Title

Order N algorithm for computation of electrostatic interactions in biomolecular systems

Permalink

https://escholarship.org/uc/item/3tf900bz

Journal

Proceedings of the National Academy of Sciences of the United States of America, 103(51)

ISSN

0027-8424

Authors

Lu, B Z
Cheng, X L
Huang, J F
et al.

Publication Date

2006-12-01

Peer reviewed

An order *N* algorithm for computation of electrostatic interactions in biomolecular systems

Benzhuo Lu [†‡ *], Xiaolin Cheng[‡¶*], Jingfang Huang[‖], and J. Andrew McCammon[†‡§¶]

[†] 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, [‖] Department of Mathematics, University of North Carolina, Chapel Hill, NC 27599-3250.

[*] To whom reprint requests should be addressed at University of California at San Diego, 9500 Gilman Drive, Mail code 0365, La Jolla, CA92093-0365. E-mail: blu@mccammon.ucsd.edu.

[*] Joint first author.

Poisson-Boltzmann (PB) electrostatics is a well established model in biophysics, however its application to large scale biomolecular processes such as protein-protein encounter is still limited by the efficiency and memory constraints of existing numerical techniques. In this paper, we present an efficient and accurate scheme which incorporates recently developed numerical techniques to enhance our computational ability. In particular, a boundary integral equation (BIE) approach is applied to discretize the linearized PB equation; the resulting integral formulas are well conditioned and are extended to systems with arbitrary numbers of biomolecules. The solution process is accelerated by Krylov subspace methods and a new version of the fast multipole method (FMM). In addition to the electrostatic energy, fast calculations of the forces and torques are made possible by using an interpolation procedure. Numerical experiments show that the implemented algorithm is asymptotically optimal $O(N)$ in both CPU time and required memory, and application to the acetylcholinesterase-fasciculin complex is illustrated.

In recent years, due to the rapid advances in biotechnology, both the temporal and spatial scales of biomolecular studies have been increased significantly: from *single* molecules to interacting molecular *networks* in a cell, and from the *static* molecular structures at different resolutions to the *dynamical* interactions in biophysical processes. In these studies, the electrostatics modeled by the well-established Poisson-Boltzmann (PB) equation has been shown to play an important role under physiological solution conditions. Therefore, its accurate and efficient numerical treatment becomes extremely important, especially in the study of large scale dynamical processes such as protein-protein association and dissociation in which the PB equation has to be solved separately during a simulation.

Traditional numerical schemes for PB electrostatics 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 the 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 with large spatial sizes, e.g. as encountered in protein association and dissociation. Alternative methods include the boundary element (BEM) and boundary integral equation (BIE) methods. In these methods, only the surfaces of the molecules are discretized, hence the number of unknowns is greatly reduced. Unfortunately, in earlier versions of BEM, the matrix is stored explicitly and the resulting dense linear system is solved using Gauss elimination, so that $O(N^2)$ storage and $O(N^3)$ operations are required, where N is the number of nodes defined on the surface to discretize the integrals as discrete summations using appropriate quadrature. Even with the acceleration afforded by Krylov subspace methods, direct evaluation of the $N(N-1)/2$ pairs of interactions in the summations still requires prohibitive $O(N^2)$ operations.

In the last twenty years, novel numerical algorithms have been developed to accelerate the calculation of this N-body problem from the original $O(N^2)$ direct method to the $O(N\log N)$ hierarchical "tree code"[1,2] and fast Fourier transform (FFT) based algorithms including the particle-mesh Ewald (PME) method,[3] and later to the asymptotically optimal $O(N)$ fast multipole method (FMM),[4] and eventually to a new version FMM with an optimized prefactor.[5] For the PB equation, however, only the original FMM and FFT based techniques have been introduced into the BEM/BIE formulations. Numerical experiments show that the original FMM,[4] although asymptotically optimal and well suited for multiscale time stepping schemes, is 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 $O(N)$.

In this paper, we present an efficient algorithm to further accelerate the solution of the PB equation. By proper coupling of single and double layer potentials as discussed by Rokhlin,[6] we derive a Fredholm second kind integral equation formulation for systems with an arbitrary number of domains (molecules). Similar formulations are used for single domain problems by Juffer et al.,[7] Liang and Subramaniam,[8] and Boschitsch et al.[9] Compared with "direct" formulations where Green's second identity is applied, the condition number of our 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 multiplications in each iteration, we use the new version FMM developed for the screened Coulombic interaction (Yukawa potential) by one of the authors and collaborators.[10] 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.

