The Fast Multipole Boundary Element Method for Molecular Electrostatics: An Optimal Approach for Large Systems

RANGANATHAN BHARADWAJ, ANDREAS WINDEMUTH, S. SRIDHARAN, BARRY HONIG,* and ANTHONY NICHOLLS

Department of Biochemistry and Molecular Biophysics and Center for Biomolecular Simulation, Columbia University, 630 W. 168th Street, New York, New York 10032

Received 9 August 1994; accepted 21 November 1994

ABSTRACT

We propose a fast implementation of the boundary element method for solving the Poisson equation, which approximately determines the electrostatic field around solvated molecules of arbitrary shape. The method presented uses computational resources of order $O(N)$ only, where N is the number of elements representing the dielectric boundary at the molecular surface. The method is based on the Fast Multipole Algorithm by Rokhlin and Greengard, which is used to calculate the Coulombic interaction between surface elements in linear time. We calculate the solvation energies of a sphere, a small polar molecule, and a moderately sized protein. The values obtained by the boundary element method agree well with results from finite difference calculations and show a higher degree of consistency due to the absence of grid dependencies. The boundary element method can be taken to a much higher accuracy than is possible with finite difference methods and can therefore be used to verify their validity.

Introduction

Electrostatic interactions are known to play a key role in determining the structure and activity of biomolecules.[1,2,3] They govern the interactions between proteins and ligands in enzymatic reactions, they are a crucial factor in molecular interactions involving lipid membranes[4] or DNA,[5] and they are important in determining the stability of folded proteins.

It has been shown that the continuum electrostatics model provides a reliable means of obtaining qualitative and quantitative insights into electrostatic interactions in solvated molecules.[6,7] The continuum electrostatics model assumes that both solvent and solute can be treated as continuous

*Author to whom all correspondence should be addressed.

CCC 0192-8651 / 95 / 070898-16

linear dielectric media with different dielectric constants. The electrostatic potential is then described by the Poisson equation:

$$\nabla(\epsilon(\mathbf{r})\nabla\phi(\mathbf{r})) = -4\pi\rho(\mathbf{r}) \quad (1)$$

where $\phi(\mathbf{r})$, $\epsilon(\mathbf{r})$, and $\rho(\mathbf{r})$ are the electrostatic potential, the dielectric constant, and the charge density, respectively. The dielectric constant $\epsilon(\mathbf{r})$ is assumed to have different values inside and outside the molecule, with the molecular surface separating the two regions. The charge distribution $\rho(\mathbf{r})$ describes charged regions within the solute and is usually represented by a number of point charges at the centers of solute atoms.

The choice of molecular surface is a critical step in the model. Simple models of macromolecules assume a geometrical shape, such as a sphere for soluble proteins or a cylinder for DNA. For such models the Poisson equation can be solved analytically.[8-11] In most cases, however, a more detailed representation of the molecule is needed, such as the molecule surface described by Richards,[12] which is defined as the envelope formed by the inner surface of a solvent probe sphere rolled around a molecule. Such a surface designates only the volume accessible to solvent as exterior and therefore serves as the dielectric boundary. Because this surface has no simple analytic representation, it becomes necessary to apply a numeric approximation technique to solve the Poisson equation.

The most common numerical method used is the finite difference method.[13,14] In the finite difference method the space in and around the solute is mapped onto a three-dimensional lattice. The Poisson equation can then be solved by any of a large number of methods for solving partial differential equations. Good results have been obtained using a successive overrelaxation (SOR) method.[15] Recently, multigrid iteration methods have also been employed.[16,17] Finite difference methods are the fastest available but have accuracy problems due to the limitations inherent in mapping physical quantities onto a lattice. Experimental measurements with biomolecular systems often resolve energies and potentials to less than 1 kT accuracy. For example, the pK_a of an ionizable group in a protein can be influenced by its local potential.[18] Experimentally measured pK_a's are accurate up to 0.1 units, corresponding to an accuracy of 0.2 kT in potential. The finite difference method does provide this accuracy while computing energies for small molecules, for which it is possible to go to very fine lattice spacings. However, attaining the same accuracy for larger systems such as biomolecules is not always feasible due mainly to the excessive memory demands imposed by the lattice resolution required to obtain reliable numbers. There is, therefore, a real need for methods to calculate electrostatic energies of large molecules accurately and rapidly. Ideally, such methods should be suited to workstations commonly available to researchers.

If the dielectric is constant everywhere but at the molecular interface, the boundary element method (BEM) can be used to solve the Poisson equation.[19-22] The BEM is based on finding a charge distribution on the dielectric boundary that accounts for the difference in dielectric across the boundary. It can be shown that this is equivalent to solving the Poisson equation directly for the potential.[23] Because the charge density is confined to the boundary while the potential is defined throughout space, the BEM can potentially be faster and require much less storage than finite difference or finite element methods because of the reduction in dimensionality. No grid is required; instead the boundary is tesselated into surface triangles, each representing a small part of the boundary.

The induced surface charge at any surface element depends linearly on the electrostatic field generated by the atomic charges and all other surface charges, resulting in a system of linear equations described by a dense matrix with $n \times n$ elements, where n is the number of surface elements. This system of equations can be solved by iterative techniques.[20-22,24] However, because the matrix elements have to be calculated and then either stored (memory intensive) or recalculated (calculationally intensive) for subsequent iteration steps, computational requirements have been prohibitive for any triangulation with more than 4000–6000 surface elements.[24,25]

The dense matrix used in the linear equations for the induced surface charges describes the Coulombic interactions between the charges. Several different methods exist for evaluating the Coulomb interactions between a large number of charges approximately and much faster than by conventional pairwise enumeration. The equation for the induced surface charges can then be solved self-consistently with the same set of iterative methods that would be used to solve the linear equations.

Appel[26] and Barnes and Hut[27] have developed tree codes for evaluating the Coulombic interac-

tion between charged particles. These programs exploit the fact that a particle interacts with a distant group of particles much as if it were interacting with a single particle at the center of mass of the distant group. Tree codes embed the computational region into a hierarchical tree structure,[28] which enables a systematic procedure to determine which particles are distant from each other. This algorithm reduces the complexity of the computation from $O(n^2)$ to $O(n \log n)$, although there is a loss in accuracy in the computation. Accuracy can be traded for compute time via a free parameter.

Greengard and Rokhlin[29,30] built on the tree code idea. They form the (infinite) multipole expansions for boxes on the lowest level of the tree and carefully combine and shift these expansions as they are passed up and down the tree. Their algorithm reduces the complexity of the n-particle problem to $O(n)$, and the errors introduced by truncating the infinite multipole expansions to a finite number of terms are well defined; arbitrary accuracy can be assured *a priori* by retaining an appropriate number of terms in the expansions. An efficient implementation was developed by Leathrum and Board,[31,32] called the Parallel Fast Multipole Algorithm (PFMA). Its efficiency derives mostly from precomputation of coefficients and from the use of spherical harmonics, and it was designed to run on parallel machines. A similar algorithm, the Cell Multipole Method (CMM), was developed independently by Ding and Goddard.[33] Speedups similar to the FMA have been observed with the CMM and tested for systems with up to a million atoms. However, its accuracy is restricted to the octupole level due to use of the more cumbersome Cartesian multipoles instead of spherical harmonics. Also, no parallel implementation of the CMM has been reported.

Hence, our work has been to utilize the efficiency of the fast multipole algorithm (FMA) to produce a method wherein the total computational work within the BEM approach is proportional to the number of surface elements. We will show that our implementation has accuracy equivalent to or better than finite difference solutions, in which the number of grid points associated with the dielectric surface is comparable with the number of surface vertices in the BEM. Furthermore, we will demonstrate (though not prove) that the number of iterations required to converge to an acceptable solution is independent of the number of vertices. Hence our method appears to have computational complexity of order lambda squared, where lambda is the grid spacing of an equivalent finite difference solution. (Complexity here is defined as how the computational expense scales with a given parameter, in this case the resolution.) This compares to the complexity of lambda to the fourth for current successive overrelaxation methods, and even lambda to the third for more advanced multigrid methods. In fact, the complexity can be considered essentially optimal for this type of problem. Moreover, because we only need to store surface quantities, the complexity of the memory requirement is also superior to grid, or volumetric, based methods.

