

Cluster algorithm to perform parallel Monte Carlo simulation of atomistic systems

N. G. Almarza^{a)} and E. Lomba

Instituto de Química Física Rocasolano (CSIC), C/Serrano 119, E-28006 Madrid, Spain

(Received 23 May 2007; accepted 25 June 2007; published online 30 August 2007)

We propose an efficient algorithm to perform Monte Carlo simulations of dense systems using multiple particle moves. The method is intended to be used in the atomistic simulation of complex systems, where the computing requirements for a single simulation run make advisable the use of parallel computing. The algorithm is based on the use of steps in which all the particle positions of the system are perturbed simultaneously. A division of the system in clusters of particles is performed, using a bonding criterion which makes feasible that the acceptance or rejection of the new particle coordinates can be carried out independently for each cluster. © 2007 American Institute of Physics. [DOI: 10.1063/1.2759924]

I. INTRODUCTION

Molecular dynamics (MD) and Monte Carlo (MC) simulation algorithms^{1,2} provide a large number of powerful techniques in the study of condensed matter physics. The simulation of a number of complex systems has become possible by the use of parallel computing. At present, MD is usually more suitable than MC simulation for parallel computing. For systems with a large number of particles interacting through short range potentials, in MD simulation one can perform the so-called domain decomposition³ of the system and distribute the load between different processors. Each processor then integrates the equations of motion of the particles located inside its assigned domain and sends and receives information (to/from) the processors in charge of the neighboring domains. On the other hand, parallel computing has also been used to perform quantum molecular dynamics. In this case the power of parallelism is applied to solve the electronic structure problem.

The use of parallel computing in MC calculations for atomistic models is not so spread as in MD simulation. One can find in the literature a number of parallel MC applications to particular problems,^{4,5} but a general strategy that could be applied to a wide class of problems is still lacking. In contrast, in the field of simple lattice models there are efficient algorithms⁶⁻⁹ that can be run in parallel machines quite easily.¹⁰ In most of these procedures a large number of particles are *moved* in a single MC step. The possibility of dividing the system in sublattices of sites makes possible the construction of effective methods which can be run without difficulty on parallel architectures. For lattice models, the cluster methods, which have become a powerful tool in the analysis of criticality, make also use of motions involving large numbers of particles and are quite amenable to be parallelized as well.

In principle one can think that a strategy inspired in the sublattice approach could be of use in a parallel version of

the Metropolis Monte Carlo simulation of continuous models. One could perturb simultaneously different particles to perform several single-particle steps in parallel. This strategy can work if and only if there is no direct interaction between the perturbed particles, and in all other cases one could group the particles in order to perform one (or several) multiparticle step(s). This circumstance makes the strategy much less effective in the simulation of condensed matter systems with continuous translational degrees of freedom, due the presence of repulsive interactions between particles² which induce a low acceptance rate for multiparticle moves.

For the particular case of models with short-ranged pairwise additive interactions we can try to explicitly choose sets of noninteracting particles to be moved simultaneously. This procedure can work satisfactorily for systems with crystalline order, but it seems quite difficult to implement in disordered systems, if detailed balance is to be kept.

In the case of lengthy simulations/large systems (e.g., in near critical conditions) and in order to take full advantage of the power of parallel computing, it seems highly desirable to devise efficient MC simulation strategies involving multiple particle moves¹¹ that are amenable to parallel coding. Such procedures could compete with existing MD algorithms for complex systems and at the same time would take advantage of the adaptability of MC methods when dealing with specific features of particular problems. In response to this need, in this contribution we propose an efficient and rigorous algorithm to perform MC simulation of dense systems using multiple particle moves. The method is based on cluster algorithms⁶⁻⁹ and partially inspired in the pioneering work of Kražčí and Hafner.¹¹ Even if our approach is designed for systems with pairwise additive potentials, we will see how different strategies can be used to cope with systems in which many-body interactions are present.

The paper is organized as follows; after this Introduction, in Sec. II we introduce our cluster algorithm as applied to an atomic system. In Sec. III we present the details of the simulation, the model we have used as a practical example, and summarize the results of the efficiency analysis for dif-

^{a)}Electronic mail: NOE@IQFR.CSIC.ES

ferent algorithms. Finally, our conclusions are to be found in Sec. IV. There we also comment on possible alternatives to extend these cluster algorithms to systems with many-body interactions.

