How to mesh up Ewald sums. I. A theoretical and numerical comparison of various particle mesh routines

Markus Deserno and Christian Holm
Max-Planck-Institut für Polymerforschung, Ackermannweg 10, 55128 Mainz, Germany

(Received 1 May 1998; accepted 6 August 1998)

Standard Ewald sums, which calculate, e.g., the electrostatic energy or the force in periodically closed systems of charged particles, can be efficiently speeded up by the use of the fast Fourier transformation (FFT). In this article we investigate three algorithms for the FFT-accelerated Ewald sum, which have attracted widespread attention, namely, the so-called particle–particle–particle mesh (P^3M), particle mesh Ewald (PME), and smooth PME method. We present a unified view of the underlying techniques and the various ingredients which comprise those routines. Additionally, we offer detailed accuracy measurements, which shed some light on the influence of several tuning parameters and also show that the existing methods — although similar in spirit — exhibit remarkable differences in accuracy. We propose a set of combinations of the individual components, mostly relying on the P^3M approach, that we regard to be the most flexible. The issue of estimating the errors connected with particle mesh routines is reserved to paper II. © *1998 American Institute of Physics.* [S0021-9606(98)51542-5]

I. INTRODUCTION

A challenging task in every computer simulation of particles which are subject to periodic boundary conditions and long range interactions is the efficient calculation of quantities like the interparticle forces or the interaction energies. The famous Ewald sum[1,2] does a remarkable job in splitting the very slowly (not even unconditionally) converging sum over the Coulomb potential into two sums which converge exponentially fast. Still, this method suffers from two deficits: First, it is computationally demanding, since one part of the problem is solved in reciprocal space, thereby implying the need for several Fourier transformations. Second, the algorithm scales like N^2 (with N being the number of charged particles in the simulation box) or at best like $N^{3/2}$, if one uses cutoffs which are optimized with respect to the splitting parameter.[3]

Several methods have been proposed to tackle the first problem, e.g., tabulation of the complete Ewald potential[4] or the use of polynomial approximations, in particular expansion of the nonspherical contributions to the Ewald potential in cubic harmonics.[4,5] Apart from the difficulty of a computational overhead which might strongly increase with the desired accuracy, all these methods do not solve the second problem: the unfavorable scaling with particle number.

The essential idea is *not* to avoid the Fourier transforms but to modify the problem in a way that permits an employment of the *fast Fourier transformation*[6] (FFT), thereby reducing the complexity of the reciprocal part of the Ewald sum to essentially order $N \log N$. If the real space cutoff is chosen small enough, this scaling applies to the complete Ewald sum. Since the FFT is a grid transformation, there are discretization problems to be solved and corresponding discretization errors to be minimized.

At present there exist several mesh implementations of the Ewald sum — similar in spirit but different in detail. In this article we will focus on the original particle–particle–particle–mesh (P^3M) method of Hockney and Eastwood[7] and two variants, namely, the particle mesh Ewald (PME) method of Darden *et al.*[8] and an upgrade of the latter by Essmann *et al.*,[9] which we will refer to as SPME (the ''S'' stands for ''smooth'').

There have been some uncertainties in the literature concerning the relative performance of these methods and it has been shown previously[10] that the P^3M approach — the oldest of the three — is actually the most accurate one and should be the preferred choice. However, since in this reference the PME method was combined with a disadvantageous charge assignment scheme and the more recent SPME could not be considered, we found it worthwhile to again test these three methods under similar well posed and reproducible conditions and a larger number of tuning parameters.

The original literature on particle mesh routines is mostly not easy to digest for the layman, obscured by the fact that the various authors approach the problem from different directions and use different notations. In this article we try to present a unified view of the common methods and analyze in detail the ingredients comprising them. By this we want to uncover the large number of possibilities for combining the different parts, thus allowing a judicious balance of accuracy, speed and ease of implementation. Moreover, we show that due to some subtle interdependencies not all combinations are advantageous, although they might appear promising at first sight.

This paper is structured as follows: First, we briefly review the idea of the Ewald sum and provide the most important formulas. Then we describe in some detail the steps which must be carried out if FFT algorithms are to be employed for the Fourier transformation, namely: charge assignment onto a mesh, solving Poisson's equation on that mesh, differentiating the potential to obtain the forces and

interpolating the mesh based forces back to the particles. All these steps can be performed in different ways and in the following section we investigate in detail their accuracy, in particular, we compare P^3M, PME and SPME with respect to their root mean square error in the force. Based on our theoretical and numerical investigations, we find that the most accurate and versatile routine is the P^3M method, supplemented by ingredients (dealing with the differentiation) of the other two approaches.

The important task of an optimal tuning of the parameters — especially the Ewald parameter α — is satisfactorily solved for the standard Ewald and the PME method, since there exist accurate analytic estimates for the root mean square error in the force. We will tackle the corresponding problem for P^3M in a second publication,[11] which we will refer to as paper II.

II. THE EWALD SUM

There are many examples of long range interactions which can be treated by Ewald techniques, but in this paper we will solely be concerned with Coulomb point charges, i.e., with an interaction potential $1/r$. Consider therefore a system of N particles with charges q_i at positions $\mathbf{r}_i$ in an overall neutral and (for simplicity) cubic simulation box of length L and volume $V_b=L^3$. If periodic boundary conditions are applied, the total electrostatic energy of the box is given by

$$E=\frac{1}{2}\sum_{i,j=1}^{N}{}'\sum_{\mathbf{n}\in\mathbb{Z}^3}\frac{q_iq_j}{|\mathbf{r}_{ij}+\mathbf{n}L|}. \tag{1}$$

The sum over $\mathbf{n}$ takes into account the periodic images of the charges and the prime indicates that in the case $i=j$ the term $\mathbf{n}=0$ must be omitted. Of course $\mathbf{r}_{ij}=\mathbf{r}_i-\mathbf{r}_j$, and our unit conventions are shortly described in Appendix A.

Strictly speaking, since this sum is only *conditionally* convergent, its value is not well defined unless one specifies the precise way in which the cluster of simulation boxes is supposed to fill the $\mathbb{R}^3$, i.e., its shape (e.g., approximately spherical[12]) and the conditions outside the cluster (e.g., vacuum or some dielectric). A thorough discussion is given elsewhere.[2,13]

The slowly decaying long range part of the Coulomb potential renders a straightforward summation of Eq. (1) impracticable. The trick is to split the problem into two parts by the following trivial identity:

$$\frac{1}{r}=\frac{f(r)}{r}+\frac{1-f(r)}{r}. \tag{2}$$

The underlying idea is to distribute the two main complications of the Coulomb potential — its rapid variation at small r and its slow decay at large r — between the two terms by a suitable choice of f. In particular:

(1) The first part $f(r)/r$ should be negligible (or even zero) beyond some cutoff $r_{\max}$, so that summation up to the cutoff is a good approximation to (or the exact result of) this contribution to the total electrostatic potential.

(2) The second part $[1-f(r)]/r$ should be a slowly varying function for *all* r, so that its Fourier transform can be represented by only a few $\mathbf{k}$ vectors with $|\mathbf{k}|\leqslant k_{\max}$. This permits an efficient calculation of this contribution to the total electrostatic potential in reciprocal space.

Since the field equations are linear, the sum of these two contributions gives the solution for the potential of the original problem. However, the two requirements on the function f mentioned above leave a large freedom of choice.[14] The traditional selection is the complementary error function $\operatorname{erfc}(r):=2\pi^{-1/2}\int_r^\infty dt\,\exp(-t^2)$, which results in the well-known Ewald formula for the electrostatic *energy* of the box:

$$E=E^{(r)}+E^{(k)}+E^{(s)}+E^{(d)}, \tag{3}$$

where the contribution from real space $E^{(r)}$, the contribution from reciprocal space $E^{(k)}$, the self-energy $E^{(s)}$ and the dipole correction $E^{(d)}$ are given by

$$E^{(r)}=\frac{1}{2}\sum_{i,j}\sum_{\mathbf{m}\in\mathbb{Z}^3}{}'\,q_iq_j\frac{\operatorname{erfc}(\alpha|\mathbf{r}_{ij}+\mathbf{m}L|)}{|\mathbf{r}_{ij}+\mathbf{m}L|}, \tag{4}$$

$$E^{(k)}=\frac{1}{2L^3}\sum_{\mathbf{k}\neq 0}\frac{4\pi}{k^2}e^{-k^2/4\alpha^2}|\tilde{\rho}(\mathbf{k})|^2, \tag{5}$$

$$E^{(s)}=-\frac{\alpha}{\sqrt{\pi}}\sum_i q_i^2, \tag{6}$$

$$E^{(d)}=\frac{2\pi}{(1+2\epsilon')L^3}\left(\sum_i q_i\mathbf{r}_i\right)^2, \tag{7}$$

and the Fourier transformed charge density $\tilde{\rho}(\mathbf{k})$ is defined as

$$\tilde{\rho}(\mathbf{k})=\int_{V_b}d^3r\,\rho(\mathbf{r})e^{-i\mathbf{k}\cdot\mathbf{r}}=\sum_{j=1}^{N}q_je^{-i\,\mathbf{k}\cdot\mathbf{r}_j}. \tag{8}$$

The inverse length α, which we will refer to as the *Ewald parameter*, tunes the relative weight of the real space and the reciprocal space contribution, but the final result is of course independent of α. The $\mathbf{k}$-vectors form the discrete set $\{2\pi\mathbf{n}/L:\mathbf{n}\in\mathbb{Z}^3\}$.

The form (7) given for the dipole correction assumes that the set of periodic replications of the simulation box tends in a spherical way toward an infinite cluster and that the medium outside this sphere is a homogeneous dielectric[2,13] with dielectric constant ϵ'. Note that the case of a surrounding vacuum corresponds to $\epsilon'=1$ and that the dipole correction vanishes for metallic boundary conditions, since then $\epsilon'=\infty$. Note also that this term is independent of α, which again shows that it is not specific to the Ewald sum but more generally reflects the problems inherent to the conditional convergence of the sum in Eq. (1). Some complications regarding the correct implementation of this term are discussed by Caillol.[13]

The advantage of rewriting Eq. (1) this way is that the exponentially converging sums over $\mathbf{m}$ and $\mathbf{k}$ in Eqs. (4) and (5) allow the introduction of relatively small cutoffs without much loss in accuracy. Typically one chooses α large enough as to employ the minimum image convention in Eq. (4). It is important to realize that at given real and reciprocal space cutoffs there exists an *optimal* α such that the accuracy

of the approximated Ewald sum is as high as possible. This optimal value can be determined easily with the help of the excellent estimates for the cutoff errors derived by Kolafa and Perram[15] — essentially by demanding that the real and reciprocal space contribution to the error should be equal.

Some more insight into the Ewald sum can be gained by the following considerations. Let $\tilde{g}(\mathbf{k}) := 4\pi/k^2$ be the Fourier transformed Green function of the Coulomb potential $1/r$ and $\tilde{\gamma}(\mathbf{k}) := \exp(-k^2/4\alpha^2)$. Then Eq. (5) can be rewritten as follows:

$$E^{(k)} = \frac{1}{2}\sum_j q_j \left(\frac{1}{L^3} \sum_{\mathbf{k}\neq 0} \tilde{g}(\mathbf{k})\tilde{\gamma}(\mathbf{k})\tilde{\rho}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}_j} \right) =: \frac{1}{2}\sum_j q_j \phi^{(k)}(\mathbf{r}_j). \tag{9}$$

Here $\phi^{(k)}(\mathbf{r}_j)$ is the electrostatic potential at the point $\mathbf{r}_j$ due to the *second* term in Eq. (2) and by definition it is clear that its Fourier transform is given by

$$\tilde{\phi}^{(k)}(\mathbf{k}) = \tilde{g}(\mathbf{k})\tilde{\gamma}(\mathbf{k})\tilde{\rho}(\mathbf{k}). \tag{10}$$

As is known, products in reciprocal space correspond to convolutions in real space. Hence Eq. (10) shows that the reciprocal space contribution to the electrostatic potential is created by a charge distribution which is obtained from the original point charge distribution by a convolution with a ''smearing function'' $\gamma(\mathbf{r})$.