The structure of the article is as follows. In the Methods section we first describe the separate components of our fast multipole boundary element (FMBE) method, the boundary element method, and the fast multipole algorithm. The Methods section describes the procedures used to combine the separate components into the FMBE method used. In the Results section, we present accuracy tests and comparisons with finite difference calculations. Comparisons are first made on a simple sphere for which an exact analytical solution of the Poisson-Boltzmann equation is available. Both the boundary element and the finite difference methods are compared with the analytic solution. This comparison is extended to include a diatomic molecule and smaller polar molecules. Next, we present comparisons on bovine pancreatic trypsin inhibitor (BPTI), comparing the FMBE with finite difference calculations at equivalent surface resolutions, and extend the FMBE to very high resolutions to compute potentials and solvation energies at high accuracy. Finally, the limitations and implications of the FMBE are discussed.

Methods

The basic model used to describe the molecule and the solvent has been discussed extensively by Gilson et al.[34] The molecule is considered to be a low dielectric cavity containing fixed-point charges at its atomic centers and is embedded in a high-dielectric solvent. To account for electronic polarization, the molecule interior is assigned a dielectric of 2, whereas the solvent water is represented as a medium of dielectric 80.

THE BOUNDARY ELEMENT METHOD

The electrostatic potential in and around a molecule in the presence of solvent is described in

the continuum model by the Poisson equation:

$$\nabla(\epsilon(\mathbf{x})\nabla\phi(\mathbf{x})) = -4\pi\rho(\mathbf{x}) \tag{2}$$

where $\phi(\mathbf{x})$ is the potential, $\epsilon(\mathbf{x})$ is the dielectric constant, and $\rho(\mathbf{x})$ is the fixed charge density. Application of the chain rule and rearrangement of terms lead to

$$\begin{aligned}\nabla^2\phi(\mathbf{x}) &= -4\pi\left[\frac{\rho(\mathbf{x})}{\epsilon(\mathbf{x})} + \frac{\nabla\epsilon(\mathbf{x})}{4\pi\epsilon(\mathbf{x})}\nabla\phi(\mathbf{x})\right] \\ &= -4\pi\rho_{\text{eff}}(\mathbf{x})\end{aligned} \tag{3}$$

where we have defined ρ_{eff} to be the quantity in square brackets. The dependence of ρ_{eff} on ϕ is disregarded. The equation can therefore be solved trivially using the Greens function for the Poisson equation:

$$\begin{aligned}\phi(\mathbf{x}) &= \int d^3x' \frac{\rho_{\text{eff}}(\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|} \\ &= \int d^3x' \frac{\rho(\mathbf{x}')}{\epsilon(\mathbf{x}')|\mathbf{x}-\mathbf{x}'|} \\ &\quad + \int d^3x' \frac{\nabla\epsilon(\mathbf{x}')}{4\pi\epsilon(\mathbf{x}')}\frac{\nabla\phi(\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|}\end{aligned} \tag{4}$$

In a model with N_q discrete point charges q_i at positions $\mathbf{x}_i$ and a piecewise constant dielectric, the first term turns into a simple sum of Coulomb potentials. Considering that $\nabla\epsilon(\mathbf{x})$ is zero everywhere but on the surface and applying Gauss's law, the second term can be rewritten as a surface integral along the molecular surface $\partial\Omega$, which divides the area of low dielectric ϵ_{in} from the solvent area with dielectric constant ϵ_{out}. This yields

$$\begin{aligned}\phi(\mathbf{x}) &= \frac{1}{\epsilon_{\text{in}}}\sum_{i=1}^{N_q}\frac{q_i}{|\mathbf{x}-\mathbf{x}_i|} \\ &\quad + \frac{\epsilon_{\text{in}}-\epsilon_{\text{out}}}{\epsilon_{\text{in}}+\epsilon_{\text{out}}}\int_{\partial\Omega} d\mathbf{a}'\cdot\frac{\nabla\phi(\mathbf{x}')}{2\pi|\mathbf{x}-\mathbf{x}'|}\end{aligned} \tag{5}$$

where $d\mathbf{a}'$ is the surface element at $\mathbf{x}'$ and is directed perpendicular to the surface. The point charges q_i are assumed to be inside Ω, thus the occurrence of ϵ_{in} in the first term. If the surface is composed of N_s small but finite surface elements at positions $\mathbf{x}_j$ with areas a_j and normal $\mathbf{n}_j$, the second term can be approximated by a sum

$$\phi(\mathbf{x}) = \frac{1}{\epsilon_{\text{in}}}\sum_{i=1}^{N_q}\frac{q_i}{|\mathbf{x}-\mathbf{x}_i|} + \sum_{j=1}^{N_s}\frac{\sigma_j}{|\mathbf{x}-\mathbf{x}_j|} \tag{6}$$

with surface charges σ_j defined as

$$\sigma_j = \frac{1}{2\pi}\frac{\epsilon_{\text{in}}-\epsilon_{\text{out}}}{\epsilon_{\text{in}}+\epsilon_{\text{out}}}a_j\mathbf{n}_j\cdot\mathbf{E}_j \tag{7}$$

in terms of the electric field at the surface $\mathbf{E}_j = \nabla\phi(\mathbf{x}_j)$. The electric field computed at the surface is the field at each element produced by all charges, fixed and induced, outside that element. Taking the derivative of eq. (6) and substituting it in eq. (7) yields a system of linear equations for the surface charges

$$\begin{aligned}\sigma_j = \frac{1}{2\pi}\frac{\epsilon_{\text{in}}-\epsilon_{\text{out}}}{\epsilon_{\text{in}}+\epsilon_{\text{out}}}a_j\Bigg[&\sum_{k=1,k\neq j}^{N_s}\frac{\sigma_k(\mathbf{x}_j-\mathbf{x}_k)}{|\mathbf{x}_j-\mathbf{x}_k|^3} \\ &+ \frac{1}{\epsilon_{\text{in}}}\sum_{i=1}^{N_q}\frac{q_i(\mathbf{x}_j-\mathbf{x}_i)}{|\mathbf{x}_j-\mathbf{x}_i'|^3}\Bigg]\cdot\mathbf{n}_j\end{aligned} \tag{8}$$

Equation (8) is a self-consistent set of equations for the σ_j and has traditionally been cast as a set of linear equations

$$\sum_{ij}^{N_s} A_{ij}\sigma_j = b_i \tag{9}$$

with the matrix

$$A_{ij} = 2\pi\frac{\epsilon_{\text{in}}+\epsilon_{\text{out}}}{\epsilon_{\text{in}}-\epsilon_{\text{out}}} - a_i\frac{(\mathbf{x}_i-\mathbf{x}_j)\cdot\mathbf{n}_i}{|\mathbf{x}_i-\mathbf{x}_j|^3} \tag{10}$$

and the vector

$$b_i = \frac{a_i}{\epsilon_{\text{in}}}\sum_{k=1}^{N_q} q_k\frac{(\mathbf{x}_i-\mathbf{x}_k)\cdot\mathbf{n}_i}{|\mathbf{x}_i-\mathbf{x}_k|^3} \tag{11}$$

The problem with solving the linear equation [eq. (9)] via an iterative, resubstitution method, as is common in finite difference approaches, is that the matrix A_{ij} is dense (i.e., contains a very large number of elements). Solving it would require either calculating all the matrix elements once and storing it for reuse at each iteration, a process which soon requires excessive memory because the number of elements goes as $O(N_s^2)$; or the A_{ij} could be calculated on the fly during each iteration saving memory, but at a cost in computer time that scales as an additional $O(N_s^2)$ operations

(square roots) per iteration. Similarly, direct methods (i.e., noniterative) suffer from the same problem of storage and require $O(N_s^6)$ operations (although once solved, the resulting inverted matrix does allow instant solutions for arbitrary charge distributions.) This has limited a broader application of the boundary element method.

Instead of casting eq. (8) as a linear matrix equation, it can be more generally written as a self-consistent set of equations

$$\sigma_i = U_i(\sigma_j) \tag{12}$$

where the functions $U_i(\sigma_j)$ happen to be linear. The essential observation is that from eq. (7) we see that calculating $U_i(\sigma_j)$ amounts to calculating the electric field **E** generated by the surface charges as well as the fixed charges at the positions of the surface charges. Thus, calculating $U_i(\sigma_j)$ is a simple Coulomb force calculation, for which the fast multipole algorithm can be used, changing the calculation time to order $O(N_s)$.