II. ALGORITHM

In what follows we will deal with a system of particles interacting with an isotropic pair potential, $V(r)$. Let us consider a given configuration of N particles, $\{\mathbf{R}_{1o}, \mathbf{R}_{2o}, \dots, \mathbf{R}_{No}\}$; we can build up a set of alternative (or test) positions for the particles $\{\mathbf{R}_{1t}, \mathbf{R}_{2t}, \dots, \mathbf{R}_{Nt}\}$ using the standard sequential algorithms in which one applies a random displacement on each particle position.¹ Our goal is now to perform a multiparticle MC step to generate a new configuration of the system. In the new configuration, $\{\mathbf{R}_{1n}, \mathbf{R}_{2n}, \dots, \mathbf{R}_{Nn}\}$, some particles will keep the old coordinates, $\mathbf{R}_{in} = \mathbf{R}_{io}$, whereas other particles will have the test coordinates, $\mathbf{R}_{jn} = \mathbf{R}_{jt}$. This approach was also followed in the approximate method¹² proposed by Krajčič and Hafner.¹¹

A. Two state per particle model

Let us consider a subset of the phase space of the system, in which each particle can adopt only two positions (*states*) represented as \mathbf{R}_{i+} and \mathbf{R}_{i-} for particle i . We can relate these N pairs of particle positions to the previous sets of coordinates. Here $\{\mathbf{R}_{io}\}$, $\{\mathbf{R}_{it}\}$, and $\{\mathbf{R}_{in}\}$ denote different choices of coordinates from the set of N pairs of particle positions. Within this description, a multiparticle MC step shows an evident analogy with the simulation of Ising-type models using collective moves. We will show how an efficient procedure can be built to sample such a subset of the phase space and then how to include these ideas in the simulation of the continuous model.

For a given pair of particles (i, j) we can find four possible values of the pair interaction, V_{ij} , depending on the choices of the coordinates \mathbf{R}_i and \mathbf{R}_j ,

$$V_{S_i, S_j}^{(ij)} = V(|\mathbf{R}_{i, S_i} - \mathbf{R}_{j, S_j}|), \quad (1)$$

where S_k is +1 or -1 for states R_{k+} and R_{k-} , respectively.

1. Cluster algorithm

We define a probability of having a bond between particles i and j at states S_i and S_j as

$$P_{ij}^b(S_i, S_j) = 1 - \min \left\{ \frac{\exp[\beta \tilde{V}_{ij}(S_i, S_j)]}{\exp[\beta \tilde{V}_{ij}(\bar{S}_i, S_j)]}, 1 \right\}, \quad (2)$$

where

$$\tilde{V}(S_i, S_j) = \frac{1}{2} [V_{S_i, S_j}^{(ij)} + V_{S_i, \bar{S}_j}^{(ij)}], \quad (3)$$

and $\beta \equiv 1/k_B T$ (k_B being the Boltzmann constant and T the absolute temperature), and $\bar{S}_k \equiv -S_k$. For each pair of particles (i, j) a random number, ξ , with uniform probability in the range (0,1] is generated, and a bond between particles i and j is assumed if ξ is less than the probability given by Eqs. (2) and (3). Then, clusters of particles are built according to the bonds generated, and a global MC step, in which a

unique choice (either modify or keep the old coordinates) is done for all particles belonging to each cluster.

Let us denote the possible two configurations of the k th cluster in the new configuration as \mathbf{S}_k^+ and \mathbf{S}_k^- , where \mathbf{S}_k^+ is the current configuration and \mathbf{S}_k^- is that built with the trial positions (or, in the Ising language, the one generated when flipping every spin of the cluster). We have tested two schemes to select the probability of flipping the clusters. Taking into account the analogies with the standard MC technique for simulating Ising models, we will refer to them as *Metropolis* and *Glauber* procedures.⁶ The respective probabilities, w , of flipping a given cluster are given by

$$w(\mathbf{S}_k^- | \mathbf{S}_k^+) = \begin{cases} \min[1, e^{-\beta U_k^-} / e^{-\beta U_k^+}], & \text{Metropolis} \\ e^{-\beta U_k^-} / (e^{-\beta U_k^+} + e^{-\beta U_k^-}), & \text{Glauber,} \end{cases} \quad (4)$$

