JCTC Journal of Chemical Theory and Computation

Article

pubs.acs.org/JCTC

How to Convert SPME to P3M: Influence Functions and Error Estimates

V. Ballenegger,*[,†] J. J. Cerdà,[‡] and C. Holm[§]

[†]Institut UTINAM, Université de Franche-Comté, UMR 6213, 16, route de Gray, 25030 Besançon cedex, France

[‡]Instituto de Física Interdisciplinar y Sistemas Complejos, IFISC (CSIC-UIB), Universitat de les Illes Balears, E-07122 Palma de Mallorca, Spain

[§]Institut für Computer Physik, Universität Stuttgart, Pfaffenwaldring 27, 70569 Stuttgart, Germany

ABSTRACT: We demonstrate explicitly how the two seemingly different particle mesh Ewald methods, the smooth particle mesh Ewald (SPME) and the particle−particle particle mesh (P3M), can be mathematically transformed into each other. This allows us in particular to convert the error estimate of the P3M method in the energy-conserving scheme (also known as "P3M with analytic differentiation") into an error estimate for the SPME method, via a simple change of the lattice Green function. Our error estimate is valid for any values of the SPME parameters (mesh size, spline interpolation order, Ewald splitting parameter, real-space cutoff distance), including odd orders of splines. The problem with the self-forces is avoided thanks to an analytical formula that allows to subtract them directly within the particle mesh calculation. Plots of the accuracy of the SPME forces are provided for a wide range of parameter values. The main use of the error estimate is to allow a simulation program to scan quickly the multidimensional parameter space to find the best set of parameters to achieve a target accuracy at the smallest computational cost. As a byproduct, we show how a SPME code can be transformed into a P3M version by changing a few lines of code. We demonstrate also that the P3M lattice Green function can be approximated by a closed form expression, computable on-the-fly, that provides essentially the same accuracy as the full function.

1. INTRODUCTION

Particle mesh methods, such as particle−particle particle mesh (P3M)[1] and smooth particle mesh Ewald (SPME),[2] are efficient methods to compute long-range Coulomb (or gravitational) forces in simulations of many-particle systems.[3,4] By using fast Fourier transform (FFT) routines to solve the Poisson equation in Fourier space, they convert the task of computing $O(N^2)$ pairwise interactions into a task that scales as $O(N \log N)$ with the number of particles. To improve the accuracy, the pair interaction is moreover decomposed into a short-ranged component that decays rapidly in real space and a smooth long-ranged component that decays rapidly in Fourier space.[5] The short-ranged interactions are computed via a double loop over particle−particle interactions, while the long-range components are computed by particle−mesh mapping of the density (charge or mass), convolution of the mesh-based density with the appropriate Green function, and mesh−particle mapping of the resulting potential.

Several variants of P3M algorithms exist, associated to different choices of the splitting function used to decompose the interaction into a short- and a long-ranged part and to different choices of the differentiation scheme used to compute forces from the mesh-based potential. The differentiation schemes fall in two classes: energy- and momentum-conserving schemes.[1,6] In momentum conserving schemes, the potential is differentiated numerically (using finite differences in real space or by multiplication by $i\boldsymbol{k}$ in Fourier space) and then interpolated to the particle positions in continuous space. In the energy conserving scheme, forces are obtained from the exact gradient of the potential energy of the system discretized on the grid. That latter scheme conserves energy but not momentum.[1,7]

The accuracy of the P3M method has been studied in depth for different choices of ingredients in the method.[1,6,8−10] Thanks to that analysis, the sources of errors in P3M algorithms are fully understood, and the accuracy of the method has been optimized via the use of error-minimizing lattice Green functions, that help reducing truncation and aliasing errors. A further benefit of that analysis is that the error of the method is under full control: An analytical formula predicts the accuracy as function of the various "free" (user defined) parameters of the method. Using this a priori error estimate, a simulation program can determine automatically the set of P3M parameters (mesh size, interpolation order, Ewald splitting parameter, real-space cutoff) that is optimal, in the sense that it achieves the prescribed accuracy at a minimal computational cost.

The purpose of this paper is to show how the SPME method can be transformed into a P3M method and vice versa, and that the error analysis of the P3M method can be adapted to understand and predict the errors of the SPME method.[2] We focus in this paper on SPME, as it supersedes particle mesh Ewald (PME)[26] in practice, and it is implemented in several mainstream molecular dynamics simulation packages, such as GROMACS,[11] AMBER,[12] and NAMD.[13] Whereas the derivation of SPME and P3M are different, the result of the SPME derivation is a special case of the more general P3M method, in which the splitting function is the error function. In the SPME approach, the Green function for the system discretized on the lattice differs from the continuous-space interaction by contributions inherited from the employed exponential

Received: March 15, 2011
Published: January 24, 2012

Euler spline approximation. This contrasts with the P3M approach, where the lattice Green function is optimized to ensure that the mesh calculation give results as close as possible, in a least-squares sense, to those of a continuous-space calculation. A further difference between the two approaches lies in the choice of the differentiation scheme. P3M is often implemented with a momentum-conserving differentiation scheme (the original implementation[14] used finite differences, several later implementations employ ***ik*** differentiation), while the SPME method uses normally the "analytical differentiation" (AD) scheme, which is an energy-conserving scheme introduced in the 1970s.[6,15] Since the AD scheme provides better accuracy in less operations than the finite differences scheme, it can be a cost-effective solution provided that the small local violation of momentum conservation inherent to that scheme can be accepted in the simulations. The optimization of the lattice Green function of a P3M method that uses the AD scheme can be found in refs 6 and 16. In this work, we give the link between the different notation and terminology used in the P3M and SPME methods, building upon the presentation and comparison of particle mesh methods of ref 9. By viewing the SPME method as a P3M method that uses analytical differentiation and a lattice Green function that differs from the optimal P3M error-minimizing one, we show how the a priori error estimates derived for P3M can be converted into an error estimate for the SPME method.

The error estimate for the SPME method introduced in this paper can be used to tune easily the parameters of the method [size of the mesh in each direction (M_1, M_2, M_3), charge interpolation order (P), Ewald splitting parameter (α), and real-space cutoff distance (r_c)] to maximize efficiency and hence save valuable computer and user time. Compared to a SPME error estimate derived recently,[17] our error estimate is more general and simpler to use. It is valid for any interpolation order, and it is also given by an analytic formula which can be evaluated quickly. Contrary to the error estimate derived in ref 17, our approach does not require an expensive calculation of the root-mean-square (rms) error arising from self-forces: We subtract these self-forces directly within the particle mesh calculation using a formula we derived in ref 10.

The paper is organized as follows. In Section 2, we recall the main formulas of the P3M method in the general case of a triclinic cell, with a special emphasis on the analytical differentiation scheme and the definition and meaning of the lattice Green function. This provides the background needed in Section 3 to derive the mathematical link between the two methods, which is embodied in the lattice Green function (eq 3.7) of the SPME method. In Section 4, we give the error estimate for the SPME method and expose also how the SPME algorithm can be modified in practice to use the optimized lattice Green function derived within the P3M approach. The numerical results presented is Section 5 demonstrate the accuracy and usefulness of the error estimate. We show that the P3M lattice Green function optimized for computing the energy, for which a closed form expression exists, provides virtually the same accuracy as the true optimal lattice Green function. For cutoffs and accuracies typically employed in atomistic simulations (e.g., 9 Å cutoff and rms force error of ~10^{-4}), the increase in accuracy brought by modifying SPME to use the optimal P3M lattice Green function turns out to be small. Section 6 contains a summary of the findings and concluding remarks.

2. SUMMARY OF P3M METHOD

We consider an overall neutral system of point charges q_i at locations $\boldsymbol{r}_i$ ($i = 1, ..., N$) in a unit cell of volume V defined by edge vectors $\boldsymbol{a}_\beta$, $\beta = 1, 2, 3$. The charges interact via Coulomb's law, and periodic boundary conditions are applied, i.e., the cell is replicated along the directions of the edge vectors $\boldsymbol{a}_\beta$'s, which need not to be orthogonal. The objective is to compute efficiently the energy E of the system

$$E = \frac{1}{2} \sum_{\boldsymbol{n} \in \mathbb{Z}^3} {\sum_{i,j=1}^{N}}' \frac{q_i q_j}{|\boldsymbol{r}_i - \boldsymbol{r}_j + \underline{\mathbf{L}} \cdot \boldsymbol{n}|} \tag{2.1}$$

and the forces acting on each particle, which are defined by $\boldsymbol{F}_i = -\nabla_i E$. In eq 2.1, $\underline{\mathbf{L}} = \mathrm{diag}(|\boldsymbol{a}_1|, |\boldsymbol{a}_2|, |\boldsymbol{a}_3|)$ is a diagonal matrix formed with the lengths of the primitive cell translation vectors, and the sum over vectors $\boldsymbol{n}$ (which have integer components) account for the periodic boundary conditions. The prime indicates the omission of the $i = j$ term when $\boldsymbol{n} = 0$. The sum is only conditionally convergent, and we assume here the usual spherical summation order with conducting boundary conditions.[18,19]

The P3M method is based on three main ideas: (i) decomposition of the pair interaction into a short-range interaction (calculated in real space) and a smooth long-range interaction with a fast decay in Fourier space; (ii) discretization of the system onto a mesh to take advantage of the FFT algorithm to solve, in Fourier space, the Poisson equation associated to the long-range forces (this equation reduces to a simple multiplication by a lattice Green function); and (iii) use of a lattice Green function that optimizes the accuracy of the calculation.

