[5] Poisson–Boltzmann Methods for Biomolecular Electrostatics

By NATHAN A. BAKER

Introduction

The understanding of electrostatic properties is a basic aspect of the investigation of biomolecular processes. Structures of proteins and other biopolymers are being determined at an increasing rate through structural genomics and other efforts; furthermore, the specific roles of these biopolymers in cellular pathways and supramolecular assemblages are being detected by genetic and other experimental efforts. The integration of this information into physical models for drug discovery or other applications requires the ability to evaluate intra- and interbiomolecular energetics. Among the various components of molecular interactions, electrostatic energetics and forces are of special importance because of their long range and the substantial charges of amino and nucleic acids.

Because of the ubiquitous nature of electrostatic interactions in biomolecular systems, a variety of computational methods have been developed for elucidating these interactions (see Refs. 1–5 and references therein). Popular computational electrostatics methods for biomolecular systems can be loosely grouped into two categories: ''explicit solvent'' methods, which treat the solvent in full molecular detail, and ''implicit solvent'' methods, which include solvent influences in averaged or continuum fashion. Although explicit solvent approaches offer more detailed insight into solvent-mediated biomolecular interactions, the necessity to integrate over the numerous solvent degrees of freedom often limits the ability of these methods to calculate thermodynamic quantities for large biomolecular systems.

[1] N. A. Baker and J. A. McCammon, *in* ''Structural Bioinformatics'' (P. Bourne and H. Weissig, eds.), John Wiley & Sons, New York, 2002.
[2] M. Gilson, *in* ''Biophysics Textbook Online'' (D. A. Beard, ed.), Biophysical Society Bethesda; T. A. Darden, *in* ''Computational Biochemistry and Biophysics'' (O. M. Becker, A. D. J. MacKerell, B. Roux, and M. Watanabe, eds.), pp. 91–114. Marcel Dekker, New York, 2001.
[3] B. Roux, *in* ''Computational Biochemistry and Biophysics'' (O. M. Becker, A. D. J. MacKerell, B. Roux, and M. Watanabe, eds.), pp. 133–152. Marcel Dekker, New York, 2001.
[4] M. E. Davis and J. A. McCammon, *Chem. Rev*. **94,** 7684 (1990).
[5] B. Honig and A. Nicholls, *Science* **268,** 1144 (1995).

0076-6879/04 $35.00

Brief Overview of Implicit Solvent Methods

Because of the sampling issues associated with explicit solvent treatments, implicit solvent methods have gained increasing popularity for elucidating the electrostatic properties of biomolecules in solution (see Refs. 1–5). As their name implies, these methods implicitly average over the configuration space of solvent and counterion species surrounding the biomolecule. The result is a polarizable continuum representation for the solvent and a mean field charge ''cloud'' for the counterion distribution. Despite some artifiacts arising from this continuum treatment (see Roux[3] and references therein), implicit solvent methods offer a significant advantage over traditional explicit solvent approaches and have become standard techniques for investigating the energetics and dynamics of biomolecular systems.

The importance of electrostatic interactions in protein behavior was recognized decades ago in work by Linderstrom-Lang[6] in developing protein titration models; Tanford and Kirkwood[7] in investigating the effects of pH and ionic strength on enzyme activity; and Flanagan *et al.*[8] in studying the energetics of dimer–tetramer assembly in hemoglobin. However, electrostatic models for protein systems were improved dramatically in 1982 by Warwicker and Watson.[9] Drawing on the increased knowledge of the three-dimensional structure of proteins increased computer power; they introduced a grid-based, finite difference approach for calculating the electrostatic potential of a nonspherical protein by solving the Poisson–Boltzmann (PB) equation, a nonlinear partial differential equation that incorporates detailed information about the biomolecular shape and charge distribution. Since this pioneering work, the PB equation has become a standard method for the detailed investigation of biomolecular electrostatics.[1,4,5,10]

In addition to PB methods, simpler approximate models have also been constructed for continuum electrostatics, including distance-dependent dielectric functions,[11] analytic continuum methods,[12] and generalized Born

[6] K. Linderstrom-Lang, *C. R. Travl. Lab. Carlsberg* **15** (1924).
[7] C. Tanford and J. G. Kirkwood, *J. Am. Chem. Soc.* **79,** 5333 (1957).
[8] M. A. Flanagan, G. K. Ackers, J. B. Matthew, G. I. H. Hanania, and F. R. N. Gurd, *Biochemistry* **20,** 7439 (1981).
[9] J. Warwicker and H. C. Watson, *J. Mol. Biol.* **157,** 671 (1982).
[10] C. Holm, P. Kekicheff, and R. Podgornik, eds. *in* ''Electrostatic Effects in Soft Matter and Biophysics,'' Vol. 46. NATO Science Series. Kluwer Academic, Boston, 2001.
[11] A. D. J. MacKerell and L. Nilsson, *in* ''Computational Biochemistry and Biophysics'' (O. M. Becker, A. D. J. MacKerell, B. Roux, and M. Watanabe, eds.), Marcel Dekker, New York, 2001; A. R. Leach, *in* ''Molecular Modelling: Principles and Applications,'' 2nd Ed., pp. xxiv and 744, and plate 16. Prentice Hall, New York, 2001.

models.[13–15] Among the most popular of these simpler methods is the generalized Born model, introduced by Still *et al*. in 1990[13] and refined by several other researchers.[14,15] This method is based on the Born ion, a canonical electrostatics model problem describing the electrostatic potential and solvation energy of a spherical ion.[16] The generalized Born method uses an analytical expression based on the Born ion model to approximate the electrostatic potential and solvation energy of small molecules. Although it fails to capture all the details of molecular structure and ion distributions provided by more rigorous models,[15,17,18] such as the Poisson–Boltzmann equation, it has gained popularity as a rapid method for evaluating approximate forces and energies for solvated molecules and continues to be vigorously developed.

Poisson–Boltzmann Methods for Biomolecular Electrostatics

As mentioned above, the Poisson–Boltzmann equation is derived from a continuum model of the solvent and counterion environment surrounding a biomolecule.[1,4,5,10] Although there are numerous derivations of the PB equation based on statistical mechanics (see Holm *et al*.[10] for review), the simplest begins with Poisson's equation[19]:

$$-\nabla \cdot \varepsilon(x)\nabla\phi(x) = \rho(x) \quad \text{for} \quad x \in \Omega \qquad \text{where} \quad \phi(x) = g(x) \quad \text{for} \quad x \in \partial\Omega \tag{1}$$

the canonical equation for describing the dimensionless electrostatic potential $\phi(x)$ generated by a charge distribution $\rho(x)$ in a polarizable continuum with dielectric constant $\varepsilon(x)$. Equation (1) is generally solved in some finite domain Ω with a fixed potential (Dirichlet boundary condition) $g(x)$ on the boundary $\partial\Omega$. In a biomolecular system, it is useful to consider two types of charge distributions. First, the partial atomic charges are typically lumped into a ''fixed'' charge distribution:

[12] M. Schaefer and M. Karplus, *J. Phys. Chem*. **100,** 1578 (1996).
[13] W. C. Still, A. Tempczyk, R. C. Hawley, and T. Hendrickson, *J. Am. Chem. Soc*. **112,** 6127 (1990).
[14] B. N. Dominy and C. L. Brooks, *J. Phys. Chem. B* **103,** 3765 (1999); D. Bashford and D. A. Case, *Annu. Rev. Phys. Chem*. **51,** 129 (2000); K. Osapay, W. S. Young, D. Bashford, C. L. Brooks, and D. A. Case, *J. Phys. Chem*. **100,** 2698 (1996).
[15] A. Onufriev, D. A. Case, and D. Bashford, *J. Comput. Chem*. **23,** 1297 (2002).
[16] M. Born, *Z. Phys*. **1,** 45 (1920).
[17] L. David, R. Luo, and M. K. Gilson, *J. Comput. Chem*. **21,** 295 (2000).
[18] R. Luo, M. S. Head, J. Moult, and M. K. Gilson, *J. Am. Chem. Soc*. **120,** 6138 (1998); J. A. Given and M. K. Gilson, *Proteins Struct. Funct. Genet*. **33,** 475 (1998).
[19] J. D. Jackson, *in* ''Classical Electrodynamics,'' 2nd Ed. John Wiley & Sons, New York, 1975.

