Treecode-based generalized Born method

Zhenli Xu,[1,a)] Xiaolin Cheng,[2,b)] and Haizhao Yang[3]
[1]*Department of Mathematics, and Institute of Natural Sciences, Shanghai Jiao Tong University, Shanghai 200240, China*
[2]*Center for Molecular Biophysics, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, USA*
[3]*Department of Mathematics, Shanghai Jiao Tong University, Shanghai 200240, China*

(Received 18 September 2010; accepted 16 January 2011; published online 9 February 2011)

We have developed a treecode-based $O(N \log N)$ algorithm for the generalized Born (GB) implicit solvation model. Our treecode-based GB (tGB) is based on the GBr6 [J. Phys. Chem. B **111**, 3055 (2007)], an analytical GB method with a pairwise descreening approximation for the R6 volume integral expression. The algorithm is composed of a cutoff scheme for the effective Born radii calculation, and a treecode implementation of the GB charge–charge pair interactions. Test results demonstrate that the tGB algorithm can reproduce the vdW surface based Poisson solvation energy with an average relative error less than 0.6% while providing an almost linear-scaling calculation for a representative set of 25 proteins with different sizes (from 2815 atoms to 65456 atoms). For a typical system of 10k atoms, the tGB calculation is three times faster than the direct summation as implemented in the original GBr6 model. Thus, our tGB method provides an efficient way for performing implicit solvent GB simulations of larger biomolecular systems at longer time scales. © *2011 American Institute of Physics*. [doi:10.1063/1.3552945]

I. INTRODUCTION

Electrostatic interaction plays an essential role in many molecular and cellular processes.[1–6] Due to its long-range nature, an accurate but efficient treatment of electrostatics is the key in fast computer simulations of biological systems.[7–11]

The Possion–Boltzmann (PB) model[12,13] is one of the most widely used implicit solvation models that describe electrostatic interactions of biological systems in aqueous solutions under a mean field approximation. PB treats the solvent as a high dielectric continuum medium and thus eliminates the degrees of freedom of the explicit solvent molecules;[14–18] the solute is described as an array of charged atoms embedded in a low dielectric medium. However, because the dielectric interface between the solute and the solvent is irregular in shape for most biomolecules, the corresponding PB equations have to be solved numerically[10] by finite difference, finite element or boundary element discretization method. These numerical methods often involve computationally intensive matrix inversion or molecular surface determination, thus, limiting the application of PB to molecular dynamics (MD) simulations of biomolecules. Only very few PB MD simulations of small proteins have been reported.

In 1990, Still *et al.*[19] proposed the generalized Born (GB) model as an alternative to the numerical solution of the PB equation. In GB, the electrostatic solvation energy is given by a sum of pairwise interactions between the solute atoms using the so-called effective Born radii of individual atoms to take into account the polarization effects of the bulk solvent. Since the work of Still *et al.*, many studies have been reported on Born radius expressions,[20–27] fast numerical algorithms,[28–34] and various successful applications.[35–43] In particular, Onufriev *et al.*[44] demonstrated that the GB formula reproduces the PB results with an accuracy of around 1% as long as the effective Born radii can be accurately computed. The effective Born radius of an atom characterizes its degree of burial inside the solute molecule and can be calculated as a volume integral over the molecular surface bounded region. Most existing methods for computing the effective Born radius are based on the Coulomb-field approximation that may lead to large errors, especially for the surface atoms. To address this difficulty, Grycuk[23] proposed an alternative method based on a R6 volume integral expression, which is derived from a spherical cavity at the conductor limit.

The R6 GB model has been shown to be more accurate in reproducing the PB results.[23,45,46] Recently, Tjong and Zhou[46] reported an efficient implementation of the R6 GB, by using the pairwise descreening approximation of Gallicchio and Levy.[32] Despite its high accuracy, an $O(N^2)$ computational complexity remains due to the pairwise nature of both the effective Born radii and the pairwise charge–charge calculations, where N is the number of particles in the solute. Therefore, for biomolecules with more than ten thousand atoms, the GB method remains computationally demanding. It is even less competitive than the explicit solvent models, for which fast algorithms such as the FFT-based particle-mesh Ewald method have been widely applied to achieve an overall $O(N \log N)$ complexity where N is the number of grid points as charges are interpolated into the grid points to use the FFT.

The GB algorithm consists of two major time-consuming steps: the calculation of effective Born radii and the summation of interactions over all the atom pairs, both of which currently scale as order N^2. In this paper, we develop an overall $O(N \log N)$ algorithm based on the GBr6 model that improves scalability in both steps. First, we introduce a

[a)] Electronic mail: xuzl@sjtu.edu.cn.
[b)] Electronic mail: chengx@ornl.gov.

cutoff scheme for the exclusion of distant atom pairs for their descreening contributions in the effective Born radii calculation. In practical simulations of large biological systems[47,48] with GB, very large cutoff radius (50 Å), or even without cutoff, has to be performed for the pairwise descreening interactions due to the longer ranged nature of the Coulomb-field approximation than the R6 model, which was implemented in AMBER.[39] In using the R6 GB, we find a cutoff radius of 8 Å provides very accurate approximation (around 0.1% in relative error in free energy) to direct pairwise summation.

Second, we develop a treecode algorithm for fast summation of all the pairwise charge–charge interactions. The hierarchical treecode algorithms[49–51] are widely used strategies to speed up the calculation of pairwise particle interactions in various applications. The basic idea of the treecode is to divide the particles into nested clusters with an octree structure. Then the aggregate effect of the particles in a cluster can be computed by their multipole expansions if an outside particle is sufficiently far from the cluster. The tree code reduces the computational complexity from $O(N^2)$ to $O(N \log N)$, which can be further reduced to $O(N)$ in the fast multipole method (FMM) (Refs. 52–55) where the far-field effects are accumulated into a single local expansion. The FMM has been successfully applied to numerical solution of the linearized PB equation based on the boundary element discretization.[56,57] However, the application of the treecode algorithm to the GB models has not been reported possibly because the interaction kernel in GB is more complex, and thus it is not so straightforward to do multipole expansions with respect to the local coordinates.

The organization of this paper is as follows. In Sec. II, we present the theory and numerical algorithm for the tGB model. In Sec. III, test calculations are performed to demonstrate the accuracy and efficiency of the algorithm. Conclusions are made in Sec. IV.

II. THEORY AND ALGORITHM

A. Generalized Born formalism

In the GB formulation of Still *et al.*,[19] the electrostatic contribution of the solvation free energy, ΔG_{elec}, is computed by the sum of the over all pairs of the atoms,

$$\Delta G_{\text{elec}} = -\frac{1}{2}\left(\frac{1}{\epsilon_{\text{i}}} - \frac{1}{\epsilon_{\text{o}}}\right) \times \sum_{i,j=1}^{N} \frac{q_i q_j}{\sqrt{r_{ij}^2 + R_i R_j \exp\left(-4r_{ij}^2/R_i R_j\right)}}, \tag{1}$$

where q_i and q_j are the atomic charges, ϵ_{i} and ϵ_{o} are the interior and exterior dielectrics, r_{ij} is the interparticle distance, N is the total number of atoms in the system, and R_i is the effective Born radius of atom i that describes how deeply the charge i is buried in the low-dielectric (biomolecular) medium. According to Eq. (1), the solvation free energy ΔG_{elec} can be decomposed into the summation of the self-energies of individual atoms and the summation of pair interaction energies. The effective Born radius R_i of the ith atom can be defined by

$$R_i = -\frac{1}{2}\left(\frac{1}{\epsilon_{\text{i}}} - \frac{1}{\epsilon_{\text{o}}}\right)\frac{q_i^2}{\Delta G_{\text{elec}}^i}, \tag{2}$$

where ΔG_{elec}^i is the self-electrostatic solvation energy of charge i in the same dielectric environment but in the absence of other charges. Ideally, the self-energy can be computed directly by numerically solving the Poisson equation, which leads to the so-called "perfect" effective Born radius.[44] This approach is, however, impractical for irregular geometries due to its high computational cost.