with

$$U_k^\pm = U_1(\mathbf{S}_k^\pm) + \frac{1}{2} \sum_{l \neq k} [U_2(\mathbf{S}_k^\pm, \mathbf{S}_l^+) + U_2(\mathbf{S}_k^\pm, \mathbf{S}_l^-)], \quad (5)$$

where $U_1(\mathbf{S}_k^\pm)$ is the sum of pair interactions between atoms belonging to cluster k in the corresponding configuration (+ or -), and $U_2(\mathbf{S}_k, \mathbf{S}_l)$ is the sum of pair interactions between clusters k and l . Let us emphasize that the terms Metropolis and Glauber are just used by analogy of the flipping probabilities $w(\mathbf{S}_k^- | \mathbf{S}_k^+)$ with those used in the standard MC procedures with single-particle moves. In addition, notice that the Glauber procedure is also known in the literature as Barker's method.¹³

2. Detailed balance

The choice of the flipping probabilities, w , must guarantee that the sampling of the different realizations of the two state per particle model is done according to the Boltzmann distribution. Let us use A and B as the symbols for two microstates of the system. If our simulation scheme fulfills detailed balance,^{1,2} we will get

$$P(A)v(A \rightarrow B) = P(B)v(B \rightarrow A), \quad (6)$$

where $P(\alpha)$ (with $\alpha = A, B$) is the probability of the configuration and $v(A \rightarrow B)$ expresses the transition probability from A to B . The transition between the two microstates can go through different realizations of particle bonds. Let us denote the bond configurations which are compatible with the transition between the two states as \mathcal{C}_l . We can write the detailed balance condition as

$$\frac{P(A) \sum_l \mathcal{P}(\mathcal{C}_l | A) v(A \rightarrow B | \mathcal{C}_l)}{P(B) \sum_l \mathcal{P}(\mathcal{C}_l | B) v(B \rightarrow A | \mathcal{C}_l)} = 1, \quad (7)$$

where $\mathcal{P}(\mathcal{C} | \alpha)$ express the probability of generating the bond configuration \mathcal{C} when the system microstate is α and $v(A \rightarrow B | \mathcal{C}_l)$ is the transition probability when the bond configuration is \mathcal{C}_l . Equation (6) is satisfied if superdetailed balance² is fulfilled,

$$\frac{P(A)}{P(B)} = \frac{\mathcal{P}(C_l|B) v(B \rightarrow A|C_l)}{\mathcal{P}(C_l|A) v(A \rightarrow B|C_l)}. \quad (8)$$

Let us consider now the rate between the bond realization probabilities in the two microstates. For a given pair of particles, i and j , consider the product $K_{ij}=S_i S_j$ in states A and B . If K_{ij} changes in the transition, then i and j cannot belong to the same cluster. In addition, according to Eqs. (2) and (3) the bonding probability P_{ij}^b only changes if K_{ij} is different in the two states. Therefore for the bond configurations compatible with the transition we can write

$$\frac{\mathcal{P}(C_l|B)}{\mathcal{P}(C_l|A)} = \frac{\exp[-(\beta/2)\sum_i^{\#}\sum_j^*(V_{S_i^A, S_j}^{(ij)} + V_{\bar{S}_i^A, \bar{S}_j}^{(ij)})]}{\exp[-(\beta/2)\sum_i^{\#}\sum_j^*(V_{S_i^B, S_j}^{(ij)} + V_{\bar{S}_i^B, \bar{S}_j}^{(ij)})]}. \quad (9)$$

The summation $\Sigma^{\#}$ is restricted over flipped particles, whereas Σ^* refers to particles whose state does not change in the transition. Notice that the result given in Eq. (9) is independent of the bonding configuration. We can write (9) in terms of the cluster interactions,

$$\frac{\mathcal{P}(C_l|B)}{\mathcal{P}(C_l|A)} = \frac{\exp\{-(\beta/2)\sum_k^{\#}\sum_m^*[U_2(\mathbf{S}_k^A, \mathbf{S}_m) + U_2(\mathbf{S}_k^B, \bar{\mathbf{S}}_m)]\}}{\exp\{-(\beta/2)\sum_k^{\#}\sum_m^*[U_2(\mathbf{S}_k^B, \mathbf{S}_m) + U_2(\mathbf{S}_k^A, \bar{\mathbf{S}}_m)]\}}. \quad (10)$$

