Parallel Monte Carlo simulations using a residence weight algorithm

M. Athènes*
CEA Saclay, Service de Recherches de Métallurgie Physique, 91191 Gif-sur-Yvette, France
(Received 23 October 2001; revised manuscript received 13 March 2002; published 3 July 2002)

A parallel Monte Carlo (MC) algorithm, the residence weight algorithm, is proposed. This algorithm is an extension to continuous ensembles of the residence time algorithm. The proposed algorithm consists of generating several trial MC moves in parallel, selecting one of them using appropriate relative probabilities and correcting the non-Boltzmannian sampling procedure by means of appropriate configuration weights. The corresponding parallel Boltzmannian scheme is based on a configurational-bias Monte Carlo scheme and will also be considered. The efficiency of both parallel algorithms has been compared in a case study: the slow relaxation dynamics in a model silicate. For a small number of trial moves generated in parallel, we observe that the residence weight algorithm performs less efficiently than its corresponding configurational-bias scheme by a factor of about 2. However, when the number of parallel trial moves increases and becomes larger than ten, we observe that the residence weight algorithm and the corresponding configurational-bias scheme perform with equivalent efficiencies.

DOI: 10.1103/PhysRevE.66.016701 PACS number(s): 02.70.Tt, 05.10.−a, 02.50.−r

I. INTRODUCTION

Equilibrium properties of a given statistical ensemble can be accurately estimated by performing Monte Carlo ensemble averages: a chain of configuration in the phase space is constructed so as to sample the desired statistical ensemble. To construct this configuration chain, we must repeatedly generate a new configuration from the current one using a stochastic process defined by some appropriate evolution rules that are to be specified. These evolution rules must obey two principles, ergodicity and detailed balance, so as to insure that the appearance probability of any generated configuration is compatible with the equilibrium statistics of the ensemble. Ergodicity states that any two configurations of the phase space can be connected with a configuration chain. Detailed balance that is a sufficient but not a necessary condition states that the probability transition fluxes between two configurations $\mathcal{C}_i$ and $\mathcal{C}_f$, the initial and final configurations, respectively, are equal. This latter principle results from both the microreversibility and incompressibility of the equations of mechanics.

For many systems of interest, and particularly for slowly relaxing amorphous structures, the need for saving computing time is a permanent request. An extensively exploited solution consists in increasing the sampling efficiency by means of ''clever'' Monte Carlo moves preferentially generated towards low energy configurations. Although the essential motivation for developing these elaborated Monte Carlo algorithms is aimed at increasing the sampling efficiency, some of these algorithms appear to be particularly well suited to parallel implementation since they offer a large portion of similar and independent tasks that can be easily distributed on distinct processors. This issue was noticed and discussed in a previous study by Esselink, Loyens, and Smit [1] who detailed the parallelization of force bias [2–5], and configurational-bias [6] Monte Carlo methods.

The scope of this paper will concern the second approach whose principle can be summarized as follows: let us consider that the basic moves to parallelize consist in random displacements applied to randomly selected particles. If the procedure allowing the construction of the new configuration $\mathcal{C}_f$ from the initial one exhibits symmetric probabilities, a possible acceptance probability [7] for these moves is $c(i \rightarrow f) = \min(1, \exp -\beta \Delta E)$, where ΔE is the variation of the internal energy. This implies that transitions with small displacement amplitudes will have a small variation ΔE of the internal energy and will therefore be accepted with a high probability, but will lead to a weak sampling efficiency. At variance, transitions with large displacement amplitudes will be hardly accepted due to a very low acceptance rate, but result in a high sampling efficiency. Since the internal energy evaluation of trial configurations represents the largest amount of the computer time, it appears beneficial to generate high amplitudes displacements in parallel and to select one with an adequate procedure [the CBMC (configurational-bias Monte Carlo) scheme] so as to increase the mean acceptance rate and thus the computational efficiency.

All the mentioned Monte Carlo methods are said to be Boltzmannian sampling schemes: consider a given configuration chain has been generated; the appearance probability of any configuration in the chain converges towards the Boltzmann weight when the length of the chain increases to infinity. There also exists non-Boltzmannian schemes for which configuration weights must be included in the sampling procedure so as to correct for the fact that the chain configurations are not distributed according to the equilibrium statistics. Although there exists numerous non-Boltzmannian Monte Carlo sampling schemes, the parallel implementation of such algorithms belonging to this second class has never been considered (to my knowledge) because existing schemes did not present large portions of independent tasks easy to parallelize.

In the present paper, we focus on the parallel implementation of a non-Boltzmannian method. For this purpose, a scheme adapted for parallelization has been developed and called the residence weight algorithm (RWA). The paper is

*Electronic address: Manuel.Athenes@cea.fr

organized as follows: the essential features of the Boltzmannian and non-Boltzmannian sampling techniques are first briefly recalled; in the following section, the residence weight algorithm is described and a proof is given that a weighted detailed balance condition is obeyed; a comparative study between the parallel residence weight algorithm and the equivalent parallel configurational-bias scheme is carried out on a case study.

II. MONTE CARLO SAMPLING SCHEMES

The two ways to perform Monte Carlo sampling, exposed previously, are recalled. Note that this simple classification is adapted for most Monte Carlo schemes currently used in statistical physics, however, from a mathematical point of view, more complete classifications have also been proposed [8].

(i) Boltzmann sampling is obtained by introducing the statistical bias associated to the Monte Carlo move into an acceptance probability [7], which is itself derived from the detailed balance equation

$$\mathcal{P}_i b(i\rightarrow f)c(i\rightarrow f)=\mathcal{P}_f b(f\rightarrow i)c(f\rightarrow i), \tag{1}$$

where $\mathcal{P}_i$ (resp. $\mathcal{P}_f$) is the Boltzmann weights of the initial configuration (resp. the final configuration), and $b(i\rightarrow f)$ [resp. $b(f\rightarrow i)$] is the probability to generate the trial configuration $\mathcal{C}_f$ (resp. $\mathcal{C}_i$) starting from configuration $\mathcal{C}_i$ (resp. $\mathcal{C}_f$). The acceptance probabilities for the forwards and backwards trial transition are $c(i\rightarrow f)$ and $c(f\rightarrow i)$, respectively. The acceptance probability proposed by Metropolis [7]

$$c(i\rightarrow f)=\min\left(1,\frac{\mathcal{P}_f b(f\rightarrow i)}{\mathcal{P}_i b(i\rightarrow f)}\right) \tag{2}$$

is compatible with Eq. (1) and implies that one must compute the *a priori* probability $b(f\rightarrow i)$ of selecting the old configuration from the new one and the statistical bias corresponds to the ratio $b(f\rightarrow i)/b(i\rightarrow f)$.

