Mathematics and Molecular Neurobiology

Nathan A. Baker[1], Kaihsu Tai[1], Richard Henchman[1], David Sept[1], Adrian Elcock[2], Michael Holst[3], and J. Andrew McCammon[1]

[1] Howard Hughes Medical Institute, Department of Chemistry and Biochemistry, and Department of Pharmacology, University of California, San Diego, La Jolla, CA 92093-0365
[2] Department of Biochemistry, University of Iowa
[3] Department of Mathematics, University of California, San Diego

Abstract. Advances in mathematics and computer technology, together with advances in structural biology, are opening the way to detailed modeling of biology at the molecular and cellular levels. One objective of such studies is the development of a more complete understanding of biological systems, including the emergence of behavior at the cellular level from that at the molecular level. Another objective is the development of more sophisticated models for structure-aided discovery of new pharmaceuticals.

1 Introduction

Computer simulations of biomolecular dynamics began in the 1970's with a nine picosecond molecular dynamics simulation of a small protein with no explicit treatment of the solvent surroundings of the protein [1]. This work was stimulated in part by the development of mathematical methods such as predictor-corrector algorithms for computing the solution of ordinary differential equation initial value problems [2]. These algorithms facilitated the solution of Newton's equations of motion for a large molecule (albeit only a small protein), a "stiff" differential equation problem due to the wide range of characteristic time scales that extends from fast bond stretching (femtoseconds) to the slow changes in overall shape of the protein.

Even with these advances, and further advances in subsequent years in the field of molecular dynamics simulation, many important problems are characterized by time scales that are inaccessible to Newtonian simulation. This has stimulated the development of a variety of coarse-grained simulation methods, including Brownian dynamics methods in which the inertial terms of Newtonian dynamics are assumed to be negligible due to rapid damping by viscous surroundings [3]. Solvent is treated implicitly in such studies. In particular, the electrostatic interactions in such models have typically been modeled by finite difference solution of the Poisson-Boltzmann equation [4, 5].

In the remainder of this chapter, we outline recent and ongoing work by our groups at the current interface between mathematics and biology. The biological applications are all relevant to the activities of nerve cells, particularly the motor neurons that control muscle contraction. These cells typically comprise a cell body that contains most of the metabolic apparatus; a long, slender process called an axon that carries the excitatory impulse toward the muscle; and a presynaptic structure that releases the neurotrasmitter acetylcholine (ACh), which diffuses to the muscle to initiate contraction [6]. Current studies make use of molecular dynamics to study the activity of single proteins, Brownian dynamics to study the assembly of protein complexes, and the Poisson-Boltzmann equation to calculate the electrostatic interactions that govern the assembly and stability of protein complexes. Among the protein systems considered here are acetylcholinesterase (AChE), the enzyme that removes ACh from the synapse to set the stage for another round of muscle contraction; fasciculin 2, a snake venom protein that acts by binding to and inhibiting AChE; and actin and tubulin, proteins that form cytoskeletal structures that are essential for the formation of neurons and the transport of materials along the axon from the cell body to the synapse.

2 Molecular Dynamics of AChE

2.1 Introduction

Acetylcholinesterase (AChE) is the enzyme responsible for the termination of signaling in cholinergic synapses such as the neuromuscular junction by degrading the neurotransmitter acetylcholine. AChE has a gorge, 2 nm deep, leading to the catalytic site. Molecular dynamics (MD) simulations have shown breathing motions of this gorge [7–9], and that an alternative portal, the back door, providing access to the catalytic site is present in AChE which may facilitate rapid solvent and product removal.

2.2 Preparation of the molecular dynamics trajectory

A previous 1 ns MD simulation of mouse acetylcholinesterase [10] was extended to afford a trajectory of 10 ns. The protein was solvated in a cubic box (9.6 nm edges) of pre-equilibrated water molecules. The simulation system had a total of 8 289 solute atoms and 75 615 solvent atoms. The simulation was performed in the isothermal-isobaric ensemble, and the solvent and solute were separately coupled to temperature reservoirs of 298.15 K with coupling times of 0.1 ps. Pressure was restrained to 1 atmosphere with a coupling time of 0.4 ps. All minimization and MD simulation steps were performed using NWChem Version 3.0 (High Performance Computational Chemistry Group, Pacific Northwest National Laboratory, Richland, Washington 99352, USA) with the AMBER 94 force field [11]. This code makes use of domain decomposition to achieve good performance on parallel machines [12]. The simulation was performed on 32 processors of a Cray T3E parallel supercomputer at the San Diego Supercomputer Center over a period of 3 years, consuming a total of about 200 processor-months of supercomputer time. Frames were collected at 1 ps intervals for the simulation length of 10.8 ns, giving 1.08×10^4 frames. The first 700 frames of this trajectory was considered the equilibration phase; only the subsequent 10 ns was used in the main analysis.

2.3 Stability of the trajectory

Various properties, such as the total potential energy, the solute (protein and counterion) potential energy, root mean square displacement from the initial crystal structure, and temperature, were inspected to ensure that the 10 ns trajectory was stable and suitable for analysis (data not shown). The isotropic temperature (B) factor was calculated from the mean square fluctuation (msf) using the equation $B = \frac{8\pi^2}{3}$ (msf) [13, 14]. The B factor plot for the crystal structure and that calculated from the MD trajectory have similar features (Figure 1). The experimental B factors include a variety of contributions (e.g., crystal contacts, static disorder in the crystal) in addition to that from the dynamical fluctuations, and these additional contributions are expected to be substantial for a structure of modest resolution [15]. Here, the experimental resolution is 0.32 nm (3.2 Å) [16]. The termini exhibited larger B factors in the MD simulation where the protein is solvated in water rather than packed in a crystal.

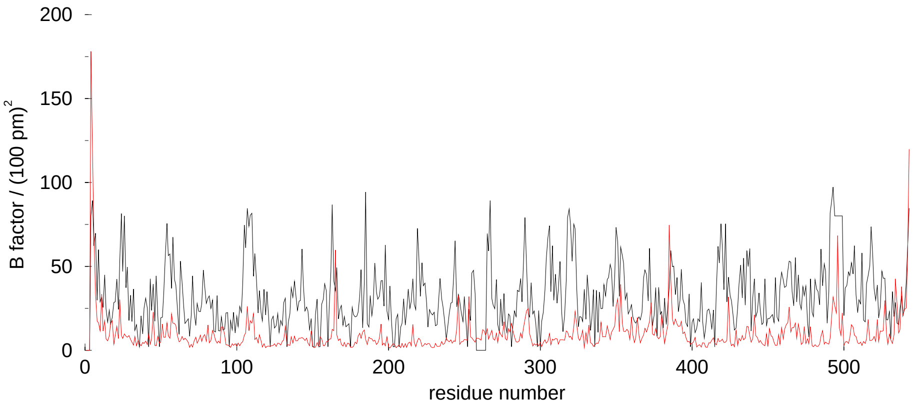

Fig. 1. The B factor for each residue from the AChE part in the fasciculin 2–mouse AChE complex crystal structure (black) and that calculated from the msf in the 10 ns MD trajectory (red).

2.4 Definition of the gorge proper radius

In order to characterize the degree of the gorge opening with a single variable, we defined the gorge proper radius $\rho(t)$ for the conformation of the snapshot at time t as the maximum radius of a spherical ligand that can go through the gorge from outside the protein to reach the bottom. Equivalently, it is the maximum probe radius with which we can still generate a solvent accessible surface with a continuous topology. We calculated the time series of the gorge proper radius at a resolution of 5 pm (Figure 2).

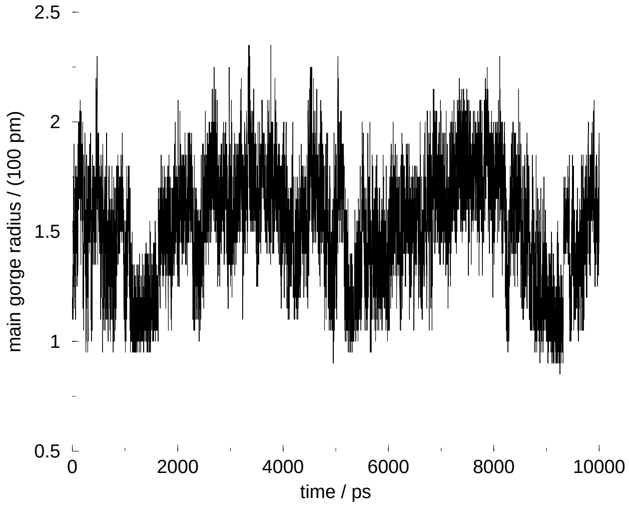

Fig. 2. The 10 ns time series for the gorge proper radius, $\rho(t)$.

2.5 Correlation of α-carbon positions and velocities with the gorge proper radius

We define the correlation between the ith degree of freedom, $r_i(t)$, and the gorge proper radius $\rho(t)$ to be

$$\frac{\left\langle \left(r_i(t) - \langle r_i\rangle_t\right)\left(\rho(t) - \langle \rho\rangle_t\right)\right\rangle_t}{\left\langle r_i(t) - \langle r_i\rangle_t\right\rangle_t \left\langle \rho(t) - \langle \rho\rangle_t\right\rangle_t}. \tag{1}$$

We calculated a correlation vector, thus defined, with 3 degrees of freedom for the α-carbon atom of each residue (Figure 3).

Similarly, the average velocity covariance is defined as

$$\left\langle \big(r_i(t) - r_i(t - 1\text{ ps})\big)\big(\rho(t) - \rho(t - 1\text{ ps})\big)\right\rangle_t \tag{2}$$

The average velocity covariance vector is shown in Figure 4.

