Title

"New-version-fast-multipole-method" accelerated electrostatic calculations in biomolecular systems

Authors

Lu, Benzhuo

Cheng, Xiaolin

McCammon, J. Andrew

Publication Date

2007-10-01

Peer reviewed

Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics xxx (2007) xxx–xxx
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

"New-version-fast-multipole-method" accelerated electrostatic interactions in biomolecular systems

Benzhuo Lu [a,*], Xiaolin Cheng [a], J. Andrew McCammon [b]

[a] *Howard Hughes Medical Institute, Center for Theoretical Biological Physics, University of California at San Diego, La Jolla, CA 92093-0365, United States*
[b] *Department of Chemistry and Biochemistry, Center for Theoretical Biological Physics, Department of Pharmacology, Howard Hughes Medical Institute, University of California at San Diego, La Jolla, CA 92093-0365, United States*

Received 15 January 2007; received in revised form 14 May 2007; accepted 16 May 2007

Abstract

In this paper, we present an efficient and accurate numerical algorithm for calculating the electrostatic interactions in biomolecular systems. In our scheme, a boundary integral equation (BIE) approach is applied to discretize the linearized Poisson–Boltzmann (PB) equation. The resulting integral formulas are well conditioned for single molecule cases as well as for systems with more than one macromolecule, and are solved efficiently using Krylov subspace based iterative methods such as generalized minimal residual (GMRES) or biconjugate gradients stabilized (BiCGStab) methods. In each iteration, the convolution type matrix–vector multiplications are accelerated by a new version of the fast multipole method (FMM). The implemented algorithm is asymptotically optimal O(N) both in CPU time and memory usage with optimized prefactors. Our approach enhances the present computational ability to treat electrostatics of large scale systems in protein–protein interactions and nano particle assembly processes. Applications including calculating the electrostatics of the nicotinic acetylcholine receptor (nAChR) and interactions between protein Sso7d and DNA are presented.

Keywords: Poisson–Boltzmann electrostatics; Boundary integral equation; New version fast multipole method; Order N; Biomolecular system

1. Introduction

In this paper, we discuss a numerical algorithm for the calculation of electrostatic interactions in biomolecular systems in solution, in which the solvent has a substantial volume with numerous mobile ions making significant contributions. Such systems are commonly encountered in biophysics, electrochemistry, and electrophoresis, such as in the study of protein–protein interactions and nano particle assembly processes in drug design and structural biology.

* Corresponding author. Tel.: +1 858 822 0168; fax: +1 858 5344974.
E-mail addresses: blu@mccammon.ucsd.edu (B. Lu), xcheng@mccammon.ucsd.edu (X. Cheng), jmccammon@ucsd.edu (J. Andrew McCammon).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.05.026

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

Although the continuum Poisson–Boltzmann (PB) equation model for these systems was introduced almost a century ago by Debye and Huckel [17] and later developed by Kirkwood [26], its numerical solutions have only been extensively explored in the last two decades [13,16,19,23,24,27,40,46,48]. Traditional schemes include the finite difference methods where difference approximations are used on structured grids describing the computational domain, and finite element methods in which arbitrarily shaped biomolecules are discretized using elements and associated basis functions. The resulting algebraic systems for both are commonly solved using multigrid or domain decomposition accelerations for optimal efficiency. However, as the grid number (and thus the storage, number of operations, and condition number of the system) increases proportionally to the volume size, finite difference and finite element methods become less efficient and accurate for systems of large spatial sizes commonly encountered in studying either macromolecules or interacting systems in the association and dissociation processes. Alternative methods include the boundary element (BEM) and boundary integral equation (BIE) methods. In these methods, only the surfaces (compared to the 3D volume) of the molecules are discretized and hence the number of unknowns is greatly reduced. In addition, the boundary elements (BEs) form a kind of surface "conforming" mesh (because they align with the surface), which therefore allows the application of the BEM to biomolecules characterized by irregular geometries, while maintaining a high level of calculation accuracy.

However, in practical biomodeling, the BEM is the least used relative to the other methods. In earlier versions of BEM, the integral equation formulations may not have been well-conditioned and the matrix was typically stored explicitly. The resulting dense linear system was often solved using direct matrix inversion such as Gauss elimination or LU decompsition, so that $O(N^2)$ storage and $O(N^3)$ operations were required, where N is the number of unknowns defined on the surface. This is extremely inefficient for any typical size system of interest. To improve the BEM efficiency, some later studies improved the condition of the integral formulation [2,24,29,30], reduced the number of the boundary elements [44] or introduced novel BEM [31]. It has been demonstrated that when the system is well-conditioned or can be effectively preconditioned, the matrix equations can be solved efficiently using iterative Krylov subspace methods which are matrix-implicit, thus eliminating the bottleneck of storage. As the number of iterations in these methods is independent of the system size for well conditioned systems, the computational cost is then dominated by the matrix–vector multiplication calculations corresponding to the N-body electrostatic particle interactions of both the Coulombic ($\kappa = 0$) and screened Coulombic ($\kappa \neq 0$) types, which require $O(N^2)$ operations using direct methods for each iteration. By introducing novel fast summation algorithms developed in the last twenty years, this cost has been reduced to $O(N \log N)$. These algorithms include the hierarchical "tree code" [4,6], fast Fourier transform (FFT) based algorithms such as the precorrected FFT (pFFT) [3,28,35] and the particle-mesh Ewald (PME) methods [15], the hierarchical SVD method [25], and FFT on multipoles [33,34]. Further improvements show that asymptotically optimal $O(N)$ complexity can be achieved by using the wavelet techniques [41,43] or the fast multipole method (FMM) [20]. FMM algorithms for the screened Coulombic interaction (Yukawa potential) have also been recently developed [9,22], which allows their direct application to the solution of the PB equation. The tree code algorithm and the FMMs based on the old scheme [20] have been implemented in former BEM PB work [7,8,10,18,34,49]. However, as revealed by previous numerical experiments, although asymptotically optimal, the original FMM [20] turns out to be less efficient for problem sizes of current interest when compared with the tree code and FFT based $O(N \log N)$ techniques, due to the huge prefactor in $O(N)$ [18].

To further accelerate the numerical solution of the PB equation, in this paper, we present an efficient algorithm using a well conditioned BIE formulation, for which the solution is accelerated by a new version of FMM first introduced by Greengard and Rokhlin [21] for the Laplace equation. By proper coupling of single and double layer potentials as in Ref. [38], a Fredholm second kind integral equation formulation for the PB equation can be derived. Similar formulations were first introduced by Juffer et al. [24] who aimed to avoid the singularity problem in deriving the complete BIE forms for the linearized PB equation. The well-conditioned property has been discussed by Liang and Subramaniam [29] and also demonstrated in the work of Boschitsch et al. [10]. We extend the formulation for systems with an arbitrary number of domains. Compared with traditional BEM formulations, the condition number of our BIE system does not increase with the number of unknowns, hence the number of iterations in the Krylov subspace based methods is bounded. For the matrix vector multiplication in each iteration, we use the new version FMM developed for the screened Coulombic

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

interaction (Yukawa potential) [22]. Compared with the original FMM, the plane wave expansion based diagonal translation operators dramatically reduce the prefactor in the O(N) new version FMM, especially in three dimensions where a break-even point of approximately 600 for 6 digits precision is numerically observed. Perhaps due to its complexity in theory and programming, we are unaware of any previous implementations of the new version FMM for the PB equation.

This paper is organized as follows. In Section 2, we discuss the methodological details of our algorithm, including the well-conditioned integral equation formulation and the new version of FMM. In Section 3, the resulting algorithm is applied to both a single protein and a two-molecule (protein–DNA) interaction cases to illustrate the performance of the algorithm. Finally in Section 4, conclusions and discussion are presented. We want to mention that some of the analyses were briefly reported previously in Ref. [30]. The purpose of this paper is to present more technical details along with two new applications.

2. Methods

In this section, we discuss the technical details of the algorithm. In particular, the boundary integral equation formulation and its discretization, the new version of FMM and how it is applied to the BEM, the Krylov subspace method, the mesh generation, and the system set up are discussed in order.

2.1. Boundary integral equation formulations

When Green's second identity is applied, traditional boundary integral equations for the linearized PB equation for a single domain (molecule) take the form

$$\frac{1}{2}\phi_p^{\text{int}} = \overset{\text{PV}}{\oint_S}\left[G_{pt}\frac{\partial\phi_t^{\text{int}}}{\partial n} - \frac{\partial G_{pt}}{\partial n}\phi_t^{\text{int}}\right]\mathrm{d}S_t + \frac{1}{D_{\text{int}}}\sum_k q_k G_{pk}, \quad p \in S, \tag{1}$$

$$\frac{1}{2}\phi_p^{\text{ext}} = \overset{\text{PV}}{\oint_S}\left[-u_{pt}\frac{\partial\phi_t^{\text{ext}}}{\partial n} + \frac{\partial u_{pt}}{\partial n}\phi_t^{\text{ext}}\right]\mathrm{d}S_t, \quad p \in S, \tag{2}$$

where ϕ_p^{int} is the interior potential at surface position p of the molecular domain Ω, $S = \partial\Omega$ is its boundary, i.e., solvent-accessible surface, ϕ_p^{ext} is the exterior potential at position p, D_{int} is the interior dielectric constant, t is an arbitrary point on the boundary, n is the outward normal vector at t, and PV represents the principal value integral to avoid the singular point when $t \to p$ in the integral equations. In the formulae, $G_{pt} = \frac{1}{4\pi|r_t - r_p|}$ and $u_{pt} = \frac{\exp(-\kappa|r_t - r_p|)}{4\pi|r_t - r_p|}$ are the fundamental solutions of the corresponding Poisson and Poisson–Boltzmann equations, respectively, r_k is the position of the kth source point charge q_k of the molecule, κ is the reciprocal of the Debye–Huckel screening length determined by the ionic strength of the solution. These equations can be easily extended to multi-domain systems in which Eq. (1) is enforced for each individual domain and the integration domain in Eq. (2) includes the collection of all boundaries [32].