(ii) Non-Boltzmannian sampling: the statistical bias is accounted for in the ensemble average so as to correct for the imposed transitions. The weighted detailed balance condition for this non-Boltzmannian sampling scheme requires

$$\frac{\mathcal{P}_n}{\tau_n}\alpha(n\rightarrow n+1)=\frac{\mathcal{P}_{n+1}}{\tau_{n+1}}\alpha(n+1\rightarrow n), \tag{3}$$

where τ_n and τ_{n+1} are the correcting weights to be included in the following biased averaging scheme that allow the evaluation of a physical quantity $\mathcal{A}$,

$$\langle\mathcal{A}\rangle_\tau=\frac{\sum_{i=1}^{\mathcal{N}}\tau_i\mathcal{A}(i)}{\sum_{i=1}^{\mathcal{N}}\tau_i}. \tag{4}$$

An essential additional condition that is to be obeyed concerns the reversibility of the scheme itself. If this point is not guaranteed, the correcting weights τ_n obtained when the configuration chain is constructed in the forwards or backwards direction would not be identical, which would imply that Eq. (3) would not be unambiguously defined.

Class (i) techniques includes, among many other schemes, the force-bias [2–5] Monte Carlo schemes (FBMC) and CBMC [6], respectively for which the parallelization was discussed previously (the parallel force-bias algorithm is called the hybrid molecular dynamics/Monte Carlo method). Class (ii) techniques comprises a wide range of algorithms. The most naive such algorithm consists in performing random sampling: a random configuration $\mathcal{C}_{n+1}$ is generated independently from $\mathcal{C}_n$ and is attributed a correcting weight $\tau_{n+1}\propto\mathcal{P}_{n+1}$, which is compatible with Eq. (3). More efficient class (ii) algorithms have however been developed for discrete systems: the residence time algorithm [9,10], detailed in Appendix A, considers all the accessible configurations from $\mathcal{C}_n$, selects one of them, $\mathcal{C}_{n+1}$, using an appropriate probability function $\alpha(n\rightarrow n+1)$, computes the correcting weight (residence time) compatible with Eq. (3) and uses it either to bias the configuration weight or to increment the time variable, depending on whether equilibrium or kinetic features are investigated.

In the random sampling scheme, the transition probability from $\mathcal{C}_n$ to $\mathcal{C}_{n+1}$ does not depend on $\mathcal{C}_n$ while in the following example, the transition probability from $\mathcal{C}_n$ to $\mathcal{C}_{n+1}$ does depend on $\mathcal{C}_n$. There also exists non-Boltzmannian algorithms for which the transition probability from $\mathcal{C}_n$ to $\mathcal{C}_{n+1}$ depends on both $\mathcal{C}_n$ and $\mathcal{C}_{n-1}$. In such a case, the algorithm is said to be non-Markovian. This feature is present in some techniques aimed at calculating free energy differences between two canonical ensembles. This is for instance the case for the Rosenbluth sampling method [11,12], which is based on the particle test insertion method [13]. This point is discussed in a companion paper [14]. In the following section, a non-Boltzmannian algorithm that can be easily parallelized is proposed. This algorithm appears to be the non-Markovian extension of the residence time algorithm (described in appendix A) to continuous systems.

III. RESIDENCE WEIGHT ALGORITHM

Let us consider that a chain of configuration $\mathcal{C}_n$ with weight τ_n has been generated using the desired Monte Carlo algorithm, where the number n is an increasing configuration index. The compatibility of the weighted chain with the equilibrium statistic is insured by imposing the equality between the forwards and backwards weighted probability fluxes between any two configurations of the chain. The microreversibility imposes that if the whole configuration chain is reversed then configuration $\mathcal{C}_n$ should always be a possible trial configuration from configuration $\mathcal{C}_{n+1}$ in the selection scheme of the algorithm, even though the probability of obtaining $\mathcal{C}_n$ from $\mathcal{C}_{n+1}$ within a finite set of trial generation is ''practically'' zero in a continuous ensemble during a finite Monte Carlo simulation.

The RWA is therefore defined as follows: $Z-1$ trial transitions are randomly generated, an additional Zth trial transition pointing backwards to the previous configuration is imposed; then a single transition, selected among these Z possible trial transitions, is finally implemented. The intro-

duced modifications imply that the Monte Carlo process possesses a memory that has to be taken into account in the probability fluxes. Since the convergence to equilibrium is insured by the equality of the forwards and backwards probability fluxes, the following weighted detailed balance condition should be obeyed:

$$\frac{1}{\tau_n}\mathcal{P}_n\alpha(n\to n+1|n-1)=\frac{1}{\tau_{n+1}}\mathcal{P}_{n+1}\alpha(n+1\to n|n+2), \tag{5}$$

where $\alpha(l\to m|k)$ is the conditional probability to transit from $\mathcal{C}_l$ to $\mathcal{C}_m$ knowing that the system was in configuration $\mathcal{C}_k$ previously. From configuration $\mathcal{C}_n$, $\alpha(n\to n+1|n-1)$ accounts for the fact that the backwards transition towards $\mathcal{C}_{n-1}$ is imposed and that $Z-1$ trial moves have to be generated prior to selecting one among them. From configuration $\mathcal{C}_{n+1}$ in Eq. (5), the forwards and imposed backwards configurations are $\mathcal{C}_n$ and $\mathcal{C}_{n+2}$ since the chain is traveled backwards.

The ''residence weights'' τ_n and τ_{n+1} correspond to the correction to introduce in the sampling scheme so as to recover the equilibrium statistics. They can be interpreted as follows: if a weight of one is attributed to configuration $\mathcal{C}_n$, then the selected configuration $\mathcal{C}_{n+1}$ must be given a weight of τ_{n+1}/τ_n so as to correct for having imposed the transition. It then follows by induction that Eq. (4) leads to a correct sampling scheme as shown below. Let us consider any two configurations $\mathcal{C}_p$ and $\mathcal{C}_q$ in the weighted chain. The transition flux from $\mathcal{C}_p$ to $\mathcal{C}_q$ is the product of the probability of generating and selecting the trial probabilities for each of the $q-p$ steps

$$K(p\to q)=\frac{\mathcal{P}_p}{\tau_p}\prod_{n=p}^{q-1}\alpha(n\to n+1|n-1), \tag{6}$$

while the probability flux for generating the reverse macromove is

$$K(q\to p)=\frac{\mathcal{P}_q}{\tau_q}\prod_{n=q}^{p-1}\alpha(n\to n-1|n+1). \tag{7}$$

Substituting Eq. (3) into Eq. (7) simplifies the residence weight ratio to

$$\frac{K(p\to q)}{K(q\to p)}=\frac{\tau_q\mathcal{P}_p}{\tau_p\mathcal{P}_q}\prod_{n=p}^{q-1}\frac{\tau_n\mathcal{P}_{n+1}}{\tau_{n+1}\mathcal{P}_n}=1. \tag{8}$$

Let us note that when an averaging procedure is initiated, the weight of the first configuration is not specified since the previous configuration does not exist. This introduced bias can be safely neglected since in practice a large number of independent configurations is actually used.

