Research Statement
Weihua Geng
Department of Mathematics, University of Michigan

I am interested in modeling problems in structural and systems biology using differential equations and integral equations, and solving these problems numerically with fast algorithms and high performance computing. My main research interest is in numerically solving interface problems described by PDEs and integral equations. These problems appear when adjacent layers with different physical, chemical and biological properties are present. For example, solvated proteins have different permittivities on solutes and solvents, and metal oxide semiconductor field effect transistors (MOSFETs) have different conductivities in metal gates, semiconductors and insulators.

Interface problems have discontinuities in equation coefficients and solution which require special numerical treatment. I am interested in: 1) matched interface and boundary (MIB) methods, and 2) boundary integral methods. MIB methods are high order, finite difference mesh-based algorithms used in solving elliptic PDEs, which have discontinuities in solutions and coefficients, and singularities on interfaces and sources. Boundary integral methods are more general methods to solve integral formulations of PDEs. I am interested in combining boundary integral methods with treecode [1], a fast algorithm for evaluating interactions of N-Body problems, to achieve high accuracy and efficiency. To describe my contribution in developing these methods, which are driven by solving practical problems, I will start by describing problems in structural biology that I have been working recently.

1. Electrostatics Computation and Biomolecular Simulation

It is well known in structural biology that structure determines function. For instance, the amino acid chain of a protein has to be coiled into specific three-dimensional shape to perform its biological function such as enzyme catalysis, signal transduction, and ligand binding. It is also well known that electrostatics, due to their long range and abundant distribution, is critical in determining structures and dynamics of solvated biomolecules. For instance, complementary electrostatics and geometry mutually determines protein-ligand binding. Numerical algorithms and softwares, which accurately and efficiently resolve electrostatics of solvated biomolecules, can play significant roles in assisting the study of protein folding, drug design, chromatin packing and so on. Driven by the practical importance and mathematical challenges of this application, I have been working on electrostatic computation of solvated biomolecules, from modeling to methods and simulations. The work was initiated from my graduate study and further extended in my postdoctoral training.

The problems I have been solving are related to modeling solvated biomolecules. There are typically two solvent models. The explicit solvent model, as shown in Fig. 1(a) with individually represented solvent atoms and mobile ions, captures all the details but requires large degree of freedoms. The implicit solvent model, as illustrated in Fig. 1(b), greatly reduces complexity of the problem by treating solvent as a continuous medium and applying statistical distribution to model mobile ions. The two most popular implicit solvent approaches are: the generalized Born model (GB) [2], which...
Figure 1: (color online) (a) explicit solvent model: a solvated protein (PDB: 451c) is surrounded by explicitly represented water molecules; (b) implicit solvent model: the solvent effect is characterized by different permittivities in domain $\Omega_1$ and $\Omega_2$ separated by molecular surface $\Gamma$; (c) surface potential mapping of the solvated protein: $\phi > 0$ (red), $\phi < 0$ (blue) [4].

is fast but more of an approximation, and the Poisson-Boltzmann (PB) model [3], which is accurate but faces many numerical challenges. My research has focused on the PB model and I plan to extend it to the GB model in the future.

The PB Model in its differential form is given as:

\begin{align}
\nabla^2 \phi_1(x) &= -\sum_{i=1}^{N_c} q_i \delta(x - x_i) \quad \text{in } \Omega_1, \\
\nabla^2 \phi_2(x) - \kappa^2 \sinh \phi_2(x) &= 0 \quad \text{in } \Omega_2, \\
\phi_1(x) &= \phi_2(x), \quad \varepsilon_1 \frac{\partial \phi_1(x)}{\partial \nu} = \varepsilon_2 \frac{\partial \phi_2(x)}{\partial \nu} \quad \text{on } \Gamma = \partial \Omega_1 = \partial \Omega_2, \\
\lim_{|x| \to \infty} \phi_2(x) &= 0,
\end{align}

where $\phi_1$ and $\phi_2$ are the electrostatic potentials in each domain of $\Omega_1$ and $\Omega_2$, $q_i$ is the partial charge carried by the $i$th atom, $i = 1, ..., N_c$, $N_c$ is the number of atoms of the solute molecule, $\varepsilon_1$ and $\varepsilon_2$ are the dielectric constants (permittivity) in solute and solvent respectively, $\kappa$ is the Debye–Hückel parameter representing ion screening effects in $\Omega_2$, and $\nu$ is the unit outward normal on the interface $\Gamma$. In Fig. 1(b), a 2-D illustration of the PB model, the most common choice for the interface $\Gamma$ is the molecular surface (the closed dashed curve) defined by the trace of a spherical probe (the red solid ball) moving over the van der Waals surface of atoms (balls with centered partial charges) in the solute. Eqs. (1) and (2) of the PB model are elliptic equations governing electrostatic potentials across multiple domains with discontinuous coefficients $\varepsilon$, an irregular geometric domain interface $\Gamma$, non-smooth solutions $\phi$ subject to the jump conditions in Eq. (3), singular sources terms ($\delta$ functions), and boundary conditions at infinity in Eq. (4). These features present challenge for numerical simulation in terms of accuracy and efficiency. Figure 1(c) shows that solving the PB model can produce electrostatic potentials on the molecular surface of solvated biomolecules (e.g. Fe
II cytochrome as shown here) [4]. The color-coded surface electrostatic potentials along with surface forces gives important affinity and polarity information, and can be used to study binding, folding, docking, and a series of static and dynamic properties for solvated biomolecules.

