Numerical Solution of the Poisson–Boltzmann Equation Using Tetrahedral Finite-Element Meshes

CHRISTIAN M. CORTIS,[1]* RICHARD A. FRIESNER[2]

[1]*Department of Applied Physics and Center for Biomolecular Simulation, Columbia University, New York, New York 10027*

[2]*Department of Chemistry and Center for Biomolecular Simulation, Columbia University, New York, New York 10027*

Received 3 September 1996; accepted 12 March 1997

ABSTRACT: The automatic three-dimensional mesh generation system for molecular geometries developed in our laboratory is used to solve the Poisson–Boltzmann equation numerically using a finite element method. For a number of different systems, the results are found to be in good agreement with those obtained in finite difference calculations using the DelPhi program as well as with those from boundary element calculations using our triangulated molecular surface. The overall scaling of the method is found to be approximately linear in the number of atoms in the system. The finite element mesh structure can be exploited to compute the gradient of the polarization energy in 10–20% of the time required to solve the equation itself. The resulting timings for the larger systems considered indicate that energies and gradients can be obtained in about half the time required for a finite difference solution to the equation. The development of a multilevel version of the algorithm as well as future applications to structure optimization using molecular mechanics force fields are also discussed. © 1997 John Wiley & Sons, Inc. *J Comput Chem* **18:** 1591–1608, 1997

*Present address: Department of Biochemistry and Molecular Biophysics, Columbia University, P&S, New York, NY 10032

Correspondence to: R. A. Friesner

Contract/grant sponsor: National Institutes of Health; contract/grant number: GM-42018

Contract/grant sponsor: FCAR

CCC 0192-8651 / 97 / 131591-18

Keywords: dielectric continuum; Poisson–Boltzmann equation; finite element; electrostatics; solvation

Introduction

The different motivations for the application of unstructured grid methods to the continuum electrostatics problem have been discussed in detail in a previous article in this issue; however, we will briefly recall them here for completeness. Hereafter, we shall refer to the former publication as part I.

In the study of solvation phenomena using continuum electrostatics, the quantity of interest is most often the polarization, or electrostatic, contribution to the solvation energy rather than the potential itself. The calculation of this quantity requires knowledge of the induced polarization charge located at the molecular surface which is a dielectric interface in this context. For this purpose, an accurate solution of the Poisson–Boltzmann equation is only required near the dielectric interface. (Even when ionic species are present in the solution, in the linear regime—which we will only consider here—the contribution from the surface charge term to the energy is overwhelming compared to that of the free ion term. The use of an unstructured grid makes it possible to resolve accurately the region around the interface while depleting the rest of the space, thus considerably reducing the total number of vertices required to obtain a converged numerical solution. At the same time, a three-dimensional formulation is retained which leads to a sparse matrix discretization. This feature is essential to the development of an efficient algorithm. Boundary element (two-dimensional) formulations, although attractive due to their greater ease of implementation, require the assembly and iterative solution of a dense linear system which makes them impractical for applications to molecular geometry optimization.

The present article is divided as follows: in the next section, we present the Galerkin finite-element formulation of the problem and the strategy used in assembling the linear system. In the third section we discuss the choice of an iterative solution strategy for the system of equations obtained, as well as issues pertaining to the generation of a multilevel version of the algorithm. The fourth section describes the methods used to calculate the polarization energy and gradient. Comparisons of the results obtained with the finite element (FE), finite difference/finite volume (FD), and boundary element (BE) methods, surface charge scaling issues; and timings are discussed in the fifth section.

Galerkin Formulation of the Problem

DISCRETIZATION

In the dielectric continuum solvent model, the solute is represented by some fixed charge distribution, $\rho(\vec{r})$, contained in a region assigned a low dielectric constant (the interior of proteins is sometimes assigned a dielectric constant between 2 and 4 to account for some degree of polarization[1]) embedded in a polarizable continuum assigned a high dielectric value. Free ions may also be present in the solvent and are assumed to be Boltzmann distributed at equilibrium. The pair distribution function for the ions is given by $g_{ij}(\vec{r}) = e^{-\beta\phi_{ij}(\vec{r})}$, where $\phi_{ij}(\vec{r})$ is the potential of mean force between the ions. This is the electrostatic interaction between two ions as screened by the presence of all the ions in the solvent at equilibrium in the field produced by the solute. For the case of a 1–1 salt, let $C = 2eI/kT$, where I is the ionic concentration in the solvent; the potential $\phi(\vec{r})$ is then the solution to the equation:

$$\nabla \cdot (\epsilon \nabla \phi) = -\frac{4\pi\rho}{kT} + 4\pi C \sinh(\phi e) \qquad (1)$$

subject to the boundary condition:

$$\lim_{\|\vec{r}\| \to \infty} \phi = 0 \qquad (2)$$

For weak ionic concentrations, the equation may be linearized, and one can write:

$$\nabla \cdot (\epsilon \nabla \phi) = -\frac{4\pi\rho}{kT} + \epsilon\kappa^2\phi \qquad (3)$$

where $\phi(\vec{r})$ is given in units of $[kT]$, and $\kappa^2 = 8\pi e^2 I/\epsilon kT$ is the inverse Debye length squared. We will focus on the linear form of the Poisson–Boltzmann equation hereafter as the finite

element method is directly applicable to it. To discretize the problem using a finite computational space, the boundary condition must then be modified, and one usually uses the asymptotic or Debye–Hückel potential, instead of the vanishing condition:

$$\phi(\vec{r})|_{\text{boundary}} = \sum_{i=1}^{Nc} \frac{q_i}{d_i} e^{-\kappa d_i} \text{ where } d_i = \|\vec{r} - \vec{r}_i\| \tag{4}$$

Given a complete set of basis functions $\{\psi_i\}$ $i = 1 \ldots N$, the electrostatic potential is expanded in a linear combination:

$$\phi(\vec{r}) = \sum_{i=1}^{N} c_i \psi_i(\vec{r}) \tag{5}$$

Substituting back in eq. (3) we obtain:

$$\sum_{i=1}^{N} c_i(\nabla \cdot (\epsilon \nabla \psi_i) - \epsilon \kappa^2 \psi_i) = -\frac{4\pi\rho}{kT} \tag{6}$$

The discretization of the problem is achieved by sampling the three-dimensional space over which the calculation is to be carried out with a set of grid points $\{\vec{r}_i,\ i = 1 \ldots N\}$ such that there is a one-to-one correspondence between the set of basis functions, ψ_i, and the set of grid points. The choice of grid points is arbitrary and can be tailored to reflect what may be known *a priori* about the solution to the equation. Furthermore, the functions, ψ_i, are only nonzero in some finite region around $\vec{r}_i$ and satisfy the conditions:

$$\psi_i(r_j) = 1 \text{ if } i = j$$
$$\psi_i(r_j) = 0 \text{ if } i \neq j$$

The coefficients c_i are then the values of the potential at the grid vertices. Hereafter, we will refer to the set of these values as the grid potential $\tilde{\phi} = c_i$, $i = 1 \ldots N$. Multiplying both sides of eq. (6) by ψ_j and integrating over all space one can write:

$$\sum_{i=1}^{N} c_i \left(\int_V \nabla \cdot (\epsilon \nabla \psi_i) \psi_j d^3\vec{r} - \int_V \epsilon \kappa^2 \psi_i \psi_j d^3\vec{r} \right) = -\int_V \frac{4\pi\rho}{kT} \psi_j d^3\vec{r} \tag{7}$$

Using Green's identity, the divergence theorem, and the vanishing condition for the potential [eq. (2)] we have:

$$\int_V \nabla \cdot (\epsilon \nabla \psi_i) \psi_j d^3\vec{r} = -\int_V \nabla(\epsilon \psi_i) \cdot \nabla \psi_j d^3\vec{r} \tag{8}$$

The potential values on the grid are obtained by solving the system of linear equations:

$$\sum_{i}^{N} A_{ij} c_i = b_j \tag{9}$$

where:

$$A_{ij} = -\int_V \nabla(\epsilon \psi_i) \cdot \nabla \psi_j d^3\vec{r} - \int_V \epsilon \kappa^2 \psi_i \psi_j d^3\vec{r}$$
$$b_j = -\int_V \frac{4\pi\rho}{kT} \psi_j d^3\vec{r} \tag{10}$$

The choice of grid points and associated basis functions will determine the structure of the linear system. The use of a three-dimensional unstructured mesh, constructed following the algorithm described in part I, will lead to a sparse system of equations that should be inexpensive to solve iteratively: because the density of the matrix assembled in eq. (10) depends on the connectivity of the mesh, and in a three-dimensional Delaunay-like triangulation this connectivity is local (i.e., a grid vertex is only coupled to its Voronoi neighbors), the resulting system of equations is sparse. This contrasts with boundary-element-based approaches in which the reduction in dimensionality achieved results in every surface element interacting with every other surface element. The resulting system of equations is therefore always dense. For a discussion of various discretization methods for partial differential equations and applications see Lapidus and Pinder.[2]

MATRIX ASSEMBLY

For simplicity, we choose linear basis functions over the tetrahedral elements. Given the nodal values of the potential, these provide a C^0 representation of the function over the mesh. The value of the basis function associated with a vertex i of element $\{i, j, k, l\}$ at a point $P(x, y, z)$ located inside the element is given by the ratio of the element volumes, $V(P, j, k, l)/V(i, j, k, l)$, as shown in Figure 1. Therefore, on a given tetrahedron in the mesh, T_{ijkl}, the basis function, ψ_i, will be given