One now considers a possible residence weight compatible with the weighted detailed balance equation:

$$\tau_n=\alpha(n\to n+1|n-1)[\mathcal{G}(\mathcal{P}_n/\mathcal{P}_{n+1})]^{-1}, \tag{9}$$

where the acceptance function $\mathcal{G}$, which will be specified by the reference moves that one wishes to parallelize, possesses the following feature:

$$\mathcal{G}(z)=z^{-1}\mathcal{G}(z^{-1}). \tag{10}$$

Since the sampling scheme [Eq. (4)] does not depend on whether the configuration chain is traveled backwards or forwards, the scheme itself must be reversible, and the residence weights [Eq. (9)] should not depend on the chain direction. Considering the fact that the weighted detailed balance condition also applies between configurations $\mathcal{C}_{n-1}$ and $\mathcal{C}_n$, the residence weight invariance means that Eq. (9) must remain invariant if $n+1$ and $n-1$ are permuted. It implies the following relationship between the probabilities $\alpha(n\to n+1|n-1)$ and $\alpha(n\to n-1|n+1)$:

$$\frac{\alpha(n\to n+1|n-1)}{\alpha(n\to n-1|n+1)}=\frac{\mathcal{G}(\mathcal{P}_n/\mathcal{P}_{n-1})}{\mathcal{G}(\mathcal{P}_n/\mathcal{P}_{n+1})}. \tag{11}$$

This condition guarantees that adequate residence weights can be deduced from Eq. (9). One now describes in details a parallel procedure satisfying Eq. (11) that allows to generate $Z-1$ trial configurations from $\mathcal{C}_n$ and to select $\mathcal{C}_{n+1}$. Let us consider the reference (Markovian) moves that one wishes to implement in parallel. For a move from $\mathcal{C}_i$ to $\mathcal{C}_j$ with respective Boltzmann weight $\mathcal{P}_i$ and $\mathcal{P}_j$, one defines b_i^j the reference *a priori* probability and $a_i^j=(b_i^j)^{-1}\mathcal{G}(\mathcal{P}_i/\mathcal{P}_j)$ the corresponding acceptance rate. In the applicative part of the present study, all *a priori* probabilities will be equal to the same constant value. We however take them into account so as to draw the attention on the fact that they should not be omitted if one wishes to derive more general algorithms.

So as to construct a chain of weighted configuration $\mathcal{C}_n$, τ_n, one successively generates $Z-1$ new trial moves with *a priori* probabilities b_n^j, and selects one of them, let us say the j_1th. The relative probability for this particular selection is $a_n^{j_1}/(a_n^-+\Sigma_{j=1}^{Z-1}a_n^j)$, where a_n^- and a_n^j are the acceptance rates for the imposed backwards move and the jth trial move $(1\leqslant j\leqslant Z-1)$, respectively. The configuration to point backwards at the following step is the current configuration, whether the selected move is backwards or forwards. Let us note a_n^+ the acceptance rate for the selected transition and b_n^- the *a priori* probability for the imposed backwards move at the nth step. One also defines the two invariant quantities $a_n=a_n^-+\Sigma_{i=1}^{Z-1}a_n^i$ and $b_n=b_n^-\Pi_{i=1}^{Z-1}b_n^i$. Then the probability to generate the nth forwards parallel move is

$$\alpha(n\to n+1|n-1)=b_na_n^+/(a_nb_n^-)=\frac{b_n}{b_n^-a_nb_n^+}\mathcal{G}(\mathcal{P}_n/\mathcal{P}_{n+1}). \tag{12}$$

The presence of both b_n^+ and b_n^- in the denominator of the right hand side term in Eq. (12) results from the substitution of a_n^+ and from the fact that the backwards transition is imposed, respectively. Since all *a priori* probabilities are constant, this form is valid whether the transition is a reversal or not. Similarly, the probability to carry out a backwards move from $\mathcal{C}_n$ is

$$\alpha(n\to n-1|n+1)=b_n a_n^-/(a_n b_n^+)=\frac{b_n}{b_n^- a_n b_n^+}\mathcal{G}(\mathcal{P}_n/\mathcal{P}_{n-1}). \tag{13}$$

Since the ratio of equality (12) with respect to equality (13) satisfies the condition of Eq. (11), the residence weight deduced from Eq. (9) does not depend on which direction the chain is generated. Omitting now the constant factor relative to the *a priori* probabilities, the residence weight is

$$\tau_n=\frac{1}{a_n}, \tag{14}$$

whose form is analogous to the conventional ''residence time'' of the residence time algorithm detailed in Appendix A. The residence weight algorithm has been validated numerically on a body-centered cubic Ising model. Simple ensemble averages were performed by means of Eq. (4). It was checked that the computed values (such as chemical potentials) did not depend on the number of parallel insertion/deletions and were in agreement with results obtained using conventional Monte Carlo schemes.

A variant form for the RWA is also detailed in Appendix C and will be used in special cases to check that the obtained results do not depend significantly on the specificity of the present algorithm form. In the following and unless otherwise stated, simple particle displacements whose components are randomly chosen in a centred cube of edge 0.4 Å will be considered for the $Z-1$ reference moves. One will also use the Glauber-like residence weight algorithm detailed previously together with the acceptance function $\mathcal{G}(z)=z^{-1/2}$. Let us now present the Boltzmannian equivalent scheme for the residence weight algorithm.

IV. CORRESPONDING BOLTZMANNIAN ALGORITHM

The corresponding algorithm appears to be the energy-biased Monte Carlo scheme [15] or the parallel configurational-bias scheme [16]. This scheme is the transposition of the CBMC scheme to simple particle displacements, and is defined as follows: Z particle random displacements are generated in parallel ($k=1,\ldots,Z$) and one of them is selected according to the Rosenbluth weights, which are expressed as

$$P_r(i\to k)=\frac{\mathcal{P}_k}{\sum_{k'=1}^{Z}\mathcal{P}_{k'}}=\frac{\mathcal{P}_k}{\mathcal{W}_i}, \tag{15}$$

where $\mathcal{P}_{k'}$ is the Boltzmann weight of a trial configuration. If $\mathcal{C}_f$ is the selected configuration and $\mathcal{W}_f$ is the Rosenbluth weight obtained by generating $Z-1$ displacement from $\mathcal{C}_f$, and imposing the Zth last one to $\mathcal{C}_i$, then the statistical bias introduced is

$$\mathcal{B}_i^f=\frac{\mathcal{W}_i\mathcal{P}_i}{\mathcal{W}_f\mathcal{P}_f}. \tag{16}$$

The selected trial displacement is then accepted or rejected according to the following acceptance probability:

$$c(i\to f)=\min\left(1,\frac{\mathcal{W}_i}{\mathcal{W}_f}\right), \tag{17}$$

whose form more closely corresponds to the Metropolis-like RWA described in Appendix C. We will adopt the standardized terminology [16] and refer to this algorithm as the parallel configurational-bias scheme (PCB).