Whereas most previous PB electrostatics algorithms have mainly focused on the energy calculations, calculations of the PB forces and torques are also essential in many cases such as in dynamics simulations. In this algorithm, we introduce an $O(N)$ interpolation scheme in the post-processing stage for calculating the forces and torques. This scheme improves previous $O(N^2)$ results based on BEM.[11,12]

Boundary integral equation formulations.

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

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

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

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, q_k is the kth source point charge of the molecule, κ is the reciprocal of the Debye-Hückel screening length determined by the ionic strength of the solution, n is the outward normal vector, t is an arbitrary point on the boundary, and PV represents the principal value integral to avoid the singular point when $t \rightarrow p$ in the integral equations. In the formulas, $G_{pt} = \frac{1}{4\pi|r_t - r_p|}$ 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. 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.

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 $D_{\text{int}}\frac{\partial\phi^{\text{int}}}{\partial n} = D_{\text{ext}}\frac{\partial\phi^{\text{ext}}}{\partial n}$. 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 g. Unfortunately, theoretical analysis shows that the corresponding equation system for f and h is in general a Fredholm integral equation of the 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", in our method, we adapt a technique introduced by Rokhlin[6] where the single and double layer potentials are combined in order to derive an optimized second kind Fredholm integral equation. Similar techniques have been used by Juffer et al.[7] and others in engineering computations,[13–15] however most of them focus on single molecule cases. In the following, we present a well-conditioned derivative BIE formulation (second kind Fredholm equation) for multiple biomolecule systems, in which $j = 1,\cdots,J$ represents the separated molecules:

$$\left(\frac{1}{2\varepsilon}+\frac{1}{2}\right)f_p = \sum_j^J \int_{S^j}^{PV} \left[(G_{pt}-u_{pt})h_t - \left(\frac{1}{\varepsilon}\frac{\partial G_{pt}}{\partial n} - \frac{\partial u_{pt}}{\partial n}\right)f_t\right]dS_t^j + \frac{1}{D_{\text{ext}}}\sum_j\sum_{k^j} q_{k^j}G_{pk^j}, \quad p\in S^i, i=1,..J, \tag{3}$$

$$\left(\frac{1}{2}+\frac{1}{2\varepsilon}\right)h_p = \sum_j^J \int_{S^j}^{PV} \left[\left(\frac{\partial G_{pt}}{\partial n_0} - \frac{1}{\varepsilon}\frac{\partial u_{pt}}{\partial n_0}\right)h_t - \frac{1}{\varepsilon}\left(\frac{\partial^2 G_{pt}}{\partial n_0\partial n} - \frac{\partial^2 u_{pt}}{\partial n_0\partial n}\right)f_t\right]dS_t^j + \frac{1}{D_{\text{ext}}}\sum_j^J\sum_{k^j} q_{k^j}\frac{\partial G_{pk^j}}{\partial n_0}, \quad p\in S^i, i=1,..J. \tag{4}$$

As our formulas have the same integrands on different domain surfaces, FMM calculation is convenient and the same as in the single molecule case. Also, due to the small number of iterations for convergence, the solution of an arbitrary system is directly obtained by solving the PB equations only once, which differs from previous ”perturbation” scheme for two-domain systems.[11,16]

New version fast multipole method.

When Eqs. (3)-(4) are discretized, the resulting 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 this section, we discuss how this can be accelerated by the new version FMM.

The fundamental observation in the multipole expansion based methods is that 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”

$$\phi(\vec{R},\theta,\phi) = \sum_{i=1}^N q_i\cdot\frac{1}{|\vec{R}-\vec{\rho}|} \approx \sum_{n=0}^P\sum_{m=-n}^{m=n} M_n^m\frac{Y_n^m(\theta,\phi)}{|\vec{R}|^{n+1}} \tag{5}$$

where Y_n^m are the spherical harmonics and M_n^m the multipole coefficients. For arbitrary distribution of particles (meshes), a hierarchical oct-tree (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,[1] and Barnes and Hut,[2] 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 in 1987.[4] In their original FMM, local expansions (under a different coordinate system)

$$\phi(\vec{R},\theta,\phi) = \sum_{i=1}^N q_i\cdot\frac{1}{|\vec{R}-\vec{\rho}|} \approx \sum_{n=0}^P\sum_{m=-n}^{m=n} L_n^m\cdot|\vec{R}|^nY_n^m(\theta,\phi) \tag{6}$$

are introduced to accumulate information from the multipole expansions in the interaction list where L_n^m are local expansion coefficients. 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. 1, the original FMM is asymptotically optimal $O(N)$. However, because the multipole to local translation requires a 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.

