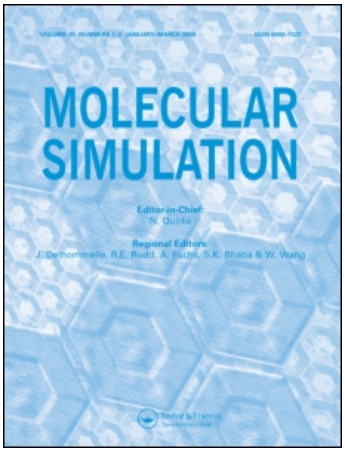


Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

Molecular Dynamics as a Mathematical Mapping. III. Efficient Evaluation of the Differentiable Force Functions and Their Derivatives

Jelena Stefanović[a]; Constantinos C. Pantelides[a]
[a] Centre for Process Systems Engineering, Imperial College of Science, Technology and Medicine, London, United Kingdom

To cite this Article Stefanović, Jelena and Pantelides, Constantinos C.(2001) 'Molecular Dynamics as a Mathematical Mapping. III. Efficient Evaluation of the Differentiable Force Functions and Their Derivatives', Molecular Simulation, 26: 5, 323 — 352
To link to this Article: DOI: 10.1080/08927020108023017
URL: http://dx.doi.org/10.1080/08927020108023017

Molecular Simulation, 2001, Vol. 26, pp. 323–352
Reprints available directly from the publisher
Photocopying permitted by license only

MOLECULAR DYNAMICS AS A MATHEMATICAL MAPPING. III. EFFICIENT EVALUATION OF THE DIFFERENTIABLE FORCE FUNCTIONS AND THEIR DERIVATIVES

JELENA STEFANOVIĆ and CONSTANTINOS C. PANTELIDES*

Centre for Process Systems Engineering, Imperial College of Science, Technology and Medicine, London SW7 2BY, United Kingdom

(Received May 2000; accepted May 2000)

The fully continuous and differentiable framework for performing molecular dynamics calculations introduced in parts I and II of this paper [1, 2] requires the evaluation of rather complex force functions and their spatial partial derivatives. This paper presents an efficient interpolation scheme for the evaluation of these quantities over a finite spatial domain.

The modified force function is approximated by a linear combination of Hermite cubic basis functions such that both the interpolant of the force and its spatial derivatives are continuous across the grid boundaries. In order to achieve better accuracy for a given grid size, a non-uniform rectilinear grid is constructed *via* iterative refinement procedure. The latter guarantees the accuracy of the force computed by interpolation within any specified tolerance $\varepsilon > 0$.

For many potential functions of practical interest, it is possible for polynomial interpolants to be constructed for parts of the force functions which are independent of the potential parameters and system density (the so-called "separable force functions"). In such cases, a single interpolation grid which is applicable for a wide range of potential parameters and system densities can be constructed *a priori*.

Keywords: Molecular dynamics; Force evaluation; Interpolation; Adaptive grids

1. INTRODUCTION

Part I of this paper [1] advocated the use of a modified interparticle force function that is almost everywhere continuous and differentiable with

*Corresponding author.

respect to particle positions. As demonstrated in part II of this paper [2], one advantage of the differentiability of this function is that it permits the accurate computation of partial derivatives of quantities computed by molecular dynamics with respect to input quantities and potential parameters. However, a significant disadvantage of the force function is its complexity. This is particularly important since typical molecular dynamics computations involve very large numbers of force evaluations.

The generalized form of the modified force function in the x-direction is:

$$\mathcal{F}^x(X,Y,Z) \equiv -\frac{1}{L}\sum_{k=-\infty}^{+\infty}\sum_{k'=-\infty}^{+\infty}\sum_{k''=-\infty}^{+\infty}\frac{X-k}{R_{kk'k''}}\frac{\partial U^{\mathrm{NB}}}{\partial R}\bigg|_{R_{ijkk'k''}} \quad \forall (X,Y,Z)\in\mathbb{R}^3\setminus\mathcal{N}^{[3]} \tag{1}$$

where U^{NB} is a general form of the non-bonded potential, which is a continuous and differentiable function of potential parameters and interparticle distance; L is the reference box size, and X, Y, Z are the normalized coordinate components of interparticle distance defined as:

$$X \equiv \frac{x_i - x_j}{L}, \quad Y \equiv \frac{y_i - y_j}{L}, \quad Z \equiv \frac{z_i - z_j}{L} \tag{2}$$

The corresponding normalized interparticle distance $R_{kk'k''}$ is given by:

$$R_{kk'k''} \equiv \sqrt{(X-k)^2 + (Y-k')^2 + (Z-k'')^2} \tag{3}$$

In our modified framework, arguments X, Y and Z can, in principle, take any real value since particles are allowed to move freely in space. However, as shown in part I of this paper [1] (Section 4, Properties I–III), the force at *any* point $(X,Y,Z)\in\mathbb{R}^3$ can be obtained from the value of the force at a point in the cube $[0,0.5]^3$. Moreover, the values of the forces in the y and z-directions can be computed using $\mathcal{F}^x(\cdot)$ (Section 4, Property IV).

Section 2 of this paper considers the computation of the modified force functions by direct evaluation of the triple summations in (1). The large computational requirements of such a task lead us to consider the use of interpolation methods. Section 3 examines the interpolation of functions in 3-dimensional space using Hermite cubic basis functions that ensure continuity of both the function interpolant and its partial derivatives at the interpolation grid boundaries. Section 4 presents practical algorithms for the construction and usage of non-uniform interpolation grids that can rigorously guarantee the accuracy of the interpolant within a specified tolerance error. Section 5 describes in detail our approach for the efficient computation of modified force functions and their derivatives, introducing the concept of separable force functions and examining the issue of the

required accuracy of interpolation. Section 6 presents some numerical results illustrating our approach. Finally, Section 7 concludes with some remarks both on the work presented here and, more generally, on the molecular dynamics framework established by this paper and its companions [1, 2].

2. DIRECT EVALUATION OF SUMMATIONS IN THE MODIFIED FORCE FUNCTIONS

Let us first consider the computation of the force function by directly evaluating the triple summation in Eq. (1). More specifically, we study the K-restricted modified force function that includes only a *finite* number of terms in the summation:

$$\mathcal{F}^{x,K}(X,Y,Z) \equiv -\frac{1}{L}\sum_{k=-K}^{+K}\sum_{k'=-K}^{+K}\sum_{k''=-K}^{+K}\frac{X-k}{R_{kk'k''}}\frac{\partial U^{\mathrm{NB}}}{\partial R}\bigg|_{R_{ijkk'k''}} \quad \forall (X,Y,Z)\in\mathbb{R}^3\setminus\mathcal{N}^{[3]} \tag{4}$$

We evaluate this interparticle force for bulk argon at a number of (X, Y, Z) points between 0 and 0.5 for different values of K. The non-bonded interaction potential is given by the Lennard–Jones model, and the box size L is taken to be 10σ. In Figure 1, we plot $\log_{10}|1-\mathcal{F}^x_K/\mathcal{F}^x_{100}|$ against K. We note the following:

1. All sums tend to the true limit as we include an increasing number of terms in the partial sum.[1]
2. The rate at which the summations converge depends on the values of X, Y and Z. Generally, convergence is faster for small values of X, Y, Z (*i.e.*, at small interparticle distances). In such cases, the first term of the summation (with $k=k'=k''$) is a strong repulsive contribution which dominates the other terms. On the other hand, as X, Y, Z approach the value of 0.5, we are in the region of the long attractive tail of the Lennard–Jones potential and the contributions of the higher order terms become more significant.
3. The rate at which the summations converge is generally fast, with the first 2–3 terms in each of the three summations usually being enough to achieve a relatively high accuracy. If more terms had to be included, techniques which improve the convergence of the series such as the

[1] We take the values with $K=100$ to be the "true" values of the force function given by Eq. (1).

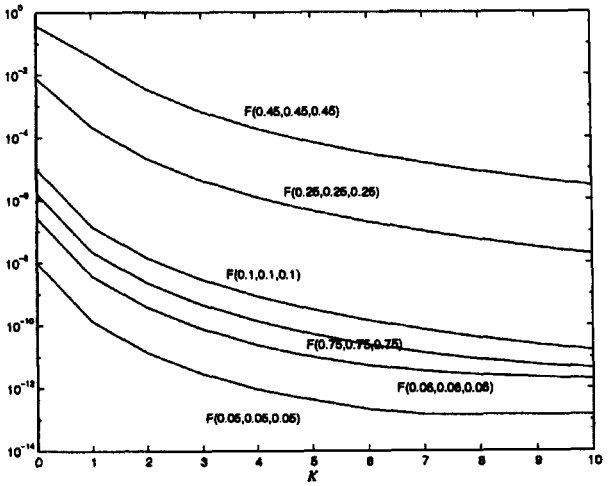


FIGURE 1 Modified force as a function of the summation range.

Shanks transformation or Richardson extrapolation could be applied [3,4]. However, the scope of benefiting from such techniques over 2–3 terms is very limited.

4. Even for $K=2$, the direct computation of the K-restricted force in Eq. (4) involves the evaluation of 125 ($=5^3$) terms. This makes the modified force function much more expensive than the corresponding conventional one which evaluates only one term per (X, Y, Z) component.

Overall, then, direct evaluation of partial sums of the form (4) is computationally unattractive. We are, therefore, led to consider interpolation schemes for the evaluation of force functions (1).