To complete the system, the solutions in the interior (Eq. (1)) and exterior (Eq. (2)) are matched by the boundary conditions $\phi^{\text{int}} = \phi^{\text{ext}}$ and $D_{\text{int}}\frac{\partial\phi^{\text{int}}}{\partial n} = D_{\text{ext}}\frac{\partial\phi^{\text{ext}}}{\partial n}$, where D_{ext} is the exterior (solvent) dielectric constant. Using these conditions, we can define $f = \phi^{\text{ext}}$ and $h = \frac{\partial\phi^{\text{ext}}}{\partial n}$ as the new unknowns and recover other quantities using boundary integrals of f and h. Unfortunately, theoretical analysis shows that the corresponding equation system for f and h is in general a Fredholm integral equation of first kind and hence ill-conditioned. i.e., when solved iteratively using Krylov subspace methods, the number of iterations increases with the number of unknowns, and the resulting algorithm becomes inefficient for large systems. Instead of this "direct formulation", Rokhlin [38] introduced a technique where the single and double layer potentials are combined in order to derive an optimized second kind Fredholm integral equation. We want to mention that a well-conditioned form actually appeared in Juffer et al.'s work [24] when they tried to derive the complete BI form for linearized PBE using a limiting process to avoid the singularity problem. The same form has been used and discussed in later BEM PB work [10,29], and similar techniques have also been applied and discussed in engineering computations [42].

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

The derivative BEM (dBEM) can be obtained by linearly combining the derivative forms of Eqs. (1) and (2):

$$\left(\frac{1}{2\varepsilon}+\frac{1}{2}\right)f_p = \overset{\text{PV}}{\oint_S}\left[(G_{pt}-u_{pt})h_t-\left(\frac{1}{\varepsilon}\frac{\partial G_{pt}}{\partial n}-\frac{\partial u_{pt}}{\partial n}\right)f_t\right]\mathrm{d}S_t+\frac{1}{D_{\text{ext}}}\sum_k q_k G_{pk},\quad p\in S, \tag{3}$$

$$\left(\frac{1}{2}+\frac{1}{2\varepsilon}\right)h_p = \overset{\text{PV}}{\oint_S}\left[\left(\frac{\partial G_{pt}}{\partial n_0}-\frac{1}{\varepsilon}\frac{\partial u_{pt}}{\partial n_0}\right)h_t-\frac{1}{\varepsilon}\left(\frac{\partial^2 G_{pt}}{\partial n_0\partial n}-\frac{\partial^2 u_{pt}}{\partial n_0\partial n}\right)f_t\right]\mathrm{d}S_t+\frac{1}{D_{\text{ext}}}\sum_k q_k\frac{\partial G_{pk}}{\partial n_0},\quad p\in S. \tag{4}$$

where n_0 is the unit normal vector at point p, $\varepsilon = D_{\text{ext}}/D_{\text{int}}$. This set of BIEs leads to a well-conditioned system of algebraic equations, which we will adopt.

For a system with an arbitrary number, e.g. J, of separate domains (molecules) surrounded by infinite homogeneous solvent, Eq. (1) holds and the integration can be performed only over one molecular surface where the evaluation point p is located, while the integrand in Eq. (2) is the combination of all the molecular surfaces. Following the same treatment, and supposing $p\in S^i$, the derivative BIEs for multiple domains are extended as:

$$\begin{aligned}\left(\frac{1}{2\varepsilon}+\frac{1}{2}\right)f_p = {}& \overset{\text{PV}}{\oint_{S^i}}\left[(G_{pt}-u_{pt})h_t-\left(\frac{1}{\varepsilon}\frac{\partial G_{pt}}{\partial n}-\frac{\partial u_{pt}}{\partial n}\right)f_t\right]\mathrm{d}S_t\\ &+\sum_{j\neq i}\oint_{S^j}\left[-u_{pt}h_t+\frac{\partial u_{pt}}{\partial n}f_t\right]\mathrm{d}S_t+\frac{1}{D_{\text{ext}}}\sum_{k^i} q_{k^i}G_{pk^i},\quad p\in S^i,\ \ i=1,\ldots J,\end{aligned} \tag{5}$$

$$\begin{aligned}\left(\frac{1}{2}+\frac{1}{2\varepsilon}\right)h_p = {}& \overset{\text{PV}}{\oint_{S^i}}\left[\left(\frac{\partial G_{pt}}{\partial n_0}-\frac{1}{\varepsilon}\frac{\partial u_{pt}}{\partial n_0}\right)h_t-\frac{1}{\varepsilon}\left(\frac{\partial^2 G_{pt}}{\partial n_0\partial n}-\frac{\partial^2 u_{pt}}{\partial n_0\partial n}\right)f_t\right]\mathrm{d}S_t\\ &+\sum_{j\neq i}\oint_{S^j}\frac{1}{\varepsilon}\left[-\frac{\partial u_{pt}}{\partial n_0}h_t+\frac{\partial^2 u_{pt}}{\partial n_0\partial n}f_t\right]\mathrm{d}S_t+\frac{1}{D_{\text{ext}}\varepsilon}\sum_{k^i} q_{k^i}\frac{\partial G_{pk^i}}{\partial n_0},\quad p\in S^i,\ \ i=1,\ldots J.\end{aligned} \tag{6}$$

However, it is noticed that in Eqs. (5) and (6) the integrand kernels for integration on surface S^i (enclosed in the first pair of square brackets) are not the same as those on molecular surface S^j (enclosed in the second pair of square brackets). This is not convenient for application of the FMM. The FMM algorithm uses hierarchical levels of boxes to group all the evaluation points (meshes), so it would be beneficial to have similar integral formulae on all molecular surfaces for every evaluation point. If we apply Green's second theorem to domain S^j, and still let $p\in S^i, i\neq j$, it is found that the following set of equations hold

$$0=\oint_{S^j}\left[G_{pt}h_t-\frac{1}{\varepsilon}\frac{\partial G_{pt}}{\partial n}f_t\right]\mathrm{d}S_t+\frac{1}{D_{\text{ext}}}\sum_{k^j} q_{k^j}G_{pk^j},\quad p\in S^i,\ \ i=1,\ldots J, \tag{7}$$

$$0=\oint_{S^j}\left[\frac{\partial G_{pt}}{\partial n_0}h_t-\frac{1}{\varepsilon}\frac{\partial^2 G_{pt}}{\partial n_0\partial n}f_t\right]\mathrm{d}S_t+\frac{1}{D_{\text{ext}}}\sum_{k^j} q_{k^j}\frac{\partial G_{pk^j}}{\partial n_0},\quad p\in S^i,\ \ i=1,\ldots J. \tag{8}$$

Combining these equations for different boundary j, it is found that Eqs. (5) and (6) have another neat form

$$\begin{aligned}\left(\frac{1}{2\varepsilon}+\frac{1}{2}\right)f_p = {}& \sum_j^J\int_{S^j}^{\text{PV}}\left[(G_{pt}-u_{pt})h_t-\left(\frac{1}{\varepsilon}\frac{\partial G_{pt}}{\partial n}-\frac{\partial u_{pt}}{\partial n}\right)f_t\right]\mathrm{d}S_t\\ &+\frac{1}{D_{\text{ext}}}\sum_j\sum_{k^j} q_{k^j}G_{pk^j},\quad p\in S^i,\ \ i=1,\ldots J,\end{aligned} \tag{9}$$

$$\begin{aligned}\left(\frac{1}{2}+\frac{1}{2\varepsilon}\right)h_p = {}& \sum_j^J\int_{S^j}^{\text{PV}}\left[\left(\frac{\partial G_{pt}}{\partial n_0}-\frac{1}{\varepsilon}\frac{\partial u_{pt}}{\partial n_0}\right)h_t-\frac{1}{\varepsilon}\left(\frac{\partial^2 G_{pt}}{\partial n_0\partial n}-\frac{\partial^2 u_{pt}}{\partial n_0\partial n}\right)f_t\right]\mathrm{d}S_t\\ &+\frac{1}{D_{\text{ext}}}\sum_j^J\sum_{k^j} q_{k^j}\frac{\partial G_{pk^j}}{\partial n_0},\quad p\in S^i,\ \ i=1,\ldots J.\end{aligned} \tag{10}$$

Now, all the calculated points can treated uniformly by this set of equations, which is similar to the case of one molecule. This is a set of well-conditioned Fredholm second kind integral equation formulations for multi-biomolecule systems. As a matter of fact, it can be more straightforward to obtain the derivative BIEs for multi-domain cases from the single domain equations, because Eqs. (1)–(4) hold not only for a single closed boundary surface, but also for any combination of separated boundaries. Compared with Eqs. (5) and (6), Eqs. (9) and (10) add more operations in the integrals and summations. But these additional operations only account for a very small part of the whole computational cost for solving the PB equation, and the summations in Eqs. (9) and (10) are also efficiently accelerated by using the FMM. In addition, as mentioned above, the symmetrized integral formulations also make the coding convenient and easy-to-maintain. It is worth noting that for the case when the interior dielectric constants are different for different molecular domains, and are same in exterior domain, a set of formulae very similar to Eqs. (9) and (10) are still available, except for that the coefficient ε is to be replaced by ε_j because it varies for different molecular surface integrals. In this case, the FMM still applies because the Green's functions are the same, but it needs to separate the terms associated with ε_j and rescale f and h to absorb ε_j on different molecular surfaces, then use the FMM. The case with different dielectric constants was studied in a recent BEM paper [47].

