UNIVERSITY OF CALIFORNIA,
IRVINE

Adaptation of PBSA and MMPBSA Methodology For Application to Membrane Protein Binding Free Energy Calculations

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTORATE OF PHILOSOPHY

in Chemistry

by

Wesley Michael Botello-Smith

Thesis Committee:
Professor Ray Luo, Chair
Professor Douglas Tobias, Co-Chair
Professor Ioan Andrichioaei

DEDICATION

To

my parents, friends and family,

but most of all,

To my loving wife,

in recognition of their worth,

and without whom I could not have come this far.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Page

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Luo, for his patience and guidance throughout my time at UCI. I would also like to acknowledge my lab colleagues, present and former, and in particular Mengjuei Hsieh, a former lab mate and the previous de-facto tech support of the lab, who was of great help when I was learning the ins and outs of scripting and coding for the PBSA program around which so much of my research revolved. I would also like to thank my collaborator Alec Follmer for all of his help during the second stage of the P2Y12R project. Lastly, I would like to acknowledge the NIH for its financial support as this work was supported in part by NIH (GM093040 & GM079383).

CURRICULUM VITAE

Wesley Michael Botello-Smith

2001-09	Learning Facilitator, Tutorial Learning Center, Santa Ana Community College
2004	A.S. in Mathematics, Santa Ana Community College
2004	A.S. in Chemistry, Santa Ana Community College
2005	STARS Research Intership, Yale
2005-09	Academic Excellence Workshop Facilitator, California State Polytechnic University of Pomona
2009	B.S. in Chemistry, California State Polytechnic University of Pomona
2009-15	Teaching Assistant, Chemistry, University of California, Irvine
2009-15	Research Assistant, Luo Lab, University of California, Irvine
2015	PhD in Chemistry, University of California, Irvine

FIELD OF STUDY

Chemistry, emphasis on computational modeling and simulation of biomolecules

PUBLICATIONS

Smith WM, Prediction of Alternate Enzyme Conformations Via Steered Molecular Dynamics. The SACNAS Conference, Denver (2005)

Smith, WM, et al, Molecular Dynamics Simulations of Surfactant Protein C Mimic in Phospholipid Bilayers, American Chemical Society National Meeting, New Orleans, April (2008)

Botello-Smith WM, Liu X, Cai Q, Li Z, Zhao H, Luo R, Numerical Poisson-Boltzmann Model for Continuum Membrane Systems. Chemical Physics Letters. 555:274-281 (2013)

Botello-Smith WM, et al, Numerical Solutions of the Poisson Boltzmann Equation for Implicit Membrane, Oral Presentation at the American Chemical Society National Meeting, Indianapolis, August (2013)

Botello-Smith WM, Cai Q, Luo R, Biological Applications of Classical Electrostatics Methods. Journal of Theoretical and Computational Chemistry. 13: 1440008 (2014)

Botello-Smith WM and Luo R, Applications of MMPBSA to Membrane Proteins I: Efficient Numerical Solutions of Periodic Poisson-Boltzmann Equation. Journal of Chemical Information and Modeling. September. 55:2187-2199 (2015)

Botello-Smith WM and Luo R, Applications of MMPBSA to Membrane Proteins II: Effects of Solution Environments and Nonpolar Solvation Upon Binding Affinity Prediction. Journal of Chemical Information and Modeling. In preparation

ABSTRACT OF THE DISSERTATION

Adaptation of PBSA and MMPBSA Methodology For Application to Membrane Protein Binding Free Energy Calculations

By

Wesley Michael Botello-Smith

Doctorate of Philosophy in Chemistry

University of California, Irvine, 2015

Professor Ray Luo, Chair

Membrane proteins are receiving an increase in attention due to their rolls in cell signaling and recognition. In particular, they are prime targets for drug design efforts. Computational prediction of binding affinity provides a useful prescreening and ranking tool for rational drug design, which may be attained using binding free energy calculations. Implicit solvent based methodologies provide efficient means of performing such computations. While generalized implicit membrane solvation has been mainly supported under Generalized Born based methodologies, which seek to approximate full Poisson-Boltzmann based computations, extension of numerical Poisson-Boltzmann solvers has occurred only relatively recently. Incorporation of implicit membrane models into the MM-PBSA framework in AMBER under the PBSA module required extension of existing accelerated linear solvers to allow support of periodic boundary conditions. Finally, the MM-PBSA implicit membrane methodology was demonstrated using the human purinergic platelet receptor (P2Y12R), for which structural and experimental binding data was recently released to the protein data bank. This included detailed examination of relevant parameters, such as choice of non-polar solvation term and selection of appropriate protein and membrane dielectric. Multi-trajectory methodology was also investigated briefly.

INTRODUCTION: Biological Applications of Classical Electrostatics Methods

Introduction

Long-range electrostatic interactions are of crucial importance to proper understanding and modeling of biomolecular structure and function.[1] Although quantum mechanical approaches provide the most detailed and accurate description of molecular structure and function, there are many cases where classical electrostatics modeling is shown to be highly effective. Such models are widely employed in applications involving highly charged biomolecules,[2] which are often far too large and / or require time scales too long to be amenable to full quantum mechanical approaches.

A key factor to be considered when modeling a biological system is the effect of solvent molecules. Most biomolecules exist and function in aqueous environments. Indeed, even membrane-bound proteins experience at least some contact with surrounding water molecules. Interactions between biomolecules and their surrounding environment play a vital role in the behavior of biomolecular systems, so that models and simulations must include relevant solute-solvent interactions.

Solvent-solute interactions are included either explicitly (i.e. by modeling each individual molecule) or implicitly (by attempting to replicate their average effect). In explicit solvent models, electrostatic interactions over all explicitly represented atoms are treated in a pair-wise fashion using Coulomb’s law. In such cases, a large amount of computational time is often devoted to model the interactions of individual atoms of the water molecules. These solvent molecules often greatly outnumber the solute molecules atoms, particularly if one needs to model a relatively large or dilute system. In such cases, periodic boundary conditions are often employed to mimic a bulk water box surrounding a solute molecule. Such setups are particularly attractive as they are conducive to the Ewald summation technique which allows for efficient computation of electrostatic forces and energies.[3]

Poisson-Boltzmann Based Implicit Solvent Methods

In most studies of biomolecules, it is the solute and not the solvent that is of interest. Thus, it often seems wasteful to devote large portions of computational time simulating water explicitly. Implicit solvation methods attempt to approximate the relevant interactions between solute and solvent so as to alleviate the resources spent upon modeling its effects. Although this approach is less accurate due to the loss of the fine-grained details of individual solute-solvent interactions, implicit models are quite useful in many biological applications. Most approaches are either based on statistical mechanics considerations or parameterized to replicate empirical results. Thus they not only reduce the number of particles to be simulated but also alleviate the need for long simulation trajectories that are required under explicit solvent models to reproduce statistical averaging of physical properties as observed experimentally.

A key component to a class of the most widely used implicit solvation methods is the Poisson-Boltzmann (PB) equation, which is employed to model solvent-induced electrostatic properties. The water is modeled as a region of high dielectric while the solute is modeled as a region of low dielectric according to Poisson's equation.[4] The solvation energy thus becomes a mean force potential, which is averaged over all the solvent degrees of freedom.[5]

In many biological systems, salt ions are also present in addition to the solute. Their inclusion is necessary to properly model relevant solvent effects. The PB equation models the salt ions as mobile charge densities, which obey a Boltzmann distribution at equilibrium. Integrating this into Poisson's equation yields: $\nabla \cdot \varepsilon \nabla \phi = -4\pi\rho_0 - 4\pi \sum_i e z_i c_i \exp\left(-e z_i \phi / k_B T\right)$ (1), where ε is the dielectric constant, ρ_0 is the solute charge density, e is the unit charge, z_i is the valence of ion type i , c_i is the number density of ion type i, k_B is the Boltzmann constant and T is absolute temperature. This is a non-linear elliptical partial differential equation in three dimensions. Its complexity is such that solutions of all but the simplest systems require sophisticated numerical solution methods. Fortunately, in many instances, it is possible to simplify this equation somewhat.

In cases where both the ionic strength and solvent potential are low,[6] and when symmetric electrolytes are considered, the PB equation can be linearize $\nabla \cdot \varepsilon \nabla \phi = -4\pi\rho_0 + \varepsilon_v \kappa^2 \phi$ (2), where, $\kappa^2 = \dfrac{8\pi e^2 I}{\varepsilon_v k_B T}$ and $C = \dfrac{ez}{k_B T}$. Here v denotes the solvent; I represents the ionic strength of the solution, and is computed as $I = z^2 c$.

Even in its linearized form, use of the PB equation requires numerical solution methods. Thus, simpler methods were developed for approximating continuum electrostatic solvation models such as the induced multipole model,[7] analytical continuum methods,[8] dielectric screening model[9] and the Generalized Born (GB) model.[10] Although an in-depth discussion of each model is beyond the scope of this paper, it is worth noting that the GB model has become one of the primary methods of choice for incorporating implicit solvation into dynamics simulations.[11]

The solution of the PB equation (eq. (1)) that one arrives at is dependent upon how the underlying model is set up. Solute charge modeling, spatial boundary conditions, solute-solvent interface description, and dielectric determination are just some of the many issues that can be considered when optimizing a PB-based implicit solvation model. Some of these issues are to be discussed in greater depth in the Force Field section below.

Point charge models are widely used as representations for atomic charge densities, ρ_0 in eq. (1). While this choice simplifies the initial setup of solute charge density descriptions, the singularities introduced by use of the delta function for ρ_0 result in complications when subsequently pursuing numerical solutions to the PB equation.[12] Regularization of eq. (1) can remediate this difficulty. Some common regularization strategies are: to decompose the solution as regular singular components then focus numerical solutions upon the regular component,[13] to introduce an additional harmonic term into a Green's function of the point charge model,[14] or the use of a matched interface boundary approach.[15] While analytical computation of the singular component requires significant effort, it allows for improved convergence and saves additional computing time during reaction energy calculations.[13]

Eq (1) cannot be solved directly in an infinite space so that either spatial boundary conditions are enforced or the portion of the infinite space exterior to the grid is handled specially.[14] Although there are a few options for implementing spatial boundary conditions, the most common choice is to mimic a free space solution. This requires leaving adequate spacing between the solute and the sides of the computational box and applying a fixed charge or potential along the spatial boundary. Doing so exactly requires computing the solution at each point along the boundary, which becomes infeasible for complex systems. The Debye-Hückel equation provides a popular alternative approximation: $\phi^{BC} = \frac{1}{\varepsilon_s}\sum_i \frac{Q_i \exp\left[-\kappa\left(R_i - r_i\right)\right]}{R_i\left(1+\kappa r_i\right)}$ (3), where $\sum_i$ sums over all atoms, Q_i is the the ith atom's charge, r_i is its radius, and R_i is the distance between the ith atom and grid center. Eq. (3) is the solution to the linear PB equation for a single spherical low dielectric region surrounded by a high dielectric region with mobile ions. This approximation holds well wherever the linearized PB holds, but it is not guaranteed to be applicable when using the non-linear PB equation.[16]

Extension of PB Solvent Models to Implicit Membranes

Recently membrane proteins have received increasing attention in modeling and simulation studies. Their roles as cell receptors and transmembrane channels make them good candidates for drug targets. Unfortunately, membrane proteins are notoriously difficult to study via crystallization, often requiring specialized detergent agents. Even then, the process in crystallization and resolution of integral membrane protein structure is usually slow, tedious, and difficult. Thus resources for membrane protein structure data are much more sparse while the number of known soluble proteins seems to grow continuously. Furthermore, study of the action of proteins in their native environment is also complicated by the presence of the membrane, particularly if the region of interest resides within the membrane. Nevertheless, modeling and simulation of membrane-bound biomolecular systems has received great attention. Indeed, the difficulties associated with experimental study of membrane proteins are a major impetus for the development of useful computational methodologies in order to supplement experimental research efforts.

Since protein structure and function is extremely sensitive to the surrounding environment, proper inclusion of a membrane is necessary to ensure accuracy when membrane proteins are being studied. Application of rationale design methodologies to membrane proteins requires adaptation of free energy calculation methods in order to include modeling of membrane environments. Here again, one chooses to either model all of the atoms or molecules in the membrane explicitly, or resort to an implicit model.

Inclusion of a membrane explicitly naturally suffers from many of the same issues as explicit water, such as the need for averaging over long trajectories or ensembles. In addition, there are also new challenges to be addressed such as the need to implement surface tension to ensure proper modeling of the membrane's physical properties when employing explicit models.

The PB methodologies discussed above can also be used to provide an implicit membrane model for study of membrane proteins. As in explicit water models, one loses the fine-grained details of membrane-to-solute interactions. However, an implicit approach can provide a reasonable approximation in many cases and alleviates the need to run extensive simulations to ensure converged statistical averaging. Thus, there has been a great deal of effort to extend PB-based implicit solvation methodologies to include a membrane region. While much of this effort has been directed toward adaptation of Generalized Born methodologies[17, 18,19,20,21] there have also been notable advances in implementing implicit membrane models under the full PB framework as well.[22] The inclusion of an implicit membrane region adds additional challenges and parameters to be considered depending on the choice of solvation model / method.

As with implicit water solvation methods, the Generalized Born method has received much attention due to its amenability toward simulations and relative ease of computation. When implementing a membrane under such methods, it is necessary to account for the location of atoms relative to the membrane, since the properties of the membrane do not necessarily match those of either the solvent or the solute.[17, 20-21, 23]

In the case of numerical PB solutions that employ a mesh or grid, the implicit membrane can be implemented as an additional solvent region with relative ease,[22] although there are certainly additional considerations. Typically a relatively low dielectric is used to mimic the membrane's hydrophobic core region.[24] Selection of dielectric constant (or dielectric constant profile) and location of the dielectric boundary between water and membrane can, of course, have a significant impact. Selection of these values depends upon the properties being studied, as well as, the nature of the system being studied.[24a] The effect of the membrane's presence upon the mobile ion distribution modeled in the PB equation must also be considered. Since ions are rarely found within the interiors of biological membranes, the mobile ion term is generally excluded from the membrane region as well. Again, this is generally easy to implement within mesh or grid based approaches. Thus, with respect to electrostatic calculations, the membrane region (or more accurately, its hydrophobic core region) has properties very similar to those of the solute region: i.e. low dielectric constant and lack of mobile electrolyte ions. In the simplest case, it can be viewed as an extension of the solute region. However, a unique dielectric is most commonly desired. In either case, inhomogeneity is introduced into the dielectric profile along the spatial boundaries. Efficient and accurate computation of periodic solutions to the PB equation is apparently not a trivial matter and is to be discussed further in the numerical solvers section.

Visualization and Structural Analyses of Biomolecules

A direct benefit of the PB models is that the electrostatic potential profile becomes available for visualization after the PB equation is solved. Molecular graphics programs have seen a boom over the last decade and there are now a myriad of options available for visualization and even editing of molecular data. Although a complete review is beyond the scope of this article, a brief review of some of the more well known and widely used programs is in order. In particular, focus will be on those with capabilities commonly used for visualization of electrostatic potentials.

Coloring the molecular surface by visualization tools, such as VMD,[25] GRASP,[26] CHIMERA,[27] and PyMOL,[28] according to the solution to the PB equation provides us with insight of the electrostatic properties of biomolecules for structural analysis.[29] The PBSA module of the AMBER package[30] was recently extended to allow output of the electrostatic potential distribution generated as solutions to the PB equation using the DX volumetric data format. This allows the potential to be easily visualized under programs such as VMD and CHIMERA. Further, when a level set based molecular surface description is employed,[31] the same DX format can be used to output the corresponding Solvent Accessible Surface. This can also be visualized in VMD as an appropriate isosurface. APBS also offers a range of options for visualization of electrostatic potentials, including export of electrostatic profiles using volumetric data file formats. The makers of APBS also provide a web service, PDB2PQR,[32] to facilitate setup and calculations for proteins. The service allows visualization via an interface to JMOL, a popular online visualization applet. Moreover, APBS provides a program to allow interactive setup of implicit membrane calculations. Finally, the PBEQ solver also provides a web-based graphical user interface to read biomolecular structures, solve the PB equation, and interactively visualize the electrostatic potential.[33]

One useful application of molecular surface visualization is identification of solute voids and channels. The presence of such interior voids is apparently an important feature with respect to protein stability, folding, and function.[34] As such processes are often difficult to examine experimentally, computational simulation and 3D modeling and visualization can be quite useful. Visualization of the electrostatic potential along the surface of such voids can provide useful insights. Electrostatic potential can have a bearing on the hydrophobicity of a molecular surface, thus the presence of an isolated void or cavity does not necessarily indicate the presence of trapped water. Additionally, with respect to modeling of implicit solvents, such trapped water is likely to exhibit more limited mobility than bulk water and the corresponding effective dielectric constant can thus be expected to be lower than the dielectric constant of the bulk solvent. The presence of trapped water can have an important bearing on the chemical environment of associated side chains. For instance, as mentioned in the discussion on application of PB to pKa determinations, the presence of water has been shown to have a stabilizing effect on ionizable side chains in their protonated or deprotonated states.[35]

Automating the location of internal voids and channels when using the common Richard or van der Waals model is, apparently, not a trivial matter. Several programs and algorithms have been investigated in the past for use with such molecular surface definitions.[15, 36] Level set based molecular surface descriptions, such as the marching front propagation approach[37] or the spline based density model[31] used in the PBSA module of the AMBER package, simplifies this process. Level set methods treat the molecular surface as an isosurface (usually the zero-isosurface) of a volumetric function. This same volumetric data can be used to allow search like algorithms to detect solute voids that are disjoint from the surrounding solvent.[37]

Another very useful and practical application of 3D visualization is in the investigation of membrane channel or gating proteins. Such proteins often contain a solvent filled channel region. If an implicit membrane model is to be used in computing electrostatics, it is first necessary to ensure that the channel region is retained as a solvent region rather than converted to a membrane region. Thus, a means of excluding the membrane dielectric from the channel region is required.

The AMBER[38] and APBS[39] packages accomplish this by allowing definition of a cylindrical region from which the membrane is excluded. Using this method, visualization of the protein prior to running the electrostatic calculations is useful in order to locate and properly model the channel region. The APBS Membrane[22c] package provides a separate graphical interface for this process, allowing the user to define a cylindrical region from which the membrane is excluded. The PBSA module of the AMBER package, which was recently extended to allow for membrane protein computations, provides a similar exclusion region definition. Location of the needed region is accomplished by exporting the molecular surface. This surface is obtained from a preliminary calculation using water-only implicit solvation to capture the molecular surface model, followed by visual inspection of the molecular surface using a 3D visualization program such as VMD. The DelPhi[22a] package can accomplish this process in a similar manner since it allows for inclusion of various user defined geometric regions with varying dielectric constants, i.e. the user can define a cylindrical or conical region with a dielectric constant matching that of the solvent and position it to envelope the channel region. Here again, preliminary visualization is needed to define the exclusion region properly.

Once the channel region and any cavity regions have been defined, the electrostatic potential distribution can be computed and mapped onto the corresponding surface or surfaces to allow visualization of the surface potential. Visualization can provide useful insights into a protein’s function and interactions. For instance, the electrostatic surface potential has a significant bearing on the hydrophobicity of the surface. In the case of a membrane protein, the regions of the protein that lie exterior to the membrane often have a positive or negative potential, whereas those that lie within the membrane tend to be closer to neutral electrostatically. For channel proteins, the channel region often exhibits a non-neutral potential in order to facilitate water or ion transport, whereas the surface of the protein exposed to the membrane is more likely to be neutral. Electrostatic surface potential and polarization is also important to binding and association since binding is stabilized or destabilized based on the electrostatic surface potentials at or near contact regions.

While visualization can provide good qualitative insight even with relatively low resolution and precision, quantitative methods require a high degree of precision to provide accurate results. The precision of a solution to the PB equation is dependent upon both the choice of solver methodology and the choice of force field as well, these are discussed further in the numerical solvers and force field sections below.

Energetics Analyses of Biomolecules

The PB models can be applied to a number of biologically relevant situations. By providing a means to approximate free energies of biomolecules in solution PB implicit solvent models have proven useful for, among other applications, prediction of pKa values for ionizable groups in biomolecules,[40] solvation free energies,[41] binding free energies,[42] and rational protein design.[43]

The MM/PBSA algorithm employs the PB models for calculation of binding free energies at much lower computational cost relative to the analogous free energy perturbation or thermodynamic integration methods that make use of an explicit solvent model.[44] Under the MM/PBSA method, a molecular dynamics simulation of the biomolecule is first performed to generate a trajectory. This trajectory's frames serve as "snapshots" of the simulation, which are then processed using a PB model to estimate conformational free energies.[45] This can provide solvation free energies when employed on a single biomolecule by subtracting the vacuum free energies (no implicit solvent, or ions) from the solvated free energies (implicit water and mobile ions). If one is able to safely assume that no significant structural changes occur upon binding,[44] then this same method can also provide binding free energy estimates using a single trajectory by subtracting the free energies of the free ligand and receptor portions from the free energy of the complex. Brown et al. adapted MM/PBSA methods to a high-throughput work flow and observed a drastic savings in computational expense of up to two orders of magnitude under virtual screening applications.[46]

The PB models show great potential for aiding peptide or protein design. Marshall et al. presented a revised finite-difference PB method with reduced representation of protein surface so that the electrostatic energy becomes pairwise decomposable by side chains and compatible with protein design calculations.[43] Kieslich et al. recently developed a computational framework known as Analysis of Electrostatic Similarity Of Proteins (AESOP) based on the PB electrostatics,[47] and applied this framework to the design of mutant proteins with enhanced immunological activity.[48]

Electrostatic interactions computed using the PB model have been used to predict pKa values for ionizable functional groups in biomolecules such as proteins and nucleic acids.[40] This is accomplished using free energy calculations of the ionizable residue and its conjugate acid or base in water, the corresponding free energies of sidechains, and the pKa of the ionizable group in water.[35] The PB implicit solvation model provides a means by which to estimate these free energies. Identification of enzyme active sites and ligand binding and catalysis studies benefit from such pKa prediction computations.[49] Additionally the use of PB implicit solvent methods for improvement of pKa calculations employed under constant pH simulations has been investigated.[50]

The native environment of membrane proteins is quite different from that of unbound globular proteins. The interior of cell membranes, which is predominantly composed of long hydrocarbon chains, is generally accepted to be quite hydrophobic. Whereas an aqueous environment favors exposure of polar and charged amino acid residues, there is a relatively large thermodynamic barrier for their insertion into the membrane interior. Conformations that limit contact between charged or polar residues and the membrane interior are favored. Similarly, the membrane interior exhibits a much lower dielectric constant than that of water. Thus, the electrostatic environment also enforces a preference for uncharged and non-polar groups near the membrane center since polar and charged groups do not receive the same stabilizing effects within the membrane interior as they do in an aqueous environment. These effects have a significant bearing toward conformational stability and preference within the membrane. The effects of these properties with respect to protein structure have been illustrated via hydropathy plots and subsequently employed for topological predictions since before the turn of the millennium.[51]

Early studies, which implemented membrane regions under Generalized Born methodology based on hydrophobicity and electrostatic considerations, have shown promise for the use of implicit membrane representations in MD simulation, such as correctly predicting insertion of non-polar helices and exclusion of polar residues.[23a, 52] Energetic analysis can provide important information regarding how processes such as insertion, folding, and association are driven. Indeed, over the past decade, rising interest in modeling and simulation of membrane proteins has prompted GB and PB implicit solvent methods to be adapted to allow inclusion of membrane regions.[19-21, 22b] While PB models are more accurate, the GB method has seen greater attention with regard to such applications thus far and has been tested and employed for a number of biologically relevant applications with surprising success. For instance, computing tilt angles and orientations of transmembrane helices,[52b, 53] computation of insertion free energies and folding for proteins and peptides.[23a, 54]

Force Field Considerations

Proper parameterization of implicit models is required in order to produce useful results. Construction of a PB solvent model is based upon system parameters such as atomic partial charges and atomic radii, which are parameterized based on empirical data or theoretical calculations. These quantities, along with all other parameters needed for classical simulations, are stored in databases called force fields.

Force fields are often constructed with particular types of systems or applications in mind. Because of this, transferability to other situations is not necessarily guaranteed. Force fields are a major topic in their own right and a detailed discussion of their design, proper application, derivation, and validation is beyond the scope of this review. However, since choice of force fields often has significant influence upon the quality of results produced with a PB solvent model,[55] a review of relevant considerations with respect to the impact of force field choice upon the accuracy of applied PB models is in order.

Ideally, implicit solvent simulations reproduce results from explicit solvent models and/or experimental results. Unfortunately, direct comparison and assessment of implicit solvent models with experiment is often limited by several factors. Entropy terms can result in differences in free energy results as large as 2 kcal/mol;[56] particularly if the ensemble studied has not sufficiently converged. Additionally, non-polar and polar solvation terms are coupled, making it hard to discern the cause of discrepancies between implicit models and experimental results.[55a] For instance, it is common to estimate non-polar solvation terms as functions of solvent accessible or solvent excluded volumes.[55a] Yet, these same volumes are also used to define the boundaries between dielectric regions in the PB electrostatic model, the topology of which strongly influences the electrostatic free energy. Lastly, the molecular mechanics force fields, being classical approximations, introduce their own errors. This is particularly relevant in the case of large macromolecules since parameterization is limited by the experimental results and/or theoretical capabilities available. For example, experimental results are only available for relatively small, neutral molecules while full quantum mechanics descriptions of electronic structure quickly become computationally intractable as system size increases. Thus PB solvent models are limited not only by their own inherent inaccuracies, but also those of the force fields with which they are employed within. Therefore, comparison of implicit solvent methods with explicit solvation results obtained under the same conditions is a more reasonable test.

