

USHER: An algorithm for particle insertion in dense fluids

R. Delgado-Buscalioni^{a)} and P. V. Coveney^{b)}

Centre for Computational Science, Department of Chemistry, University College London, 20 Gordon Street, London WC1H 0AJ, United Kingdom

(Received 12 February 2003; accepted 10 April 2003)

The insertion of solvent particles in molecular dynamics simulations of complex fluids is required in many situations involving open systems, but this challenging task has been scarcely explored in the literature. We propose a simple and fast algorithm (USHER) that inserts the new solvent particles at locations where the potential energy has the desired prespecified value. For instance, this value may be set equal to the system's excess energy per particle in such a way that the inserted particles are energetically indistinguishable from the other particles present. During the search for the insertion site, the USHER algorithm uses a steepest-descent iterator with a displacement whose magnitude is adapted to the local features of the energy landscape. The only adjustable parameter in the algorithm is the maximum displacement, and we show that its optimal value can be extracted from an analysis of the structure of the potential energy landscape. We present insertion tests in periodic and nonperiodic systems filled with a Lennard-Jones fluid whose density ranges from moderate to high values. © 2003 American Institute of Physics.

[DOI: 10.1063/1.1579475]

I. INTRODUCTION

Many dynamical processes of chemical, biological, and physical interest occur in open systems where matter and energy are exchanged with the surroundings. The main focus of attention has been in (grand canonical) Monte Carlo algorithms, which are particularly suited for the study of equilibrium states. However, more recently, there has been significant attention focused on molecular dynamics (MD) algorithms adapted for open systems. One of the biggest challenges faced in the investigation of such phenomena by MD simulation is the problem of the efficient insertion of solvent particles in dense liquids. Indeed, as the scope and scale of MD increases, a growing variety of applications and methods are in need of *on the fly* particle insertion algorithms (see Refs. 1–5 and references therein). Among those, a particularly relevant family of methods is hybrid schemes that couple the particle domain to an outer region described by continuum fluid dynamics.^{4,5} The present method was devised for use in such hybrid schemes, but we believe its application within molecular simulation may prove to be more widespread.

The problem of inserting a solvent molecule in a dense fluid is commonly encountered in grand canonical Monte Carlo (GCMC) methods—for instance, in Gibbs ensemble calculations for phase equilibria or evaluation of the chemical potential. A number of techniques have been proposed to overcome this problem (see Refs. 6 and 7 and references therein). For instance, cavity-biased procedures search for domains within the fluid with a small local value of number density, as these cavities are more susceptible to accommodate a new molecule. In doing so, a bias is introduced and,

according to the rules of MC simulation, the bias has to be precalculated and corrected so that the scheme adheres to detailed balance. Even so, recent comparisons show that if the molecules are smaller than the mean size of the cavities, the GCMC method is nearly 10 times faster than grand canonical molecular dynamics (GCMD).⁸ We believe that the insertion algorithm proposed here may be used to improve the efficiency of the GCMD schemes.

The acme of a particle insertion protocol for MD is one that, in just a few iterations, is able to place the new particle within the required subdomain of the simulation space at a site where the potential energy takes exactly the desired value. This last condition ensures that no extra energy is introduced into the system, and therefore such an insertion algorithm would not require thermostating after each insertion. Indeed, even for moderate liquid densities, these are difficult requirements to satisfy and the few insertion protocols proposed in the literature^{1,3} are far from fulfilling them.

For instance, Goodfellow *et al.*³ introduce solvent (water) molecules in the cavities of proteins to investigate their structural stability. Once the protein's cavities are found, the insertion protocol consists of several steps that involve operations over the whole system. Solvent molecules are introduced with arbitrary orientation and locations within the selected cavity. As a consequence, the energy of the system increases sharply after each solvent molecule insertion and, to allow its relaxation, 200 energy minimization steps of the whole system (protein+water) are then performed, followed by a 1-ps molecular dynamics simulation. This expensive insertion procedure, which involves substantial alterations of the microscopic dynamics, could be avoided if the solvent particle were initially introduced at the desired potential energy site.

The work by Ji, Cagin, and Pettitt, towards the grand canonical molecular dynamics method¹ (see also Ref. 2 for

^{a)}Electronic mail: R.Delgado-Buscalioni@ucl.ac.uk

^{b)}Electronic mail: P.V.Coveney@ucl.ac.uk

further development and applications), is an example of an open-system MD simulation which does take care of the potential energy at the insertion site. In their method, new particles contribute a fractional number to the total number of particles. These fractional or scaled particles must be inserted at positions where the potential energy is equal to that of the former (added or deleted) fractional particle. As explained in Ref. 1, the authors first use a grid method to slice the MD domain into a number of boxes that is the same as or a little larger than the total number of particles. The most favorable boxes (with the least number of neighbors) are selected as candidates to add the new particle. Then the new solvent candidates are placed within each of these boxes and 200 possible molecular orientations of these new solvent candidates are computed. For each box, the orientation that yields the potential energy closest to the desired value is chosen and a first steepest descent procedure of $O(10)$ steps follows. If this does not lead to any site with the desired insertion energy, they finally perform a much larger steepest-descent procedure with at least 100 steps on the most favorable box. As the authors acknowledge, this numerically expensive protocol still yields numerical errors that can disturb the system.¹

Our main concern is the insertion of solvent particles in the framework of a hybrid (particle–continuum) scheme. In recent work we proposed a hybrid scheme that is able to deal not only with momentum but also with mass and energy exchange between the continuum (C) and the particle regions (P).⁵ In particular, particles need to be inserted in the overlapping $C \rightarrow P$ regions where the C fluxes are imposed on the P domain. In a real liquid (with interacting potential energy), mass and energy exchanges are strongly coupled and we showed that, in order to balance the energy flux, the new particles have to be inserted at positions where the potential energy equals the value prescribed by the continuum domain. In that work⁵ we used a particle insertion algorithm (called USHER) which is able to tackle this task in a rather efficient way (see Ref. 5 for a brief description). Further research has led to an enhanced version of the USHER algorithm. Here we shall describe this new version of the USHER protocol and, for the sake of consistency, we shall briefly review the one presented in Ref. 5.

The rest of the article proceeds as follows: We first formulate the root-finding problem in Sec. II. In Sec. III we describe a reference scheme against which we compare the USHER algorithm and then the USHER protocol. Insertion tests are described in Sec. IV, and the results are discussed in Sec. V. In Sec. VI we present an analysis of the potential energy landscape that proves to be very useful for the optimization of the algorithm's parameters. Finally, conclusions and directions for future research on insertion algorithms are given in Sec. VII.

II. THE INSERTION PROBLEM

We consider a set of N particles inside a box of volume V which interacts via pair potentials $\mathcal{V}(r)$. At any instant the potential energy can be defined at any point \mathbf{r} by evaluating $U(\mathbf{r}) = \frac{1}{2} \sum_{i=0}^N \mathcal{V}(|\mathbf{r} - \mathbf{r}_i|)$. The force that a test particle would

feel at any point can be measured by $\mathbf{f}(\mathbf{r}) = -\nabla U(\mathbf{r})$. In this work, we consider a Lennard-Jones fluid whose interparticle potential $\mathcal{V}(r) = 4(r^{-12} - r^{-6})$ is written in the usual units of length (the effective radius σ) and energy (the potential well ϵ).

