

# A pedagogical introduction to quantum Monte Carlo

Michel Caffarel and Roland Assaraf

Laboratoire de Chimie Théorique, Université Pierre et Marie Curie, 4 Place Jussieu, 75252 Paris, France

(Received: )

Quantum Monte Carlo (QMC) methods are powerful stochastic approaches to calculate ground-state properties of quantum systems. They have been applied with success to a great variety of problems described by a Schrödinger-like Hamiltonian (quantum liquids and solids, spin systems, nuclear matter, *ab initio* quantum chemistry, etc...). In this paper we give a pedagogical presentation of the main ideas of QMC. We develop and exemplify the various concepts on the simplest system treatable by QMC, namely a  $2 \times 2$  matrix. First, we discuss the Pure Diffusion Monte Carlo (PDMC) method which we consider to be the most natural implementation of QMC concepts. Then, we discuss the Diffusion Monte Carlo (DMC) algorithms based on a branching (birth-death) process. Next, we present the Stochastic Reconfiguration Monte Carlo (SRMC) method which combines the advantages of both PDMC and DMC approaches. A very recently introduced optimal version of SRMC is also discussed. Finally, two methods for accelerating QMC calculations are sketched: (a) the use of integrated Poisson processes to speed up the dynamics (b) the introduction of “renormalized” or “improved” estimators to decrease the statistical fluctuations.

PACS No:

## I. INTRODUCTION

Quantum Monte Carlo methods are powerful methods to compute the ground-state properties of quantum systems. They have been applied with success to a large variety of problems described by a Schrödinger-like equation (see, *e.g.*, [1], [2], [3], [4], [5]). Although many variants can be found in the literature all the methods rest on the same idea. In essence, QMC approaches are power methods [6]. Starting from some approximation of the ground-state, some simple function of the Hamiltonian matrix (or operator for continuous spaces) is applied a large number of times on it and the convergence to the exact ground-state is searched for. In contrast with the usual power method the multiplication -matrix times vector- is not done exactly by summing explicitly all components of the current vector but estimated in average through some stochastic sampling. Such a procedure becomes particularly interesting when the linear space has a size prohibiting any attempt of multiplying or even storing matrices (say, a size much larger than  $10^9$ ). For such problems QMC is in general the most reliable and accurate method available. The price to pay is that the stochastic sampling of the product -matrix times vector- at the heart of power methods has to be done in a very efficient way. The key ingredient to realize this is to introduce some “importance sampling” techniques. This is done via a trial vector whose role is to guide the stochastic “walkers” (points of the stochastic chain built on the computer) in the “important” regions (regions corresponding to an important contribution to the averages). In addition to this, it can be shown that there exists a most important “zero-variance property”: the closer the trial vector is to the exact one, the smaller the statistical fluctuations are, and the more rapid the convergence to the exact eigenvector is. Accordingly, the efficiency of QMC methods is directly related to the quality of the trial vector used. It should be noted that powerful methods have been developed to construct and optimize trial vectors within a QMC framework [7].

A second central aspect determining the efficiency of QMC is the way the sign of the exact wave function is sampled. For distinguishable particles or bosons described by a standard Hamiltonian operator (kinetic + potential terms) it can be shown that the ground-state components (defined in the usual basis) have a constant sign (say, positive). Therefore, they can be directly interpreted as a probabilistic density and no “sign problem” arises. Highly accurate simulations of systems including thousands of bosons have been done [2] and QMC methods can be considered as the most reliable and powerful approaches for computing the ground-state properties of such systems. In contrast, for fermions the wave function has no longer a constant sign (as a consequence of the Pauli principle). We are then faced to the problem of sampling a quantity -the sign- which fluctuates a lot (regions of configuration space with different signs are highly intricate [8]) and has essentially zero-average (because of the orthogonality between the Fermi and Bose ground-states). Even worse, it

can be shown that the signed (fermionic) stochastic averages which are expressed as a difference of two positive (bosonic) quantities (one associated with the positive part of the wave function, the other with the negative one) have a statistical error which increases *exponentially* with the number of iterations of the power method [9], [10], [11] (with a constant in the exponential which also increases with the number of fermions!). This fundamental problem is known as the “sign problem”. Its solution is one of the central issues of present-day computational physics. Despite this major difficulty, to simulate fermions is not a desperate task. A first commonly employed solution consists in removing the sign from the simulations by imposing to the walkers to remain in regions of constant sign for the trial function. This approach, known as the fixed-node method, is stable (we define a set of independent sign-free bosonic simulations in each nodal domain) but is not exact due to the approximate location of the trial nodes (fixed-node bias). The quality of the fixed-node approximation is very dependent on the specific system studied and on the nature of the physics involved. In some cases the result obtained can be very satisfactory [12] (at least for the total energies). On the other hand, a number of exact schemes for treating fermions have also been developed such as transient methods, nodal release approaches, etc... In short, it is found that exact fermion simulations can be made to converge in a reasonable amount of CPU time only when the difference between the Fermi and Bose ground-state energies is not too large and/or when the starting trial wave function is good enough.

We conclude these general considerations by some remarks about the application of QMC methods to *ab initio* quantum chemistry. The calculation of electronic properties of atoms and molecules is certainly one of the most difficult applications for QMC techniques. The reasons are two-fold. First, as already emphasized above, electrons are fermions and the sign of the wave function has to be properly taken into account. However, as a number of actual simulations have shown, this problem is not dramatic for molecules (at least for the calculation of total electronic energies). By using the nodes of Hartree-Fock wave functions accurate energies can be obtained [12]. Second, and this is the most severe difficulty, the electronic density resulting from the Pauli principle (shell-structure) and the highly inhomogeneous nuclear coulombic attraction (strongly attractive nuclear centers at localized positions) displays a very intricate and subtle structure. In particular, the simulation must properly take into account the very different energy scales present in atoms (both high-energy core electrons and low-energy valence electrons). Since we are mainly interested in small differences of energies (bond energies, affinities, etc...) any error in the regions where the density of energy is large can be responsible for uncontrolled fluctuations in low-energy regions of physical or chemical interest. As a consequence, highly accurate trial wave functions are needed to get reliable results [13], [14]. Using as trial wave functions standard approximations of quantum chemistry (Hartree-Fock, small CI, etc...) and adding to them some correlated part including explicitly the interelectronic coordinate, a number of all-electron calculations of total energies have been performed for small atoms and molecules [4], [5]. Results are excellent (essentially, the exact values are obtained). The maximum number of electrons tractable is very dependent on the number of nuclei, the magnitude of the nuclear charges, the quality of the trial wave function, etc... However, for more than a dozen of electrons all-electron simulations can rapidly become very time consuming. For more electrons all-electron calculations are almost impossible because of the large statistical fluctuations resulting from the core electrons. A natural solution consists in freezing in some way the high-energy core electrons as it is routinely done in *ab initio* quantum chemistry with basis sets. Without entering into the details of the various proposals presented in the QMC literature (model potentials [15], pseudohamiltonians [16], non-local pseudopotentials [17], [18], etc...) it is important to mention that some simple and stable schemes are now at our disposal. By using pseudopotentials and the fixed-node approximation it is possible to restrict the simulations to valence electrons only in a coherent way [19], [20], [21]. This represents a real breakthrough toward a systematic use of QMC techniques in *ab initio* quantum chemistry. In order to illustrate what QMC can achieve we present in Table I a short and very arbitrary selection of results obtained for total energies and binding energies of several atoms and molecules. Of course, much more applications have been done (in particular calculations of properties other than energy, excited states, etc...) the interested readers are referred to the literature (for example, [4], [5]).

TABLE I. A short selection of QMC calculations for atoms and molecules. Total energies results correspond to all-electron fixed-node (approximate) or release-node (exact) calculations. Results for the binding energies are fixed-node calculations with pseudopotentials.

Total energies (atomic units)	$E_0$ (Hartree-Fock)	$E_0$ (QMC)	$E_0$ ("exact" <sup>a</sup> )	
He[no approx.]	-2.86168 <sup>b</sup>	-2.9037244(1) <sup>c</sup>	-2.90372437 <sup>d</sup>	
H <sub>2</sub> [no approx.]	-1.13363	-1.174(1) <sup>e</sup>	-1.17447 <sup>f</sup>	
LiH[fixed-node]	-7.987	-8.0680(6) <sup>g</sup>	-8.07021 <sup>a</sup>	
LiH[release-node]	-7.987	-8.07021(5) <sup>h</sup>	-8.07021 <sup>a</sup>	
H <sub>2</sub> O[fixed-node]	-76.0675	-76.377(7) <sup>i</sup>	-76.4376 <sup>a</sup>	
F <sub>2</sub> [fixed-node]	-198.7701	-199.487(1) <sup>j</sup>	-199.5299	

Binding energies(eV) of hydrocarbons <sup>k</sup>	HF <sup>l</sup>	LDA <sup>m</sup>	QMC	Exp.
Methane (CH <sub>4</sub> )	14.20	20.59	18.28(5)	18.19
Acetylene (C <sub>2</sub> H <sub>2</sub> )	12.70	20.49	17.53(5)	17.59
Ethane (C <sub>2</sub> H <sub>6</sub> )	23.87	35.37	31.10(5)	30.85
Benzene (C <sub>6</sub> H <sub>6</sub> )	44.44	70.01	59.2(1)	59.24

Binding energies(eV/atom) of Si clusters <sup>n</sup>	HF <sup>l</sup>	LDA <sup>m</sup>	QMC	Exp.
Si <sub>2</sub> (D <sub>2h</sub> )	0.85	1.98	1.580(7)	1.61(4)
Si <sub>3</sub> (C <sub>3v</sub> )	1.12	2.92	2.374(8)	2.45(6)
Si <sub>7</sub> (D <sub>5h</sub> )	1.91	4.14	3.43(2)	3.60(4)
Si <sub>10</sub> (C <sub>3v</sub> )	1.89	4.32	3.48(2)	...
Si <sub>20</sub> (C <sub>3v</sub> )	1.55	4.28	3.43(3)	...

