JOURNAL OF COMPUTATIONAL PHYSICS 115, 276–287 (1994)

Spline Approximation of "Effective" Potentials under Periodic Boundary Conditions

R. T. FAROUKI AND S. HAMAGUCHI

IBM Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598

Received May 12, 1993; in revised form December 16, 1993; accepted May 19, 1994

The use of spline functions to approximate the "effective" interparticle potentials that result from taking into account all image particles in periodic-boundary-condition Monte Carlo or molecular dynamics simulations is described. Such approximations are intrinsically very "smooth," easy to construct, relatively inexpensive to evaluate, and can provide a high degree of accuracy. The asymptotic properties of systems governed by long-range interactions may thus be determined using relatively small particle numbers. A number of implementation issues are discussed in detail, including the choice of end conditions, economical storage of the spline coefficients, conversion to B-spline form, and efficient evaluation procedures. Applied to the problem of locating the melting temperature T_m of a Yukawa system by means of molecular dynamics simulations, we observe values for T_m that are virtually independent of the particle number N if the pair potential includes the spline correction term and $N \gtrsim 250$, whereas using only the "minimum image" method gives T_m values that systematically decrease and attain the asymptotic value only for $N \gtrsim 5000$. © 1994 Academic Press, Inc.

1. INTRODUCTION

The use of periodic boundary conditions is a standard means of eliminating "surface effects" [1] in Monte Carlo or molecular dynamics calculations of the bulk properties of systems of particles whose interactions are describable by a scalar potential function $\phi(r)$, where r is the interparticle distance. If the simulation employs N particles in a cubical volume $V = L^3$ of side L, one is usually interested in asymptotic properties of the system in the limit $N \to \infty$ and $L \to \infty$ such that the mean density N/V remains constant.

Ideally, using periodic boundary conditions in three dimensions amounts to substituting, in lieu of the potential $\phi(r)$, the function defined over the interior of V by the infinite lattice sum

$$\Phi(\mathbf{r}) = \sum_{\mathbf{n}} \phi(|\mathbf{r} + \mathbf{n}L|), \tag{1}$$

where the components (λ, μ, ν) of the vector $\mathbf{n}$ assume all integer values. For each particle $i = 1, ..., N$ in V, the "effective" potential (1) describes the interaction with another particle j $(\neq i)$ in V at relative position $\mathbf{r} = \mathbf{r}_j - \mathbf{r}_i$, and with every *periodic image* of the latter when all space is imagined to be occupied by identical contiguous replicas of V.

Note that although the basic pair potential ϕ is spherically symmetric—depending only on the *magnitude* of the separation vector $\mathbf{r}$—the function Φ has only cubical symmetry; it depends explicitly on the *direction* of $\mathbf{r}$.

For short-range potentials, such as the familiar Lennard–Jones form

$$\phi(r) \propto \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}$$

for which it is customary to ignore all interactions at separations greater than a few times σ, a drastic simplification of (1) is usually adequate. Namely, it is generally possible to take N such that $\sigma/L \ll 1$, and the sum (1) is then dominated by a *single term*, for which $-1 \leq \lambda, \mu, \nu \leq +1$. Geometrically, this amounts to surrounding the basic volume V by 26 copies V' of itself, each having a face, edge, or vertex in common with V. Every particle $i = 1, ..., N$ in V then interacts with particle j $(\neq i)$ in V or with the image of j in one of the duplicate volumes V', according to whichever is nearest. Introduced by Metropolis *et al.* [2] in Monte Carlo equation-of-state calculations, this has come to be known as the *minimum image* method.

For simplicity, we shall focus on potentials $\phi(r)$ of "intermediate range," for which the sum (1) is absolutely convergent, but one must nevertheless include many terms for satisfactory accuracy at values of N that will yield reasonable simulation times. As a typical example of such potentials, we cite the Yukawa or screened-Coulomb form

$$\phi(r) \propto \frac{1}{r}\exp(-kr), \tag{2}$$

for which the sum (1) is absolutely convergent when $k > 0$ (we assume that kL is not large compared to unity, i.e., the minimum image method is inadequate). The Yukawa form (2) is the self-consistent pair potential [3] for the interaction of charged particles in colloidal or plasma suspensions, under the

0021-9991/94 $6.00

Debye–Hückel approximation applied to a background electrolyte or plasma medium of fixed temperature and mean density. It has been widely used to investigate the thermodynamic behavior of such systems [4–13].

The procedures described below are not limited to exponentially decaying potentials, however, and may be employed to approximate *any* smooth, nonsingular, cubically symmetric function that is expensive to evaluate. With the classical "one-component plasma" (OCP), for example, expression (1) includes contributions from discrete ions interacting through a bare Coulomb potential $\phi(r) \propto r^{-1}$ and a strictly uniform background of electrons giving overall charge neutrality. The sum (1) is then only conditionally convergent, and one appeals to the Ewald method [14, 15] to re-write it as the union of a spherically symmetric term and an infinite sum that converges rapidly:

$$\Phi(\mathbf{r}) \propto \frac{\operatorname{erfc}(\sqrt{\pi}r/L)}{r} - \frac{1}{L} + \sum_{\mathbf{n}\neq 0}\left[\frac{\operatorname{erfc}(\sqrt{\pi}|\mathbf{r}+\mathbf{n}L|/L)}{|\mathbf{r}+\mathbf{n}L|} + \frac{\exp(-\pi|\mathbf{n}|^2)\cos(2\pi\mathbf{n}\cdot\mathbf{r}/L)}{\pi|\mathbf{n}|^2 L}\right]. \tag{3}$$

The complementary error, exponential, and trigonometric functions make a direct summation for the cubically symmetric term in $\Phi(\mathbf{r})$ impractical in simulations, and many methods have been proposed to approximate it by forms that can be efficiently evaluated (see Section 2 below). We have used the spline approximation method in molecular dynamics simulations [16] to estimate the first anharmonic coefficient of the crystalline OCP energy, obtaining good agreement with recent lattice-dynamics studies [17, 18].

The plan of this paper is as follows. Section 2 briefly reviews related work on the approximation of complicated "effective" potentials by forms that are cheaper to evaluate. In Section 3 we develop the technique used here, namely, interpolation of a three-dimensional grid of sampled values by a cubic C^2 tensor-product spline function. The cardinal basis is the natural form in which to express this interpolant, requiring only the solution of tridiagonal linear systems and allowing easy incorporation of end conditions. Section 4 describes the conversion of the interpolant to B-spline representation—an essential step in facilitating its efficient evaluation. Finally, Section 5 presents empirical data on the accuracy achievable in practice and the expense of using full periodic boundary conditions as compared to the minimum image method, while Section 6 summarizes our results and makes some concluding remarks.

The methods discussed below are well-established in the mathematical theory of spline functions. However, they are not readily accessible without a substantial investment of time on the specialist literature of that subject. Our purpose here is to present a relatively simple and self-contained account, confined to the application at hand, that will encourage more widespread use of spline approximations for the periodic-boundary correction term in Monte Carlo and molecular dynamics simulations. As indicated in Section 2 below, there appear to be widespread misconceptions within the computational physics literature as to *what* constitutes a "spline" function.

2. APPROXIMATION OF POTENTIALS

Although the approximation of complicated "effective" potentials is a key ingredient of condensed-matter simulations, this problem appears to have received only anecdotal treatment in the literature. A simple scheme [1] for univariate (i.e., spherically symmetric) potentials involving many terms—see, for example, [19]—employs interpolation between pre-computed values in a look-up table. Andrea, Swope, and Andersen [20] described a piecewise-quintic Hermite interpolant to values and first and second derivatives sampled from a univariate potential. (Allen and Tildesley [1] deem this a "spline fit," but the method is strictly local in nature and hence cannot yield the global smoothness properties of true spline functions; see Section 3 below. Note also that Hermite interpolation to higher-order derivatives of a function on a given interval does not always yield better approximations to that function.)

As noted above, the approximation of trivariate potentials is important in handling the convergent series that arises when using the Ewald method for the one-component plasma. The pioneering study of Brush, Sahlin, and Teller [14] used a crude three-dimensional Taylor expansion to approximate this term. Hansen [15] obtained greater accuracy by means of an "optimized expansion in Kubic harmonics." Further enhancements were subsequently achieved by DeWitt and Hubbard [21] with interpolation between values in a three-dimensional table, and by Slattery, Doolen, and DeWitt [22] with a least-squares fit to an expansion in cubic harmonics of degree 22.