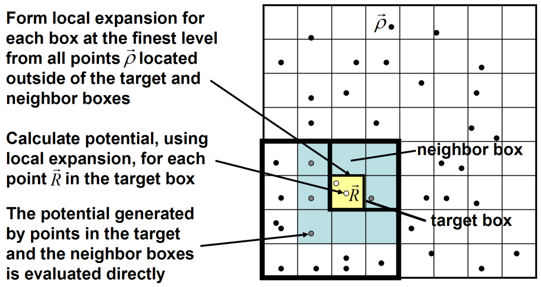

Form local expansion for each box at the finest level from all points $\vec{\rho}$ located outside of the target and neighbor boxes
Calculate potential, using local expansion, for each point $\vec{R}$ in the target box
The potential generated by points in the target and the neighbor boxes is evaluated directly
$\vec{\rho}$
neighbor box
$\vec{R}$
target box

Figure 1: Schematic showing the source points $\vec{\rho}$ and evaluation point $\vec{R}$ in the new version FMM. In BEM implementation, the source points are centered at the surface triangular elements.

In 1997, a new version of FMM was introduced by Greengard and Rokhlin[5] 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 $40P^2+2P^3$. In our algorithm, we adapt the new version of FMM for the screened Coulomb interactions (corresponding to the linearized PB kernel) developed by one of the authors and his collaborators.[10] 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. However, the new version FMM is more complicated than the original FMM in programming and theory, and we are unaware of any previous implementations for the linearized PB equation.

Krylov subspace methods and mesh generation

In our algorithm, a parallel iterative methods package for systems of linear equations PIM23[17] is used. Several iterative schemes are available in the package including the GMRES method, biconjugate gradients stabilized (BiCGStab) method, and transpose-free quasi-minimal residual (TFQMR) algorithm. Preliminary numerical experiments show that the 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). Currently a detailed comparison of different Krylov subspace methods is being performed and results will be reported in later papers.

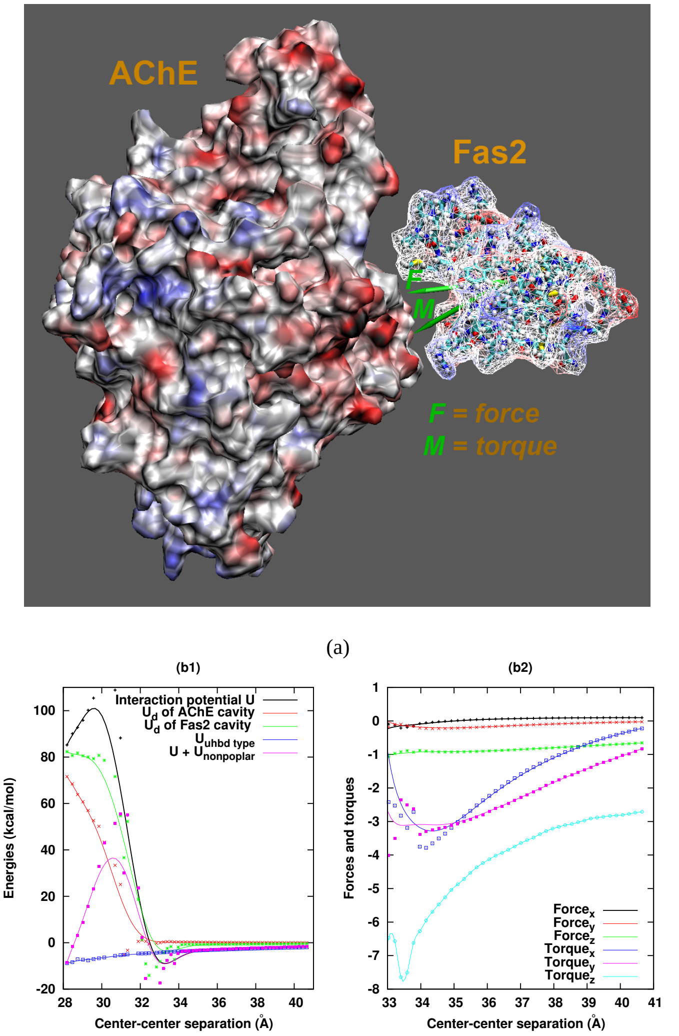
To discretize the boundary integral equations, a triangular mesh is generated using the package MSMS,[18] and zero and extremely small area elements are modified by a mesh checking procedure in our algorithm. A typical mesh is shown in the Fig. 2 (top right).