2.1. Decomposition of the Pair Interaction. The splitting of the interaction into a short- and a long-range part is performed in P3M using the same principle as in the Ewald method,[5,20] namely by introducing a compensating charge distribution around each particle. In the Ewald method, this screening distribution is a Gaussian of width $1/(2\alpha)$, which leads to the following decomposition of the interaction:

$$\frac{1}{r} = \psi(r) + \varphi(r) = \frac{1 - \mathrm{erf}(\alpha r)}{r} + \frac{\mathrm{erf}(\alpha r)}{r} \tag{2.2}$$

where $\mathrm{erf}(x)$ is the error function and the free parameter α controls the relative importance of both contributions. In that decomposition, the long-range part $\phi(r) = \mathrm{erf}(\alpha r)/r$ corresponds to the interaction of a point charge with a Gaussian charge distribution of width $1/(2\alpha)$, while the short-range part $\psi(r) = (1-\mathrm{erf}(\alpha r))/r$ corresponds to the interaction of a point charge with a point charge screened by the Gaussian charge distribution. In the P3M method, several shapes for the screening charge distribution have been considered, among which the Gaussian shape (called S3) corresponds to the above Ewald splitting.[21,22] Since the various screening distributions gave similar accuracies,[1,23] it is assumed here that P3M is implemented with standard Gaussian screening (for other choices, see e.g. ref 24). Since the potential $\psi(r)$ is short ranged, ψ can be computed easily by direct summation in real space. The long-range interaction ϕ is sometimes called the "reciprocal-space interaction", because interactions ϕ between the particles are most easily computed in reciprocal (Fourier) space. The Fourier transform of that interaction:

$$\hat{\varphi}(\boldsymbol{k}) = \int \varphi(\boldsymbol{r}) \mathrm{e}^{-i\boldsymbol{k}\cdot\boldsymbol{r}} \mathrm{d}^3\boldsymbol{r} = \frac{4\pi}{k^2} \exp(-k^2/(4\alpha^2)) \tag{2.3}$$

has a fast decay with k. Upon rewriting the long-range interactions between particles in Fourier space (Ewald method), the energy of the system can be written as the sum of the real-space energy $E(r) = 1/2\sum_{n\in\mathbb{Z}^3} \sum_{i,j=1}^{N}{}' q_i q_j \psi(|\boldsymbol{r}_i - \boldsymbol{r}_j + \underline{\mathrm{L}}\cdot\boldsymbol{n}|)$, the reciprocal-space energy

$$E^{(k)} = \frac{1}{2}\frac{1}{V}\sum_{\boldsymbol{k}\neq 0} \hat{\varphi}(\boldsymbol{k})|\hat{\rho}(\boldsymbol{k})|^2 \tag{2.4}$$

and the self-energy $E^{(s)} = -(\alpha/\sqrt{\pi})\sum_i q_i^2$. In eq 2.4, the structure factor

$$\hat{\rho}(\boldsymbol{k}) = \sum_{i=1}^{N} q_i \mathrm{e}^{-i\boldsymbol{k}\cdot\boldsymbol{r}_i} \tag{2.5}$$

is the Fourier transform of the charge density $\rho(\boldsymbol{r}) = \sum_i q_i \delta(\boldsymbol{r} - \boldsymbol{r}_i)$. The sum over $\boldsymbol{k}$ runs over all reciprocal lattice points (except $\boldsymbol{k} = 0$): The wave vector $\boldsymbol{k}$ is of the form

$$\boldsymbol{k_n} = 2\pi(n_1\boldsymbol{a}_1^* + n_2\boldsymbol{a}_2^* + n_3\boldsymbol{a}_3^*) \tag{2.6}$$

with $\boldsymbol{n} = (n_1, n_2, n_3) \in \mathbb{Z}^3$ a triplet of integers, and the reciprocal lattice vectors $\boldsymbol{a}_\alpha^*$ are defined by the relations $\boldsymbol{a}_\alpha^*\cdot\boldsymbol{a}_\beta = \delta_{\alpha\beta}$ (the Kronecker δ), for $\alpha, \beta = 1, 2, 3$. The reciprocal lattice vectors are thus given by $\boldsymbol{a}_1^* = (1/V)(\boldsymbol{a}_2 \times \boldsymbol{a}_3)$ with cyclic permutations for $\boldsymbol{a}_2^*$ and $\boldsymbol{a}_3^*$, and $V = \boldsymbol{a}_1\cdot(\boldsymbol{a}_2 \times \boldsymbol{a}_3)$. Notice that the self-energy term $E^{(s)}$ subtracts the Ewald self-energy $\phi(0)$ of each particle included in the reciprocal energy $E^{(k)}$.

2.2. Mesh Calculation of the Reciprocal Energy. The computation of eq 2.4 requires a priori $O(N^2)$ operations (or $O(N^{3/2})$ if the cutoff used in the real-space calculation depends on the Ewald splitting parameter α in a specific way),[25] because eq 2.5 is a sum of N terms and needs to be evaluated for a number of wave vectors that grows linearly with the volume of the simulation box and hence with the number of particles. The basic idea in particle mesh methods is to reduce the computational cost to $O(N \log N)$ by replacing the continuous Fourier transform $\hat{\rho}(\boldsymbol{k})$ in eq 2.4 by a FFT. To this end, the system is discretized onto a grid with fixed lattice spacing, that is, each direction $\boldsymbol{a}_\alpha$ of the unit cell is divided into N_α mesh points. The edges of a mesh cell have lengths $h_1 = |\boldsymbol{a}_1|/N_1$, $h_2 = |\boldsymbol{a}_2|/N_2$, and $h_3 = |\boldsymbol{a}_3|/N_3$. The charge q_{r_n} at each mesh point $\boldsymbol{r_n}$ is computed using an assignment function $W(r)$:

$$q_{\boldsymbol{r_n}} \equiv \sum_i q_i W(\boldsymbol{r_n} - \boldsymbol{r}_i) \tag{2.7}$$

where periodic boundary conditions are implicitly assumed in eq 2.7. If analytical differentiation is used, $W(r)$ must be differentiable. An assignment function that distributes a charge to its P nearest mesh points (separately in each dimension) is called a P order assignment function. It is usually chosen to be of the product form

$$W^{(P)}(\boldsymbol{r}) = w(s_1)w(s_2)w(s_3) \tag{2.8}$$

where $w(s)$ is an even function with finite support $[-P/2, P/2]$ and

$$\boldsymbol{r} = \frac{s_1}{N_1}\boldsymbol{a}_1 + \frac{s_2}{N_2}\boldsymbol{a}_2 + \frac{s_3}{N_3}\boldsymbol{a}_3 \tag{2.9}$$

where $s_\alpha = N_\alpha \boldsymbol{a}_\alpha^*\cdot\boldsymbol{r} \in [0, N_\alpha]$ is the coordinate of $\boldsymbol{r}$ along direction $\boldsymbol{a}_\alpha$ scaled by the number of mesh points N_α. Hockney and Eastwood's choice for $w(s)$ is a B-spline of order P. More details on charge assignment can be found in refs [1,9,16,21,23] We can introduce the mesh-based charge density by setting

$$\rho_M(\boldsymbol{r_n}) = \frac{N_1N_2N_3}{V} q_{\boldsymbol{r_n}} \tag{2.10}$$

where $\boldsymbol{r}_n \in \mathbb{M}$ and $\mathbb{M} = \{\boldsymbol{r}_n = (n_1/N_1)\boldsymbol{a}_1 + (n_2/N_2)\boldsymbol{a}_2 + (n_3/N_3)\boldsymbol{a}_3\text{: } n_\alpha = 0, 1, ..., N_\alpha - 1, \alpha = 1, 2, 3\}$ denotes the set of all mesh points. Notice that $\rho_M(\boldsymbol{r}_n)$ can be written as a convolution of the (periodic) charge density $\rho(\boldsymbol{r})$ with the assignment function:

$$\rho_M(\boldsymbol{r}) = \frac{N_1N_2N_3}{V}\int_{\mathbb{R}^3} \rho(\boldsymbol{r}')W(\boldsymbol{r} - \boldsymbol{r}')\mathrm{d}^3\boldsymbol{r}' \tag{2.11}$$

Once the charges are assigned to the mesh, the finite Fourier transform of the mesh-based charge density

$$\tilde{\rho}_M(\boldsymbol{k}) = \mathrm{FFT}[\rho_M] = \frac{V}{N_1N_2N_3}\sum_{\boldsymbol{r_n}\in\mathbb{M}} \rho_M(\boldsymbol{r_n})\mathrm{e}^{-i\boldsymbol{k}\cdot\boldsymbol{r_n}} \tag{2.12}$$

may be computed via the FFT algorithm (we denote finite Fourier transforms by a tilde instead of a hat to distinguish them from usual Fourier transforms). Notice that the volume element $\mathrm{d}^3\boldsymbol{r}$ of the usual transform takes here the form of the volume of a mesh cell $V/(N_1N_2N_3)$. $\tilde{\rho}_M\,(\boldsymbol{k})$ is defined for wave vectors in the (finite) reciprocal lattice

$$\begin{aligned}\tilde{\mathbb{M}} = \{&\boldsymbol{k_n} = 2\pi(n_1\boldsymbol{a}_1^* + n_2\boldsymbol{a}_2^* + n_3\boldsymbol{a}_3^*)\text{: } n_\alpha \in \mathbb{Z},\\ &|n_\alpha| \leq (N_\alpha - 1)/2,\ \alpha = 1, 2, 3\}\end{aligned} \tag{2.13}$$

In the P3M method, the reciprocal Ewald energy eq 2.4 is approximated by the mesh formula

$$E_{\mathrm{P3M}}^{(k)} = \frac{1}{2}\frac{1}{V}\sum_{\substack{\boldsymbol{k}\neq 0\\ \boldsymbol{k}\in\tilde{\mathbb{M}}}} \tilde{G}(\boldsymbol{k})|\tilde{\rho}_M(\boldsymbol{k})|^2 \tag{2.14}$$

The reciprocal interaction $\hat{\varphi}(\boldsymbol{k})$ is thus replaced by a lattice Green function (or "influence function") $\tilde{G}(\boldsymbol{k})$, which will be chosen, as suggested by Eastwood,[6] in a way that optimizes the accuracy of the mesh calculation. For later purposes, it is useful to note that eq 2.4 can be expressed in real-space as

$$E^{(k)} = \frac{1}{2}\sum_{i,j=1}^{N} q_i q_j G_{\mathrm{cont.}}(\boldsymbol{r}_i - \boldsymbol{r}_j) \tag{2.15}$$

where $G_{\mathrm{cont.}}(\boldsymbol{r}) = V^{-1}\sum_{\boldsymbol{k}\neq 0} \exp(i\boldsymbol{k}\cdot\boldsymbol{r})\hat{\varphi}(\boldsymbol{k})$ is the continuous-space Green function of the system under periodic boundary conditions. If that Green function reduces to $\hat{\varphi}(\boldsymbol{k})$ in Fourier space $[\hat{G}_{\mathrm{cont.}}(\boldsymbol{k}) = \hat{\varphi}(\boldsymbol{k})$ for $\boldsymbol{k}\in\tilde{\mathbb{M}}]$, it differs from $\phi(r)$ in real space, since it is given by the inverse Fourier series (rather than full) transform of $\hat{\varphi}(\boldsymbol{k})$. That difference accounts precisely for the periodic boundary conditions that are embedded in the Green function. When the system is discretized onto the mesh, the energy is computed using eq 2.14 or equivalently in real space by

$$E_{\mathrm{P3M}}^{(k)} = \frac{1}{2}\sum_{\boldsymbol{r_n},\boldsymbol{r_m}\in\mathbb{M}} q_{\boldsymbol{r_n}} q_{\boldsymbol{r_m}} G(\boldsymbol{r_n} - \boldsymbol{r_m}) \tag{2.16}$$

where $G(\boldsymbol{r}_n) = \mathrm{FFT}^{-1}[\tilde{G}](\boldsymbol{r}_n)$ is the lattice Green function in real space. The reciprocal-space energy (eq 2.15) can also be written as $E^{(k)} = (1/2)\sum_i q_i \Phi^{(k)}(r_i)$ in term of the (reciprocal space) electrostatic potential at point r defined by

$$\Phi^{(k)}(\boldsymbol{r}) = \int_V \rho(\boldsymbol{r}')G_{\text{cont.}}(\boldsymbol{r} - \boldsymbol{r}')\mathrm{d}^3\boldsymbol{r}' \tag{2.17}$$