The vectors reveal how much each residue moves in concert with the gorge. The moiety of AChE that includes the gorge has remarkable concerted motion

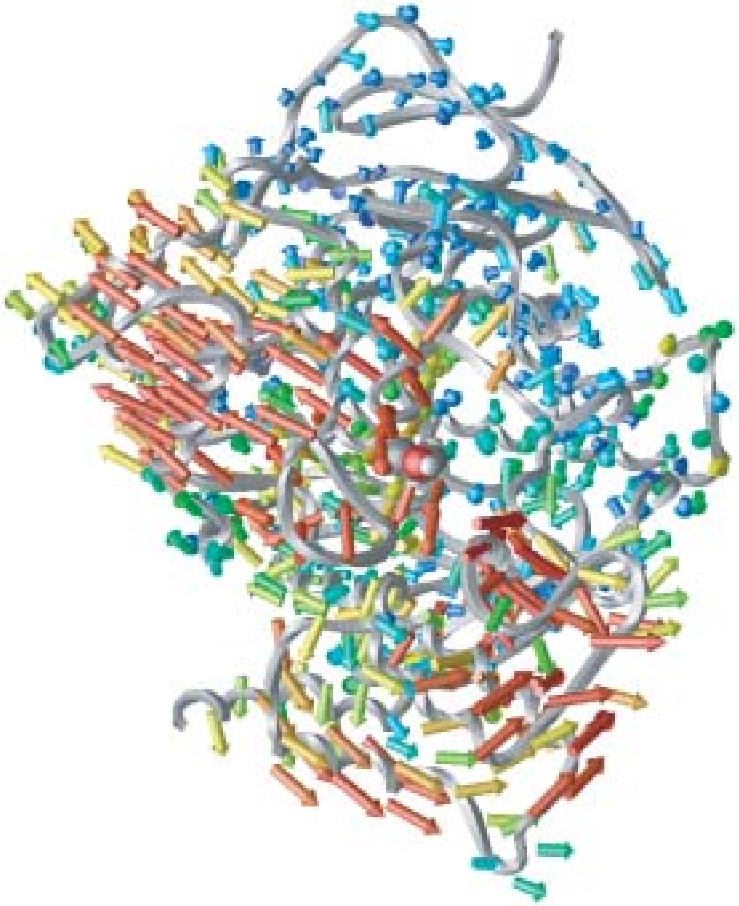

Fig. 3. The correlation vector (see text for definition) for each residue of acetylcholinesterase (AChE). The vector lengths are in arbitrary units. The vectors are colored by their length; the longest one being red. This is the conventional orientation for displaying AChE, with the N-terminus on the top right, and the C-terminus on the bottom left. The viewer is looking into the gorge, with the active site Ser203 marked by the spacefill model. It is easy to see that the moiety (front) containing the gorge has large correlated motions with the gorge. A virtual reality markup language (VRML) file for this plot is available on the web at http://mccammon.ucsd.edu/.

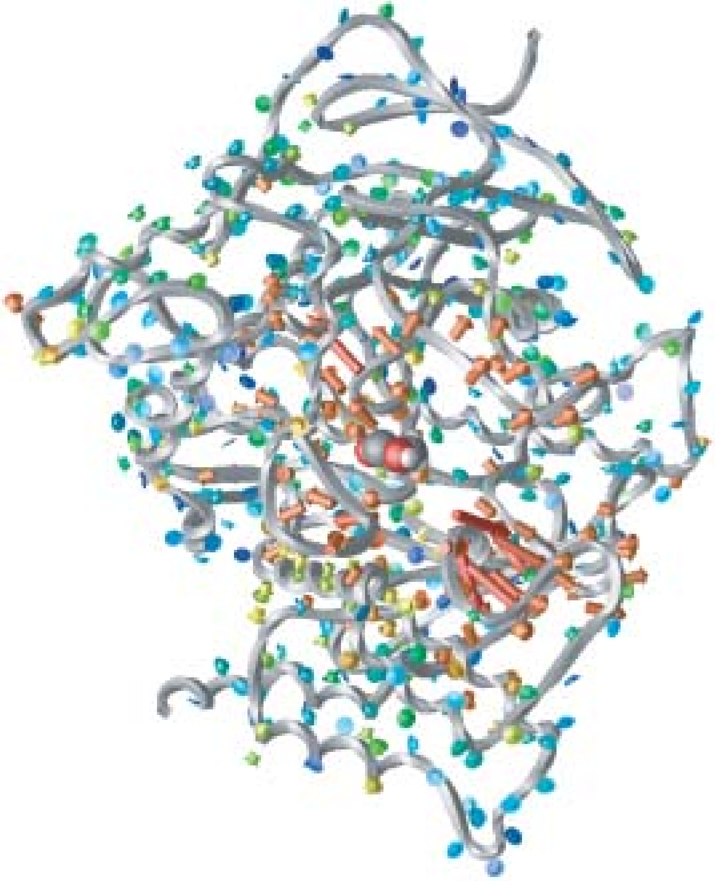

Fig. 4. The average velocity covariance vector (see text for definition) for each residue of acetylcholinesterase (AChE). The vector lengths are in arbitrary units. The vectors are colored by their length; the longest one being red. The orientation is the same as in Figure 3.

with the gorge proper radius. The residues in this moiety generally have correlation vectors pointing away from the gorge. These residues apparently move away from the gorge when the gorge opens. Some residues that are in the other moiety, which are closer to the base of the gorge, have correlation vectors that are generally smaller. The discrimination of concerted motions is even more pronounced in Figure 4: The residues that construct the gorge itself have the largest average velocity covariance vectors. The vectors decrease in magnitude; the residues that are farthest from the gorge have the smallest average velocity covariance vectors.

We speculate that inhibitors such as fasciculin 2 may decrease the likelihood of gorge opening by restricting loops on the surface of AChE that have large concerted motions, in addition to sterically occluding the entrance to the gorge. This remains to be confirmed by our current work on the MD simulation of the fasciculin 2–mouse AChE complex.

2.6 Contribution of the principal components to the gorge proper radius fluctuations

Principal component analysis [17] of $N = 530$ α-carbons (Leu9 to Lys538; 1590 degrees of freedom) was performed. We then have a transformation matrix T, whose columns are the eigenvectors $\mathbf{v}_c$. That is,

$$\mathsf{T} = \left[\, \mathbf{v}_1 \; \mathbf{v}_2 \cdots \mathbf{v}_{3N} \,\right] \tag{3}$$

such that

$$\mathit{\Delta}\mathbf{r}(t) := \mathbf{r}(t) - \langle \mathbf{r} \rangle_t = \mathsf{T}\mathbf{p}(t) \tag{4}$$

where $\mathbf{p}(t)$ is a vector containing the projection in each of the principal components $c = 1$ to $3N$ at time t, and $\mathbf{r}(t)$ is a vector containing the Cartesian coordinates for residue 1 direction x, residue 1 direction y, ..., up to residue N direction z at time t.

We found that the Phe338 $\mathrm{C}_{\epsilon 2}$–Tyr124 O_{H} distance correlated well with $\rho(t)$, with a correlation coefficient of 0.91. The Phe338 C_α–Tyr124 C_α distance and $\rho(t)$ had a correlation coefficient of 0.55.

It can be proved that one can express exactly the square of the Phe338 C_α–Tyr124 C_α distance $d_{124,338}$ in terms of the projection time series:

$$\left[d_{124,338}(t)\right]^2 = S + \sum_{c=1}^{3N} R_c p_c(t) + \sum_{c_1=1}^{3N} \sum_{c_2=1}^{3N} Q_{c_1 c_2} p_{c_1}(t) p_{c_2}(t) \tag{5}$$

where S, the R_cs, and the $Q_{c_1 c_2}$s are functions of the average structure $\langle \mathbf{r} \rangle_t$ and the transformation matrix T only, neither of which is time-dependent. We calculated the time-dependent contributions to $\left[d_{124,338}(t)\right]^2$, namely, the linear terms $\sum_c R_c p_c(t)$ and the cross terms $\sum_{c_1} \sum_{c_2} Q_{c_1 c_2} p_{c_1}(t) p_{c_2}(t)$ (Figure 5).

The contribution of the cross terms is not negligible, since they are at times of opposite direction but of the same magnitude as the linear term. Therefore,

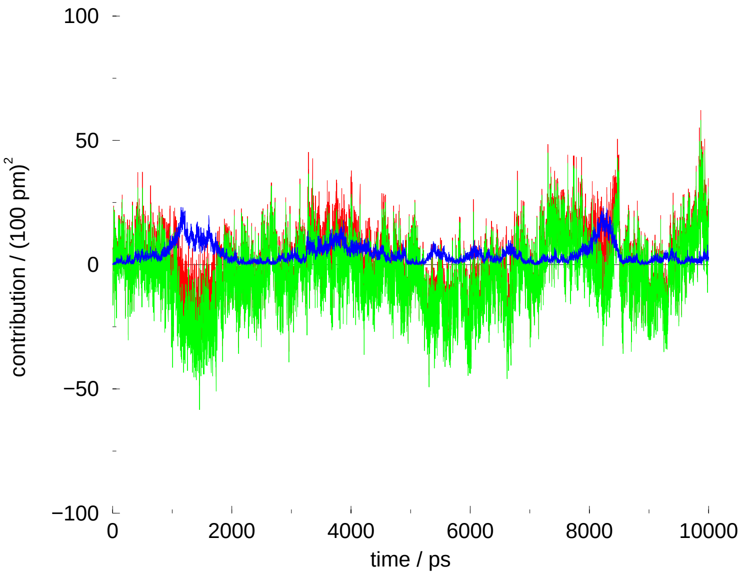

Fig. 5. Contribution of the linear ($\sum_c R_c p_c(t)$, green) and the cross ($\sum_{c_1} \sum_{c_2} Q_{c_1 c_2} p_{c_1}(t) p_{c_2}(t)$, blue) terms to the square distance $[d_{124,338}(t)]^2$, and the sum of these two (red). See text for definitions.

it is not justified to assume that only the linear terms contribute to the square distance $[d_{124,338}(t)]^2$.