Finally, Helfer, McCrory, and Van Horn [23] appear to have attained even greater accuracy using a scheme to locally interpolate values and certain *second* derivatives of the function $\Phi(\mathbf{r}) - 1/r$ on a three-dimensional grid. Again, this so-called "modified cubic spline interpolation formula" is not a true spline function. In fact, it would appear to have *discontinuous first derivatives* across the boundaries of the cubical elements in the subdivision (the method is difficult to fully fathom since the authors provide few details and there are several confusing typographical errors). While this artifact may not be important in Monte Carlo calculations, it is undesirable in dynamical simulations, since particles would experience discontinuous force fields.

We believe that—depending on the available memory and certain trade-offs between the speed of evaluation and the volume of stored data—spline functions are capable of matching or exceeding the performance of any of the above methods in terms of accuracy of approximation, ease of construction, and efficiency of evaluation. Such functions are, moreover, quite versatile; the same code can be used to construct approximants to the periodic-boundary correction term for a variety of potentials. The input to the code is simply a set of *values* for the correction term on a regular grid; the method need not rely on derivatives of this term, which can be cumbersome to compute. Another important attribute is the overall "smoothness" of spline functions (i.e., their tendency to suppress oscillations) when the interpolated data is well behaved. While other approaches may also be capable of achieving a high approximation accuracy, this does not per se guard against microscopic fluctuations of the potential—which could, for example, induce spurious dynamics in condensed systems at low temperatures.

It will be convenient to extract the $\mathbf{n} = \mathbf{0}$ term from (1) and write

$$\Phi(\mathbf{r}) = \Phi_0(r) + \Phi_c(\mathbf{r}) = \phi(r) + \sum_{\mathbf{n}\neq\mathbf{0}} \phi(|\mathbf{r} + \mathbf{n}L|), \qquad (4)$$

$\Phi_0(r) = \phi(r)$ being the "basic" pair potential under the standard minimum image convention, while the "correction" term $\Phi_c(\mathbf{r})$ represents the potential for the interaction of one particle with all periodic images of the other. Each component of the separation vector $\mathbf{r}$ lies between $-L/2$ and $+L/2$.

We shall focus henceforth on the approximation of $\Phi_c(\mathbf{r})$ only. This has the disadvantage, in a Monte Carlo or molecular dynamics context, that the cost of including the effect of all periodic images is *in addition* to that of using the basic (minimum-image) pair potential. Since these two terms are cleanly separated, however, one can readily assess the importance of the $\Phi_c(\mathbf{r})$ contribution in any simulation by switching it on and off.

The decomposition (4) is motivated by the fact that $\phi(r)$ is singular at $r = 0$ in many cases of interest, and inclusion of this singularity complicates the direct approximation of $\Phi(\mathbf{r})$. Even if a short-range cutoff is imposed, $\Phi_c(\mathbf{r})$ typically has a smaller overall variation than $\Phi(\mathbf{r})$, making an accurate approximation of the former much easier. The Ewald potential (3) also has the form $\Phi_0(r) + \Phi_c(\mathbf{r})$, although in this case one should *not* interpret the spherically symmetric term $\Phi_0(r)$ as a "basic" pair potential and the infinite sum $\Phi_c(\mathbf{r})$ as the corresponding effect of all image particles [14, 15].

3. CONSTRUCTION OF THE INTERPOLANT

While there are several excellent comprehensive treatises on spline functions, e.g., [24–26], most "all-purpose" numerical analysis texts offer only a token discussion that is inadequate for our purposes. Accordingly, we commence with a gentle introduction to the construction and basic properties of splines, in the specific context of approximating the function $\Phi_c(\mathbf{r})$, that can serve as a point of departure for consulting the specialist literature. To keep matters simple and concise without omitting essential principles, we shall confine our attention to *cubic* splines defined on *uniform* sequences of nodes or "knots."

3.1. *Univariate Interpolants*

Beginning with the univariate case, consider the problem of finding piecewise-polynomial functions that assume specified values $s_1, ..., s_n$ at the integer points $t = 1, ..., n$ and, furthermore, have given derivatives s_1' and s_n' at the initial and final points. Clearly, there are infinitely many such functions, and—as noted above—the term "spline" has often been loosely used to denote piecewise-polynomial functions in general.

If we impose the additional condition that the desired function be of class C^2 (i.e., have continuous first and second derivatives), however, there is a unique piecewise-*cubic* $S(t)$ that matches the given data. Moreover, among *all* C^2 functions $f(t)$ interpolating the data, this piecewise-cubic is the one that minimizes the integral

$$\int_1^n [f''(t)]^2\, dt. \qquad (5)$$

It is only such functions—and their (tensor-product) generalizations to the multivariate case—that we shall henceforth refer to as "splines" (for details on higher-order splines with greater degrees of continuity, and splines defined over non-uniform knot sequences, see [24–26]).

The idea is that, if $|f'(t)| \ll 1$ for $1 \leq t \leq n$, the integral (5) approximates the strain energy of an "elastica" (i.e., a thin elastic beam) that has been bent so as to pass through the points (k, s_k) for $k = 1, ..., n$ and clamped at $t = 1$ and n so as to assume slopes s_1' and s_n' at these endpoints (formally, the strain energy is the integral of the square of the curvature

$$\kappa = (1 + f'^2)^{-3/2} f''$$

of the curve $x = t$, $y = f(t)$ with respect to arc length s, where $ds/dt = (1 + f'^2)^{1/2}$). Since $S(t)$ minimizes the approximate energy integral (5), it mimics the shape of an elastica—it is in this sense that the spline may be regarded as the "smoothest" function interpolating the given data.

The construction of $S(t)$ is not difficult. If $S^{(i)}(\tau)$ denotes the ith span $t \in [i, i + 1]$ of $S(t)$, considered as a function of the local variable $\tau = t - i \in [0, 1]$, we can express $S^{(i)}(\tau)$ in the *cubic Hermite basis* as

$$S^{(i)}(\tau) = s_i\, \alpha_0(\tau) + s_{i+1}\, \alpha_1(\tau) + s_i'\, \beta_0(\tau) + s_{i+1}'\, \beta_1(\tau), \qquad (6)$$

where s_i, s_{i+1} are specified values for $S(t)$ at the knots $t = i$, $i + 1$ and s_i', s_{i+1}' are the corresponding derivatives—as yet unknown, apart from the given end values s_1' and s_n'. Here the Hermite basis functions

$$\alpha_0(\tau) = 2\tau^3 - 3\tau^2 + 1, \quad \alpha_1(\tau) = -2\tau^3 + 3\tau^2,$$
$$\beta_0(\tau) = \tau^3 - 2\tau^2 + \tau, \quad \beta_1(\tau) = \tau^3 - \tau^2,$$

are the unique cubic polynomials that satisfy the boundary conditions

$$\begin{bmatrix} \alpha_0(0) & \alpha_0(1) & \alpha_0'(0) & \alpha_0'(1) \\ \alpha_1(0) & \alpha_1(1) & \alpha_1'(0) & \alpha_1'(1) \\ \beta_0(0) & \beta_0(1) & \beta_0'(0) & \beta_0'(1) \\ \beta_1(0) & \beta_1(1) & \beta_1'(0) & \beta_1'(1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Writing each span of $S(t)$ in the form (6) clearly guarantees first-order continuity at the juncture of spans i and $i + 1$ for $i = 1, ..., n - 2$. To ensure second-order continuity, we must match second derivatives at these nodes. On differentiating (6), this gives rise to the system of linear equations

$$s_{i-1}' + 4s_i' + s_{i+1}' = 3(s_{i+1} - s_{i-1}), \quad i = 2, ..., n - 1, \qquad (7)$$

for the unknown derivatives $s_2', ..., s_{n-1}'$. Together with the specified values s_1' and s_n' this amounts to a tridiagonal system, and solution of this system thus corresponds to a construction of the desired spline function.

The spline described above, interpolating both a sequence $s_1, ..., s_n$ of values *and* end-derivatives s_1' and s_n', is known as a "complete" spline. If the end-derivatives are unknown or difficult to compute, however, one will have fewer equations than unknowns. Methods for *automatically* selecting s_1', s_n' values that agree with the overall "shape" of the data $s_1, ..., s_n$ are then needed (the so-called "natural" spline, defined by setting $S''(1) = S''(n) = 0$, does not give very satisfactory results in general).