Force and torque calculations

In addition to energy calculation, an improved procedure is implemented to calculate the force and torque. Compared with previous $O(N^2)$ schemes,[11,12] the complexity of the new procedure is $O(N)$. In the calculation, the full stress tensor on the boundary includes contributions from conventional Maxwell stress tensor as well as the ionic pressure is given by[19]

$$T_{ij} = D_{\text{ext}}E_iE_j - \frac{1}{2}D_{\text{ext}}E^2\delta_{ij} - \frac{1}{2}D_{\text{ext}}\kappa^2\phi^2\delta_{ij}, \tag{7}$$

where E is the electrostatic field and δ_{ij} the Kronecker delta function. For the gradient of the potential, an interpolation scheme is used



(a)

Figure 2: (a) The surface potential map of AChE and Fas2 at separation of 14 *Å*. The two green arrows indicated as *F* and *M* show the force (0.10, -0.03, -0.69) and torque (-0.35, -1.03, -2.8), respectively, which are scaled for visualization. (b1) The electrostatic potential profiles as functions of separations along a predefined direction: the total electrostatic interaction potential U, electrostatic desolvation energies due to AChE and Fas2 cavities, respectively. $U_{\text{uhbd-type}}$ means the interaction energy obtained in a similar way as that in UHBD where the ligand polarization is not present. U_{nonpolar} is a nonpolar contribution from a simple surface term. (b2) The $x-, y-, z-$ components of forces, and torques as functions of the separation distances along a predefined direction.

to construct a trivariate function in the vicinity of the molecular surface. For each triangular element on the surface, we construct a small three-sided prism as shown in Fig. 3. In the prism, the potential is linearly interpolated, and the total PB force F and torque M acting on each molecule are calculated by integrations of $F = \int_S T(x) \cdot dS(x)$ and $M = \int_S r_c(x) \times [T(x) \cdot dS(x)]$, where $r_c(x)$ represents a vector from the center of mass of the target molecule to the surface point x, and the dot and cross vector multiplications are applied to the vector and tensor quantities.

Computational performance.
To assess the accuracy of the algorithm, we first consider a spherical cavity of radius 50 *Å* with one positive charge located at its center, and compare the numerical solutions with analytical ones. The surface is discretized at various resolution levels (from 320 to 81, 920 elements) by recursively subdividing an icosahedron. Numerical results show that the relative potential error decreases with increased number of elements, from $\sim$ 8% (320 BEs) to less than 0.01% (81,920 BEs).

As for the efficiency, we noticed that regardless of the surface resolution, the GMRES iteration steps never exceed 10, which numerically confirms that the derivative BEM formulation is well-conditioned.

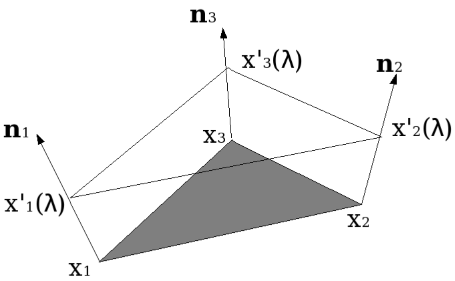


Figure 3: The prism constructed on a triangular element. The shadowed triangle is one of the boundary elements, n_1, n_2, n_3 are three unit normal vectors at the three nodes, λ is a parameter to describe the third dimensional position of the prism.