We have proposed previously [18] that the gorge proper radius exhibits fractal-like behavior without characteristic scale of length or time. Our results here are in concert with that proposal.

2.7 Water molecules in the gorge

Since AChE catalysis is observed to occur at the diffusion-controlled limit [19], it is important to understand how the behavior of water molecules in AChE may affect its function. In particular, it is desirable to know the mechanism and frequency of water molecules diffusing into and out of the gorge, for it is these waters that must be displaced by a substrate trying to enter.

In order to measure the population of gorge water molecules, gorge waters are defined to be that cluster of waters all within 4 Å of each other around the active site residue Ser203. If such a cluster extends outside the gorge to the bulk, a junction water is defined between the gorge and bulk. This molecule assumes a gorge or bulk identity depending on from where it came. Such a definition reduces noise in gorge population at the gorge-bulk interface. The overall gorge water population defined in this way ranges from 16 to 22 molecules (Figure 6), the dominant population being 19–20 waters. It is interesting to note

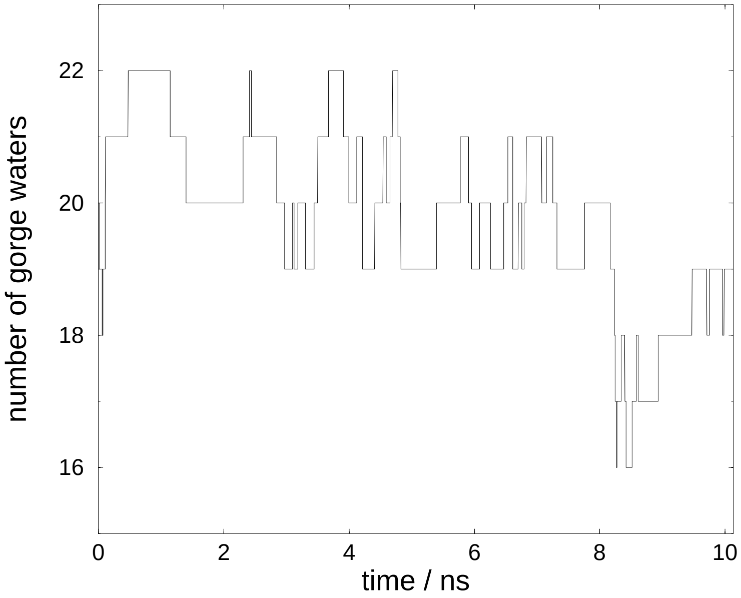

Fig. 6. Population of gorge waters as a function of time.

that the change in gorge water volume due to such population fluctuations is approximately equivalent to a typical substrate molecule.

There are a number of possible entrances to the gorge [10]. The four paths found connecting the gorge to the bulk are the main gorge proper, as described earlier, a second "front door", and two "back doors", all illustrated in Figure 7. Of these, the main gorge is approximately 70 % of the time open wide enough for a water molecule, while the other three are never open more than 1 % of the time [18]. 53 water transits are observed between the gorge and the bulk, all of them via the main gorge, while none are observed to pass by the other paths. During these transits, no obvious motions of the protein side chains lining the main gorge are detected. This is in agreement with the previous finding that the opening of the gorge is due to larger scale protein motions. In addition to water transits, waters are also occasionally observed to pass into or out of (become separated by > 4 Å from the main cluster) the residues making up the side of the gorge itself.

One of the striking observations about gorge waters is that they are much less mobile than in bulk. Although diffusion in the confined gorge is anisotropic, the average mean-squared displacement for gorge waters changes two orders of magnitude more slowly than that in bulk. Evidently, the mobility of the substrate would be severely impeded if too many water molecules are present in the gorge.

What all these observations suggest is that substrate entry is likely to occur after the prior exit of a sufficient number of water molecules rather than both of

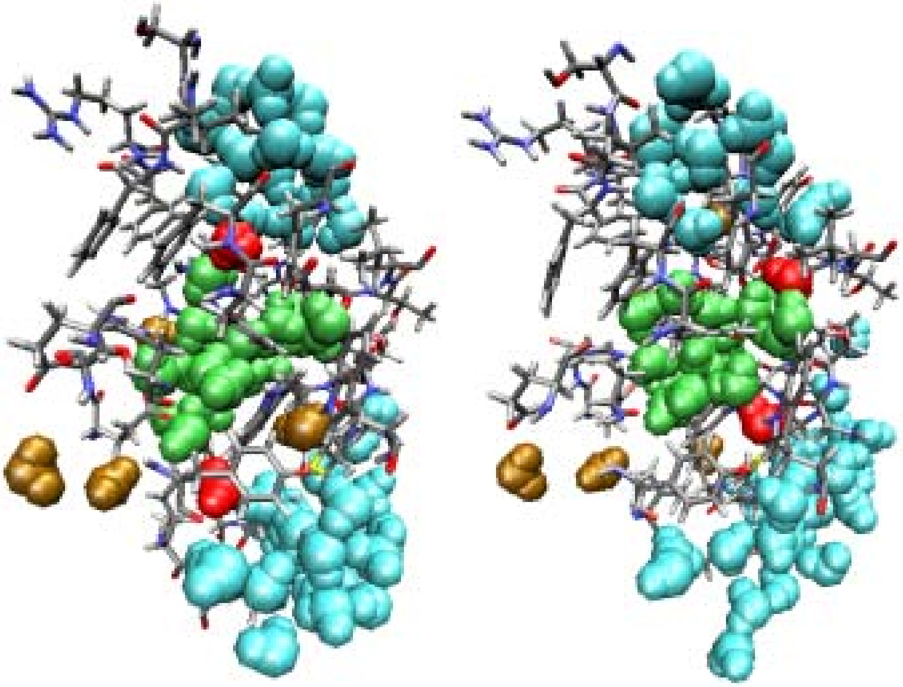

Fig. 7. Two structures showing the four paths into the gorge, indicated by the junction water molecules (red = junction, blue = bulk, green = gorge, brown = buried). Going clockwise from the top-left, they are the main gorge (i.e., front door 1), front door 2, back door 2, and back door 1.

these processes occurring simultaneously. Further study is required to determine which of the water molecules in the gorge are likely to be excluded, whether there is any relationship between fluctuations in gorge water population and the gorge proper radius, and how fluctuations in gorge water population may be contributing to the speed of AChE catalysis.

3 Brownian Dynamics of Protein-Protein Association

As evidenced in the previous section, molecular dynamics can be very useful in simulating the dynamics of proteins, as well as the interactions between the protein and its solvent environment. However, because of the amount of detail involved in these calculations, they are really limited to simulating processes that occur on the 10 ns to 100 ns timescale. For this reason, when we are interested in dynamics or interactions that occur on much longer time scales, we often turn to Brownian Dynamics (BD) simulations. As always, there must be some compensation between speed and accuracy, and BD simulations achieve their gain in speed through two simplifying measures: 1) the internal dynamics of the proteins are typically ignored and the structures are held rigid, and 2) the explicit solvent molecules are replaced with a continuum model. These assumptions allow us to study much larger proteins and to explore a wide variety of processes such as the kinetics of protein-protein association [20–26].

The basis of BD simulations is the Ermak-McCammon equation [27]

$$R(t+\Delta t) = R(t) + \frac{D\Delta t}{k_B T}F + S \tag{6}$$

where D is the diffusion constant, k_B is the Boltzmann constant and T is the temperature. In this equation, the position of the protein $R(t)$ changes due to forces and torque between the proteins (F) and stochastic interactions with the solvent (S). The forces between the molecules are usually only due to electrostatic interactions, found by solving the Poisson-Boltzmann equation [4, 5]. One of the major differences between BD and MD simulations in that the solvent is only included implicitly, dynamically through the stochastic term S, and by using a large value for the solvent dielectric (typically 78-80) when solving the Poisson-Boltzmann equation. By allowing the proteins to diffuse subject to these electrostatic and solvent forces, and by keeping count of the number of times they form a successful bound complex, we can determine the association rate constant (see [28] for details).

As we move to increasingly larger protein systems, the workload in calculating the electrostatic interactions between the proteins again becomes a limiting factor. To alleviate this restriction, some of our recent work has employed the ‘effective charge’ model of Gabdoulline and Wade [29]. Within this framework, the ‘true’ charges of the biomolecules are replaced with a simplified set of charges that reproduce the same electrostatic potentials. This method has proved very successful in the study of barnase and barstar [23], as well as two studies which we will highlight here.

3.1 AChE-fasciculin

We have already been introduced to the enzyme acetylcholinesterase (AChE) in the previous section, but the interruption of this enzyme by other proteins is the basis of many types of nerve agents and neurotoxins. One such protein is fasciculin, a 61 residue peptide from the venom of the green mamba, which binds to AChE with a very high affinity. It has been shown that the formation of the AChE-fasciculin complex is diffusion limited, and also that electrostatics plays an important role. These two factors make this an ideal system for study using BD techniques [24].