2.2. Discretization of the BIEs

Similar to our former work [32], the discretized form of the BIEs (9) and (10) can be written as

$$\left(\frac{1}{2\varepsilon}+\frac{1}{2}\right)f_p = \sum_t^T \left(A_{pt}h_t - B_{pt}f_t\right) + \frac{1}{D_{\text{ext}}}\sum_k q_k G_{pk}, \tag{11}$$

$$\left(\frac{1}{2}+\frac{1}{2\varepsilon}\right)h_p = \sum_t^T \left(C_{pt}h_t - D_{pt}f_t\right) + \frac{1}{D_{\text{ext}}}\sum_k q_k \frac{\partial G_{pk}}{\partial n_0}, \tag{12}$$

where T is the total number of discretized patches of the combined boundaries, which is half of the total unknowns (f or h) of the system, and here the $\sum_k$ encompasses all the source charges in the considered system. The coefficient matrices are defined as follows:

$$\begin{aligned} A_{pt} &= \int_{\Delta S_t} (G_{pt} - u_{pt})\mathrm{d}S, \quad B_{pt} = \int_{\Delta S_t}\left(\frac{1}{\varepsilon}\frac{\partial G_{pt}}{\partial n} - \frac{\partial u_{pt}}{\partial n}\right)\mathrm{d}S, \\ C_{pt} &= \int_{\Delta S_t}\left(\frac{\partial G_{pt}}{\partial n_0} - \frac{1}{\varepsilon}\frac{\partial u_{pt}}{\partial n_0}\right)\mathrm{d}S, \quad D_{pt} = \int_{\Delta S_t}\frac{1}{\varepsilon}\left(\frac{\partial^2 G_{pt}}{\partial n_0 \partial n} - \frac{\partial^2 u_{pt}}{\partial n_0 \partial n}\right)\mathrm{d}S. \end{aligned} \tag{13}$$

where the integrations are performed on the small patch ΔS_t. To obtain above form, it is assumed that the solution f and h are constants in every small patch ΔS_t. For nearby patches p and t, Eq. (13) is performed by direct integration. For far field, the kernels for each patch integral are taken as constants (depending on the relative positions of p and t). The linear system can be written in a matrix form:

$$\begin{pmatrix} \left(\frac{1}{2\varepsilon}+\frac{1}{2}\right)I + B & -A \\ \left(\frac{1}{2}+\frac{1}{2\varepsilon}\right)I + D & -C \end{pmatrix}\begin{pmatrix} f \\ h \end{pmatrix} = \begin{pmatrix} \frac{1}{D_{\text{ext}}}\sum_k q_k G_{pk} \\ \frac{1}{D_{\text{ext}}}\sum_k q_k \frac{\partial G_{pk}}{\partial n_0} \end{pmatrix}, \tag{14}$$

where I is the identity matrix. The linear system is well-conditioned and can be solved efficiently using Krylov subspace methods. As the number of iterations is bounded, the most time consuming part becomes the convolution type matrix vector multiplication in each iteration. In the following section, we discuss how this can be accelerated by the new version FMM.

2.3. New version fast multipole method

The original idea of FMM is to subdivide the summation system of N particles into hierarchical groups of particles, and the potentials produced by far-field particles for a given particle are approximated by using the multipole expansions (Fig. 1a). The fundamental observation in the multipole expansion based methods is that

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

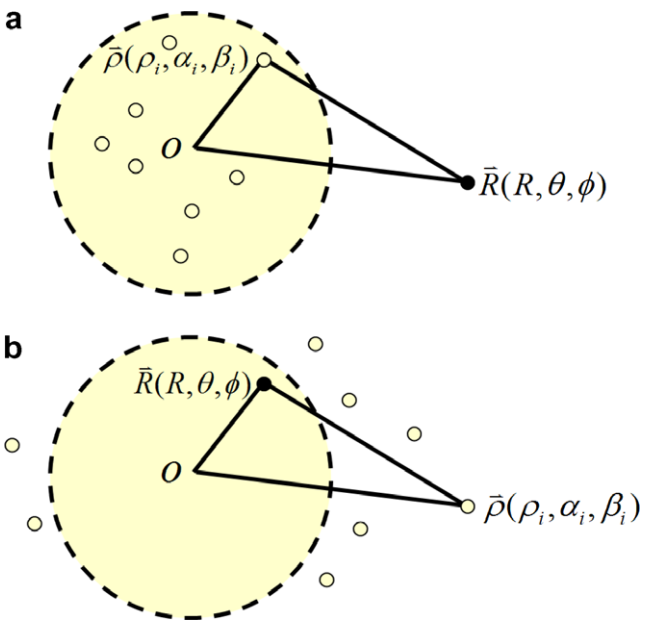

Fig. 1. Series expansion approximations of the function $\frac{1}{r}$. (a) For any point $R(R,\theta,\phi)$ located outside of a sphere S_a of radius a, the potential generated by N charges located inside of S_a with spherical coordinates $\rho(\rho_i,\alpha_i,\beta_i)$, respectively, can be described using *multipole expansions*; (b) in the opposite case, for any point $R(R,\theta,\phi)$ located inside of S_a, the potential generated by N charges located outside of S_a with spherical coordinates $\rho(\rho_i,\alpha_i,\beta_i)$, respectively, can be described using *local expansions*.

the numerical rank of the far field interactions is relatively low and hence can be approximated by P terms (depending on the prescribed accuracy) of the so-called "multipole expansion",

$$\Phi(R,\theta,\phi)=\sum_{i=1}^{N} q_i\cdot\frac{1}{|\vec{R}-\vec{\rho}_i|}\approx\sum_{n=0}^{P}\sum_{m=-n}^{m=n} M_n^m\frac{Y_n^m(\theta,\phi)}{R^{n+1}}, \tag{15}$$

where the multipole coefficients,

$$M_n^m=8\sum_{i=1}^{N} q_i\cdot Y_n^{-m}(\alpha_i,\beta_i), \tag{16}$$

where the spherical harmonic function of order n and degree m is defined according to the formula [1],

$$Y_n^m(\theta,\phi)=\sqrt{\frac{(2n+1)(n-|m|)!}{4\pi(n+|m|)!}}\cdot P_n^{|m|}(\cos\theta)\mathrm{e}^{\mathrm{i}m\phi}. \tag{17}$$

For the Debye–Huckel (screened Coulombic) interaction, a similar expansion can be written as follows:

$$\Phi(R,\theta,\phi)=\sum_{i=1}^{N} q_i\cdot\frac{\mathrm{e}^{-\kappa|\vec{R}-\vec{\rho}_i|}}{|\vec{R}-\vec{\rho}_i|}\approx\sum_{n=0}^{P}\sum_{m=-n}^{m=n} M_n^m\cdot k_n(\kappa R)\cdot Y_n^m(\theta,\phi), \tag{18}$$

where the multipole coefficients,

$$M_n^m=8\kappa\sum_{i=1}^{N} q_i\cdot i_n(\kappa\rho_i)\cdot Y_n^{-m}(\alpha_i,\beta_i), \tag{19}$$

where $i_n(r)$ and $k_n(r)$ are modified spherical Bessel and modified spherical Hankel functions respectively. The modified spherical Bessel and modified spherical Hankel functions are defined in terms of the conventional Bessel function via [1],

$$I_v(r) = \mathrm{i}^{-v} J_v(ir), \tag{20}$$

$$K_v(r) = \frac{\pi}{2\sin v\pi}[I_{-v}(r) - I_v(r)], \tag{21}$$

$$i_n(r) = \sqrt{\frac{\pi}{2r}} I_{n+1/2}(r), \tag{22}$$

$$k_n(r) = \sqrt{\frac{\pi}{2r}} K_{n+1/2}(r). \tag{23}$$

For arbitrary distributions of particles, a hierarchical octree (in 3D) is generated so each particle is associated with different boxes at different levels, and a divide-and-conquer strategy is applied to account for the far field interactions at each level in the tree structure. In the "tree code" developed by Appel [4], and Barnes and Hut [6], as each particle interacts with 189 boxes in its "interaction list" through P terms of multipole expansions at each level and there are $O(\log N)$ levels, the total amount of operations is approximately $189P^2N\log N$. The tree code was later improved by Greengard and Rokhlin [20]. In their original FMM, local expansions (under a different coordinate system) were introduced to accumulate information from the multipole expansions in the interaction list (Fig. 1b)

$$\Phi(R,\theta,\phi) = \sum_{i=1}^{N} q_i \cdot \frac{1}{|\vec{R} - \vec{\rho}_i|} \approx \sum_{n=0}^{P} \sum_{m=-n}^{m=n} L_n^m \cdot R^n Y_n^m(\theta,\phi), \tag{24}$$

where L_n^m are local expansion coefficients.

$$L_n^m = 8\sum_{i=1}^{N} q_i \cdot \frac{Y_n^{-m}(\alpha_i,\beta_i)}{\rho_i^{n+1}}. \tag{25}$$

For the screened Coulombic interaction, a similar expansion can be written as follows:

$$\Phi(R,\theta,\phi) = \sum_{i=1}^{N} q_i \cdot \frac{\mathrm{e}^{-\kappa|\vec{R}-\vec{\rho}_i|}}{|\vec{R} - \vec{\rho}_i|} \approx \sum_{n=0}^{P} \sum_{m=-n}^{m=n} L_n^m \cdot i_n(\kappa R) \cdot Y_n^m(\theta,\phi), \tag{26}$$

where

$$L_n^m = 8\kappa\sum_{i=1}^{N} q_i k_n(\kappa\rho_i) \cdot Y_n^{-m}(\alpha_i,\beta_i). \tag{27}$$

As the particles only interact with boxes and other particles at the finest level, and information at higher levels is transferred using a combination of multipole and local expansions as explained in Fig. 2, the original FMM is asymptotically optimal $O(N)$. However, because the multipole to local translation requires prohibitive $189P^4$ operations for each box, the huge prefactor makes the original FMM less competitive with the tree code and other FFT based methods. In 1997, a new version of FMM was introduced by Greengard and Rokhlin [21] for the Laplace equation. Compared with the original FMM, a plane wave expansion based diagonal translation operator is introduced and the original $189P^4$ operations were reduced to $40P^2 + 2P^3$.