The objective of the algorithms presented below is to find a position \mathbf{r}_0 for which the potential energy equals a prescribed value U_0 ; therefore, $U(\mathbf{r}_0) = U_0$. In most practical situations a less stringent requirement needs to be fulfilled: namely, $\langle U(\mathbf{r}_0) \rangle = U_0$, where brackets denote an average over a certain (small) number of insertions.

Even for a simple system such as a Lennard-Jones fluid, the structure of the energy landscape is very complex, with large energy gradients and complicated energy isosurface shapes. A typical energy distribution along the whole space spreads over several orders of magnitude, but for the typical (moderate) temperatures usually considered in applications, the particles need to be placed at positions with extremely low energies compared with the range of the energy distribution. The result is that the mean specific excess energy resulting from the equation of state, $u_{\text{eos}} = U_{\text{eos}}(\rho, T)/N$, is a very low energy compared with the typical energies found at any arbitrary point of the space. As the fluid particle density increases, the situation worsens, reflecting the fact that particles tend to reside within deep potential wells. The relation of the chemical potential μ with density ρ shows clear evidence of this fact. At moderate densities the value of μ is close to u_{eos} , but above a certain density, μ steeply increases above u_{eos} , meaning that the typical energy needed to insert a particle becomes much larger than the mean potential energy per particle.

Therefore, if one needs to insert particles at positions with energies close to the mean excess energy per particle, u_{eos} , one needs to find extremely-low-energy sites, particularly in dense systems. The main problem to be faced is that the energy landscape presents many energy “holes” whose local minima range from intermediate to large energy values. Here, we define a “hole” as a region of space enclosed by an isosurface of energy in such a way that $\nabla U(\mathbf{r}) \cdot \mathbf{n} > 0$ at the hole surface, where \mathbf{n} is the (outward) normal surface vector. Usually, these holes act as traps for widely used energy-minimization algorithms based on standard steepest-descent or conjugate-gradient methods.⁹ As a matter of fact, we soon discovered that it was very inefficient to move downhill over the potential energy surface by means of any of the standard versions of the steepest-descent method used in molecular simulation (see, for instance, Ref. 9). The purpose of the present study is to present a steepest-descent-like iterative procedure that can avoid the intermediate-energy holes. To this end, the USHER algorithm does not rely on line minimization along the steepest-descent direction,¹⁰ but instead on a displacement size which is adapted on the fly, according to the local topology of the potential energy landscape. Another important advantage is the facile implementation of the USHER code.

III. DESCRIPTION OF THE ALGORITHMS

We shall now describe some general aspects of the problem of particle insertion and the common features of the algorithms concerned.

In any insertion procedure the first step is to place the new particle at a starting position $\mathbf{r}^{(0)}$. In all the tests presented in Sec. IV, we chose $\mathbf{r}^{(0)}$ at random. We also tried to select $\mathbf{r}^{(0)}$ according to a cavity-biased procedure (as in Ref. 1). As explained in Sec. I, this procedure incurs a number of operations of $O(N)$ prior to the insertion algorithm itself and we found that, when using the USHER algorithm presented below, it did not reduce the total number of iterations with respect to the (much cheaper) random choice.

During successive iterations, the iterator's position is moved according to the update rule, which, in general, may be a function of the mechanical quantities at the previous iteration, $\mathbf{r}^{(n+1)} = \mathbf{r}^{(n+1)}(\mathbf{r}^{(n)}, U^{(n)}, \mathbf{f}^{(n)})$. The search terminates if the new position $\mathbf{r}^{(n+1)}$ is a site with the desired energy. This is determined by the following condition

$$|\xi|^{(n+1)} < \xi_{\max}, \quad \text{with} \quad \xi^{(n+1)} \equiv \frac{U^{(n+1)} - U_0}{|U_0|}, \quad (1)$$

where ξ_{\max} is a predetermined parameter—namely, the half-width of the interval of the accepted energies around U_0 —and ξ is defined as the relative difference of the potential energy $U^{(n+1)}$ at the $(n+1)$ iteration with respect to the desired value U_0 .

Finally, once the new particle is correctly inserted, the force that it exerts on its neighbors is calculated and its velocity is also assigned. This velocity is drawn from a Maxwellian distribution with the desired temperature T and the desired mean velocity $\langle \mathbf{v} \rangle$:

$$P(\mathbf{v}) = \left(\frac{1}{2\pi mkT} \right)^{3/2} \exp\left(-\frac{m(\mathbf{v} - \langle \mathbf{v} \rangle)^2}{2mkT} \right). \quad (2)$$

While the algorithm is guiding a new particle to a correct location, the positions of all the other particles remain frozen. This means that one insertion iteration only involves the evaluation of the force on a single particle (that is, the force exerted by all the particles at the site $\mathbf{r}^{(n)}$).

The starting position determines whether the following iterations will have to be downhill (if $U^{(0)} > U_0$) or uphill (if $U^{(0)} < U_0$). A simple way to unify both cases in a single scheme is to rescale the potential energy as $U(\mathbf{r}) \rightarrow \text{sgn} U(\mathbf{r})$, where $\text{sgn} \equiv (U^{(0)} - U_0) / (|U^{(0)} - U_0|)$. By doing so, the forces $\mathbf{f} = -\nabla U$ are also redefined and, in particular, a case with $\text{sgn} = -1$ then implies that the redefined forces point uphill of the (unscaled) potential energy throughout the entire course of that particular particle insertion. In the following presentation we shall assume that the energy and force field are already rescaled as $\text{sgn} U(\mathbf{r})$ and $\text{sgn} \mathbf{f}$, so we will not explicitly include “sgn” in the equations.

During the iterative process, the algorithm will encounter three different situations which may require separate treatment (for instance, different update rules). We denote these situations as follows: *downhill* move $U^{(n+1)} < U^{(n)}$, *uphill* move $U^{(n+1)} > U^{(n)}$, and *confinement* $U^{(n+1)} < U_0 < U^{(n)}$. In an optimal insertion one expects to keep going

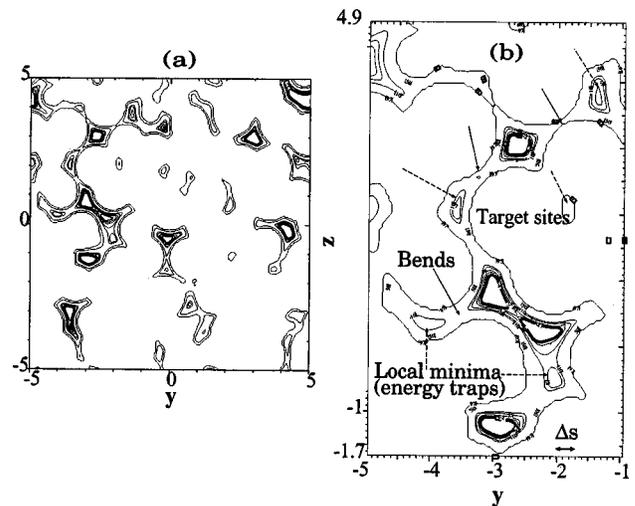


FIG. 1. (a) Cut along the $x=0$ plane of the contour plot of the potential energy landscape for energies lower than 100, showing the typical low-energy tubelike structures. (b) Close-up of the leftmost region, indicating with thicker solid lines a possible targeted energy, at $U = -4$. Some bends and saddle points of the energy surface and some energy traps (local minima with energy larger than the target) are indicated with solid and dashed arrows, respectively. The snapshot corresponds to a LJ fluid with $\rho=0.6$ and $T=2.5$ inside a 3D cubic box of side $L=10\sigma$. The size of the maximum displacement Δs used by the USHER algorithm for this density is also indicated.

downhill (with respect to the rescaled potential energy) until the confinement is attained. Then the desired location lies within the segment $\delta\mathbf{r}^{(n)} = \mathbf{r}^{(n+1)} - \mathbf{r}^{(n)}$ and can be determined by means of standard one-dimensional (1D) root-finding algorithms (such as the Newton–Raphson or bisection methods).