In Eq. (10) the sums are defined over clusters, and the meaning of the sum superscripts is similar to that of Eq. (9). On the other hand, taking into account Eq. (4), the ratio between conversion rates (within a given bond configuration) can be written as

$$X_{B|A} \equiv \frac{v(B \rightarrow A|C_l)}{v(A \rightarrow B|C_l)} = \prod_k^{\#} \frac{w(\mathbf{S}_k^A|\mathbf{S}_k^B)}{w(\mathbf{S}_k^B|\mathbf{S}_k^A)}, \quad (11)$$

where the product is restricted to the flipping clusters, since the corresponding terms for the other clusters do not depend on the configuration (for a fixed bond distribution). Considering Eq. (5) we get

$$X_{B|A} = \frac{\exp[-W_f^A - W_{ff}^A - \frac{1}{2}W_{fn}^A - \frac{1}{2}W_{fn}^A]}{\exp[-W_f^B - W_{ff}^B - \frac{1}{2}W_{fn}^B - \frac{1}{2}W_{fn}^B]}, \quad (12)$$

with

$$W_f^\alpha = \beta \sum_k^{\#} U_1(\mathbf{S}_k^\alpha), \quad (13)$$

$$W_{ff}^\alpha = \beta \sum_k^{\#} \sum_{l>k}^{\#} U_2(\mathbf{S}_k^\alpha, \mathbf{S}_l^\alpha), \quad (14)$$

$$W_{fn}^\alpha = \beta \sum_k^{\#} \sum_m^* U_2(\mathbf{S}_k^\alpha, \mathbf{S}_m), \quad (15)$$

$$W_{fn}^\alpha = \beta \sum_k^{\#} \sum_m^* U_2(\mathbf{S}_k^\alpha, \bar{\mathbf{S}}_m), \quad (16)$$

with α being either A or B . Equation (10) can be rewritten as

$$\frac{\mathcal{P}(C_l|B)}{\mathcal{P}(C_l|A)} = \frac{\exp[-\frac{1}{2}W_{fn}^A - \frac{1}{2}W_{fn}^B]}{\exp[-\frac{1}{2}W_{fn}^B - \frac{1}{2}W_{fn}^A]}. \quad (17)$$

From Eqs. (12) and (17) we finally get

$$\frac{\mathcal{P}(C_l|B) v(B \rightarrow A|C_l)}{\mathcal{P}(C_l|A) v(A \rightarrow B|C_l)} = \frac{\exp[-W_f^A - W_{ff}^A - W_{fn}^A]}{\exp[-W_f^B - W_{ff}^B - W_{fn}^B]}. \quad (18)$$

Notice that in (18) the energy terms between brackets on the right hand side are precisely those that change in the transition. Therefore, we conclude that the procedure fulfills (6)–(8) and the sampling procedure over the two state per particle model is correct.

B. General procedure

The procedure of simulation of the continuous model is now straightforward. In order to generate a new configuration from the current one, we generate a set of N trial positions (using the *symmetric* standard methods^{1,2}) for the N particles of the system and apply the procedure detailed in the description of the two state per particle cluster simulation. In the new configuration of the system some of the particles will have changed their positions while others will remain fixed. Repeating the procedure from the new position(s) we can build our chain of configurations to sample the phase space of the continuous model.

III. SIMULATION DETAILS

As a benchmark system we have used a Lennard-Jones atomic fluid, in which the pair interaction potential takes the form

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]. \quad (19)$$

The interaction range was truncated using the minimum image criterion.¹ The simulations were performed at reduced temperature $k_B T/\epsilon = 1.0$ and reduced density $\rho\sigma^3 = 0.80$, using several system sizes, $N = 64, 108,$ and 256 , and considering different values for the maximum displacement parameter, δx_{\max} .¹

Four simulation algorithms have been used to analyze their respective efficiencies for the determination of equilibrium properties. The first algorithm is the standard Metropolis MC algorithm involving one particle trial displacements,¹ we will denote this procedure as 1-MC. The second algorithm is also based in the Metropolis MC algorithm, but with the simultaneous displacement of all particles when building a trial configuration. This algorithm will be referred to as N-MC. Finally we will use the cluster algorithms proposed above, which will be denoted as G-CMC (Glauber) and M-CMC (Metropolis), with flipping criteria given by Eq. (4).