The incorporation of fast FMM into BEM-based PB models has been successfully pursued by several groups [7,10,8,49]. However, all past implementations have used an older scheme of the FMM algorithm. As we mentioned above, the cost associated with those types of algorithms is approximately $189P^2N\log N$ (the tree code) or $189P^3N$ (in the original FMM scheme). Although it scales better than the direct computation, considerable speed up can only be achieved for systems of over 20,000 particles due to the large value of the prefactor. Recent work by Greengard and Rokhlin, which introduces a plane wave expansion during the repeated multipole to local transitions, significantly reduces the cost and breaks even with direct calculation for a reasonable value of N (~1000). The new version of FMM has subsequently been extended to screened Coulomb interactions (corresponding to the linearized PB kernel) in three dimensions [22]. Although mathematically more complicated, the new version of FMM makes it practical to be combined with the boundary element based solution of the linear PB equation. In our algorithm, we adapt the new version of FMM for the

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

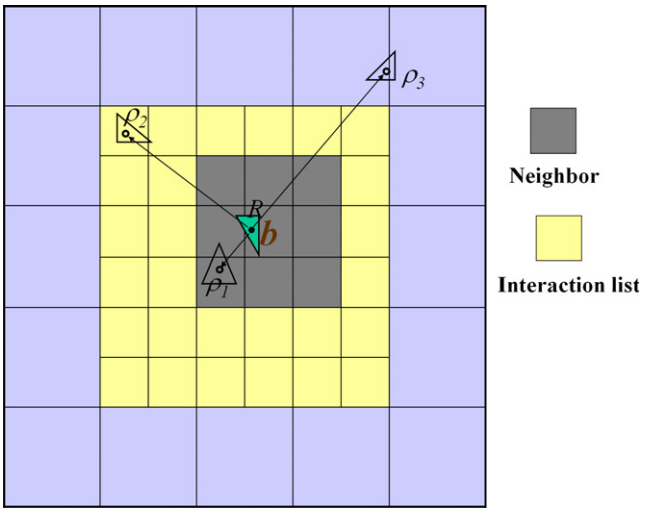

Fig. 2. Schematic showing the hierarchical divided boxes for recording the neighbor boxes and interaction list in the new version FMM. The neighbor boxes (up to 27 including itself in three dimensions) of the target box b are darkly shaded, while its interaction list (up to 189 boxes in three dimensions) is indicated in yellow. The remaining far-field boxes are indicated in light blue. Also shown are the source points ρ_i and evaluation point R (field). In BEM implementation, the source particles are located at the centers of the surface triangular elements.

screened Coulomb interactions. Preliminary numerical experiments show that the overall break even point of the new version FMM becomes 600 with 6-digit accuracy and about 400 for 3-digit.

Before proceeding to describe how the new version of FMM is used in the context of the BEM solution of the linearlized PB equation, we first introduce how the gradient of the local expansion coefficients can be calculated in FMM. If we define $Q^{\kappa}_{n,m} = i_n(\kappa r)\cdot Y^m_n(\theta,\phi)$ (in the limiting case when $\kappa = 0$, then $Q^0_{n,m} = r^n \cdot Y^m_n(\theta,\phi)$), then a very useful recursive relationship for the gradient of $Q^{\kappa}_{n,m}$ can be expressed as linear combinations of $Q^{\kappa}_{n,m}$ of different order and degree:

$$\nabla Q^{\kappa}_{n,m} = \frac{\kappa}{2n+1}[B]\left(\begin{pmatrix}(n+m-1)(n+m)Q^{\kappa}_{n-1,m-1}\\ (n+m)Q^{\kappa}_{n-1,m}\\ Q^{\kappa}_{n-1,m+1}\end{pmatrix}\cdot\frac{1}{s} - \begin{pmatrix}(n-m+1)(n-m+2)Q^{\kappa}_{n+1,m-1}\\ -(n-m+1)Q^{\kappa}_{n+1,m}\\ Q^{\kappa}_{n+1,m+1}\end{pmatrix}\cdot s\right), \tag{28}$$

where s is the scaling factor to avoid under-over flow ($s = 1$, if $\kappa r > 1$ and $s = \kappa r$, if $\kappa r \leqslant 1$). Note that the above relationship is applicable for all $0 < m < n-1$, for $m = 0$,

$$\nabla Q^{\kappa}_{n,0} = \frac{\kappa}{2n+1}[B]\left(\begin{pmatrix}Q^{\kappa}_{n-1,-1}\\ nQ^{\kappa}_{n-1,0}\\ Q^{\kappa}_{n-1,1}\end{pmatrix}\cdot\frac{1}{s} - \begin{pmatrix}Q^{\kappa}_{n+1,-1}\\ -(n+1)Q^{\kappa}_{n+1,0}\\ Q^{\kappa}_{n+1,1}\end{pmatrix}\cdot s\right), \tag{29}$$

for $m = n-1$,

$$\nabla Q^{\kappa}_{n,m} = \frac{\kappa}{2n+1}[B]\left(\begin{pmatrix}(n+m-1)(n+m)Q^{\kappa}_{n-1,m-1}\\ (n+m)Q^{\kappa}_{n-1,m}\\ 0\end{pmatrix}\cdot\frac{1}{s} - \begin{pmatrix}6Q^{\kappa}_{n+1,m-1}\\ -2Q^{\kappa}_{n+1,m}\\ Q^{\kappa}_{n+1,m+1}\end{pmatrix}\cdot s\right), \tag{30}$$

for $m = n$,

$$\nabla Q^{\kappa}_{n,m} = \frac{\kappa}{2n+1}[B]\left(\begin{pmatrix}(n+m-1)(n+m)Q^{\kappa}_{n-1,m-1}\\ 0\\ 0\end{pmatrix}\cdot\frac{1}{s} - \begin{pmatrix}2Q^{\kappa}_{n+1,m-1}\\ -Q^{\kappa}_{n+1,m}\\ Q^{\kappa}_{n+1,m+1}\end{pmatrix}\cdot s\right), \tag{31}$$

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

where,

$$[B] = -\frac{1}{2}\begin{pmatrix} 1 & 0 & -1 \\ \mathrm{i} & 0 & \mathrm{i} \\ 0 & -2 & 0 \end{pmatrix}. \tag{32}$$

Higher order derivatives can be easily obtained by recursive application of Eq. (28). For example, the second derivatives can be obtained by inserting the first order derivatives into the right side of Eq. (28). The recursive relationship for $\mathcal{Q}^{\kappa}_{n,m}$ is a very useful property for applying FMM to the BEM solution of PB equation, which will become apparent in the following section.

2.4. FMM in the context of BEM

The solution of the f_i and h_i can be obtained by inverting the $2N \times 2N$ matrix in Eq. (14). As mentioned above, direct methods such as Gaussian elimination and LU decomposition are too expensive in terms of both CPU time and memory. To this end, an iterative procedure will be used in the present algorithm. Another important feature of these iterative methods is that no explicit matrix needs to be stored or calculated; only the calculation of matrix–vector multiplication is required. The multiplication of a matrix (A, B, C, and D) and a vector (f and h) is analogous to calculating electrostatic potentials for $2N$ locations induced by $2N$ point charges. In the present FMM implementation, for each evaluation point p, the evaluation of the left-hand side of matrix Eq. (14) can be divided into two parts: (1) contributions from all of the far-field elements of element p (located outside the finest level box encompassing the evaluation point p) will be calculated using local expansions; (2) contributions from all remaining near neighbor elements (inside the same childless box that contains evaluation point p) must be evaluated directly (Fig. 2).

It is convenient to convert the normal derivatives of functions G and u at ρ into the spatial gradients of G and u at R (Fig. 3) using the following equations:

$$\begin{aligned} \frac{\partial G}{\partial n} &= -\nabla_R G(R,\rho)\cdot n, \quad \frac{\partial^2 G}{\partial n_0\,\partial n} = -n_0\cdot\nabla_R^2 G(R,\rho)\cdot n, \\ \frac{\partial u}{\partial n} &= -\nabla_R u(R,\rho)\cdot n, \quad \frac{\partial^2 u}{\partial n_0\,\partial n} = -n_0\cdot\nabla_R^2 u(R,\rho)\cdot n, \end{aligned} \tag{33}$$

where $n = (n_x, n_y, n_z)$ is the unit normal vector at point ρ, $n_0 = (n_{0x}, n_{0y}, n_{0z})$ is the unit normal vector at point R. Substituting Eq. (33) into Eq. (13) yields,

$$A_{pt} = (G_{pt} - u_{pt})\Delta S_t, \tag{34}$$

$$B_{pt} = \left(-\frac{1}{\varepsilon}\nabla_R G_{pt}\cdot n + \nabla_R u_{pt}\cdot n\right)\Delta S_t, \tag{35}$$

$$C_{pt} = \left(\nabla_R G_{pt}\cdot n_0 - \frac{1}{\varepsilon}\nabla_R u_{pt}\cdot n_0\right)\Delta S_t, \tag{36}$$

$$D_{pt} = \frac{1}{\varepsilon}(-n_0\cdot\nabla_R^2 G_{pt}\cdot n + n_0\cdot\nabla_R^2 u_{pt}\cdot n)\Delta S_t. \tag{37}$$

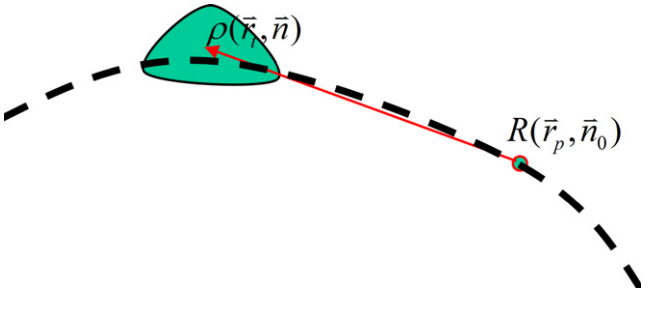

Fig. 3. Schematic showing the location of the evaluation point $R(\vec{r}_p,\vec{n}_0)$ (R_p) and a BE location ρ_t.