The most problematic iteration corresponds to the uphill move, and it merits some discussion. To illustrate the $U^{(n+1)} > U^{(n)}$ scenario we refer to the energy landscape shown in Fig. 1. Even for moderate densities (Fig. 1 corresponds to $\rho=0.6$), the low-energy regions conform to a complex tubelike structure. The insertion algorithm will have to usher the new particle into these energy tubes before arriving at a correct location. An uphill iteration may arise when the iterator faces either of two features of the energy landscape: intermediate-energy holes or sharp bends (including saddle points). Note that both kinds of features induce completely different decisions. The best thing to do when encountering an energy trap is to give up the search and restart from another initial position $\mathbf{r}^{(0)}$. By contrast, if a bend in the energy landscape leads to a low-energy valley, it may be worthwhile to use an update rule that can efficiently deflect the iterator's trajectory. Unfortunately, once an uphill move occurs, it is not possible to distinguish between these two features within only one iteration. On the other hand, the number of uphill iterations rapidly increases as the displacement $|\mathbf{r}^{(n+1)} - \mathbf{r}^{(n)}|$ is made larger than a specified maximum threshold. In fact, an important issue for the algorithm design is first to estimate this threshold and only then to determine the best decision to take upon an uphill move (see Sec. VI).

It is also convenient to introduce a restart condition in order to avoid any possible stagnation of the algorithm around energy holes. In particular, if $n > n_{\max}$, the search is

restarted from another initial position $\mathbf{r}^{(0)}$. Particularly at high densities (typically above 0.75), the overall number of iterations is sensitive to n_{\max} . A very large value of n_{\max} corresponds to many unsuccessful and time-consuming iterations, while a value for n_{\max} that is too small prevents most of the potentially successful trials from terminating successfully. We found that the best compromise between these two extremes is to make $n_{\max} \sim 0.8\langle n \rangle$, where $\langle n \rangle$ is the number of iterations averaged over a certain number (~ 20) of insertions. The value of n_{\max} depends on the density. It may be determined from an initial test run or, alternatively, reassigned on the fly according to the value of $\langle n \rangle$ determined during the simulation.

In the remainder of this section we first define a “reference” scheme against which we can then discuss and compare the USHER algorithm in more detail.

A. Reference scheme

In order to better understand the behavior of the USHER algorithm it is helpful to compare its performance with a reference scheme based on a combination of well-established methods widely used in the literature for root finding and energy minimization. While moving downhill, the reference scheme uses a basic steepest-descent step with a fixed displacement Δs_1 . The update rule is

$$\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} + \frac{\mathbf{f}^{(n)}}{f^{(n)}} \Delta s_1, \quad (3)$$

where, according to standard notation, $f^{(n)}$ is the modulus of $\mathbf{f}^{(n)}$.

If an uphill move is made, the reference scheme will first try to deflect the iterator’s trajectory in order to adapt itself to a possible bend in the potential energy surface. By construction of the update rule, Eq. (3), the potential energy decreases locally at $\mathbf{r}^{(n)}$ in the direction $\delta\mathbf{r}^{(n)} \equiv \mathbf{r}^{(n+1)} - \mathbf{r}^{(n)}$. Therefore, if $U^{(n+1)} > U^{(n)}$, there must exist a location \mathbf{r}_m where $U(\mathbf{r}_m) = \min\{U(\mathbf{r}_\lambda) | \mathbf{r}_\lambda = \mathbf{r}^{(n)} + \lambda \delta\mathbf{r}^{(n)}\}$. The reference scheme finds the position \mathbf{r}_m by means of a line minimization of the potential energy along the segment $\delta\mathbf{r}^{(n)}$ (see Ref. 10 for details). The new position is then recalculated by a steepest descent step starting from \mathbf{r}_m and with a displacement Δs_2 :

$$\mathbf{r}^{(n+1)} = \mathbf{r}_m + \frac{\mathbf{f}_m}{f_m} \Delta s_2. \quad (4)$$

The line minimization itself requires an inner iterative procedure (see Ref. 10). In view of the narrowness of the potential energy tubes, we used no more than three iterations for the estimation of \mathbf{r}_m . Better estimates of \mathbf{r}_m do not improve the efficiency, but instead lead to a larger number of force evaluations in the overall scheme. When a local minimum of potential energy is found, the iterator’s position will bounce back, moving subsequently upwards and downwards in energy as it hits the walls of the energy hole. To avoid this situation, after several (typically three) consecutive uphill iterations the scheme determines that it has been trapped in a local energy minimum and consequently restarts the search from another initial position $\mathbf{r}^{(0)}$. Finally, once the root has

been confined in the segment $\delta\mathbf{r}^{(n)} = \mathbf{r}^{(n+1)} - \mathbf{r}^{(n)}$, it is expressed as $\mathbf{r}_0 = \mathbf{r}^{(n+1)} + \lambda_0 \delta\mathbf{r}^{(n)}$, with λ_0 a real number in (0,1). To find λ_0 , the reference scheme uses a 1D root-finding algorithm which combines the Newton–Raphson method with a robust bisection method to ensure confinement in case of a failure of the Newton–Raphson step (due, for instance, to $f^{(n)} \sim 0$). It typically took less than three iterations to calculate the value of \mathbf{r}_0 . The optimum choice for the parameters Δs_1 and Δs_2 is presented in Sec. VI.

B. USHER scheme

The basic idea of the USHER insertion algorithm is to use an update rule to move downhill that can adapt the iterator’s displacement according to the local topology of the low-energy landscape. This is reflected in the update rule

$$\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} + \frac{\mathbf{f}^{(n)}}{f^{(n)}} \delta s^{(n)}. \quad (5)$$

Equation (5) is essentially a steepest-descent scheme with a displacement $\delta s^{(n)}$ that depends on the iterator’s position. The success of the method resides in a judicious choice of $\delta s^{(n)}$. Optimal performance was obtained using the following expression for $\delta s^{(n)}$, which depends on both the local potential energy $U^{(n)}$ and force $\mathbf{f}^{(n)}$:

$$\delta s^{(n)} = \begin{cases} \Delta s_{\text{ovlp}}, & \text{if } U^{(n)} > U_{\text{ovlp}}, \\ \min\left(\Delta s, \frac{U^{(n)} - U_0}{f^{(n)}}\right), & \text{if } U^{(n)} < U_{\text{ovlp}}. \end{cases} \quad (6)$$

The best way to illustrate how the adaptive displacement of Eq. (6) works is to describe how the USHER scheme performs one insertion. As long as the starting position $\mathbf{r}^{(0)}$ is chosen at random, there is a large chance of overlap with a preexisting particle, leading to a very large value of $U^{(0)}$. The displacement Δs_{ovlp} quoted in the first line of Eq. (6) can be constructed to remove the overlap in (typically) one iteration. For this reason, U_{ovlp} is chosen to be a very large energy representing an overlap position—say, $U_{\text{ovlp}} \sim 10^4$. As the hard-core part of the interparticle potential goes like $4r^{-12}$, the distance from a site with energy $U^{(n)} > U_{\text{ovlp}}$ to the center of the overlapped particle is $r = (4/U^{(n)})^{1/12}$. Therefore, by choosing $\Delta s_{\text{ovlp}} = r_\sigma - (4/U^{(n)})^{1/12}$, we can guarantee that the next iterator’s position $\mathbf{r}^{(n+1)}$ will be moved a distance r_σ away from the center of the overlapped particle and, by virtue of Eq. (5), in a direction of lower potential energy. The value of r_σ should be close to or slightly smaller than a characteristic contact distance between particles (e.g., the distance given by the maximum in the radial distribution function). For the pure Lennard-Jones fluid under consideration here, we have used $r_\sigma = 0.9$ (in units of σ).