THE FAST MULTIPOLE ALGORITHM

The FMA is a numerically stable method for evaluating Coulombic interactions between large numbers of particles in a time that increases linearly rather than as the square of the number of particles. The FMA is based on the expansion of the Coulomb potential of a bounded charge distribution in multipoles

$$\Phi(\mathbf{x}) = 4\pi \sum_{l,m} \frac{M_{lm} Y_{lm}(\theta, \phi)}{(2l+1) r^{l+1}} \tag{13}$$

where (r, θ, ϕ) are the spherical coordinates associated with the Cartesian coordinate vector $\mathbf{x}$, and Y_{lm} are the complex valued spherical harmonics functions. This expansion is exact and valid everywhere outside the smallest sphere enclosing all charges. For a collection of n point charges q_i at positions $\mathbf{x}_i$, the coefficients M_{lm} are given by

$$M_{lm} = \sum_{i=1}^{n} q_i Y_{lm}^*(\theta_i, \phi_i) r_i^l \tag{14}$$

where Y_{lm}^* denotes the complex conjugate of the spherical harmonics functions. In addition to the multipole expansion of eq. (13), the FMA also uses the local expansion

$$\Phi(\mathbf{x}) = 4\pi \sum_{l,m} L_{lm} Y_{lm}(\theta, \phi) r^l \tag{15}$$

which is equivalent to a Taylor expansion and valid within the largest sphere around the origin that does not contain any of the charges contributing to it.

The expansions of eqs. (13) and (15) are exact (i.e., within their regions of validity they provide the exact value of the potential). However, they also consist of an infinite number of terms. The approximation in the FMA consists of truncating the expansions after a certain number of terms with the truncation limit p being the largest value of l included in the sums. It has been shown that the same level of truncation is appropriate for both the multipole and the local expansions. The essential idea of the FMA is to build a hierarchy of multipoles, each containing the contribution of a subset of charges of limited extent. For efficiency, cubic boxes are used in the FMA. The smallest boxes on the lowest level contain only a small number of charges (10–20), and the boxes on successively higher levels are the union of eight lower level boxes, until one single box on the highest level contains all other boxes and therefore all charges. The number of levels L is chosen to provide the optimal tradeoff between multipole and direct interactions and depends on the number of charges in the system as $N \sim 8^L$.

The multipoles associated with the lowest level boxes are calculated according to eq. (14). Multipoles of higher level boxes are calculated not from the charges but from the multipoles of the lower levels. To obtain the multipole expansion for one box, the eight multipole expansions of its subboxes are shifted to a common origin and added up. Shifting the origin of a multipole expansion involves finding the coefficients of $\Phi'(\mathbf{x})$ such that

$$\Phi'(\mathbf{x}) = \Phi(\mathbf{x} + \mathbf{x}_t) \tag{16}$$

The relationship between the old and new coefficients is linear. That is,

$$M'_{lm} = \sum_{l',m'} T^{MM}_{lm,l'm'} M_{l'm'} \tag{17}$$

and the translation matrix is given by

$$T^{MM}_{lm,l'm'} = 4\pi \frac{(2l+1) a_{l'm'} a_{l-l',m-m'}}{(2l'+1)(2l-2l'+1) a_{lm}} \times (-r_t)^{l-1} Y^*_{l-l',m-m'}(\theta_t, \phi_t) \tag{18}$$

with the auxiliary numbers a_{lm} defined as

$$a_{lm} = (-1)^{l+m}\sqrt{\frac{2l+1}{4\pi(l+m)!(l-m)!}} \quad (19)$$

Because the coefficients $T^{MM}_{lm,l'm'}$ depend only on the shift vector $\mathbf{x}_t$, and because in the regular cubic arrangement of multipoles the same shift vectors occur many times, the coefficients can be precomputed for the L levels and eight possible directions and then reused efficiently during the calculations.

After the multipoles $\Phi(\mathbf{x})$ of all boxes have been calculated, a local expansion $\Psi(\mathbf{x})$ is constructed for each box, describing the potential inside the box caused by all distant charges (i.e., all charges except those in the same or neighboring boxes). An essential element of the algorithm is the recursive use of the local expansion from the next higher level, that is,

$$\Psi_l(\mathbf{x}) = \Psi'_l(\mathbf{x}) + \Psi_{l-1}(\mathbf{x}) \quad (20)$$

where $\Psi'_l(\mathbf{x})$ is the local expansion of all multipoles not already contained in $\Psi_{l-1}(\mathbf{x})$. The number of those multipoles is never larger than $6^3 - 3^3 = 189$; thus the number of operations for this step is independent of the size of the system for each box. Because the number of boxes depends linearly on the number of charges, this step, like all others, is scalable. The coefficients of the local expansion of a given multipole expansion are given by

$$L'_{lm} = \sum_{l',m'} T^{LM}_{lm,l'm'} M_{l'm'} \quad (21)$$

with the transformation matrix

$$T^{LM}_{lm,l'm'} = 4\pi \frac{a_{l'm'}a_{lm}}{(2l'+1)(2l+1)a_{l+l',m-m'}} \times \frac{(-1)^{l'+m'}}{r_t^{l'+l+1}} Y^*_{l+l',m-m'}(\theta_t, \phi_t) \quad (22)$$

and they can be precomputed in the same way as $T^{MM}_{lm,l'm'}$. Most of the time spent evaluating the multipole interactions is in this transformation. The translation of $\Psi_{l-1}(\mathbf{x})$ to the origin of $\Psi_l(\mathbf{x})$ is done similar to the shifting of multipoles. That is,

$$L'_{lm} = \sum_{l',m'} T^{LL}_{lm,l'm'} L_{l'm'} \quad (23)$$

with the transformation matrix

$$T^{LL}_{lm,l'm'} = 4\pi \frac{a_{lm}a_{l'-l,m'-m}}{(2l+1)(2l'-2l+1)a_{l'm'}} \times (-r_t)^{l'-l} Y_{l'-l,m'-m}(\theta_t, \phi_t) \quad (24)$$

Once the local expansions at the lowest level $\Psi_L(\mathbf{x})$ are known, the electrostatic potential at position $\mathbf{x}_i$ due to all other charges can be calculated as

$$U(\mathbf{x}_i) = \Psi_{L,i}(\mathbf{x}_i) + \sum_j \frac{q_j}{|\mathbf{x}_i - \mathbf{x}_j|} \quad (25)$$

where $\Psi_{L,i}$ is the local expansion for the box that contains charge $\mathbf{x}_i$. The sum in the second term is restricted to charges within that box or within adjacent boxes.

The code used for the incorporation of the FMA in this study is based on an efficient implementation by Leathrum and Board[31,32] and was further improved by us to avoid the waste of storage space and calculation time for empty boxes. This improvement was essential because the spatial distribution of surface charges is intrinsically and strongly inhomogeneous.

ITERATIVE SOLUTION OF THE BOUNDARY ELEMENT EQUATION

When the boundary element equation, [eq. (12)] is cast in a matrix form, as in eq. (9), the resultant matrix is nonsymmetric and is generally dense. Therefore, the usual fast methods that can be used to solve finite difference matrices,[15] which are sparse and symmetric, do not apply here. The method of successive overrelaxation (SOR) has been applied previously with success in solving such unsymmetric matrices.[35] In the SOR method, given eq. (12) and a starting value of σ, which we denote as $\sigma^{(1)}$, we obtain $\sigma^{(2)}$, the next better approximation to σ from the equation

$$\sigma_i^{(n+1)} = (1-\omega)\sigma_i^{(n)} + \omega U_i\left(\sigma_j^{(n)}\right) \quad (26)$$

Here ω is a parameter, referred to as the relaxation factor. When $\omega = 1$, we get the Jacobi method. Equation (26) is repeated to get values of $\sigma^{(3)}$, $\sigma^{(4)}$, and so on until the chosen criteria for convergence are satisfied. An excellent overview of these methods can be found in Young et al.[35] We have developed and used an adaptive SOR procedure in our implementation of the BEM. As explained later, adaptive SOR proceeds by improving the value used for the relaxation factor each successive iteration.