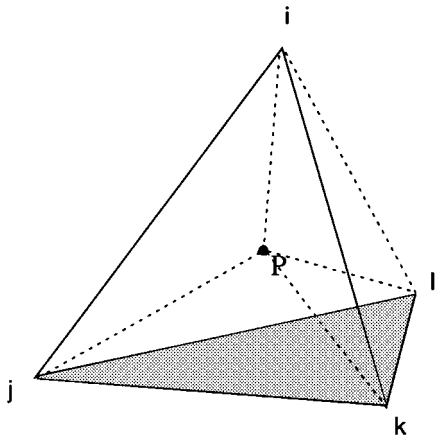

FIGURE 1. Tetrahedral mesh element. The basis functions associated with vertices (i, j, k, l) at point P are given by the volume ratios, $V(P, j, k, l) / V$, $V(P, k, l, i) / V$, $V(P, l, i, j) / V$, and $V(P, i, j, k) / V$.

by the ratio of the determinants:

$$\psi_i(x, y, z) = \frac{\begin{vmatrix} x - x_l & y - y_l & z - z_l \\ x_j - x_l & y_j - y_l & z_j - z_l \\ x_k - x_l & y_k - y_l & z_k - z_l \end{vmatrix}}{\begin{vmatrix} x_i - x_l & y_i - y_l & z_i - z_l \\ x_j - x_l & y_j - y_l & z_j - z_l \\ x_k - x_l & y_k - y_l & z_k - z_l \end{vmatrix}}$$

$$\text{for } (x, y, z) \in T_{ijkl} \qquad (11)$$

$$\psi_i(x, y, z) = 0 \qquad \text{otherwise}$$

In evaluating the integrals in eq. (10) one therefore needs only to take nearest neighbor overlaps into account. All the entries in row i of A will therefore be determined by integration over all the tetrahedra that share vertex i. In its present form, the tessellation algorithm produces a mesh where a tetrahedron shares a given vertex with a relatively small number of elements, Ntp_i, where $Ntp_i \sim 50$. A given row of A will therefore only have $2\,Ntp_i - 4$ nonzero elements. The resulting system of equations is therefore sparse and can be solved efficiently with any of several existing iterative methods. The A_{ij} coefficients are then calculated by cumulating the contribution of the integrals over a tetrahedron to all the relevant matrix elements. Because none of the element edges cross any of the discontinuous boundaries for ϵ and κ, the integrals in eq. (10) are easily evaluated in closed form for linear basis functions, eq. (11). Contributions from elements having one or more vertices belonging to the outer boundary on which condition eq. (4) is imposed, are multiplied by the boundary condition at the vertex and subtracted from the corresponding row of the right side of eq. (9). The electrostatic potential at the grid vertices is obtained by solving the resulting set of linear equations.

Iterative Schemes for the Linear System and a V-Cycle Multigrid Method

SELECTION OF ITERATIVE SOLVER

There are a number of iterative methods which could be employed to solve the set of linear equations obtained from the finite element discretization. The majority of the more sophisticated methods developed to date, however, are tailored to problems in which a predetermined grid structure can be specified. For the problem at hand, in addition to the development of an optimal successive overrelaxation algorithm by Nicholls and Honig,[3] a considerable amount of work has been done on a cubic multigrid formulation of the problem.[4,5] In both these cases, the use of the regular grid structure was central to the development of the algorithm.

In our unstructured grid approach, there is little *a priori* information available about the mesh structure which will result from the triangulation process. For this reason, there is almost no flexibility in the development of an optimized iterative solver without a complicated relabeling of the mesh vertices through the use of graph theoretical methods. The efficiency gain from such a reordering would at least in part be offset by the CPU cost of performing the relabeling. Among the currently available methods, we have experimented with the various algorithms provided as part of the ITPACK[6] set of programs for solving general sparse linear systems. Because the minimization of the total computing time has been the central aim in developing a finite element approach to the problem, we have tested the various algorithms available in this package for time to convergence rather than number of iterations. For the largest test system considered here, the crambin protein (Brookhaven Protein Data Bank code 1CRN), the Jacobi conjugate gradient (JCG) algorithm has proven the most efficient even though, depending on the grid parameters employed, it required approximately twice as many iterations as the symmetric successive overrelaxation conjugate gradi-

ent (SSORCG) algorithm to converge. This finding is consistent with the benchmark calculations reported in the ITPACK technical notes.[7]

In the light of this result, we have used our own implementation of the JCG algorithm both as a single level iterative solver and as a smoothing and coarse level operator in the two-level strategy discussed in what follows. A discussion of the basic conjugate gradient algorithm can be found in many texts on numerical analysis and iterative methods so it will not be further discussed here.[8] The Jacobi variation of the algorithm uses diagonal preconditioning so that given the linear system:

$$Ax = b \tag{12}$$

and the matrix D of diagonal elements of A, the following symmetrized system is solved using a conjugate gradient method:

$$(D^{-1/2}AD^{-1/2})D^{1/2}x = D^{-1/2}b \tag{13}$$

where symmetrized preconditioning is required as A is stored as a sparse symmetric matrix.

It is now recognized that for most structured elliptic problems, multigrid methods are the most efficient solvers.[9] In particular, ideal nested multigrid methods have been shown to exhibit $O(N)$ scaling, where N is the number of grid points in the mesh.[5] However, as will be discussed, a resolution of about 2 grid points per angstrom (gp/Å) results in an error on the order of 1 kcal/mol in the calculated polarization contribution to the solvation free energy. With this method one is therefore still bound by the $O(N_a)$ scaling of the grid size with the number of atoms for globular molecules which constitute ideal cases. In principle, a multigrid method based on the adaptive grid structure would combine the efficiency of multilevel methods with the approximate $O(N_a^{2/3})$ grid size scaling. This scaling estimate reflects the fact that the nonuniform grid-point distribution used in the algorithm[10] is designed to grow with the number of atoms contributing to the interface rather than with all the atoms in the molecule.

There are reports in the fluid dynamics literature of the development of unstructured multigrid solvers in two[11,12] and three dimensions.[13] Even though these methods make extensive use of unnested meshes (the boundaries of the element at one level intersect the boundaries of the elements at a different level) for which no convergence theorems exist,[5] they appear to have been effective in solving Euler equations in complicated geometries. The success of these methods has been the motivation behind the development of a multilevel method for our problem.

Unlike most computational fluid dynamics problems, we do not seek a high accuracy solution to the problem because of the inherent limitations of the model used. Instead, in experimenting with multigrid methods, our aim has been to reduce the time necessary to obtain an acceptable polarization energy. For practical purposes we will consider converged energies within 0.25 kcal/mol of the result obtained for the same system using a cubic finite-difference method at 2 grid points/Angstrom (gp/Å) to be acceptable. Consequently, for a given grid structure, rather than attempting to further refine the mesh, we seek coarser level descriptions of the problem that preserve the basic structure of the different physical interfaces in the system.

TWO-LEVEL MULTIGRID STRATEGY

The basic principle of multilevel iteration algorithms rests on the fact that iterative schemes for linear systems tend to converge the high-frequency components of the solution more quickly than the low-frequency ones. The strategy is then to use coarser representations of the linear operator obtained from the discretization of the partial differential equation, to converge the low frequency components of the correction to a given approximate solution to the linear system. A discussion of the efficiency of the method for the Poisson problem based on an analysis of the spectral properties of the discretized Laplacian operator can be found in the text by Hackbush.[9] Here, we briefly review the basic principle behind the multilevel method.

The x^0 be the exact solution to 3.1 and let $\bar{x}$ be an approximation to x^0. Define:

$$v = \bar{x} - x^0 \tag{14}$$

Acting on both sides of eq. (14) with A, we obtain:

$$A\bar{x} - Ax^0 = Av \tag{15}$$

where Av will be referred to as the defect. Let B be an approximate inverse to operator A, and multiply through by B:

$$BA\bar{x} - BAx^0 = BAv \Leftrightarrow$$
$$BA\bar{x} - Bb = \bar{v} \tag{16}$$

where the second equation defines the approximate defect $\bar{v}$. With the definition $\bar{\bar{x}} = \bar{x} - \bar{v}$, and I

the identity operator we obtain:

$$\bar{\bar{x}} = (I - BA)\bar{x} + Bb \tag{17}$$

Here $\bar{\bar{x}}$ is a new approximation to x^0. Setting $\bar{\bar{x}} = x^{n+1}$, $\bar{x} = x^n$ the general form of an iterative scheme for the solution of the linear system is obtained. The specific choice of the approximate inverse B determines the class of methods used, such as Jacobi, Gauss–Seidel, and others. Under certain conditions, it can be shown that if A is the discretized form of an elliptic operator, the defect [eq. (15)] is a smoother function that the approximate solution $\bar{x}$. Now, we define a sequence of finite element spaces $\{H^l,\ l = 1, \ldots, L\}$ with the property $H^l \subset H^{l+1}$ and an associated sequence of discretizations of operator A, which we denote $\{A^l,\ i = 1, \ldots, L\}$. We call l the level of discretization of the problem. We also introduce projection operators between different levels, I_m^l, defined by:

$$\begin{aligned} &H^l \to H^m \\ &v_l : \to \tilde{v}_m = I_m^l(v_l) \end{aligned} \tag{18}$$

The set of projection operators satisfies $I_l^l = I$. Also, we define P, the prolongation operator, by $P = I_{l+1}^l$ and R, the restriction operator, by $R = I_l^{l+1}$.