Both the topology of the solute-solvent interface and the function used to describe transitioning of the dielectric constant across this interface influence the results obtained from PB solvent models.[57] Thus, proper optimization of cavity radii plays an important role in fine-tuning the performances of the PB solvent models. Several sets of optimized cavity radii are currently available.[55d, 57d, 58] Since most training sets previously used in parameterization tend to focus on small to mid size molecules, transferability to large molecules (usually biomolecules out of the training set) is often an issue. Testing of transferability for cavity radii sets against biomolecules outside their respective training sets does not appear to be well pursued, although it has been brought up by Swanson *et al*.[57d] Nevertheless, transferability of cavity radii cannot be taken for granted, as was illustrated in a study using the NMA dimer as a test case.[55d]

Under the PB method, atomic radii is used to construct the molecular surface and / or volume which in turn is used to define mappings for the dielectric constant and mobile ion distributions. There are a number of possible methods to choose from for the construction of molecular surfaces from atomic radii.

Hard sphere surfaces are among the oldest and most well known. These include the van der Waal surface (VDWS), the solvent accessible surface (SAS),[59] and the solvent excluded surface (SES).[60] While hard sphere surfaces tend to be intuitive, relatively straightforward to implement, and can be computed fairly rapidly, there are also important drawbacks to be noted. For instance, VDWS often introduces potentially unphysical solvent pockets within the solute volume, while SAS can be used to avoid such complications. However, SAS also introduces enlarged atomic cavities that skew electrostatic computations. The SES can overcome these complications and allow PB solvent models to yield electrostatics which agrees with explicit solvent models.[61] However, this surface is not differentiable and thus destabilizes some dynamics simulations.[62] Finally, all hard sphere methods tend to yield surfaces with sharp cusps or edges. This in turn leads to sharp transitions from low dielectric to high dielectric. Such sharp transitions often result in artificial grid / mesh setup sensitivity when using common finite differencing or finite element schemes, once again reducing accuracy and convergence properties of electrostatic computations. Fortunately, harmonic averaging of the solute and solvent dielectric constants provides a means of smoothing these transitions.[63]

Another recent option is to use a smoother Gaussian surface[57b] or a spline-based dielectric surface model.[61] In this approach, a distance-dependent pseudo-density function is used to define each atomic volume. The molecular surface can then be defined as a level set of this function, i.e. the surface is defined as the locus of all points for which the function takes on a particular value. Such methods generally result in much smoother surfaces and various geometric properties, such as local coordinate bases and curvatures, can be computed from the level set, which can prove useful when applying the Immersed Interface Method (IIM) or Immersed Interface and Boundary (MIB) method to improve the accuracy of surface potential and force calculations as will be discussed in the numerical solvers section. Furthermore, use of a level set based surface model has the added advantage of allowing the surface to be exported as a volumetric map over the same grid and format used to export the electrostatic potential. As discussed earlier, this allows it to be easily imported into many modern visualization software packages. The cost of computing this volume exclusion/density function, however, introduces a performance bottleneck. In all cases, these atom-centered surfaces seem to result in overestimation of solvation energies,[64] which has lead to the proposal of a modified Van Der Waals surface to improve the PB methods.[62]

Accurate representation of solute polarization is also an important consideration. For instance, it has long been known that the local electrostatic environment can have an impact on the chemical behavior of side-chain groups, such as modulation of the pKa of ionizable residues (see next section). Polarizable force field approaches, such as the Drude Oscillator,[65] or induced dipole and multipole methods such as those employed in OPLS-AA,[66] CHARMM,[67] AMOEBA,[68] and AMBER,[69] have been receiving increased attention, particularly now that modern CPUs have progressed to the point that the required processing power is more readily available and accessible. While polarizable methods provide a much more detailed and accurate representation of solute electrostatic interactions, point charge representations still seem to be the dominant method choice due to their speed and ease of implementation.

Under many point charge based force fields, permanent dipoles are commonly modeled as atomic partial charges, which are often derived from *ab inito* gas-phase calculations performed for small compounds. This provides electronic densities, which are then fitted to electrostatic potentials (ESP)[70] to produce the needed partial charge data sets. One of the initial drawbacks of this method is that its results are sensitive to molecular conformations. This led to the development of the restrained ESP-fit (RESP) model.[71] Polarization interactions still lead to inaccuracies under this model, particularly since the molecular systems upon which the *ab initio* calculations are typically performed are modeled in the gas phase whereas the biomolecular systems to which they are being here applied exist in the condensed phase. Introduction of scaling factors compensates for this.[72] Such approximations make good sense when conformational flexibility within the system of study is relatively high. However, in cases where polarization plays a key role, more sophisticated methods are warranted.

In lieu of using a polarizable force field, adjustment of solute dielectric constant can be used to account for polarization effects. When using this approach, it is important to note that such approximations are not universally transferable and the value needed to provide good results often varies greatly depending upon application.[73] For instance, in one study the Kirkwood-Fröhlich dielectric theory predicted a dielectric constant between 2.5 and 4 for a folded protein,[74] but this same theory predicted a dielectric constant of 15-40 when applied to dipole fluctuations recorded under MD simulations due to side chain fluctuations. From these findings, it has been deduced that protein dielectric properties are more accurately modeled by a heterogeneous scheme, with relatively low dielectric in core regions where crowding limits conformational motility, and relatively high dielectric in regions where side chain conformations are more fluid-like, such as near the protein's surface.[75] Thus, for applications where polarization properties and interactions play the most dominant role, such as in pKa predictions, a relatively high dielectric constant is used,[76] while applications such as binding affinity, for which non-polar interactions play a strong part, often seem to require a relatively low dielectric constant in order to obtain accurate results.[77] Attempts to efficiently model protein dielectric heterogeneity via distance-dependent dielectric model[78] or nonuniform charge scaling,[79] have had some success, but their results are often difficult to interpret. Alternatively, efforts were also reported to develop a self-consistent continuum polarizable force field with non-unity solute dielectric constant to represent the polarization effect in a continuum manner.[80]

Numerical Solvers

Analytical solution of the PB equation is all but impossible, except in the case of extremely simplistic model systems. Therefore, once the PB model for a system has been defined, a numerical solver is employed to compute the solution. Currently, the major numerical methods that are most widely used fall into one of three broad categories.

The first and most widely used is the finite-difference method (FDM).[81] The popularity of the FDM owes largely to its speed and ease of implementation. The first step in the FDM is to define a grid, a regular cubic lattice, unto which relevant system properties shall be mapped. Point charge based models lend themselves readily to this method. A simple spreading function is often employed here to reduce sensitivity to grid spacing and orientation. The dielectric constant is often mapped to grid edges rather than grid points, for reasons to be discussed shortly. Before this can occur, however, each grid point is first assigned an occupancy value to define its location relative to the molecular volume / surface. Edges connecting grid points that are within the same region are assigned the value appropriate to that region. Edges that traverse the molecular surface are assigned a value based on the choice of a dielectric smoothing option and the relative proportions lying on each side of the interface. The governing partial differential equations are then discretized. In the case of the linearized PB equation, a set of linear algebraic equations results. As mentioned earlier, regularization schemes are often employed in order to reduce the error in potentials computed for grid points that are proximal to point charges. One of the major drawbacks to this method is that potentials near the molecular surface are the most prone to error. Adaptive Cartesian grid schemes provide one route to addressing this issue.[82] Another avenue is to generate additional equations in order to enforce exact boundary conditions at the interface, such as in the matched interface and boundary[83] method and immersed interface method.[84]

Recent developments in the finite-difference methods have also been focused on membrane protein systems. This is accomplished by adding a third dielectric region, often with a unique, relatively low dielectric constant. Unfortunately, this introduces inhomogeneity in the dielectric constant and mobile ion distribution along the spatial boundary. As with explicit membrane models, periodic boundary or mixed boundary conditions help address this problem. This also alleviates the need for computing virtual charges or potentials along the boundary via the Debye-Huckel equation as is required to simulate free boundary conditions. However, depending on the numerical solver method used, introduction of periodic boundary conditions complicates or slows the numerical solution process in other manners. For instance, preconditioned solver methods that are often used to accelerate convergence,[85] such as the incomplete Cholesky preconditioned conjugate gradient method, need to be re-derived in order to satisfy the periodic boundary condition.[86] In the case of boundary / interfacial based approaches, such as the immersed interface method, the membrane region needs additional attention[22b] to account for the introduction of new interfacial boundaries.

The second approach is the finite-element method (FEM).[12, 87] Under this method, a weak variational approach is used to approximate the potential distribution as a superposition of basis functions. The resulting linear or nonlinear system for the coefficients is then solved. One of the major advantages to this method is that it allows an unstructured mesh to be employed, which lends itself well to adaptive fitting schemes. Alternatively, a body-fitted mesh can be employed in order to enhance the quality of the solution near the molecular surface. The primary drawback of this method, however, is that constant re-meshing becomes necessary if the topology of the dielectric mapping changes frequently, such as when a biomolecule undergoes significant conformational changes during the course of a molecular dynamics simulation. This presents a bottleneck to solution efficiency.

The third approach is the boundary-element method (BEM).[88] In BEM, the dielectric boundary is discretized into a surface mesh. Onto this mesh, the Poisson or Poisson-Boltzmann equation is then cast in terms of either induced surface charge[88a, 88d, 88g] or the normal component of electric displacement.[88b, 88c, 88e, 88f] Efficient generation of the surface mesh is critical to the success of this method. MSMS is widely used for this reason.[89] Recently, Chen et al. developed the TMSmesh program, which is capable of handling biomolecules consisting of more than one million atoms.[90] A hybrid FDM / BEM[91] has also been investigated. Research into improving speed and efficiency of BEM methods continues to be an active area, particularly with respect to the development of efficient, parallelizable algorithms.[92]

The aforementioned methods have been incorporated into a number of mainstream PB programs, such as Delphi,[81d, 93] UHBD,[81b] PBEQ,[61] PBSA,[81e] and APBS.[87a] One very interesting and appealing twist to PB solver methods was recently proposed by Fenley et al. by employing a correlated Monte Carlo approach. Solution of the PB equation was achieved with CPU time exhibiting logarithmic scaling with respect to the number of atoms.[94] Such an approach is particularly appealing as research pushes the envelope of system size and complexity.

Distributive and Parallel Implementation

The iterative methods of numerical PB solvers can be loosely grouped into two types: stationary methods and Krylov subspace methods. Since many stationary methods are employed as preconditioners for Krylov subspace methods,[95] however, this is only a loose classification scheme. Stationary methods, such as Jacobi, successive over-relaxation, and Gauss-Seidel, are amenable to distributive computing environments via implementation of special ordering schema such as multi-coloring[96] or multi-splitting approaches.[97]

Adaptation of Krylov subspace methods under distributive computing environments was achieved via various domain decomposition[98] strategies and distributive preconditioners.[99] Hsieh et al. explored a distributive multi-block focusing technique to analyze accuracy and performance requirements for parallelization of numerical PB solutions for large biomolecular systems. A highly scalable parallel implementation of these methods has been incorporated into the AMBER/PBSA program.[100]

Multi-grid based methods[85b, 87a, 101] provide another possible alternative, along with seemingly related electrostatic focusing approaches in PB. Focusing techniques allow efficient calculation of electrostatics by adaptively scaling grid size near the solute without needing to waste resources on fine grid computations in distant solvent regions. Such regions are often not of interest and since the dielectric constant is often uniform, therein, grid size refinement has little impact upon accuracy. The focusing approach computes a series of finite-difference runs, performed with successively finer grids. Each run has boundary conditions calculated from the potential map of its predecessor.[102] While this method can provide good improvement in efficiency, the memory usage for the finest grid is still a consideration. Division of fine grids into multiple blocks has been investigated as a means to break the problem into pieces that can be managed by multiple individual computing nodes.[39]

Lastly, recent developments in graph analysis / combinatorial solvers,[103] shows promise for providing an alternative class of numerical solvers that would be highly amenable to distributive computing. However, these methods are apparently still maturing and their applications to PB methods have not yet been investigated.

Limitations

In the studies of comparison between the PB model and the explicit water model, water mediated salt-bridging is one of the major discrepancies, although overall agreement between the two models can be observed.[104] This problem can hardly be solved unless explicit water molecules are present surrounding the solute in the model, and therefore the implicit/explicit model has received attention.[105] More research work has been conducted on the effects of ion size and a modified Poisson-Boltzmann model has been accordingly used to include these effects.[106]

A study by Mobley et al., on hydration behavior of polar solutes, found solvation of solutes exhibiting dense negative charge distribution counter-balanced by diffuse positive charge distribution are more energetically favored than solvation of solutes with opposite charge distributions due to the asymmetric distribution of charge centers in water molecules. This behavior cannot be captured by current implicit PB models, which exhibit a symmetrical response with respect to charge distribution.[107] However, in cases where solvent model details like charge symmetry or distribution profiles need to be incorporated as, for example, when studying protein hydration and aggregation,[108] a statistical mechanics approach such as RISM[109] may provide a viable alternative avenue for modeling solvation effects implicitly.

Estimation of the entropy under the PB model is most often accomplished via a quasi-harmonic analysis or normal mode analysis. Unfortunately, these methods are often unreliable,[110] difficult to converge, and / or applicable only to systems which exhibit small conformational changes.[110-111] Moreover, when studying binding between highly polar compounds, deviations of the resulting estimates have been shown to be very large in some cases.[112] Inclusion of a buffer region during normal mode analysis has been suggested to address this problem, and has shown some utility toward mitigating these deviations.[113] An alternative solution, proposed by Wittayanarakul et al. is to include explicit water molecules in the PB model. This was shown to significantly improve the entropy estimate.[114] In such cases, the MM/PBSA provides superior performance for analysis and comparison of the relative binding energies for a set of similar ligands.[44]

Lastly, while PB methods have been proven useful in molecular dynamics studies of small organic molecules,[115] such approaches often become intractable for large biomolecules. Indeed, the per-step simulation cost when employing FD based PB solvent models is higher than when explicit water is employed. This holds even with grid spacing as coarse as 1/4 Å. The upside, fortunately, is that implicit solvents do not require the long simulation durations needed under explicit solvent models, which require either a long simulation time or an ensemble of replicas in order to properly represent a converged average of solvent solute interactions. Nevertheless, the inefficiency of current PB methods, when applied to large systems, sharply limits its practical applications toward dynamics simulations of macromolecules.

Conclusion

Much effort has been invested to improve the accuracy and performance of the Poisson-Boltzmann solvent models, which has been instrumental in promoting their acceptance in studying interesting biological problems. Despite the past successes, however, there is still much that needs to be done. These methods require improvement to their utility and transferability from the small molecular systems, upon which they were developed and validated, to the larger and more complex macromolecular systems being pursued as advances in technology continue to push forward and provide greater computational resources to researchers.

References

1. (a) Davis, M. E.; McCammon, J. A., *Chemical Reviews* **1990,** *90* (3), 509-521; (b) Perutz, M. F., *Science* **1978,** *201* (4362), 1187-1191.
2. Honig, B.; Nicholls, A., *Science* **1995,** *268* (5214), 1144-1149.
3. Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G., *J. Chem. Phys.* **1995,** *103* (19), 8577-8593.
4. Warwicker, J.; Watson, H. C., *Journal of Molecular Biology* **1982,** *157* (4), 671-679.
5. Wang, J.; Tan, C. H.; Tan, Y. H.; Lu, Q.; Luo, R., *Communications in Computational Physics* **2008,** *3* (5), 1010-1031.
6. Hill, T. L., Dilute Electrolyte Solutions and Plasmas. In *An Introduction to Statistical Thermodynamics*, Dover Publications, Inc.: New York, 1986; pp 321-331.
7. Davis, M. E., *J. Chem. Phys.* **1994,** *100* (7), 5149-5159.
8. Schaefer, M.; Karplus, M., *Journal of Physical Chemistry* **1996,** *100* (5), 1578-1599.
9. Luo, R.; Moult, J.; Gilson, M. K., *Journal Of Physical Chemistry B* **1997,** *101* (51), 11226-11236.
10. (a) Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T., *Journal of the American Chemical Society* **1990,** *112* (16), 6127-6129; (b) Onufriev, A.; Case, D. A.; Bashford, D., *Journal of Computational Chemistry* **2002,** *23* (14), 1297-1304.
11. Shivakumar, D.; Deng, Y.; Roux, B., *Journal of Chemical Theory and Computation* **2009,** *5* (4), 919-930.
12. Chen, L.; Holst, M. J.; Xu, J. C., *SIAM Journal on Numerical Analysis* **2007,** *45*, 2298-2320.
13. Cai, Q.; Wang, J.; Zhao, H. K.; Luo, R., *J. Chem. Phys.* **2009,** *130* (14), 145101.
14. Chern, I.; Liu, J.-G.; Wang, W.-C., *Methods and Applications of Analysis* **2003,** *10* (2), 309-328.
15. Geng, W.; Yu, S.; Wei, G., *The Journal of chemical physics* **2007,** *127*, 114106.

16. (a) Rocchia, W., *Mathematical and Computer Modelling* **2005,** *41* (10), 1109-1118; (b) Boschitsch, A. H.; Fenley, M. O., *Journal of Computational Chemistry* **2007,** *28* (5), 909-921.
17. Spassov, V. Z., Yan, L., and Szalma, S. , *J Phys. Chem. B* **2002,** *106* (8726-38).
18. Forsten, K. E.; Kozack, R. E.; Lauffenburger, D. A.; Subramaniam, S., *Journal of Physical Chemistry* **1994,** *98* (21), 5580-5586.
19. Im, W., Feigh, M., and Brooks III, C. L. , *Biophysical Journal* **2003,** *85*, 2900-18.
20. Tanizaki, S.; Feig, M., *Journal of Physical Chemistry B* **2006,** *110* (1), 548-556.
21. Tanizaki, S.; Feig, M., *J. Chem. Phys.* **2005,** *122* (12).
22. (a) Rocchia, W.; Sridharan, S.; Nicholls, A.; Alexov, E.; Chiabrera, A.; Honig, B., *Journal Of Computational Chemistry* **2002,** *23* (1), 128-137; (b) Botello-Smith, W. M.; Liu, X.; Cai, Q.; Li, Z.; Zhao, H.; Luo, R., *Chemical Physics Letters* **2012**; (c) Callenberg, K. M.; Choudhary, O. P.; de Forest, G. L.; Gohara, D. W.; Baker, N. A.; Grabe, M., *Plos One* **2010,** *5* (9).
23. (a) Ulmschneider, M. B.; Ulmschneider, J. P.; Sansom, M. S.; Di Nola, A., *Biophysical Journal* **2007,** *92* (7), 2338-2349; (b) Im, W.; Feig, M.; Brooks III, C. L., *Biophysical Journal* **2003,** *85* (5), 2900-2918.
24. (a) Nymeyer, H.; Zhou, H.-X., *Biophysical Journal* **2008,** *94* (4), 1185-1193; (b) Stern, H. A.; Feller, S. E., *The Journal of chemical physics* **2003,** *118*, 3401.
25. Humphrey, W.; Dalke, A.; Schulten, K., *Journal of Molecular Graphics & Modelling* **1996,** *14* (1), 33-38.
26. Nicholls, A.; Sharp, K. A.; Honig, B., *Proteins* **1991,** *11* (4), 281-296.
27. Pettersen, E. F.; Goddard, T. D.; Huang, C. C.; Couch, G. S.; Greenblatt, D. M.; Meng, E. C.; Ferrin, T. E., *Journal of Computational Chemistry* **2004,** *25* (13), 1605-1612.
28. Schrodinger, LLC, The PyMOL Molecular Graphics System, Version 1.3r1. 2010.
29. Yang, X. L.; Xie, J.; Niu, B.; Hu, X. N.; Gao, Y.; Xiang, Q.; Zhang, Y. H.; Guo, Y.; Zhang, Z. G., *Journal of Molecular Graphics & Modelling* **2005,** *23* (5), 389-394.
30. Case, D.; Darden, T.; Cheatham III, T.; Simmerling, C.; Wang, J.; Duke, R.; Luo, R.; Walker, R.; Zhang, W.; Merz, K., *University of California, San Francisco* **2012**.
31. Ye, X.; Wang, J.; Luo, R., *J. Chem. Theory Comput.* **2010,** *6* (4), 1157-1169.
32. Unni, S.; Huang, Y.; Hanson, R. M.; Tobias, M.; Krishnan, S.; Li, W. W.; Nielsen, J. E.; Baker, N. A., *Journal of Computational Chemistry* **2011,** *32* (7), 1488-1491.
33. Jo, S.; Vargyas, M.; Vasko-Szedlar, J.; Roux, B.; Im, W., *Nucleic Acids Research* **2008,** *36*, W270-W275.
34. Cuff, A. L.; Martin, A. C. R., *Journal of Molecular Biology* **2004,** *344* (5), 1199-1209.
35. Alexov, E.; Mehler, E. L.; Baker, N.; Baptista, A. M.; Huang, Y.; Milletti, F.; Nielsen, J. E.; Farrell, D.; Carstensen, T.; Olsson, M. H. M.; Shen, J. K.; Warwicker, J.; Williams, S.; Word, J. M., *Proteins-Structure Function and Bioinformatics* **2011,** *79* (12), 3260-3275.
36. (a) Dundas, J.; Ouyang, Z.; Tseng, J.; Binkowski, A.; Turpaz, Y.; Liang, J., *Nucleic Acids Research* **2006,** *34*, W116-W118; (b) Petrek, M.; Otyepka, M.; Banas, P.; Kosinova, P.; Koca, J.; Damborsky, J., *Bmc Bioinformatics* **2006,** *7*; (c) Sonavane, S.; Chakrabarti, P., *Plos Comput Biol* **2008,** *4* (9); (d) Bakowies, D.; van Gunsteren, W. F., *Proteins* **2002,** *47* (4), 534-545; (e) Krone, M.; Falk, M.; Rehm, S.; Pleiss, J.; Ertl, T., *Comput Graph Forum* **2011,** *30* (3), 673-682.
37. Can, T.; Chen, C. I.; Wang, Y. F., *Journal of Molecular Graphics & Modelling* **2006,** *25* (4), 442-454.

38. Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J., *Journal of Computational Chemistry* **2005,** *26* (16), 1668-1688.

39. Baker, N. A.; Sept, D.; Joseph, S.; Holst, M. J.; McCammon, J. A., *Proceedings of the National Academy of Sciences of the United States of America* **2001,** *98* (18), 10037-10041.

40. (a) Georgescu, R. E.; Alexov, E. G.; Gunner, M. R., *Biophysical Journal* **2002,** *83* (4), 1731-1748; (b) Nielsen, J. E.; McCammon, J. A., *Protein Sci.* **2003,** *12* (2), 313-326; (c) Warwicker, J., *Protein Sci.* **2004,** *13* (10), 2793-2805; (d) Tang, C. L.; Alexov, E.; Pyle, A. M.; Honig, B., *Journal of Molecular Biology* **2007,** *366* (5), 1475-1496.

41. (a) Shivakumar, D.; Deng, Y. Q.; Roux, B., *J. Chem. Theory Comput.* **2009,** *5* (4), 919-930; (b) Nicholls, A.; Mobley, D. L.; Guthrie, J. P.; Chodera, J. D.; Bayly, C. I.; Cooper, M. D.; Pande, V. S., *Journal of Medicinal Chemistry* **2008,** *51* (4), 769-779.

42. (a) Swanson, J. M. J.; Henchman, R. H.; McCammon, J. A., *Biophysical Journal* **2004,** *86* (1), 67-74; (b) Bertonati, C.; Honig, B.; Alexov, E., *Biophysical Journal* **2007,** *92* (6), 1891-1899; (c) Brice, A. R.; Dominy, B. N., *Journal of Computational Chemistry* **2011,** *32* (7), 1431-1440.

43. Marshall, S. A.; Vizcarra, C. L.; Mayo, S. L., *Protein Sci.* **2005,** *14* (5), 1293-1304.

44. Homeyer, N.; Gohlke, H., *Molecular Informatics* **2012,** *31* (2), 114-122.

45. Kollman, P. A.; Massova, I.; Reyes, C.; Kuhn, B.; Huo, S. H.; Chong, L.; Lee, M.; Lee, T.; Duan, Y.; Wang, W.; Donini, O.; Cieplak, P.; Srinivasan, J.; Case, D. A.; Cheatham, T. E., *Accounts of Chemical Research* **2000,** *33* (12), 889-897.

46. Brown, S. P.; Muchmore, S. W., *Journal of Chemical Information and Modeling* **2007,** *47* (4), 1493-1503.

47. Kieslich, C. A.; Morikis, D.; Yang, J.; Gunopulos, D., *Biotechnology Progress* **2011,** *27* (2), 316-325.

48. Pyaram, K.; Kieslich, C. A.; Yadav, V. N.; Morikis, D.; Sahu, A., *Journal of Immunology* **2010,** *184* (4), 1956-1967.

49. Baker, N. A., *Numerical Computer Methods, Pt D* **2004,** *383*, 94-+.

50. (a) Dlugosz, M.; Antosiewicz, J. M.; Robertson, A. D., *Phys. Rev. E* **2004,** *69* (2), 021915; (b) Dlugosz, M.; Antosiewicz, J. M., *Journal Of Physical Chemistry B* **2005,** *109* (28), 13777-13784; (c) Machuqueiro, M.; Baptista, A. M., *Journal of Physical Chemistry B* **2006,** *110* (6), 2927-2933.