The boundary element method has been coded in C and has been optimized to minimize computer memory usage via efficient utilization of dynamic memory allocation. The most time-intensive step in the solution is the calculation of the electrostatic field at each element due to induced surface charges present at all the other elements at each iteration, which is done using the FMA. Comparison with the time taken by direct calculation indicates that usage of the FMA leads to an acceleration by a factor of 30 for a molecular surface comprised of roughly 40,000 surface elements.

The speedup that can be attained by using the fast multipole algorithm depends largely on two factors: the truncation limit used in its multipole expansions and the number of surface charges present in each cube. For a given truncation limit, an optimum number of surface charges need to be present in each cube to yield the maximum speedup in computation.[29] For a homogeneous charge distribution, the number of particles per box depends on the number of boxes n_b that the FMA partitions in the space containing the charges. This number is related to the level by the relation

$$n_b = 8^{\text{level}}$$

The optimal level l_{opt} to be used in a FMA computation is then related to the number of charges N in the computation by the formula

$$l_{\text{opt}} = \log_8\left(\frac{N}{n_{\text{opt}}}\right) \tag{27}$$

where n_{opt} is the optimal number of particles to be present in each box. However, surface charges calculated from the BEM are not distributed homogeneously. Application of the FMA to such a system leads to a situation in which charges localize in only a few cubes instead of being distributed uniformly in all the cubes. Hence eq. (27) cannot be applied to compute the optimal level. We have resorted to a trial-and-error method for obtaining the optimal level to be used in the FMA invocations in a BEM computation.

ADAPTIVE SOR PROCEDURE FOR SOLUTION OF THE BOUNDARY ELEMENT METHOD MATRIX

We have developed an adaptive procedure that finds the best relaxation parameter based on results from previous iterations performed using a relaxation parameter of 1.0. The induced charges calculated for each iteration are stored in memory. Using them, it is possible to calculate the induced surface charge at the end of the nth iteration for any arbitrary relaxation factor ω from

$$\sigma^{(n)'} = \sum_{i=0}^{n} \binom{i}{n} \omega^i (1-\omega)^{n-i} \sigma^{(i)} \tag{28}$$

where $\sigma^{(n)}$ and $\sigma^{(n)'}$ are the induced charges at a surface element calculated using relaxation factors of 1 and ω, respectively. Proof is by induction from the three-term recurrence relation inherent in eq. (26). Using eq. (28) and the total induced surface charges calculated at each iteration using a relaxation factor of 1, one may calculate the induced surface charges using any relaxation factor, ω, in a trivial amount of computer time. In our adaptive SOR procedure, we store the induced surface charges calculated from each iteration. For each iteration after the third, we use a range of relaxation factors ranging from 0.5 to 1.0 to calculate the induced surface charges. (Note that we are actually underrelaxing, not overrelaxing. If the relaxation parameter is set to 1.0 or greater—i.e., overrelaxing—the procedure invariably does not converge). For each relaxation factor, the appropriate convergence test is then used on the calculated surface charges, and an interval containing the relaxation factor that yields the best value for the convergence parameter used is selected. The foregoing procedure is then repeated using relaxation factors at regularly spaced intervals within this interval to yield yet a smaller interval for the optimal relaxation factor. This may be repeated several times before proceeding with the next iteration or stopping depending on whether the convergence criterion is fulfilled. Note that although this method in general requires the storage of each surface charge moment, for the choice of certain convergence criteria, such as the total surface charge or the resultant solvation energy, only the sum of surface charges within each moment is required, eliminating any memory burden.

CALCULATION OF POTENTIAL AND SOLVATION ENERGY

Electrostatic potentials were calculated at various positions within molecules using both the finite difference and the boundary element methods. Using the formula

$$V = \sum_{k=1}^{n_e} \frac{\sigma_k}{|\mathbf{x} - \mathbf{x}_k|} + \sum_{i=1}^{N_q} \frac{q_i}{\epsilon_{\text{in}} |\mathbf{x} - \mathbf{x}_i|}$$

the electrostatic potential at any point x is calculated from the fixed charges and values calculated for the induced surface charges from the boundary element method. The solvation energy SE of the system was calculated using the relation[21]

$$SE = \frac{1}{2}\sum_{i=1}^{N_s}\sum_{j=1}^{N_q}\frac{q_j\sigma_i}{\epsilon_{in}|\mathbf{x}_i - \mathbf{x}_j|} \quad (29)$$

The analytical solvation energies for a point charge placed in a low-dielectric sphere surrounded by high-dielectric solvent were computed using the formula

$$SE = \frac{q^2}{2\epsilon_{in}}\left[\frac{1}{r} + \sum_{n=0}^{\infty}\left[\frac{r^{2n}}{b^{2n+1}}\left[\frac{(n+1)(\epsilon_{in} - \epsilon_{out})}{(n+1)\epsilon_{out} + n\epsilon_{in}}\right]\right]\right]$$

as derived by Hill,[34,36] where ϵ_{in} and ϵ_{out} represent the internal and external dielectric constants, respectively, and q is the point charge at a distance r from the center of a sphere of radius b.

If we know the solvation energy at the end of each iteration performed with a relaxation parameter of 1, we can calculate the solvation energy for the iterations performed with any chosen relaxation parameter ω using the relation

$$SE^{(n)'} = \sum_{i=0}^{n}\binom{i}{n}\omega^i(1-\omega)^{n-i}SE^{(i)} \quad (30)$$

where $SE^{(n)'}$ is the solvation energy at the end of the nth iteration using a relaxation parameter ω, and $SE^{(i)}$ refers to the solvation energy at the end of the ith iteration performed using a relaxation parameter of 1. The preceding relationship maybe proven by substituting eq. (28) in eq. (29).

CONVERGENCE CRITERIA

Several criteria for evaluating attainment of convergence were studied, specifically the absolute change in solvation energy, the root mean square change in induced surface charge, and the root mean square relative change in induced surface charge. Absolute change in solvation energy, ΔS, was calculated as

$$\Delta SE = |SE^{(n)} - SE^{(n-1)}|$$

where $SE^{(n)}$ is the solvation energy calculated using the induced surface charges computed from the nth iteration. The root mean square change in induced surface charge, $\Delta\sigma$, is similarly calculated as

$$\Delta\sigma = \sqrt{\frac{\sum_{i=1}^{n_e}(\sigma_i^n - \sigma_i^{n-1})^2}{n_e}}$$

The root mean square relative change in induced surface charge, $\Delta\sigma_{rel}$; is calculated using the formula

$$\Delta\sigma_{rel} = \sqrt{\frac{\sum_{i=1}^{n_e}\left(\frac{\sigma_i^n - \sigma_i^{n-1}}{\sigma_i^n}\right)^2}{n_e}}$$

Except for $\Delta\sigma_{rel}$, none of the parameters were observed to decrease linearly or predictably as the iterations progressed. Hence for all computations, convergence was considered attained if for three consecutive iterations, the calculated root mean square relative change in surface charge kept reducing in magnitude and was less than a minimum value. This minimum value was an external input to the method. Further analysis of the solvation energy calculated after convergence with different values for this minimum indicated that a value of 10^{-5} or lower for this parameter yielded solvation energies that did not alter by more than 0.1 kT from each other. Hence the convergence parameter in all calculations was set at 10^{-5}.

SURFACE CHARGE NORMALIZATION

Application of the continuity rule and Gauss's law and integration over the entire molecular surface[21] offers the following equation relating the total fixed charge in a molecule to the total induced charge on the molecular surface:

$$\sum_{i=1}^{n_e}\sigma_i a_i = \frac{\epsilon_{in} - \epsilon_{out}}{\epsilon_{in}\cdot\epsilon_{out}}\sum_{i=1}^{n_e} q_i$$

As described previously by Rashin,[21] the net surface charge obtained after convergence can be uniformly scaled up or down to normalize to the theoretically expected surface charge. However, the foregoing idea is not applicable to an electrically neutral molecule, because the theoretically expected surface charge for such a system is zero. For such a molecule with both positive and negative charges, we use a variant based on the same idea. The boundary element calculation is done

twice, the first time with only positive charges in the molecule and the second time with only negative charges in the molecule. The induced surface charges obtained from each calculation are normalized to the theoretical amount calculated for the molecule with only positive or negative charges. The induced surface charge on each element is then obtained by summing normalized induced surface charges obtained from both these calculations.