Once any possible initial overlap is sorted out ($U < U_{\text{ovlp}}$), the second line of Eq. (6) is designed to drive the new particle downhill in energy, towards the target value U_0 . Here resides the main difference with respect to the reference scheme. At large energies, the typical slope of the potential energy is very large ($f^{(n)} \gg 1$), meaning that the energy drop along the steepest-descent direction is governed by the linear

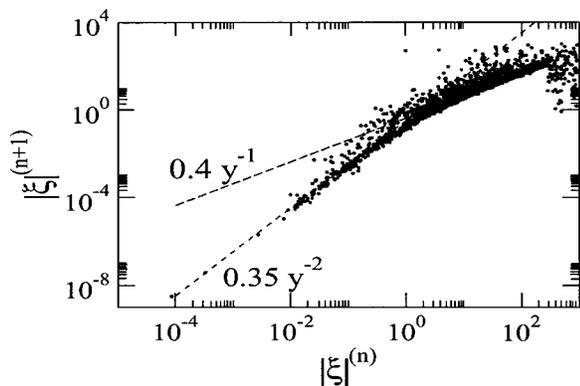


FIG. 2. Absolute value of the error at the $n+1$ iteration plotted against its value at the previous iteration n . The data correspond to insertions made by the USHER algorithm in a periodic box $L=10\sigma$ and for $0.6 \leq \rho \leq 0.75$. As illustrated by the dashed lines, the convergence is linear for $\xi > O(1)$ and quadratic for $\xi < O(1)$.

term of the Taylor expansion in the displacement, $\Delta U = f^{(n)}\delta s + O(\delta s^2)$. The second line of Eq. (6) makes use of this fact and takes $\Delta U = U^{(n)} - U_0$ for extracting a displacement adapted to the (maximum) local energy gradient $\delta s = \Delta U / f^{(n)}$. Note that at large energies $U^{(n)} - U_0 \sim U^{(n)}$, so after one iteration one expects the energy to decrease in (at least) a fraction of $U^{(n)}$, implying linear convergence. The local curvature of the potential energy landscape becomes dominant when approaching a local minimum ($f^{(n)} \sim 0$) and in this case Eq. (6) limits the displacement to a maximum value Δs . The maximum displacement is the only variable parameter in the algorithm and, as discussed in Sec. VI, its optimal value is about the width of the low-energy tubes of the potential energy landscape (see Fig. 1).

At low energies, as the iterator approaches the energy target U_0 , the displacement $\delta s = (U^{(n)} - U_0) / f^{(n)}$ behaves like a Newton–Raphson step made along the steepest-descent direction. Due to this feature, the convergence of the USHER algorithm increases notably near the target. In particular, this kind of displacement enables the error ξ to decrease quadratically once $\xi < O(1)$. This fact is illustrated in Fig. 2 by plotting the absolute value of the error $|\xi|^{(n+1)}$ against its value at the previous iteration $|\xi|^{(n)}$. As explained above, for $\xi > O(1)$ the algorithm converges linearly with $|\xi|^{(n+1)} \approx 0.4|\xi|^{(n)}$, while $|\xi|^{(n+1)} \approx 0.35(\xi^{(n)})^2$ once $\xi < O(1)$. In the same way, it may be possible to further increase the convergence rate by implementing a displacement based upon higher-order methods such as Halley’s or Bailey’s scheme¹¹ (for such purposes one would need to calculate the Hessian matrix and project it onto the steepest-descent direction).

For the sake of completeness, we also describe here the older version of the USHER displacement used in Ref. 5. This earlier version used a similar displacement rule for $U^{(n)} > U_{\text{ovlp}}$ to that quoted in Eq. (6), but for lower energies it used $\delta s^{(n)} = \min(\Delta s, \frac{1}{2}f^{(n)}\Delta t)$ where the optimal choice for the parameter Δt ranged within (0.05, 0.15). This scheme is around two times slower than the improved USHER scheme discussed here.

One of the important issues of the algorithm design con-

cerned the optimal strategy to deploy for uphill iterations. We compared two different strategies. The first one, which we shall call *indirect* USHER, performs a line minimization¹⁰ of the energy along the direction $\delta \mathbf{r}^{(n)} \equiv \mathbf{r}^{(n+1)} - \mathbf{r}^{(n)}$, similar to that described in Sec. III A and Eq. (4). The second alternative, called *direct* USHER, gives up the initial search and restarts a new one from another random position $\mathbf{r}^{(0)}$ once an uphill move is encountered. Interestingly, the insertion tests (see Sec. IV) clearly show that the *direct* USHER is about two times faster than the *indirect* version. This indicates that most of the uphill moves encountered using the update rule of Eqs. (5) and (6) are due to energy holes and therefore suggests that Eq. (6) enables the USHER algorithm to properly deflect its trajectory at most of the bends of the low-energy tubes encountered. A less restrictive version of the *direct* USHER allows a line minimization iteration only if the uphill move is done near enough to the target [for instance, if $|\xi| \leq O(1)$]. This alternative gives slightly better results at large densities.

In the insertion tests presented below in Sec. IV the reference scheme is compared with the most efficient version of the USHER algorithm—i.e., with the *direct* USHER. To avoid any possible confusion, in the remainder of the paper we shall simply call this the USHER algorithm.

IV. INSERTION TESTS

The insertion algorithms presented in Sec. III were evaluated in two kinds of systems: with and without periodic boundary conditions. We stress that no thermostat was used in any of the insertion tests. This ensures that the temperature of the system does not spuriously increase due to the dissipation of possible additional internal energy introduced by particle insertions in nonappropriate (higher-energy) locations.

In order to investigate the functioning of the insertion algorithm we shall drive the system through a specific thermodynamic process (see below) and compare the values of the thermodynamic variables computed during the simulations with those arising from thermodynamics. The system contains N particles within a volume V and its total energy is $E = 3NT/2 + U(\mathbf{r}^N)$, the energy per particle being $e = E/N$. The thermodynamic processes will be specified by the variation of the number of particles, ΔN (or density $\Delta \rho$), and the change of energy per particle, Δe . We now use standard thermodynamics to derive the changes in the system’s other variables.