Given an initial set of f_t and h_t at any element locations, then for any evaluation point p, the far-field contribution to the left-hand side of Eq. (14) can be written as local expansions that sum contributions from a collection of far-field elements (denoted as $t \in \{L\}$),

$$\Phi_{1p} \simeq \sum_{n=0}^{P} \sum_{m=-n}^{m=n} \left(-\frac{1}{\varepsilon} \{\nabla_R Q^0_{n,m}(R)\}\{E^0_{n,m}\} - \{Q^0_{n,m}(R)\}\{H^0_{n,m}\} + \{\nabla_R Q^\kappa_{n,m}(R)\}\{E^\kappa_{n,m}\} + \{Q^\kappa_{n,m}(R)\}\{H^\kappa_{n,m}\} \right), \tag{38}$$

$$\begin{aligned} \Phi_{2p} \simeq \sum_{n=0}^{P} \sum_{m=-n}^{m=n} & \left(-\{n_{0x}, n_{0y}, n_{0z}\}\{\nabla_R Q^0_{n,m}(R)\}\{F^0_{n,m}\} - \frac{1}{\varepsilon}\{n_{0x}, n_{0y}, n_{0z}\}\{\nabla^2_R Q^0_{n,m}(R)\}\{E^0_{n,m}\} \right. \\ & \left. + \frac{1}{\varepsilon}\{n_{0x}, n_{0y}, n_{0z}\}\{\nabla_R Q^\kappa_{n,m}(R)\}\{F^\kappa_{n,m}\} + \frac{1}{\varepsilon}\{n_{0x}, n_{0y}, n_{0z}\}\{\nabla^2_R Q^\kappa_{n,m}(R)\}\{E^\kappa_{n,m}\} \right). \end{aligned} \tag{39}$$

The local expansion coefficients $\{E^0_{n,m}, H^0_{n,m}, F^0_{n,m}, E^\kappa_{n,m}, H^\kappa_{n,m}, F^\kappa_{n,m}\}$ for all of the elements $t \in \{L\}$, are

$$\begin{aligned} \{E^0_{n,m}\} &= \frac{1}{4\pi} \sum_{t\in\{L\}} \begin{pmatrix} n_x f_t \\ n_y f_t \\ n_z f_t \end{pmatrix} L^0_{n,m}(\rho)\mathrm{d}S_t, \quad \{E^\kappa_{n,m}\} = \frac{1}{4\pi} \sum_{t\in\{L\}} \begin{pmatrix} n_x f_t \\ n_y f_t \\ n_z f_t \end{pmatrix} L^\kappa_{n,m}(\rho)\mathrm{d}S_t, \\ \{H^0_{n,m}\} &= \frac{1}{4\pi} \sum_{t\in\{L\}} h_t L^0_{n,m}(\rho)\mathrm{d}S_t, \quad \{H^\kappa_{n,m}\} = \frac{1}{4\pi} \sum_{t\in\{L\}} h_t L^\kappa_{n,m}(\rho)\mathrm{d}S_t, \\ \{F^0_{n,m}\} &= \frac{1}{4\pi} \sum_{t\in\{L\}} f_t L^0_{n,m}(\rho)\mathrm{d}S_t, \quad \{F^\kappa_{n,m}\} = \frac{1}{4\pi} \sum_{t\in\{L\}} f_t L^\kappa_{n,m}(\rho)\mathrm{d}S_t, \end{aligned} \tag{40}$$

where $n_x f_t \Delta S_t$, $n_y f_t \Delta S_t$, $n_z f_t \Delta S_t$, $h_t \Delta S_t$ can be considered as groups of effective charges respectively. In Eqs. (38) and (39), the operation between two curly braces could be scalar–scalar product, or vector–vector dot product, or matrix–vector/vector–matrix multiplication, depending on the property of the quantities in the curly braces. It is worth noting that, for both $\frac{\partial^2 G}{\partial n_0 \partial n}$ and $\frac{\partial^2 u}{\partial n_0 \partial n}$, the first derivative is with respect to R (evaluation points) and the second is with respect to ρ (source points), so there is only a little computational overhead (<10%) compared to the original non-derivative formulation.

At this point, we are ready to summarize the FMM algorithm in the context of BEM solution of the PB equation, which proceeds as follows (Fig. 2):

1. Develop an octree structure encompassing all of the boundary elements by recursively dividing each box into eight child boxes until any child box contains no more than s BEs.
2. Compute multipole expansion coefficients at the tree's finest level; for each parent box, form a multipole expansion by merging multipole expansions from its eight children. (Note: eight sets of multipole coefficients will be needed considering four sets of effective charges $n_x f_t \Delta S_t$, $n_y f_t \Delta S_t$, $n_z f_t \Delta S_t$, $h_t \Delta S_t$ for both $\kappa = 0$ and $\kappa \neq 0$ cases.)
3. Start at the tree's coarsest level, compute local expansion coefficients by converting the multipole expansions at any well-separated boxes (interaction list) into a local expansion around the target center and by directly adding contributions due to local near source points (neighbor boxes).
4. For each parent box, translate the local expansion to each of its children.
5. Go to step 3 until the finest level is reached. (Note: again eight sets of local coefficients will be needed considering four sets of effective charges $n_x f_t \Delta S_t$, $n_y f_t \Delta S_t$, $n_z f_t \Delta S_t$, $h_t \Delta S_t$ for both $\kappa = 0$ and $\kappa \neq 0$ cases.)
6. For each childless box, evaluate the potential at each target location from the local expansions, and compute the remaining near neighbor interactions directly.

2.5. Krylov subspace methods and mesh generation

In our algorithm, a parallel iterative methods package for systems of linear equations PIM23 [14] is used. Several iterative schemes are available in the package including the GMRES method, BiCGStab method, and transpose-free quasi-minimal residual (TFQMR) algorithm. Preliminary numerical experiments show that the

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

GMRES method converges faster than other methods, which agrees with existing analyses. Because the memory required by the GMRES method increases linearly with the iteration number k, and the number of multiplications scales like $\frac{1}{2}k^2N$, for large k, the GMRES procedure becomes very expensive and requires excessive memory storage. For these reasons, instead of a full orthogonalization procedure, GMRES can be restarted every k_0 steps where $k_0 < N$ is some fixed integer parameter. The restarted version is often denoted as GMRES (k_0). For other alternative methods as BiCGStab method and TFQMR algorithm, the storage required is independent of iteration number k, and the number of multiplications grows only linearly as a function of k. Currently a detailed comparison of different Krylov subspace methods is being performed and results will be reported in later papers.

There are normally three types of "surface" used to define the molecular boundary dividing the low dielectric (interior) and high dielectric (exterior) regions: the *van der Waals* surface is the surface area of the volume formed by placing van der Waals spheres at the center of each atom in a molecule. The *solvent-accessible surface* [36] is formed by rolling a solvent, or a probe, sphere over the van der Waals surface. The trajectory of the center of the solvent sphere defines the solvent-accessible surface. Whereas, the *solvent-excluded surface* is defined as the trajectory of the boundary of the solvent sphere in contact with the van der Waals surface. The solvent-excluded surface is also referred to as the molecular surface. In our BEM, to discretize the boundary integral equations, a triangular mesh of molecular surface is generated using the software MSMS [39], and elements of zero and extremely small area are removed by a mesh checking procedure in our algorithm. The node density and probe radius are input parameters of MSMS to control the fineness of the output mesh; the typical values are 1.0 $Å^2$ and 1.5 Å, respectively. Mesh generation is a fast step and takes only a few seconds for medium-sized molecules. A typical mesh of a molecule with 8362 atoms is shown in Fig. 4, which contains 32,975 vertices and 65,982 triangles and is generated within 3 s of cpu time.

2.6. System set up

For all calculations, the AMBER [12] atomic charges and radii were assigned for protein atoms. A probe radius of 1.5 Å was used to define the dielectric interface. The relative dielectric constants were taken as 2.0 for solute and 80.0 for solvent. In the protein–DNA interaction calculation, the ion concentration was set to 50 mM, which is equivalent to a Debye–Huckel screening length of 13.8 Å. The meshes were generated at a density of 1.0 $Å^{-2}$. A single mesh was generated if two molecular surfaces were separated by less than 3 Å, while for the further separations the system was treated as two separate domains with two sets of meshes.

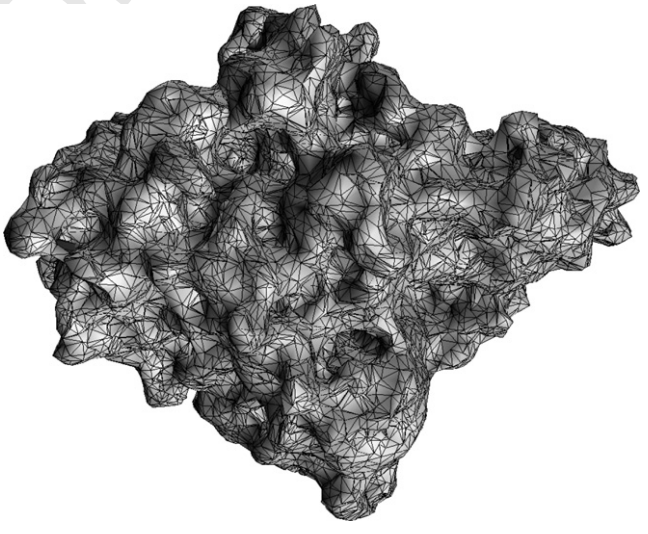

Fig. 4. A typical surface triangulated mesh of a protein (Acetylcholinsterase).

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

3. Results

3.1. Computational performance

As a first numerical experiment, we compared the speed and memory usage of the FMM to the direct calculation in one GMRES iteration step. The position and parameters of BEs were randomly generated but uniformly distributed on the surface of a sphere of radius 40 Å. As expected, the error of the FMM calculation is bounded when the number of multipole expansion terms P is set. Similar to what is observed in the original FMM implementation [22], our algorithm breaks even with the direct calculation at about $N = 400$ for 6-digit precision ($P = 9$), and $N = 600$ for 6-digit precision ($P = 16$). As shown in Fig. 5a, in contrast to the quadratic increase in direct calculation, the actual CPU time required by our fast algorithm grows approximately linearly with the number of BEs. On the other hand, Fig. 5b displays some non-linearity for the growth of memory usage, whereas in theory it should also follow a linear growth. This is because we have used a non-adaptive FMM. In the non-adaptive FMM, as the number of levels increases, there is a cubic increase of number of boxes (storing the local expansion coefficients for each box is the main source of memory usage), leading to a slightly non-linear growth of memory usage.

