Towards an Accurate Representation of Electrostatics in Classical Force Fields: Efficient Implementation of Multipolar Interactions in Biomolecular Simulations

ARTICLE *in* THE JOURNAL OF CHEMICAL PHYSICS · FEBRUARY 2004

Impact Factor: 2.95 · DOI: 10.1063/1.1630791 · Source: PubMed

3 AUTHORS, INCLUDING:

Celeste Sagui
North Carolina State University
104 PUBLICATIONS **2,152** CITATIONS
SEE PROFILE

Thomas Darden
OpenEye Scientific Software
145 PUBLICATIONS **6,063** CITATIONS
SEE PROFILE

Towards an accurate representation of electrostatics in classical force fields: Efficient implementation of multipolar interactions in biomolecular simulations

Celeste Sagui
Department of Physics, North Carolina State University, Raleigh, North Carolina 27695

Lee G. Pedersen
Chemistry Department, UNC-Chapel Hill, Chapel Hill, North Carolina 27599-3290 and Laboratory of Structural Biology, National Institute of Environmental Health Sciences, Research Triangle Park, North Carolina 27709

Thomas A. Darden
Laboratory of Structural Biology, National Institute of Environmental Health Sciences, Research Triangle Park, North Carolina 27709

(Received 27 August 2003; accepted 9 October 2003)

The accurate simulation of biologically active macromolecules faces serious limitations that originate in the treatment of electrostatics in the empirical force fields. The current use of ''partial charges'' is a significant source of errors, since these vary widely with different conformations. By contrast, the molecular electrostatic potential (MEP) obtained through the use of a *distributed* multipole moment description, has been shown to converge to the quantum MEP outside the van der Waals surface, when higher order multipoles are used. However, in spite of the considerable improvement to the representation of the electronic cloud, higher order multipoles are not part of current classical biomolecular force fields due to the excessive computational cost. In this paper we present an efficient formalism for the treatment of higher order multipoles in Cartesian tensor formalism. The Ewald ''direct sum'' is evaluated through a McMurchie–Davidson formalism [L. McMurchie and E. Davidson, J. Comput. Phys. **26**, 218 (1978)]. The ''reciprocal sum'' has been implemented in three different ways: using an Ewald scheme, a particle mesh Ewald (PME) method, and a multigrid-based approach. We find that even though the use of the McMurchie–Davidson formalism considerably reduces the cost of the calculation with respect to the standard matrix implementation of multipole interactions, the calculation in direct space remains expensive. When most of the calculation is moved to reciprocal space via the PME method, the cost of a calculation where all multipolar interactions (up to hexadecapole–hexadecapole) are included is only about 8.5 times more expensive than a regular AMBER 7 [D. A. Pearlman *et al.*, Comput. Phys. Commun. **91**, 1 (1995)] implementation with only charge-charge interactions. The multigrid implementation is slower but shows very promising results for parallelization. It provides a natural way to interface with continuous, Gaussian-based electrostatics in the future. It is hoped that this new formalism will facilitate the systematic implementation of higher order multipoles in classical biomolecular force fields. © *2004 American Institute of Physics.* [DOI: 10.1063/1.1630791]

I. INTRODUCTION

Atomic interactions in classical force fields are roughly divided into *short-ranged*, *bonded* interactions and *long-ranged*, *nonbonded* interactions. Nonbonded interaction potentials include terms modeling the electrostatic, exchange-repulsion and dispersion interactions, and are largely responsible for the loss of accuracy in classical force fields.[1] A classical force field description is largely valid outside the van der Waals surface of the molecule. Electrostatic interactions have traditionally been modeled using an atom-centered point charge (''partial charge'') representation of the molecular charge density.[2] The most popular methods for extracting charges from molecular wavefunctions are based on fitting atomic charges to the molecular electrostatic potential (MEP), computed with *ab initio*, density functional theory (DFT) or semiempirical wave functions. The charge fitting procedure consists of minimizing the squared deviation between the Coulombic potential produced by the atomic charges and the MEP. These nonbond potentials are then expressed as a sum of spherically isotropic atom–atom potentials. Such representations are believed to be an important source of error in current force fields.[1]

The fit to the MEP can be improved either by adding more charge sites[3] or by including higher order multipoles at the atomic sites or bonds. Even with these improvements the fit to the MEP remains poor in regions near the atomic nuclei, where the charge densities overlap. As a consequence, the electrostatic interaction energy must be corrected for ''penetration'' effects at close range[4] (usually this error is absorbed into the exchange repulsion term); and the optimal values of the point multipoles may be poorly determined.[5–7] Nevertheless the use of off-center charges and/or higher or-

der atomic point multipoles can significantly improve the treatment of electrostatics. For instance, partial charges (monopoles) can vary significantly with conformational change.[7] A realistic physical molecular representation requires dipole moments (e.g., for the lone pairs), quadrupole moments (e.g., for the π bonds), etc. It has been shown that true convergence to the quantum MEP outside the van der Waals surface of the molecule requires the inclusion of higher order multipoles—up to hexadecapoles (i.e., fourth-order expansion of the electrostatic potential).[1,8–12] Depending on the atom and the molecular context, a good representation may need one or two orders less than the fourth order. Individual multipole moments may be less reliable since they depend on the conformational variation as well as on the basis set/method used for the quantum mechanical calculations. However, their sum is reliable and stable and can accurately represent the electron density outside the van der Waals surface.

In a multipole expansion, the electrostatic potential or energy of a charge distribution is expanded in powers of the Coulomb potential (or Green's function) $1/R=1/|\mathbf{r}-\mathbf{r}'|$, where $\mathbf{r}'$ is the position of a charge element and $\mathbf{r}$ is the point in space where the electrostatic potential is evaluated. The expansion can be carried out either in spherical harmonics or in a Taylor series in Cartesian coordinates. Both methods have advantages, and are used in diverse chemical physics and quantum chemistry programs.

There are two ways to perform a multipole expansion: the one-center multipole expansion which is taken with respect to a center in a molecule (i.e., the center of mass), or the *distributed* multipole analysis (DMA) first introduced by Stone,[1] where distributed multipole moments are assigned to several sites in the molecule (i.e., atoms and bond midpoints). Both the one-center multipole expansion and the DMA expansion have the inherent assumption that the distance R is much larger than the radius of the sphere that encloses a given charge, but since the charge density extends to infinity—although exponentially decreasing—the $1/R$ expansion converges only asymptotically. It has been shown that in a multipole expansion there is an ideal number of terms where the series can be considered as having converged, and divergence—if observed—occurs a few higher-order terms after that ideal number.

The one-center multipole expansion and the DMA expansion are not equivalent. The DMA expansion converges much faster than the one-center expansion and avoids the ''shape'' convergence problem that arises for nonspherical molecules.[13–17] The fact that the DMA expansion converges faster is also illustrated by the well-known case of a system with a discrete set of point charges or dipoles. This system is exactly represented by the distributed charges or dipoles (i.e., the first terms in a DMA expansion) but requires an infinite number of terms in the one-center multipole expansion. Jansen, Hättig *et al.*[18,19] examined the interaction energy of several small homomolecular dimers. Using Bader's topological theory of atoms in molecules (AIM),[9,20] the authors showed that the distributed electrostatic properties calculated using this formalism indeed solve the ''shape'' convergence problem. In addition, there are two ways of performing the multipole expansion: either by keeping the terms in the energy expansion up to the maximum power of $1/R$ ($1/R$ expansion) or by keeping all the terms that contain the highest rank of permanent and induced multipole moments (maximum rank expansion). Thus, in a DMA $1/R$ expansion, if one wants to consider interaction energies up to $1/R^5$, the highest order terms in the expansion include charge–hexadecapole, dipole–octupole, and quadrupole–quadrupole interactions. On the other hand, in a DMA maximum rank expansion up to hexadecapoles, the interaction energies contain terms like the hexadecapole–hexadecapole interaction term (which decays as $1/R^9$). Jansen, Hättig *et al.*[18,19] favored the maximum rank expansion for two reasons. First, unlike the case of the one-center expansion, where the distance R is well-defined (measured from the center of the molecule), in the DMA expansion there is not a single R, since it varies according to the position of the charge elements. Second, the authors showed that the $1/R$ expansion can sometimes give unphysical positive induction and dispersion energies, while the maximum rank expansion ensures their correct sign.

In the original DMA formalism of Stone the charges, multipoles, and other molecular properties are distributed solely on the basis of the relative positions of the basis functions with respect to the nuclei. The AIM charge partitioning, on the other hand, uses the concept of a *gradient vector path* which follows the gradient of the electronic density around the atoms in the molecule, thus defining a unique atomic partitioning of the charge density. As such, the method is totally independent of the basis function, and the atomic properties are uniquely determined by the total electron density. This method determines *bound* domains for each atom (for atoms sitting on the surface of the molecule, one of the atomic boundary surfaces is given by the van der Waals surface). This means that multipoles are defined as integrations over finite volumes, and formally avoid convergence problems. Recently, Popelier, Kosov *et al.*[10–12] used the $1/R$ expansion in spherical harmonics with the AIM method to show convergence of the electrostatic energy.

However, even though a higher order multipole expansion considerably improves the representation of the electronic cloud, its implementation is *not* common. The obvious reason for this is the associated cost. In previous work,[21] we implemented classical Ewald[22] and particle mesh Ewald (PME) (Refs. 23, 24) based treatments of fixed and induced point dipoles into the sander molecular dynamics module of AMBER 6 and 7.[25] During molecular dynamics (MD) the induced dipoles can be propagated along with the atomic positions either by iteration to self-consistency at each time step, or by a Car–Parrinello technique using an extended Lagrangian formalism. We have also implemented an $O(N)$ multigrid-based method for the efficient, parallel calculation of the long-range electrostatic forces in the partial charge representation.[26]

In this paper we present an efficient implementation of higher order multipoles in a cartesian tensor formalism, adopting the maximum rank expansion. The long-range electrostatic interactions are divided in two sums according to the usual Ewald scheme: the *direct* sum, which evaluates the fast-varying, particle–particle interactions, considered up to

a given cutoff in real space; and the "*reciprocal*" sum, which evaluates the smoothly varying, long-range part of the interaction. To speed up the evaluation of the direct part, we have implemented a McMurchie–Davidson formalism[27] which was originally developed for the evaluation of quantum molecular integrals over Cartesian Gaussians, with extensions due to Challacombe *et al.*[28] The reciprocal part has been implemented herein in three different ways: using an Ewald scheme, a PME-based approach and a multigrid-based approach (where the "reciprocal" part is evaluated in real space). In this paper we report results up to hexadecapoles.

The standard matrix implementation of multipole interactions up to hexadecapole–hexadecapole costs $\sim 35^2$ times more than point charge–point charge interactions. In this work we found that even though the use of the McMurchie–Davidson formalism considerably reduces the cost of the direct calculation with respect to the standard matrix implementation of multipole interactions by speeding up the assembly of the matrix elements, the calculation in direct space still remains expensive. It is often argued that the steep decay of the higher-order multipole interactions justifies the use of a cutoff. However, this is not the case: most of the cost of the interactions originates in the **direct** part, even under relatively short cutoffs. Additionally, too short of a cutoff produces serious force and energy errors. On the other hand, transferring more of the computation of the interactions to reciprocal space not only preserves accuracy but also has a moderate cost, which allows for a reduced cutoff for the direct space. In fact, we found that the use of a "regular" cutoff of 8 Å for the Coulomb summation (with the acceleration provided by the McMurchie–Davidson scheme) is approximately six times more expensive than the complete calculation for which most of the interaction is computed in reciprocal space via the PME method. In addition, the relative force error in the Coulomb summation with an 8 Å cutoff is about two orders of magnitude larger than the corresponding error in the calculation with the complete long-range interaction. Generally, by switching most of the calculation to reciprocal space in the PME method, a highly accurate calculation of interactions up to hexadecapole–hexadecapole costs only a factor of 8.5 more (for relative force errors of $\sim 5\times 10^{-4}$) than a regular AMBER implementation with only charge–charge interactions. The multigrid method is slower than the PME method, but shows promising results for parallelization, and it provides a natural way to interface with continuous, Gaussian-based electrostatics. Our implementation, therefore, should facilitate the systematic implementation of higher order multipoles in classical biomolecular force fields.