We describe two such methods, known as spline *end conditions:*

- *quadratic end-spans.* Instead of specifying s_1' and s_n' values, we close the system (7) by requiring $S'''(t) \equiv 0$ for $t \in [1, 2]$ and $[n - 1, n]$, i.e., the initial and final spans are just *quadratic* polynomials. Equations (7) are thus augmented by

$$s_1' + s_2' = 2(s_2 - s_1)$$

$$s_{n-1}' + s_n' = 2(s_n - s_{s-1}),$$

giving altogether a tridiagonal system of n equations.

- *"not-a-knot" condition.* Here we require *third*-order continuity at the knots $t = 2$ and $n - 1$, i.e., a *single* cubic polynomial is used for the first two and the last two spans. Equating third derivatives at the junctures $t = 2$ and $n - 1$ gives

$$s_3' - s_1' = 2s_1 - 4s_2 + 2s_3$$

$$s_n' - s_{n-2}' = 2s_{n-2} - 4s_{n-1} + 2s_n.$$

In order not to compromise the tridiagonal nature of the system (7), we add the first of the above equations to member $i = 2$ of that system, and subtract the second from member $i = n - 1$, thereby obtaining

$$4s_2' + 2s_3' = -s_1 - 4s_2 + 5s_3$$

$$2s_{n-2}' + 4s_{n-1}' = -5s_{n-2} + 4s_{n-1} + s_n.$$

Once the new tridiagonal system of $n - 2$ equations has been solved for $s_2', ..., s_{n-1}'$ we may complete the solution by setting

$$s_1' = s_3' - 2s_1 + 4s_2 - 2s_3$$

$$s_n' = s_{n-2}' + 2s_{n-2} - 4s_{n-1} + 2s_n.$$

Another end condition, appropriate to *periodic* splines, is to match first and second derivatives at $t = n$ with those at $t = 1$ (the equations to be solved are then no longer tridiagonal). Here the function $S(t)$—defined nominally over $t \in [1, n]$—is imagined to be periodic, being repeated indefinitely outside this interval. This might seem to be the appropriate choice for our present application, since the effective potential (1) is evidently periodic in each coordinate. However, for reasons given in Section 2, we want to approximate not the full effective potential $\Phi(\mathbf{r})$, but rather the quantity $\Phi_c(\mathbf{r})$ of equation (4), with the $\mathbf{n} = \mathbf{0}$ term missing, which is *not* a periodic function.

3.2. THE CARDINAL BASIS

The above construction was specific to a given set of values $s_1, ..., s_n$ at the nodes $t = 1, ..., n$ and end-derivatives s_1', s_n' or "end conditions." If several different sets of values are to be interpolated, or if one desires to interpolate values on a uniform array of nodes in more than one dimension, a somewhat different approach is recommended.

The complete splines that interpolate *arbitrary* values $s_1, ..., s_n$ and end-derivatives s_1', s_n' constitute a vector space of dimension $n + 2$ that may be spanned by any set of $n + 2$ linearly independent spline *basis* functions. In Section 4 we discuss perhaps the most fundamental basis, namely, the *B-spline* basis. For the moment, however, we wish to consider the basis that is best suited to interpolation problems: the *cardinal* basis.[1]

For $i = 1, ..., n$ the ith cardinal basis function $C_i(t)$ is simply the complete spline, constructed as described above, that has end-derivatives $C_i'(1) = C_i'(n) = 0$ and interpolates the values

$$C_i(j) = \delta_{ij} = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \qquad (8)$$

[1] We adopt this nomenclature following Ahlberg *et al.* [24], whereas de Boor [25] and others use "cardinal" to denote any spline that is defined on a uniform sequence of knots.

for $j = 1, \ldots, n$; i.e., it vanishes at every node except $t = i$, where it has the value unity. Two additional basis functions are needed, $C_0(t)$ and $C_{n+1}(t)$, which satisfy $C_0'(1) = C_{n+1}'(n) = 1$ and $C_0'(n) = C_{n+1}'(1) = 0$, but vanish at *every* knot $t = 1, \ldots, n$. Thus, any complete spline may clearly be expressed as

$$S(t) = \sum_{i=0}^{n+1} s_i C_i(t), \quad \text{where } s_0 = s_1', s_{n+1} = s_n'.$$

A cardinal basis with built-in end conditions may also be defined. Here, one dispenses with $C_0(t)$ and $C_{n+1}(t)$, while the basis functions $C_i(t)$ for $i = 1, \ldots, n$ again interpolate the values (8), but incorporating the desired end conditions (see Section 3.1 above) rather than setting $C_i'(1) = C_i'(n) = 0$. Any spline satisfying the specified end conditions and interpolating given values $s_1, \ldots, s_n$ at $t = 1, \ldots, n$ may then be immediately identified as

$$S(t) = \sum_{i=1}^{n} s_i C_i(t).$$

(The above form is the spline analog of the perhaps-familiar expression for the *polynomial* interpolant $P(t)$, of degree $n - 1$, to the values $s_1, \ldots, s_n$:

$$P(t) = \sum_{i=1}^{n} s_i L_i(t), \quad \text{where } L_i(t) = \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{t - j}{i - j},$$

in terms of the *Lagrange basis* $L_1(t), \ldots, L_n(t)$ for the nodes $t = 1, \ldots, n$.)

3.3. *Trivariate Interpolants as Tensor Products*

The above methods generalize directly to the smooth interpolation of data on uniform rectangular grids of arbitrary dimension. While the bivariate case is the simplest context in which to discuss this, we shall address the trivariate case here, since the periodic-boundary correction term $\Phi_c(\mathbf{r})$ that we wish to approximate is explicitly three dimensional.

Let Φ_{ijk}, $1 \leq i, j, k \leq n$, be a regular cubical grid of values for the function $\Phi_c(\mathbf{r})$ defined in Eq. (4), obtained within a prescribed tolerance by direct summation. In view of the cubical symmetry of $\Phi_c(\mathbf{r})$, it suffices to sample values in the octant $x, y, z \in [0, L/2]$ of the control volume. Also, it will be convenient to transform the physical coordinates $x = (i - 1)\Delta$, $y = (j - 1)\Delta$, $z = (k - 1)\Delta$ of the sample points, where $\Delta = L/2(n - 1)$, into the nonnegative integer triplets (i, j, k) for $1 \leq i, j, k \leq n$.

We now consider an extended data set, Φ_{ijk} for $0 \leq i, j, k \leq n + 1$, that includes certain *boundary derivatives*, defined as follows. If $i = 0$ or $n + 1$, the quantity Φ_{ijk} will represent the partial derivative

$$\left.\frac{\partial \Phi_c}{\partial x}\right|_{(x,y,z)=(1,j,k)} \quad \text{or} \quad \left.\frac{\partial \Phi_c}{\partial x}\right|_{(x,y,z)=(n,j,k)}$$

of Φ_c with respect to x, and likewise for the indices j, k and coordinates y, z. Similarly, if two or more indices i, j, k are equal to 0 or $n + 1$, then Φ_{ijk} denotes the appropriate second or third mixed derivative of $\Phi_c(\mathbf{r})$ at the indicated boundary point. Thus, the additional data that we specify amounts to:

- the values of the first derivatives $\partial\Phi_c/\partial x$, $\partial\Phi_c/\partial y$, and $\partial\Phi_c/\partial z$, at each of the n^2 grid points on the two control-volume faces parallel to the $y - z$, $z - x$, and $x - y$ planes, respectively;
- the values of the mixed second derivatives $\partial^2\Phi_c/\partial x\partial y$, $\partial^2\Phi_c/\partial y\partial z$, and $\partial^2\Phi_c/\partial z\partial x$, at each of the n grid points on the four control-volume edges parallel to the z, x, and y axes, respectively;
- and the values of the mixed third derivative $\partial^3\Phi_c/\partial x\partial y\partial z$ at each of the eight corners of the control volume.