In our calculation, the majority of computer memory is allocated to store the neighboring list and the corresponding near-field coefficients, the size of which mainly relies on the total number of BEs and the level for

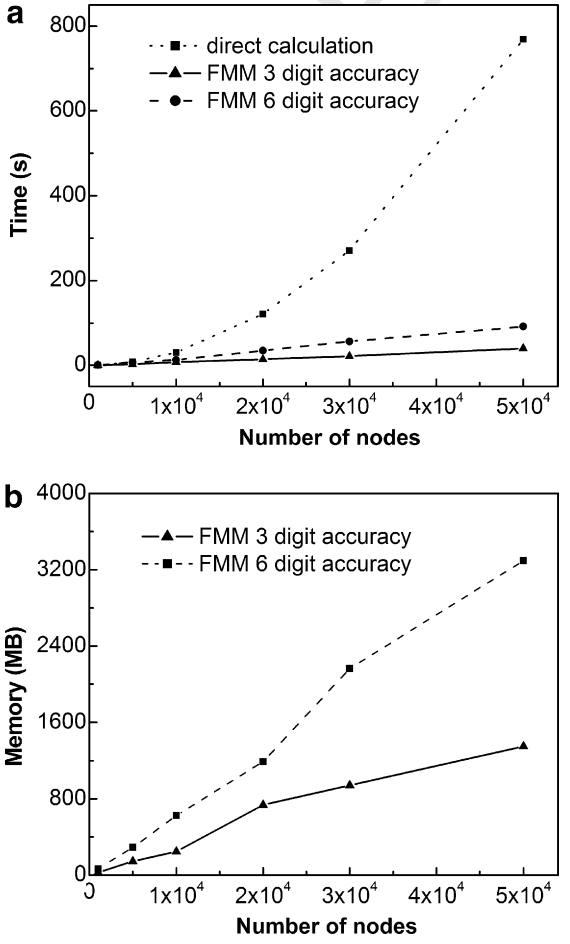

Fig. 5. The CPU time (a) and memory usage (b) of our fast BEM-PB algorithm as compared to those from the direct calculation in one GMRES iteration step.

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

box subdivision. Depending on a tradeoff between memory and speed, at each iterative step these coefficients can either be saved as in a memory-intensive mode or be discarded as in a memory-saving mode. In a non-adaptive FMM case, the number of neighboring boxes of a box (therefore any BE located within this box) is 27 (including itself). If we further assume that the maximum number of elements per box at the finest level is s, then it is easy to see that the number of near-field elements for each BE can normally be up-bounded by a fixed number $27s$. Hence, the size of neighboring list is also up-bounded by $27sN$; this and the fact that there are at most $2N/s$ boxes in the tree structure lead to $O(N)$ overall memory usage.

When using the FMM, it is important to keep a load balance between the number of BEs in the local list (calculated directly) and the number of BEs in the far-field (calculated using local expansions). If the number of local BEs is too large, then the advantage of using multipole expansions is not fully taken. Conversely, if the number of local BEs is too small, then more boxes will be needed, which usually means more operations of expansions. We assessed the performance of the FMM on a 10,000 BE system (again in a single GMRES iteration step) using various levels and terms (P); results are presented in Table 1. The total timing T_{total} is broken into the T_{fmm} for far-field calculation and T_{local} for local direct calculation. For both 3- and 6-digit accuracy, the optimal level is 4. Having more levels (more boxes, fewer local BEs) and fewer levels (fewer boxes, more local BEs) both slow down the overall speed because of the unbalanced T_{fmm} and T_{total}. Generally, the optimal level of box subdivision depends on number of terms P, so that the maximum number of BEs in the lowest level box s is comparable to $P^{\frac{3}{2}}$.

To assess the performance of the FMM BEM algorithm in solving the PB equation, we next calculate the Born solvation energy of a point charge +50 e located at the center of a spherical cavity with a radius of 50 Å. The surface is discretized at various resolution levels by recursively subdividing an icosahedron. Table 2 summarizes the timing results (on a Dell dual 2.0 GHz P4 desktop with 2 GB memory) and some related control parameters using a FMM accelerated BEM (denoted by FMM BEM) and a direct BEM without invoking any fast algorithms (denoted by direct BEM). Due to memory constraints, the PC can not handle higher levels of subdivision on the sphere (more than 300 k BEs). As for the efficiency, we noticed that regardless of the surface resolution, all the GMRES iteration steps are below 5, which numerically confirms that the derivative BEM formulations are well-conditioned. The CPU time for the new version of FMM linearly increases with the number of BEs, while it quadratically increases for the direct integration method. Note that whenever switching to a higher level of box division, there will be a small jump of CPU time due to the increased boxes,

Table 1
Timing results for the FMM on a 10,000-node-system (in one GMRES iteration step) using various levels and terms (P)

P	Levels	T_{fmm} (s)	T_{local} (s)	T_{total} (s)
9	3	0.7	10.6	11.3
9	4	1.2	2.4	3.6
9	5	5.7	0.4	6.1
16	3	2.3	10.6	12.9
16	4	4.9	2.4	7.3
16	5	26.9	0.4	27.3

Table 2
BEM performance on a spherical cavity case with different surface mesh sizes[a]

Number of elements	$T_{\text{direct BEM}}$ (s)	$T_{\text{FMM BEM}}$ (s)	Level	Iteration steps	$E_{\text{solvation}}$ (error%)	Error (%) in	
						f	h
320	0.13	0.18	2	5	−4227.5 (4.5)	6.6	5.6
1280	1.56	0.82	3	5	−4134.5 (2.2)	2.8	2.5
5120	19.67	3.39	3	5	−4088.6 (1.1)	1.4	1.1
20,480	247.20	15.86	4	5	−4066.5 (0.5)	0.7	0.6
81,920	3122.10	87.96	5	5	−4050.6 (0.3)	0.2	0.4

[a] Of radius 50 Å with a point charge +50 e located at the center. The exact Born solvation energy $E_{\text{solvation}}$ of the cavity is −4046.0 (energy is in kcal/mol).

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

which leads to some deviation from performance linearity for the FMM. For a system with 81,920 surface elements, the O(N) new version FMM is approximately 40 times faster than the direct method.

The numerical error of our BEM algorithm is on the first order of the grid size of the mesh. In the spherical case in Table 2, when the mesh is refined to a higher level, the number of BEs is quadrupled, and the size of each element is reduced by half. The relative errors of the calculated energy, f, and h compared with the analytical results also show that the computational accuracies are nearly linearly improved upon the refinement of mesh scale. More discussion on the accuracies of BEMs can be found in Ref. [31].

To further illustrate the performance of our fast BIE technique on protein electrostatic calculations, we computed the electrostatic solvation energies of fasciculinII, a 68 residue protein, and compared the algorithm performance with the multigrid finite difference algorithm, as implemented in the widely used software APBS [5] (see Table 3). We want to mention that the two program codes employ very different algorithms and data structures, hence an exact comparison between them would be difficult. Also, APBS is designed primarily for massively parallel computing; it has an integrated mesh generation routine, while the current BEM only runs on a single CPU, and needs a pre-generated mesh as an input. Two sets of meshes at different resolutions were generated for BEM and APBS calculations, respectively. Similar convergence trends are observed for both energy calculations. At low mesh resolution (with small number of nodes and faces), the BEM seems to require more memory than APBS does. The reason is that we use the same level of 4 of box subdivision for all the BEM calculations, which consumes a large portion of the total memory, and may not be optimal for small systems. When system size increases, the memory usage shows a slower increase, as does the CPU time cost. It should also be noted that APBS solves the PB equation twice to obtain the solvation energy, while BEM only solves it once. However, if the potentials and forces at the "volume" grid points away from the surface are needed, they are readily available in APBS solutions, while in BEM it is necessary to calculate again by integrating the PB equation solutions on the boundary.

3.2. Electrostatics of the nicotinic acetylcholine receptor (nAChR)

nAChR is one of ligand-gated ion channels that mediate fast synaptic transmission between cells. The roles of electrostatic interactions in governing the agonist binding, ion conduction and anesthetic action in nAChR have been implicated in many previous studies. As a test of our PB solver, we calculated the electrostatic potentials of the human α7 nAChR. The receptor structure including both the extra-cellular and intra-cellular domains was built up by homology modeling based on the cryo-electron microscopy structure of *Torpedo* nAChR (PDB code: 2BG9) [45]. The modeled structure contains 1880 residues, has a total length of about

Table 3
Comparison between BEM[a] and APBS[b]

Methods	Mesh	Memory (MB)		CPU (s)		$E_{solvation}$ (kcal/mol)	Iteration steps
		a	*b*	*a*	*b*		
BEM	8894 *F*, 4449 *V*	224	54	22	35	−556.1	14
	12044 *F*, 6024 *V*	289	59	26	53	−540.3	13
	15046 *F*, 7525 *V*	350	63	32	75	−534.6	13
	18046 *F*, 9025 *V*	411	67	36	98	−525.5	13
	21430 *F*, 10717 *V*	481	72	44	129	−522.0	13
APBS	65 × 65 × 65	78		39		−552.1	–
	97 × 65 × 97	150		64		−542.3	–
	127 × 97 × 127	341		131		−531.0	–
	161 × 129 × 161	742		258		−525.0	–
	225 × 161 × 225	1784		599		−522.8	–

F denotes the number of faces, *V* the vertices; *a* and *b* denote the memory-intensive and memory-saving calculation modes, respectively.
[a] Using the same level = 4 for all FMM calculations in BEM.
[b] APBS using focusing procedure, and solving the PB equation two times in each solvation energy calculation. When a much finer mesh (321 × 321 × 321, which can not be handled in a 2 GB memory PC) is used to run APBS again, a solvation energy 521.1 kcal/mol is obtained. This could be taken as a reference solvation energy.

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

160 Å and a diameter of about 40 Å parallel to the membrane surface. The BEM calculation was performed with 194,428 triangular elements and 97,119 vertices.