$$\rho_{\mathrm{f}}(x) = \frac{4\pi e_{\mathrm{c}}^2}{kT} \sum_{i=1}^{M} Q_i \delta(x - x_i) \tag{2}$$

which models the M atomic partial charges of the biomolecule as delta functions $\delta(x - x_i)$ located at the atom centers $\{x_i\}$ with magnitudes $\{Q_i\}$. The scaling coefficients ensure the dimensionless form of the potential and include e_{c}, the charge of an electron, and kT, the thermal energy of the system. Second, the contributions of counterions are modeled in a continuous (or ''mean field'') fashion by a Boltzmann distribution, giving rise to the ''mobile'' charge distribution

$$\rho_{\mathrm{m}}(x) = \frac{4\pi e_{\mathrm{c}}^2}{kT} \sum_{j}^{m} c_j q_j \exp\left[-q_j \phi(x) - V_j(x)\right] \tag{3}$$

for m counterion species with charges $\{q_j\}$, bulk concentrations $\{c_j\}$, and steric potentials $\{V_j\}$ (i.e., potentials that prevent biomolecule–counterion overlap). In the case of a one-to-one electrolyte such as NaCl, Eq. (3) reduces to

$$\rho_{\mathrm{m}}(x) = \overline{\kappa}^2(x) \sinh \phi(x) \tag{4}$$

where the coefficient $\overline{\kappa}^2(x)$ describes both ion accessibility (indirectly via $\exp[-V(x)]$) and bulk ionic strength. Combining the expressions for the fixed [Eq. (2)] and mobile [Eq. (4)] counterion distributions with Poisson's equation [Eq. (1)] gives the Poisson–Boltzmann equation for a one-to-one electrolyte:

$$-\nabla \cdot \varepsilon(x) \nabla \phi(x) + \overline{\kappa}^2(x) \sinh \phi(x) = \frac{4\pi e_{\mathrm{c}}^2}{kT} \sum_{i} q_i \delta(x - x_i)$$
$$\text{for} \quad x \in \Omega; \quad \text{where} \quad \phi(x) = g(x) \quad \text{for} \quad x \in \partial\Omega \tag{5}$$

As described above, details of the biomolecular structure enter into the coefficients of the PB equation (see Fig. 1). Most obviously, the atom locations appear in the delta functions as the set of points $\{x_i\}$; the atomic positions and radii also enter into the definitions of coefficients $\varepsilon(x)$ and $\overline{\kappa}^2(x)$. The dielectric function $\varepsilon(x)$ has been represented by a variety of models, including discontinuous transitions at the molecular surface,[20,21] smooth spline-based definitions,[22,23] and Gaussian-based descriptions.[24]

[20] B. Lee and F. M. Richards, *J. Mol. Biol.* **55,** 379 (1971).
[21] M. L. Connolly, *J. Mol. Graphics* **11,** 139 (1993).
[22] W. Im, D. Beglov, and B. Roux, *Comput. Phys. Commun.* **111,** 59 (1998).
[23] C. L. Bajaj, V. Pasucci, R. J. Holt, and A. N. Netravali, *in* ''Fourth Issue of the Special Series of Discrete Applied Mathematics on Computational Biology,'' 2001.

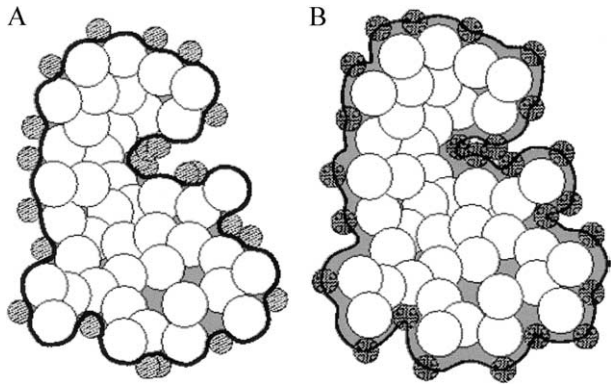

FIG. 1. Surface definitions used to evaluate coefficients of PB equation. (A) Representation of the molecular surface, a popular definition for the dielectric coefficient $\varepsilon(x)$. (B) Representation of the inflated van der Waals surface, the common definition for the ion accessibility coefficient $\overline{\kappa}^2(x)$.

Among these various dielectric coefficient models, the most popular have been definitions based on the molecular[21] and Lee–Richards surfaces.[20] In these models, $\varepsilon(x)$ is discontinuous along the biomolecular surface and assumes solute dielectric values inside and bulk solvent values outside the surface. The ion accessibility function $\overline{\kappa}^2(x)$ has typically been characterized in terms of the inflated van der Waals surface: the boundary of the union of spheres centered at the atomic positions with radii equal to the atomic van der Waals radii plus the counterion species radius.

The ''full'' or nonlinear form of the problem given in Eq. (5) is often simplified to the linearized PB equation by replacing the $\sinh\phi(x)$ term with its first-order approximation, $\sinh\phi(x) \approx \phi(x)$, to give

$$-\nabla \cdot \varepsilon(x)\nabla\phi(x) + \overline{\kappa}^2(x)\varphi(x) = \frac{4\pi e_c^2}{kT}\sum_i q_i\delta(x - x_i) \quad \text{for}$$
$$x \in \Omega \qquad \text{where} \quad \phi(x) = g(x) \quad \text{for} \quad x \in \partial\Omega \tag{6}$$

However, although the linearized version of the problem can be somewhat simpler to solve, it is only as useful as the underlying approximation $\sinh\phi(x) \approx \phi(x)$. Specifically, this linearization is appropriate only at small potential values where the nonlinear contributions to $\sinh\phi(x)$ are negligible.

It is worthwhile to note that the PB equation (nonlinear or linearized) is an approximate theory and therefore cannot be applied blindly to all

[24] J. A. Grant, B. T. Pickup, and A. Nicholls, *J. Comput. Chem.* **22,** 608 (2001).

biomolecular systems. Specifically, the PB equation is derived from mean field or saddle point treatments of the electrolyte system and therefore neglects counterion correlations and fluctuations that can affect the energetics of highly charged biomolecular systems such as DNA, RNA, and some protein systems. A good review of the scope and impact of these deviations from PB theory can be found in Holm *et al.*[10] In short, Poisson–Boltzmann theory gives reasonable quantitative results for biomolecules with low linear charge density in monovalent symmetric salt solutions; however, PB theory can be qualitatively incorrect for highly charged biomolecules or more concentrated multivalent solutions. Therefore, application of PB theory and software requires some discretion on the part of the user.

Since the PB equation was first applied to biomolecular systems, methods for the solution of the PB equation have been repeatedly revisited and refined to further improve the efficiency of electrostatics calculations, including improved finite difference, finite element, and boundary element methods; as well as a host of new and/or improved algorithms for using PB data in energy and force evaluations, pK_a calculations, and biomolecular simulations. The following sections describe the various aspects of these methods for obtaining and using solutions to the PB equation.

Numerical Solution of Poisson–Boltzmann Equation

Few analytical solutions of the PB equation exist for realistic biomolecular geometries and charge distributions. Therefore, this equation is usually solved numerically by a variety of computational methods. These methods typically rely on a discretization to project the continuous solution down onto a finite dimensional set of basis functions. In the case of the linearized PB equation [see Eq. (6)], the resulting equations take the usual linear matrix vector form, which can be solved directly. However, the nonlinear equations obtained from the full PB equation require more specialized techniques, such as Newton methods, to determine the solution to the discretized algebraic equation.[25] Specifically, Newton methods start with an initial solution guess and iteratively improve this guess by solving related linear equations for corrections to the current solution. The remainder of this section describes the most common discretization and solver methods used for both linearized and nonlinear PB models.

[25] M. J. Holst and F. Saied, *J. Comput. Chem.* **16,** 337 (1995).