For the standard Ewald sum $\gamma(\mathbf{r})$ is a Gaussian, i.e., $\gamma(\mathbf{r}) = \alpha^3\pi^{-3/2}\exp(-\alpha^2 r^2)$, but this is merely a consequence of choosing the splitting function f in Eq. (2) to be the complementary error function. In fact, an alternative method[16] to motivate the splitting which was done in Eq. (2) is to replace the point charge distribution ρ by a screened charge distribution $\rho - \rho\star\gamma$ and compensate this screening by adding the smeared charge distribution $\rho\star\gamma$. (The star denotes the convolution operation.) From a mathematical point of view these two interpretations are perfectly equivalent: Instead of splitting the potential one splits the charge density.

At this point a word of caution seems appropriate: Whereas the electrostatic *potential* depends linearly on the charge density, the electrostatic *energy* does not. Thus, calculating the energies resulting from the charge densities $\rho - \rho\star\gamma$ and $\rho\star\gamma$ and adding these contributions together would *not* give the energy of the charge density ρ. Consequently, $E^{(k)}$ is not the electrostatic energy of a charge density $\rho\star\gamma$ but the *Fourier space contribution* to the electrostatic energy of the charge density ρ. We want to make this subtle point more clear by writing down the energy explicitly. If we denote with ϕ_ρ the potential originating from ρ, we have due to the linear dependence of ϕ_ρ on ρ an equation like $\phi_\rho = \phi_{\rho-\rho\star\gamma} + \phi_{\rho\star\gamma}$. Hence we can obtain for the electrostatic energy the following expression:

$$\begin{aligned} E' &= \frac{1}{2}\int \mathrm{d}^3 r\, \rho(\mathbf{r})\, \phi_\rho(\mathbf{r}) \\ &= \frac{1}{2}\int \mathrm{d}^3 r\, \rho(\mathbf{r})\, [\phi_{\rho-\rho\star\gamma}(\mathbf{r}) + \phi_{\rho\star\gamma}(\mathbf{r})] \\ &= \frac{1}{2}\int \mathrm{d}^3 r\, \rho(\mathbf{r})\, \phi_{\rho-\rho\star\gamma}(\mathbf{r}) + \frac{1}{2}\int \mathrm{d}^3 r\, \rho(\mathbf{r})\, \phi_{\rho\star\gamma}(\mathbf{r}). \end{aligned} \tag{11}$$

The two terms in the last line are the real space and the Fourier space contribution to the energy, but neither of them can be interpreted as the energy of a charge distribution $\rho - \rho\star\gamma$ or $\rho\star\gamma$! Moreover, the quantity E' contains unphysical self-energy contributions, i.e., energy due to the interaction of a charge (or γ-smeared charge) with itself. In the actual Ewald sum the self-energy contribution of the real space part is canceled by omitting the term $\mathbf{m}=0$ for $i=j$ in Eq. (4), whereas the self-energy contribution of $E^{(k)}$ must be subtracted separately (this is the origin of the term $E^{(s)}$).

Finally, the force $\mathbf{F}_i$ on particle i is obtained by differentiating the electrostatic potential energy E with respect to $\mathbf{r}_i$, i.e.,

$$\mathbf{F}_i = -\frac{\partial}{\partial \mathbf{r}_i} E. \tag{12}$$

Using Eqs. (3)–(8) one obtains the following Ewald formula for the forces:

$$\mathbf{F}_i = \mathbf{F}_i^{(r)} + \mathbf{F}_i^{(k)} + \mathbf{F}_i^{(d)} \tag{13}$$

with the real space, Fourier space and dipole contributions, respectively, given by

$$\mathbf{F}_i^{(r)} = q_i \sum_j{}' q_j \sum_{\mathbf{m}\in\mathbb{Z}^3} \left(\frac{2\alpha}{\sqrt{\pi}} \exp(-\alpha^2 |\mathbf{r}_{ij} + \mathbf{m}L|^2) + \frac{\operatorname{erfc}(\alpha|\mathbf{r}_{ij}+\mathbf{m}L|)}{|\mathbf{r}_{ij}+\mathbf{m}L|} \right) \frac{\mathbf{r}_{ij}+\mathbf{m}L}{|\mathbf{r}_{ij}+\mathbf{m}L|^2}, \tag{14}$$

$$\mathbf{F}_i^{(k)} = \frac{q_i}{L^3}\sum_j q_j \sum_{\mathbf{k}\neq 0} \frac{4\pi\mathbf{k}}{k^2} \exp\left(-\frac{k^2}{4\alpha^2} \right) \sin(\mathbf{k}\cdot\mathbf{r}_{ij}), \tag{15}$$

$$\mathbf{F}_i^{(d)} = -\frac{4\pi q_i}{(1+2\epsilon')L^3} \sum_j q_j \mathbf{r}_j. \tag{16}$$

Since the self-energy from Eq. (6) is independent of particle positions, it does not contribute to the force.

III. EWALD SUMMATION ON A GRID

Performing the Fourier transformations inherent to the reciprocal space part of the Ewald sum by FFT routines is by no means a straightforward business. First, the point charges with their continuous coordinates have to be replaced by a grid based charge density, because the FFT is a *discrete* and *finite* Fourier transformation. Second, it is neither obvious nor true that the best grid approximation to the continuum solution of Poisson's equation is achieved by using the *continuum* Green function. Third, there are at least three ways

for implementing the differentiation needed in Eq. (12), which differ in accuracy and speed. And fourth, the procedure of assigning the forces calculated on the mesh back to the actual particles can — under certain circumstances — lead to unwanted violations of Newton's third law, which can be anything between harmless and disastrous.

The four steps involved in a particle mesh calculation are sources for various kinds of errors, originating, e.g., from discretization, interpolation or aliasing problems (with the latter we want to denote inaccuracies resulting from the fact that a *finite* grid cannot represent arbitrarily large **k** vectors). Since these contributions are not independent of each other (reducing one might enhance another), the only reasonable demand is the minimization of the *total* error at given computational effort.

One of our aims is to compile some of the possibilities for each step, in order to draw a comparison between the three mesh implementations mentioned in Sec. I — PME, SPME and P^3M. Like the Ewald sum, all these algorithms can be extended to a *triclinic* simulation cell by reverting to general dual basis vectors and one can also use a different number of grid points along each direction. However, in order to keep the notation simple, we restrict ourselves to the case of a cubic box and employ the same number of mesh points in each direction. How the generalizations can be done is described, e.g., in the references on PME[8] or SPME.[9]

A. Charge assignment

The actual *procedure* of assigning the charges to the grid can be written down very easily. We will first discuss the one-dimensional case, i.e., particles with coordinates $x \in [0;L] \subset \mathbb{R}$ have to be assigned to the mesh points $x_p \in \mathbb{M} = \{p\ h : p = 0, \ldots, N_M - 1\}$, where N_M is the *number* of mesh points and $h := L/N_M$ is their *spacing*. To keep the notation simple, we will abstain from *explicitly* taking into account that any x value, which is outside $[0;L]$, has to be folded back into this interval in order to conform to periodic boundary conditions. Rather, we assume that this is done as necessary, i.e., all calculations are to be understood ''modulo L.''

Define the even function $W(x)$ such that the fraction of charge which is assigned to the mesh point x_p due to a unit charge at position x is given by $W(x - x_p)$. If the charge density of the system is $\rho(x)$, then the *mesh based* charge density ρ_M, defined at the mesh points x_p, can be written as the following convolution:

$$\rho_M(x_p) = \frac{1}{h} \int_0^L \mathrm{d}x\, W(x_p - x)\, \rho(x). \tag{17}$$

The prefactor $1/h$ merely ensures that ρ_M is in fact a *density*. Henceforth we will refer to any such W as a *charge assignment function*.

The important question is: What *properties* should $W(x)$ have in order to be a suitable choice? The following wish list summarizes some desirable features:

(1) Charge conservation, i.e., the fractional charges of one particle, which have been distributed to the surrounding grid points, sum up to the total charge of that particle.
(2) Finite and if possible small support, because the computational cost increases with the number of mesh points among which the charge of each particle is distributed. (The support of a real-valued function f defined on X is (the closure of) the set $\{x \in X : f(x) \neq 0\}$, i.e., basically the range of values for which the function is nonzero.)
(3) Localization of discretization errors, i.e., inaccuracies in the force between two particles due to the discretization should become small with increasing particle separation.
(4) Large degree of smoothness, i.e., the fractional charge of particle i which is assigned to some mesh point x_p should be a smoothly varying function of the position of particle i.
(5) Minimization of aliasing errors, i.e., since on a finite grid there is only space for a limited number of **k** vectors, the charge assignment function should decay sufficiently rapidly in Fourier space.
(6) Easy and transparent implementation.

It is important to realize that these characteristics cannot be achieved all at the same time. Although some properties are positively correlated (e.g., a large degree of smoothness implies a fast decay in reciprocal space and thus minimizes aliasing errors), some other properties exclude each other (e.g., minimization of aliasing errors implies a sufficient localization in reciprocal space which is incompatible with a small support in real space). Thus, a good charge assignment function is always a compromise between these different demands.

We also want to stress that the choice of $W(x)$ is *not* independent of the other decisions made for the mesh implementation. In Appendix B we show that if one sticks to the *continuum* Coulomb Green function in the mesh calculation, the requirement of localization of discretization errors is enough to restrict the charge assignment to a *Lagrange interpolation scheme*. This is in fact the combination used for PME[8,17] (see also Appendix B). Hence, other influence functions can only be competitive if the Coulomb Green function is somehow adjusted at the same time.

The choice for the charge assignment function of Hockney and Eastwood is as follows: In a Pth order assignment scheme (i.e., the charge of one particle is distributed between its P nearest mesh points) define the Fourier transformed charge assignment function as

$$\tilde{W}^{(P)}(k) = h \left(\frac{\sin(kh/2)}{kh/2} \right)^P. \tag{18}$$

Transforming this back to real space gives

$$W^{(P)}(x) = \underbrace{(\chi_{[-\frac{1}{2},\frac{1}{2}]} \star \cdots \star \chi_{[-\frac{1}{2},\frac{1}{2}]})}_{P\text{-fold convolution}}(x/h) \tag{19}$$

with $\chi_{[-1/2,1/2]}$ being the characteristic function of the interval $[-\frac{1}{2},\frac{1}{2}]$, i.e., the function that is 1 within this interval and 0 outside. Thus, e.g., by the central limit theorem, the Pth order charge assignment function resembles (in this case) for increasing P more and more closely a centered Gaussian

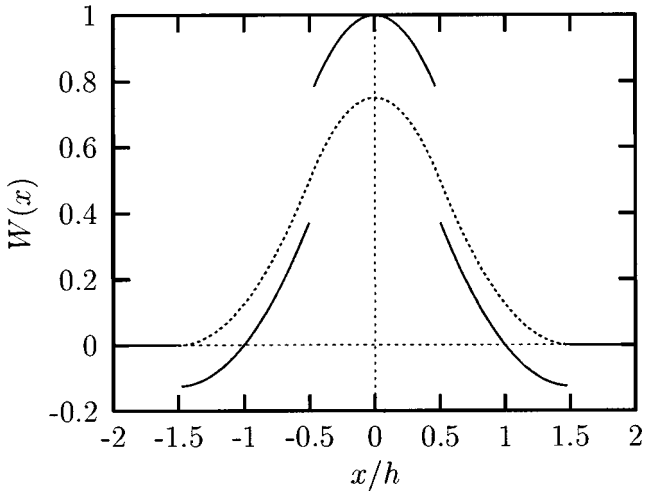

FIG. 1. Third-order charge assignment function for the Lagrange scheme [Refs. 8 and 17 — solid line] and the spline scheme [Ref. 7 — dotted line, see also Eq. (19)]. Both assignment functions have support $[-\frac{3}{2}h;\frac{3}{2}h]$ and are piecewise quadratic. While for all odd assignment orders the Lagrange assignment function is discontinuous (for even assignment order it is continuous but not differentiable), the spline assignment function is in general $P-2$ times continuously differentiable by construction. Note, however, that the Lagrange assignment function is optimized with respect to a different property, see Appendix B.

