## Boundary Element Solution of the Linear Poisson–Boltzmann Equation and a Multipole Method for the Rapid Calculation of Forces on Macromolecules in Solution

A. J. BORDNER, G. A. HUBER

Department of Bioengineering, University of California, San Diego, 9500 Gilman Drive, San Diego, California 92093-0412

Received 7 May 2002; Accepted 23 August 2002

**Abstract:** The Poisson–Boltzmann equation is widely used to describe the electrostatic potential of molecules in an ionic solution that is treated as a continuous dielectric medium. The linearized form of this equation, applicable to many biologic macromolecules, may be solved using the boundary element method. A single-layer formulation of the boundary element method, which yields simpler integral equations than the direct formulations previously discussed in the literature, is given. It is shown that the electrostatic force and torque on a molecule may be calculated using its boundary element representation and also the polarization charge for two rigid molecules may be rapidly calculated using a noniterative scheme. An algorithm based on a fast adaptive multipole method is introduced to further increase the speed of the calculation. This method is particularly suited for Brownian dynamics or molecular dynamics simulations of large molecules, in which the electrostatic forces must be calculated for many different relative positions and orientations of the molecules. It has been implemented as a set of programs in C++, which are used to study the accuracy and speed of this method for two actin monomers.

© 2003 Wiley Periodicals, Inc. J Comput Chem 24: 353-367, 2003

Key words: Poisson–Boltzmann equation; boundary element method; cell multipole method; Brownian dynamics; molecular dynamics

#### Introduction

Electrostatic interactions are important in determining the structure, thermodynamic properties, and reaction kinetics of macromolecules in solution.<sup>1</sup> In particular, for noncovalent protein association reactions Monte Carlo calculations of the thermodynamic quantities and Brownian dynamics calculations of reaction rates require evaluating the electrostatic interaction energy and force, respectively, many times during the calculation. The water may be represented in atomic detail, as in molecular dynamics simulations; however, this is computationally intensive. Alternatively, treating the water as a continuum medium with macroscopic properties, such as a bulk dielectric constant, allows a faster computation of the electrostatic and hydrodynamic effects.<sup>1–3</sup>

Methods for evaluating the electrostatic potential in continuum solvation models include the generalized Born method,<sup>4,5</sup> the effective charge method of ref. 6, and various numerical methods for direct solution of the Poisson–Boltzmann equation. In the latter case the charge density of the molecules is usually approximately represented by point charges located at the atomic centers. The

interior of the molecule, delimited by the solvent-excluded surface, has a low dielectric constant (~2–4), whereas the exterior has a high dielectric constant (~80). We will be primarily interested in simulations of proteins under physiologic conditions in which ions are present and are included in the electrostatic calculation by assuming that the electric potential satisfies the Poisson–Boltzmann equation. When the electrostatic energy of the ions is much less than their thermal energy, i.e.  $q\phi/kT \ll 1$ , the nonlinear Poisson–Boltzmann equation may be linearized. This approximation is reasonably accurate for molecules with a relatively low charge density in solutions at physiologic ion concentrations (~0.1 M).<sup>7</sup>

The boundary element method (BEM) for solving the linear Poisson–Boltzmann equation has several advantages over two

*Correspondence to:* A. J. Bordner, The Scripps Research Institute, 10550 North Torrey Pines Rd., TPC-28, San Diego, CA 92037; e-mail: bordner@scripps.edu

Contract/grant sponsor: National Institutes of Health (to A.J.B.)

other numerical methods: the finite difference and finite element methods. One is that infinite domains may be simply treated without introducing a large artificial boundary. Another advantage is that the equations are defined over the two-dimensional molecular surface rather than a three-dimensional domain so that a smaller number of finite elements, resulting from discretizing these equations on a mesh, are required. Also, there is no need to distribute the partial atomic charges over nearby node points, as in the other two methods. Finally, because the electrostatic potential is represented by charge distributions on the molecular surfaces it may be rapidly calculated for a large number of different relative configurations by calculating the perturbation of the charge density for an isolated molecule due to the electric field of the other molecule. This is because, unlike the other two numerical methods, the discretization mesh moves with the molecules and is not fixed in space. This property will be the basis for our method.

The BEM was first applied to the case of zero ionic strength, in which the potential is a solution of the Poisson equation.<sup>8–11</sup> Studies that include ions in solution may be found in refs. 12–16. Several recent articles have used multipole or adaptive grid methods to reduce the size of the matrix resulting from the BEM for the Poisson equation and thus reducing storage requirements and solution time. References 17–19 used methods based on the cell multipole algorithm of refs. 20 and 21, whereas ref. 22 used a multilevel grid on the surface to reduce the size of the system of linear equations.

In this article we will first describe a BEM for solving the linear Poisson–Boltzmann equation for two proteins in an ionic solution based on a single-layer formulation of the equations. This gives a simpler set of equations on the boundary and hence a more efficient starting point for solving them than the direct formulation of the boundary integral equations based on Green's theorem used in previous studies. More importantly, this will allow us to use a method based on the cell multipole algorithm to rapidly evaluate the force and torque on one molecule due to the electric potential of the other. It should be noted that the single-layer formulation of the Poisson equation yields matrix equations of half the size of those presented in this article and, hence, are preferable in the case of zero ionic strength.<sup>11,23</sup>

Our cell multipole method is applicable to rigid molecules because the calculation of the fields of the isolated molecules needs to be done only once before the simulation begins. The relatively slow calculations, namely that of the surface charge distribution on an isolated molecule's surface using the BEM and this distribution's multipole moments, up to quadrupole order, within a multilevel set of cubic cells are done at leisure before the simulation begins. The calculation of the total force and torque on a molecule are then done rapidly by using an adaptive cell multipole algorithm. Basically, the potential at one molecule due to the charges of the other are calculated using the cell multipole algorithm and a local Taylor expansion about the center of a cell in the first molecule. The contributions to the force and torque due to this quadratic approximation to the electric field within the cells on the first molecule are then summed. Actually, linear combinations of the multipole and Taylor coefficients, that are coefficients in an expansion in orthogonal polynomials within each cell, are used.

The next section describes the boundary element solution of the Poisson–Boltzmann equation and some associated numerical

methods. Then we discuss the adaptive multipole algorithm for fast calculation of the force and torque on a rigid molecule. The error and computational speed are then tested on a system of two actin monomers, and finally, the results and possible areas of further investigation are discussed.

# Boundary Element Solution of the Poisson–Boltzmann Equation

#### **Boundary Integral Equations**

The first step in the BEM is to formulate the solution of a boundary value problem for a partial differential equation in a region in terms of integral equations on its boundary, called the boundary integral equations (BIE). In the literature, previous applications of the boundary element method to solve the linear Poisson–Boltzmann equation for the electric potential of macromolecules in an ionic solution used the direct formulation of the BIE, in which the BIE are derived using Green's theorem. However the single-layer formulation of a boundary integral equation, when it exists, always yields simpler equations, and hence, a more efficient computational scheme.

We consider the case of two molecules in an ionic solution which occupy regions  $\Omega_1$  and  $\Omega_2$ , delimited by the solvent-excluded, or Connolly, surfaces of the respective molecules. The solvent-excluded surface is formed by the boundary of the region in which a probe sphere, typically the size of a water oxygen Van der Waals radius of around 1.4 Å, is excluded from the Van der Waals volume of the molecule.<sup>24</sup> Outside of these regions, i.e., in the solution, the potential  $\phi$  satisfies the linear Poisson–Boltzmann equation without sources

$$-\vec{\nabla}^2 \phi(\vec{x}) + \kappa^2 \phi(\vec{x}) = 0 \tag{1.1}$$

where the inverse Debye length  $\kappa$  is

$$\kappa^2 = \frac{8\,\pi\bar{n}e^2}{\varepsilon_{\rm out}kT}\tag{1.2}$$

for a neutral solution of monovalent ions of concentration  $\bar{n}$ . e is the fundamental electronic charge,  $\varepsilon_{out}$  is the dielectric constant in the solution, k is the Boltzmann constant and T is the absolute temperature.

Inside each region the potential satisfies the Poisson equation

$$-\vec{\nabla}^{2}\phi(\vec{x}) = \frac{4\pi}{\varepsilon_{\text{in}}}\rho(\vec{x}) = \frac{4\pi}{\varepsilon_{\text{in}}}\sum_{k=1}^{N_{c}^{(j)}} q_{k}^{(j)}\delta(\vec{x}-\vec{x}_{k}^{(j)}), \quad \vec{x} \in \Omega_{j}, j = 1, 2$$
(1.3)

where the partial atomic charges  $q_k^{(j)}$  are located at the atomic centers  $\vec{x}_k^{(j)}$  within  $\Omega_j$  and  $\varepsilon_{\rm in}$  is the dielectric constant inside the molecules. The fundamental solution or Green's function  $E_P(\vec{x})$  for the Poisson equation is

$$E_P(\vec{x}) = \frac{1}{4\pi |\vec{x}|}$$
(1.4)

and satisfies

$$-\vec{\nabla}^2 E_P(\vec{x}) = \delta(\vec{x}). \tag{1.5}$$

Next define

$$u_{j} = \frac{4\pi}{\varepsilon_{\rm in}} \int_{\Omega_{j}} d\vec{x}' E_{P}(\vec{x} - \vec{x}') \rho(\vec{x}'),$$
$$= \frac{1}{\varepsilon_{\rm in}} \sum_{k=1}^{N_{c}^{(j)}} \frac{q_{k}^{(j)}}{|\vec{x} - \vec{x}_{k}|}$$
(1.6)

and

$$\tilde{\phi}_j(\vec{x}) \equiv \phi(\vec{x}) - u_j(\vec{x}). \tag{1.7}$$

 $\tilde{\phi}_j(\vec{x})$  then satisfies the Laplace equation

$$-\vec{\nabla}^2 \vec{\phi}_i(\vec{x}) = 0 \tag{1.8}$$

inside  $\Omega_j$ , because

$$-\vec{\nabla}^2 u_j(\vec{x}) = \frac{4\pi}{\varepsilon_{\rm in}} \rho(\vec{x}). \tag{1.9}$$

 $\tilde{\phi}_j(\vec{x})$  inside  $\Omega_j$  may then be represented by a single-layer representation in terms of a surface charge density  $f_j(\vec{x})$  as<sup>23</sup>

$$\tilde{\phi}_j(\vec{x}) = \int_{\partial \Omega_j} d\vec{x}' E_P(\vec{x} - \vec{x}') f_j(\vec{x}')$$
(1.10)

so that the potential inside  $\Omega_i$  is

$$\phi(\vec{x}) = \frac{1}{\varepsilon_{\rm in}} \sum_{k=1}^{N_c^{(j)}} \frac{q_k^{(j)}}{|\vec{x} - \vec{x}_k^{(j)}|} + \frac{1}{4\pi} \int_{\partial \Omega_j} d\vec{x}' \, \frac{f_j(\vec{x}')}{|\vec{x} - \vec{x}'|}.$$
 (1.11)

Because there are no charges in the exterior region the potential there satisfies the Poisson–Boltzmann equation without sources [eq. (1.1)]. The fundamental solution  $E_{PB}(\kappa, \vec{x})$  for this equation is

$$E_{PB}(\kappa, \vec{x}) \equiv \frac{e^{-\kappa |\vec{x}|}}{|\vec{x}|}$$
(1.12)

which satisfies

$$(-\vec{\nabla}^2 + \kappa^2) E_{PB}(\kappa, \vec{x}) = \delta(\vec{x}). \tag{1.13}$$

The potential in the exterior region  $\vec{x} \in \Omega_1^c \cap \Omega_2^c$  may be described by a single-layer representation in terms of charge densities  $h_i(\vec{x})$  as

$$\begin{split} \phi(\vec{x}) &= \sum_{j=1,2} \int_{\partial \Omega_j} d\vec{x}' E_{PB}(\kappa, \, \vec{x} - \vec{x}') h_j(\vec{x}') \\ &= \frac{1}{4\pi} \sum_{j=1,2} \int_{\partial \Omega_j} d\vec{x}' \; \frac{e^{-\kappa |\vec{x} - \vec{x}'|}}{|\vec{x} - \vec{x}'|} h_j(\vec{x}') \end{split}$$
(1.14)