Finite Difference Discretization

Some of the most popular discretization techniques employ Cartesian meshes to subdivide the domain in which the PB equation is to be solved. Of these, the finite difference (FD) method has been at the forefront of PB equation solvers (see Refs. 5, 26–28, and references therein). In its most general form, the finite difference method solves the PB equation on a non-uniform Cartesian mesh, as shown in Fig. 2A for a two-dimensional domain. In this general setting, the Laplacian or Poisson differential operator is transformed into a sparse difference matrix by means of a Taylor expansion. The resulting matrix equations are then solved (either directly or over the course of a Newton iteration) by a variety of linear algebra techniques. Although FD grids offer relatively simple problem setup, they provide little control over how unknowns are placed in the solution domain. Specifically, as shown by Fig. 2A, the Cartesian or tensor-product nature of the mesh makes it impossible to locally increase the accuracy of

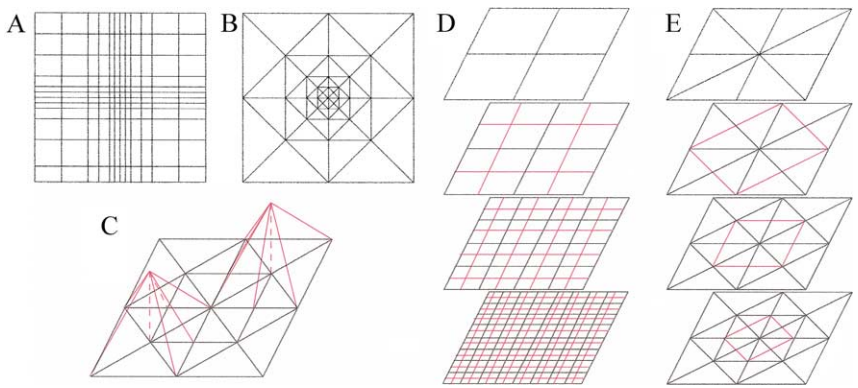

FIG. 2. Discretization schemes and hierarchies used in Poisson–Boltzmann (PB) solvers. (A) Cartesian mesh suitable for finite difference (FD) calculations. (B) Finite element (FE) mesh exhibiting adaptive refinement. (C) Examples of typical piecewise linear basis functions used in FE methods. (D) The multilevel hierarchy used to solve the PB equation for an FD discretization; red lines denote the additional unknowns added at each level of the hierarchy. (E) The multilevel hierarchy used to solve the PB equation for FE discretizations; red lines denote the simplex subdivisions used to introduce additional unknowns at each level of the hierarchy.

[26] M. E. Davis, J. D. Madura, B. A. Luty, and J. A. McCammon, *Comput. Phys. Commun.* **62,** 187 (1991).
[27] W. Rocchia, S. Sridharan, A. Nicholls, E. Alexov, A. Chiabrera, and B. Honig, *J. Comput. Chem.* **23,** 128 (2002).
[28] M. E. Davis and J. A. McCammon, *J. Comput. Chem.* **12,** 909 (1991).

the solution in a specific region without increasing the number of unknowns across the entire grid.

Adaptive Finite Element Discretization

Unlike finite difference methods, adaptive finite element (FE) discretizations[29] offer the ability to place computational effort in specific regions of the problem domain. Finite element meshes (see Fig. 2B) are composed of simplices (e.g., triangles or tetrahedra) that are joined at edges and vertices. The solution is constructed from piecewise polynomial basis functions (see Fig. 2C), which are associated with mesh vertices and typically are nonzero only over a small set of neighboring simplices. Solution accuracy can be increased in specific areas by locally increasing the number of vertices through simplex refinement. As shown in Fig. 2E, the number of unknowns (vertices) is generally increased only in the immediate vicinity of the simplex refinement and not throughout the entire problem domain, as in FD methods. This ability to locally increase the solution resolution is called ''adaptivity'' and is the major strength of finite element methods applied to the PB equation.[30–33] As with the FD method, this discretization scheme leads to sparse symmetric matrices with a small number of nonzero entries in each row.[29]

Multilevel Solvers

Multilevel solvers,[34] in conjunction with the Newton methods described above, have been shown to provide the most efficient solution of the algebraic equations obtained by discretization of the PB equation with either finite difference or finite element techniques.[1,25,30,32,35,36] Most sizable

[29] O. Axelsson and V. A. Barker, *in* ''Finite Element Solution of Boundary Value Problems: Theory and Computation.'' Academic Press, San Diego, CA, 1984; D. Braess, ''Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics. Cambridge University Press, Cambridge, 1997; S. C. Brenner, and L. R. Scott, ''The Mathematical Theory of Finite Element Methods,'' 2nd Ed., pp. xv and 361. Springer-Verlag, New York, 2002.

[30] M. Holst, N. Baker, and F. Wang, *J. Comput. Chem*. **21,** 1319 (2000).

[31] N. Baker, M. Holst, and F. Wang, *J. Comput. Chem*. **21,** 1343 (2000).

[32] N. A. Baker, D. Sept, M. J. Holst, and J. A. McCammon, *IBM J. Res. Dev*. **45,** 427 (2001).

[33] C. M. Cortis and R. A. Friesner, *J. Comput. Chem*. **18,** 1591 (1997); C. M. Cortis and R. A. Friesner, *J. Comput. Chem*. **18,** 1570 (1997).

[34] W. Hackbusch, *in* ''Multigrid Methods and Applications.'' Springer-Verlag, Berlin, 1985; W. L. Briggs, *in* ''A Multigrid Tutorial,'' pp. ix and 88. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1987.

[35] M. Holst, *Adv. Comput. Math*. **15,** 139 (2001).

[36] N. A. Baker, D. Sept, S. Joseph, M. J. Holst, and J. A. McCammon, *Proc. Natl. Acad. Sci. USA* **98,** 10037 (2001).

algebraic equations are solved by iterative methods, which start with an initial guess and repeatedly apply a set of operations to improve this guess until a solution of the desired accuracy is reached. However, the speed of traditional iterative methods has been limited by their inability to quickly reduce low-frequency (long-range) error in the solution.[34] This problem can be overcome by projecting the discretized system onto meshes (or grids) at multiple resolutions (see Fig. 2D and E). The advantage of this multiscale representation is that the slowly converging low-frequency components of the solution on the finest mesh are quickly resolved on coarser levels of the system. This coupling of scales gives rise to a ''multilevel'' solver algorithm, where the algebraic system is solved directly on the coarsest level and then used to accelerate solutions on finer levels of the mesh.

As shown in Fig. 2D and E, assembly of the multilevel hierarchy depends on the method used to discretize the PB equation. For FD types of methods, so-called multigrid methods are used: the nature of the FD grid lends itself to the assembly of a hierarchy with little additional work.[25,34] In the case of adaptive finite element discretizations, ''algebraic multigrid'' methods[32,35] are employed. For FE meshes, the most natural multiscale representation is constructed by refinement of an initial mesh, which typically constitutes the coarsest level of the hierarchy.

The advantage of multiscale algorithms in solving numeric problems cannot be overstated as they enable the solution of large-scale numerical systems in an optimal fashion; that is, the time required to solve an $N \times N$ linear algebraic system scales as $O(N)$.[25,30,34] Figure 3 illustrates the theoretical scaling behavior of a variety of numerical methods (both traditional and multilevel) implemented in popular PB software (see Table I). This plot was calibrated on the basis of timing data from real-world multigrid calculations. For comparison, the PB equation for a typical small protein (i.e., lysozyme) could be solved to reasonable accuracy (0.5–1.0 Å) using a finite difference discretization on a relatively small ($65 \times 65 \times 65$) grid by any of the methods shown in Fig. 3; the expected calculation time would be roughly 1 s. However, to study systems only five times larger (i.e., ribosomes, microtubules, polymerases, etc.) could require up to six orders of magnitude more time with a standard iterative method (e.g., successive overrelaxation) than with multigrid. On the basis of this scaling behavior, it should be clear that multilevel methods are essential to the efficient solution of the PB equation in an era in which the scale of experimentally resolved structures is continuously increasing.

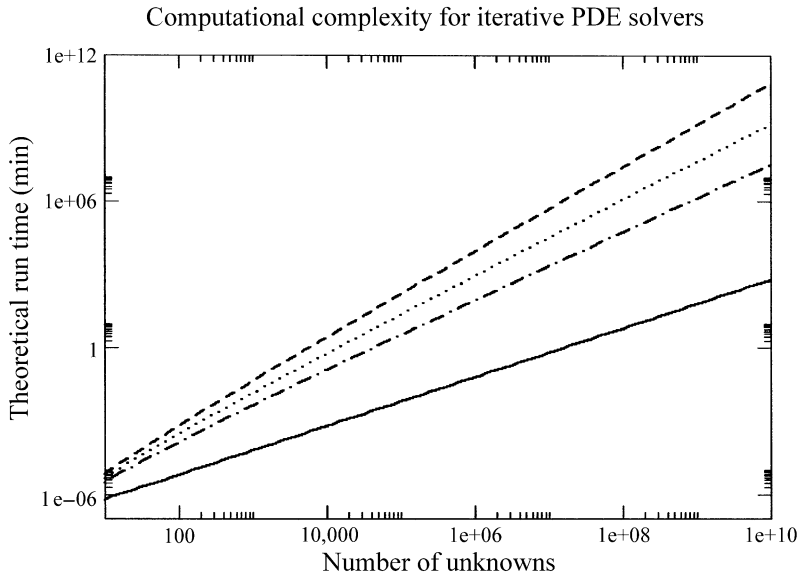

FIG. 3. Computational complexity for common iterative PD equation solvers: multigrid (—), successive overrelaxation (– • –), conjugate gradient (•••), and Gauss–Seidel (– – –). Theoretical run times were calculated from asymptotic scaling estimates for these methods[25,37] and calibrated by the observed multigrid run time with APBS software[36,38] for $65 \times 65 \times 65$ unknowns.