51. (a) Tusnady, G. E.; Simon, I., *Journal of Molecular Biology* **1998,** *283* (2), 489-506; (b) White, S. H., Hydropathy plots and the prediction of membrane protein topology. In *Membrane Protein Structure*, Springer: 1994; pp 97-124; (c) Von Heijne, G., *Journal of Molecular Biology* **1992,** *225* (2), 487-494.

52. (a) Bond, P. J.; Sansom, M. S., *Journal of the American Chemical Society* **2006,** *128* (8), 2697-2704; (b) Ulmschneider, M. B.; Sansom, M. S.; Di Nola, A., *Biophysical Journal* **2006,** *90* (5), 1650-1660.

53. Lomize, A. L.; Pogozheva, I. D., Solvation Models and Computational Prediction of Orientations of Peptides and Proteins in Membranes. In *Membrane Proteins*, Springer: 2013; pp 125-142.

54. Ulmschneider, M. B.; Ulmschneider, J. P., *Molecular Simulations and Biomembranes: From Biophysics to Function* **2010,** *20*, 91.

55. (a) Tan, C.; Tan, Y.-H.; Luo, R., *Journal of Physical Chemistry B* **2007,** *111* (42), 12263-12274; (b) Lwin, T. Z.; Zhou, R. H.; Luo, R., *Journal of Chemical Physics* **2006,** *124* (3); (c)

Wen, E. Z.; Hsieh, M. J.; Kollman, P. A.; Luo, R., *Journal of Molecular Graphics & Modelling* **2004,** *22* (5), 415-424; (d) Tan, C.; Yang, L.; Luo, R., *Journal of Physical Chemistry B* **2006,** *110* (37), 18680-18687; (e) Korman, T. P.; Tan, Y.-H.; Wong, J.; Luo, R.; Tsai, S.-C., *Biochemistry* **2008,** *47* (7), 1837-1847; (f) Wen, E. Z.; Luo, R., *Journal of Chemical Physics* **2004,** *121* (5), 2412-2421.

56. Mobley, D. L.; Dill, K. A.; Chodera, J. D., *Journal of Physical Chemistry B* **2008,** *112* (3), 938-946.

57. (a) Wagoner, J.; Baker, N. A., *Journal of Computational Chemistry* **2004,** *25* (13), 1623-1629; (b) Grant, J. A.; Pickup, B. T.; Nicholls, A., *Journal Of Computational Chemistry* **2001,** *22* (6), 608-640; (c) Nina, M.; Im, W.; Roux, B., *Biophysical Chemistry* **1999,** *78* (1-2), 89-96; (d) Swanson, J. M. J.; Adcock, S. A.; McCammon, J. A., *Journal Of Chemical Theory And Computation* **2005,** *1* (3), 484-493.

58. (a) Banavali, N. K.; Roux, B., *Journal of Physical Chemistry B* **2002,** *106* (42), 11026-11035; (b) Nina, M.; Beglov, D.; Roux, B., *Journal of Physical Chemistry B* **1997,** *101* (26), 5239-5248; (c) Sitkoff, D.; Sharp, K. A.; Honig, B., *Journal of Physical Chemistry* **1994,** *98* (7), 1978-1988.

59. Lee, B.; Richards, F. M., *Journal of Molecular Biology* **1971,** *55* (3), 379-&.

60. Richards, F. M., *Annual Review of Biophysics and Bioengineering* **1977,** *6*, 151-176.

61. Im, W.; Beglov, D.; Roux, B., *Computer Physics Communications* **1998,** *111* (1-3), 59-75.

62. Lu, Q.; Luo, R., *Journal of Chemical Physics* **2003,** *119* (21), 11035-11047.

63. Davis, M. E.; McCammon, J. A., *Journal Of Computational Chemistry* **1991,** *12* (7), 909-912.

64. Swanson, J. M. J.; Mongan, J.; McCammon, J. A., *Journal Of Physical Chemistry B* **2005,** *109* (31), 14769-14772.

65. Jiang, W.; Hardy, D. J.; Phillips, J. C.; MacKerell Jr, A. D.; Schulten, K.; Roux, B., *The journal of physical chemistry letters* **2010,** *2* (2), 87-92.

66. Jorgensen, W. L.; Maxwell, D. S.; TiradoRives, J., *Journal of the American Chemical Society* **1996,** *118* (45), 11225-11236.

67. (a) Patel, S.; Brooks, C. L., *Journal of Computational Chemistry* **2004,** *25* (1), 1-15; (b) Patel, S.; Mackerell, A. D.; Brooks, C. L., *Journal of Computational Chemistry* **2004,** *25* (12), 1504-1514.

68. Schnieders, M. J.; Baker, N. A.; Ren, P. Y.; Ponder, J. W., *J. Chem. Phys.* **2007,** *126* (12), 124114.

69. (a) Wang, J.; Cieplak, P.; Li, J.; Hou, T.; Luo, R.; Duan, Y., *Journal of Physical Chemistry B* **2011,** *115* (12), 3091-3099; (b) Wang, J.; Cieplak, P.; Li, J.; Wang, J.; Cai, Q.; Hsieh, M.; Lei, H.; Luo, R.; Duan, Y., *Journal of Physical Chemistry B* **2011,** *115* (12), 3100-3111; (c) Wang, J.; Cieplak, P.; Cai, Q.; Hsieh, M. J.; Wang, J. M.; Duan, Y.; Luo, R., *Journal of Physical Chemistry B* **2012,** *116* (28), 7999-8008; (d) Wang, J. M.; Cieplak, P.; Li, J.; Cai, Q.; Hsieh, M. J.; Luo, R.; Duan, Y., *Journal of Physical Chemistry B* **2012,** *116* (24), 7088-7101.

70. (a) Weiner, S. J.; Kollman, P. A.; Case, D. A.; Singh, U. C.; Ghio, C.; Alagona, G.; Profeta, S.; Weiner, P., *Journal of the American Chemical Society* **1984,** *106* (3), 765-784; (b) Weiner, S. J.; Kollman, P. A.; Nguyen, D. T.; Case, D. A., *Journal of Computational Chemistry* **1986,** *7* (2), 230-252.

71. Bayly, C. I.; Cieplak, P.; Cornell, W. D.; Kollman, P. A., *Journal of Physical Chemistry* **1993,** *97* (40), 10269-10280.

72. (a) Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A., *Journal of the American Chemical Society* **1995,** *117* (19), 5179-5197; (b) Duan, Y.; Wu, C.; Chowdhury, S.; Lee, M. C.; Xiong, G. M.; Zhang, W.; Yang, R.; Cieplak, P.; Luo, R.; Lee, T.; Caldwell, J.; Wang, J. M.; Kollman, P., *Journal of Computational Chemistry* **2003,** *24* (16), 1999-2012.

73. (a) Kukic, P.; Nielsen, J. E., *Future Medicinal Chemistry* **2010,** *2* (4), 647-666; (b) Sheinerman, F. B.; Norel, R.; Honig, B., *Current Opinion in Structural Biology* **2000,** *10* (2), 153-159.

74. Gilson, M. K.; Honig, B. H., *Biopolymers* **1986,** *25* (11), 2097-2119.

75. Pitera, J. W.; Falta, M.; van Gunsteren, W. F., *Biophysical Journal* **2001,** *80* (6), 2546-2555.

76. (a) Antosiewicz, J.; McCammon, J. A.; Gilson, M. K., *Journal of Molecular Biology* **1994,** *238* (3), 415-436; (b) Nielsen, J. E.; Vriend, G., *Proteins* **2001,** *43* (4), 403-412.

77. (a) Yang, T.; Wu, J. C.; Yan, C.; Wang, Y.; Luo, R.; Gonzales, M. B.; Dalby, K. N.; Ren, P., *Proteins-Structure Function and Bioinformatics* **2011,** *79* (6), 1940-1951; (b) Hou, T. J.; Wang, J. M.; Li, Y. Y.; Wang, W., *Journal of Chemical Information and Modeling* **2011,** *51* (1), 69-82.

78. Morozov, A. V.; Kortemme, T.; Baker, D., *J. Phys. Chem. B* **2003,** *107* (9), 2075-2090.

79. Schwarzl, S. M.; Huang, D. Z.; Smith, J. C.; Fischer, S., *Journal of Computational Chemistry* **2005,** *26* (13), 1359-1371.

80. (a) Tan, Y.-H.; Tan, C.; Wang, J.; Luo, R., *Journal of Physical Chemistry B* **2008,** *112* (25), 7675-7688; (b) Tan, Y.-H.; Luo, R., *Journal of Chemical Physics* **2007,** *126* (9).

81. (a) Davis, M. E.; McCammon, J. A., *Journal Of Computational Chemistry* **1989,** *10* (3), 386-391; (b) Luty, B. A.; Davis, M. E.; McCammon, J. A., *Journal of Computational Chemistry* **1992,** *13* (9), 1114-1118; (c) Holst, M.; Saied, F., *Journal of Computational Chemistry* **1993,** *14* (1), 105-113; (d) Rocchia, W.; Alexov, E.; Honig, B., *Journal of Physical Chemistry B* **2001,** *105* (28), 6507-6514; (e) Luo, R.; David, L.; Gilson, M. K., *Journal Of Computational Chemistry* **2002,** *23* (13), 1244-1253.

82. (a) Boschitsch, A. H.; Fenley, M. O., *Journal of Chemical Theory and Computation* **2011,** *7* (5), 1524-1540; (b) Mirzadeh, M.; Theillard, M.; Gibou, F., *Journal of Computational Physics* **2011,** *230* (5), 2125-2140.

83. Chen, D.; Chen, Z.; Chen, C.; Geng, W.; Wei, G.-W., *Journal of Computational Chemistry* **2011,** *32* (4), 756-770.

84. (a) Wang, J.; Cai, Q.; Li, Z. L.; Zhao, H. K.; Luo, R., *Chemical Physics Letters* **2009,** *468* (4-6), 112-118; (b) C. Wang, J. W., Q. Cai, Z. Li, H. Zhao, and R. Luo, *Computational and Theoretical Chemistry* **2013,** *1022*, in press.

85. (a) Cai, Q.; Hsieh, M. J.; Wang, J.; Luo, R., *Journal of Chemical Theory and Computation* **2010,** *6* (1), 203-211; (b) Wang, J.; Luo, R., *Journal of Computational Chemistry* **2010,** *31* (8), 1689-1698.

86. Meijerink, J. A.; van der Vorst, H. A., *Journal of Computational Physics* **1981,** *44* (1), 134-155.

87. (a) Holst, M.; Baker, N.; Wang, F., *Journal Of Computational Chemistry* **2000,** *21* (15), 1319-1342; (b) Shestakov, A. I.; Milovich, J. L.; Noy, A., *Journal of Colloid and Interface Science* **2002,** *247* (1), 62-79; (c) Bond, S. D.; Chaudhry, J. H.; Cyr, E. C.; Olson, L. N., *Journal of Computational Chemistry* **2010,** *31* (8), 1625-1635; (d) Xie, D.; Zhou, S., *Bit Numerical Mathematics* **2007,** *47* (4), 853-871.

88. (a) Zauhar, R. J.; Morgan, R. S., *Journal of Computational Chemistry* **1988,** *9* (2), 171-187; (b) Yoon, B. J.; Lenhoff, A. M., *Journal of Computational Chemistry* **1990,** *11* (9), 1080-1086; (c) Juffer, A. H.; Botta, E. F. F.; Vankeulen, B. A. M.; Vanderploeg, A.; Berendsen, H. J. C., *Journal of Computational Physics* **1991,** *97* (1), 144-171; (d) Vorobjev, Y. N.; Scheraga, H. A., *Journal Of Computational Chemistry* **1997,** *18* (4), 569-583; (e) Boschitsch, A. H.; Fenley, M. O.; Zhou, H. X., *Journal of Physical Chemistry B* **2002,** *106* (10), 2741-2754; (f) Lu, B. Z.; Cheng, X. L.; Huang, J. F.; McCammon, J. A., *Proceedings of the National Academy of Sciences of the United States of America* **2006,** *103* (51), 19314-19319; (g) Bardhan, J. P., *J. Chem. Phys.* **2009,** *130* (9).

89. Sanner, M. F.; Olson, A. J.; Spehner, J. C., *Biopolymers* **1996,** *38* (3), 305-320.

90. Chen, M.; Lu, B., *Journal of Chemical Theory and Computation* **2011,** *7* (1), 203-212.

91. Boschitsch, A. H.; Fenley, M. O., *Journal of Computational Chemistry* **2004,** *25* (7), 935-955.

92. (a) Zhang, B.; Lu, B.; Cheng, X.; Huang, J.; Pitsianis, N. P.; Sun, X.; McCammon, J. A., *Communications in Computational Physics* **2013,** *13*, 90-106; (b) Geng, W.; Krasny, R., *Journal of Computational Physics* **2013**.

93. (a) Li, L.; Li, C.; Sarkar, S.; Zhang, J.; Witham, S.; Zhang, Z.; Wang, L.; Smith, N.; Petukh, M.; Alexov, E., *BMC biophysics* **2012,** *5* (1), 9; (b) Li, C.; Li, L.; Zhang, J.; Alexov, E., *Journal of Computational Chemistry* **2012,** *33* (24), 1960-1966.

94. Fenley, M. O.; Mascagni, M.; McClain, J.; Silalahi, A. R. J.; Simonov, N. A., *Journal of Chemical Theory and Computation* **2010,** *6* (1), 300-314.

95. Strang, G., Iterative method for Ax=b. In *Linear algebra and its application*, 3 ed.; Brooks/Cole: 1988; pp 380-387.

96. Evans, D. J., *Parallel Comput* **1984,** *1* (1), 3-18.

97. Oleary, D. P.; White, R. E., *Siam Journal on Algebraic and Discrete Methods* **1985,** *6* (4), 630-640.

98. (a) Chen, W.; Poirier, B., *Journal of Computational Physics* **2006,** *219* (1), 185-197; (b) Chen, W.; Poirier, B., *Journal of Computational Physics* **2006,** *219* (1), 198-209; (c) Bank, R. E.; Holst, M., *Siam Journal on Scientific Computing* **2000,** *22* (4), 1411-1443.

99. (a) Ma, S., *Ieice Transactions on Fundamentals of Electronics Communications and Computer Sciences* **2008,** *E91A* (9), 2578-2587; (b) Iwashita, T.; Shimasaki, M., *Ieee Transactions on Magnetics* **2002,** *38* (2), 429-432.

100. Hsieh, M.-J.; Luo, R., *Journal of Molecular Modeling* **2011,** *17* (8), 1985-1996.

101. (a) Baker, N.; Holst, M.; Wang, F., *Journal Of Computational Chemistry* **2000,** *21* (15), 1343-1352; (b) Cai, Q.; Hsieh, M.-J.; Wang, J.; Luo, R., *Journal of Chemical Theory and Computation* **2010,** *6* (1), 203-211.

102. Klapper, I.; Hagstrom, R.; Fine, R.; Sharp, K.; Honig, B., *Proteins* **1986,** *1* (1), 47-59.

103. (a) Kelner, J. A.; Orecchia, L.; Sidford, A.; Zhu, Z. A. In *A simple, combinatorial algorithm for solving sdd systems in nearly-linear time*, Proceedings of the 45th annual ACM symposium on Symposium on theory of computing, ACM: 2013; pp 911-920; (b) Spielman, D. A.; Teng, S.-H., *arXiv preprint cs/0607105* **2006**; (c) Koutis, I.; Miller, G. L.; Peng, R. In *Approaching optimality for solving SDD linear systems*, Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on, IEEE: 2010; pp 235-244; (d) Koutis, I.; Miller, G. L.; Peng, R. In *A nearly-m log n time solver for sdd linear systems*, Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on, IEEE: 2011; pp 590-598.

104. (a) Tan, C. H.; Yang, L. J.; Luo, R., *Journal Of Physical Chemistry B* **2006,** *110* (37), 18680-18687; (b) Salari, R.; Chong, L. T., *Journal of Physical Chemistry Letters* **2010,** *1* (19), 2844-2848.

105. Lin, Y.; Baumketner, A.; Deng, S.; Xu, Z.; Jacobs, D.; Cai, W., *J. Chem. Phys.* **2009,** *131* (15).

106. (a) Silalahi, A. R. J.; Boschitsch, A. H.; Harris, R. C.; Fenley, M. O., *Journal of Chemical Theory and Computation* **2010,** *6* (12), 3631-3639; (b) Zhou, S.; Wang, Z.; Li, B., *Phys. Rev. E* **2011,** *84* (2).

107. Mobley, D. L.; Barber, A. E., II; Fennell, C. J.; Dill, K. A., *Journal of Physical Chemistry B* **2008,** *112* (8), 2405-2414.

108. Chong, S.-H.; Ham, S., *Proceedings of the National Academy of Sciences* **2012,** *109* (20), 7636-7641.

109. (a) Genheden, S.; Luchko, T.; Gusarov, S.; Kovalenko, A.; Ryde, U., *The Journal of Physical Chemistry B* **2010,** *114* (25), 8505-8516; (b) Luchko, T.; Gusarov, S.; Roe, D. R.; Simmerling, C.; Case, D. A.; Tuszynski, J.; Kovalenko, A., *Journal of Chemical Theory and Computation* **2010,** *6* (3), 607-624.

110. Srinivasan, J.; Cheatham, T. E.; Cieplak, P.; Kollman, P. A.; Case, D. A., *Journal of the American Chemical Society* **1998,** *120* (37), 9401-9409.

111. Gohlke, H.; Case, D. A., *Journal of Computational Chemistry* **2004,** *25* (2), 238-250.

112. Kongsted, J.; Soderhjelm, P.; Ryde, U., *Journal of Computer-Aided Molecular Design* **2009,** *23* (7), 395-409.

113. Kongsted, J.; Ryde, U., *Journal of Computer-Aided Molecular Design* **2009,** *23* (2), 63-71.

114. Wittayanarakul, K.; Hannongbua, S.; Feig, M., *Journal of Computational Chemistry* **2008,** *29* (5), 673-685.

115. (a) Cai, Q.; Ye, X.; Luo, R., *Physical Chemistry Chemical Physics* **2012,** *14* (45), 15917-15925; (b) Cai, Q.; Ye, X.; Wang, J.; Luo, R., *Chemical Physics Letters* **2011,** *514* (4-6), 368-373; (c) Xiao, L.; Cai, Q.; Ye, X.; Wang, J.; Luo, R., *The Journal of Chemical Physics* **2013,** *139* (9), 094106.

Chapter 1: Numerical Poisson-Boltzmann Model for Continuum Membrane Systems

Introduction

Energetic analyses of solvated systems are of fundamental importance in theoretical and computational studies of molecular biophysics. Due to the sizes of many biologically relevant compounds and systems, and the need for extensive sampling required to recover observable properties, explicit inclusion of solvent molecules can become quite demanding. Implicit/continuum solvation methods allow energetic calculations to be computed with far less computational expense by approximating discrete solvent molecules with a continuum. These types of methods have been routinely applied in many biomolecular applications such as protein-ligand binding affinity and docking pose predictions.[1]

Recent interest in membrane proteins has spurred extension of continuum solvation treatments to incorporate membrane models.[2] To implement a continuum membrane, an additional solvent region must be incorporated into the solvation model. Since most membranes are typically non-polar at the interior, energy terms that would be accounted for in an aqueous system by using solute volume or surface area can be adapted to include a membrane by taking the union of the solute and membrane regions. This manuscript focuses on the treatment of the electrostatic energy contributions. Electrostatic energy contributions can be accounted for by treating the membrane as a region with a low dielectric constant, often close to that of the solute. While this approach is relatively straightforward on the surface, the solvent region now includes a heterogeneity that may have significant bearing on handling boundary conditions and interface conditions.

The Poisson-Boltzmann equation (PBE) is the basis of electrostatic energy calculations for many continuum solvation methods.[3] However, closed-form analytical solution is only possible for a few special cases, such as systems with radial symmetry. Indeed the popular Generalized-Born (GB) methods are derived from PBE by making specific simplifying assumptions to allow a closed-form approximated solution to be obtained.[4] Although GB is very fast compared to most full numerical solution techniques, its inherent assumptions are often a source of debate and the methods have been shown to result in undesirable errors in some cases.[5] Furthermore, while application to heterogeneous cases (such as membrane protein systems) is possible, it poses significant challenges.[2b, 2c, 6] In most cases the only recourse is to seek a numerical solution. Many methods have been proposed and investigated for this purpose. Finite-difference (FD) [7] methods are amongst the most popular. Despite their somewhat lower adaptivity compared to finite-element [8] or boundary-element methods,[3e, 3h, 9] they are often preferred for large-scale computations due to their speed, efficiency, and ease of implementation.[10]

Current numerical methods often assume free or zero potential boundary conditions. Systems with more complex setups, e.g. when a membrane region is included, may suffer from artifacts due to edge effects when these boundary conditions are used. The problem is particularly profound in the FD methods. For such systems, periodic boundaries become an attractive alternative. Implementation of the periodic boundary condition requires modifying the algebraic equations generated for grid nodes at the FD grid edges to include terms imposing periodicity. An additional benefit of imposing periodicity, however, is that it eliminates the need for adding boundary charge distributions required for the free boundary conditions. Additionally, Fourier transform based solvers become an attractive option due to their innately periodic formulation and log-linear scaling.[11] Apparently, limitation of these algorithms to use for constant coefficient equations prevents them from solving the general PBE with inhomogeneous coefficients. The augmented Immersed Interface Method (IIM) allows this limitation to be overcome for systems where each region has a distinct uniform dielectric constant.[12]

In the following section, we first present a continuum membrane model based on the level set formulism and a pseudo density function approach [13]. This is followed with the description of two periodic FD solvers, the finite volume/periodic conjugate gradient (FV/PCG) and augmented Immersed Interface Method/Fast Fourier Transform solver (IIM/FFT) to solve the PBE for the continuum membrane systems. Finally a detailed validation of the new continuum model and analysis of the numerical algorithms are presented.

Method

Membrane Setup

Dielectric Model Setup and Interface Location via Level Set Continuum solvation can be extended to include a membrane by modeling it as an additional dielectric region. We first focused on the case where the membrane dielectric matches the solute's dielectric, so that the membrane region is simply an extension of the solute region. This requires us to derive, define, and merge a membrane level set into the molecular level set originally used to generate a dielectric mapping for globular proteins.[13] Further modification to allow unique membrane dielectric constant can make use of this infrastructure, but it will be left to a future study. The resulting linear systems were then solved assuming the periodic boundary condition.

Solute-Solvent Level Set Construction We begin by reviewing the level set formulation for globular proteins. The level set is defined such that it will be positive on one side of the interface (solvent region in this case), negative on the other side (solute or membrane region) and zero on the interface. The interface is thus the locus of all points where the level set is zero.

Earlier attempts used the "signed" distance function as the level set, which was defined as the distance to the unsmoothed molecular surface,[13] However, the straightforward definition prohibits the use of higher accuracy methods, such as IIM,[12, 14] which requires smooth and continuous level set functions. In this study a smooth and continuous density function [13] was used instead. In this approach, the solute-solvent level set ϕ at a point $[\mathbf{p}]$ is derived from the sum of atomic density contributions defined by the equations:

$$\phi[\mathbf{p}] = 1 - \sum_{i=1}^{natom} \rho_i[\mathbf{p}] \tag{1.1}$$

$$\rho_i = \begin{cases} k \cdot d_i[\mathbf{p}] & d_i \le 0 \\ f_{spline}(d_i[\mathbf{p}]) & d_i > 0 \end{cases} \tag{1.2}$$

$$d_i[\mathbf{p}] = r_i[\mathbf{p}] - R_i \tag{1.3}$$

where the *natm* is the number of atoms in our system and $\rho_i[\mathbf{p}]$ is the density function for atom *i* at $[\mathbf{p}]$. The density $\rho_i[\mathbf{p}]$ is computed based upon the signed distance between $[\mathbf{p}]$ and the solvent accessible surface of the atom. Function f_{spline} is a splined density function as defined and optimized in [13], and k is the constant needed to ensure the function smoothness across the interface. Lastly, d_i is the signed distance to the interfacial surface for atom i, where r_i is the distance from atom i to $[\mathbf{p}]$ and R_i is the atomic cavity radius.

Membrane Level Set To implement a continuum membrane model, a corresponding level set function is needed to merge with the solute–solvent level set. The simplest membrane resembles a slab-like region with two planar interfaces parallel to the x-y plane. Thus the total level set function (considering both the solute and membrane) can be expressed as:

$$\phi[\mathbf{p}] = 1 - \rho_m[\mathbf{p}] - \sum_{i=1}^{natom} \rho_i[\mathbf{p}] \tag{1.4}$$

$$\rho_m[\mathbf{p}] = \begin{cases} g_{memb}(d_m[\mathbf{p}]) & d_i \le 0 \\ f_{spline}(d_m[\mathbf{p}]) & d_i > 0 \end{cases} \tag{1.5}$$

$$\begin{aligned} g_{memb} &= \begin{cases} a \cdot d_m^3 + b \cdot d_m^2 + c \cdot d_m + e & 0 > d_m > R_p \\ k_p & d_m < R_p \end{cases} \\ k_p &= a \cdot R_p^3 + b \cdot R_p^2 + c \cdot R_p + e \\ a &= 2.108572 \\ b &= -6.108572 \\ c &= 4.527143 \\ e &= 0 \end{aligned} \tag{1.6}$$

$$d_m = z_{mctr} - p_z - m/2 \tag{1.7}$$

Here $\rho_m[\mathbf{p}]$ is the membrane level set density contribution at $[\mathbf{p}]$ and f_{spline} is the same as the spline function for atomic contributions. The function g_{memb} is a monotonic, concave up, cubic polynomial function constructed to transition the level set to a constant value near the membrane center while preserving smoothness and continuity. Coefficients a, b, and c were parameterized to ensure that 1) the first derivative of g_{memb}, with respect to d_m, matched the first derivative at $d_m = 0$ and 2) $0 > d_m > R_p$. The coefficient e must remain 0 since we want $f_{spline}[0] = g_{memb}[0] = 0$. See Figure 1.1 for an illustration of the construction of $\rho_m[\mathbf{p}]$ and the corresponding level set density sum contribution starting from $d_m[\mathbf{p}]$. Finally, $d_m[\mathbf{p}]$ is the distance from [**p**] to the nearest membrane surface, where z_{mctr} is the z coordinate of the membrane center, $\mathbf{p}_z$ is the z coordinate of [**p**] and m is the membrane thickness.