Given $\bar{x}_l$, an approximate solution obtained on level l, the aim is to obtain a correction to $\bar{x}_l$ from a correction computed at a coarser level of discretization. On level l, we define the defect d_l from:

$$A^l\bar{x}_l - b = A^l v_l \equiv d_l \tag{19}$$

As mentioned previously, the spectral properties of v_l allow us to approximate it with a projection of a correction obtained at level $l - 1$. Setting $v_l \approx \tilde{v}_l$, we obtain the latter correction from:

$$\begin{aligned} \tilde{v}_l &= Pv_{l-1} \\ &= P(A^{l-1})^{-1} d_{l-1} \\ &= P(A^{l-1})^{-1} Rd_l \end{aligned} \tag{20}$$

To write down an algorithm for the general method, we introduce a more compact notation for the iterative scheme described in eq. (17) associated with a given level of discretization:

$$\begin{aligned} x_l^{n+1} &= (I - B^l A^l)x_l^n + B^l b_l \\ &\equiv S_l(x_l^n, b_l) \\ &= S_l^{(n)}(x_l^1, b_l) \end{aligned} \tag{21}$$

With this notation, an approximate solution, x_l^{n+1}, is obtained at level l through the application of the operator S_l^n. The defect is then computed using eq. (19) and is used to obtain a correction through application of the operator $P(A^{l-1})^{-1}R$. The iteration sequence is then resumed on level l with operator S_l^n starting with the corrected solution $\tilde{x}_l^{n+1} = x_l^{n+1} - \tilde{v}_l$ and extended until convergence is achieved. Because the effect of the fine level iterations using operator S is to damp the high-frequency modes, these are usually referred to as smoothing iterations. The algorithm is represented schematically by:

$$\begin{aligned} &\text{Presmoothing iterations} && x \leftarrow S_l^{\nu_1}(x, b) \\ &\text{Coarse grid correction} && \begin{bmatrix} d \leftarrow R(A^l x - b) \\ v \leftarrow (A^{l-1})^{-1} d \\ x \leftarrow x - Pv \end{bmatrix} \\ &\text{Postsmoothing iterations} && x \leftarrow S_l^{\nu_2}(x, b) \end{aligned} \tag{22}$$

General multilevel methods employ a similar technique to solve the linear system, which appears in the calculation of the coarse grid correction. The process is then recursive and is extended up to the coarsest level, where ideally the defect can be computed exactly. The number of smoothing iterations, ν_1, ν_2, is usually determined empirically to optimize the rate of convergence of the algorithm. In a very restricted set of cases, the optimal number of smoothing operations can be determined *a priori*. Due to the constraints imposed by the geometry of the problem and the specific aims of the method, our current implementation uses a two-level mesh so that the calculation of the coarse grid defect must be done approximately. The two levels of discretization used are defined as:

(A) Grid used in single-level calculations. The corresponding mesh includes the result of the constrained Delaunay triangulation plus the surface refinement stage with four surface images.

(B) Grid used in single-level calculations without surface images. The corresponding mesh includes only the Delaunay triangulation of the initial grid.

The mesh levels (A) and (B) are almost identical throughout most of space except near the Richards surface. However, as the surface mesh generated by our algorithm contains approximately the same

number of vertices as the rest of the three-dimensional mesh, the linear system corresponding to the discretization on level (A) is about three times as large as that on level (B). The surface mesh is generated by constructing four images of each molecular surface vertex along the normal direction to the surface and directly meshing the volume spanned by these vertices (see part I). This local refinement is used to calculate accurately the normal component of the electric field on the molecular surface by finite differencing the potential in the normal direction. This procedure has been found to be far more reliable than differentiation of the expansion in eq. (5), which, with the basis functions used, leads to a constant gradient over the tetrahedral elements.

The number of JCG iterations performed on level (B) to compute the approximate grid defect d_B will be referred to as the number of *coarse grid iterations*, also the restriction operator R is simply taken to be the injection from level (A) to level (B). A distance weighted linear interpolation scheme is used as a prolongation operator to extend the correction from level (B) to level (A).

Polarization Energy and Energy Gradient Calculation

The method used here for calculating the polarization contribution to the solvation energy is similar to the one discussed by Cortis and coworkers.[14] This term is obtained from the equation:

$$G_{\text{pol}} = \frac{1}{2}\int_S \sigma\phi_0 d^3\vec{r} + \frac{1}{2}\int_V \rho_{\text{ion}}\phi_0 d^3\vec{r} \qquad (23)$$

where σ is the surface polarization charge, ρ_{ion} is the density of free ions in the solvent, and ϕ_0 is the electrostatic potential produced by the fixed charge distribution of the solute. These are calculated using the discontinuity of the electric field at the dielectric interface and the polarization vector as follows:

$$\sigma(\vec{r}) = \frac{1}{4\pi}\vec{n}\cdot(\nabla\phi^i - \nabla\phi^o)$$
$$\rho_{\text{ion}}(\vec{r}) = \nabla\cdot\vec{P} \qquad (24)$$

The electrostatic potential gradients at the surface are then numerically evaluated using the surface mesh described previously. In the linear approximation [eq. (3)], the term ρ_{ion} is proportional to the potential ϕ, so the second term in eq. (23) depends only on the integral of ϕ^2 in the ion-accessible region. The finite element discretization provides a simple scheme for evaluating this term by Gaussian quadrature. At physiological conditions, however, the contribution of the free ion term to the polarization energy is extremely small compared to the reaction field term. Therefore, numerical results will only be reported for the case of zero ionic concentration.

Two methods have been implemented for the calculation of the energy gradient. The first has been described in detail in the context of quantum-mechanical geometry optimization calculations.[14] It is based on differentiation of the operator obtained from the discretization of the PDE and has been referred to as a mesh operator derivative method (MOD). In this scheme, use is made of the discretized form of the expression for the energy:

$$\Delta G^{\text{pol}} = \vec{z}\cdot\tilde{\phi} + \vec{y}\cdot\tilde{\phi}^2 \qquad (25)$$

where $\vec{z}$ and $\vec{y}$ are vectors of integration weights depending on the coordinates of grid vertices and $\tilde{\phi}$ and $\tilde{\phi}^2$ are vectors containing the first and second powers of the grid potential values. Differentiating this expression, and making use of the linear equation satisfied by the grid potential, the following expression is obtained:

$$\frac{\partial\Delta G^{\text{pol}}}{\partial q} = \frac{\partial\vec{z}}{\partial q}\cdot\phi + \frac{\partial\vec{y}}{\partial q}\cdot\phi^2 + \vec{g}\cdot\left(\frac{\partial\vec{b}}{\partial q} - \frac{\partial A}{\partial q}\cdot\tilde{\phi}\right) \qquad (26)$$

where $\vec{g}$ satisfies the following equation:

$$A\cdot\vec{g} = \vec{z} + 2\tilde{\phi}\cdot(\vec{y}^T\cdot I) \qquad (27)$$

The adaptive grid structure is then used to evaluate numerically the various weight derivatives appearing in eq. (26) by using the dependence of each grid vertex on the nuclear coordinates as follows:

$$\frac{\partial w_i}{\partial q} = \sum_k \frac{\partial w_i}{\partial p_k}\frac{\partial p_k}{\partial q}$$
$$\frac{\partial a_{ij}}{\partial q} = \sum_k \frac{\partial a_{ij}}{\partial p_k}\frac{\partial p_k}{\partial q} \qquad (28)$$

where w_i is any of $\vec{z}$, $\vec{y}$, or $\vec{b}$; a_{ij} is any element of A; and p_k is a Cartesian coordinate of any vertex in the grid.