Parallel Methods

Regardless of the scalability of the numerical algorithm used to solve the PB equation, there are some systems that are simply too large to be solved sequentially (i.e., on one processor). For example, although small- to medium-sized protein systems (100–1000 residues) are amenable to sequential calculations, there are an increasing number of structures of macromolecular assemblages with tens to hundreds of thousands of residues (e.g., microtubules, entire viral capsids, ribosomes, polymerases, etc.). Studies of these large systems are not feasible on most sequential platforms; instead, they require multiprocessor computing platforms to solve the PB equation in a parallel fashion.

Traditional parallel algorithms for solving the PB equation introduced parallelism at the "fine-grain" level[16,39]; for example, during matrix–vector

[37] G. H. Golub and C. F. van Loan, *in* "Matrix Computations," 3rd Ed. Johns Hopkins University Press; M. D. Baltimore, 1996; W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *in* "Numerical Recipes in C." Cambridge University Press, New York, 1992.

[38] N. A. Baker, *in* "Adaptive Poisson–Boltzmann Solver (APBS)." Washington University in St. Louis, St. Louis, MO, 1999–2003. http://agave.wustl.edu/apbs/

TABLE I
POISSON–BOLTZMANN SOFTWARE FOR BIOMOLECULAR SYSTEMS[a]

Software package	Description	URL	Availability
APBS[36,38]	Solves PBE in parallel with FD MG and FE AMG solvers	http://agave.wustl.edu/apbs/	Windows, all Unix; free, open source
DelPhi[27,40]	Solves PBE sequentially with highly optimized FD GS solver	http://trantor.bioc.columbia.edu/delphi/	SGI, Linux, AIX; $250 academic
GRASP[41]	Visualization program with emphasis on graphics; offers sequential calculation of qualitative PB potentials	http://trantor.bioc.columbia.edu/grasp/	SGI; $500 academic
MEAD[42]	Solves PBE sequentially with FD SOR solver	http://www.scripps.edu/bashford	Windows, all Unix; free, open source
UHBD[26,43]	Multipurpose program with emphasis on SD; offers sequential FD SOR PBE solver	http://mccammon.ucsd.edu/uhbd.html	All Unix; $300 academic
MacroDox	Multipurpose program with emphasis on SD; offers sequential FD SOR PBE solver	http://pirn.chem.tntech.edu/macrodox.html	SGI; free, open source
Jaguar[33,44]	Multipurpose program with emphasis on QM; offers sequential FE MG, SOR, and CG solvers	http://www.schrodinger.com/Products/jaguar.html	Most Unix; commercial
CHARMM[45]	Multipurpose program with emphasis on MD; offers sequential FD MG solver and can be linked with APBS	http://yuri.harvard.edu	All Unix; $600 academic

[a] *Notation:* PBE, Poisson–Boltzmann equation; MG, multigrid; AMG, algebraic multigrid; FD, finite difference; FE, finite element; GS, Gauss–Seidel; CG, conjugate gradient; SOR, successive overrelaxation; SD, stochastic dynamics; QM, quantum mechanics; MM, molecular mechanics; MD, molecular dynamics.

multiplication and other linear algebra operations. Although such methods are among the most common form of parallelization, they require substantial communication. An optimal parallel algorithm using P processors performs P times more work than the same algorithm using one processor; that is, the work performed by an optimal algorithm scales linearly with P. However, the substantial communication required by traditional fine-grain parallel PB solver algorithms destroys the parallel performance and offers only sublinear (suboptimal) parallel scaling. In other words, there is a diminishing return in the amount of work achieved by increasing the number of processors. Therefore, suboptimal parallel algorithms are not suitable for scaling PB calculations to larger biomolecular systems. However, two new methods have been developed for the coarse-grained parallel solution of the PB equation.[32,36] These highly efficient parallel algorithms were designed for use with both the FD and FE discretizations explained above and are suitable for the study of large biomolecular systems consisting of millions of atoms. The two methods are described in detail in the following sections.

Parallel Finite Element Methods. In 2000, a new algorithm (Bank–Holst) was described for the parallel adaptive finite element solution of elliptic partial differential equations with negligible interprocess communication.[46] As mentioned above, adaptive finite element techniques generate accurate solutions of these equations by locally enriching the basis set in regions of high error through refinement of the domain discretization. The Bank–Holst algorithm exploits this local refinement and constructs a parallel algorithm by confining local mesh refinement to regions of the problem domain assigned to particular processors.

---

[39] A. Ilin, B. Bagheri, L. R. Scott, J. M. Briggs, and J. A. McCammon, *in* "Parallelization of Poisson–Boltzmann and Brownian Dynamics Calculations." American Chemical Society Symposium Series, Vol. 592, pp. 170–185. American Chemical Society, Washington, DC, 1995.

[40] A. Nicholls and B. Honig, *J. Comput. Chem*. **12,** 435 (1991).

[41] A. Nicholls, K. A. Sharp, and B. Honig, *Proteins* **11,** 281 (1991).

[42] D. Bashford, *in* "Scientific Computing in Object-Oriented Parallel Environments" (Y. Ishikawa, R. R. Oldehoeft, J. V. W. Reynders, and M. Tholburn, eds.). Springer-Verlag, Berlin, 1997.

[43] J. D. Madura, J. M. Briggs, R. C. Wade, M. E. Davis, B. A. Luty, A. Ilin, J. Antosiewicz, M. K. Gilson, B. Bagheri, L. R. Scott, and J. A. McCammon, *Comput. Phys. Commun*. **91,** 57 (1995).

[44] G. Vacek, J. K. Perry, and J.-M. Langlois, *Chem. Phys. Lett*. **310,** 189 (1999).

[45] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *J. Comput. Chem*. **4,** 187 (1983); A. D. J. MacKerell, B. Brooks, C. L. I. Brooks, L. Nilsson, B. Roux, Y. Won, and M. Karplus, *in* "The Encyclopedia of Computational Chemistry" (P. V. R. Schleyer, ed.), pp. 271–277. John Wiley & Sons, Chichester, 1998.

[46] R. E. Bank and M. Holst, *Siam J. Sci. Comput*. **22,** 1411 (2000).

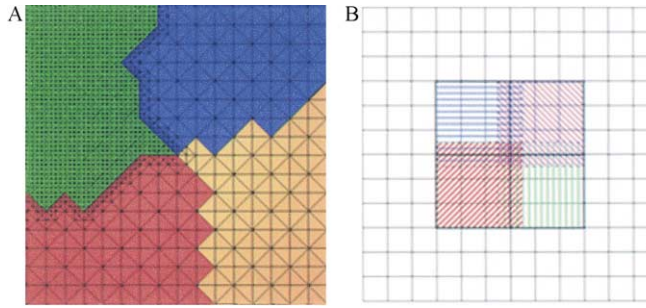

FIG. 4. Domain decomposition for parallel methods; colors denote mesh partitions belonging to individual processors of a parallel computer. (A) The Bank–Holst parallel FE method; the mesh shown has been refined from a coarser mesh by the green processor. (B) The parallel focusing method; each processor focuses from the larger coarse mesh to its particular smaller colored region.

The algorithm (illustrated in Fig. 4A) proceeds as follows: the first step in the Bank–Holst parallel finite element algorithm is the solution of the equation over the entire problem domain, using a coarse resolution basis set by each processor. This solution is then used, in conjunction with an error estimator, to partition the problem domain into P subdomains that are assigned to the P processors of a parallel computer. The algorithm is load balanced by equidistributing the error estimates across the subdomains. Each processor then solves the partial differential equation over the global mesh, but confines its adaptive refinement to the local subdomain and a small surrounding overlap region. This procedure results in an accurate representation of the solution over the local subdomain. After all processors have completed their local adaptive solution of the equation, a global solution is constructed by the piecewise assembly of the solutions in each subdomain.

It can be rigorously shown that the piecewise-assembled solution is as accurate as the solution to the problem on a single global mesh.[35] Because each processor performs all of its computational work independently, the Bank–Holst algorithm requires little interprocess communication and exhibits excellent parallel scaling.