Traditionally, the effective Born radius is determined by using the Coulomb-field approximation (CFA),[35] which assumes the electric displacement flux remains in the Coulombic form when the dielectric varies from ϵ_{i} to ϵ_{o} in the solvation process, leading to a volume integral expression of kernel $1/r^4$ over the exterior domain of the solute, Ω_{ex},

$$\frac{1}{R_i} = \frac{1}{4\pi}\int_{\Omega_{\text{ex}}} \frac{d^3\mathbf{r}}{|\mathbf{r} - \mathbf{r}_i|^4}. \tag{3}$$

The accuracy of the CFA has been assessed by comparing to reference solutions[20,35,44,58,59] such as the exact solutions of simplified models and the numerical solutions of PB equations. It was found that the CFA heavily overestimates the Born radii for off-center charges. For example, the maximum relative error may reach 50% for atoms near the surface of a spherical solute. Empirical corrections have been proposed by combining the CFA with other volume integrals,[20,21] which provided better approximations to the "perfect" Born radii.

An alternative also with a single volume integral is the R6 model proposed by Grycuk,[23] which is the volume integral of kernel $1/r^6$,

$$\frac{1}{R_i^3} = \frac{3}{4\pi}\int_{\Omega_{\text{ex}}} \frac{d^3\mathbf{r}}{|\mathbf{r} - \mathbf{r}_i|^6}. \tag{4}$$

Equation (4) is an unbounded integral on the exterior domain of the molecule, which is often transformed into the exterior integral of the van der Waals (vdW) sphere subtracting a bounded one,

$$\frac{1}{R_i^3} = \frac{1}{a_i^3} - \frac{3}{4\pi}\int_{\Omega_{\text{in}}\setminus A_i} \frac{d^3\mathbf{r}}{|\mathbf{r} - \mathbf{r}_i|^6}, \tag{5}$$

where A_i is the vdW sphere of atom i and a_i is its radius. The basic idea behind the R6 model is that the self-energy of every atom in a spherical solute can be exactly represented by a series with the Kirkwood expansion.[60] In the conductor limit of the solvent dielectric, the solution leads to the integral in Eq. (4). For solutes of arbitrary geometries, the R6 model can be solved by discretizing the volume integral, which has also been shown to yield very good approximation[45,46,61] to the PB energies.

B. Pairwise descreening approximation

Most of the existing GB methods used in MD simulations are based on the analytical pairwise descreening summation, in which the union of the atomic vdW spheres comprises the molecular volume of the solute, and its surface is known as the vdW surface. The basic idea of the pairwise descreening summation is to split the integral in Eq. (5) into a weighted sum of integrals defined on each of vdW spheres, where the weights are the scaling coefficients used to offset of the overlap between spheres. In the GBr6,[46] the integral (5) is approximated by

$$\frac{1}{R_i^3} = \frac{1}{a_i^3} - \frac{3}{4\pi}\sum_j s_{ji} H_{ji}(a_i, a_j, r_{ij}), \tag{6}$$

with

$$H_{ji}(a_i, a_j, r_{ij}) = \int_{A_j \setminus A_i} \frac{1}{|\mathbf{r} - \mathbf{r}_i|^6} d^3\mathbf{r}, \tag{7}$$

where A_j is the vdW sphere of radius a_j of the jth atom, and the sum runs over all the spheres with interparticle distance r_{ij} larger than a_i, and $A_j \setminus A_i$ denotes the domain of A_j excluding the overlapping portion with A_i. The integral H_{ji} is available in an analytical form as $A_j \setminus A_i$ is a regular domain. The scaling factors s_{ji} are used to account for the possible overlap of atomic spheres, because the summation of all the spherical integrals overcounts the molecular region, leading to an overestimate of the effective Born radius.[35] The principle to choose the scaling factors is such that Eq. (6) approximates the integral (5). To address this issue, a number of strategies[32,36,62–65] have been developed to calculate the scaling factors, depending on different approximations of the overlap regions. We will focus on the method of Gallicchio and Levy,[32] which has been recently extended to the R6 expression.[46] To precisely define the overlap regions, we start from the inclusion–exclusion principle in combinatorics:

$$V = \sum_i V_i - \sum_{i<j} V_{ij} + \sum_{i<j<k} V_{ijk} + \cdots, \tag{8}$$

where $V_i = \frac{4\pi}{3}a_i^3$ is the spherical volume of atom i, V_{ij} and V_{ijk} are, respectively, the second- and third-order intersection volumes. The self-volume $\tilde{V}_i$ of atom i is defined by

$$\tilde{V}_i = V_i - \frac{1}{2}\sum_j V_{ij} + \frac{1}{3}\sum_{j<k} V_{ijk} + \cdots, \tag{9}$$

which obviously satisfies $V = \sum_i \tilde{V}_i$. The ratio between the self-volume $\tilde{V}_i$ and the spherical volume V_i of atom i, $s_i = \tilde{V}_i / V_i$, is a measure of the overlapping of atom i with all other atoms. Note in the calculation of s_j, the atom i is not excluded. So by adding back the contribution of atom i to the self- volume of atom j, the scaling factor s_{ji} can be written as

$$s_{ji} \approx s_j + \frac{1}{2}\frac{V_{ji}}{V_j}. \tag{10}$$

Exact calculations of the intersection volumes are expensive, and an approximate algorithm was developed by Grant and Pickup,[66] in which the volume density of each atom is described by a Gaussian density function

$$\rho_i(\mathbf{r}) = \frac{4\pi}{3}\left(\frac{\mu}{\pi}\right)^{3/2} e^{-c_i(\mathbf{r}-\mathbf{r}_i)^2}, \tag{11}$$

where $c_i = \mu/a_i^2$ and $\mu = 2.227$ is a dimensionless parameter which was empirically determined,[66] and the integration of the density function on the full 3D space leads to the volume of the vdW sphere of atom i. The overlap volume of atoms $j_1, \cdots, j_m$ can then be approximated by the integral of the product of their density functions, $\int d^3\mathbf{r}\rho_{j_1}(\mathbf{r})\cdots\rho_{j_m}(\mathbf{r})$, which can be evaluated analytically, resulting in the following expression:

$$V_{j_1,\cdots,j_m} \approx \left[\frac{4\pi}{3}\left(\frac{\mu}{\pi}\right)^{3/2}\right]^m \left(\frac{\pi}{C}\right)^{\frac{3}{2}} \times \exp\left(-\frac{1}{C}\sum_{p=1}^{m}\sum_{q=p+1}^{m} c_{j_p} c_{j_q} r^2_{j_p,j_q}\right), \tag{12}$$

with $C = \sum_{p=1}^m c_{j_p}$.

The integral H_{ji} can be computed exactly due to the simple integration region. Direct evaluation of H_{ji} and s_{ji} has an $O(N^2)$ complexity. The FFT-accelerated method[30,31] has been developed for the Born radii calculations to achieve a linear arithmetic complexity with respect to the number of grid points. In this grid-based algorithm, the number of grid points is usually taken as a large number in order to sample the dielectric jump on the molecular surface, which makes the algorithm less efficient for many systems of interest. Actually, the linear-scaling calculation for H_{ji} and s_{ji} can be obtained by a simple idea. Note from Eq. (7) that in the R6 formulation, H_{ji} decays rapidly as the distance, r_{ij}, between spheres A_i and A_j increases (the same as the dispersion term in the Lennard-Jones potential). Therefore, a direct cutoff scheme can be used without loss of much accuracy, leading to a linear-scaling algorithm. The cell method[67] is employed to search for neighbors of the atoms within the cutoff distance in $O(N)$ complexity. The cell method partitions the simulation box into disjoint cells. For a given atom i, all the atoms in the same and neighboring cells are considered neighbors of atom i. It should be noted that, when the method is applied in dynamics simulations, atoms could go in and out of the cut distance and the neighboring list of each atom should be updated at certain time steps. The analytical results of H_{ji} and their derivatives are given below in four possible situations:

1. If the spheres A_i and A_j do not overlap and their interparticle distance $a_i + a_j < r_{ij} < L_c$, where L_c is the cutoff distance, then

$$\begin{cases} H_{ji} = \dfrac{a_j^3}{\left(r_{ij}^2 - a_j^2\right)^3} \\ \dfrac{\partial H_{ji}}{\partial r_{ij}} = \dfrac{-6 r_{ij} a_j^3}{\left(r_{ij}^2 - a_j^2\right)^4} \\ \dfrac{\partial^2 H_{ji}}{\partial r_{ij}^2} = \dfrac{6 a_j^3\left(7 r_{ij}^2 + a_j^2\right)}{\left(r_{ij}^2 - a_j^2\right)^5} \end{cases}. \tag{13}$$