The variation of energy per particle upon insertion of ΔN particles into the system is $\Delta e = \Delta E/N - e\Delta N/N$. For a system having no contact with a thermostat or a manostat, as for the one considered here, the variation of the total energy upon insertion of ΔN particles is exactly $\Delta E = \langle \epsilon' \rangle \Delta N$, where $\langle \epsilon' \rangle$ is the energy of the inserted particle averaged over ΔN insertions: $\Delta N \langle \epsilon' \rangle = \sum_i^{\Delta N} \epsilon'_i$ (ϵ'_i being the energy of the i th inserted particle). Thus,

$$\Delta e = (\langle \epsilon' \rangle - e)\Delta N/N = (\langle \epsilon' \rangle - e)\Delta \rho/\rho. \quad (7)$$

The variation of temperature can now be calculated from the equation

$$\Delta T = \frac{1}{c_v} \left[\Delta e - \left(\frac{\partial e}{\partial \rho} \right)_T \Delta \rho \right], \quad (8)$$

where $\Delta \rho = (\Delta N)/V$, c_v is the specific heat at constant volume, and $(\partial e / \partial \rho)_T$ is obtained from the equation of state for the excess energy per particle, $u = U/N$, reported by Johnson *et al.*¹²

Therefore, for given initial values of the system's density and temperature (ρ_0, T_0) , the time evolution of the thermodynamic variables is determined by the (specified) temporal variation of density $\partial \rho / \partial t$. The rate of temperature variation, obtained from Eq. (8), enables us to calculate the temperature at each instant in the process. The pressure and excess energy per particle can then be obtained from equations of state [$P = P_{\text{eos}}(\rho, T)$ and $u = u_{\text{eos}}(\rho, T)$].

In the tests presented below we considered a thermodynamic process in which the density increases at constant specific energy $\Delta e = 0$. According to Eq. (7), during the process the average energy of the inserted particles $\langle \epsilon' \rangle$ is set equal to the mean specific energy of the system, e . This condition is similar to that required for the energy balance conditions in the hybrid (particle–continuum) scheme of Delgado-Buscalioni and Coveney.⁵ In fact, the process with $\Delta e = 0$ can also be sought as a test for energy conservation in this hybrid scheme.

The thermodynamic relations, such as Eq. (8), are meaningful at least under condition of local equilibrium. This imposes a limit on the rate of particle insertion, because within each subdomain of the system the insertions of particles need to be sufficiently well spaced out in time for the system to be able to recover the equilibrium distribution. Consider a small subvolume of size λ , large enough to be representative of the system's distribution function. For instance, λ may be the distance at which the radial distribution function converges to one ($\sim 3\sigma$). To be sure that the system is able to restore its equilibrium distribution, the rate of particle insertion in each of these subdomains, $\lambda^3(\partial \rho / \partial t)$, has to be smaller than the inverse of the collision time, which for a simple fluid can be estimated using the hard-sphere approximation by $\tau_c \approx 0.14 \rho^{-1} T^{-1/2} (\sigma^2 m / \epsilon)^{1/2}$ (see, e.g., Ref. 13). In our calculations we used $(\partial \rho / \partial t) \leq 0.01$, so the characteristic insertion time was ~ 3 , much larger than the collision time $\tau_c \approx 0.3$.

A. Insertions in a periodic box

The first set of tests were performed in systems contained within a cubic periodic box of side lengths $L = \{7, 8, 10\}\sigma$. The initial density was set to a moderate value $\rho(t=0) = 0.4$ and was increased until $\rho = 1.0$. The maximum rate of density increase used was $\partial \rho / \partial t \sim 0.01$. The temperature, pressure, excess energy per particle, $u = U/N$, and total energy per particle, e , are plotted in Fig. 3 versus the density. Results correspond to particle insertions in a box with $L = 10\sigma$ at a constant density increase rate of $\partial \rho / \partial t = 0.01$. Particles were inserted at sites where the potential energy equals the specific excess energy of the system, $U_0 = U/N = u$, and with velocities drawn from a Maxwellian distribution at the instantaneous system's (kinetic) temperature T .

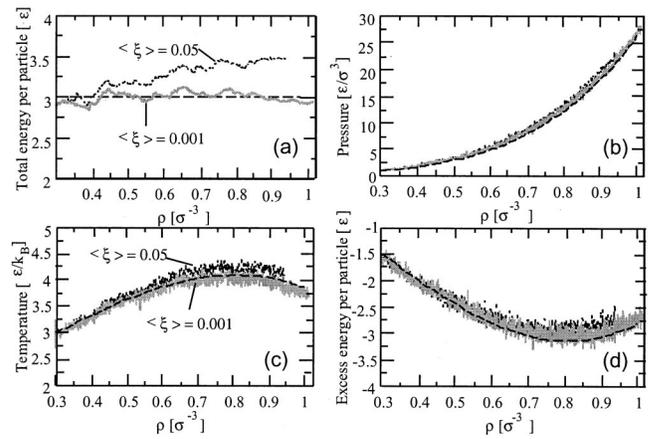


FIG. 3. (a) Total energy per particle, e , (b) pressure P , (c) temperature, and (d) excess energy per particle u vs the density, obtained in a particle insertion test made in a cubic periodic box with side length $L = 10\sigma$. The density increases linearly with time at a rate of $\partial \rho / \partial t = 0.01$. The insertions were made to guarantee a process with $\Delta e = 0$ (see text). The dashed lines correspond to the thermodynamic variables extracted from Eq. (8), using the equations of state for $u_{\text{eos}}(\rho, T)$ and $P_{\text{eos}}(\rho, T)$.

The dashed lines in Fig. 3 correspond to the thermodynamic variables obtained from the equation of state according to the process of Eq. (8).

B. Insertions in an open flow

The second test was done in open fluid flows—i.e., in systems with open boundary conditions. We considered a system with density N/V within a cuboidal domain of sides $L_x = 40\sigma$ and $L_y = L_z = 9\sigma$. The system is periodic in the y and z directions and has open boundaries at $x=0$ and $x=L_x$. Particles were inserted with potential energies close to the specific excess energy U_{eos}/N and with velocities drawn from a Maxwellian distribution at a temperature T_0 , which was fixed throughout the simulation. Insertions were done within a region of width Δx around $x=0$ and at a rate $A\rho v_{\text{in}}$, where v_{in} is a parameter that determines the flow velocity normal to the surface vector of the open boundary (\mathbf{n}) and $A = L_y L_z$ is the area of the boundary. At the right-hand boundary, particles are extracted at the same rate, so the overall density of the system remains constant throughout the simulation N/V . In order to couple the particle region to the outer pressure we used our hybrid particle–continuum scheme at the $x=0$ and $x=L_x$ surfaces.⁵

Note that in this case particle insertion in the x direction is restricted to a region Δx , which is set to $\Delta x = 2.0\sigma$ (the volume available to insert particles being $\Delta x L_y L_z$). To ensure that insertions are done in this region two different strategies were implemented. The first one is a simple reflection of the position $\mathbf{r}^{(n+1)}$ back to the insertion domain when the USHER iterator crosses the $x=0$ and $x=\Delta x$ boundaries. In the second implementation we imposed an artificial sharp potential well at $x=0$ and $x=\Delta x$, which acts only during the evaluation of the forces in the iteration procedure: i.e., it was not included in the evaluation of $U(\mathbf{r}^{(n+1)})$. Both alternatives worked equally well and resulted in a similar number of required iterations.