A number of interpolation schemes that aim to reduce the computational expense has been applied in work dealing with long-range (*e.g.*, polarization or Coulomb) forces in classical simulations [5,6] and in work combining quantum mechanical with classical simulations [7,8]. In our approach we develop a rigorous interpolation algorithm with the following objectives:

1. to preserve the continuous nature of the force and its partial spatial derivatives;
2. to achieve high accuracy for a given grid size;
3. to construct the interpolation grid in an efficient manner so that it can be used repeatedly for a wide range of potential parameters and system densities.

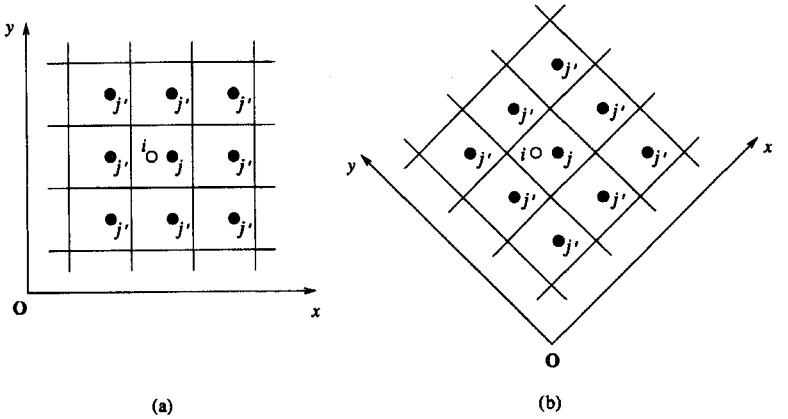


FIGURE 2 Potential energy as a function of coordinate axes orientation.

A peculiarity that needs to be addressed within the molecular dynamics framework developed in this series of papers is that the potential energy describing the interaction between a pair of particles is a function of the distance between the two particles along the 3 coordinate axes x, y and z rather than merely a function of their cartesian distance r. This point is illustrated in Figures 2(a) and (b), which show two particles i and j. The positions of the particles i and j with respect to the origin **O** are identical in both figures. However, the positions of the *images* of j (and, consequently, their interactions with i) depend on the orientation of the x and y axes, and the periodic partitioning that this orientation induces on the domain.

This property also implies that there is no advantage in constructing interpolants for potential energy functions $U(\cdot)$ rather than force functions $\mathcal{F}(\cdot)$ since both depend on *three* independent variables X, Y, Z. Therefore, we prefer to interpolate $\mathcal{F}(\cdot)$ since this is the quantity of direct interest to molecular dynamics computations.

3. INTERPOLATION OF FUNCTIONS IN 3-DIMENSIONAL SPACE USING HERMITE CUBIC BASIS FUNCTIONS

Interpolation methods attempt to strike a balance between accuracy on one hand, and speed of evaluation and storage requirements on the other. Generally, higher accuracy may be achieved by employing more complex approximating functions over finer grids. However, increasing function

complexity increases the computational cost, while increasing grid resolution leads to higher storage requirements – a trade-off that is dependent on the size of the system and the objective of the molecular dynamics computation.

Let us consider a general function $f(X, Y, Z)$ defined over a finite, rectangular domain:

$$\mathbb{D} \equiv \{(X, Y, Z) | X \in [X^L, X^U], Y \in [Y^L, Y^U], Z \in [Z^L, Z^U]\} \tag{5}$$

We wish to approximate this by an interpolant $\hat{f}(X, Y, Z)$ over a rectilinear 3-dimensional grid $\mathbb{G}$ of $n_x \times n_y \times n_z$ points defined as the Cartesian product of three 1-dimensional grids in X, Y and Z respectively:

$$\mathbb{G} \equiv \{X_{k_x}, k_x = 1, \ldots, n_x\} \times \{Y_{k_y}, k_y = 1, \ldots, n_y\} \times \{Z_{k_z}, k_z = 1, \ldots, n_z\} \tag{6}$$

where:

$$\begin{aligned} &X_1 = X^L; \quad X_{n_x} = X^U; \quad X_{k_x} < X_{k_x+1}, \quad k_x = 1, \ldots, n_x - 1 \\ &Y_1 = Y^L; \quad Y_{n_y} = Y^U; \quad Y_{k_y} < Y_{k_y+1}, \quad k_y = 1, \ldots, n_y - 1 \\ &Z_1 = Z^L; \quad Z_{n_z} = Z^U; \quad Z_{k_z} < Z_{k_z+1}, \quad k_z = 1, \ldots, n_z - 1 \end{aligned} \tag{7}$$

As an illustration, Figure 3 shows an example of a similar 2-dimensional non-uniform grid in X and Y with $n_x = 8$ and $n_y = 10$.

In constructing the interpolant $\hat{f}$, we seek to ensure that both $\hat{f}$ and its partial derivatives $\partial\hat{f}/\partial X$, $\partial\hat{f}/\partial Y$ and $\partial\hat{f}/\partial Z$ are continuous across the grid boundaries. This is essential in our case as the interparticle force and its spatial partial derivatives must be continuous functions of the interparticle distance (*cf.* parts I and II of this work [1, 2]). In order to achieve this continuity, we choose the interpolant so that it computes both force and its partial derivatives with respect to (X, Y, Z) *exactly* at each node of the grid:

$$\left.\begin{aligned} &\hat{f}(X_{k_x}, Y_{k_y}, Z_{k_z}) = f(X_{k_x}, Y_{k_y}, Z_{k_z}) \\ &\left.\frac{\partial\hat{f}}{\partial\gamma}\right|_{X_{k_x}, Y_{k_y}, Z_{k_z}} = \left.\frac{\partial f}{\partial\gamma}\right|_{X_{k_x}, Y_{k_y}, Z_{k_z}}, \quad \gamma = X, Y, Z \end{aligned}\right\} \quad \forall \begin{array}{l} k_x = 1, \ldots, n_x \\ k_y = 1, \ldots, n_y \\ k_z = 1, \ldots, n_z \end{array} \tag{8}$$

These conditions can be satisfied with relative ease if we approximate the force function in the kth element $[s_k, s_{k+1}]$ *via* a linear combination of

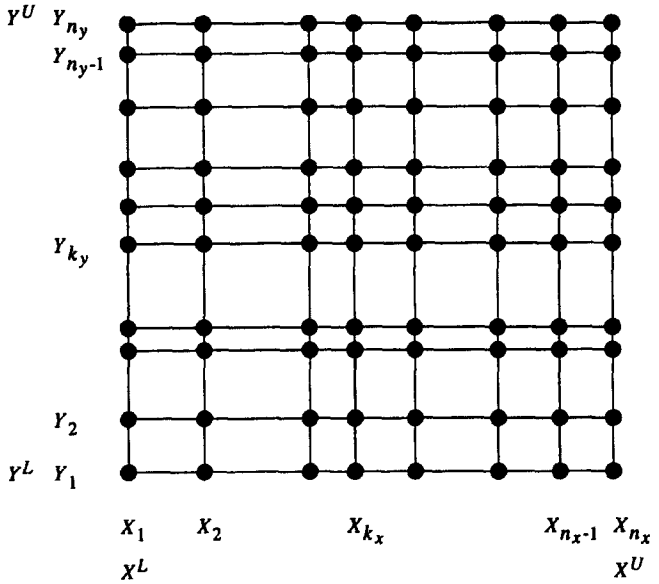


FIGURE 3 2-dimensional non-uniform grid.

Hermite cubic basis functions. This leads to the following cubic interpolant:

$$
\begin{aligned}
\hat{f}(X,Y,Z) = {} & H_1(u)[H_1(v)(H_1(w)f_{k_x,k_y,k_z} + H_3(w)f_{k_x,k_y,k_z+1}) \\
& + H_3(v)(H_1(w)f_{k_x,k_y+1,k_z} + H_3(w)f_{k_x,k_y+1,k_z+1})] \\
& + H_3(u)[H_1(v)(H_1(w)f_{k_x+1,k_y,k_z} + H_3(w)f_{k_x+1,k_y,k_z+1}) \\
& + H_3(v)(H_1(w)f_{k_x+1,k_y+1,k_z} + H_3(w)f_{k_x+1,k_y+1,k_z+1})] \\
& + H_2(u)[H_1(v)(H_1(w)f^{[x]}_{k_x,k_y,k_z} + H_3(w)f^{[x]}_{k_x,k_y,k_z+1}) \\
& + H_3(v)(H_1(w)f^{[x]}_{k_x,k_y+1,k_z} + H_3(w)f^{[x]}_{k_x,k_y+1,k_z+1})]h^x_{k_x} \\
& + H_4(u)[H_1(v)(H_1(w)f^{[x]}_{k_x+1,k_y,k_z} + H_3(w)f^{[x]}_{k_x+1,k_y,k_z+1}) \\
& + H_3(v)(H_1(w)f^{[x]}_{k_x+1,k_y+1,k_z} + H_3(w)f^{[x]}_{k_x+1,k_y+1,k_z+1})]h^x_{k_x} \\
& + H_2(v)[H_1(u)(H_1(w)f^{[y]}_{k_x,k_y,k_z} + H_3(w)f^{[y]}_{k_x,k_y,k_z+1}) \\
& + H_3(u)(H_1(w)f^{[y]}_{k_x+1,k_y,k_z} + H_3(w)f^{[y]}_{k_x+1,k_y,k_z+1})]h^y_{k_y} \\
& + H_4(v)[H_1(u)(H_1(w)f^{[y]}_{k_x,k_y+1,k_z} + H_3(w)f^{[y]}_{k_x,k_y+1,k_z+1}) \\
& + H_3(u)(H_1(w)f^{[y]}_{k_x+1,k_y+1,k_z} + H_3(w)f^{[y]}_{k_x+1,k_y+1,k_z+1})]h^y_{k_y}
\end{aligned}
$$