The simulation lengths will be given in *sweeps*. A sweep is defined as N single-particle trial displacements in the 1-MC procedure, and as a trial configuration in the collective motion algorithms. In order to evaluate reliable efficiency estimations we have carried out very long simulation runs. Typically we have used $\sim 2 \times 10^6$ sweeps in 1-MC and cluster algorithms, and $\sim 8 \times 10^6$ sweeps for N-MC algorithm.

TABLE I. System size dependence of various thermodynamic properties for different simulation procedures. Error bars, computed as twice the standard deviation of the mean are shown between parentheses and given in units of the last quoted figure.

Algorithm	N	δx_{\max}	$\langle U \rangle / N\epsilon$	$p\sigma^3 / \epsilon$	c_v^{ex} / k_B
1-MC	64	0.16	-5.1271(9)	1.618(5)	0.876 (5)
1-MC	108	0.16	-5.3024(6)	1.334(3)	0.889 (5)
1-MC	256	0.16	-5.4362(5)	1.161(3)	0.889 (5)
M-CMC	64	0.10	-5.1257(12)	1.625(7)	0.877 (7)
M-CMC	108	0.10	-5.3031(9)	1.332(5)	0.883 (7)
M-CMC	256	0.10	-5.4361(6)	1.162(3)	0.881 (6)
G-CMC	64	0.10	-5.1266(13)	1.619(7)	0.874 (8)
G-CMC	108	0.10	-5.3023(11)	1.336(6)	0.883 (8)
G-CMC	256	0.10	-5.4364(8)	1.161(4)	0.891 (7)
N-MC	64	0.015	-5.122(4)	1.643(19)	0.891(19)
N-MC	108	0.015	-5.299(4)	1.356(18)	0.88 (3)
N-MC	256	0.010	-5.435(4)	1.167(18)	0.87 (5)

The averages were performed after discarding the first $\sim 10^5$ sweeps.

In the MC runs we compute the potential energy, U , pressure p , and excess contribution to the heat capacity at constant volume (per particle) c_v^{ex} which is related with the fluctuations of the potential energy in the canonical ensemble,¹

$$c_v^{\text{ex}} = \frac{\delta U^2}{Nk_B T^2}, \quad (20)$$

where δA^2 stands for the fluctuations of the quantity A ,

$$\delta A^2 = \langle A^2 \rangle - \langle A \rangle^2. \quad (21)$$

As expected, the results for the different equilibrium properties do not depend on the simulation algorithm. In Table I we present the results for the system sizes considered. The N dependence of U and p is due to the truncation scheme.

The efficiency of a given MC simulation procedure depends on the computational effort required to generate a certain number of *effectively* independent samples of the system. The standard simulation procedures of condensed systems generate a sequence of correlated configurations. Therefore, it is required to estimate the CPU time per sweep and the correlation *time*, τ , of the different simulation techniques in order to compare their performances. Roughly speaking, τ provides a measure of the number of simulation sweeps between effectively independent configurations. In order to evaluate τ we compute the averages of the potential energy along blocks of \mathcal{N}_b sweeps, $U_b(\mathcal{N}_b)$. Within one simulation run, the results of the statistical analysis of the values of U_b for different block lengths furnish a reliable estimate of τ as

$$\tau \approx \mathcal{N}_b \frac{\delta U_b^2}{\delta U^2}. \quad (22)$$

For large blocks we expect¹ the result given in (22) to reach a constant (within statistical uncertainties) value.

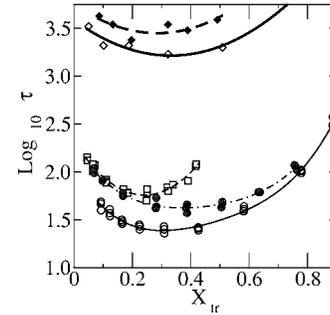


FIG. 1. Logarithm of correlation time τ (in MC sweeps) as a function of the average of the fraction of translated particles, X_{tr} per sweep. Empty circles and thin continuous line: 1-MC algorithm. Filled circles and dotted-dashed line: M-CMC algorithm. Squares and dashed lines: G-CMC algorithm. Empty diamonds and thick continuous line: N-MC algorithm for $N=64$. Filled diamonds and thick continuous line: N-MC algorithm for $N=108$. The results for 1-MC, G-CMC, and M-CMC include simulation of different system sizes: $N=64, 108$, and 256 , for these algorithm no significant dependence with the system size was found.