The second method we consider for obtaining the gradient components is based on the Maxwell stress tensor (MST) formulation of Gilson et al.[15] Starting from the expression for the electrostatic free energy derived by Sharp and Honig,[16] Gilson and coworkers obtained the following expression for the force density $\vec{f}$ associated with the Poisson–Boltzmann equation. In terms of ϕ, a solution of eq. (3), we have:

$$\vec{f} = \rho^f \vec{E} - \frac{1}{2} E^2 \nabla\epsilon - \frac{1}{2}\epsilon\kappa^2\phi^2\nabla\lambda \tag{29}$$

where ρ^f is the fixed charge density associated with the molecule, $\vec{E} = -\nabla\phi$, and $\lambda = \lambda(\vec{r})$ is 0 in the region of space from which free ions are excluded and 1 otherwise. It can be shown that the force density $\vec{f}$ can be expressed as the divergence of a tensor $\mathbf{P}$,[17] so that the force acting on a volume V is given by the expression:

$$\vec{F} = \int_V \nabla \cdot \mathbf{P} d^3\vec{r} \tag{30}$$

where $\mathbf{P}$ is the sum of the Maxwell stress tensor and a diagonal ionic pressure tensor. The decomposition has the following form:

$$\mathbf{P} = \vec{E}\vec{E} - \frac{1}{2}\epsilon\|\vec{E}\|^2 - kT\sum_{i=1}^{N}\left[(e^{q_i\phi/kT} - 1)c_i\right]\lambda \cdot I \tag{31}$$

where the coefficients c_i are the bulk concentrations of the N different ionic species. To obtain the forces acting on the atomic centers we must first determine the net force on the dielectric boundary. Applying eq. 30 to the case where V is the volume enclosed in this boundary, the difference in the stress tensor above and below the boundary is used to compute the net force on the object:

$$\vec{F}_{\epsilon\ \text{boundary}} = -\int_S \hat{n}(\hat{n}[\mathbf{P}_i - \mathbf{P}_o]\hat{n}) d^2\vec{r} \tag{32}$$

We rewrite this equation as:

$$\vec{F}_{\epsilon\ \text{boundary}} = -\int_S \frac{1}{2}(\epsilon_0 - \epsilon)\vec{E}_0 \cdot \vec{E} d^2\vec{r} \tag{33}$$

Alternately, relating the integrand to the surface charge density σ we obtain an expression more convenient for numerical evaluation:

$$\vec{F}_{\epsilon\ \text{boundary}} = -\int_S \left[2\pi\sigma(\vec{r})\epsilon_0 + \frac{1}{8\pi}(\epsilon - \epsilon_0)\|\vec{E}_0\|^2\right] d^2\vec{r} \tag{34}$$

The force is transferred from the dielectric interface to the atoms following a procedure suggested by Gilson and McCammon.[15] The algorithm can be described as follows:

- Partition the interface by assigning every region either to the surface of an atom or a probe position used in generating the concave parts of the interface.
- Assign a force to each atom and probe position by summing the contributions from the integrals on its assigned interface regions.
- The force on a probe position is transferred to the atoms with which it is in contact by decomposing the vector along the directions defined by the probe center and its contact atom centers. The decompositions is unique whether the probe is in contact with two or three atoms. In other words, given a force $\vec{F}p$ acting on a probe position p, the contribution f_i to the force acting on atoms i is obtained by solving the system:

$$\hat{n}_j \cdot \vec{F}p = \sum_i \hat{n}_j \cdot \hat{n}_i f_i \tag{35}$$

  where $\hat{n}_i$ is a unit vector from atom i to probe position p.

The contribution from the ionic boundary forces to the atomic forces can be treated with a similar procedure. The remaining contribution to the forces on the atomic centers comes from integrating eq. (29) over a volume enclosing all of ρ^f. Because we want to exclude the mutual Coulomb interaction of the fixed charges, and these are assumed to lie well within the surface, we subtract out the vacuum term:

$$\vec{F}_{\text{sources}} = \int_V \rho^f\left(\vec{E}^r - \vec{E}_0\right) d^3\vec{r} = \int_V \rho^f \vec{E}^r d^3\vec{r} \tag{36}$$

where $\vec{E}^r$ is the electric field produced by the surface charge density. One possible implementa-

tion of this method, using a boundary element discretization, has been reported by Zauhar,[18] but no comparisons with virtual work calculations or any other method are described. Results of calculations performed with the two different methods are discussed next.

Results

BENCHMARK CALCULATIONS AND CONVERGENCE ISSUES

We begin by comparing the energies obtained using different methods at varying resolutions for a benchmark set of small molecules. A dielectric constant of 80.0 was used everywhere for the solvent continuum along with a probe radius of 1.4 Å, corresponding to a model of aqueous solvation. The interior of all the molecules considered was assigned vacuum dielectric properties. The convergence criterion for the norm of the residual in the iterative solver was set to $5 \cdot 10^{-6}$ and energies are reported in kT units at 298 K. The first set of comparisons is done between the cubic finite-difference (FD) discretization implemented in the DelPhi program[3,19] and our adaptive finite element discretization, implemented in the PBF program. The FD calculations were performed at a 4.5 gp/Å resolution on grids of 111^3 vertices, while the FEM calculations used the highest density grids that could be generated and triangulated for these molecules using our algorithm. The corresponding parameters used in the grid/mesh generation are described in the footnotes to Table I. For all molecules in the set the FD results performed in 95^3 boxes at the same resolution differed by less than 0.01 kT unit, indicating that the calculations are converged, at least with respect to the imposition of boundary conditions of eq. (4). For a subset of these molecules, however, the results obtained at 4.0 and 4.5 gp/Å still differed by up to 0.12 kT units, indicating that convergence with respect to spatial resolution at 1% accuracy was still not achieved. Nevertheless, the converged FD results shown in Table I indicate that small differences remained between the saturated FEM and FD discretizations. A well-understood property of finite element meshes is that the error in the solution (defined as the difference between the solution to the discretized problem and the exact solution to the posed problem) depends not only on the density of the mesh but also on its quality. This property has been quantified in different ways in the literature, but can, in general, be related to a measure of the aspect ratio of the elements in the mesh. The differences observed between the FEM and converged FD calculations are then unlikely to be reduced simply by further increasing the mesh density. A local error analysis procedure coupled with a mesh refinement algorithm would be necessary to further converge the FEM results. A summary of the issues and methods used in error analysis on FE meshes, based on the work of Babuška and Zienkiewicz, can be found in a review by Kelly et al.[20] We have deferred the implementation of this type of technology in our algorithm for two reasons. First, the computational cost of an error analysis procedure, at the mesh generation stage, or subsequent to calculating a first solution to the discretized equations, would have to be offset by an overall speed-up of the rest of the algorithm for the method to remain competitive. Given that the current implementation exhibits approximately linear scaling with the size of the system (see subsequent text); further increase in efficiency would be difficult to achieve. Second, an exact solution to the continuum equations is unnecessary given the level of accuracy at which

TABLE I.
Comparison of FD (DelPhi) and FEM (PBF) Grid-Saturated Energy Calculations for Benchmark Set of Molecules.

Molecule	DelPhi	PBF
2-Pentanone	−7.52	−7.47
4-Methyl-2-pentanone	−13.47	−13.41
3,5-Dimethylpyridine	−11.78[a]	−11.78
Acetone	−11.31[a]	−11.19
Butadiene	−2.89	−2.85
cis-*N*-methylacetamide	−17.47	−17.40
Diethylsulfide	−6.15	−6.17
N,*N*-dimethylacetamide	−7.57	−7.42
Dimethylsulfide	−6.73	−6.68
Ethylamine	−11.51[a]	−11.32
Ethylbenzene	−4.80	−4.74
Ethylthiol	−5.09	−5.06
Methylthiol	−5.51	−5.45
Methylethylsufide	−5.48	−5.40
n-Butylamine	−11.30[a]	−11.22
n-Propylamine	−11.50	−11.28
Phenol	−13.45	−13.39

All energies are shown in kT units at 300 K. Grid generation parameters used in PBF: $n_{sh} = 14$, $n_{sf} = 5$, $l_{sf} = 29$, $g_{d0} = 0.15$, $g_{d1} = 0.42$, $g_{d2} = 0.70$, $g_{db} = 8.00$, $\alpha_1 = 0.0$, $\alpha_2 = 0.0$.
[a]In these FD calculations the differences between 4.0- and 4.5-gp/Å resolutions were greater than about 3%.

the theory is useful.[21] An estimate of the margin of error in the energy that could be obtained by performing calculations at several different resolutions is sufficient to assess the reliability of the discretization. For small molecules, such as those listed in Table I, convergence of the polarization energy to 0.35 kT units is considered sufficient for self-consistent reaction field calculations,[21,22] whereas, in molecular mechanics applications involving proteins, an error on the order of 1 kT can be tolerated.

A separate convergence question also arises. It is specific to the problem of dielectric interface representation. The use of tetrahedral elements and linear basis functions results in the molecule effectively being represented as a complex polyhedron with a finite number of faces. This implies that, in addition to the discretization error associated with any finite element representation of the solution, there is an additional error, which we refer to as the *representation error*, associated with any of the calculations. This is the difference between the exact solution to the problem if the interface were truly a polyhedron with a finite number of faces, and the solution to the complete problem where the dielectric interface is really the Richards surface described earlier. To estimate the size of these effects we compare the results of FEM calculations and boundary element calculations[23] (BEM) as implemented in Bharadwaj et al.'s BEM-FMA program. In our comparison, we use the same dielectric interface in both the BEM and FEM calculations. This is done by using the triangulated Richards surface obtained in the mesh generation stage as input to the BEM-FMA program. The results are shown in Table II.

An average difference of $1.50 \pm 1.25\%$ is obtained between the DelPhi and PBF calculations and of $0.70 \pm 0.17\%$ between the DelPhi and BEM calculations. Because an identical representation of the dielectric interface has been used in both the BEM and FEM calculations, this suggests that the *discretization error*, rather than the *representation error*, is the main contributor to the observed differences. Any further optimization of the method, therefore, should address the issue of mesh quality to minimize the *discretization error* away from the surface before attempting to improve the representation of the interfaces.

PROBLEM OF SURFACE CHARGE SCALING

When the boundary value problem under consideration is cast in Galerkin form [eq. (8)], it can

TABLE II.
Comparison of FEM (PBF) and Corresponding BEM Calculations (BEM-FMA) with Benchmark FD (DelPhi) Energy Calculations.