The solutions of the Poisson–Boltzmann equation for the inside regions [eq. (1.11)], and the outside region [eq. (1.14)], must satisfy the following boundary conditions on  $\partial \Omega_i$  for  $\vec{y} \in \partial \Omega_i$ 

$$\lim_{\substack{\vec{x} \to \vec{y} \\ \vec{x} \in \Omega_j}} \phi(\vec{x}) = \lim_{\substack{\vec{x} \to \vec{y} \\ \vec{x} \in \Omega^c}} \phi(\vec{x}), \tag{1.15}$$

$$\lim_{\substack{\vec{x} \to \vec{y} \\ \vec{x} \in \Omega_j}} \varepsilon_{\text{in}} \, \hat{n} \cdot \vec{\nabla} \phi(\vec{x}) = \lim_{\substack{\vec{x} \to \vec{y} \\ \vec{x} \in \Omega_j^c}} \varepsilon_{\text{out}} \, \hat{n} \cdot \vec{\nabla} \phi(\vec{x}) \tag{1.16}$$

with  $\hat{n}$  the outward pointing unit length normal vector to  $\partial \Omega_j$ . The integrals in the expressions for the potential in each region also satisfy the following jump conditions at the boundaries  $\partial \Omega_j$  for  $\tilde{y} \in \partial \Omega_j^{23}$ 

$$\lim_{\substack{\vec{x} \to \vec{y} \\ \vec{x} \in \Omega_j}} \int_{\partial \Omega_j} d\vec{x}' E_p(\vec{x} - \vec{x}') f(\vec{x}') = \int_{\partial \Omega_j} d\vec{x}' E_p(\vec{y} - \vec{x}') f(\vec{x}'), \quad (1.17)$$

$$\lim_{\substack{\vec{x}\to\vec{y}\\\vec{x}\in\Omega_j^c}} \int_{\partial\Omega_j} d\vec{x}' E_{PB}(\vec{x}-\vec{x}') f(\vec{x}') = \int_{\partial\Omega_j} d\vec{x}' E_{PB}(\vec{y}-\vec{x}') f(\vec{x}'),$$
(1.18)

$$\lim_{\substack{\vec{x} \to \vec{y} \\ \vec{x} \in \Omega_j}} \int_{\partial \Omega_j} d\vec{x}' \hat{n} \cdot \vec{\nabla}_x E_P(\vec{x} - \vec{x}') f(\vec{x}')$$
$$= \int_{\partial \Omega_j} d\vec{x}' \hat{n} \cdot \vec{\nabla}_y E_P(\vec{y} - \vec{x}') f(\vec{x}') + \frac{1}{2} f(\vec{y}), \quad (1.19)$$

$$\lim_{\substack{\vec{x} \to \vec{y} \\ \vec{x} \in \Omega_j^c}} \int_{\partial \Omega_j} d\vec{x}' \hat{n} \cdot \vec{\nabla}_x E_{PB}(\vec{x} - \vec{x}') f(\vec{x}')$$
$$= \int_{\partial \Omega_j} d\vec{x}' \hat{n} \cdot \vec{\nabla}_y E_{PB}(\vec{y} - \vec{x}') f(\vec{x}') - \frac{1}{2} f(\vec{y}). \quad (1.20)$$

Using these jump conditions eq. (1.15) becomes

c



**Figure 1.** Mapping  $\vec{x}(\xi_1, \xi_2)$  from local coordinates to a triangular quadratic surface element.

$$\frac{1}{\varepsilon_{\text{in}}} \sum_{k=1}^{N_{c}^{(j)}} \frac{q_{k}^{(j)}}{|\vec{x} - \vec{x}_{k}|} + \frac{1}{4\pi} \int_{\partial \Omega_{j}} d\vec{x}' \frac{f_{j}(\vec{x}')}{|\vec{x} - \vec{x}'|} \\
= \frac{1}{4\pi} \sum_{k=1,2} \int_{\partial \Omega_{k}} d\vec{x}' \frac{e^{-\kappa |\vec{x} - \vec{x}'|}}{|\vec{x} - \vec{x}'|} h_{k}(\vec{x}'), \quad \vec{x} \in \partial \Omega_{j}$$
(1.21)

and eq. (1.16) becomes

$$\frac{1}{\varepsilon_{\text{in}}} \sum_{k=1}^{N_{c}^{(j)}} q_{k}^{(j)} \hat{n}(\vec{x}) \cdot \vec{\nabla}_{x} \frac{1}{|\vec{x} - \vec{x}_{k}|} \\
+ \frac{1}{4\pi} \int_{\partial \Omega_{j}} d\vec{x}' f_{j}(\vec{x}') \hat{n}(\vec{x}) \cdot \vec{\nabla}_{x} \frac{1}{|\vec{x} - \vec{x}'|} + \frac{1}{2} f_{j}(\vec{x}) \\
= \frac{\varepsilon_{\text{out}}}{\varepsilon_{\text{in}}} \left[ \frac{1}{4\pi} \sum_{k=1,2} \int_{\partial \Omega_{k}} d\vec{x}' h_{k}(\vec{x}') \hat{n}(\vec{x}) \cdot \vec{\nabla}_{x} \frac{e^{-\kappa |\vec{x} - \vec{x}'|}}{|\vec{x} - \vec{x}'|} \\
- \frac{1}{2} h_{j}(\vec{x}) \right], \quad \vec{x} \in \partial \Omega_{j}.$$
(1.22)

#### Surface Discretization and Boundary Element Equations

The BIE of eqs. (1.21) and (1.22) are then solved for  $f_{1,2}(\vec{x})$  and  $h_{1,2}(\vec{x})$  by first discretizing both the boundary surfaces and the function space in terms of finite elements. We choose to use quadratic surface elements and linear function elements. The solvent-excluded surfaces of the molecules  $\partial\Omega_{1,2}$  are approximated by a closed mesh of triangles with quadratic curves as edges. Local coordinates  $(\xi_1, \xi_2)$  for each triangle  $T_m^{(j)}$  on the mesh for surface  $\partial\Omega_j$ , such that they map the standard triangle, with vertices at (0, 0), (1, 0), and (0, 1) onto the triangular surface element defined by the six vertices  $\{\vec{v}_1, \ldots, \vec{v}_6\}$  are (see Fig. 1)

$$\vec{x}_{T_m}(\xi_1,\,\xi_2) = \sum_{j=1}^{\circ} N_j(\xi_1,\,\xi_2) \vec{v}_j \tag{1.23}$$

with the shape functions  $N_i(\xi_1, \xi_2)$  defined as

$$N_{1}(\xi_{1}, \xi_{2}) = (2\xi_{1} + 2\xi_{2} - 1)(\xi_{1} + \xi_{2} - 1),$$

$$N_{2}(\xi_{1}, \xi_{2}) = \xi_{1}(2\xi_{1} - 1),$$

$$N_{3}(\xi_{1}, \xi_{2}) = \xi_{2}(2\xi_{2} - 1),$$

$$N_{4}(\xi_{1}, \xi_{2}) = 4\xi_{1}(1 - \xi_{1} - \xi_{2}),$$

$$N_{5}(\xi_{1}, \xi_{2}) = 4\xi_{2}(1 - \xi_{1} - \xi_{2}),$$

$$N_{6}(\xi_{1}, \xi_{2}) = 4\xi_{1}\xi_{2}.$$
(1.24)

The function  $N_j(\xi_1, \xi_2)$  is 1 at vertex  $\vec{v}_j$  and 0 at the other vertices. A function defined on the surface  $h_j(\vec{x}), \vec{x} \in \partial \Omega_j$  is approximated by linear function elements  $\{H_k^{(j)}(\vec{x}), k = 1, \dots, N_V\}$  as

$$\tilde{h}(\vec{x}) = \sum_{k=1}^{N_V^{(j)}} (\vec{h}_j)_k H_k^{(j)}(\vec{x})$$
(1.25)

where  $N_V^{(j)}$  is the number of vertices in the triangulation of  $\partial \Omega_j$ . A component of the vector  $\vec{h}_j$ ,  $(\vec{h}_j)_k$ , is the value of  $h_j(\vec{x})$  at vertex k.  $H_k^{(j)}(\vec{x})$  is linear in the local coordinates  $\xi_1$  and  $\xi_2$ , 1 at vertex k and 0 at the other vertices so that only the vertices of triangles containing  $\vec{x}$  contribute to the sum in eq. (1.25). On surface element  $T_m H_k(\tilde{x}_{T_m}(\xi_1, \xi_2))$  is one of the functions  $S_j(\xi_1, \xi_2)$ , j = 1, 2, 3 defined by

$$S_{1}(\xi_{1}, \xi_{2}) = 1 - \xi_{1} - \xi_{2},$$

$$S_{2}(\xi_{1}, \xi_{2}) = \xi_{1},$$

$$S_{3}(\xi_{1}, \xi_{2}) = \xi_{2}$$
(1.26)

depending on whether vertex k is at (0, 0), (1, 0), or (0, 1), respectively, in the local coordinates on  $T_m^{(j)}$ . We denote the appropriate function for  $H_k^{(j)}(\vec{x})$  on  $T_m^{(j)}$  as  $S_{k,m}^{(j)}(\xi_1, \xi_2)$ .

The four BIE of eqs. (1.21) and (1.22) are then solved using the method of collocation in which the discrete approximation to the solution is assumed to be exact at a finite number of node points, which in this case are the corner vertices of the triangles. This then yields a system of linear equations in terms of the coefficients in the expansion of the surface charge densities  $f_{1,2}(\vec{x})$  and  $h_{1,2}(\vec{x})$  in terms of the finite elements. Define the following integrals

$$\begin{split} I_{1l}^{(j)}(\vec{x}, \kappa) &\equiv \frac{1}{2\pi} \sum_{m \in \mathscr{F}_{l}^{(j)}} \int_{T_{m}^{(j)}} d\vec{x}' \frac{e^{-\kappa |\vec{x} - \vec{x}'|}}{|\vec{x} - \vec{x}'|} H_{l}^{(j)}(\vec{x}'), \\ &= \frac{1}{2\pi} \sum_{m \in \mathscr{F}_{l}^{(j)}} \int_{0}^{1} d\xi_{1} \int_{0}^{1-\xi_{1}} d\xi_{2} \left| \frac{\partial \vec{x}_{T_{m}^{(j)}}}{\partial \vec{\xi}} \right| \\ &\times \frac{e^{-\kappa |\vec{x} - \vec{x}_{T_{m}^{(j)}}(\xi_{1}, \xi_{2})|}}{|\vec{x} - \vec{x}_{T_{m}^{(j)}}(\xi_{1}, \xi_{2})|} S_{l,m}^{(j)}(\xi_{1}, \xi_{2}), \\ I_{2l}^{(j)}(\vec{x}, \kappa) &\equiv -\hat{n}(\vec{x}) \cdot \vec{\nabla}_{x} I_{1l}^{(j)}(\vec{x}, \kappa) \end{split}$$
(1.27)

where  $\mathcal{F}_{l}^{(j)}$  is the set of indices for surface elements that contain vertex *l* on the mesh for  $\partial \Omega_{i}$ . The Jacobian factor in the integral

$$\left|\frac{\partial \vec{x}_{T_m^{(j)}}}{\partial \vec{\xi}}\right| = \vec{\nabla}_{\xi_1} \vec{x}_{T_m^{(j)}}(\xi_1, \xi_2) \times \vec{\nabla}_{\xi_2} \vec{x}_{T_m^{(j)}}(\xi_1, \xi_2).$$
(1.28)

may be calculated from eq. (1.23). The integrals in eqs. (1.27) are calculated using two-dimensional Gaussian quadrature as described in Appendix A.

