The Fast Multipole Method and Point Dipole Moment Polarizable Force Fields

Jonathan P. Coles\textsuperscript{1} and Michel Masella\textsuperscript{2}

\textsuperscript{1}Exascale Computing Research Laboratory, Versailles Saint-Quentin-en-Yvelines University, 45 avenue des Etats-Unis, Versailles, France

\textsuperscript{2}Laboratoire de Chimie du Vivant, Service d’ingénierie moléculaire des protéines, Institut de biologie et de technologies de Saclay, CEA Saclay, F-91191 Gif sur Yvette Cedex, France

(Dated: August 29, 2014)

We present a momentum conserving implementation of the fast multipole method for computing coulombic electrostatic and polarization forces from polarizable force-fields based on induced point dipole moments. We demonstrate the expected \( O(N) \) scaling of that approach by performing single energy point calculations on hexamer protein subunits of the mature HIV-1 capsid. We also show the long time energy conservation in molecular dynamics at the nanosecond scale by performing simulations of a protein complex embedded in a coarse-grained solvent using both a standard integrator and a multiple time step one. Our tests show the applicability of FMM combined with state-of-the-art chemical models in molecular dynamical systems.

\section{INTRODUCTION}

In \( N \)-body simulations, the long-range potentials, such as gravitational or electrostatic potentials, pose the greatest computational difficulty as their cost scales with the number of particles as \( O(N^2) \). Reducing this complexity has been the subject of intense study for over 40 years and has seen the development of many successful algorithms. Particularly important have been the Ewald summation scheme, which achieved \( O(N^{3/2}) \) \cite{1}, and its extension, the particle-mesh-Ewald (PME) scheme \cite{2} based on fast Fourier transforms (FFTs), which scales as \( O(N \log N) \). The tree-code from Barnes and Hut \cite{3} also reduced the cost to \( O(N \log N) \) using multipole expansions of the potential. From a purely algorithmic perspective, the most performant algorithm should be the fast multipole method (FMM) proposed by Greengard and Rokhlin \cite{4}, which scales as \( \tilde{O}(N) \). However, despite volumes of theoretical work, the FMM, on which we focus here, has not been widely used in production molecular dynamics (MD) simulations due to the perceived complexity to implement it, hidden constants that affect its scaling, issues with multiple time step integrators, and early concerns that energy cannot be conserved well enough unless a prohibitively high accuracy is used \cite{5,6}. That explains why PME is presently very popular among many molecular dynamics codes \cite{7} to handle periodic molecular systems containing up to millions of atoms \cite{8}, even if FFT does not scale well on modern supercomputer architectures due to high communication overhead.

More recently, interest in FMM has resurfaced partly due to these problems arising with FFT and the PME methods and partly due to the desire to solve ever larger systems \cite{9,10}. Several successful implementations have existed in the astrophysics community for many years to evolve the extremely large number of particles typically representing dark matter (e.g \cite{11,12}). In this paper we present an implementation of FMM to perform molecular simulations based on advanced many-body interatomic potentials (force-fields) including polarization effects described according to the induced point dipole approach. The development of polarizable force-fields represents a major challenge in the molecular modeling field, as they have been shown to greatly improve the description of complex microscopic systems, such as physical interfaces and those involving charged species \cite{13,19}. They are also suspected to greatly improve the description of amino-acid interactions, which will lead to further promising computational techniques devoted to theoretically investigations of protein folding \cite{20}.

Among the polarizable force-fields proposed so far, the second author has developed a particular kind of force-field which was shown to be easily and efficiently coupled with a polarizable coarse grained approach to model the solvent \cite{21,23}. Here, a complex solute (e.g., a protein) is modelled at the atomic level while the solvent is modelled as a set of polarizable pseudo-particles. The main feature of this approach is the systematic cut off of the solute-solvent interactions whose computation scales as \( O(N) \) (with \( N \) the atomic solute size), while no cut off is considered in computing intra-solute interactions. In other words, the computation of intra-solute electrostatic interactions corresponds to an electrostatic free boundary condition problem. Hence, coupling the modeling approach and an FMM scheme to compute the intra-solute interactions will lead to a full \( O(N) \) polarizable approach to investigate the solvation of large and complex solutes.

Our FMM method is based on the work of Dehnen \cite{24} that uses a binary tree spatial decomposition and conserves linear momentum by construction (via a dual tree walk approach to compute multipole interactions). Accuracy is controlled by an opening angle tolerance parameter \( \theta \) at fixed multipole expansion order.

In this paper we first review in \cite{11} the induced point dipole method and the mathematical basis of FMM. In \cite{111} we discuss some technical details of our implementa-
II. THEORY

A. Induced point dipole polarizable force-fields

Molecular modeling approaches consider the total interatomic potential $U$ as a sum of different energy terms. For polarizable force-fields, a sum of four terms is commonly considered

$$U = U^{\text{short}} + U^{\text{rel}} + U^{\text{pair}} + U^{\text{pol}}$$

Here, $U^{\text{short}}$ is a short range energy term describing the short range atom-atom repulsive effects and, usually, dispersion effects. The term is normally additive in nature, however, there have been attempts to consider short range many-body potentials to account for short range electronic cloud reorganization effects (see, e.g., Ref. [18]). $U^{\text{rel}}$ is the sum of the common stretching, bending and torsional potentials, which describe the interactions among covalently bonded atoms.