In Fourier space, this convolution becomes the product $\hat{\Phi}^{(k)}(\boldsymbol{k}) = \hat{\rho}(\boldsymbol{k})\hat{G}_{\text{cont.}}(\boldsymbol{k})$, which can be approximated on the mesh by

$$\hat{\Phi}^{(k)}(\boldsymbol{k}) \simeq \tilde{\rho}_M(\boldsymbol{k})\tilde{G}(\boldsymbol{k}) \tag{2.18}$$

2.3. Calculation of Forces. Once the potential $\Phi^{(k)}$ is known, the forces acting on the particles are obtained by differentiating the energy. This step can be performed in several ways:

(i) finite-differences scheme: differentiation by computing finite differences in real space

(ii) analytical differentiation (AD) scheme: forces are obtained by differentiating exactly the mesh-based energy (eq 2.16):

$$\begin{aligned} \boldsymbol{F}_i^{(k)} &= -\nabla_i E_{\text{P3M}}^{(k)} \\ &= -\sum_{\boldsymbol{r_n}, \boldsymbol{r_{n'}} \in \mathbb{M}} (\nabla_i q_{\boldsymbol{r_n}}) q_{\boldsymbol{r_{n'}}} G(\boldsymbol{r_n} - \boldsymbol{r_{n'}}) \\ &= -q_i \sum_{\boldsymbol{r_n} \in \mathbb{M}} \Phi^{(k)}(\boldsymbol{r_n}) \nabla_i W(\boldsymbol{r}_i - \boldsymbol{r}_n) \end{aligned} \tag{2.19}$$

where we used that $G(\boldsymbol{r}_n)$ is even and $\Phi^{(k)}(\boldsymbol{r}_n) \equiv \sum_{r_{n'} \in \mathbb{M}} q_{r_{n'}} G(\boldsymbol{r}_n - \boldsymbol{r}_{n'})$ is the potential at mesh point $\boldsymbol{r}_n$. Forces are therefore calculated in this scheme by interpolating the mesh-based potential back to a particle position using the gradient $\nabla W(\boldsymbol{r})$ of the charge assignment function.

(iii) *ik*-differentiation scheme: differentiation in Fourier space by multiplying the potential by *ik*.

In the first two schemes, one computes the inverse FFT of the (scalar) electrostatic potential (eq 2.18) to get $\Phi^{(k)}(\boldsymbol{r}_n)$ in real space and calculate then the forces either by using finite differences or by using the exact analytic gradient $\nabla W(\boldsymbol{r})$. It should be emphasized that the analytical differentiation scheme amounts to computing the exact gradient of an approximate energy (the energy, eq 2.16, of the discretized system), while the *ik*-differentiation scheme corresponds to approximating the formula for the force (i.e., the gradient of the exact energy, eq 2.15) as well as possible on a mesh.

The *ik*-differentiation scheme, also known as force-interpolation scheme, is the most accurate one, but it requires to compute the inverse FFT of a vectorial field. The force-interpolation scheme amounts to calculating the electric (or gravitational) field at each mesh point and to interpolating this vectorial "force" field back to the particle positions. In the context of particle mesh methods, force interpolation was introduced first in the PME method[26] and then in P3M by Bertschwinger et al.[8] The *ik*-differentiation and finite-differences schemes are known as "momentum conserving schemes", as they conserve momentum but not the energy.[1]

The forces obtained with analytical differentiation are conservative (since they are the exact gradient of an energy), so the AD scheme is an energy-conserving scheme. In Lewis' original energy-conserving scheme,[15] analytical differentiation is used together with a potential that is solved by using a specific finite-difference form of Poisson's equation, but Langdon showed that almost any potential solver can be used.[7] The AD scheme conserves the energy (in the limit of small time steps) but not momentum.[1,7] When using this scheme in a simulation, a correction must be applied to the forces to enforce at least the conservation of the center-of-mass momentum.[2] That correction has unfortunately the collateral effect of breaking the exact conservation of energy.

For a more thorough discussion of the merits and drawbacks of each scheme and comparison of accuracies, see refs 1, 3, and 9. P3M is often associated with finite differences because Hockney and Eastwood advocated the use of this scheme in their book[1] (that scheme turned out to be the most effective at that time where memory constraints were strong and where the computations were done at low force accuracy [interpolation order 1 or 2] due to the small computing power). We will refer to a P3M method with *ik*-differentiation as "P3M-ik" and to a P3M method with analytical differentiation as "P3M-AD".

2.4. Optimal Lattice Green Functions. In the simplest implementation of P3M, the lattice Green function would be taken to be equal to the continuous-space Green function:

$$\tilde{G}(\boldsymbol{k_n}) = \tilde{G}_{\text{cont.}}(\boldsymbol{k_n}) = \hat{\varphi}(\boldsymbol{k_n}) \tag{2.20}$$

But this simple Poisson solver gives poor accuracy, because it neglects the reshaping effect due to the charge assignment onto the mesh, which amounts to giving a finite size to the particles. The mesh-based charge density can indeed be seen as resulting from the sampling, at grid points, of a continuous smoothed charge density obtained by replacing each point charge by a cloud of charge density $W(\boldsymbol{r})$.[1] We stress that this reshaping effect is due purely to charge assignment, and it is not related to the Ewald Gaussian screening contained in the reciprocal interaction $\hat{\varphi}(\boldsymbol{k})$. Because of this assignment (or interpolation) smoothing, the high-frequency harmonics of the mesh-based charge density $\tilde{\rho}_M(\boldsymbol{k})$ are damped compared to those of $\hat{\rho}(\boldsymbol{k})$. In P3M, the lattice Green function is adjusted to compensate for this damping and also to reduce other errors arising from truncation and aliasing.

Since eq 2.11 is a convolution, the (continuous) Fourier transform of $\rho_M(\boldsymbol{r})$ is given by the product $\tilde{\rho}_M(\boldsymbol{k}) = \hat{U}(\boldsymbol{k})\hat{\rho}(\boldsymbol{k})$, where $\hat{U}(\boldsymbol{k})$ is equal to the Fourier transform $\hat{W}(\boldsymbol{k})$ of the charge assignment function divided by the volume of a mesh cell. For a B-spline assignment function of order P, one has

$$\begin{aligned} \hat{U}(\boldsymbol{k}) &\equiv \frac{N_1N_2N_3}{V}\hat{W}(\boldsymbol{k}) \\ &= \left(\frac{\sin(\pi n_1/N_1)}{\pi n_1/N_1}\right)^P \left(\frac{\sin(\pi n_2/N_2)}{\pi n_2/N_2}\right)^P \left(\frac{\sin(\pi n_3/N_3)}{\pi n_3/N_3}\right)^P \end{aligned} \tag{2.21}$$

If the harmonic content of function $\hat{\rho}_M(\boldsymbol{k})$ is negligible beyond the Nyquist frequency π/h_β (for each direction $\beta = 1, 2, 3$), no information is lost by the sampling on the mesh, and the discrete Fourier transform yields the exact spectrum: $\tilde{\rho}_M(\boldsymbol{k}) = \hat{\rho}_M(\boldsymbol{k}) = \hat{U}(\boldsymbol{k})\hat{\rho}(\boldsymbol{k})$. From this relation, and by comparing eqs 2.14 to 2.4, we find that the accuracy of the Poisson solver can be increased by setting the lattice Green function to

$$\tilde{G}(\boldsymbol{k_n}) = \frac{\hat{\varphi}(\boldsymbol{k_n})}{\hat{U}^2(\boldsymbol{k_n})} \tag{2.22}$$

That function contains a "sharpening" factor $1/\hat{U}^2(\boldsymbol{k}_n)$ that compensates for the smoothing introduced by the mapping of the charges onto the mesh and by the back interpolation of the mesh results to continuous space. The use of the adjusted Green function, eq 2.22, in place of eq 2.20 improves substantially the accuracy of the particle mesh calculation.

A key idea in P3M is to consider $\tilde{G}(\boldsymbol{k}_n)$ as an adjustable function, that has to be optimized such that the results of the mesh calculation are as close as possible to the results of the original continuous problem. The optimal P3M lattice Green function is the one that minimizes the rms error of the mesh calculation. The optimal lattice Green function includes corrections beyond the previous sharpening factor. The optimization of the lattice Green function enhances the accuracy at no computational cost because: (i) if one disregards caching issues, $\tilde{G}(\boldsymbol{k}_n)$ is a fixed quantity (apart in constant pressure simulations) that can be tabulated; and (ii) the lattice Green function optimized for computing the energy, which can be computed on-the-fly since it is known in closed analytical form, provides in practice virtually the same accuracy as the true optimal lattice Green function associated to the P3M-AD and P3M-ik force calculations (see Section 5).

To derive the optimal P3M lattice Green function, one needs the exact relation between the structure factors $\tilde{\rho}_M(\boldsymbol{k})$ and $\hat{\rho}(\boldsymbol{k})$, which can be obtained using the sampling and convolution theorems (see ref 1 or eq 5.2 of ref 27):

$$\tilde{\rho}_M(\boldsymbol{k}) = \sum_{\boldsymbol{m}\in\mathbb{Z}^3} \hat{U}(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})\hat{\rho}(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}), \quad \boldsymbol{k}=\boldsymbol{k}_{\boldsymbol{n}} \in \tilde{\mathbb{M}} \tag{2.23}$$

where $\underline{\mathbf{N}} = \mathrm{diag}(N_1, N_2, N_3)$ is a diagonal matrix formed with the numbers of mesh points in each dimension. In agreement with eq 2.6, the vector

$$\begin{aligned}\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}} &= 2\pi((n_1+m_1N_1)\boldsymbol{a}_1^* + (n_2+m_2N_2)\boldsymbol{a}_2^* \\ &\quad + (n_3+m_3N_3)\boldsymbol{a}_3^*) = \boldsymbol{k}_{\boldsymbol{n}} + \boldsymbol{k}_{\underline{\mathbf{N}}\boldsymbol{m}}\end{aligned} \tag{2.24}$$

is equal to the vector $\boldsymbol{k}_n$ shifted outside the first Brillouin zone by reciprocal lattice vector $\boldsymbol{k}_{\underline{\mathbf{N}}m}$. Equation 2.23 embodies in a simple formula the effects on the spectrum induced by the charge assignment and the sampling of the charge density on a mesh. The sum over integer triplets $\boldsymbol{m}$ shows that spurious contributions from the high-frequency spectrum of $\hat{U}(\boldsymbol{k})\hat{\rho}(\boldsymbol{k})$ are introduced into the first Brillouin zone $\tilde{\mathbb{M}}$. These unwanted copies of the other Brillouin zones into the first one are known as aliasing errors.[1] To reduce aliasing errors, the charge assignment function $\hat{U}(\boldsymbol{k})$ must decay quickly in k-space (it must also decay to zero over a short distance in real-space to avoid having to assign the charges to a too large number of grid points). If aliasing ($\boldsymbol{m} \neq 0$) contributions are neglected in eq 2.23, one recovers the relation $\tilde{\rho}_M(\boldsymbol{k}) = \hat{U}(\boldsymbol{k})\hat{\rho}(\boldsymbol{k})$ that leads to the modified lattice Green function, eq 2.22.