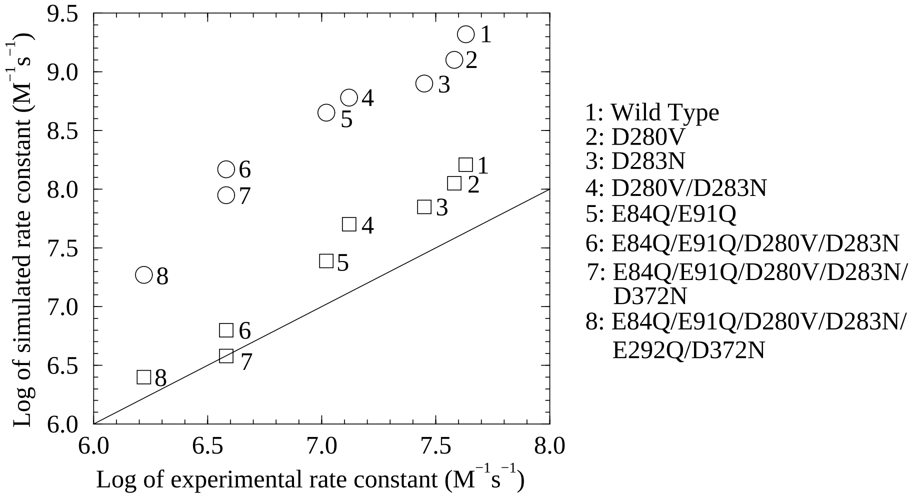

Fig. 8. Comparison between simulated and experimental rate constants for simulations performed only with electrostatic interactions (open circles) and with both electrostatic and desolvation effects (open squares).

The interest in performing simulations of AChE-fasciculin association is not simply to reproduce the experimentally measured rate constants, but to explain the binding kinetics of a number of mutants that have been produced. In these mutants, charged amino acids were replaced with neutral ones, thus altering the electrostatic interaction between the two proteins. These changes resulted in a decrease in the association rate constant, in some cases by almost two orders of magnitude. It is a true test of our simulation methods to see if this change can be explained simply on the basis of electrostatic interactions. Figure 8 shows a plot of the experimental rates versus the calculated rates for wild type fasciculin as well as six mutants. The agreement between the two data sets is quite reasonable, even in case of one mutant where a total of six residues were replaced.

From these results, it appears we can characterize the net effect of mutations on the association rate, however we can still make improvements to our model.

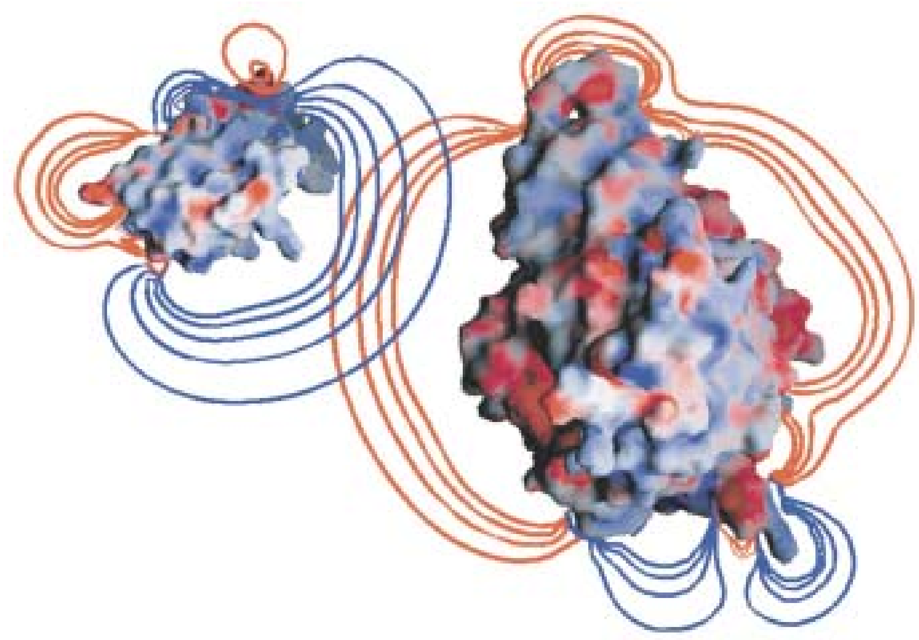

Fig. 9. Contours of the electrostatic potential of AChE (right) and fasciculin (left) showing the favorable electrostatic interactions.

Figure 9 shows the electrostatic complementarity between AChE and fasciculin. Despite these favorable interactions, more complete Poisson Boltzmann calculations show that the desolvation penalty incurred to bringing the two proteins together is larger than the gain from favorable electrostatics. Because of the general form of the Ermak-McCammon equation (6), we are not limited in adding additional forces, such as those arriving from desolvation, into the simulation (see [24] for details). When this is done, the absolute agreement between the experimental and calculated results is improved (Figure 8), as is the ionic strength dependence of the rate constant [24]. The general conclusion that can be drawn from these simulations is that electrostatic interactions alone can certainly account for the qualitative effects of mutation, but more accurate, quantitative results can be achieved through the addition of electrostatic desolvation effects.

3.2 Actin Polymerization

Another system of interest is the polymerization of actin filaments. Actin filaments are key components of the cell's cytoskeleton and are responsible for many cellular functions. What makes them so interesting is their polymerization properties, since one end of these polar filaments grows much faster than the other. Like the AChE-fasciculin system, electrostatics and diffusion control the binding, making this an ideal system to study using BD techniques.

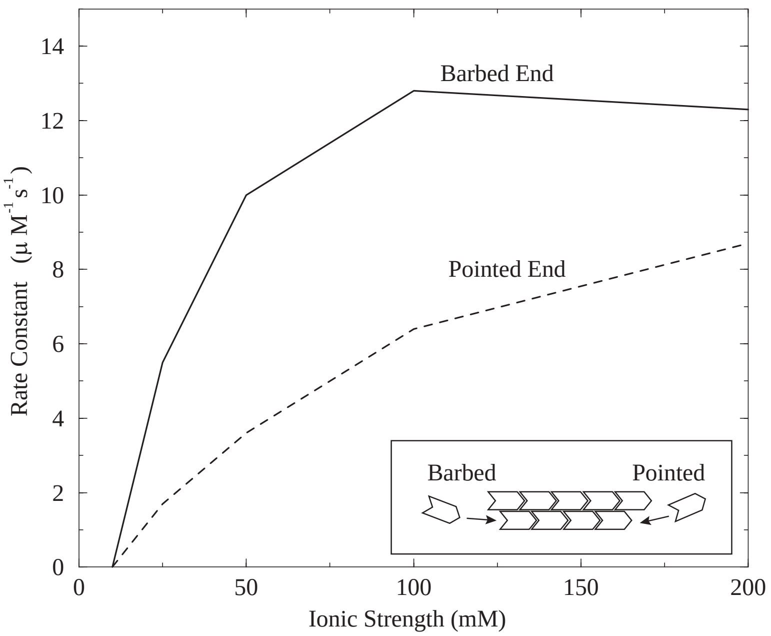

Fig. 10. The simulated association rate constants for actin polymerization where the results have been uniformly scaled so that the barbed end rate at 50 mM is equal to 10 $\mu\mathrm{M}^{-1}\mathrm{s}^{-1}$. The inset is meant to emphasize the fact that there are structural differences between the two ends of the filament.

Actin filaments are double-stranded helical polymers that are assembled from the 375-residue protein actin. The monomers attach in a "head-to-tail" fashion which results in the two ends of the filaments, the 'barbed' and 'pointed' ends, having different properties. The underlying reason for the difference in polymerization rates between the two ends was not known, and our goal was to try and determine the basis for this difference. To date there has been no crystal structure of the filament, however the crystal structure of the monomer has been docked into x-ray fiber data for actin filaments by Holmes et al. [30]. Using this structure as the basis for our study, we performed BD simulations to observe the effects of ionic strength on the polymerization rate, and to see if there was any difference in the polymerization rates between the barbed and pointed ends [25]. Figure 10 shows the polymerization rates at the two ends of the filaments as a function of ionic strength. Again we see that we are fairly successful in reproducing experimentally measured rates (data not shown), but more importantly, our simulations reproduced the asymmetry between the barbed and pointed end polymerization rates. In experiments, depending on the exact conditions, the barbed end typically grows about 10 times faster than the pointed end. In our simulations, this ratio is only about 3 to 4, but based on the fact that increasing the stringency in the binding criteria leads to a higher ratio, it seems electrostatic interactions account for a large portion of the polymerization differences (see [25] for more details).

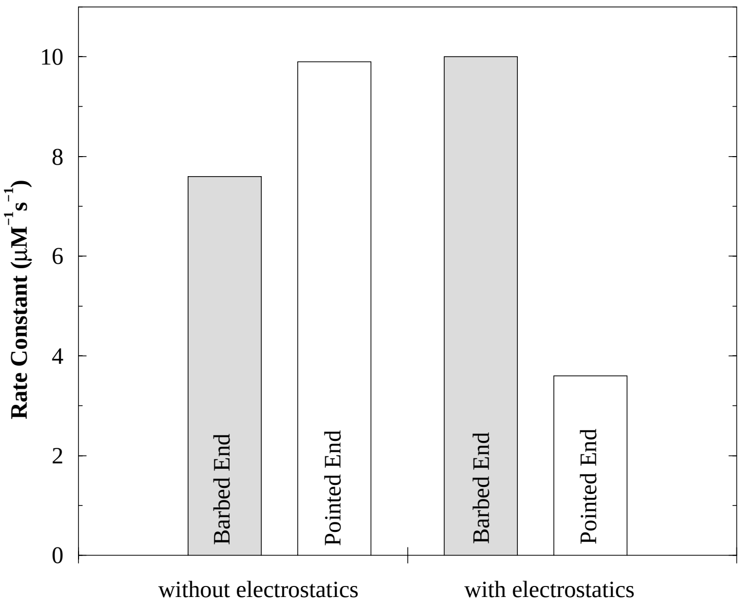

Fig. 11. Plot showing the effect of electrostatic interactions on the simulated association rate constants at the barbed and pointed ends of an actin filament.