$U^{\text{pair}}$ is the standard pairwise Coulomb potential based on static charges centered on the system atoms

$$U^{\text{pair}} = \frac{1}{2} \sum_i \Phi_i = \frac{1}{8\pi\varepsilon_0} \sum_i q_i \sum_j^* q_j \phi(x_i - x_j)$$

where the Green’s function is $\phi(x_i - x_j) = |x_i - x_j|^{-1}$. The superscript $^*$ denotes sums for which a subset of atoms is excluded (commonly, the atoms $j$ separated by less than two chemical bonds from an atom $i$).

$U^{\text{pol}}$ is the polarization term. When considering an induced dipole moment polarization approach, a set of additional degrees of freedom is introduced, the induced dipole moments $\mu_i$, that obey

$$\mu_i = \alpha_i \cdot (E_i^q + E_i^\mu)$$

Here, $\alpha_i$ is the polarizability tensor of the polarizable atom $i$, $E_i^q$ and $E_i^\mu$ are the electric fields generated on atom $i$ by the surrounding static charges $q_j$ and the surrounding induced dipoles $\mu_j$, respectively. These two electric fields are defined as

$$E_i^q = -\sum_j^* q_j \nabla \phi(x_i - x_j)$$

and

$$E_i^\mu = -\sum_j \mu_j \cdot T_{ij}$$

The dipolar tensor $T_{ij}$ is the second derivative of the Green’s function

$$T_{ij} = \nabla^2 \phi(r = x_i - x_j) = \frac{1}{r^3} \left( 1 - \frac{3 r \otimes r}{r^2} \right)$$

where $\otimes$ is the tensor outer product. $T_{ij}$ is, however, usually altered to account for interatomic short range damping effects. Following the original ideas of B. T. Thole [27], this is achieved by adding to $T_{ij}$ a specific short range dipolar tensor $T_{ij}^{\text{damp}}$.

The interactions of the dipoles with the above two electric fields give rise to two additional potentials, namely

$$\Phi_i^{\mu q} = \mu_i \cdot E_i^q \quad \text{and} \quad \Phi_i^{\mu \mu} = \mu_i \cdot E_i^\mu$$

$U^{\text{pol}}$ also typically accounts for a third energy term, $A(\mu)$, quantifying the energy cost to create a dipole $\mu$. In standard induced dipole moment implementations, this third term leads to the fundamental relationship

$$\frac{\partial U^{\text{pol}}}{\partial \mu} = 0$$

For instance, when the dipoles obey the linear relation $[3]$ and the polarizability tensors $\alpha_i$ are taken as isotopic polarizabilities (i.e., they are a scalar quantity), the above condition is met by considering

$$A(\mu) = \frac{1}{2} \sum_i |\mu_i|^2$$

Note that alternative induced dipole moment approaches where the dipoles obey more complex relations have been proposed (see, e.g., Ref. [26]). In that case, a specific $A(\mu)$ energy term is considered that fulfills the relation [8]. Regardless of the form of $A(\mu)$, the total dipole potential energy nevertheless obeys

$$U^{\text{pol}} = A(\mu) - \sum_i \Phi_i^{\mu q} - \frac{1}{2} \sum_i \Phi_i^{\mu \mu}$$

B. The Fast Multipole Method

The fast multipole method (FMM) [4] is a technique for approximating a long range potential

$$\tilde{\Phi}(x_b) = \sum_{a \in P} q_a \phi(x_b - x_a)$$

of a set of particles $P$ via a multipole expansion of the Green’s function $\phi$. Typically, the particles are organized by a hierarchical spatial decomposition such as an oct-tree or binary tree. We consider the interaction of the multipole expansions of the particles in pairs of tree nodes $A, B$ centered at $z_A, z_B$, and containing particles labeled $a, b$, respectively.
The Taylor expansion of $\phi$ to $p^{th}$ order about both centers in Cartesian coordinates yields the following expression

$$\phi(x_b - x_a) \approx \sum_{|n| \leq p} \sum_{m \leq -|n|} \frac{(-1)^{|n|}}{n!m!} r_n^m r_a^m \varphi^{n+m}(z_B - z_A)$$

(12)

where $r_b = x_b - z_B$ and $r_a = x_a - z_A$. Grouping terms, we can express the multipole expansion for a node $A$ (or $B$) as

$$M_n(z_A) = \sum_{a \in A} q_a \frac{(-1)^{|n|}}{n!} r_n^a$$

(13)

Multipoles can be computed efficiently for all parent nodes by combining the multipoles of child nodes using the shifting formula

$$M_n(z + x) = \sum_{|k| \leq |n|} \frac{x^k}{k!} M_{n-k}(z)$$

(14)

If two nodes are sufficiently distant for the expansions to be accurate—as determined by an implementation specific multipole acceptance criteria (MAC)—the multipoles may be transformed into a local expansion (or field tensor) about another point, e.g.,

$$F_n(z_B) = \sum_{|m| \leq p - |n|} M_m(z_A) \nabla^{n+m} \phi(z_B - z_A)$$

(15)

Performing this evaluation on both $A$ and $B$ symmetrically naturally satisfies Newton’s third law.