There exists a *unique* trivariate piecewise-polynomial function $S(x, y, z)$ interpolating the extended data Φ_{ijk} for $0 \leq i, j, k \leq n + 1$ that is cubic in each variable and of class C_2^6 (i.e., all its partial derivatives of order ≤ 6 that involve no more than twofold differentiation with respect to each coordinate are continuous). Indeed, in terms of the complete univariate cardinal basis $C_0(t), \ldots, C_{n+1}(t)$ defined above, this spline function may be immediately identified as

$$S(x, y, z) = \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} \sum_{k=0}^{n+1} \Phi_{ijk} C_i(x) C_j(y) C_k(z). \tag{9}$$

Furthermore, among *all* trivariate functions $f(x, y, z)$ that interpolate the given data and are of class C_2^6, the complete tensor-product spline interpolant (9) is the one that minimizes the quantity [24]:

$$\int_1^n \int_1^n \int_1^n \left[\frac{\partial^6 f}{\partial x^2 \partial y^2 \partial z^2}\right]^2 dx\, dy\, dz. \tag{10}$$

The above integral may be regarded as a measure of "smoothness" for the hypersurface passing through given points (i, j, k, Φ_{ijk}) in a Euclidean space of dimension 4, although the physical analog with the strain energy of such a hypersurface is somewhat more tenous than in the univariate case.

In order to obtain the boundary derivative data for the term

$$\Phi_c(\mathbf{r}) = \sum_{\mathbf{n} \neq \mathbf{0}} \phi(|\mathbf{r} + \mathbf{n}L|)$$

needed by the complete spline interpolant (9), we write the derivatives of the pair potential ϕ with respect to its argument as ϕ', ϕ'', ϕ''' and, for each $\mathbf{n}$, we denote the direction cosines of the vector $\mathbf{r} + \mathbf{n}L$ by $\cos\theta_x$, $\cos\theta_y$, $\cos\theta_z$. Then it is easily verified, for example, that

$$\frac{\partial\Phi_c}{\partial x}(\mathbf{r}) = \sum_{\mathbf{n}\neq 0} \phi'(|\mathbf{r} + \mathbf{n}L|)\cos\theta_x,$$

$$\frac{\partial^2\Phi_c}{\partial x\partial y}(\mathbf{r}) = \sum_{\mathbf{n}\neq 0}\left[\phi''(|\mathbf{r} + \mathbf{n}L|) - \frac{\phi'(|\mathbf{r} + \mathbf{n}L|)}{|\mathbf{r} + \mathbf{n}L|}\right]\cos\theta_x\cos\theta_y,$$

$$\frac{\partial^3\Phi_c}{\partial x\partial y\partial z}(\mathbf{r}) = \sum_{\mathbf{n}\neq 0}\left[\phi'''(|\mathbf{r} + \mathbf{n}L|) - 3\frac{\phi''(|\mathbf{r} + \mathbf{n}L|)}{|\mathbf{r} + \mathbf{n}L|} + 3\frac{\phi'(|\mathbf{r} + \mathbf{n}L|)}{|\mathbf{r} + \mathbf{n}L|^2}\right]\cos\theta_x\cos\theta_y\cos\theta_z,$$

with analogous expressions for the other first and mixed second derivatives. For the Ewald potential (3), the quantity $\Phi_c(\mathbf{r})$ is not precisely of the above form, and it is slightly more laborious to compute the desired derivatives.

Note that, to allow for the scaling of the sample-point coordinates, the physical values for the derivatives of Φ_c given above must be multiplied by appropriate powers of the sample interval Δ before they are interpolated.

The computation of boundary derivative data can, of course, be omitted (at the cost of some degradation in the approximation accuracy; see Section 5 below) by using a cardinal basis $C_1(t)$, ..., $C_n(t)$ with built-in end conditions. For such a basis,

$$S(x, y, z) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \Phi_{ijk}C_i(x)C_j(y)C_k(z)$$

gives an interpolant to just the *values* ϕ_{ijk} for $1 \leq i, j, k \leq n$ of $\Phi_c(\mathbf{r})$ on a cubical grid. The end conditions at $t = 1$, representing points on midplanes of the simulation volume, should be $C_i'(1) = 0$, since the symmetric distribution of image particles about such planes gives an exact cancellation of the normal force due to them. At $t = n$, representing points on faces of the simulation volume, we have no equivalent information, and quadratic end spans or the "not-a-knot" condition must suffice.

We conclude this section by noting that, in view of the cubical symmetry of (9), it is not necessary to compute or store all $(n + 2)^3$ of the values Φ_{ijk} for $0 \leq i, j, k \leq n + 1$. In fact, the tetrahedral array defined by

$$0 \leq k \leq j \leq i \leq n + 1$$

contains all the *distinct* values that we need, which number

$$\sum_{i=0}^{n+1}\sum_{j=0}^{i}\sum_{k=0}^{j} 1 = \frac{(n + 2)(n + 3)(n + 4)}{6}$$

in total. We can store only the distinct values in a linear array, with

$$m = \tfrac{1}{6}i(i + 1)(i + 2) + \tfrac{1}{2}j(j + 1) + k + 1$$

being the approximate index into this array when arbitrary indices i, j, k are permuted so that $i \geq j \geq k$.

4. CONVERSION TO B-SPLINE FORM

While the cardinal form (9) facilitated a simple *construction* of the spine approximant to $\Phi_c(\mathbf{r})$, it is not useful as a representation for simulations in which this approximant will be repeatedly evaluated. Other than at the grid points, $(n + 2)^3$ nonzero terms evidently contribute to the value of the sum (9), and as n is increased to suppress the approximation error, the evaluation cost would become prohibitive.

As noted in Section 3, the complete splines on a given sequence of knots admit representations in many different bases. We will now convert (9) to a form that admits efficient evaluation, at a cost that is *independent* of the number of knots. The distinguishing feature of the B-spline basis $B_0(t)$, ..., $B_{n+1}(t)$, in which we wish to represent (9), is the *compact support* of the basis functions, i.e., the fact that each is nonzero over only a subset of the domain $1 \leq t \leq n$. Thus, if we compute the B-spline coefficients $\Psi_{\alpha\beta\gamma}$ for $S(x, y, z)$ and express it in the form

$$S(x, y, z) = \sum_{\alpha=0}^{n+1}\sum_{\beta=0}^{n+1}\sum_{\gamma=0}^{n+1} \Psi_{\alpha\beta\gamma}B_\alpha(x)\,B_\beta(y)B_\gamma(z), \qquad (11)$$

then for any (x, y, z) only a relatively small number of combinations α, β, γ of the indices will contribute to the value of $S(x, y, z)$.

4.1. *The B-Spline Basis*

To define a B-spline basis for the domain $1 \leq t \leq n$, the knot sequence must be augmented by three initial knots at $t = -2, -1, 0$ and three final knots at $t = n + 1, n + 2, n + 3$. For uniform knots, the B-spline basis is actually just a set of translates of a unique nonnegative C^2 piecewise-cubic, $B_\alpha(t)$ say, which we may regard as being defined by the following properties:

- $B_\alpha(t) \equiv 0$ for $t \leq \alpha - 2$ and $t \geq \alpha + 2$,
- $B_\alpha(\alpha - 1) = B_\alpha(\alpha + 1) = \frac{1}{6}$ and $B_\alpha(\alpha) = \frac{2}{3}$.

The above values are chosen to give "normalized" B-splines, for which

$$\int_{-\infty}^{+\infty} B_\alpha(t)\,dt = 1.$$

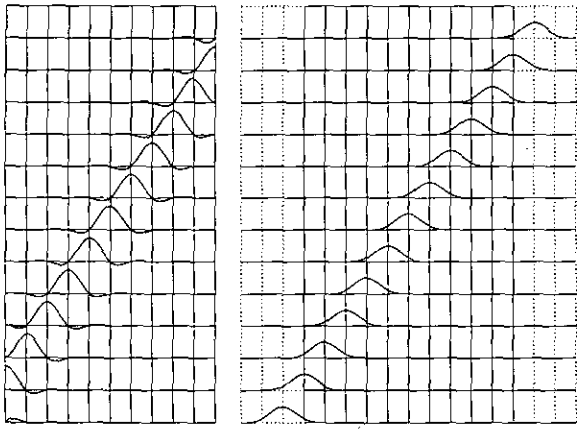

FIG. 1. Comparison of the complete cardinal and B-spline bases on a sequence of eleven uniformly spaced knots.