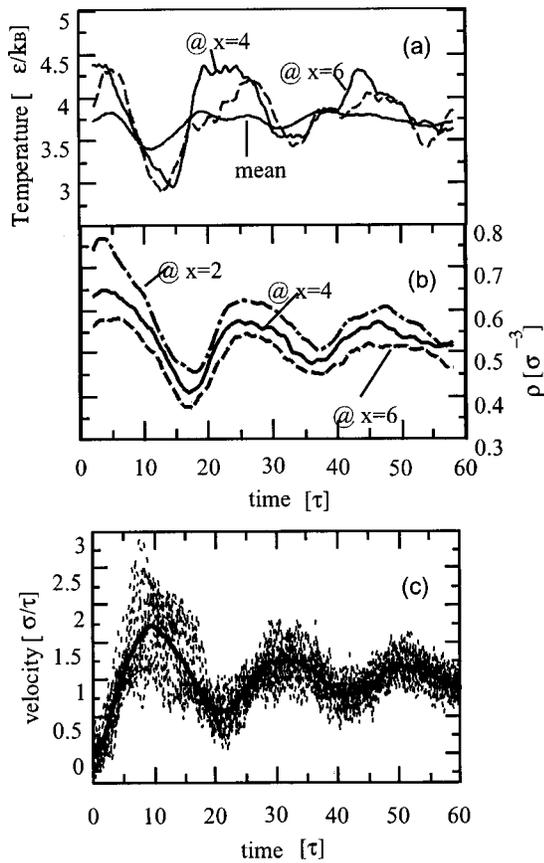


FIG. 4. Evolution in time of various hydrodynamic variables in an insertion test on an open flow in a “box” with sides $L_x=40\sigma$, $L_y=L_z=9\sigma$. Particles were inserted with zero mean velocity at $x=0$ at a rate $s=L_z L_y \rho_0 v_{in}$ and extracted at $x=L_x$ at the same rate. The overall density was $\rho_0=0.5$ and $v_{in}=1.0$. (a) Mean and local temperature and (b) local density at slices of width $\Delta x=2$; (c) the local velocity at each slice (dotted line) and the mean velocity (solid line) are shown. Flat stationary x profiles of density ($\rho=0.5$) and velocity ($v_x=v_{in}=1.0$) are reached after several sound transversal periods.

The simulation starts from an initial state with zero mean velocity and constant density profile along the x direction. As time goes by, the particle insertions concentrated in the region around $x=0$ lead to the production of a density wave that expands at the sound velocity for $x>0$. This density wave transports momentum along the x direction and, after a transient time, the density profile converges to the flat stationary density profile; throughout the simulation cell, the mean-flow velocity in the x -direction tends to the value v_{in} .

The hydrodynamic and thermodynamic variables were measured over slices of width Δx along the x direction. Figure 4 shows the local density at some of the leftmost slices $x < L_x/2$ together with the mean (slice averaged) velocity and total temperature of the system. The oscillatory behavior of the local density is a desired feature of these tests as it enables us to determine the dependence of the number of iterations, n , on the density for a range of values of ρ in each simulation. We refer to our previous paper⁵ for a detailed comparison between theoretical hydrodynamic trends and results obtained from hybrid continuum-particle simulations in different relaxing flows, also involving mass exchange.

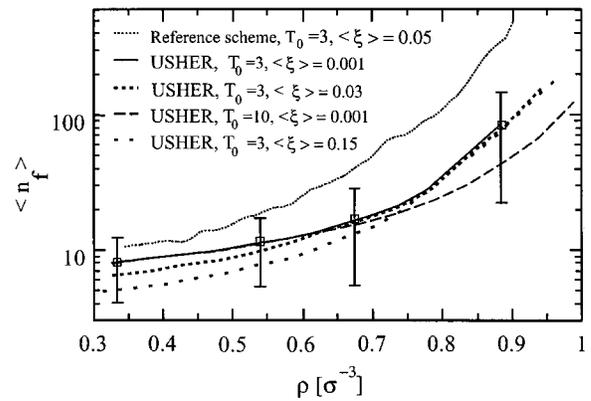


FIG. 5. Average number of force evaluations needed to insert a new particle, $\langle n_f \rangle$. The results correspond to insertions in a cubic periodic box of side $L=10\sigma$. For all the curves $\langle \xi \rangle$ indicates the maximum value of the averaged errors and the error bars corresponds to the standard deviation upon 100 insertions. The results are for processes with $\Delta e=0$, starting from an initial temperature T_0 [see Eq. (8) and Fig. 3].

V. RESULTS

Figure 5 presents the average number of single-force evaluations, $\langle n_f \rangle$, needed to perform an insertion versus the density. The most expensive part of the insertion algorithms is the evaluation of the force. Consequently, in order to compare the performance of the algorithms we have used n_f , rather than the total number of iterations, n . We note that some of the steps discussed in Sec. III (such as line minimization) require several subiterations and so $n \leq n_f$. The USHER and reference algorithms are compared in Fig. 5, in a test corresponding to insertions in a periodic box. In this kind of test (and for a similar average error $\langle \xi \rangle < 0.05$) the USHER algorithm is more than 2 times faster than the reference scheme for $\rho > 0.5$ and more than 4 times faster for $\rho > 0.8$. The reference scheme is slightly slower when insertions are constrained to a smaller region, as occurs in the open-fluid-flow tests. But notably, for both open-flow and periodic box tests, the USHER scheme gives similar values of $\langle n_f \rangle$. This means that the USHER algorithm does not pay any extra cost for restricting the size of the domain of insertion. This may be understood by looking at the distance between the initial trial and final insertion positions, $\Delta r = |\mathbf{r}^{(0)} - \mathbf{r}^{(n)}|$, shown in Fig. 6. For a wide range of densities the maximum value of

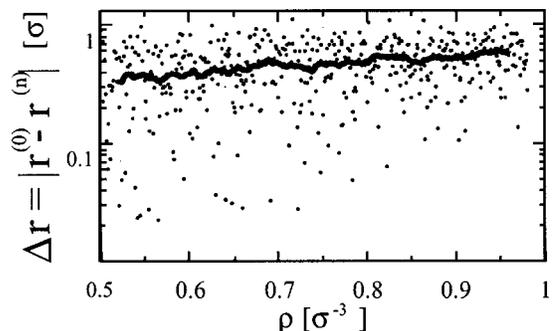


FIG. 6. Distance traveled by the USHER algorithm between the initial trial position and final insertion site, $\Delta r = |\mathbf{r}^{(0)} - \mathbf{r}^{(n)}|$ (on a logarithmic scale) vs the density. The test corresponds to particle insertions in a cubic periodic box of side $L=10\sigma$.

Δr is smaller than 1.0σ (its average being typically less than 0.5σ), indicating that most particles are inserted before the USHER iterator reaches the boundaries of the insertion domain. This feature is important for many applications. For instance, in the hybrid particle–continuum schemes the insertions are assigned (and restricted) to finite cells arising from a discretization of the space,^{4,5} and in the water-insertion method,³ the water molecules have to be placed in assigned protein cavities.

Figure 5 illustrates how the number of force evaluations varies with the maximum averaged error when using the USHER algorithm. In particular, we compare the results for the insertion tests done at an initial temperature of $T_0=3$ (as in Fig. 2) and with different values of the maximum error averaged over 30 insertions, $\langle \xi \rangle$. To decrease the error, from 0.15 to 0.03 one typically needs one more iteration. Another iteration leads to $\langle \xi \rangle = 10^{-3}$. This fast (quadratic) error reduction is made possible by the Newton–Raphson-like displacement implemented in Eq. (6) (see Fig. 2). A systematic nonvanishing value of $\langle \xi \rangle$ has a direct effect on the thermodynamic variables, as shown in Fig. 3. For instance, a value of $\langle \xi \rangle = 0.05$ maintained during the insertion process leads to a systematic drift from the $\Delta e = 0$ line and also has an effect on the temperature evolution.