The analysis of results with different system sizes shows that for 1-MC, M-CMC, and G-CMC procedures the correlation time depends on δx_{\max} , but does not depend (within statistical error bars) on the system size. For the 1-MC algorithm the most efficient runs, $\tau \approx 24$, used $\delta x_{\max} / \sigma \approx 0.16$, with an average of the fraction of particles moved per sweep, $X_{\text{tr}} \approx 0.3$. For the G-CMC algorithm the best results ($\tau \approx 57$) were found for $\delta x_{\max} / \sigma \approx 0.10$, with $X_{\text{tr}} \approx 0.24$, whereas with the M-CMC procedure we get $\tau \approx 42$, ($\delta x_{\max} \approx 0.10$) with $X_{\text{tr}} \approx 0.37$. The results for the N-MC algorithm show, as expected,² large correlation lengths and a strong dependence of τ with the system size. In the best cases we found $\tau \sim 2 \times 10^3$ for $N=64$ and $\tau \sim 3.5 \times 10^3$ for $N=108$. The results for these simulation runs are collected in Fig. 1.

In addition, we have looked at the particle diffusion of the system. We have computed *ad hoc* self-diffusion coefficients D as¹

$$D(t) = \frac{1}{6t} \langle |\mathbf{r}_i(t_0 + t) - \mathbf{r}_i(t_0)|^2 \rangle, \quad (23)$$

where the time, t , is measured in sweeps. After a number of sweeps $D(t)$ reaches a constant value (with some statistical noise) D . When comparing the results for different simulation methods, larger values of D for fluid systems should imply more efficient simulation procedures (given similar computational costs).^{1,2} In Table II we present the results for the different algorithms with values of δx_{\max} chosen in each

TABLE II. *Ad hoc* diffusion coefficients for different simulation procedures. The results for 1-MC, M-CMC, and G-CMC do not show a significant system size dependence. The displacement parameter, δx_{\max} , is chosen in each case to maximize the algorithm efficiency.

Algorithm	N	$\delta x_{\max} / \sigma$	$10^4 \times D / (\sigma^2 \text{ sweep}^{-1})$
1-MC	64; 108; 256	0.16	2.7(2)
M-CMC	64; 108; 256	0.10	1.5(1)
G-CMC	64; 108; 256	0.10	1.0(1)
N-MC	64	0.015	0.030(1)
N-MC	108	0.015	0.017(1)
N-MC	256	0.010	0.007(1)

TABLE III. Mean number of bonds per particle and average of the fraction of particles moved per MC sweep in the cluster algorithms.

N	δx_{\max}	$\langle N_{\text{bonds}} \rangle / N$	$X_{\text{tr}}^{\text{G-CMC}}$	$X_{\text{tr}}^{\text{M-CMC}}$
64	0.04	0.044	0.442	0.755
108	0.04	0.044	0.445	0.760
108	0.06	0.096	0.389	0.635
108	0.08	0.160	0.321	0.508
108	0.10	0.234	0.250	0.387
108	0.15	0.437	0.111	0.168
108	0.20	0.648	0.046	0.069

case according to the best efficiencies reached in the previous tests. The results for the diffusion coefficients are consistent with the values of τ given before.

Let us now estimate the computational cost of a MC sweep in each algorithm. Firstly we will pay attention to the number of pair interactions, N_p , to be evaluated in one simulation sweep, which is expected to be the most time consuming task in the programs. The N-MC procedure will require the lowest number of pair interaction evaluations: $N_p \approx N^2/2$ in our case and $N_p \approx Nn_n/2$ (with n_n being the typical number of particles within the interaction range of a given one) if the potential range is much smaller than the simulation box. In the case of 1-MC algorithm encoded using the usual approach (i.e., evaluating the interaction of the moving particle in both the current and trial positions) we will have $N_p \approx 2N^2$ or $N_p \approx 2Nn_n$ depending, as above, on the potential range. Finally in the cluster methods, as applied in the program used in this work, one has $N_p \approx 2N^2$ (or $N_p \approx 2Nn_n$), since in each sweep one has to compute the interactions between two sets of coordinates (original and perturbed). Using N-MC as a reference, we must take into account other tasks (cluster treatment, random number generator calls, etc....) that can contribute to increase the computational effort in 1-MC, G-CMC, and M-CMC algorithms. In the model we are dealing with, we have found that within our implementation the relative CPU times per sweep t_{CPU} were $t_{\text{CPU}}(1\text{-MC})/t_{\text{CPU}}(\text{N-MC}) \approx 5.0$ and