$$
\begin{aligned}
&+ H_2(w)[H_1(u)(H_1(v)f^{[z]}_{k_x,k_y,k_z} + H_3(v)f^{[z]}_{k_x,k_y+1,k_z}) \\
&\quad + H_3(u)(H_1(v)f^{[z]}_{k_x+1,k_y+1,k_z} + H_3(v)f^{[z]}_{k_x+1,k_y+1,k_z})]h^z_{k_z} \\
&+ H_4(w)[H_1(u)(H_1(v)f^{[z]}_{k_x,k_y,k_z+1} + H_3(v)f^{[y]}_{k_x,k_y+1,k_z+1}) \\
&\quad + H_3(u)(H_1(v)f^{[z]}_{k_x+1,k_y+1,k_z+1} + H_3(v)f^{[z]}_{k_x+1,k_y+1,k_z+1})]h^z_{k_z} \\
&\qquad \forall (k_x, k_y, k_z) \in [1, n_x - 1] \times [1, n_y - 1] \times [1, n_z - 1]; \\
&\qquad \forall (X, Y, Z) \in [X_{k_x}, X_{k_x+1}] \times [Y_{k_y}, Y_{k_y+1}] \times [Z_{k_z}, Z_{k_z+1}]
\end{aligned} \tag{9}
$$

defined on the rectilinear grid $\mathbb{G}$ (*cf.* Eq. (6)). Here u, v, w denote normalized positions within a cell in the grid:

$$
u \equiv \frac{X - X_{k_x}}{h^x_{k_x}}, \quad v \equiv \frac{Y - Y_{k_y}}{h^y_{k_y}}, \quad w \equiv \frac{Z - Z_{k_z}}{h^z_{k_z}} \tag{10}
$$

where

$$
h^x_{k_x} \equiv X_{k_x+1} - X_{k_x}, \quad h^y_{k_y} \equiv Y_{k_y+1} - Y_{k_y}, \quad h^z_{k_z} \equiv Z_{k_z+1} - Z_{k_z} \tag{11}
$$

The functions $H_i(\alpha)$, $i = 1, \ldots, 4$, $\alpha = u$, v, w are the cubic Hermitian polynomials:

$$
H_1(\alpha) \equiv (1 - \alpha)^2(1 + 2\alpha) \tag{12}
$$

$$
H_2(\alpha) \equiv \alpha(1 - \alpha)^2 \tag{13}
$$

$$
H_3(\alpha) \equiv \alpha^2(3 - 2\alpha) \tag{14}
$$

$$
H_4(\alpha) \equiv \alpha^2(\alpha - 1) \tag{15}
$$

The quantities f_{k_x,k_y,k_z} and $f^{[\gamma]}_{k_x,k_y,k_z}$ denote values of the exact function being interpolated and its partial derivatives at the grid nodes:

$$
f_{k_x,k_y,k_z} \equiv f(X_{k_x}, Y_{k_y}, Z_{k_z}) \quad \text{and} \quad f^{[\gamma]}_{k_x,k_y,k_z} \equiv \frac{\partial f}{\partial \gamma}(X_{k_x}, Y_{k_y}, Z_{k_z}) \tag{16}
$$

It can be shown [9] that the interpolant defined by Eqs. (9)–(16) is continuous at the grid boundaries, *i.e.*, on the faces that separate adjacent

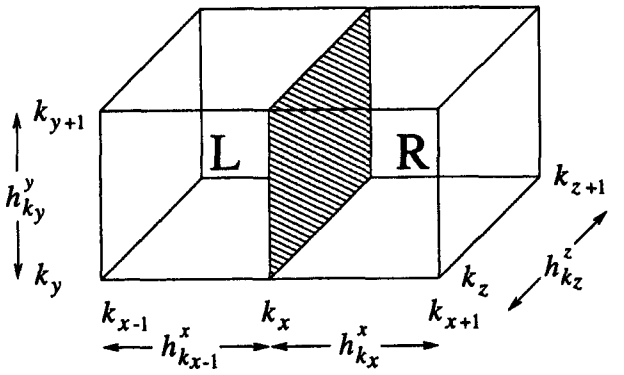


FIGURE 4 Continuity of interpolant and its derivatives on grid boundaries.

partitions defined by the grid $\mathbb{G}$. Consider, for example, the two grid cells shown in Figure 4 and the interface separating them (shown as the shaded area). Then, if $\hat{f}_L$ and $\hat{f}_R$ denote the interpolants in, respectively, the left and right cells of Figure 4, the following relations hold:

$$\left.\begin{aligned}
&\hat{f}_L(X_{k_x}, Y, Z) = \hat{f}_R(X_{k_x}, Y, Z)\\
&\frac{\partial \hat{f}_L}{\partial X}(X_{k_x}, Y, Z) = \frac{\partial \hat{f}_R}{\partial X}(X_{k_x}, Y, Z)\\
&\frac{\partial \hat{f}_L}{\partial Y}(X_{k_x}, Y, Z) = \frac{\partial \hat{f}_R}{\partial Y}(X_{k_x}, Y, Z)\\
&\frac{\partial \hat{f}_L}{\partial Z}(X_{k_x}, Y, Z) = \frac{\partial \hat{f}_R}{\partial Z}(X_{k_x}, Y, Z)
\end{aligned}\right\} \quad \forall (Y, Z) \in [Y_{k_y}, Y_{k_y+1}] \times [Z_{k_z}, Z_{k_z+1}] \qquad (17)$$

Therefore, the partial derivatives of the interpolant with respect to X, Y and Z are continuous everywhere at the interface.

4. GRID CONSTRUCTION AND USAGE

The previous section has considered how a function $f(X, Y, Z)$ defined over a finite rectangular domain $\mathbb{D}$ can be approximated by a cubic Hermitian interpolant $\hat{f}(X, Y, Z)$ defined over a given rectilinear grid $\mathbb{G}$. The accuracy of interpolation at any point (X, Y, Z) in the domain under consideration

can be measured in terms of the approximation error ε defined as:

$$\varepsilon(X,Y,Z) \equiv |f(X,Y,Z) - \hat{f}(X,Y,Z)| \quad \forall (X,Y,Z) \in \mathbb{D} \tag{18}$$

A question of practical importance is how to construct a grid such that the above error does not exceed a specified tolerance $\varepsilon^{\max}$ at any point in the domain of interest, *i.e.*,

$$\varepsilon(X,Y,Z) \leq \varepsilon^{\max}, \quad \forall (X,Y,Z) \in \mathbb{D} \tag{19}$$

4.1. Iterative Grid Refinement

We consider an iterative procedure that starts with an initial grid and refines it repeatedly until condition (19) is satisfied. The basic idea of this procedure is that, given a grid $\mathbb{G}$, it searches for a point $(X^*, Y^*, Z^*) \in \mathbb{D}$ at which the error ε is maximum. If $\varepsilon(X^*, Y^*, Z^*)$ does not exceed the tolerance $\varepsilon^{\max}$, the algorithm terminates as condition (5) is already satisfied. Otherwise, the point (X^*, Y^*, Z^*) is inserted in the current grid to derive a new grid $\mathbb{G}$. The procedure is then repeated.

In practice, the exact determination of the point $(X^*, Y^*, Z^*) \in \mathbb{D}$ that maximizes ε is, in itself, a difficult mathematical problem especially since we require the *global* maximum of ε in the domain $\mathbb{D}$. Consequently, here we prefer to obtain only an approximate solution by "sampling", *i.e.*, by evaluating ε at a (relatively large) number of points in $\mathbb{D}$ and simply selecting the point that corresponds to the largest value of ε.

More specifically, we choose N_s sampling points (u_s, v_s, w_s), $s = 1, \ldots, N_s$ in the unit cube (*i.e.*, $u_s, v_s, w_s \in [0, 1]$). We then evaluate ε at the following points:

$$(X_{k_x} + h^x_{k_x} u_s, Y_{k_y} + h^y_{k_y} v_s, Z_{k_z} + h^z_{k_z} w_s), \quad s = 1, \ldots, N_s \tag{20}$$

where $h^x_{k_x}$, $h^y_{k_y}$ and $h^z_{k_z}$ are given by Eq. (11), for $k_x = 1, \ldots, n_x$, $k_y = 1, \ldots, n_y$, $k_z = 1, \ldots, n_z$. As the grid generation procedure is executed once only, we can afford to use a large number of sampling points to ensure that the condition (19) is indeed satisfied by the final grid that will be employed by our molecular dynamics computations.

We now present our grid generation procedure as a formal algorithm. The key operations of the algorithm are also illustrated schematically in Figure 5 for the case of a 2-dimensional domain.

Given

- a finite domain $\mathbb{D} \subset \mathbb{R}^3$ defined by Eq. (5)
- a continuous and differentiable function $f : \mathbb{D} \to \mathbb{R}$
- a tolerance $\varepsilon^{\max}$
- a set of normalized sampling points $\{(u_s, v_s, w_s),\ s = 1, \ldots, N_s\}$ in the unit cube.