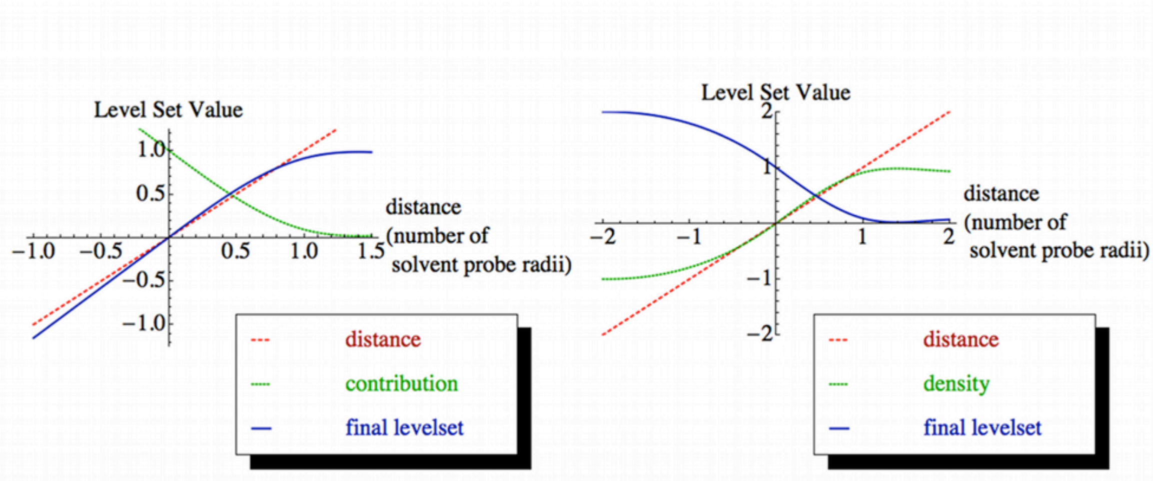

Figure 1.1: Construction of Solute and Membrane Level Set Densities

Illustrations of level set construction starting from signed distance functions. Signed distance to van der Waals surface is shown in red. Summation contribution computed using spline and/or polynomial scaling functions is shown in green. Final level set function is shown in blue. Left: Diagram for solute level set function construction. Right: Diagram for membrane level set function construction

Inclusion of Channels The simple slab-like membrane setup may cause problems if a solute contains pore- or channel-like region(s) that need to retain a solvent dielectric constant. To describe these proteins more accurately, an appropriate cylindrical region will be removed from the membrane region [2f]. Removal of a cylindrical region could lead to sharp changes in the level set if not properly transitioned. Use of g_{memb} in (1.5), rather than a linear function as in (1.2), ensures that there will not be a cusp at the membrane center. What remains is to transition the level set contribution smoothly at the excluded region's surface. The membrane density is modified when a cylindrical region is needed, as follows:

$$\rho_m[\mathbf{p}] = \begin{cases} f_{spline}\left(\sqrt{d_m^2[\mathbf{p}] + \mathbf{d}_p^2[\mathbf{p}]}\right) & d_m > 0, d_p < 0 \\ f_{spline}\left(d_m[\mathbf{p}]\right) & d_m > 0, d_p > 0 \\ f_{spline}(d_p[\mathbf{p}]) & d_m < 0, d_p < 0 \\ g_{memb}\left(-\sqrt{d_m[\mathbf{p}] \cdot Min\left[d_p[\mathbf{p}], d_m[\mathbf{p}]\right]}\right) & d_m < 0, d_p > 0 \end{cases} \tag{1.8}$$

$$\begin{aligned} d_p[\mathbf{p}] &= r_p[\mathbf{p}] - R_p \\ r_p[\mathbf{p}] &= \sqrt{(\mathbf{p}_x - \mathbf{c}_x)^2 + (\mathbf{p}_y - \mathbf{c}_y)^2} \end{aligned} \tag{1.9}$$

Here d_p is the signed distance to the surface of the cylindrical region, where [**c**] is the coordinates of the center of the cylindrical region. Again, f_{spline} and g_{memb} scale the signed distances to match a density summation approach. Distances $d_m[\mathbf{p}]$ and $d_p[\mathbf{p}]$ are defined from point [**p**] to the membrane and cylindrical region interfacial surfaces respectively. Finally $r_p[\mathbf{p}]$ is the distance to the centeral axis of the cylindrical region running along x = $\mathbf{c}_x$ and y =

$\mathbf{c}_y$ membrane interior level set and the cylindrical regions level set for points residing near the interfacial surface interior to the membrane. See

Figure 1.2 for an illustration of the construction of Eq. (1.9) from signed distances $d_m[\mathbf{p}]$ and $d_p[\mathbf{p}]$ for a model channel membrane setup on a 200×200 cross-sectional grid.

Once the membrane level set is computed and added to the solute-solvent level set, the existing dielectric map setup procedure can be used to map the needed dielectric distribution on the FD grid.[3g, 3i] The upper expressions in equation (1.8) represent the region that is assigned solvent dielectric constant. This includes the region above or below the membrane or within the cylindrical region (pore) of the membrane. The top most equation is used for the region just above the cylindrical exclusion region and is used to transition the level set smoothly as the distance to the upper or lower edge of the intersection of the cylinder and membrane regions. Finally, the last equation uses a geometric average to provide a smooth transition.

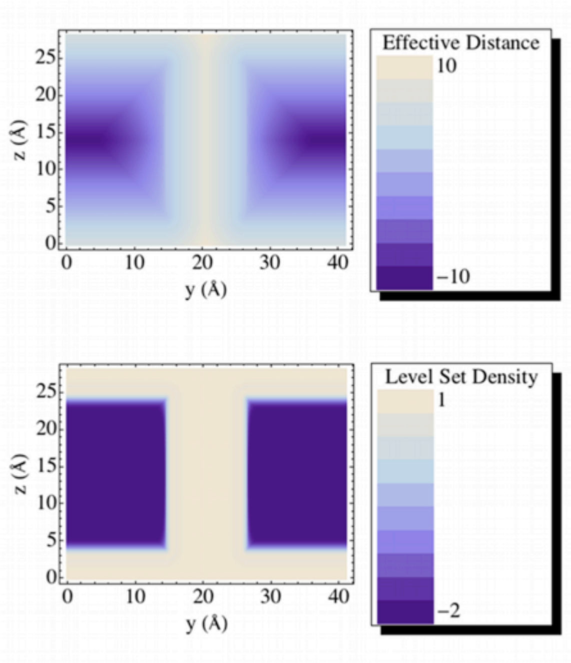

Figure 1.2: Effective signed distance (Å) and level set density distribution cross sections for model membrane

Signed distance and level set density distributions on the cross section of a 20 Å model membrane containing a cylindrical exclusion region with a radius of 6 Å. Top: Signed distance distribution. Bottom: Level set density distribution.

Adaptation of the Numerical FD Solvers for Periodic Boundary Conditions

Now that we have an appropriate model for the discretization for the dielectric map of our membrane, solvent, and solute regions, we must implement an appropriate method to solve the resulting systems of equations. However, unlike the globular protein model wherein non-uniformity in the dielectric constant was confined to the interior of our system, the membrane model extends the solute dielectric constant to the edges of the grid. To overcome potential computational artifacts, due to edge effects, a periodic boundary formulation is used when a membrane model is employed. This requires adapting our solver methods from the isolated free-boundary formulation to a periodic boundary formulation. In this study, we first consider the Finite Volume / Periodic Conjugate Gradient and the Immersed Interface Method / FFT approaches.

Periodic Conjugate Gradient Solver The existing conjugate gradient solvers, either unconditioned or conditioned, can be used to solve the algebraic equations from the finite volume discretization. This approach, reviewed in the supplemental material section, was initially developed for isolated systems when the free boundary condition is used. Briefly, the effect of the dielectric is modeled using a dielectric map, which assigns a dielectric value to each edge connecting a pair of grid nodes in a regular rectangular lattice. The dielectric map is then used to construct a series of algebraic equations based on an appropriate stencil according to the finite volume representation of the linearized PBE. To utilize this method under free boundary conditions, pseudo charges would need to be computed and mapped to the boundary of the grid. This is unnecessary for a periodic system. Implementation involves modifying the linear system by coupling grid nodes on one edge with "adjacent" nodes on the opposite edge (see equation 9 of supplementary materials). For the unconditioned conjugate gradient method, this can be accomplished with a single pass over each edge node performed prior to the main pass of each iteration step.

FFT Solver The augmented immersed interface method,[12, 14-15] briefly reviewed in the supplementary materials, provides an alternative to the finite volume discretization. Under this approach, the effect of the non-uniform dielectric constant of the system is modeled by introducing an effective surface charge distribution along the interface(s) between regions with differing dielectric constants. This results in introduction of potential and field jump conditions. The field jump condition is used as an augmented variable which is converged iteratively. This requires solving PBE for a system with point charges, a surface charge distribution and uniform dielectric constant, at each step. This proves to be the most time consuming step of each iteration. Periodicity may be implemented by implementing periodic boundary conditions for the Poisson's equation solver. However, since the system now has a uniform dielectric constant, we may employ a rapid ellipitical FFT based solver to accelerate this step.[11] Due to the nature of the FFT approach, the resulting solutions will naturally be periodic without any further modification. The details of the development and implementation of our FFT based solver are reviewed in the supplementary materials.

Computational Details

For the FV/PCG method, edges connecting points within a dielectric region are assigned to that region's value; edges that cross an interface are assigned a value using a weighted harmonic average [16]. For the IIM/FFT method, dielectric is modeled using surface charges and corresponding surface jump conditions.[12] For both approaches, a level set function provides the means of locating the interface(s). To provide a consistent testing framework, the PBSA module in the AMBER 12 simulation and modeling package [17] was used to implement our methods. In each case, single point electrostatic energy calculations were computed. The atomic cavity radii were set to be the default *mbondi* set in the Amber package, except all hydrogen radii were set to be 1.0Å. A classical two-dielectric model was used to set the dielectric distribution where region within the solute/membrane is set to 1 and region outside is set to 80. Default options were used for all parameters except those specifically noted here or in the corresponding discussion in the results and discussion section. All models except for the quadrapole system and Aquaporin C-terminal coil system were parameterized directly from their corresponding models from pdb entries. The quadrapole system was modeled as a single sphere of 2.0 Ångstroms and four point charges of zero net charge and zero net dipole moment. The Aquaporin coil system was constructed by excising the last 19 C-terminal residues of the Aquaporin model (1IH5).

Biomolecular systems were implemented in a realistic membrane with a thickness of 20 Ångstroms. In all cases except the Aquaporin systems, a simple slab-like membrane was sufficient. For the Aquaporin system, both a simple slab-like model and cylindrical exclusion pore model were tested. The pore radius was set to 6.0 Ångstroms to ensure that no solvent in the channel region would be overwritten.

Results and Discussion

Consistency between Periodic FD Solvers

The FFT solver may be used to calculate potentials for *in vacuo* systems without the augmented IIM method. This allows us to compare the FFT and PCG solvers for *in vacuo* systems directly. To ensure that both methods give consistent results, electrostatic potential distributions were generated for the complete Aquaporin system in vacuum. Contour plots were generated using the *Mathematica* software package and are shown in **Figure 1.3**. As is evident from the contour plots, the two methods yield equivalent electrostatic potential distribution in vacuum, even for the tested large complex molecule. Electrostatic energies reported for AMBER were also identical for both solvers (Table 1). More detailed analysis shows that discrepancy in computed energies is below the corresponding tolerance set for PCG. Given their high numerical consistency, we next proceed to validate our numerical models in more complex dielectric setups.

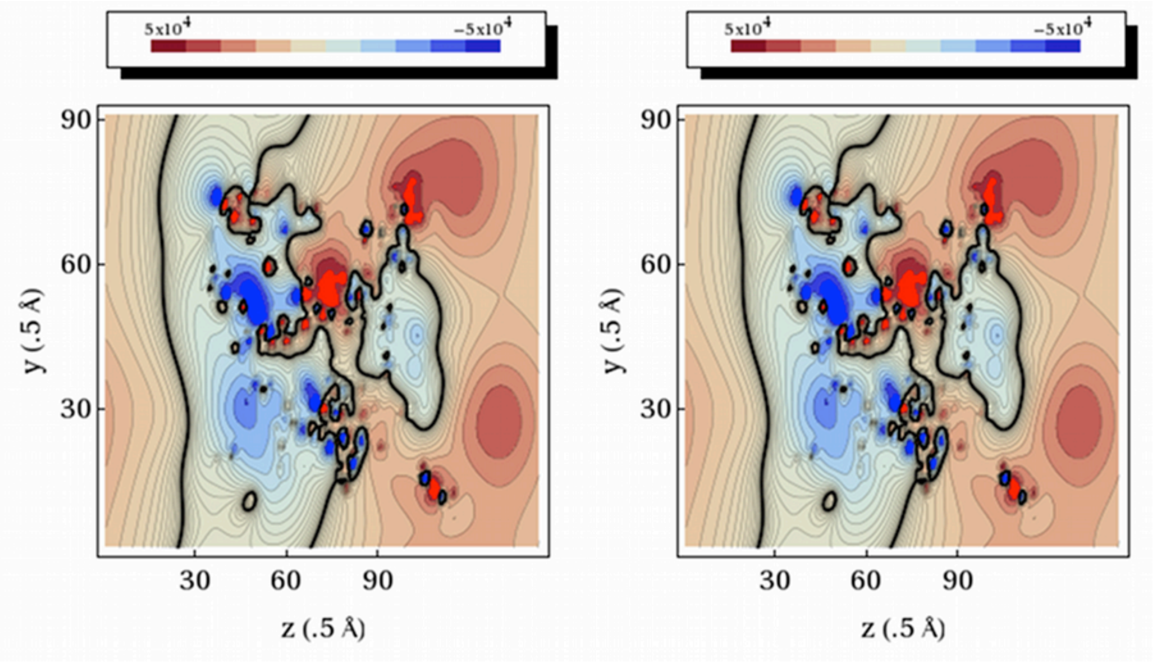

Figure 1.3: Electrostatic potential distribution (kcal/mol-e$^-$) of Aquaporin in vacuum.

Left: Results for the FFT solver. Right: Results for the PCG solver. Contours are taken along the *yz* plane through the center of the finite-difference grid.

Quadrapole in a Membrane

We first used a simple quadrapole system in a 2 Ångstrom low dielectric sphere as an initial test case for the membrane setup. The membrane region was represented as a rectangular slab oriented with its normal running parallel to the *z* axis and centered on the middle of the simulation box with dielectric constant equal to that of the solute (e.g. acting as an extension of the solute region). Cross-sectional electrostatic potential distributions for the water only and water + membrane systems were generated using *VMD* and displayed with the solute/solvent boundary grid points overlaid as shown in Figure 1.4. Comparison of the left and right panels clearly shows that the low dielectric solute region is indeed being extended to include the slab-like membrane. It is also evident that the level set density method leads to smoothing out of what would otherwise be a sharp transition between the spherical solute and the rectangular membrane.

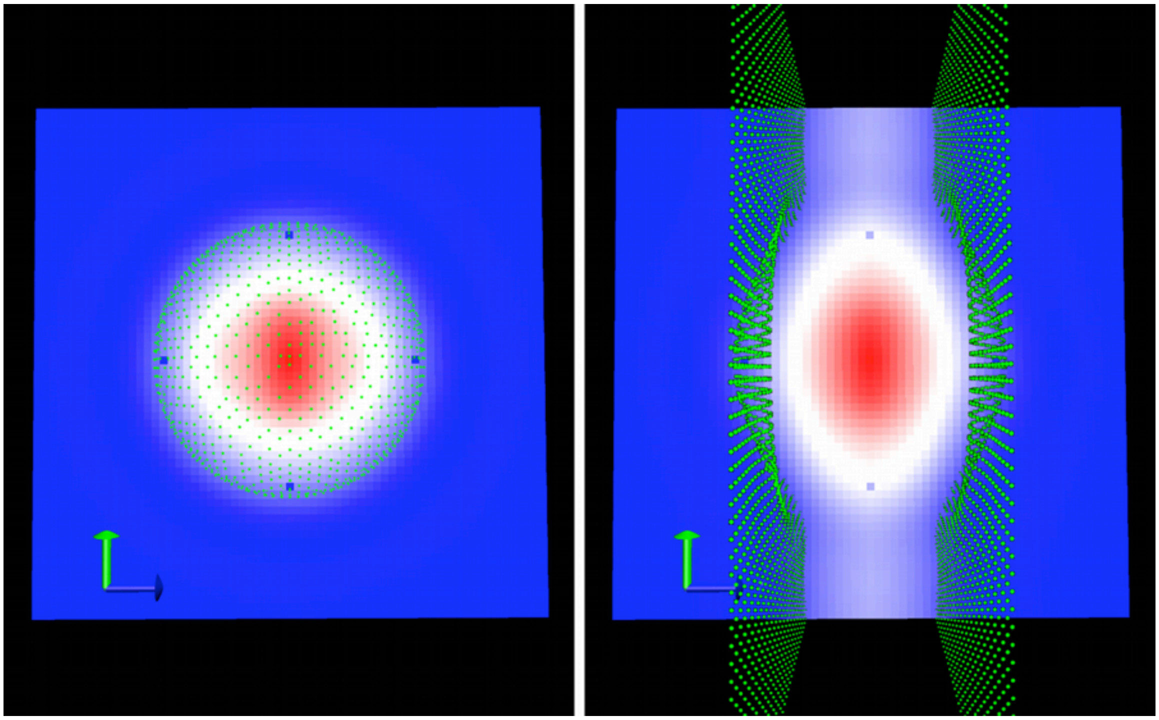

Figure 1.4: Level Set Cross Sections and Boundary Grid Points for Quadrapole System

Left: System solvated in water. Right: System solvated in water and 2 Å slab-like membrane. Boundary grid points are overlaid as green points. For both plots, red indicates the solute/membrane interior region, and blue indicates the solvent region.

Small to Mid-Sized Membrane Peptides and Proteins

To test the numerical setups for moderate-sized membrane protein systems, we first ran computations on the C-terminal trans-membrane alpha helix of Aquaporin. Computations were run for vacuum, continuum water, and continuum water + 20 Ångstrom membrane. The results are visualized in Figure 1.5 where molecule itself is visualized using the van der Waals surface with the electrostatic potential distribution mapped onto it. The boundary grid points are also shown to indicate the solute solvent interface. The effect of adding the high dielectric solvent is clearly evident when comparing the left (vacuum) and middle (water only) panels, which shows reduction in intensity and contrast of the color mapping. The effect of the membrane region is evident from comparing the right (water + membrane) panels with the middle and left panels. The upper and lower portions of the molecule that extend beyond the membrane exhibit potentials that most closely resemble the water-solvated potentials while the region interior to the membrane exhibits potential more closely resembling that of the vacuum potentials.

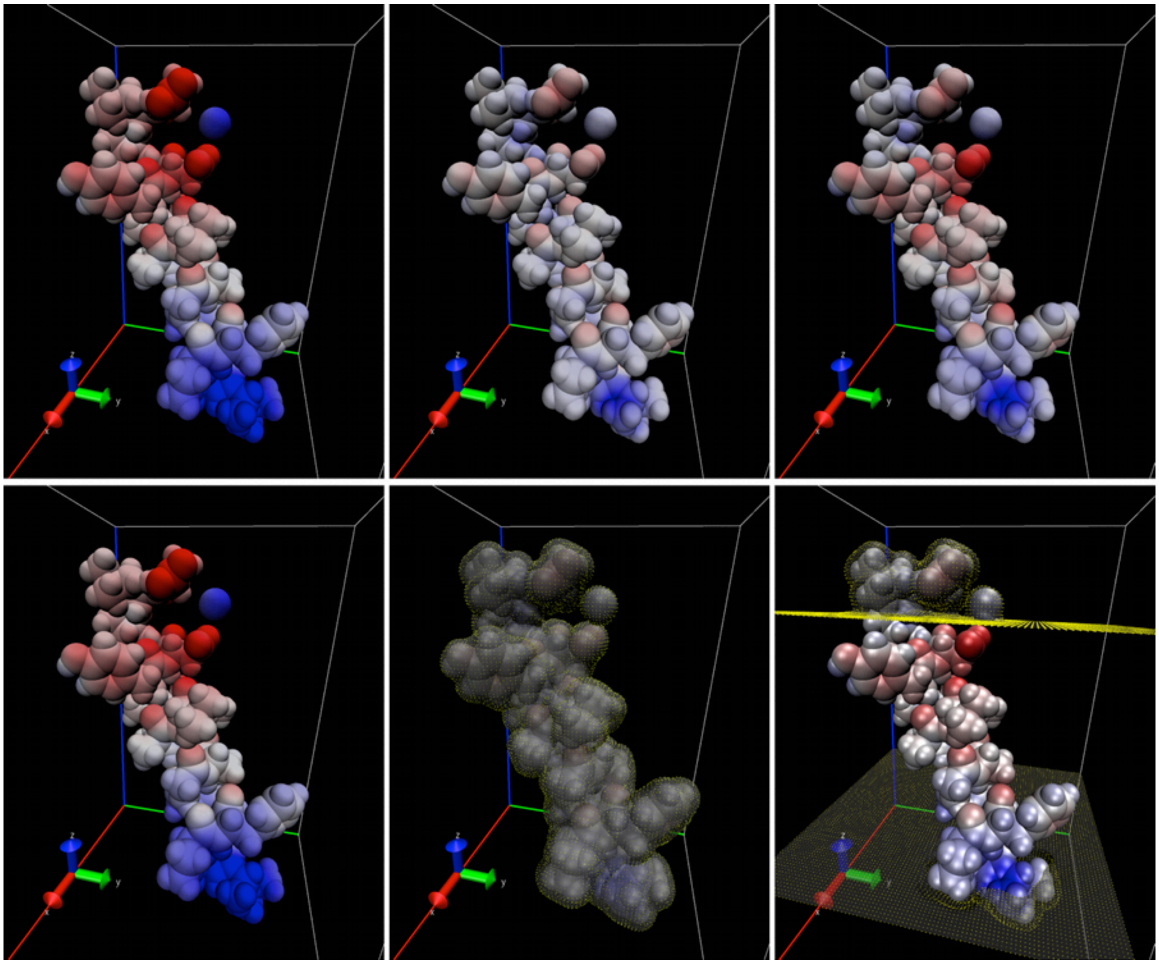

Figure 1.5: Electrostatic potential distribution (kcal/mol-e$^-$) and boundary grid points for the Aquaporin coil system

Top: Electrostatic potential maps only. Bottom: Electrostatic potential maps overlaid with boundary grid points in yellow. Left: Vacuum System. Middle: Solvated in water + 20 Å slab-like membrane. Right: Solvated in water only.

Next we tested our models directly on membrane protein systems and ran computations on several proteins ranging from 19 to 200 amino acids. Electrostatic energies were computed using both the FV/PCG and IIM/FFT methods. Results are shown in FV/PCG and IIM/FFT methods **Table 1.1** with columns 3 and 4 showing the reaction field energy for each system.

Columns 5 and 6 show the change in the reaction field energy due to the addition of the membrane region. We again notice that the membrane solvation yields energy values between the water only and vacuum cases, as would be expected from extension of the low dielectric solute region (for which the dielectric constant matches the vacuum dielectric). Furthermore, both numerical methods yield energies highly consistent with each other, with difference less than 1-2%, for most cases, demonstrating the consistency between the very different handlings of the heterogeneous dielectrics by FV/PCG and IIM/FFT methods.