Formula 2.23 can be used to predict analytically, for any positions of two particles within the simulation box, the error of the P3M pair interaction (energy, force, etc.) compared to the exact continuous-space interaction. By minimizing the rms error of this pair interaction, one finds the optimal P3M lattice Green function. The optimal lattice Green function for computing the energy reads[27]

$$\begin{aligned}\tilde{G}_{\mathrm{opt}}(\boldsymbol{k}_{\boldsymbol{n}}) &= \frac{\sum_{\boldsymbol{m}} \hat{\varphi}(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})\hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})}{(\sum_{\boldsymbol{m}} \hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}))^2} \\ &\simeq \hat{\varphi}(\boldsymbol{k}_{\boldsymbol{n}})\frac{\hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}})}{(\sum_{\boldsymbol{m}} \hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}))^2}\end{aligned} \tag{2.25}$$

where the last expression holds thanks to the fast decay of $\hat{\varphi}(\boldsymbol{k})$. The sum over $\boldsymbol{m}$ in the numerator originates from the minimization of truncation errors, which are small if $\hat{\varphi}(\boldsymbol{k})$ decays rapidly, and the sum over $\boldsymbol{m}$ in the denominator arises from aliasing errors. As can be seen by comparing the optimal lattice Green function, eq 2.25, to the simpler result, eq 2.22, the "sharpening" factor $1/\hat{U}^2(\boldsymbol{k})$ is somewhat damped in eq 2.25 to compensate partially for aliasing errors. The denominator in eq 2.25 can actually be calculated analytically. For the first few spline interpolation orders, one has for instance:[23]

$$\begin{aligned}&\sum_{m\in\mathbb{Z}} \hat{U}^2(2\pi(n+Nm)) \\ &= \begin{cases} 1 - 2z^2/3 & \text{for } P = 2 \\ 1 - z^2 + 2z^4/15 & \text{for } P = 3 \\ 1 - 4z^2/3 + 2z^4/5 + 4z^6/315 & \text{for } P = 4 \\ 1 - 5z^2/3 + 7z^4/9 - 17z^6/189 & \text{for } P = 5 \\ \quad + 2z^8/2835 & \end{cases}\end{aligned} \tag{2.26}$$

where $z = \sin\,(\pi n/N)$ and the full function is

$$\sum_{\boldsymbol{m}} \hat{U}^2(\mathbf{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}) = \prod_{j=1,2,3} \sum_{m\in\mathbb{Z}} \hat{U}^2(2\pi(n_j + N_j m)) \tag{2.27}$$

This provides a closed form expression for the lattice Green function, eq 2.25, which is most useful for computing this function on-the-fly.

Expressions for the optimal lattice Green function to compute forces, both with finite differences or ik-differentiation, are given in refs 1 and 8. The optimal influence function for the P3M-AD algorithm reads[6,16]

$$\begin{aligned}\tilde{G}_{\mathrm{opt}}(\boldsymbol{k}_{\boldsymbol{n}}) &= \frac{\sum_{\boldsymbol{m}} \hat{\varphi}(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})k_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}^2\hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})}{(\sum_{\boldsymbol{m}} \hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}))\sum_{\boldsymbol{m}} \hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})k_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}^2} \\ &\simeq \hat{\varphi}(\boldsymbol{k}_{\boldsymbol{n}})\frac{\hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}})k_{\boldsymbol{n}}^2}{(\sum_{\boldsymbol{m}} \hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}))\sum_{\boldsymbol{m}} \hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})k_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}^2}\end{aligned} \tag{2.28}$$

In all earlier expressions, the sum over $\boldsymbol{m} \in \mathbb{Z}^3$ converges quickly. The lattice Green functions can therefore be evaluated in practice by truncating the sum at $|m_\beta| = 2$ in each direction ($\beta = 1, 2, 3$).

3. LINK BETWEEN SPME AND P3M: THE SPME LATTICE GREEN FUNCTION

The SPME method corresponds to a specific form of particle—particle particle mesh algorithm. It uses the traditional Ewald splitting, eq 2.2, of the interaction and an FFT-based Poisson solver to compute the reciprocal-space potential (contrary to P3M, the lattice Green function of the Poisson solver is not optimized in SPME). The SPME method differs from the standard P3M method mainly by the choice of the analytical differentiation scheme and by the point of view taken when discretizing the system onto a mesh (which leads to the SPME lattice Green function).

The combination of an FFT-based Poisson solver with the analytical differentiation (energy-conserving) scheme corresponds to the P3M-AD algorithm introduced in the previous section. In a comparison of schemes, Eastwood found that the P3M-AD method was not competitive and discarded the method, but his analysis was limited to one-dimensional problems.[6] In three dimensions, the use of the AD scheme results in a very fast and efficient algorithm, since it requires fewer FFT's than *ik*-differentiation (the main drawback of the method lies in local violations of momentum conservation). That choice of ingredients makes the SPME method very effective.

In the SPME methods, the discretization of eqs 2.4 and 2.5 onto a mesh is viewed alternatively as interpolation of the exponential factor $\exp(i\boldsymbol{k}{\cdot}\boldsymbol{r}_i)$ from its values at nearby grid points. This is done using Lagrange interpolation in the PME method, while SPME uses exponential Euler splines. The latter approximation amounts to setting

$$\hat{\rho}(\boldsymbol{k}) \approx b(\boldsymbol{k})\tilde{\rho}_M(\boldsymbol{k}) \tag{3.1}$$

and hence to calculate

$$E^{(k)}_{\mathrm{SPME}} = \frac{1}{2}\frac{1}{V}\sum_{\substack{\boldsymbol{k}\neq 0\\ \boldsymbol{k}\in\widetilde{\mathbb{M}}}} \hat{\varphi}(\boldsymbol{k})|\hat{b}(\boldsymbol{k})|^2|\hat{\rho}_M(\boldsymbol{k})|^2 \tag{3.2}$$

where $\tilde{\rho}_M(\boldsymbol{k})$ is computed using a B-spline of order P as assignment function, which is identical, up to a shift of $P/2$, to the charge assignment function introduced in P3M:

$$M^{(P)}(s) = w(s - P/2) \tag{3.3}$$

The shift of the charge mesh in SPME by $P/2$ compared to eq 2.7 is of course irrelevant since the system obeys periodic boundary conditions. In eq 3.2, $b(\boldsymbol{k})$ is given by $b(\boldsymbol{k}_n) = b(n_1)b(n_2)b(n_3)$, where for even P (see ref 2 for the treatment of odd P values)

$$b(n_\alpha) = \frac{\mathrm{e}^{2\pi i P n_\alpha/N_\alpha}}{\sum_{l=1}^{P-1} M^{(P)}(l)\mathrm{e}^{2\pi i n_\alpha l/N_\alpha}} \tag{3.4}$$

The SPME formula, eq 3.2, for the energy is identical to the P3M formula, eq 2.14, with specific choices of the lattice influence function and of the charge assignment function. In the SPME algorithm, the exponential Euler spline approximation gives thus rise to a modified lattice Green function

$$\tilde{G}_{\mathrm{SPME}}(\boldsymbol{k}) = \hat{\varphi}(\boldsymbol{k})|b(\boldsymbol{k})|^2 \tag{3.5}$$

That expression for $\tilde{G}_{\mathrm{SPME}}(\boldsymbol{k})$ corrects and generalizes eq 28 of ref 9 to general triclinic unit cells. We note that the denominator in $b(\boldsymbol{k})$ is a FFT of $M^{(P)}(s)$ and is hence given by an aliasing sum:

$$\begin{aligned} b(\boldsymbol{k_n}) &= \frac{\mathrm{e}^{i\boldsymbol{k_n}\cdot(\frac{\boldsymbol{a}_1}{N_1}+\frac{\boldsymbol{a}_2}{N_2}+\frac{\boldsymbol{a}_3}{N_3})P}}{\frac{N_1N_2N_3}{V}\sum_{\boldsymbol{m}}\hat{M}(-\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})} \\ &= \frac{\mathrm{e}^{i\boldsymbol{k_n}\cdot(\frac{\boldsymbol{a}_1}{N_1}+\frac{\boldsymbol{a}_2}{N_2}+\frac{\boldsymbol{a}_3}{N_3})P/2}}{\sum_{\boldsymbol{m}}\hat{U}(k_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})} \end{aligned} \tag{3.6}$$

where we used eqs 3.3, 2.8, and 2.21 and the symmetry $\hat{U}(-\boldsymbol{k}) = \hat{U}(\boldsymbol{k})$ in the second equality. The lattice Green function of SPME can eventually be rewritten as

$$\tilde{G}_{\mathrm{SPME}}(\boldsymbol{k_n}) = \frac{\hat{\varphi}(\boldsymbol{k_n})}{(\sum_{\boldsymbol{m}}\hat{U}(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}))^2} \tag{3.7}$$

in terms of the harmonics $\hat{U}(\boldsymbol{k})$ of the spline assignment function defined in P3M (eq 2.21). Comparing eq 3.7 to the optimal P3M influence function for the energy, eq 2.25, we see that if aliasing terms $\boldsymbol{m} \neq 0$ are neglected, both functions reduce to the same lattice Green function, eq 2.22, containing the sharpening factor $1/\hat{U}^2(\boldsymbol{k})$.

We note finally that Eastwood's idea of minimizing rms errors of the mesh calculation can also be implemented in a SPME approach, and this has been done in the case of a SPME method with *ik*-differentiation.[19,28] The lattice Green function of the resulting optimized SPME-ik algorithm should then be equivalent to the optimal P3M-ik lattice Green function (both methods were found in ref 19 to have a similar accuracy). Contrary to the P3M approach, the SPME route does however not enlighten the connection between the adjusted lattice Green function and the errors that originate from interpolation smoothing and aliasing.

4. FROM P3M TO SPME AND BACK: CONVERTING ALGORITHMS AND ERROR ESTIMATES

To convert a SPME algorithm into a P3M algorithm, or vice versa, with the same differentiation scheme (that is SPME-ik ↔ P3M-ik or SPME ↔ P3M-AD), it suffices to change the influence function in the program. Since P3M is not always implemented with Gaussian screening, one might furthermore have to make sure that the real-space part of the P3M algorithm does compute the real-space energy associated to the Ewald splitting, eq 2.2.