A more interesting experiment is to compare simulations performed with and without electrostatic interactions. What we observe is that without the force term in Eq. 6, the pointed end, ie. the slow growing end, actually polymerizes faster than the barbed end (see Figure 11). Comparing these results with those calculated at 50 mM ionic strength, we see that the net effect of electrostatic interactions is to accelerate monomer addition at the fast growing barbed end, while at the same time decreasing the binding rate at the pointed end. This explains why the barbed end is observed to be diffusion limited and the pointed end is not, and it also indicates that asymmetry between the two ends has both a structural and electrostatic basis. The relative contributions of both factors could hopefully be resolved with an increase in the accuracy of both the structure of the actin filaments and the treatment of electrostatic interactions.

3.3 Future of Brownian Dynamics

There are two limitations that we face in using BD techniques: 1) we cannot routinely account or allow for protein flexibility, and 2) we are limited in the accuracy of the electrostatic interactions when the proteins are in close proximity. The first point needs to be carefully considered since it is by dealing with rigid protein structures that we can simulate for longer times than is possible using molecular dynamics. Perhaps by allowing some limited form of flexibility, we will someday be able to capture the effect of internal protein dynamics without incurring the workload of too many degrees of freedom. Overcoming the second limitation is much more feasible with the introduction of better and faster solvers for the Poisson-Boltzmann equation. One such method using adaptivity appears to be very promising, and we will discuss it next.

4 Parallel adaptive finite element solution of the Poisson-Boltzmann equation for large biomolecules

Both the thermodynamics and kinetics of a variety of biomolecular processes are strongly influenced by electrostatic interactions [31, 32]. However, due the long-range nature of these interactions, the accurate modeling of the contributions of solvent, counterions, and protein charges to the electrostatic field is often difficult and typically presents a computational bottleneck for a variety of molecular simulations. Continuum methods, such as the Poisson-Boltzmann equation (PBE), partially reduce the computational effort required to evaluate electrostatic interactions by approximating the explicit solvent as a polarization continuum and replacing discrete counterion charges by a continuous distribution. Despite these simplifications, current methods for the calculation of electrostatic properties from the PBE still require significant computational effort and typically do not scale well with increasing problem size [33].

Recently, new adaptive multilevel finite element methods have been introduced for the efficient solution of the PBE for large biomolecular systems [33–35]. Using these adaptive finite element algorithms as implemented in the FEtk (Finite Element toolkit) library [33, 36], a new software package APBS (Adaptive Poisson-Boltzmann Solver) [33, 34] has been developed for the modeling of electrostatic interactions in biomolecular systems. The APBS program provides parallel and sequential implementations of these multilevel adaptive finite element techniques to solve the linear and nonlinear versions of the PBE around biomolecules in ionic solutions. APBS makes extensive use of FEtk [36] for a variety of tasks, including the implementation of finite element mesh structures, refinement algorithms and data structures, assembly and solution of the linear and nonlinear equations, spectral bisection mesh partitioning, and residual-based error estimation. Because of the underlying hardware abstraction design of FEtk, the APBS code is designed for portability and can be used, with no modifications, on both single-processor workstations and massively parallel supercomputers. Information about obtaining APBS is available at http://wasabi.ucsd.edu/∼nbaker/pbe.html and information regarding FEtk and related tools is available from http://www.fetk.org/.

4.1 The Poisson-Boltzmann equation

The Poisson-Boltzmann equation is a second-order elliptic partial differential equation which describes the electrostatic potential around a fixed charge distribution in an ionic solution. More thorough reviews of this equation and its role in biological electrostatics calculations are presented by Davis and McCammon [31] and Sharp and Honig [37]. The three components of the solvated biomolecular system which must be considered to accurately model the electrostatic potential are the solute molecule, the solvent, and the solvated ions. The solute molecule is treated as a dielectric continuum of low polarizability (dielectric constant between 2 and 20) embedded in a dielectric medium of high polarizability (dielectric constant near 80) which represents the solvent. In most cases, the atomic charge distribution inside the biomolecule is represented by a collection of delta functions scaled by the atomic partial charges. Finally, the solvated ions surrounding the biomolecule are modeled according to a continuous Boltzmann distribution. The nonlinear PBE (NPBE) can be thought of as a combination of this Boltzmann distribution with the Poisson equation to give

$$-\nabla \cdot (\epsilon(x)\nabla u(x)) + \overline{\kappa}^2(x)\sinh\left(u(x)\right) = f(x), \quad u(\infty) = 0. \tag{7}$$

The linearized PBE (LPBE) arises from the NPBE by linearization of the hyperbolic sine term

$$-\nabla \cdot (\epsilon(x)\nabla u(x)) + \overline{\kappa}^2(x)u(x) = f(x), \quad u(\infty) = 0. \tag{8}$$

In both equations, the source term is a sum of delta functions,

$$f(x) = \frac{4\pi e_c^2}{kT}\sum_{i=1}^{N_m} z_i\delta(x - x_i). \tag{9}$$

In Eqs. 7 and 8, the variable $u(x) = e_c\phi(x)/kT$ represents a dimensionless electrostatic potential, $\epsilon(x)$ is the dielectric coefficient, $\overline{\kappa}^2$ is the Debye-Hückel screening parameter, which describes ion concentration and accessibility, kT is the thermal energy, e_c is the electron charge, N_m is the number of protein charges, z_i is the partial charge of each protein atom, and x_i is the position of each atom. Figure 12 shows a schematic of a solute (here taken to be a protein), ions, and solvent system modeled by the Poisson-Boltzmann equation. The dielectric coefficient ϵ changes by nearly two orders of magnitude across the protein-solvent boundary (defined by the molecular surface [38] shown as a solid line in Fig. 12) and the screening parameter jumps from zero to a value proportional to the bulk ionic strength across the "exterior" boundary (defined by the inflated van der Waals surface shown as a dashed line in Fig. 12).

As described here, the PBE equation contains three sources of discontinuities. Both the dielectric coefficient ϵ and the screening parameter $\overline{\kappa}^2$ have jump

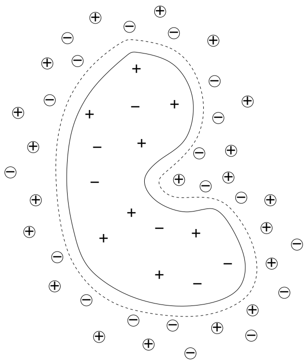

Fig. 12. Schematic of a model protein-solvent system. Charges within the protein are depicted as plus and minus symbols. The first protein-solvent boundary (solid line) represents discontinuities in the dielectric coefficient ϵ, while the second boundary (dashed line) represents discontinuities in the screening parameter $\overline{\kappa}^2$. Finally, the solvated ions surrounding the protein are depicted by the circled plus and minus symbols.

discontinuities over surfaces near near the protein-solvent interface. The final discontinuity is found in the source term $f(x)$, which models the point charges at the protein atoms by a sum of delta functions. While these jump and delta function discontinuities of coefficients in the PBE can pose serious numerical difficulties for traditional uniform or nonadaptive mesh partial differential equation solvers, these features can be efficiently described using adaptive finite element techniques [33, 34, 36].

4.2 Adaptive multilevel finite element methods

This section briefly describes the theory behind the parallel multilevel adaptive finite element scheme used to solve the PBE for the electrostatic potential around biomolecules. A much more detailed treatment of these methods and the corresponding theory can be found in texts by Axelsson and Barker [39] and Braess [40].

Finite element discretization In order to solve the PBE on a finite computational platform, the infinitely large problem domain implicit in Eqs. 7 and 8 must be truncated and the solution (the electrostatic potential) must be discretized into some basis supported on that domain. Finite element methods solve the equation inside a polygonal domain $\Omega \subset \mathbb{R}^3$ subject to some Dirichlet boundary condition, i.e., $u(x) = g$ on $\partial\Omega$, where $\partial\Omega$ denotes the boundary of Ω. To discretize the problem, the domain Ω is subdivided into tetrahedral simplices; this mesh forms the structure over which the function space $V_h = \text{span}\{v_i\}$ is defined in terms of the piecewise-polynomial basis functions $\{v_i\}$. Although a variety of polynomial orders are possible in such finite element bases, APBS currently uses the piecewise-linear finite element support provided by FEtk [36]. The solution to the PBE is then approximated in terms of this solution space V_h and a trace function $\overline{u}_h$ which satisfies the Dirichlet boundary conditions, such that $u_h \in \overline{u}_h + V_h$. Such a solution can be constructed by a linear combination of the finite element basis functions, $u_h(x) = \sum_i^N \alpha_i v_i(x)$.

In order for the construction of u_h from piecewise-linear functions to be successful, we must restate the PBE equations in their "weak" form. Clearly, the second derivative (as required by Eqs. 7 and 8) of a piecewise-linear function is not well defined. This difficulty can be overcome by integrating the PBE with a test function $\tilde{v}$ and applying integration by parts to the second-order differential term to give

$$\int_\Omega \left(\epsilon \nabla u \cdot \nabla \tilde{v} + \overline{\kappa}^2 \sinh(u)\tilde{v}\right) dx = \int_\Omega f\tilde{v} dx. \tag{10}$$

Equation 10 can also be written as

$$\langle F(u), \tilde{v}\rangle_{L^2(\Omega)} = \int_\Omega \left(\epsilon \nabla u \cdot \nabla \tilde{v} + \overline{\kappa}^2 \sinh(u)\tilde{v} - f\tilde{v}\right) dx = 0, \tag{11}$$

where $\langle \cdot, \cdot \rangle_{L^2(\Omega)}$ denotes the $L^2(\Omega)$ inner product and $F(u)$ is the strong form of the residual. This allows us to restate the PBE in weak form:

$$\text{Find } u_h \in \overline{u} + V_h \text{ such that } \langle F(u_h), v_i \rangle = 0 \text{ for all } v_i \in V_h. \tag{12}$$

This form of the PBE requires only one order of differentiation under an integral and is therefore a "weaker" formulation of the PBE than the original second-order differential equations (7 and 8). Although the above discussion used the NPBE, similar manipulations can be performed for the LPBE to produce an expression for the residual $F(u)$ which is linear in u.