There are many PB solvers embedded in popular molecular simulation packages including AMBER, CHARMM, NAMD/VMD etc. However, these solvers apply straightforward finite difference method and lack rigorous treatments of the discontinuities and singularities, suffering from accuracy reduction, particularly near the interfaces. Interface methods improve the accuracy and at the same time achieve efficiency using fast algorithms and high performance computing.

1.1 Matched interface and boundary (MIB) methods

Under the direction of Professor Guowei Wei, my PhD thesis [5] developed a matched interface and boundary methods based PB (MIBPB) solver to rigorously treat discontinuities and singularities. The MIBPB solver: 1) repeatedly uses interface jump conditions to capture the non-smoothness of solutions [6], 2) adaptively applies high order local interpolation to track the geometric singularities [7], and 3) analytically takes Green’s function based decomposition to regularize the singularities of the source terms [4]. By computing a series of benchmark tests, the MIBPB solver has been proved to be by far the most accurate Cartesian grid based PB solver, which now is open source, serving the bio-oriented community [8].

In addition to developing the static MIBPB solver, I also developed a dynamic MIBPB solver, which produces electrostatic solvation forces for molecular dynamics simulation. Developing the dynamic PB solver requires a series of theoretical derivations, and goes through numerical challenges such as electrostatic forces redefinition, computation, integration, and distribution. The dynamic PB solver has been embedded in AMBER to simulate molecular dynamics [9].

1.2 Treecode accelerated PB solver in the boundary integral formulation

During my postdoctoral training with Professor Robert Krasny, I continued working on the PB model, but with a different approach: boundary integral methods [11]. The Cartesian treecode together with its parallelization serves a critical role in accelerating boundary integral methods. Professor Krasny and I submitted a NSF grant proposal in my first semester at University of Michigan. His expertise in treecode and boundary integral method combined with my experience in solving and modeling electrostatics for solvated bimolecules were recognized by experts and we were awarded an NSF Grant to support our research on this project for three years starting September 1, 2009. So far we are near the completion of the boundary integral PB solver, which showed stable convergence rate, designated accuracy, greatly reduced memory use in comparison with finite difference method, and fast speed with convenient parallelization. The corresponding publication is to be submitted [10], and the code will be published after extensive testing. The following are a few important details about this approach.

**Boundary Integral Methods**

The boundary integral formulation of the PB equation achieves both high accuracy and efficiency with
the assistance of the treecode algorithm and parallel computing. For any collocation point $x \in \Gamma$, a well posed integral formulation of PB model [11] can be expressed as a set of coupled equations for the electrostatic potentials $\phi_1$ and their normal derivatives $\frac{\partial \phi_1}{\partial n}$ on the interface:

$$\frac{1}{2} \left( 1 + \varepsilon \right) \phi_1(x) = \int_{\Gamma} \left[ L_1(x, y) \frac{\partial \phi_1(y)}{\partial n_y} + L_2(x, y) \phi_1(y) \right] dS_y + \sum_{k=1}^{N_c} q_k G_0(x, y_k), \quad (5)$$

$$\frac{1}{2} \left( 1 + \frac{1}{\varepsilon} \right) \frac{\partial \phi_1(x)}{\partial n_x} = \int_{\Gamma} \left[ L_3(x, y) \frac{\partial \phi_1(y)}{\partial n_y} + L_4(x, y) \phi_1(y) \right] dS_y + \sum_{k=1}^{N_c} q_k \frac{\partial G_0(x, y_k)}{\partial n_x}, \quad (6)$$

where $L_1-L_4$ are kernels involving Coulomb potential $G_0(x, y) = \frac{1}{4\pi|x-y|}$, Screened Coulomb potential $G_{\kappa}(x, y) = e^{-\kappa|x-y|}/4\pi|x-y|$ and their derivatives. In this context, boundary integral methods have the following advantages: 1) a well-posed Fredholm 2nd kind integral formulation to achieve rapid convergence, 2) fewer unknowns in comparison with methods that discretize the entire domain, 3) exact imposed boundary conditions at infinity, 4) a high precision surface represented by triangular boundary elements, 5) analytically expressed singular source terms, 6) explicitly enforced continuity conditions, and 7) treecode conveniently parallelized. In a related project in progress, we use a domain decomposition method (DDM) preconditioning to further improve the efficiency of the boundary integral method in collaboration with Professor Yassine Boubendir from NJIT.