Table 1.1: Electrostatic Potentials for Continuum Water and Water + Membrane Solvation for Various Systems

System	Vacuum	FFT Total	PCG Total	FFT Rxn	PCG Rxn	FFT Δ	PCG Δ
Quadrapole Water	-0.713	-12.914	-12.930	-12.201	-12.217	NA	NA
Quadrapole Water + Memb.	-0.713	-4.175	-4.133	-3.461	-3.420	8.739	8.797
Aquaporin C-term. Coil Water	-1385.22	-1670.02	-1668.51	-284.797	-283.295	NA	NA
Aquaporin C-term. Coil Water + Memb.	-1385.22	-1587.68	-1587.03	-202.456	-201.814	82.341	81.480
Inf HAFD Water	-1723.99	-2130.11	-2135.61	-406.114	-411.618	NA	NA
Inf HAFD Water + Memb.	-1723.99	-1878.14	-1890.20	-154.143	-166.207	251.971	245.410
Rr LHC Water	-3358.85	-4084.69	-4094.11	-725.843	-735.258	NA	NA
Rr LHC Water + Memb.	-3358.85	-4042.27	-4062.88	-683.419	-704.028	42.424	31.231
SFV MBP Water	-1235.18	-1719.84	-1722.70	-484.655	-487.521	NA	NA
SFV MBP Water + Memb.	-1235.18	-1575.80	-1585.48	-340.616	-350.299	144.039	137.223
OMPX Water	-10584.29	-12219.19	-12270.85	-1634.91	-1686.57	NA	NA
OMPX Water + Memb	-10584.29	-11811.29	-11858.25	-1227.01	-1273.96	407.900	412.606

Results for electrostatic energy calculations with IIM/FFT and FV/PCG for various systems, in kcal/mol. Vacuum energies are shown as reference in the first column. Rxn refers to reaction field energies. The “FFT △” and “PCG △” columns show the difference between the reaction field energies solvated in water and in water + membrane. All membranes except for the Influenza Hemmagglutination Fusion Domain (Inf HAFD) and the Semiliki Forest Virus Membrane Binding Peptide (SFV MBP) were centered on the protein/peptide. For Inf HAFD and SFV MBP, the membrane was offset by 5 Å in the negative *z* direction to simulate partial insertion from the top of the membrane rather than transmembrane behavior as with the other systems.

Aquaporin Channel in Membrane

The new model is next tested on a typical membrane protein system, Aquaporin, whose transmembrane channel also offers an opportunity to test the implementation of the cylindrical exclusion feature. The tested Aquaporin was oriented such that its central solvent channel ran roughly parallel to the z axis. A grid spacing of 0.5 Ångstrom and a fill ratio of 1.125 were used due to its large size. Default AMBER settings were used in all other cases. Four continuum solvation setups were tested: in vacuum, in water, in water + membrane, and in water + membrane with pore. The width and placement of the pore exclusion region was chosen such that membrane dielectric would not be assigned to the solvent channel of the Aquaporin while also ensuring that the exclusion region remained within the bounds of the membrane bound protein region.

Figure 1.6 demonstrates the proper implementation of the solvated membrane system with the transmembrane channel. The top panels (membrane with the pore exclusion region) show that the pore region is set to the solvent dielectric when the membrane pore exclusion feature is turned on; whereas the pore region is set to the solute/membrane dielectric when the pore exclusion feature is turned off. This indicates that the membrane pore exclusion feature functions as expected.	Figure 1.7 further visualizes the electrostatic potential distributions of the membrane protein with or without membrane, as well as, with and without pore. The addition of solvent (top right panel) clearly reduces the magnitude of the electrostatic potential in the solvent region when compared to the *in vacuo* run (top left panel). The addition of the slab-like membrane region to the solvated protein (bottom left) results in an increase in the magnitude of the potential within the membrane region as expected. When an excluded pore region is included (bottom right), the magnitude of the potential in the pore region becomes more closely matching that in the protein solvated in water.

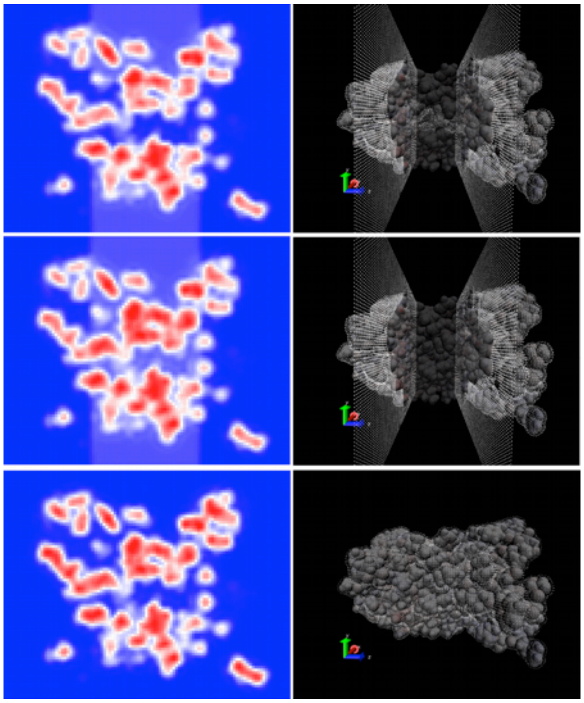

Figure 1.6 : Cross-sectional distribution of level set density function and boundary grid points for the Aquaporin system

Left: Cross-sectional distribution of the level set density function taken along the *yz* plane through the center of the channel. Red indicates the solute region, white indicates the membrane region, and blue indicates the solvent region. Right: van der Waals surface of the Aquaporin system overlaid with boundary grid points in white. The van der Waals surface is made transparent to allow viewing of buried boundary grid points. Top panels: Solvated in water + 20 Å membrane with a 6 Å cylindrical exclusion region. Middle panels: Solvated in water + 20 Å membrane. Bottom panels: Solvated in water.

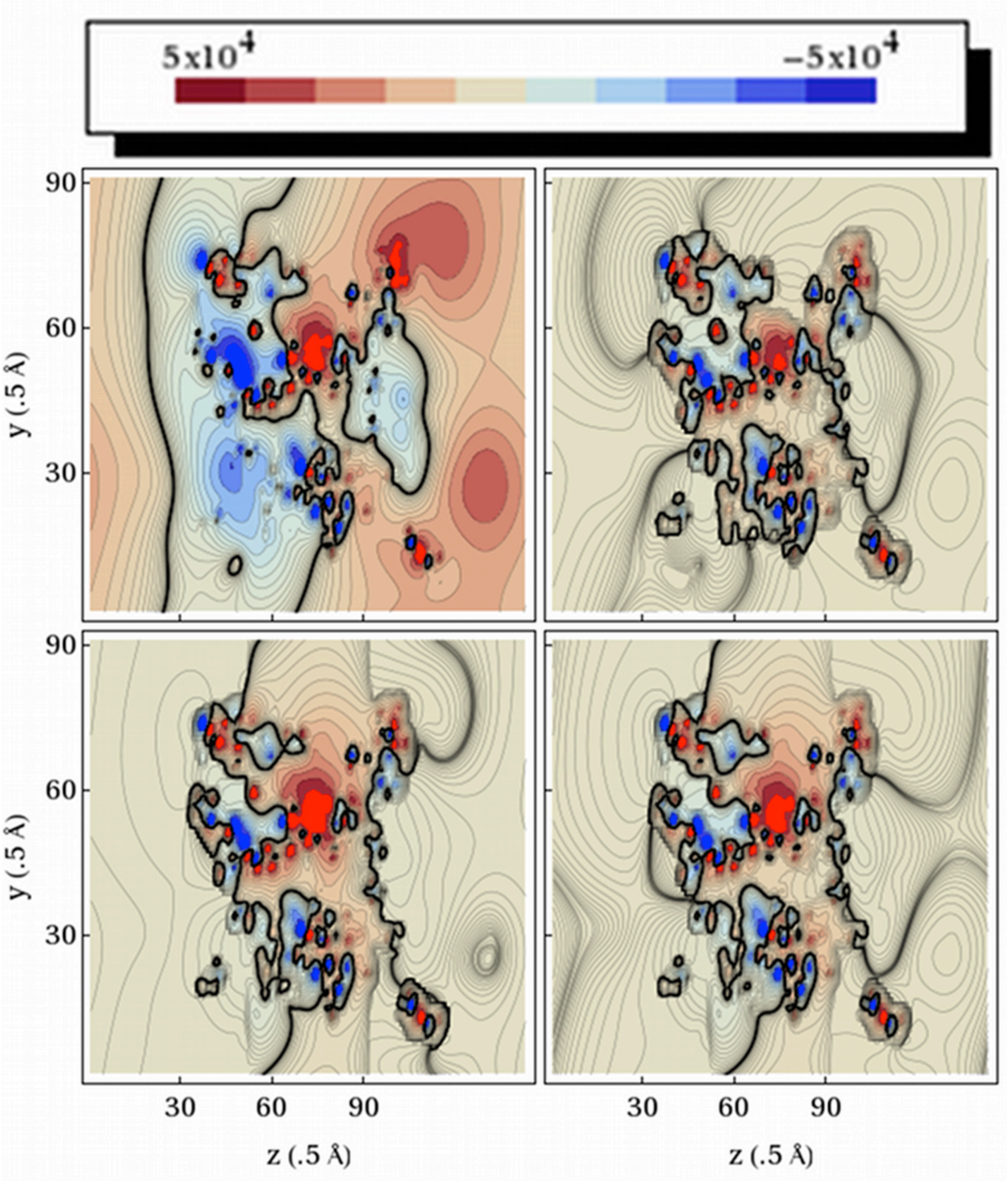

Figure 1.7 : Cross-sectional distributions of electrostatic potential (kcal/mol-e$^-$) for the Aquaporin system

Top Left: Vacuum. Top right: Solvated in water. Bottom Left: Solvated in water + 20 Å membrane. Bottom Right: Solvated in water + 20 Å membrane and a 6 Å cylindrical exclusion region. Contour plots are taken along the *yz* plane through the center of the channel.

Conclusions and Future Directions

In this study, we explored a continuum slab-like membrane model based on the density function strategy. The optional cylindrical exclusion region was also implemented with the assistance of the level set function for easily mapping of heterogeneous dielectric distributions in the continuum representation of the membrane systems, in order to accommodate the existence of transmembrane channel. To mitigate the artifacts of edge effects of the finite system sizes, the periodic boundary condition was also utilized. The continuous and smooth density level set function also allows higher-order PBE solvers to be utilized in the current setup.

Visualization of the tested systems for the water only and water + membrane setups indicate that the membrane level set scheme functions properly. Comparison of reaction field energies between the water only and the water + membrane setups for various small to mid-sized peptides and proteins indicate that the addition of a membrane region lowers the magnitude of the reaction field energies. This is expected since the membrane is an extension of the low dielectric region and thus should produce results that fall somewhere between the water only and vacuum environments. Finally, the cylindrical exclusion feature, as illustrated in the visualizations for the Aquaporin system, was shown to have the desired effect of preventing the solute dielectric from being mapped to the channel region of the protein, which should retain the high solvent dielectric. The implementation was also confirmed by comparison of the electrostatic potential in the channel region between the simple slab-like membrane setups with and without the cylindrical exclusion. The magnitude of the electrostatic potential in the channel region more closely matches that from the water only setup.

The results of our continuum membrane model are encouraging; however, there are still many details that must be addressed before it can be routinely applied in biomembrane system modeling and analysis. We are actively working on implementation of unique dielectric constants profiles for the membrane and reformulation of the preconditioned linear PBE solvers[18] to suit the periodic boundary condition for more efficient numerical calculations of large biomembrane systems. In addition, the extension of the continuum membrane model, specifically, inclusion of charged head group distributions and incorporation of the hydrophobic effects, is quite interesting.[19] Finally, optimization of model parameters for simulation of common membranes should lead to robust modeling of the membrane systems.

References

1. (a) Gohlke, H.; Case, D. A., *Journal of Computational Chemistry* **2004,** *25* (2), 238-250; (b) Srinivasan, J.; Cheatham, T. E.; Cieplak, P.; Kollman, P. A.; Case, D. A., *Journal of the American Chemical Society* **1998,** *120* (37), 9401-9409.
2. (a) Forsten, K. E.; Kozack, R. E.; Lauffenburger, D. A.; Subramaniam, S., *Journal of Physical Chemistry* **1994,** *98* (21), 5580-5586; (b) Spassov, V. Z., Yan, L., and Szalma, S. , *J Phys. Chem. B* **2002,** *106* (8726-38); (c) Im, W., Feigh, M., and Brooks III, C. L. , *Biophysical Journal* **2003,** *85*, 2900-18; (d) Tanizaki, S.; Feig, M., *Journal of Chemical Physics* **2005,** *122* (12); (e) Tanizaki, S.; Feig, M., *Journal of Physical Chemistry B* **2006,** *110* (1), 548-556; (f) Callenberg, K. M., Choudhary, O. P., de Forest, G. L., Gohara, D. W., Baker, N. A., and Grabe, M. , *PLoS One* **2010,** *5* (9), 1-11.
3. (a) Miertus, S.; Scrocco, E.; Tomasi, J., *Chemical Physics* **1981,** *55* (1), 117-129; (b) Hoshi, H.; Sakurai, M.; Inoue, Y.; Chujo, R., *Journal of Chemical Physics* **1987,** *87* (2), 1107-1115; (c) Zauhar, R. J.; Morgan, R. S., *Journal of Computational Chemistry* **1988,** *9* (2), 171-187; (d) Davis, M. E.; McCammon, J. A., *Chemical Reviews* **1990,** *90* (3), 509-521; (e) Rashin, A. A., *Journal of Physical Chemistry* **1990,** *94* (5), 1725-1733; (f) Juffer, A. H.; Botta, E. F. F.; Vankeulen, B. A. M.; Vanderploeg, A.; Berendsen, H. J. C., *J. Comput. Phys.* **1991,** *97* (1), 144-171; (g) Honig, B.; Nicholls, A., *Science* **1995,** *268* (5214), 1144-1149; (h) Liang, J.; Subramaniam, S., *Biophysical Journal* **1997,** *73* (4), 1830-1841; (i) Roux, B.; Simonson, T., *Biophysical Chemistry* **1999,** *78* (1-2), 1-20; (j) Cramer, C. J.; Truhlar, D. G., *Chemical Reviews* **1999,** *99* (8), 2161-2200; (k) Bashford, D.; Case, D. A., *Annual Review Of Physical Chemistry* **2000,** *51*, 129-152; (l) Chen, J. H.; Im, W. P.; Brooks, C. L., *Journal of the American Chemical Society* **2006,** *128* (11), 3728-3736; (m) Feig, M.; Chocholousova, J.; Tanizaki, S., *Theoretical Chemistry Accounts* **2006,** *116* (1-3), 194-205; (n) Im, W.; Chen, J. H.; Brooks, C. L., Peptide

and protein folding and conformational equilibria: Theoretical treatment of electrostatics and hydrogen bonding with implicit solvent models. In *Peptide Solvation and H-Bonds*, Academic Press: 2006; Vol. 72, pp 173-198; (o) Lu, B. Z.; Zhou, Y. C.; Holst, M. J.; McCammon, J. A., *Communications in Computational Physics* **2008,** *3* (5), 973-1009; (p) Wang, J.; Tan, C. H.; Tan, Y. H.; Lu, Q.; Luo, R., *Communications in Computational Physics* **2008,** *3* (5), 1010-1031; (q) Yap, E.-H.; Head-Gordon, T., *Journal of Chemical Theory and Computation* **2010,** *6* (7), 2214-2224; (r) Geng, W.; Wei, G. W., *J. Comput. Phys.* **2011,** *230* (2), 435-457.

4. (a) Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T., *Journal of the American Chemical Society* **1990,** *112* (16), 6127-6129; (b) Srinivasan, J.; Trevathan, M. W.; Beroza, P.; Case, D. A., *Theoretical Chemistry Accounts* **1999,** *101* (6), 426-434.

5. Zhou, R., *PROTEINS: Structure, Function, and Genetics* **2003,** *53*, 148-161.

6. Ulmschneider, M. B., Ulschneider, J. P., Sansom, M. S., and Di Nola, A. , *Biophysical Journal* **2007,** *92*, 2338-49.

7. (a) Davis, M. E.; McCammon, J. A., *Journal of Computational Chemistry* **1989,** *10* (3), 386-391; (b) Nicholls, A.; Honig, B., *Journal of Computational Chemistry* **1991,** *12* (4), 435-445; (c) Bashford, D., *Lecture Notes in Computer Science* **1997,** *1343*, 233-240.

8. Baker, N.; Holst, M.; Wang, F., *Journal of Computational Chemistry* **2000,** *21* (15), 1343-1352.

9. (a) Bajaj, C.; Chen, S.-C.; Rand, A., *Siam Journal on Scientific Computing* **2011,** *33* (2), 826-848; (b) Boschitsch, A. H.; Fenley, M. O.; Zhou, H. X., *Journal of Physical Chemistry B* **2002,** *106* (10), 2741-2754; (c) Lu, B.; Cheng, X.; Huang, J.; McCammon, J. A., *Journal of Chemical Theory and Computation* **2009,** *5* (6), 1692-1699; (d) Totrov, M.; Abagyan, R., *Biopolymers* **2001,** *60* (2), 124-133.

10. Koehl, P., *Current Opinion in Structural Biology* **2006,** *16*, 142-151.

11. Eastwood, J. W., and Hockney, R. W. , *Computer Simulation Using Particles*. IOP Publishing Ltd.: Philadelphia, PA, USA, 1988.

12. Li, Z.; Ito, K., *The immersed interface method : numerical solutions of PDEs involving interfaces and irregular domains*. Society for Industrial and Applied Mathematics: Philadelphia, 2006; p xvi, 332 p.

13. Ye, X.; Wang, J.; Luo, R., *J. Chem. Theory Comput.* **2010,** *6* (4), 1157-1169.

14. Li, Z., *Taiwanese Journal of Mathematics* **2003,** *7* (1), 1-49.

15. Li, Z., *Numerical Algorithms* **1997,** *14* (269-293).

16. (a) Wang, J.; Cai, Q.; Xiang, Y.; Luo, R., *Journal of Chemical Theory and Computation* **2012,** *8* (8), 2741-2751; (b) Bruccorleri, R. E., Novotny, J., and Davis, M. E. , *Journal of Computational Chemistry* **1997,** *18* (2), 268-76.

17. Case, D. A.; Darden, T. A.; Cheatham, T. E. I.; Simmerling, C. L.; Wang, J.; Duke, R. E.; Luo, R.; Walker, R. C.; Zhang, W.; Merz, K. M.; Roberts, B.; Hayik, S.; Roitberg, A.; Seabra, G.; Swails, J.; Gotz, A. W.; Kolossvary, I.; Wong, K. F.; Paesani, F.; Vanicek, J.; Wolf, R. M.; Liu, J.; Wu, X.; Brozell, S. R.; Steinbrecher, T.; Gohlke, H.; Cai, Q.; Ye, X.; Wang, J.; Hsieh, M.-J.; Cui, G.; Roe, D. R.; Mathews, D. H.; Seetin, M. G.; Salomon-Ferrer, R.; Sagui, C.; Babin, V.; Luchko, T.; Gusarov, S.; Kovalenko, A.; Kollman, P. A. *AMBER 12*, University of California, San Francisco, 2012.

18. (a) Cai, Q.; Hsieh, M.-J.; Wang, J.; Luo, R., *Journal of Chemical Theory and Computation* **2010,** *6* (1), 203-211; (b) Wang, J.; Luo, R., *Journal of Computational Chemistry* **2010,** *31* (8), 1689-1698.

19. Ting, C. L.; Wang, Z. G., *Biophysical Journal* **2011,** *100* (5), 1288-1297.

Chapter 2: Applications of MMPBSA to Membrane Proteins I: Efficient Numerical Solutions of Periodic Poisson-Boltzmann Equation

Introduction

Electrostatic interactions play a major role in the function, structure, and dynamics of biomolecular systems. Modeling of these interactions in an accurate and efficient manner is thus of great importance and continues to be an active topic [1]. Most biomolecular systems exist in an aqueous environment. The effect of this solvent environment upon a biomolecular system must be accounted for when performing computation modeling and simulation. Such effects can be treated explicitly – i.e. by modeling each individual solvent molecule, or they may be treated implicitly – wherein solvent molecules are not included explicitly but instead are represented as a continuum. In the implicit treatment, the Poisson-Boltzmann (PB) equation has been established as a fundamental equation for modeling of continuum solvent electrostatic interactions [1].

The PB equation is a non-linear elliptical partial differential equation. Efficient and accurate solution for complex systems such as biomolecules is not trivial. In general, closed form solutions are not available and thus numerical methods are required. Incorporating the PB equation in a typical molecular simulation, or even using it as a post processing method to perform free energy and binding affinity calculations, involves computing solutions for numerous conformations. Thus, solving the equation and interpolating or processing the electrostatic energies, potential distributions, and so on must be accomplished very efficiently for it to become a useful computational model. Analytic solution of the PB equation is only attainable for systems with simple, highly symmetric geometry. Biomolecular systems, however, often exhibit extremely complex geometries. Thus a numerical solution is required. The finite difference method (FDM) [2] is apparently the most widely adopted method. The FDM is quite intuitive and straight-forward to construct. Its computations proceed quite rapidly. FDM solvers for the PB equation have been implemented in several programs, such as DelPhi [2a, 2c, 2i], UHBD [2b, 2d], APBS [2e, 2k], and in related modules of Amber [2j, 3] and CHARMM [2c, 2h]. The FDM proceeds by employing a grid, most often uniform and rectangular, to discretize the equation, building up a set of linear equations that may be solved by standard linear algebra methods. A description of the molecular surface is first constructed and then from it, the dielectric constant is mapped onto the grid. Classical FDM's lead to highly efficient solvers, such as preconditioned conjugate gradient or multi-grid algorithms, which have been developed to solve the equation [2b, 2j, 4]. Other numerical options include boundary element method (BEM) [5], and the finite-element method (FEM) [6]. The BEM seeks to obtain a linear system whose unknowns are either the induced surface charges [5a-d, 5h, 5i, 5k, 5l] or the normal components of the electric displacement [5e-g, 5j, 5m, 5n] on the boundary, providing a highly accurate description at the interface**.** The FEM [6] is based on the weak variational formulation. The electrostatic potential to be solved is approximated by a superposition of a set of basis functions.

In this study, we focus on the applicability of the PB equation to membrane bound systems, which have recently received increasing attention in modeling and simulation studies. Indeed, their roles as cell receptors and transmembrane channels make them good candidates for drug targets. Since protein structure and function is extremely sensitive to the surrounding environment, proper inclusion of a membrane is necessary to ensure accuracy when membrane proteins are studied. Therefore application of rationale design methodologies to membrane proteins requires properly modeled membrane environments. The PB equation can be used to provide an implicit membrane model to study membrane proteins. There has been a great deal of efforts to extend PB equation-based implicit solvent models to include a membrane region. While much of this effort has been directed toward adaptation of Generalized Born methodologies [20, 21,22,23,24] there have also been notable advances in implementing implicit membrane models under the full PB equation [25-27]. The inclusion of an implicit membrane region adds additional challenges and parameters to be considered depending on the choice of solvation model / method. This study is limited to the widely used finite-different methods to applications in membrane bound systems. A weighted harmonic averaging treatment was used for the dielectric constant at the solute-solvent-membrane interface [7], which is essentially the first-order immerse interface method [8]. This method was shown to converge quadratically as far as electrostatic solvation energies are concerned. In particular, the focus here is on applications to systems that must be modeled with periodic boundary conditions.

The use of periodic boundary conditions is a common practice in modeling and simulation of molecular systems and has long been applied to electrostatic calculations methodologies developed for systems with homogenous dielectric constants. One of the main advantages in using periodic boundary conditions is alleviation of computational artifacts resulting from edge effects. This turns out to be an idea setup to extend the PB solvers to the heterogeneous membrane/water environments. Modeling of an implicit membrane under the PB equation is typically accomplished by the inclusion of an additional dielectric region into the solvent model [9]. Edge effects become a pronounced concern under this model because the dielectric interface extends infinitely along the membrane plane. Thus the coefficients of the corresponding grid discretization are no longer uniform along the edges of the computational grid, causing difficulties in setting the widely used free boundary condition.

Implementation of the periodic boundary condition under a conjugate gradient (CG) solver [10] is relatively straight forward when employing an unconditioned solver. Unfortunately, such methods may require many iterations to converge [2b, 2j, 4, 10-11]. Preconditioning treatments, such as the modified Incomplete Cholesky preconditioner [2b, 2j, 4], and multi-grid based methods [2e, 11-12] can greatly enhance performance [2b, 4a, 11]. Adapting the preconditioner and multi-grid algorithms to allow periodic boundary conditions, however, is apparently much less trivial.

In the case of the modified Incomplete Cholesky preconditioned CG method, the structure of the preconditioner is dependent upon the band structure of the operator matrix being conditioned. Unfortunately, systems with periodic boundary conditions require operator matrices with an expanded band structure to account for interactions between opposing boundaries. More specifically, there are two additional bands for each periodic dimension (one in the upper triangular portion and one in the lower triangular portion). In the case of an explicitly conditioned algorithm [11], such changes must be reflected in the preconditioner matrix as well since it is applied directly to both the operator matrix and the starting conditions as well.

In the case of the geometric multi-grid method, implementation of periodic boundary conditions requires updating the core restriction, relaxation, and prolongation operators. Adaptation of the prolongation operator in particular requires special care since operator based prolongation is required to preserve convergence when solving the PB equation due to the presence of a spatially varying dielectric coefficient [11, 13]. Furthermore, care must be taken to ensure consistency of the periodic boundary between the restriction, relaxation, and prolongation operators.

This paper documents the implementation of the order 1 modified Incomplete Cholesky Conjugate Gradient (MICCG1 or ICCG) [2b, 2j, 4, 11] and geometric Multi-Grid (MG) [2e, 11-13] under the periodic boundary conditions as implemented in the Amber/PBSA module, along with subsequent optimization. The new periodic solvers were then tested on various systems to explore accuracy and efficiency, and also in a realistic MMPBSA application to highlight the importance of modeling the heterogenous membrane/water environment in biological applications.