Given u_h as a linear combination of the finite element basis functions and the above weak form of the PBE (12), we have a discretization of the partial differential equation suitable for numerical solution. In the case in which $F(u)$ is linear (LPBE), Eq. 12 explicitly defines a sparse matrix equation that can be solved using standard linear algebra methods or the multilevel methods described in Section 4.2. However, when $F(u)$ is nonlinear (NPBE), a damped inexact Newton iteration can be used in conjunction with the multilevel linear methods to solve the nonlinear equations [33, 41–45].

Adaptive refinement While these finite element methods can be used to determine the solution on a given finite element mesh, they do not provide information about the accuracy of the numerical solution u_h or indicate how it can be improved. The answers to these two questions lie within the domain of error estimation and adaptive refinement techniques. Again, only a cursory overview of this topic is presented here, with discussion of the *a posteriori* error estimation and adaptive refinement techniques that are applied to the PBE in the present work. For more detailed information about the implementation of these methods in the solution of the PBE, see Holst, Baker, and Wang [33]. *A posteriori* error estimation has been the subject of several publications [36, 46–50] which provide much more information about the theory and implementation of these methods.

Adaptive refinement methods typically employ error-estimation techniques to approximate the difference between the numerical and true solution $\|u - u_h\|_X$ (using some norm $\|\cdot\|_X$) and determine the regions of the problem domain where this error is large. The mesh is then refined in regions of excess error and the equation is re-solved to provide a more accurate finite element representation of the solution. This error-based refinement of the mesh can also be interpreted as the local enrichment of the finite element basis set in regions where the true solution is not adequately represented. In general, an *a posteriori* error estimator is used to determine the error in each simplex. Although simpler *a priori* or geometry-based error estimators can also be used, the reduction of error in the solution with each level of refinement is typically less efficient. APBS employs the residual-based *a posteriori* error estimation framework provided by FEtk which generates a per-simplex error estimate η_s in simplex s by using the residual defined by the strong form (Eqs. 7 and 8) of the PBE [33, 49]. An estimate of the global error over the problem domain is obtained as the root mean square of the per-simplex estimates $\eta_{\text{global}} = \left(\sum_i \eta_s^2\right)^{1/2}/N$. Although this η_{global} provides only an upper bound (within a constant) of the true error in the solution, it offers a practical measure for the reduction of error during solution of the PBE. Given a per-simplex error estimate, those simplices with errors above a certain tolerance η_{tol} are marked for subdivision. The iterative subdivision algorithms in FEtk [36] are used to refine the marked simplices, together with a few neighboring simplices needed for mesh conformity.

Multilevel solution The time required to solve the linear algebra equations, either within the Newton steps for NPBE or explicitly defined by the LPBE, generally controls the performance of solution methods for the PBE. Multilevel methods are well-established techniques for efficiently solving such equations through algebraic hierarchies [41, 51–56]. Such methods have been shown to give optimal (for uniformly refined meshes [57]) or nearly optimal (for adaptively refined meshes [58]) time and memory complexity for the solution of the linear matrix equations.

APBS employs the multilevel finite element solver technology in FEtk [36] to form an algebraic hierarchy of problems based on the refinement of the mesh [33, 36, 59]. Specifically, a prolongation operator P_k is constructed which relates basis functions on refinement levels k and $k+1$ of the finite element mesh. Given operator A_k on level k of the mesh, the prolongation operator P_k can be used to reconstruct the problems A_{k+1} from coarser levels of the mesh by applying P_k to the current problem A_k via $A_{k+1} = P_k^T A_k P_k$. Using this prolongation-based reconstruction, the problem can then be solved in a multilevel fashion, employing a direct solver for the problem on the coarsest level.

Parallel finite element methods Using the parallel refinement techniques of Bank and Holst [60], the methods described in the previous sections can be performed in a parallel fashion. In the parallel implementation, each of P processors is given the same initial mesh. Using the previously described sequential finite element techniques, each processor then solves the problem on this coarse mesh and generates an *a posteriori* error estimate for every simplex in the mesh. This error estimate is used to weight a partitioning which divides the mesh into P pieces of approximately equal error. Finally, each of these mesh partitions M_i is assigned to a different processor i. After partitioning, the usual solution and adaptive refinement methods of the previous sections are performed with only a small modification: When the per-simplex error estimates are calculated on processor i, only simplices within the local partition M_i and a boundary region of size σ surrounding M_i are given nonzero error estimates. The simplices with errors greater than a specified tolerance are marked for refinement, and these marked simplices (which are located only in M_i and the overlap) are subdivided. The mesh refinement algorithm then proceeds as usual, refining marked and any additional simplices (from any partition) required to ensure a conforming mesh. The initial error-based partitioning step acts as the load-balancing mechanism for this algorithm; the number of simplices in an error-based adaptive refinement is related to the total error in in the mesh region it is refining. By partitioning the mesh such that all processors have roughly the same amount of error, this algorithm provides a reasonable amount of *a priori* load balancing.

The overlap region surrounding each mesh partition is implemented by APBS in a simple fashion. Let x_i be the center of geometry of partition M_i, and let R_i be the radius of the sphere circumscribing M_i. The parameter $\sigma \geq 1$ is the desired relative size of the overlap region with respect to R_i. APBS then enforces parallel refinement with partition overlap by only allowing error-based simplex marking (on processor i) of simplices within a distance σR_i of the center x_i of partition M_i. A two-dimensional example of this method applied to a four processor system is shown in Fig. 13. In this example, all simplices within $\sigma = 1.2$ times the radius of the green partition were given the same error (which was chosen to be greater than the error-based marking tolerance). The resulting refinement over the green processor's partition and the overlap region is evident, as is the additional refinement outside the radius σR_i required for conformity.

As noted by Bank and Holst [60], this error-decoupling parallel algorithm essentially trades computation for communication. While the algorithm requires little or no communication between processors, it compensates by duplicating the computational effort spent in some portions of the solution algorithm. Specifically, partitioning steps of the mesh and computations on the the solution in overlap regions are duplicated across processors. Although the overlap region can be neglected for some problems [60], a nonzero overlap region proportional to the size of M_i must generally be implemented to satisfy the requirements underlying the decoupled error estimates [60, 61].

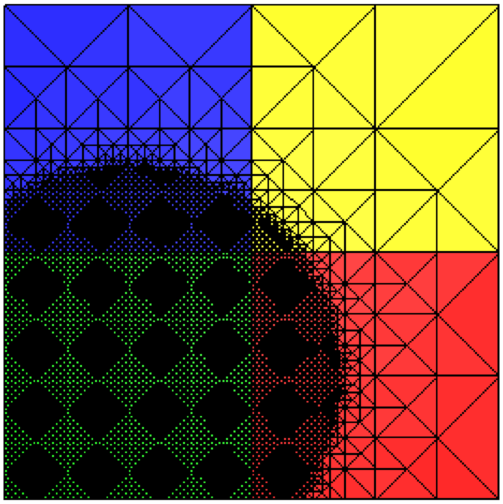

Fig. 13. Overlapping refinement on a partitioned mesh. All simplices within $\sigma = 1.2$ times the radius of the green partition were given the same error (greater than the simplex marking tolerance). The checkerboard pattern within the refined region is an artifact of the image, a Moiré pattern due to the high density of simplex edges.

4.3 Application to microtubules

One of the advantages of applying adaptive finite element methods to the PBE is the ability to study large biomolecular systems which would be inaccessible using uniform mesh techniques. One such system of interest is the cytoskeleton, the complex array of filaments and proteins within every eukaryotic cell. The largest cytoskeletal component, the microtubule, is a hollow cylindrical filament (see Fig. 14) assembled from the long protofilaments composed of tubulin subunits [62, 63]. The microtubule cylinders are 25 nm in diameter and, depending on function, can have lengths from nanometers to several millimeters. While microtubules are the most rigid structures in the cell and play an important structural role, they are also involved in variety of other functions, including cellular transport, motility, and division. Many of these more dynamic functions involve interactions with other proteins or filaments in the cell, often through electrostatic interactions. For this reason, the ability to calculate the electrostatic properties of a microtubule can provide important insight into many cellular processes. It is the large size of microtubules that poses tremendous computational challenges; for example, the atomically detailed solution of the PBE for a 1 μm long microtubule requires more than 21 million delta functions in the source term of the PBE to model the charge distribution to full atomic detail.