In this paper we only consider permanent multipoles. In the case of the induction and dispersion contributions to the energy, the "shape" convergence problem can also be avoided with the use of *distributed* polarizabilities or charge-density susceptibility functions. Future work will address the implementation of higher-order, distributed charge-density susceptibility functions.

II. TREATMENT OF LONG-RANGED ELECTROSTATICS INTERACTIONS

A. Ewald summation with multipolar interactions

True convergence to the quantum MEP outside the van der Waals surface of the molecule requires the inclusion of higher order multipoles, approximately up to hexadecapoles,[1,8–12] in a DMA representation. In this section we generalize the Ewald formalism to include Cartesian point multipoles up to hexadecapoles. To motivate the formalism, first note that the electrostatic potential $\phi(\mathbf{r}_1)$ at position $\mathbf{r}_1$ due to a point charge q_2 together with a set of point multipoles at position $\mathbf{r}_2$ is given by

$$\phi(\mathbf{r}_1)=(q_2-\mathbf{p}_2\cdot\nabla_1+\mathbf{Q}_2\!:\!\nabla_1\nabla_1-\mathbf{O}_2\vdots\nabla_1\nabla_1\nabla_1 + \mathbf{H}_2::\nabla_1\nabla_1\nabla_1\nabla_1)\frac{1}{|\mathbf{r}_2-\mathbf{r}_1|}, \tag{2.1}$$

where the subscript 1 or 2 on ∇ denotes differentiation at the point $\mathbf{r}_1$ or $\mathbf{r}_2$, and the symbols **p**, **Q**, **O**, and **H** denote dipole, quadrupole, octupole, and hexadecapole, respectively. The different "dot" products stand for the usual tensor contraction, i.e., $\mathbf{H}::\nabla\nabla\nabla\nabla=\Sigma_{i,j,k,l}H_{ijkl}\ (d/dx_i)(d/dx_j)\times(d/dx_k)(d/dx_l)$. For simplicity of notation, we will introduce the multipolar operator $\hat{L}_i$ by

$$\hat{L}_i=(q_i+\mathbf{p}_i\cdot\nabla_i+\mathbf{Q}_i\!:\!\nabla_i\nabla_i+\mathbf{O}_i\vdots\nabla_i\nabla_i\nabla_i+\mathbf{H}_i::\nabla_i\nabla_i\nabla_i\nabla_i). \tag{2.2}$$

Of course, since $\nabla_j=-\nabla_i$ when applied to any function that depends on $|\mathbf{r}_i-\mathbf{r}_j|$, the corresponding operator $\hat{L}_j$ becomes $\hat{L}_j=(q_j-\mathbf{p}_j\cdot\nabla_i+\mathbf{Q}_j\!:\!\nabla_i\nabla_i-\mathbf{O}_j\vdots\nabla_i\nabla_i\nabla_i+\mathbf{H}_j::\nabla_i\nabla_i\nabla_i\nabla_i)$.

Suppose there are N point charges and multipoles at positions $\mathbf{r}_1,\mathbf{r}_2,\ldots,\mathbf{r}_N$ within the unit cell and suppose $q_1+q_2+\cdots+q_N=0$. The edges of the unit cell are denoted by vectors $\mathbf{a}_\alpha,\alpha=1,2,3$, which need not be orthogonal. The conjugate reciprocal vectors $\mathbf{a}_\alpha^*$ are defined by the relations $\mathbf{a}_\alpha^*\cdot\mathbf{a}_\beta=\delta_{\alpha\beta}$ (the Kronecker delta), for $\alpha,\beta=1,2,3$. The point charge q_i at position $\mathbf{r}_i$ has fractional coordinates $s_{\alpha i},\alpha=1,2,3$ defined by $s_{\alpha i}=\mathbf{a}_\alpha^*\cdot\mathbf{r}_i$. The charges and multipoles interact with each other, and with their periodic images. Thus each and every component of a multipole set $\{q_i,\mathbf{p}_i,\mathbf{Q}_i,\mathbf{O}_i,\mathbf{H}_i\}$ at position $\mathbf{r}_i$ interacts with each and every component of another multipole set $\{q_j,\mathbf{p}_j,\mathbf{Q}_j,\mathbf{O}_j,\mathbf{H}_j\}$ at positions $\mathbf{r}_j$, $j\neq i$, as well as with their periodic images at positions $\mathbf{r}_j+n_1\mathbf{a}_1+n_2\mathbf{a}_2+n_3\mathbf{a}_3$ for all integer triples (n_1,n_2,n_3). It also interacts with its own periodic images at $\mathbf{r}_i+n_1\mathbf{a}_1+n_2\mathbf{a}_2+n_3\mathbf{a}_3$ for all such triples with n_1,n_2,n_3 not all zero. The electrostatic energy of the unit cell is then written as

$$U(\mathbf{r}_1,\ldots,\mathbf{r}_N)=\frac{1}{2}{\sum_{\mathbf{n}}}'\sum_i\sum_j\hat{L}_i\hat{L}_j\left(\frac{1}{|\mathbf{r}_i-\mathbf{r}_j+\mathbf{n}|}\right), \tag{2.3}$$

where the outer sum is over the vectors $\mathbf{n}=n_1\mathbf{a}_1+n_2\mathbf{a}_2+n_3\mathbf{a}_3$, the prime indicating that terms with $i=j$ and $\mathbf{n}=0$ are omitted.

As in the usual Ewald or particle-mesh treatment of Coulombic interactions under periodic boundary conditions, this sum is split into a short-range term which is handled in the direct sum, plus a long-range, smoothly varying term,

handled in the reciprocal sum by means of Fourier methods. In molecular systems, corrections are introduced to account for the "masked pairs," which are atom pairs $(i,j)\in\mathcal{M}$, $\mathcal{M}$ the masked list, whose nonbond interactions should not be calculated, since they are accounted for by other terms in the potential. These masked pairs are easily omitted from the direct sum. However, since all pair interactions are unavoidably included in the Fourier treatment of the reciprocal sum, the reciprocal part of the masked pair contributions must be separately subtracted. Computationally, this is conveniently achieved by skipping these pairs in the direct sum and separately adding the sum over $(i,j)\in\mathcal{M}$ of the "adjusted" interaction energy, which is the difference between the direct sum interaction and the standard Coulomb interaction. Similarly the self-energy, which is the reciprocal part of the interactions of multipole components with themselves, must be removed. The electrostatic energy can then be written as a sum of four terms, the direct, the reciprocal, the adjusted and the self terms: $U=U_{\rm dir}+U_{\rm rec}+U_{\rm adj}+U_{\rm self}$. Finally, the electrostatic field and force on atom i at position $\mathbf{r}_i$ are computed as the negative gradient of the electrostatic potential $\phi(\mathbf{r}_i)$ and electrostatic energy $U(\mathbf{r}_i)$, respectively,

$$\mathbf{E}(\mathbf{r}_i)=-\nabla_i\phi(\mathbf{r}_i), \tag{2.4}$$

$$\begin{aligned}\mathbf{F}(\mathbf{r}_i)&=-\nabla_i U(\mathbf{r}_i)\\&=-\hat{L}_i\nabla_i\phi(\mathbf{r}_i).\end{aligned} \tag{2.5}$$

1. Reciprocal part

For the reciprocal part, we generalize the equations of Smith[22] for the inclusion of Cartesian point multipoles into the Ewald formalism. The reciprocal energy $U_{\rm rec}$ is similar in form to the reciprocal sum in the Coulombic case, but requires a generalization of the structure factor $S(\mathbf{m})$ to include the multipolar interactions. Define the reciprocal lattice vectors $\mathbf{m}$ by $\mathbf{m}=m_1\mathbf{a}_1^*+m_2\mathbf{a}_2^*+m_3\mathbf{a}_3^*$ with m_1,m_2,m_3 integers not all zero. The generalized structure factor $S(\mathbf{m})$ for N particles is written as

$$S(\mathbf{m})=\sum_{j=1}^{N}\tilde{L}_j(\mathbf{m})\exp(2\pi i\mathbf{m}\cdot\mathbf{r}_j), \tag{2.6}$$

where $\exp(2\pi i\mathbf{m}\cdot\mathbf{r}_j)=\exp(2\pi i(m_1s_{1j}+m_2s_{2j}+m_3s_{3j}))$, $s_{\alpha j},\alpha=1,2,3$ are the fractional coordinates of site j, defined above, and $\tilde{L}_j(\mathbf{m})$ is the Fourier transform of the multipolar operator $\hat{L}_j$, given by

$$\begin{aligned}\tilde{L}_j(\mathbf{m})=q_j&+2\pi i\mathbf{p}_j\cdot\mathbf{m}-(2\pi)^2\mathbf{Q}_j\!:\!\mathbf{m}\ \mathbf{m}\\&-(2\pi)^3 i\mathbf{O}_j\vdots\mathbf{m}\ \mathbf{m}\ \mathbf{m}\\&+(2\pi)^4\mathbf{H}_j::\ \mathbf{m}\ \mathbf{m}\ \mathbf{m}\ \mathbf{m}.\end{aligned} \tag{2.7}$$

The reciprocal electrostatic potential at position $\mathbf{r}_i$ is given by

$$\begin{aligned}\phi_{\rm rec}(\mathbf{r}_i)=&\frac{1}{\pi V}\sum_{\mathbf{m}\neq 0}\frac{\exp(-\pi^2\mathbf{m}^2/\beta^2)}{\mathbf{m}^2}S(\mathbf{m})\\&\times\exp(-2\pi i\mathbf{m}\cdot\mathbf{r}_i),\end{aligned} \tag{2.8}$$

where $V=\mathbf{a}_1\cdot\mathbf{a}_2\times\mathbf{a}_3$ is the volume of the unit cell, and hence the reciprocal energy—with the introduction of the generalized structure factor—acquires the familiar form of

$$U_{\rm rec}=\frac{1}{2\pi V}\sum_{\mathbf{m}\neq 0}\frac{\exp(-\pi^2\mathbf{m}^2/\beta^2)}{\mathbf{m}^2}S(\mathbf{m})S(-\mathbf{m}). \tag{2.9}$$

The reciprocal contributions to the electric field $\mathbf{E}(\mathbf{r}_i)$ and force $\mathbf{F}(\mathbf{r}_i)$ are given by the real parts of

$$\begin{aligned}\mathbf{E}_{\rm rec}(\mathbf{r}_i)=&\frac{2i}{V}\sum_{\mathbf{m}\neq 0}\mathbf{m}\frac{\exp(-\pi^2\mathbf{m}^2/\beta^2)}{\mathbf{m}^2}\\&\times\exp(-2\pi i\mathbf{m}\cdot\mathbf{r}_i)S(\mathbf{m}),\end{aligned} \tag{2.10}$$

$$\begin{aligned}\mathbf{F}_{\rm rec}(\mathbf{r}_i)=&\frac{2i}{V}\sum_{\mathbf{m}\neq 0}\mathbf{m}\frac{\exp(-\pi^2\mathbf{m}^2/\beta^2)}{\mathbf{m}^2}\tilde{L}_i^*(\mathbf{m})\\&\times\exp(-2\pi i\mathbf{m}\cdot\mathbf{r}_i)S(\mathbf{m}),\end{aligned} \tag{2.11}$$

where $\tilde{L}_i^*(\mathbf{m})$ is the complex conjugate of the multipolar operator in Fourier space given in Eq. (2.7).

2. Direct and adjusted parts

The direct electrostatic potential at position $\mathbf{r}_i$ is given by

$$\phi_{\rm dir}(\mathbf{r}_i)=\sum_{\mathbf{n}}^{*}\sum_{j=1}^{N}\hat{L}_j\frac{\operatorname{erfc}(\beta|\mathbf{r}_j-\mathbf{r}_i+\mathbf{n}|)}{|\mathbf{r}_j-\mathbf{r}_i+\mathbf{n}|}, \tag{2.12}$$

where the asterisk over the sum denotes that the terms with $\mathbf{n}=0$ and either $j=i$ or $(i,j)\in\mathcal{M}$ are omitted. The direct sum energy is then given by

$$U_{\rm dir}=\frac{1}{2}\sum_{\mathbf{n}}^{*}\sum_{i,j=1}^{N}\hat{L}_i\hat{L}_j\frac{\operatorname{erfc}(\beta|\mathbf{r}_j-\mathbf{r}_i+\mathbf{n}|)}{|\mathbf{r}_j-\mathbf{r}_i+\mathbf{n}|}. \tag{2.13}$$

The adjusted part contributions are obtained by replacing $\operatorname{erfc}(\beta|\mathbf{r}_j-\mathbf{r}_i+\mathbf{n}|)$ in the above expressions by $-\operatorname{erf}(\beta|\mathbf{r}_j-\mathbf{r}_i|)$, and carrying the sum over the masked atom pairs $(i,j)\in\mathcal{M}$. In our previous treatment of dipolar interactions,[21] we used the formalism of Smith[22] by defining a generalized source function $B_0(r)=\operatorname{erfc}(\beta r)/r$ and all related higher-order functions $B_l(r)$ $l>0$ obtained through a recursion relation, $B_l(r)=[(2l-1)B_{l-1}(r)+(2\beta^2)^l\exp(-\beta^2r^2)/(\beta\sqrt{\pi})]r^{-2}$. In this work, we have opted for a McMurchie–Davidson formalism (applied also to the adjusted part), that will be introduced in the following sections.