Next we define the following matrix elements in terms of the integrals in eq. (1.27)

$$(\mathbf{A}_{j})_{mn} \equiv I_{1n}^{(j)}(\vec{v}_{m}^{(j)}, 0), \quad j = 1, 2$$

$$(\mathbf{B}_{jk})_{mn} \equiv I_{1n}^{(k)}(\vec{v}_{m}^{(j)}, \kappa), \quad j, k = 1, 2$$

$$(\mathbf{C}_{j})_{mn} \equiv I_{2n}^{(j)}(\vec{v}_{m}^{(j)}, 0) - \delta_{mn}, \quad j = 1, 2$$

$$(\mathbf{D}_{jj})_{mn} \equiv \frac{\varepsilon_{\text{out}}}{\varepsilon_{\text{in}}} (I_{2n}^{(j)}(\vec{v}_{m}^{(j)}, \kappa) + \delta_{mn}), \quad j = 1, 2$$

$$(\mathbf{D}_{jk})_{mn} \equiv \frac{\boldsymbol{\varepsilon}_{\text{out}}}{\boldsymbol{\varepsilon}_{\text{in}}} I_{2n}^{(k)}(\vec{v}_m^{(j)}, \boldsymbol{\kappa}), \quad j \neq k = 1, 2$$
(1.29)

and the vector elements

$$s_{k}^{(j)} \equiv \frac{2}{\varepsilon_{\text{in}}} \sum_{m=1}^{N_{c}^{(j)}} \frac{q_{m}^{(j)}}{|\vec{v}_{k}^{(j)} - \vec{x}_{m}^{(j)}|}$$
$$t_{k}^{(j)} \equiv \frac{2}{\varepsilon_{\text{in}}} \sum_{m=1}^{N_{c}^{(j)}} \frac{q_{m}^{(j)} \hat{n}(\vec{v}_{k}^{(j)}) \cdot (\vec{v}_{k}^{(j)} - \vec{x}_{m}^{(j)})}{|\vec{v}_{k}^{(j)} - \vec{x}_{m}^{(j)}|^{3}}$$
(1.30)

where  $\vec{v}_m^{(j)}$  is the coordinate of vertex *m* on surface  $\Omega_j$  and  $\vec{x}_m^{(j)}$  is the coordinate of atomic charge  $q_m^{(j)}$ . Furthermore, we define vectors of coefficients  $\vec{f}_1$  and  $\vec{f}_2$  in the expansion of the surface charge densities  $f_1(\vec{x})$  and  $f_2(\vec{x})$ , respectively, in terms of the finite function elements according to eq. (1.25). Likewise, vectors  $\vec{h}_1$  and  $\vec{h}_2$  are defined for the surface charge densities  $h_1(\vec{x})$  and  $h_2(\vec{x})$ , respectively. Equations (1.21) and (1.22) then become

$$\vec{v}_{1} + \mathbf{A}_{1} \cdot \vec{f}_{1} = \mathbf{B}_{11} \cdot \vec{h}_{1} + \mathbf{B}_{12} \cdot \vec{h}_{2},$$
  

$$\vec{v}_{2} + \mathbf{A}_{2} \cdot \vec{f}_{2} = \mathbf{B}_{21} \cdot \vec{h}_{1} + \mathbf{B}_{22} \cdot \vec{h}_{2},$$
  

$$\vec{w}_{1} + \mathbf{C}_{1} \cdot \vec{f}_{1} = \mathbf{D}_{11} \cdot \vec{h}_{1} + \mathbf{D}_{12} \cdot \vec{h}_{2},$$
  

$$\vec{w}_{2} + \mathbf{C}_{2} \cdot \vec{f}_{2} = \mathbf{D}_{21} \cdot \vec{h}_{1} + \mathbf{D}_{22} \cdot \vec{h}_{2}.$$
 (1.31)

These linear equations may then be solved for the charge density vectors  $\vec{f}_{1,2}$  and  $\vec{h}_{1,2}$  using a suitable iterative method with preconditioner.

#### Iterative Solution of Boundary Element Equations

Another approach to solving eq. (1.31) is a perturbative method in which the surface charge densities for two molecules in isolation are first calculated and then the relatively small change in these densities due to the charges of the other molecule are determined. This is a fast method to calculate the charge densities for a large number of different relative configurations of the two rigid molecules. Although we will use a faster, but approximate, noniterative method that is described later to solve for the polarization charge in the fast multipole version of the BEM, we first review an iterative method of this type, developed by Zhou in ref. 14, that may be adapted to solve eq. (1.31). The results of this method will be used for comparison with our one-step algorithm. In the iterative scheme the surface charge densities for an isolated molecule in solution are first given by solving the equations

$$\vec{f}_{j}^{0} = \mathbf{A}_{j}^{-1} (\mathbf{B}_{jj} \cdot \vec{h}_{j} - \vec{s}_{j}),$$
$$\vec{h}_{j}^{0} = \mathbf{U}_{j}^{-1} \cdot [\vec{t}_{j} - \mathbf{C}_{j} \cdot \mathbf{A}_{j}^{-1} \cdot \vec{s}_{j}]$$
(1.32)

where

$$\mathbf{U}_{j} \equiv \mathbf{D}_{jj} - \mathbf{C}_{j} \cdot \mathbf{A}_{j}^{-1} \cdot \mathbf{B}_{jj}.$$
(1.33)

Then substituting  $\vec{f}_j = \vec{f}_j^0 + \Delta \vec{f}_j$  and  $\vec{h}_j = \vec{h}_j^0 + \Delta \vec{h}_j$  in eq. (1.31) gives the following equations for the changes in the surface charge densities due to the electric field of the other molecule,  $\Delta \vec{h}_j$  and  $\Delta \vec{f}_j$ , as

$$\Delta \vec{h}_1 = \mathbf{U}_1^{-1} \cdot [-\mathbf{D}_{12} + \mathbf{C}_1 \cdot \mathbf{A}_1^{-1} \cdot \mathbf{B}_{12}] \cdot (\vec{h}_2 + \Delta \vec{h}_2),$$
  
$$\Delta \vec{f}_1 = \mathbf{A}_1^{-1} \cdot (\mathbf{B}_{11} \cdot \Delta \vec{h}_1 + \mathbf{B}_{12} \cdot (\vec{h}_2 + \Delta \vec{h}_2))$$
(1.34)

and likewise for  $\Delta \vec{f}_2$  and  $\Delta \vec{h}_2$ . These equations may be solved to the desired accuracy by starting with  $\Delta \vec{f}_j = \Delta \vec{h}_j = \vec{0}$  and iteratively substituting the previous values for  $\Delta \vec{f}_j$  and  $\Delta \vec{h}_j$  into eq. (1.34) to obtain the value for the next iteration.

#### **One-Step Solution of Boundary Element Equations**

Next, we will derive a one-step calculation to approximate  $\Delta \vec{h}_j$  and  $\Delta \vec{f}_j$  as a faster alternative to the iterative solution of eq. (1.34). The method is to approximate the change in the charge density of molecule 1, after introducing molecule 2, by solving the Poisson–Boltzmann boundary value problem for molecule 1 without any internal point charges, i.e. simply a dielectric cavity, and in the presence of an externally applied potential due only to molecule 2. Of course, the resulting change in charge density on the other one and these corrections must be evaluated iteratively to obtain an exact result. However, for values of the dielectric constants relevant to protein molecules in water, each succeeding correction to the charge density is much smaller than the previous one, and hence, the first term is a reasonably accurate approximation.

The boundary value problem for the dielectric cavity of molecule 1 in the presence of an external field is again solved using the boundary element method. There is a surface charge density on the surface  $\partial \Omega_1$  due to an externally applied electric field with potential  $\phi_{ext}(\vec{x})$ , no free charges inside  $\partial \Omega_1$ , and a constant dielectric coefficient  $\varepsilon_{in}$  inside of  $\partial \Omega_1$  and  $\varepsilon_{out}$  outside of  $\partial \Omega_1$ , as in the previous derivation. Also, as before, ions are present outside of  $\partial \Omega_1$  but excluded from the inside. The boundary integral equations for each region are then

$$\phi(\vec{x}) = \phi_{\text{ext}}(\vec{x}) + \int_{\partial \Omega_1} d\vec{x}' E_P(\vec{x} - \vec{x}') \Delta f_1(\vec{x}'), \quad \vec{x} \in \Omega_1,$$
  
$$\phi(\vec{x}) = \phi_{\text{ext}}(\vec{x}) + \int_{\partial \Omega_1} d\vec{x}' E_{PB}(\vec{x} - \vec{x}') \Delta h_1(\vec{x}'), \quad \vec{x} \in \Omega_1^c \quad (1.35)$$

with  $E_P(\vec{x} - \vec{x}')$  and  $E_{PB}(\vec{x} - \vec{x}')$  defined in eq. (1.4) and eq. (1.12), respectively. Matching the solutions in the two regions at the boundary using eq. (1.15) and (1.16) and using the jump conditions of eq. (1.17)–(1.20) gives the boundary integral equations

$$\int_{\partial\Omega_1} d\vec{x}' \Delta f_1(\vec{x}') \frac{1}{|\vec{x} - \vec{x}'|} = \int_{\partial\Omega_1} d\vec{x}' \Delta h_1(\vec{x}') \frac{e^{\kappa |\vec{x} - \vec{x}'|}}{|\vec{x} - \vec{x}'|}, \quad \vec{x} \in \partial\Omega$$
(1.36)

$$\frac{1}{4\pi} \int_{\partial\Omega_{1}} d\vec{x}' \Delta f_{1}(\vec{x}') \hat{n}(\vec{x}) \cdot \vec{\nabla}_{x} \frac{1}{|\vec{x} - \vec{x}'|} + \frac{1}{2} \Delta f_{1}(\vec{x})$$

$$= \frac{\varepsilon_{\text{out}}}{\varepsilon_{\text{in}}} \left[ \frac{1}{4\pi} \int_{\partial\Omega_{1}} d\vec{x}' \Delta h_{1}(\vec{x}) \hat{n}(\vec{x}) \cdot \vec{\nabla}_{x} \frac{e^{\kappa |\vec{x} - \vec{x}'|}}{|\vec{x} - \vec{x}'|} - \frac{1}{2} \Delta h_{1}(\vec{x}) \right]$$

$$+ \left( \frac{\varepsilon_{\text{out}}}{\varepsilon_{\text{in}}} - 1 \right) \hat{n}(\vec{x}) \cdot \vec{\nabla} \phi_{\text{ext}}(\vec{x}), \quad \vec{x} \in \partial\Omega_{1}. \quad (1.37)$$

After discretizing the BIE, as before, and defining the components of vector  $\vec{u}$  as

$$u_{k} \equiv 2 \left( \frac{\varepsilon_{\text{out}}}{\varepsilon_{\text{in}}} - 1 \right) \hat{n}(\vec{x}_{k}) \cdot \vec{\nabla} \phi_{\text{ext}}(\vec{x}_{k})$$
(1.38)

where the  $\vec{x}_k$  are the nodal coordinates for the mesh of  $\partial \Omega_1$  and using the matrices  $\mathbf{A}_1$ ,  $\mathbf{B}_{11}$ ,  $\mathbf{C}_1$ , and  $\mathbf{D}_{11}$ , defined in eq. (1.29), gives the linear equations

$$\mathbf{A}_1 \cdot \Delta \vec{f}_1 = \mathbf{B}_{11} \cdot \Delta \vec{h}_1, \qquad (1.39)$$

$$\mathbf{C}_1 \cdot \Delta \vec{f}_1 = \mathbf{D}_{11} \cdot \Delta \vec{h}_1 - \vec{u}. \tag{1.40}$$

Only  $\Delta \vec{h}_1$  is needed, with its value given by solving the preceding equations to yield

$$\Delta \vec{h}_1 = \mathbf{U}_1^{-1} \vec{u}, \tag{1.41}$$

where  $U_1$  is defined in eq. (1.33). Substituting the potential due to an isolated molecule 2 for the external potential gives the charge on molecule 1 due to presence of molecule 2 as

$$(\Delta \vec{h}_{1})_{j} = \frac{1}{2\pi} \left( \frac{\varepsilon_{\text{out}}}{\varepsilon_{\text{in}}} - 1 \right) \sum_{k} (\mathbf{U}_{1}^{-1})_{jk}$$
$$\times \left[ \hat{n}(\vec{x}) \cdot \vec{\nabla}_{x} \int_{\partial \Omega_{2}} d\vec{x}' \, \frac{e^{-\kappa |\vec{x} - \vec{x}'|}}{|\vec{x} - \vec{x}'|} \, h_{2}(\vec{x}') \right]_{\vec{x} = \vec{x}_{k}} \quad (1.42)$$

which, upon substituting the discretized form of the integral, gives

$$\Delta \vec{h}_1 = \left(\frac{\varepsilon_{\rm in}}{\varepsilon_{\rm out}} - 1\right) \mathbf{U}_1^{-1} \mathbf{D}_{12} \vec{h}_2. \tag{1.43}$$

The corresponding equation for  $\Delta h_2$  may be obtained by exchanging indices  $1 \leftrightarrow 2$ .