<sup>a</sup> From experimental data analysis; <sup>b</sup> Ref. [22]; <sup>c</sup> Ref. [14]; <sup>d</sup> Ref. [23]; <sup>e</sup> Ref. [24]; <sup>f</sup> Ref. [25]; <sup>g</sup> Ref. [30]; <sup>h</sup> Ref. [31]; <sup>i</sup> Ref. [10]; <sup>j</sup> Ref. [32]; <sup>k</sup> From Ref. [33]; <sup>l</sup> Hartree-Fock approximation; <sup>m</sup> Local Density Approximation (Density Functional Theory); <sup>n</sup> From Ref. [34];

In this paper we attempt to give a pedagogical presentation of QMC. It is intended to newcomers in the field or any person interested in the general ideas of QMC. As we shall see, QMC is a method of very broad applicability. It can be used for studying problems defined in continuous configuration space (Schrödinger equation for particles moving in a continuum), for models described by an infinite matrix (*e.g.* problems defined on a lattice), and of course for finite matrices. In order to focus on QMC and not on problems arising from particular physical systems, we develop and exemplify the various concepts on the simplest system treatable by QMC, namely a  $2 \times 2$  matrix. We have chosen to use the standard mathematical notation employed in linear algebra instead of the most specialized notation employed in quantum physics (use of Dirac's notation in terms of "bras" and "kets").

The contents of the paper is as follows. In Sec. 2.1. we introduce our toy problem (matrix  $2 \times 2$ ). In Sec. 2.2 the central idea of QMC approaches is presented: In essence, QMC can be viewed as a stochastic implementation of the power method. Section 2.3 is devoted to the presentation of the fundamental relation between the matrix to be diagonalized and a stochastic matrix. The simplest expression for the energy resulting from this connection is then given. Sec. 2.4 presents the Pure Diffusion Monte Carlo (PDMC) method which we consider to be the most natural implementation of QMC concepts. Then, in Sec. 2.5 we discuss the Diffusion Monte Carlo (DMC) algorithms based on a branching (birth-death) process. Next, we present in Sec. 2.6 the Stochastic Reconfiguration Monte Carlo (SRMC) method which combines the advantages of both PDMC and DMC approaches. An optimal version of SRMC very recently introduced is also discussed. In Sec. 2.7 two methods for accelerating QMC calculations are sketched: (a) the use of integrated Poisson processes to speed up the dynamics (Sec. 2.7.1) (b) the introduction of "renormalized" or "improved" estimators to decrease the statistical fluctuations (Sec. 2.7.2). Finally, in Sec. 3 we present a number of aspects related to the implementation of QMC to an arbitrary problem (QMC for a  $N \times N$  matrix: Sec. 3.1; QMC in a continuum: Sec. 3.2).

## II. QMC FOR A $2 \times 2$ MATRIX

### A. The $2 \times 2$ matrix

The simplest system treatable by QMC is a  $2 \times 2$  matrix. Let us denote by  $H$  this matrix. It is written under the form

$$H = \begin{pmatrix} v_0 & -t \\ -t & v_1 \end{pmatrix} \quad (1)$$

where, without loss of generality,  $v_0$  is taken smaller or equal to  $v_1$  (the labels for the two basis vectors can be chosen arbitrarily) and  $t$  is chosen positive (choice of the overall sign for the matrix). The lowest eigenvalue (ground-state energy) is given by

$$E_0 = \frac{v_0 + v_1 - \sqrt{(v_1 - v_0)^2 + 4t^2}}{2} \quad (2)$$

and the corresponding ground-state eigenvector is

$$\mathbf{u}_0 = \begin{pmatrix} \frac{t}{\sqrt{(E_0 - v_0)^2 + t^2}} \\ \frac{v_0 - E_0}{\sqrt{(E_0 - v_0)^2 + t^2}} \end{pmatrix} \quad (3)$$

Note that  $E_0 \leq v_0$  and that both components of  $\mathbf{u}_0$  are positive.

### B. QMC is a power method

The central idea of QMC is to extract from a known trial vector  $\mathbf{v}$  its exact ground-state component,  $\mathbf{u}_0$ . This is realized by using a matrix  $G(H)$  acting as a filter:

$$\lim_{n \rightarrow \infty} G(H)^n \mathbf{v} \sim \mathbf{u}_0 \quad (4)$$

where  $H$  is the original matrix. This idea is similar to what is done in power methods where the maximal eigenvector (associated with the eigenvalue of maximum modulus) is obtained by applying a large number of times the matrix on an arbitrary initial vector [6]. A most popular choice for  $G(H)$  is

$$G(H) \equiv 1 - \tau(H - E_T) \quad (5)$$

where “1” denotes the identity matrix ( $[1]_{ij} = \delta_{ij}$ ) and  $\tau$  is some positive constant (a “time-step” in physical applications) to be adjusted (see below) and  $E_T$  is some arbitrary reference energy.

In our toy example, the initial trial vector  $\mathbf{v}$  can be decomposed within the basis defined by the eigenvectors of  $H$ :

$$\mathbf{v} = c_0 \mathbf{u}_0 + c_1 \mathbf{u}_1. \quad (6)$$

The action of  $G(H)$  at iteration number  $n$  defined as follows

$$\mathbf{v}^{(n)} \equiv G(H)^n \mathbf{v} = [1 - \tau(H - E_T)]^n \mathbf{v} \quad (7)$$

can therefore be expressed as

$$\mathbf{v}^{(n)} = c_0 [1 - \tau(E_0 - E_T)]^n \mathbf{u}_0 + c_1 [1 - \tau(E_1 - E_T)]^n \mathbf{u}_1. \quad (8)$$

Provided that  $c_0$  is non-zero, that is, there exists some overlap between the trial vector  $\mathbf{v}$  and the exact solution, the vector  $\mathbf{v}^{(n)}$  will converge geometrically to the exact ground-state eigenvector (up to some inessential normalization factor). The correction at finite  $n$  is essentially exponential in  $n$  and in the gap in energy  $E_1 - E_0$

$$\mathbf{v}^{(n)} \sim_{n \rightarrow \infty} \mathbf{u}_0 + O[\exp(-n\tau(E_1 - E_0))]. \quad (9)$$

Note that this result is independent on the value of  $E_T$  if  $\tau$  is chosen positive. In practice, the exact eigenvalue can be extracted by considering the large- $n$  limit of the following ratio, known as the Rayleigh quotient

$$E_0(n) = \frac{\mathbf{v}^T H \mathbf{v}^{(n)}}{\mathbf{v}^T \mathbf{v}^{(n)}} \sim_{n \rightarrow \infty} E_0 \quad (10)$$

Finally, the problem reduces to compute the action of the matrix  $H$  over some arbitrary vector,  $\mathbf{v}^{(n)}$ . When the size of the linear space is small this can be easily done by a simple matrix multiplication. Here, we have

$$\begin{aligned} c_0^{(n+1)} &= [1 - \tau(v_0 - E_T)]c_0^{(n)} + t\tau c_1^{(n)} \\ c_1^{(n+1)} &= t\tau c_0^{(n)} + [1 - \tau(v_1 - E_T)]c_1^{(n)} \end{aligned} \quad (11)$$

where  $c_i^{(k)}$  are the two components of the vector  $\mathbf{v}^{(k)}$ .

In Table II the convergence of  $c_0^{(n)}$ ,  $c_1^{(n)}$ , and  $E_0(n)$  are shown. The parameters chosen are the following:  $v_0 = 1, v_1 = 2, t = 1, \tau = 0.1, E_T = 3$ , and  $c_0^{(0)} = c_1^{(0)} = 1/\sqrt{2}$ . As it is seen the convergence to the exact values is quite rapid as a function of the iteration number  $n$ .

TABLE II. Convergence of  $c_0^{(n)}$ ,  $c_1^{(n)}$ , and  $E_0(n)$  as a function of  $n$ .

$n$	$c_0^{(n)}$	$c_1^{(n)}$	$E_0^{(n)}$
$n = 10$	0.83253...	0.55397...	0.403295...
$n = 20$	0.84813...	0.52978...	0.385023...
$n = 30$	0.85029...	0.52630...	0.382401...
$n = 40$	0.85060...	0.52581...	0.382027...
$n = 50$	0.85064...	0.52574...	0.381974...
$n = 60$	0.85064...	0.52573...	0.381967...

$n = 70$	0.85065...	0.52573...	0.381966...
$n = 80$	0.85065...	0.52573...	0.381966...
Exact values	0.85065... <sup>a</sup>	0.52573... <sup>a</sup>	0.381966... <sup>b</sup>

<sup>a</sup> See equation (3).

<sup>b</sup> See equation (2).