3. Virial tensor

The stress tensor Π, necessary for constant pressure simulations using Ewald summations, is computed in the usual manner.[29] The direct part has a similar expression to that reported before.[21,24] It can be written as

$$V\Pi_{\alpha\beta}^{\rm dir}=\frac{1}{2}\sum_{\mathbf{n}}^{*}\sum_{i,j=1}^{N}(\mathbf{r}_{ji\mathbf{n}})_\alpha(\mathbf{F}_{ji\mathbf{n}}^{\rm dir})_\beta, \tag{2.14}$$

where $(\mathbf{F}_{ji\mathbf{n}}^{\rm dir})_\beta$ is the β component of the direct force between particles i and j, whose relative position is given by $\mathbf{r}_{ji\mathbf{n}}=\mathbf{r}_j-\mathbf{r}_i+\mathbf{n}$. A similar expression, with $(i,j)\in\mathcal{M}$ and $\mathbf{n}=0$ is obtained for the adjusted part. The computation of the

TABLE I. For each multipole level, this table gives the number of *Cartesian* multipole components in that level; the *cumulative* number of independent cartesian components, as explained in Sec. II B; the independent components of the field vector Ξ as given in Eq. (2.32); and the number of R^n_{tuv} functions [Eqs. (2.27) and (2.29)].

Multipole level	Independent Cartesian multip. components	Total independent components	Ξ field vector components	Number of R^n_{tuv} functions
0	1	1	4	5
1	3	4	10	35
2	6	10	20	126
3	10	20	35	330
4	15	35	56	715
5	21	56	84	1365
6	28	84	120	2380
7	36	120	165	3876
8	45	165	220	5985
9	55	220	286	8855

reciprocal tensor, on the other hand, gives new terms compared to the monopolar contribution. Following a standard procedure,[21,29] one can write the different multipolar contributions to the reciprocal virial tensor [here referred to just as $\Pi^{(l)}$, where l is the multipole order] as

$$V\Pi^{(0)}_{\alpha\beta}=\frac{1}{2\pi V}\sum_{\mathbf{m}\neq 0}\frac{\exp(-\pi^2\mathbf{m}^2/\beta^2)}{\mathbf{m}^2}S(\mathbf{m})S(-\mathbf{m})\times\left(\delta_{\alpha\beta}-2\frac{1+\pi^2\mathbf{m}^2/\beta^2}{\mathbf{m}^2}\mathbf{m}_\alpha\mathbf{m}_\beta\right) \tag{2.15}$$

for the monopolar case [where the structure factor $S(\mathbf{m})$ is independent of the vectors $\mathbf{a}_\alpha, \alpha=1,2,3$]. The contribution of the higher-order multipoles to the reciprocal virial arises from the dependence of $S(\mathbf{m})$ on the reciprocal lattice vector. The different multipolar contributions are given by

$$\begin{aligned}
V\Pi^{(1)}_{\alpha\beta}&=\sum_i\frac{\partial\phi_{\rm rec}}{\partial x_\alpha}p_\beta(i),\\
V\Pi^{(2)}_{\alpha\beta}&=\sum_i\frac{\partial^2\phi_{\rm rec}}{\partial x_\alpha\partial x_\mu}(Q_{\beta\mu}(i)+Q_{\mu\beta}(i)),\\
V\Pi^{(3)}_{\alpha\beta}&=\sum_i\frac{\partial^3\phi_{\rm rec}}{\partial x_\alpha\partial x_\mu\partial x_\nu}(O_{\beta\mu\nu}(i)+O_{\mu\beta\nu}(i)+O_{\mu\nu\beta}(i)),\\
V\Pi^{(4)}_{\alpha\beta}&=\sum_i\frac{\partial^4\phi_{\rm rec}}{\partial x_\alpha\partial x_\mu\partial x_\nu\partial x_\omega}(H_{\beta\mu\nu\omega}(i)+H_{\mu\beta\nu\omega}(i)+H_{\mu\nu\beta\omega}(i)+H_{\mu\nu\omega\beta}(i)).
\end{aligned} \tag{2.16}$$

In this expression, sum over repeated indices μ, ν and ω is assumed, where each index can take the value 1, 2 or 3 and, for instance, $H_{1233}=H_{xyzz}$. The total reciprocal virial is the sum of $\Pi^{(l)}, l=0,\ldots 4.$,

B. Unidimensional formulation of the multipole moments

In order to make our code efficient for the evaluation of the direct part, it was necessary to recast all the multipole moments into a long unidimensional vector $\mathbf{M}=(q,p_x,p_y,p_z,Q_{xx},Q_{yy},Q_{zz},Q_{xy},Q_{xz},Q_{yz},O_{xxx},O_{yyy},O_{zzz},O_{xxy},O_{xxz},O_{yyx},O_{yyz},\ldots)$, which is defined in the program. In addition, to facilitate the presentation of the equations, we define a unidimensional vector $\mathbf{D}$, containing partial derivatives, according to $\mathbf{D}=(1,D_x,D_y,D_z,D_{xx},D_{yy},D_{zz},D_{xy},D_{xz},D_{yz},D_{xxx},D_{yyy},D_{zzz},D_{xxy},D_{xxz},D_{yyx},\ldots)$ where, for instance, $D_{xyz}=d^3/dxdydz$. We will indicate the components of these vectors either by the standard way, i.e., the jth component of $\mathbf{M}$ is $\mathbf{M}(j)$, or by a set of three subindices $\{tuv\}$, M_{tuv}. The latter refers to the number of times each cartesian component appears. For instance, $M_{300}=\mathbf{M}(11)=O_{xxx}$, $M_{111}=\mathbf{M}(20)=O_{xyz}$, etc. The tuv notation is useful to relate to the Hermite Gaussians used in the McMurchie–Davidson recursion relation presented in the next section. The unidimensional multipolar vector is similar to the ''polytensor'' introduced by Applequist.[30,31] In the present definition, the individual components of $\mathbf{M}$ contain the sum of all the components that are related by symmetry in the original multipolar tensor. If we indicate with a prime the original multipolar tensor, this simply means that $M_{300}=\mathbf{M}(11)=O_{xxx}=O'_{xxx}$, $M_{210}=\mathbf{M}(14)=O_{xxy}=O'_{xxy}+O'_{xyx}+O'_{yxx}=3O'_{xxy}$, etc. Notice that $u+t+v$ indicates the order of the multipole.

With the use of this notation, we can re-write the direct part of the electrostatic potential (2.12) as

$$\phi_{\rm dir}(\mathbf{r}_i)=\sum_{\mathbf{n}}^{*}\sum_{j=1}^{N}(-1)^{u+t+v}M_{tuv}(j)D_{tuv}(i)\times\left[\frac{\operatorname{erfc}(\beta|\mathbf{r}_j-\mathbf{r}_i+\mathbf{n}|)}{|\mathbf{r}_j-\mathbf{r}_i+\mathbf{n}|}\right], \tag{2.17}$$

where the Einstein convention is used, i.e., summation is carried out over repeated indices $\{tuv\}$ for the allowed combinations of tuv. $M_{tuv}(j)$ is the tuv multipolar component at position $\mathbf{r}_j$, while the derivatives $D_{tuv}(i)$ are taken with respect to the cartesian components of $\mathbf{r}_i$.

The multipole order as well as the number of independent components at each level is shown in Table I. For each level l, the number of independent *Cartesian* components for the level is $\frac{1}{2}(l+1)(l+2)$. Therefore, the total number of degrees of freedom, i.e., the *cumulative* number of indepen-

dent components up to level l is $\binom{l+3}{3}$. The fourth column in the table gives the number of independent derivatives (including the zero-order derivative, i.e., the function itself) needed to compute the electrostatic field and forces at each level. The result is equal to the number of degrees of freedom in the next order multipole, as given in column three.

C. McMurchie–Davidson formalism for the treatment of the direct interactions

McMurchie and Davidson developed a scheme for the evaluation of quantum molecular integrals over Cartesian Gaussians.[27] Later, Challacombe *et al.* extended the formalism to the Cartesian multipole interaction tensor.[28] In this section, we adapt this scheme for the classical computation of the direct energy and forces in the Ewald formalism. Our treatment of the recursion relations shown below follows that presented by Helgaker *et al.*[32] This formalism combines properties of Hermite Gaussians and the Boys functions. Hermite Gaussians are defined as

$$\Lambda_{tuv}(\mathbf{r},p,\mathbf{P})=\left(\frac{\partial}{\partial P_x}\right)^t\left(\frac{\partial}{\partial P_y}\right)^u\left(\frac{\partial}{\partial P_z}\right)^v \exp(-p|\mathbf{r}-\mathbf{P}|^2). \tag{2.18}$$

For $x>0$ the Boys function of order n ($n\geqslant 0$) is defined as

$$F_n(x)=\int_0^1 \exp(-xt^2)t^{2n}dt \tag{2.19}$$

while the complementary Boys function is defined as

$$F_n^C(x)=\int_1^\infty \exp(-xt^2)t^{2n}dt. \tag{2.20}$$

These functions are related to the error function and complementary error function according to

$$\mathrm{erf}(x)=\frac{2x}{\sqrt{\pi}}F_0(x^2), \tag{2.21}$$

$$\mathrm{erfc}(x)=\frac{2x}{\sqrt{\pi}}F_0^C(x^2). \tag{2.22}$$

The Boys functions can be generated through *upward* and *downward* recursions. While the upward recursions tend to be numerically unstable at small x, the downward recursions have proved quite useful for our purposes. These are given by

$$F_n(x)=\frac{2x\ \ F_{n+1}(x)+\exp(-x)}{2n+1}, \tag{2.23}$$

$$F_n^C(x)=\frac{2x\ \ F_{n+1}^C(x)-\exp(-x)}{2n+1}. \tag{2.24}$$

One can introduce a function analogous to the Hermite Coulomb integral,[27,32]

$$R_{tuv}(r_{ij})=\frac{2\beta}{\sqrt{\pi}}\left(\frac{\partial}{\partial X}\right)^t\left(\frac{\partial}{\partial Y}\right)^u\left(\frac{\partial}{\partial Z}\right)^v F_0^C(\beta^2 r_{ij}^2), \tag{2.25}$$

where $X=x_j-x_i$, $Y=y_j-y_i$, and $Z=z_j-z_i$. Thus, the potential in Eq. (2.17) can be rewritten as

$$\phi_{\mathrm{dir}}(\mathbf{r}_i)=\sum_j M_{tuv}(j)\ \ R_{tuv}(r_{ij}), \tag{2.26}$$

where one can take advantage of the rapid decay of $R_{tuv}(r_{ij})$ with distance to restrict the sum to those j atoms within a cutoff distance r_c of i. One can also form a unidimensional vector $\mathbf{R}$ for each atomic pair distance r_{ij} whose components are R_{tuv},

$$\mathbf{R}(r_{ij})=\frac{2\beta}{\sqrt{\pi}}\mathbf{D}(j)\ \ F_0^C(\beta^2 r_{ij}^2). \tag{2.27}$$

The advantage of this formalism is that one can use recursion relations to generate all the R_{tuv} functions up to a given order $t+u+v$. Since the recursion relations to a certain order necessitate higher order derivatives, one first defines higher-order "source" functions,

$$R_{000}^n=\frac{2\beta}{\sqrt{\pi}}(-2\beta^2)^n F_n^C(\beta^2 r^2) \tag{2.28}$$

and then generates the $\{tuv\}$ terms by

$$R_{tuv}^n=\left(\frac{\partial}{\partial X}\right)^t\left(\frac{\partial}{\partial Y}\right)^u\left(\frac{\partial}{\partial Z}\right)^v R_{000}^n. \tag{2.29}$$

Finally, the recursion relations are given by[32]

$$\begin{aligned}
R_{t+1,u,v}^n &= t\ \ R_{t-1,u,v}^{n+1}+X\ \ R_{tuv}^{n+1},\\
R_{t,u+1,v}^n &= u\ \ R_{t,u-1,v}^{n+1}+Y\ \ R_{tuv}^{n+1},\\
R_{t,u,v+1}^n &= v\ \ R_{t,u,v-1}^{n+1}+Z\ \ R_{tuv}^{n+1}.
\end{aligned} \tag{2.30}$$

How many functions do we need to generate? From the recursion relations, one can work out that the number of R_{tuv}^n functions needed for a given level l must be such that $0\leqslant n+t+u+v\leqslant(2l+1)$. For instance, for dipoles ($2l+1=3$) there is 1 function with $n+t+u+v=0$ (R_{000}^0), 4 functions with $n+t+u+v=1$ (R_{100}^0, R_{010}^0, R_{001}^0 and R_{000}^1), 10 functions with $n+t+u+v=2$ (R_{200}^0, R_{020}^0, R_{002}^0, R_{110}^0, R_{101}^0, R_{011}^0, R_{100}^1, R_{010}^1, R_{001}^1 and R_{000}^2), and 20 functions with $n+t+u+v=3$: a total of 35 functions. In general, the number of R_{tuv}^n functions is

$$\sum_{j=0}^{2l+1}\binom{j+3}{3}=\binom{(2l+1)+4}{4}. \tag{2.31}$$

These numbers for the first few multipolar orders are given in Table I.