The conversion SPME → P3M-AD is done for example by replacing eqs 3.7 by 2.28, which is done in practice by replacing the SPME array $|b(\boldsymbol{k}_n)|^2 = (1/(\sum_m\hat{U}(\boldsymbol{k}_{n+\underline{\mathrm{N}}m}))^2$, where $\hat{U}(\boldsymbol{k})$ is defined in eq 2.21 by the array

$$|b_{\mathrm{opt}}(\boldsymbol{k_n})|^2 = \frac{\hat{U}^2(\boldsymbol{k_n})\boldsymbol{k}_{\boldsymbol{n}}^2}{(\sum_{\boldsymbol{m}}\hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}))\sum_{\boldsymbol{m}}\hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})\boldsymbol{k}^2_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}} \tag{4.1}$$

$$\simeq \frac{\hat{U}^2(\boldsymbol{k_n})}{(\sum_{\boldsymbol{m}}\hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}))^2} \tag{4.2}$$

The last expression, eq 4.2, corresponds to using the lattice Green function, eq 2.25, optimized for the energy instead of eq 2.28. Numerical tests have shown that the use of this simpler lattice Green function, which does not need to be tabulated since it can be written in closed form using eq 2.26, does not lead to any noticeable loss of accuracy.

Using the array $|b_{\text{opt}}(\boldsymbol{k_n})|^2$ in a SPME algorithm will ensure that the FFT Poisson solver is optimized, which will improve somewhat the accuracy at no computational cost. As mentioned in the previous section, an optimized lattice Green function for SPME in the case of the *ik*-differentiation scheme has been derived within the SPME approach, resulting in an algorithm that is basically identical to the P3M-ik algorithm.[19,28]

Similarly, one can convert the error estimate derived for the P3M-AD method into an error estimate for the SPME method. The mean-square error of the forces

$$(\Delta F)^2 \equiv \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{F}_i - \boldsymbol{F}_i^{\text{exact}})^2 \tag{4.3}$$

can be written as the sum of a real- and a reciprocal-space contributions:

$$(\Delta F)^2 = (\Delta F^{(r)})^2 + (\Delta F^{(k)})^2 \tag{4.4}$$

if both errors are assumed to be statistically independent. These errors can be predicted exactly for random uniformly distributed charge distributions, as function of the system's parameters (number N of charged particles, valencies $\{q_i\}$, volume V) and of the method's parameters: Ewald splitting parameter α, real-space cutoff r_c, mesh size (N_1, N_2, N_3) and interpolation order P of the charge assignment function. The accuracy of the real-space part of the calculation is controlled by a formula derived by Kolafa and Perram:[29]

$$\Delta F^{(r)} = \frac{2\mathrm{e}^{-\alpha^2 r_c^2}}{\sqrt{r_c N V}}\sum_{i=1}^{N} q_i^2 \tag{4.5}$$

The accuracy of the reciprocal-space part of the calculation is controlled by[9]

$$\Delta F^{(k)} = V^{-1}\sqrt{\frac{Q}{N}}\sum_{i=1}^{N} q_i^2 \tag{4.6}$$

where

$$Q \equiv \int_V \int_V (F_{\text{P3M}}(\boldsymbol{r}_1, \boldsymbol{r}_2) - F_{\text{exact}}^{(k)}(\boldsymbol{r}_1, \boldsymbol{r}_2))^2 \mathrm{d}^3\boldsymbol{r}_1 \mathrm{d}^3\boldsymbol{r}_2 \tag{4.7}$$

measures the error in the P3M force $F_{\text{P3M}}(\boldsymbol{r}_1, \boldsymbol{r}_2)$ on a test unit charge at $\boldsymbol{r}_1$ due to a unit charge at $\boldsymbol{r}_2$, for all possible positions of both particles in the simulation box [$F_{\text{exact}}^{(k)}(\boldsymbol{r}_1, \boldsymbol{r}_2)$] is the exact force which can be calculated with an Ewald sum]. Rewriting eq 4.7 in Fourier space and using the explicit expressions for the approximate and the exact forces, Hockney and Eastwood showed that error Q is given by a quadratic polynomial in the influence function

$$Q = \sum_{\substack{\boldsymbol{k}\neq 0\\ \boldsymbol{k}\in\widetilde{\mathbb{M}}}} [A(\boldsymbol{k})\tilde{G}^2(\boldsymbol{k}) - 2B(\boldsymbol{k})\tilde{G}(\boldsymbol{k})] + C \tag{4.8}$$

where C is the constant defined by

$$C = \sum_{\boldsymbol{k}\neq 0} \hat{\varphi}(\boldsymbol{k})\boldsymbol{k}^2 = \sum_{\substack{\boldsymbol{k_n}\neq 0\\ \boldsymbol{k_n}\in\widetilde{\mathbb{M}}}} \sum_{\boldsymbol{m}} \hat{\varphi}(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}^2 \tag{4.9}$$

and expressions for $A(\boldsymbol{k})$ and $B(\boldsymbol{k})$ for both *ik* and analytical differentiation schemes can be found in refs 8 and 16 and are recalled below for convenience:

$$A(\boldsymbol{k_n}) = \begin{cases} (\sum_{\boldsymbol{m}} \hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}))(\sum_{\boldsymbol{m}'} \hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}'})\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}'}^2) & \text{for AD scheme} \\ (\sum_{\boldsymbol{m}} \hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}))^2\boldsymbol{k}_{\boldsymbol{n}}^2 & \text{for } i\boldsymbol{k} \text{ scheme} \end{cases} \tag{4.10}$$

and

$$B(\boldsymbol{k_n}) = \begin{cases} \sum_{\boldsymbol{m}} \hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})\hat{\varphi}(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}}^2 & \text{for AD scheme} \\ \sum_{\boldsymbol{m}} \hat{U}^2(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})\hat{\varphi}(\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}})\boldsymbol{k}_{\boldsymbol{n}}\cdot\boldsymbol{k}_{\boldsymbol{n}+\underline{\mathbf{N}}\boldsymbol{m}} & \text{for } i\boldsymbol{k} \text{ scheme} \end{cases} \tag{4.11}$$

The optimal lattice Green function is the one for which error Q is minimal. It is easy to see that minimizing eq 4.8 yields $\tilde{G}_{\text{opt}}(\boldsymbol{k}) = B(\boldsymbol{k})/A(\boldsymbol{k})$. The numerator in eq 2.28 is therefore the expression of $B(\boldsymbol{k})$ in the case of analytical differentiation, while the denominator is $A(\boldsymbol{k})$.

Formula 4.6 together with eq 4.8 predicts the accuracy of the particle mesh calculation as function of the system's and method's parameters. It applies not only to the P3M, P3M-ik, and P3M-AD methods [in which case Q reduces to its minimal value $Q_{\text{min}} = C - \sum_{\boldsymbol{k}\neq 0} B^2(\boldsymbol{k})/A(\boldsymbol{k})$], but also to the PME and SPME methods. In the case of the SPME method, we simply substitute the SPME influence function, eq 3.7, in eq 4.8 and use the expressions for $A(\boldsymbol{k})$ and $B(\boldsymbol{k})$ that correspond to analytical differentiation.

When using the analytical differentiation scheme (i.e., the SPME or P3M-AD method), each particle is subjected to a self-force that depends on the particle position relative to a mesh cell.[1,10,30] It is important to note that the contribution of the self-forces to the rms error $\Delta F^{(k)}$ is not included in the above error estimate. The rms error due to self-interactions can also be predicted analytically, as has been done for example in the case of the P3M error estimate for the energy[27] (which includes contributions due to self-energies) and for self-energies and self-torques in a dipolar P3M algorithm.[31] Instead of including the contribution of self-interactions into the error estimate, it is actually better to simply subtract the self-interactions directly within the particle mesh calculation. This can be done easily thanks to the analytical formulas for the self-interactions (self-energy and self-force in the case of the SPME and P3M-AD algorithms) derived in ref 10. Alternatively, the self-forces can also be measured and tabulated before the start of a simulation, as first suggested by Cerutti et al.[30] The analysis of ref 10 shows that the subtraction of self-forces in the P3M-AD or SPME algorithm improves the accuracy of the calculation by a factor between 0% and 30%, depending on the values of the method's parameters. All numerical results presented in the next section apply to a SPME algorithm in which the self-forces are subtracted.

5. NUMERICAL RESULTS

We choose the same test system as in ref 9. In the smallest setup, it consists in $N = 100$ particles randomly distributed within a cubic box of length $L = 10$, half of them carry a positive, the other half a negative unit charge. Forces are measured in units of $\mathcal{C}^2/\mathcal{L}^2$, where the unit of charge $\mathcal{C}$ and the unit of length $\mathcal{L}$ are arbitrary ($\mathcal{C}$ and $\mathcal{L}$ could be, for example, the electronic charge and one nanometer or a solar mass and a parsec in a cosmological simulation). By appropriately choosing the unit $\mathcal{L}$, the density of the system, $n = N/L^3 = 0.1\ \mathcal{L}^{-3}$, can correspond to any desired physical density. Once the unit $\mathcal{L}$ is chosen, results for another density n' can moreover be deduced from the results for density n by simply scaling the length $\mathcal{L}$ by a factor $s = (n/n')^{1/3}$. Under the scaling $\mathcal{L} \rightarrow s\mathcal{L}$, the density is indeed multiplied by $s^{-3} = n'/n$, while the forces are multiplied by s^{-2}, the Ewald parameter α is multiplied by s^{-1}, and the mesh spacing h and the real-space cutoff r_c are multiplied by s. The numerical results presented here for the test system exhibit therefore the accuracy of the SPME method for homogeneous (and uncorrelated) systems of arbitrary density. Since the charge valencies $\{q_i\}$ appear moreover only via the simple factor $\sum_i q_i^2$ in the error estimate, eq 4.6, the results can be converted straightforwardly to a system with other charge valencies. An example showing how to predict the accuracy of the SPME forces in a simulation of water from the accuracy curves of the test system will be given below (Section 5.3.3).

By extensivity, identical results would be obtained in larger systems, for instance $N = 100\,000$ particles in a box of length $L = 100$, provided the size of the mesh is increased at the same time proportionally (here 10 times). No new effect is expected indeed to appear when increasing the box size because all wavelengths are uncoupled (the sole characteristic length in this test system is the mean interparticle distance). We checked this point explicitly by performing calculations in larger simulation boxes (all at density $0.1\ \mathcal{L}^{-3}$), in particular $N = 800$ particles in a box $L = 20$ and $N = 12\,500$ in a box $L = 50$. We found that the small setup ($L = 10$) has already reached the thermodynamic limit, as far as the accuracy of forces is concerned, for almost all values of the Ewald splitting parameter α, except for very small ones; a slight system size dependence is observed when $L < 20$ and α is in region $r_c > 13$ in Figure 1. To show curves corresponding to the infinitely extended system even in that region, we opted to use a box with $N = 800$ particles and $L = 20$.

5.1. Predicted versus Actual Errors.

We demonstrate the accuracy of the error estimate, eq 4.6, by comparing predicted errors versus actual errors measured on the test system with $L = 20$ defined previously. Exact reference forces are computed by a well converged standard Ewald sum. For better statistics, the measured errors were calculated by averaging the rms error obtained for 10 different random configurations of the charges. We recall that, in our implementation of SPME, the self-forces are subtracted within the particle mesh calculation. We varied the mesh size from 16^3 to 128^3, the real-space cutoff from r_c = 1 to 13 and the spline order P from 2 to 7.