Further, in each iteration, we compare the new version of FMM with direct method for different resolutions (up to 81,920 BEs). Numerical results in Fig. 4 show that the CPU time (on a Dell dual 2.0 GHz P4 desktop with 2 GB memory) for the new version of FMM scales linearly with the number of BEs with correlation coefficient 0.984, and quadratically for the direct integration method with correlation coefficient 0.999. For a system with 81,920 surface elements, the $O(N)$ new version FMM is approximately 40 times faster than direct method.

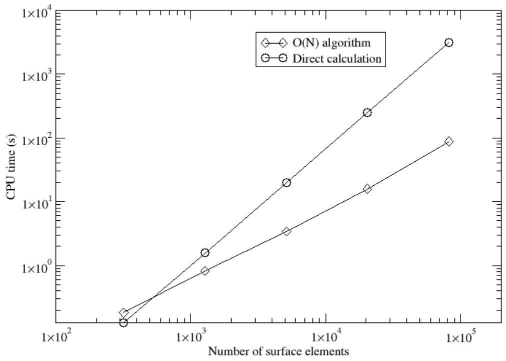


Figure 4: Log-log plot of CPU time vs. the number of elements for the calculation on a sphere case.

The memory requirements of our methods are tested on large biomolecule systems with N_{atom} atoms. Numerical experiments show that the overall memory requirement scales linearly with the number of surface elements. Compared with existing finite difference and finite element schemes, orders of magnitude reduction in memory usage has been observed in simulations on a nicotinic receptor (30, 385 atoms) with 194, 428 elements and 97, 119 vertices. In our algorithm, we noticed that 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 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. We note that without sacrificing the accuracy, the number of near-field elements for each vertex can normally be up-bounded by a fixed number s. Hence, the size of neighboring list is also up-bounded by sN, which leads to $O(N)$ overall memory usage.

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.[20] For all calculations, the AMBER atomic charges and radii were assigned.

A probe radius of 1.5 $\AA$ was used to define the dielectric interface. The dielectric constants were taken as 2.0 for solute and 80.0 for solvent. 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, and it solves the PB equations twice to obtain the solvation energy. Nevertheless, the preliminary results given below show that present algorithm provides better speed and memory performance than the current version of APBS. For APBS calculations, when using a $161 \times 129 \times 161$ grid with grid spacing of 0.25 $\AA$, the computed electrostatic solvation energy is -525.5 kcal/mol, and the calculation takes 250.8 seconds of total CPU time and 742.8 megabytes of memory on our desktop machine. When using a finer grid of $225 \times 161 \times 225$, the total CPU time is increased to 599.9 seconds, memory increased to 1784.6 megabytes and the total solvation energy is -522.8 kcal/mol. For our calculation to achieve the same level of accuracy, the surface mesh was generated with vertex density of 3 $\AA^{-2}$, which results in a total of 21, 430 triangular elements and 10, 717 vertices. In this case, the computed solvation energy is -522.0 kcal/mol, and the calculation takes 129 seconds requiring only 90 megabytes of memory if running in a memory-saving mode while the job completes in 44 seconds requiring 486 megabytes of memory if running in a memory-intensive mode.

Protein-protein interaction of the acetylcholinesterase (AChE) and fasciculinII (Fas2). Many experimental and theoretical studies have established that electrostatic interactions dominate the AChE-Fas2 binding process, and increase the binding rate by about two orders of magnitude.[21,22] However, in the initial Brownian dynamics (BD) simulations of AChE-Fas2 encounter, the methods for solving electrostatics are not rigorous in the sense that the polarization and electrostatic desolvation effects are neglected to reduce the computational cost. Using these approximate methods, the calculated encounter rates tend to be overestimated especially at high ionic concentration. In seeking to demonstrate the energetic discrepancies that may occur by using our BEM-based method as compared to the previous one, we calculate the interaction energies, forces and torques for a series of structures at different separation distances between AChE and Fas2. These structures are generated by displacing Fas2 away from the binding site, along a selected direction with possibly least clashes. The AChE-Fas2 distance in bound complex is about 28.3 $\AA$. In all calculations, the ion concentration is set to 50 mM, which is equivalent to a Debye-Hückel screening length of 13.8 $\AA$. The meshes are generated at a density of 1.0 $\AA^{-2}$. A single mesh is generated if two molecular surfaces are separated by less than 3 $\AA$, while for the further separations the system is treated as two separate domains with two meshes.