#### Force and Torque Calculation

We next describe a simple method to calculate the force and torque on one molecule due to the electric field of the other using the surface charge densities of the boundary element representation. Consider the force,  $\vec{F}_{12}$ , on molecule 1 due to the field of molecule 2. This may be calculated by integrating the normal component of the electromagnetic stress tensor

$$T_{jk}(\vec{x}) \equiv \frac{\varepsilon_{\text{out}}}{4\pi} \left( E_j(\vec{x}) E_k(\vec{x}) - \frac{1}{8} |\vec{E}(\vec{x})|^2 \delta_{jk} \right) - kT \left( \sum_i n_i(\vec{x}) \right) \delta_{jk}$$
$$= \frac{\varepsilon_{\text{out}}}{4\pi} \left( E_j(\vec{x}) E_k(\vec{x}) - \frac{1}{8} |\vec{E}(\vec{x})|^2 \delta_{jk} \right) - 2\bar{n}q\phi(\vec{x})$$
(1.44)

over a surface *S* surrounding molecule 1. The second equation is obtained by substituting the value of the ionic concentrations of the two species of oppositely charged ions implicit in the linear Poisson–Boltzmann equation  $\bar{n}_+(\vec{x}) + \bar{n}_-(\vec{x}) = 2\bar{n}q\phi(\vec{x})/kT$ . Because the last term for the contribution of the ionic pressure is negligible for the range of ionic concentrations considered under physiologic conditions we use only the electric field contribution<sup>25</sup>

$$T_{jk}^{E}(\vec{x}) \equiv \frac{\varepsilon_{\text{out}}}{4\pi} \left( E_{j}(\vec{x}) E_{k}(\vec{x}) - \frac{1}{8} |\vec{E}(\vec{x})|^{2} \delta_{jk} \right)$$
(1.45)

and the force on molecule 1 is then

$$\vec{F}_{12} = \int_{S} d\vec{x} \mathbf{T}^{E}(\vec{x}) \cdot \hat{n}(\vec{x})$$
(1.46)

where  $\hat{n}(\vec{x})$  is the unit normal to the surface at  $\vec{x}$ . For our purposes, the important fact about this expression is that the force depends only on the value of the electric field on the surface *S*. Thus, any other charge distributions that generates the same values of  $\vec{E}(\vec{x})$ 

on S will experience the same force. Because the boundary integral representation of the potential in the region exterior to the molecular surfaces, eq. (1.14) is the same as that for surface charge densities

$$\sigma_j(\vec{x}) = \frac{\varepsilon_{\text{out}}}{4\pi} h_j(\vec{x}), \quad \vec{x} \in \partial\Omega_j \tag{1.47}$$

on the respective molecular surfaces in a surrounding medium with dielectric constant  $\varepsilon_{out}$  and Debye parameter  $\kappa$  the force is simply

$$\vec{F}_{12} = \int_{\partial \Omega_1} d\vec{x} \sigma_1(\vec{x}) \vec{E}_2(\vec{x}).$$
(1.48)

Note that  $\sigma_j(\vec{x})$  from the boundary integral representation is not the same as the physical charge density that exists at the dielectric interface. Substituting

$$\vec{E}_{2}(\vec{x}) = -\vec{\nabla}\phi_{2}(\vec{x}) = -\frac{1}{4\pi} \int_{\partial\Omega_{2}} d\vec{x}' h_{2}(\vec{x}')\vec{\nabla}_{x} \frac{e^{-\kappa|\vec{x}-\vec{x}'|}}{|\vec{x}-\vec{x}'|} \quad (1.49)$$

gives the expression for the total force

$$\vec{F}_{12} = -\frac{\varepsilon_{\text{out}}}{16\pi^2} \int_{\partial\Omega_1} d\vec{x} \int_{\partial\Omega_2} d\vec{x}' h_1(\vec{x}) h_2(\vec{x}') \vec{\nabla}_x \frac{e^{-\kappa |\vec{x} - \vec{x}'|}}{|\vec{x} - \vec{x}'|}.$$
 (1.50)

Using a similar argument, the torque on molecule 1 due to the charges on molecule 2 is

$$\vec{T}_{12} = -\frac{\varepsilon_{\text{out}}}{16\pi^2} \int_{\partial\Omega_1} d\vec{x} \int_{\partial\Omega_2} d\vec{x}' h_1(\vec{x}) h_2(\vec{x}')(\vec{x} - \vec{x}_0) \times \vec{\nabla}_x \frac{e^{-\kappa |\vec{x} - \vec{x}'|}}{|\vec{x} - \vec{x}'|}.$$
(1.51)

#### Fast Force Evaluation Using an Adaptive Multipole Algorithm

Next we describe a method that uses a variation of the cell multipole method along with the BEM for solving the linear Poisson–Boltzmann equation shown earlier to rapidly calculate the force and torque on the molecules due to electrostatic interactions. As will be shown in the next section, the polarization corrections to the charge densities of each molecule due to the electric field of the other only adds a small correction to the calculated force and torque. Because it would be computationally expensive, relative to their magnitude, to calculate these corrections only the charge densities  $\tilde{h}_j^0$  from the BEM solution of the isolated molecules are used in the multipole method. Also, because this method is designed to be part of a Brownian dynamics or molecular dynamics simulation, as many quantities as possible are precomputed before the simulation begins. The speed of these computations are therefore considered unimportant.

The cell multipole method is a fast, efficient algorithm to calculate the electric potential due to a set of particles that interact according to a pairwise potential.<sup>20,21</sup> In this method, a cubic cell is defined such that it contains all of the particles of interest. This cell is then equally subdivided into eight smaller cubic cells, and this procedure iterated until no cell at the finest level contains too many particles. The grouping of the cells is represented by an octatree data structure in which any given cell is linked to its eight daughter cells. Contributions to the potential are then calculated using a multipole expansion about the center of a cell containing the source charges and a Taylor expansion about the center of a cell containing the field point. The cells are chosen such that the source cells and field cells each form a set that covers the largest cubic cell and the sizes of the cells are such that the approximation error is less than a specified value.

It is useful to define a set of unique indices for each cubic cell in the octatree structure with  $N_L$  levels. The cells will be denoted by  $C_{j,\vec{k}}$ , where  $j = 0, ..., N_L - 1$  is the level number, and the cells within a level are indexed by a vector  $\vec{k}$  with integer components and  $0 \le (\vec{k})_l < 2^j$ . The cell  $C_{j,\vec{k}}$  is then defined by the region

$$\vec{x} \in \mathbf{R}^3$$
,  $-a2^{-j} < (\vec{x} - \vec{c}_{j,\vec{k}})_m < a2^{-j}$ ,  $m = 1, 2, 3$  (2.52)

with the center

$$\vec{c}_{j,\vec{k}} \equiv a \bigg( 2^{-j} \vec{k} + \frac{1}{2} (2^{-j} - 1)(1, 1, 1) \bigg).$$
 (2.53)

This is a convenient notation because whether a cubic region at the next higher level is contained in a given cubic region may be determined simply from the index as

$$C_{j+1,\vec{l}} \subset C_{j,\vec{k}} \Leftrightarrow \left[\frac{(\vec{l})_m}{2}\right] = (\vec{k})_m, \quad m = 1, 2, 3$$
 (2.54)

where [x] denotes the integer part of x. Each cell at level  $j < N_L - 2$  has exactly eight daughter cells at level j + 1 contained within it. The unique level 0 cell is centered at the origin, i.e.,  $\vec{c}_{0,\vec{0}} = \vec{0}$ , and has sides of length a.

The set of polynomials  $\Phi_{j,\vec{k}}^{m}(\vec{x})$  for the multipole expansions within each cell are chosen to be mutually orthonormal over the cell  $C_{i,\vec{k}}$  so that

$$d\vec{x} \Phi_{j,\vec{k}}^{m}(\vec{x}) \Phi_{j,\vec{k}}^{n}(\vec{x}) = \delta_{m,n}$$
(2.55)

A set of polynomials up to quadratic order that are orthonormal in the cubic region centered at the origin with sides of length 1,  $C_{0,1}^1$ , are given in Table 1. The polynomials for cell  $C_{j,\bar{k}}$  are then defined in terms of these by

$$\Phi_{j,\vec{k}}^{m}(\vec{x}) \equiv 8^{j/2} \Phi^{m}(2^{j}(\vec{x} - \vec{c}_{j,\vec{k}}))$$
(2.56)

with

$$\Phi^{m}(\vec{x}) \equiv a^{-(3/2)} P_{m}(a^{-1}\vec{x})$$
(2.57)

Table 1. Polynomials	That Are	Orthonormal	on	the	Unit	Cube
Centered at the Origin	ı.					

$P_{01}$ $P_{11}$ $P_{12}$ $P_{13}$	$1  2\sqrt{3}x  2\sqrt{3}y  2\sqrt{3}z$
P <sub>21</sub>	$6\sqrt{5}x^2 - \frac{\sqrt{5}}{2}$
P <sub>22</sub>	$6\sqrt{5}y^2 - \frac{\sqrt{5}}{2}$
P <sub>23</sub>	$6\sqrt{5}z^2 - \frac{\sqrt{5}}{2}$
P <sub>24</sub> P <sub>25</sub> P <sub>26</sub>	12 <i>xy</i> 12 <i>yz</i> 12 <i>xz</i>

where  $\vec{c}_{j,\vec{k}}$  is the center of the cell  $C_{j,\vec{k}}$  and  $P_m(\vec{x})$  is one of the polynomials in Table 1. It will sometimes be convenient to refer to the indices m, j, and  $\vec{k}$  for the function  $\Phi_{j,\vec{k}}^m(\vec{x})$  collectively by a single index, denoted by a capital letter, i.e.,  $\Phi_I(\vec{x})$  or by the corresponding cell  $C_{j,\vec{k}}$  and the index m.