V. COMPARATIVE STUDY

A. Model problem

The comparative study between the two algorithms has been carried out in a model silicate. The kinetics of the slow structural relaxation observed in amorphous materials below its melting point is used as a benchmark for the study. Note that in this comparative study, the RWA and the PCB scheme are implemented with particle displacements and not particle insertions or deletions as it is usually the case when the PCB is implemented [1,17]. The reason is that in our model silicate, the slow relaxation kinetics offers a convenient and reliable tool to measure the algorithm efficiencies. Moreover, by varying the particle amplitude displacement, the mean acceptance rate will also vary and its influence on the efficiencies can be easily investigated.

The interatomic potentials are Born-Mayer-Huggins pair potentials with additional three-body terms involving O-Si-O and Si-O-Si triplets so as to obtain a better agreement with the experimental structures. Potential parameters are given elsewhere [18–20]. One has randomly inserted 384 silicon and 768 oxygen atoms in a cubic computational box of size 25.9 Å, which corresponds to a density of 2.21 g/cm^3. The box is partially equilibrated by performing Monte Carlo random atom displacements at the low temperature of 550 K. The final configuration is taken as the starting configuration for the subsequent study.

B. Preliminary definitions

In this feasibility study, the parallelization is actually not implemented, but will be studied by monitoring appropriate efficiency parameters. Technical details about the parallelization are given elsewhere [17]. The evolution of the relaxation kinetics will be monitored as a function of the number of parallel displacements, which corresponds to the computational time if the selection procedure and the communication time between processors is neglected. The evolution of the internal energy will also be monitored as a function of the total number of trial displacements, which corresponds to the total computational cost if the selection procedure and the communication time between processors is neglected.

One thus defines the parallel efficiency π_Z, which corresponds to the gain in computing time when implementing the moves in parallel with respect to the reference scheme considering only one trial move,

$$\pi_Z=\frac{\delta a_Z}{\delta a_X}\begin{cases}X=2 & \text{if RWA}\\ X=1 & \text{if PCB,}\end{cases} \tag{18}$$

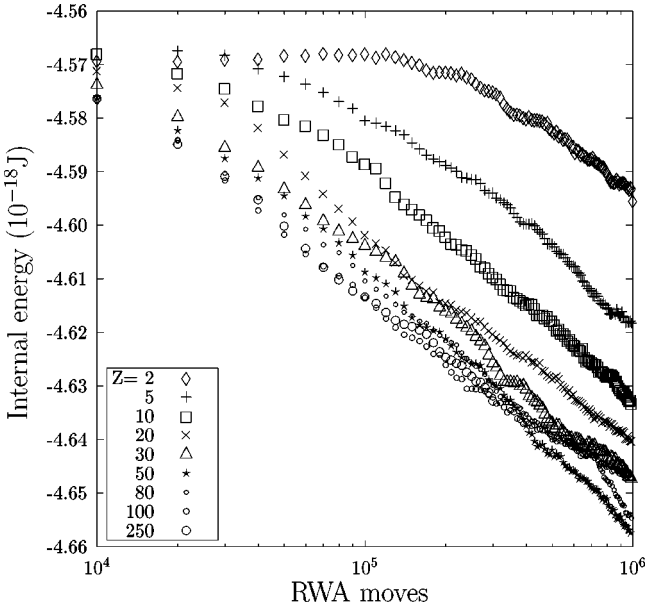

FIG. 1. Evolution of the internal energy as a function of the number of moves (abscissa) with the RWA and varying the number of parallel displacements, Z.

where δa_Z corresponds to the relaxation rate of the internal energy as a function of the number of Monte Carlo steps obtained with Z parallel moves. One also defines the sequential efficiency σ_Z, which corresponds to the gain in total CPU (central processing unit) (i.e., if sequential implementation). Depending on whether RWA or PCB is implemented, the relationship with the parallel efficiency is

$$\sigma_Z = \begin{cases} \pi_Z(Z-1) & \text{if RWA} \\ \pi_Z(2Z-1) & \text{if PCB.} \end{cases} \tag{19}$$

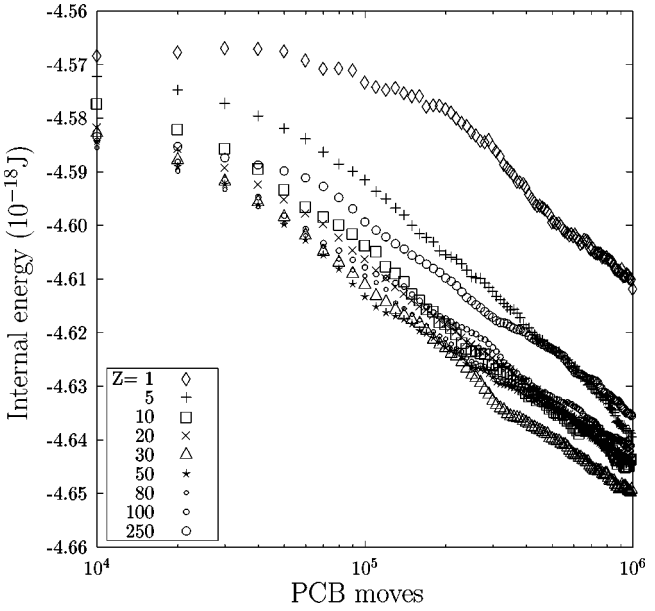

FIG. 2. Evolution of the internal energy as a function of the total number of trial displacements (abscissa) with the RWA and varying Z.

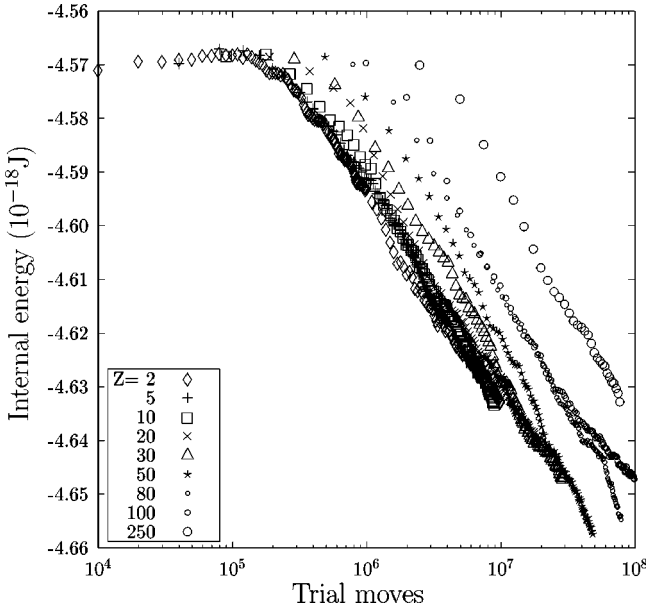

FIG. 3. Evolution of the internal energy as a function of the number of PCB attempted moves with varying Z.