Molecule	DelPhi (G_{pol}^{FD})	BEM-FMA (G_{pol}^{BEM})	$\Delta(G_{pol}^{FD}, G_{pol}^{BEM})$ %	PBF (G_{pol}^{FEM})	$\Delta(G_{pol}^{FD}, G_{pol}^{FEM})$ %
2-Pentanone	−7.52	−7.57	0.65	−7.48	0.53
4-Methyl-2-pentanone	−13.47	−13.62	1.11	−13.41	0.45
3,5-Dimethylpyridine	−11.78[a]	−11.95	1.44	−11.38	3.40
Acetone	−11.31[a]	−11.36	0.44	−11.27	0.35
Butadiene	−2.89	−2.87	0.70	−2.78	3.81
cis-*N*-methylacetamide	−17.47	−17.68	1.20	−17.57	0.57
Diethylsulfide	−6.15	−6.19	0.65	−6.16	0.16
N,*N*-dimethylacetamide	−7.57	−7.53	0.53	−7.40	2.25
Dimethylsulfide	−6.73	−6.81	1.19	−6.62	1.63
Ethylamine	−11.51[a]	−11.38	1.13	−11.34	1.48
Ethylbenzene	−4.80	−4.79	0.21	−4.68	2.50
Ethylthiol	−5.09	−5.12	0.59	−5.07	0.39
Methylthiol	−5.51	−5.52	0.18	−5.43	1.45
Methylethylsulfide	−5.48	−5.47	0.18	−5.40	1.46
n-Butylamine	−11.30[a]	−11.36	0.53	−11.24	0.53
n-Propylamine	−11.50	−11.39	0.96	−11.21	2.52
Phenol	−13.45	−13.47	0.15	−13.18	2.00

Percent differences are shown with respect to the FD calculations. All energies are shown in kT units at 298 K. Grid generation parameters used in PBF: $n_{sh} = 9$, $n_{sf} = 4$, $l_{sf} = 20$, $g_{d0} = 0.15$, $g_{d1} = 0.40$, $g_{d2} = 0.70$, $g_{db} = 8.00$, $\alpha_1 = -1.50$, $\alpha_2 = 1.50$.
[a]See Table I.

be shown that the exact solution to the discretized problem $\tilde{\phi}$ satisfies internal boundary conditions on the surface of discontinuity for the coefficient in the equation. For the specific case of the electrostatics problem, this implies that the electric field[24] obtained from $\tilde{\phi}$ should satisfy:

$$\epsilon_i \vec{E}^i \cdot \hat{n} = \epsilon_o \vec{E}^0 \cdot \hat{n} \tag{37}$$

on the molecular surface.

Now, let ξ be some coordinate such that the molecular surface can be written as $f_{MS}(\xi) = c$, where c is an unspecified constant. Application of Gauss' law on a surface at $\xi - \epsilon$, and use of eq. (37) fixes the magnitude of the total surface polarization charge as follows. Taking $\xi \to 0$ we have:

$$\oint_S \sigma(\vec{r}) d^2\vec{r} = -\frac{1}{4\pi}\left(1 - \frac{\epsilon_0}{\epsilon}\right) Q_{\text{solute}} \tag{38}$$

where Q_{solute} is the total charge on the solute contained within the molecular surface. Because eq. (37) is not enforced for every iterate in the solution process, the total induced surface charge will only approximately satisfy eq. (38). To reduce the error introduced by this approximate internal boundary condition, some form of surface charge scaling is required. It should be noted, at this point, that the so-called FD representation used in DelPhi[19] is technically a finite volume formulation (as the discretized equation is an integrated form of the PDE), which enforces the charge conservation condition exactly at each iteration. Therefore, no scaling procedure is required on the cubic grid.

In the FE calculations, the rescaling of the total induced polarization charge is performed in a postprocessing step. One possible approach, used in BE discretizations, is to solve the problem separately for positive and negative sources (assuming a point charge representation of the solute charge distribution) and to rescale the positive and negative surface charges separately. This uniform rescaling approach has been shown to be effective in improving total solvation energies, but does not guarantee an improvement in the surface charge calculated locally on the surface. Furthermore, it requires two iterative solutions of the linear eqs. (9). For these reasons, we have chosen a different approach to the problem. After computing an initial solution to eqs. (9), we introduce the discretized form of eq. (38) as an additional equation in the linear system. The overdetermined system of equations can now be solved iteratively by a least squares method. By using the solution to the invertible problem as an initial guess, only a few iterations (1–10) are necessary to converge the total induced polarization charge to the predicted value.

For neutral molecules, such as the ones used in the benchmark calculations, the change in energy produced by the charge scaling step is negligible ($\leq 0.01\ kT$). To better assess the effect of charge scaling on the calculations, we now turn to a different set of test cases. We have chosen to examine the crambin protein in its folded conformation, obtained from x-ray diffraction experiments (PDB code 1CRN) and associated fragments. Because of its small size, it was possible to use it in FD and FEM calculations at varying resolution without exhausting system resources to compare the performance of the different algorithms. However, because this particular protein is not soluble in water we emphasize that the energies calculated in what follows should serve only as an illustration of the method and are not meant to reflect the physical properties of the system. The structure is composed of 46 residues and carries a net charge of approximately 1.17e, as assigned using rounded-off charges from the AMBER[25] molecular mechanics potential (round-off contribution is 0.17e). For our purposes, the substructures generated by truncating the residue sequence at various points are also useful test systems, although they are not chemically meaningful. Similarly, the present discussion is independent of the particular charge distribution used because we do not address the question of predicting any of the actual physical properties of the molecule. The number of atoms corresponding to the truncated sequences are given in Table V.

It is apparent from Tables III and IV that the surface charge scaling step produces a systematic improvement in the calculated energies (taking the DelPhi calculations at the highest available resolution as a benchmark). In only one case ($N_{res} = 8$, $l_{sf} = 14$) does the discrepancy between the FEM and FD calculations increase slightly after the scaling step. For the higher resolution calculations (Table IV), the magnitude of the correction in the polarization energy resulting from this step is, in all cases, correlated with the error in the induced surface charge predicted from the FEM calculation. For the lower resolution calculations shown in Table III the correlation is not evident.

TABLE III.
Effect of Surface Charge Scaling on Polarization Energies.

N_{res}	Q_t	Q^0_{ind}	Q^1_{ind}	PBF-G^{FEM-1}_{pol}	Q^2_{ind}	PBF-G^{FEM-2}_{pol}	DelPhi-G^{FD}_{pol}
5	0.99	−0.98	−0.95	−149.49	−0.96	−150.05	−149.78
8	1.01	−1.00	−0.98	−175.09	−0.99	−175.55	−174.93
12	2.05	−2.02	−1.97	−339.32	−2.00	−342.02	−340.81
24	2.03	−2.00	−1.97	−424.48	−2.00	−426.03	−428.28
36	2.13	−2.10	−2.03	−447.48	−2.09	−450.70	−451.00
46	1.17	−1.16	−1.10	−492.71	−1.15	−494.31	−495.92

Q_t: total charge on the system; Q^0_{ind}: induced charge predicted from eq. (38); Q^1_{ind}: induced charge from solution to FEM equations; Q^2_{ind}: induced charge after the scaling step; G^{FEM-1}_{pol}: polarization energy from solution to FEM equations; G^{FEM-2}_{pol} polarization energy after charge scaling step; G^{FD}_{pol} polarization energy from FD calculation at 3.0 gp / Å resolution in a 129^3 box. Grid generation parameters used in PBF: $n_{sh} = 8$, $n_{sf} = 3$, $l_{sf} = 14$, $g_{d0} = 0.15$, $g_{d1} = 0.40$, $g_{d2} = 0.00$, $g_{db} = 8.00$, $\alpha_1 = -1.50$, $\alpha_2 = 0.50$.

TABLE IV.
Effect of Surface Charge Scaling on Polarization Energies.

N_{res}	Q_t	Q^0_{ind}	Q^1_{ind}	PBF-G^{FEM-1}_{pol}	Q^2_{ind}	PBF-G^{FEM-2}_{pol}	DelPhi-G^{FD}_{pol}
5	0.99	−0.98	−0.96	−149.64	−0.97	−150.08	−149.78
8	1.01	−1.00	−0.98	−175.60	−1.00	−175.97	−174.93
12	2.05	−2.02	−1.98	−339.63	−2.01	−341.77	−340.81
24	2.03	−2.00	−1.98	−426.16	−2.00	−427.10	−428.28
36	2.13	−2.10	−2.04	−447.04	−2.09	−449.98	−451.00
46	1.17	−1.16	−1.10	−493.98	−1.15	−495.33	−495.92

Q_t: total charge on the system; Q^0_{ind}: induced charge predicted from eq. (38); Q^1_{ind}: induced charge from solution to FEM equations; Q^2_{ind}: induced charge after the scaling step; G^{FEM-1}_{pol}: polarization energy from solution to FEM equations; G^{FEM-2}_{pol} polarization energy after charge scaling step; G^{FD}_{pol} polarization energy from FD calculation at 3.0-gp / Å resolution in a 129^3 box. Grid generation parameters used in PBF: $n_{sh} = 8$, $n_{sf} = 3$, $l_{sf} = 16$, $g_{d0} = 0.15$, $g_{d1} = 0.40$, $g_{d2} = 0.00$, $g_{db} = 8.00$, $\alpha_1 = -1.50$, $\alpha_2 = 0.50$.