#### Precomputation

All of the calculations in this section are repeated for each molecule in isolation before the simulation begins. The molecules are also assumed to be rigid so that these solutions remain valid throughout the simulation. The molecule and its associated highest level cell  $C_{0,\bar{0}}$  should be centered at the origin. First a smooth representation of the solvent-excluded surface  $\partial \Omega_j$  of molecule j(j = 1, 2) is triangulated and the surface charge densities at each node,  $\tilde{h}_j^0$ , are calculated with the BEM using eq. (1.32). Next, a cubic region that contains the molecule is found and is recursively divided into eight daughter cells until the cells at the finest level have a size that is at least about five times the average length of the triangles' sides. These cubes are contained within an octatree data structure described in the previous section. Next, the multipole coefficients of the surface charge density at the highest (finest) level in the octatree structure are calculated using

$$b_{N_{L}-1,\bar{k}}^{m} = \int_{\partial\Omega} d\vec{x} \tilde{h}^{0}(\vec{x}) \Phi_{N_{L}-1,\bar{k}}^{m}(\vec{x})$$
(2.58)

where  $\tilde{h}^0(\vec{x})$  is the boundary element solution of an isolated molecule calculated using  $\vec{h}_j^0$  in eq. (1.32) and the approximation by linear function element in eq. (1.25), i.e.,

$$\tilde{h}_0(\vec{x}) = \sum_{\{l \mid \vec{x} \in C_{N_i-1,\vec{k}}\}} (\vec{h}_j^0)_l H_l(\vec{x}).$$
(2.59)

where the sum is over the indices for vertices of the  $\partial \Omega_j$  mesh that are in cell  $C_{N_L-1,\vec{k}}$ . The multipole coefficients of the surface charge density  $\tilde{h}_0(\vec{x})$  for the lower levels are then calculated by recursively applying

$$b_{j\bar{k}}^{m} = \sum_{n=1}^{10} \sum_{\alpha=1}^{8} A_{\alpha}^{m,n} b_{j+1,\bar{d}(\bar{k},\alpha)}^{n}$$
(2.60)

with

$$\begin{aligned} A_{\alpha}^{m,n} &= \int_{C_{0,\vec{0}}} d\vec{x} \Phi^{m}(\vec{x}) \Phi_{1,\vec{d}(\vec{0},\alpha)}^{n}(\vec{x}) \\ &= \int_{C_{0,\vec{0}}} d\vec{x} \Phi^{m}(\vec{x}) \Phi^{n} \left( 2\vec{x} - \left( \alpha_{4} - \frac{1}{2}, \, \alpha_{2} - \frac{1}{2}, \, \alpha_{1} - \frac{1}{2} \right) \right) \end{aligned}$$
(2.61)

where  $\vec{d}(\vec{k}, \alpha)$  is the vector index of the  $\alpha^{th}$  daughter cell

$$\vec{d}(\vec{k}, \alpha) = 2\vec{k} + (\alpha_4, \alpha_2, \alpha_1)$$
 (2.62)

and  $\alpha_k \equiv [\alpha/k]$ , k = 1, 2, 4 is the value of the appropriate bit in the binary representation of  $\alpha$ . The sum in eq. (2.60) is over the multipole coefficients for the eight cells contained within cell  $C_{j,\vec{k}}$ . As an aside we note that the orthogonal polynomials may be interpreted as scaling functions in a multiresolution analysis and that except for the fact that only the scaling function coefficients, not the wavelet coefficients are computed, this calculation of the lower level scaling coefficients is the same as the fast wavelet transform algorithm.<sup>26</sup>

#### Force and Torque Evaluation during the Simulation

During the simulation, the multipole coefficients calculated beforehand will be used for fast evaluation of the electrostatic force and torque on a molecule. For simplicity, it is assumed here that only molecule 2 moves and molecule 1 remains at the origin, i.e., only relative motion is considered. Let  $\vec{b}$  be the displacement of molecule 2 relative to the origin and **A** be the  $3 \times 3$  matrix that effects a rotation of the molecule relative to its initial orientation. A point  $\vec{x}$  on the molecule when it is initially situated at the origin is then transformed as  $\vec{x} \rightarrow A\vec{x} + \vec{b}$ . The expansion of  $\tilde{h}_0(\vec{x})$  for molecule 2 at the origin

$$\tilde{h}_0(\vec{x}) = \sum_l b_l^{(2)} \Phi_l(\vec{x})$$
(2.63)

with  $b_I$  the coefficients calculated using eqs. (2.58) and (2.60) becomes

$$\tilde{h}_0(\vec{x}) = \sum_I b_I^{(2)} \Phi_I(\mathbf{A}^{-1}(\vec{x} - \vec{b}))$$
(2.64)

for molecule 2 in the new position and orientation. The coefficients  $b_I^{(2)}$  in the latter expansion are the same as those in the former.

To calculate the force on molecule 1 at the origin due to molecule 2 at the position and orientation specified by  $\vec{b}$  and **A** using the multipole representation of the charge densities the integral kernel in eq. (1.14) must be calculated by

$$\vec{\mathbf{K}}_{I,J} \equiv -\int d\vec{x} \int d\vec{x}' \Phi_I(\vec{x}) \vec{\nabla}_x \frac{e^{-\kappa |\vec{x} - \vec{x}'|}}{|\vec{x} - \vec{x}'|} \Phi_J(\mathbf{A}^{-1}(\vec{x}' - \vec{b}))$$
$$= -\int d\vec{x} \int d\vec{x}' \Phi_I(\vec{x}) \vec{\nabla}_x \frac{e^{-\kappa |\vec{x} - \mathbf{A}\vec{x}' - \vec{b}|}}{|\vec{x} - \mathbf{A}\vec{x}' - \vec{b}|} \Phi_J(\vec{x}'). \quad (2.65)$$

Using the definition of eq. (2.56) for the polynomials this becomes

$$\vec{\mathbf{K}}_{I,J} = -s(j)^{-3} \int_{C_{j,\vec{k}_1}} d\vec{x} \int_{C_{j,\vec{k}_2}} d\vec{x}' P_{m_1}(s(j)^{-1}(\vec{x} - \vec{c}_{j,\vec{k}})) \times P_{m_2}(s(j)^{-1}(\vec{x}' - \vec{c}_{j,\vec{k}})) \vec{\nabla}_x \frac{e^{-\kappa |\vec{x} - \mathbf{A}\vec{x}' - \vec{b}|}}{|\vec{x} - \mathbf{A}\vec{x}' - \vec{b}|}$$
(2.66)

where the polynomial indices are  $I = (m_1, j, \vec{k}_1)$  and  $J = (m_2, j, \vec{k}_2)$  and  $s(j) \equiv 2^{-j}a$  is the length of a side of a cell at level *j*. Because only matrix elements between cells at the same level are needed, the level indices are the same. Defining  $\vec{y} \equiv s^{-1}(\vec{x} - \vec{c}_{j,\vec{k}_1})$  and  $\vec{y}' \equiv s^{-1}(\vec{x}' - \vec{c}_{j,\vec{k}_2})$  the previous equation becomes

 $\vec{\mathbf{K}}_{I,J}$ 

$$= -s(j) \int_{C_{0,\vec{0}}} d\vec{x} \int_{C_{0,\vec{0}}} d\vec{x}' \frac{e^{-s(j)\kappa|s(j)^{-1\vec{k}'}+\vec{y}-\mathbf{A}\vec{y}'|}}{|s(j)^{-1}\vec{R}'+\vec{y}-\mathbf{A}\vec{y}'|} P_{m_1}(\vec{y}) P_{m_2}(\vec{y}')$$
(2.67)

with  $\vec{R}' \equiv \vec{c}_{j,\vec{k}_1} - \mathbf{A}\vec{c}_{j,\vec{k}_2} - \vec{b}$ . Defining

$$\vec{K}'(\kappa, \vec{R}, \vec{x}, \vec{x}') \equiv -\vec{\nabla}_x \frac{e^{-\kappa |\vec{R} + \vec{x} - A\vec{x}'|}}{|\vec{R} + \vec{x} - A\vec{x}'|}$$
(2.68)

and its associated matrix elements at level 0

$$\vec{\mathbf{K}}_{m_1,m_2}'(\kappa,\vec{R}) = \int_{C_{0,\vec{0}}} d\vec{x} \int_{C_{0,\vec{0}}} d\vec{x}' \vec{K}'(\kappa,\vec{R},\vec{x},\vec{x}') P_{m_1}(\vec{x}) P_{m_2}(\vec{x}')$$
(2.69)

gives the relation between matrix elements of  $\vec{K}'$  and  $\vec{K}$ 

$$\vec{\mathbf{K}}_{I,J} = s(j)\vec{\mathbf{K}}'_{m_1,m_2}(s(j)\kappa, s(j)^{-1}\vec{R}').$$
(2.70)

This implies that the matrix elements of the integral kernel  $\vec{K}(\vec{x}, \vec{x}')$  defined in eq. (2.65) for any choice of indices are related to the level 0 coefficients of  $\vec{K}'_{m_1,m_2}$ ,  $m_1$ ,  $m_2 = 1, \ldots, 10$ . Analytical approximations for these latter coefficients are given in Appendix B.

According to eq. (1.50) the force on molecule 1 due to the electric field of molecule 2 may be calculated using the multipole coefficients as

$$\vec{F}_{12} = \frac{\varepsilon_{\text{out}}}{16\pi^2} \sum_{(C,C')\in S} \sum_{m,m'=1}^{10} b^{(1)}_{(C,m)} b^{(2)}_{(C',m')} \vec{\mathbf{K}}_{(C,m),(C',m')}$$
(2.71)

where the sum is over pairs of cells belonging to the set S. We next describe a method to find the pairs of cells in S such that the approximation error resulting from the sparse representation of the matrix  $\mathbf{\vec{K}}$  is less than a specified value. Throughout this article we make the assumption that the cells and corresponding octatree structures for each molecule are the same. This simplifies the discussion; however, it is straightforward to use different sets of cells for each molecule. A choice of indices consistent with the desired error bound  $\delta$  is to include matrix elements  $\vec{\mathbf{K}}_{(C,m),(C',m')}$ between cells that are as large as possible, subject to the condition that  $d(C, C') < \delta$  for a specified  $\delta < 1$ . d(C, C') is the upper bound on the approximation error described in appendix B. If this condition is not satisfied for nearby cells, even at the finest level, then the matrix element is calculated directly using the original boundary element representation. We denote the octatree structures for molecule 1 and molecule 2 as list 1 and list 2, respectively. A recursive algorithm for choosing such pairs of cells for which their corresponding indices are summed over in eq. (2.71) is

- 1. Let *j* be the lowest level (largest cells) for which cells exist in both list 1 and list 2
- 2. For each pair of nonempty cells  $(C^{(1)}, C^{(2)})$  at level *j*, one from each of list 1 and list 2
- 3. If  $d(C^{(1)}, C^{(2)}) < \delta$  for these cells calculate and store the matrix elements  $\{\vec{\mathbf{K}}_{(C^{(1)},m_1),(C^{(2)},m_2)}, m_1, m_2 = 1, \dots, 10\}$

If j = 0

Mark for direct calculation and stop

Else

For pairs of nonempty daughter cells, one from each of  $C^{(1)}$  and  $C^{(2)}$ 

The resulting matrix  $\vec{\mathbf{K}}$  will be sparse with nonzero elements appearing in the sum in eq. (2.71). Because the charge distribution is two-dimensional, many cells at the finer levels are empty. The condition that both cells corresponding to the matrix element are nonempty is checked before calculating the approximation error. The large number of empty cells also contributes to the sparsity of  $\mathbf{\tilde{K}}$ . The terms for the pairs of cells that are marked for direct calculation are calculated using the boundary element representation of the charge density on mesh triangles within the corresponding cells. Because the matrix is sparse the calculation of the force using eq. (2.71) will be fast, particularly if the molecules are widely separated. In fact, if the molecules are sufficiently separated only the matrix elements corresponding to the unique lowest level cell for each molecule will be used. This choice of cells is the same as would be used in the cell multipole method if the approximation error controlling the choice of cells were  $d(C^{(1)}, C^{(2)})$  for both the multipole expansion and local Taylor expansion about the field evaluation point.