Knowledge of the behavior of both parallel and sequential efficiencies as a function of the number of parallel displacements Z allows to optimize the use of a parallel computer. However, in our comparative study, it will also be instructive to define and monitor the relative efficiency of the two algorithms. This one can be obtained from the ratio of the relaxation rate for the RWA with respect to the one obtained for the PCB scheme as follows:

$$\rho_Z = [\delta a_Z^{RWA}(Z-1)]/[\delta a_Z^{PCB}(2Z-1)] \quad \text{if} \quad Z>2 \tag{20}$$

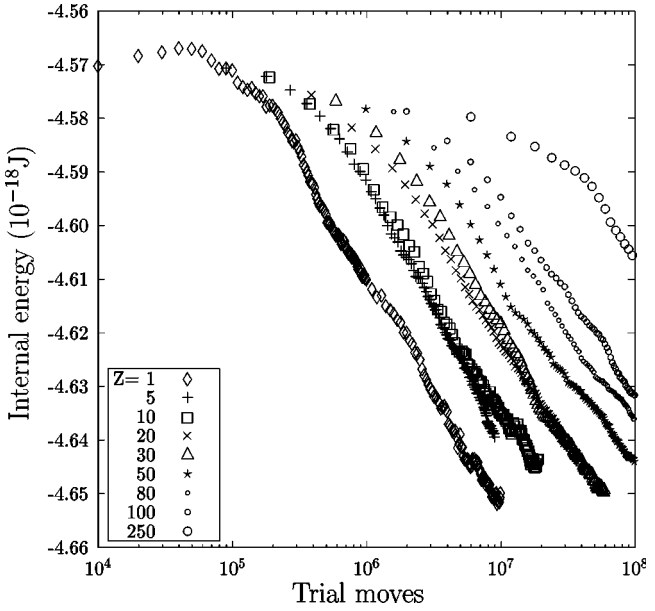

FIG. 4. Evolution of the internal energy as a function of the total number of trial moves (abscissa) with the PCB scheme and varying Z.

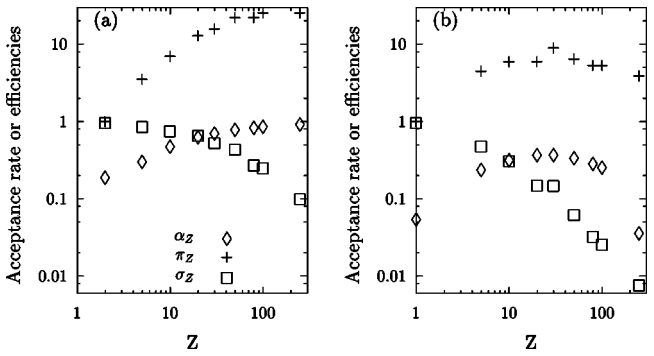

FIG. 5. Evolution of the mean acceptance rate α_Z, and of the parallel and sequential efficiencies (π_Z and σ_Z) as a function of the number of parallel trial moves *Z*: (a) for the RWA and (b) for the PCB scheme and the same legend as in (a).

and

$$\rho_2 = \delta a_2^{RWA} / \delta a_1^{PCB}. \tag{21}$$

C. Results and discussion

1. Parallel efficiency

Prior to comparing the residence weight algorithm to the parallel configurational-bias scheme, the influence of the number of parallel trial moves is investigated for both methods separately. Relaxation kinetics of the internal energy have been computed using the residence weight algorithm with various numbers of parallel displacements $Z(Z=2, 5, 10, 20, 30, 50, 80, 100, 250)$ and have been displayed in Figs. 1 and 2, respectively. The efficiency parameters have been estimated from the number of relaxation steps required by the system to reach $-4.618\ 10^{-18}$ J/atom. The quantities π_Z and σ_Z have been plotted in Fig. 5(a) so as to give an indication of the general trend. One observes that the efficiency π_Z increases with the increasing number of parallel displacements below $Z=50$ and then stabilizes beyond $Z=50$. The optimal choice is thus about $Z=50$ with a gain π_Z about 20. Relaxation kinetics of the internal energy have been computed using the parallel configurational-bias scheme with various numbers of parallel displacements Z $(Z=1, 5, 10, 20, 30, 50, 80, 100, 250)$ and are displayed in Figs. 3 and 4. The quantities π_Z and σ_Z have also been estimated from the number of relaxation steps required by the system to reach $-4.63\ 10^{-18}$ J/atom and have been plotted in Fig. 5(b). One observes that the efficiency of the algorithm π_Z increases with the increasing number of parallel displacements until $Z=30$ and then abruptly decreases beyond $Z=30$. The optimal choice is thus about 30 with a gain of about $\pi_Z=9$.

So as to analyze and compare the observed behaviors, the mean acceptance rates α_Z for both the RWA and the PCB scheme were computed and displayed in Fig. 5 as a function of *Z* together with π_Z and σ_Z. A move is said to be accepted with the RWA when it corresponds to a forwards move $(\mathcal{C}_{n+1} \neq \mathcal{C}_{n-1})$. The mean acceptance rate for the standard Metropolis algorithm corresponds to α_1 for the PCB scheme and was found equal to 5.6%. For both parallel schemes, when the number of parallel generation first increases, the probability to have generated at least one favorable displacements is also enhanced, which explains the subsequent increase of both α_Z and π_Z. For large increasing *Z* values and the RWA, α_Z is greater than 80% and saturates. The efficiency π_Z thus also saturates since only a single favorable displacement can be carried out at once. For large increasing *Z* values and the PCB scheme, the observed decreasing π_Z is also accompanied by a decreasing α_Z value and is indeed a consequence of this weak acceptance rate.

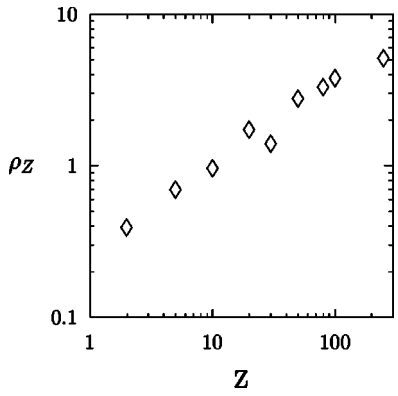

FIG. 6. Evolution of the relative efficiency ρ_Z as a function of the number of parallel displacements, *Z*.