2. If the two spheres overlap but neither is completely inside the other one, i.e., $|a_i - a_j| \le r_{ij} < a_i + a_j$, then

$$\begin{cases} H_{ji} = \dfrac{1}{16r_{ij}}\left[-6\left(\dfrac{1}{a_i^2}-\dfrac{1}{(r_{ij}+a_j)^2}\right)+8r_{ij}\left(\dfrac{1}{a_i^3}-\dfrac{1}{(r_{ij}+a_j)^3}\right)-3(r_{ij}^2-a_j^2)\left(\dfrac{1}{a_i^4}-\dfrac{1}{(r_{ij}+a_j)^4}\right)\right] \\ \dfrac{\partial H_{ji}}{\partial r_{ij}} = -\dfrac{3}{16a_i^4 r_{ij}^2}\left[-2a_i^2+a_j^2+r_{ij}^2+\dfrac{a_i^4\left(a_j^2+4a_j r_{ij}+r_{ij}^2\right)}{(a_j+r_{ij})^4}\right] \\ \dfrac{\partial^2 H_{ji}}{\partial r_{ij}^2} = \dfrac{3}{8a_i^4 r_{ij}^3}\left[-2a_i^2+a_j^2+\dfrac{a_i^4\left(a_j^3+5a_j^2 r_{ij}+10a_j r_{ij}^2+2r_{ij}^3\right)}{(a_j+r_{ij})^5}\right] \end{cases} . \tag{14}$$

3. If sphere A_i is completely included in A_j, i.e., $r_{ij} < |a_i - a_j|$ and $a_j \geq a_i$,

$$\begin{cases} H_{ji} = \dfrac{1}{a_i^3} - \dfrac{a_j^3}{\left(a_j^2 - r_{ij}^2\right)^3} \\ \dfrac{\partial H_{ji}}{\partial r_{ij}} = \dfrac{-6r_{ij}a_j^3}{\left(r_{ij}^2 - a_j^2\right)^4} \\ \dfrac{\partial^2 H_{ji}}{\partial r_{ij}^2} = \dfrac{6a_j^3\left(7r_{ij}^2 + a_j^2\right)}{\left(r_{ij}^2 - a_j^2\right)^5} \end{cases} . \tag{15}$$

4. If sphere A_j is completely included in A_i,

$$H_{ji} = \frac{\partial H_{ji}}{\partial r_{ij}} = \frac{\partial^2 H_{ji}}{\partial r_{ij}^2} = 0. \tag{16}$$

Once H_{ji} and their derivatives are computed, the effective Born radii are obtained from Eq. (6), and their derivatives with respect to Cartesian coordinates can be given by using the chain rule,

$$\frac{\partial R_i}{\partial x_l} = \frac{R_i^4}{3}\sum_{j\neq i} s_{ji}\frac{\partial H_{ji}}{\partial r_{ij}}\frac{\partial r_{ij}}{\partial x_l}, \tag{17}$$

$$\frac{\partial^2 R_i}{\partial x_k \partial x_l} = \frac{4}{R_i}\frac{\partial R_i}{\partial x_k}\frac{\partial R_i}{\partial x_l} - \frac{R_i^4}{3}\sum_{j\neq i} s_{ji} \times \left(\frac{\partial^2 H_{ji}}{\partial r_{ij}^2}\frac{\partial r_{ij}}{\partial x_k}\frac{\partial r_{ij}}{\partial x_l} + \frac{\partial H_{ji}}{\partial r_{ij}}\frac{\partial^2 r_{ij}}{\partial x_k \partial x_l}\right), \tag{18}$$

where $\mathbf{r}_i = (x_1, x_2, x_3)$ and $k, l = 1, 2, 3$. The scaling coefficient s_{ji} actually depends on the position of atom i, and thus it will improve the accuracy if we add the terms of its derivatives. However, mostly because V_{ji} is much smaller than V_j, we found Eqs. (17) and (18) are good approximations.

In summary, the calculation of the effective Born radii and their derivatives is composed of three steps: (i) set the cutoff radius L_c, and find the neighbors of each atom i by the cell method; (ii) compute s_{ji} and H_{ji} of each atom i with all the j atoms in the neighbor list of atom i; (iii) sum up pairwise descreening contributions from all the atoms in the neighbor list.

C. Treecode implementation for the pair-energies

To achieve an overall $O(N \log N)$ calculation, we implement the treecode algorithm to compute the solvation free energy [Eq. (1)]. In the present treecode algorithm, the particles are grouped into a hierarchical structure of nested boxes.[51,68–70] The root box, the smallest box that can encompass all the particles, is first uniformly divided into eight boxes as its children. Each child box is then recursively subdivided into eight boxes until the particle number in a box is less than a specified value N_0. By traversing the hierarchical tree, the total interactions with atom j can be represented by a sum of particle–cluster interactions for the far-field approximation and direct particle–particle interactions in the near field. A multipole acceptance criterion (MAC) is used to determine whether a particle–cluster interaction is allowed by comparing the box size, $h_A = \sqrt{\Delta x_1^2 + \Delta x_2^2 + \Delta x_3^2}/2$ where Δx_1, Δx_2, and Δx_3 are the sizes of the three box dimensions, and the particle–cluster distance, r_{jA},

$$\frac{h_A}{r_{jA}} \leq \theta, \tag{19}$$

for a predefined error-control parameter θ.

Let us consider the particle–cluster interaction of j and cluster A, where the reaction potential energy is expressed as

$$q_j\Phi_{jA} = \sum_{i\in A} q_i\Phi_j(\mathbf{r}_i) = \sum_{i\in A}\frac{\beta q_i q_j}{\sqrt{r_{ij}^2 + R_i R_j e^{-r_{ij}^2/4R_iR_j}}}, \tag{20}$$

where $\Phi_j(\cdot)$ is the reaction potential function due to charge j, and the coefficient $\beta = -1/2(1/\epsilon_{\rm i} - 1/\epsilon_{\rm o})$. To obtain a simple multipole expansion, we need to note that the influence of the exponentials is not significant, as was analyzed in the spherical model (see Fig. 5 in Grycuk[23]). So the exponential term can be re-expressed in terms of the average over all the particles in the cluster, which is a first-order approximation $O(h_A)$:

$$q_j\Phi_{jA} \approx \sum_{i\in A}\frac{\beta q_i q_j}{\sqrt{r_{ij}^2 + R(\mathbf{r}_i)\tilde{R}_{jA}}}, \text{ with } \tilde{R}_{jA} = R_j e^{-r_{jA}^2/4R_AR_j}, \tag{21}$$

where r_{jA} is the distance between $\mathbf{r}_j$ and the center of the cluster, $\mathbf{r}_A$, R_A is the average Born radius of A, and $R(\mathbf{r})$ is the Born radius function which will be approximated by using $\{R_i, i = 1, \cdots, N\}$ and their derivatives at atomic locations. For the approximation with $\tilde{R}_{jA}$ in Eq. (21), we have used the slow varying property of the exponential. Then the multipole

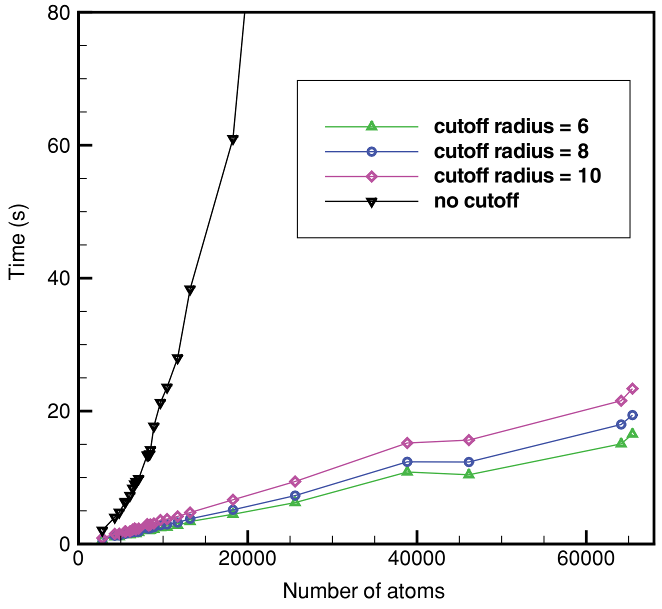

FIG. 1. The CPU times as a function of the number of atoms for calculating the Born radii of the 25 proteins with different cutoff radii $L_c = 6, 8, 10$ Å, and without cutoff. The proteins are distinguished by the number of atoms.

expansion for Φ_{jA} is given by

$$q_j \Phi_{jA} \approx \sum_{||\alpha||=0}^{p} \frac{1}{\alpha!} \partial^\alpha \Phi_j(\mathbf{r}_A) \sum_{i \in A} q_i (\mathbf{r}_i - \mathbf{r}_A)^\alpha$$

$$= \beta \sum_{||\alpha||=0}^{p} T_\alpha M_A^\alpha, \qquad (22)$$

where $\alpha = (\alpha_1, \alpha_2, \alpha_3)$ is the multi-index notation, $T_\alpha = \frac{1}{\beta \alpha!} \partial^\alpha \Phi_j(\mathbf{r}_A)$ and $M_A^\alpha = \sum_{i \in A} q_i (\mathbf{r}_i - \mathbf{r}_A)^\alpha$, respectively, represent the αth Taylor expansion coefficient (divided by β) and the moment of the cluster A. Because the moment of each cluster is only required to be computed once when generating the octree, the operations of evaluating the particle–cluster interactions are independent of the particle number inside the cluster. Thus, the complexity for computing the energy of each atom is $O(\log N)$ which is the number of the clusters interacting with the particle.