Note that the convergence as a function of the iteration number  $n$  of the power method can be greatly enhanced by using all information present in the set of iterated vectors,  $\mathbf{v}^{(p)}$ ,  $p = 0 \dots n$ . This idea is implemented by introducing a so-called Krylov space [26], [27] associated with the pair  $(H, \mathbf{v})$  and defined as the linear space spanned by the vectors  $\mathbf{v}^{(p)}$ ,  $p = 0 \dots n$ . By diagonalizing the matrix  $H$  within this subspace the optimal solution at iteration  $n$  is obtained under the form  $\mathbf{u} = \sum_{i=0}^n c_i \mathbf{v}^{(i)}$  instead of  $\mathbf{u} = \mathbf{v}^{(n)}$ , and the convergence as a function of  $n$  is much more rapid. Many variations of this idea have been presented in the literature for treating “big” matrices of size ranging from  $10^3$  to  $10^9$ . Among them two well-known variants are the Lanczòs [26] and Davidson [28] algorithms. For matrices of much larger size, stochastic approaches are to be used.

### C. Introducing a stochastic matrix

In order to perform the product -matrix times vector- stochastically we shall rewrite the matrix elements  $[1 - \tau(H - E_T)]_{ij}$  in terms of a so-called stochastic matrix  $P_{i \rightarrow j}(\tau)$  and some residual contribution  $w_{ij}$  (“weight” matrix). As we shall see below,  $P_{i \rightarrow j}(\tau)$  will be interpreted as the probability of moving from state  $i$  to state  $j$  and will be used to construct a random sequence of states. To be valid this probabilistic interpretation requires the two following conditions. First, to define a probability, the matrix elements must all be positive:

$$P_{i \rightarrow j}(\tau) \geq 0 \quad . \quad (12)$$

Second, the sum over final states must be equal to one:

$$\sum_j P_{i \rightarrow j}(\tau) = 1 \quad . \quad (13)$$

An arbitrary matrix is said to be stochastic when conditions (12) and (13) are fulfilled. Here, we define  $P_{i \rightarrow j}(\tau)$  as follows

$$P_{i \rightarrow j}(\tau) = \frac{v_j}{v_i} [1 - \tau(H - E_L)]_{ij}. \quad (14)$$

where  $E_L$  is a diagonal matrix written as

$$[E_L]_{ij} = \delta_{ij} E_L(i) \quad (15)$$

and the vector  $E_L(i)$  -called the local energy- is given by

$$E_L(i) = \frac{[H\mathbf{v}]_i}{v_i}. \quad (16)$$

Here  $\tau$  is a parameter which will play the role of a time step. Note that the magnitude of the fluctuations of the local energy is a measure of the quality of the trial vector used. The closer the trial vector is to the exact one, the smaller the fluctuations are. In the optimal case corresponding to  $\mathbf{v} = \mathbf{u}_0$ , the local energy becomes constant (independent on the state  $i$ ) and equal to the exact energy.

Now, using the definition of the stochastic matrix our fundamental quantity  $[1 - \tau(H - E_T)]_{ij}$  can be rewritten as

$$[1 - \tau(H - E_T)]_{ij} = \frac{v_i}{v_j} P_{i \rightarrow j}(\tau) w_{ij} \quad (17)$$

where the weight  $w_{ij}$  is defined as

$$w_{ij} = \frac{[1 - \tau(H - E_T)]_{ij}}{[1 - \tau(H - E_L)]_{ij}}. \quad (18)$$

This quantity represents the residual contribution resulting from the fact that the trial vector is not exact. When  $\mathbf{v} = \mathbf{u}_0$ , all weights reduce to a constant (which can be made equal to one by taking  $E_T = E_0$ ).

In our simple example the stochastic matrix writes

$$\begin{pmatrix} P_{0 \rightarrow 0}(\tau) & P_{1 \rightarrow 0}(\tau) \\ P_{0 \rightarrow 1}(\tau) & P_{1 \rightarrow 1}(\tau) \end{pmatrix} = \begin{pmatrix} 1 - \tau t \frac{v_1}{v_0} & \tau t \frac{v_0}{v_1} \\ \tau t \frac{v_1}{v_0} & 1 - \tau t \frac{v_0}{v_1} \end{pmatrix} \quad (19)$$

Since the quantities  $t, \tau$ , and  $v_i$  are positive the diagonal elements can be made positive by choosing the time-step  $\tau$  small enough:

$$\tau \leq \text{Min}\left[\frac{v_0}{v_1 t}, \frac{v_1}{v_0 t}\right] \quad (20)$$

Off-diagonal elements being also positive, condition (12) is fulfilled. We also check from (19) that  $P_{0 \rightarrow 0}(\tau) + P_{0 \rightarrow 1}(\tau) = 1$  (starting from 0 the probability of arriving at 0 or 1 is equal to one) and  $P_{1 \rightarrow 0}(\tau) + P_{1 \rightarrow 1}(\tau) = 1$  (*idem* when starting from 1), so that condition (13) is also valid. The general case is not difficult to verify. Condition (12) implies the following constraint on  $\tau$

$$0 \leq \tau \leq \frac{1}{\text{Max}_i[H_{ii} - E_L(i)]} \quad (21)$$

and condition (13) results from (14) and (16).

Let us now write the quantities of interest in terms of the stochastic matrix. Here, we shall focus our attention on the Rayleigh quotient (10) and, more fundamentally, on the expression of the  $n$ -th order iterated vector  $\mathbf{v}^{(n)}$

$$\mathbf{v}^{(n)} = [1 - \tau(H - E_T)]^n \mathbf{v} \quad (22)$$

Writing explicitly the product of matrices present in (22), the components of the vector  $\mathbf{v}^{(n)}$  can be written as

$$v_{i_n}^{(n)} = \sum_{i_{n-1}} \dots \sum_{i_0} [1 - \tau(H - E_T)]_{i_n i_{n-1}} \dots [1 - \tau(H - E_T)]_{i_1 i_0} v_{i_0} \quad (23)$$

Introducing the weight matrix (18) we obtain

$$v_{i_n}^{(n)} = \sum_{i_{n-1}} \dots \sum_{i_0} w_{i_n i_{n-1}} [1 - \tau(H - E_L)]_{i_n i_{n-1}} \dots w_{i_1 i_0} [1 - \tau(H - E_L)]_{i_1 i_0} v_{i_0} \quad (24)$$

and, finally (after permuting the indices and using the definition (14) of the stochastic matrix)

$$v_{i_n}^{(n)} = \sum_{i_0} \dots \sum_{i_{n-1}} v_{i_0}^2 P_{i_0 \rightarrow i_1} \dots P_{i_{n-1} \rightarrow i_n} w_{i_0 i_1} \dots w_{i_{n-1} i_n} \frac{1}{v_{i_n}}. \quad (25)$$

Now, by defining the following quantity:

$$P(i_0, \dots, i_n) = v_{i_0}^2 P_{i_0 \rightarrow i_1} \dots P_{i_{n-1} \rightarrow i_n} \quad (26)$$

we can rewrite the energy at iteration  $n$  as follows:

$$E_0(n) = \frac{\sum_{i_0} \dots \sum_{i_n} P(i_0, \dots, i_n) E_L(i_n) \prod_{k=0}^{n-1} w_{i_k i_{k+1}}}{\sum_{i_0} \dots \sum_{i_n} P(i_0, \dots, i_n) \prod_{k=0}^{n-1} w_{i_k i_{k+1}}} \quad (27)$$

Up to that point, it is most important to realize that no probabilistic aspect has been introduced so far. In other words, (27) can be viewed as a simple algebraic rewriting of (10). When we are able to compute exactly the sums involved in (27) both equations give the same answer. When this is possible we are in the realm of standard power methods and their improved versions (Lanczòs, Davidson,...). In that case we can refer to such approaches as “deterministic” power methods. If the sums are too large (for a linear space of size  $N$  there are  $(N+1)^n$  terms in the sums and an iterative computation requires about  $nN^2$  multiplications), some sort of “stochastic” power method (=QMC) is needed.



The essence of QMC is to reinterpret the quantity  $P(i_0, \dots, i_n)$  as the  $(n+1)$ -time probability distribution of a (Markov) chain defined by the transition probability distribution  $P_{i \rightarrow j}$ , (14). Once this is done, the various sums in equation (27) can be reinterpreted and computed as stochastic averages over the successive states drawn according to  $P_{i \rightarrow j}$  (construction of a “stochastic trajectory”). In practice, the sequence of states is easily built on a computer as follows. For a given state  $i$  we first determine the set of states  $\{j_1, \dots, j_k, \dots, j_K\}$  such that  $\mathcal{P}_{j_k} \neq 0$  and the probability  $\mathcal{P}_{j_k} = P_{i \rightarrow j_k}(\tau)$  for each state  $j_k$  is computed. A uniform random number  $\xi$  on the interval  $(0,1)$  is drawn and the state is chosen among this set according to the repartition function

$$\begin{aligned}
& \text{State } j_1 \text{ chosen if } \xi \in (0, \mathcal{P}_{j_1}) \\
& \text{State } j_2 \text{ chosen if } \xi \in (\mathcal{P}_{j_1}, \mathcal{P}_{j_1} + \mathcal{P}_{j_2}) \\
& \dots \\
& \text{State } j_k \text{ chosen if } \xi \in \left( \sum_{l=0, k-1} \mathcal{P}_{j_l}, \sum_{l=0, k} \mathcal{P}_{j_l} \right) \\
& \dots
\end{aligned} \tag{28}$$

This step can be iterated, thus giving rise to a chain of states, i.e, to a stochastic trajectory  $i_0, i_1, \dots$ . Finally, the estimate of the energy can be expressed as an average along the stochastic trajectory  $(i_0, i_1, i_2, \dots)$  as follows

$$E_0 = \frac{E_L(i_0) + E_L(i_1)w_{i_0 i_1} + E_L(i_2)w_{i_0 i_1}w_{i_1 i_2} + E_L(i_3)w_{i_0 i_1}w_{i_1 i_2}w_{i_2 i_3} + \dots}{1 + w_{i_0 i_1} + w_{i_0 i_1}w_{i_1 i_2} + w_{i_0 i_1}w_{i_1 i_2}w_{i_2 i_3} + \dots} \tag{29}$$

Note that passing from (27) to (29) is possible because the stochastic process used is recurrent or ergodic (see, *e.g.*, Ref. [29]). As an important consequence, the quantity calculated in (29) does not depend neither on the particular stochastic trajectory used nor on the initial condition  $i_0$ . In other words, any stochastic trajectory of infinite length realizes the various transition probability distributions defined by the stochastic matrix.