The final evaluation of the potential proceeds as follows. The field tensors from node-node interactions are accumulated from parent to child down the tree by the shifting formula

$$F_n(z + x) = \sum_{|k| \leq p - |n|} \frac{x^k}{k!} F_{n+k}(z)$$

(16)

such that $F_n$ in the leaf nodes will be the sum of all the field tensors of all parent nodes and any node-node interactions it had itself. The approximated potential (or any $k^{th}$ order derivative) at the positions of particles in a leaf node $B$ is then

$$\nabla^k \phi(x_b) = q_b \sum_{|n| \leq p - k} \frac{1}{n!} (x_b - z_B)^n F_{n+k}(z_B)$$

(17)

C. FMM and induced point dipole moments

An induced point dipole $\mu_i$ can be viewed as resulting from a pair of particles with charges $\pm q_i^p$ located at a distance $|\delta l_i|$ from the dipole center such that

$$\mu_i = 2|q_i^p| \delta l_i$$

(18)

From the above relation, it is straightforward to compute the polarization energy and forces using FMM. With the dipole charge set $\{q_i^p\}$ and by considering $x_i^\pm = x_i \pm \delta l_i$, we can reformulate the potentials in $U^{pol}$ such that they take a similar form as the standard Coulomb potential

$$\Phi_i^\mu = \mu_i \cdot E_i$$

$$= q_i^p \sum_j \phi(x_i^+ - x_j) - \phi(x_i^- - x_j)$$

(19)

and

$$\Phi_i^\mu = \mu_i \cdot E_i^\mu$$

$$= q_i^p \sum_j q_j^i \left[ \phi(x_i^+ - x_j^+) - \phi(x_i^- - x_j^-) \right]$$

$$- \left[ \phi(x_i^+ - x_j^-) - \phi(x_i^- - x_j^+) \right]$$

(20)

Note that if $U^{pol}$ obeys Eq. [8], then the derivatives of $U^{pol}$ w.r.t. the charges $\{q_i^p\}$ and the vectors $\delta l_i$ vanish. Hence, all the formulas discussed in the above section can be used as such to compute the polarization energy and forces. The same $p^{th}$ order expansion is used to estimate equations (19) and (20) as for the Coulomb potential despite $\mu_i$ being related to a second order derivative of the Green’s function. Our later tests suggest this has no incidence on the resulting precision of the simulations.

Moreover, the electric fields in equation (6), which the induced dipole moments obey, are also computable using FMM. The static electric field $E_i^\mu$ can be computed by considering the spatial derivatives of the electrostatic potential acting on atom $i$ and generated by the surrounding charges $q_j$. Likewise, the dynamical electric field $E_i^\mu$ can be computed by considering the spatial derivatives of the electrostatic potential generated on atom $i$ by the surrounding set of charges $q_j^\mu$.

III. IMPLEMENTATION

Here we will describe briefly our implementation of the FMM in our own molecular dynamics code POLARIS(MD) [21, 22, 27]. The implementation is similar to that described in [21], but we account for the Coulomb, dipole-static electric field, and dipole-dipole interactions. Moreover, we also account for the specificities of the polarizable force-fields implemented in the code POLARIS(MD) by allowing only a subset of atoms to generate the static electric field acting on a second subset of polarizable atoms (both these subsets of atoms can be different, and the charges $q_i^p$ generating the electric field can also be different from the Coulomb $q_i$ ones; see Ref. [28]). The atomic polarization tensor $\alpha_i$ is taken to be isotropic and is replaced by a scalar $\alpha_i$.

Regardless of the charge set generating a potential $\Phi$, that potential consists of two components

$$\Phi = \Phi^{direct} + \Phi^{fmm}$$

(21)
The first component is computed via direct particle-particle interactions while the second is computed via FMM. The direct component includes particles that are too close for the multipole expansion to be accurate enough and also for interactions that are cheaper to compute directly than with the expansions. Other effects, such as damping the polarization effects at short range and the atom-atom repulsive potential \( U^{\text{rep}} \) are also computed directly using a list of neighbors when a cell has a "self"-interaction. We explicitly consider the induced dipole moments \( \{ \mu_i \} \) for direct particle-particle interactions and not the \( q_i^p \) and \( \delta_i \) quantities.