Fig. 2 (a) shows the mutually polarized electrostatic potentials mapped to the molecular surfaces of AChE and Fas2 at a $\sim 14\AA$ displacement of Fas2. Not surprisingly, the potential surfaces exhibit qualitative electrostatic complementarity at the binding interface. Fig. 2 (b1) shows the electrostatic interaction energy and electrostatic desolvation profiles for the AChE-Fas2 complex as a function of center-center distances. Although the data at short range may not be quantitatively accurate due to the atomic and mesh clashes when AChE and Fas2 are close in (from $\sim$29.0 $\AA$ to 33.0 $\AA$), Fig. 2 shows some interesting results that will not be expected from previous approximate models. Clearly, the electrostatic interaction energy (black line) is favorable for binding at separations further than 33 $\AA$, but becomes increasingly positive at closer separations. The long-range electrostatic attraction is the dominant driving force for Fas2-AChE binding, which accounts for the observed electrostatic enhancement of the binding rate in experiments. However, given the fact that the AChE-Fas2 complex has a high binding affinity, the unfavorably high positive electrostatic energies at closer distances seem to be surprising. This should be balanced by the non-polar interactions. If we take a simple model to add the surface term $U_{\text{nonpolar}} = \gamma \Delta S$, $\gamma = 0.058$ kcal$\cdot$mol$^{-1} \cdot \AA^{-2}$,[23] to account for the nonpolar contributions, the total binding energy profile (the purple line shown in Fig. 2 (b1)) will show favorable interactions for the AChE-Fas2 complex.

The origin of the large unfavorable electrostatic interaction at closer 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 (Elcock et al.[22]). The green and red lines in Fig. 2 (b1) show the electrostatic desolvation energies of AChE and Fas2 respectively. When AChE and Fas2 stay close, there are large desolvation penalties, but the electrostatic desolvation energies decrease rapidly when two molecules are separated by $\sim$5 $\AA$ or further.

Another interesting observation is that the electrostatic interaction energy profile shows a minimum at the distance of $\sim 33\AA$ (Fig. 2 (b1)), which corresponds to a 5 $\AA$ displacement of Fas2 from the complex structure. The presence of this energy minimum arises from the polarization effects that tend to minimize the total interaction energy when Fas2 is close to AChE.

The present BEM method gives the full PB interaction energy that inherently takes into account both the desolvation and polarization contributions from two proteins. In conventional electrostatic interaction calculations as in the UHBD package[24] for protein-substrate systems, the reaction field of only one molecule (usually protein) is computed, and then acts on the atomic charges of the other one. The blue line in Fig. 2 (b1) shows the interaction energies obtained with this type of calculation. Whereas it is in good agreement with the full PB energies at large separations ($> 8\AA$), it deviates greatly at short distances, which emphasizes the importance of using more rigorous PB electrostatics in simulating the AChE-Fas2 encounter process.

Although more work is presently underway to combine this code with BD simulations for calculating association rates of enzyme-substrate or protein-protein encounters, we show some early results on the force and torque calculations in Fig. 2 (b2). The forces and torques are more sensitive to the atom/mesh clashes, which exhibit very large fluctuations at short ranges below 5 Å(data not shown). Across the whole separation range, the forces along x and y directions are close to zero, while the z component varies from -1.0 to -0.65 kcal$\cdot mol^{-1} \cdot \AA^{-1}$. Since the direction of the Fas2 displacement is close to the z-axis (-0.4, 0.2, 0.89), the force results are consistent with the energy calculations, and also suggest that this direction may be close to one of the real association pathways within this spatial range. Fig. 2 (b2) also shows torque calculation in all three x, y and z directions. Those significant values suggest that the molecular orientation will be adjusted along this association pathway.

Conclusions and discussion

In this paper, an efficient algorithm with optimal computational complexity is presented for the numerical solution of the linearized PB electrostatics. It uses a BIE formulation with unknowns defined only on the surface, and is accelerated by the new version of FMM and Krylov subspace methods. The algorithm enables the computational study of relatively large biological systems ($\sim$ hundreds of thousands atoms) on a PC computer, and has been applied to the simulation of AChE and Fas2 protein-protein interactions.