Another point which deserves to be emphasized is that the stationary density of the stochastic process  $\pi_i$  is given by the squared components of the trial vector

$$\pi_i = v_i^2 \tag{30}$$

The stationary density has a simple meaning: this is the distribution in configuration space which remains unchanged after application of the stochastic matrix. In practice, this is the average distribution obtained on the computer when the equilibrium regime is reached. Mathematically, this condition is written as

$$\sum_i \pi_i P_{i \rightarrow j}(\tau) = \pi_j. \tag{31}$$

This relation is a simple consequence of the definition of the stochastic matrix (14).

In practice, it is found that a direct implementation of formula (29) is very inefficient. As the simulation time increases, no convergence of the energy is observed. The reason for that is the rapid increase of variance of the products of weights  $\prod w_{i_i i_{i+1}}$  as a function of the number of iterations (number of weights). This important aspect is discussed in the next section.

#### D. Pure Diffusion Monte Carlo (PDMC)

In order to control the convergence of the energy as a function of the simulation time, the two limits  $n \rightarrow \infty$  (number of iterations of the power method) and  $P \rightarrow \infty$  (number of Monte Carlo steps for a given trajectory) must be treated *separately*. In other words,  $E_0(n)$  is computed

by collecting from the trajectory all contributions in (29) associated with  $n$  products of weights. Explicitly, it gives

$$\begin{aligned}
E_0(0) &= \frac{E_L(i_0) + E_L(i_1) + E_L(i_2) + \dots}{1 + 1 + 1 + \dots} \\
E_0(1) &= \frac{E_L(i_1)w_{i_0i_1} + E_L(i_2)w_{i_1i_2} + E_L(i_3)w_{i_2i_3} + \dots}{w_{i_0i_1} + w_{i_1i_2} + w_{i_2i_3} \dots} \\
E_0(2) &= \frac{E_L(i_2)w_{i_0i_1}w_{i_1i_2} + E_L(i_3)w_{i_1i_2}w_{i_2i_3} + E_L(i_4)w_{i_2i_3}w_{i_3i_4} + \dots}{w_{i_0i_1}w_{i_1i_2} + w_{i_1i_2}w_{i_2i_3} + w_{i_2i_3}w_{i_3i_4} \dots} \\
&\dots
\end{aligned} \tag{32}$$

In practice it is found that, for not too large values of  $n$  (limited number of weights in the product  $\prod w_{i_i i_{i+1}}$ ),  $E_0(n)$  can be obtained with a good accuracy.

The method based on (32) is known in the literature as the Pure Diffusion Monte Carlo (PDMC) approach [29], [24]. Although intrinsically unstable at large  $n$ , PDMC has proven to be very useful as illustrated by a number of applications (*e.g.* [24], [35], [36], [37], [38], [39]). This is the case when the trial vector is accurate enough to allow the convergence of the various estimators before the large fluctuations at large iteration numbers arise.

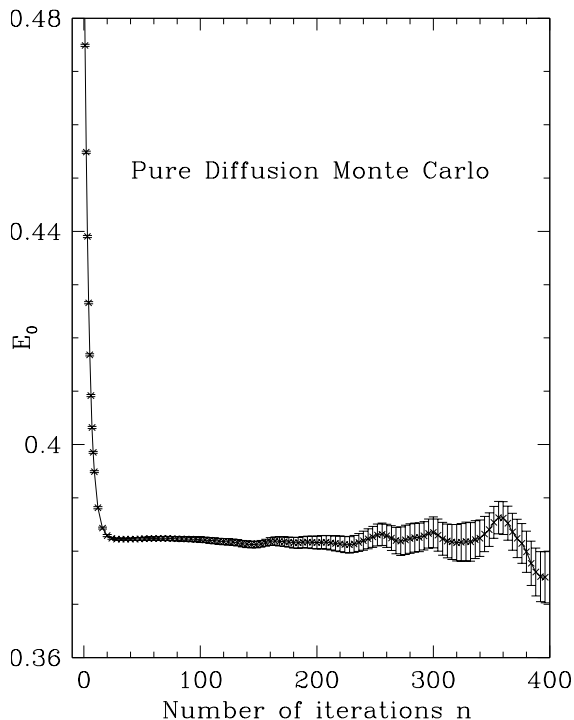


FIG. 1. Pure Diffusion Monte Carlo calculation.

Fig.1 presents for the toy model the convergence of  $E_0(n)$  as a function of  $n$ . The parameters of the matrix  $H$  are identical to those used in the previous application and the trial vector chosen is  $\mathbf{v} = (1/\sqrt{2}, 1/\sqrt{2})$ . In order to get an estimate of the error made for a finite  $P$  at a given  $n$ , a number of independent calculations of  $E_0(n)$  are made and the statistical errors are calculated using standard statistical techniques.

At  $n = 0$  the Rayleigh quotient is given by

$$E_0(0) = \frac{\mathbf{v}^T H \mathbf{v}}{\mathbf{v}^T \mathbf{v}}. \quad (33)$$

This quantity is also known as the ‘‘variational’’ energy associated with the trial vector. Here, the exact value for this quantity is 0.5 and we get 0.5001(2). When the iteration number  $n$  is increased the estimate of the energy converges to the exact value [number of iterations of about 40,  $E_0(40) = 0.3822(2)$ ]. At larger values of  $n$  ( $n > 100$ ) the difficulty associated with the problem of the many-weight product becomes severe and no stabilization is observed.

Note that in a very recent work a detailed mathematical analysis of the behavior of the PDMC algorithm as a function of the iteration number has been presented [40]. It has been shown in a rigorous way that the ‘‘effective’’ probability associated with a given state  $i$  as it is built in a PDMC scheme, namely

$$P_i = \sum_{P=0}^{\infty} \sum_{i_0} \dots \sum_{i_{P-1}} \pi_{i_0} P_{i_0 \rightarrow i_1} \dots P_{i_{P-1} \rightarrow i} w_{i_0 i_1} \dots w_{i_{P-1} i} \quad (34)$$

and considered as a random variable, does not converge as a function of  $P$ .

### E. Diffusion Monte Carlo (DMC)

A most commonly employed method to escape from the difficulty associated with the weights at large  $n$  is to remove them from the averages and incorporate them in the stochastic process itself. This idea is realized by introducing a so-called birth-death or branching process. In practice, it consists in adding to the standard stochastic move of PDMC (28) a new step in which the current configuration is destroyed or copied a number of times proportional to the local weight. Denoting  $m$  the number of copies (multiplicity), we can take

$$m \equiv \text{int}(w_{ij} + \eta) \quad (35)$$

where  $\text{int}(x)$  denotes the integer part of  $x$ , and  $\eta$  a uniform random number on  $(0, 1)$ . The use of (35) is justified because it gives the correct weight in average

$$\int_0^1 d\eta (w_{ij} + \eta) = w_{ij} \quad (36)$$

Note that, in contrast with PDMC which could be defined with one single walker, to simulate a branching process requires the use of a large enough population of walkers (in practice, a few thousands). Mathematically, adding a branching process can be viewed as sampling with a generalized transition probability  $P_{i \rightarrow j}^*(\tau)$  defined as

$$P_{i \rightarrow j}^*(\tau) \equiv P_{i \rightarrow j}(\tau) w_{ij} = \frac{v_j}{v_i} [1 - \tau(H - E_T)]_{ij} \quad (37)$$

Of course, the normalization is no longer constant,  $\sum_i P_{i \rightarrow j}^* \neq 1$  and is a function of  $j$ , and therefore  $P^*$  is not a genuine transition probability. However, we can still define a stationary density for it. From (37) it is easily shown that the stationary condition is obtained when  $E_T$  is chosen to be the exact energy  $E_0$ , and that the stationary density is now  $v_i u_{0i}$

$$\sum_i \pi_i^* P_{i \rightarrow j}^* = \pi_j^* \quad (38)$$

with

$$\pi_i^* = v_i u_{0i} \quad \text{and} \quad E_T = E_0 \quad (39)$$

By using a stabilized population of configurations the exact energy may be now obtained as

$$E_0 = \langle\langle E_L \rangle\rangle \equiv \lim_{P \rightarrow \infty} \frac{1}{P} \sum_{l=0}^{P-1} E_L(i_l) \quad (40)$$

where a population of states (walkers) is evolved according to (28) and branched according to (35). Such a Monte Carlo algorithm is known as the Diffusion Monte Carlo (DMC) method. Now, since in DMC the number of walkers can fluctuate, some sort of population control is required. Indeed, nothing prevents the total population from exploding or collapsing entirely. Various solutions to this problem have been proposed. Two common approaches consist either in performing from time to time a random deletion/duplication step or in varying slowly the reference energy  $E_T$  to keep the average number of walkers approximately constant. In both cases, a finite bias is introduced by the population control step. In order to minimize this undesirable source of error it is important to control the size of population as rarely as possible and in the most gentle way [1]. Here, to illustrate this idea, we do the population control by adjusting the reference energy to the fluctuations of population by using a formula of the type

$$E_T(t + \tau) = E_T(t) + K/\tau \ln[M(t)/M(t + \tau)] \quad (41)$$

where  $M(t)$  is the total number of walkers at time  $t$  and  $K$  is some positive constant. When  $M(t + \tau) > M(t)$  the reference energy is reduced and more walkers are killed at the next step. In the opposite case  $E_T$  is raised and more walkers are duplicated.