(with a variance P times as large as the variance of $W^{(1)}$), but it has *finite* support $[-Ph/2, Ph/2]$. This assignment function is very smooth for large P, since it is a spline of order P and thus $P-2$ times differentiable.[18] As a matter of convenience we decided to tabulate the corresponding charge fractions $W_p^{(P)}(x)=W^{(P)}(x-x_p)$ for $P\in\{1,\ldots,7\}$ in Appendix E.

In Fig. 1 the third-order assignment functions for the Lagrange and the spline interpolation scheme are plotted. Note that while the spline function is in general $P-2$ times continuously differentiable, the Lagrange assignment function is not as smooth: Generally, for even assignment orders it is continuous, but the derivative is not, while for odd assignment orders it is discontinuous right away. Incidentally, for $P=1$ and $P=2$ both schemes coincide.

The SPME method uses in essence the same charge assignment functions as the P^3M method, but this is discussed more appropriately in Sec. III B.

Charge assignment in more than one dimension can be achieved by a simple factorization approach. For example, the three-dimensional charge assignment function $W(\mathbf{r})$ can be written as

$$W(\mathbf{r})=W(x)W(y)W(z). \tag{20}$$

This is certainly not the only possibility,[7] but it is computationally advantageous.

The generalization of Eq. (17) to three dimensions can be written as

$$\rho_M(\mathbf{r}_p)=\frac{1}{h^3}\int_{L^3}\mathrm{d}^3r\, W(\mathbf{r}_p-\mathbf{r})\,\rho(\mathbf{r}) \tag{21}$$

$$=\frac{1}{h^3}\sum_{i=1}^{N} q_i W(\mathbf{r}_p-\mathbf{r}_i). \tag{22}$$

In Eq. (22) the reader should not confuse the coordinate of particle i, $\mathbf{r}_i$, with the coordinate of mesh point p, $\mathbf{r}_p$.

B. Solving Poisson's equation

For the standard Ewald sum the Fourier space contribution to the electrostatic energy is given by Eq. (5). How is this equation to be modified, now that we are working on a discrete mesh?

The simplest approach is used in the PME method,[8] where it is assumed that this equation is appropriate in the discrete case as well. The only difference is that the Fourier transformed charge density $\tilde{\rho}$ from Eq. (8) is replaced by the *finite Fourier transform* of the mesh based charge density, $\hat{\rho}_M$, which we define as

$$\hat{\rho}_M(\mathbf{k}):=h^3\sum_{\mathbf{r}_p\in\mathbb{M}}\rho_M(\mathbf{r}_p)e^{-i\,\mathbf{k}\cdot\mathbf{r}_p}, \tag{23}$$

where $\Sigma_{\mathbf{r}_p\in\mathbb{M}}$ is the sum over the (three-dimensional) mesh in real space and the $\mathbf{k}$-vectors are from the corresponding Fourier space mesh. Of course, the way back to real space is also done by a (inverse) finite Fourier transform. (In order to distinguish between the usual and the finite Fourier transform, we indicate the latter by a caret and not by a tilde.) As discussed in Sec. III A and Appendix B, the usage of the continuum Coulomb Green function is best accompanied by a Lagrange interpolation scheme for the charge assignment. See, e.g., Petersen[17] for a tabulation of the corresponding polynomials and their implementation.

A second algorithm, the SPME method, was presented by Essmann *et al.*[9] It uses a smooth charge assignment scheme and hence an adjusted Green function. The reasoning is as follows: Starting with Eqs. (5) and (8) it is argued that charge assignment onto the mesh can equivalently be viewed as interpolating exponentials of the form $\exp(ikx)$ at discrete grid points. This problem has a particularly elegant solution, the so-called *exponential Euler splines*.[18] If x is the continuous particle coordinate, we have for even P (a recipe for the treatment of odd P can be found in the original SPME reference—Ref. 9):

$$e^{ikx}\approx b(k)\sum_{l\in\mathbb{Z}}M^{(P)}(x-lh)e^{iklh} \tag{24}$$

with

$$b(k)=\frac{e^{ikPh}}{\Sigma_{l=1}^{P-1}M^{(P)}(lh)e^{iklh}}. \tag{25}$$

The function $M^{(P)}$ is a cardinal-B-spline of order P, and Essmann *et al.*[9] give a recursive definition. They also point out that (in our notation for $h=1$) the function $M^{(P)}$ is identical to the probability distribution of the sum of P independent random variables, each distributed uniformly on the unit interval. Since this distribution is given by the P-fold convolution of the characteristic function $\chi_{[0;1]}$ with itself, we can see by comparison with Eq. (19) that the charge assignment functions from the SPME and the P^3M method are in fact

identical up to a translation: $M^{(P)}(x)=W^{(P)}(x-Ph/2)$. Of course, $M^{(P)}$ has finite support $[0;Ph]$, so the sum in Eq. (24) is actually finite.

Note that $M^{(P)}$ is not an even function, and in the original reference on SPME[9] the charge assignment differs slightly from our Eq. (22). However, the only effect of the translation is that the original system is represented by a *shifted* mesh system, which is from a practical point of view irrelevant, because this shift is undone in the back-interpolation (if accomplished with the same assignment function).

Now the following approximation for $E^{(k)}$ can be derived by inserting Eqs. (24) and (25) into Eq. (5):

$$E^{(k)}\approx\frac{1}{2}\sum_{\mathbf{r}_p\in\mathbb{M}}h^3\rho_M(\mathbf{r}_p)[\rho_M\star G](\mathbf{r}_p). \tag{26}$$

Here the star denotes the finite convolution

$$[\rho_M\star G](\mathbf{r}_p)=h^3\sum_{\mathbf{r}_q\in\mathbb{M}}\rho_M(\mathbf{r}_q)G(\mathbf{r}_p-\mathbf{r}_q) \tag{27}$$

(again, the periodic closure is not written down explicitly) and the function G is given by its finite Fourier transform

$$\hat{G}(\mathbf{k})=B(\mathbf{k})\sum_{\mathbf{m}\in\mathbb{Z}^3}\frac{4\pi}{\left(\mathbf{k}+\frac{2\pi}{h}\mathbf{m}\right)^2}\tilde{\gamma}\left(\mathbf{k}+\frac{2\pi}{h}\mathbf{m}\right) \tag{28}$$

with $B(\mathbf{k}):=|b(k_x)b(k_y)b(k_z)|^2$. Following Hockney and Eastwood we will refer to G as the *influence function*. The nice thing about G is that it is by construction independent of particle coordinates and can therefore be precomputed.

Equation (26) can be made plausible in the following way: G plays the role of a Coulomb Green function which has incorporated the ''smearing'' with the Gaussian γ. Hence, its convolution with the mesh based point charge density gives the mesh based electrostatic potential of γ-smeared charges. Multiplying this with the mesh based charge $h^3\rho_M$ and summing over all mesh points gives the Fourier space contribution to the electrostatic energy up to a factor 1/2, which merely cancels some double counting. This should be compared to Eq. (9) or the second term in Eq. (11).

As pointed out before, a charge assignment different from the Lagrange interpolation scheme can only be competitive if the Coulomb Green function is changed at the same time. The replacement of the usual (and smeared) Green function $g\star\gamma$ with the influence function G, which essentially differs by the additional prefactor B in Fourier space, achieves exactly that. Conversely, PME uses a Lagrange interpolation scheme together with the unchanged Coulomb Green function, i.e., Eqs. (26) and (28) with $B\equiv 1$.

Finally, the alias sum occurring in Eq. (28) is substituted according to the following rule:[9] If N_M is the number of mesh points in each direction and the vector $\mathbf{k}$ on the left-hand side is given by $\mathbf{k}=2\pi\mathbf{n}/L$, $\mathbf{n}\in\{0,\ldots,N_M-1\}^3$, then define $\hat{G}(\mathbf{k})=B(\mathbf{k})\tilde{g}(\mathbf{k}')\tilde{\gamma}(\mathbf{k}')$, where $\mathbf{k}'=2\pi\mathbf{n}'/L$ and $n_i'=n_i$ for $0\leqslant n_i\leqslant N_M/2$ and $n_i'=n_i-N_M$ otherwise ($i=x,y,z$).

A third possibility — the so-called P^{3}M method — was presented by Hockney and Eastwood:[7] Their objective was an optimization of the influence function G in Eq. (26), which causes the final result of the *mesh* calculation to be as close as possible to the original *continuum* problem. So in order to proceed one first has to make the statement ''as close as possible'' more quantitative, and this can be done as follows:

Take two particles with coordinates $\mathbf{r}_1$ and $\mathbf{r}_2$ and define $\mathbf{r}:=\mathbf{r}_1-\mathbf{r}_2$. The true force between these particles should be a function of $\mathbf{r}$ only, but in any mesh implementation the actual force also depends on the positions of the particles relative to the mesh, say, on the position of the first particle within its mesh cell (i.e., the original translational symmetry is broken by the mesh). This suggests the following measure for the error: Integrate the square of the difference between the calculated force $\mathbf{F}$ and the true reference force $\mathbf{R}$ over all values of $\mathbf{r}$ and average this quantity over all positions of, e.g., the first particle within one particular mesh cell:

$$Q:=\frac{1}{V_c}\int_{V_c}\mathrm{d}^3r_1\int_{V_b}\mathrm{d}^3r[\mathbf{F}(\mathbf{r};\mathbf{r}_1)-\mathbf{R}(\mathbf{r})]^2. \tag{29}$$

Here $V_c=h^3$ is the volume of one mesh cell. The solution of Poisson's equation is accomplished in essence by Eq. (26) (it is only written down somewhat differently), and the derivative in Eq. (12) is performed by applying finite difference operators to the mesh based electrostatic potential (see below). Since the discretization error Q can be regarded as a functional of $\hat{G}$, the *optimal* influence function $\hat{G}_{\mathrm{opt}}$ can be obtained by setting the *functional derivative* of Q with respect to $\hat{G}$ to zero, i.e.,

$$\frac{\delta Q}{\delta\hat{G}}\Big|_{\hat{G}=\hat{G}_{\mathrm{opt}}}=0. \tag{30}$$

Starting from this idea, Hockney and Eastwood were able to derive the following expression for $\hat{G}_{\mathrm{opt}}$:[7]

$$\hat{G}_{\mathrm{opt}}(\mathbf{k})=\frac{\tilde{\mathbf{D}}(\mathbf{k})\cdot\sum_{\mathbf{m}\in\mathbb{Z}^3}\tilde{U}^2\left(\mathbf{k}+\frac{2\pi}{h}\mathbf{m}\right)\tilde{\mathbf{R}}\left(\mathbf{k}+\frac{2\pi}{h}\mathbf{m}\right)}{|\tilde{\mathbf{D}}(\mathbf{k})|^2\left[\sum_{\mathbf{m}\in\mathbb{Z}^3}\tilde{U}^2\left(\mathbf{k}+\frac{2\pi}{h}\mathbf{m}\right)\right]^2}. \tag{31}$$

Here $\tilde{\mathbf{D}}(\mathbf{k})$ is the Fourier transform of the employed differentiation operator (see Sec. III C and Appendix C), $\tilde{U}(\mathbf{k})=\tilde{W}(\mathbf{k})/V_c$ is the Fourier transform of the charge assignment function divided by the volume of one mesh cell and $\tilde{\mathbf{R}}(\mathbf{k})$ is the Fourier transform of the true reference force, given by

$$\tilde{\mathbf{R}}(\mathbf{k})=-i\mathbf{k}\tilde{g}(\mathbf{k})\tilde{\gamma}(\mathbf{k}). \tag{32}$$

Note that this differs from the expression from the book of Hockney and Eastwood,[7] who use $\tilde{\gamma}^2$ instead of $\tilde{\gamma}$. The reason is that Eq. (32) describes the true reference force between a γ-smeared charge and a point charge, while Hockney and Eastwood choose a slightly different approach in which they need the force between two γ-smeared charges.