**Cartesian Treecode**

Cartesian treecode is a tree-based algorithm that replaces particle-particle interactions by particle-cluster interactions, which reduces computational cost from $O(N^2)$ to $O(N \log N)$. The treecode has many numerically attractive features such as 1) wide application in different kernels/interactions such as vortex sheet [12], screened Coulomb potential [13], etc., 2) simplicity in coding and effective recurrence relation in computing Taylor coefficients, and 3) $O(N)$ memory saving nature and convenience in parallelization. Solving Eqs. (5) and (6) requires evaluating particle-particle interactions of kernels $L_1 - L_4$, which can be efficiently accelerated by the treecode. The treecode can also be used to efficiently solve the generalized Born (GB) model, which is included in our research plans.

**Parallel Computing**

The treecode is an $O(N \log N)$ algorithm, which loops over all particles; the clusters are predetermined with moments pre-computed. This feature makes the distributed memory based parallel computing, e.g. MPI, straightforward. In our first paper [10], the numerical results demonstrated nearly perfect parallelization. More importantly, the treecode uses very limited memory compared with other fast algorithms, e.g. FMM, and therefore is a strong candidate for GPU computing. We are investigating different parallelizing and vectorizing algorithms [14].

2. Simulating Combinatorial Complexity in Genetic Networks

I am also interested in problems from systems biology. In systems biology, a major difficulty in simulating genetic networks is combinatorial complexity. Although a network may involve a relatively low number of proteins (e.g. 10), complexes can form leading to a very large ($2^{10}$) number of species
which must be simulated. Even writing down equations of each of these species is very difficult. To simulate these networks, Professor Daniel Forger and I have been working on algorithms and codes to automatically generate and simulate genetic networks with combinatorial complexity. Our simulations demonstrate that the specific mechanisms of complex formation can drastically change the time scales and steady state behavior of these networks. Using these methods we have begun to model one of the most well studied genetic networks with combinatorial complexity: the mammalian circadian (24-hour) clock which times biological events in our body. This work is in preparation [15]. We are also working with a drug company who will use these techniques in drug discovery.

3. Future Research Plans

My graduate and postdoctoral training has taught me advanced numerical methods and provided me experience in modeling and solving problems in structural and systems biology. With this training, I plan to 1) continue developing numerical algorithms in collaboration with faculty in biology/biochemistry/biophysics to assist their study of molecular dynamics and genetic network, and 2) collaborate with faculty in the Engineering school to assist their research by providing solutions to interface problems (e.g. Material Science). In addition here are two areas where my current projects will be extended.

Molecular Dynamics

Molecular dynamics allows scientists to study the motion of biomolecules in atomic details to compare with experiment results, or in a way which is not possible in laboratory experiments. Many important features of the boundary integral PB formulation can be applied in computing electrostatic solvation forces, which are critical in simulating the molecular dynamics of solvated biomolecules. The electrostatic solvation forces are composed of reaction forces, dielectric boundary forces and ionic boundary forces; the last two appear due to the existence of the interface. Computing dielectric and ionic boundary forces requires the normal derivatives of the potentials on both sides of the interface, which is a big challenge for the finite difference method. However, under the boundary integral formulation, their computation can be done in a convenient way [16].

As an application of the treecode accelerated boundary integral PB solver, we are collaborating with Professor Tamar Schlick to study her mesoscopic and coarse-grained models of chromatin packing [17]. She invited me to her lab for one-week training in chromatin packing. The boundary integral PB solver will assist the study of chromatin packing by producing more accurate electrostatic potentials and fields. Furthermore, the study of ionic atmosphere affecting chromatin packing inspires us to develop an adaptive nonlinear PB solver.

We also plan to implement a treecode-accelerated electrostatic forces driver in a large-scale molecular dynamics software package such as AMBER/CHARMM/NAMD. Professor Klaus Schulten, the leading developer of NAMD/VMD from UIUC [18], has shown interest in our approach when I attended the NAMD/VMD biophysics workshop at UCSD this summer.
**Improved and Extended PB Model**

The PB model is effective but has inherent limitations. For example, the PB model cannot characterize hydrophobic effects, which are critical in the folding process of globular proteins with hydrophobic cores. The PB model also does not include the viscosity that water molecules contribute by randomly colliding and impeding the motion of solutes through their van der Waals repulsion. Furthermore, the PB model cannot trace the directionality of protein-water hydrogen bonds and strong ionic effects. I expect to collaborate with biophysicists and biochemists to improve the PB model with the assistance of our PB solver. In addition, the PB model can be extended to solve more complicated coupled systems. For example, the Poisson-Nernst-Planck equation can simulate electro-diffusion by describing the flux of ions under the influence of both an ionic concentration and an electric field [19], and the coupled Poisson-Schrödinger equations can be used to model and simulate nano-electronic structures [20]. The integral formulation of PB model can be incorporated in these extended PB models, utilizing its advantage in both efficiency and accuracy.

4. Summary

Scientific computing and mathematical biology are some of the most exciting and challenging interdisciplinary research areas. Researches on structural and systems biology evolve with the increased scale and complexity, and rely more and more on numerical algorithms and high performance computing. My achievements during graduate and postdoctoral training have demonstrated my ability to conduct independent and collaborated research on these "hot" topics. As a motivated individual, taking academia as my career, I am looking forward to new challenges.

References