In Table III the DMC energy as a function of the size  $M$  of the population is shown. The bias for small population is clearly seen. For  $M$  sufficiently large the bias becomes smaller than the statistical fluctuations.

TABLE III. Convergence of the energy in a DMC calculation as a function of the size  $M$  of the population for a fixed number of Monte Carlo steps. The total number of MC steps is about  $2.10^7$

Size $M$	$E_0(M)$
$M = 20$	0.3832(3.2)
$M = 40$	0.3829(3.2)
$M = 60$	0.3827(3.5)
$M = 80$	0.3822(3.2)
$M = 100$	0.3820(3.5)
$M = 200$	0.3820(3.4)
Exact value <sup>a</sup>	0.381966...

<sup>a</sup> See equation (2).

## F. Stochastic reconfiguration Monte Carlo (SRMC)

Very recently, following an idea initially introduced by Hetherington [41], the use of stochastic reconfiguration processes has been reconsidered in Diffusion Monte Carlo [42], [43], [44], [40]. The motivation is to combine the best of both worlds: efficiency of DMC (most of the weights are treated in the stochastic process and the algorithm is stable) and absence of bias as in PDMC (no population control step). The approach is derived within a PDMC framework (the walkers “carry” some weight) but the population is “reconfigured” using some specific rules. The reconfiguration is done in a such way that the number of walkers is kept constant at each step and therefore no population control is required. The idea consists in carrying many walkers simultaneously and introducing a global weight associated with the entire population instead of a local weight for each walker. The global weight  $W$  of the population at iteration (or “time”)  $l$  is defined as the average of the  $M$  local weights  $w_{i_l i_{l+1}}^{(k)}$

$$W \equiv \frac{1}{M} \sum_{k=1}^M w_{i_l i_{l+1}}^{(k)} \quad (42)$$

where  $M$  is the number of walkers considered. By increasing the number  $M$  of walkers the fluctuations of the global weight is reduced and the gain in computational efficiency can be very important. Now, the method consists in defining a PDMC scheme in the enlarged linear space defined by the tensorial product ( $M$  times) of the initial linear space. In this new space the full stochastic matrix is defined as the tensorial product of individual stochastic matrices. Note that no correlation between the stochastic moves of different walkers is introduced at this level. Second, and this is the important point, each individual weight is rewritten as a function of the global weight

$$w_{i_l i_{l+1}}^{(k)} = \tilde{w}_{i_l i_{l+1}}^{(k)} W$$

with

$$\tilde{w}_{i_l i_{l+1}}^{(k)} \equiv \frac{w_{i_l i_{l+1}}^{(k)}}{W} \quad (43)$$

This re-writing allows to introduce the global weight as a weight common to all walkers and thus to define a standard PDMC scheme in the tensorial product of spaces. To take into account the new weight  $\tilde{w}$  a so-called reconfiguration process is introduced. At each step the total population of  $M$  walkers is “reconfigured” by drawing  $M$  walkers independently with probability proportional to  $\tilde{w}$ . Note that, at this point, some correlation between different walkers is introduced since the duplication/deletion process for a walker depends on the global weight  $W$ .

In Fig.2 we present for our toy problem the calculation of the energy with the stochastic reconfiguration Monte Carlo method. A population of  $M = 20$  walkers has been used. In sharp contrast with PDMC (see Fig.1) no problem at large values of  $n$  is observed. The converged value for the energy is 0.3819(6.4) is in excellent agreement with the exact value.

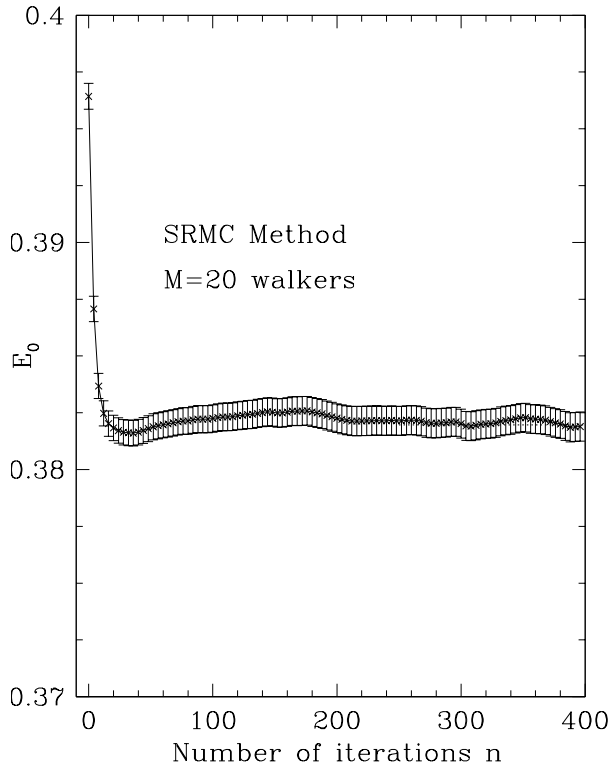


FIG. 2. Stochastic Reconfiguration Monte Carlo calculation.

In order to improve further the algorithm it is possible to define a so-called optimal Stochastic Reconfiguration Monte Carlo method [40]. For that let us first discuss the two important limits of the algorithm, namely the case of an infinite number of walkers,  $M \rightarrow \infty$  and the case of constant weights,  $w \rightarrow 1$ . When  $M \rightarrow \infty$  the global weight (42) converges to its stationary exact value. As a consequence, the different weights  $\tilde{w}$  associated with each walker (as given by (43)) become independent from each other and the reconfiguration process reduces to the usual branching process (35) without population control and systematic bias since the population is infinite. However, in the limit  $w \rightarrow 1$  the method does not reduce to the standard DMC approach. Indeed, the reconfiguration step “reconfigures” the entire population whatever the values of the weights. In order to improve the efficiency of the SRMC method this undesirable source of fluctuations must be reduced. To do that we divide the population of walkers into two different sets. A first set of walkers corresponds to all walkers verifying  $\tilde{w} \geq 1$ . These walkers can be potentially duplicated and will be called “positive” walkers. The other walkers verify  $0 \leq \tilde{w} < 1$ , they can be potentially destroyed and will be called “negative walkers”. The number of reconfigurations is defined as

$$N_{\text{Reconf}} = \sum_{k+} |\tilde{w}_{i_i i_{i+1}}^{(k)} - 1| = \sum_{k-} |\tilde{w}_{i_i i_{i+1}}^{(k)} - 1| \quad (44)$$

where  $\sum_{k+}$  (resp.  $\sum_{k-}$ ) indicates that the summation is done over the set of positive (resp. negative) walkers. The equality in (44) is a simple consequence of the definition of positive and negative walkers. Note that in general  $N_{\text{Reconf}}$  is not integer. An integer number of reconfigurations can be obtained by considering  $\text{int}(N_{\text{Reconf}} + \eta)$  where  $\eta$  is a uniform random number on the interval (0,1). Once the number of reconfigurations has been drawn, a set of  $N_{\text{Reconf}}$  walkers is duplicated and another set of  $N_{\text{Reconf}}$  walkers is destroyed from the current population. This is done by drawing separately  $N_{\text{Reconf}}$  walkers among the lists of positive and negative walkers. It is easily verified that by doing this no source of systematic error has been introduced as in the original reconfiguration process of Hetherington. However, in contrast with the latter the average number of reconfigurations is kept minimal and, consequently, the efficiency of the simulation is significantly enhanced. In addition, the average number of reconfigurations vanishes as the weights

become constant. In other words, the reconfiguration method reduces in this limit to the standard DMC method with  $w = 1$ .

TABLE IV. Convergence of  $E_0(n = 0)$  as a function of the size  $M$  of the population. The values for the standard Stochastic Reconfiguration Monte Carlo (SRMC) method and the optimal one are compared. The closer  $E_0(n = 0)$  is to the exact result  $E_0$  the closer the corresponding SRMC stationary density is to the exact DMC density, (39). The total number of MC steps is about  $2 \cdot 10^7$

Size $M$	$E_0(n = 0)$ SRMC method	$E_0(n = 0)$ Optimal SRMC
$M = 10$	0.4091(5.7)	0.3940(4.7)
$M = 20$	0.3964(5.6)	0.3872(4.5)
$M = 100$	0.3848(5.7)	0.3830(4.1)
$M = 200$	0.3835(5.9)	0.3824(4.3)
$M = 400$	0.3829(5.7)	0.3822(4.4)
Exact value $E_0^a$	0.381966...	0.381966...

<sup>a</sup> See equation (2).