Also, we keep the factor 4π in the Fourier transformed Green function $\tilde{g}$ and do not to hide it somewhere else.

The alias sums over $\mathbf{m}$ in Eq. (31) are typically well converged for $|\mathbf{m}| \leqslant 2$ and the sum in the denominator could even be done analytically. Again we want to emphasize that the calculation of the influence function has to be done only once prior to the actual simulation and thus does not produce any runtime overhead. Note also that the expression (28) differs from the optimal form (31) and hence cannot be optimal.

A final word concerning the implementation: Although the convolution $\rho_M \star G$ in Eq. (26) is a nice and compact notation, the whole purpose of these particle mesh routines is to employ the *convolution theorem* and use efficient FFT routines to calculate $\rho_M \star G$. The central steps are as follows:

(1) Calculate the finite Fourier transform $\hat{\rho}_M$ of the mesh based charge density ρ_M.
(2) Multiply $\hat{\rho}_M$ with the precomputed Fourier space representation of the influence function, $\hat{G}$.
(3) Apply an inverse finite Fourier transform to this product to end up with the finite convolution of ρ_M with G. Formally this can be symbolized as

$$\rho_M \star G = \overleftarrow{\mathrm{FFT}}\left[\overrightarrow{\mathrm{FFT}}[\rho_M] \times \overrightarrow{\mathrm{FFT}}[G] \right]. \tag{33}$$

Note that in this way one only needs $\hat{G}$ to calculate $\rho_M \star G$ but actually never G itself.

This is the important part which all particle mesh algorithms have in common. The various methods differ, e.g., in their choice of G, the assignment function W or the implementation of the derivative in Eq. (12).

C. Differentiation

After the calculation of the electrostatic energy, the forces on the particles are obtained by differentiation according to Eq. (12). However, for the Fourier space part of particle mesh methods there are several possibilities to implement this procedure. In other words, there exist several possible substitutes for Eq. (15), in particular:

(1) differentiation in Fourier space,
(2) analytic differentiation of the assignment function in real space,
(3) discrete differentiation on the mesh in real space.

Differentiation in Fourier space is easy, since it merely involves a multiplication with the Fourier transformed differentiation operator $\tilde{\mathbf{D}}(\mathbf{k})$, which is a fast, local and accurate operation. Although one might want to use Fourier transforms of discrete difference operators — allowing for the fact that one is actually working on a mesh — the best results are obtained when the Fourier transform of the usual differential operator, namely $i\mathbf{k}$, is employed. Therefore we will refer to this method as $i\mathbf{k}$ differentiation. The basic idea is not to calculate the mesh based electrostatic potential $\phi^{(k)}(\mathbf{r}_p)$ via Eq. (33) but the mesh based electric field $\mathbf{E}(\mathbf{r}_p)$ by the following simple change to this equation:

$$\begin{aligned}\mathbf{E}(\mathbf{r}_p) &= -\frac{\partial}{\partial \mathbf{r}_p}\phi^{(k)}(\mathbf{r}_p) = -\frac{\partial}{\partial \mathbf{r}_p}[\rho_M \star G](\mathbf{r}_p) \\ &= -\overleftarrow{\mathrm{FFT}}[i\mathbf{k} \times \hat{\rho}_M \times \hat{G}](\mathbf{r}_p).\end{aligned} \tag{34}$$

This method is employed in the PME algorithm and, as shown later, leads to the most accurate force calculations, if it is used in conjunction with the optimal influence function from Eq. (31). Note, however, that since $\mathbf{k}$ is a vector, there are in fact *three* inverse three-dimensional Fourier transforms to be calculated in Eq. (34), which is obviously computationally demanding.

The electrostatic energy calculated on the mesh depends on the particle coordinates through the arguments of the charge assignment function W. As the creators of the SPME method point out,[9] a smooth charge assignment scheme permits an *analytic* differentiation of the energy, since the quantity ρ_M, which contains the particle coordinates $\mathbf{r}_i$, depends in a differentiable way on the $\mathbf{r}_i$. Using Eqs. (12) and (26) and the fact that G is independent of particle coordinates and an even function (since $\hat{G}$ is even), one can derive

$$\begin{aligned}\mathbf{F}_i &\approx -\frac{\partial}{\partial \mathbf{r}_i}\frac{1}{2}\sum_{\mathbf{r}_p \in \mathbb{M}} h^3 \rho_M(\mathbf{r}_p)[\rho_M \star G](\mathbf{r}_p) \\ &= -\frac{1}{2}h^6 \sum_{\mathbf{r}_p, \mathbf{r}_q \in \mathbb{M}} \left(\frac{\partial \rho_M}{\partial \mathbf{r}_i}(\mathbf{r}_p)\rho_M(\mathbf{r}_q) \right. \\ &\quad \left. + \rho_M(\mathbf{r}_p)\frac{\partial \rho_M}{\partial \mathbf{r}_i}(\mathbf{r}_q) \right) G(\mathbf{r}_p - \mathbf{r}_q) \\ &= -h^6 \sum_{\mathbf{r}_p, \mathbf{r}_q \in \mathbb{M}} \frac{\partial \rho_M}{\partial \mathbf{r}_i}(\mathbf{r}_p)\rho_M(\mathbf{r}_q)\frac{1}{2}\left(G(\mathbf{r}_p - \mathbf{r}_q) \right. \\ &\quad \left. + G(\mathbf{r}_q - \mathbf{r}_p) \right) \\ &= -h^3 \sum_{\mathbf{r}_p \in \mathbb{M}} \frac{\partial \rho_M}{\partial \mathbf{r}_i}(\mathbf{r}_p)[\rho_M \star G](\mathbf{r}_p).\end{aligned} \tag{35}$$

From Eq. (21) it is obvious that the array $[\partial \rho_M / \partial \mathbf{r}_i](\mathbf{r}_p)$ is essentially obtained by a charge assignment scheme which uses the *gradient* of the assignment function W and can thus be calculated conveniently at the same time as ρ_M. Since only one Fourier transform back to real space is necessary, this procedure is indeed very fast. Unfortunately, this differentiation scheme leads to a small random particle drift, since momentum is not conserved any more (see Sec. III D). Although the *total* momentum of the simulation box can be kept constant by subtracting the mean force $(1/N)\,\Sigma_i \mathbf{F}_i$ from each particle, the small reduction in the accuracy of the particle forces due to these *local* random fluctuations can only be compensated marginally by this *global* correction.

A third possibility for implementing the derivative in Eq. (12) is the use of finite difference operators, which calculate the force on one mesh point from the potential at the neighboring mesh points. This is basically the method which is favored by Hockney and Eastwood for P^3M. Higher accuracy is achieved by considering not only the nearest neighbors but also mesh points farther away, i.e., using linear combinations of nearest neighbor, next nearest neighbor, etc., difference operators. In Appendix C we show how these

approximations are constructed systematically. In the P^3M method the Fourier transforms of these operators are needed for the calculation of the optimal influence function (31). This approach as well needs only one Fourier transformation back to real space, like in the method of analytic differentiation. But unlike the latter, it conserves momentum (if the difference operators are chosen correctly[7]) and thus has no problems with spurious particle drifts and resulting errors in the force. However, using the neighboring points is a nonlocal approach and increasing its accuracy can only be done by taking into account more neighbors — which makes it even more nonlocal and more costly.

Obviously, there is no unique optimal way for doing the differentiation. Each approach joins together advantages and drawbacks which have to be balanced against each other under the constraint of required accuracy and available computational resources. Let us make just one example: If the required accuracy is not very high, using only nearest neighbors for the discrete differentiation on the mesh might be accurate enough. Certainly, multiplication in Fourier space by $i\mathbf{k}$ gives better results, but let us assume that this approach is actually slower due to the two additional Fourier transformations. However, if the required accuracy increases, the finite difference approximation calls for more neighbors and thus becomes more and more costly, whereas the $i\mathbf{k}$-approach right away gives the best result possible by discrete differentiation. This is because increasing the order of the differentiation scheme means that in Fourier space the transformed operators approximate $i\mathbf{k}$ to higher and higher truncation order (actually, that is how these approximations are constructed, see Appendix C). In other words, accepting the two additional Fourier transformations can be competitive. Moreover, the method of analytic differentiation could be faster than the discrete difference method even for $J=1$. Thus, in cases where the latter is less accurate than analytic differentiation, there is no reason for using it.

Whether there exists a break-even point between these methods and — if so — where it is located can depend on tuning parameters like mesh size and interpolation order as well as on details of the implementation or the computational facilities one is working with. A general statement seems to be difficult.

D. Backinterpolation

At some stage of any particle mesh method a backinterpolation of the mesh based results to the actual particles is necessary. As we have seen in Sec. III C, this can be done before or after the Fourier transformation back to real space, i.e., the mesh points can contain either the potential or the components of the electric field.

Basically, this backinterpolation is done in a similar way as the distribution of the charges to the mesh at the beginning of the calculation, via some assignment function W. For example, the force on particle i is given by

$$\mathbf{F}_i = q_i \sum_{\mathbf{r}_p \in \mathbb{M}} \mathbf{E}(\mathbf{r}_p) W(\mathbf{r}_i - \mathbf{r}_p) \tag{36}$$

with $\mathbf{E}(\mathbf{r}_p)$ being the electric field on mesh point $\mathbf{r}_p$ from Eq. (34). The interpretation of Eq. (36) is the following: Due to the discretization each particle is replaced by several ''subparticles,'' which are located at the surrounding mesh points and carry a certain fraction of the charge of the original particle. The force on each subparticle is given by its charge times the electric field at its mesh point, and the force on the original particle is the sum of the forces of its subparticles.

From a technical point of view it is convenient to use the *same* function W for the assignment onto and from the mesh, because if in the first step one does not only calculate the *total charge* accumulated at some mesh point but additionally memorizes to what extent the *individual particles* contributed to this charge, the interpolation back can be done without a single function call to W.

However, there is also a more subtle reason which suggests a symmetric interpolation, and this is related to the conservation of momentum. As demonstrated by Hockney and Eastwood,[7] the force which a particle acts onto itself is zero and Newton's third law is obeyed (up to machine precision), if

(1) charge assignment and force interpolation are done by the same function W and
(2) the approximations to the derivatives are correctly space centered.

The second requirement states that if the electric field at some mesh point $\mathbf{r}_p$ can formally be written as $\Sigma_{\mathbf{r}_q} \mathbf{d}(\mathbf{r}_p, \mathbf{r}_q) \rho_M(\mathbf{r}_q)$, then $\mathbf{d}(\mathbf{r}_p, \mathbf{r}_q) = -\mathbf{d}(\mathbf{r}_q, \mathbf{r}_p)$.

The method of analytical differentiation mentioned in Sec. III C does not use mesh based derivatives and this is the approach chosen in the implementation of SPME.[9] In their paper the authors state that the sum of the electrostatic forces on the atoms is not zero but a random quantity of the order of the rms error in the force. We believe that these fluctuations have their origin in a violation of the above conditions, although strictly speaking these are only *sufficient* conditions for momentum conservation.