$t_{\text{CPU}}(\text{G-CMC, M-CMC})/t_{\text{CPU}}(\text{N-MC}) \approx 5.8$. Notice, however, that in the 1-MC algorithm one can save some computing time by storing, for each particle, the sum of interactions with the rest of the particles; within this strategy the old pair interaction energies only have to be calculated for the accepted steps in order to update the corresponding partial sums of energy interactions. Also, for the cluster MC algorithms one could store the value of the pair interactions in the current configuration, which corresponds to about one fourth of the pair interactions to be calculated at every sweep in the procedure that we have used. In any case, these improvements do not substantially modify the ratios of computational efficiency between the different methods.

In Table III we present a simple analysis of the cluster distribution. The results for the mean number of bonds suggest that, at the best performance conditions, $\delta x_{\max} \approx 0.10$, most of the particles are either independent or belong to clusters with a small number of particles. This fact has two main consequences: firstly the small size of the clusters makes possible the use of efficient parallelization schemes for sys-

tems with short range interactions (we will discuss this issue in the next section), and secondly, the extra computational effort needed to build up clusters and compute the intercluster interactions is modest and scales well with the system size. To some extent, the crucial feature that explains the efficiency of the cluster algorithms here proposed is the ability of forcing some pairs of particles—that in a N-MC trial configuration could otherwise overlap—to move in a correlated manner when applying cluster MC methods.

IV. DISCUSSION

Taking into account both the correlation length and the computational costs, we find that, in the present application, 1-MC and the cluster procedures (G-CMC and M-CMC) are much more cost effective than N-MC (even with relatively small systems). The single-particle algorithm provides the best performance, but the cluster methods can offer results with a similar precision at about twice the computing effort. This fact can make interesting the parallel implementations of the cluster techniques proposed in this work when the simulation of large systems and/or lengthy simulation runs are required (e.g., situations close to critical points, inhomogeneous systems, etc.). The consideration of simultaneous perturbation of the N particles of the system in a trial configuration helps to remove most of the problems of fulfilling detailed balance on a parallel MC scheme using domain decomposition strategies. One can then use the versatility of the MC simulation technique to build up efficient simulation procedures in situations where molecular dynamics procedures show an insufficient performance.

From a practical standpoint, parallelization can be implemented both at the level of the cluster construction [Eq. (2)] and cluster move acceptance/rejection criterion [Eq. (4)] as well. The test configuration might be generated on a master process and subsequently broadcast to the slave processes, or alternatively one could use a parallel random number generator so that this trial configuration is constructed cooperatively.¹⁴ Then, if the interaction is sufficiently short ranged (a typical requirement for an efficient parallelization unless a replicated data strategy is used) one can employ a domain decomposition scheme to construct the clusters, exchanging connectivity information between neighboring domains. From Table III, one sees the average cluster size is relatively small, which implies that most clusters will be entirely contained in a single domain or will span mostly a few neighboring domains (if located at the domain boundaries). If such is the case, the acceptance/rejection procedure can also be performed independently on each domain, as long as the required data (energies and connectivity) of neighboring clusters have been exchanged in advance. This problem is similar to the one found when implementing a domain decomposition scheme in parallel MD of chain molecules.¹⁵

V. CONCLUSIONS

We have presented an efficient, general purpose, cluster MC algorithm involving multiparticle moves. In our application example the cluster algorithm has shown an efficiency only slightly inferior to that of the standard single-particle

algorithms. This suggests that such an approach will be most useful for the simulation of complex systems that require a large computational effort. In this case a N -particle move strategy can be implemented in a parallel computing environment using similar strategies to those already developed for parallel MD. The cluster MC approach can be very fruitful in the study of those complex systems for which MD exhibits inherent limitations (low self-diffusion coefficient, compositional sampling, etc.).