The atoms are organized via an adaptive kd-tree spatial decomposition. A binary tree, as opposed to an oct-tree, is particularly well suited for non-uniformly distributed structures, or structures that are not roughly cubic in extent. This is an important feature for simulating a complex solute solvated within a coarse-grained solvent box (see below). For efficiency, we build one tree with all the atoms \( T^{\text{atom}} \) and a second tree with only the polarizable atoms \( T^\mu \).

The tree is constructed by recursively splitting the bounding volume across the longest dimension at the location of the geometric center of the particles within each cell. The recursion stops once a cell has no more than \( N_{\text{bucket}} \) particles, where \( N_{\text{bucket}} \) is a parameter we set to 8. Larger values decrease the size of the tree at the expense of increased direct sum work.

In addition to typical tree book keeping data, each cell stores the tuple \( (z, x_{\text{min}}, x_{\text{max}}, r_{\text{renc}}, M_i, F_i, \alpha_i) \) for each of the three categories of particles: all atoms \( (i = 1) \), atoms generating \( E^q \) \((i = 2)\), and polarizable atoms \((i = 3)\). The tuple contains the geometric center (the mean of the particle positions), the bounds of the tightest enclosing box, and the distance from \( z \) to the most distant corner of the bounding box. This choice of radius allows parent cell sizes and positions to be easily derived from the children. We also store in the tuple a multipole acceptance criteria (MAC) where \( \theta \) defines the opening angle of each particle via the quantity \( |z \cdot z_{\text{A}} - z_{\text{B}}| > (r_{\text{renc}} + r_{\text{renc}}) \) (22).

Unlike the original FMM of Greengard and Rokhlin [4], where \( \theta \) is effectively equal to unity, we do not increase the expansion order \( p \) of the multipoles to gain accuracy. Very recent work by [20] suggests that at \( \theta = 1 \) only small improvements to the precision are gained by large increases in \( p \), while at \( \theta < 1 \) large gains in precision are possible for small changes in \( p \).

Interactions are determined by descending the tree in a dual-walk fashion whereby we maintain a stack of interaction pairs and at each iteration of the walk we consider the interaction of the pair at the top of the stack.

The walk begins with \( A, B \) equal to the root node placed on the stack. If two nodes are equal we consider a direct self interaction. If there are too many particles (> ~64) for direct computation the node is decomposed into its left and right children \( A_l, A_r \) and the new interactions \( A_lA_r, A_lA_l, \) and \( A_rA_r \) are placed on the stack.

If two nodes are not equal then we first consider a direct interaction. If that is rejected, the nodes must pass the MAC. If they satisfy the MAC the field tensors are accumulated for the mutual interaction of the nodes, thus ensuring conservation of momentum. If the MAC is not accepted, the node with the larger \( r_{\text{renc}} \) is opened, unless the node is a leaf. If, for instance, node \( B \) was the larger, the interactions \( AB_t, AB_r, \) and \( B_tB_r \) are placed on the stack. If both nodes are leaves then a direct interaction is performed. The procedure ends when there is no more work left on the stack.

Calculating all the long range forces involves several applications of the above procedure applied in three stages. First, we calculate the coulomb forces \( \nabla \Phi^q \) and static electric field \( E^q \) using \( T^{\text{atom}} \). All particles with non-zero \( q_i \) interact with each other, but only those with non-zero \( q_i^E \) are allowed to contribute to the electric field. The electric field is only evaluated at the positions of polarizable particles (those with non-zero \( \mu_i \)). Second, we iteratively solve for the dipoles \( \mu_i = \alpha_i E_i \) using \( T^\mu \). At each iteration we recalculate the moments and perform an FMM interaction tree walk. The details of the iterative procedure have been described earlier in Ref. [21]. This has been shown to allow large time steps to be used to solve the Newtonian equations of motion in MD simulations [30], as well as multiple time step MD integrators [31]. Finally, with the converged solution for the dipoles, we update the moments for the dipoles stored in \( T^{\text{atom}} \) and compute the dipole-dipole forces as well as the dipole-\( q^E \) forces.

IV. TESTS

To assess the quality of our implementation, both in terms of precision and efficiency, we performed a number of accuracy and performance tests. We compare the force evaluations with those generated by the standard \( O(N^2) \) implementation, which we refer to as a “direct” summation. We measure the relative error in the forces \( a_i \) of each particle via the quantity

\[
\frac{|a_i^{\text{fmm}} - a_i^{\text{dir}}|}{|a_i^{\text{dir}}|}
\]

We chose to compare the force rather than potential as the force is the directly integrated quantity and because the potential is smoother and thus less likely to reveal any computational issues.
Most of the tests were performed by considering different kinds of molecular systems in vacuum. We also performed a series of tests with a protein complex solvated in a coarse grained solvent box. In that case, we use the approach proposed by the second author [21, 22] which doesn’t consider any cut-off for computing the atom-atom interactions within the solute, whereas the solvent-solute and solute-solvent interactions have cut-offs at 12 Å and 7 Å, respectively. The computation of electrostatic solute-solvent interactions thus correspond to a free boundary condition problem, while the algorithmic complexity concerning the solute-solvent and solvent-solvent interactions scales as $O(N_p)$ and $O(N_s)$, where $N_p$ and $N_s$ are the number of protein atoms and solvent pseudo-particles, respectively. As only atoms are involved in long range forces, we will ignore pseudo-particles for discussions regarding the protein.