The electrostatic field $\mathbf{E}$ and electrostatic force $\mathbf{F}$ in Eqs. (2.4) and (2.5) take the following form for the direct part,

$$\mathbf{E}_{\mathrm{dir}}(\mathbf{r}_i)=-\nabla_i\ \ \phi_{\mathrm{dir}}(\mathbf{r}_i), \tag{2.32}$$

$$\begin{aligned}
\mathbf{F}_{\mathrm{dir}}(\mathbf{r}_i) &= -\nabla_i U_{\mathrm{dir}}(\mathbf{r}_i)\\
&= -[M_{tuv}(i)\ \ D_{tuv}(i)]\ \ \nabla_i\left[\sum_j M_{pqs}(j)\ \ R_{pqs}(r_{ij})\right].
\end{aligned} \tag{2.33}$$

However, the straightforward application of the derivatives is impractical, because many derivatives are repeated. For computational efficiency, one can define a unidimen-

sional, generalized field vector Ξ that contains the electrostatic potential and its various derivatives for each atom *pair*,

$$\Xi_{tuv}(i,j)=(-1)^{t+u+v}\ D_{tuv}(j)[R_{pqs}(r_{ij})\ M_{pqs}(j)], \tag{2.34}$$

where $D_{tuv}(j)$ contains derivatives with respect to $X=x_j-x_i$, $Y=y_j-y_i$ and $Z=z_j-z_i$, and $D_{tuv}(i)=(-1)^{t+u+v}\ D_{tuv}(j)$. As an example, consider two sets of charges and dipoles located at $\mathbf{r}_i$ and $\mathbf{r}_j$. Let ϕ be the electrostatic potential produced at $\mathbf{r}_i$ by the charges and dipoles at $\mathbf{r}_j$. The field vector Ξ can then be explicitly written as

$$\Xi=\left(\phi,-\frac{\partial\phi}{\partial X},-\frac{\partial\phi}{\partial Y},-\frac{\partial\phi}{\partial Z},\frac{\partial^2\phi}{\partial X^2},\frac{\partial^2\phi}{\partial Y^2},\frac{\partial^2\phi}{\partial Z^2},\frac{\partial^2\phi}{\partial X\partial Y},\frac{\partial^2\phi}{\partial X\partial Z},\frac{\partial^2\phi}{\partial Y\partial Z}\right).$$

One can use a mapping function $\mathcal{I}$ that maps the appropriate components of Ξ that are necessary to compute the energy or the forces. With the use of the generalized pair field and the mapping function, the direct electrostatic energy and force at $\mathbf{r}_i$ become

$$U_{\mathrm{dir}}=\sum_{i<j}\ \mathbf{M}(i)\cdot\mathcal{I}_{\mathrm{energy}}[\Xi(i,j)], \tag{2.35}$$

$$\mathbf{F}_{\mathrm{dir}}(\mathbf{r}_i)=-\sum_{j}\ \mathbf{M}(i)\cdot\mathcal{I}_{\mathrm{force}}[\Xi(i,j)]. \tag{2.36}$$

For the example above, $\mathcal{I}_{\mathrm{energy}}$ is such that $U_{\mathrm{dir}}=M_{000}\,\Xi(1)+M_{100}\,\Xi(2)+M_{010}\,\Xi(3)+M_{001}\,\Xi(4)=(q_i-\mathbf{p}_i\cdot\nabla_j)\phi(i)$. For the x-component of the force, $\mathcal{I}_{\mathrm{force}}$ is such that $F_{x,\mathrm{dir}}(\mathbf{r}_i)=-[M_{000}\,\Xi(2)+M_{100}\,\Xi(5)+M_{010}\,\Xi(8)+M_{001}\,\Xi(9)]=-\{[-q_i\nabla_j\phi(i)]_x-[(\mathbf{p}_i\cdot\nabla_j)\nabla_j\phi(i)]_x\}=-[q_i\nabla_i\phi(i)-\mathbf{p}_i\cdot\nabla_i\mathbf{E}_i]_x$.

Notice that the number of components of the pair field vector Ξ at each level is exactly the number of independent components of the immediately higher multipole level, and is given in the fourth column of Table I. Addition over all atoms j that interact with i can convert the *pair* field vector Ξ into a "potential vector" $\mathbf{\Phi}_{\mathrm{dir}}$ defined at each atomic position $\mathbf{r}_i$. Summation upon j, with $j\neq i$ in Eq. (2.34) gives

$$\mathbf{\Phi}_{\mathrm{dir}}=\sum_{j,j\neq i}\ \Xi(i,j)\equiv\mathbf{D}(\mathbf{r}_i)\ \ \phi_{\mathrm{dir}}(\mathbf{r}_i), \tag{2.37}$$

where $\mathbf{D}(\mathbf{r}_i)$ is the derivatives vector and $\phi_{\mathrm{dir}}(\mathbf{r}_i)$ is given in Eq. (2.17). In the $\mathbf{\Phi}_{\mathrm{dir}}$ vector, $\mathbf{\Phi}_{\mathrm{dir}}(1)$ is the electrostatic potential, $\mathbf{\Phi}_{\mathrm{dir}}(2)$, $\mathbf{\Phi}_{\mathrm{dir}}(3)$ $\mathbf{\Phi}_{\mathrm{dir}}(4)$ are the first derivatives of $\phi_{\mathrm{dir}}(\mathbf{r}_i)$ (or equivalently, the negative of the electric field), $\mathbf{\Phi}_{\mathrm{dir}}(5)\cdots\mathbf{\Phi}_{\mathrm{dir}}(10)$ are second derivatives of $\phi_{\mathrm{dir}}(\mathbf{r}_i)$, etc.

The formalism for the direct part presented here carries over to the adjusted part, when one replaces the complementary Boys function $F_0^C(\beta^2r_{ij}^2)$ in the Hermite Coulomb integral in Eq. (2.25), by the Boys function $F_0(\beta^2r_{ij}^2)$, and carries the sum over the masked atom pairs $(i,j)\in\mathcal{M}$.

Programming for efficiency: In order to optimize the performance of the direct part computation, we introduced a McMurchie–Davidson formalism[27] based on Hermite Coulomb integrals. This allows the use of the recursion relations in Eqs. (2.30) and the downward recursions in the Boys functions in Eqs. (2.23) to speed up the computation. One can therefore tabulate the "source" functions R_{000}^n in Eq. (2.28), and then use interpolation to evaluate these functions at the different atom-pair distances r_{ij}. The program evaluates lists of atom pairs, whose distances r_{ij} are stored, and the calculations are performed upon these distances. To optimize computation and memory requirements we found that it is better to divide the list of atom pairs into blocks, and perform the calculations over these blocks (25 pairs per block at a time seem to work best in our tests). These preprocessed lists avoid indirection problems by accessing pairs that are continuous in memory in the recursion equations. With the use of appropriate pointers, the five-index matrix whose elements are given by $R_{tuv}^n(r_{ij})$ (the indices are t, u, v, n and the index that identifies the pair distance) can be transformed into a two-dimensional matrix (with one index to represent t, u, v, and n and the other to identify the atom pair) on which the recursion relations (2.30) are applied. The resulting Hermite Coulomb integrals are then used in Eq. (2.34) to compute the pair field vector Ξ, which in turn is used to compute the energy and forces according to Eqs. (2.35) and (2.36), as well as the potential vector $\mathbf{\Phi}_{\mathrm{dir}}$.

D. PME method with multipolar interactions

The PME method approximates the electrostatic structure factors by replacing the complex exponentials appearing in Eq. (2.6) by appropriate linear combinations of their values at nearby grid points. The resulting trigonometric sums over grid points can be rapidly evaluated by the fast Fourier transform (FFT). The current version[24,33] uses approximation by cardinal B-splines $\theta_p(u)$. This leads to approximate structure factors and hence approximate reciprocal potential ϕ_{rec} and energy U_{rec} which can be analytically differentiated to give the different components of the potential field vector $\mathbf{\Phi}_{\mathrm{rec}}$. Since this method has been described in detail in the literature[23,24,33] here we just present the results as generalized for the multipolar case. Given positive integers K_1,K_2,K_3 and a point $\mathbf{r}$ in the unit cell, we denote its scaled fractional coordinates by u_1,u_2,u_3, i.e., $u_\alpha=K_\alpha\mathbf{a}_\alpha^*\cdot\mathbf{r}$, for $\alpha=1,2,3$. Due to the periodic boundary conditions, we may assume that $0\leqslant u_\alpha<K_\alpha$. Then

$$\exp(2\pi i\mathbf{m}\cdot\mathbf{r})=\exp\left(2\pi i\frac{m_1u_1}{K_1}\right)\times\exp\left(2\pi i\frac{m_2u_2}{K_2}\right)\times\exp\left(2\pi i\frac{m_3u_3}{K_3}\right). \tag{2.38}$$

The pth order Cardinal B-spline $\theta_p(u)$ is defined as follows:

$$\theta_1(u)=\begin{cases}1 & \text{if } 0\leqslant u\leqslant 1\\ 0 & \text{otherwise}\end{cases} \tag{2.39}$$

and

$$\theta_{p+1}(u)=\int_{-\infty}^{\infty}\theta_p(u-s)\,\theta_1(s)\,ds=\theta_p\star\theta_1(u). \tag{2.40}$$

The interpolation of complex exponentials is achieved by the simple Euler exponential spline. Using a spline of order p, the interpolating spline g_p can be written as

$$\exp\left(2\pi i \frac{m_i}{K_i} u_i\right) \approx g_p\left(\frac{m_i}{K_i}, u_i\right), \tag{2.41}$$

$$g_p\left(\frac{m_i}{K_i}, u_i\right) = b_i\left(\frac{m_i}{K_i}\right) \sum_{k=-\infty}^{\infty} \theta_p(u_i - k) \cdot \exp\left(2\pi i \frac{m_i}{K_i} k\right).$$

Since the Cardinal B-spline θ_p is a continuous function that is zero outside an interval of length p, the above sum actually involves only p terms. The coefficients $b_i(m_i/K_i)$ are given by

$$b_i\left(\frac{m_i}{K_i}\right) = \exp(2\pi i (p-1) m_i/K_i) \times \left[\sum_{k=0}^{p-2} \theta_p(k+1) \exp(2\pi i m_i k/K_i)\right]^{-1}. \tag{2.42}$$

Using the symmetry of the B-splines ($\theta_p(p-u) = \theta_p(u), 0 \leqslant u \leqslant p$) and the definition (2.42), one can directly show that $g_p(m_i/K_i, u_i) = \exp(2\pi i (m_i/K_i) u_i)$ for integer values of u_i. The spline can further be optimized to fit the complex exponential in the least-squares sense.[33,34] In general, a function fitted with polynomial splines of order p is $p-2$ times continuously differentiable. Therefore, for a given multipole order l, the function is $p-2-l$ times continuously differentiable.