In Table IV we present the values obtained for the energy at zero time (without any correction due to the global weight), for the SRMC method and its optimal version. This quantity is particularly interesting since it is an indicator of the quality of the stationary density obtained. The closer the SRMC stationary density is to the DMC density (39), the smaller the difference between  $E_0(n=0)$  and  $E_0(n=\infty) = E_0$  is. As it should be, both methods give an energy which converges to the exact value as the number of walkers goes to infinity. However, the finite bias at finite  $M$  is found to be much smaller for the optimal version. This result is a direct consequence of the fact that fluctuations due to the reconfiguration process are smaller in the optimal SRMC.

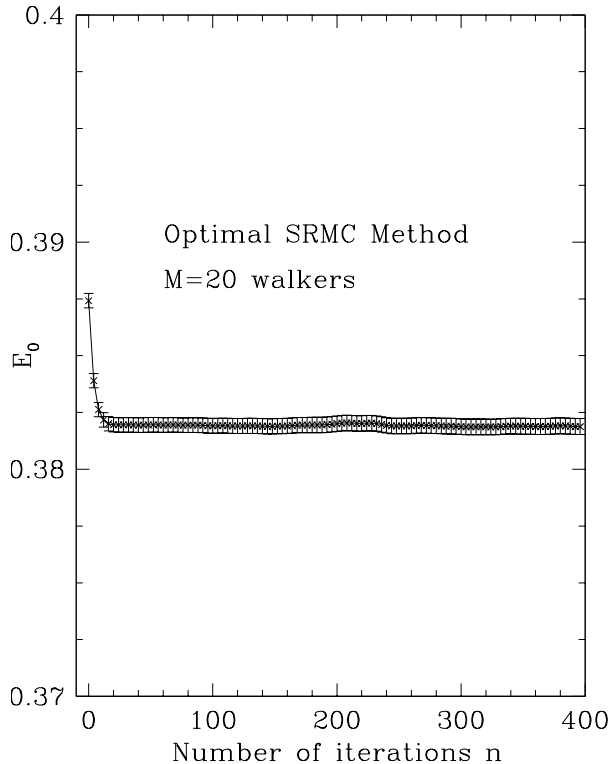


FIG. 3. Optimal Stochastic Reconfiguration Monte Carlo calculation.

In Fig.3 the convergence of the energy as a function of the iteration number is shown. The curve is similar to what has been obtained previously with the standard SRMC method. However, for the same number of Monte Carlo steps the accuracy obtained is better. The converged value obtained for the energy is 0.3819(3.5).

### G. Accelerating QMC

QMC calculations can be computationally demanding. Starting from a few thousands walkers, several thousands of Monte Carlo steps are performed for each walker. When the system studied is complex, this can indeed require an important numerical effort. In order to treat large systems and/or to get a better accuracy it is therefore desirable to accelerate as much as possible the convergence of Monte Carlo simulations. By “accelerating” it is meant to decrease significantly the statistical fluctuations without changing much the amount of calculations. To realize this, it is natural to imagine two main directions. A first direction consists in improving in some way the dynamics of the stochastic process. For example, configurations can be generated so that they are better distributed in configuration space or more independent. A second direction consists in decreasing as much as possible the fluctuations of the various estimators to be averaged. In what follows, we present two very recently proposed methods illustrating both aspects. Note that these approaches are of general interest (valid for almost any type of QMC methods).



1. Use of integrated Poisson processes

In our toy example a stochastic trajectory can be represented as a sequence of states 0 and 1 such as:

$$(0, 0, 0, 1, 0, 1, 1, 0, 0, 1, \dots)$$

During a certain amount of “time” (or iterations) the system is blocked in a given state  $i$  (0 or 1) and then at some time escapes to a different state (here,  $0 \rightarrow 1$  or  $1 \rightarrow 0$ ). It is therefore possible to define the probability that the system remains in a given configuration  $i$  a number of times equal to  $n$ . It is given by

$$\begin{aligned} \mathcal{P}_i(n) \equiv P(i_1 = i, \tau; \dots; i_{n-1} = i, \tau; i_n \neq i, \tau) = \\ [P_{i \rightarrow i}(\tau)]^n [1 - P_{i \rightarrow i}(\tau)] \end{aligned} \quad (45)$$

$\mathcal{P}_i(n)$  defines a normalized discrete Poisson distribution

$$\mathcal{P}_i(n) = [1 - P_{i \rightarrow i}(\tau)] \exp[n \ln(P_{i \rightarrow i}(\tau))] \quad (46)$$

where the integer  $n$  runs from zero to infinity. It is quite easy to compute the average number of times the system is “trapped” in a given state  $i$ :

$$\bar{n}_i = \sum_{n=0}^{\infty} n \mathcal{P}_i(n) = \frac{P_{i \rightarrow i}(\tau)}{1 - P_{i \rightarrow i}(\tau)} \quad (47)$$

Now, to describe transitions towards states  $j$  different from  $i$  we introduce the following “escape” transition probability

$$\tilde{P}_{i \rightarrow j} = \frac{P_{i \rightarrow j}(\tau)}{1 - P_{i \rightarrow i}(\tau)} \quad j \neq i. \quad (48)$$

Using (14)  $\tilde{P}_{i \rightarrow j}$  is rewritten in the most explicit form

$$\tilde{P}_{i \rightarrow j} = \frac{H_{ij} v_j}{\sum_{k \neq i} H_{ik} v_k} \quad j \neq i. \quad (49)$$

Once this “integrated” stochastic process is introduced the elementary stochastic trajectory is replaced by a trajectory between *different* states and to each state is associated an average “trapping time” given by

$$\bar{\theta}_i = \bar{n}_i \tau \quad (50)$$

All expressions presented previously can be re-expressed in terms of the “escape” transition probability and average trapping times only [45]. To give an example, within a DMC framework the energy estimate can be rewritten as:

$$E_0 = \frac{\langle\langle \bar{\theta}_i E_L(i) \rangle\rangle}{\langle\langle \bar{\theta}_i \rangle\rangle} \quad (51)$$

where the average  $\langle\langle \dots \rangle\rangle$  is defined from a population of walkers evolving according to the escape transition probability and where the branching contribution associated with the transition from state  $i$  to state  $j$  is given now by the following “integrated” weight

$$\tilde{w}_i = \sum_{n=0}^{+\infty} \mathcal{P}_i(n) w_{ii}^n \quad (52)$$

This formula represents the average contribution of the weight  $w_{ii}$  when the system is blocked in state  $i$ .

Integrating out the entire time evolution of the system when blocked in a given state can improve a lot the convergence of QMC. This is particularly true when the system studied displays very different average “trapping” times (for example, when deep potential wells are present). Let us now illustrate this idea with some calculations. First, note that the application of the method to our  $2 \times 2$  matrix is not representative of the general case. Indeed, for the  $2 \times 2$  matrix, formula (51) degenerates to the ratio of two deterministic quantities (since the series of states is imposed:  $\dots 0 \rightarrow 1 \rightarrow 0 \rightarrow 1 \rightarrow 0 \dots$ ). To test the method we need to consider a problem with at least three states. We have chosen a simple generalization of our toy model, namely

$$H = \begin{pmatrix} v_0 & -t & -t \\ -t & v_1 & -t \\ -t & -t & v_2 \end{pmatrix} \quad (53)$$

We take  $t = 1, v_0 = 1$  and  $v_1 = 2$ . Here also, we consider a trial vector whose components are constant,  $\mathbf{v} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$ . In Table V we present some calculations done with the standard DMC method and the improved version presented in this section. We consider the case of an intermediate value for  $v_2$  ( $v_2=3$ ) and the case of a larger value ( $v_2=10$ ). Of course, the same number of Monte Carlo steps are performed in each case. When  $v_2=3$  a clear improvement is observed. A factor of about 4 in statistical error has been gained. For the larger value, we get a factor of about 11. This important improvement is due to the fact that when  $v_2$  is large the exact eigenvector is strongly localized on components 0 and 1 and therefore the on-site time integration is particularly effective.

TABLE V. Integrating out the dynamics,  $t = 1, v_0 = 1, v_1 = 2$ .

Value of the potential $v_2^a$	$v_2 = 3$	$v_2 = 10$
$E_0(\text{standard})^b$	-0.21416(19)	0.18772(111)
$E_0(\text{integrated})^c$	-0.21430(4.4)	0.18805(10)
$E_0(\text{exact})$	-0.2143197...	0.1880341...

<sup>a</sup> Defined by (53).

<sup>b</sup> Formula (40).

<sup>c</sup> Formula (51).