For a more detailed discussion of related effects and the connection between momentum conserving and energy conserving methods see Hockney and Eastwood.[7]

IV. INVESTIGATING THE ACCURACY

An investigation of the errors connected with particle mesh Ewald methods is important for several reasons. First, the complete procedure of discretization introduces new sources of errors in addition to the ones originating from real and reciprocal space cutoffs. Second, comparing the efficiency of different mesh methods is only fair if it is done at the same level of accuracy. And third, the tuning parameters should be chosen in such a way as to run the algorithm at its optimal operation point.

However, there is no unique or optimal *measure* of accuracy. If molecular dynamics simulations are performed, the main interest lies in errors connected with the *force*, while in Monte Carlo simulations one is concerned with accurate *energies*. In the simulation of ensemble averages it is the *global* accuracy — measured, e.g., by root mean square

quantities — which is important, but in the simulation of rare events *local* accuracy and maximal errors are also relevant. Errors in the force can be due to their *magnitude* or due to their *direction*. And finally, one might be interested in *absolute* or *relative* errors.

Whatever quantity one decides to look at, it can be investigated as a function of *system* parameters like particle separation or distribution, *tuning* parameters like α, mesh size or interpolation order and *components* of the algorithm, e.g., interpolation or differentiation scheme or splitting function $f(r)$. Obviously this gives rise to a very large number of combinations. In other words, the corresponding parameter space is large and nontrivial, i.e., general statements concerning the performance of one method cannot usually be extracted from low-dimensional cuts through this space, because different methods scale differently with respect to their parameters.

Nevertheless, we want to present some numerical accuracy measurements at important points of this parameter space for the following reasons: As we pointed out, there are several options for the implementation of each step of a mesh calculation, e.g., three ways for doing the derivative in Eq. (12). This freedom of choice and its impact on the overall accuracy has not been systematically investigated so far, although a qualitative understanding of at least *typical* influences of the different parts on the performance permits a judicious assessment and comparison of the resulting algorithms, in particular P^3M, PME and SPME. We want to show which combinations are attractive and which should definitely be avoided. And finally we want to present easily reproducible measurements which should allow the reader a comparison with his own implementations of particle mesh Ewald routines. However, we will *not* present large accurate tables, which provide an easy way for tuning these algorithms under all circumstances. On the contrary, we want to encourage any potential user to perform some of these simple measurements on his own and thereby not only gain insight but also the possibility to optimize his tuning parameters. We want to stress that parameters which are only roughly estimated or even historically handed down should be used with great care.

A. One possible measure of accuracy

In this paper we will solely be concerned with one measure of accuracy, namely the root mean square (rms) error in the force, given by

$$\Delta F := \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\mathbf{F}_i - \mathbf{F}_i^{\mathrm{exa}})^2}, \tag{37}$$

where $\mathbf{F}_i$ is the force on particle number i calculated via some mesh method and $\mathbf{F}_i^{\mathrm{exa}}$ is the *exact* force on that particle, calculable, e.g., by a well converged standard Ewald sum. There exist error estimates for the real space and Fourier space contribution to this error for the standard Ewald sum[15] and for the PME method[17] which greatly simplify the determination of the optimal value of α. For the case of the (in particular $i\mathbf{k}$-differentiated) P^3M method we will derive such an estimate in paper II.[11]

B. Error as a function of α

We investigated the rms error (37) for a system of 100 particles (50 carry a positive and 50 a negative unit charge), which were randomly placed within a simulation box of length $L=10$, as a function of the Ewald parameter α. In order to make our results fully reproducible, we describe in Appendix D, how our actual random configuration was generated.

For small α the result of the Ewald sum (or any of the described particle mesh methods) is dominated by the real space contribution (4) while for large α it is the Fourier part (5) which is important. This is a simple consequence of the fact that in the real space sum α occurs in the *numerator* of the exponential function (or — to be precise — of the complementary error function) while in the Fourier space sum it occurs in the *denominator* and thus influences the decay of both contributions in a converse way. Hence, *at given cutoffs*, the same applies to the errors. Since with increasing α the real space contribution becomes more accurate while the Fourier space contribution degrades in accuracy, one can expect an optimal α to exist at which the *total* error is minimal. This is approximately at the point where real space and Fourier space errors are equal. Since the different mesh methods we investigate all coincide in the treatment of the real space part, their errors should all be the same for sufficiently small α.

In Fig. 2 we plot the rms error of the force as a function of α, which was obtained by investigating our system with various mesh methods. They all share a mesh size of $N_M = 32$ (and thus have 32^3 mesh points in total), an interpolation order $P=7$ and a real space cutoff $r_{\max}=4$. We find indeed the general features described above, like a low accuracy for very small or very large values of α and an optimal value in-between. However, the various methods differ considerably in their accuracy. (Note that in this and the following figures the vertical scale is logarithmic!)

The solid line corresponds to PME. This method comprises some elements which make one think about possible improvements: a not very smooth charge assignment scheme (namely, the Lagrange interpolation) and the use of the plain continuum Green function. There is clearly no *obvious* advantageous replacement for the latter, but it is easy to replace the Lagrange scheme by the smooth spline interpolation (by just changing the assignment function). Yet, the result of this supposed improvement, shown in line 2, is in fact disappointing. As we mentioned several times, the continuum Green function is best accompanied by a Lagrange interpolation scheme, because this leads to a cancellation of certain discretization errors. Changing the assignment scheme destroys this effect and the resulting error completely shatters the desired improvement in accuracy.

Upgrading PME requires a proper treatment of both elements — charge assignment *and* Green function. This is in fact what the remaining two algorithms (SPME and P^3M) accomplish. Since they both use a smooth spline interpolation, they are both potential candidates for analytic differentiation. In fact, the SPME method, as described in the original publication,[9] chooses this implementation of the

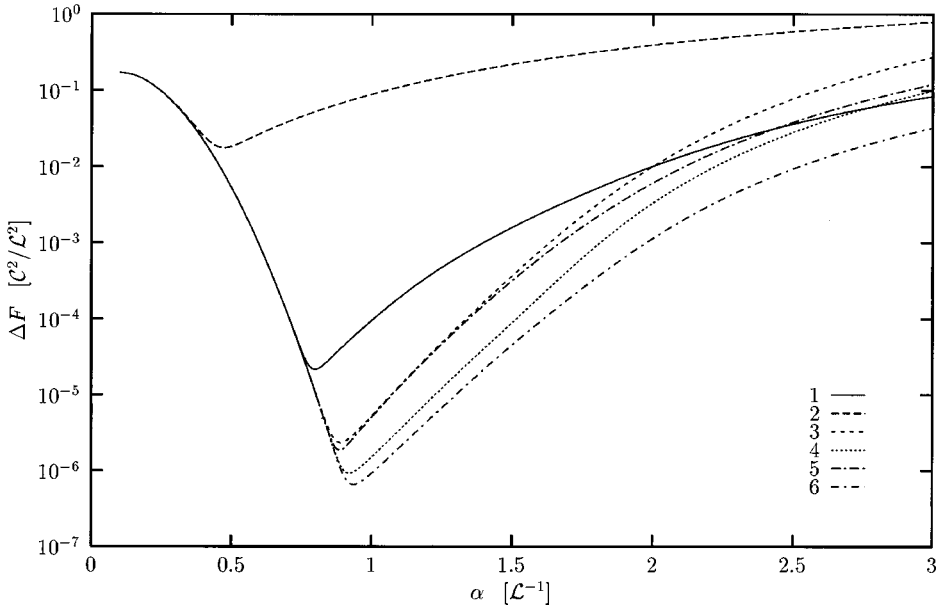

FIG. 2. Comparison of different mesh methods: The rms error ΔF from Eq. (37) for a system of 100 charged particles randomly distributed within a cubic box of length $L=10$ (see Appendix D) is shown as a function of the Ewald parameter α for 6 mesh algorithms, which all share $N_M=32$, $P=7$ and $r_{\max}=4$. Line 1 is PME. Line 2 corresponds to an algorithm which is obtained from PME by retaining the continuum Green function but changing to the spline charge assignment. Lines 3 and 4 are analytically and $i\mathbf{k}$-differentiated SPME, respectively, and lines 5 and 6 are analytically and $i\mathbf{k}$-differentiated P^3M, respectively. Note the logarithmic vertical scale in this and the following figures.

derivative, because it is very fast (line 3). Nevertheless, the $i\mathbf{k}$ method is still possible and leads to an even better result (line 4), which admittedly has to be paid with two additional FFT calls. Analytically differentiated P^3M gives an error almost identical to analytically differentiated SPME, but if one implements the $i\mathbf{k}$ derivative, P^3M improves a little bit on SPME. From a theoretical point of view the latter is not too surprising: After all, if P^3M uses an optimal differentiation (in view of accuracy) and an optimal influence function, it can be expected to constitute a kind of lower bound for the error. However, if the optimal differentiation is replaced by the analytic differentiation, a new source of error appears (namely, the random force fluctuations described in Sec. III D). If this contribution dominates, the fact that P^3M uses a better influence function than SPME cannot make a large difference. In our case the analytically differentiated SPME is a factor 9.2 more accurate than PME, while the $i\mathbf{k}$-differentiated P^3M method is more accurate than PME by a factor of about 33. However, one must realize that SPME and P^3M have different execution times, since P^3M needs two additional FFT calls compared to SPME. But apart from the analytically differentiated curves all methods summarized in Fig. 2 need *exactly* the same time for a mesh calculation. This comes from the fact that the methods differ only in parts which normally are tabulated anyway, like the influence function.

There is another surprising thing to note about SPME: For the chosen values of N_M and P the curves for PME and (analytically differentiated) SPME *intersect*, i.e., the latter is not necessarily more accurate. It could be argued that at least for the *optimal* value of α SPME is better, but this optimal α of course depends on the real space cutoff $r_{\max}$ as well. If this cutoff is decreased, the real space contribution to the error is increased. In fact, using the estimate of Kolafa and Perram[15] one finds that at the intersection point of PME and SPME this contribution will have the same size as the Fourier space contribution for (in this case) $r_{\max}\approx 1$. Thus, for even smaller values of $r_{\max}$, PME would actually be more accurate than SPME.

Now that we have compared various particle mesh *methods*, we want to examine in a little more detail some *parts* of the algorithm. We will always use the P^3M method for illustration. Corresponding plots for PME or SPME would look qualitatively very similar and hence are not presented in this paper.

In Fig. 3 we took all parameters of the $i\mathbf{k}$-differentiated P^3M method from Fig. 2 but varied the charge assignment order from $P=1$ to $P=7$. Increasing P improves the accuracy by more than four orders of magnitude (from $P=1$ to $P=7$). However, the reward for going from P to $P+1$ is larger for small P. Note also that the optimal value of α depends on P.

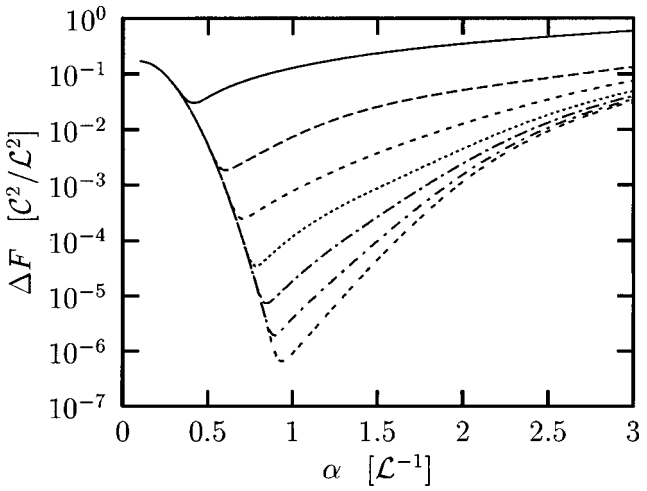

FIG. 3. Influence of the charge assignment order: The rms error ΔF for our model system from Appendix D is calculated for the $i\mathbf{k}$-differentiated P^3M method with $N_M=32$ and $r_{\max}=4$. From top to bottom the order P of the (spline) charge assignment scheme is increased from 1 to 7.