Additionally, Fig. 5 illustrates how $\langle n_f \rangle$ varies with the system's temperature as shown in the data for $T_0=3$ and $T_0=10$. At larger temperatures it becomes much easier to insert particles once $\rho > 0.6$. The reason is that the target energy $U_0 [= u_{\text{eos}}(\rho, T)]$ increases much faster with the temperature at larger densities than it does at lower densities. For instance, $u_{\text{eos}}(0.4, 3) \approx -1.9$ and $u_{\text{eos}}(0.4, 10) \approx -1.2$, while for a larger density $u_{\text{eos}}(0.85, 3) \approx -3.1$ and $u_{\text{eos}}(0.85, 10) \approx -0.6$. For the same reason, if particles were inserted with potential energies similar to the chemical potential $\langle U_0 \rangle = \mu_{\text{eos}}(\rho, T)$, the slope of $\langle n_f \rangle$ with ρ would be flatter than those data shown in Fig. 5. We also performed insertions at subcritical temperatures $T < 1.3$, for liquid densities and also inside the liquid–vapor coexistence region. In these calculations the number of iterations needed to insert a particle was very similar to that presented in Fig. 5 for $T_0=3$. However, fluctuations of n_f were larger inside the coexistence region as a consequence of the inhomogeneity of the density field.

VI. CHOOSING AN OPTIMAL PARAMETER SET

We now wish to provide a physical interpretation of the performance of the particle insertion algorithms, based on the structure of the energy landscape. Such insight will be very useful for extensions of the USHER parameters to the simulation of other kinds of fluids. In fact, our experience is that, instead of a simple parametric study, it is advisable to perform an analysis of the structure of the potential energy landscape to obtain information about the typical shapes and length scales of the low-energy regions. This kind of structural analysis for the Lennard-Jones (LJ) fluid considered here not only provided important clues for the algorithm design, but provided the key relationship between the optimal displacement Δs and the density.

Low-energy holes

In order to investigate the structure of the low-energy holes we devised the following procedure. In a standard MD simulation in a periodic box and at time intervals separated by several collision times, we seek a point \mathbf{r}_0 with a very low prespecified energy U_0 . In particular U_0 is chosen to be the mean excess energy per particle. Initially, the search for the point \mathbf{r}_0 was done by the “basic” update rule of the USHER algorithm mentioned in Sec. III B. Once \mathbf{r}_0 was found, the energy landscape was probed in radial directions from this point. For each azimuthal angle $\psi \in [0, 2\pi]$ and longitudinal angle $\theta \in [0, \pi]$, the energy $U(\mathbf{r})$ was measured for increasing radial coordinate and a radial distance was recorded when $U(\mathbf{r}) \geq U_{\text{iso}}$. The radial distance will be denoted as $R(\psi, \theta)$. Note that $R(\psi, \theta)$ determines the shape of each hole; in particular, the mean radius and mean of the squared radius were computed for each hole:

$$\langle R \rangle_{\theta, \phi} \equiv \frac{1}{2\pi^2} \int_0^{2\pi} d\phi \int_0^\pi R(\theta, \phi) d\theta, \quad (9)$$

$$\langle R^2 \rangle_{\theta, \phi} \equiv \frac{1}{2\pi^2} \int_0^{2\pi} d\phi \int_0^\pi R^2(\theta, \phi) d\theta. \quad (10)$$

The effective shape of the hole can be estimated by the quantity

$$\sigma_R \equiv \left(\frac{\langle R^2 \rangle_{\theta, \phi}}{\langle R \rangle_{\theta, \phi}^2} - 1 \right)^{1/2}. \quad (11)$$

Clearly, for a sphere $\sigma_R = 0$, while σ_R is positive for any other elongated shape. A glance at the low- and intermediate-energy regions of a typical contour plot of the potential energy (see Fig. 1) suggests that it is possible to estimate the characteristic length scales of the low-energy regions by fitting σ_R and $\langle R \rangle_{\theta, \phi}$ to ellipsoids. In particular, due to the symmetry of the LJ fluid, it is enough to use asymmetric ellipsoids for this estimation. For an ellipsoid with semiminor and semimajor axes given, respectively, by R_s and $R_l = \chi R_s$, the following parametric relations fit within 1% to the exact analytical results:

$$\sigma_R = 0.56 \ln \chi, \quad (12)$$

$$\langle R \rangle_{\theta, \phi} = R_s (1 + 0.25 \ln \chi). \quad (13)$$

For given values of σ_R and $\langle R \rangle_{\theta, \phi}$, one can estimate the eccentricity $\chi = R_l/R_s$ and the semiminor axis R_s using Eqs. (12) and (13). The values of $\langle R \rangle_{\theta, \phi}$ and σ_R and the estimations of R_s and R_l , averaged over a set of about 80 holes, are shown in Fig. 7 versus the density. To ensure that these values are representative of the shape of low- and intermediate-energy regions (such as those shown in Fig. 1), the averages were obtained for a relatively wide range of (intermediate) energy isovalues $U_{\text{iso}} \in [0, 50]$. The error bars determine the maximum variation of these quantities for this range of U_{iso} .

In Fig. 7 we included the optimum choice for the reference algorithm described in Sec. III A (Δs_1 and Δs_2). The optimization of these parameters was performed independently, for a wide range of densities, $\rho \in [0.4–0.92]$ (see Fig. 8). It is interesting to note that Δs_1 closely follows the trend

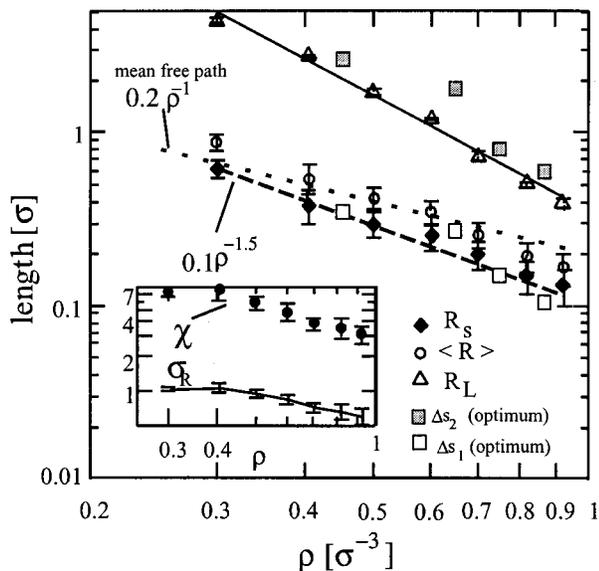


FIG. 7. Mean radius of the low-energy regions $\langle R \rangle_{\theta, \phi}$ [Eq. (9)] along with the smallest R_s and largest R_L characteristic lengths of the low-energy regions estimated by Eqs. (13) and (12). Squares correspond to the optimum values of the reference scheme Δs_1 , obtained from a parametric study. The dashed line ($0.1\rho^{-1.5}$) corresponds to our choice for the optimum USHER maximum displacement Δs . The mean free path (hard-sphere estimate $0.2\rho^{-1}$) is also shown. The inset shows the normalized variance σ_R given by Eq. (11) and the estimated mean eccentricity of the low-energy holes $\chi = R_L/R_s$.

obtained for the smallest effective radius R_s , while Δs_2 lies above the longest radius R_L of the intermediate energy regions. The interpretation of the results for Δs_1 seems quite evident, meaning that the displacement of the steepest-descent method when moving downhill should be about half the minimum characteristic diameter of the low-energy valleys. From this we readily understand why the best choice