Step 0 Grid initialization

Define an initial grid $\mathbb{G} = \{X^L, X^U\} \times \{Y^L, Y^U\} \times \{Z^L, Z^U\}$ covering the entire domain $\mathbb{D}$ as a single partition, *i.e.*,

$$n_x = 2, X_1 = X^L, X_2 = X^U; n_y = 2, Y_1 = Y^L, Y_2 = Y^U;$$

$$n_z = 2, Z_1 = Z^L, Z_2 = Z^U$$

Step 1 Approximately locate point of maximum error

Set $\varepsilon^* := 0$
FOR $k_x := 1$ TO $n_x - 1$; $k_y := 1$ TO $n_y - 1$; $k_z := 1$ TO $n_z - 1$ DO
  FOR $s := 1$ TO N_s DO
    Set $X := X_{k_x} + h^x_{k_x} u_s$; $Y := Y_{k_y} + h^y_{k_y} v_s$; $Z := Z_{k_z} + h^z_{k_z} w_s$
    Evaluate function $f(X, Y, Z)$
    Evaluate interpolant $\hat{f}(X, Y, Z)$ using Eq. (9)
    Evaluate approximation error $\varepsilon = |f - \hat{f}|$ (*cf.* Eq. (18))
    IF $\varepsilon > \varepsilon^*$ THEN
      Set $\varepsilon^* := \varepsilon$
      Set $X^* := X$; $Y^* := Y$; $Z^* := Z$
      Set $k_x^* := k_x$; $k_y^* := k_y$; $k_z^* := k_z$
    END
  END
END

Step 2 Check for termination

IF $\varepsilon^* \leq \varepsilon^{\max}$ THEN stop: grid $\mathbb{G}$ satisfies condition (5)

Step 3 Refine grid by introducing new point (X^*, Y^*, Z^*)

$$\text{Set } \mathbb{G} := \{X_1, \ldots, X_{k_x^*}, X^*, X_{k_x^*+1}, \ldots, X_{n_x}\}$$
$$\times \{Y_1, \ldots, Y_{k_y^*}, Y^*, Y_{k_y^*+1}, \ldots, Y_{n_y}\}$$
$$\times \{Z_1, \ldots, Z_{k_z^*}, Z^*, Z_{k_z^*+1}, \ldots, Z_{n_z}\}$$

This is equivalent to:

- increasing n_x, n_y, n_z by 1
- shifting all grid points for which $k_x > k_x^*$, $k_y > k_y^*$, $k_z > k_z^*$ one position to the right:

$$X_{k_x} := X_{k_x-1} \quad \text{from} \quad k_x := n_x \quad \text{down to} \quad k_x := k_x^* + 2$$

$$Y_{k_y} := Y_{k_y-1} \quad \text{from} \quad k_y := n_y \quad \text{down to} \quad k_y := k_y^* + 2$$

$$Z_{k_z} := Z_{k_z-1} \quad \text{from} \quad k_z := n_z \quad \text{down to} \quad k_z := k_z^* + 2$$

- inserting the new point (X^*, Y^*, Z^*) in the grid

$X_{k_x+1} := X^*$; $Y_{k_y+1} := Y^*$; $Z_{k_z+1} := Z^*$

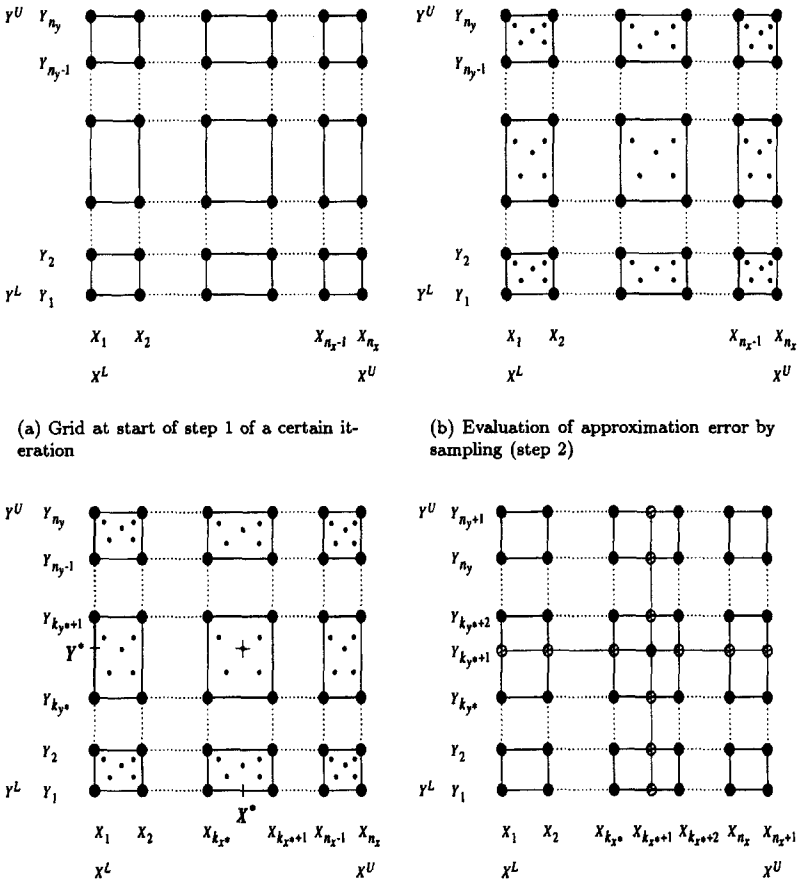
Step 4 Repeat from step 1

We note that the above algorithm takes direct account of the approximation error in locating the grid points. Consequently, the final grid is likely to be much more accurate than a uniform rectilinear grid with the same number of points. On the other hand, the grid is not necessarily optimal in the sense of being able to satisfy condition (5) with the *minimum* number of points. However, with the increasing availability of computer memory, obtaining the absolutely minimal number of grid points is not crucial, at least as far as storage demands are concerned. Of course, one also has to consider the efficiency of the utilization of the grid *during* the molecular dynamics computations. This issue is considered below.

4.2. Efficient Evaluation of the Interpolant

Section 4.1 has demonstrated how a rectilinear non-uniform grid $\mathbb{G}$ can be constructed over a finite domain $\mathbb{D} \subset \mathbb{R}^3$ so that a continuous and differentiable function $f : \mathbb{D} \to \mathbb{R}$ can be approximated using a restricted Hermitian interpolant to within an arbitrarily small positive tolerance $\varepsilon^{\max}$. For the class of applications of interest to this work, the interpolant will be evaluated a very large number of times; it is, therefore, important for each such evaluation to be as efficient as possible.

The evaluation of the interpolant $\hat{f}(X, Y, Z)$ at a given point $(X, Y, Z) \in \mathbb{D}$ involves two steps. These are described below.



(a) Grid at start of step 1 of a certain iteration

(b) Evaluation of approximation error by sampling (step 2)

(c) Approximate location of maximum error (end of step 2)

(d) Grid refinement (step 3)

FIGURE 5 Iterative grid construction algorithm.

4.2.1. Location of Point in Grid

In order to apply the correct interpolant function (Eq. (9)), the partition of the domain which contains a point (X, Y, Z) must be determined. This is equivalent to determining k_x, k_y, k_z such that:

$$X \in [X_{k_x}, X_{k_x+1}]; \quad Y \in [Y_{k_y}, Y_{k_y+1}]; \quad Z \in [Z_{k_z}, Z_{k_z+1}]$$

An efficient procedure for achieving this for a non-uniform grid is *binary search*. This exploits the fact that the grid points in each coordinate are

sorted in strictly ascending order (*cf.* Eq. (7)). The procedure is illustrated for the case of the X coordinate.

```
Given X:
1   Set k_L := 1
2   Set k_U := n_x
3   WHILE k_U − k_L > 1 DO
4     Set k_x := ⌊(k_L + k_U)/2⌋
5     IF X > X_{k_x} THEN
6       Set k_L := k_x
7     ELSE
8       Set k_U := k_x
9     END
10  END
```

The procedure maintains throughout two indices k_L and k_U such that:

$$X_{k_L} \leq X \leq X_{k_U}$$

These indices are initialized to 1 and n_x respectively (Steps 1 and 2). The algorithm then undertakes an iterative reduction of the interval of indices $[k_L, k_U]$ until $k_U = k_L + 1$ (Step 3). The interval reduction procedure bisects the index interval $[k_L, k_U]$ to obtain the index k_x (Step 4).[2] Depending on the value of X_{k_x} in relation to X, k_x is then used to replace either k_L or k_U (Steps 5–9).

Since the above search procedure bisects the index interval $[k_L, k_U]$ at each iteration, terminating when it becomes of length 1, the number of iterations required to locate any X within the given grid is $\lfloor \log_2 n_x \rfloor$, the cost per iteration being primarily one of an integer division (Step 4) and a real number comparison (Step 5). Since the procedures for the other coordinate directions are entirely analogous, the total number of iterations is simply:

$$\lfloor \log_2 n_x \rfloor + \lfloor \log_2 n_y \rfloor + \lfloor \log_2 n_z \rfloor$$

It should be noted that even for an exceedingly fine grid of $n_x = n_y = n_z = 1000$ (*i.e.*, 10^9 grid points), the above number is just 27.

[2]Note that if $k_U > k_L + 1$, then k_x will always be different to both k_L and k_U.

4.2.2. Evaluation of the Interpolant Function

Once we determine the partition of the domain $\mathbb{D}$ which contains the point of interest (X, Y, Z), we need to evaluate the interpolant $\hat{f}(X, Y, Z)$ for this particular partition (*cf.* Eq. (9)). This requires a sequence of operations as shown above.