CREATION OF ANALYTICAL SURFACES FOR A SPHERE

An icosahedron approximates a sphere. Its 12 vertices lie on the sphere and comprise 20 triangles. Each of these triangles may be subdivided into four triangles by connecting the midpoints of its sides. These midpoints can then be repositioned to lie on the surface of the sphere. This process yields a surface with four times as many triangles. This procedure was iteratively repeated to yield molecular surfaces for a sphere with 1280, 5120, 20,480, 81,920, and 327,680 triangles, respectively. The number of vertices, n_v, is related to the number of triangles, n_t, by the Euler relation

$$n_v = \frac{n_t}{2} + 2$$

CREATION OF THE MOLECULAR SURFACE

A grid-based molecular surfacing program developed by Sridharan et al.[37] has been used to generate molecular surfaces for our study. The standard Richards definition of molecular surface[12] is used here. The surface is rendered on a cubic grid, and the marching cubes algorithm[38] is applied to triangulate it. Finally, the vertices obtained through the triangulation are scaled onto the molecular surface using an approximate but efficient process. The grid representation of this surface was also used as input for finite difference computations done for comparison.

The input to this program consists of a file in the Brookhaven Protein Databank format containing the x, y, z coordinates of the atoms comprising the molecule along with a size file containing the radii of the various atoms and the required grid spacing to use for surfacing. Its output is the positional coordinates of a set of vertices, along with the x, y, z components of the surface normals at these vertices and the connectivity between the vertices to form a triangulated surface. Each vertex was assigned an area equal to one third of the sum of areas of all the triangles to which it belongs. Each vertex along with its associated area and normal was then considered as an element for the boundary element method.

CALCULATION OF FINITE DIFFERENCE RESULTS

A modified version of the program DelPhi,[14] which uses the same surfacing algorithm as used for the boundary element method, has been used for finite difference calculations. In the calculations by Gilson et al.,[14] the potential map (i.e., the potential values at all grid points in the finite difference method) was used to interpolate linearly the potentials to any site of interest. However, one may use the potential map to calculate induced surface charges from a finite difference potential map.[39] The induced surface charges so calculated lie on the discrete finite difference grid. If these induced surface charges are repositioned to lie on the true molecular surface and then solvation energies are calculated as with eq. (29), this method reproduces analytical solvation energies much more accurately.[39] The scaled (i.e., repositioned) surface charges, along with the fixed real charges, have also been used to calculate potentials at any point using the same formula as in the boundary element method. Potentials calculated using this method are, in general, more accurate than those obtained by interpolating from the potential map.

COMPARISON OF THE FINITE DIFFERENCE AND BOUNDARY ELEMENT METHODS

To facilitate comparison of the finite difference and boundary element methods, we have used identical conditions as far as possible. The dielectric maps used by the finite difference method and by the surfacing program for the boundary element method were identical. This implies that the grids used for surfacing and for generating the dielectric map are identically shaped and oriented for any pair of comparable results from both methods. Accordingly, comparisons of both methods usually plot the grid size (in grids/angstrom) along the horizontal axis. For the finite difference method, this refers to the scale used for calculation of the dielectric map used by the computation, whereas for the boundary element method this refers to the grid scale used for molecular surfacing.

Results

This section is divided into three parts. We first establish the parameters to use with the boundary element method in order to compute solvation energies optimally without losing accuracy. A comparison of solvation energies calculated from the finite difference and boundary element methods follows. Finally, we compare potentials calculated using both methods with analytical potentials for a test case of a point charge in a sphere, in order to probe the shortcomings and strengths of each method.

Errors that arise while using the boundary element method with the fast multipole algorithm may be partitioned into two sources: (1) inaccuracies intrinsic to the boundary element method and (2) inaccuracies stemming from approximations used in the fast multipole algorithm, in particular the truncation limit used in its multipole expansions. Obtaining a good result for the solvation energy of a system involves minimizing errors from both these steps. To measure the errors arising from the boundary element method alone, it was decided to do calculations using the boundary element method without any fast multipole implementation, hereinafter referred to as the direct BEM, on a test case for which an analytical solution is available. A point charge placed eccentrically in a low-dielectric sphere embedded in a high-dielectric solvent was used. Solvation energy calculations using the direct BEM were done on such a system. The radius of the sphere was taken to be 1.6 Å, and a point charge of unit magnitude was placed within the sphere at varying distances from its center. The internal dielectric was set to be 2 and the external dielectric 80. A triangulated spherical surface comprised of 2562 vertices and containing 5120 triangles was used for the direct BEM computations. It was observed that net induced surface charge obtained from the direct BEM did not equal the theoretically expected value. Consequently, both normalized and unnormalized surface charge distributions were used to calculate normalized and unnormalized solvation energies for the system and the relative percent errors computed in each case.

Figure 1 shows the relative percent error plotted against the distance of the point charge from the center of the sphere. Normalizing the surface charge yields better accuracy of the solvation energies calculated by the direct BEM. We have also

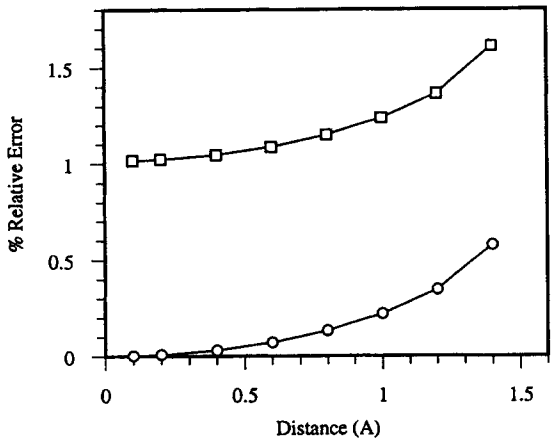

FIGURE 1. Graphs of percent relative error in electrostatic solvation energy calculated for a sphere of 1.6 Å radius with a dielectric of 2 embedded in a solvent of dielectric 80 and containing a unit point charge eccentrically placed. The *x*-axis refers to the distance of the unit charge from the center of the sphere, whereas the *y*-axis refers to the percent relative error in unnormalized and normalized electrostatic solvation energies calculated using the direct boundary element method, compared to the analytical result obtained using the formulation of Gilson et al.[34] A molecular surface comprised of 2562 vertices and 5120 triangles was used for the calculation. In the graph, squares and circles represent percent relative errors calculated using unnormalized and normalized solvation energies, respectively.

observed that surface charge normalization, when used in direct BEM calculations on arbitrarily shaped molecules, yields solvation energies that show much better agreement with results from other methods such as the finite difference method. Consequently, it was decided to use normalized solvation energies for all our boundary element calculations.

Solvation energy calculations using the boundary element method incorporating the fast multipole algorithm (FMBE) were done on the same system. To study the dependence of accuracy on the parameters of the fast multipole algorithm, the numbers of terms used in its multipole expansions were varied and the calculated solvation energies were compared with those obtained from the direct BEM implementation.

The percent relative differences in solvation energies obtained using the FMBE method, considering the direct BEM solvation energy as the standard, were calculated and plotted against the distance of the point charge from the center of the sphere. The aforementioned calculations were re-

peated using different numbers of terms in the multipole expansions while using the fast multipole algorithm. The percent relative differences between the direct BEM implementation and the FMBE method become negligibly small as we use three or more terms in the multipole expansions. As the point charge comes closer to the molecular surface, the differences increase in magnitude. Graphs for similar calculations done using surface approximations with differing numbers of triangles corroborate these observations. Consequently, we have used four terms in the multipole expansions in all our calculations using the FMBE method.

Comparison of the time taken per iteration for the direct method versus the four term FMA method is shown in Figure 2. It is clear that the saving in computer time is substantial once the number of vertices is greater than 10,000. When 40,000 vertices are used, the ratio of direct to FMA enhanced is about 30-fold. As discussed later, such vertex counts are not unusual in the study of even small proteins, such as bovine pancreatic trypsin inhibitor, used for this and later calculations.