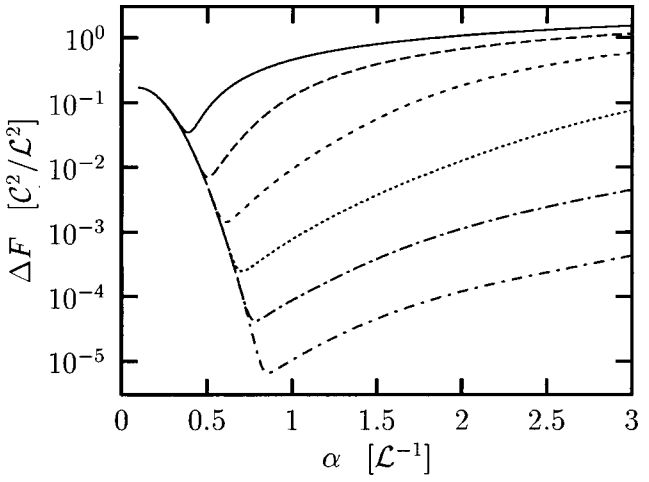

FIG. 4. Influence of the mesh size: The rms error ΔF for our model system from Appendix D is calculated for the $i\mathbf{k}$-differentiated P^{3}M method with $P=3$ and $r_{\max}=4$. From top to bottom the mesh size N_M is given by 4,8,16,32,64 and 128. [Note that the total number of mesh points in this three-dimensional system is given by $(N_M)^3$.]

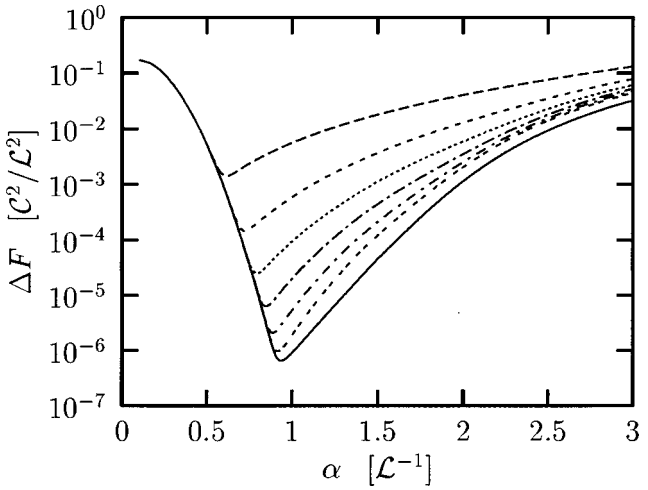

FIG. 5. Influence of the differentiation scheme: The rms error ΔF for our model system from Appendix D is calculated for the P^{3}M method with $N_M=32$, $P=7$ and $r_{\max}=4$. Shown are six mesh based approximations to the differentiation operator $i\mathbf{k}$ (from top to bottom: $\Delta^{(1)}, \ldots, \Delta^{(6)}$, see Appendix C) as well as the result for $i\mathbf{k}$ itself (lowest curve, solid line).

In Fig. 4 we fix the order of the charge assignment scheme to $P=3$ and vary the number N_M of Fourier mesh points. Qualitatively the behavior is similar to Fig. 3: Improving the method reduces the error and shifts the optimal α to the right. Note that from a computational point of view Figs. 3 and 4 are sort of conjugate: The accuracy depends on both N_M and P, but increasing one parameter does not influence the performance of the other. In other words, the charge assignment scales as P^3 independent of N_M and the FFT scales as $(N_M \log N_M)^3$ independent of P. Optimal performance requires a suitable combination of N_M and P.

Next we investigated the differentiation scheme. To this end we employed the P^{3}M method with $N_M=32$ and $P=7$ and used various orders J of the mesh based approximation to the difference operator (see Appendix C). (Actually, the calculations were done by a multiplication in Fourier space with the transformed approximations $\tilde{\mathbf{D}}^{(J)}$.) The result is shown in Fig. 5, which looks pretty much like Fig. 3 but was generated quite differently. With increasing order of the difference approximation the errors decrease. However, the result of the $i\mathbf{k}$-differentiation scheme forms a lower bound to the error of this method. [After all, $i\mathbf{k}$ is the Fourier representation of the exact differential operator, and in the standard Ewald sum the differentiation is also done this way—compare Eq. (15).] Here the bound is reached in the minimum at $J=7$, so further improving the differentiation order is of no use at all. Of course, if the accuracy of the lower bound is smaller (e.g., because the charge assignment order is lower) the $i\mathbf{k}$ bound will be reached already by smaller values of J. Note that in this example the method of analytic differentiation gives approximately the same accuracy as a fifth-order difference scheme (compare to Fig. 2). Since analytic differentiation is much faster, it should be preferred to the finite difference approach in cases where the latter is less accurate anyway.

The last part of this section deals with the determination of the optimal α value. There exist rather good estimates for the real and reciprocal space error of the standard Ewald sum[15] and the reciprocal space error of the PME method.[17] The optimal α value of these two methods and the corresponding accuracy can be obtained very precisely by just calculating the intersection point of the real and corresponding reciprocal space estimates. Their high quality is clearly demonstrated in Fig. 6. The existence of these formulas is certainly a big advantage of the PME method, since it permits an *a priori* determination of the optimal operation point as a function of system specifications (like box length, par-

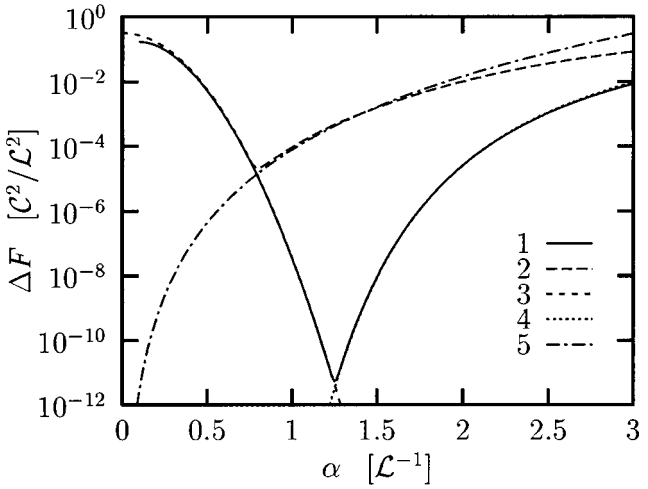

FIG. 6. Comparison of a mesh method with the standard Ewald sum: The rms error ΔF for the Ewald (line 1) and PME (line 2) method are calculated for our model system from Appendix D. The parameters for PME are the same as in Fig. 2 and the Fourier space cutoff for the Ewald sum was set to $k_{\max}=20\times 2\pi/L$. This value is interesting to compare with the PME method, because it corresponds to the same number of $\mathbf{k}$ vectors (since $\frac{4}{3}\pi\, 20^3 \approx 32^3$). Also shown is the estimate for the real space error[15] (line 3), the Fourier space error for Ewald (line 4, we used the slightly better estimate from Petersen— Ref. 17) and the Fourier space error for PME (Ref. 17)—line 5). Note that the estimates for the Ewald sum can hardly be distinguished from line 1.

ticle number or valence) or method parameters (like mesh size or assignment order). The derivation of a similar error estimate for the P^{3}M method is the focus of paper II.[11] This is basically the last step which is missing to advocate P^{3}M as the most accurate and versatile Ewald mesh method.

A final word concerning the accuracy of mesh methods compared to the Ewald sum follows: The optimal α value for a standard Ewald summation of our system with $k_{\max}=20\times 2\pi/L$ (which thus has the same number of $\mathbf{k}$ vectors, because $\frac{4}{3}\pi 20^3 \approx 32^3$) is approximately 1.25 and the corresponding total error is of the order 5×10^{-12} (see Fig. 6). Although much optimization effort has been put into mesh methods in order to reduce errors, we must face the fact that one generally loses many orders of magnitude in accuracy due to discretization. So if high accuracy is essential but speed is not an issue, the conventional Ewald method is unsurpassed: it is much easier to program and the desired accuracy can be increased up to machine precision without any additional programming effort. However, it would be misleading to infer that particle mesh methods sacrifice accuracy in favor of speed, because due to the more advantageous scaling with particle number (essentially $N\log N$ compared to $N^{3/2}$) there will always be a critical number N^*, such that the mesh method will be faster than the Ewald sum for particle numbers $N>N^*$. See, e.g., Petersen[17] for a discussion of the break-even value N^* for PME.

C. Error as a function of minimum image distance

Instead of calculating the rms error for a complete configuration, it is also worthwhile to investigate it as a function of the minimum image distance r between just *two* particles. This is a possibility to monitor the distance dependence of the accuracy for the various methods. Thus, we randomly created a pair of particles inside the simulation box (again, $L=10$) with *given* minimum image separation r and calculated the rms error from Eq. (37). This was repeated for 5×10^4 separations equally spaced between 0 and $\frac{1}{2}\sqrt{3}L$, which is the largest possible minimum image separation. As this is done at constant α for each method, the real space contribution to the force always cancels when performing the difference in Eq. (37), so this plot is only sensitive to the *Fourier* contribution and it is not necessary to specify a real space cutoff. A grid with $N_M=32$ was chosen and the charge assignment order was set to $P=7$. However, as can be seen from Fig. 2, different methods have their optimal operation point at different values of α. Therefore we found it more sensible to compare the different methods at their *individual* optimal value of α, which can be obtained from Fig. 2. Although the curves in Fig. 2 correspond to a system which contains 100 particles (and not just 2), we believe that this has no influence on the optimal value of α, since, e.g., the error formulas for the Ewald sum derived by Kolafa and Perram[15] show that the real space and the Fourier space contribution to ΔF display the same dependence on particle number. (See also our general discussion of the scaling of ΔF with particle number and valence in paper II.[11])

Note that the Coulomb problem in the given periodic geometry lacks spherical symmetry and due to the existence of a grid also the translational symmetry is broken. So — strictly speaking — ΔF is not just a function of r but also depends on the orientation of the particles and their location within the box. This manifests itself in the fact that the measured points $\Delta F(r)$ do not collapse onto a single smooth curve but show some scatter. Since we are not interested in this effect, we averaged the scatter by binning 50 points together at one time and additionally performing a Gaussian smoothing (with width 0.1). This just makes the data easier to plot and digest.

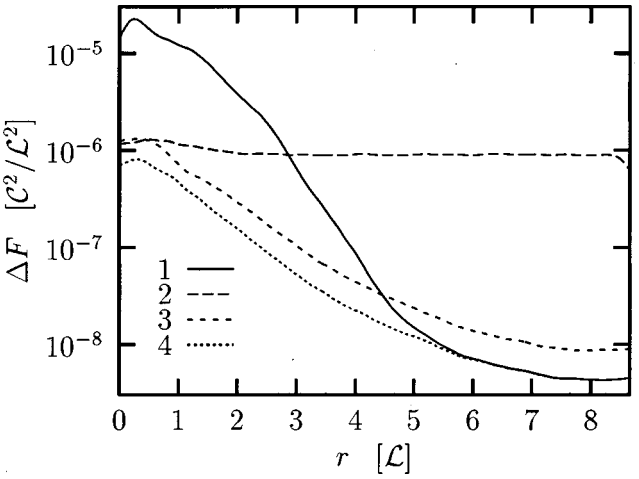

FIG. 7. Distance dependency: The rms error ΔF as a function of the minimum image separation r between two particles is shown for several mesh methods, which all share $N_M=32$ and $P=7$. Line 1 corresponds to PME, lines 2 and 3 are analytically and $i\mathbf{k}$-differentiated SPME, respectively, and line 4 is $i\mathbf{k}$-differentiated P^{3}M. For each method α was individually set to its optimal value from Fig. 2: $\alpha_1=0.8$, $\alpha_2=0.89$, $\alpha_3=0.92$ and $\alpha_4=0.94$.