To compute T_α, we need to use the Taylor expansion of function $R(\mathbf{r})$,

$$R(\mathbf{r}) \approx \sum_{||\alpha||=0}^{p} \frac{\partial^\alpha R_A}{\alpha!} (\mathbf{r} - \mathbf{r}_A)^\alpha. \qquad (23)$$

We compute the multipole expansion until $p = 2$, and thus $\partial^\alpha R_A$ for $||\alpha|| \leq 2$ are required, which are given by taking the average within the cluster,

$$\partial^\alpha R_A = \frac{1}{N_A} \sum_{i \in A} \partial^\alpha R_i, \qquad (24)$$

where analytical expressions of $\partial^\alpha R_i$ are provided in Eqs. (17) and (18). The accuracy of the average operations depends on the particle distribution inside the cluster, which is again an error of $O(h_A)$ theoretically. But the atomic distribution in biomolecules is usually uniform, so the averaging approximation is close to the second order of accuracy, which is the same level as the multipole expansion. For those atoms near the surface, the atomic distribution in a cluster is not uniform, but their effective Born radii are small, and thus the average operation is still accurate. It should be also noted that although the approximation of the effective Born radii is $O(h_A)$, the multipole expansion (22) may be still accurate as the effective Born radii are more slowly varying than the distance r_{ij} in the denominator of the potential function. By using the Taylor expansion of $R(\mathbf{r})$, the multipole expansion coefficients can be expressed as

$$T_{\mathbf{0}} = \frac{1}{\left(r_{jA}^2 + \tilde{R}_{jA} R_A\right)^{1/2}}, \qquad (25)$$

$$T_{\mathbf{e}_k} = -\frac{1}{2}[2(\mathbf{r}_A - \mathbf{r}_j) \cdot \mathbf{e}_k + \tilde{R}_{jA} \partial^{\mathbf{e}_k} R_A](T_{\mathbf{0}})^3, \qquad (26)$$

$$T_{2\mathbf{e}_k} = \frac{3}{2} \frac{(T_{\mathbf{e}_k})^2}{T_{\mathbf{0}}} - \frac{2 + \tilde{R}_{jA} \partial^{2\mathbf{e}_k} R_A}{4} (T_{\mathbf{0}})^3, \qquad (27)$$

$$T_{\mathbf{e}_k + \mathbf{e}_l} = \frac{3 T_{\mathbf{e}_k} T_{\mathbf{e}_l}}{T_{\mathbf{0}}} - \frac{\tilde{R}_{jA} \partial^{\mathbf{e}_k + \mathbf{e}_l} R_A}{2} (T_{\mathbf{0}})^3, k \neq l, \qquad (28)$$

where $\mathbf{e}_k$ is the unit vector of the x_k axis. Here the derivatives of the effective Born radii that are required by the multipole expansion of the reaction potential are calculated analytically, leading to more accurate results than those in our previous work[70] where the derivatives were reconstructed from the difference of the effective Born radii at nonuniform locations.

The extension of the treecode implementation to the force calculation on individual atoms is straightforward. The

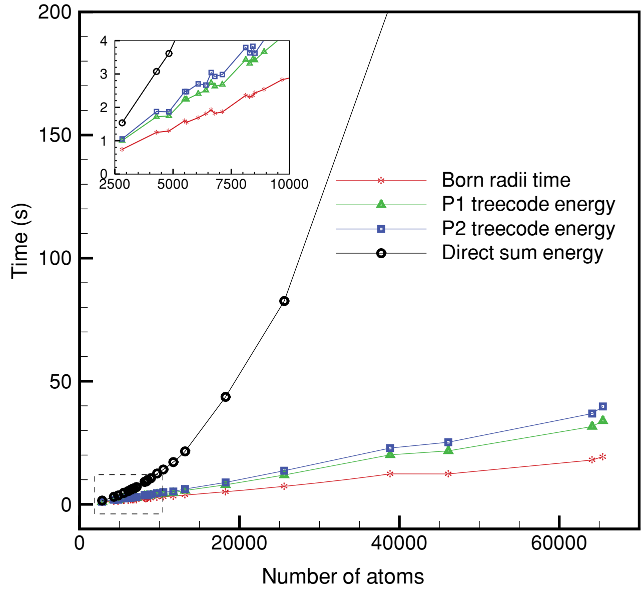

FIG. 2. The CPU times as the number of atoms for the cutoff Born radii calculation with $L_c = 8$ Å (red stars), and the effective Born radii + the pair-energies calculation with the P1 (blue Deltas) and P2 (green squares) treecode algorithms and the direct summation (black circles). The embedded frame is an enlarged picture of the dashed portion. A total of 25 proteins are calculated with each protein corresponding to a symbol on the plot.

TABLE I. List of 25 proteins, their properties (atom number N and charge Q), and PB and GB (without cutoff and with cutoff of radius 8 Å) electrostatic solvation energies (kcal/mol).

PDB	N	Q	ΔG_{PB}	ΔG_{GB}	$\Delta G_{GB}(L_c, 8\,\text{Å})$	PDB	N	Q	ΔG_{PB}	ΔG_{GB}	$\Delta G_{GB}(L_c, 8\text{Å})$
1NP4	2815	−1	−3651.218	−3663.180	−3667.844	3HL1	8515	−32	−17167.37	−17075.55	−17091.37
1FT7	4288	−17	−7327.665	−7366.804	−7375.693	3K4K	8911	−18	−11351.29	−11333.71	−11350.71
2HXK	4822	−15	−9136.084	−9106.049	−9114.244	2Q4H	9685	−12	−11289.57	−11238.23	−11256.71
3K7X	5501	−18	−8509.207	−8495.940	−8506.645	1TXG	10466	−20	−15188.16	−15306.10	−15325.28
2WZX	5579	−1	−6408.980	−6453.201	−6463.427	3K3W	11719	−12	−14979.11	−14767.00	−14791.93
1Q5U	6082	−4	−8188.450	−8140.053	−8151.062	3LHV	13188	−36	−20349.90	−20451.27	−20476.68
2II6	6409	−9	−8045.526	−8071.206	−8082.554	2WJN	18256	−1	−16409.28	−16544.59	−16578.68
3LHU	6638	−18	−9189.670	−9256.034	−9268.265	3AEU	25598	−25	−29737.34	−29839.26	−29890.13
3MEM	6793	−13	−9136.576	−9124.018	−9135.584	3KCV	38842	3	−35654.68	−35969.98	−36039.66
1V9K	7116	18	−11580.32	−11602.76	−11615.40	3HHQ	46129	21	−61990.30	−61866.17	−61956.42
3HLV	8122	−12	−10587.69	−10543.96	−10558.67	2VAT	64119	−56	−77379.36	−75517.52	−75651.76
1HL6	8291	−21	−15020.45	−15008.23	−15023.24	1VPX	65456	−175	−174231.8	−172933.3	−173063.6
3N8H	8428	10	−10147.95	−10252.84	−10267.04						

forces are defined by the negative gradient of the electrostatic solvation energy, $\mathbf{F}_i = -\nabla_{\mathbf{r}_i}(\Delta G_{\text{elec}})$, for $i = 1, \cdots, N$. The necessary multipole expansion coefficients for individual force components are obtained by taking the derivatives of Eqs. (25)–(28), which are cumbersome and not presented.