Now we discuss the calculation of the reciprocal interactions using differentiation of the spline approximants g_p. A generalized grid multipolar array $Q^R(k_1, k_2, k_3)$ is defined to include the effect of all the multipoles,

$$Q^R(k_1, k_2, k_3) = \sum_{i=1}^{N} \sum_{n_1, n_2, n_3} \hat{L}_i \; [\theta_p(u_{1i} - k_1 - n_1 K_1) \times \theta_p(u_{2i} - k_2 - n_2 K_2) \times \theta_p(u_{3i} - k_3 - n_3 K_3)], \tag{2.43}$$

where the operator $\hat{L}_i$ is given in Eq. (2.2). In Eq. (2.43) the inner sums are over all integers n_1, n_2, n_3 (as above each sum is actually finite). The discrete Fourier transform of this array is

$$Q^F(m_1, m_2, m_3) = \sum_{k_1=0}^{K_1-1} \sum_{k_2=0}^{K_2-1} \sum_{k_3=0}^{K_3-1} Q^R(k_1, k_2, k_3) \times \exp\left[2\pi i \cdot \left(\frac{m_1 k_1}{K_1} + \frac{m_2 k_2}{K_2} + \frac{m_3 k_3}{K_3}\right)\right]. \tag{2.44}$$

In these and other equations, the superscripts R and F denote arrays in real and Fourier space, respectively. The structure factor (2.6) is then approximated by

$$S(\mathbf{m}) \simeq b_1\left(\frac{m_1}{K_1}\right) b_2\left(\frac{m_2}{K_2}\right) b_3\left(\frac{m_3}{K_3}\right) \; Q^F(m_1, m_2, m_3)$$

$$\text{if} \quad -\frac{1}{2} \leqslant \frac{m_\alpha}{K_\alpha} < \frac{1}{2}, \quad \alpha = 1,2,3$$

$$= 0, \quad \text{otherwise.} \tag{2.45}$$

In the computation of the reciprocal potential, the prefactor

$$\left| b_1\left(\frac{m_1}{K_1}\right) b_2\left(\frac{m_2}{K_2}\right) b_3\left(\frac{m_3}{K_3}\right) \right|^2$$

arising from $|S(\mathbf{m})|^2$ is absorbed into the definition of a generalized influence function $G^F(\mathbf{m})$ in Fourier space defined by

$$G^F(m_1, m_2, m_3) = \frac{1}{\pi V} \frac{\exp(-\pi^2 \mathbf{m}^2/\beta^2)}{\mathbf{m}^2} \left| b_1\left(\frac{m_1}{K_1}\right) \right|^2 \cdot \left| b_2\left(\frac{m_2}{K_2}\right) \right|^2 \cdot \left| b_3\left(\frac{m_3}{K_3}\right) \right|^2$$

$$-\frac{1}{2} \leqslant \frac{m_\alpha}{K_\alpha} < \frac{1}{2}, \quad \mathbf{m}^2 \neq 0$$

$$= 0, \quad \text{otherwise.} \tag{2.46}$$

The reciprocal electrostatic potential at $\mathbf{r}_i$ is then given

$$\phi_{\text{rec}}(\mathbf{r}_i) = \frac{1}{\pi V} \sum_{\mathbf{m} \neq 0} \frac{\exp(-\pi^2 \mathbf{m}^2/\beta^2)}{\mathbf{m}^2} g_p\left(\frac{-m_1}{K_1}, u_1\right) \cdot g_p\left(\frac{-m_2}{K_2}, u_2\right) \cdot g_p\left(\frac{-m_3}{K_3}, u_3\right) \cdot S(\mathbf{m})$$

$$= \sum_{\mathbf{n}} \theta_p(u_{1i} - n_1) \;\; \theta_p(u_{2i} - n_2) \;\; \theta_p(u_{3i} - n_3) \cdot (G^R \star Q^R)(\mathbf{n}), \tag{2.47}$$

where Parseval's identity is used to obtain the latter equality. The potential on the grid, $\phi(\mathbf{n}) = (G^R \star Q^R)(\mathbf{n})$ is the convolution of the multipole array Q^R and the pair potential G^R, which is the inverse discrete Fourier transform of $G^F(\mathbf{m})$. The convolution is evaluated at the real-space grid points $\mathbf{n}$.

The approximate reciprocal energy is given by

$$U_{\text{rec}} = \frac{1}{2\pi V} \sum_{\mathbf{m} \neq 0} \frac{\exp(-\pi^2 \mathbf{m}^2/\beta^2)}{\mathbf{m}^2} S(\mathbf{m}) S(-\mathbf{m})$$

$$= \frac{1}{2} \sum_{\mathbf{m} \neq 0} G^F(\mathbf{m}) \cdot Q^F(\mathbf{m}) \cdot Q^F(-\mathbf{m})$$

$$= \frac{1}{2} \sum_{m_1=0}^{K_1-1} \sum_{m_2=0}^{K_2-1} \sum_{m_3=0}^{K_3-1} Q^R(\mathbf{m}) \cdot (G^R \star Q^R)(\mathbf{m}). \tag{2.48}$$

Now we examine in more detail the intermediate steps in the previous computation, as well as the calculation of the field and the forces. We proceed in the same spirit as for the direct part. We use the unidimensional multipolar vector $\mathbf{M}$. In the calculation of the generalized grid multipolar array $Q^R(k_1, k_2, k_3)$, the application of the operator $\hat{L}_i$ to the spline product in Eq. (2.43) involves terms that contain the

"dot" products of each multipole with gradients of the scaled fractional coordinates u_1, u_2, u_3, with $u_\alpha = K_\alpha \mathbf{a}_\alpha^* \cdot \mathbf{r}$, for $\alpha = 1,2,3$. For instance, the octupole gives rise to terms containing $[\mathbf{O}\vdots\nabla u_\alpha \nabla u_\alpha \nabla u_\alpha \, (\partial^3 \theta_\alpha / \partial u_\alpha^3) \, \theta_\beta \, \theta_\gamma]$, $[\mathbf{O}\vdots\nabla u_\alpha \nabla u_\alpha \nabla u_\beta \, (\partial^2 \theta_\alpha / \partial u_\alpha^2) \, (\partial \theta_\beta / \partial u_\beta) \, \theta_\gamma]$, and $[\mathbf{O}\vdots\nabla u_1 \nabla u_2 \nabla u_3 \, (\partial \theta_1 / \partial u_1) \, (\partial \theta_2 / \partial u_2) \, (\partial \theta_3 / \partial u_3)]$; where θ_α is a short-hand notation for $\theta_p(u_\alpha - n_\alpha)$. Since $\nabla u_\alpha = K_\alpha \mathbf{a}_\alpha^*$, for $\alpha = 1,2,3$, one can define transformation vectors

$$A_{\alpha,j} = K_\alpha a_{\alpha,j}^* \tag{2.49}$$

which in turn are used to define the elements of a 35×35 (through hexadecapoles) transformation matrix. This matrix converts the unidimensional *Cartesian* multipole vector $\mathbf{M}$ into a *Fractional* multipole vector $\mathbf{M}'$, whose elements are transformed according to

$$\begin{aligned} q' &= q, \\ p_i' &= \sum_r A_{ir} p_r, \\ Q_{ij}' &= \sum_{rs} A_{ir} A_{js} Q_{rs}, \\ O_{ijk}' &= \sum_{rst} A_{ir} A_{js} A_{kt} O_{rst}, \\ H_{ijkl}' &= \sum_{rstu} A_{ir} A_{js} A_{kt} A_{lu} H_{rstu}, \end{aligned} \tag{2.50}$$

where the summation indices r, s, t, and u take the values 1, 2, and 3. On applying this conversion, one must remember that the element H_{rstu} is actually the sum of all the elements related by symmetry, and therefore H_{ijkl}' must also be computed in the same way.

Next, we cast each of the spline functions and their corresponding derivatives into three unidimensional vectors $\mathbf{\Theta}_1$, $\mathbf{\Theta}_2$, and $\mathbf{\Theta}_3$. Each spline of order p spans p points in one of the directions of the grid. The number of derivatives DO needed at each multipole level l is $DO = l$ when force interpolation is carried out, or $DO = l + 1$ when spline interpolation is used for the forces. Thus each atom spans a "spline space" of size $(DO+1) \times p$ in each direction. For N atoms, the dimension of each vector $\mathbf{\Theta}$ is therefore $(DO+1) \times p \times N$. With the correct mapping of indices, the computation of the generalized grid multipolar array $Q^R(k_1, k_2, k_3)$ in Eq. (2.43) can be efficiently carried out as a sum of products among elements of $\mathbf{M}'$, $\mathbf{\Theta}_1$, $\mathbf{\Theta}_2$, and $\mathbf{\Theta}_3$.

Next, the grid multipolar array $Q^R(k_1, k_2, k_3)$ and the reciprocal energy and reciprocal potential are computed according to Eqs. (2.44)–(2.48). In analogy to Eq. (2.37), one can obtain a *fractional* potential vector $\mathbf{\Phi}_{\text{rec}}'$ which contains $\phi_{\text{rec}}(\mathbf{r}_i)$ in Eq. (2.47) and all its derivatives (up to the multipolar order) with respect to u_1, u_2, u_3. The fractional force is obtained by a sum of products between elements of $\mathbf{\Phi}_{\text{rec}}'$ and $\mathbf{M}'$, and the reciprocal energy can also be computed in such a way. The last step in the reciprocal computation is to bring the fractional forces and the fractional potential vector $\mathbf{\Phi}_{\text{rec}}'$ into its Cartesian counterparts. For that, one defines the vector transformation

$$A_{j,\alpha} = K_\alpha a_{j,\alpha}^* \tag{2.51}$$

and proceeds in a manner analogous to (2.50) to determine the elements of the new transformation matrix. Finally, the energies and forces from the direct, reciprocal and adjusted contributions are added up, and so are each of the components of the potential vector $\mathbf{\Phi}(\mathbf{r}_i)$, which contains not only the potential and the electrostatic field at each atomic position $\mathbf{r}_i$ but also different derivatives of the electrostatic field which are needed in the computation of the torques and forces.

Programming for efficiency: A very important property of the PME method is its favorable scalability, $O(N \ln(N))$, when compared to the Ewald method $(O(N^2))$. When it comes to the implementation of the multipoles, the PME method has an additional advantage, the factorability of the spline functions along each dimension. The dimension of each spline function therefore increases only by a factor of l (the multipole order) with respect to the function for charges alone. As a consequence there is a significant efficiency gain when filling the charge grid array in Eq. (2.43). In particular, memory access and loading becomes inexpensive, increasing only by a factor of 2 when going from charges to hexadecapoles, for a fixed cutoff and spline order. As a consequence, calculations in reciprocal space become inexpensive compared to those in the direct sum, even with the use of the McMurchie–Davidson formalism. Strategically, therefore, it becomes convenient to move the bulk of the calculation into reciprocal space. This point will be further discussed in the Results.

E. Multigrid method with multipolar interactions

When it comes to parallelization issues, only the U_{rec} term in the electrostatic energy is problematic in the Ewald and PME methods. This is because these methods depend on the efficient implementation of the 3D-FFTs, which still is rather difficult on the current parallel machines. Essentially, this is due to the large communication overhead associated with the global shifting of data between different processors for 3D-FFT evaluations.

Recently, we presented an $O(N)$ multigrid-based method for the efficient calculation of the long-range electrostatic forces needed for biomolecular simulations, that is suitable for implementation on massively parallel architectures.[26] Along general lines, the method consists of: (i) a charge assignment scheme, which both interpolates and smoothly assigns the charges onto a grid; (ii) the solution of Poisson's equation on the grid via multigrid methods; (iii) the back-interpolation of the forces and energy from the grid to the particle space. Careful approaches for the charge assignment and the force interpolation, and a Hermitian approximation of Poisson's equation on the grid allow for the generation of the high-accuracy solutions required for high-quality molecular dynamics simulations. In this section, we present an extension to the lattice Gaussian multigrid (LGM) method,[26] which uses Gaussian functions for both interpolation and smoothing.