Finally, we test the scaling properties of our FMM implementation using subsets of the HIV-1 capsid system.

### A. Sphere Tests

We first tested the FMM force accuracy using a randomly generated sphere of atoms. The sphere was generated by uniformly sampling 4096 points inside a radius of 45 Å with a minimum interparticle separation of 2 Å. An equal number of $q = \pm e$ charges were chosen such that the sphere would be charge neutral. Each atom was also assigned a random $q^E = \pm e$ charge for the electric field, but again in equal number to ensure neutrality.

In Figure 1 we show the relative error of the force with respect to the direct evaluation. Polarization effects are disabled so that only the Coulombic forces are considered. Three opening angles $\alpha = 0, 1, 2, 3$ have been assumed for all atoms. The tolerance parameter is fixed at $\theta = 0.5$. We highlight in blue the $(\alpha, \theta) = (0, 0.5)$ case presented in Figure 1. The left vertical axes has been normalized such that the area under each curve is 1. Filled curves using the right vertical axis represent the cumulative histograms.

In Figure 2 we fix $\theta = 0.5$ enable polarization and recompute the forces. The polarization scalar for all atoms is taken to be $\alpha = \{0, 1, 2, 3\}$, for each test, respectively. In the case of $\alpha = 0$ the test is equivalent to that shown previously in Figure 1. In all cases there is no significant change in the error distribution.

We also considered the effect of our choice of $|\delta I|$, the separation distance for the point dipole approximation. Even though we tested values over 5 orders of magnitude, from $10^{-5}$ to $10^{-1}$ Å, as shown in Figure 3, we again found no difference in the errors.

A direct comparison of these parameters with other implementations is difficult as many small details can affect the results. The most robust measure is to demand a maximum error tolerance. We therefore chose for all of our future tests the parameters $\theta = 0.5, p = 5, |\delta I| = 0.0001$ Å to ensure a maximum error of $10^{-2}$.

![Figure 1](image1.png)

**FIG. 1.** The relative FMM error of the force on each atom of the random sphere compared with a direct evaluation. Polarization is disabled. Three values of the tolerance parameter $\theta = 0.5, 0.7, 0.99$ are considered, from very accurate to less accurate. The left vertical axes has been normalized such that the area under each curve is 1. Filled curves using the right vertical axis represent the cumulative histograms.

![Figure 2](image2.png)

**FIG. 2.** The relative FMM error of the force on each atom of the random sphere compared with a direct evaluation. Polarization is enabled with where the polarization scalar $\alpha \in \{0, 1, 2, 3\}$ has been assumed for all atoms. The tolerance parameter is fixed at $\theta = 0.5$. We highlight in blue the $(\alpha, \theta) = (0, 0.5)$ case presented in Figure 1. The left vertical axes has been normalized such that the area under each curve is 1. Filled curves on the right vertical axis represent the cumulative histograms.

### B. Molecular dynamics of a solvated protein complex

We used the Pancreatic Trypsin Inhibitor-Trypsin complex (PDB [32] label 2PTC [33]) to test the stability and accuracy of the FMM method used in conjunction
FIG. 3. Force error distribution of FMM for the spherical test case without polarization considering different values of the dipole separation distance $\delta l = |\delta l|$. Our choice for this approximation makes little difference over 5 orders of magnitude. Here we assume an opening angle of $\theta = 0.5$. We highlight in green the $(\alpha, \delta l) = (1, 0.0001)$ case presented in Figure 2. The left vertical axes has been normalized such that the area under each curve is 1. Filled curves using the right vertical axis represent the cumulative histograms.

FIG. 4. The Pancreatic Trypsin Inhibitor-Trypsin complex (4,114 atoms) immersed in a pseudo-particle solvent (28,258 particles) non-periodic box. We perform six 1 ns simulations of the system to verify our FMM implementation.

with three MD schemes over 1 ns. The protein complex is made of 2 non-bonded proteins and consists of 4,114 atoms [34]. We immersed the system in a single level solvent box made of 28,258 polarizable pseudo-particles (see Figure 3), according to the procedure described in Ref. 22. As already mentioned, the computations concerning the electrostatic interactions within the protein complex were performed in a non-periodic environment without any cut-off for the atom-atom interactions within the solute and cut-offs at 12 Å and 7 Å for the solvent-solvent and solute-solvent interactions, respectively.