III. NUMERICAL RESULTS

Numerical tests of our tGB method are performed on a set of 25 proteins (the PDB access codes: 1NP4, 1FT7, 2HXK,3K7X, 2WZX, 1Q5U, 2II6, 3LHU, 3MEM, 1V9K, 3HLV, 1HL6, 3N8H, 3HL1, 3K4K, 2Q4H, 1TXG, 3K3W, 3LHV, 2WJN, 3AEU, 3KCV, 3HHQ, 2VAT, 1VPX for small to large sizes) collected from the RCSB Protein Data Bank. These protein structures are randomly chosen with the criterion that their atom numbers cover a wide range in order to assess the scalability of the treecode algorithm. These proteins contain 2815 to 65456 atoms and have the total charges ranging from −175 to +21e, as summarized in Table I. We use the PDB2PQR program[71] to add missing hydrogen atoms and assign AMBER atomic charges. The van der Waals radii of the atoms are assigned to the Bondi's parameters:[72] C, 1.7 Å; H, 1.2 Å; O, 1.5 Å; N, 1.55 Å and S, 1.8 Å.

The PB benchmark results were carried out with the APBS package[73] that includes a multigrid solver[74] to accelerate the computation of linear systems. To be consistent with the GB calculations, the vdW surface is specified to define the solute boundary by using the "srad 0" option in the APBS input file, where the Bondi's parameters are also used for the atomic radii. The spacing size of grids is 0.5 Å and the dimensions are 193 × 193 × 193 for the 13 proteins from 1NP4 to 3N8H, 289 × 289 × 289 for the 10 proteins from 3HL1 to 3HHQ. The last two proteins 2VAT and 1VPX are calculated with 481 × 481 × 481 grids.

As the main purpose of this paper is to develop an efficient algorithm to calculate the effective Born radii and the electrostatic solvation energy for the GB method, we fixed dielectric constants $\epsilon_{\text{i}} = 1$, $\epsilon_{\text{o}} = 78.5$, and the temperature at 300 K in all the calculations. The obtained effective Born radius is corrected by adding a constant value to each of inverse radius, $R_{\text{corr}}^{-1} = R^{-1} + 0.028\ \text{Å}^{-1}$, which was shown to greatly improve the accuracy.[45] For the dielectric-dependent scaling parameters used in pair charge–charge interactions with various dielectrics, the reader could refer to Refs. 24, 46, and 75–77. In the treecode algorithm, we set the MAC parameter $\theta = 0.4$ and the maximum number of particles in a leaf box, $N_0 = 20$, unless specified otherwise. All the calculations are performed on a Linux machine with 2.67 GHz CPUs and 48 G memory, and installed with the GNU FORTRAN 95 compiler.

A. The effective Born radii calculation

The cutoff scheme in GB simulation can be used to control the truncation of the charge–charge interactions and the descreening contributions in the effective Born radii calculation. The charge–charge interactions decay slowly with the distance, so a simple cutoff approach will be insufficient. For more efficient charge–charge interactions, we have developed a treecode algorithm that will be discussed further below. Here we focus on assessing the accuracy of a cutoff scheme for computing the descreening contributions in the R6 form.

TABLE II. Statistics for relative errors (standard deviation, average unsigned error, average signed error, and maximum error) between Poisson and GB energies,[a] $(\Delta G_{PB} - \Delta G_{GB})/\Delta G_{PB}$, and between the cutoff and non-cutoff GB energies,[b] of the set of 25 proteins. The GB energies are computed with three cutoff radii and without cutoff.

	$L_c = 6\,\text{Å}$ (%)	$L_c = 8\,\text{Å}$ (%)	$L_c = 10\,\text{Å}$ (%)	no cutoff
Standard deviation[a] (%)	0.7963	0.7837	0.7812	0.7801
Signed error[a] (%)	−0.3747	−0.1186	−0.0391	0.0189
Unsigned error[a] (%)	0.6924	0.5952	0.5919	0.5907
Maximum error[a] (%)	1.9149	2.2326	2.3312	2.4061
Standard deviation[b] (%)	0.0877	0.0318	0.0140	–
Signed errror[b] (%)	−0.3936	−0.1375	−0.0580	–
Unsigned error[b] (%)	0.3936	0.1375	0.0580	–
Maximum error[b] (%)	0.5906	0.2060	0.0872	–

[a] Errors between Poisson and GB energies.

[b] Errors between cutoff and non-cutoff GB energies.

TABLE III. Electrostatic GB solvation energies (kcal/mol) with the P1 and P2 treecodes for 25 proteins. The effective Born radii are calculated with cutoff radius 8 Å. The results are approximations to those in Table I. The parameters for the treecodes are $\theta = 0.4$ and $N_0 = 20$.

PDB	P1	P2	PDB	P1	P2
1NP4	−3659.552	− 3666.846	3HL1	− 17065.39	−17088.75
1FT7	− 7363.102	− 7375.358	3K4K	− 11352.21	−11354.68
2HXK	− 9142.064	− 9120.951	2Q4H	− 11284.30	−11260.12
3K7X	− 8509.107	− 8514.632	1TXG	− 15333.17	−15323.01
2WZX	− 6449.590	− 6469.903	3K3W	− 14834.73	−14789.34
1Q5U	− 8118.744	− 8141.391	3LHV	− 20481.17	−20481.22
2II6	− 8095.792	− 8078.902	2WJN	− 16583.05	−16586.86
3LHU	− 9279.414	− 9271.568	3AEU	− 29935.92	−29865.99
3MEM	− 9138.576	− 9135.540	3KCV	− 36121.63	−36024.44
1V9K	− 11632.82	− 11615.93	3HHQ	− 61961.89	−61938.01
3HLV	− 10566.47	− 10560.50	2VAT	− 75721.71	−75668.36
1HL6	− 15012.09	− 15023.19	1VPX	− 173114.6	−173079.6
3N8H	− 10271.22	− 10262.71			

The cutoff scheme has been previously used in AMBER,[39,78] CHARMM[79] and GROMACS.[80] To achieve a better balance between the stable simulation and the computational efficiency, various smoothing schemes have also been implemented in many MD simulation packages. However, the descreening contribution in the R6 form should converge within a short distance due to its higher order decay. We study the effect of the cutoff parameter on the accuracy of the effective Born radii and the electrostatic energy calculations by comparing the cutoff results with those of PB and GBr6. The GB electrostatic solvation energies are computed with GBr6 that has been extensively tested and shown to be very accurate in reproducing the PB energies. In Table I, the PB, GB, and GB with cutoff solvation energies of the 25 proteins are listed. Table II summarizes the statistics of the relative solvation energy errors in the entire protein dataset, including the standard deviation, average and maximum errors, with different cutoff values. Very good agreement between the GB and Poisson energies can be seen. For example, when the cutoff radius $L_c = 8$ Å, the signed and unsigned average errors are only −0.119% and 0.595% (Table II). The maximum error is 2.233% for the protein 2VAT.

The convergence of the solvation energies in comparison with the original GBr6 model without cutoff is evident with the increase of L_c. As shown in Table II, the maximum relative errors for $L_c = 6$, 8, and 10 Å are 0.591%, 0.206%, and 0.087%, respectively. Overall, the cutoff scheme provides a reasonable approximation to the noncutoff calculation, even when a small cutoff radius, 6 Å, is used. These results are not very surprising as the convergence rate of the external integral of $1/r^6$ kernel is $1/L_c^3$ that decay more rapidly with the increase of the cutoff radius. Figure 1 shows the CPU timings for computing the effective Born radii of the 25 proteins with different cutoff distances. Overall, the cutoff approach is much more efficient than the noncutoff one, showing an almost linear increase with the increase of the number of atoms. In comparison to the noncutoff approach, the use of a 8 Å cutoff speeds up the calculation by a factor of 3.8 for a protein of around 5k atoms, and a factor of 7.7 for a protein of around 10k atoms.

TABLE IV. Statistics for relative errors between treecode and direct-sum GB energies, $(\Delta G_{GB,\text{direct}} - \Delta G_{GB,\text{tree}})/\Delta G_{GB,\text{direct}}$, and between treecode GB and Poisson energies, $(\Delta G_{PB} - \Delta G_{GB,\text{tree}})/\Delta G_{PB}$, of the set of 25 proteins. The effective Born radii in the GB energies are computed with the cutoff radius $L_c = 8$ Å.