Parallel Focusing. Unfortunately, although the Bank–Holst algorithm works well for adaptive techniques, such as finite elements, it is not directly applicable to the fixed-resolution finite difference methods that are the standard methods used by the biological community. However, by combining the Bank–Holst method with other techniques, a new parallel focusing method was developed by Baker and co-workers[36] to study large biomolecular systems in parallel, using traditional finite difference discretizations of the PB equation.

Electrostatic "focusing" is a popular technique in finite difference methods for generating accurate solutions to the PB equation in subsets of the problem domain, such as a binding or titratable sites within a protein.[4,5,47] The first step in electrostatic focusing is the calculation of a low-accuracy solution on a coarse finite difference mesh spanning the entire problem domain. This coarse solution is then used to define the boundary conditions for a much more accurate calculation on a finer discretization of the desired subdomain. As noted previously,[46] this focusing technique is superficially related to the Bank–Holst algorithm, where the local enrichment of a coarse, global solution has been replaced by the solution of a fine, local multigrid problem using the solution from a coarse, global problem for boundary conditions.

Parallel focusing is simply a combination of standard focusing techniques with the Bank–Holst algorithm into a new method for the parallel solution of the PB equation with finite difference discretization. This algorithm begins with each processor independently solving a coarse global problem. The per-processor subdomains are chosen in a heuristic fashion as outlined in Fig. 4. Like standard focusing calculations, only a subset of the global mesh surrounding the area of interest is used for the parallel calculations. This subset is partitioned into P approximately equal subdomains that are distributed among the processors. Each processor then performs a fine-scale finite difference calculation over this subdomain and an overlap region that usually spans about 5–10% of the neighboring subdomains. The overlap regions are included to compensate for inaccuracies in the boundary conditions derived from the global coarse solution; however, these regions are not used to assemble the fine-scale global solution and do not contribute to calculations of observables such as forces.

Unlike previous parallel algorithms for solving the PB equation, this method has excellent parallel complexity, permitting the treatment of large biomolecular systems on massively parallel computational platforms. Furthermore, the finite difference discretization on a regular mesh allows for fast solution by highly efficient multigrid solvers. Because of the efficiency of the parallel and underlying multigrid algorithms, implementation of the parallel focusing algorithm in APBS software (see Table I) has enabled the largest PB calculations to date.[36] Specifically, Baker and co-workers used APBS on the National Partnership for Advanced Computational Infrastructure (NPACI). Blue Horizon supercomputer to solve the PB equation for the 88,000-atom small ribosomal subunit,[48] the 95,000-atom

[47] M. K. Gilson and B. H. Honig, *Nature* **330,** 84 (1987).
[48] A. P. Carter, W. M. Clemons, D. E. Brodersen, R. J. Morgan-Warren, B. T. Wimberly, and V. Ramakrishnan, *Nature* **407,** 340 (2000).

large ribosomal subunit,[49] and a 1.2-million atom microtubule structure derived from coordinates by Nogales and co-workers.[50] Figure 5 illustrates some of the specific results obtained from the microtubule calculations and the optimal parallel scaling of the algorithm. The use of parallel focusing in routine APBS calculations is straightforward; users simply specify

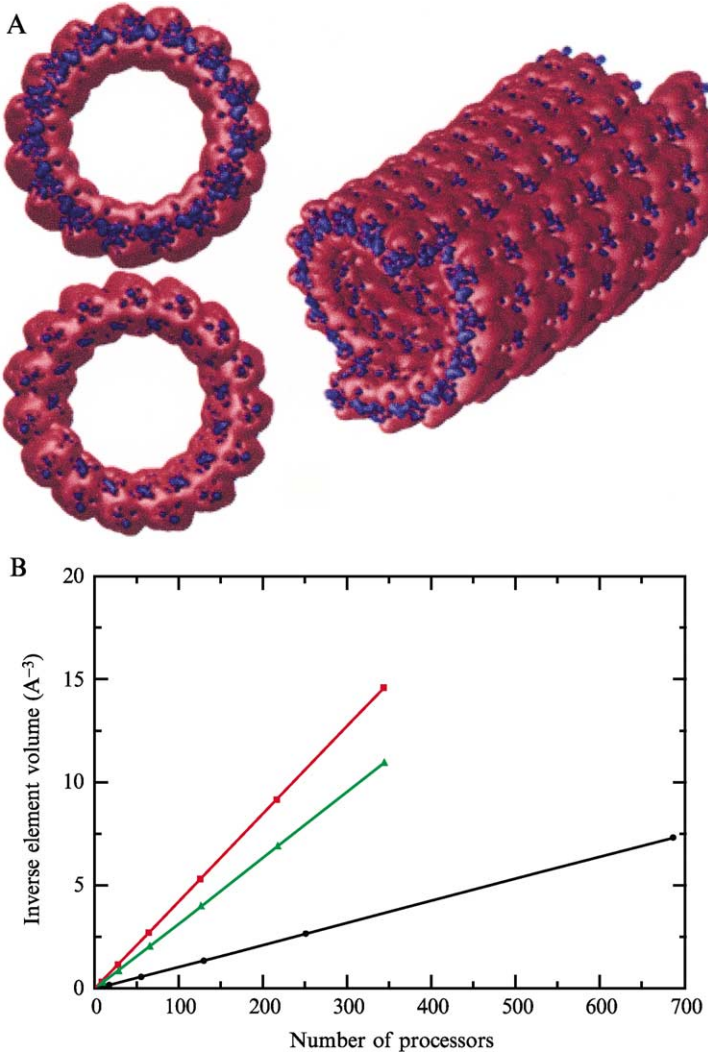

FIG. 5. Results from parallel focusing calculations.[36] (A) Views of electrostatic isocontours for a 1.2-million atom microtubule structure (blue, $+1$ kT/e; red, -1 kT/e). (B) Graph illustrating the optimal efficiency of parallel focusing algorithm.

the desired number of processors and an overlap parameter and the remaining aspects of the run are calculated for them.

Software for Computational Electrostatics

Table I presents a list of the major software currently used to solve the Poisson–Boltzmann equation for biomolecular systems. There are a variety of such programs, ranging from multipurpose computational biology packages (e.g., CHARMM, Jaguar, UHBD, and MacroDox) to specialized PB solvers (e.g., APBS, MEAD, and DelPhi).

In addition to the traditional "stand-alone" software packages listed in Table I, Web-based services are also becoming available for solving the PB equation. Such services often have the advantage of removing the troublesome details of software installation from the user. Quite often, Web services also simplify and/or automate the process of setting up calculations. The first of these Web-based packages is the APBS Web Portal (https://gridport.npaci.edu/apbs/), a service for preparing, submitting, and organizing electrostatics calculations on supercomputing platforms. This Web portal is aimed at quantitative electrostatics calculations and currently does not offer substantial analysis or visualization options. On the other hand, the GRASS (http://trantor.bioc.columbia.edu/) Web service supports *qualitative* electrostatics calculations with extensive visualization capabilities. GRASS allows users to easily calculate and visualize electrostatic potentials and other properties using uploaded structures or PDB entries.

Applications to Biomedical Sciences

There have been far too many contributions of electrostatics calculations to the biological community to list them all in detail; see Refs. 1, 4, and 5, and the numerous references therein. However, we provide a brief overview of some of the applications of computational electrostatics with a particular focus on the practical application of PB methods to common biomolecular electrostatics problems.

Qualitative and Quantitative Potential Analysis

Perhaps the most readily recognizable aspects of computational electrostatics are the images produced by coloring biomolecular surfaces according to electrostatic potential (see Fig. 6A) or by plotting electrostatic isosurfaces surrounding a molecule (see Fig. 6B). By far, the most popular

[49] N. Ban, P. Nissen, J. Hansen, P. B. Moore, and T. A. Steitz, *Science* **289,** 905 (2000).
[50] E. Nogales, M. Whittaker, R. A. Milligan, and K. H. Downing, *Cell* **96,** 79 (1999).