Methods

Finite Volume Discretization of the Poisson-Boltzmann Equation

In most implicit solvation methods, the electrostatic energy is modeled by the Poisson-Boltzmann (PB) equation [1a, 1c, 1e-g, 1i, 1j, 1l, 1n, 14]. In its most general form, this is given by

$$\nabla[\varepsilon(\nabla\phi)] = -4\pi\rho - 4\pi\sum_i e_i z_i \lambda \exp(\frac{-z_i\phi}{k_b T}) \qquad (2.1)$$

This relates the electrical potential ϕ and dielectric constant ε to the solute charge distribution ρ and the charge distribution due to mobile solvated ions, as given by the summation in the second term on the right hand side. Here c_i is the bulk concentration of solute ion i with effective z_i, k_b and T are the Boltzmann constant and temperature, and λ is the Stern layer masking function which is 0 within the layer or 1 outside.

For sufficiently dilute ion concentrations, the second term on the right hand side may be linearized to yield

$$\nabla[\varepsilon(\nabla\phi)] = -4\pi\rho - 4\pi\sum_i \frac{c_i z_i^2 \lambda\phi}{k_b T} \qquad (2.2)$$

Due to its complexity a numerical solution is most often needed whether the full or linearized version is to be solved for anything but the simplest cases. Finite difference or finite volume approaches are commonly used for this due to their speed and relatively straight forward application [11].

As in a standard particle mesh setup [3, 11], the first step is to overlay a regular rectangular grid onto the system and then map the atomic point charges onto the grid using an appropriate assignment function. Next the dielectric constant is assigned to edges connecting each pair of neighboring grid nodes. The PB equation can then be discretized at each grid node, yielding the following operator stencil

$$\left[\begin{array}{l}\left(\begin{array}{l}\varepsilon_x[i,j,k]\phi[i+1,j,k]+\varepsilon_x[i-1,j,k]\phi[i-1,j,k]+\\ \varepsilon_y[i,j,k]\phi[i,j+1,k]+\varepsilon_y[i,j-1,k]\phi[i,j-1,k]+\\ \varepsilon_z[i,j,k]\phi[i,j,k+1]+\varepsilon_z[i,j,k-1]\phi[i,j,k-1]\end{array}\right)-\\ \left(\begin{array}{l}\varepsilon_x[i-1,j,k]+\varepsilon_x[i,j,k]+\varepsilon_y[i,j-1,k]+\\ \varepsilon_y[i,j,k]+\varepsilon_z[i,j,k-1]+\varepsilon_z[i,j,k]\end{array}\right)\phi[i,j,k])\end{array}\right]+h^2\lambda[i,j,k]\kappa^2\phi[i,j,k] \quad =\frac{-4\pi\rho[i,j,k]}{h}$$

(2.3)

Here $\varepsilon_x, \varepsilon_y,$ and ε_z represent the dielectric constants for grid edges along the x, y, and z directions respectively, h represents the grid spacing, and $\kappa^2 = 4\pi\sum_i c_i z_i^2 / k_b T$. The last detail to consider is the treatment of nodes at the edge of the computational grid, since these nodes have no neighbors along at least one direction. This requires defining a set of rules known as the boundary conditions. Implementations of boundary conditions depend on the particular linear solver algorithms used to solve the system of linear equations.

Matrix Representation of the Discrete Operator

It is useful to first discuss the existing linear system solvers, developed previously for isolated systems. Since all linear PB solvers are essentially solving a matrix vector equation, it is worthwhile to cast the problem under the framework of a matrix vector equation

$$\mathbf{Ax} = \mathbf{b} \qquad (2.4)$$

where **A** is a matrix representation of the PB operator stencil as defined in eqn (2.3), **x** is the desired potential solution, and **b** is the charge distribution of the solute. Since **b** is a vector representation of the computational grid, matrix **A** requires a number of entries equal to the square of the number of grid nodes. This quickly grows intractable as the size of the computational grid increases. Fortunately, the matrix itself has a band-like structure that is quite sparse. Further, it never needs to be constructed or stored explicitly in practice when applying a solver.

For non-periodic solvers, the terms of the **A** matrix that connect nodes at the edge of the grid to non-existent nodes were simply omitted. The most naïve variant of this leads to the zero (or conductor) boundary condition. This boundary condition models the computation grid as being placed in an infinitely large metal (with an infinitely high dielectric constant), which is a reasonable approximation for aqueous solution given the relatively large dielectric constant of water. A more realistic but also more time-consuming approach is the free boundary condition, i.e., the computation grid is modeled as being isolated in infinitely large dielectric medium, e.g. water.

The periodic boundary condition, on the other hand, essentially mimics an infinite periodic lattice, wherein the computation grid is representative of the central cell. This is accomplished by treating nodes on the edge of the computation grid as if they were adjacent to corresponding nodes from the opposing edge or face, i.e.

$$\begin{aligned} \phi[1,j,k] &= \phi[x_m+1,j,k] \\ \phi[i,1,k] &= \phi[i,y_m+1,k] \\ \phi[i,j,1] &= \phi[i,j,z_m+1] \end{aligned} \tag{2.5}$$

where x_m, y_m, z_m are the maximum grid point indices in the x, y, and z dimensions respectively. For illustrative purposes, an operator matrix constructed for a 3x4x5 grid with uniform dielectric constant of 1 is shown in . Upon inspection, the additional band structure required for periodic boundary conditions is clearly visible and these bands must be properly taken care of so that the desired periodic boundary condition is enforced upon the solution.

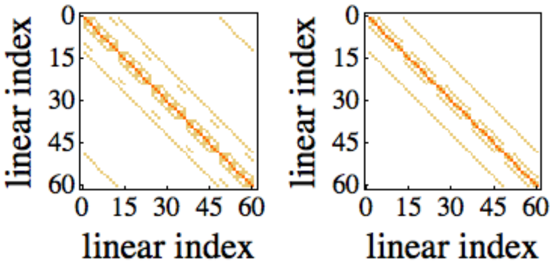

Figure 2.1. Discretized Laplacian Operator for 3x4x5 Grid

Left: Illustration of the band structure for the finite-difference discretization of the Laplacian operator under periodic boundary conditions. Right: Illustration of the band structure for the finite-difference discretization of the Laplacian operator under fixed potential boundary conditions.

Treatment of Charged Solutes

Poisson Equation As mentioned above, a system with the periodic boundary condition effectively mimics the physical case of an infinite lattice, with the computational grid being analogous to a central or unit cell of this lattice. For a system with non-zero net charge, a uniform neutralizing plasma must be used, as is standard for electrostatic calculations involving periodic ionic systems [15]. This may be accomplished by subtracting the net charge uniformly from all grid nodes before solving the linear systems.

The complication of the standard practice, in the case of periodic systems, is the unknown constant potential offset that is introduced. It is well known that the Poisson equation allows a family of solutions that only differ by a constant. When the boundary potentials are specified, the constant can then be uniquely determined. Unfortunately this is not possible in periodic systems. Thus the use of uniform plasma further complicates the issue of unknown potential offset in the solution of periodic Poisson systems. In particular, this poses a problem when attempting to obtain consistent results from different linear system solvers since it is not guaranteed that different linear system solvers lead to the same constant potential offset. Indeed our numerical tests show that different linear solvers do give different potential offsets.

In order to obtain consistent results among different solvers, it is necessary to impose a consistent potential offset to remove any possible difference caused by different numerical solvers. This can be easily realized by subtracting the mean potential on all grid nodes. Apparently the extra step incurs little additional CPU time and does not change any derivatives of the potential distribution.

Poisson-Boltzmann Equation Under physiological conditions, it is desirable to include a description of salt or ionic strength in the implicit solvation models. Under the PB framework, this is modeled by the addition of a second operator term that is itself a function of the electrostatic potential. This paper examines only the case where the term may be approximated by a linear function, such as the case with solutions containing relatively dilute (on the order of a few hundred mM) ionic concentrations in the weak electrostatic field.

Physically the salt term acts as an additional charge term that in principle provides a means to neutralize the system. This turns out to be the case because the PB equation is satisfied when the system electrostatic free energy is at its minimum [16]. It is apparent that the charge of the unit cell must be neutral for the free energy to be at its minimum at all since either positive or negative net charge leads to a diverging and positive free energy for the infinite periodic lattice. Note too that there is sufficient freedom to set the amount of charge by the salt term since we have an open system. Therefore a PB solver effectively looks for a solution that neutralizes the charge of the unit cell. Our numerical experiment shows that indeed this is the case for a large test set of nucleic acids of very different net charges as shown in Results and Discussion. There is, however, a final detail to consider. In cases where the ionic strength is very low, the PB operator approaches the Poisson operator. While such cases remain well posed, it is possible that some numerical solvers may experience difficulties. Testing was therefore run to examine the numerical stability of all implemented periodic linear solvers over a range of salt concentrations as presented in Results and Discussion.

Next, it is also worthwhile to point out that it is not necessary to reset the potential offset either, because the constant potential is no longer a solution of the PB equation due to the existence of the potential-dependent salt term. The requirement of minimum electrostatic free energy has uniquely determined the solution of the PB equation [16], even if the periodic boundary condition is used. Our numerical tests show that subtraction of the mean potential from each grid nodes does not change the electrostatic free energies for the tested nucleic acids when a non-zero ionic strength term is used.

Adaptation of Conjugate Gradient Type Solvers to Periodic Boundary Condition

The CG method is an iterative method for numerical solution of a matrix vector problem. CG attempts to effectively expand the solution in terms of mutually orthogonal components. At each step the solution vector is updated by adding to it, a vector orthogonal to the current solution vector, as shown in the following pseudo code

```
$r_0 = b - Ax_0$
$p_0 = r_0$
$while(unconverged)$
    $\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k A p_k$                                        (2.6)
    $if(| r_{k+1} | / | r_k |< tol) -> return(r_{k+1})$
    $\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
    $p_{k+1} = r_{k+1} + \beta_k p_k$
    $k = k + 1$
$end\, while$
```

Updating CG for periodic boundary conditions primarily involves implementing periodicity (via an indexing array) when applying the matrix vector multiplication (**Ap**$_k$). As noted earlier, this is equivalent to adding a set of additional bands to the **A** matrix. While these additional bands in the **A** matrix pose little problem for the unconditioned CG algorithm, the more efficient modified Incomplete Cholesky (MIC) preconditioner [11] that was formulated for non-periodic systems must be reformulated in order to incorporate periodicity.

When a preconditioner is used, the additional bands due to periodicity must be integrated into the preconditioning matrix as well. To do this we begin with the MIC preconditioner currently for non-periodic systems as detailed by Wang and Luo [11]

$$(\mathbf{M}^{-1}\mathbf{A}\mathbf{M}^{-1})(\mathbf{M}\mathbf{x}) = \mathbf{M}^{-1}\mathbf{b} \tag{2.7}$$

$$\mathbf{M} = (\tilde{\mathbf{D}} + \mathbf{L})\tilde{\mathbf{D}}^{-1}(\tilde{\mathbf{D}} + \mathbf{L}^{\mathbf{T}}) \tag{2.8}$$

where **M** is the preconditioning matrix**,** $\tilde{\mathbf{D}}$ is a positive diagonal matrix derived from the diagonal of **A**, and **L** is the lower triangular portion of **A** with diagonal excluded. The values in $\tilde{\mathbf{D}}$ are computed as

$$\begin{aligned}\tilde{d}_i^{-1} = a_{i,i} &- a_{i-\Delta x,i-\Delta x}(a_{i-\Delta x,i-\Delta x} + \alpha a_{i-\Delta x,i-\Delta y} + \alpha a_{i-\Delta x,i-\Delta y} + a_{i-\Delta x,i-\Delta z}) \cdot \tilde{d}_{i-\Delta x} \\ &- a_{i-\Delta y,i-\Delta y}(\alpha a_{i-\Delta y,i-\Delta x} + a_{i-\Delta y,i-\Delta y} + \alpha a_{i-\Delta y,i-\Delta z}) \cdot \tilde{d}_{i-\Delta y} \\ &- a_{i-\Delta z,i-\Delta z}(\alpha a_{i-\Delta z,i-\Delta x} + \alpha a_{i-\Delta z,i-\Delta y} + a_{i-\Delta z,i-\Delta z}) \cdot \tilde{d}_{i-\Delta z}\end{aligned} \tag{2.9}$$

This is equivalent to equation 11 in Wang and Luo [11] but with a slight notation change. Here *i* represents the 1-d sequential index of an arbitrary grid point. The off-diagonal entries along each row, therefore, correspond to the adjacent grid nodes along the x, y, or z direction. The values of Δx, Δy, and Δz represent the shifts in 1-d sequential indexing that correspond to 1 unit shifts along the x, y, and z dimensions, respectively. They can be derived as follows. The 1-d sequential grid index, *i*, corresponding to a 3-d spatial grid index [*x,y,z*] may be calculated as

$$i[x, y, z] = x + (y-1) \cdot x_m + (z-1) \cdot x_m \cdot y_m \tag{2.10}$$

Thus the needed shifts can be computed as

$$\begin{aligned}\Delta x &= 1 \\ \Delta y &= x_m \\ \Delta z &= x_m \cdot y_m\end{aligned} \tag{2.11}$$

Note also that the matrix described here by equation (2.9) represents a lower triangular matrix. Due to the symmetry of the **A** matrix, only the lower triangular portion needs to be considered in constructing *d*. The upper triangular portion is simply its transpose.

In this notation, inclusion of the additional bands due to periodicity becomes relatively straightforward. Each additional band in **A** corresponds to a “wrapping” from one edge of the computational grid to the opposite side. Since we are concerned only with bands occurring in columns prior to a given matrix diagonal entry, only those bands corresponding to wrapping from the last node in a dimension to the first node along the same dimension are added. Updated equation (2.9) then yields

$$
\begin{aligned}
\tilde{d}_i^{-1} = \quad & a_{i,j} - a_{i-\Delta x,j-\Delta x}(a_{i-\Delta x,j-\Delta x} + \alpha a_{i-\Delta x,j-\Delta y} + \alpha a_{i-\Delta x,j-\Delta z} \\
& +\alpha a_{i-\Delta x,j-wx} + \alpha a_{i-\Delta x,j-wy} + \alpha a_{i-\Delta x,j-wz}) \cdot \tilde{d}_{i-\Delta x} \\
& -a_{i-\Delta y,j-\Delta y}(\alpha a_{i-\Delta y,j-\Delta x} + a_{i-\Delta y,j-\Delta y} + \alpha a_{i-\Delta y,j-\Delta z} \\
& +\alpha a_{i-\Delta y,j-wx} + \alpha a_{i-\Delta y,j-wy} + \alpha a_{i-\Delta y,j-wz}) \cdot \tilde{d}_{i-\Delta y} \\
& -a_{i-\Delta z,j-\Delta z}(\alpha a_{i-\Delta z,j-\Delta x} + \alpha a_{i-\Delta z,j-\Delta y} + a_{i-\Delta z,j-\Delta z} \\
& +\alpha a_{i-\Delta z,j-wx} + \alpha a_{i-\Delta z,j-wy} + \alpha a_{i-\Delta z,j-wz}) \cdot \tilde{d}_{i-\Delta z} \\
& -a_{i-wx,j-wx}(\alpha a_{i-wx,j-\Delta x} + \alpha a_{i-wx,j-\Delta y} + \alpha a_{i-wx,j-\Delta z} \\
& +a_{i-wx,j-wx} + \alpha a_{i-wx,j-wy} + \alpha a_{i-wx,j-wz}) \cdot \tilde{d}_{i-wx} \\
& -a_{i-wy,j-wy}(\alpha a_{i-wy,j-\Delta x} + \alpha a_{i-wy,j-\Delta y} + \alpha a_{i-wy,j-\Delta z} \\
& +\alpha a_{i-wy,j-wx} + a_{i-wy,j-wy} + \alpha a_{i-wy,j-wz}) \cdot \tilde{d}_{i-wy} \\
& -a_{i-wz,j-wz}(\alpha a_{i-wz,j-\Delta x} + \alpha a_{i-wz,j-\Delta y} + \alpha a_{i-wz,j-\Delta z} \\
& +\alpha a_{i-wz,j-wx} + \alpha a_{i-wz,j-wy} + a_{i-wz,j-wz}) \cdot \tilde{d}_{i-wz}
\end{aligned}
\tag{2.12}
$$

where *wx*, *wy,* and *wz* are the shifts in 1-d sequential indexing that correspond to wrapping from x=x_m to x=1, y=y_m to y=1, and z=z_m to z=1 respectively. Computation of *wx*, *wy*, and *wz* will be discussed shortly.

Implementation of the newly developed preconditioning algorithm poses one further complication. The preconditioner developed for non-periodic systems used padded arrays to store the coefficient matrices required. This was done to permit the use of a single inner loop over all entries of each array rather than using a separate loop to iterate over each spatial dimension. This setup then facilitated the CPU pipeline optimization. This previous approach requires extending the size of each array by an amount proportional to roughly twice the number of elements on the grid's *z* faces, which corresponds to the largest 1-d sequential shift needed to describe shifting along the various 3-d grid dimensions.

In the case of a periodic system, however, grids along the $z = 1$ boundary must access grids at the $z = z_m$ boundary. To determine how much padding is needed, the 1-d sequential shifts *wx, wy, and wz* are derived as follows

$$\begin{aligned}
i[x,y,z] &= x + x_m(y-1+y_m(z-1)) \\
i[1,y,z] &= 1 + x_m(y-1+y_m(z-1)) \\
i[x_m,y,z] &= x_m + x_m(y-1+y_m(z-1)) \\
i[x,1,z] &= x + x_m(y_m(z-1)) \\
i[x,y_m,1] &= x + x_m(y_m-1) \\
i[x,y,1] &= x + x_m(y-1) \\
i[x,y,z_m] &= x + x_m(y-1+y_m(z_m-1))
\end{aligned}$$

$$\begin{matrix} i[x_m+1,y,z] = i[1,y,z] \\ i[1-1,y,z] = i[x_m,y,z] \end{matrix} \Rightarrow \begin{matrix} i[x_m+1,y,z] = i[x_m-(x_m-1),y,z] = i[x_m,y,z]-(x_m-1) \\ i[1-1,y,z] = i[1+(x_m-1),y,z] = i[1,y,z]+(x_m-1) \end{matrix}$$

$$wx = x_m - 1$$

$$\begin{matrix} i[x,y_m+1,z] = i[x,1,z] \\ i[x,1-1,z] = i[x,y_m,z] \end{matrix} \Rightarrow \begin{matrix} i[x,y_m-(y_m-1),z] = i[x,y_m,z]-x_m(y_m-1) \\ i[x,1+(y_m-1),z] = i[x,1,z]+x_m(y_m-1) \end{matrix}$$

$$wy = x_m y_m - x_m$$

$$\begin{matrix} i[x,y,z_m+1] = i[x,y,1] \\ i[x,y,1-1] = i[x,y,z_m] \end{matrix} \Rightarrow \begin{matrix} i[x,y,z_m+1] = i[x,y,z_m-(z_m-1)] = i[x,y,z_m]-x_m y_m(z_m-1) \\ i[x,y,1-1] = i[x,y,1+(z_m-1)] = i[x,y,1]+x_m y_m(z_m-1) \end{matrix}$$

$$wz = x_m y_m z_m - x_m y_m$$

(2.13)

Thus the number of padding elements required is equal to twice $x_m y_m(z_m$-1). Unfortunately, this would nearly triple the amount of storage space required. Thus three-dimensional array was used instead.

To preserve the previous single loop structure, an indexing array was introduced to map the one-dimensional indices into the corresponding three-dimensional indices. The extra two-grid points worth of padding along each dimensions would handle the needed wrapping.

Adaptation of Successive Over Relaxation to Periodic Boundary Condition

Successive Over Relaxation (SOR) is another method of iteratively solving a system of linear algebraic equations. It bears a striking resemblance to the Jacobi method (or the damped variant thereof), but seems to provide better convergence properties. In terms of the matrix vector formulation, the damped Jacobi (or SOR) may be expressed as

$$x^{(k+1)} = (1-\omega)x^{(k)} + \omega D^{-1}(b - Rx^{(k)}) \qquad (2.14)$$

where *k* is the *k*th iteration, ω is a damping coefficient, **D** is the diagonal of **A**, and **R** is (**A**-**D**). The primary difference in terms of implementation is that Jacobi may be performed out of place while SOR and Gauss-Siedel are performed in place. A discussion of the convergence properties of Jacobi iteration (which extends to SOR) can be found in the textbook by Hackbusch [17]. One is also much more restricted in choice of damping coefficient when using Jacobi, which requires a coefficient of less than unity, whereas SOR requires a coefficient under 2. Thus SOR may employ coefficients between values of 1 and 2, allowing for more rapid convergence. Although convergence is not necessarily guaranteed for either method when considering a general matrix, it can be shown that convergence is guaranteed for symmetric positive-definite matrices, such as those which arise from a finite-difference or finite-volume discretization of the linearized PB equation.

Like the unconditioned CG method, however, SOR still converges very slowly compared to conditioned or multigrid methods. However, as noted in the text by Hackbusch [17], an analysis of the convergence properties with respect to the spectral content of the residual shows that the Jacobi method (and SOR and Gauss-Seidel methods as well) is able to attenuate the short wavenumber components of the residual very rapidly and that the poor convergence is due to the long wavenumber components for which the Jacobi method converges very slowly. Thus, Jacobi (or SOR) may be conceived as a "smoother" of the residual for multigrid solvers [17].

Adaptation of single grid SOR to periodic boundary conditions is relatively straightforward. Similar to CG, the primary difference between implementation of periodic boundary conditions versus fixed potential boundary conditions lies in the presence of additional band structure in the operator matrix, **A**. As noted above, these additional bands can be implemented by making use of an indexing array (instead of padding working arrays as in conductor or free boundary conditions).

Like unconditioned CG, SOR (and Jacobi) often require a very large number of iterations, and correspondingly, very slow to converge when compared with more advanced methods, as is shown in Results and Discussion. Thus SOR is best utilized as a smoother in the multi-grid methods as discussed below.

Adaptation of Geometric Multigrid to Periodic Boundary Condition

The final solver method to consider is the geometric multi-grid solver. In this study, focus is limited to the geometric multigrid approaches as the infrastructure is already in place in Amber/PBSA as documented by Wang and Luo [11], but other multi-grid variants such as algebraic multigrid [11], and more recently, combinatorial multigrid [18] or Lean algebraic multigrid [19] may also provide viable options.

Geometric MG functions by generating a hierarchy of successively sparser grids. In the current implementation, a four level V-cycle scheme, grid spacing is doubled at each successive level, and a total of four levels are used. Thus, the coarsest grid is smaller by a factor 512 when compared with the finest grid. See the paper by Wang and Luo [11] for an in-depth discussion of the original formulation of the geometric multi-grid method and other linear solvers, geared toward non-periodic systems. There are three operations that must be performed at each level during the course of a single cycle: restriction (projecting the residual of the current grid onto the next coarser grid), relaxation (smoothing of the residual of the current grid), and prolongation (projection of the current grid onto the next finer grid).

The restriction operation, which interpolates values from a finer grid onto the next coarser grid, is carried out as a tri-linear interpolation using three-dimensional indexing. Adaptation of this operator for use with periodic boundary conditions simply requires use of modular arithmetic when computing shifts. In addition, restriction of the **A** matrix must be applied in a periodic fashion at the boundaries. Again, the existing infrastructure is easily adapted by utilization of modular arithmetic.

The relaxation operation is carried out as several iterations of an appropriate iterative method, in this case, the Gauss-Siedel method, which is equivalent to a special case of SOR with a relaxation coefficient of unity. As mentioned in the earlier paper by Wang and Luo [11], SOR does a poor job at maintaining solution smoothness and would disrupt convergence. Thus Gauss-Siedel is used to smooth between prolongation and restriction and SOR is used only as a direct solver for the coarsest level. Extension of the matrix operator to allow periodic boundary conditions can be accomplished via an indexing array, and a separate indexing array must be generated for each level of the multi-grid hierarchy.

The prolongation is the most complicated operation. In cases where the coefficients of the underlying partial differential equation are either spatially uniform (or at least vary smoothly and do not deviate by an order of magnitude or more), tri-linear interpolation would suffice [12b]. Unfortunately, unless solving a vacuum or uniform dielectric system, this is not the case. Thus, a more complex prolongation operator is required. This is because the spectral content of the solution on a fine grid exhibits contents of high wave numbers. They cannot be expressed on a coarser grid. Consequently a linear interpolation of a coarse grid solution onto the fine grid introduces an artificial smoothing of the solution that in turn greatly disrupts convergence properties. This can be handled by using a operator-based prolongation [12b]. The method was previously implemented [11] using the one-dimensional array indexing. Adapting the algorithm for use with three-dimensional indexing grid is apparently non-trivial, so that a padding/virtual grid approach is utilized. In this case, the current coarse grid is padded on each side. The corresponding periodic images are projected onto the padded edges. This padded grid can then be fed into the existing prolongation algorithm and treated as if it is a non-periodic system.

MMPBSA Calculations

As reviewed one of the major applications of numerical PB solvers is in the prediction of protein-ligand binding affinities. Currently, the AMBER suite provides the MMPBSA module that automates the computation of binding affinity from a molecular dynamics trajectory. Addition of the heterogeneous membrane/water model in the numerical PB solvers clearly facilitates extension of this widely used method to membrane-bound receptors. The benefit of modeling the membrane implicitly for MMPBSA calculation was demonstrated here with recently published crystal structures and binding affinities for the P2Y12 human platelet receptor [20]. The study analyzed the protein in complex with three different ligands. Additionally binding affinities for the D294N mutant in complex with the same three ligands were also provided. This makes P2Y12R an interesting initial candidate to demonstrate the effect of the membrane via the implicit PB solvent model.