An alternative way of improving QMC methods consists in decreasing as much as possible the fluctuations of the quantity to be averaged. Since the ideas we shall present are in fact extremely general (valid for any type of Monte Carlo algorithm) let us work with very general notations. Let us call  $O$  an arbitrary observable defined in some discrete configuration space and  $\pi$  the normalized probability distribution of the Monte Carlo algorithm, we want to calculate the following average

$$\langle O \rangle = \sum_i \pi_i O_i \quad (54)$$

The magnitude of the statistical fluctuations associated with the use of a finite sample of configurations drawn according to  $\pi$

$$\langle O \rangle = \frac{1}{P} \sum_{i=1}^P O_i \quad (55)$$

is directly related to the variance of the observable

$$\sigma(O)^2 \equiv \langle O^2 \rangle - \langle O \rangle^2. \quad (56)$$

In order to accelerate the simulations we can use a so-called “improved” or “renormalized” estimator, that is, a new observable  $\tilde{O}$  having the same average as the original one

$$\langle O \rangle = \langle \tilde{O} \rangle \quad (57)$$

but a lower variance:

$$\sigma(\tilde{O}) < \sigma(O). \quad (58)$$

Various improved estimators have been introduced in the literature (See, e.g., [46], [47]). The basic idea is to construct new estimators by integrating out some intermediate degrees of freedom and, therefore, removing the corresponding source of fluctuations. Very recently, we have proposed a general and systematic approach to construct such estimators [45]. The method is valid for any type of observable and any type of Monte Carlo algorithm. It is based on a zero-variance principle. The ideas are very simple and can be summarized as follows. We introduce a so-called trial matrix  $\tilde{H}$  and a trial vector  $\mathbf{u}$ . The basic requirements are that the trial matrix must be self-adjoint and must admit  $\sqrt{\pi}$  as eigenstate with zero eigenvalue:

$$\sum_j \tilde{H}_{ij} \sqrt{\pi_j} = 0 \quad (59)$$

In the general case constructing a matrix verifying such conditions is not difficult. The “renormalized” or improved estimator is then defined as follows:

$$\tilde{O}_i = O_i + \frac{\sum_j \tilde{H}_{ij} u_j}{\sqrt{\pi_i}}. \quad (60)$$

It is quite easy to verify that both estimators have the same average. It is also easy to realize that the variance of  $\tilde{O}$  will be minimal and equal to zero if the following fundamental equation is fulfilled:

$$\sum_j \tilde{H}_{ij} u_j = -(O_i - \langle O \rangle) \sqrt{\pi_i} \quad (61)$$

This equation can be obtained by imposing the renormalized observable to be constant and equal to the exact average. Now, a very important point is that, whatever the choice of the trial matrix and trial vector, it is always possible to construct an improved estimator having a lower variance than the bare observable  $O$ . The proof is as follows. Let us introduce  $\alpha$  a multiplicative factor for

the trial vector  $\mathbf{u}$  and let us minimize the variance of the renormalized estimator with respect to this parameter. We write:

$$\sigma(\alpha)^2 = \langle \tilde{O}^2 \rangle - \langle \tilde{O} \rangle^2 = \sigma(\alpha = 0)^2 + 2\alpha \langle OHu/\sqrt{\pi} \rangle + \alpha^2 \langle (Hu/\sqrt{\pi})^2 \rangle \quad (62)$$

Minimizing the variance with respect to  $\alpha$  we get

$$\alpha_{opt} = - \langle OHu/\sqrt{\pi} \rangle / \langle (Hu/\sqrt{\pi})^2 \rangle \quad (63)$$

and

$$\sigma(\alpha_{opt})^2 = \sigma(\alpha = 0)^2 - \langle OHu/\sqrt{\pi} \rangle^2 / \langle (Hu/\sqrt{\pi})^2 \rangle \quad (64)$$

The correction to  $\sigma(\alpha = 0)^2$  being negative, we see that from *any* trial vector we can get a better estimator just by optimizing the multiplicative factor. Applications of the zero-variance property to a number of classical and quantum Monte Carlo simulations have shown that very important gains in computational effort can be obtained [48].

### III. QMC BEYOND THE $2 \times 2$ MATRIX

All ideas and results presented in the preceding sections can be extended to a general matrix or a continuous operator. In this section we discuss a number of aspects related to the general case. For the sake of clarity we shall discuss separately the cases of a discrete or a continuous configuration space.

#### A. QMC for a $N \times N$ matrix

For a finite matrix of arbitrary dimension the various formulas presented above can be readily used (most of the formulas have already been written for an arbitrary size  $N$  of the linear space). For our  $2 \times 2$  matrix we have restricted ourselves to the case of a negative off-diagonal element,  $H_{12} = -t < 0$ . It was justified since it corresponds to choose arbitrarily the overall sign of the matrix. In the general case ( $N > 2$ ), off-diagonal elements  $H_{ij}$  can now have a positive or negative sign. This fact has to be taken into account when defining the stochastic matrix. Let us recall that the value of the stochastic matrix between two different states is given by (see, (14))

$$P_{i \rightarrow j}(\tau) = -\frac{v_j}{v_i} \tau H_{ij} \quad i \neq j \quad (65)$$

To allow a probabilistic interpretation this quantity needs to be positive. As explained in the introduction this is true for a number of quantum systems describing bosons or distinguishable particles. For fermions, both  $H_{ij}$  and the trial vector components have a non-constant sign. As already discussed a number of practical solutions have been proposed to solve this famous “sign problem”. We shall not discuss further this point and we refer the interested reader to the abundant literature on this subject.

Now, it is worth emphasizing that the quantity which controls the efficiency of QMC is the trial vector used,  $\mathbf{v}$ . When  $\mathbf{v}$  is close to the exact eigenvector, the sampling of the configuration space is done efficiently and the statistical fluctuations on the energy can be very small. When the trial vector is good enough, the size of the linear space treatable can be almost as large as desired (virtually infinite). Many QMC applications on lattice models have illustrated this property (see, *e.g.* [3]). To give a concrete example, in a recent study of a metal-insulator phase transition in a generalized one-dimensional Hubbard model [45], the size of the largest matrix to be diagonalized was  $N \sim 1.2 \cdot 10^{28}$  (32 particles moving on a 32 sites-ring). For this matrix and using some physically meaningful trial vectors the values obtained for the ground-state energy were:  $E_0 = -52.13056(15)$  in the better case (small electron-electron repulsion) and  $E_0 = -23.7118(13)$  in the worst case (large electron-electron repulsion).

## B. QMC in a continuum

For continuous configuration spaces all ideas are identical but their implementation can appear in a quite different way. Since the Schrödinger Hamiltonian has a simple form in the  $x$ -representation, QMC is developed within this representation. The fundamental step of the power method is written as

$$\psi^{(n+1)}(\mathbf{x}) = \int d\mathbf{x}' [G(H)]_{\mathbf{x}\mathbf{x}'} \psi^{(n)}(\mathbf{x}') \quad (66)$$

where  $[G(H)]_{\mathbf{x}\mathbf{x}'}$  denotes the kernel of the operator  $G(H)$  (matrix elements in a continuous space),  $\mathbf{x}$  represents a vector in the  $d$ -dimensional space  $\mathbf{x} = (x_1, \dots, x_d)$  and  $\psi^{(n)}(\mathbf{x})$  is the  $n$ -th order iterated trial function. The vast majority of QMC applications have been concerned with a Hamiltonian operator of the form

$$H = -\frac{1}{2} \sum_{i=1}^d \frac{\partial^2}{\partial x_i^2} + V(x), \quad (67)$$

that is, the standard quantum-mechanical Hamiltonian used to describe a set of interacting particles. The first part in (67) represents the kinetic energy and  $V(x)$  is the potential function. For such an operator the kernel of  $H$  is highly singular (derivatives of Dirac distributions). Therefore, in contrast with what is done for finite linear spaces, the operator  $G(H) = 1 - \tau(H - E_T)$  cannot be used and some regular kernels must be introduced. The most common choice is

$$G(\mathcal{H}) = e^{-\tau(\mathcal{H} - E_T)}. \quad (68)$$

For continuous systems the QMC approach based on (68) is in general referred to as Diffusion Monte Carlo (note that several denominations exist in the literature, we give here what we think to be the most current one). Another possible choice is

$$G(\mathcal{H}) = \frac{1}{1 + \tau(\mathcal{H} - E_T)}. \quad (69)$$

The main advantage of this second choice is to define a QMC method free of the so-called short-time approximation (see below). The methods based on (69) are usually referred to as Green's function Monte Carlo [1], [10].

In the same way as in the discrete case we can define a stochastic kernel (matrix) or transition probability distribution as follows:

$$P(\mathbf{x} \rightarrow \mathbf{y}, \tau) = \frac{\psi_T(\mathbf{y})}{\psi_T(\mathbf{x})} [e^{-\tau(H - E_L)}]_{\mathbf{x}\mathbf{y}} \quad (70)$$

where  $\psi_T(\mathbf{x})$  is some trial function and  $E_L$  is the local energy function

$$E_L(\mathbf{x}) = H\psi_T(\mathbf{x})/\psi_T(\mathbf{x}). \quad (71)$$

This definition is the direct generalization of (14) and (16). Note that  $P(\mathbf{x} \rightarrow \mathbf{y}, \tau) \geq 0$  and that  $\int d\mathbf{y} P(\mathbf{x} \rightarrow \mathbf{y}, \tau) = 1$ . In order to sample  $P(\mathbf{x} \rightarrow \mathbf{y}, \tau)$  we need to interpret it as a probability distribution with respect to  $\mathbf{y}$  for a fixed value of  $\mathbf{x}$ . This can be done in a natural way since  $P(\mathbf{x} \rightarrow \mathbf{y}, \tau)$  obeys the following Fokker-Planck equation

$$\frac{\partial P}{\partial \tau} = LP \quad (72)$$

where  $L$  is a Fokker-Planck operator

$$L = \frac{1}{2} \nabla^2 \cdot - \nabla[\mathbf{b} \cdot] \quad (73)$$

and  $\mathbf{b}$  the so-called drift vector given by

$$\mathbf{b} = \frac{\nabla\psi_{\mathbf{T}}}{\psi_{\mathbf{T}}}. \quad (74)$$

The derivation of this result can be found in Ref. [29]. For a presentation of the Fokker-Planck equation and material related to diffusion processes, the reader is referred to [49]. In a continuum  $P(\mathbf{x} \rightarrow \mathbf{y}, \tau)$  is therefore interpreted as the transition probability density of a standard drifted continuous diffusion process in configuration space. Stochastic trajectories can be obtained by using the Langevin equation associated with the diffusion process [49]

$$d\mathbf{x}(t) = \mathbf{b}[\mathbf{x}(t)]dt + d\mathbf{W}(t) \quad (75)$$

where  $\mathbf{W}(t)$  is the standard multi-dimensional Wiener process. In practice, this equation is discretized under the form

$$\mathbf{x}(t + \tau) = \mathbf{x}(t) + \mathbf{b}[\mathbf{x}(t)]\tau + \sqrt{\tau}\eta \quad (76)$$

where  $\eta$  is a vector whose components are independent gaussian random variables verifying  $\langle \eta_i \rangle = 0$  and  $\langle \eta_i \eta_j \rangle = \delta_{ij}$ . This latter equation is particularly simple to implement on a computer. This is the continuous counterpart of (28) used in the discrete case. Note that the discretization in time introduces a bias (short-time approximation). To obtain exact quantities several simulations with different time-steps are performed and the results are then extrapolated to a zero time-step. Note that this source of error does not exist in the discrete case.