	Floating point arithmetic operations		
	Add/subtract	*Multiply*	*Divide*
Evaluate normalized coordinates u, v, w (*cf.* Eq. (10))	3		3
Evaluate Hermitian polynomials H_1, H_2, H_3, H_4 at u, v, w (*cf.* Eqs. (12)–(15))	12	18	
Evaluate the interpolant f (*cf.* Eq. (9))	31	62	
Total arithmetic operations	46	80	3

Although the total number of arithmetic operations may appear to be large, it has to be remembered that this number is independent both of the complexity of the original function $f(X, Y, Z)$ and of the degree of accuracy of approximation required. Consider, for example, the modified force function of interest to this work. The evaluation of even the K-partial form of this function described by Eq. (4) involves the computation of $(2K+1)^3$ terms, each one of which can be quite complex. Even for quite modest K (*e.g.*, $K=1$), this is likely to be more expensive than evaluating the interpolant (9).

5. INTERPOLATION OF MODIFIED FORCE FUNCTIONS

Sections 3 and 4 have presented a general approach for constructing and using efficient interpolants of functions $f(X, Y, Z)$ defined over finite 3-dimensional domains $\mathbb{D}$. The approach is applicable to the modified force function $\mathcal{F}(X, Y, Z)$ since we have shown [1] that, irrespectively of the values that X, Y, Z take during a molecular dynamics computation, the force function needs to be evaluated only in the domain $\mathbb{D} = [0, 0.5]^3$. Moreover, the force components in the y- and z-directions can be computed in terms of the force component in the x-direction. Consequently, the interpolation grid needs to be constructed only for the function $\mathcal{F}^x(X, Y, Z)$ defined by Eq. (1).

Overall, then, it is possible to apply the generic methodology of Sections 3 and 4 directly to the force function $\mathcal{F}^x(X, Y, Z)$. However, some additional issues that are specific to this function and the types of application for which

molecular dynamics computations may be used or within which they can be embedded can be identified. These issues pertain to the frequency with which the interpolation grid needs to be computed.

More specifically, in addition to the particle positions, the modified force function $\mathcal{F}^x(\cdot)$ also depends on the reference box size, L, and the non-bonded potential parameters, ϑ. In the context of a "one-off" molecular dynamics simulation in the microcanonical ensemble, both L and ϑ are just constants and can, therefore, be ignored as far as the construction of an appropriate interpolant is concerned. However, such a simplification may not be possible in more complex situations. Examples include:

(a) The computation of partial derivatives of molecular dynamics mappings (*cf.* part II of this work [2]) requires the partial derivatives of the force function with respect to ϑ and/or L. Consequently, the dependence of the interpolant on these quantities needs to be made explicit.
(b) The molecular dynamics computation may be embedded within a procedure that attempts to estimate the parameters ϑ by matching the molecular dynamics predictions with actual experimental data. In this case, the molecular dynamics computation will have to be executed many times, each with a different set of values of ϑ. To generate an accurate interpolation grid at the start of each such computation is undesirable from the point of view of both computational efficiency and mathematical consistency.[3]
(c) In the context of molecular dynamics computations in the (N, P, T) ensemble, the reference box size L is actually a function of time; hence, it cannot be assumed to be a constant for the purposes of constructing the force function interpolant.

Ideally, we would like to be able to construct the polynomial interpolant for parts of the force function that are independent of both ϑ and L. This construction would be done once and for all, and the interpolants would be stored for future use by any application making use of this potential. In fact, as shown below, this is possible for some commonly used potential functions.

5.1. Interpolation of Separable Force Functions

In this section, we consider non-bonded potential functions $U^{\mathrm{NB}}(r, \vartheta)$ of such form that the quantity $\partial U^{\mathrm{NB}}/\partial R$ that appears in Eq. (1) can be

[3]For example, the use of a different grid at each invocation of the molecular dynamics computation may prevent the results of the latter from being a well-defined function of the value of ϑ.

written as:

$$\frac{\partial U^{\mathrm{NB}}}{\partial R} = \sum_{\kappa} A_{\kappa}(L, \vartheta) B_{\kappa}(R) \tag{21}$$

i.e., as the sum of one or more terms $\kappa = 1, 2, \ldots$, each of which can be expressed as the product of two factors; the first factor $A_{\kappa}(L, \vartheta)$ is *exclusively* a function of the reference box size L and potential parameters ϑ, while the second factor $B_{\kappa}(R)$ is *exclusively* a function of the normalized interparticle distance R.

The modified force function corresponding to (21) is given by:

$$\mathcal{F}^{x}(X, Y, Z, L, \vartheta) = -\frac{1}{L} \sum_{\kappa} A_{\kappa}(L, \vartheta) f_{\kappa}(X, Y, Z) \tag{22}$$

where $f_{\kappa}(X, Y, Z)$ is defined as:

$$f_{\kappa}(X, Y, Z) \equiv \sum_{k=-\infty}^{+\infty} \sum_{k'=-\infty}^{+\infty} \sum_{k''=-\infty}^{+\infty} \frac{X - k}{R_{kk'k''}} B(R_{kk'k''}), \quad \kappa = 1, 2, \ldots \tag{23}$$

Each of the functions $f_{\kappa}(X, Y, Z)$ can be approximated by a separate interpolant $\hat{f}_{\kappa}$. Thus, an overall interpolant $\hat{\mathcal{F}}^{x}$ for the modified force function can be obtained from:

$$\hat{\mathcal{F}}^{x}(X, Y, Z, L, \vartheta) = -\frac{1}{L} \sum_{\kappa} A_{\kappa}(L, \vartheta) \hat{f}_{\kappa}(X, Y, Z) \tag{24}$$

5.1.1. Practical Examples of Separable Force Functions

As an example, consider the Lennard–Jones potential:

$$U^{\mathrm{NB}}(r, \vartheta) = 4k_b\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^{6} \right] \tag{25}$$

In this case, the quantity $\partial U^{\mathrm{NB}}/\partial R$ can be written in the following form consistent with Eq. (21):

$$\frac{\partial U^{\mathrm{NB}}}{\partial R} = -\frac{48k_b\epsilon\sigma^{12}}{L^{12}} \times \frac{1}{R^{13}} + \frac{24k_b\epsilon\sigma^{6}}{L^{6}} \times \frac{1}{R^{7}} \tag{26}$$

In particular, we can see that there are two distinct terms ($\kappa = 1$ and $\kappa = 2$), with:

$$A_1(L, \vartheta) = -\frac{48 k_b \epsilon \sigma^{12}}{L^{12}}; \quad B_1(R) = 1/R^{13} \tag{27}$$

$$A_2(L, \vartheta) = \frac{24 k_b \epsilon \sigma^{6}}{L^{6}}; \quad B_2(R) = 1/R^{7} \tag{28}$$

By inserting the above expressions for $B_1(R)$ and $B_2(R)$ in Eq. (23), we obtain the two functions for which interpolants need to be constructed, defined as:

$$f_1(X, Y, Z) \equiv \sum_{k=-\infty}^{+\infty} \sum_{k'=-\infty}^{+\infty} \sum_{k''=-\infty}^{+\infty} \frac{X - k}{((X-k)^2 + (Y-k')^2 + (Z-k'')^2)^7} \tag{29}$$

$$f_2(X, Y, Z) \equiv \sum_{k=-\infty}^{+\infty} \sum_{k'=-\infty}^{+\infty} \sum_{k''=-\infty}^{+\infty} \frac{X - k}{((X-k)^2 + (Y-k')^2 + (Z-k'')^2)^4} \tag{30}$$

Another example of a separable force function is provided by the Coulomb potential of the form:

$$U(r) = \frac{qq'}{Dr} \tag{31}$$

where q and q' are the charges on the interacting particles, and D is the dielectric constant. In this case,

$$\frac{\partial U}{\partial R} = -\frac{qq'}{DL} \times \frac{1}{R^2} \tag{32}$$

hence $A_1(L, \vartheta) \equiv qq'/(DL)$ and $B_1(R) \equiv 1/R^2$ (*cf.* Eq. (21)).

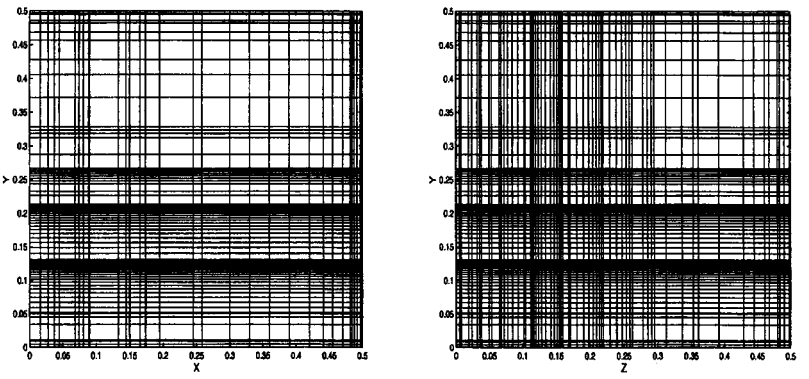
5.1.2. Error Control for Interpolation of Separable Force Functions

Returning to the general form of the interpolant (24), we note that the interpolants $\hat{f}_\kappa(X, Y, Z)$ can be constructed by applying the procedure of Section 3 *individually* to each one of the functions $f_\kappa(X, Y, Z)$ defined by Eq. (23). In this case, each interpolant $\hat{f}_\kappa(\cdot)$ will have its own interpolation grid which will generally be different to that of any other interpolant $\hat{f}'_\kappa$, $k' \neq k$. This may make it difficult to control the error in the overall interpolant (24).