Instead of computing $U_{\rm rec}$ in reciprocal space via FFT, it is possible to solve a Poisson's equation for the compensating cloud of ''co-ion'' Gaussians traditionally used in the Ewald formalism,

$$\nabla^2 \phi_{\rm rec}(\mathbf{r}) = -4\pi\rho_s(\mathbf{r}), \tag{2.52}$$

where $\rho_s(\mathbf{r})$ is the smoothly varying Gaussian charge distribution. In the presence of multipoles, this can be generalized as

$$\rho_s(\mathbf{r}) = \frac{\beta^3}{\pi^{3/2}} \sum_{j=1}^{N} \hat{L}_j \exp(-\beta^2|\mathbf{r}-\mathbf{r}_j|^2), \tag{2.53}$$

where the operator $\hat{L}_j$ is given in Eq. (2.2). As discussed in our previous work,[26] a particular accurate implementation of the FFT-based solution of the Poisson's equation is the fsimulations implementationast Fourier Poisson (FFP) method,[35] where the electrostatic energy can be expressed as

$$\begin{aligned} U_{\rm dir} &= \frac{1}{2} \sum_{\mathbf{n},i,j}{}' \ \hat{L}_i \hat{L}_j \frac{\mathrm{erfc}(\beta|\mathbf{r}_j-\mathbf{r}_i+\mathbf{n}|/\sqrt{2})}{|\mathbf{r}_j-\mathbf{r}_i+\mathbf{n}|}, \\ U_{\rm rec} &= \frac{1}{2} \int \rho_s(\mathbf{r}) \ \ \phi_{\rm rec}(\mathbf{r}) \ \ d^3r. \end{aligned} \tag{2.54}$$

This equation has the advantage that one can use the charge density Gaussians as interpolating functions. The $U_{\rm rec}$ term, computed via five 3DFFT's in the original FFP method, is now computed via a multigrid method.

Multigrid methods are used to solve linear and nonlinear elliptic partial differential equations, and integrodifferential equations.[36–42] Essentially, the multigrid approach is based on the observation that the individual frequency components in Fourier space of the error are most effectively reduced on grids whose spacing is similar in magnitude to the wavelength of the error component.[43] Thus, when one considers an iteration on a fine grid, the iteration readily reduces the high-frequency components of the error, but is rather slow in eliminating the corresponding low-frequency components. Multigrid algorithms aim to take advantage of these observations by working with different grids. In our implementation, we have used a standard weighted restriction operator to restrict the *residual* from the finer to the coarser grids. In this operation, each coarse-grid point is obtained as the weighted average of the corresponding fine-grid point and the 26 fine-grid points surrounding it. The interpolation from coarse to fine grid is performed through trilinear interpolation. The relaxation is carried through the *red–black Gauss–Seidel* method. For high accuracy, we have applied a deferred defect correction scheme,[38,43,44] which uses a double discretization for the Poisson's equation. On the finest grid, we use a Hermitian representation of Poisson's equation, while on the coarser grids the seven-point central finite-difference operator for the Laplacian is used. The gain in accuracy in the Hermitian methods is obtained not by including more points in the Laplacian representation, but by considering that the entire differential equation is satisfied at several grid points. The resulting formulas, therefore, involve the value of the function *and the derivatives* at several grid points. They are called Hermitian formulas by analogy with Hermite's interpolation formula (which essentially is a generalization of Taylor's theorem).[45] The big advantage of the Hermitian representation versus a higher order representation of the ∇^2 operator is that for a given accuracy, the former requires less grid points, and therefore, less communication between processors in a parallel implementation. The accelerated convergence on all length scales provided by the introduction of the coarse grids, allows the solution of elliptical partial differential equations on grids with K points in $O(K)$ operations.[43] For condensed-matter systems with a density near that of water, the total number of grid points K is linearly proportional to the number of atoms N; i.e., the multigrid is truly an $O(N)$ method.

Our solution follows the general structure described in the previous sections. The direct part calculation is exactly the same for all three methods. The unidimensional treatment of the multipoles and associated derivatives is also adopted for the multigrid method. The steps of this method are the following:

(1) **Charge assignment**: Both charge interpolation and charge smoothing are carried out in the same step. If $\mathbf{x}$ indicates a point on the lattice, and $\mathbf{r}_j$ indicates the coordinates of particle j, the charge assignment is performed directly by the Gaussian:

$$\rho_s(\mathbf{x}) = \frac{\beta^3}{\pi^{3/2}} \sum_{j=1}^{N} \ \ M_{tuv}(j) \ \ D_{tuv}(j) \exp(-\beta^2|\mathbf{x}-\mathbf{r}_j|^2) \tag{2.55}$$

which is computed up to a cutoff R_G. This assignment works better for relatively high β's, since the Gaussian decays faster, and a smaller R_G is needed for convergence.

(2) **Solution on the grid**: Via the multigrid method enhanced with the deferred defect correction scheme.

(3) **Back interpolation**: Once the electrostatic potential is computed on the grid, it can be interpolated back to the particle positions. The reciprocal energy can be directly computed on the grid,

$$\begin{aligned} \phi_{\rm rec}(\mathbf{r}_i) &= \frac{\beta^3}{\pi^{3/2}} \sum_{\mathbf{x}} \ \exp(-\beta^2|\mathbf{x}-\mathbf{r}_i|^2) \ \ \phi_{\rm rec}(\mathbf{x}) \ \ h_x h_y h_z, \\ U_{\rm rec} &= \tfrac{1}{2} \sum_{\mathbf{x}} \ \rho_s(\mathbf{x}) \ \ \phi_{\rm rec}(\mathbf{x}) \ \ h_x h_y h_z, \end{aligned} \tag{2.56}$$

where h_x, h_y, and h_z are the mesh sizes. As done in the direct part and in the PME method, one can obtain a potential vector $\mathbf{\Phi}_{\rm rec}$ which contains $\phi_{\rm rec}(\mathbf{r}_i)$ in the equation above and all its derivatives (up to the multipolar order). By replacing $\rho_s(\mathbf{x})$ by its expression, Eq. (2.55), one can take the derivative of the energy with respect to the coordinates of particle i, to obtain the corresponding force

$$\begin{aligned} \mathbf{F}_{{\rm rec},i} = &-\frac{2\beta^5}{\pi^{3/2}} M_{tuv}(i) \ \ D_{tuv}(i) \sum_{\mathbf{x}} \ (\mathbf{x}-\mathbf{r}_i) \\ &\times \exp(-\beta^2|\mathbf{x}-\mathbf{r}_i|^2) \, \phi_{\rm rec}(\mathbf{x}) \ \ h_x h_y h_z. \end{aligned} \tag{2.57}$$

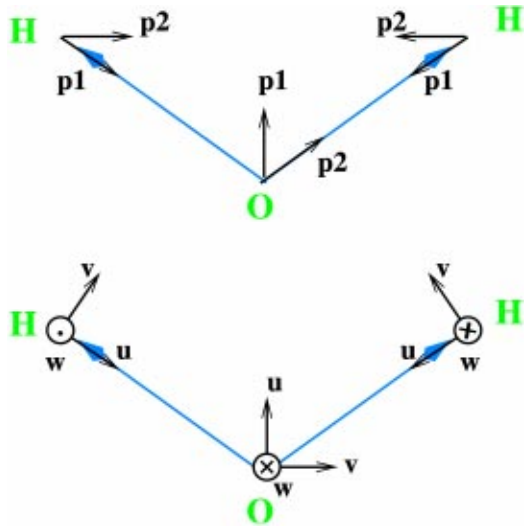

FIG. 1. Scheme of the local axes $\mathbf{p}_1$ and $\mathbf{p}_2$ for each atom in a water model and the corresponding local orthogonal frames that they define.

F. Global frames and treatment of torques

All the multipolar components that appear in the previous sections are referred to a *global* coordinate system. It is necessary therefore to transform the *local* multipole moments—generally defined in reference to the molecule—to a global framework before any calculation starts. This is achieved by defining ''frames'' (local orthogonal coordinate systems) for the molecules. We have implemented a generalization of the method that we introduced previously for the dipolar case.[21] This method is also similar to that provided in the optimal partitioning of electric properties (OPEP) code.[46] In general lines, the method consists of selecting for each atom two reference axes, based on the general geometry of the molecule. For instance, one axis could be the vector pointing towards the neighbor, the bisector between two nearest neighbors, the sum of the vectors pointing to three nearest neighbors, etc. Once these vectors have been chosen one proceeds to orthogonalize them. For instance, if $\mathbf{p}_1$ and $\mathbf{p}_2$ represent the first and second axis respectively for each atom, one can define an orthogonal local frame for each atom, represented by the orthogonal *unit* vectors $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ as follows: $\mathbf{u}=\mathbf{p}_1/|\mathbf{p}_1|$; $\mathbf{y}=\mathbf{p}_2-(\mathbf{p}_2\cdot\mathbf{u})\,\mathbf{u}$ and $\mathbf{v}=\mathbf{y}/|\mathbf{y}|$; $\mathbf{w}=\mathbf{u}\times\mathbf{v}$. For the case of water, the choice of frames is particularly simple, as shown in Fig. 1.

Once the local frames are defined, we can implement our ''unidimensional machinery'' to find the global multipolar components. A transformation matrix is defined for all the sites in the system (including those sites not centered at the atomic positions). The components of these unit vectors ($u_x=u_1$, $u_y=u_2$, etc.) are used to define a 3×3 matrix in a similar way to that in Eq. (2.49). For $\alpha=1,2,3$ it is[46]

$$A_{\alpha,1}=u_\alpha,\qquad A_{\alpha,2}=v_\alpha,\qquad A_{\alpha,3}=w_\alpha, \tag{2.58}$$

which in turn is used to define the elements of a 35×35 transformation matrix. This matrix converts the unidimensional *local* multipole vector $\mathbf{M}$ into a *global* multipole vector $\mathbf{M}'$, whose elements are transformed according to Eq. (2.50). In the equations presented in the previous sections we omitted the prime and always used the global multipole vector.

To carry out molecular dynamics, we need to convert the torques produced by every multipole into atomic forces. The first step is the computation of the torques at each site. Starting with the infinitesimal counter-clockwise rotations around axes x, y, and z,

$$\delta R^x=\begin{pmatrix}0&0&0\\0&0&1\\0&-1&0\end{pmatrix}, \tag{2.59}$$

$$\delta R^y=\begin{pmatrix}0&0&-1\\0&0&0\\1&0&0\end{pmatrix}, \tag{2.60}$$

$$\delta R^z=\begin{pmatrix}0&1&0\\-1&0&0\\0&0&0\end{pmatrix}, \tag{2.61}$$

it is possible to form a transformation matrix for the unidimensional multipole vector following the transformation rules given in Eq. (2.50). Thus, if $R_{ij}=\delta_{ij}$ represents the unitary matrix and δR^α_{ij} represents an infinitesimal rotation around the axis α, with $\alpha=x,y,z$, the infinitesimally rotated multipole vector $\mathbf{M}^\alpha$ around α will have the following components:

$$\begin{aligned}
q^\alpha&=0,\\
p_i^\alpha&=\sum_r\ \delta R^\alpha_{ir}p_r,\\
Q_{ij}^\alpha&=\sum_{rs}\ (R_{ir}\delta R^\alpha_{js}+\delta R^\alpha_{ir}R_{js})\ \ Q_{rs},\\
O_{ijk}^\alpha&=\sum_{rst}\ (R_{ir}R_{js}\delta R^\alpha_{kt}+R_{ir}\delta R^\alpha_{js}R_{kt}+\delta R^\alpha_{ir}R_{js}R_{kt})\ \ O_{rst},\\
H_{ijkl}^\alpha&=\sum_{rstu}\ (R_{ir}R_{js}R_{kt}\delta R^\alpha_{lu}+R_{ir}R_{js}\delta R^\alpha_{kt}R_{lu}+R_{ir}\delta R^\alpha_{js}R_{kt}R_{lu}\\
&\quad+\delta R^\alpha_{ir}R_{js}R_{kt}R_{lu})\ \ H_{rstu}.
\end{aligned} \tag{2.62}$$

The torque can then be easily obtained by making the scalar product between $\mathbf{M}^\alpha$ and the potential vector $\mathbf{\Phi}$, i.e.,

$$\tau_\alpha(\mathbf{r}_i)=M^\alpha_{tuv}(i)\ \ \Phi_{tuv}(i). \tag{2.63}$$

Thus, this procedure recovers the well-known equation of the torque produced by a dipole, $\boldsymbol{\tau}=\mathbf{p}\times\mathbf{E}$, that for a quadrupole, $\tau_\alpha=-2\Sigma_{j=1}^3\epsilon_{\alpha\beta\gamma}Q_{\beta,j}(\partial^2\phi/\partial x_\gamma\partial x_j)$, where $\epsilon_{\alpha\beta\gamma}$ is the Levi–Civita symbol, etc.