The same method used to calculate  $\vec{F}_{12}$  may also be used to calculate the torque  $\vec{T}_{12}$ . A new integral kernel  $\vec{L}(\vec{x}, \vec{x}')$  with multipole coefficients

$$\vec{\mathbf{L}}_{I,J} \equiv -\int d\vec{x} \int d\vec{x}' \Phi_I(\vec{x}) \left[ x \times \vec{\nabla}_x \frac{e^{-\kappa |\vec{x} - \vec{x}'|}}{|\vec{x} - \vec{x}'|} \right] \Phi_J(\mathbf{A}^{-1}(\vec{x}' - \vec{b}))$$

$$= -\int d\vec{x} \int d\vec{x}' \Phi_I(\vec{x}) \left[ x \times \vec{\nabla}_x \frac{e^{-\kappa |\vec{x} - \mathbf{A}\vec{x}' - \vec{b}|}}{|\vec{x} - \mathbf{A}\vec{x}' - \vec{b}|} \right] \Phi_J(\vec{x}').$$
(2.72)

is required. As before, these coefficients may be calculated using only the level 0 coefficients

$$\vec{\mathbf{L}}'_{m_1,m_2}(\kappa,\vec{R}) = \int_{C_{0,\bar{0}}} d\vec{x} \int_{C_{0,\bar{0}}} d\vec{x}' \vec{L}'(\kappa,\vec{R},\vec{x},\vec{x}') P_{m_1}(\vec{x}) P_{m_2}(\vec{x}') \quad (2.73)$$

of the kernel

$$\vec{L}'(\kappa, \vec{R}, \vec{x}, \vec{x}') \equiv -x \times \vec{\nabla}_x \frac{e^{-\kappa \vec{R} + \vec{x} - \mathbf{A}\vec{x}'|}}{|\vec{R} + \vec{x} - \mathbf{A}\vec{x}'|}$$
(2.74)

by using

$$\vec{\mathbf{L}}_{I,J} = s(j)\vec{\mathbf{L}}'_{m_1,m_2}(s(j)\kappa, s(j)^{-1}\vec{R}').$$
(2.75)

Analytical approximations of the coefficients  $\tilde{\mathbf{L}}'_{m_1,m_2}(\kappa, \vec{R})$  are given in Appendix B. The torque about  $\vec{x}_0$  may then be calculated from

$$\vec{T}_{12} = \frac{\boldsymbol{\varepsilon}_{\text{out}}}{16\pi^2} \sum_{(C,C')\in S} \sum_{m,m'=1}^{10} b^{(1)}_{(C,m)} b^{(2)}_{(C',m')} (\vec{\mathbf{L}}_{(C,m),(C',m')} + (\vec{c} - \vec{x}_0) \times \vec{\mathbf{K}}_{(C,m),(C',m')}) = \frac{\boldsymbol{\varepsilon}_{\text{out}}}{16\pi^2} \sum_{(C,C')\in S} \sum_{m,m'=1}^{10} b^{(1)}_{(C,m)} b^{(2)}_{(C',m')} (\vec{\mathbf{L}}_{(C,m),(C',m')} + \vec{c} \times \vec{\mathbf{K}}_{(C,m),(C',m')}) - \vec{x}_0 \times \vec{F}_{12}.$$
(2.76)

The indices to sum over are chosen using the same algorithm described above with the same value of the parameter  $\delta$ .

#### **Sample Calculations**

The method described in this article for fast calculation of the electrostatic forces on large molecules in solution has been implemented as a program package in C++. In this section we give the results of some calculations of the electrostatic force between two proteins, both actin monomers (G-actin), as a

demonstration of the method. G-actin was chosen because it is sufficiently large, containing 5837 atoms with a maximum separation of 76 Å between their centers, and because of our interest in studying actin polymerization using Brownian dynamics simulations. The atomic coordinates were those used in ref. 27 with the bound Ca ion and ADP removed. AMBER force-field values were used for the partial atomic charges and Van der Waals radii.<sup>28</sup> A triangulation of the solvent-excluded surface with probe radius 1.5 Å was generated using the program MSMS.<sup>29</sup> Because this triangulation contained many triangles with a large vertex angle (near 180 degrees), which lead to large interpolation errors in the boundary element calculation, it was repeatedly refined until 4634 corner vertices and 9272 faces remained.<sup>23</sup> All calculations were done on a 500-MHz Pentium workstation.

Dielectric constants  $\varepsilon_{in} = 4.0$  and  $\varepsilon_{out} = 80.0$  were used as typical values for a protein molecule in water. All calculations were repeated for Debye parameter  $\kappa = 0.0$  and  $\kappa = 0.1027$ Å<sup>-1</sup> corresponding to pure water and 0.1 M monovalent ions at 300 K, the approximate physiologic concentration of Na<sup>+</sup> ions in solution, respectively. The center of mass separations of the two molecules was set to 50, 75, 100, and 200 Å. A 50-Å center of mass separation corresponds to a separation of only 8.0 Å for the nearest vertices on each molecular surface.

First, the accuracy and speed of the fast noniterative calculation of the polarization correction to the total force,  $F_{12}^{1-\text{step}}$ , is compared with the value,  $F_{12}^{\text{iter}}$ , calculated from an iterative solution of  $\Delta \vec{h}_j$ , j = 1, 2 using eq. (1.34). Equation (1.34) was iterated until  $|(\Delta \vec{h}_1, \Delta \vec{h}_2)| < 1.0 \times 10^{-10}$ . The average computation time for the noniterative method was 1390 s, which is about 36% of the time required for the iterative method. The relative errors in the total force  $\vec{F}_{12}$  for different separations are given in Table 2. The relative error in the polarization force increases for larger separation distances for  $\kappa = 0.0$  but decreases for  $\kappa = 0.1027$ . However, the relative error in the force remains small (<0.3%).

Next, iterative solutions of the boundary element equations were used to compare the fraction of the force resulting from the perturbation of the charge density on one molecule due to the electric field of the other,  $\vec{F}_{12}^{\text{polariz}}$ . The results are also shown in Table 2. As expected, polarization effects are largest when the molecules are near one another but the error remains below 10%, thus justifying the neglect of these corrections in the multipole version of the force calculation.

The force was then calculated using the fast multipole algorithm with 4<sup>3</sup> cells at the finest level for each molecule. Twentyfour of the cells were empty and the remaining ones contained an average of 116 vertices per cell. The results for  $\kappa = 0.0$  and  $\kappa = 0.1027$  Å<sup>-1</sup> are shown in Tables 3 and 4, respectively. Although a single value for the error bound  $\delta$  for different relative configurations would be used in practice, smaller values of  $\delta$  were chosen at larger separation distances to demonstrate results for a range of different numbers of nonzero matrix elements for a given separation distance. Also, smaller values of  $\delta$  were chosen for  $\kappa = 0.1027$  Å<sup>-1</sup> compared to  $\kappa = 0.0$  to achieve a similar sparsity of matrix  $\vec{K}$  because the electric field decreases much faster in the former case. For the same reason, the relative error for the same distance and matrix sparsity is

ecule and Error in the

Table 2. Fraction of Force from Polarization in the Electric Field of the Other Molecule and Error in the
Force Calculated Using the Fast Noniterative Method of Eq. (1.43) Relative to That Calculated
Using an Iterative Method of Eq. (1.34).

	κ =	= 0.0	$\kappa = 0.1027$		
r (Å)	$\frac{ \vec{F}_{12}^{\text{polariz}} }{ \vec{F}_{12}^{\text{iter}} }$	$\frac{ \vec{F}_{12}^{1-\text{step}} - \vec{F}_{12}^{\text{iter}} }{ \vec{F}_{12}^{\text{iter}} }$	$\frac{ \vec{F}_{12}^{\text{polariz}} }{ \vec{F}_{12}^{\text{iter}} }$	$\frac{ \vec{F}_{12}^{1-\text{step}}-\vec{F}_{12}^{\text{iter}} }{ \vec{F}_{12}^{\text{iter}} }$	
50	$9.35 \times 10^{-2}$	$5.64 \times 10^{-4}$	$6.90 \times 10^{-2}$	$2.95 \times 10^{-3}$	
75	$4.01 \times 10^{-2}$	$2.87 \times 10^{-3}$	$4.69 \times 10^{-3}$	$1.01 \times 10^{-4}$	
100	$1.92 \times 10^{-2}$	$2.66 \times 10^{-3}$	$3.91 \times 10^{-4}$	$8.19 \times 10^{-6}$	
200	$1.78 \times 10^{-3}$	$1.37 \times 10^{-3}$	$< 1.0 \times 10^{-6}$	$< 1.0 \times 10^{-6}$	

r is the center of mass separation distance.

larger for  $\kappa = 0.1027$  Å<sup>-1</sup> than for  $\kappa = 0.0$ ; however, for simulations, one is usually interested in the absolute error, which remains small.

### on an isolated molecule is done before the simulation and the force evaluation during the simulation is done quickly using the multipole representation of this charge density.

When the separation of the molecular surfaces is much closer than the 8 Å tested in the previous section it is expected that the continuum approximation of the ionic solution implicit in the Poisson–Boltzmann equation breaks down and that the method should be modified. This is true for any method that uses a continuum representation of the solvent. Numerical errors also become important when the separation of the molecular surfaces is on the order of the size of the surface elements. One solution to these problems is to merge the molecular surfaces near their interface. Another possibility is to replace the solvent layer separating the molecules by explicit molecules whose conformations

## Conclusion

We have given a detailed description of a calculational scheme, which is a combination of the BEM and the cell multipole method, which allows a fast calculation of the electrostatic force and torque on a macromolecule for use in Brownian dynamics or molecular dynamics simulations. Because the molecules are assumed to be rigid, the relatively slow calculation of the surface charge density

**Table 3.** Errors and Execution Times for Calculating the Force on Two Actin Monomers in Pure Water ( $\kappa = 0$ ) Whose Centers of Mass Are Separated by *r* Å.

r (Å)	$\delta$ Error bound	# Matrix elements	# Direct elements	$ \vec{F}_{12}^{\text{mult}} - \vec{F}_{12}^{1-\text{step}} $ (10 <sup>-6</sup> dynes)	$\frac{ \vec{F}_{12}^{\text{mult}} - \vec{F}_{12}^{1-\text{step}} }{ \vec{F}_{12}^{1-\text{step}} }$	$\frac{t(\vec{F}_{12}^{\text{mult}})}{t(\vec{F}_{12}^{1-\text{step}})}$
50	$1.0 \times 10^{-3}$	1372	444	1.44	0.145	0.31
	$1.0  imes 10^{-2}$	1297	190	1.58	0.159	0.12
	$1.0  imes 10^{-1}$	1071	94	2.53	0.256	$5.0  imes 10^{-2}$
75	$1.0 \times 10^{-3}$	1071	89	0.656	0.149	$3.6 \times 10^{-2}$
	$1.0  imes 10^{-2}$	367	10	0.991	0.225	$3.8 \times 10^{-3}$
	$1.0  imes 10^{-1}$	367	0	0.988	0.225	$3.1 \times 10^{-5}$
100	$1.0  imes 10^{-4}$	1297	121	0.110	$4.17 \times 10^{-2}$	$5.0 \times 10^{-2}$
	$1.0  imes 10^{-3}$	367	0	0.414	0.156	$3.1 \times 10^{-5}$
	$1.0  imes 10^{-2}$	139	0	0.550	0.208	$1.3 \times 10^{-5}$
200	$1.0 \times 10^{-6}$	1600	147	$5.64 \times 10^{-3}$	$7.67 \times 10^{-3}$	$6.2 \times 10^{-2}$
	$1.0 \times 10^{-5}$	367	0	$5.18  imes 10^{-2}$	$7.04  imes 10^{-2}$	$3.1 \times 10^{-5}$
	$1.0  imes 10^{-4}$	1	0	$3.94 \times 10^{-2}$	$5.36 \times 10^{-2}$	$< 1.0 \times 10^{-6}$

Each molecular surface of the monomer has 9272 triangular surface elements.  $\delta$  is the error bound used in the multipole algorithm described in the text. The number of nonzero elements in the matrix  $\vec{K}$  and the number of elements calculated directly using the boundary elements are given.  $\vec{F}_{12}^{\text{null}}$  and  $\vec{F}_{12}^{1-\text{step}}$  are the forces calculated using the multipole algorithm and the direct one-step BEM with the charge density of eq. (1.43), respectively. Execution times for the multipole method,  $t(\vec{F}_{12}^{\text{null}})$ , are given as a fraction of the time for the same calculation using the boundary element method without multipoles,  $t(\vec{F}_{12}^{1-\text{step}})$ .  $t(\vec{F}_{12}^{1-\text{step}})$  is approximately 1390 s on a 500-MHz Pentium workstation.