Three different kinds of simulations were performed using both the direct and FFM approaches by solving the Newtonian equations of motion using the velocity Verlet integrator with two time steps to handle non-bonded interactions, namely 1 fs and 2 fs, and the multiple time step integrator devoted to induced dipole moment based potentials [35]. In the latter case, we use a time step of 1 fs for short range non-bonded interactions and 5 fs for long range ones. To handle the quickly varying energy terms (those handling bonded atom interactions), we systematically used a time step of 0.25 fs, regardless of the MD integrator. Chemical bonds X-H and bending angles H-X-H were constrained to their initial values using the RATTLE algorithm [36] with a convergence criterion of $10^{-6}$ Å. The induced dipole moments were solved iteratively with a convergence criterion of $10^{-6}$ Debye per polarizable center. However, the iterations continue until the greatest difference between two successive iterations of the induced dipole moment for a single polarizable center is smaller than $20 \times 10^{-6}$ Debye. A temperature of 300 K was monitored along the trajectories using the GGMT thermostat [37] (with a coupling constant of 0.5 ps). The system center of mass kinetic energy was subtracted before a simulation begins. Thus, a priori, we expect there to be no drift in the center of mass. Lastly, based on the above spherical tests we use the FMM parameters $\theta = 0.5, p = 5, |\delta l| = 0.0001$ Å. All the systems were equilibrated by performing 100 ps runs before starting the 1 ns production runs.

The number of iterations needed to converge the dipole moments along the trajectories were the same for each integrator when using FMM or direct summation, within less than 0.1 iterations on average. The difference originates mainly from the different portions of the system potential energy surface explored along the trajectories due to the chaotic effects affecting MD simulations.

The energy plots from running the 6 test cases for 1 ns are presented in the upper plot of Figure 5. The instantaneous total energy has been adjusted by the initial energy $E_0$ and then scaled by the total injected kinetic energy $E_{\text{kin}}^{\text{inject}}$ over the course of the simulation (here $E_{\text{inject}}^{\text{kin}} = 63,477$ kcal/mol). The initial 100 ps used to equilibrate the system is not shown in the figure.

Because the dipoles are solved iteratively, a small energy drift is expected [38] and at the ns scale we observed this mainly for the MTS runs. However, the drifts are small and all clearly under 0.1%. Most importantly, all the FMM runs behave consistently in terms of energy conservation when compared with the direct summation based runs, exhibiting the high precision level of
FIG. 5. (Top) The energy drift as a fraction of the total injected kinetic energy $E_{\text{kin, inject}}$. For the 2PTC protein simulation using fixed time step and multiple time step integrators $E_{\text{kin, inject}} = 63477$ kcal/mol. The general behavior using FMM is similar to that of the direct sum computation. (Bottom) The kinetic energy $T_{\text{cm}}$ of the system center of mass with FMM, expressed in K. The small, nearly constant value for both fixed and multiple time steps demonstrates the momentum conserving nature of the implementation. The 1 fs and 2 fs data has been offset vertically from the MTS run for clarity.

the present FMM implementation. There may be a small inaccuracies due to the discontinuity introduced when a particle crosses a cell boundary and transitions from a direct interaction to an FMM one, however, these are largely dominated by errors in the iterative dipole scheme and do not manifest themselves in the final results.

The lower plot of the same figure highlights the conservation of momentum of the full system center of mass. There is essentially no change in the value over the length of all simulations, thus exhibiting the momentum conservation ability of the present FMM approach.

C. FFM scaling

We performed scaling tests using a series of subset structures of the mature HIV-1 capsid full system recently investigated in liquid water using a PME approach and a non-polarizable standard pairwise force field [8]. The molecular structures considered here were extracted from the PDB files 1VU{4–9}. Our base capsid subsys-

FIG. 6. The relative scaling of FMM (normalized by the smallest system) applied to a single energy point calculation of the HIV capid subsystem for increasing system size (blue triangles, left axis). We also show the speed up of FMM over a direct summation computation (grey triangles, right axis). Despite lacking vectorization optimizations, FMM is more than 6x faster for the largest system.

tem is an hexamer of the HIV-1 capsid protein. That subsystem contains 21,612 atoms (including hydrogen). Larger capsid subsystems were generated by replicating this base system up to a 5x larger system. Each base hexamer system interacts with the others according to the interaction scheme of the HIV-1 capsid. The largest system considered is thus made of about $10^5$ atoms. All computations were performed in vacuum and represent a single energy point computation using the same protocol to solve the dipole moments as for 2PTC and including all short range forces in addition to the long range Coulomb and polarization interactions. The calculations were performed on a single core Xeon E5-2680 2.7GHz CPU.

The results of the scaling tests are displayed in Figure 6. The blue triangles show the relative scaling of the FMM with respect to the the smallest system. The grey triangles show the speed up of the FMM over a direct summation method. Small fluctuations in the scaling are due to minor differences in the number of dipole iterations. It must also be emphasized that the FMM implementation does not yet include any vectorization optimizations, which the direct sum approach is currently able to benefit from. Despite this, FMM is over 6x faster for the largest system.