TABLE V.
Number of Atoms in Truncated Sequences of Crambin Protein.

N_{res}	5	8	12	24	36	46
N_{atoms}	66	112	171	349	527	649

RESULTS FROM POLARIZATION ENERGY GRADIENT CALCULATIONS

We now compare the performance of the two gradient algorithms discussed previously. In a first test, all the gradient components were calculated with both methods, as well as by energy finite difference (EFD), for the water molecule at different mesh resolutions. The EFD gradient components were computed using:

$$\frac{\partial E}{\partial q_i} \approx \frac{E\left(\vec{a}_i + \vec{\Delta} q\right) - E\left(\vec{a}_i - \vec{\Delta} q\right)}{2\|\vec{\Delta} q\|} \qquad (39)$$

where $\vec{a}_i$ is the position of atom i and $\vec{\Delta} q$ is a displacement vector along one of the Cartesian axes. The results are shown in Table VI for all the gradient components in the water molecule and in Table VII for the components depending on the carbonyl carbon (C_2) of the 2-pentanone molecule. Both the MOD and MST methods produce results that are in good agreement with the EFD results although, for the largest components, the MOD numbers are somewhat more consistent across the range of mesh resolutions. Also, the cancellation of error in the latter method prevents the appearance of a spurious net force on the system as is evident from the last three columns in the table (the component sums along each axis are not identically zero but are ~ 100 times smaller than those obtained with the MST method). This term would have to be removed by rescaling the forces on each atom which would introduce an additional error in a geometry optimization calculation. For this reason, the MOD method has been selected over the

TABLE VI.

Gradient Components in Units of kT / Å at 298 K for Water.

l_{sf}	O_x	O_y	O_z	$H1_x$	$H1_y$	$H1_z$	$H2_x$	$H2_y$	$H2_z$	Σ_x	Σ_y	Σ_z
MOD												
8	−0.02	0.20	35.45	0.27	−6.42	−17.72	−0.25	6.22	−17.73	0.00	0.00	0.00
11	0.78	−0.19	35.73	−0.47	−5.88	−17.84	−0.31	6.07	−17.89	0.00	0.00	0.00
14	0.06	0.15	37.63	−0.02	−3.99	−18.80	−0.04	3.84	−18.83	0.00	0.00	0.00
16	0.25	−0.18	35.01	−0.16	−5.17	−17.51	−0.09	5.35	−17.50	0.00	0.00	0.00
19	−0.02	0.01	35.19	−0.02	−4.72	−17.60	0.04	4.71	−17.59	0.00	0.00	0.00
MST												
8	0.10	0.10	36.45	0.05	−6.31	−18.06	0.00	6.23	−18.16	0.16	0.02	0.23
11	−0.06	−0.48	35.60	−0.20	−5.20	−17.71	0.07	5.32	−17.68	−0.19	−0.36	0.21
14	−0.05	−0.02	37.25	−0.25	−3.85	−18.72	0.25	3.86	−18.56	−0.05	−0.01	−0.03
16	0.01	−0.00	33.69	−0.04	−4.61	−16.84	−0.00	4.59	−16.83	−0.03	−0.02	0.02
19	−0.04	0.04	34.88	−0.01	−4.48	−17.24	0.05	4.46	−17.24	0.01	0.02	0.40
EFD (Δq)												
0.05	−0.02	0.06	35.15	0.01	−4.85	−17.87	0.02	4.86	−18.02			
0.01	−0.01	0.08	35.03	0.01	−4.29	−17.62	0.01	4.17	−17.57			

Results from the different gradient calculation methods are shown at various grid densities determined by the parameter l_{sf}. The sum of gradient components in each direction is listed in the last three columns. The results from energy finite differences are given in the last two rows. The atomic displacement Δq is given in angstroms. Grid generation parameters used in PBF, for MOD, MST, and EFD calculations: $n_{sh} = 8$, $n_{sf} = 3$, $g_{d0} = 0.15$, $g_{d1} = 0.40$, $g_{d2} = 0.00$, $g_{db} = 8.00$, $\alpha_1 = -1.50$, $\alpha_2 = 0.70$; and, for energy FD (EFD) calculations: $l_{sf} = 19$.

MST method as the algorithm of choice in self-consistent reaction field (SCRF) geometry optimizations.[14]

However, the CPU cost of the MOD approach is comparable to that of an entire FE calculation for the energy. It would therefore be impractical to consider this method in the context of molecular mechanics applications to large molecules where the error introduced by the net force removal would be tolerable. In this respect, the MST algorithm presents a significant advantage as it depends on only the ease with which integrals on the dielectric interface can be calculated. Because a triangulated description of the surface as well as the exterior electric field values are directly available in the finite element formulation, the CPU cost of evaluating the relevant terms is small compared to that of the rest of the calculation (see Table IX).

TIMINGS AND ALGORITHM EFFICIENCY

We now turn to the central question of efficiency of the algorithm. To do so, we attempt to reproduce the results of single-grid FD calculations for the various truncated sequences of the crambin protein with a set of transferable grid generation parameters. The target energies we compare with are those from FD calculations at the 3.0 gp/Å resolution, which are converged with respect to the imposition of boundary conditions, as the change in energy with respect to box size is less than 0.5%. The results obtained for the small molecule set indicate that this level of convergence with respect to spatial resolution may not be achieved at this grid spacing. The 3.0 gp/Å resolution benchmark is nevertheless consistent with a 1 kT convergence requirement in energy, which is adequate for obtaining estimates of relative energies between different protein conformations. The results of polarization energy calculations and associated timings obtained from the FD formulation with different resolutions and box sizes are shown in Table VIII. The converged FD results are compared with those from the FE calculations in Table IX. Unless specified otherwise, FEM timings are for the complete calculation with the JCG algorithm used as a single-level iterative solver. Also, in all FE calculations on the protein fragments, a single interior surface shell was used in the surface mesh generation stage[10] instead of four shells for the high-resolution calculations on the small molecule set (Table I).

The results listed in Table IX suggest that the FEM grids used with a surface resolution level set at $l_{sf} = 14$ can produce results of comparable accu-

TABLE VII.

Gradient Components in Units of kT / Å at 298 K for the C2 Carbon of 2-Pentanone.

l_{sf}	$C2_x$	$C2_y$	$C2_z$	Σ_x	Σ_y	Σ_z
MOD						
8	−3.98	−9.47	−6.85	0.00	0.00	0.00
11	−9.27	−8.30	−5.17	0.00	0.00	0.00
14	−9.35	−7.86	−4.96	0.00	0.00	0.00
16	−9.21	−7.94	−4.92	0.00	0.00	0.00
19	−8.90	−7.75	−5.35	0.00	0.00	0.00
28	−9.12	−7.85	−4.95	0.00	0.00	0.00
MST						
8	−9.55	−7.89	−4.76	0.57	−0.43	0.22
11	−9.00	−7.81	−5.03	0.12	0.62	0.13
14	−9.20	−7.90	−5.27	0.21	−0.09	0.05
16	−9.08	−7.84	−4.88	−0.01	0.48	0.07
19	−9.05	−7.95	−5.13	−0.06	0.22	0.03
28	−9.14	−7.90	−5.00	0.10	−0.06	−0.03
FD (Δq)						
0.05	−9.54	−7.79	−5.08			
0.01	−9.62	−7.46	−5.75			

Results from the different gradient calculation methods are shown at various grid densities determined by the parameter l_{sf}. The sum of gradient components in each direction is listed in the last three columns. The results from energy finite differences are given in the last row. The atomic displacement Δq is given in angstroms. Grid generation parameters used in PBF for MOD and MST calculations: $n_{sh} = 8$, $n_{sf} = 3$, $g_{d0} = 0.15$, $g_{d1} = 0.40$, $g_{d2} = 0.00$, $g_{db} = 8.00$, $\alpha_1 = -1.50$, $\alpha_2 = 0.70$; and for energy FD calculations: $l_{sf} = 21$, $n_{sh} = 9$, $n_{sf} = 4$; $g_{d0} = 0.15$, $g_{d1} = 0.40$, $g_{d2} = 0.00$, $g_{db} = 8.00$, $\alpha_1 = -1.50$, $\alpha_2 = 0.50$.

TABLE VIII.

Crambin Fragment Energies and Timings for Different Resolutions and Grid Sizes for FD Calculations.