To prepare for the MMPBSA calculation, all-atom simulations of the P2Y12R receptor-ligand structures, both the wild type and mutant, were first conducted with the Lipid14 force field [21] following the protocol in Ref. [22]. The membrane protein system was constructed using the web-based CHARMM membrane builder GUI [23]. The MODELER program [24] was used to generate structures of missing loops. Once molecular dynamics trajectories of the six complexes were attained, the MMPBSA method was employed, by hand, to process 100 frames (1 ns) of each complex trajectory. The change in relative binding free energies between all pairwise systems (ΔΔG) were then computed and compared with the measured values as discussed in Results and Discussion.

Computational Details

The periodic linear PB solvers were implemented under the PBSA module of the 2015 release of AmberTools [25]. Both a protein test set [11] and a nucleic acid test set [26] were utilized to validate the implementations. Charges and modified bond radii were assigned according to Cornell et. al. [27]. The ratio of grid dimension over the solute dimension was set to a value of 2 if not specified otherwise. Grid spacing was set at 0.5 Å. For easy comparison, all periodic solvers were tested with grid dimensions in the multiples of 16 as for the geometric multi-grid solver. Water dielectric was set to a value of 80 and vacuum dielectric was set to 1 if not specified otherwise. The dielectric constants along solute boundary interpolated via weighted harmonic averaging [7]. For SOR, the relaxation coefficient was set to 1.95. For ICCG, the relaxation coefficient was first optimized over a range of values from -1 to 1 using a short peptide as a test system. The possibility of non-equal coefficients for periodic and non-periodic relaxation coefficients for the preconditioner was investigated as well.

Results and Discussion

Validation of Periodic Numerical Solvers

The electrostatic energy of periodic and non-periodic systems should converge to the same value as the size of the system or central cell is extended toward infinity provided that the system is neutral in charge. This provides a means of validating the various periodic solvers by comparing the energies with those attained by the non-periodic solvers, which are known to function properly. Similarly, the deviation between the energies predicted should reduce as the distance between the boundary of the grid and the boundary of the solute is increased.

The periodic CG (PCG) solver was first evaluated with three simple model systems, each consisting of a low dielectric spherical cavity of 2 Angstrom imbedded with a dipole, quadrapole, or octapole, respectively. Testing was performed using the PBSA program [7, 11, 26] of the Amber simulation package [25] to verify proper convergence of energies by the periodic and conductor boundary conditions to those by the free boundary condition as the dimensions of the box is increased. The fill ratio of the solute dimension over the finite-difference grid dimension as defined by Wang and Luo [11] was used to control the grid dimension. The relative deviation between the periodic and free boundary conditions and the conductor and free boundary conditions were then plotted as a function of fill ratio on a log-log scale. The results are shown in . Inspection of the plot shows that the difference in computed electrostatic energies converges rapidly to zero as the fill ratio is increased. Specifically, the differences are in the order of 10^{-4} in the tested solvated cases when the commonly used fill ratio of 2 is used, indicating highly consistent energy calculations. The difference comes down slower in vacuum cases as expected due to the lack of solvent screening. This is also consistent with previous observations when comparing the vacuum electrostatic energies computed with the periodic CG solver and those with FFT or PME methods in the Amber simulation package [28].

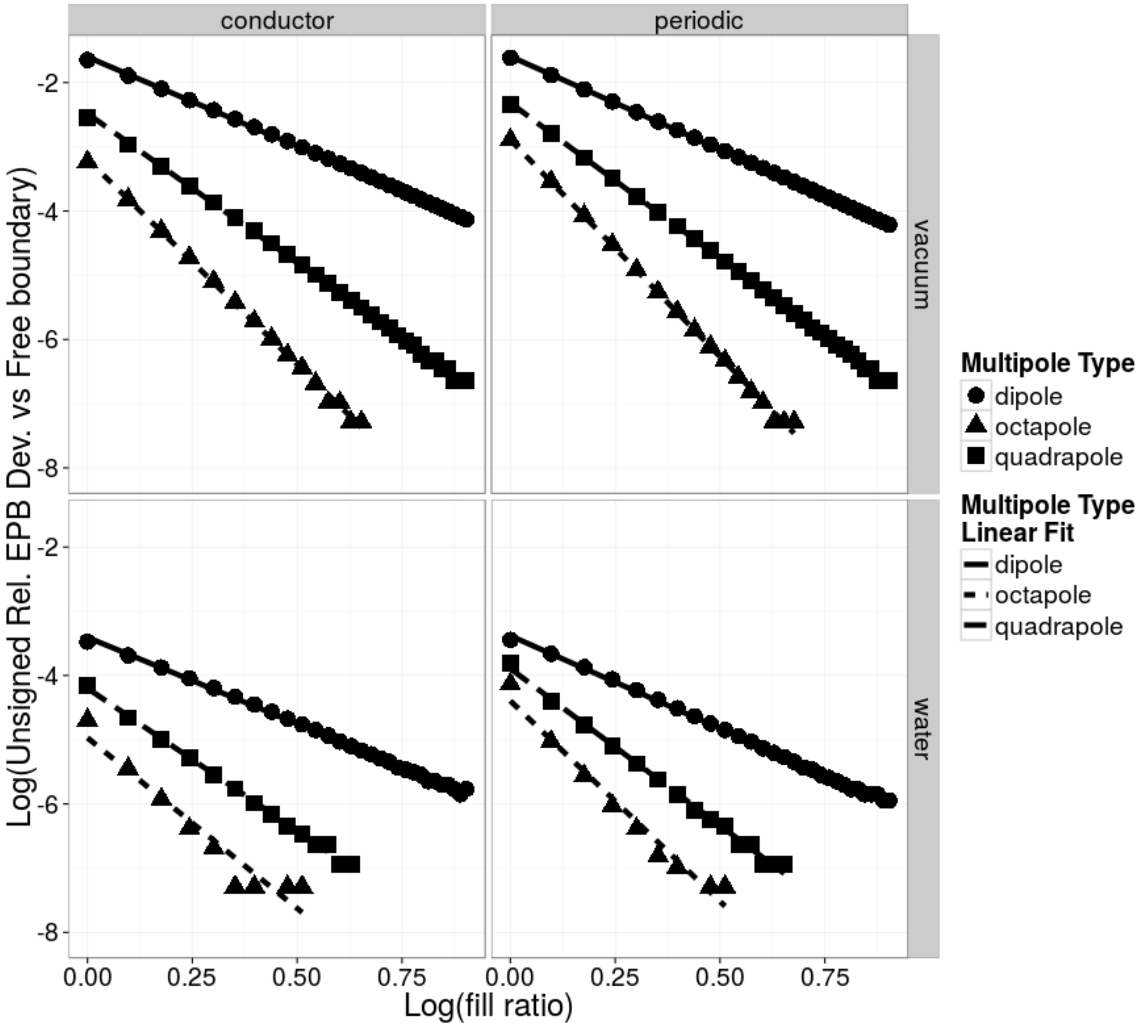

Figure 2.2. Differences between Electrostatic Energies by Different Boundary Conditions vs Fill Ratios

This figure illustrates the grid-dimension dependence in deviations of electrostatic energies between the periodic/conductor boundary condition and the free boundary condition in vacuum (upper) and in water (lower) for three model systems: low dielectric spherical cavities with a radius of Angstrom imbedded with a dipole, quadrapole, or octapole, respectively. The dipole consists of unit positive and negative charges located at positive and negative .5 Å from the cavity center along the x axis. The quadrapole consists of four unit positive and negative charges located at the vertices of a unit square in the x-y plane and centered at the cavity center. The octapole consists of eight unit positive and negative charges located at the vertices of a unit cube centered at the cavity center. The program reports the sum of the Coulomb and reaction field energies as a single electrostatic energy.

To validate the other periodic solvers, i.e. periodic SOR (PSOR), periodic incomplete cholesky conditioned conjugate gradient (PICCG), and periodic geometric multigrid (PMG), electrostatic energies for peptides and proteins of sizes ranging between ten and three hundred residues were computed and compared with the electrostatic energies with the PCG method. The grid dimensions and origins were chosen to be exactly the same across all tested solvers for each molecule to remove the discretization discrepancy. Implicit water solvation was implemented by setting the exterior dielectric constant to 80 and the interior dielectric constant to 2. The relative deviations between tested solvers and the CG solver were plotted as a function of the grid volume as shown in . The comparison was conducted for all three commonly used boundary conditions, i.e. free, periodic, and conductor, respectively. Overall, relative deviations less than the convergence criterion used, i.e. 1 ppm, were observed for all three testing conditions of all tested solvers.

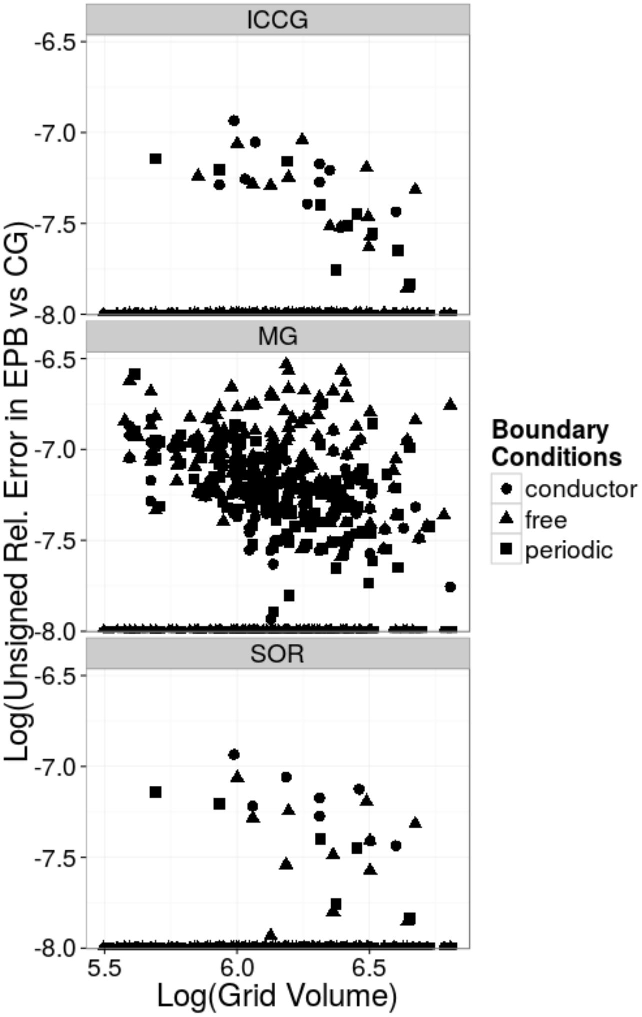

Figure 2.3. Differences between Electrostatic Energies by Different Solvers vs Grid Volumes

Log-Log plots of differences in electrostatic energies by PICCG (top), PMG (middle), and PSOR (bottom) with respect to PCG versus computation grid volumes for the protein test set.

Optimization of Conditioning Coefficients for ICCG

The PICCG solver may be optimized by adjusting the relaxation coefficient for the preconditioner. This coefficient is applied to the terms of each band in the algorithm's operator matrix during construction of the preconditioner. In the case of non-periodic systems, most of these bands contain roughly equal numbers of elements. In the case of periodic systems, additional bands in the operator matrix are present due to periodic wrapping that occurs on the faces of the computation grid.

These bands tend to have far fewer elements than the bands associated with connectivity of interior grid nodes. It is therefore unclear as to whether or not the same conditioning coefficient can be used to retain the high performance of the preconditioner.

To ensure optimal parameterization of the preconditioner, a series of tests were set up to examine the number of iterations required for convergence relative to choice of periodic and non-periodic relaxation coefficients. This series was constructed over a range of scaling coefficients ranging from positive to negative .975 in increments of .025 using a model of the c-terminal transmembrane helix of aquaporin (pdb ID: 1lH5). The results of this parameter scanning are shown as a contour plot in . Examination of the plot clearly shows that the optimal values for both periodic and non-periodic relaxation coefficients lie in a single basin with values nearby .90. Although further testing may be required to investigate possible dependence of optimal values upon system specific factors such as grid size, net charge, etc. The initial screening indicates that the same optimal value can be used for both periodic and non-periodic coefficients and is consistent with the previously reported value for the non-periodic coefficient that was optimized with different molecules [2j].

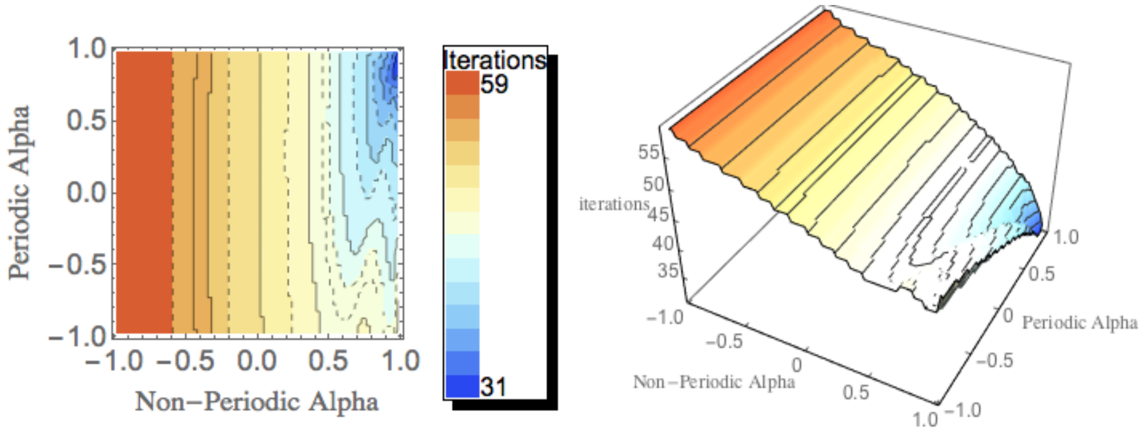

Figure 2.4. Optimization of Scaling Coefficients for PICCG

Contour plot of the number of iterations required for convergence of the PICCG method on a small model peptide system as a function of the scaling coefficients for periodic and non-periodic bands.

Efficiency Analysis of Numerical Solvers

In order to investigate the efficiency of the various linear solvers under different boundary conditions, additional timing analyses were conducted for the same set of computations used in the validation above. It was observed in the paper by Wang and Luo [11] that a major portion of the time used when applying the non-periodic solver is spent in computing the necessary virtual charges to be applied along the grid boundary surfaces, as is needed when setting up the free boundary condition. Thus two timing metrics were examined. First, the times used by the linear solver were recorded with the boundary condition setup time excluded. Second, the total times from start to finish were recorded. The total number of iterations required by the solvers was the first metric used to compare the effectiveness of the conditioning used, given different boundary conditions. The number of iterations required for convergence for conditioned and unconditioned periodic and non-periodic algorithms are shown in . Inspection of clearly shows that both periodic and non-periodic conditioned solvers require on the order of ten to twenty times fewer iterations than their unconditioned counterparts. It can also be seen that the non-periodic solvers perform slightly more efficiently than the periodic solvers in terms of number of iterations required to converge. Worth noting is the scaling of the MG solver regardless of the boundary conditions.

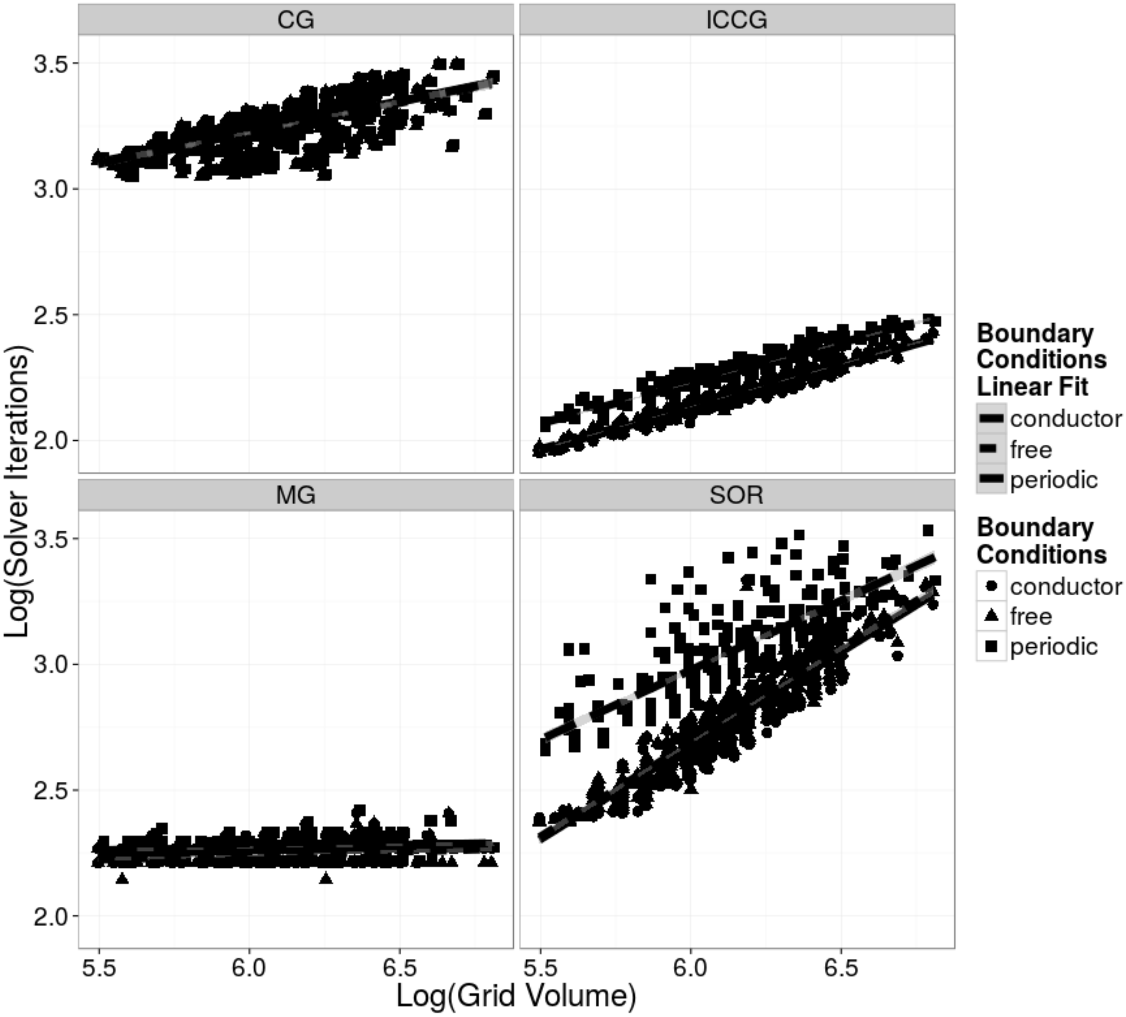

Figure 2.5. Solver Iteration Scaling: Log Iteration Required vs Log Grid Volume

Log-log plots of iteration steps required to reach convergence versus total number of grid points (grid volume). In the case of the PMG method, "effective" iteration steps are used. This is computed as the sum of iteration steps at each grid level divided by the scaling factor of that grid level relative to the finest grid level, e.g. factors of 1, 8, 64, and 512, respectively. Top Left: PICCG, Top Right: PCG, Bottom Left: 4 Level V-Cycle PMG with SOR/Gauss-Siedel Relaxation, Bottom Right: PSOR.

The analysis presented in **Figure 5** does not, however, consider the fact that a larger grid volume takes more time at each iteration step. Thus we further analyzed the total CPU time used by the linear solvers to reach convergence for the same tested systems. illustrates relative efficiencies in terms of solver CPU times for all tested solvers at tested boundary conditions. From the plots on the left-hand side, it can clearly be seen that the periodic solvers are less efficient than the corresponding non-periodic solvers. The reduced efficiency is expected, however, since the periodic solvers' operator matrices include roughly twice as many bands as the equivalent non-periodic solvers due to the inclusion of extra terms required to implement periodic boundaries.

Despite the reduction in efficiency of the solver algorithm itself, the periodic solvers are actually slightly faster than the non-periodic solvers when the free boundary condition is used, as shown in , which plots total computation time v.s. grid size. This is evident from the plot on the left-hand side. As noted previously, this is most likely due to the cost to compute the virtual charges needed to implement the free boundary condition. Indeed, the fraction of solver time over total computation time goes down when the grid volume is increased as shown in . This effect is more pronounced for the more efficient solvers.

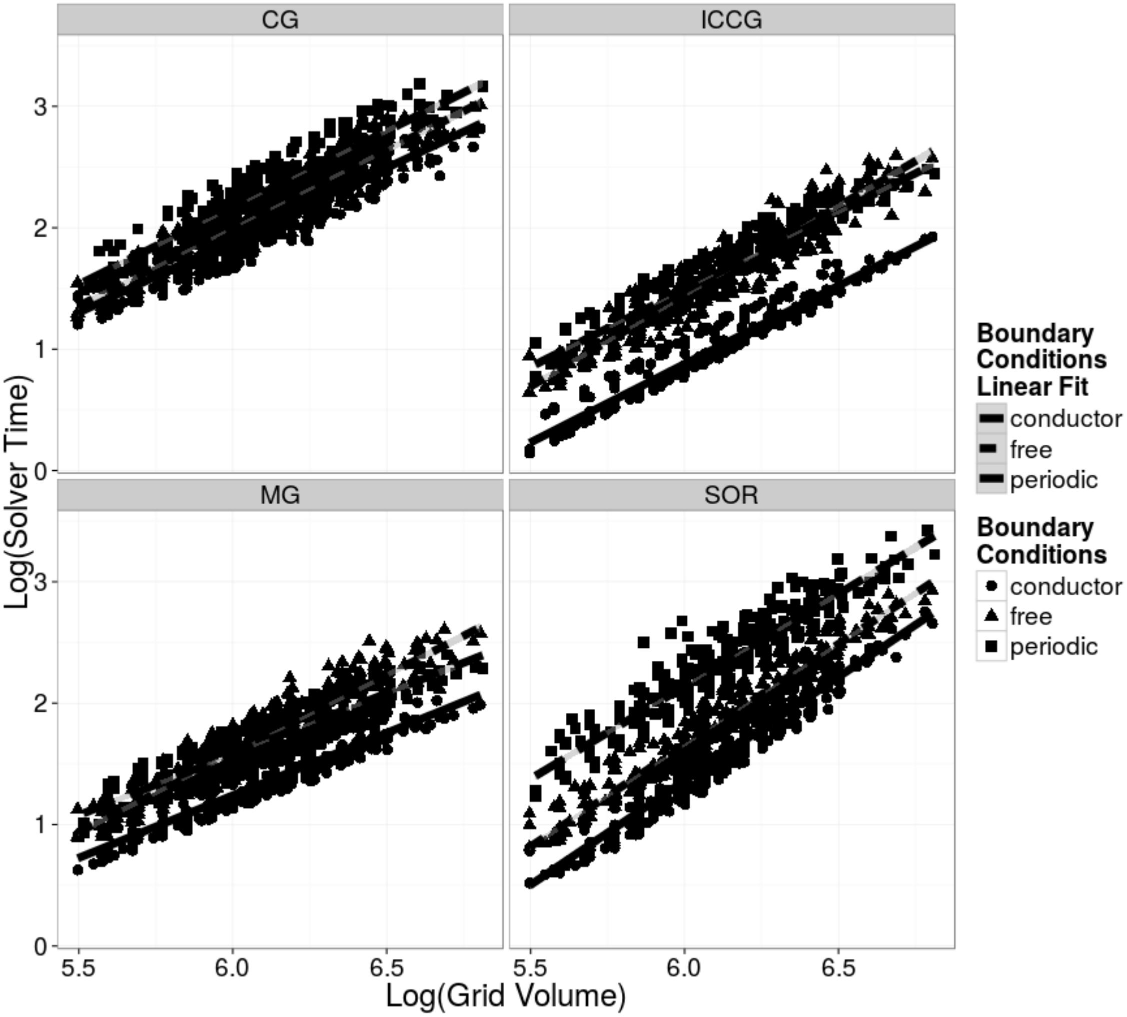

Figure 2.6. Solver Time Scaling: Log Solver Time Required vs Log Grid Volume

Log-log plots of the solver computation time versus total number of grid points (grid volume). The solver computation time excludes other time such as energy calculation and molecular surface generation.