V. CONCLUSIONS

We have implemented a momentum conserving version of the fast multipole method in the molecular dynamics code Polaris(MD) to compute the Coulomb
forces as well as polarization interactions associated with the induced dipole polarization method in $O(N)$ time. Through a series of artificial and real-world test cases—most notably recent HIV-1 capsid subsystems from [8]—we have shown we are able to achieve very good accuracy by combining a multipole expansion to 5th order and an opening angle tolerance criteria of $\theta = 0.5$. During 1 ns MD simulations of the 2PTC protein, we observe a small energy drift of less than 0.1% of the total kinetic energy injected in the system and observe no change in the total momentum—consistent with standard $O(N^2)$ summation-based simulations.

The next three major steps in the development of the present FFM approach will be to 1) develop an efficient parallel scheme using the new features of the MPI v3 specification; 2) explore existing alternatives to the Cartesian expansions to reduce the number of operations required by the multipole evaluation (e.g., Ref. [39] and personal communications with Walter Dehnen and Joachim Stadel); and 3) adapt the POLARIS(MD) code to modern and forthcoming computational systems (in particular, multi-threaded architectures with overall reduced memory resources). While very technical in nature, these three goals are critical for taking advantage of the latest advances in computing and will pave the way for tackling many of the large scale challenges of the future.

ACKNOWLEDGMENTS

We thank Othman Bouizi (Exascale Computing Research Laboratory, a joint INTEL/CEA/UVSQ/GENCI laboratory) for his help in developing the code POLARIS(MD). This work was granted access to the HPC resources of [CCRT/CINES/IDRIS] under the allocation 2014-[6100] by GENCI (Grand Equipement National de Calcul Intensif). We acknowledge access to the Stampede supercomputing system at TACC/UT Austin, funded by NSF award OCI-1134872.

Appendix A: Additional 2PTC Tests

In Figure 7 we show the distribution of errors after a single energy calculation. The distribution is remarkably similar the spherical tests with a peak in log-space just below $10^{-4}$. In addition, we present in Figure 8 the temporal evolution of the system potential energy $E_{pot}$ over the course of the 1 ns simulations. An initial 100 ps was used to relax the system and is not shown in the figure. The relaxation phase is itself chaotic and will result in slightly different initial conditions for each run. Therefore, this plot shows that different portions of the system potential energy surface are explored along each trajectory, and explains the difference in the number of iterations needed to converge the dipole in direct sum and FMM simulations. The original data has been smoothed with a Hanning filter for clarity and to reduce the noise.

Appendix B: Generating the multipole expansions

Computing multipoles to order $p = 5$ is a non-trivial task to perform by hand. We have instead implemented a program in Python [40] that symbolically generates the code for all FMM operations. The equations are then symbolically simplified using the SymPy library [41]. By

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig7.png}
\caption{Force error distribution of FMM after one time step of the 2PTC protein complex compared with a direct sum calculation. The left vertical axes has been normalized such that the area under each curve is 1. The filled curve using the right vertical axis represents the cumulative histogram.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig8.png}
\caption{The potential energy $E_{pot}$ for the six 1 ns simulations of the 2PTC protein complex. The drift is due to the presence of a thermostat and chaotic effects in the MD trajectories. The original data has been smoothed with a Hanning filter for clarity and to reduce the noise.}
\end{figure}
eliminating common subexpressions we see up to a ten fold decrease in operation count, although scaling still behaves as $O(p^4)$.

**Appendix C: Multi-index notation**

The multi-index notation simplifies the tensor representation in the derivation of the FMM operations. Following from property (C1) addition and subtraction operations are only valid for those tuples that have all $n_i \geq 0$.

\[
\mathbf{n} = (n_x, n_y, n_z) \text{ where } n_i \geq 0 \quad (C1)
\]

\[
|\mathbf{n}| \equiv n_x + n_y + n_z 
\]

\[
|\mathbf{n}|! \equiv n_x! \cdot n_y! \cdot n_z! 
\]

\[
n \pm k \equiv (n_x \pm k, n_y \pm k, n_z \pm k) \quad (C4)
\]

\[
r^n \equiv r_x^{n_x} \cdot r_y^{n_y} \cdot r_z^{n_z} 
\]

\[n \equiv \prod_{i=1}^{3} x_i^{n_i} \quad (C2)
\]

\[n! \equiv \prod_{i=1}^{3} n_i! \quad (C3)
\]

\[
(C5)
\]

\[
(C6)
\]

[1] J. W. Perram, H. G. Petersen, and S. W. De Leeuw, Molecular Physics 65, 875 (1988). http://dx.doi.org/10.1080/0026897880101471

[2] T. Darden, D. York, and L. Pedersen, Jcp 98, 10089 (1993)

[3] J. Barnes and P. Hut, Nature (London) 324, 446 (1986).

[4] L. Greengard and V. Rokhlin, Journal of Computational Physics 73, 325 (1987)

[5] T. C. Bishop, R. D. Skeel, and K. Schulten, Journal of Computational Chemistry 18, 1785 (1997).