Moreover, the amount of computer memory required for storing the interpolants will be proportional to the number of terms in Eq. (21). Also, there will be increased costs during the simulation as each evaluation of the force function at a given point (X, Y, Z) will involve locating this point in several different grids (*cf.* Section 4.2.1) and evaluating the Hermite polynomials at a different set of normalized positions u, v, w for each grid (*cf.* Section 4.2.2).

In order to illustrate the above points, Figures 6 and 7 present the interpolation grids constructed for the repulsive (29) and attractive (30) components of the Lennard–Jones force[4] for the same grid size of $75 \times 75 \times 75$. As can be seen by comparing these figures, the two grids are clearly different. Moreover, the accuracies of the corresponding interpolants are also different: the individual absolute interpolation errors are approximately 1.219 and 1.616×10^{-2} respectively.

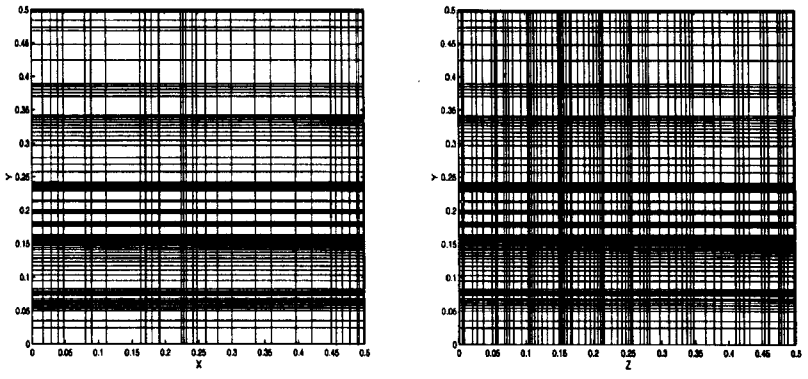
The alternative is to construct a *common* interpolation grid by considering all functions $f_\kappa(X, Y, Z)$ simultaneously. This can be achieved by modifying the definition of the approximation error ε (*cf.* Eq. (18)) to take account of



(a) X-Y coordinate plane (b) Z-Y coordinate plane

FIGURE 6 Interpolation grid for the repulsive Lennard–Jones force component.

[4]The "exact" force and its partial derivatives were computed using the K-restricted forms of Eqs. (29) and (30) with $K = 10$ (*cf.* Eq. (4)). The grid was constructed using $N_s = 729$ $(= 9^3)$ sampling points placed at normalized positions $u_s = 0.1i$, $v_s = 0.1i'$, $w_s = 0.1i''$, $\forall i, i' \in [1, \ldots, 9]$ (*cf.* Eq. (20)). For improved accuracy at very small interparticle distances, the central terms of the summations (*i.e.*, those for $k = k' = k'' = 0$) were omitted. Consequently, these terms will have to be computed separately during the molecular dynamics computations, to be added to the force values determined by the corresponding interpolants. This is similar to the approach used by Sangester and Dixon [6] in their work on interpolants for Ewald summation.



(a) X-Y coordinate plane (b) Z-Y coordinate plane

FIGURE 7 Interpolation grid for the attractive Lennard–Jones force component.

all parts of the force function. In particular, by subtracting Eq. (24) from Eq. (22), we obtain:

$$\mathcal{F}^x(X,Y,Z,L,\vartheta) - \hat{\mathcal{F}}^x(X,Y,Z,L,\vartheta) = -\frac{1}{L}\sum_{\kappa} A_\kappa(L,\vartheta)(f_\kappa(X,Y,Z) - \hat{f}_\kappa(X,Y,Z)) \tag{33}$$

Taking absolute values of the both sides of the above equation yields:

$$\varepsilon(X,Y,Z,L,\vartheta) \le \sum_{\kappa} \left|\frac{A_\kappa(L,\vartheta)}{L}\right| \varepsilon_\kappa(X,Y,Z) \tag{34}$$

where we have defined the overall error:

$$\varepsilon(X,Y,Z,L,\vartheta) \equiv \left|\mathcal{F}^x(X,Y,Z,L,\vartheta) - \hat{\mathcal{F}}^x(X,Y,Z,L,\vartheta)\right| \tag{35}$$

and the individual errors:

$$\varepsilon_\kappa(X,Y,Z) \equiv \left|f_\kappa(X,Y,Z) - \hat{f}_\kappa(X,Y,Z)\right|, \quad \kappa = 1,2,\ldots \tag{36}$$

Of course, both L and ϑ may vary in a particular application, and we need to ensure that $\varepsilon(X, Y, Z, L, \vartheta)$ remains acceptable for whatever value they may take. Therefore, we replace (34) by:

$$\varepsilon(X,Y,Z,L,\vartheta) \le \sum_{\kappa} C_\kappa \varepsilon_\kappa(X,Y,Z) \tag{37}$$

where we have introduced constants C_κ such that:

$$C_\kappa \equiv \max_{L,\vartheta} \left| \frac{A_\kappa(L,\vartheta)}{L} \right| \tag{38}$$

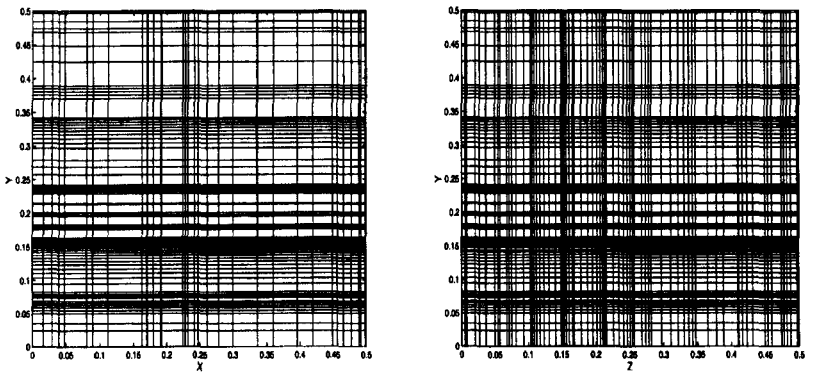
with the maximization being carried out over all allowable values of L and ϑ. For example, in the case of the Lennard–Jones potential considered earlier in this section, the constants C_κ are given by:

$$C_1 = \frac{48 k_b \epsilon_{\max} \sigma_{\max}^{12}}{L_{\min}^{13}} \quad \text{and} \quad C_2 = \frac{24 k_b \epsilon_{\max} \sigma_{\max}^{6}}{L_{\min}^{7}} \tag{39}$$

where $\epsilon_{\max}$ and $\sigma_{\max}$ are the maximum values of ϵ and σ, and $L_{\min}$ the minimum value of L that are under consideration.

The right hand side of Eq. (37) provides an upper bound on the overall error ε. If we employ this instead of Eq. (19) in the algorithm of Section 4, we can ensure that the value of the force function computed by Eq. (24) by combining the individual interpolants $\hat{f}_\kappa(\cdot)$ will always be accurate to (at least) the specified tolerance $\varepsilon^{\max}$.

Figure 8 shows the *combined* interpolant grid of size $75 \times 75 \times 75$ constructed for a Lennard–Jones fluid in a manner analogous to the grids shown in Figures 6 and 7. The quantities $\epsilon_{\max}$ and $\sigma_{\max}$ appearing in Eq. (39) were set at 120 K and 4 Å respectively, while $L_{\min}$ was set at 20 Å. Interestingly in this case, the combined interpolant grid is very similar to that obtained by the consideration of the attractive force term alone



(a) X-Y coordinate plane (b) Z-Y coordinate plane

FIGURE 8 Interpolation grid for the combined Lennard–Jones force components.

(*cf.* Fig. 7). This indicates that the dominant contribution (*i.e.*, the one that determines the placement of the grid points) to the interpolant error (*cf.* Eq. (37)) is that of the attractive term. This is a consequence of the fact that the central summation term ($k = k' = k'' = 0$) has been omitted from the computation of the interpolant (see footnote). Therefore, the longer range attractive forces dominate over the repulsive ones even for very small values of X, Y and Z.

For a given potential function $U^{\mathrm{NB}}(r)$, the interpolation grid and the interpolant coefficient (*cf.* Eq. (9)) for the individual functions f_κ can be computed once and for all and stored for future use. For example, the grid presented in Figure 8 ensures an interpolation error in the force function not exceeding $10^{-6} k_b \epsilon_{\max}/\sigma_{\max} \simeq 4.142 \times 10^{-18}$ N. Moreover, its rigorous theoretical basis guarantees *at least* this level of accuracy for all Lennard–Jones fluids with $\epsilon \leq \epsilon_{\max} = 120$ K and $\sigma \leq \sigma_{\max} = 4$ Å in boxes of size $L \geq L_{\min} = 20$ Å. Such systems include both argon ($\epsilon = 119.8$ K, $\sigma = 3.405$ Å) and all linear alkanes modeled according to the NERD potential [10], as well as their mixtures.[5] The minimum box size of $L_{\min} = 20$ Å guarantees that, for a system of, say, 256 molecules, the above accuracy is obtained for densities up to:

$$\rho_{\max} = \frac{256}{6.023 \times 10^{23} \times (20 \times 10^{-10})^3} = 5.3 \times 10^4 \text{ mol/m}^3$$

It is worth pointing out that, for many applications, it will be more efficient to combine the individual interpolant coefficients into a *single* set according to Eq. (24) *before* the start of the molecular dynamics computations. For example, in the case of molecular dynamics simulations in the (N, V, E) or (N, V, T) ensembles, both ϑ and L are given constants. Hence, we can calculate the quantities $A_\kappa(L, \vartheta)$, and then use them to pre-compute the interpolation coefficients for the overall force function (*cf.* Eq. (24)); this is straightforward since all interpolants $\hat{f}_\kappa(\cdot)$ share the same interpolation grid. We can then use these combined interpolation coefficients during the subsequent molecular dynamics simulation. This is clearly more efficient than using (24) directly within the simulation. On the other hand, such a pre-combination of the interpolants is not possible for molecular dynamics computations in the (N, P, T) ensemble as, in this case, L varies with time.