The result of this measurement is shown in Fig. 7. Several interesting things can be observed: All algorithms produce their largest errors at small distances and get considerably more accurate at larger values of r — with one exception — the analytically differentiated SPME method almost immediately settles to a (comparatively large) constant error. Since the only difference between lines 2 and 3 is the differentiation scheme, it must be the random force fluctuations discussed in Sec. III D which are responsible for this effect. Note that PME at some distance gives better results than $i\mathbf{k}$-differentiated SPME. Also it is most surprising that at large distances PME and P^{3}M give identical errors, although they differ considerably in the charge assignment scheme as well as in the employed Coulomb Green function. Finally, the $i\mathbf{k}$-differentiated P^{3}M method is most accurate for all distances. In this case this is not very surprising, because the quantity Q from Eq. (29), with respect to which P^{3}M is optimized, is essentially the integral over any of these curves in Fig. 7, weighted with the probability density of the minimum image distance r.

V. CONCLUSIONS

Based on our theoretical considerations and the results of our numerical experiments we draw the following conclusions.

(1) The error of all Ewald calculations — be it the standard Ewald sum or any particle mesh method — depends very sensitively on the Ewald parameter α. Hence, finding the optimal value of α is not merely an option but *absolutely essential*. Abstaining from a proper α-tuning results (at best) in wasting accuracy and (at worst) in the calculation of wrong forces or energies. For the standard Ewald sum and for PME there exist estimates for the real and reciprocal space contribution to the error, which allow an *a priori* determination of the optimal value for α, depending on the relevant system parameters. For the P^3M method we will tackle this problem in paper II.[11]

(2) At given computational effort the errors produced by different methods do not just vary marginally but by orders of magnitude, so putting some effort into this topic is certainly worth the trouble.

(3) Generally, the total error is a combination of several contributions. If one of them dominates, there is no point in improving the other parts of the method. Assume for instance that we are using a finite difference scheme for the derivative and that the overall accuracy is actually limited by the discretization errors resulting from a low order charge assignment scheme. In this case, increasing the order J of the differentiation scheme would be useless. For example, in Fig. 5 going beyond $J=7$ would not yield any improvement. If in Fig. 5 the assignment order P was 3 and not 7, it would even suffice to use $J=2$ (compare with Fig. 3).

(4) Different methods scale differently with respect to their parameters. For example, since the α dependency of the error for PME is not the same as for SPME (there is not just a constant factor between them), the error curves can intersect (see Fig. 2). The (almost trivial) consequence is that if one method is more accurate than another method in a specific region of the space of tuning parameters, this need not be the case in another region or even for all choices of parameters.

(5) There exist many possibilities for combining the various parts of a mesh calculation — like charge assignment or differentiation scheme. This freedom of choice can be exploited to suit ones particle mesh algorithm to already existing constraints in the complete simulation program. However, these combinations should always be tested thoroughly, since naive ''improvements'' can turn out to be disastrous (see line 2 in Fig. 2). There are, so to speak, several incompatible roads toward optimization, and one step away from a local optimum is in general a disimprovement.

(6) If method A is at the same computational effort 10 times more accurate than method B, this can be advantageous even if one is happy with the accuracy of B: Almost surely method A — tuned down to the accuracy of B — will be faster than B, because, e.g., the number of mesh points could be reduced.

(7) If one wants to use the continuum Green function, a Lagrange interpolation scheme should be used. However, our tests show that a combination of the smooth spline interpolation with an appropriately adjusted Green function — like in the P^3M and SPME approach — should be preferred, since this can be made more accurate. We recommend the P^3M approach, because it uses the analytically derived optimal influence function from Eq. (31), which minimizes the force errors, and — due to its smooth charge assignment — permits all investigated differentiation schemes. In particular, the $i\mathbf{k}$-method is the most accurate implementation of the derivative [and comes most closely to the Ewald method, see Eq. (15)], whereas the analytic differentiation (introduced by Essmann *et al.*[9] originally for the SPME method) — although somewhat less accurate — is a fast and attractive alternative.

(8) Compared to a standard Ewald sum, which uses the same number of $\mathbf{k}$-vectors, all mesh algorithms are much less accurate (see Fig. 6). However, the accuracy which is actually *needed* in a simulation is typically not too large, since most simulations employ at the same time some kind of thermostat, and it is a waste of time to calculate the electrostatic forces much more accurately than the random fluctuations of the thermostat. (However, it is to be expected that the fluctuations originating from particle mesh errors are not δ-correlated in time.)

Note added in proof. After the submission of our paper T. Darden kindly brought to our attention a publication[19] where he performed a numerical comparison of the P^3M to the SPME method, leading to results which are in agreement with our findings.

ACKNOWLEDGMENTS

The authors are grateful to W. F. van Gunsteren for initiating this research, U. Micka and Q. Spreiter for fruitful discussions, and K. Kremer for encouragement and helpful comments. C. H. further thanks the DFG for financial support.

APPENDIX A: SOME COMMENTS ON UNITS

Different people and communities prefer different conventions for units, especially if it comes to electrostatics. In this brief appendix we present our choice.

We write the Coulomb potential generated by a point particle with charge q located at the position $\mathbf{r}_0$ as

$$\phi(\mathbf{r}) = \frac{q}{|\mathbf{r}-\mathbf{r}_0|}. \tag{A1}$$

Thus, its dimension is charge divided by length. In other words, if we measure all lengths in multiples of some unit length $\mathcal{L}$ and all charges in multiples of some unit charge $\mathcal{C}$, the dimension of the electrostatic potential is $\mathcal{C}/\mathcal{L}$. As a consequence, the dimension of electrostatic energy is $\mathcal{C}^2/\mathcal{L}$ and of electrostatic force is $\mathcal{C}^2/\mathcal{L}^2$.

In this paper there is no need to specify $\mathcal{C}$ or $\mathcal{L}$, and if it comes to the final result (be it formulas or numbers), it can always be embellished with prefactors like $1/4\pi\varepsilon_0$.

We give just one example: If one chooses $\mathcal{L}=\text{Å}=10^{-10}\,\text{m}$, $\mathcal{C}=e_0\approx 1.6022\times 10^{-19}\,\text{C}$ and includes the

standard-SI-prefactor $1/4\pi\varepsilon_0$, the numerical value of the original expression $q_1 q_2 / r^2$ gives the force in units of 2.3071×10^{-8} N.

Another common unit of force — especially among chemists — is kcal mol^{-1} Å^{-1}. Obviously we have

$$\frac{\text{kcal}}{\text{mol Å}} \approx \frac{4.186 \times 10^3 \text{ J}}{6.02217 \times 10^{23}\, 10^{-10} \text{ m}} \approx 6.9510 \times 10^{-10} \text{ N}.$$

Thus, if one prefers to measure forces in units of kcal mol^{-1} Å^{-1}, one only has to multiply the numerical value of the original $q_1 q_2 / r^2$ by a factor of approximately 331.9.

APPENDIX B: CONTINUUM GREEN FUNCTION AND LAGRANGE INTERPOLATION SCHEME

In this appendix we show how the implemented Green function and the charge assignment scheme are related to each other. More specifically, we demonstrate that the use of the *continuum* version of the Coulomb Green function, as it appears in the conventional Ewald sum, suggests a so-called *Lagrange interpolation scheme*, because this leads to a nice cancellation of certain discretization errors. We closely follow the notation of Hockney and Eastwood.[7]

We consider only the one-dimensional case. The electrostatic potential at position x' due to a unit charge residing at position x is *not* just a function of $|x'-x|$ but also depends on the distances of this charge from its neighboring mesh points. This artifact of the mesh can be quantified as follows: Let $g(x)$ be the continuum Coulomb Green function and $W_p(x) = W(x - x_p)$ the charge assigned to mesh point p at position x_p due to a unit charge at position x. The electrostatic potential at position x' can then be written as

$$\phi(x') = \sum_{p=1}^{P} W_p(x)\, g(x' - x_p) \tag{B1}$$

where the sum is taken over all P mesh points to which the particle at position x contributed some fraction of its charge, i.e., the P mesh points which are closest to x. Taylor expanding $g(x'-x_p)$ about $(x'-x)$ gives:

$$\phi(x') = \sum_{p=1}^{P} W_p(x) \sum_{n=0}^{\infty} \frac{(x-x_p)^n}{n!} g^{(n)}(x-x'). \tag{B2}$$

It is possible to cancel the artificial terms in the n sum (i.e., the ones which depend on $x-x_p$) up to order P by choosing the charge fractions $W_p(x)$ such that

$$\sum_{p=1}^{P} W_p(x)\, (x-x_p)^{n-1} = \delta_{1,n}, \quad n = 1, \ldots, P. \tag{B3}$$

By induction with respect to n one can show that this may equivalently be expressed as

$$\sum_{p=1}^{P} W_p(x)\, x_p^{n-1} = x^{n-1}, \quad n = 1, \ldots, P. \tag{B4}$$

This system of P linear equations has a unique solution for the $W_p(x)$ since the coefficient matrix x_p^{n-1} is a Vandermonde matrix for the distinct points $x_1, \ldots, x_P$ and hence has full rank. The $W_p(x)$ are thus polynomials of degree $P-1$. Since in particular Eq. (B4) must be true at the mesh points, it follows

$$\sum_{p=1}^{P} W_p(x_q)\, x_p^{n-1} = x_q^{n-1}, \quad n, q = 1, \ldots, P \tag{B5}$$

which — again due to the invertibility of x_p^{n-1} — can only be true if

$$W_p(x_q) = \delta_{pq}, \quad p, q = 1, \ldots, P. \tag{B6}$$

Equation (B6) suffices to determine the polynomials $W_p(x)$. They are referred to as the *fundamental polynomials for the Lagrange interpolation problem*.[20] Petersen[17] tabulates them for $P = 3, \ldots, 7$ and their implementation is explained in detail. If one needs these assignment functions for higher values of P, one has to solve the system of linear equations (B4) or the interpolation problem (B6).

APPENDIX C: SYSTEMATIC DIFFERENCE APPROXIMATIONS TO THE DIFFERENTIAL OPERATOR

In this appendix we show how mesh approximations for the differential operator d/dx can systematically be written as convex combinations of difference operators. In this way one can implement optimal combinations of these operators into the program right-away, so an empirical tuning of the coefficients[7,10] is no longer necessary. We describe the idea only for one dimension, the generalization to higher dimensions can be done easily via the Cartesian components.

First we define the jth-neighbor centered difference operator Δ_j by

$$(\Delta_j f)(x) := \frac{f(x+jh) - f(x-jh)}{2jh}, \tag{C1}$$

where h is the mesh spacing and x some mesh point. Applying this operator on a function f can be written as the convolution $D_j \star f$, where $D_j(x)$ is defined as

$$D_j(x) := \frac{\delta(x+jh) - \delta(x-jh)}{2jh}. \tag{C2}$$

From the convolution theorem it follows that in Fourier space the derivative is given by $\tilde{D}_j(k)\tilde{f}(k)$, where it is easily verified that the Fourier transform of D_j is

$$\tilde{D}_j(k) = i \frac{\sin(jkh)}{jh}. \tag{C3}$$

(Note that in the limit $h \downarrow 0$ this reduces to the Fourier representation of d/dx, namely ik.)

Since one can expect to achieve better approximations for the differential operator by using linear combinations of the difference operators Δ_j, we define a Jth-order difference operator by

$$\Delta^{(J)} := \sum_{j=1}^{J} c_j \Delta_j. \tag{C4}$$

TABLE I. Optimal form for the weighting coefficients of the Jth-order difference operator $\Delta^{(J)}$ from Eq. (C4) for several values of J.