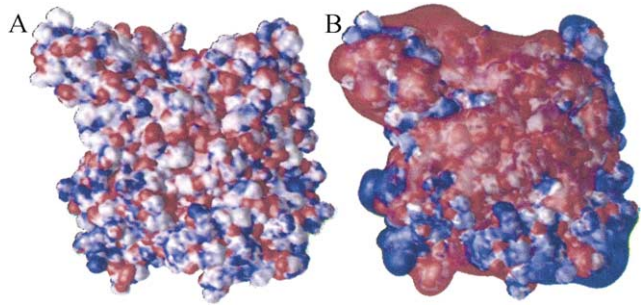

FIG. 6. Electrostatic potential of acetylcholinesterase (PDB ID 1MAH[51]). Potential calculation by solution of the PB equation with APBS. (A) Electrostatic potential mapped onto molecular surface; regions with negative potential are shaded red, regions with positive potential are shaded blue. (B) Isocontours of electrostatic potential; blue contour at $+1kT/e$, red contour at $-1kT/e$.

software package for generating such packages is GRASP,[41] developed by the Honig laboratory. Images of the electrostatic potential such as in Fig. 6 provide insight into the electrostatic properties of the protein—both at the molecular surface and surrounding the biomolecule. Such insight cannot simply be obtained by mapping atomic partial charges onto the surface of the biomolecule; the complex shapes of proteins, together with their low dielectric interior, have been shown to perturb the overall electrostatic potential, often focusing it into functionally important regions.[4,5,52] In addition to visual presentation, the electrostatic potentials calculated by the PB and other methods have also been analyzed to identify active and ligand-binding sites,[53–55] to predict protein–protein[56,57] and protein–membrane interfaces,[58–60] and to categorize biomolecules on the basis of the potential

[51] Y. Bourne, P. Taylor, and P. Marchot, *Cell* **83,** 503 (1995).
[52] I. Klapper, R. Hagstrom, R. Fine, K. Sharp, and B. Honig, *Proteins* **1,** 47 (1986).
[53] A. H. Elcock, *J. Mol. Biol.* **312,** 885 (2001); M. J. Ondrechen, J. G. Clifton, and D. Ringe, *Proc. Natl. Acad. Sci. USA* **98,** 12473 (2001).
[54] Z. Y. Zhu and S. Karlin, *Proc. Natl. Acad. Sci. USA* **93,** 8350 (1996).
[55] A. M. Richard, *J. Comput. Chem.* **12,** 959 (1991).
[56] R. Norel, F. Sheinerman, D. Petrey, and B. Honig, *Protein Sci.* **10,** 2147 (2001); S. M. de Freitas, L. V. de Mello, M. C. da Silva, G. Vriend, G. Neshich, and M. M. Ventura, *FEBS Lett.* **409,** 121 (1997); L. Lo Conte, C. Chothia, and J. Janin, *J. Mol. Biol.* **285,** 2177 (1999); J. Janin and C. Chothia, *J. Biol. Chem.* **265,** 16027 (1990); V. A. Roberts, H. C. Freeman, A. J. Olson, J. A. Tainer, and E. D. Getzoff, *J. Biol. Chem.* **266,** 13431 (1991).
[57] R. C. Wade, R. R. Gabdoulline, and F. De Rienzo, *Int. J. Quantum Chem.* **83,** 122 (2001); J. Novotny and K. Sharp, *Prog. Biophys. Mol. Biol.* **58,** 203 (1992); A. J. McCoy, V. Chandana Epa, and P. M. Colman, *J. Mol. Biol.* **268,** 570 (1997).
[58] A. Arbuzova, L. B. Wang, J. Y. Wang, G. Hangyas-Mihalyne, D. Murray, B. Honig, and S. McLaughlin, *Biochemistry* **39,** 10330 (2000).

at and surrounding their surfaces.[55,57,60,61] It is expected that such methods for automatic processing of structural will become increasingly important as the number of experimentally resolved structures increases dramatically as the result of structural genomics efforts.[1,53,62]

Biomolecular Energetics

Evaluation of biomolecular energetics is another common application of PB and other computational electrostatics methods. PB methods have been used for numerous medicinal chemical and ligand/ion-binding calculations. In particular, the MM/PBSA method (a hybrid molecular mechanics–PB technique), developed by Massova and Kollman, is popular for the calculation of binding free energies.[63] In addition, Poisson–Boltzmann solvers have also aided in the investigation and simulation of several supramolecular systems, including protein–protein,[64,65] protein–nucleic acid,[66] and protein–membrane interactions.[59,60] Poisson–Boltzmann and continuum electrostatics energetics have been central to protein titration and pK_a calculations.[7,67] Such pK_a calculations play a role in several aspects of computational biology, including the setup of biomolecular simulations, the investigation of mechanisms for ligand binding and catalysis, and the identification of enzyme active sites. Finally, PB methods have also been used to investigate DNA and RNA systems, examining

[59] J.-H. Lin, N. A. Baker, and J. A. McCammon, *Biophys. J.* **83,** 1374 (2002); C. Fleck, R. R. Netz, and H. H. von Grunberg, *Biophys. J.* **82,** 76 (2002).

[60] D. Murray and B. Honig, *Mol. Cell* **9,** 145 (2002).

[61] S. A. Botti, C. E. Felder, J. L. Sussman, and I. Silman, *Protein Eng.* **11,** 415 (1998); E. Demchuk, T. Mueller, H. Oschkinat, W. Sebald, and R. C. Wade, *Protein Sci.* **3,** 920 (1994); L. T. Chong, S. E. Dempster, Z. S. Hendsch, L. P. Lee, and B. Tidor, *Protein Sci.* **7,** 206 (1998); N. Blomberg, R. R. Gabdoulline, M. Nilges, and R. C. Wade, *Proteins Struct. Funct. Genet.* **37,** 379 (1999); L. P. Lee and B. Tidor, *Protein Sci.* **10,** 362 (2001); E. Kangas and B. Tidor, *Phys. Rev. E* **59,** 5958 (1999).

[62] H. M. Berman, T. N. Bhat, P. E. Bourne, Z. Feng, G. Gilliland, H. Weissig, and J. Westbrook, *Nat. Struct. Biol.* **7**(Suppl.), 95 (2000).

[63] I. Massova and P. A. Kollman, *Perspect. Drug Discov. Design* **18,** 113 (2000); L. T. Chong, Y. Duan, L. Wang, I. Massova, and P. A. Kollman, *Proc. Natl. Acad. Sci. USA* **96,** 14330 (1999).

[64] A. H. Elcock, D. Sept, and J. A. McCammon, *J. Phys. Chem. B* **105,** 1504 (2001).

[65] J. A. McCammon, *Curr. Opin. Struct. Biol.* **8,** 245 (1998).

[66] S. W. W. Chen and B. Honig, *J. Phys. Chem. B* **101,** 9113 (1997); F. Fogolari, A. H. Elcock, G. Esposito, P. Viglino, J. M. Briggs, and J. A. McCammon, *J. Mol. Biol.* **267,** 368 (1997); V. K. Misra, J. L. Hecht, A. S. Yang, and B. Honig, *Biophys. J.* **75,** 2262 (1998); K. A. Sharp, R. A. Friedman, V. Misra, J. Hecht, and B. Honig, *Biopolymers* **36,** 245 (1995).

[67] J. E. Nielsen and G. Vriend, *Protein Struct. Funct. Genet.* **43,** 403 (2001); J. Antosiewicz, J. A. McCammon, and M. K. Gilson, *Biochemistry* **35,** 7819 (1996); D. Bashford and M. Karplus, *Biochemistry* **29,** 10219 (1990).

issues of nucleic acid–ion binding,[68] nucleic acid structure minimization, and conformational analysis.[69]

The remainder of this section describes how electrostatic energies are obtained from solution of the PB equation and used to study biological systems. The total electrostatic energy $G[\phi]$ can be obtained by integrating the solution $\phi(x)$ to the PB equation over the solution domain Ω[70]:

$$G[\phi] = \int_{\Omega} \left[\rho_f \phi - \frac{\varepsilon}{2} (\nabla \phi)^2 - \overline{\kappa}^2 (\cosh \phi - 1) \right] dx \tag{7}$$

The first term of Eq. (7) is the energy of inserting the protein charges into the electrostatic potential and can be interpreted as the interaction energy between charges. However, unlike analytic representations of charge–charge interactions, this energy also includes large ''self-energy'' terms associated with the interaction of a particular charge with itself. These self-energy terms are highly dependent on the discretization of the problem; as the mesh spacing increases, these terms become larger. In general, self-energies are treated as artifacts of the calculation and are removed by a reference calculation using the same discretization (see later section). The second term of Eq. (7) can be interpreted as the energy of polarization for the dielectric medium. Finally, the third term is the energy of the mobile counterion distribution. Both the dielectric and mobile ion energies do not include self-energy terms and therefore do not need to be corrected by reference calculations. As with the PB equation, the energy can be linearized by noting $\cosh\phi(x) - 1 \approx \phi^2(x)/2$ to give the simplified expression

$$G[\phi] = \int_{\Omega} \left[\rho_f \phi - \frac{\varepsilon}{2} (\nabla \phi)^2 - \frac{\overline{\kappa}^2}{2} \phi^2 \right] dx = \frac{1}{2} \int_{\Omega} \rho_f \phi dx \tag{8}$$

where the second equality is derived from substitution of the linearized PB equation [Eq. (6)] into the integrand. The following sections provide a few examples showing the application of Eq. (7) and Eq. (8) to biomolecular systems.