[5] This is true since the Lorentz–Berthelot combining rules always yield values for ε and σ that lie between the corresponding values for the pure species.

5.2. Computation of Partial Derivatives of the Modified Force Functions

As we have seen in part II of this work [2], in addition to the values of the modified force functions themselves, the computation of partial derivatives of molecular dynamics mappings also requires the values of the partial derivatives of the force function with respect to the positions X, Y, Z and the potential parameters θ. These partial derivatives are themselves quite complex functions and can, therefore, benefit from the application of interpolation techniques of the type considered in this chapter. The corresponding interpolants can be obtained in two different ways:

(a) by differentiating the interpolant of the modified force function with respect to the corresponding quantities; or,
(b) by constructing new interpolants for each of these partial derivatives.

Option (a) has the advantage that the values of derivatives obtained are exactly consistent with the values of the force functions used. It also implies that a single interpolation grid is sufficient for both the force function and its partial derivatives.

Option (b) has the advantage that the partial derivatives have the same degree of differentiability as the force functions themselves – and not one less as is the case for option (a). On the other hand, the values obtained may not be entirely consistent with the values of the force function approximated *via* the corresponding interpolant. In other words, consider a function $f(w)$ and its interpolant $\hat{f}(w)$, and also the partial derivatives $\partial f/\partial w$ and their interpolants $\widehat{\partial f/\partial w}$. Then, in general, interpolation and differentiation are not commutative operations, *i.e.*:

$$\frac{\partial \hat{f}}{\partial w} \neq \frac{\widehat{\partial f}}{\partial w}$$

Another disadvantage of option (b) is that multiple interpolation grids may be required.

In view of the above, we choose option (a). By differentiating (24) with respect to the required quantities (X, Y, Z, L, ϑ), we obtain the following expressions:

$$\frac{\partial \widehat{\mathcal{F}}^{X}}{\partial \gamma} = -\frac{1}{L} \sum_{\kappa} A_{\kappa}(L, \vartheta) \frac{\partial \hat{f}_{\kappa}}{\partial \gamma}, \quad \gamma \in \{X, Y, Z\} \tag{40}$$

$$\frac{\partial \widehat{\mathcal{F}}^x}{\partial \vartheta} = -\frac{1}{L} \sum_{\kappa} \frac{\partial A_\kappa}{\partial \vartheta} \hat{f}_\kappa(X, Y, Z) \tag{41}$$

$$\frac{\partial \widehat{\mathcal{F}}^x}{\partial L} = \frac{1}{L} \sum_{\kappa} \left(\frac{A_\kappa(L, \vartheta)}{L} - \frac{\partial A_\kappa}{\partial L} \right) \hat{f}_\kappa(X, Y, Z) \tag{42}$$

The partial derivatives $\partial \hat{f}_\kappa / \partial \gamma$ appearing on the right hand side of Eq. (40) can be determined by differentiating the interpolant given by Eq. (9) with respect to X, Y or Z and making use of the chain rule:

$$\frac{\partial \hat{f}_\kappa}{\partial X} = \frac{\partial \hat{f}_\kappa}{\partial u} \times \frac{\partial u}{\partial X} = \frac{1}{h^x_{k_x}} \frac{\partial \hat{f}_\kappa}{\partial u} \tag{43}$$

and similarly:

$$\frac{\partial \hat{f}_\kappa}{\partial Y} = \frac{1}{h^y_{k_y}} \frac{\partial \hat{f}_\kappa}{\partial v}, \quad \frac{\partial \hat{f}_\kappa}{\partial Z} = \frac{1}{h^z_{k_z}} \frac{\partial \hat{f}_\kappa}{\partial w} \tag{44}$$

In the specific example of the Lennard–Jones potential, the partial derivatives on the right hand side of Eqs. (41)–(42) take the following forms (*cf.* Eqs. (27)–(28)):

$$\frac{\partial A_1}{\partial \epsilon} = -\frac{48 k_b \sigma^{12}}{L^{12}} \quad \frac{\partial A_1}{\partial \sigma} = -\frac{576 k_b \epsilon \sigma^{11}}{L^{12}} \quad \frac{\partial A_1}{\partial L} = \frac{576 k_b \epsilon \sigma^{12}}{L^{13}} \tag{45}$$

$$\frac{\partial A_2}{\partial \epsilon} = \frac{24 k_b \sigma^6}{L^6} \quad \frac{\partial A_2}{\partial \sigma} = \frac{144 k_b \epsilon \sigma^5}{L^6} \quad \frac{\partial A_2}{\partial L} = -\frac{144 k_b \epsilon \sigma^6}{L^7} \tag{46}$$

5.3. Required Accuracy of Interpolation

As we have seen in Section 4, we can always construct an interpolation grid such that the force can be computed within a specified tolerance $\varepsilon^{\max} > 0$. An interesting question that arises in this context concerns the appropriate value for $\varepsilon^{\max}$. In fact, the same question arises more generally in the context of molecular dynamics whenever the force exerted on a particle is approximated in some sense (*e.g.*, *via* the use of cut-off distances). An answer can be obtained by considering the sensitivity of the computed outputs of the molecular dynamics computation with respect to this tolerance.

We consider a perturbed version of the Newton's equations of motion written as:

$$\dot{\boldsymbol{r}}_i = \boldsymbol{v}_i \quad \forall i = 1, \ldots, N \tag{47}$$

$$\dot{\boldsymbol{v}}_i = \frac{1}{m_i}(\boldsymbol{F}_i(\boldsymbol{r}, \alpha) + \varepsilon^{\max}) \quad \forall i = 1, \ldots, N \tag{48}$$

subject to:

$$\boldsymbol{r}(0) = \boldsymbol{r}^0(\alpha) \tag{49}$$

$$\boldsymbol{v}(0) = \boldsymbol{v}^0(\alpha) \tag{50}$$

We note that (48) recognizes explicitly the fact that the force $\boldsymbol{F}_i$ is not computed exactly but may be subject to an error $\varepsilon^{\max}$.

The thermodynamic property p of interest is given by time-integral expressions of the form:

$$p(\alpha, \varepsilon^{\max}) = \int_0^{t_f} \varphi(\boldsymbol{r}(t, \alpha, \varepsilon^{\max}), \boldsymbol{v}(t, \alpha, \varepsilon^{\max}), \alpha) dt \tag{51}$$

We note that the value of this property will also depend on the error in the force function since the particle positions and velocities will both be affected by $\varepsilon^{\max}$. To a first-order approximation, the prediction error Δp that is caused by $\varepsilon^{\max}$ is given by:

$$\Delta p \simeq \left.\frac{\partial p}{\partial \varepsilon^{\max}}\right|_{\varepsilon^{\max}=0} \varepsilon^{\max} \tag{52}$$

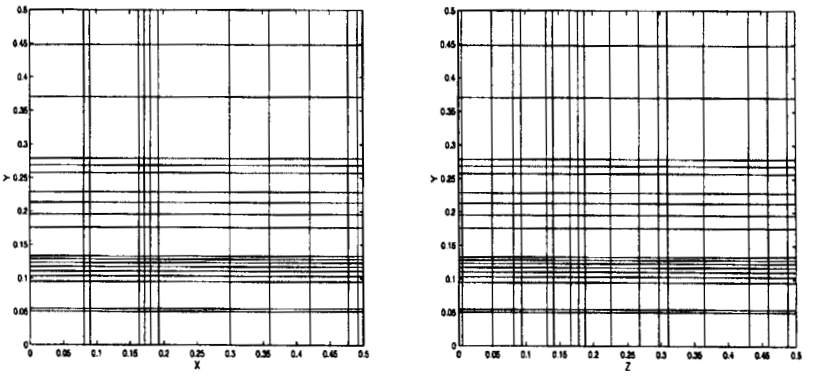
The error $\varepsilon^{\max}$ in the force in Eq. (48) needs to be sufficiently small for property p to be within a specified acceptable predictive accuracy Δp^*. Therefore, we obtain the following approximate expression for $\varepsilon^{\max}$:

$$\varepsilon^{\max} \simeq \left| \frac{\Delta p^*}{(\partial p / \partial \varepsilon^{\max})|_{\varepsilon^{\max}=0}} \right| \tag{53}$$

An estimate of the sensitivity $(\partial p / \partial \varepsilon^{\max})|_{\varepsilon^{\max}=0}$ appearing in the above equation can be obtained by finite difference perturbations; this generally produces a result that is sufficiently accurate for the purposes of Eq. (53). Alternatively, the adjoint or sensitivity equations corresponding to (47)–(51) can be formulated and solved [9].