The cluster algorithms presented in this work are restricted to additive pair potentials. In principle, this limitation makes them of little use when dealing with models containing many-body interactions. *Ab initio* simulations^{16,17} are perhaps the most significant N -body problems to which one would like to extend this approach. In these cases, first principles MD is considerably more cost effective than any MC method based on single particle sequential moves. Unfortunately, the extension of this work's cluster algorithms to deal with *ab initio* problems is far from straightforward. For these systems one then might resort to the "fuzzy" MC approach of Krajić and Hafner.¹¹ In some cases, however, it could be possible to exploit the advantages of MC techniques, if the Hamiltonian can be expressed as a sum of pair and many-body terms. For not too large systems and appropriate maximum displacement parameters, one should expect that the electronic band energy varies smoothly with the nuclei translations (as compared to the interparticle repulsion). In such a case, when a trial set of positions is chosen, we can adopt a similar strategy to the so-called early rejection scheme.² That is, we first consider the pair interaction in order to generate a new configuration using the cluster algorithm, then we will accept or reject the proposed configuration by performing a Metropolis acceptance test on the change of the *perturbation* energy. Such a possible scheme would make simulations easier since the calculation of forces will not be required (of course with the penalty of losing the information on the dynamics). Nevertheless, the efficiency of such procedure must be tested for particular quantum models, and this is beyond the scope of this work.

ACKNOWLEDGMENTS

The authors acknowledge support from the Dirección General de Investigación Científica y Técnica under Grant No. FIS2004-02954-C03-01 and the Dirección General de Universidades e Investigación de la Comunidad de Madrid under Grant No. S0505/ESP/0299, program MOSSNOHO-CM.

- ¹M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Clarendon, Oxford, 1986).
- ²D. Frenkel and B. Smit, *Understanding Molecular Simulation* (Academic, New York, 2002).
- ³G. Eisenhauer and K. Schwan, *J. Parallel Distrib. Comput.* **35**, 76 (1996); J. R. Stadler, R. Mikulla, and H.-R. Trebin, *Int. J. Mod. Phys. C* **8**, 1131 (1997); D. C. Rapaport, *The Art of Molecular Dynamics Simulation* (Cambridge University Press, Cambridge, 1995).
- ⁴K. Essenlink, L. D. J. C. Loyens, and B. Smit, *Phys. Rev. E* **51**, 1560 (1995).
- ⁵M. Athènes, *Phys. Rev. E* **66**, 016701 (2002).
- ⁶K. Binder and D. W. Hermann, in *Monte Carlo Simulation in Statistical Physics*, Springer Series in Solid-State Sciences Vol. 80, 3rd ed. (Springer, Heidelberg, 1997).
- ⁷R. H. Swendsen and J. S. Wang, *Phys. Rev. Lett.* **58**, 86 (1987).
- ⁸U. Wolf, *Phys. Rev. Lett.* **62**, 361 (1989).
- ⁹G. T. Barkema and M. E. J. Newman, in *New Monte Carlo Algorithms*, Monte Carlo Methods in Chemical Physics Vol. 105, edited by D. M. Ferguson, J. I. Siepmann, and D. G. Truhlar (InterScience, New York, 1999).
- ¹⁰D. W. Heermann and A. N. Burkitt, in *The Monte Carlo Method in Condensed Matter Physics*, Series Topics in Applied Physics Vol. 71, 2nd ed., edited by K. Binder (Springer, Heidelberg, 1995).
- ¹¹M. Krajić and J. Hafner, *Phys. Rev. Lett.* **74**, 5100 (1995).
- ¹²R. C. Ball, T. M. A. Fink, and N. E. Bowler, *Phys. Rev. Lett.* **91**, 030201 (2003).
- ¹³A. A. Barker, *Aust. J. Phys.* **18**, 119 (1965).
- ¹⁴S. Aluru, G. M. Prabju, and J. Gustaffson, *Parallel Comput.* **18**, 839 (1992).
- ¹⁵J. M. Ilnytskyi and M. R. Wilson, *Comput. Phys. Commun.* **148**, 43 (2002).
- ¹⁶D. Marx and J. Hutter, in *Ab Initio Molecular Dynamics: Theory and Implementation*, Modern Methods and Algorithms of Quantum Chemistry Vol. I, edited by J. Grotendorst (John von Neumann Institute for Computing, Jülich, 2000), pp. 301–449.
- ¹⁷C. M. Goringe, D. R. Bowler, and E. Hernández, *Rep. Prog. Phys.* **60**, 1447 (1997).