In Fig. 6, the molecular surface of nAChR is colored according to electrostatic potentials such that the most negative region is in red while the most positive region is in blue. The interior of the channel vestibule formed by the pentameric assembly of the ligand-binding domains shows very negative potentials. This would be expected to increase the local concentration of permeant cations (i.e. Na^+ and K^+ ions), and is consistent with earlier suggestions [11]. Moreover, deeper inside the channel, more negative potentials are observed, which reach the minimum roughly in the middle of pore. The existence of an electrostatic potential gradient across the channel pore may facilitate passage of ions through the channel. The surface potentials can be divided into two regions: the membrane-spanning domain that is dominated by positive potentials, and the extra/intra-cellular domains that are dominated by negative potentials. The strong negative potentials on the extra-cellular surface are expected to impose electrostatic steering attraction to positive ligands (e.g. acetylcholine) and cations.

We also performed the calculation in the zero ionic strength condition. The results turn out to be very different where the surface potentials are almost all negative (data not shown). The difference indicates that the ionic strength has a great impact on the electrostatic character of nAChR.

3.3. Electrostatic interactions between Sso7d and DNA

We studied the electrostatic interactions between two molecules: Sso7d and DNA based on a crystal structure (PDB code: 1AZQ) [37]. Sso7d is a small chromosomal protein from the hyperthermophilic archaeabacteria *Sulfolobus solfataricus*. The protein has high thermal, acid and chemical stability. It binds DNA without marked sequence preference. In the crystal structure, Sso7d has 66 residues in complex with a short double-stranded DNA with 8 bp. Sso7d binds in the minor groove of DNA and causes sharp kink in DNA. The protein–DNA complexes are normally highly charged. Sso7d is positively charged (+6), whereas the complex is negatively-charged (−8) overall due to the additional 14 negative charges carried by the DNA phosphate

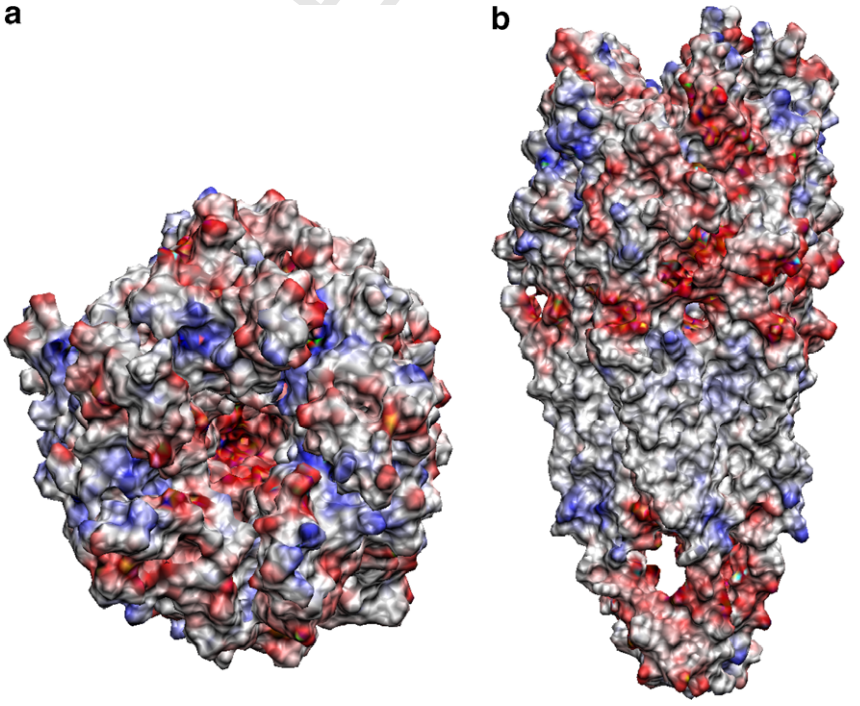

Fig. 6. The surface potential map of nAChR from different views. The increasing potential from negative to positive value is represented by changing the color from red to blue.

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

groups. To investigate the role of electrostatics in the Sso7d–DNA association process, the interactions between Sso7d and DNA at different separation distances are calculated. These structures are generated by displacing DNA away from the binding site along the center-to-center direction in the Sso7d–DNA complex.

The BEM calculation is performed on a two-domain system if two molecules move away and two separate meshes are generated or on a single domain system if two molecules are close enough to 'merge' and only a single mesh generated. For intermolecular electrostatics, the present BEM method provides the full PB interaction energy that takes into account both the desolvation and mutual polarization contributions from the two molecules. Fig. 7a shows the electrostatic potentials mapped on the molecular surfaces of Sso7d and DNA at a separation of 10 Å. The potential surfaces exhibit good electrostatic complementarity at the binding interface. Electrostatic attraction governs the intermolecular interaction at distances larger than ~5 Å (Fig. 7b, black line). Nevertheless, the electrostatic interaction becomes repulsive at close distances. A closer inspection of the complex structure suggests that a significant component of the binding free energy is due to the non-electrostatic interactions, made in large part by the interfacial hydrophobic residues [37]. The origin of the large unfavorable electrostatic interaction at close separations can be attributed to the electrostatic desolvation, an effect due to the unfavorable exclusion of the high dielectric solvent around one protein when the other one approaches. As a comparison, we also calculate the screened Coulomb interactions by summing up all the atomic pair contributions between Sso7d and DNA (see Fig. 7b, red line). In this treatment, it is found that the interactions are all attractive across the whole separations. The values are close to the full BEM calculations at long distances, but the desolvation effects are obviously missed at close distances.

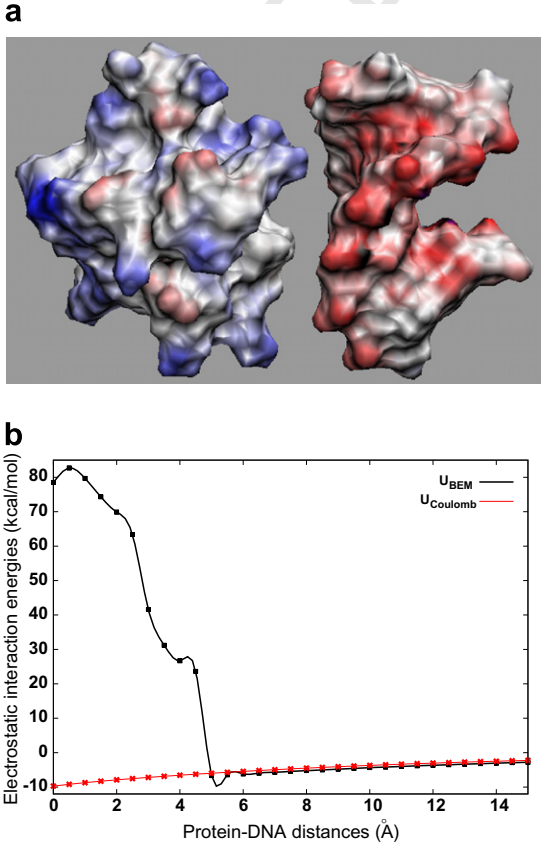

Fig. 7. Electrostatics of Sso7d-DNA. (a) The surface potential map of Sso7d and DNA at separation of 10 Å. (b) The electrostatic interaction energies as functions of separation along the center-to-center unbinding direction. U_{BEM} is the full electrostatic interaction energy determined by our BEM, and $U_{Coulomb}$ is the sum of all the atomic pair screened Coulomb interactions between Sso7d and DNA. The curves connect the calculation points (denoted by the diamond and star symbols) consecutively by fitting with cubic splines.

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

The electrostatic interaction characteristics displayed in Fig. 7 are very similar to the acetylcholinesterase and fasciculinII complex system as has been demonstrated previously [30], which also shows long-range attraction ($\gtrsim 5$ Å) and close-range repulsion. Another common observation of these two systems is that the electrostatic interactions start abruptly increasing from around 5 Å. This is the distance where the water molecules between the two molecules are squeezed out, and the two molecules begin to collapse into a compact complex structure. At the same time, the generated molecular surface meshes of the two molecules also begin to merge into a single mesh. This implies that the interfacial non-polar interaction, hydrophobic packing, and possibly local conformational rearrangement upon binding take effects from around 5 Å of association and become dominant binding forces in the final stage of complex formation.

4. Conclusions and discussion

In this paper, an algorithm with an optimal computational complexity is presented by introducing the new version of FMM. This is combined with the well-conditioned BIE formula to solve the linearized PB electrostatics for systems of arbitrary numbers of biomolecules. This algorithm enhances our computational capability to treat large and complex biological systems. The method could possibly be further developed for applications to dynamical simulations (Monte Carlo, Brownian dynamics, and all-atom molecular dynamics, etc.) of proteins with full PB electrostatics. However, although the speed has been greatly improved, the dynamical computation for biosystems using the current algorithm still exceed the presently available computer capability. For a typical Brownian dynamics simulation with tens of millions of steps, the one-step PB solution needs to be completed within no more than a few tenths of a second to meet the total wall-clock time constraint. Based on this estimation, the present BEM solver is still about one order slower. Several techniques can be pursued to further increase the efficiency of the present algorithm. Using an adaptive FMM scheme can greatly decrease the memory usage because most ''volume'' boxes away from the boundary need not be stored and counted. Curvilinear boundary element methods can reduce the number of boundary elements by a few to several folds while achieving the similar calculation accuracy. Improvement on mesh generation can also help in dynamical simulation when changing boundaries occur due to conformational flexibility, because the surface meshes need to be regenerated on the fly. Finally, another way to increase the computational speed is to implement an efficient parallelization of the code. Very good scalability of parallelization is anticipated, due to the nature of both the BEM and FMM algorithms.

Acknowledgement

We thank Prof. Jingfang Huang for supplying the FMM source code and helpful discussion, and Dr. Justin Gullingsrud for visualization aid. The work was supported in part by the National Institutes of Health, the National Science Foundation, the Howard Hughes Medical Institute, the National Biomedical Computing Resource, the National Science Foundation Center for Theoretical Biological Physics, the San Diego Supercomputing Center, the W.M. Keck Foundation, and Accelrys Inc.