The fact that α_Z decreases results from the way the PCB scheme operates. The statistical bias used by the scheme requires to compute an *a priori* probability for the system to generate and select the initial configuration from the final configuration. If the forwards move leads to an energetically favorable configuration, and if favorable moves are also generated from the final configuration, which is likely for *Z* large enough, then it means that one imposes the selection of an unfavorable configuration for the reverse *a priori* move among a set of configurations containing one or more energetically favorable configurations. For *Z* large enough and assuming that favorable displacements present a spectrum of energy variations that is similar from both $\mathcal{C}_i$ and $\mathcal{C}_f$, the ratio $\mathcal{W}_i/\mathcal{W}_f$ of the Rosenbluth factors, which enters the ac-

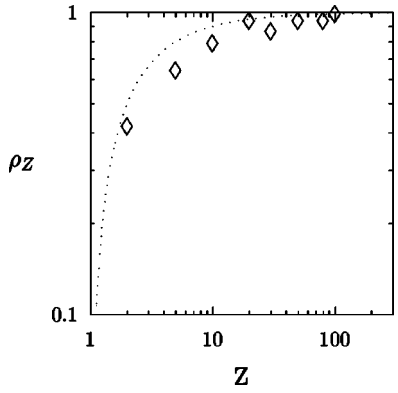

FIG. 7. Evolution of the relative efficiency ρ_Z as a function of the number of parallel displacements, *Z*, with a mean acceptance rate of $a_m = 3.2 \times 10^{-3}$. The dotted line corresponds to the function $(Z-1)/Z$.

ceptance probability for the move is expected to be similar to $\mathcal{P}_i/\mathcal{P}_f$. This explains why the mean acceptance rate α_Z decreases, as observed in Fig. 5. Note that the statement about the Rosenbluth ratio would not be correct if the trial sites were placed into a region of phase space different from the old site, e.g., the particle swap move in the Gibbs ensemble.

The above mentioned limitation does not appear with the RWA. The reason lies in the structure of the algorithm itself. Let us consider that the system has transited from $\mathcal{C}_{n-1}$ to an energetically favorable configuration $\mathcal{C}_n$. For large enough Z one can assume that the spectrum of the Z energy variations is equivalent from both $\mathcal{C}_{n-1}$ or $\mathcal{C}_n$, because trial sites are placed near the old site. The algorithm will generate an energetically favorable configuration $\mathcal{C}_{n'}$ with a high probability and will select it ($\mathcal{C}_{n+1} \equiv \mathcal{C}_{n'}$) with a high probability as well, although the correcting bias associated to configuration $\mathcal{C}_n$ will be very small [residence weight τ_n in Eq. (14)]. Favorable forwards moves are thus preferentially selected to the detriment of unfavorable backwards moves.

2. Sequential efficiency

Let us now examine the sequential efficiencies σ_Z. Figure 5 displays σ_Z for both the RWA and the PCB scheme. We observe that both efficiencies decreases as Z increases with Z. This means that if sequential implementation of the random displacements is indeed carried out, both the RWA and the PCB scheme do not present any advantages with respect to the standard Metropolis algorithm or the RWA with a single displacement ($Z=2$), respectively. The reason for this behavior may be explained as follows: both parallel algorithms carry out single moves at each step and the probability to generate more than one favorable move increases with Z; since only a single move is implemented at best, the number of favorable moves that are not implemented also increases with Z, and the sequential efficiency decreases.

Note that the loss of CPU for the RWA and $Z=50$ (stabilization of the parallel efficiency) is about a factor of 2.5 ($\sigma_Z=0.4$), while the loss for the PCB scheme and $Z=30$ (optimal parallel efficiency) is about a factor of 6.6 ($\sigma_Z=0.15$). The RWA seems to require less total CPU. This suggests to investigate the relative efficiency of the RWA with respect to the PCB scheme.

3. Relative efficiency

In the previous discussions, the parallel and sequential efficiencies were considered independently for each algorithms. We now examine the relative efficiency for these two algorithms.

One observes in Fig. 6 that, the relative efficiency is lower than 1 when Z is lower than 5: the RWA performs less efficiently. To explain this behavior, a first reason can be suggested: forwards probabilities in the present RWA that are smaller than the ones that would be obtained with the Metropolis form (detailed in Sec. III) are responsible for the relative slowing down of the relaxation dynamics. One has therefore implemented the Metropolis-like RWA to check this point. However, one did not observe any substantial differences in the relaxation rates with respect to the ones obtained with the Glauber-like RWA. The reason for the weak performance of the RWA can be finally attributed to the fact that when Z is small, an energetically favorable move is more likely followed by an unfavorable move due to the form of the selection probabilities, if we compare to the conventional Metropolis algorithm. Even though an unfavorable move subsequently leads to numerous reversals, it only results in slowing down the relaxation dynamics by a factor of about 2 with respect to the conventional dynamics.

At variance, the relative efficiency is greater than 1 when Z is larger than 10: the RWA outperforms the PCB scheme. However, when Z is larger than the mean acceptance rate a_m^{-1} for Metropolis moves, the PCB acceptance rate α_Z decreases drastically for the structural reasons explained previously and it is indeed this fact that explains the relative outperformance of the RWA.

The outlined limitation of the PCB scheme is purely theoretical and is not relevant from a practical view point, since the parallel configurational-bias scheme should be implemented in cases where the mean acceptance rate is about or lower than 10^{-3}. It is thus instructive to monitor the relative efficiency in such a case.

We have therefore carried out additional simulations with a lower mean acceptance rate ($a_m^{-1} \approx 3.210^{-3}$). This was artificially obtained by setting the particle displacement amplitude to 1.2 Å instead of 0.4 Å. The relative efficiency has been monitored with varying the number of parallel trial moves up to 100. It is observed in Fig. 7 that the relative efficiency tends towards 1 with increasing Z. In this case the RWA does not outperform the PCB scheme.

The fact that the performance of both RWA and PCB scheme are similar for difficult problems results from the structure of the RWA. Since the reversal rate is important with the RWA even for large numbers of parallel trial moves ($Z=100$), a forwards move will be likely followed by several reversals. The algorithm oscillates around an energetically favorable configuration and performs numerous reversals leading to unfavorable configurations. From an unfavorable configuration the system will likely return to the favorable configuration but will waste $2Z-2$ trial moves (including both the forwards and backwards transition). This schematic representation implies that for $Z=2$, the RWA is less efficient by a factor of 2 than the Metropolis algorithm (PCB with $Z=1$) and that the dependence on Z of the relative efficiency can be approximated by $(Z-1)/Z$. This simple predictive law is approximately observed in Fig. 7.

Note that the observed trend seems to be general. Indeed, additional simulations were carried out using a different form for the acceptance function $\mathcal{G}(z)=\min(1,z^{-1})$. This form more closely corresponds to the Rosenbluth weight. No significant differences were observed with the results obtained using the standard form [$\mathcal{G}(z)=z^{-1/2}$].

VI. CONCLUSION

The residence weight algorithm developed in this paper is a Monte Carlo sampling scheme. It considers Z parallel moves, among which $Z-1$ new trial moves have been generated and the Zth move points backwards to the previous

configuration, selects a single move to carry out among the Z transitions and attributes a correcting ''residence weight'' to the current configuration. These features make this algorithm to be both a non-Boltzmannian and non-Markovian sampling scheme, which can be considered as the extension of the standard residence time algorithm to continuous ensembles.