Now we need to transform these site torques, that may also be sitting on extra-nuclear sites, into atomic forces. The

TABLE II. This table shows results for the PME implementation of the permanent electrostatic multipole moments. The timing was performed on Intel Xeon, 3.06 GHz, 512 KB cache, 2 GB memory (compiler is g77), 1 processor. The averaged timings are per step. The system consists of 4 096 water molecules (12 288 atomic sites) in a 49.6 Å cubic box. The table shows the *relative* energy and force rms errors, the order of the spline used for interpolation, the mesh size h_x, the Ewald parameter β, and the cutoff radius R_c for the direct interactions. Timings shown are for the direct interactions (they also include the van der Waals interactions out to 8 Å cutoff with continuous correction), the reciprocal interactions, the "other" interactions (which include the adjusted and self terms and a variety of tensor operations needed for the calculations), and the time for the total force. The parameters were chosen so that the relative force error is $\sim 5\times10^{-4}$. The last two rows give timings for one-order and two-order reduction in relative force error.

Multipole level	Energy error	Force error	Spline order	h_x (Å)	β (Å^{-1})	R_c (Å)	Time (s) direct	Time (s) reciprocal	Time (s) other	Time (s) force
0	2.3×10^{-5}	4.8×10^{-4}	5	0.775	0.50	5.63	0.21	0.18	0.03	0.42
1	1.9×10^{-5}	4.9×10^{-4}	5	0.775	0.50	5.60	0.33	0.21	0.04	0.58
2	4.9×10^{-5}	5.0×10^{-4}	6	0.688	0.55	5.10	0.57	0.32	0.07	0.96
3	3.4×10^{-5}	4.7×10^{-4}	8	0.620	0.70	4.25	0.93	0.60	0.17	1.70
4	1.3×10^{-6}	4.9×10^{-4}	8	0.459	0.85	3.60	1.54	1.12	0.39	3.05
4	1.9×10^{-6}	5.0×10^{-5}	10	0.344	0.90	3.85	1.81	2.23	0.38	4.42
4	2.2×10^{-7}	4.8×10^{-6}	12	0.344	0.95	4.06	2.00	2.66	0.37	5.03

pair of vectors $\{\mathbf{p}_1,\mathbf{p}_2\}$ used to build the local frames are also useful to convert the site torques to forces. We have devised a method to convert the site torques into atomic forces. First, the site multipoles are converted into torques according to the procedure described above. Since the local frames are defined in terms of the atomic positions, the torques produced by the extra-nuclear sites have to be referred to the nuclear sites by adding the translational contribution $\mathbf{r}\times\mathbf{F}$, where $\mathbf{r}$ is the vector from the reference site to the extra-nuclear site and $\mathbf{F}$ is the force on the extra-nuclear site. In general terms, the change of energy produced by a torque $\boldsymbol{\tau}$ under an infinitesimal rotation $d\boldsymbol{\Omega}$ is given by

$$dU_\Omega = -\boldsymbol{\tau}\cdot d\boldsymbol{\Omega}. \tag{2.64}$$

It is possible then to carry out infinitesimal rotations of $\mathbf{p}_1$ and $\mathbf{p}_2$ such that the infinitesimal vector change, say $d\mathbf{p}_1$, is related to the infinitesimal rotation $d\boldsymbol{\Omega}$ by $d\mathbf{p}_1 = d\boldsymbol{\Omega}\times\mathbf{p}_1$. For instance, if the vector $\mathbf{p}_1$ in the oxygen frame in Fig. 1 rotates around the $\mathbf{w}$ axis an infinitesimal angle $d\Omega_w$, the change in $\mathbf{p}_1$ is given by $d\mathbf{p}_1 = d\Omega_w \mathbf{w}\times\mathbf{p}_1 = \mathbf{v}\, d\Omega_w |\mathbf{p}_1| = dp_{1v}\,\mathbf{v}$ (remember that $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ are *unit* vectors). Therefore the change in energy due to $d\Omega_w$ is $dU = -\tau_w\, d\Omega_w = -\tau_w\, dp_{1v}/|\mathbf{p}_1|$ or $\partial U/\partial p_{1v} = -\tau_w/|\mathbf{p}_1|$. In this way, we can relate infinitesimal variations in energy due to infinitesimal rotations of the vectors $\mathbf{p}_1$ and $\mathbf{p}_2$ to the torques defined with respect to each frame. When doing this, two things should be kept in mind. First, that several sites can contribute to the torque in each frame and second, it is simpler to choose rotations of $\mathbf{p}_1$ that leave $\mathbf{p}_2$ invariant, and vice versa. Next, we convert the infinitesimal changes in energy into local cartesian coordinates, i.e.,

$$\frac{dU}{dp_{1x}} = \frac{dU}{dp_{1u}}u_1 + \frac{dU}{dp_{1v}}v_1 + \frac{dU}{dp_{1w}}w_1,$$

etc. Finally, we apply the chain rule to obtain the gradients of the energy in the global reference system

$$\frac{dU}{dx_\alpha} = \sum_\beta \left[\frac{dU}{dp_{1\beta}}\frac{dp_{1\beta}}{dx_\alpha} + \frac{dU}{dp_{2\beta}}\frac{dp_{2\beta}}{dx_\alpha}\right], \tag{2.65}$$

where x_α, $\alpha = 1,2,3$ are the atomic coordinates.

III. RESULTS AND DISCUSSION

Historically, the reason why higher order multipolar interactions have not been implemented into macromolecular codes is their high cost. In principle, to take into account multipoles up to hexadecapoles means that there are 35 degrees of freedom, and the interaction matrix between them has 1225 components. In a standard, *fixed-cutoff* implementation of these interactions, the computation time grows with the square of the multipolar components and therefore becomes expensive when compared to charge–charge interactions. The belief that—since higher-order multipole interactions decay much faster than charge interactions—the use of a cutoff alleviates the problem proves incorrect. In fact, *most of the cost of the interactions originates in the* **direct** *part*, even under relatively short cutoffs. Additionally, too short of a cutoff produces serious force and energy errors. This is further illustrated by the fact that truncation of the dispersive van der Waals interactions (which decay as $1/r^6$) leads to artifactual behavior unless large cutoffs ($R_c \sim 30$ Å) are used, or a smaller cutoff (say, $R_c \sim 8$ Å) is used along with some treatment for the remainder of the long-range interaction. On the other hand, moving the majority of the computation of the interactions into reciprocal space not only preserves accuracy but also has a moderate cost, which justifies the reduction of the cutoff for the direct space. This is true as far as the PME and multigrid methods are concerned; the traditional Ewald method still scales as $\sim O(1225N^2)$ and therefore is not competitive.

To time the performance of our code we have augmented a TIP3P model with charges and multipoles up to hexadecapoles, whose values were obtained for a different water model.[47] This "hybrid" model does not necessarily describe a physically correct water model, but provides an easy model to test all the possible multipolar interactions with reasonable multipole parameters. The force and energy relative rms errors in Tables II and III are computed with respect to costly "exact" values calculated using very large cutoffs, high spline orders, and very small mesh sizes. The parameters are chosen such that the *relative* force error ϵ stays the same for all multipole levels (in this case, $\epsilon \simeq 5\times10^{-4}$). This requires

TABLE III. This table shows results for the MULTIGRID implementation of the permanent electrostatic multipole moments. The timing was performed on Intel Xeon, 3.06 GHz, 512 KB cache, 2 GB memory (compiler is g77), 1 processor. The averaged timings are per step. The system consists of 4 096 water molecules (12 288 atomic sites) in a 49.6 Å cubic box. The table shows the *relative* rms energy and force errors, the mesh size h_x, the Ewald parameter β, the cutoff radius R_c for the direct interactions, the cutoff radius R_G for the Gaussians in the Poisson's equation, and the averaged timing for the reciprocal interactions. The parameters were chosen so that the relative force error is $\sim 5\times 10^{-4}$. The timings for ''other'' calculations is the same as in Table II and not shown here.

Multipole level	Energy error	Force error	h_x (Å)	β (Å^{-1})	R_c (Å)	R_G (Å)	Time (s) direct	Time (s) reciprocal
0	3.9×10^{-4}	5.1×10^{-4}	0.620	0.60	5.20	3.50	0.20	2.30
1	2.9×10^{-4}	5.0×10^{-4}	0.620	0.61	5.20	3.63	0.29	2.66
2	4.0×10^{-4}	5.1×10^{-4}	0.516	0.70	4.80	3.45	0.52	4.64
3	3.5×10^{-4}	5.0×10^{-4}	0.443	0.75	4.25	3.10	0.94	8.42
4	2.9×10^{-4}	5.0×10^{-4}	0.388	0.79	4.25	3.05	2.21	15.71

that as the multipolar order increases, the spline order and the Ewald parameter β be increased, while the mesh space h_x and the cutoff radius R_c (and the cutoff radius R_G for the Gaussians in the multigrid method) be decreased. Increasing β and decreasing R_c effectively moves most of the calculation into reciprocal (or ''multigrid'') space. Both Tables II and III are obtained for a system with 12 288 atoms (or 4096 waters). Table II shows the parameters used for the PME multipole implementation and the corresponding results. The ''other'' time column in the table includes the contributions from the adjusted and self terms as well as a variety of tensor operations needed for the calculations. This time is relatively insignificant compared to that from the direct and reciprocal contributions. The last two rows in Table II give timings for one-order and two-order reduction in relative force error for the full multipolar calculation. There is an increase in computer time of ~ 1.45 times to reduce the force error an order of magnitude from $\epsilon \simeq 5\times 10^{-4}$ to $\epsilon \simeq 5\times 10^{-5}$ and of ~ 1.14 times to reduce the force error an order of magnitude from $\epsilon \simeq 5\times 10^{-5}$ to $\epsilon \simeq 5\times 10^{-6}$. There is an increase in computer time of ~ 1.65 to reduce the force error two orders of magnitude from $\epsilon \simeq 5\times 10^{-4}$ to $\epsilon \simeq 5\times 10^{-6}$.

For reference, an AMBER 7 run with sander auto settings (8 Å cutoff for the direct part; Ewald coefficient $\beta = 0.348\,64$; spline order 4; mesh size 0.992 Å) yields a comparable relative rms force error of 3.8×10^{-4} with direct sum time 0.23 s; reciprocal 0.13 s; and total Ewald sum time 0.36 s. Our new implementation with all multipolar interactions up to hexadecapole–hexadecapole costs therefore only a factor of $\simeq 8.5$ more than the standard AMBER implementation with charge–charge interactions.

For the sake of comparison, we have also implemented a Coulomb code with an optimized McMurchie–Davidson acceleration. We find that a ''regular'' cutoff of 8 Å increases the computation time by a factor of ~ 6 when compared to the total force time in the first hexadecapole row in Table II. In addition, the relative rms force error increases about two orders of magnitude with respect to that reported in Table II. It is fair to argue that the cutoff should only be used for quadrupoles and higher order multipoles. We have therefore repeated the calculation where the charges and dipoles of the model have been set to zero (that is, all the interactions are quadrupole–quadrupole or higher order and should converge absolutely). In our tests we found that the relative rms force error is greater than 1% at a 7 Å cutoff and about 8×10^{-3} at an 8 Å cutoff. A run with a 7 Å cutoff costs approximately three times more than the optimized PME which uses a 3.6 Å cutoff (and computes the additional charge and dipole interactions).