As displayed in Figures 1 and 2, we observe excellent agreement between the actual errors and those predicted by eqs 4.5 (real-space error estimate) and 4.6–4.11 (reciprocal-space error estimate). The curves in Figure 1 hold for a B-spline interpolation order $P = 4$, a value that is often a good compromise between accuracy and speed of calculation. Figure 2 shows, in the case of a mesh of size $M = 16$, how the accuracy depends on order P. Increasing P improves of course the accuracy, but this comes at an increased cost for the steps of mapping charges onto the mesh and back-interpolating results from the mesh; these operations scale indeed as $O(P^3)$ with respect to the spline order.

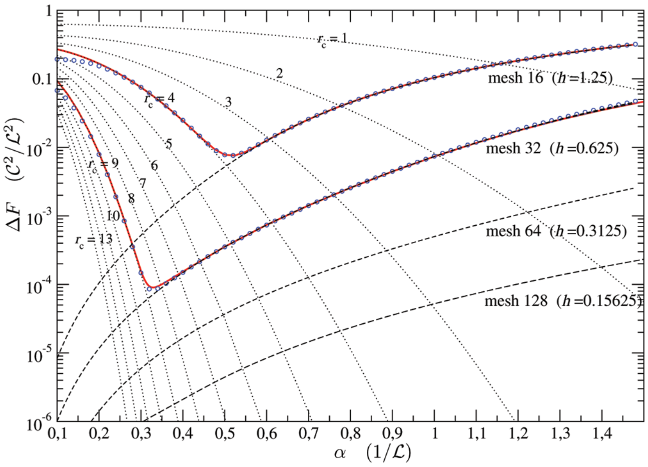

Figure 1. Actual and estimated errors of the SPME method as function the Ewald splitting parameter α. The system has a uniform density equal to 0.1 charged particles per $\mathcal{L}^3$ (800 particles in box $L = 20$). Dotted lines are the estimated errors for the direct part of the Ewald summation, for real-space cutoffs (r_c) varying from 1 to 13 (in units of $\mathcal{L}$). Dashed lines are the error estimate for the reciprocal-space part of the SPME method with self-forces subtracted. The B-spline interpolation order is set to 4, and various mesh sizes are considered: 16^3, 32^3, 64^3, and 128^3. Red lines show the estimate for the total error for two choices of parameters: mesh size 16^3 with $r_c = 4$ and mesh size 32^3 with $r_c = 9$. Actual measured errors for these two sets of parameters are shown as circles.

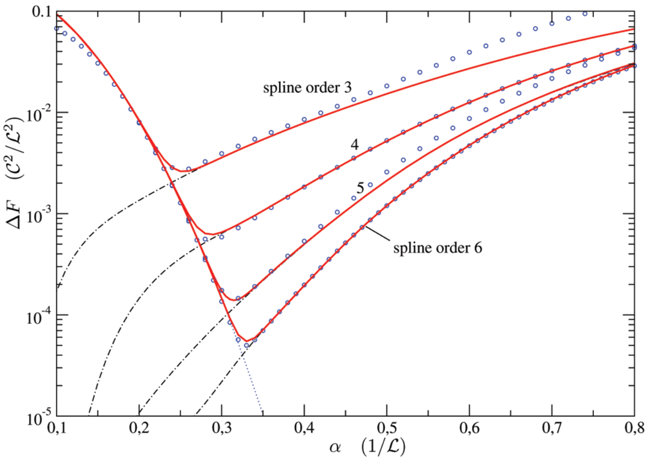

Figure 2. Actual and estimated errors of the SPME method as function of α. The system is the same as in Figure 1. The cutoff in real space is set to $r_c = 9\ \mathcal{L}$ and the mesh size is 16^3 (lattice spacing $h = L/M = 1.25\ \mathcal{L}$). The B-spline order is varied from 3 to 6. Dot-dashed lines show the error estimate for the reciprocal-space part of the SPME calculation. Red lines are the estimated total errors. Measured errors are shown as circles for interpolation orders 3–6.

Since the analytical error estimate is built by uniformly averaging the squared error of the pair interaction over all possible positions of the two particles within the simulation cell, this perfect agreement when comparing to uncorrelated systems (charges located at random) is expected by construction. Real systems have of course correlations, but we will show in Section 5.3, on several examples, that the uncorrelated error estimate still predicts reasonably well the accuracy in "real" systems with correlations. The uniform uncorrelated test system, for which the accuracy is fully understood and under control, serves thus as a useful generic guide on the accuracy. Since it contains no other characteristic length scale apart from the mean interparticle distance, that system exhibits conveniently how the accuracy of a method to compute long-range forces depends intrinsically on its parameters.

Figure 1 displays how the accuracy increases with respect to the number M of mesh points per direction and with respect to the magnitude of the real-space cutoff. At small values of the Ewald splitting parameter α, errors in the direct space part of the calculation dominate, while reciprocal-space errors dominate when α is large. For a given mesh size M and a given cutoff r_c, the parameter α affects only the accuracy and not the computational cost of the calculation. The value of α must therefore be chosen to be at, or at least close to, the point where the error reaches a minimum. The location of that point of highest accuracy is given in good approximation by the condition that the real- and reciprocal-space errors have the same magnitude: $\Delta F^{(r)} = \Delta F^{(k)}$. That equation, which defines the optimal value of α for a given M and r_c, can easily be solved numerically. From Figure 1, we see that to reach an accuracy of, say, 10^{-4} (in units of $\mathcal{C}^2/\mathcal{L}^2$), one can use for example a mesh of size 32^3 in association with a real-space cutoff $r_c = 9$ (and $\alpha \simeq 0.32$) or a mesh 64^3 with a real-space cutoff $r_c = 5$ (and $\alpha \simeq 0.58$). The smaller real-space cutoff makes the direct space part of the calculation faster, but the increased size of the mesh makes the reciprocal space part of the calculation slower. The best choice between the previous two sets of parameters is naturally the one that takes, overall, the least computational time. Note that if particles interact not only via Coulomb's law (or the force of gravity) but also via other short-range forces, it might be necessary to keep the real-space cutoff r_c above a minimum value to ensure that these other short-range forces are computed within the required accuracy. This constraint on the cutoff may be relaxed in some cases since short-range forces that derive from a central potential can be computed using a P3M computation with a Green function adjusted to get the right force.

Figures 1 and 2 demonstrate that the error estimate is accurate for most possible values of the parameters (r_c, M, P, α) of the SPME method, and specifically it is very accurate near the parameter set yielding minimal error. Such plots, or the formulas 4.5 and 4.6 themselves, can be used in association with timings measurements to determine optimal working parameters for the SPME method, i.e., parameters that yield a target accuracy at a minimal computational cost. The open-source soft-matter simulation package ESPResSo[32] contains such an automatic tuning routine for the parameters of a P3M method.

5.2. SPME with Optimized Lattice Green Function.

The FFT Poisson solver is not fully optimized in the SPME method with analytical differentiation, since the SPME lattice Green function is determined by the exponential Euler spline interpolation employed in the charge mapping onto the mesh, whereas this function (which may be tabulated in the algorithm) can actually be adjusted to improve the accuracy of the method, as done in the P3M approach. We recall that if P3M is implemented with analytical differentiation (P3M-AD algorithm), then the SPME and P3M-AD methods differ, at a computational level, only by the choice of the lattice Green function. As shown in Section 4, the lattice Green function in a SPME algorithm can be changed to the error-minimizing lattice Green function of P3M-AD by setting the tabulated array $|b(\boldsymbol{k})|^2$ to expression 4.1 instead of its usual SPME expression.

Figure 3 shows that modifying the lattice Green function to use the P3M-AD optimal lattice Green function improves the accuracy only marginally. This may be traced back to the smallness of the aliasing ($\boldsymbol{m} \neq 0$) terms in the lattice Green functions. Neglecting entirely all aliasing terms, the SPME and P3M lattice Green functions both reduce to the same expression, namely the influence function 2.22 which contains a sharpening factor $1/\hat{U}^2(\boldsymbol{k}_n)$ that compensates for interpolation smoothing. Equation 2.22 may be called the $\boldsymbol{m} = 0$ approximation for the lattice Green function, though we stress that it can be derived trivially without any reference to the aliasing theorem (see reasoning before eq 2.22). If one uses eq 2.22 as the lattice Green function, the resulting accuracy is close to those of the SPME and P3M-AD methods (in-between the two curves or slightly less accurate than SPME depending on the value of α).

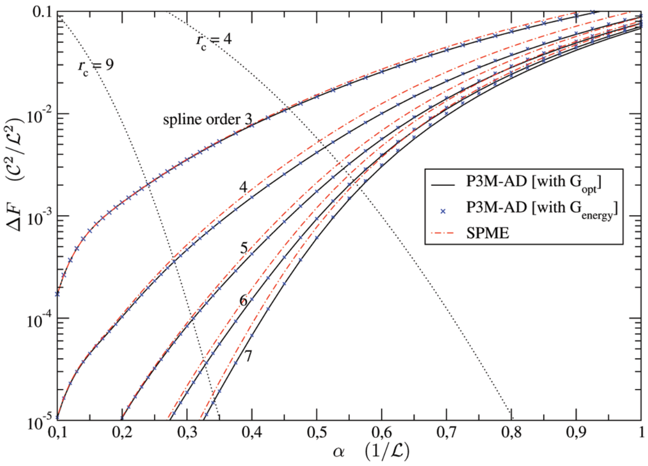

Figure 3. Comparison of reciprocal-space rms errors forces as function of α for three mesh methods: standard SPME (dashed lines), P3M-AD [or equivalently SPME with optimized lattice Green function] (solid lines), and P3M-AD with the closed form approximation 2.25 as the lattice Green function (crosses). Self-forces are subtracted within the particle mesh calculations. The system is the same as in Figure 2. The mesh size is 16^3 (lattice spacing $h = 1.25\ \mathcal{L}$), and the B-spline order is varied from 3 to 7. The real-space error estimate is shown for $r_c = 9$ and 4 $\mathcal{L}$ (dotted lines).

Interestingly, the lattice Green function optimized for the energy, eq 2.25, provides essentially the same accuracy as the true optimal P3M-AD lattice Green function, as evidenced in Figure 3. This is a useful result because eq 2.25 is known in closed analytical form and can be computed on-the-fly in simulations. The small deviations at high values of α are due to truncation errors; these unimportant deviations can be removed by using the full expression for the P3M lattice Green function optimized for the energy, i.e., keeping the aliasing sum in the numerator of eq 2.25. The fact that P3M-AD is very slightly more accurate than SPME demonstrates that the aliasing sum that optimizes the accuracy is indeed the one that appears in eq 2.25 $[(\sum_m \hat{U}^2(\boldsymbol{k}_{n+\underline{\mathbf{N}}m}))^2/\hat{U}^2(\boldsymbol{k}_n)]$ and not the one $[(\sum_m \hat{U}(\boldsymbol{k}_{n+\underline{\mathbf{N}}m}))^2]$ that appears in SPME.