The residence weight algorithm has been compared to the parallel configurational-bias scheme that appears to be the corresponding Boltzmannian scheme. The main difference of this latter scheme with respect to the former algorithm lies in the fact that the correcting biases are introduced in Metropolis-like acceptance probabilities. The two methods are worth being implemented on parallel processors if the reference Monte Carlo moves require similar and long computational times relatively to the selection procedure, which appears to be the case when continuous interatomic potentials are involved. Computer time can then be saved if the evaluation of the Z internal energy is distributed on separate processors. The efficiency of the two Monte Carlo algorithms have been compared in a case study. For difficult problems where the mean acceptance rate of the individual moves is very small and where parallelization is worth being implemented, we observe that: (i) when the number of parallel moves is small (below five), the residence weight algorithm performs slower by a factor of about 2; (ii) when the number of parallel moves increases above five, the relative efficiency of the residence weight algorithm with respect to the parallel configurational-bias scheme increases and tends towards 1.

The reasons for this feature were found to lie in the structure of the schemes themselves. Because the residence weight algorithm can return to an older configuration (reversal), it requires twice more Monte Carlo steps than the parallel configurational-bias scheme but utilizes the same subset of trial moves twice. It results that the efficiency of both schemes becomes equivalent as soon as a sufficient number of trial moves are generated in parallel.

The choice for implementing the parallel configurational-bias scheme or its corresponding non-Boltzmannian variant form should therefore be dictated by the nature of the considered problem. We show in a forthcoming paper [14] that the residence weight algorithm is particularly well suited to the computation of chemical potentials by means of arbitrarily biased Monte Carlo schemes for inserting or deleting particles.

ACKNOWLEDGMENTS

The author is grateful to Dr. J.-L. Bocquet, Dr. Y. Limoge, Dr. D. Ghaleb, Dr. G. Martin, and Professor P. Bellon for fruitful and stimulating discussions.

APPENDIX A: RESIDENCE TIME ALGORITHM

The residence time algorithm (RTA), from which the residence weight algorithm is derived, is recalled here. The RTA was proposed by Lanore [9] as a mathematical trick allowing to save computing time in Monte Carlo simulations dealing with many independent Poissonian processes. It consists of considering the stochastic process product of all the Poissonian processes, which is fortunately a Poissonian process with an appropriate relaxation rate. Similar arguments were applied to the Ising model [10] in which each spin was considered to be an independent Poissonian process.

Here, we recall the approach of Novotny [21] who showed, using very simple probability arguments, that the RTA can be derived from the reformulation of a conventional Metropolis-like algorithm. Since the computational box is finite and the considered systems are discrete, one can define Z_i the finite number of accessible transitions from configuration $\mathcal{C}_i$ and $a_i^j = c(i \to j)$ the acceptance probabilities. The residence time algorithm operates as follows: one computes the Z_i Metropolis acceptance rates a_i^j and their sum $a_i = \Sigma_{j=1}^{Z_i} a_i^j$; one selects a configuration $\mathcal{C}_f$ using the relative probabilities $\alpha_i^j = a_i^j / a_i (1 \leqslant j \leqslant Z_i)$, and attributes to configuration $\mathcal{C}_i$ the weight $\tau_i = 1/(a_i Z_i)$ thereafter denominates the residence time. Let us outline that from a given configuration $\mathcal{C}_i$, the residence times τ_i is independent of the previous configurations because all accessible transitions are considered and so the previous configuration is naturally taken into account. This limits the use of the algorithm to discrete systems. The equivalence with a Metropolis scheme results from the correspondence between the mean number of Metropolis unsuccessful attempts before leaving the current configuration $\mathcal{C}_i$ and the mean residence time on that configuration

$$\tau_i = \sum_{k=1}^{\infty} k(1 - a_i/Z_i)^{k-1} a_i/Z_i = (a_i/Z_i)^{-1}, \qquad \text{(A1)}$$

where $(1 - a_i/Z_i)^{k-1} a_i/Z_i$ is to be interpreted as the probability of $k-1$ successive Metropolis rejections before an acceptation. Moreover α_i^j corresponds to the escape probability through exit j since $\alpha_i^j = \Sigma_{k=1}^{\infty} (1 - a_i/Z_i)^{k-1} a_i^j/Z_i$.

The use of a Metropolis algorithm thus seems to be more efficient when the mean number of rejections is smaller than the number of accessible transition Z_i, which is equivalent to a residence time smaller than $Z_i (a_i > 1)$. At variance, when τ_i is large compared to $Z_i (a_i < 1)$, the use of the residence algorithm becomes preferable over the Metropolis-like algorithm. Although the parallel implementation of this algorithm on separate processors seems straightforward, it was never carried out. The reason is that the residence time algorithm is usually implemented with appropriate tabulations [10] in systems where the evaluation of the configurational energy only represents a small fraction of the computing time so that the gain that can be obtained from parallelization is not significant.

Let us now consider the various forms that can be used for the acceptance probabilities. In equilibrium Monte Carlo studies, the Metropolis-like probabilities $a_{ij} = \min[1, \exp -\beta(E_j - E_i)]$ are used very often and are very convenient. However, when kinetics are studied [22,23] a different choice should be made: transition rates depending on an activation barrier with the form $a_i^f = \exp -\beta(E_s - E_i)$ are to be used instead, where E_i is the internal energy of the initial configuration $\mathcal{C}_i$ and E_s is the energy at the saddle position

$\mathcal{C}_s$, respectively. The reverse rates are similarly defined as $a_f^i = \exp-\beta(E_s - E_f)$, where E_f and is the internal energies of the final configuration $\mathcal{C}_f$ and E_s is identical. The transition rate ratio ensures that detailed balance Eq. (5) is still obeyed for any two consecutive residence times chosen along the configuration chain, even though a_f^i and a_i^f are no more probabilities.

Finally, since the algorithm proposed in Sec. III appears to be a straightforward extension of the residence time algorithm the name ''residence weight algorithm'' was chosen. The terminology ''residence time'' was replaced by ''residence weight'' because the non-Markovian nature of this algorithm prevents from any statistical equivalence to the dynamics of a Metropolis-like algorithm. Note that non-Markovian residence time algorithms have also been developed [23] and are currently used in lattice kinetic studies [24,25] to improve the computational efficiency. The principle of these algorithms consists in eliminating all the reversal events that would be generated if the standard algorithm was used. To do this, generalized residence times are computed to account for the numerous reversal events that are subsequently not carried out. At a given configuration, the generalized residence time used by such a non-Markovian algorithm indeed corresponds to a mean residence time over all the accessible configurations and, therefore, cannot be interpreted as a correcting weight as in the detailed balance equation (3).

APPENDIX B: CONFIGURATIONAL MONTE CARLO SCHEMES

The principle of the configurational-bias Monte Carlo scheme, on which is based the parallel configurational-bias scheme is briefly recalled. This efficient method is devoted to the sampling of polymer conformations and consists of inserting a polymer chain one bead after the other. For each polymer bead, several, let say *Z*, trial locations are generated in parallel, but only one particular trial location is selected according to an adequate probability function, which associates the highest probability to the move towards the lowest energy configuration. The commonly used Rosenbluth probability function satisfies this property. With this procedure, each bead location has been optimized, so the resulting acceptance probability is enhanced by a factor increasing ''almost'' exponentially with the polymer length when compared to a direct method generating unbiased polymer conformations. This gain thus largely compensates the additional cost of generating the equivalent *Z* polymer chains, cost that grows linearly with the polymer length. This explains why the CBMC method reveals to be very efficient. The method is general and variants have been developed for polymer sampling [26,27].