Table 4.	Errors and	Execution	Times for	Calculating	the Fo	rce on	Two	Actin	Monomers	in 0.1	M Solution
of Mone	ovalent Ions	$(\kappa = 0.10$	27 Å <sup><math>-1</math></sup> ).								

r (Å)	δ Error bound	# Matrix elements	# Direct elements	$\frac{ \vec{F}_{12}^{\text{mult}} - \vec{F}_{12}^{1-\text{step}} }{(10^{-6} \text{ dynes})}$	$\frac{ \vec{F}_{12}^{\text{mult}} - \vec{F}_{12}^{1-\text{step}} }{ \vec{F}_{12}^{1-\text{step}} }$	$\frac{t(\vec{F}_{12}^{\text{mult}})}{t(\vec{F}_{12}^{1-\text{step}})}$
50	$1.0 \times 10^{-3}$	1297	369	$4.56 \times 10^{-2}$	$5.21 \times 10^{-2}$	0.25
	$1.0 \times 10^{-2}$	1297	190	$8.23 \times 10^{-2}$	$9.41 \times 10^{-2}$	0.12
	$1.0  imes 10^{-1}$	1071	94	0.152	0.174	$5.0  imes 10^{-2}$
75	$5.0  imes 10^{-5}$	1297	430	$3.33 \times 10^{-3}$	0.126	0.28
	$1.0  imes 10^{-4}$	1297	291	$4.52 \times 10^{-3}$	0.171	0.19
	$1.0 \times 10^{-3}$	611	69	$1.08 \times 10^{-2}$	0.406	$3.0 \times 10^{-2}$

The notation is the same as in Table 3.

are sampled from an equilibrium ensemble using either a Monte Carlo or a molecular dynamics simulation.

Although quadratic orthogonal polynomials were used for the multipole expansion, it would be more efficient to use spherical harmonics up to a given order if polynomials of greater than quadratic order are used. This is because the error bound estimate, at least for the Coulomb case, may be formulated in terms of spherical harmonics, and there are fewer of these than independent polynomials of the same order.<sup>20</sup> This was not done here because the number of spherical harmonics up to quadratic order (9) is not sufficiently smaller than the number of independent quadratic polynomials (10) to justify the added complication.

There are several ways in which the method may be made faster. First, a higher order multipole approximation would be expected to improve performance because the computation time is currently dominated by direct boundary element contributions. This would require using spherical harmonics, as described above. Two approximations that would improve the speed of the boundary element calculation are using a lower order Gaussian quadrature formula for the off-diagonal elements and using larger surface elements such as described in ref. 30. Although these two approximations would lower the accuracy of the direct boundary element contribution to the force, this is acceptable if the multipole error bound is comparable.

It is also possible to apply the methods described here for solving the linear Poisson–Boltzmann equation to more highly charged systems. Reference 7 describes a method for scaling the solution of the linear Poisson–Boltzmann equation so that it agrees fairly well with the nonlinear Poisson–Boltzmann equation solution. The authors of this reference also found that the correction was unnecessary for systems with low charge density at physiologic ionic strengths, where the solutions are virtually identical.

Finally, it would be useful to apply some of the methods used in this article to solve for the surface charge density of an isolated molecule as well. Although this would not speed up the calculations during the simulation, it would reduce time and memory constraints on the initial step and allow the use of larger surface meshes.

#### Acknowledgements

We thank D. Sept for the actin coordinates and D. Steiger for many useful discussions.

#### Numerical Integration Over Boundary Elements

The integrals over the standard triangles, with vertices (0, 0), (1, 0), and (0, 1) in eq. (1.27) are calculated using Gaussian quadrature. The two-dimensional integrals are evaluated using a nonproduct form of the Gaussian quadrature scheme, i.e., not simply using a one-dimensional form in each dimension. Although it is more difficult to calculate the coefficients, the nonproduct form requires less function evaluations, in general. The integral of a function w(x, y) f(x, y) over a region T in this approximation is evaluated using

$$\int_{T} w(x, y) f(x, y) \, dx \, dy = \sum_{j=1}^{N(P)} w_j f(x_j, y_j).$$
(1.77)

The  $w_j$ ,  $x_j$ , and  $y_j$  depend on both the region *T* and the weight function w(x, y). These coefficients are chosen such that this equation is exact for  $f(x, y) = x^m y^n$ ,  $m + n \le P$ . Because the integrands for the diagonal matrix elements in eq. (1.29) have a 1/rsingularity, the weight function is chosen to be w(x, y) = $1/\sqrt{x^2 + y^2}$ . The weight function w(x, y) = 1 for the integrals in the off-diagonal elements. A third-order formula due to Radon is used, with the coefficients for w(x, y) = 1 taken from ref. 31 for the off-diagonal matrix elements and a fifth-order Radon formula, with the coefficients for  $w(x, y) = 1/\sqrt{x^2 + y^2}$  calculated using Macsyma, is used for the diagonal elements.<sup>32</sup> The coefficients for the latter are given in Table 5.

## Approximate Integral Kernel Matrix Elements and Error Bounds

All of the matrix elements of the integral kernels  $\vec{K}$  and  $\vec{L}$  may be calculated from the level 0 matrix elements of the kernel  $\vec{K}'$  using

**Table 5.** Weights  $w_j$  and Evaluation Points  $(x_j, y_j)$  for Fifth-Order Gaussian Quadrature Over a Triangular Region with Vertices (0, 0), (1, 0), and (0, 1) and with Weight Function  $w(x, y) = 1/\sqrt{x^2 + y^2}$ .

W	X	у
$9.087600805043129  imes 10^{-2}$	$4.519108816401833  imes 10^{-1}$	$4.519108816401833  imes 10^{-1}$
$2.622451129254795 \times 10^{-2}$	$5.502601270767638  imes 10^{-2}$	$5.502601270767638 \times 10^{-2}$
$9.915323050713348  imes 10^{-2}$	$2.767054424221354 \times 10^{-1}$	$2.767054424221354 \times 10^{-1}$
$7.682740556667928  imes 10^{-2}$	$7.720289712733758 \times 10^{-1}$	$1.054562060711828 \times 10^{-1}$
$7.682740556667928  imes 10^{-2}$	$1.054562060711828 \times 10^{-1}$	$7.720289712733758 \times 10^{-1}$
$6.409495859505721  imes 10^{-2}$	$4.010643051428878  imes 10^{-1}$	$5.478385084588691 \times 10^{-2}$
$6.409495859505721  imes 10^{-2}$	$5.478385084588691 \times 10^{-2}$	$4.010643051428878 \times 10^{-1}$

eq. (2.70) and the matrix elements of the kernel  $\vec{L}'$  using eq. (2.75), respectively. A closed form expression for these matrix elements may be obtained by using the Taylor series expansion of the kernel to  $\mathbb{O}(x_j^m x_k'^n)$ , m + n = 2 and using the orthonormality of the polynomials  $P(\vec{x})$  in eqs. (2.69) and (2.73).

Defining

$$F_{1} \equiv \frac{(\kappa R + 1)}{R^{3}},$$

$$F_{2} \equiv \frac{(\kappa^{2}R^{2} + 3\kappa R + 3)}{R^{5}},$$

$$F_{3} \equiv \frac{(\kappa^{3}R^{3} + 6\kappa^{2}R^{2} + 15\kappa R + 15)}{R^{7}},$$
(2.78)

the matrix elements for the integral kernel  $\vec{K}'(\kappa, \vec{R}, \vec{x}, \vec{x}')$  are

$$\begin{aligned} (\vec{\mathbf{K}}_{0,0})_{l} &= e^{-\kappa R} F_{1} \left( 1 + \frac{1}{12} \kappa^{2} \right) \vec{R}_{l}, \\ (\vec{\mathbf{K}}_{1j,0})_{l} &= \frac{e^{-\kappa R}}{2\sqrt{3}} \left( F_{1} \delta_{jl} - F_{2} \vec{R}_{j} \vec{R}_{l} \right), \\ (\vec{\mathbf{K}}_{0,1j})_{l} &= \frac{e^{-\kappa R}}{2\sqrt{3}} \left( F_{2} (R\mathbf{A})_{j} \vec{R}_{l} - F_{1} \mathbf{A}(l,j) \right), \\ (\vec{\mathbf{K}}_{1j,1k})_{l} &= \frac{e^{-\kappa R}}{24} \left( F_{2} (\delta_{jl} (R\mathbf{A})_{k} + \delta_{kl} (R\mathbf{A})_{j} + A_{jk} \vec{R}_{l} + \mathbf{A}_{kj} \vec{R}_{l} + \mathbf{A}_{lj} \vec{R}_{k} + A_{lik} \vec{R}_{j} \right) - F_{3} (\vec{R}_{j} (R\mathbf{A})_{k} \vec{R}_{l} + (R\mathbf{A})_{j} \vec{R}_{k} \vec{R}_{l})), \\ (\vec{\mathbf{K}}_{2j,0})_{l} &= \frac{e^{-\kappa R}}{12\sqrt{5}} \left( F_{3} \vec{R}_{j}^{2} \vec{R}_{l} - F_{2} (2\delta_{jl} \vec{R}_{j} + \vec{R}_{l}) \right), \quad j = 1, 2, 3 \\ (\vec{\mathbf{K}}_{0,2j})_{l} &= \frac{e^{-\kappa R}}{12\sqrt{5}} \left( F_{3} (R\mathbf{A})_{j}^{2} \vec{R}_{j} - F_{2} (\mathbf{A} (2\mathbf{A} (l, j) (R\mathbf{A})_{j} + \vec{R}_{l})), \quad j = 1, 2, 3 \end{aligned}$$

$$(\vec{\mathbf{K}}_{24,0})_{l} = \frac{e^{-\kappa R}}{12} (F_{3}\vec{R}_{1}\vec{R}_{2}\vec{R}_{l} - F_{2}(\delta_{l1}\vec{R}_{2} + \delta_{l2}\vec{R}_{1})),$$

$$(\vec{\mathbf{K}}_{25,0})_{l} = \frac{e^{-\kappa R}}{12} (F_{3}\vec{R}_{2}\vec{R}_{3}\vec{R}_{l} - F_{2}(\delta_{l2}\vec{R}_{3} + \delta_{l3}\vec{R}_{2})),$$

$$(\vec{\mathbf{K}}_{26,0})_{l} = \frac{e^{-\kappa R}}{12} (F_{3}\vec{R}_{1}\vec{R}_{3}\vec{R}_{l} - F_{2}(\delta_{l1}\vec{R}_{3} + \delta_{l3}\vec{R}_{1})),$$

$$(\vec{\mathbf{K}}_{0,24})_{l} = \frac{e^{-\kappa R}}{12} (F_{3}(R\mathbf{A})_{1}(R\mathbf{A})_{2}\vec{R}_{l} - F_{2}(\mathbf{A}(l, 1)(R\mathbf{A})_{2} + (R\mathbf{A})_{1}\mathbf{A}(l, 2))),$$

$$(\vec{\mathbf{K}}_{0,25})_{l} = \frac{e^{-\kappa R}}{12} (F_{3}(R\mathbf{A})_{2}(R\mathbf{A})_{3}\vec{R}_{l} - F_{2}(\mathbf{A}(l, 2)(R\mathbf{A})_{3} + (R\mathbf{A})_{2}\mathbf{A}(l, 3))),$$

$$(\vec{\mathbf{K}}_{0,26})_{l} = \frac{e^{-\kappa R}}{12} (F_{3}(R\mathbf{A})_{1}(R\mathbf{A})_{3}\vec{R}_{l} - F_{2}(\mathbf{A}(l, 1)(R\mathbf{A})_{3} + (R\mathbf{A})_{2}\mathbf{A}(l, 3))),$$

$$(\vec{\mathbf{K}}_{1j,2k})_{l} = (\vec{\mathbf{K}}_{2k,1j})_{l} = 0, \quad j = 1, 2, 3; k = 1, \dots, 6$$

$$(\vec{\mathbf{K}}_{2j,2k})_{l} = 0, \quad j, k = 1, \dots, 6. \qquad (2.79)$$

The matrix elements for the integral kernel  $\vec{L}'(\kappa, \vec{R}, \vec{x}, \vec{x}')$  are

$$(\vec{\mathbf{L}}_{1j,0}) = \frac{e^{-\kappa R}}{2\sqrt{3}} F_1 \sum_{k=1}^{3} \varepsilon_{ljk} \vec{R}_k,$$
  
$$(\vec{\mathbf{L}}_{1j,1k})_l = \frac{e^{-\kappa R}}{12} \sum_{m=1}^{3} \varepsilon_{ljm} (F_2(R\mathbf{A})_k \vec{R}_m - F_1 \mathbf{A}(m,k)),$$
  
$$(\vec{\mathbf{L}}_{2j,0}) = -\frac{e^{-\kappa R}}{6\sqrt{5}} F_2 \vec{R}_j \sum_{m=1}^{3} \varepsilon_{ljm} \vec{R}_m, \quad j = 1, 2, 3$$