From the practical point of view, the fact that the SPME lattice Green function is very close to the true optimal lattice Green function confirms the effectiveness of the SPME approach in the standard case where the charge density is discretized on a single mesh. The accuracy gain provided by the true optimal lattice Green function increases when using coarser meshes. If changing the lattice Green function in a standard SPME algorithm may seem unnecessary, it can be expected that it can provide substantial improvement in the accuracy in the case of the interlaced (also known as staggered)

SPME algorithm,[30] since interlaced P3M-AD can be more than 10 times more accurate than interlaced SPME.[33]

5.3. Applications to Correlated Systems. The a priori error estimate introduced in Section 4 is expected to predict only approximately the accuracy in correlated systems, since it assumes all interionic distances to be equiprobable [recall the uniform weighting in definition, eq 4.7, of the mean-square error Q], whereas in reality some distances are more likely than others because of correlations. We observed, however, on several examples of real systems that the uncorrelated error estimate still predicts reasonably well the error even in systems with strong correlations, in agreement with the findings of ref 9. We present comparisons of predicted versus measured errors for three different correlated condensed matter systems: an ionic melt, an ionic crystal, and a liquid water system.

5.3.1. System 1: Ionic Melt. We consider the same NaCl melt as in ref 34. The system consists of a cubic box of size L = 25.557 Å containing 512 particles (256 Na^+ and 256 Cl^- ions) that interact via the Fumi−Tosi potential:

$$v_{ij} = A_{ij}\exp(B(\sigma_{ij} - r)) - \frac{C_{ij}}{r^6} - \frac{D_{ij}}{r^8} + \frac{q_i q_j}{r} \tag{5.1}$$

Parameters A_{ij}, B, σ_{ij}, C_{ij}, and D_{ij} are set to the values given in Table 1 of ref 34. Short-range forces are truncated at r_c = 9 Å, while the long-range Coulomb forces are computed with the P3M method. A short simulation of duration 100 ps was run, at temperature fixed to 1078 K using a Langevin thermostat, to get 10 independent representative configurations of the system. Exact reference forces for these 10 configurations were computed using a well-converged Ewald summation.

Figure 4a shows that the actual rms errors of the Coulombic forces, computed using the SPME method for the 10 configurations, are somewhat smaller (by a factor between 1 and 2) than the errors expected from the error estimate for a random system. The presence of correlations in this system reduces hence the rms errors with respect to an uncorrelated system. This can be understood as arising from the screening of the electric field by the counterion atmosphere: The field (and hence the forces) decays faster than in the uncorrelated system where no such screening occurs. It can moreover be seen that the optimal value of α predicted by the error estimate (for a given mesh size and spline interpolation order) corresponds also essentially to the optimal value for the ionic melt system. As the actual accuracy is higher than the expected accuracy, it is safe, for this particular system, to use the uncorrelated error estimate when tuning the parameters of the SPME method.

As pointed out earlier, Figure 1 contains results on the accuracy of forces in systems of arbitrary density. The density of the test system (0.1 charged particle per $\mathcal{L}^3$) can be made equal to the density of the ionic melt system (0.03 charged particle per Å^3) by setting the unit of length to $\mathcal{L}$ = 1.48 Å. Since the employed mesh spacing is h = 25.557/16 ≃ 1.6 Å = 1.08 $\mathcal{L}$, the expected accuracy of the mesh calculation in the ionic melt system (solid line in Figure 4a) lies in between the curves for mesh spacing h = 1.25 and 0.625 in Figure 1. One may take also the opposite path, i.e., convert the accuracy curves from Figure 1 into accuracy curves for Figure 4a, which implies switching to units e^2/Å^2 and adjusting the density to that of the ionic melt. This is achieved by setting $\mathcal{L}$ = 1 Å and by converting the curves in Figure 1 via the scaling $\mathcal{L} \to s\mathcal{L}$ with s = 1.48, as explained at the beginning of Section 5.

5.3.2. System 2: NaCl Crystal. We consider a NaCl crystal, which is a system where the charges are very strongly correlated. We take again the same parameters for the simulation box and for the Fumi−Tosi interactions as in ref 34. The crystal is made up of 512 particles in a box of size 22.728 Å, which is equivalent to 4 × 4 × 4 units cells with lattice parameter 5.682 Å (the equilibrium value at ambient pressure). Exact reference forces were computed for 10 independent configurations extracted from a 100 ps long simulation of the crystal at 300 K.

Figure 4b shows the comparison between the predicted and the actual errors for SPME forces computed with a mesh of size 16^3. One observes again that the mesh calculation provides more accurate results than expected on the basis of a random system, at least for the region of α values of interest. This lessening of the rms errors in the correlated system is due to partial cancellations of the forces (at zero temperature, all forces would vanish exactly by symmetry).

5.3.3. System 3: SPC/E Water. Since aqueous simulation boxes are ubiquitous in chemical physics and biophysical simulations, we check how the error estimate performs in the case of a water simulation box. We take the water box provided with the GROMACS molecular dynamics simulation package.[11] It contains,

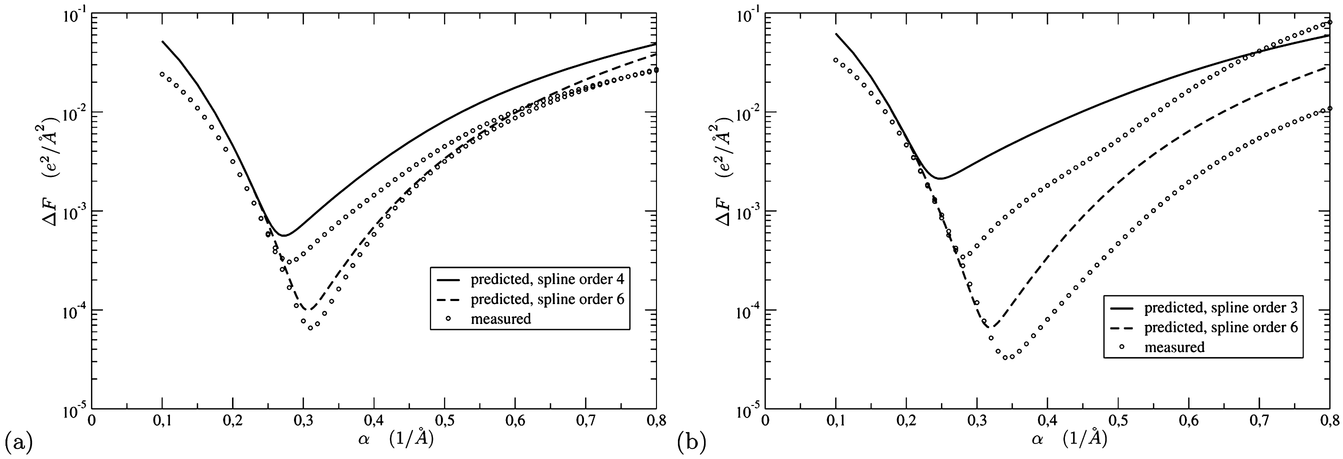

Figure 4. Predicted (solid and dashed lines) and measured (circles) rms accuracy of Coulombic forces as function of α in a simulation box containing 256 Na^+ and 256 Cl^- ions for two different state points: (a) a liquid state (temperature T = 1078 K, box size L = 25.557 Å) and (b) a crystalline state (T = 300 K, L = 22.728 Å). The SPME method is used with a real-space cutoff r_c = 9 Å and a mesh size M = 16^3; the used values of the B-spline interpolation order are indicated in the legend box.

in a box of size L = 18.6206 Å, 216 water molecules defined by the SPC/E water model.[35] There are therefore in total 648 charged sites in the box: 216 negative sites (O) with charge −0.8476 e and 432 positive sites (H) with charge 0.4238 e. In this system, the charges are strongly correlated within the molecules (rigid model) and between molecules via the hydrogen-bond network.

The considered water system has a density of 0.1 charged particle/Å^3. Setting the unit of length $\mathcal{L}$ in Figures 1 and 2 to 1 Å, the plots exhibit directly the expected accuracy [in units e^2/Å^2 = 1.389 × 10^4 kJ/(mol·nm)] for a system with that density. The parameter file provided with GROMACS for this water simulation box specifies, either directly or indirectly, the use of the following SPME parameters: real-space cutoff r_c = 9 Å, mesh size M = 16^3, interpolation order P = 4, and Ewald parameter α = 0.347 Å. The employed lattice spacing is therefore $h \simeq$ 1.16 Å. Since the lattice spacing used in Figure 2 is slightly coarser (1.25 Å), the reciprocal-space errors will be slightly smaller in the water simulation than in Figure 2.

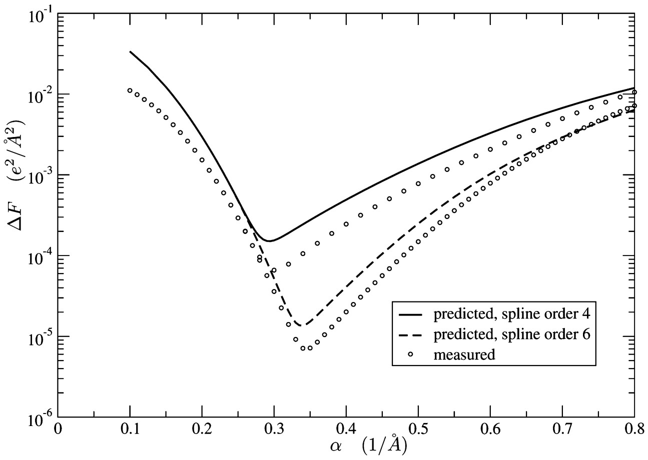

Figure 5. Predicted (solid and dashed lines) and measured (circles) rms accuracy of the forces as function of α, in a simulation box of size L = 18.6206 Å containing 216 SPC water molecules. The parameters for the SPME method are: real-space cutoff r_c = 9 Å, mesh size M = 16^3, and two values are considered for the B-spline interpolation order: P = 4 and 6.

The predicted and measured errors for the water simulation box are shown in Figure 5. In that figure, the predicted accuracies have been recalculated using the error estimate, eq 4.4 , in the case of the water simulation box of size L = 18.6206 Å. Compared to the rms errors in Figure 2, the reciprocal-space errors are reduced by a factor 0.359 due to the different charge valencies factor ($\sum_i q_i^2$):

$$\frac{216\cdot(-0.8476^2 + 2\cdot 0.4238^2)}{648} = \frac{232.8}{648} \simeq 0.359 \qquad (5.2)$$

One can also observe a further reduction of the reciprocal-space errors by about 25% due to the use of a slightly smaller mesh spacing. The error estimate, valid in principle only for uniform and uncorrelated systems, predicts the actual accuracy of the Coulombic forces in the water system within a factor of about 3.[17] As the errors are overestimated, the error estimate can be used safely in that system. It is not surprising to see a decrease of the rms errors because charges are associated into neutral molecules, which leads to partial cancellations between the forces acting on the charges (at large distances, the force between two molecules reduces to a dipole−dipole interaction).