Since the configurational-bias method involves the computation of several independent operations (tasks) in parallel, a natural parallelization of the method would consist of distributing the computations on separate processors, as shown by Smit and co-workers in a previous study [1]. In this variant form, several polymer conformations themselves were generated in parallel, corresponding Rosenbluth factors were computed and used for the selection of one particular conformation. Because large portions of the computation are implemented in parallel, the configurational-bias method results in a net gain in computing time. As outlined by the authors [1], this property is general and can be exploited in cases other than polymer sampling.

APPENDIX C: VARIANT FORM FOR THE RESIDENCE WEIGHT ALGORITHM

Variant forms for the residence weight algorithm can be proposed. For instance, a different choice for the selecting probabilities is possible. The present residence weight algorithm reduces to an algorithm with a Glauber form when a single trial move is generated ($Z=2$). Since Glauber-like acceptance probabilities are smaller than their corresponding Metropolis probabilities, it can be suggested that using a residence weight algorithm with a Metropolis form might be advantageous. A parallel residence weight algorithm with a Metropolis form can be obtained by introducing the following modifications to the algorithm detailed previously: one generates the $Z-1$ trial moves; using the same notation as previously, one computes the modified selecting probabilities $a_{n'}^i/(a_n - a_n^{n-1})$ and then selects $\mathcal{C}_{n'}$ among the $Z-1$ generated configurations; one decides to transit to $\mathcal{C}_{n'}$ with probability $p^+ = \min[1;(a_n - a_n^{n-1})/(a_n - a_n^{n'})]$ or returns to $\mathcal{C}_{n-1}$ with probability $p^- = 1 - [(a_n - a_n^{n-1})/(a_n - a_n^{n'})]$; one computes the residence weight of the current configuration from Eq. (9):

$$\tau_n = \begin{cases} \min[(a_n - a_n^{n-1})^{-1}; (a_n - a_n^{n+1})^{-1}] & \text{if} \quad \mathcal{C}_{n+1} \equiv \mathcal{C}_{n'} \\ (a_n - a_n^{n-1})^{-1} - (a_n - a_n^{n'})^{-1} & \text{if} \quad \mathcal{C}_{n+1} \equiv \mathcal{C}_{n-1}. \end{cases} \tag{C1}$$

One can check that τ_n is invariant with respect to a chain reversal and notice that this algorithm reduces to a Metropolis form for $Z=1$.

Finally, let us outline that in the derivation of the algorithm, the acceptance rates are not necessarily acceptance probabilities. These quantities can be derived from any appropriate acceptance function $\mathcal{G}$ as long as the residence weight invariance with respect to a chain reversal is guaranteed. We have thus carried out several simulations with a different form for the acceptance rates. For instance, we have chosen $a_n^{n'} = \min(1, \mathcal{P}_{n'}/\mathcal{P}_n)$ and did not notice any significant differences on the efficiencies with respect to the choice made in the present study. However, an essential point to note is that the acceptance rates should always be a function of Boltzmann weight ratios. Let us consider a case where this point is not respected, for instance, if the acceptance rate $a_n^{n'} = \mathcal{P}_{n'}$ was chosen. The selection probability would become equal to the Rosenbluth weight or to the selection probability for the conventional algorithm [$\mathcal{G}(z) = z^{-1/2}$] at half the actual temperature. It would then appear that the algorithm would generate a configuration chain that would ''approximately'' samples the ensemble at half the desired temperature (by approximate sampling we mean if the residence weights were omitted from the ensemble average).

The use of the adequate weighted sampling procedure would correct for the fact that the configuration chain is distributed very far from the desired temperature. Nevertheless, the use of such a Rosenbluth selection probability, even if correct in principle, would yield a statistically inefficient sampling scheme.

[1] K. Esselink, L. D. J. C. Loyens, and B. Smit, Phys. Rev. E **51**, 1560 (1995).
[2] P. J. Rossky, J. D. Doll, and H. L. Friedman, J. Chem. Phys. **69**, 4628 (1978).
[3] M. Rao and B. J. Berne, J. Chem. Phys. **71**, 129 (1979).
[4] B. Mehlig, D. W. Heermann, and B. M. Forrest, Phys. Rev. B **45**, 679 (1992).
[5] B. Mehlig, D. W. Heermann, and B. M. Forrest, Mol. Phys. **76**, 1347 (1992).
[6] J. A. Siepmann and D. Frenkel, Mol. Phys. **75**, 59 (1992).
[7] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).
[8] C. Robert, *Méthodes de Monte Carlo par Chaînes de Markov* (Economica, Paris, 1996).
[9] J.-M. Lanore, Radiat. Eff. **22**, 153 (1974).
[10] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz, J. Comput. Phys. **17**, 10 (1975).
[11] D. Frenkel, G. C. Mooij, and B. Smit, J. Phys.: Condens. Matter **4**, 3053 (1992).
[12] D. Frenkel, G. C. Mooij, and B. Smit, Mol. Phys. **75**, 983 (1992).
[13] B. Widom, J. Chem. Phys. **39**, 2802 (1963).
[14] M. Athènes (unpublished).
[15] E. Leontidis and U. W. Suter, Mol. Phys. **83**, 489 (1994).
[16] D. Frenkel and B. Smit, *Understanding Molecular Simulation* (Academic, San Diego, 2001).
[17] T. J. Vlugt, Mol. Simul. **23**, 63 (1999).
[18] J.-M. Delaye, CEA- srmp 93-59, 1993 (unpublished).
[19] J.-M. Delaye, CEA- srmp 93-58, 1994 (unpublished).
[20] J.-M. Delaye and D. Ghaleb, J. Nucl. Mater. **244**, 22 (1997).
[21] M. A. Novotny, Mol. Simul. **9**, 46 (1995).
[22] M. Athènes, P. Bellon, G. Martin, and H. Haider, Acta Mater. **44**, 4739 (1996).
[23] M. Athènes, P. Bellon, and G. Martin, Philos. Mag. A **76**, 527 (1997).
[24] M. Athènes, P. Bellon, and G. Martin, Acta Mater. **48**, 2675 (2000).
[25] T. Rautiainen and A. Sutton, Phys. Rev. B **59**, 13 681 (1999).
[26] S. Consta, N. B. Wilding, D. Frenkel, and Z. Alexandrowicz, J. Chem. Phys. **110**, 3220 (1999).
[27] S. Consta, T. J. Vlugt, H. J. W. Hoeth, D. Smit, and B. Frenkel, Mol. Phys. **97**, 1243 (1999).