Figure 1 compares complete cardinal and B-spline bases over uniform knots. From the specified values and continuity properties of $B_\alpha(t)$ we may deduce, in terms of the "local" variable $\tau = t - \alpha$, the explicit representation

$$6\,B_\alpha(\tau) = \begin{cases} (\tau+2)^3 & \text{for } -2 \le \tau \le -1, \\ -3\tau^3 - 6\tau^2 + 4 & \text{for } -1 \le \tau \le 0, \\ 3\tau^3 - 6\tau^2 + 4 & \text{for } 0 \le \tau \le 1, \\ (2-\tau)^3 & \text{for } 1 \le \tau \le 2. \end{cases} \tag{12}$$

It follows that for any (noninteger) t between 1 and n, exactly *four* of the B-spline basis functions $B_0(t)$, ..., $B_n(t)$ will be nonzero. In particular, if $\alpha < t < \alpha + 1$, say, the nonvanishing basis functions are $B_{\alpha-1}(t)$, $B_\alpha(t)$, $B_{\alpha+1}(t)$, $B_{\alpha+2}(t)$. At each knot, however, there are only *three* nonvanishing basis functions; if $t = \alpha$, say, they are $B_{\alpha-1}(t)$, $B_\alpha(t)$, $B_{\alpha+1}(t)$.

Thus, if we can convert the interpolant (9) to the B-spline representation (11), there will be only $4^3 = 64$ rather than $(n+2)^3$ nonzero contributions to $S(x, y, z)$ at an arbitrary point. Even with the B-spline form, however, a careful analysis is needed to formulate efficient procedures for evaluating $S(x, y, z)$. We will return to this matter in Section 4.3—for now we focus on obtaining the B-spline coefficients $\Psi_{\alpha\beta\gamma}$ from the interpolated values Φ_{ijk}.

We achieve this by noting that the complete cardinal splines defined in Section 3.2 can be expressed as linear combinations of B-splines:

$$C_i(t) = \sum_{\alpha=0}^{n+1} \Lambda_{i\alpha} B_\alpha(t) \quad \text{for } i = 0, ..., n+1. \tag{13}$$

Thus, by substituting the above into (9), the B-spline coefficients are seen to given by a threefold contraction of the array of interpolant values with the $(n+2) \times (n+2)$ *basis conversion matrix* $\mathbf{\Lambda}$ having elements $\Lambda_{i\alpha}$,

$$\Psi_{\alpha\beta\gamma} = \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} \sum_{k=0}^{n+1} \Phi_{ijk} \Lambda_{i\alpha} \Lambda_{j\beta} \Lambda_{k\gamma}, \quad 0 \le \alpha, \beta, \gamma \le n+1. \tag{14}$$

4.2. *The Basis Conversion Matrix*

The construction of the basis conversion matrix has a simple formulation in terms of solving $n + 2$ systems of tridiagonal equations. Noting that, at each node $\alpha = 1, ..., n$, the only nonzero B-splines have the values $B_\alpha(\alpha) = \frac{2}{3}$ and $B_{\alpha-1}(\alpha) = B_{\alpha+1}(\alpha) = \frac{1}{6}$, we obtain

$$\begin{aligned} \Lambda_{i,\alpha-1} + 4\Lambda_{i\alpha} + \Lambda_{i,\alpha+1} &= 6\delta_{i\alpha} \quad \text{if } 1 \le i \le n, \\ \Lambda_{i,\alpha-1} + 4\Lambda_{i\alpha} + \Lambda_{i,\alpha+1} &= 0 \quad \text{if } i = 0 \text{ or } n+1, \end{aligned}$$

in view of the known nodal values of the complete cardinal basis functions. For fixed i, the above form an incomplete tridiagonal system of n equations in $n + 2$ unknowns. To complete the system, we match the derivatives of both sides of (13) at $t = 1$ and $t = n$. From the form (12) it is readily verified that $B'_\alpha(\alpha) = 0$ and $-B'_{\alpha-1}(\alpha) = B'_{\alpha+1}(\alpha) = \frac{1}{2}$. Hence we deduce that

$$\begin{aligned} &\Lambda_{i2} - \Lambda_{i0} = 2, \quad \Lambda_{i,n+1} - \Lambda_{i,n-1} = 0 \quad \text{if } i = 0, \\ &\Lambda_{i2} - \Lambda_{i0} = 0, \quad \Lambda_{i,n+1} - \Lambda_{i,n-1} = 0 \quad \text{if } 1 \le i \le n, \\ &\Lambda_{i2} - \Lambda_{i0} = 0, \quad \Lambda_{i,n+1} - \Lambda_{i,n-1} = 2 \quad \text{if } i = n+1. \end{aligned}$$

Solving each of the $n + 2$ tridiagonal systems thus defined for $i = 0, ..., n+1$ then gives all the elements of $\mathbf{\Lambda}$ and allows the computation of the B-spline coefficients using (14). Again, we take advantage of the cubical symmetry, computing and storing only those values for which $0 \le \gamma \le \beta \le \alpha \le n + 1$.

If a cardinal basis with built-in end conditions has been used, the matrix $\mathbf{\Lambda}$ will be of dimension $n \times (n+2)$, since we dispense with $C_0(t)$ and $C_{n+1}(t)$. The systems of equations defining its entries are exactly as given before, except that members reflecting these end conditions must be used instead of $C'_i(1) = C'_i(n) = 0$ for $i = 1, ..., n$. Actually, for reasons stated in Section 3.3, it is advisable to keep the equation $\Lambda_{i2} - \Lambda_{i0} = 0$ corresponding to $C'_i(1) = 0$, and for the other end condition use either

$$\Lambda_{i,n+1} - 3\Lambda_{in} + 3\Lambda_{i,n-1} - \Lambda_{i,n-2} = 0$$

for a quadratic end span over $t \in [n-1, n]$, or

$$\Lambda_{i,n+1} - 4\Lambda_{in} + 6\Lambda_{i,n-1} - 4\Lambda_{i,n-2} + \Lambda_{i,n-3} = 0$$

for the not-a-knot condition with continuity of $C'''_i(t)$ at $t =$

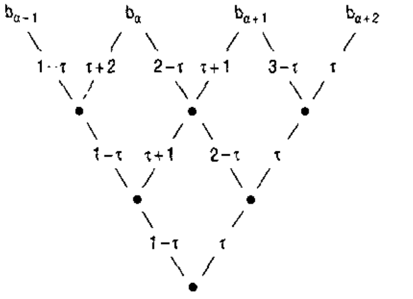
$b_{\alpha-1}$
b_α
$b_{\alpha+1}$
$b_{\alpha+2}$
$1-\tau$ $\tau+2$ $2-\tau$ $\tau+1$ $3-\tau$ τ
$1-\tau$ $\tau+1$ $2-\tau$ τ
$1-\tau$ τ

FIG. 2. Computational graph for the de Boor algorithm. Given B-spline coefficients $b_{\alpha-1}$, b_α, $b_{\alpha+1}$, $b_{\alpha+2}$ and the local variable $\tau = t - \alpha$, each node "•" to be filled in is obtained by multiplying the values diagonally above by the factors on the edges leading to that node and adding the results.

$n - 1$. Both the above equations can be combined with other members of the system, so as not to destroy its overall tridiagonal nature.

We conclude by noting that the values Φ_{ijk} could, of course, be directly interpolated using the representation (11). The resulting system of linear equations is, however, quite cumbersome and of much larger bandwidth. The basis-conversion approach that we have taken is more transparent, simple to implement, and numerically stable (it requires only a tridiagonal solver, and all the matrices involved are diagonally dominant).

4.3. *Efficient Evaluation Procedures*

Even after conversion from the cardinal form (9) to the B-spline form (11), careful consideration must still be given as to how the latter is evaluated. For potentials which require inclusion of the periodic-correction term, it will usually be necessary to explicitly take all pairwise interactions between the N simulation particles into account. This means that the evaluation of (11) will be done $O(N^2)$ times per simulation step.

The standard means of evaluating a univariate spline function

$$S(t) = \sum_{i=0}^{n+1} b_i B_i(t) \tag{15}$$

with given B-spline coefficients $b_0, ..., b_{n+1}$ is through a recursive formula known as the *de Boor algorithm*. If we wish to evaluate (15) at a point t such that $\alpha \leq t < \alpha + 1$, say, it can be verified from (12) that $S(t)$ reduces to

$$\begin{aligned} S(t) = \tfrac{1}{6}[&b_{\alpha-1}(1-\tau)^3 + b_\alpha(3\tau^3 - 6\tau^2 + 4) \\ &+ b_{\alpha+1}(-3\tau^3 + 3\tau^2 + 3\tau + 1) + b_{\alpha+2}\tau^3], \end{aligned} \tag{16}$$

where $\tau = t - \alpha$. For the simple case of uniform knots, the de Boor algorithm can be represented by the computational graph shown in Fig. 2. Starting with the appropriate B-spline coefficients $b_{\alpha-1}$, b_α, $b_{\alpha+1}$, $b_{\alpha+2}$, the values "•" to be filled in at each level are obtained by multiplying the entries diagonally above to the left and right by the factors on the edges leading to those entries and adding the results. Dividing the final • by 6 then gives the value (16).