[6] E. Figueirido, R. M. Levy, R. Zhou, and B. J. Berne, The Journal of Chemical Physics 106, 9835 (1997).

[7] C. Sagui and T. A. Darden, Annual Review of Biophysics and Biomolecular Structure 28, 155 (1999) PMID: 10410799.

[8] G. Zhao, J. R. Perilla, E. L. Yufenyuy, X. Meng, B. Chen, J. Ning, J. Ahn, A. M. Gronenborn, K. Schulten, C. Aiken, and P. Zhang, Nature 497, 541 (2002).

[9] A. Arnold, F. Fahrenberger, C. Holm, O. Lenz, M. Bolten, H. Ducheel, R. Halver, I. Kabashow, F. Gähler, F. Heber, J. Iseringhausen, M. Hofmann, M. Pippig, D. Potts, and G. Sutmann, Phys. Rev. E 88, 063308 (2013).

[10] R. Yokota and L. A. Barba, Int. J. High Perform. Comput. Appl. 26, 337 (2012).

[11] J. Diemand, M. Kuhlen, P. Madau, M. Zemp, B. Moore, D. Potter, and J. Stadel, Nature (London) 454, 735 (2008) arXiv:0805.1244.

[12] J. Stadel, D. Potter, B. Moore, J. Diemand, P. Madau, M. Zemp, M. Kuhlen, and V. Quilis, mnras 398, L21 (2009) arXiv:0808.2981.

[13] L. Dang, J. Chem. Phys. 113, 266 (2000).

[14] P. Lopes, B. Roux, and A. MacKerell Jr, Theor. Chem. Acc. 124, 11 (2009).

[15] P. Cieplak, F.-Y. Dupradeau, Y. Duan, and J. Wang, J. Phys.: Condens. Matter 21, 333102 (2009).

[16] T.-M. Chang and L. X. Dang, Chem. Rev. 106, 1305 (2006).

[17] P. Jungwirth and D. J. Tobias, Chem. Rev. 106, 1259 (2006).

[18] F. Réal, M. Trumm, Schimmelpfennig, M. Masella, and V. Vallet, J. Comput. Chem. 34, 707 (2013).

[19] M. Salanne and P. Madden, Mol. Phys. 109, 2299 (2011).

[20] D. Scarpazza, D. Ierardi, A. Lerer, K. Mackenzie, A. Pan, J. Bank, E. Chow, R. Dror, J. Grossman, D. Killebrew, M. A. Moraes, C. Predescu, J. Salmon, and D. Shaw, IEEE 27th International Symposium on Parallel and Distributed Processing 35, 933 (2013).

[21] M. Masella, D. Borgis, and P. Cuniasse, Journal of Computational Chemistry 29, 1707 (2008).

[22] M. Masella, D. Borgis, and P. Cuniasse, Journal of Computational Chemistry 32, 2064 (2008).

[23] M. Masella, D. Borgis, and P. Cuniasse, Journal of Computational Chemistry 34, 1112 (2013).

[24] W. Dehnen, Journal of Computational Physics 179, 27 (2002) astro-ph/020512.

[25] B. T. Thole, Chemical Physics 59, 341 (1981).

[26] T. Hadioung, S. Phan, M. Marchi, and D. Borgis, J. Chem. Phys. 117, 541 (2002).

[27] M. Masella, D. Borgis, and P. Cuniasse, Journal of Computational Chemistry 34, 1112 (2013).

[28] M. Masella and P. Cuniasse, The Journal of Chemical Physics 119, 1866 (2003).

[29] W. Dehnen, ArXiv e-prints (2014), arXiv:1405.2255 astro-ph.IM.

[30] W. Wang and R. D. Skeel, The Journal of Chemical Physics 123, 164107 (2005).

[31] M. Masella, Molecular Physics 104, 415 (2006).

[32] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, Nucleic Acids Research 28, 235 (2000).

[33] M. Marquart, J. Walter, J. Deisenhofer, W. Bode, and R. Huber, Acta Crystallographica Section B 39, 480 (1983).

[34] M. Marquart, J. Walter, J. Deisenhofer, W. Bode, and R. Huber, Acta Crystallogogr. Sect.B 39, 480 (1983).

[35] M. Masella, Molecular Physics 104, 415 (2006).

[36] H. C. Andersen, Journal of Computational Physics 52, 24 (1983).

[37] Y. Liu and M. E. Tuckerman, Jcp 112, 1685 (2000).

[38] A. Toukmaji, C. Sagui, J. Board, and T. Darden, J. Comput. Chem. 34, 1685 (2000).

[39] H. Cheng, L. Greengard, and V. Rokhlin, J. Comput. Phys. 155, 468 (1999).

[40] G. Rossum, Python Reference Manual Tech. Rep. (Amsterdam, The Netherlands, The Netherlands, 1995).

[41] SymPy Development Team, SymPy: Python library for symbolic mathematics (2014).