The optimal values of α predicted by the error estimate ($\alpha_{\text{opt}} \simeq$ 0.29 in the case P = 4) are also the optimal values for the water system. Figure 5 shows incidentally that value α = 0.347 Å specified in the GROMACS parameter file is slightly too large when used with the order P = 4: The rms error of the forces would be reduced by almost a factor of 2 if the optimal value of α were used. In the case P = 6, the specified value is optimal.

The previous three examples show that correlations in real homogeneous systems tend to reduce the rms errors, with respect to those expected for an uncorrelated system, because of screening effects. If this seems to be the general tendency, it cannot be excluded that correlations can lead to a build up of errors in some specific systems, especially if the system is both strongly correlated and strongly inhomogeneous. When using the error estimate, together with timings measurements, to find an efficient set of particle mesh parameters for a given "real" system, one has to keep in mind that the real accuracy might differ from the prescribed accuracy (which is guaranteed to be satisfied only in the absence of correlations). It is therefore recommended to perform once (or periodically if correlations build up during a simulation) an explicit check of the real accuracy. If the real accuracy turns out to be higher than the expected accuracy, as in the previous examples, the parameters are safe to use. In the opposite case, other parameters must be found, for instance by specifying a higher target accuracy in the tuning routines based on the error estimate.

6. CONCLUSIONS

We exploited the theoretical link between the SPME and P3M-AD methods to demonstrate that the P3M-AD error estimate can be adapted straightforwardly to predict the accuracy of the SPME method as well. On the theoretical side, the existence of the link between these two methods, embodied in expression 3.7 of the SPME lattice Green function, allows to understand the origin of errors in the SPME method using the same powerful concepts and tools (Fourier analysis, sampling theorem, harmonics aliasing) introduced in the error analysis and optimization of the P3M method.

We demonstrated that the a priori error estimate [eqs 4.6 and 4.7] is pinpoint accurate for random uncorrelated systems. It is valid for any values of the SPME parameters, including odd B-spline interpolations orders, contrary to the error estimate derived in ref 17. Our approach avoids also the expensive calculation of the contribution of the self-forces to the rms errors, since we subtract the self-forces within the particle mesh algorithm using the analytical formula derived in ref 10.

The plots in Figures 1−3 can be used to check the accuracy of a SPME calculation and to find the optimal value of the Ewald splitting parameter, for almost any values of the parameters. The figures cover a large range of real-space cutoffs (r_c from 1 to 13) of mesh spacings (from 0.156 to 1.25), and the B-spline order is varied from 2 to 6. A simple scaling of the values can be introduced to deduce from the provided plots the accuracy of the SPME forces in a system of arbitrary density, containing charges with arbitrary valencies. An example of such a scaling is given to predict the accuracy of the forces in a box containing water molecules at normal conditions. In agreement with previous findings, the error estimate is found to predict quite well the accuracy, even though the charges in this system are correlated. The main use of the a priori error estimate is to allow a simulation program to determine automatically the best set of parameters to achieve a target accuracy at a minimal computational cost. Optimal parameters determined in this way depend in general on implementation details of the algorithm and on the machine architecture over which the program is running. An example of an automatic tuning routine for the parameters of the P3M method can be found in the open-source soft-matter simulation package ESPResSo.[32]

We showed moreover that by changing the lattice Green function in a SPME algorithm in favor of the optimized lattice Green function of P3M (which can be either tabulated or computed on-the-fly thanks to a closed form approximation that does not lead to any accuracy loss), the accuracy of the algorithm can be improved marginally. If this change might seem unnecessary for the standard SPME algorithm, it can be expected that it can provide a substantial improvement in accuracy in the case of the interlaced (also known as staggered) SPME algorithm.[30] Our derived error estimate for SPME can moreover be generalized easily to the case of the interlaced SPME method, since the effect of interlacing can be straightforwardly introduced into the a priori error estimate 4.7 (see refs 1 and 33). A detailed analysis of the accuracy and performance of the interlaced SPME and interlaced P3M algorithms will be the subject of a future work.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: vincent.ballenegger@univ-fcomte.fr

Notes

The authors declare no competing financial interest.

■ REFERENCES

(1) Hockney, R. W.; Eastwood, J. W. *Computer Simulation Using Particles*; IOP: London, U.K., 1988.

(2) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. A smooth particle mesh Ewald method. *J. Chem. Phys.* **1995**, *103*, 8577.

(3) Sagui, C.; Darden, T. A. Molecular dynamics simulations of biomolecules: Long-range electrostatic effects. *Annu. Rev. Biophys. Biomol. Struct.* **1999**, *28*, 155−179.

(4) Arnold, A.; Holm, C. In Efficient methods to compute long range interactions for soft matter systems. *Advanced Computer Simulation Approaches for Soft Matter Sciences II*; Holm, C., Kremer, K., Eds.; Advances in Polymer Sciences; Springer: Berlin, Germany, 2005; Vol. *II*; pp 59−109.

(5) Ewald, P. P. Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Ann. Phys.* **1921**, *369*, 253−287.

(6) Eastwood, J. W. Optimal particle-mesh algorithms. *J. Comput. Phys.* **1975**, *18*, 1−20.

(7) Langdon, A. B. "Energy-conserving" plasma simulation algorithms. *J. Comput. Phys.* **1973**, *12*, 247−268.

(8) Ferrel, R.; Bertschinger, E. Particle-mesh methods on the connection machine. *Int. J. Mod. Phys. C* **1994**, *5*, 933.

(9) Deserno, M.; Holm, C. How to mesh up Ewald sums. I. A theoretical and numerical comparison of various particle mesh routines. *J. Chem. Phys.* **1998**, *109*, 7678.

(10) Ballenegger, V.; Cerda, J. J.; Holm, C. Removal of spurious self-interactions in particle-mesh methods. *Comput. Phys. Commun.* **2011**, *182*, 1919−1923.

(11) Lindahl, E.; Hess, B.; van der Spoel, D. Particle mesh Ewald: An N log(N) method for Ewald sums in large systems. *J. Mol. Mod.* **2001**, *7*, 306−317.

(12) Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. The amber biomolecular simulation programs. *J. Computat. Chem.* **2005**, *26*, 1668−1688.

(13) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. Scalable molecular dynamics with NAMD. *J. Comput. Chem.* **2005**, *26*, 1781−1802.

(14) Eastwood, J. W.; Hockney, R. W.; Lawrence, D. N. P3M3DP−the three-dimensional periodic particle-particle/ particle-mesh program. *Comput. Phys. Commun.* **1980**, *19*, 215−261.

(15) Lewis, H. R. Energy-conserving numerical approximations for Vlasov plasmas. *J. Comput. Phys.* **1970**, *6*, 136.

(16) Stern, H. A.; Calkins, K. G. On mesh-based Ewald methods: Optimal parameters for two differentiation schemes. *J. Chem. Phys.* **2008**, *128*, 214106.

(17) Wang, H.; Dommert, F.; Holm, C. Optimizing working parameters of the smooth particle mesh Ewald algorithm in terms of accuracy and efficiency. *J. Chem. Phys.* **2010**, *133*, 034117.

(18) de Leeuw, S. W.; Perram, J. W.; Smith, E. R. Simulation of electrostatic systems in periodic boundary conditions. I. lattice sums and dielectric constants. *Proc. R. Soc. Lond., A* **1980**, *373*, 27−56.

(19) Darden, T. A.; Toukmaji, A.; Pedersen, L. G. Long-range electrostatic effects in biomolecular simulations. *J. Chim. Phys.* **1997**, *94*, 1346−1364.

(20) Frenkel, D.; Smit, B. *Understanding Molecular Simulation*, 2nd ed.; Academic Press: San Diego, CA, 2002.

(21) Eastwood, J. W. Optimal P3M Algorithms For Molecular Dynamics Simulations. In *Computational methods in classical and quantum physics*; Hooper, M., Ed.; Advance Publications Limited: Dublin, Ireland, 1976; pp 206−228.

(22) Luty, B. A.; Tironi, I. G.; van Gunsteren, W. F. Lattice-sum methods for calculating electrostatic interactions in molecular simulations. *J. Chem. Phys.* **1995**, *103*, 3014−21.

(23) Pollock, E. L.; Glosli, J. Comments on P3M, FMM, and the Ewald method for large periodic Coulombic systems. *Comput. Phys. Commun.* **1996**, *95*, 93−110.

(24) Huenenberger, P. H. Optimal charger-shaping functions for the particle-particle-particle-mesh (P3M) method for computing electrostatic interactions in molecular simulations. *J. Chem. Phys.* **2000**, *113*, 10464−10476.

(25) Perram, J. W.; Petersen, H. G.; de Leeuw, S. W. An algorithm for the simulation of condensed matter which grows as the 3/2 power of the number of particles. *Mol. Phys.* **1988**, *65*, 875−893.

(26) Darden, T.; York, D.; Pedersen, L. Particle mesh Ewald - an N.log(N) method for Ewald sums in large systems. *J. Chem. Phys.* **1993**, *98*, 10089.

(27) Ballenegger, V.; Cerdà, J. J.; Lenz, O.; Holm, C. The optimal P3M algorithm for computing electrostatic energies in periodic systems. *J. Chem. Phys.* **2008**, *128*, 034109.

(28) Sagui, C.; Darden, T. A. *P3M and PME: A comparison of the two methods. In Simulation and Theory of Electrostatic Interactions in Solution;* Pratt, L. R., Hummer, G., Eds.; American Institute of Physics: Melville, NY, 1999; ISBN 1-56396-906-8.

(29) Kolafa, J.; Perram, J. W. Cutoff errors in the Ewald summation formulae for point charge systems. *Mol. Simul.* **1992**, *9*, 351−368.

(30) Cerutti, D.; Duke, R. E.; Darden, T. A.; Lybrand, T. P. Staggered mesh Ewald: An extension of the smooth particle-mesh Ewald method adding great versatility. *J. Chem. Theory Comput.* **2009**, *5*, 2322−2338.

(31) Cerdà, J. J.; Ballenegger, V.; Lenz, O.; Holm, C. P3M algorithm for dipolar interactions. *J. Chem. Phys.* **2008**, *129*, 234104.

(32) Limbach, H. J.; Arnold, A.; Mann, B. A.; Holm, C. ESPResSo − an extensible simulation package for research on soft matter systems. *Comput. Phys. Commun.* **2006**, *174*, 704−727.

(33) Neelov, A.; Holm, C. Interlaced P3M algorithm with analytical and ik-differentiation. *J. Chem. Phys.* **2010**, *132*, 234103.

(34) Anwar, J.; Frenkel, D.; Noro, M. G. Calculation of the melting point of NaCl by molecular simulation. *J. Chem. Phys.* **203**, *118*, 728−735.

(35) Berendsen, H. J. C.; Grigera, J. R.; Straatsma, T. P. The missing term in effective pair potentials. *J. Phys. Chem.* **1987**, *91*, 6269−6271.