	P1, GB	P2, GB	P1, PB	P2, PB
Standard deviation (%)	0.1785	0.0574	0.7854	0.7845
Signed error (%)	−0.1712	−0.1397	−0.1521	−0.1208
Unsigned error (%)	0.2133	0.1397	0.5956	0.5969
Maximum error (%)	0.4586	0.2588	2.1422	2.2112

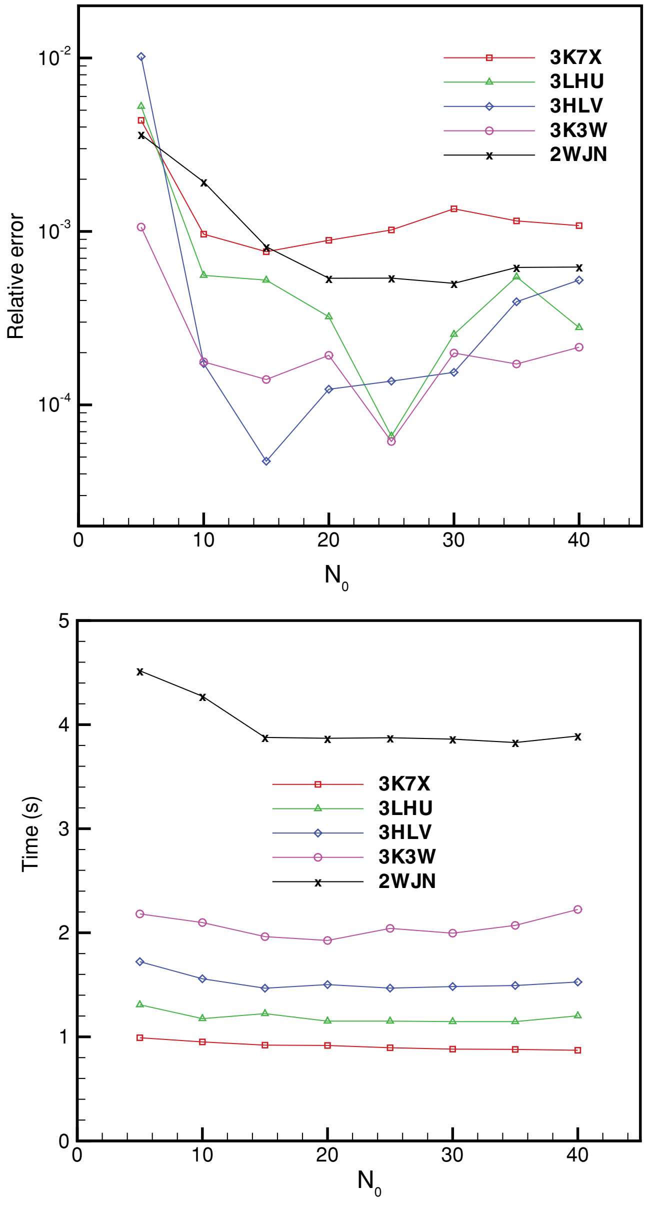

FIG. 3. Error performance in solvation energy and CPU times of the P2 treecode results for the five proteins (3K7X, 3LHU, 3HLV, 3K3W, 2WJN) for varying N_0.

TABLE V. Error performance (average and maximum errors) for the force calculation with the P2 treecode for 25 proteins. The MAC parameters $\theta = 0.4$ and $\theta = 0.3$.

	Absolute error (kcal/mol/Å)		Relative error (%)			Absolute error (kcal/mol/Å)		Relative error (%)	
PDB	$\theta = 0.4$	$\theta = 0.3$	$\theta = 0.4$	$\theta = 0.3$	PDB	$\theta = 0.4$	$\theta = 0.3$	$\theta = 0.4$	$\theta = 0.3$
1NP4	3.627e-3	8.496e-4	3.441	0.806	3HL1	1.820 e-3	5.638e-4	2.082	0.645
1FT7	4.739e-3	1.254e-3	4.037	1.068	3K4K	1.862 e-3	6.071e-4	3.443	1.122
2HXK	2.403e-3	7.635e-4	2.260	0.718	2Q4H	1.838 e-3	6.060e-4	3.427	1.129
3K7X	3.553e-3	1.082e-3	4.767	1.452	1TXG	2.353 e-3	7.546e-4	3.581	1.148
2WZX	3.178e-3	1.046e-3	4.170	1.373	3K3W	1.709 e-3	5.444e-4	3.170	1.009
1Q5U	2.670e-3	8.676e-4	3.351	1.088	3LHV	1.539 e-3	5.051e-4	3.267	1.072
2II6	2.306e-3	7.249e-4	3.362	1.056	2WJN	1.336 e-3	4.358e-4	3.697	1.206
3LHU	2.941e-3	9.529e-4	4.555	1.476	3AEU	1.768 e-3	5.399e-4	5.141	1.569
3MEM	2.331e-3	7.633e-4	3.578	1.171	3KCV	1.276 e-3	4.187e-4	4.407	1.445
1V9K	2.330e-3	7.430e-4	2.850	0.909	3HHQ	1.081 e-3	3.697e-4	3.574	1.222
3HLV	2.411e-3	8.348e-4	3.604	1.248	2VAT	8.272 e-4	3.473e-4	2.402	1.734
1HL6	1.908e-3	6.232e-4	2.505	0.818	1VPX	1.336 e-3	2.789e-4	3.697	0.809
3N8H	2.581e-3	8.058e-4	4.231	1.321					

B. Performance of the treecode algorithm

To assess the performance of the treecode algorithm, we carry out calculations for all the 25 proteins with the cutoff scheme ($L_c = 8$ Å) for the effective Born radii calculation. The solvation energies of the 25 proteins with the P1 (the first order multipole expansion) and P2 (the second order expansion) treecode algorithms are listed in Table III, while the statistics for the relative errors by comparing with the direct summation GB and the Poisson energies are given in Table IV. The average unsigned error of the P1 tree code for the 25 proteins is 0.2133%, and that of the P2 is 0.1397%, demonstrating that the treecode algorithm provides a good approximation to the pairwise interactions. It is also shown that the treecode-based GB energies approximate the Poisson energies with an average relative error less than 0.6%.

The CPU times for computing the effective Born radii, the P1 and P2 treecode energies, along with the direct summation, are shown in Fig. 2. The CPU times for both the effective Born radii calculation and the treecode algorithm increase almost linearly with the particle number. The CPU times for both the P1 and P2 treecodes are already less than that of the direct summation for the smallest protein (1NP4, 2815 atoms), suggesting the prefactor for the treecode algorithm is reasonably small. Accounting the CPU times for the Born radii calculation with the cutoff scheme ($L_c = 8$ Å), when the particle numbers are 10k, 20k, and 40k, the total speed-ups of P1 are 3.10, 5.96, and 10.74, and those of P2 are 2.79, 5.26, and 9.41, respectively.

The parameter N_0 is defined as the maximum number of particles in the leaf box. To study how the parameter N_0 influences the performance of the treecode algorithm, we perform the calculation with the P2 multipole expansion on five example proteins. The relative energy errors in comparison with those computed with the direct summation remain small (0.1%) for $N_0 \in [15, 25]$, as shown in Fig. 3. But the errors increase dramatically if N_0 is too small, which is reasonable because the averaging operation in Eq. (24) is statistically less accurate if the particle number in a box is too small. Furthermore, the CPU time curves remain roughly flat for $N_0 \in [15, 25]$, indicating both the computational accuracy and cost are insensitive to N_0 when $N_0 \in [15, 25]$. But dramatic changes in the CPU time such as proteins 3K3W and 2WJN are observed if N_0 takes a value outside this interval.

C. Accuracy of the force calculation

We also implement the code for the force calculation on individual atoms. To test it, the effective Born radii are computed with a cutoff radius of $L_c = 8$ Å. We use the P2 expansion, $N_0 = 20$, and two MAC parameters $\theta = 0.4$ and 0.3. Compared with the energy calculation, the force calculation

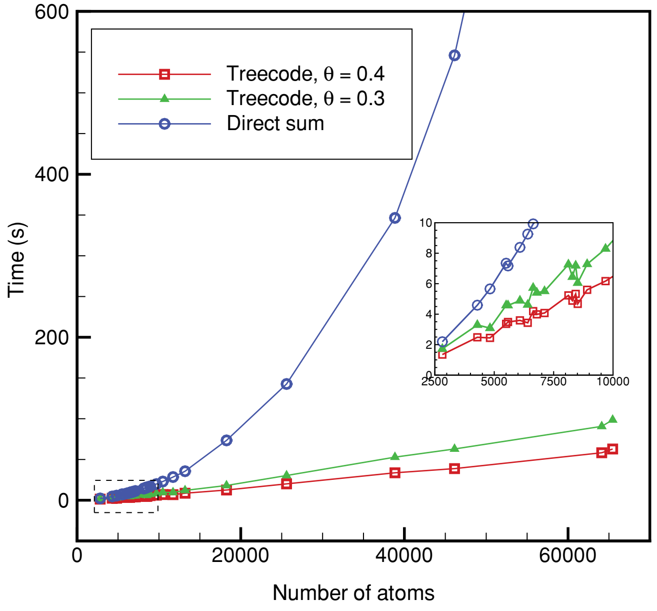

FIG. 4. The CPU times of the 25 proteins for the pair-energies with the P2 treecode algorithm ($\theta = 0.4$ and 0.3) and the direct summation, together with the time of Born radii calculation. The embedded frame is an enlarged picture of the dashed portion.