As described previously by Baker et al. [35], APBS was used to solve the LPBE for a 40 nm long microtubule consisting of 605 205 atoms with a net charge of $-1800e$. The microtubule structure was assembled by D. Sept using

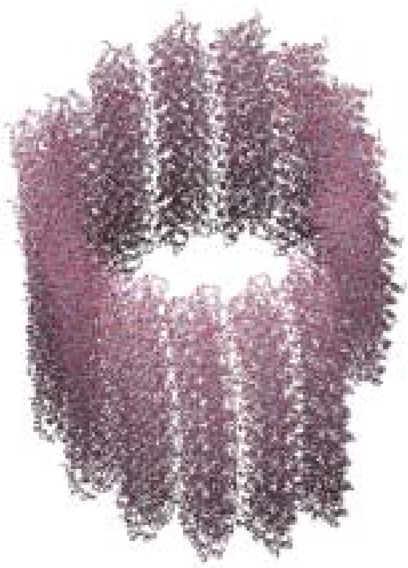

Fig. 14. The amino acid backbone atoms of a microtubule fragment. The structure shown here is 25 nm in diameter, 60 nm in length and has 901 083 atoms and a $-3000e$ charge.

microtubule structures derived from the work of Nogales, Whittaker, Milligan, and Downing [64]. The biomolecule was assigned an internal dielectric constant of 2 and surrounded by a solvent of dielectric 78.54 and ionic strength of 150 mM. The molecular volume was defined with 0.14 nm radius solvent probes, and the ion accessibility was calculated using 0.20 nm probes. For each P-processor calculation, the pre-mesh was uniformly refined to over $100P$ simplices and partitioned by error-weighted spectral bisection. Each partition in the mesh was then subjected to the solve-estimate-refine adaptive refinement loop using the residual-based *a posteriori* error estimator until each processor had the target number of vertices (40 000). These calculations were performed on 1, 2, 4, 8, 16, and 32 processors of the NPACI Blue Horizon supercomputer. Blue Horizon is a massively parallel computational platform with eight-way SMP IBM 222 MHz Power3 nodes and 4 gigabytes of RAM per node. Jobs requiring less than 32 processors were run with all 8 processors per node, giving each processor roughly 400 megabytes of heap memory. However, to provide adequate memory for the initial mesh refinement and partitioning steps, the 32-processor job was run with 4 processors per node, allowing approximately 800 megabytes of memory per processor.

The parallel efficiency of these calculations was very good. Although it was anticipated that each calculation would take roughly the same amount of execution time with the various processor configurations, the actual runs showed a decrease in the execution time with increasing number of processors. The *global* number of simplices in the mesh $L(P)$ (the sum of the number of simplices from each partition) was fit to a straight line $L(P) = l_1 P + l_0$ with correlation coefficient $r^2 = 0.999$, slope $l_1 = (2.05 \pm 0.03) \times 10^5$ simplices/processor, and intercept $l_0 = (1.7 \pm 0.6) \times 10^5$ simplices. Finally, the parallel efficiency was defined as $E(P) = L(P)/PL(1)$ to measure the increase in problem size at each processor configuration. The efficiency was also fit to a linear polynomial $E(P) = e_1 P + e_0$ with correlation coefficient $r^2 = 0.61$, slope $e_1 = (7 \pm 3) \times 10^{-3}$ per processor, and intercept $e_0 = (1.08 \pm 0.05)$. The mean efficiency of the six runs was $\overline{E} = 1.0 \pm 0.1$.

Due to the large size of the resulting electrostatic potential data sets, it was not possible to visualize the results of the above parallel calculations. However, a much lower resolution calculation was performed on a slightly larger (60 nm long, 901 083 atoms, $-3000e$ charge) microtubule to generate the electrostatic potential contours shown in Figure 15. As expected, the highly charged microtubule shows mostly negative electrostatic potential near the molecular surface (Figure 15, red contour). However, several regions of positive potential are visible, especially near the ends of the microtubule. Such localized variations in electrostatic potential often play important roles in molecular recognition and binding and suggest interesting modes of microtubule assembly and stability.

4.4 Outlook

New methods for the parallel solution of elliptic partial differential equations have made it possible to leverage the teraflops computing power of parallel computers to perform electrostatic calculations on biomolecular systems at scales approaching the cellular level. Using the APBS and FEtk software on the NPACI Blue Horizon supercomputer, it was possible to solve the Poisson-Boltzmann equation for the electrostatic potential around a microtubule containing more than 600 000 atoms. The code showed excellent parallel scaling, providing incentive to attempt further calculations to probe the structure and function of even larger macromolecular assemblages. Such calculations may also serve as a benchmark for parameterizing simpler methods for modeling large-scale biomolecular electrostatics.

Acknowledgements

N. Baker's work was supported by predoctoral fellowships from the Howard Hughes Medical Institute and the Burroughs Wellcome La Jolla Interfaces in Science program. N. Baker is indebted to Amit Majumdar, Giridhar Chukkapalli, and Greg Johnson at the San Diego Supercomputer Center for help with technical issues on the Blue Horizon and visualization of the large data sets.

K. Tai thanks Mr. Ulf Börjesson, Dr. Marios Philippopoulos, and Mr. Tongye Shen for many good ideas and fruitful discussions, Dr. Tjerk P. Straatsma for

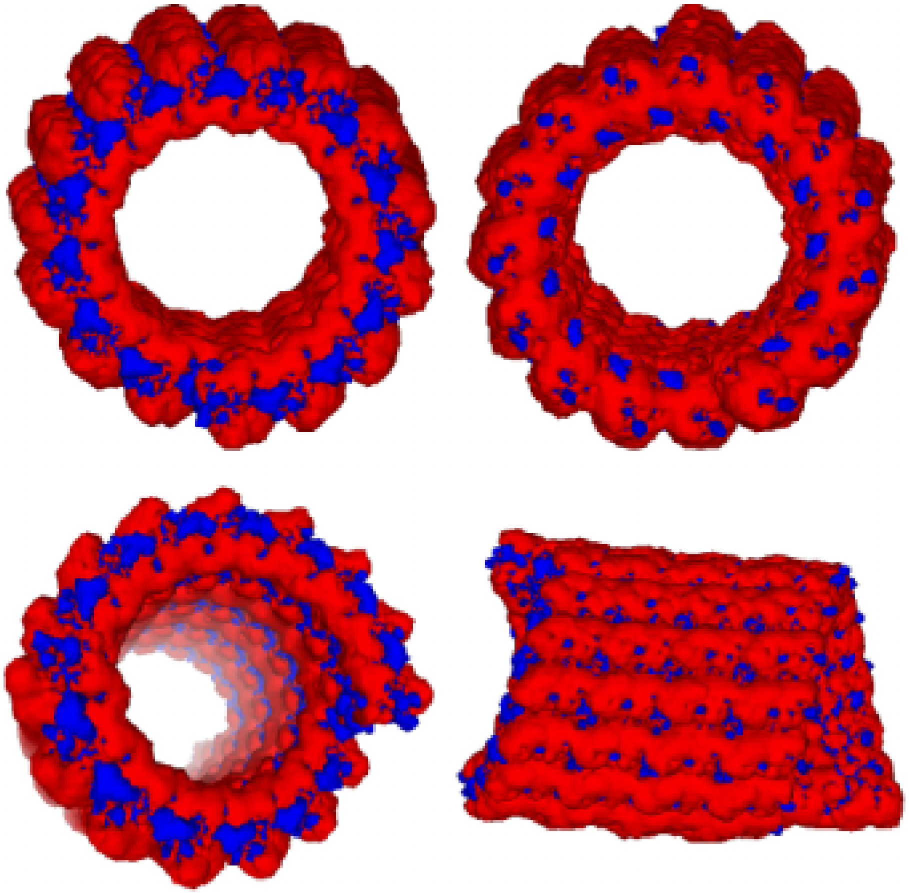

Fig. 15. Electrostatic potential of a 901 083-atom microtubule fragment at 150 mM ionic strength. Potential contours are shown at $+0.5kT/e$ (blue) and $-0.5kT/e$ (red). Each image represents a different view of the macromolecule. The upper images show the electrostatic complementarity at ends of the microtubule, while the lower images show the varying positive and negative regions of the potential on the exterior and interior of the protein.

providing and maintaining the NWChem software, and Dr. Sylvia Tara for setting up and equilibrating the MD simulation. Gratitude is expressed to Molecular Simulations, Inc., San Diego for generously providing us with the *Insight II* software. K. Tai also wishes to acknowledge the La Jolla Interfaces in Science interdisciplinary training program and the Burroughs Wellcome Fund for fellowship support.

M. Holst's work was supported in part by a UCSD Hellman Fellowship, and in part by NSF CAREER Award 9875856.

J.A. McCammon's work was supported in part by grants from NIH, NSF, and SDSC.

References

1. J.A. McCammon, B.R. Gelin, and M. Karplus. Dynamics of folded proteins. *Nature*, 267:585–590, 1977.
2. C.W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Englewood Cliffs, New Jersey, 1971.
3. D.L. Ermak and J.A. McCammon. Brownian dynamics with hydrodynamic interactions. *J. Chem. Phys.*, 69:1352–1360, 1978.
4. F.B. Sheinerman, R. Norel, and B. Honig. Electrostatic aspects of protein-protein interactions. *Curr. Opin. Struc. Biol.*, 10:153–159, 2000.
5. M.E. Davis and McCammon J.A. Solving the finite difference linearized poisson-boltzmann equation: A comparison of relaxation and conjugate gradient methods. *J. Comp. Chem.*, 10:386–391, 1989.
6. E.R. Kandel, J.H. Schwartz, and T.M. Jessell. *Principles of neural science*. Appleton & Lange, Norwalk, Connecticut, 1991.
7. Stanislaw T. Wlodek, Terry W. Clark, L. Ridgway Scott, and J. Andrew McCammon. Molecular dynamics of acetylcholinesterase dimer complexed with tacrine. *Journal of the American Chemical Society*, 119:9513–9522, 1997.
8. Huan-Xiang Zhou, Stanislaw T. Wlodek, and J. Andrew McCammon. Conformation gating as a mechanism for enzyme specificity. *Proceedings of the National Academy of Sciences of the United States of America*, 95:9280–9283, 1998.
9. M. K. Gilson, T. P. Straatsma, J. A. McCammon, D. R. Ripoll, C. H. Faerman, P. H. Axelsen, I. Silman, and J. L. Sussman. Open "back door" in a molecular dynamics simulation of acetylcholinesterase. *Science*, 263:1276–1278, 1994.
10. Sylvia Tara, T. P. Straatsma, and J. Andrew McCammon. Mouse acetylcholinesterase unliganded and in complex with huperzine A: a comparison of molecular dynamics simulations. *Biopolymers*, 50:35–43, 1999.
11. Wendy D. Cornell, Piotr Cieplak, Christopher I. Bayly, Ian R. Gould, Kenneth M. Merz, Jr., David M. Ferguson, David C. Spellmeyer, Thomas Fox, James W. Caldwell, and Peter A. Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *The Journal of the American Chemical Society*, 117:5179–5197, 1995.
12. T. P. Straatsma, M. Philippopoulos, and J.A. McCammon. NWChem: Exploiting parallelism in molecular simulation. *Comp. Phys. Commun.*, 128:377–385, 2000.
13. J.A. McCammon and S.C. Harvey. *Dynamics of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, 1987.