To study the dependence of the surface charge scaled solvation energy on the resolution of the molecular surface used, surface charge scaled solvation energies for molecular surfaces of different resolution were computed for the system used in Figure 1 using the FMBE method. The calculations were done using four terms in the multipole expansions. The relative percent error with respect to the analytical solvation energy was calculated. Figure 3 plots this error as a function of the distance of the charge from the center of the sphere for surfaces comprised of 642, 2562, 10,242, and 40,962 vertices, respectively. As expected, better resolved surfaces yield more accurate solvation energies. Loss in accuracy becomes significant as the charge moves closer to the molecular surface.

We compared potentials calculated from the boundary element and finite difference methods with analytical potentials for a test case of a charge in a sphere. As discussed previously, experimentally measured pK_a's are accurate up to 0.1 units, corresponding to an accuracy of 0.2 kT potential. This accuracy is attainable in potential calculations on small molecules using the finite difference method. However, this accuracy has been difficult to achieve in similar calculations on much larger systems, such as biomolecules. This has been mainly due to excessive computer memory requirements of the large grids required.

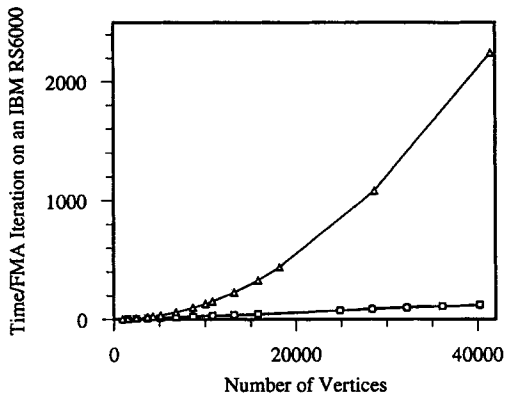

FIGURE 2. Graph showing time taken in seconds per boundary element iteration with and without usage of the Fast Multipole Algorithm versus the number of surface elements on the molecular surface. Squares and triangles refer to iteration times with and without usage of the Fast Multipole Algorithm, respectively. Multipole expansions beyond the first four terms were neglected for the Fast Multipole Algorithm computations. As evident, the FMA iteration time scales linearly and provides a considerable speedup to the boundary element method.

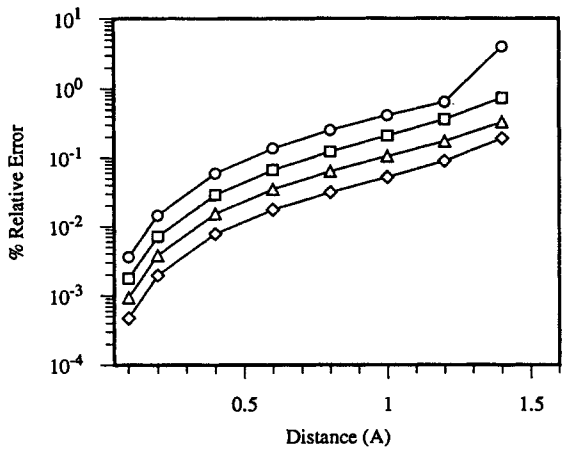

FIGURE 3. Graphs of percent relative error in electrostatic solvation energy calculated for a sphere of 1.6-Å radius with a dielectric of 2 embedded in a solvent of dielectric 80 and containing a unit point charge eccentrically placed. The *x*-axis refers to the distance of the unit charge from the center of the sphere, whereas the *y*-axis refers to the percent relative error in calculated electrostatic solvation energy when compared to the analytical result. Four levels of resolution were used for the molecular surface. In the graph, circles, squares, triangles, and diamonds represent results using surfaces with 642, 2562, 10,242, and 40,962 vertices, respectively.

A moderately sized protein may be approximated roughly by a sphere of 30 Å radius. Typically one is interested in knowing the potential 2–3 Å away from a charge in a sidechain in the protein, itself situated 1–2 Å beneath the molecular surface. The true potentials at any point in the sphere due to a charge placed anywhere within it can be calculated from analytical formulas.[34,36] The typical case may be illustrated by a point charge placed 1 Å from the surface of the sphere. We have compared potentials calculated by both the finite difference and the boundary element methods for such a system. Potentials were calculated using both the finite difference and boundary element methods for a point charge located at regular intervals from the center of a sphere. These values were calculated for points distributed throughout a plane containing the charge inside the sphere.

A sphere of radius 30 Å with an inner dielectric of 2 and surrounded by solvent of dielectric 80 was used for the calculation. Finite difference potentials for such a system had also been computed by Gilson et al.,[14] using an older version of DelPhi. For our comparison, finite difference potentials were calculated using two schemes. The first scheme interpolates potentials from the potential map output by the finite difference calculation, whereas the second, as described in the previous section, calculates induced surface charges from the potential map, repositions or scales them to the molecular surface, and uses these scaled surface charges for potential calculations.

Figure 4 shows graphs of maximum errors in potentials calculated using the boundary element and finite difference methods, when the charge is placed 1 Å from the spherical surface, plotted against the distance from the charge. Potentials calculated from the FMBE and finite difference methods at equivalent resolutions are compared. The FMBE computations used a surface comprised of 10,242 vertices. The finite difference results were computed at a scale of 0.52 grids/Å. This corresponds to a dielectric map with around 10,200 dielectric boundary points. Whereas finite difference potentials directly interpolated from the grid are not accurate close to the charge, finite difference potentials calculated using the scaled surface charges are seen to be more accurate. The scaled surface charge method confers an order of magnitude improvement in accuracy over calculating potentials by direct interpolation to the finite difference method. As discussed in ref. 39, this improvement in accuracy extends to the solvation energies calculated using the scaled surface charge

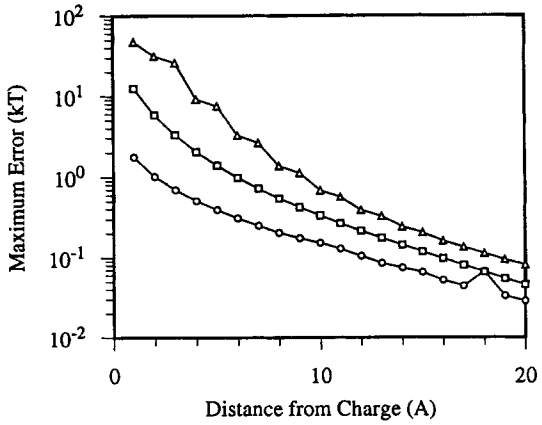

FIGURE 4. Graphs showing the maximum error in calculated potential versus distance from charge for a point charge situated 1 Å inside a 30-Å low-dielectric sphere in water. The potentials were compared for points situated uniformly inside the sphere. Circles, triangles, and squares refer to graphs calculated from potentials obtained using the boundary element method with the fast multiple algorithm, the finite difference method with potential map interpolation, and the finite difference method with scaled surface charges, respectively. The boundary element computations used a surface comprised of 10,242 vertices, whereas finite difference calculations were done at a resolution that yielded around the same number of dielectric boundary points.

method in finite difference calculations. The FMBE potentials are more accurate than the scaled surface charge finite difference potentials for the case considered. Our implementation of the FMBE assumes the induced charge on each surface element to be discretized at a point. In an alternate implementation of the boundary element method,[24] Zauhar et al. assume a smoothly varying charge density and perform an elegant integration over each surface element instead. The integration results in a smoothing out of the surface charge over elements instead of their discretization. In yet another innovative implementation, Juffer et al.[20] split the potential at each boundary element into a weakly varying or smooth and a strongly varying nonsmooth component. The strongly varying component is cast as a known function and moved into the right-hand side of the integral equation. The unknowns to be solved for are therefore the weakly varying components. Because these vary smoothly, the accuracy of their calculations is improved substantially. Incorporation of such techniques within the FMBE could further increase the accuracy of potentials calculated here.