N_{res}	Grid size	N (gp / Å)	DelPhi—G^{FD}_{pol}	CPU[a] (s)
5	65^3	2.5	−149.77	19.14
5	65^3	3.0	−149.78	20.93
8	85^3	2.5	−175.34	49.34
8	93^3	3.0	−174.93	66.37
12	65^3	2.0	−339.91	28.20
12	95^3	2.5	−340.98	79.42
12	125^3	3.0	−340.81	82.05
24	95^3	2.0	−420.66	115.63
24	95^3	2.5	−423.29	117.63
24	125^3	3.0	−428.28	247.04
36	95^3	2.0	−441.32	143.36
36	125^3	2.5	−444.66	300.80
36	125^3	3.0	−451.00	305.28
46	95^3	2.0	−482.62	336.45
46	125^3	2.5	−490.78	340.80
46	125^3	3.0	−495.92	347.58

[a] DelPhi timings were measured on a Convex C210 and then converted for comparison with timings for an IBM 370 RS6000 workstation. IBM 370 calculations using the version of DelPhi available on our system were only possible on grids up to 65^3. The measured conversion factor for the same 65^3 calculation on the C210 and the IBM 370 was 0.64. For example, the C210 timing for the $N_{res} = 5$, 2.5 gp / Å case was 29.90 s.

racy to the ones obtained from the FD formulation at 3.0 gp/Å resolution with a box size sufficiently large to converge the boundary conditions. Furthermore, the FEM calculation can be carried out in approximately one half the CPU time required to perform the FD calculation. Also, the results obtained with the different surface resolution parameters indicate that some degree of grid optimization may reduce the overall number of grid points per atom necessary to obtain the same quality results, thus reducing even further the CPU time required by the method. Although both methods in their current form are still far too expensive to carry out molecular dynamics or Monte Carlo simulations if complete solution to the continuum electrostatics problem were required at every step, the FEM formulation presents a significant advantage over the FD methods, considering that the cost of evaluating the energy gradient components using the MST method is 10–20% that of the total calculation (Table IX, column 6). In the context of solution-phase geometry optimizations using molecular mechanics force fields and a dielectric continuum solvent, the speed-up obtained by using the FE instead of a FD or BE discretization should be greater than the observed twofold factor. A further advantage of the method lies in the possibility of deforming the mesh in the geometry optimization process and making use of the converged solution obtained in the previous molecular configuration as an initial guess to distribute the mesh generation cost over several steps and reduce the number of iterations required to converge the potential.

Figures 2–6 illustrate the general scaling properties of our implementation with system size where we have chosen the number of atoms in the system as an obvious measure of size. Timings obtained on an IBM 370 RS6000 workstation are reported for a breakdown of the calculation into four principal parts, namely *grid generation*, *triangulation*, *matrix assembly*, and *iterative solution*, as

TABLE IX.
Crambin Fragment Energies and Timings for Different Grid Structures for FEM and BEM Calculations.

N_{res}	l_{sf}	N_p / atom	PBF-G_{pol}^{FEM}	CPU[a] (s)	CPU[b] (s)
5	8	82	−148.65	7.45	0.62
5	11	117	−149.89	10.18	0.80
5	14	133	−150.01	11.95	0.97
5	16	152	−150.24	13.44	1.03
8	8	78	−174.09	13.35	1.46
8	11	111	−175.17	18.60	1.97
8	14	129	−175.70	24.74	2.33
8	16	148	−175.88	25.43	2.56
12	8	68	−338.62	21.35	2.58
12	11	98	−340.52	29.02	3.70
12	14	114	−342.08	35.96	4.23
12	16	129	−341.98	37.99	4.75
24	8	62	−427.59	56.09	9.03
24	11	90	−422.89	78.99	12.43
24	14	105	−425.54	92.50	14.56
24	16	119	−427.08	101.66	16.37
36	8	52	−444.92	83.85	15.52
36	11	75	−446.36	110.37	21.83
36	14	88	−450.20	135.03	26.06
36	16	101	−449.37	142.90	28.38
46	8	50	−487.48	106.37	21.32
46	11	72	−490.47	145.63	30.74
46	14	84	−494.28	174.64	36.22
46	16	96	−494.79	192.53	39.61

Timings shown are for the FEM calculations carried out on an IBM 370 RS6000 workstation. Grid generation parameters used in PBF: $n_{sh} = 8$, $n_{sf} = 3$, $g_{d0} = 0.15$, $g_{d1} = 0.40$, $g_{d2} = 0.00$, $g_{db} = 8.00$, $\alpha_1 = -1.50$, $\alpha_2 = 0.70$.
[a]Total CPU time required to solve to the PB equation (includes grid generation, triangulation, matrix assembly, and iterative solution of linear system).
[b]CPU time required to compute the polarization energy gradient using the MST method.

well as for the complete calculation. The results suggest that, apart from the grid generation stage, the three other parts of the algorithm scale approximately linearly with the size of the system. The nonlinear behavior of the grid generation procedure can be traced to the work needed to build the molecular surface. Because the construction depends on pairs and triplets of atoms, its scaling properties are necessarily nonlinear. Furthermore, because the system is still rather small to benefit from a surface-to-volume effect, the number of surface atoms grows approximately linearly, leading to the scaling behavior implied by the data in Figure 2. Hence, the variability of the surface structure from one molecule to another makes it

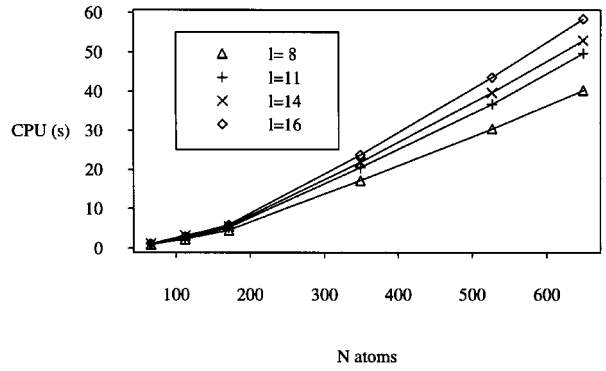

FIGURE 2. CPU time to generate grid. Timings include operations needed to generate Richards surface information. Scaling with number of atoms is shown for different grid structures determined by the parameter l_{sf}. Other grid generation parameters used: $n_{sh} = 8$, $n_{sf} = 3$, $g_{d0} = 0.15$, $g_{d1} = 0.40$, $g_{d2} = 0.00$, $g_{db} = 8.00$, $\alpha_1 = -1.50$, $\alpha_2 = 0.70$.

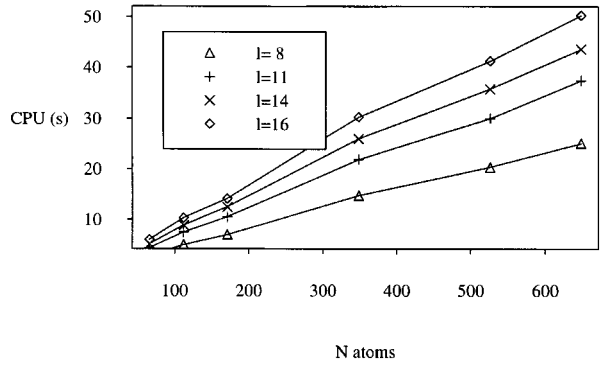

FIGURE 3. CPU time to construct mesh using a constrained Delaunay triangulation algorithm. Notation as in Figure 2.

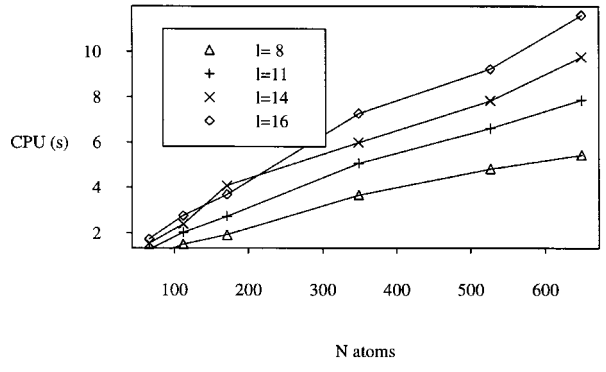

FIGURE 4. CPU time to assemble finite element equations. Notation as in Figure 2.

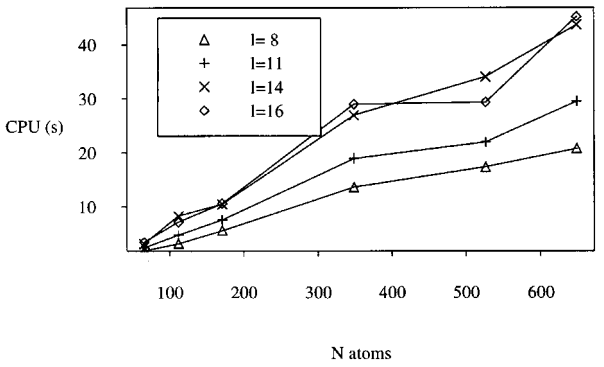

FIGURE 5. CPU time to solve linear system using the JCG algorithm. Notation as in Figure 2.

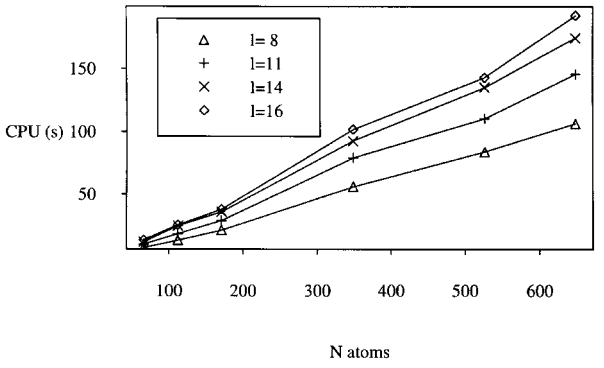

FIGURE 6. Total CPU time. Notation as in Figure 2.

difficult to draw any general conclusions with regard to system size for this algorithm. Only the behavior as a function of the complexity of the surface can be reasonably estimated. Nevertheless, the timings obtained for the complete procedure (Fig. 6) suggest that the general approach of using a high grid density near the two-dimensional interface and a sparse grid elsewhere has been effective in producing an algorithm which combines the scaling behavior of a boundary-element-like method with the speed of a three-dimensional structured grid method.