References

[1] M. Abramowitz, I.A. Stegun, Handbook of Mathematical Functions, Dover Publications, New York, 1965.
[2] M. Altman, J. Bardhan, J. White, B. Tidor, An accurate surface formulation for biomolecule electrostatics in non-ionic solutions, Conf. Proc. IEEE Eng. Med. Biol. Soc. 7 (NIL) (2005) 7591–7595.
[3] M.D. Altman, J.P. Bardhan, B. Tidor, J.K. White, FFTSVD: a fast multiscale boundary-element method solver suitable for Bio-MEMS and biomolecule simulation, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 25 (2) (2006) 274–284.
[4] A.W. Appel, An efficient program for many-body simulations, SIAM J. Sci. Stat. Comput. 6 (1985) 85–103.
[5] N.A. Baker, D. Sept, S. Joseph, M.J. Holst, J.A. McCammon, Electrostatics of nanosystems: application to microtubules and the ribosome, Proc. Natl. Acad. Sci. USA 98 (18) (2001) 10037–10041.
[6] J. Barnes, P. Hut, A hierarchical O($n\log n$) force-calculation algorithm, Nature 324 (4) (1986) 446–449.
[7] R. Bharadwaj, A. Windemuth, S. Sridharan, B. Honig, A. Nicholls, The fast multipole boundary-element method for molecular electrostatics – an optimal approach for large systems, J. Comput. Chem. 16 (7) (1995) 898–913.
[8] A.J. Bordner, G.A. Huber, Boundary element solution of the linear Poisson–Boltzmann equation and a multipole method for the rapid calculation of forces on macromolecules in solution, J. Comput. Chem. 24 (3) (2003) 353–367.

Please cite this article in press as: B. Lu et al., ''New-version-fast-multipole-method'' accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

[9] A.H. Boschitsch, M.O. Fenley, W.K. Olson, A fast adaptive multipole algorithm for calculating screened Coulomb (Yukawa) interactions, J. Comput. Phys. 151 (1) (1999) 212–241.
[10] A.H. Boschitsch, M.O. Fenley, H.X. Zhou, Fast boundary element method for the linear Poisson–Boltzmann equation, J. Phys. Chem. B 106 (10) (2002) 2741–2754.
[11] C.E. Capener, H.J. Kim, Y. Arinaminpathy, M.S.P. Sansom, Ion channels: structural bioinformatics and modelling, Hum. Mol. Genet. 11 (20) (2002) 2425–2433.
[12] W.D. Cornell, P. Cieplak, C.I. Bayly, I.R. Gould, K.M. Merz, D.M. Ferguson, D.C. Spellmeyer, T. Fox, J.W. Caldwell, P.A. Kollman, A 2nd generation force-field for the simulation of proteins, nucleic-acids, and organic-molecules, J. Am. Chem. Soc. 117 (19) (1995) 5179–5197.
[13] C.M. Cortis, R.A. Friesner, An automatic three-dimensional finite element mesh generation system for the Poisson–Boltzmann equation, J. Comput. Chem. 18 (13) (1997) 1570–1590.
[14] R.D. Dacunha, T. Hopkins, The parallel iterative methods (PIM) package for the solution of systems of linear equations on parallel computers, Appl. Numer. Math. 19 (1–2) (1995) 33–50.
[15] T. Darden, D. York, L. Pedersen, Particle mesh Ewald: an $n\log(n)$ method for Ewald sums in large systems, J. Chem. Phys. 98 (12) (1993) 10089–10092.
[16] M.E. Davis, J.A. McCammon, Electrostatics in biomolecular structure and dynamics, Chem. Rev. 90 (3) (1990) 509–521.
[17] P. Debye, E. Huckel, Zur theorie der elektrolyte, Phys. Zeitschr. 24 (1923) 185–206.
[18] F. Figueirido, R.M. Levy, R.H. Zhou, B.J. Berne, Large scale simulation of macromolecules in solution: combining the periodic fast multipole method with multiple time step integrators, J. Chem. Phys. 106 (23) (1997) 9835–9849.
[19] M.K. Gilson, A. Rashin, R. Fine, B. Honig, On the calculation of electrostatic interactions in proteins, J. Mol. Biol. 184 (3) (1985) 503–516.
[20] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (2) (1987) 325–348.
[21] L. Greengard, V. Rokhlin, A new version of the fast multipole method for the laplace equation in three dimensions, Acta Numer. 6 (1997) 229–269.
[22] L.F. Greengard, J.F. Huang, A new version of the fast multipole method for screened coulomb interactions in three dimensions, J. Comput. Phys. 180 (2) (2002) 642–658.
[23] M. Holst, N. Baker, F. Wang, Adaptive multilevel finite element solution of the Poisson–Boltzmann equation i. algorithms and examples, J. Comput. Chem. 21 (15) (2000) 1319–1342.
[24] A.H. Juffer, E.F.F. Botta, B.A.M. Vankeulen, A. Vanderploeg, H.J.C. Berendsen, The electric-potential of a macromolecule in a solvent – a fundamental approach, J. Comput. Phys. 97 (1) (1991) 144–171.
[25] S. Kapur, D.E. Long, IES3: Efficient electrostatic and electromagnetic simulation, IEEE Comput. Sci. Eng. 5 (4) (1998) 60–67.
[26] J.G. Kirkwood, On the theory of strong electrolyte solutions, J. Chem. Phys. 2 (1934) 767–781.
[27] I. Klapper, R. Hagstrom, R. Fine, K. Sharp, B. Honig, Focusing of electric fields in the active site of Cu–Zn superoxide dismutase: effects of ionic strength and amino-acid modification, Proteins 1 (1) (1986) 47–59.
[28] S.S. Kuo, M.D. Altman, J.P. Bardhan, B. Tidor, J.K. White, Fast methods for simulation of biomolecule electrostatics, in: ICCAD '02: Proceedings of the 2002 IEEE/ACM International Conference on Computer-aided Design, ACM Press, New York, NY, USA, 2002.
[29] J. Liang, S. Subramaniam, Computation of molecular electrostatics with boundary element methods, Biophys. J. 73 (4) (1997) 1830–1841.
[30] B.Z. Lu, X.L. Cheng, J.F. Huang, J.A. McCammon, Order N algorithm for computation of electrostatic interactions in biomolecular systems, Proc. Natl. Acad. Sci. USA 103 (51) (2006) 19314–19319.
[31] B.Z. Lu, J.A. McCammon, Improved boundary element methods for Poisson–Boltzmann electrostatic potential and force calculations, J. Chem. Theory. Comput. 3 (3) (2007) 1134–1142.
[32] B.Z. Lu, D.Q. Zhang, J.A. McCammon, Computation of electrostatic forces between solvated molecules determined by the Poisson–Boltzmann equation using a boundary element method, J. Chem. Phys. 122 (21) (2005) 214102.
[33] E.T. Ong, K.H. Lee, K.M. Lim, A fast algorithm for three-dimensional electrostatics analysis: fast fourier transform on multipoles (FFTM), Int. J. Numer. Meth. Eng. 61 (5) (2004) 633–656.
[34] E.T. Ong, K.M. Lim, K.H. Lee, H.P. Lee, A fast algorithm for three-dimensional potential fields calculation: fast Fourier transform on multipoles, J. Comput. Phys. 192 (1) (2003) 244–261.
[35] J.R. Phillips, J.K. White, A precorrected-FFT method for electrostatic analysis of complicated 3-D structures, IEEE Trans. Comput.-Aided Des. Integr. Circuit Syst. 16 (10) (1997) 1059–1072.
[36] F.M. Richards, Areas, volumes, packing, and protein-structure, Annu. Rev. Biophys. Bioeng. 6 (1977) 151–176.
[37] H. Robinson, Y.G. Gao, B.S. Mccrary, S.P. Edmondson, J.W. Shriver, A.H.J. Wang, The hyperthermophile chromosomal protein Sac7d sharply kinks DNA, Nature 392 (6672) (1998) 202–205.
[38] V. Rokhlin, Solution of acoustic scattering problems by means of second kind integral equations, Wave Motion 5 (3) (1983) 257–272.
[39] M.F. Sanner, A.J. Olson, J.C. Spehner, Reduced surface: an efficient way to compute molecular surfaces, Biopolymers 38 (3) (1996) 305–320.
[40] K.A. Sharp, B. Honig, Electrostatic interactions in macromolecules – theory and applications, Annu. Rev. Biophys. Biophys. Chem. 19 (1990) 301–332.
[41] W. Shi, J. Liu, N. Kakani, T. Yu, A fast hierarchical algorithm for 3D capacitance extraction, in: DAC '98: Proceedings of the 35th Annual Conference on Design Automation, ACM Press, New York, NY, USA, 1998.

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026

[42] M. Tanaka, V. Sladek, J. Sladek, Regularization techniques applied to boundary element method, AMSE Appl. Mech. Rev. 47 (1994) 457–499.
[43] J. Tausch, J. White, A multiscale method for fast capacitance extraction, in: DAC '99: Proceedings of the 36th ACM/IEEE Conference on Design automation, ACM Press, New York, NY, USA, 1999.
[44] M. Totrov, R. Abagyan, Rapid boundary element solvation electrostatics calculations in folding simulations: successful folding of a 23-residue peptide, Biopolymers 60 (2) (2001) 124–133.
[45] N. Unwin, Refined structure of the nicotinic acetylcholine receptor at 4 angstrom resolution, J. Mol. Biol. 346 (4) (2005) 967–989.
[46] J. Warwicker, H.C. Watson, Calculation of the electric-potential in the active-site cleft due to alpha-helix dipoles, J. Mol. Biol. 157 (4) (1982) 671–679.
[47] W. Xin, A.H. Juffer, A boundary element formulation of protein electrostatics with explicit ions, J. Comput. Phys. 223 (2007) 416–435.
[48] R.J. Zauhar, R.S. Morgan, A new method for computing the macromolecular electric-potential, J. Mol. Biol. 186 (4) (1985) 815–820.
[49] R.J. Zauhar, A. Varnek, A fast and space-efficient boundary element method for computing electrostatic and hydration effects in large molecules, J. Comput. Chem. 17 (7) (1996) 864–877.

Please cite this article in press as: B. Lu et al., "New-version-fast-multipole-method" accelerated ..., J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.05.026