Having estimated the parameters and methods to use with FMBE, we now compare solvation energies calculated from the finite difference and FMBE methods. The middle graph in Figure 5 compares solvation energies calculated using the FMBE and the finite difference methods for a two-atom molecule. The molecular interior and solvent were assigned dielectrics of 1 and 80, respectively. The two atoms separated by 3 Å have charges of +0.5 and −0.5 electron units and have radii of 1 Å and 2 Å, respectively. Analytical molecular surfaces of differing resolution were created using a probe radius of 1.4 Å and were used for the boundary element calculations. These surfaces were also used as a basis for creating the dielectric maps used in the finite difference calculations. The solvation energies computed are plotted against the surfacing scale used to create the molecular surfaces. There is good agreement between the solvation energies computed by the two different methods. The bottom graph in Figure 5 compares solvation energies computed by the boundary element and finite difference methods for methanol. The molecular surface used for the boundary element calculations was generated from the grid-based molecular surfacing program discussed in the Methods section. A probe radius of 1.4 Å was used. The atomic sizes and charges used are obtained from Jean-Charles et al.[40] Similar results were obtained for other small molecules within that reference.

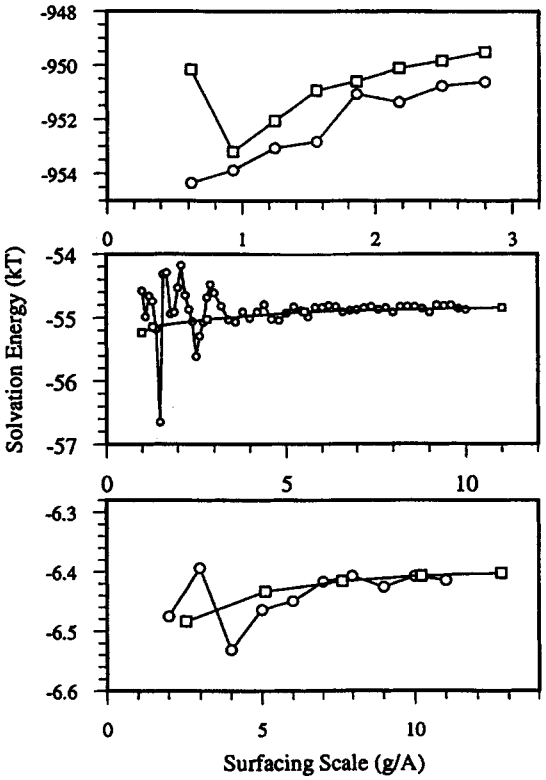

FIGURE 5. Comparison of electrostatic solvation energies obtained via the boundary element and the finite difference methods for a diatomic molecule, methanol, and BPTI, a small protein. The calculated electrostatic solvation energy (*y*-axis) is plotted against the lattice scale in grids / angstrom used either to create the boundary element description or in the finite difference calculation. Circles and squares represent results calculated using the finite difference and boundary element methods, respectively. The top, middle, and bottom graphs refer to solvation energy calculations for BPTI, a diatomic molecule, and methanol, respectively.

Whereas solvation energies calculated by the finite difference and boundary element methods were observed to agree well for small molecular systems, it remained to be determined whether the same was also true for much larger molecules, such as proteins. Previous boundary element calculations on proteins have used molecular surfaces with up to 5000 boundary elements. Whereas a surface with 5000 boundary elements is a good approximation for the dielectric interface of a small molecule, it fails to represent the molecular surface in sufficient detail for accurate calculations on even a small protein. For example, when finite difference calculations are done on a 65^3 grid for bovine pancreatic trypsin inhibitor (BPTI)—a small protein of only 451 heavy atoms—at a scale such that the protein occupies roughly 75% of the grid, there are approximately 6000 dielectric boundary points. Were such a dielectric map used in our surfacing program, it would yield around 12,000 surface vertices. More typical finite difference calculations on proteins are done with grids of size 129^3. Hence a comparison of the boundary element and finite difference methods has not been done for large molecules at equivalently high resolutions. However, using the FMA, we have performed boundary element calculations on BPTI with molecular surfaces containing around 40,000 vertices. The top graph in Figure 5 plots solvation energies for BPTI calculated using the boundary element and finite difference methods against the scale used. For the finite difference calculations, the scale corresponds to the grid spacing used in the finite difference calculations, whereas for the boundary element method, it refers to the grid spacing that was used to create the molecular surface. For the finite difference calculation and for creating the molecular surface for the boundary element method, a grid

size of 129^3 was employed. The BPTI molecule was charged at only its ionizable sidechains and at its N- and C-terminal ends. This resulted in a molecule with a charge of +5.0 comprised of positive and negative charges of +11.0 and −6.0, respectively. The molecular interior was assigned a dielectric of 2, whereas the solvent was assigned a dielectric of 80. The solvation energies calculated by the two methods differ by around 1 kT. Whereas this is experimentally significant, these errors are comparatively small given the large magnitude of the solvation energies. The solvation energies calculated by these two different methods are seen to agree to within 0.2%. Solvation energies calculated with the FMBE method are observed to show less variation with scale than those calculated with the finite difference implementation. This may be because atomic charges are discretized on a grid in the finite difference implementation. From Figure 5, it is also evident that the FMBE method overestimates solvation energy magnitudes at lower resolutions for all cases. Hence there is a progressive decrease in the solvation energy magnitude as the surface resolution increases. This may be related to our assumption of constant induced surface charge density at each surface element and the fact that we collapse all the charge induced at each surface element onto its center in our computations.

We now turn to our claim that the computer time used by the FMBE method increases as $O(n)$, where n is the number of boundary elements. Figure 6 plots the time per iteration against the number of vertices used for a boundary element calculation. The times reported are for calculations performed on an IBM RS6000 workstation. The

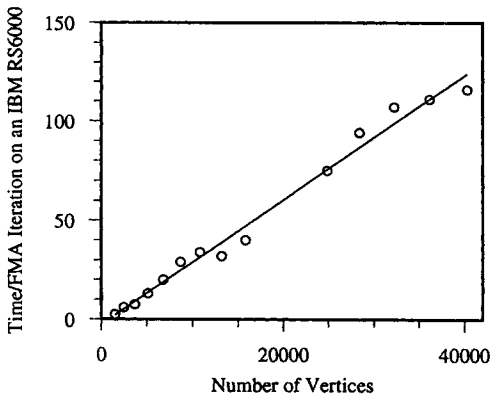

FIGURE 6. Dependence of the time taken per boundary element iteration with the fast multiple algorithm on the number of vertices present in the molecule surface. As predicted by theory, the dependence is linear.

number of iterations to convergence was observed to vary for different molecular systems. For surfaces from different molecules containing the same numbers of vertices, different numbers of iterations are required for convergence. The number of iterations required for convergence is reduced by approximately a factor of 3 on using the adaptive SOR technique. However, the technique can require storage of induced surface charges at all iterations. Hence it can be computer memory intensive as the number of vertices on the molecular surface increases. However, if the convergence criterion is chosen to depend linearly on the surface charge, it is possible to circumvent this problem. For example, if the difference in solvation energy, ΔSE, is chosen as the convergence criterion, eq. (30) can be used to calculate the solvation energy at the end of the nth iteration using a relaxation parameter ω, if we know the solvation energies calculated at the end of each iteration performed using a relaxation parameter of 1. To implement this scheme, we need to store only the solvation energy, and not the surface charges at the end of each iteration, thereby leading to much smaller memory requirements. With either method, the number of iterations is roughly constant for a given system with increasing surface resolution. Hence the boundary element implementation scales linearly as the number of vertices on the molecular surface.

We now compare the FMBE and FDPB methods in terms of speed for a given accuracy. Finite difference computations on a Convex C220, which runs at a comparable speed to the workstation used for our FMBE calculations, take around 15 seconds for a calculation with a grid size of 65^3, and around 2–3 minutes for a grid size of 129^3. These times include the time taken for creating the dielectric map. Figure 2 shows the time taken as a function of vertex number, with FMA operative and not. An exact comparison is difficult because the finite difference calculation times depend on the grid size chosen, which does not vary continuously with the scale in our implementation. It is clear that in spite of the fact that the fast multipole algorithm speeds up the boundary element method by over an order of magnitude, boundary element computations are still not as rapid as finite difference computations. However, with increasing resolution, they do not impose the excessive memory demands that plague finite difference computations. Furthermore, the cost of an FMBE computation scales better as the size of the problem, or the surface resolution used, increases. Consider a

molecule in a finite difference grid with n points on its dielectric interface. If the grid spacing were halved, the grid would contain eight times as many lattice points, and the time for each iteration would increase by this factor. Further, it will take twice as many iterations to converge. Hence the finite difference calculation takes 16 times longer when the grid scale increases by a factor of 2. With FMBE, the number of surface elements increases by a factor of 4 and hence the workload only increases by this amount (i.e., four times less than finite difference).