Finally, the weights can be defined via the fundamental relation between the operator  $G(H)$  and the stochastic kernel. We can write

$$\frac{\psi_{\mathbf{T}}(\mathbf{y})}{\psi_{\mathbf{T}}(\mathbf{x})} [e^{-\tau(H-E_T)}]_{\mathbf{xy}} = \frac{\psi_{\mathbf{T}}(\mathbf{y})}{\psi_{\mathbf{T}}(\mathbf{x})} [e^{-\tau[(H-E_L)+(E_L-E_T)]}]_{\mathbf{xy}}. \quad (77)$$

In the short-time approximation (or ‘‘Trotter break up’’, see [50]) this equation can be rewritten as

$$\frac{\psi_{\mathbf{T}}(\mathbf{y})}{\psi_{\mathbf{T}}(\mathbf{x})} [e^{-\tau(H-E_T)}]_{\mathbf{xy}} \sim_{\tau \rightarrow 0} P(\mathbf{x} \rightarrow \mathbf{y}, \tau) e^{-\tau[E_L(\mathbf{x})-E_T]} \quad (78)$$

and the weight appears now as a simple multiplicative operator given by

$$w(\mathbf{x}) = e^{-\tau[E_L(\mathbf{x})-E_T]}. \quad (79)$$

Using expression (70) for the transition probability distribution [stochastically realized via (76)] and expression (79) for the weight, all ideas presented in Sec. II can be implemented.

## Acknowledgments

We would like to acknowledge many discussions with Anatole Khelif (Paris 7) and some useful comments on the manuscript from Peter Reinhardt (Paris 6). This work was supported by the ‘‘Centre National de la Recherche Scientifique’’ (CNRS). Part of the work has been done by R. Assaraf at SISSA under EU sponsorship, TMR Network FULPROP.

- [1] D.M. Ceperley and M.H. Kalos, in *Monte Carlo Method in Statistical Physics*, edited by K.Binder (Springer-Verlag, Heidelberg, 1992).
- [2] D.M. Ceperley, *Rev. Mod. Phys.* **67**, 279 (1995).
- [3] W. von der Linden, *Phys. Rep.* **220**, 53 (1992).

- [4] B.L. Hammond, W.A. Lester, Jr., and P.J. Reynolds in *Monte Carlo Methods in Ab Initio Quantum Chemistry*, World Scientific Lecture and course notes in chemistry Vol.1 (1994).
- [5] J.B. Anderson in *Reviews in Computational Chemistry*, Vol.13, edited by K.B. Lipkowitz and D.B. Boyd (Wiley, 1999).
- [6] A.R. Gourlay and G.A. Watson, *Computational Methods for Matrix Eigenproblems*, Chichester: Wiley.
- [7] C. J. Umrigar, K. G. Wilson, and J. W. Wilkins, Phys. Rev. Lett. **60**, 1719 (1988).
- [8] D.M. Ceperley, J. Stat. Phys. **63**, 1237 (1991).
- [9] K.E. Schmidt and M.H. Kalos in *Monte Carlo methods in Statistical Physics II*, edited by K. Binder, Springer, (1984).
- [10] D.M. Ceperley and B.J. Alder, J. Chem. Phys. **64**, 5833 (1984).
- [11] J.B. Anderson in *Understanding Chemical Reactivity*, S.R. Langhoff, Ed. Kluwer (1995).
- [12] J.B. Anderson, Int. Rev. Phys. Chem. **14**, 95 (1995).
- [13] K.E Schmidt and J.W. Moskowitz, J. Chem. Phys. **93**, 4172 (1990).
- [14] M.P. Nightingale and C.J. Umrigar in *Recent Advances in Computational Chemistry* Vol.2, edited by W.A. Lester Jr., World Scientific (1997).
- [15] T. Yoshida and K. Igushi, J. Chem. Phys. **88**, 1032 (1988).
- [16] G.B. Bachelet, D.M. Ceperley, and M.G.B. Chiochetti, Phys. Rev. Lett **62**, 2088 (1989).
- [17] G.B. Bachelet, D.M. Ceperley, M. Chiochetti, and L. Mitas, "Atomic pseudohamiltonians for quantum Monte Carlo", Dordrecht, Kluwer (1989)
- [18] B.L. Hammond, P.J. Reynolds, and W.A. Lester, Jr., Phys. Rev. Lett **61**, 2312 (1988).
- [19] B.L. Hammond, P.J. Reynolds, and W.A. Lester, Jr., J. Chem. Phys. **87**, 1130 (1987).
- [20] L. Mitas, E.L. Shirley, and D.M. Ceperley, J. Chem. Phys. **95**, 3467 (1991).
- [21] M.M. Hurley and P.A. Christiansen, J. Chem. Phys. **86**, 1069 (1987).
- [22] E. Clementi and C. Roetti, *Atomic Data and Nuclear Data Tables* **14**, 177(1974)
- [23] K. Frankowski and C.L. Pekeris, Phys. Rev. **146**, 46 (1966).
- [24] M. Caffarel and P. Claverie, J. Chem. Phys. **88** 1100 (1988).
- [25] W. Kolos and L. Wolniewicz, J. Chem. Phys. **43**, 2429 (1965).
- [26] B.N. Parlett, *The symmetric Eigenvalue Problem*, Prentice-Hall Series in Computational Mathematics (1980).
- [27] M. Caffarel, F.X. Gadea, and D.M. Ceperley, Europhys. Lett. **16**, 249 (1991).
- [28] E.R. Davidson, J. Comput. Phys. **17**, 87 (1975)
- [29] M. Caffarel and P. Claverie, J. Chem. Phys. **88** 1088 (1988).
- [30] M. Caffarel and D.M. Ceperley, J. Chem. Phys. **97** 8415 (1992)
- [31] B. Chen and J.B. Anderson, J. Chem. Phys. **102**, 4491 (1995).
- [32] C. Filippi and C.J. Umrigar, J. Chem. Phys. **105**, 213 (1996).
- [33] J.C. Grossman, L. Mitas, and K. Raghavachari, Phys. Rev. Lett. **75**, 3870 (1995).
- [34] J.C. Grossman, and L. Mitas, Phys. Rev. Lett. **74**, 1323 (1995).
- [35] K.J. Runge, Phys. Rev. B **45**, 7229 (1992).
- [36] D.M. Ceperley and B. Bernu, J. Chem. Phys. **89** 6316 (1988).
- [37] B. Bernu, D.M. Ceperley, and W.A. Lester Jr., J. Chem. Phys. **93** 552 (1990).
- [38] M. Caffarel, M. Rérat, and C. Pouchan, Phys. Rev. A **47**, 3704 (1993).
- [39] F. Schautz and H.J. Flad, J. Chem. Phys. **110** 11700 (1999).
- [40] R. Assaraf, M. Caffarel, and A. Khelif, "Diffusion Monte Carlo with a fixed number of walkers", Phys. Rev. E (April 2000)
- [41] J.H. Hetherington Phys. Rev. A **30**, 2713 (1984).
- [42] S. Sorella, Phys. Rev. Lett. **80**, 4558 (1998).
- [43] M. Calandra Buonauro and S. Sorella, Phys. Rev. B **57**, 11446 (1998).
- [44] S. Sorella and L. Capriotti, Phys. Rev. B **61**, 2599 (2000).
- [45] R. Assaraf, P. Azaria, M. Caffarel, and P. Lecheminant, Phys. Rev. B **60**, 2299 (1999).
- [46] M. Sweeny, Phys. Rev. B **27**, 4445 (1983); G. Parisi, R. Petronzio, and F. Rapuano, Phys. Lett. B **128**, 418 (1983); U. Wolff, Nucl. Phys. B **334**, 581 (1990)
- [47] B. Ammon, H.G. Evertz, N. Kawashima, M. Troyer, and B. Frischmuth, Phys. Rev. B **58**, 4304 (1998).
- [48] R. Assaraf and M. Caffarel, Phys. Rev. Lett. **83**, 4682 (1999)
- [49] C.W. Gardiner, *Handbook of Stochastic Methods* (Springer, Berlin, 1983).
- [50] H.F. Trotter, Proc. Am. Math. Soc. **10**, 545 (1959); M. Suzuki, Commun. Math. Phys. **51**, 183 (1976).