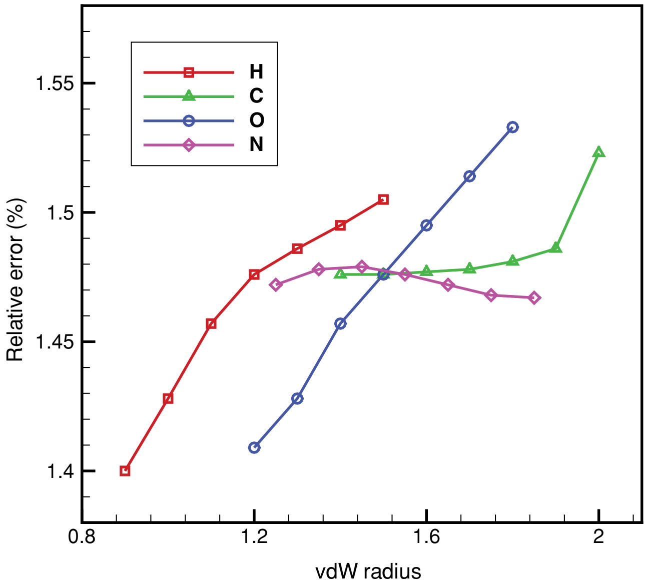

FIG. 5. The average relative errors of the force calculations with the change of atomic radii for 3LHU.

decreases one order of the accuracy in the multipole expansion because we need to take the first order derivatives of Eq. (22), and thus the relative order will be increased in comparison to the energy calculation. To compensate it, the MAC parameter θ should be reduced. We found $\theta = 0.3$ provides a reasonable approximation.

In order to test the performance of the treecode algorithm, the average absolute and relative errors are calculated, where the absolute error is formulated by

$$E = \sqrt{E_x^2 + E_y^2 + E_z^2}, \tag{29}$$

with E_x, E_y, and E_z being the x, y, and z components of the root mean square of errors of the forces, and the reference forces are computed from direct summation of pairwise interactions. The relative error is defined by the ratio of the mean absolute error and the mean force by the direct summation.

The resulting error and time performance are shown in Table V. It is obvious that the force calculation is less accurate than the solvation energy calculation, mainly because of the loss of one order of accuracy for the force in the multipole expansion. Moreover, the averaging operations may have a more significant influence on the accuracy of force than that of energy. The average relative errors are about 3.54% when the MAC parameter takes $\theta = 0.4$. The errors are decreased if θ is reduced to 0.3, with the average relative errors less than 1.15% and the maximum error less than 1.73%. The maximum absolute error for $\theta = 0.3$ is 0.001254 kcal/mol.Å. The speed-up for the force calculation is still attractive, as shown in Fig. 4. For instance, when $\theta = 0.4$, the overall speedups including the effective Born radii calculation for 10k, 20k, and 40k particles are 3.25, 6.27, and 10.96, respectively, while, for $\theta = 0.3$, the corresponding factors are 2.38, 4.28, and 6.94, respectively.

Finally, it is important to assess the error performance of the treecode for different sets of atomic radii because we suspect that the accuracy of the averaging operation used in Eq. (24) would be affected by the relative Born radius sizes among atoms in a cluster. Since the local environments are similar for neighboring atoms, the initial atomic radii become a major source of differences in the calculated effective Born radii. To test it, the vdW radii are varied from the Bondi's parameters by ±0.3 Å for the four elements H, C, O, and N. The force calculations are performed using $\theta = 0.3$, $N_0 = 20$, and the P2 treecode on protein 3LHU. Figure 5 shows the average relative root mean square errors of the force with varying atomic radii. The biggest change of the four curves is O whose errors vary from 1.41% to 1.53%. This demonstrates that the errors of the treecode are not sensitive to the vdW radii. Similar results are obtained for the solvation energy calculations, which are not shown.

IV. CONCLUSIONS

In this study, we have developed a treecode-based GB (tGB) method for the electrostatic solvation energy and force calculations with an $O(N \log N)$ complexity, where N is the number of particles in the solute. The method is composed of a cutoff scheme for the effective Born radii calculation and a hierarchical treecode algorithm for the pairwise charge–charge interaction calculations. Given these two accelerations, the overall computational cost is reduced from $O(N^2)$ to $O(N \log N)$, which will facilitate GB MD simulations of larger biomolecules at longer time scales.

Test examples for a set of 25 proteins demonstrate that the cutoff scheme for the effective Born radii calculation is reasonably accurate, with average unsigned relative errors of the solvation energies less than 0.6% in comparison to the Poisson energies, for a relatively small cutoff radius $L_c = 8$ Å (Table II). The treecode algorithm with both the P1 and P2 multipole expansions maintains a comparable level of accuracy (0.6%) for the electrostatic solvation energy calculation (Table IV). The calculations also demonstrate that the accuracy of the tGB method does not decrease with the increase of the overall charges or the number of atoms in the system, and thus will work for a wide range of biomolecules. The extension to the force calculation shows that the treecode algorithm is less accurate than the corresponding energy calculation, it still provides reasonably accurate force results with the maximum absolute error of 0.00125 kcal/mol/Å with the P2 tree code and $\theta = 0.3$. In MD simulations, the force computed as the gradient of a discontinuous potential would result in the energy drift and instability of the simulation, so high numerical accuracy is often required. In our tGB algorithm, two approximations could lead to the discontinuity in the potentials: one is the cutoff scheme, and the other is the treecode algorithm with finite multipole expansion terms. For the cutoff scheme, as we have shown the $1/r^6$ kernel allows the descreening contribution to be truncated at a much shorter distance due to its faster decay with distance. In addition, a smoothing function can be combined with the cutoff scheme, which will lead to stable simulations with smaller cutoff values. For the treecode algorithm, higher order multipole expansion may be required to obtain stable trajectory, which is under investigation by the authors to reach a balance between the computational accuracy and efficiency.

ACKNOWLEDGMENTS

Z.X. is funded by the Chinese Ministry of Education (NCET-09-0556) and NSFC (Grant No. 11026057). X.C. is funded by the U.S. Department of Energy Field Work Proposal ERKJE84. The authors thank the reviewers for helpful comments and suggestions.