**Figure 2.** The recursive algorithm for choosing pairs of cells is illustrated by showing one step. The error bound d(C, C') for two cells in (a) is larger than the desired value  $\delta$  so the error bounds for each pair of daughter cells are compared with  $\delta$ . If the bound for these cells is found to be less than  $\delta$ , as in (b), the recursion terminates. The algorithm then continues checking other pairs of cells at the coarser level.

$$(\vec{\mathbf{L}}_{24,0}) = \frac{e^{-\kappa R}}{12} F_2(-\vec{R}_1 \vec{R}_3, \vec{R}_2 \vec{R}_3, \vec{R}_1^2 - \vec{R}_2^2),$$

$$(\vec{\mathbf{L}}_{25,0}) = \frac{e^{-\kappa R}}{12} F_2(\vec{R}_2^2 - \vec{R}_3^2, -\vec{R}_1 \vec{R}_2, \vec{R}_1 \vec{R}_3),$$

$$(\vec{\mathbf{L}}_{26,0}) = \frac{e^{-\kappa R}}{12} F_2(\vec{R}_1 \vec{R}_2, \vec{R}_3^2 - \vec{R}_1^2, -\vec{R}_2 \vec{R}_3),$$

$$(\vec{\mathbf{L}}_{0,0})_l = 0,$$

$$(\vec{\mathbf{L}}_{0,1j})_l = 0,$$

$$(\vec{\mathbf{L}}_{0,2j})_l = 0, \quad j = 1, \dots, 6$$

$$(\vec{\mathbf{L}}_{1j,2k})_l = (\vec{\mathbf{L}}_{2k,1j})_l = 0, \quad j = 1, 2, 3; k = 1, \dots, 6$$

$$(\vec{\mathbf{L}}_{2j,2k})_l = 0, \quad j, k = 1, \dots, 6.$$

$$(2.80)$$

Because many matrix elements of  $\vec{K}'$  and  $\vec{L}'$  are zero, sparse matrix multiplication is used in the program implementing the method.

We next derive a bound on the approximation error of each term in the sum of eq. (2.71) for  $\vec{F}_{12}$  resulting from truncating the Taylor series expansion of the kernel  $\vec{K}'$  to obtain the matrix elements given above. This bound is used to determine which matrix elements in the compressed form of the orthonormal polynomial basis of  $\vec{K}(\vec{R}, \vec{x} - \vec{x}')$  are nonzero. Because truncating the Taylor series expansion for  $\vec{K}'$  to  $\mathcal{O}(\vec{x}_{j}^{m}\vec{x}_{k}^{(n)})$ , m + n = 2 corresponds to the same approximation for  $\vec{K}$  we examine the error in the latter. We also know that the maximum approximation error for each term in eq. (1.50) occurs when the charge distribution in the cells corresponding to each index is a delta function, i.e., point

charge, at the closest pair of points. The error bound for the terms  $\sum_{m,m'=1}^{10} b_{(C,m)}^{(1)} b_{(C',m')}^{(2)} \vec{\mathbf{K}}_{(C,m),(C',m')}$  is denoted by d(C, C') and is

$$d(C, C') = \max_{\vec{x} \in C, \vec{x}' \in C'} |\vec{\mathbf{K}}(\vec{R}, \vec{x} - \vec{x}') - \vec{\mathbf{K}}_2(\vec{R}, \vec{x} - \vec{x}')|$$
(2.81)

where  $\vec{\mathbf{K}}_2(\vec{R}, \vec{x} - \vec{x}')$  is the Taylor series expansion up to quadratic order in  $\vec{x} - \vec{x}'$ 

$$\vec{\mathbf{K}}_{2}(\vec{R}, \vec{x} - \vec{x}') \equiv \vec{d}_{0}(\vec{R}) + \sum_{k=1}^{3} \vec{d}_{1}^{k} (\vec{x} - \vec{x}')_{k}$$
(2.82)  
+  $\sum_{k=1}^{3} \vec{d}_{k}^{kl} (\vec{x} - \vec{x}')_{k} (\vec{x} - \vec{x}')_{k}$ 

Defining  $\vec{y} \equiv \vec{x}' - \vec{x}$  the kernel function is

$$\vec{\mathbf{K}}(\vec{R},\vec{y}) = e^{-\kappa |\vec{R}-\vec{y}|} \left( \frac{1}{|\vec{R}-\vec{y}|^3} + \frac{\kappa}{|\vec{R}-\vec{y}|^2} \right) (\vec{R}-\vec{y}). \quad (2.83)$$

Next, we assume that the maximum in eq. (2.81) occurs when  $\vec{R} \cdot \vec{y} = |\vec{R}| |\vec{y}|$ . Defining  $R \equiv |\vec{R}|$ ,  $r \equiv |\vec{y}|$ , and  $\hat{R} \equiv \vec{R}/|\vec{R}|$  yields

$$\vec{\mathbf{K}}(R,r) = e^{-\kappa(R-r)} \left( \frac{1}{(R-r)^2} + \frac{\kappa}{R-r} \right) \hat{R}$$
(2.84)

and the scalar function K(R, r) may be defined by

$$\vec{\mathbf{K}}(R,r) \equiv K(R,r)\hat{R} \tag{2.85}$$

and likewise for  $K_2(R, r)$  to give

$$K_{2}(R, r) = \frac{e^{-\kappa R}}{R^{2}} \left[ 1 + \kappa R + \frac{(\kappa^{2}R^{2} + 2\kappa R + 2)}{R} r + \frac{1}{2} \frac{(\kappa^{3}R^{3} + 3\kappa^{2}R^{2} + 6\kappa R + 6)}{R^{2}} r^{2} \right].$$
 (2.86)

The error bound of eq. (2.81) then becomes

$$d(C, C') = \max_{0 \le r \le \sqrt{3}s(j)} K(R, r) - K_2(R, r)$$
(2.87)

with s(j) the length of one side of the cells *C* and *C'* and the  $\sqrt{3}$  factor coming from the relative configuration of the cells with the largest error in which the line connecting their centers also passes through a corner of each cell. This expression is the remainder term for the Taylor series, whose value may be found from the Lagrange form

$$R_{3}(r, c) = \frac{1}{3!} \frac{\partial^{3} K(R, c)}{\partial r^{3}} r^{3}, \quad 0 < c < r.$$
(2.88)

Because

$$\frac{\partial^3 K(R, r)}{\partial r^3} = e^{-\kappa(R-r)} \left( \frac{24}{(R-r)^5} + \frac{24\kappa}{(R-r)^4} + \frac{12\kappa^2}{(R-r)^3} + \frac{4\kappa^3}{(R-r)^2} + \frac{\kappa^4}{R-r} \right)$$
(2.89)

is monotonically decreasing for r < R an upper bound d(C, C')on the approximation error may be found from the corresponding upper bound on the remainder term of eq. (2.88) to give

$$d(C, C') \le 3^{3/2} e^{-\kappa a} \left(\frac{4}{a^3} + \frac{4\kappa}{a^2} + \frac{2\kappa^2}{a} + \frac{2}{3}\kappa^3 + \frac{1}{6}\kappa^4 a\right) (s(j))^3.$$
(2.90)

with  $a \equiv R - \sqrt{3}s(j)$ . However, rather than using this upper bound on the remainder term we simply evaluate it directly from eq. (2.87) with  $r = \sqrt{3}s(j)$ . This is because the smaller error bound that results implies that there are fewer matrix elements of  $\vec{K}(\vec{R}, \vec{x} - \vec{x}')$  in the truncated matrix, which more than compensates for the additional computational time needed to evaluate d(C, C') directly.

## References

- 1. Honig, B.; Nicholls, A. Science 1995, 268, 1144.
- 2. Davis, M.; McCammon, J. Chem Rev 1990, 90, 509.
- 3. Sharp, K.; Honig, B. Annu Rev Biophys Biophys Chem 1990, 19, 301.
- 4. Constanciel, R.; Contreras, R. Theoret Chem ACTA 1984, 65, 1.
- Still, W.; Tempczyk, A.; Hawley, R.; Hendrickson, T. J Am Chem Soc 1990, 112, 6127.
- 6. Gabdoulline, R.; Wade, R. J Phys Chem 1996, 100, 3868.

- 7. Fogolari, F.; Zuccato, P.; Esposito, G.; Viglino, P. Biophys J 1999, 76, 1.
- 8. Miertus, S.; Scrocco, E.; Tomasi, J. Chem Phys 1981, 55, 117.
- 9. Rashin, A.; Namboodiri, K. J Phys Chem 1987, 91, 6003.
- 10. Zauhar, R. J.; Morgan, R. J Mol Biol 1985, 186, 815.
- 11. Zauhar, R. J.; Morgan, R. J Comput Chem 1988, 9, 171.
- 12. Yoon, B. J.; Lenhoff, A. J Comput Chem 1990, 11, 1080.
- Juffer, A. H.; Botta, E. F.; van Keulen, B. A.; van der Ploeg, A.; Berendsen, H. J. J Comput Phys 1991, 97, 144.
- 14. Zhou, H.-X. Biophys J 1993, 65, 955.
- Horvath, D.; van Belle, D.; Lippens, G.; Wodak, S. J Chem Phys 1996, 104, 6679.
- 16. Liang, J.; Subramaniam, S. Biophys J 1997, 73, 1830.
- Bharadwaj, R.; Windemuth, A.; Sridharan, S.; Honig, B.; Nicholls, A. J Comput Chem 1995, 16, 898.
- 18. Purisima, E. O. J Comput Chem 1998, 19, 1494.
- 19. Zauhar, R. J.; Varnek, A. J Comput Chem 1996, 17, 864.
- 20. Greengard, L.; Rokhlin, V. J Comput Phys 1987, 73, 325.
- 21. Ding, H.-Q.; Karasawa, N.; Goddard, W. A., III. J Chem Phys 1992, 97, 4309.
- 22. Vorobjev, Y. N.; Scheraga, H. A. J Comput Chem 1997, 18, 569.
- Chen, G.; Zhou, J. Boundary Element Methods; Academic Press: San Diego, CA, 1992.
- 24. Connolly, M. J Appl Crystallogr 1983, 16, 548.
- Gilson, M. K.; Davis, M. E.; Luty, B. A.; McCammon, J. A. J Phys Chem 1993, 14, 3591.
- 26. Mallat, S. Transac Am Math Soc 1989, 315, 69.
- 27. Sept, D.; Elcock, A. H.; McCammon, J. A. J Mol Biol 1999, 294, 1181.
- Cornell, W.; Cieplakand, P.; Bayly, C.; Gould, I.; Merz, K., Jr.; Ferguson, D.; Spellmeyer, D.; Fox, T.; Caldwell, J.; Kollman, P. J Am Chem Soc 1995, 117, 5179.
- 29. Sanner, M.; Olson, A. J.; Spehner, J. C. Biopolymers 1996, 38, 305.
- 30. Totrov, M.; Abagyan, R. Biopolymers 2001, 60, 124.
- Stroud, A. Approximate Calculation of Multiple Integrals; Prentice-Hall: Englewood Cliffs, NJ, 1971.
- 32. Radon, J. Monat Math 1948, 52, 286.