Given τ, the graph of Fig. 2 incurs 12 multiplications and six additions (corresponding to the number of its edges and • nodes). A further five additions are needed to form the quantities $3 - \tau$, $2 - \tau$, $1 - \tau$, $\tau + 1$, $\tau + 2$ and one multiplication for the final normalization; the total cost is thus 13 multiplies and 11 adds. This is rather high in view of the fact that we are effectively just evaluating a cubic polynomial; if the power form

$$c_{\alpha,0} + c_{\alpha,1}\tau + c_{\alpha,2}\tau^2 + c_{\alpha,3}\tau^3 \tag{17}$$

of $S(t)$ for $t \in [\alpha, \alpha + 1]$ were available, it could be evaluated at a quarter of the cost—just three multiplies and three adds—by Horner's method.

However, storing power coefficients for each span of $S(t)$ requires $4(n - 1)$ memory locations, instead of just $n + 2$ for the B-spline coefficients. The problem is even more severe for the trivariate spline (11), where the $(n + 2)^3$ B-spline coefficients must be replaced by $64(n - 1)^3$ power coefficients to allow nested Horner evaluation (since $S(x, y, z)$ is cubic in each of three variables, it requires $4^3 = 64$ power coefficients for its specification within each of $(n - 1)^3$ cubical regions). Allowing for symmetry reduces both these numbers by a factor of ~6, but does not appreciably alter their relative magnitudes.

As the approximation error decreases roughly in proportion to n^{-4} (see Section 5) there is a fundamental trade-off between accuracy and efficiency. For given memory, storing only the B-spline coefficients allows n values ~4 times larger than if pre-computed power coefficients c_{ijk} for each span are stored, yielding an approximation error ~256 times smaller. However, accepting the lower accuracy entailed by storage of the power coefficients allows evaluation ~4 times faster than when only the B-spline coefficients are known.

Since the coefficients of the local power form (17) are given by multiplying the appropriate B-spline coefficients $b_{\alpha-1}$, b_α, $b_{\alpha+1}$, $b_{\alpha+2}$ by the matrix

$$\mathbf{M} = \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix},$$

the power coefficients appropriate to each cubical span $1 \leq \alpha$, β, $\gamma \leq n - 1$ of the tensor product (11) are obtained by threefold multiplication with $\mathbf{M}$:

$$c_{\alpha\beta\gamma,ijk} = \sum_{l=1}^{4} \sum_{m=1}^{4} \sum_{n=1}^{4} M_{il} M_{jm} M_{kn} \, \Psi_{\alpha+l-2,\beta+m-2,\gamma+n-2} \tag{18}$$

for $0 \leq i, j, k \leq 3$. The local power form of (11) is then

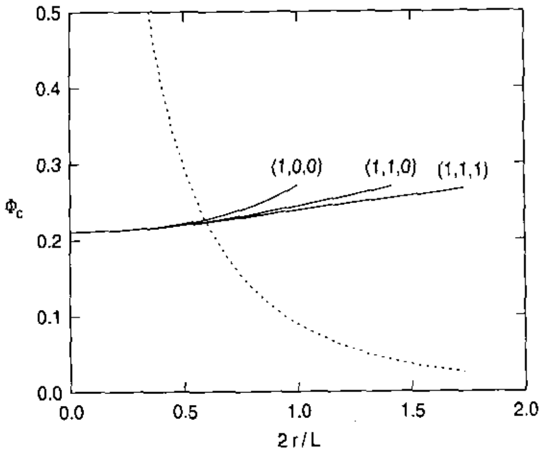

FIG. 3. The magnitude of the periodic-correction term Φ_c in the three directions (1, 0, 0), (1, 1, 0), and (1, 1, 1), compared with the "bare" Yukawa potential (dashed curve) for the case $\kappa = 0.25$ and $N = 128$ ($L/a \approx 8.1$).

$$S(x, y, z) = \sum_{i=0}^{3} \sum_{j=0}^{3} \sum_{k=0}^{3} c_{\alpha\beta\gamma,ijk}(x - \alpha)^i(y - \beta)^j(z - \gamma)^k,$$

which may be rapidly evaluated using a nested Horner scheme.

If memory constraints allow only the B-spline coefficients to be stored, it is possible to improve slightly (in terms of total arithmetic count, at least) on the algorithm illustrated in Fig. 2 for computing the value (16). With $\sigma = 1 - \tau$, we may rewrite (16) in the form

$$S(t) = \tfrac{1}{6}[b_{\alpha-1}\sigma^3 + 3(b_\alpha\sigma + b_{\alpha+1}\tau)(\sigma\tau + 1) \\ + b_\alpha + b_{\alpha+1} + b_{\alpha+2}\tau^3],$$

which (including the formation of σ and the final normalization) requires just 12 multiplies and seven adds for evaluation. Whether this results in a discernible speedup, however, depends in detail on the platform and implementation (the IBM RISC System/6000 workstations, for example, are capable of executing a multiplication *and* an addition in a single machine cycle [27] if the operands satisfy certain constraints).

In dynamical simulations, it is the force-field $F_c = -\nabla\Phi_c$ corresponding to the periodic correction potential Φ_c that will be frequently evaluated; considerations similar to those given above apply to its efficient evaluation. (It is also usually necessary to compute several higher derivatives of $\mathbf{F}_c$ when starting up the numerical integration scheme.)

5. EMPIRICAL RESULTS

The methods described above have been implemented and tested using both the Yukawa potential (2) for charged particles in a responsive neutralizing background, and the Ewald potential (3) for the one-component plasma. For brevity, we give representative results for only the Yukawa system (this has been used to model suspensions of small polystyrene spheres in water, which exhibit phase transitions and other interesting phenomena on scales that can be probed by visible light [28, 29]).

For N particles in a cubical simulation volume of side L, the Wigner–Seitz radius is defined by $a = (3/4\pi N)^{1/3}L$. For the Yukawa potential (2), the dimensionless quantity $\kappa = ka$ measures the strength of particle interactions, while the magnitude of $kL = (4\pi N/3)^{1/3}\kappa$ determines how rapidly the sum (1) converges and how important the correction $\Phi_c(\mathbf{r})$ is relative to the "bare" Yukawa potential. The minimum image method is valid only when $kL \gg 1$.

We use the case $\kappa = 0.25$ and $N = 128$, for which $kL \approx 2$ and inclusion of the periodic correction term is thus essential, to illustrate the performance of the spline approximation procedure. Figure 3 compares the variation of $\phi(r)$ and $\Phi_c(\mathbf{r})$ in this case; note that the latter substantially *exceeds* the former except near the center of the simulation volume. In Figure 4 we show, in the (x, y) plane, the force fields that correspond to these terms.

For the complete spline interpolant to a uniform grid of $n \times n \times n$ values and the boundary derivative data enumerated in Section 3.3, Fig. 5 shows measured root-mean-square and maximum errors in the spline approximation to Φ_c and its associated gradient for various values of n. Here the data that

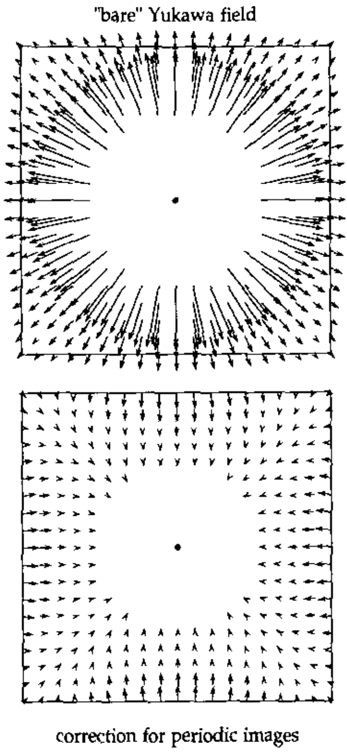

FIG. 4. Comparison of the "bare" Yukawa force-field in the (x, y)-plane with the correction due to periodic images, for the case $\kappa = 0.25$ and $N = 128$. Note that the latter exactly cancels the former at the center of each face and edge, and at each vertex of the control volume.