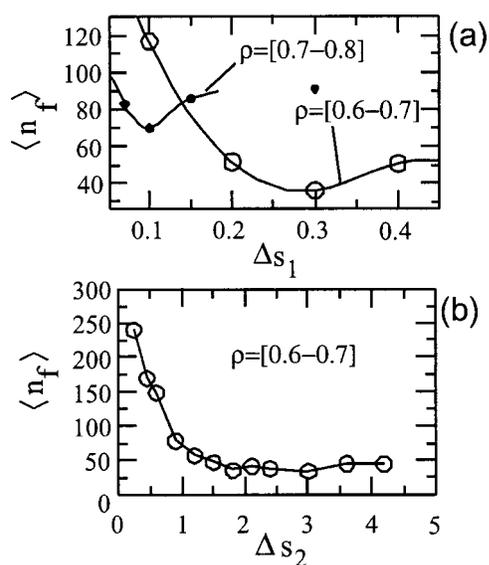


FIG. 8. Average number of force evaluations per insertion vs the parameters (a) Δs_1 and (b) Δs_2 of the reference scheme of Sec. III A. The evaluations were done by inserting particles in a cubic periodic box of side $L = 10\sigma$. The range of densities at which the evaluations were made is indicated in each figure.

for the maximum displacement of the USHER algorithm is $\Delta s \approx R_s \approx 0.1\rho^{-1.5}$. Quite remarkably, the estimate of the mean free path based on the hard-sphere fluid, $0.2\rho^{-1}$ (shown in Fig. 7), is close to the typical radius of the low-energy holes, $\langle R \rangle_{\theta, \phi}$. This indicates that such kinetic information, if available, may be of great help for the first adjustment of the maximum displacement of the algorithm, when inserting particles (or minimizing the energy) in other kinds of fluids.

Figure 8(b) sheds light on the interpretation of the result for Δs_2 . The optimal value of Δs_2 may be taken to be any value larger than a certain threshold, which according to Fig. 8(b) has to exceed the largest typical longest diameter within the low-energy regions. This confirms that once an uphill move is made, the fastest option is to completely traverse the energy valley and continue the iterations from a high-energy site, instead of trying to pursue possible further line minimizations. This conclusion, obtained from the reference scheme, suggested that the best procedure was to give up the search once $U^{(n)} > U^{(n-1)}$. As stated in Sec. III B, this is indeed what we have found when comparing the *direct* and *indirect* versions of the USHER scheme.

VII. CONCLUSIONS

An increasing number of methods involving molecular dynamics simulations of open systems⁴⁻⁷ require one to insert particles at precise locations where the potential energy is set equal to a prespecified value. Moreover, insertions need to be done on the fly and the performance of these methods will greatly depend on the efficiency of the insertion algorithm. At high densities this may seem a formidable task and indeed this sort of insertion algorithm has scarcely been explored in the literature. The main purpose of our paper is to show that this problem can be solved efficiently. To this end we have devised a particle insertion procedure called the USHER algorithm. To give an example, to insert a particle in a Lennard-Jones fluid with $\rho = 0.5$ and $T = 3.0$, at positions where the potential energy equals the mean specific energy of the system, the algorithm requires around eight extra evaluations of a (single-particle) force and 25 if $\rho = 0.8$.

The USHER algorithm essentially consists of a steepest-descent iteration procedure [see Eq. (5)] with a displacement adapted to the local shape of the energy landscape. In particular, by using an initial displacement which depends on the value of the potential energy at the initial trial position, $1 - (4/U^{(0)})^{1/2}$, the algorithm avoids in (about) one iteration any possible overlap with a preexisting particle. We confirmed that this feature makes it advantageous to choose the initial trial position at random, instead of using a much more expensive grid method to slice up the entire space in search of the less dense region (as is done in the cavity-biased Monte Carlo or in the grand canonical MD scheme proposed by Pettitt and co-workers^{1,2}). In subsequent iterations the displacement is given by the Newton-Raphson step measured along the steepest-descent direction and has an upper bound of Δs , to avoid uncontrolled jumps near local minima.

As another relevant conclusion, we wish to caution the reader about the usage of line minimization, normally implemented in conjunction with the steepest-descent method.^{9,10}

We clearly observed that, in these complex landscapes, it is better to use a (small enough) maximum displacement to ensure that most iterations are made downhill; then, if a single iteration is made uphill, the best option is to restart the search from another random position, rather than performing a line minimization. There are two reasons for this: first, line minimization is expensive and, second, and more importantly, we observed that if the maximum displacement is optimal, most of the uphill moves are due to the presence of an energy trap (i.e., a local minimum at an energy larger than the target).

An important part of this work was to give a physical interpretation of the optimal maximum displacement Δs . As stated earlier, it is essential to optimize its value for the complex topology of the low-energy tubes. To that end we analyzed the structure of the energy landscape of the Lennard-Jones fluid considered here and concluded that the width of low-energy tubes scales as $0.1\rho^{-1.5}$. A much more computationally expensive parametric study clearly showed that the optimal displacement follows a similar trend. This means that, in order to extend the insertion algorithm to other fluids, it is strongly advisable to first investigate the structure (in terms of shape and length scales) of the low-energy regions. Apart from this, our insertion protocol is based uniquely on mechanical variables (force and potential energy) readily calculated in any standard MD simulation: therefore, the same kind of protocol can be used for inserting solvent molecules in fluids consisting of (small) polyatomic molecules or even in polar fluids with nonadditive potentials. Nevertheless, the optimal algorithm design may require modifications, depending on the specific molecule considered. For instance, in fluids whose molecules have rotational degrees of freedom, the

insertion update step could be modified so as to first update the position of the center of mass of the molecule and then to use the local torque to orientate the molecule to the most favorable position. Such an investigation will form the subject of future work.

ACKNOWLEDGMENTS

We gratefully acknowledge useful conversations with E. Flekkoy and P. Español. R.D.-B. wishes to thank G. Ciccotti and R. Winkler for fruitful discussions and S. Saumitra for comments. This work is supported by the European Union through a Marie Curie Fellowship (Grant No. HPMF-CT-2001-01210) to R.D.-B. Support from Project No. BFM2002-0290 is also acknowledged.

- ¹Jie Ji, Tahir Cagin, and B. M. Pettitt, *J. Chem. Phys.* **96**, 1333 (1992).
- ²G. Lynch and B. M. Pettitt, *Chem. Phys.* **107**, 8594 (1997); **258**, 405 (2000).
- ³J. M. Goodfellow, M. Knaggs, M. A. Williams, and J. M. Thornton, *Faraday Discuss.* **103**, 339 (1996).
- ⁴E. G. Flekkø, G. Wagner, and J. Feder, *Europhys. Lett.* **52**, 271 (2000).
- ⁵R. Delgado-Buscalioni and P. V. Coveney, *Phys. Rev. E* **67**, 046704 (2003).
- ⁶M. Strnad and I. Nezbeda, *Mol. Simul.* **22**, 183 (1999).
- ⁷M. Mezei, *Mol. Simul.* **9**, 257 (1992).
- ⁸M. Mezei, *J. Chem. Phys.* **112**, 1059 (2000).
- ⁹A. R. Leach, *Molecular Modeling. Principles and Applications* (Addison-Wesley Longman, Essex, England, 1996).
- ¹⁰W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing* (Cambridge University Press, Cambridge, England, 1992).
- ¹¹F. B. Hildebrand, *Introduction to Numerical Analysis* (Dover, New York, 1987).
- ¹²K. Johnson, J. A. Zollweg, and K. E. Gubbins, *Mol. Phys.* **78**, 591 (1993).
- ¹³K. Huang, *Statistical Mechanics* (Wiley, Singapore, 1987).