Unfortunately, all-atom molecular dynamics and BD simulations with full PB calculation for large systems still exceed the presently available computer capability. To overcome this hurdle, several techniques are being pursued to further increase the efficiency of the present algorithm for dynamical simulations, including (a) parallelization of the present code and (b) a new multiscale time stepping method which utilizes the efficiency of our algorithm for electrostatics calculations. For (a), previous studies show that the BIE method and new version of FMM have excellent scalability for parallel computation; and for (b), as different temporal scales are readily available in the FMM structures, larger time step sizes can be used for the slowly varying far field interactions represented by the local and multipole expansions, and smaller step sizes for the rapid local interactions. In addition, the present fast BIE framework can be readily extended to solving other equations, such as the diffusion equations arising from the study of ion permeation and ligand diffusion processes. Results along these directions will be reported in the future.

The work of BL, XC and JAM was supported in part by the NIH, NSF, the Howard Hughes Medical Institute, National Biomedical Computing Resource, the NSF Center for Theoretical Biological Physics, SDSC, the W. M. Keck Foundation, and Accelrys, Inc. The work of JH was supported in part by the NSF under grants DMS0411920 and DMS0327896.

References

1. Appel, A. W. (1985) *SIAM J. Sci. Stat. Comput.* **6**, 85–103.
2. Barnes, J & Hut, P. (1986) *Nature* **324**, 446 – 449.
3. Darden, T, York, D, & Pedersen, L. (1993) *J. Chem. Phys.* **98**, 10089–10092.
4. Greengard, L & Rokhlin, V. (1987) *J. Comput. Phys.* **73**, 325–348.
5. Greengard, L & Rokhlin, V. (1997) *Acta Numerica* **6**, 229–269.
6. Rokhlin, V. (1983) *Wave Motion* **5**, 257–272.
7. Juffer, A. H, Botta, E. F. F, Vankeulen, B. A. M, Vanderploeg, A, & Berendsen, H. J. C. (1991) *J. Comput. Phys.* **97**, 144–171.
8. Liang, J & Subramaniam, S. (1997) *Biophys. J.* **73**, 1830–1841.
9. Boschitsch, A. H, Fenley, M. O, & Zhou, H. X. (2002) *J. Phys. Chem. B* **106**, 2741–2754.
10. Greengard, L. F & Huang, J. F. (2002) *J. Comput. Phys.* **180**, 642–658.
11. Lu, B. Z, Zhang, D. Q, & McCammon, J. A. (2005) *J. Chem. Phys.* **122**, 214102.
12. Lu, B. Z, Cheng, X. L, Hou, T. J, & McCammon, J. A. (2005) *J. Chem. Phys.* **123**, 084904.
13. Rudolphi, T. J. (1991) *Math. Comput. Model.* **15**, 269–278.
14. Ingber, M. S & Rudolphi, T. J. (1990) *Appl. Math. Model.* **14**, 536–543.
15. Tanaka, M, Sladek, V, & Sladek, J. (1994) *AMSE Appl. Mech. Rev.* **47**, 457–499.
16. Zhou, H. X. (1993) *Biophys. J.* **65**, 955–963.
17. da Cunha, R. D & Hopkins, T. (1997) *URL, http://www.mat.ufrgs.br/pim-e.html*.
18. Sanner, M. F, Olson, A. J, & Spehner, J. C. (1996) *Biopolymers* **38**, 305–320.
19. Gilson, M. K, Davis, M. E, Luty, B. A, & McCammon, J. A. (1993) *J. Phys. Chem.* **97**, 3591–3600.
20. Baker, N. A, Sept, D, Joseph, S, Holst, M. J, & McCammon, J. A. (2001) *Proc. Natl. Acad. Sci. U. S. A.* **98**, 10037–10041.
21. Radic, Z, Quinn, D. M, McCammon, J. A, & Taylor, P. (1997) *J. Biol. Chem.* **272**, 23265–23277.
22. Elcock, A. H, Gabdoulline, R. R, Wade, R. C, & McCammon, J. A. (1999) *J. Mol. Biol.* **291**, 149–162.
23. Nicholls, A, Sharp, K. A, & Honig, B. (1991) *Proteins* **11**, 281–296.
24. Madura, J. D, Briggs, J. M, Wade, R. C, Davis, M. E, Luty, B. A, Ilin, A, Antosiewicz, J, Gilson, M. K, Bagheri, B, Scott, L. R, & McCammon, J. A. (1995) *Comput. Phys. Commun.* **91**, 57–95.