[68] V. A. Bloomfield and I. Rouzina, *Methods Enzymol.* **295,** 364 (1998); G. Lamm, L. Wong, and G. R. Pack, *Biopolymers* **34,** 227 (1994); V. K. Misra and D. E. Draper, *J. Mol. Biol.* **294,** 1135 (1999).

[69] R. A. Friedman, and B. Honig, *Biopolymers* **32,** 145 (1992); J. L. Hecht, B. Honig, Y. K. Shin, and W. L. Hubbell, *J. Phys. Chem.* **99,** 7782 (1995); V. K. Misra, K. A. Sharp, R. A. Friedman and B. Honig, *J. Mol. Biol.* **238,** 245 (1994).

[70] K. A. Sharp and B. Honig, *J. Phys. Chem.* **94,** 7684 (1990); M. Gilson, K. Sharp, and B. Honig, *Proteins* **4,** 7 (1988); A. M. Micu, B. Bagheri, A. V. Ilin, L. R. Scott, and B. M. Pettitt, *J. Comput. Phys.* **136,** 263 (1997).

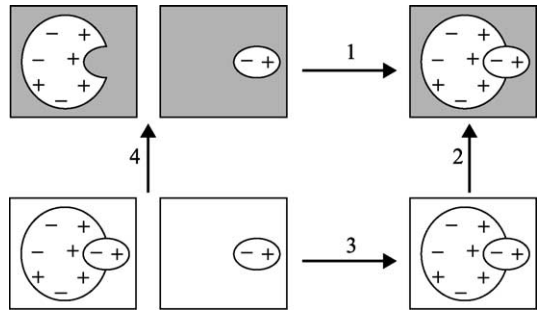

FIG. 7. Typical free energy calculations for a biomolecular system. The white background indicates an environment identical to the biomolecular interior with a low dielectric constant $\varepsilon(x) = \varepsilon_p$ and zero ionic strength $\bar{\kappa}^2(x) = 0$. The gray background indicates an environment identical to the biomolecular exterior with (generally) higher dielectric coefficient $\varepsilon(x) = \varepsilon_s$ and nonzero ionic strength $\bar{\kappa}^2(x) > 0$. Arrow 1 indicates the binding of two biomolecules in their physical state (with different internal and external environments). Arrow 2 indicates the solvation energy calculation for the biomolecular complex; the reference state has the same internal and external environment. Arrow 3 indicates the binding of the biomolecules in a constant environment identical to the biomolecular interior. Arrow 4 denotes the solvation energy calculation for the isolated biomolecular components.

Calculating Solvation Energy. The most basic component of an electrostatic energy calculation is the solvation energy, shown as processes 1 and 4 of the thermodynamic cycle depicted in Fig. 7. Specifically the solvation energy is computed as

$$\Delta G_{\text{solv}} = G_{\text{sys}} - G_{ref} \tag{9}$$

Here G_{sys} is the electrostatic free energy of the system of interest with differing dielectric constants inside (ε_p) and outside (ε_s) the biomolecule, a fixed charge distribution corresponding to the atom locations and charges, and a varying ion accessibility coefficient ($\bar{\kappa}^2$) that is zero inside the biomolecule and equal to the bulk ionic strength outside. The reference free energy G_{ref} uses the same fixed charge distribution but has a constant dielectric coefficient equal to the value in biomolecular interior* $\varepsilon(x) = \varepsilon_p$ and a constant zero ion accessibility coefficient. Therefore a solvation energy calculation consists of the following steps:

* Note that ε_p may not necessarily be equal to 1 (vacuum) and therefore the computed quantity may not be directly comparable to an experimentally observed solvation free energy. See later sections for information on how the desired solvation energy can be calculated when $\varepsilon_p \neq 1$.

1. Calculate ϕ_{sys} by solving the PB equation for the system of interest, using a particular FD or FE discretization.
2. Calculate ϕ_{ref} by solving the PB equation for the reference, using the same FD or FE discretization.
3. Calculate the solvation free energy from these two solutions:

$$\Delta G_{\text{solv}} = \int_{\Omega} \left[\rho_f \left(\phi_{\text{sys}} - \frac{\phi_{\text{ref}}}{2} \right) - \frac{\varepsilon}{2} \left(\nabla \phi_{\text{sys}} \right)^2 - \bar{\kappa}^2 \left(\cosh \phi_{\text{sys}} - 1 \right) \right] dx \quad (10)$$

An important aspect of this calculation is the use of the same discretizations (i.e., same FD grid spacing or FE refinement) in steps 1 and 2. Both the G_{sys} and G_{ref} energies contain large self-energy terms that are dependent on the discretization; however, by choosing the same meshes or grids for the both calculations, these self-energies cancel in computing ΔG_{solv} via Eq. (10).

The electrostatic solvation energy is often supplemented with an estimate of the nonpolar contribution to biomolecular energetics. There are several approximate methods for evaluating the nonpolar term (see Ref. 3), but the most popular is simply a linear function of the solvent-accessible surface area:

$$\Delta G_{\text{solv}} = \int_{\Omega} \left[\rho_f \left(\phi_{\text{sys}} - \frac{\phi_{\text{ref}}}{2} \right) - \frac{\varepsilon}{2} \left(\nabla \phi_{\text{sys}} \right)^2 - \bar{\kappa}^2 \left(\cosh \phi_{\text{sys}} - 1 \right) \right] dx + \gamma A \quad (11)$$

where A is the solvent-accessible surface area and γ is the energetic coefficient or surface tension.

Given the basic role of solvation energy calculations in PB methods, most PB software packages offer some sort of automatic way to calculate Eq. (11) without requiring users to be too involved in the setup of the reference and system calculations. However, knowledge of the details of these steps is important when designing and troubleshooting more complex energy calculations.

Calculating Binding Energies. The concept of removing self-energies through solvation energy reference calculations is a common theme in electrostatic energetics. Figure 7 illustrates this procedure in the context of a binding energy calculation thermodynamic cycle. Following the usual conventions, the binding free energy is calculated as

$$\Delta G_{bind} = \Delta G_1 = \Delta G_2 + \Delta G_3 - \Delta G_4 \quad (12)$$

where ΔG_2 is the solvation energy of the complex, ΔG_4 is the total solvation energy of the isolated components, and ΔG_3 is the energy of the component charge distributions interacting in a uniform dielectric continuum. In practice, the solvation energies ΔG_2 and ΔG_4 are obtained by

numerical solution of the PB equation and Eq. (11). Although ΔG_3 can also be evaluated numerically, it is typically desirable to use the analytical expression for this energy (i.e., Coulomb's law) to achieve more accurate results.

In general, both binding energy calculations and other types of energy calculations (see below) are not automatically calculated by most PB software. Therefore, users need to be aware of the details (especially the importance of removing self-energies) when setting up these computations and basing their calculations on free energy cycles as in Fig. 7.

Other Energy Calculations. As mentioned above, there are several types of energetic calculations that can be performed with electrostatic methods. However, most of these use the same general formula as outlined for binding in later sections. Consider a general transition of a biomolecular system from state A to state B (e.g., ligand binding, conformational change, protonation, etc.). To calculate the free energy for the transition $G_{sys}^{A\to B}$, one must calculate

- $\Delta G_{\text{solv}}^{\text{A}}$: The solvation energy of state A. This is determined from a numerical solution to the PB equation and Eq. (11).
- $\Delta G_{\text{solv}}^{B}$: The solvation energy of state B. This is determined from a numerical solution to the PB equation and Eq. (11) (with the same discretization as for state [A]).
- $\Delta G_{\text{ref}}^{A\to B}$: The energy of going from state A to B in the reference environment; homogeneous dielectric $\varepsilon(x) = \varepsilon_p$ and zero ionic strength $\overline{\kappa}^2(x) = 0$. This is determined either from a numerical solution of the PB equation or an analytical expression such as Coulomb's law.