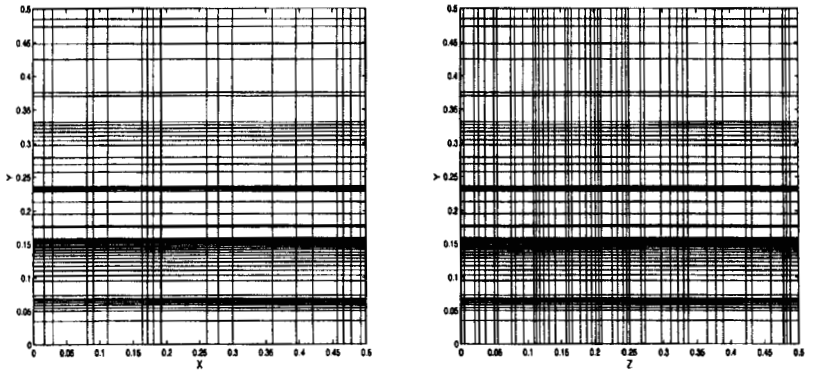
6. NUMERICAL EXPERIMENTS

In this section, we conduct numerical experiments aimed to study the effect of the force interpolation error on the accuracy of the quantities determined by molecular dynamics computations. To this end, we have constructed interpolation grids of different sizes for the combined Lennard–Jones force components using $\epsilon_{\max} = 120\,\mathrm{K}$ and $\sigma_{\max} = 4\,\text{Å}$ in the manner described in Section 5.1. Figures 9 and 10 show combined interpolant grids of size $20 \times 20 \times 20$ and $50 \times 50 \times 50$ respectively. The combined interpolant grid



(a) X-Y coordinate plane (b) Z-Y coordinate plane

FIGURE 9 Interpolation grid of size $20 \times 20 \times 20$.



(a) X-Y coordinate plane (b) Z-Y coordinate plane

FIGURE 10 Interpolation grid of size $50 \times 50 \times 50$.

TABLE I Maximum interpolant error as a function of grid size

Grid size	*Maximum interpolant error* $\varepsilon^{max}(k_b\varepsilon^{max}/\sigma^{max})$
$20 \times 20 \times 20$	7.862×10^{-5}
$50 \times 50 \times 50$	1.139×10^{-5}
$75 \times 75 \times 75$	4.937×10^{-6}

TABLE II Interpolation of modified force functions

Simulation of argon fluid ($N = 256$)						
		$\Delta T\ (K)$			$\Delta P\ (MPa)$	
	$T(K)$	ΔT_{act}	ΔT_{est}	$P\ (MPa)$	ΔP_{act}	ΔP_{est}
Interpolation over grid						
$20 \times 20 \times 20$	177.0	−0.2	−0.7	359.1	−0.7	−1.9
$50 \times 50 \times 50$	177.3	0.1	−0.2	360.2	0.4	−0.8
$75 \times 75 \times 75$	177.2	0.0	0.1	359.8	0.0	−0.2
Direct summation	177.2	–	–	359.8	–	–

of size $75 \times 75 \times 75$ has already been shown in Figure 8. Table I lists the maximum grid interpolation error $\varepsilon^{\max}$ attained by these grids.

Table II compares the results of the molecular dynamics computations obtained using the direct summation of the modified force functions[6] with those obtained using the interpolants on the above grids. The simulation involves 256 argon particles at a density of 39,960 mol/m^3 and energy of −3397 J/mol.

The differences between the temperature and pressure values calculated using the interpolants and the direct modified force summation are denoted as ΔT_{act} and ΔP_{act} respectively:

$$\Delta T_{\text{act}} \equiv T_{\text{int}} - T_{\text{MOD-10}}$$

$$\Delta P_{\text{act}} \equiv P_{\text{int}} - P_{\text{MOD-10}}$$

As can be seen, these differences are quite small; they improve with increasing grid resolution and, indeed, vanish (to 4 significant digits for both T and P) for the $75 \times 75 \times 75$ grid.

Table II also compares the actual errors, ΔT_{act} and ΔP_{act}, to the corresponding errors ΔT_{est} and ΔP_{est} estimated by the analysis presented in Section 5.3. These can be obtained from Eq. (52) using the values of $\varepsilon^{\max}$

[6]Here the force was computed using the MOD-10 framework (*cf.* Eq. (4)); thus each evaluation involves the summation of 9261 ($=21^3$) distinct terms.

listed in Table I. The comparison indicates that the analysis of Section 5.3 tends to provide an *upper bound* on the effects of the interpolation error, rather than a precise estimate. There are two reasons for this: first, the value of ε^{max} is itself an upper bound to the error of the interpolant – indeed, the actual error over most of the interpolation domain is expected to be better than this. Secondly, Eq. (48) assumes that the immediate effect of the interpolation error is always to *increase* the force by ε^{max}; in reality, this error could be negative as well as positive. Nevertheless, the value of ε^{max} that can be obtained from such an analysis (*cf.* Eq. (53)) is a useful, albeit somewhat conservative, estimate that will guarantee the required accuracy in T and P.

7. CONCLUDING REMARKS

Interpolation of potential functions has been quite common in the molecular simulation literature, with forces being calculated by differentiating these interpolants. Much of this work has been focused on specific types of potential, with the accuracy of interpolation being adjusted by trial-and-error. The continuity of the force across the grid boundaries was not always maintained.

The interpolation scheme proposed in this paper can be applied to essentially any continuous and differentiable force function defined over a finite spatial domain. Obviously, it is of maximum benefit in the case of complex force functions of the type of interest to this work. The resulting interpolant is continuous and differentiable throughout the domain under consideration. Moreover, the grid generation procedure guarantees the accuracy of the computed force within a user-specified tolerance.

It has also been shown that, for some forms of potential function, one can construct, once-and-for-all, interpolants of guaranteed accuracy over specified ranges of the density and of the parameters appearing in the potential. Such constructions are possible for some potential functions that are commonly used to describe non-bonded particle interactions, including the Lennard–Jones and Coulombic ones.

It has also been argued that it is both desirable and feasible to relate the error tolerance in the force calculation to the required precision of the predictions of the molecular dynamics computations. The procedure proposed for achieving this is of general interest as it is applicable to any molecular dynamics computation which employs some approximation in the calculation of force (*e.g.*, *via* the use of cut-off distances).

In conclusion, this paper and its companions represent an attempt to establish a formal framework for molecular dynamics that is consistent with, and depends on, two fundamental premises [1]:

(a) the physical system of interest exhibits a spatial periodicity expressed in terms of the infinite replication of a cube of given size;
(b) the non-bonded interactions in the system of particles under consideration can be described in terms of a set of pairwise interactions, each characterized by a continuous and twice-differentiable potential function.

No other assumptions or simplifications have been necessary. This allows molecular dynamics computations to be viewed as mathematical mappings that are continuous and differentiable functions of their arguments. It is also possible [2] to establish rigorous procedures for the computation of all partial derivatives of this mapping within the same degree of numerical accuracy as the results of the molecular dynamics computation itself; this information is likely to be valuable in a wide variety of practical applications. Finally, the rigorous and efficient force interpolation schemes proposed in this paper imply that there is little, if any, computational penalty associated with using the new molecular dynamics framework in comparison with its conventional counterparts.

Acknowledgments

The financial support of Mitsubishi Chemical Corporation through a Mitsubishi Fellowship to J. Stefanović is gratefully acknowledged. Partial funding has also been provided by the United Kingdom Engineering and Physical Sciences Research Council (EPSRC) under the IRC grant to the Centre for Process Systems Engineering at Imperial College.

References

[1] Stefanović, J. and Pantelides, C. C. (2000). *Molecular dynamics as a mathematical mapping. I. Differentiable force functions.* To appear in Mol. Sim.
[2] Stefanović, J. and Pantelides, C. C. (2000). *Molecular dynamics as a mathematical mapping. II. Partial derivatives in the microcanonical ensemble.* To appear in Mol. Sim.
[3] Swillinger, D. (1995). *CRC Standard Mathematical Tables and Formulae.* CRC Press, Washington, 30th edition.
[4] Bender, C. M. and Orszag, S. A., *Advanced Mathematical Methods for Scientists and Engineers.* McGraw-Hill, New York, 1978.
[5] Andrea, T. A., Swope, W. C. and Andersen, H. C. (1983). The role of long ranged forces in determining the structure and properties of liquid water. *J. Chem. Phys.*, **79**, 4576–4584.
[6] Sangester, M. J. L. and Dixon, M. (1976). Interionic potentials in alkali halides and their use in simulations of the molten salts. *Adv. Phys.*, **63**, 247–342.

[7] Berweger, C. D., van Gunsteren, W. F. and Müller-Plathe, F. (1997). Finite element interpolation for combined classical quantum mechanical molecular dynamics simulations. *J. Comput. Chem.*, **18**, 1484–1495.
[8] Berweger, C. D., van Gunsteren, W. F. and Müller-Plathe, F. (1998). Molecular dynamics simulation with an *ab initio* potential energy function and finite element interpolation: The photoisomerization of *cis*-stilbene in solution. *J. Chem. Phys.*, **108**, 8773–8781.
[9] Stefanović, J. (2000). *On the Mathematics of Molecular Dynamics. Ph.D. Thesis*, University of London.
[10] Nath, S. K., Escobedo, F. A. and de Pablo, J. J. (1998). On the simulation of the vapor–liquid equilibria for alkanes. *J. Chem. Phys.*, **108**, 9905–9911.