TIMING RESULTS FOR TWO-LEVEL ITERATIVE SOLVER

Finally, we present some timing comparisons between the two-level V-cycle and JCG iterative solvers. We have used the five-residue fragment of crambin and a grid constructed with a value of $l_{sf} = 16$ as a test case. The comparison is done by measuring the total time required to obtain a residual norm of $5.0 \cdot 10^{-6}$ with both methods. The solution time required for the single-level JCG algorithm, which requires 271 iterations to converge, is 11.46 s. The number of coarse grid iterations N_c used to obtain an approximate solution to the linear equations for the defect d_B and the number of presmoothing iterations ν_1 are varied in an attempt to minimize the number of postsmoothing iterations ν_2 required to converge the system. The results and timings are shown in Table X.

The results indicate that no increase in efficiency is obtained from the two-level solver., Although for several choices of N_c, ν_1 the total number of fine-level (A) iterations $\nu_1 + \nu_2$ required to converge the equations is less than 271, the number of iterations required by the single grid solver, the efficiency is lost due to the cost of performing the coarse grid iterations and of transferring information across the two mesh levels. This observation would still hold even without the 2.5-s penalty which must be paid for assembling

TABLE X.
Results of Parameter Search for an Optimal Number of Postsmoothing Iterations ν_2.

$N_c \setminus \nu_1$	10	20	30	40	50	80	100
10	254 / 11.69	248 / 11.84	240 / 11.88	229 / 11.82	222 / 11.86	203 / 12.46	189 / 12.72
20	252 / 11.73	244 / 11.89	239 / 12.03	227 / 11.96	219 / 12.06	199 / 12.46	187 / 12.79
30	248 / 11.78	242 / 11.92	233 / 11.94	226 / 12.11	218 / 12.18	195 / 12.46	188 / 12.99
40	245 / 11.80	243 / 12.16	235 / 12.22	223 / 12.14	216 / 12.28	195 / 12.60	186 / 13.12
50	245 / 12.01	242 / 12.30	235 / 12.41	226 / 12.44	216 / 12.44	194 / 12.69	187 / 13.30
60	244 / 12.13	242 / 12.44	237 / 12.63	224 / 12.51	218 / 12.69	196 / 13.03	187 / 13.50
70	—	—	—	224 / 12.69	217 / 12.83	195 / 13.09	187 / 13.47
80	—	—	—	—	—	194 / 13.27	186 / 13.82

Table entries show the number of iterations ν_2 and the total CPU time in seconds (ν_2 / T) required to converge the linear system for different choices of the number of presmoothing iterations ν_1 and coarse grid iterations N_c. The test system chosen is the one- to five-residue fragment of crambin. See text for the choice of grid-generation parameters.

TABLE XI.
Norm of Residual $\|\vec{r}\|$ for Linear System at Start of Postsmoothing Iterations for Different Numbers of Coarse Grid Iterations and 100 Presmoothing Iterations.

N_c	$\|\vec{r}\|$
10	0.006497
20	0.005993
30	0.005700
40	0.005656
50	0.005645
60	0.005644
70	0.005641
80	0.005640

the coarse level operator in addition to the fine level one. Although the method appears to work formally, we do not obtain a speed-up from it. The inefficiency of the coarse grid iterations can be easily understood if the behavior of the residual is examined more closely. For the $\nu_1 = 100$ case, the presmoothing iterations decrease the norm of the residual from its initial value of 34.784 to 0.012. The new residual norm, obtained after calculation of the coarse grid correction, is shown in Table XI for the different number of coarse grid iterations.

It is clear that the effect of the coarse grid correction saturates after about 70 iterations, because further iterations do not contribute to reducing the residual at the start of the postsmoothing iterations. We have identified two possible causes for this problem, although others may also exist. There may be a problem with the form of the coarse-level operator implemented in the solver so that the defect computed from the coarse-level equations is incorrect. Alternately, the restriction and prolongation operators we have chosen for the problem, to which these methods are known to be very sensitive (this issue has been discussed extensively by Holst[5]), are inadequate and, although the correct defect may be available on mesh (B), the error introduced by the interpolation onto mesh (A) sets a limit on how much uncorrupted information can be transferred this way. Clearly, there is still much to be done in terms of identifying better mesh decomposition schemes and more reliable interpolation operators if anything is to be gained from the implementation of an unstructured multilevel solver algorithm.

Conclusion

The FE element method we have presented for the solution of the linear Poisson–Boltzmann equation appears to present some advantages over FD or BE formulations. These arise from the significant reduction in the number of vertices required to cover the space associated with the solute and a retention of the sparse matrix structure associated with a three-dimensional discretization. An indirect advantage of the approach is the availability of a triangulated molecular surface and accurate values of the electric field in its neighborhood, which can be used to calculate derivatives of the polarization energy with respect to the nuclear coordinates through application of the MST method proposed by Gilson and coworkers.[15] Current efforts are being focused on further optimization of the method and coupling to molecular mechanics force fields to carry out geometry optimization on biomolecular structures. Future work will involve the development of an alternate grid structure which should be better suited to the application of multilevel methods as well as the development of a formulation for the full nonlinear equation.

Acknowledgments

The authors thank Anthony Nicholls, Ranganathan Bharadwaj, Sundaram Sridharan, Andreas Windemuth, and Barry Honig for many instructive discussions regarding molecular surfaces and continuum electrostatic models as well as for the use of the DelPhi program. C. M. C. thanks Michael Holst for helpful discussions regarding multigrid methods and the problem of continuum electrostatics. This work was supported through funding from NIH grant NIHGM40526. C. M. C. acknowledges support from the FCAR fund.

References

1. B. Honig and A. Nicholls, *Science*, **268,** 1144 (1995).
2. L. Lapidus and G. F. Pinder, *Numerical Solution of Partial Differential Equations in Science and Engineering*, Wiley, New York, 1982.
3. A. Nicholls and B. Honig, *J. Comput. Chem.*, **4,** 435 (1991).
4. M. Holst and F. Saied, *J. Comput. Chem.*, **14,** 105 (1993).
5. M. J. Holst, Ph.D. thesis, University of Illinois at Urbana-Champaign, 1993.
6. D. Young and D. Kincaid, *The ITPACK Package for Large Sparse Linear Systems*, Academic Press, New York, 1981.

7. D. R. Kincaid, et al. *ITPACK* 2*C*: *A FORTRAN Package for Solving Large Sparse Linear Systems by Adaptive Accelerated Iterative Methods*, Center for Numerican Analysis, U.T. Austin, Austin TX, 1993.
8. A. Ralston, *A First Course in Numerical Analysis*, McGraw-Hill, New York, 1965.
9. W. Hackbush, *Multi-Grid Methods and Applications*, Vol. 4: *Springer Series in Computational Mathematics*, Springer, Berlin, 1985.
10. C. M. Cortis and R. A. Friesner, *J. Comput. Chem.*, (this issue).
11. M. P. Leclercq and B. Stoufflet, *J. Comput. Phys.*, **104,** 329 (1993).
12. D. J. Mavriplis, *AIAA J.*, **26,** 824 (1988).
13. D. J. Mavriplis, *AIAA J.*, **30,** 1753 (1992).
14. C. M. Cortis and R. A. Friesner, *J. Chem. Phys*. (in press).
15. M. Gilson, M. E. Davis, B. A. Luty, and J. A. McCammon, *J. Phys. Che.*, **97,** 3591 (1993).
16. K. Sharp and B. Honig, *J. Phys. Chem.*, **94,** 7684 (1990).
17. M. Gilson, M. E. Davis, B. A. Luty, and J. A. McCammon, Data not included in paper (obtained directly from author), January 1995.
18. R. J. Zauhar, *J. Comput. Chem.*, **12,** 575 (1991).
19. I. Klapper, R. Hagstrom, R. Fine, K. Sharp, and B. Honig, *Proteins*, **1**, 47 (1986).
20. D. W. Kelly, J. P. De, S. R. Gago, O. C. Zienkiewicz, and I. Babuška, *Int. J. Numer. Meth. Eng.*, **19,** 1593, 1621 (1983).
21. B. Marten, K. Kim, C. Cortis, R. B. Murphy, M. N. Ringnalda, R. A. Friesner, D. Sitkoff, and B. Honig, *J. Phys. Chem.*, **100,** 11775 (1996).
22. D. J. Tannor, B. Marten, R. Murphy, R. A. Friesner, D. Sitkoff, A. Nicholls, M. Ringnalda, W. A. Goddard III, and B. Honig, *J. Am. Chem. Soc.*, **116,** 11875 (1994).
23. R. Bharadwaj, A. Windemuth, A. Nicholls, S. Sridharan, and B. Honig, *J. Comput. Chem.*, **16,** 898 (1995).
24. G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, NJ, 1973.
25. S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta Jr., and P. Weiner, *J. Am. Chem. Soc.*, **106,** 765 (1984).