14. Charles L. Brooks, III, Martin Karplus, and B. Montgomery Pettitt. *Proteins: a theoretical perspective of dynamics, structure, and thermodynamics*, volume LXXI of *Wiley Series on Advances in Chemical Physics*. Wiley-Interscience, New York, 1988.
15. M. Karplus and J.A. McCammon. The internal dynamics of globular proteins. *CRC Crit. Revs. in Biochem.*, 9:293–349, 1981.
16. Y. Bourne, P. Taylor, and P. Marchot. Acetylcholinesterase inhibition by fasciculin: crystal structure of the complex. *Cell*, 83:503–512, 1995.
17. Angel E. García. Large-amplitude nonlinear motions in proteins. *Physical Review Letters*, 68:2696–2699, 1992.
18. T. Y. Shen, Kaihsu Tai, and J. Andrew McCammon. Statistical analysis of the fractal gating motions of the enzyme acetylcholinesterase. *Physical Review E*, 63:041902, 2001.
19. D. M. Quinn. Acetylcholinesterase: Enzyme structure, reaction dynamics, and virtual transition states. *Chem. Rev.*, 87:955–979, 1987.
20. P. Nambi, A. Wierzbicki, and S.A. Allison. Intermolecular interaction between bovine pancreatic trypsin inhibitor moecules probed by brownian dynamics simulation. *J. Phys. Chem.*, 95:9595–9600, 1991.
21. S.H. Northrup, K.A. Thomasson, C.M. Miller, P.D. Barker, L.D. Eltis, J.G. Guillemette, S.C. Inglis, and A.G. Mauk. Effects of charged amino acid mutations on the biomolecular kinetics of reduction of yeast iso-1-ferricytochrome c by bovine ferrocytochrome b_5. *Biochem.*, 32:6613–6623, 1993.
22. R.E. Kozack, M.J. d'Mello, and S. Subramanium. Computer modeling of electrostatic steering and orientational effects in antibody-antigen association. *Biophys. J.*, 68:807–814, 1995.
23. R.R. Gabdoulline and R.C. Wade. Simulation of the diffusional association of barnase and barstar. *Biophys. J.*, 72:1917–1929, 1997.
24. A.H. Elcock, R.R. Gabdoulline, R.C. Wade, and J.A. McCammon. Computer simulation of protein-protein association kinetics: Acetylcholinesterase-Fasciculin. *J. Mol. Biol.*, 291:149–162, 1999.
25. D. Sept, A. H. Elcock, and J. A. McCammon. Computer simulations of actin polymerization can explain the barbed-pointed end assymetry. *J. Mol. Biol.*, 294:1181–1189, 1999.
26. A. H. Elcock, D. Sept, and J. A. McCammon. Computer simulation of protein-protein interactions. *J. Phys. Chem.*, in press.
27. D. L. Ermak and J. A. McCammon. Brownian dynamics with hydrodynamic interactions. *J. Chem. Phys.*, 69:1352–1360, 1978.
28. S.H. Northrup, S.A. Allison, and J.A. McCammon. Brownian dynamics simulations of diffusion-influenced biomolecular reactions. *J. Chem. Phys.*, 80:1517–1524, 1984.
29. R.R. Gabdoulline and R.C. Wade. Effective charges for macromolecules in solvent. *J. Phys. Chem.*, 100:3868–3878, 1996.
30. K.C. Holmes, D. Popp, W. Gebhard, and W. Kabsch. Atomic model of the actin filament. *Nature*, 347:44–49, 1990.
31. M. E. Davis and J. A. McCammon. Electrostatics in biomolecular structure and dynamics. *Chem. Rev.*, 94:7684–7692, 1990.
32. B. Honig and A. Nicholls. Classical electrostatics in biology and chemistry. *Science*, 268:1144–1149, 1995.
33. M. J. Holst, N. A. Baker, and F. Wang. Adaptive multilevel finite element solution of the Poisson-Boltzmann equation I: algorithms and examples. *J. Comput. Chem.*, 21:1319–1342, 2000.

34. N. A. Baker, M. J. Holst, and Wang. F. Adaptive multilevel finite element solution of the Poisson-Boltzmann equation II: refinement at solvent accessible surfaces in biomolecular systems. *J. Comput. Chem.*, 21:1343–1352, 2000.
35. N. A. Baker, D. Sept, M. J. Holst, and J. A. McCammon. The adaptive multilevel finite element solution of the Poisson-Boltzmann equation on massively parallel computers. *IBM Journal of Research and Development*, in press.
36. Michael J. Holst. *Adaptive multilevel finite element methods on manifolds and their implementation in MC.* (In preparation; currently available as a UCSD Dept. of Mathematics technical report and user's guide to the MC software).
37. K. A. Sharp and B. Honig. Calculating total electrostatic energies with the nonlinear Poisson-Boltzmann equation. *J. Phys. Chem.*, 94:7684–7692, 1990.
38. B. Lee and F. M. Richards. The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol.*, 55:379–400, 1971.
39. O. Axelsson and V. A. Barker. *Finite element solution of boundary value problems. Theory and computation.* Academic Press, San Diego, CA, 1984.
40. D. Braess. *Finite elements. Theory, fast solvers, and applications in solid mechanics.* Cambridge Univ. Press, New York, 1997.
41. M. Holst and F. Saied. Numerical solution of the nonlinear Possion-Boltzmann equation: developing more robust and efficient methods. *J. Comput. Chem.*, 16:337–364, 1995.
42. R. E. Bank and R. K. Smith. Parameter selection for Newton-like methods applicable to nonlinear partial differential equations. *SIAM J. Numer. Anal.*, 17:806–822, 1980.
43. R. E. Bank and R. K. Smith. Global approximate Newton methods. *Numer. Math.*, 37:279–295, 1981.
44. R. E. Bank and R. K. Smith. Analysis of a multilevel iterative method for nonlinear finite element equations. *Math. Comp.*, 39:453–465, 1982.
45. R. S Dembo, S. C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19:400–408, 1982.
46. M. Holst and D. Bernstein. Adaptive finite element solution of the initial-value problem in general relativity: Theory and algorithms. *Comm. Math. Phys.*, submitted.
47. I. Babuška and W. C. Rheinboldt. Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.*, 15:736–754, 1978.
48. I. Babuška and W. C. Rheinboldt. A posteriori error estimates for the finite element method. *Int. J. Numer. Meth. Engrg.*, 12:1597–1615, 1978.
49. R. Verfürth. A posteriori error estimates for nonlinear problems. Finite element discretization of elliptic equations. *Math. Comp.*, 62:445–475, 1994.
50. R. Verfürth. *A review of a posteriori error estimation and adaptive mesh-refinement techniques.* John Wiley, New York, 1996.
51. R. E. Bank and J. Xu. The hierarchical basis multigrid method and incomplete LU decomposition. In D. Keyes and J. Xu, editors, *Seventh international symposium on domain decomposition methods for partial differential equations*, pages 163–173. AMS, 1994.
52. A. Brandt. Algebraic multigrid theory: the symmetric case. *Appl. Math. Comp.*, 19:23–56, 1986.
53. A. Brandt, S. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In D. J. Evans, editor, *Sparsity and its applications.* Cambridge Univ. Press, 1984.
54. M. Holst and F. Saied. Multigrid solution of the Poisson-Boltzmann equation. *J. Comput. Chem.*, 14:105–113, 1993.

55. J. W. Ruge and K. Stüben. Algebraic multigrid (AMG). In S. F. McCormick, editor, *Multigrid methods*, volume 3 of *Frontiers in applied mathematics*, pages 73–130. SIAM, Philadelphia, 1987.
56. W. Hackbusch. *Multi-grid methods and applications.* Springer-Verlag, Berlin, 1985.
57. J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34:581–613, 1992.
58. R. E. Bank, T. F. Dupont, and H. Yserentant. The hierarchical basis multigrid method. *Numer. Math.*, 52:427–458, 1988.
59. M. Holst and S. Vandewalle. Schwarz methods: to symmetrize or not to symmetrize. *SIAM J. Numer. Anal*, 34:699–722, 1997.
60. R. Bank and M. Holst. A new paradigm for parallel adaptive meshing algorithms. *SIAM J. Sci. Comput.*, in press.
61. Jinchao Xu and Aihui Zhou. Local and parallel finite element algorithms based on two-grid discretizations. *Math. Comp.*, 69:881–909, 2000.
62. P. Dustin. *Microtubules.* Springer-Verlag, Berlin, 1984.
63. B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson. *Molecular biology of the cell.* Garland Publishing, New York, 1994.
64. E. Nogales, M. Whittaker, R. A. Milligan, and K. H. Downing. High-resolution model of the microtubule. *Cell*, 96:79–88, 1999.