Table III shows the parameters used for the multigrid multipole implementation and the corresponding results. The ''other'' time (that includes the contributions from the adjusted and self terms as well as a variety of tensor operations needed for the calculations) is the same as the ''other'' time column in Table II and therefore is not included here. The multigrid is implemented with the specific goal of parallelization. The parallel implementation of the ''reciprocal'' part is a straightforward generalization of our previous method.[26] From the timings shown in Table III (obtained in one processor) one would expect that a relatively efficient parallel scheme would make this method quite competitive. Since the direct part is computed with the Ewald coefficient β, the width of the Gaussians that define the charge density in the Poisson's equation is determined by $\sqrt{2}\beta$. Large β's (narrow Gaussians) need smaller Gaussian cutoff radius R_G for the representation of the Gaussian on the grid. However, accuracy requires that many points of the Gaussian be sampled (since gradients of the distribution are needed), which in turn means that narrow Gaussians need smaller mesh sizes, if they are to subtend a sufficient number of grid points. Therefore, for a fixed accuracy, a maximum efficiency will be achieved as a compromise between reducing the Gaussian cutoff radius R_G and increasing the mesh size h_x (in such a way as to preserve accuracy). In principle, there is a coefficient β that will optimize the results of this compromise, but of course this β also weighs the relative contributions of U_{dir} and U_{rec}. The ideal set of h_x, β, R_C, and R_G that optimize the performance of the multigrid method will be determined once parallelization of the entire code is complete.

As a test of our implementation within the sander molecular dynamics code, we performed a 1 ns simulation (1 fs time step) of 216 waters using this ''hybrid'' model at the hexadecapole level. The model proved difficult to equilibrate, likely due to the imbalance between van der Waals and electrostatic forces, which we did not attempt to alleviate. Consequently the average temperature was too high (≈ 330 K) and the potential energy too negative for a realistic water simulation. Nonetheless, running under NVE con-

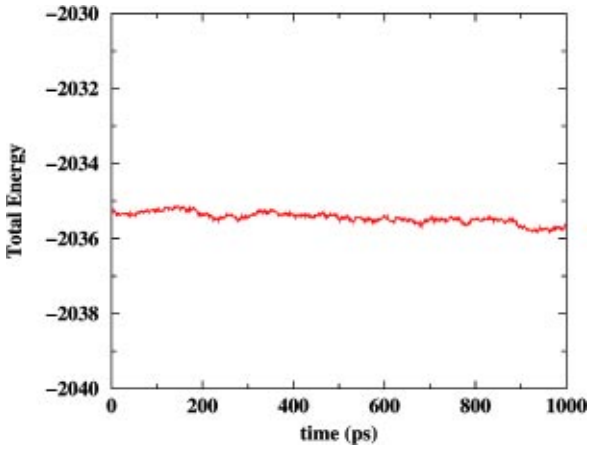

FIG. 2. Total energy for a bath of 216 waters. The water model is a ''hybrid'' model (TIP3P water enhanced with multipoles up to hexadecapole, see text).

ditions the simulation displayed reasonable energy conservation, as shown in Fig. 2. Over short runs of several picoseconds, using the parameters from the first hexadecapole row in Table II the energy conservation was comparable to that of TIP3P water with standard PME parameters. However, over the course of a nanosecond noticeable heating occurred, with an energy drift of about 2 kcal/mol out of a total of -2035.4 kcal/mol. Increasing the direct space cutoff to 3.9 Å, corresponding to a direct space relative rms error of 1×10^{-4}, eliminated the heating, with a resulting energy drift of -0.4 kcal/mol/ns (about 3–4 times that of TIP3P using standard PME parameters). In addition, we have implemented a more realistic water model: a polarizable atomic multipole water model by Ren and Ponder,[48] that assigns permanent atomic monopole, dipole, and quadrupole moments to each atomic center. The model also includes polarization effects via atomic dipole polarizabilities. This is a highly realistic model with fully flexible potentials that compares very well with experimental and high level *ab initio* results. Preliminary simulations for liquid water under periodic boundary conditions show very good energy conservation. This work is still in progress and results will be reported in the near future.

IV. CONCLUSIONS

The recognition that the inclusion of higher order electrostatic multipoles allows convergence to the molecular electrostatic potential outside the van der Waals surface has been a great step towards the accurate description of electrostatics. The fact that different approaches like DMA and AIM converge at the hexadecapole level is in fact most encouraging. What—until now—has been missing is the bridge between this knowledge and the practical, systematic implementation of multipoles in classical biomolecular codes: in fact, the cost of such implementation has often been considered unsurmountable. In this paper we have presented an efficient implementation of higher order multipoles in cartesian tensor formalism which makes systematic use of multipoles in classical MD affordable at quite a low cost. The much improved electrostatics allowed by such a description will probably become very valuable in the elucidation of those—all too common—''sensitive'' cases which do not yield gracefully to the point-charge description.

In our treatment, as in the usual Ewald scheme, the long-range electrostatic interactions are divided in two sums: the *direct* sum, which evaluates the fast-varying, particle–particle interactions, considered up to a given cutoff in real space; and the ''*reciprocal*'' sum, which evaluates the smoothly varying, long-range part of the interaction. To speed up the evaluation of the direct part, we have implemented a McMurchie–Davidson formalism which was originally developed for the evaluation of quantum molecular integrals over Cartesian Gaussians. The reciprocal part has been implemented in three different ways: using an Ewald scheme, a PME-based approach and a multigrid-based approach, which evaluates the Poisson's equation in real space. Our code is flexible to include multipoles to any desired order. In this paper we report results to hexadecapoles.

We found that the combination of the McMurchie–Davidson formalism with the PME method increases the cost of multipolar interactions (which include all order interactions up to hexadecapole–hexadecapole) by a factor of only ~8.5 with respect to the regular AMBER charge-only implementation. Even with the efficiency gained by the McMurchie–Davidson recursion relations, the direct part is considerably expensive and most of the savings are obtained by using a short cutoff for the direct part, and by performing most of the calculation in reciprocal space via the PME method. Not only does this method present the advantage of good scaling with the number of particles [as $O(N\ln(N))$] but also it has the additional advantage—important for our present purposes—of the factorability of the spline functions along each dimension. The dimension of each spline function therefore increases only by a factor of l (the multipole order) with respect to the function for charges alone. As a consequence the filling of the charge grid array in Eq. (2.43) is very efficient. In particular, memory access and loading becomes computationally inexpensive, increasing only by a factor of 2 when going from charges to hexadecapoles, for a fixed cutoff and spline order. The multigrid method is slower in one processor but shows promising results for parallelization, and it provides a natural way to interface with continuous, Cartesian Gaussian-based electrostatics in the future. Immediate future work includes its parallelization using the MPI library. We are also working on a a new method of calculating the distributed multipoles of a molecular charge density based on the maximally-localized Wannier functions.[49,50] Future work will address the implementation of higher-order, distributed charge-density susceptibility functions, to deal with the induction and dispersion contributions to the energy.

ACKNOWLEDGMENT

Support for this work has been provided by a NSF-ITR Award No. 0121361.

[1] A. J. Stone, *The Theory of Intermolecular Forces* (Clarendon, Oxford, 1996).
[2] C. Sagui and T. A. Darden, Annu. Rev. Biophys. Biomol. Struct. **28**, 155 (1999).
[3] R. Dixon and P. Kollman, J. Comput. Chem. **18**, 1632 (1997).
[4] R. Wheatley and J. Mitchell, J. Comput. Chem. **15**, 1187 (1994).
[5] C. Chipot, J. Angyan, and C. Millot, Mol. Phys. **94**, 881 (1998).
[6] C. Bayly, P. Cieplak, W. Cornell, and P. Kollman, J. Phys. Chem. **97**, 10269 (1993).
[7] M. M. Francl and L. A. Chirlian, in *Reviews in Computational Chemistry*, edited by K. Lipkowitz and D. B. Boyd (VCH, New York, 1999), Vol. 14.
[8] S. Price, in *Reviews in Computational Chemistry*, edited by K. Lipkowitz and D. B. Boyd (VCH, New York, 1999), Vol. 14.
[9] P. Popelier, *Atoms in Molecules: An Introduction* (Prentice–Hall, Harlow, 2000).
[10] D. S. Kosov and P. L. A. Popelier, J. Phys. Chem. A **104**, 7339 (2000).
[11] P. L. A. Popelier, L. Joubert, and D. S. Kosov, J. Phys. Chem. A **105**, 8254 (2001).
[12] P. L. A. Popelier and D. S. Kosov, J. Chem. Phys. **114**, 6539 (2001).
[13] A. J. Stone, Chem. Phys. Lett. **83**, 233 (1981).
[14] A. J. Stone, Mol. Phys. **56**, 1065 (1985).
[15] F. Vigné-Maeder and P. Clavérie, J. Chem. Phys. **88**, 4934 (1988).
[16] P. Fowler and A. Buckingham, Chem. Phys. Lett. **176**, 11 (1991).
[17] C. Dykstra, Chem. Rev. **93**, 2339 (1993).
[18] G. Jansen, C. Hättig, B. Hess, and J. Ángyán, Mol. Phys. **88**, 69 (1996).
[19] C. Hättig, G. Jansen, B. Hess, and J. Ángyán, Mol. Phys. **91**, 145 (1997).
[20] R. Bader, *Atoms in Molecules: A Quantum Theory* (Clarendon, Oxford, 1990).
[21] A. Toukmaji, C. Sagui, J. A. Board, and T. Darden, J. Chem. Phys. **113**, 10913 (2000).
[22] W. Smith, CCP5 Information Quarterly **4**, 13 (1982).
[23] T. A. Darden, D. M. York, and L. G. Pedersen, J. Chem. Phys. **98**, 10089 (1993).
[24] U. Essmann *et al.*, J. Chem. Phys. **103**, 8577 (1995).
[25] D. A. Pearlman *et al.*, Comput. Phys. Commun. **91**, 1 (1995).
[26] C. Sagui and T. A. Darden, J. Chem. Phys. **114**, 6578 (2001).
[27] L. McMurchie and E. Davidson, J. Comput. Phys. **26**, 218 (1978).
[28] M. Challacombe, E. Schwegler, and J. Almlöf, Chem. Phys. Lett. **241**, 67 (1995).
[29] S. Nose and M. Klein, Mol. Phys. **50**, 1055 (1983).
[30] J. Applequist, J. Math. Phys. **24**, 736 (1982).
[31] C. Dykstra, J. Comput. Chem. **9**, 476 (1988).
[32] T. Helgaker, P. Jørgensen, and J. Olsen, *Molecular Electronic-Structure Theory* (Wiley, England, 2000).
[33] T. Darden, A. Toukmaji, and L. Pedersen, J. Chim. Phys. Phys.-Chim. Biol. **94**, 1346 (1997).
[34] C. Sagui and T. A. Darden, in *Simulation and Theory of Electrostatic Interactions in Solution*, edited by L. R. Pratt and G. Hummer (AIP, Melville, 1999), Vol. 492.
[35] D. York and W. Yang, J. Chem. Phys. **101**, 3298 (1994).
[36] W. Hackbush and U. Trottenburg, *Multigrid Methods* (Springer-Verlag, Berlin, 1982).
[37] D. Jespersen, *Studies in Numerical Analysis* (Mathematical Association of America, Washington, D.C., 1984), Vol. 24.
[38] W. Hackbush, *Multigrid Methods and Applications* (Springer-Verlag, Berlin, 1985).
[39] W. Briggs, V. Henson, and S. McCormick, *A Multigrid Tutorial*, 2nd ed. (SIAM, Philadelphia, 2000).
[40] S. McCormick, *Multilevel Adaptive Methods for Partial Differential Equations* (SIAM, Philadelphia, 1989).
[41] J. Adams, Appl. Math. Comput. **34**, 113 (1989).
[42] W. Press, S. A. Teukolsky, W. T. Vettering, and B. P. Flannery, *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd ed. (Cambridge University Press, Cambridge, 1992).
[43] A. Brandt, Math. Comput. **31**, 333 (1977).
[44] E. L. Briggs, D. J. Sullivan, and J. Bernholc, Phys. Rev. B **54**, 14362 (1996).
[45] L. Collatz, *The Numerical Treatment of Differential Equations* (Springer-Verlag, Berlin, 1960), p. 164.
[46] J. G. Angyán, C. Chipot, F. Dehez, C. Hättig, G. Jansen, and C. Millot, OPEP: A tool for the optimal partitioning of electric properties, Equipe de chimie et biochimie théoriques, Unité mixte de Recherche CNRS/UHP No 7565, Université Henri Poincaré, B.P. 239, 54506 Vandoeuvre-lés-Nancy Cedex, France, version 1.0-β.
[47] C. Millot *et al.*, J. Phys. Chem. A **102**, 754 (1998).
[48] P. Ren and J. W. Ponder, J. Comput. Chem. **23**, 1497 (2002).
[49] G. Wannier, Phys. Rev. **52**, 191 (1937).
[50] C. Sagui, P. Pomorski, T. A. Darden, and C. Roland (preprint).