[1] C. Sagui and T. A. Darden, Annu. Rev. Biophys. Biomol. Struct. **28**, 155 (1999).
[2] M. E. Davis and J. A. McCammon, Chem. Rev. **90**, 509 (1990).
[3] A. Warshel and S. T. Russell, Q. Rev. Biophys. **17**, 283 (1984).
[4] T. Simonson, Rep. Prog. Phys. **66**, 737 (2003).
[5] A. Warshel, P. K. Sharma, M. Kato, and W. W. Parson, Biochim. Biophys. Acta **1764**, 1647 (2006).
[6] J. Wang, C. Tan, Y. H. Tan, Q. Lu, and R. Luo, Comm. Comp. Phys. **3**, 1010 (2008).
[7] W. F. van Gunsteren and J. J.C. Berendsen, Angew. Chem., Int. Ed. **29**, 992 (1990).
[8] N. A. Baker, Curr. Opin. Struct. Biol. **15**, 137 (2005).
[9] P. Koehl, Curr. Opin. Struct. Biol. **16**, 142 (2006).
[10] B. Z. Lu, Y. C. Zhou, M. J. Holst, and J. A. McCammon, Comm. Comp. Phys. **3**, 973 (2008).
[11] Z. Xu and W. Cai, SIAM Rev. (in press) available at http://math.uncc.edu/~wcai/fastReviewFinal.pdf (2010).
[12] P. Debye and E. Hückel, Phys. Z. **24**, 185 (1923).
[13] F. Fogolari, A. Brigo, and H. Molinari, J. Mol. Biol. **15**, 377 (2002).
[14] T. A. Darden, D. M. York, and L. G. Pedersen, J. Chem. Phys. **98**, 10089 (1993).
[15] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. Pedersen, J. Chem. Phys. **103**, 8577 (1995).
[16] J. Shimada, H. Kaneko, and T. Takada, J. Comput. Chem. **14**, 867 (1993).
[17] B. A. Luty, M. E. Davis, I. G. Tironi, and W. F. Van Gunsteren, Mol. Simul. **14**, 11 (1994).
[18] Y. Lin, A. Baumketner, S. Deng, Z. Xu, D. Jacobs, and W. Cai, J. Chem. Phys. **131**, 154103 (2009).
[19] W. C. Still, A. Tempczyk, R. C. Hawley, and T. Hendrickson, J. Am. Chem. Soc. **112**, 6127 (1990).
[20] M. S. Lee, F. R. Salsbury, and C. L. Brooks III, J. Chem. Phys. **116**, 10606 (2002).
[21] M. S. Lee, M. Feig, F. R. Salsbury, and C. L. Brooks III, J. Comput. Chem. **24**, 1348 (2003).
[22] A. N. Romanov, S. N. Jabin, Y. B. Martynov, A. V. Sulimov, F. V. Grigoriev, and V. B. Sulimov, J. Phys. Chem. A **108**, 9323 (2004).
[23] T. Grycuk, J. Chem. Phys. **119**, 4817 (2003).
[24] G. Sigalov, P. Scheffel, and A. Onufriev, J. Chem. Phys. **122**, 094511 (2005).
[25] G. Sigalov, A. Fenley, and A. Onufriev, J. Chem. Phys. **124**, 124902 (2006).
[26] J. Mongan, C. Simmerling, J. A. McCammon, D. A. Case, and A. Onufriev, J. Chem. Theory Comput. **3**, 156 (2007).
[27] J. P. Bardhan, J. Chem. Phys. **129**, 144105 (2008).
[28] M. Scarsi, J. Apostolakis, and A. Caflisch, J. Phys. Chem. A **101**, 8098 (1997).
[29] A. Ghosh, C. S. Rapp, and R. A. Friesner, J. Phys. Chem. B **102**, 10983 (1998).
[30] W. Cai, Z. Xu, and A. Baumketner, J. Comput. Phys. **227**, 10162 (2008).
[31] T. Schupbach, V. Zoete, B. Tsakam-Sotche, and O. Michielin, J. Comput. Chem. **31**, 649 (2010).
[32] E. Gallicchio and R. M. Levy, J. Comput. Chem. **25**, 479 (2004).
[33] E. Gallicchio, K. Paris, and R. M. Levy, J. Chem. Theory Comput. **5**, 2544 (2009).
[34] R. Anandakrishnan and A. V. Onufriev, J. Comput. Chem. **31**, 691 (2010).
[35] D. Bashford and D. A. Case, Annu. Rev. Phys. Chem. **51**, 129 (2000).
[36] V. Tsui and D. A. Case, J. Am. Chem. Soc. **122**, 2489 (2000).
[37] M. Feig, A. Onufriev, M. S. Lee, W. Im, D. A. Case, and C. L. Brooks III, J. Comput. Chem. **25**, 265 (2004).
[38] M. Feig and C. L. Brooks III, Curr. Opin. Struct. Biol. **14**, 217 (2004).
[39] D. A. Case, T. E. Cheatham III, T. Darden, H. Gohlke, R. Luo, K. M. Merz, A. Onufriev, C. Simmerling, B. Wang, and R. J. Woods, J. Comput. Chem. **26**, 1668 (2005).
[40] J. Chen, C. L. Brooks III, and J. Khandogin, Curr. Opin. Struct. Biol. **18**, 140 (2008).
[41] A. Okur, L. Wickstrom, and C. Simmerling, J. Chem. Theory Comput. **4**, 488 (2008).
[42] A. Onufriev, Annu. Rep. Comp. Chem. **4**, 125 (2008).
[43] A. Onufriev, *Modeling Solvent Environments* (Wiley, 2010), chapter “Continuum electrostatics solvent modeling with the generalized Born model,” pp. 127–165.
[44] A. Onufriev, D. A. Case, and D. Bashford, J. Comput. Chem. **23**, 1297 (2002).
[45] J. Mongan, W. A. Svrcek-Seiler, and A. Onufriev, J. Chem. Phys. **127**, 185101 (2007).
[46] H. Tjong and H.-X. Zhou, J. Phys. Chem. B **111**, 3055 (2007).
[47] J. Z. Ruscio and A. Onufriev, Biophys. J. **91**, 4121 (2006).
[48] R. E. Amaro, X. Cheng, I. Ivanov, D. Xu, and J. A. McCammon, J. Am. Chem. Soc. **131**, 4702 (2009).
[49] A. Appel, SIAM J. Sci. Comput. (U.S.A.) **6**, 85 (1985).
[50] J. Barnes and P. Hut, Nature (London) **324**, 446 (1986).
[51] Z. H. Duan and R. Krasny, J. Chem. Phys. **113**, 3492 (2000).
[52] L. Greengard and V. Rokhlin, J. Comput. Phys. **73**, 325 (1987).
[53] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems* (MIT, Cambridge, 1987).
[54] H. Cheng, L. Greengard, and V. Rokhlin, J. Comput. Phys. **155**, 468 (1999).
[55] L. Ying, G. Biros, and D. Zorin, J. Comput. Phys. **196**, 591 (2004).
[56] B. Lu, X. Cheng, J. Huang, and J. A. McCammon, Proc. Natl. Acad. Sci. U.S.A. **103**, 19314 (2006).
[57] B. Lu, X. Cheng, J. Huang, and J. A. McCammon, Comput. Phys. Commun. **181**, 1150 (2010).
[58] M. Schaefer and C. Froemmel, J. Mol. Biol. **216**, 1045 (1990).
[59] M. Schaefer and M. Karplus, J. Phys. Chem. **100**, 1578 (1996).
[60] J. G. Kirkwood, J. Chem. Phys. **2**, 351 (1934).
[61] H. Tjong and H. X. Zhou, J. Chem. Phys. **126**, 195102 (2007).
[62] G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, Chem. Phys. Lett. **146**, 122 (1995).
[63] G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, J. Phys. Chem. **100**, 19824 (1996).
[64] M. J. Dudek and J. W. Ponder, J. Comput. Chem. **16**, 791 (1995).
[65] B. N. Dominy and C. L. Brooks III, J. Phys. Chem. B **103**, 3765 (1999).
[66] J. A. Grant and B. T. Pickup, J. Phys. Chem. **99**, 3503 (1995).
[67] M. P. Allen and D. J. Tildesley, *Computer Simulations of Liquids* (Oxford University Press, Oxford, 1987).
[68] K. Lindsay and R. Krasny, J. Comput. Phys. **172**, 879 (2001).
[69] P. Li, H. Johnston, and R. Krasny, J. Comput. Phys. **228**, 3858 (2009).
[70] Z. Xu, Phys. Rev. E **81**, 020902 (2010).
[71] T. J. Dolinsky, J. E. Nielsen, J. A. McCammon, and N. A. Baker, Nucleic Acids Res. **32**, W665 (2004).
[72] A. Bondi, J. Phys. Chem. **68**, 441 (1964).
[73] N. A. Baker, D. Sept, S. Joseph, M. J. Holst, and J. A. McCammon, Proc. Natl. Acad. Sci. U,S.A. **98**, 10037 (2001).
[74] M. Holst and F. Saied, J. Comput. Chem. **14**, 105 (1993).
[75] M. Feig, W. Im, and C. L. Brooks III, J. Chem. Phys. **120**, 903 (2004).
[76] M. Wojciechowski and B. Lesyng, J. Phys. Chem. B **108**, 18368 (2004).
[77] H. Tjong and H. X. Zhou, J. Chem. Phys. **125**, 206101 (2006).
[78] L. J. Yang, C. H. Tan, M. J. Hsieh, J. M. Wang, Y. Duan, P. Cieplak, J. Caldwell, P. A. Kollman, and R. Luo, J. Phys. Chem. B **110**, 13166 (2006).
[79] A. D. MacKerell, Jr., D. Bashford, M. Bellott, R. L. Dunbrack, Jr., J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher III, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus, J. Phys. Chem. B **102**, 3586 (1998).
[80] B. Hess, C. Kutzner, D. van der Spoel, and E. Lindahl, J. Chem. Theory Comput. **4**, 435 (2008).