Discussion

We have shown that the limitations of the boundary element method, as applied to the Poisson-Boltzmann equation for macromolecular electrostatics, can be circumvented by use of the fast multipole method. In particular, such problems have excessive memory requirements and poor scaling of computational time with resolution. We have now a method in which the algorithmic complexity for both such quantities is optimal. We have used this method to extract accurate solvation energies for small proteins, and we are currently proceeding to use the approach in the study of complex formation and other biological phenomena. The method is still inherently limited in that it is not well suited to the calculation of ionic effects or to problems better represented by more than two dielectrics. Extension to such is possible, however, and would benefit from FMA technology in both cases.

In spite of the speedup conferred through use of the FMA, it is clear that the FMBE is still computationally expensive in comparison with other methods. Although the number of iterations needed for convergence with the FMBE is low, the time for each FMBE iteration is still large. Several avenues await exploration to counter this problem.

The molecular surfaces that we have used are grid based and therefore contain surface elements distributed uniformly throughout. In a boundary element calculation, little charge is induced on surface elements far away from atomic charges, whereas more charge is induced on surface elements near such. Hence, without sacrificing accuracy, one could reduce the number of elements used in the FMBE and therefore the time taken for each FMBE iteration by coalescing surface elements far away from all charges or where the induced surface charge is low. This could be done by using adaptively tesselated surfaces. This could even be done dynamically at runtime, based on the magnitude of the surface charge from the previous iteration.

The multigrid approach has yielded impressive speedups when applied to the finite difference method.[16,17] This method proceeds to solve a set of linear algebraic equations resulting from the discretization of a partial differential equation on a fine mesh by using auxiliary solutions on coarser meshes. Because much of the work in converging to the solution is done on the coarser and computationally inexpensive meshes, this accelerates the solution greatly. However, the multigrid method is not restricted to solutions of only partial differential equations. Multigrid methods of the second kind are formulations of the method for integral equations of the kind appearing in the boundary element method. They are also applicable when the equations are cast in a self-consistent form, as in the FMBE. Implementation of such a method in the FMBE would confer a significant speedup to our current method.

Our approach to the generation of the surface mesh can also be improved. Preliminary investigation has suggested that this is still an important source of error. Incorporation of a fast analytic surface method would solve remaining ambiguity. Also, use of surface integration techniques should improve accuracy without sacrificing computational speed. Finally, our appropriation of FMA technology has been somewhat off the shelf. A more careful integration of the technique may provide substantial savings in computer time.

The finite difference code, DelPhi, to which this method is consistently compared has undergone much development in terms of optimization. The current version is two orders of magnitude faster than the original implementation. Although we would be fortunate to obtain that level of refinement in our current code, it is clear that the combination of the aforementioned approaches should make FMBE an increasingly attractive choice for the quantitative study of the electrostatics of large molecule assemblages, such as occur in many biological systems.

Acknowledgments

This work was supported by the National Center for Research Resources division of the Biomedi-

cal Technology Program at the NIH (P41 RR06892) at Columbia University, NSF grant DIR-9207256, and NIH grant 30518. We are also grateful to the Pittsburgh Supercomputing Center for providing us access to the Cray through an NIH National Center for Research Resources grant (1 P41RR06009).

References

1. M. F. Perutz, *Science*, **201,** 1187 (1978).
2. J. Warwicker and H. C. Watson, *J. Mol. Biol.*, **157,** 671 (1982).
3. A. Warshel and S. T. Russell, *Q. Rev. Biophys.*, **17,** 283 (1984).
4. B. Honig, W. Hubbell, and R. Flewelling, *Ann. Rev. Biophys. Biophys. Chem.*, **15,** 163 (1986).
5. B. Jayaram, K. A. Sharp, and B. Honig, *Biopolymers*, **28,** 975 (1989).
6. K. A. Sharp and B. Honig, *Ann. Rev. Biophys. Biophys. Chem.*, **19,** 301 (1990).
7. M. E. Davis and J. A. McCammon, *J. Comp. Chem.*, **11,** 401 (1990).
8. J. G. Kirkwood, *J. Chem. Phys.*, **2,** 351 (1934).
9. C. Tanford and J. G. Kirkwood, *J. Am. Chem. Soc.*, **79,** 5333 (1957).
10. T. Alfrey, P. W. Berg, and H. Morawetz, *J. Polym. Sci.*, **7,** 543 (1951).
11. C. S. Murthy, R. J. Bacquet, and P. J. Rossky, *J. Phys. Chem.*, **89,** 701 (1985).
12. F. M. Richards, *Ann. Rev. Biochem. Bioeng.*, **6,** 151 (1977).
13. I. Klapper, R. Hagstrom, R. Fine, K. Sharp, and B. Honig, *Proteins*, **1,** 47 (1986).
14. M. K. Gilson, K. A. Sharp, and B. Honig, *J. Comp. Chem.*, **9,** 327 (1987).
15. A. Nicholls and B. Honig, *J. Comp. Chem.*, **12,** 435 (1991).
16. M. Holst and F. Saied, *J. Comp. Chem.*, **14,** 105 (1993).
17. H. Oberoi and N. M. Allewell, *Biophys. J.*, **65,** 48 (1993).
18. A.-S. Yang, M. R. Gunner, R. Sampogna, K. Sharp, and B. Honig, *Proteins*, **15,** 252 (1993).
19. R. J. Zauhar and R. S. Morgan, *J. Mol. Biol.* **186,** 815 (1985).
20. A. H. Juffer, E. F. F. Botta, B. A. M. van Keulen, A. van der Ploeg, and H. J. C. Berendsen, *J. Comp. Phys.*, **97,** 8 (1991).
21. A. A. Rashin, *J. Phys. Chem.*, **94,** 1725 (1988).
22. Y. N. Vorobjev, J. A. Grant, and H. A. Scheraga, *J. Am. Chem. Soc.*, **114,** 3189 (1992).
23. J. R. Reitz and F. J. Milford, *Foundations of Electromagnetic Theory*, Addison-Wesley, Reading, MA, 1967, p. 70.
24. R. J. Zauhar and R. S. Morgan, *J. Comp. Chem.*, **9,** 171 (1988).
25. A. A. Rashin, *Proteins*, **13,** 120 (1992).
26. A. Appel, *SIAM J. Sci. Stat. Comput.*, **6,** 85 (1985).
27. J. E. Barnes and P. Hut, *Nature*, **324,** 446 (1986).
28. L. Hernquist, *Comp. Phys. Comm.*, **48,** 107 (1988).
29. L. Greengard and V. Rokhlin, *J. Chem. Phys.*, **73,** 325 (1987).
30. L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, MIT Press, Cambridge, MA, 1988.
31. J. F. Leathrum, Jr. and J. A. Board, Jr., *The Parallel Fast Multipole Algorithm in Three Dimensions*, technical report, Duke University, Department of Engineering, 1992.
32. J. A. Board, Jr., J. W. Causey, J. F. Leathrum, Jr., A. Windemuth, and K. Schulten, *Chem. Phys. Lett.*, **198,** 89 (1992).
33. H.-Q. Ding, N. Karasawa, and W. A. Goddard, *J. Chem. Phys.*, **97,** 4309 (1992).
34. M. K. Gilson, A. Rashin, R. Fine, and B. Honig, *J. Mol. Biol.*, **183,** 503 (1985).
35. D. M. Young, *Comp. Phys. Comm.*, **53,** 1 (1989).
36. T. L. Hill, *J. Phys. Chem.*, **60,** 253 (1956).
37. S. Sridharan, A. Nicholls, and B. Honig, in preparation.
38. W. E. Lorensen and H. E. Cline, *Comp. Graph.*, **21,** 163 (1987).
39. A. Nicholls and B. Honig, in preparation.
40. A. Jean-Charles, A. Nicholls, K. Sharp, B. Honig, A. Tempczyk, T. F. Hendrickson, and W. C. Still, *J. Am. Chem. Soc.*, **113,** 1454 (1991).