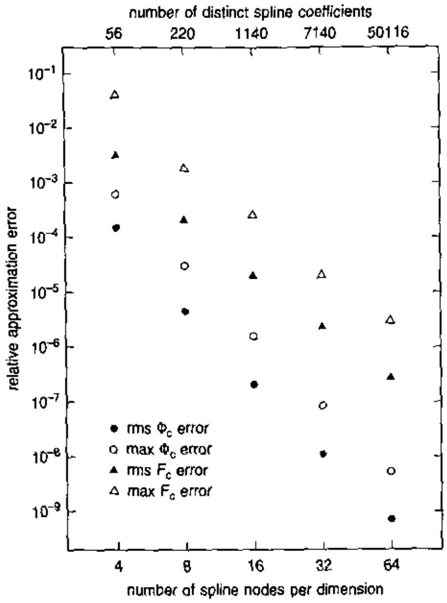

FIG. 5. Measured root-mean-square and maximum relative errors in the spline approximation to the periodic correction $\Phi_c(\mathbf{r})$ and its corresponding force field $\mathbf{F}_c(\mathbf{r}) = -\nabla\Phi_c(\mathbf{r})$, for the case $\kappa = 0.25$ and $N = 128$.

is interpolated, and against which the approximation accuracy is tested, was obtained by direct summation to a relative error of 10^{-12} or better.

To obtain measures representative of the fields experienced by particles in a simulation, the data of Fig. 5 were based on sampling at 10^4 randomly selected points. If the sum (1) is carried sufficiently far in computing the interpolant values, the error in the spline approximation to Φ_c is essentially zero at each node, but the gradient of the approximation will generally have a nonzero error everywhere.

In Monte Carlo calculations only Φ_c is of interest, whereas in dynamical simulations the periodic-correction force field $\mathbf{F}_c = -\nabla\,\Phi_c$ is primarily used. While approximations to $\mathbf{F}_c$ of greater accuracy and an additional order of continuity could be obtained by directly interpolating it componentwise, it is preferable (e.g., for computing the system energy) to avoid forces that are not explicitly derived from known potential functions.

The decline of the approximation error with the number of spline nodes n per dimension seen in Fig. 5 agrees well with standard error estimates [25] for spline interpolation. In the univariate case, for example, it may be shown that the complete spline $S(t)$ interpolating n uniformly spaced values of a function $f(t)$ on $t \in [0, 1]$ satisfies

$$\max_{0\le t\le 1} |S(t) - f(t)| \le \frac{5}{384}\frac{1}{(n-1)^4}\max_{0\le t\le 1}|f^{(4)}(t)|,$$

while the error in the first derivative of $S(t)$ may be characterized by

$$\max_{0\le t\le 1} |S'(t) - f'(t)| \le \frac{1}{24}\frac{1}{(n-1)^3}\max_{0\le t\le 1}|f^{(4)}(t)|.$$

The weaker decay with n of the error in $\mathbf{F}_c = -\nabla\Phi_c$ than of that in Φ_c itself is clearly evident in Fig. 5.

As the computation of the boundary derivative data needed by complete spline interpolants is somewhat cumbersome, especially for more complicated potentials, we have also tried interpolating just a grid of *values* for Φ_c using cardinal bases with built-in end conditions. Specifically, zero derivatives at x, y, $z = 0$ and either quadratic end spans or the not-a-knot condition at x, y, $z = L/2$ were used (as described in Section 3.3). In each case, significant increases of the rms and maximum approximation errors were observed for both Φ_c and $\nabla\Phi_c$, by a factor of ~4 for the not-a-knot case and by about an order of magnitude for quadratic end spans. Thus, the complete spline is preferred for greatest accuracy, but if the derivative data is difficult or impossible to compute, the not-a-knot condition will give somewhat better performance than quadratic end spans.

A comparison of the program speed under various strategies for evaluating the correction term Φ_c was performed. When n was not too large and the local power coefficients (18) could be stored, allowing the evaluation of Φ_c by a nested Horner scheme, the run time was found to increase by a factor of ~3.7 as compared to cases using only the "bare" potential ϕ. On a small workstation with 16 Mbytes of memory, for example, a $25 \times 25 \times 25$ mesh for the approximation could easily be accommodated by this scheme (giving rms Φ_c and $\mathbf{F}_c$ errors $<10^{-7}$ and 10^{-5}, respectively) without paging problems.

If only the B-spline coefficients were stored, however, the evaluation of Φ_c became intolerably slow—the program ran about 30 times slower than the "bare" ϕ case! This may, perhaps, be partly attributable to our use of function calls (rather than in-line code) for nested execution of the de Boor algorithm. Nevertheless, the latter requires about 4 times as much arithmetic per evaluation than Horner's method (see Section 4.3), and a significant loss of speed was to be expected. Thus, relying on the B-spline coefficients only is not recommended unless one wants to use all the available memory to obtain very high accuracy in Φ_c at the cost of a severe price in performance. Note that even relatively coarse grids (e.g., $n = 10$) yield approximations that are adequate for most purposes and that make fairly modest demands (<0.5 Mbyte) on memory when using the more-efficient evaluation scheme.

To illustrate the physical importance of the periodic correction term Φ_c, we consider the problem of determining the asymptotic ($N \to \infty$) melting temperature of a $\kappa = 0.5$ Yukawa system. If κ is not large, the bcc lattice is known to be the stable form of the frozen Yukawa system. Our experiments consist of integrating the equations of motion for $N = 2\nu^3$ particles, placed initially at bcc lattice sites (the integer ν is the number of conventional cells). Dynamical equilibrium at a given temperature T is achieved by periodically renormalizing

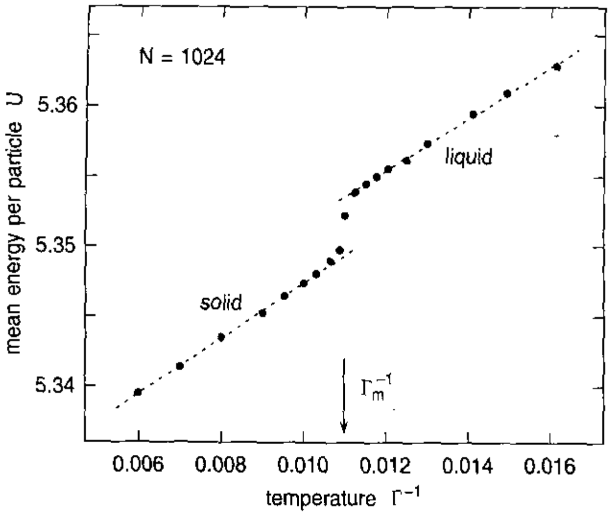

FIG. 6. Mean internal energy per particle U as a function of Γ^{-1} for a system of $N = 1024$ particles interacting through a $\kappa = 0.5$ Yukawa potential with the periodic correction term included. A transition from crystalline to fluid states is apparent as a jump in U at $\Gamma_m \approx 92$ (the dashed lines represent least-squares fits to U for the solid and liquid phases).

the particle velocities [30]. In lieu of the temperature, we use the dimensionless coupling parameter

$$\Gamma = \frac{Q^2 \exp(-\kappa)}{4\pi\varepsilon_0 a\, k_B T},$$

roughly equal to the ratio of mean potential to kinetic energy per particle (Q is the charge on each particle, while ε_0 and k_B are the permittivity of free space and the Boltzmann constant). The Yukawa-system thermodynamics is completely describable in terms of the quantities κ and Γ.

We wish to compare the dependence of the value Γ_m at which the frozen Yukawa system melts on the number of particles N, when using: (i) just the minimum image method and (ii) full periodic boundary conditions based on the spline correction term $\Phi_c(\mathbf{r})$. The phase transition is identified by a jump in the equilibrium mean energy per particle,

$$U = \frac{1}{N}\left[\sum_{i=1}^{N} \tfrac{1}{2} m|\mathbf{v}_i|^2 + \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \Phi(\mathbf{r}_i - \mathbf{r}_j)\right],$$

as Γ is decreased. Figure 6 shows such values of U, in units of $Q^2/4\pi\varepsilon_0 a$, obtained from the simulations—these values represent time averages after an initial "setting" period. (The total internal energy of the plasma/charged-particle system differs from NU by only an additive constant.) With runs at many different Γ values, it is possible to isolate Γ_m to high accuracy.