The desired energy for the transition is then calculated from these three quantities:

$$\Delta G_{\text{sys}}^{A\to B} = \Delta G_{\text{ref}}^{A\to B} + \Delta G_{\text{solv}}^{B} - \Delta G_{\text{solv}}^{A} \tag{13}$$

Biomolecular Dynamics

Not surprisingly, PB calculations have also found numerous applications in the study of biomolecular dynamics. In general, solutions to the PB equation are used to derive forces[22,71,72] for incorporation in an implicit solvent stochastic dynamics simulation. Among the most popular dynamics

[71] M. K. Gilson, M. E. Davis, B. A. Luty, and J. A. McCammon, *J. Phys. Chem.* **97,** 3591 (1993).
[72] B. Z. Lu, W. Z. Chen, C. X. Wang, and X. J. Xu, *Proteins* **48,** 497 (2002); M. E. Davis and J. A. McCammon, *J. Comput. Chem.* **11,** 40 (1990); M. Friedrichs, R. H. Zhou, S. R. Edinger, and R. A. Friesner, *J. Phys. Chem. B* **103,** 3057 (1999).

applications of PB theory is the Brownian dynamics (BD) simulation[73–75] of diffusional encounter. There have been numerous applications of BD simulations to biomolecular systems, but the most common use of these methods is the calculation of diffusive reaction rates for biomolecule–biomolecule or biomolecule–ligand binding for a variety of protein, membrane, and nucleic acid systems.[65,73,76] In general, such diffusional encounter simulations offer little or no atomistic flexibility and generally consider the rigid body motion of the biomolecules. However, PB electrostatics have also been used to supply forces for atom-level stochastic dynamics simulations using either Brownian or Langevin[75,77] equations of motion. The size of the molecules studied by stochastic dynamics has generally been limited by the calculation of electrostatic interactions—the computational bottleneck for these algorithms. Therefore previous atom-level simulations were constrained to much smaller molecules and generally used for the purposes of conformational sampling.[17,78] However, several improvements to the solution of the PB equation and calculation of forces have been made to accelerate such calculations and facilitate the stochastic dynamics simulations of much larger molecules. These improvements include new coarse-grain parallelization methods described later, effective charge methods for approximating electrostatic potentials,[79] and optimization of accuracy and update frequencies for PB force calculation methods.[80]

PB forces can be derived from functional differentiation of the free energy integral (see Refs. 22 and 71):

$$\mathbf{F}_i = -\int_\Omega \left[\phi \left(\frac{\partial \rho_f}{\partial \mathbf{y}_i} \right) - \frac{(\nabla \phi)^2}{2} \left(\frac{\partial \varepsilon}{\partial \mathbf{y}_i} \right) - (\cosh \phi - 1) \left(\frac{\partial \overline{\kappa}^2}{\partial \mathbf{y}_i} \right) \right] dx \qquad (14)$$

where $\mathbf{F}_i$ is the force on atom i, and $\partial/\partial \mathbf{y}_i$ denotes the derivative with respect to displacements of atom i. The first term of Eq. (14) represents

[73] D. L. Ermak and J. A. McCammon, *J. Chem. Phys*. **69,** 1352 (1978).
[74] T. J. Murphy and J. L. Aguirre, *J. Chem. Phys*. **57,** 2098 (1972); A. C. Branka, and D. M. Heyes, *Phys. Rev. E* **58,** 2611 (1998); W. F. van Gunsteren and H. J. C. Berendsen, *Mol. Phys*. **45,** 637 (1982).
[75] C. W. Gardiner, *in* "Handbook of Stochastic Methods for Physics, Chemistry, and the Natural Sciences," 2nd Ed., pp. xix and 442. Springer-Verlag, New York, 1985.
[76] S. H. Northrup, S. A. Allison, and J. A. McCammon, *J. Chem. Phys*. **80,** 1517 (1984); R. R. Gabdoulline and R. C. Wade, *Methods* **14,** 329 (1998).
[77] W. F. van Gunsteren and H. J. C. Berendsen, *Mol. Simul*. **1,** 173 (1988).
[78] J. L. Smart, T. J. Marrone, and J. A. McCammon, *J. Comput. Chem*. **18,** 1750 (1997); T. Y. Shen, C. F. Wong, and J. A. McCammon, *J. Am. Chem. Soc*. **105,** 8028 (2001); M. K. Gilson, *J. Comput. Chem*. **16,** 1081 (1995).
[79] R. R. Gabdoulline and R. C. Wade, *J. Phys. Chem*. **100,** 3868 (1996).
[80] R. Luo, L. David, and M. K. Gilson, *J. Comput. Chem.* **23,** 1244 (2002).

the force density for displacements of atom i in the potential; it can also be rewritten in the classic form $q_i \nabla \phi$ for a charged particle in an electrostatic field. The second term is the dielectric boundary pressure; that is, the force exerted on the biomolecule by the high dielectric solvent surrounding the low-dielectric interior. Finally, the third term is equivalent to the osmotic pressure or the force exerted on the biomolecule by the surrounding counterions.

Not surprisingly, like their energetic counterparts, force evaluations must also be performed in the context of reference calculation due to the presence of self-interactions in the charge distribution. Specifically, the first term of Eq. (14) contains large artificial ''self-force'' contributions from the charge distribution. Therefore, to calculate the PB-based forces on a biomolecule one must calculate the following three quantities:

- $\mathbf{F}_i^{\text{sys}}$: The total electrostatic force on atom i for the system of interest due to all atoms. This is calculated from the numerical solution of the PB equation and Eq. (14); the dielectric and ion accessibility coefficients are inhomogeneous.
- $\mathbf{F}_i^{\text{ref}}$: The total electrostatic force on atom i for the reference system due to all atoms. This is calculated from the numerical solution of the PB equation and Eq. (14); the dielectric and ion accessibility coefficients are homogeneous; namely, $\varepsilon(x) = \varepsilon_p$ and $\overline{\kappa}^2(x) = 0$. As with the computation of free energies, this calculation must be performed using the same discretization as $\mathbf{F}_i^{\text{sys}}$.
- $\mathbf{f}_i^{\text{ref}}$: The total electrostatic force on atom i due to all atoms except atom i (i.e., self-interactions are removed) for the reference system. Because the reference system is a set of point charges in a homogeneous dielectric medium, this force is obtained analytically from Coulomb's law.

Given these quantities, the electrostatic force on atom i with self-interactions removed is

$$\mathbf{f}_i^{\text{sys}} = \mathbf{f}_i^{\text{ref}} + \mathbf{F}_i^{\text{sys}} - \mathbf{F}_i^{\text{ref}} \tag{15}$$

Like PB energy evaluations, continuum electrostatics force calculations are often supplemented with approximations for nonpolar influences on the biomolecule. Using the solvent-accessible nonpolar term described in Section III.B2, we have an alternate form for the force that is directly analogous to the energy expression Eq. (11):

$$\mathbf{f}_i^{\text{sys}} = \mathbf{f}_i^{\text{ref}} + \mathbf{F}_i^{\text{sys}} - \mathbf{F}_i^{\text{ref}} - \int_\Omega \gamma \left(\frac{\partial A}{\partial \mathbf{y}_i} \right) dx \tag{16}$$

This extra apolar term plays an important role in electrostatic force calculations. The solvation forces obtained from PB calculations work to maximize the solvent–solute boundary surface area, thereby providing the maximum solvation for the biomolecule. Apolar forces, on the other hand, tend to drive the system to a conformation with minimum surface area. Therefore, solvation and apolar forces typically act in a delicate balance by exerting opposing forces on molecular structure.

PB force evaluation is currently available only in a small number of software packages: in particular APBS, CHARMM, and UHBD. Of these, CHARMM currently provides the most straightforward integration of PB forces into dynamics and other molecular mechanics applications.

Conclusions

The importance of electrostatics to molecular biophysics is well established; electrostatics have been shown to influence various aspects of nearly all biochemical reactions. Evaluation of the electrostatic properties of proteins and nucleic acids has a long history in the study of biopolymers and continues to be a standard practice for the investigation of biomolecular structure and function. Foremost among the models used for biomolecular electrostatics is the Poisson–Boltzmann equation. Several numerical methods have been developed to solve the PB equation; of these, finite difference is the most popular. However, new finite element techniques offer potentially better efficiency through adaptivity and may gain popularity over finite difference methods in future.

Despite over two decades of use, PB methods are still actively developed. The increasing size and number of biomolecular structures have necessitated the development of new parallel finite difference and finite element methods for solving the PB equation. These parallel techniques allow users to leverage supercomputing resources to determine the electrostatic properties of single-structure calculations on large biological systems consisting of millions of atoms. In addition, advances in improving the efficiency of PB force calculations have paved the way for a new era of biomolecular dynamics simulation methods using detailed implicit solvent models. It is anticipated that PB methods will continue to play an important role in computational biology as the study of biological systems grows from macromolecular to cellular scales.

Acknowledgment

Support for this work was provided by the National Partnership for Advanced Computational Infrastructure and Washington University in St. Louis, School of Medicine.