Order J	c_1	c_2	c_3	c_4	c_5	c_6
1	1					
2	4/3	$-1/3$				
3	3/2	$-3/5$	1/10			
4	8/5	$-4/5$	8/35	$-1/35$		
5	5/3	$-20/21$	5/14	$-5/63$	1/126	
6	12/7	$-15/14$	10/21	$-1/7$	2/77	$-1/462$

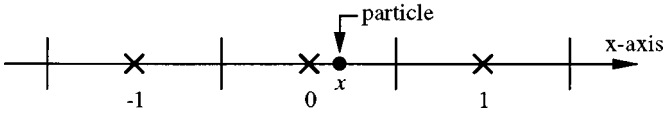

FIG. 8. Schematic picture for a three-point charge assignment. The crosses are the mesh points and the lines indicate the (Wigner–Seitz) cell boundaries of each point (the mesh spacing is $h=1$). All particles with $x \in [-\frac{1}{2},+\frac{1}{2}]$ distribute their charge between the mesh points at -1, 0 and $+1$ and the corresponding charge fractions are $W_p(x)$, $p \in \{-1,0,+1\}$. (After Hockney and Eastwood—Ref. 7).

Using the Fourier representation of the differential and the jth-neighbor centered difference operator from Eq. (C3), we demand

$$\sum_{j=1}^{J} c_j \, i \frac{\sin(jkh)}{jh} = ik + \mathcal{O}((kh)^{2J+1}) \tag{C5}$$

or

$$\sum_{j=1}^{J} c_j \cos(jkh) = 1 + \mathcal{O}((kh)^{2J}), \tag{C6}$$

where Eq. (C6) follows from differentiating Eq. (C5). Taylor expanding the cosine in Eq. (C6) and equating coefficients gives J linear equations for the J unknowns c_j. The first few are given in Table I.

Note that in the case of the second-order approximation the weighting $(\frac{4}{3},-\frac{1}{3})$, which empirically was found to be optimal,[7] is reproduced.

APPENDIX D: THE MODEL SYSTEM

The rms error in the force for a system of 100 particles randomly distributed in the simulation box is somewhat sensitive to details of the generated configuration, e.g., the actual minimum distance. In order to make our measurements fully reproducible we decided to present our configuration as well.

We found it easier not to list the particle positions but to describe the procedure which was used to generate them. The coordinates of the 100 particles were constructed by first drawing 300 random numbers $\mathcal{R}_n$ between 0 and 1. If L is the box length then particle 1 gets the coordinates $(L\mathcal{R}_1, L\mathcal{R}_2, L\mathcal{R}_3)$, particle 2 gets $(L\mathcal{R}_4, L\mathcal{R}_5, L\mathcal{R}_6)$ and so on. Moreover, particles with an even/odd number will get a positive/negative unit charge.

The choice of the random number generator is the following: If a_n is a positive integer, define its successor a_{n+1} via:

$$a_{n+1} := (1\,103\,515\,245 a_n + 12\,345) \bmod 2^{32}. \tag{D1}$$

Now define the pseudo-random number $\mathcal{R}_n \in [0;1[$ by

$$\mathcal{R}_n := \frac{(a_n \div 65\,536) \bmod 32\,768}{32\,769}, \tag{D2}$$

where ''$\div$'' should denote an integer division which discards any division rest. Choosing $a_0=1$ we obtained the sequence of random numbers 16 838, 5758, 10 113, . . . , of which the first 300 were used for positioning the particles, e.g., with $L=10$, the first particle has coordinates 5.138 . . . ,1.757 . . . ,3.086 . . .) and a negative unit charge. The smallest minimum image distance is approximately 0.370 264 and occurs between particle 46 and particle 98.

Incidentally, we did not choose this random number generator because it is particularly good (it is not), but it is very easy to implement. Many `C` libraries provide a function `rand`, which relies on Eqs. (D1) and (D2).

APPENDIX E: CHARGE ASSIGNMENT WITH SPLINES

In this appendix we describe in a little more detail the procedure of charge assignment and present the charge fractions which are needed for a Pth order assignment scheme à la Hockney and Eastwood[7] [see Eqs. (18) and (19)].

Let the units be chosen such that the grid spacing is 1. For any P consecutive mesh points there exists an interval $\mathcal{I}$ of length 1 such that the charge of a particle with coordinate $x \in \mathcal{I}$ is distributed between these mesh points. By simple shifting we can assume this interval to be $[-\frac{1}{2},+\frac{1}{2}]$. Then the P mesh points will lie at $-(P-1)/2, -(P-1)/2+1, \ldots, (P-1)/2$. For $P=3$ this is schematically shown in Fig. 8. The charge fraction $W_p(x)$, which will be assigned to the mesh point x_p, is related to the charge assignment function $W(x)$ via $W_p(x)=W(x-x_p)$.

The basic steps which have to be done for a particle with coordinate x (generally not in $[-\frac{1}{2},+\frac{1}{2}]$) during a Pth order charge assignment are thus:

(1) Define $\overline{x}$ to be the coordinate of the particle's nearest mesh point (if P is odd) or the midpoint between the two nearest mesh points (if P is even).
(2) Find the P mesh points x_p which are closest to x. They will be indexed by their relative position to $\overline{x}$, so $p \in \{-(P-1)/2, -(P-1)/2+1, \ldots, (P-1)/2\}$.
(3) The fraction of charge which is assigned to each of these mesh points is given by $W_p(x-\overline{x})$.

In this way the charge fractions are written as a function of the separation $x-\overline{x} \in [-\frac{1}{2},+\frac{1}{2}]$. Hockney and Eastwood refer to the cases $P=1$, 2 and 3 as NGP (nearest grid point), CIC (cloud in cell) and TSC (triangular shaped cloud), respectively. Generally, for $P \in \{1, \ldots, 7\}$ the charge fractions $W_p^{(P)}(x)$ are given by the following polynomials:

$$P=1\text{: } W_0^{(1)}(x)=1;$$

$$P=2\text{: } W_{-1/2}^{(2)}(x)=\tfrac{1}{2}(1-2x),$$

$$W^{(2)}_{+1/2}(x) = \tfrac{1}{2}(1+2x);$$

$$P=3\text{: } W^{(3)}_{-1}(x) = \tfrac{1}{8}(1-4x+4x^2),$$

$$W^{(3)}_{0}(x) = \tfrac{1}{4}(3-4x^2),$$

$$W^{(3)}_{+1}(x) = \tfrac{1}{8}(1+4x+4x^2);$$

$$P=4\text{: } W^{(4)}_{-3/2}(x) = \tfrac{1}{48}(1-6x+12x^2-8x^3),$$

$$W^{(4)}_{-1/2}(x) = \tfrac{1}{48}(23-30x-12x^2+24x^3),$$

$$W^{(4)}_{+1/2}(x) = \tfrac{1}{48}(23+30x-12x^2-24x^3),$$

$$W^{(4)}_{+3/2}(x) = \tfrac{1}{48}(1+6x+12x^2+8x^3);$$

$$P=5\text{: } W^{(5)}_{-2}(x) = \tfrac{1}{384}(1-8x+24x^2-32x^3+16x^4),$$

$$W^{(5)}_{-1}(x) = \tfrac{1}{96}(19-44x+24x^2+16x^3-16x^4),$$

$$W^{(5)}_{0}(x) = \tfrac{1}{192}(115-120x^2+48x^4),$$

$$W^{(5)}_{+1}(x) = \tfrac{1}{96}(19+44x+24x^2-16x^3-16x^4),$$

$$W^{(5)}_{+2}(x) = \tfrac{1}{384}(1+8x+24x^2+32x^3+16x^4);$$

$$P=6\text{: } W^{(6)}_{-5/2}(x) = \tfrac{1}{3840}(1-10x+40x^2-80x^3+80x^4-32x^5),$$

$$W^{(6)}_{-3/2}(x) = \tfrac{1}{3840}(237-750x+840x^2-240x^3-240x^4+160x^5),$$

$$W^{(6)}_{-1/2}(x) = \tfrac{1}{1920}(841-770x-440x^2+560x^3+80x^4-160x^5),$$

$$W^{(6)}_{+1/2}(x) = \tfrac{1}{1920}(841+770x-440x^2-560x^3+80x^4+160x^5),$$

$$W^{(6)}_{+3/2}(x) = \tfrac{1}{3840}(237+750x+840x^2+240x^3-240x^4-160x^5),$$

$$W^{(6)}_{+5/2}(x) = \tfrac{1}{3840}(1+10x+40x^2+80x^3+80x^4+32x^5);$$

$$P=7\text{: } W^{(7)}_{-3}(x) = \tfrac{1}{46\,080}(1-12x+60x^2-160x^3+240x^4-192x^5+64x^6),$$

$$W^{(7)}_{-2}(x) = \tfrac{1}{23\,040}(361-1416x+2220x^2-1600x^3+240x^4+384x^5-192x^6),$$

$$W^{(7)}_{-1}(x) = \tfrac{1}{46\,080}(10543-17340x+4740x^2+6880x^3-4080x^4-960x^5+960x^6),$$

$$W^{(7)}_{0}(x) = \tfrac{1}{11\,520}(5887-4620x^2+1680x^4-320x^6),$$

$$W^{(7)}_{+1}(x) = \tfrac{1}{46\,080}(10543+17340x+4740x^2-6880x^3-4080x^4+960x^5+960x^6),$$

$$W^{(7)}_{+2}(x) = \tfrac{1}{23\,040}(361+1416x+2220x^2+1600x^3+240x^4-384x^5-192x^6),$$

$$W^{(7)}_{+3}(x) = \tfrac{1}{46\,080}(1+12x+60x^2+160x^3+240x^4+192x^5+64x^6).$$

[1] P. Ewald, Ann. Phys. (Leipzig) **64**, 253 (1921).
[2] S. W. De Leeuw, J. W. Perram, and E. R. Smith, Proc. R. Soc. London, Ser. A **373**, 27 (1980).
[3] J. W. Perram, H. G. Petersen, and S. W. De Leeuw, Mol. Phys. **65**, 875 (1988).
[4] M. J. L. Sangester and M. Dixon, Adv. Phys. **63**, 247 (1976).
[5] D. J. Adams and G. S. Dubey, J. Comput. Phys. **72**, 156 (1987).
[6] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. (Cambridge University Press, Cambridge, 1992), Chap. 12; see also references therein.
[7] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (IOP, Bristol, 1988).
[8] T. Darden, D. York, and L. Pedersen, J. Chem. Phys. **98**, 10089 (1993).
[9] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. Pedersen, J. Chem. Phys. **103**, 8577 (1995).
[10] B. A. Luty, I. G. Tironi, and W. F. van Gunsteren, J. Chem. Phys. **103**, 3014 (1995).
[11] M. Deserno and C. Holm, J. Chem. Phys. **109**, 7694 (1998), following paper.
[12] However, the idea that something as symmetric as extending spheres will necessarily converge is deceiving. As a warning see, e.g., D. Borwein, J. M. Borwein, and K. F. Taylor, J. Math. Phys. **26**, 2999 (1985).
[13] J.-M. Caillol, J. Chem. Phys. **101**, 6080 (1994).
[14] D. M. Heyes, J. Chem. Phys. **74**, 1924 (1981).
[15] J. Kolafa and J. W. Perram, Mol. Simul. **9**, 351 (1992).
[16] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Clarendon, Oxford, 1987).
[17] H. G. Petersen, J. Chem. Phys. **103**, 3668 (1995).
[18] I. J. Schoenberg, *Cardinal Spline Interpolation* (SIAM, Philadelphia, PA, 1973).
[19] T. A. Darden, A. Toukmaji, and L. G. Pedersen, J. Chim. Phys. **94**, 1346 (1997).
[20] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, 2nd ed. with appl. (Academic, New York, 1985).