Figure 7 shows the observed dependence of the Γ_m value on $N = 2\nu^3$ for $\nu = 4, \ldots, 10$. While Γ_m is seen to increase steadily with N when using just the minimum image convention, it is *nearly constant* when the Φ_c term is included (excepting, perhaps, some residual small-number effects for $\nu \lesssim 5$). The Γ_m values based on the minimum image method are within ~10% of the asymptotic value indicated by runs using the periodic correction term only when $N \gtrsim 2000$ (for which $kL \gtrsim 10$). Indeed, extrapolating the data shown in Fig. 7 suggests that, to within the measurement error, the minimum-image Γ_m attains the asymptotic value only for $\nu \gtrsim 12$ (i.e., $N \gtrsim 3456$).

Previous simulations of Yukawa systems have focused mostly on the large κ regime, where the "minimum image" method gives a reasonably adequate representation of periodic boundary conditions. For example, Robbins *et al.* [7] present results for cases with $\kappa \gtrsim 1$ (note that κ and the parameter λ used in [7] are related by $\kappa = (3/4\pi)^{1/3}\,\lambda \approx 0.62\,\lambda$). Since they also truncate pairwise interactions at a cutoff radius $r_c \approx 3a$, the results in [7] probably have some systematic error at smaller κ values. For the weak-screening regime $\kappa \lesssim 1$, one certainly needs full periodic boundary conditions for quantitative thermodynamic measurements representative of infinite systems.

The results presented above are for illustrative purposes only; accurate determinations of Γ_m are based on finding the intersection of free energy curves for the solid and fluid phases (obtained by fitting energy data from the simulations to appropriate functional forms, e.g., as in [22] for the OCP).

6. CONCLUDING REMARKS

While periodic boundary conditions may incur nonphysical constraints on the simulated dynamics of condensed systems

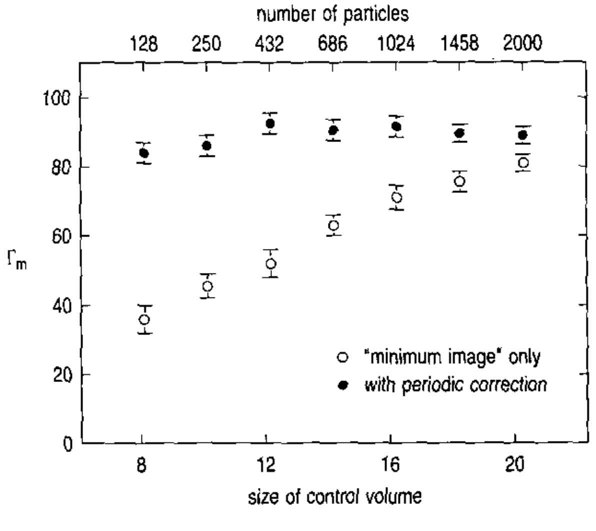

FIG. 7. Dependence of the observed melting point Γ_m of a Yukawa system with $\kappa = 0.5$ on the particle number N when: (a) the "minimum-image" convention only is used (open dots); and (b) the periodic correction term is incorporated (solid dots).

[31–34], they are still the best means of avoiding gross errors due to "surface terms" when estimating intensive thermodynamic properties by small-N Monte Carlo or molecular dynamics calculations. We have described a general-purpose approximation scheme for incorporating *full* periodic boundary conditions that may be used with a variety of intermediate-range pair potentials (for which the "minimum image" method would be unsatisfactory).

The accuracy of the periodic-boundary correction term is limited only by the available memory for storing the spline coefficients, and the speed of evaluation is essentially *independent* of the accuracy. For potentials such as those discussed in Section 1, relative approximation errors of 10^{-6} or better are easily within the scope of modern scientific workstations, allowing the quantitative study of dense "strongly coupled" systems at low temperatures.

Although, for a given N, inclusion of the correction term increases the computational cost by a constant factor, it should be borne in mind that values for thermodynamic variables representative of the $N \to \infty$ limit are obtainable with far smaller particle numbers under full periodic boundary conditions (except for short-range interactions). As shown in Section 5 above, when using full periodic boundary conditions, physically meaningful results may be observed with N an order of magnitude or more smaller than the minimum-image method requires. Since the run time is proportional to N^2, the scheme proposed above amounts to a significant enhancement in the speed with which physical data of a prescribed quality can be deduced from simulations.

ACKNOWLEDGMENT

We thank C. A. Micchelli for helpful discussions.

REFERENCES

1. M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford Univ. Press, Oxford, 1987).
2. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *J. Chem. Phys.* **21**, 1087 (1953).
3. E. J. W. Verwey and J. Th. G. Overbeek, *Theory of the Stability of Lyophobic Colloids* (Elsevier, New York, 1948).
4. D. Hone, S. Alexander, P. M. Chaikin, and P. Pincus, *J. Chem. Phys.* **79**, 1474 (1983).
5. S. Alexander, P. M. Chaikin, P. Grant, G. J. Morales, P. Pincus, and D. Hone, *J. Chem. Phys.* **80**, 5776 (1984).
6. K. Kremer, M. O. Robbins, and G. S. Grest, *Phys. Rev. Lett.* **57**, 2694 (1986).
7. M. O. Robbins, K. Kremer, and G. S. Grest, *J. Chem. Phys.* **88**, 3286 (1988).
8. R. O. Rosenberg and D. Thirumalai, *Phys. Rev. A* **33**, 4473 (1986).
9. R. O. Rosenberg and D. Thirumalai, *Phys. Rev. A* **36**, 5690 (1987).
10. E. J. Meijer and D. Frenkel, *J. Chem. Phys.* **94**, 2269 (1991).
11. E. C. Whipple, T. G. Northrop, and D. A. Mendis, *J. Geophys. Res.* **90**, 7405 (1985).
12. H. Ikezi, *Phys. Fluids* **29**, 1764 (1986).
13. R. T. Farouki and S. Hamaguchi, *Appl. Phys. Lett.* **61**, 2973 (1992).
14. S. G. Brush, H. L. Sahlin, and E. Teller, *J. Chem. Phys.* **45**, 2102 (1966).
15. J.-P. Hansen, *Phys. Rev. A* **8**, 3096 (1973).
16. R. T. Farouki and S. Hamaguchi, *Phys. Rev. E* **47**, 4330 (1993).
17. H. Nagara, Y. Nagata, and T. Nakamura, *Phys. Rev. A* **36**, 1859 (1987).
18. D. H. E. Dubin, *Phys. Rev. A* **42**, 4972 (1990).
19. J. A. Barker, R. A. Fisher, and R. O. Watts, *Mol. Phys.* **21**, 657 (1971).
20. T. A. Andrea, W. C. Swope, and H. C. Andersen, *J. Chem. Phys.* **79**, 4576 (1983).
21. H. E. DeWitt and W. B. Hubbard, *Astrophys. J.* **205**, 295 (1976); see Ref. [22] also.
22. W. L. Slattery, G. D. Doolen, and H. E. DeWitt, *Phys. Rev. A* **21**, 2087 (1980).
23. H. L. Helfer, R. L. McCrory, and H. M. Van Horn, *J. Stat. Phys.* **37**, 577 (1984).
24. J. H. Ahlberg, E. N. Nilson, and J. L. Walsh, *The Theory of Splines and Their Applications* (Academic Press, New York, 1967).
25. C. de Boor, *A Practical Guide to Splines* (Springer-Verlag, New York, 1978).
26. L. L. Schumaker, *Spline Functions: Basic Theory* (Wiley, New York, 1981).
27. R. Bell, Document GG24-3611 IBM International Technical Support Center, Poughkeepsie, NY, 1990 (unpublished).
28. D. W. Schaefer and B. J. Ackerson, *Phys. Rev. Lett.* **35**, 1448 (1975).
29. R. Williams, R. S. Crandall, and P. J. Wojtowicz, *Phys. Rev. Lett.* **37**, 348 (1976).
30. F. F. Abraham, *Adv. Phys.* **35**, 1 (1986).
31. M. J. Mandell, *J. Stat. Phys.* **15**, 299 (1976).
32. D. J. Adams, *Chem. Phys. Lett.* **62**, 329 (1979).
33. L. R. Pratt and S. W. Haan, *J. Chem. Phys.* **74**, 1864 (1981).
34. J. D. Honeycutt and H. C. Andersen, *Chem. Phys. Lett.* **108**, 535 (1984).