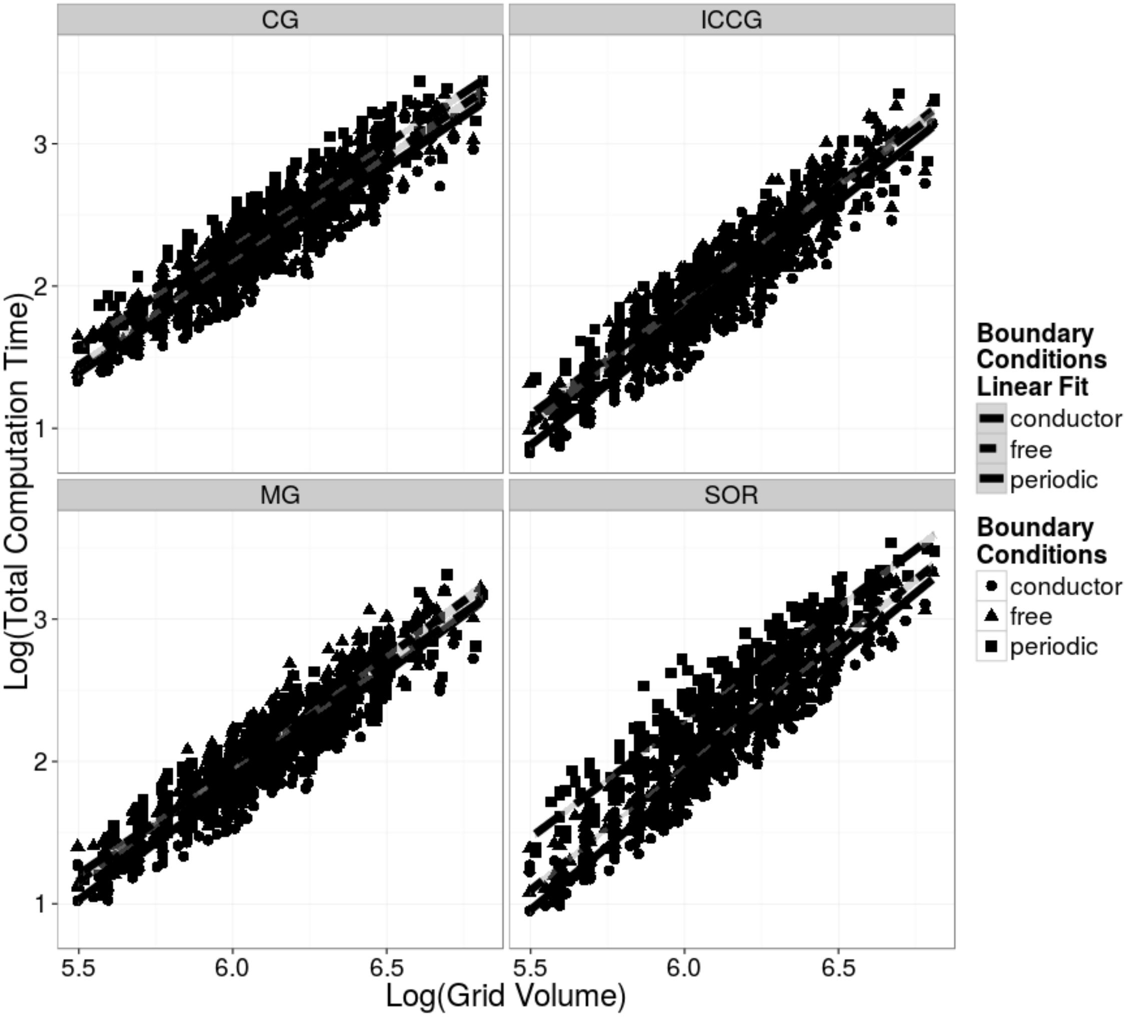

Figure 2.7. Total PB Time Scaling: Log Computation Time vs Log Grid Volume

Log-log plots of the total computation time required versus total number of grid points (grid volume). The total computation time includes all time needed for the PB calculations to finish normally, such as energy calculation and molecular surface generation.

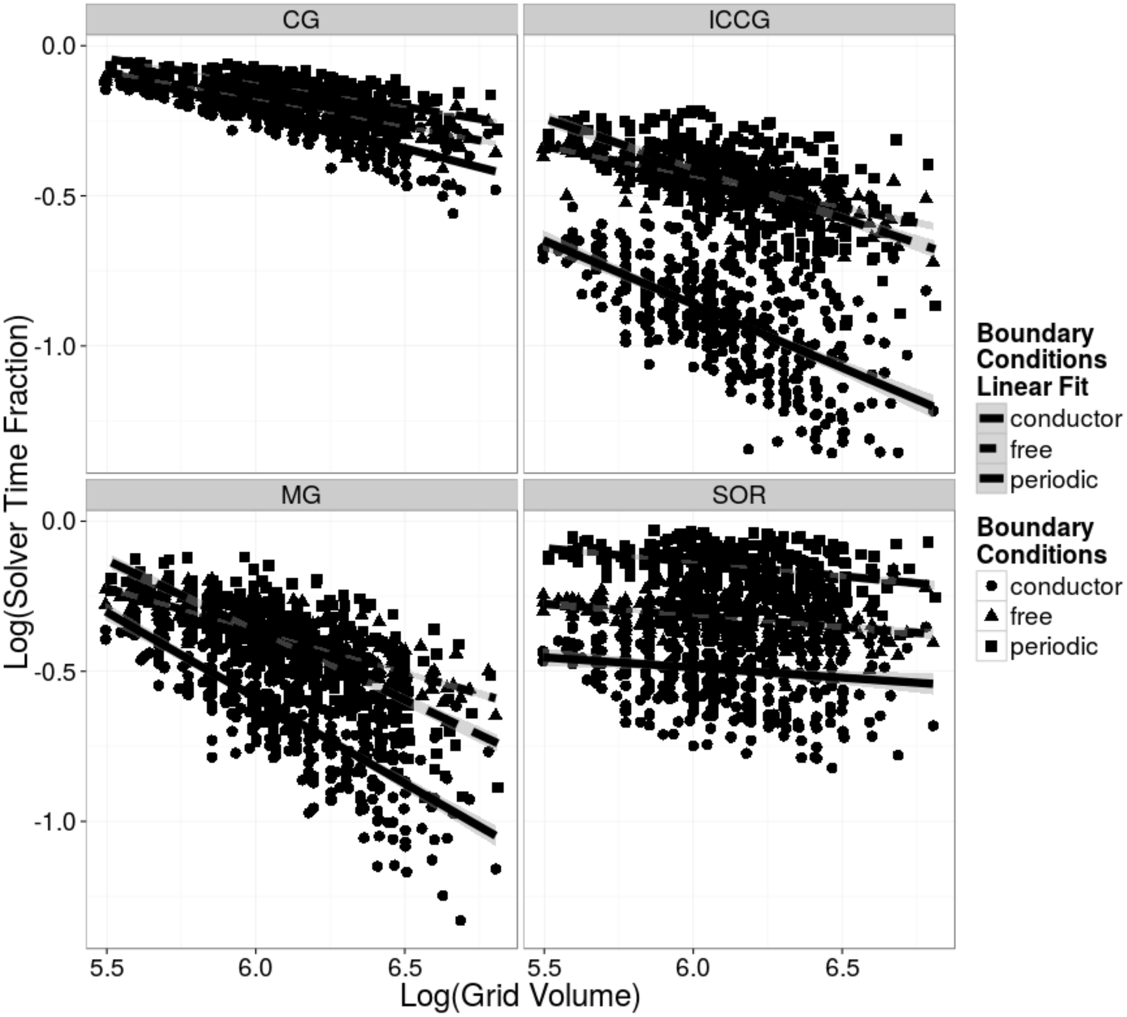

Figure 2.8. Relative Solver Time Scaling: Log Relative Solver Time vs Log Grid Volume

Log-log plots of the fractional solver computation time versus total number of grid points (grid volume). The fractional solver computation time is with respect to the total computation time.

Numerical Stability in Simulations of Charged Solutes

As was noted in the discussion of charged solutes in Methods, the addition of the linear term used to model ionic strength alleviates the need to pre-neutralize a PB solution systems as is necessary when solving the Poisson equation. This is demonstrated using a large test set of biomolecules with net charges of hundreds of electron charges under the physiological condition of 150mM. plots the unsigned relative deviations (with respect to the solute net charges) from the charge neutrality condition versus solute net charges. It is clear that the error in enforcing the charge neutrality condition is about ~10^{-5}, or roughly a factor of 10 larger than the convergence criterion, 1 ppm, used in the iteration.

Lastly, illustrates the numerical performance of all four periodic solvers with respect to the ionic strength ranging 25 to 1000 mM using a small charged molecule – dimethyl phosphate. Worth pointing out is that all solvers can achieve convergence and their electrostatic energies are all consistent with each other at all tested conditions. However the SOR performance degrades extremely rapidly when the ionic strength approaches zero. Performance degradation is not very apparent in the unconditioned CG, and the best behaving periodic solvers are ICCG and MG in this test case.

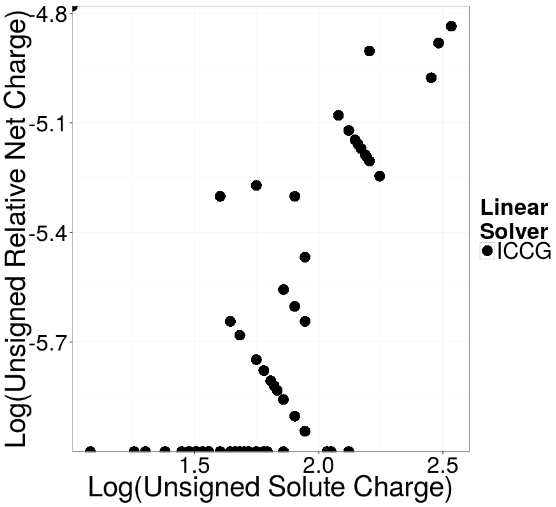

Figure 2.9. Charge Neutrality Achieved: Log Unsigned Relative Net Charge vs Log Absolute Net Solute Charge

Log-Log plot of the unsigned relative net charge of the combined solute and mobile ion charge distributions versus the net solute charge for various nucleic acid systems under periodic boundary conditions. The relative net charge is computed with respect to the absolute solute net charge.

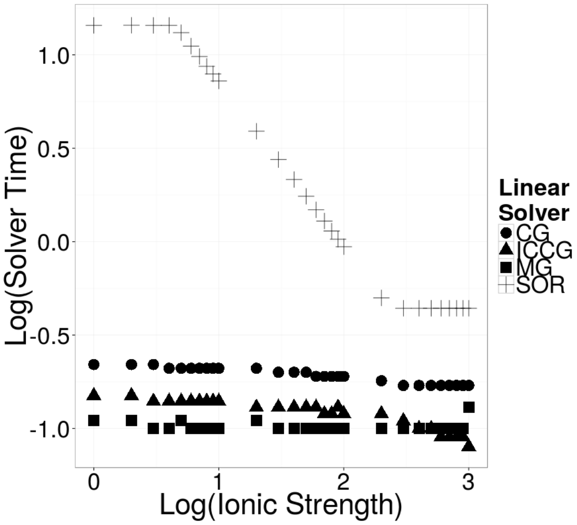

Figure 2.10. Dependence of Solver Performance upon Ionic Strength: Log Solver Time vs Log Ionic Strength

Log-log plot of the solver computation time versus the ionic strength. Note that the solver time for PSOR degrades rapidly, following a seemingly exponential increase in time as the ionic strength approaches zero.

Effect of Implicit Membrane in MMPBSA Analysis of P2Y12R

To demonstrate the effectiveness of the heterogeneous membrane/water model, the same MMPBSA calculation was run using both the membrane/water model and the homogeneous water model under both free and periodic boundary conditions. The correlations between the experimental ΔΔG's and each of the three different computational ΔΔG's were analyzed. In each case only the boundary condition and solvent setup (homogenous water or membrane/water) were changed. Here the membrane was modeled as a 40-Ångstrom slab of dielectric constant of 1.0. The dielectric constant of the protein region was set to 4.0 to accommodate the high net charges of the systems. All other parameters were left at default values, including the nonpolar solvent treatment[29] in the MMPBSAprotocol.

The results of the binding free energy calculation are shown in . Upon inspection, it is clear that the heterogeneous membrane/water model leads to a marked improvement in the correlation between computed and measured ΔΔG's. The homogenous water model yields poor correlations whether the free boundary or periodic boundary was used. Thus the different correlation is not the cause of the periodic boundary condition that must be used in the heterogeneous membrane/water model. Of course the deviations from experimental values are still quite large, as is the case for most MMPBSA calculations, at least with the single-trajectory approach that is often applied in the literature.

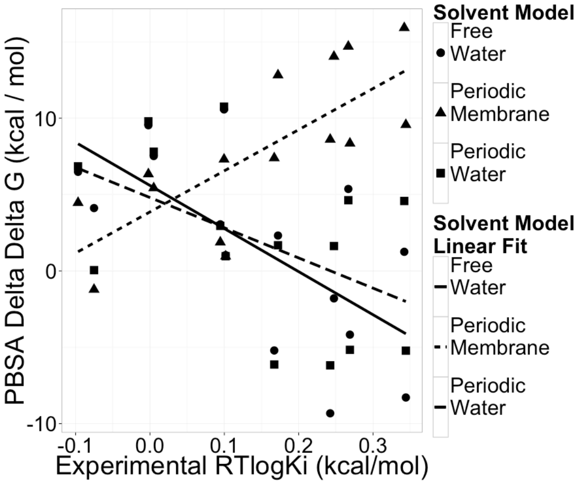

Figure 2.11. Effect of Implicit Solvent Modeling on Binding Affinity Calculation for P2Y12R

Correlation plots of computed and measured relative binding free energies (ΔΔG) for P2Y12R complexes.

Given that the focus of the current study are on the development of efficient numerical PB solvers for MMPBSA calculations; it is informative to look into the timing data of the MMPBSA calculation when existing and newly developed solvers are used. Here, the PBSA jobs were rerun for one snapshot of the ligand-protein complex (with periodic boundaries and implicit membrane model) using ICCG and CG solvers in addition to the MG solver used in the above analysis. The CPU times were 139.04, 55.12, and 25.00 for the CG, ICCG, and MG solvers, respectively. Given that typically hundreds of snapshots were processed for typical MMPBSA calculations, the efficiency gain of the new solver is noticeable, especially when the MG method is used.

Conclusions and Future Directions

The linear solvers currently available in the PBSA module of the Amber package, including SOR, modified ICCG, and geometric MG, were extended to include periodic boundary conditions. Accuracy testing, conducted using the previously developed and verified unconditioned CG solver indicated that all solvers are capable of achieving convergence and produce consistent electrostatic energies within the specified convergence criterion. Thorough efficiency testing was performed to investigate solution time scaling with respect to system sizes. As in the previous work by Wang and Luo [11], multi-grid and conditioned conjugate gradient exhibit superior performance when compared with SOR and unconditioned CG. Interestingly while ICCG was shown, on average, to be the most efficient algorithm when applied to small to medium sized systems, scaling projections seem to indicate that MG would eventually outperform ICCG when applied to sufficiently large systems.

The results of the solver timing analysis have shown that, for large systems, the fraction of time spent on the linear solver portion of the computation comprised successively smaller portions of the total time. This is an indication that future work should include optimization of the other computationally intensive tasks required by PBSA, such as computation of the molecular surface and assignment of virtual charges. Since many of these tasks are amenable to parallelization, it may be beneficial to focus upon parallel implementations. Directions for future work may include adaptation to MPI, or Open-MP frameworks or to development of code suitable for use with GPU that can greatly accelerate computational time for highly parallel tasks.

As mentioned in the introduction, a primary motivation for adaptation of current linear PB solvers as previously investigated by Wang and Luo[11] to include periodic boundary conditions is to facilitate the use of heterogeneous membrane/water models for MMPBSA calculations within the current PBSA framework [3, 28b, 30]. Previous work has already been published, providing the details for adapting the current level-set molecular surface [30] and formulation to allow inclusion of a planar membrane region and even inclusion of a cylindrical pore-like region in the case of channel or gating proteins [30]. However, application of such methods requires use of periodic boundary conditions in order to mitigate introduction of edge effects induced by the presence of non-uniform dielectric mapping along the grid boundary. Although the unconditioned CG method, developed previously, provides a means of performing such computations, its relatively poor convergence properties limit its practical applications. Adaptation of the accelerated MG and ICCG linear solvers, as detailed here, should allow for competitive applications of the new membrane/water model.

While the tested P2Y12R membrane system is by no means a fully representative sample, it nevertheless provides a good platform to demonstrate the utility of the heterogeneous membrane/water model for MMPBSA calculations. Potential avenues for further development includes the incorporation of implicit membrane in the classical solvent excluded molecular surface for the PB surface definition, a potential update of the solvent accessible volume / surface area based non-polar computation [29] for ligand binding that is within the membrane region, and implementation of the membrane/water model under the existing MM-PBSA protocol [31] for fully automatic applications in binding free energy computations to the end users. Apparently more detailed analysis on the influence of various computational setups and parameters is in the P2Y12R system also needed for thorough assessment of the new MMPBSA proptocol and is left as a future development.

Acknowledgements

This work is supported in part by NIH (GM093040 & GM079383).

References

1. (a) Davis, M. E.; McCammon, J. A., *Chemical Reviews* **1990,** *90* (3), 509-521; (b) Honig, B.; Sharp, K.; Yang, A. S., *J. Phys. Chem.* **1993,** *97* (6), 1101-1109; (c) Honig, B.; Nicholls, A., *Science* **1995,** *268* (5214), 1144-1149; (d) Beglov, D.; Roux, B., *Journal of Chemical Physics* **1996,** *104* (21), 8678-8689; (e) Cramer, C. J.; Truhlar, D. G., *Chemical Reviews* **1999,** *99* (8), 2161-2200; (f) Bashford, D.; Case, D. A., *Annual Review Of Physical Chemistry* **2000,** *51*, 129-152; (g) Baker, N. A., *Current Opinion in Structural Biology* **2005,** *15* (2), 137-143; (h) Chen, J. H.; Im, W. P.; Brooks, C. L., *Journal of the American Chemical Society* **2006,** *128* (11), 3728-3736; (i) Feig, M.; Chocholousova, J.; Tanizaki, S., *Theoretical Chemistry Accounts* **2006,** *116* (1-3), 194-205; (j) Koehl, P., *Current Opinion in Structural Biology* **2006,** *16* (2), 142-151; (k) Im, W.; Chen, J. H.; Brooks, C. L., *Peptide Solvation and H-Bonds* **2006,** *72*, 173-+; (l) Lu, B. Z.; Zhou, Y. C.; Holst, M. J.; McCammon, J. A., *Communications in Computational Physics* **2008,** *3* (5), 973-1009; (m) Onufriev, A., Chapter 7 - Implicit Solvent Models in Molecular Dynamics Simulations: A Brief Overview. In *Annual Reports in Computational Chemistry*, Ralph, A. W.; David, C. S., Eds. Elsevier: 2008; Vol. Volume 4, pp 125-137; (n) Wang, J.; Tan, C. H.; Tan, Y. H.; Lu, Q.; Luo, R., *Communications in Computational Physics* **2008,** *3* (5), 1010-1031; (o) Altman, M. D.; Bardhan, J. P.; White, J. K.; Tidor, B., *Journal of Computational Chemistry* **2009,** *30* (1), 132-153; (p) Cai, Q.; Wang, J.; Hsieh, M.-J.; Ye, X.; Luo, R., Chapter Six - Poisson–Boltzmann Implicit Solvation Models. In *Annual Reports in Computational Chemistry*, Ralph, A. W., Ed. Elsevier: 2012; Vol. Volume 8, pp 149-162; (q) Botello-Smith, W. M.; Cai, Q.; Luo, R., *Journal of Theoretical and Computational Chemistry* **2014,** *13* (03), 1440008.

2. (a) Klapper, I.; Hagstrom, R.; Fine, R.; Sharp, K.; Honig, B., *Proteins Structure Function and Genetics* **1986,** *1* (1), 47-59; (b) Davis, M. E.; McCammon, J. A., *J Comput Chem* **1989,** *10* (3), 386-391; (c) Nicholls, A.; Honig, B., *Journal of Computational Chemistry* **1991,** *12* (4), 435-445; (d) Luty, B. A.; Davis, M. E.; McCammon, J. A., *Journal of Computational Chemistry* **1992,** *13* (9), 1114-1118; (e) Holst, M.; Saied, F., *J Comput Chem* **1993,** *14* (1), 105-113; (f) Forsten, K. E.; Kozack, R. E.; Lauffenburger, D. A.; Subramaniam, S., *Journal of Physical Chemistry* **1994,** *98* (21), 5580-5586; (g) Bashford, D., *Lecture Notes in Computer Science* **1997,** *1343*, 233-240; (h) Im, W.; Beglov, D.; Roux, B., *Computer Physics Communications* **1998,** *111* (1-3), 59-75; (i) Rocchia, W.; Alexov, E.; Honig, B., *Journal of Physical Chemistry B* **2001,** *105* (28), 6507-6514; (j) Luo, R.; David, L.; Gilson, M. K., *Journal of Computational Chemistry* **2002,** *23* (13), 1244-1253; (k) Holst, M. J.; Saied, F., *Journal of Computational Chemistry* **1995,** *16* (3), 337-364.

3. Lu, Q.; Luo, R., *Journal of Chemical Physics* **2003,** *119* (21), 11035-11047.

4. (a) Meijerink, J. A.; Vandervorst, H. A., *Mathematics of Computation* **1977,** *31* (137), 148-162; (b) Gustafsson, I., *BIT Numerical Mathematics* **1978,** *18* (1), 142-156; (c) Eisenstat, S. C., *Siam Journal on Scientific and Statistical Computing* **1981,** *2* (1), 1-4; (d) Meijerink, J. A.; Vandervorst, H. A., *Journal of Computational Physics* **1981,** *44* (1), 134-155.

5. (a) Miertus, S.; Scrocco, E.; Tomasi, J., *Chemical Physics* **1981,** *55* (1), 117-129; (b) Hoshi, H.; Sakurai, M.; Inoue, Y.; Chujo, R., *Journal of Chemical Physics* **1987,** *87* (2), 1107-1115; (c) Zauhar, R. J.; Morgan, R. S., *Journal of Computational Chemistry* **1988,** *9* (2), 171-187; (d) Rashin, A. A., *Journal of Physical Chemistry* **1990,** *94* (5), 1725-1733; (e) Yoon, B. J.; Lenhoff, A. M., *Journal of Computational Chemistry* **1990,** *11* (9), 1080-1086; (f) Juffer, A. H.; Botta, E. F. F.; Vankeulen, B. A. M.; Vanderploeg, A.; Berendsen, H. J. C., *J. Comput. Phys.* **1991,** *97* (1), 144-171; (g) Zhou, H. X.,

Biophysical Journal **1993,** *65* (2), 955-963; (h) Bharadwaj, R.; Windemuth, A.; Sridharan, S.; Honig, B.; Nicholls, A., *Journal of Computational Chemistry* **1995,** *16* (7), 898-913; (i) Purisima, E. O.; Nilar, S. H., *Journal of Computational Chemistry* **1995,** *16* (6), 681-689; (j) Liang, J.; Subramaniam, S., *Biophysical Journal* **1997,** *73* (4), 1830-1841; (k) Vorobjev, Y. N.; Scheraga, H. A., *Journal of Computational Chemistry* **1997,** *18* (4), 569-583; (l) Totrov, M.; Abagyan, R., *Biopolymers* **2001,** *60* (2), 124-133; (m) Boschitsch, A. H.; Fenley, M. O.; Zhou, H. X., *Journal of Physical Chemistry B* **2002,** *106* (10), 2741-2754; (n) Lu, B. Z.; Cheng, X. L.; Huang, J. F.; McCammon, J. A., *Proceedings of the National Academy of Sciences of the United States of America* **2006,** *103* (51), 19314-19319.

6. (a) Cortis, C. M.; Friesner, R. A., *Journal of Computational Chemistry* **1997,** *18* (13), 1591-1608; (b) Holst, M.; Baker, N.; Wang, F., *Journal of Computational Chemistry* **2000,** *21* (15), 1319-1342; (c) Baker, N.; Holst, M.; Wang, F., *Journal of Computational Chemistry* **2000,** *21* (15), 1343-1352; (d) Shestakov, A. I.; Milovich, J. L.; Noy, A., *Journal of Colloid and Interface Science* **2002,** *247* (1), 62-79; (e) Chen, L.; Holst, M. J.; Xu, J. C., *Siam Journal on Numerical Analysis* **2007,** *45*, 2298-2320; (f) Xie, D.; Zhou, S., *BIT Numerical Mathematics* **2007,** *47* (4), 853-871.

7. Wang, J.; Cai, Q.; Xiang, Y.; Luo, R., *Journal of Chemical Theory and Computation* **2012,** *8* (8), 2741-2751.

8. (a) Wang, J.; Cai, Q.; Li, Z.-L.; Zhao, H.-K.; Luo, R., *Chemical Physics Letters* **2009,** *468* (4-6), 112-118; (b) Wang, C.; Wang, J.; Cai, Q.; Li, Z.; Zhao, H.-K.; Luo, R., *Computational and Theoretical Chemistry* **2013,** *1024* (0), 34-44.

9. (a) Callenberg, K. M., Choudhary, O. P., de Forest, G. L., Gohara, D. W., Baker, N. A., and Grabe, M. , *PLoS One* **2010,** *5* (9), 1-11; (b) Feig, M., Implicit Membrane Models for Membrane Protein Simulation. In *Methods in Molecular Biology*, Kukol, A., Ed. Humana Press: Totowa, NJ. USA, Vol. 443, pp 181-196; (c) Im, W., Feigh, M., and Brooks III, C. L. , *Biophysical Journal* **2003,** *85*, 2900-18; (d) Spassov, V. Z., Yan, L., and Szalma, S. , *J Phys. Chem. B* **2002,** *106* (8726-38); (e) Ulmschneider, J. P.; Ulmschneider, M. B., *J Chem Theory Comput* **2007,** *3* (6), 2335-2346; (f) Ulmschneider, M. B., Ulschneider, J. P., Sansom, M. S., and Di Nola, A. , *Biophysical Journal* **2007,** *92*, 2338-49; (g) Ulmschneider, M. B.; Sansom, M. S. P.; Di Nola, A., *Proteins* **2005,** *59* (2), 252-265.

10. Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P., *Numerical recipes : the art of scientific computing*. Cambridge University Press: Cambridge [Cambridgeshire]; New York, 1986.

11. Wang, J.; Luo, R., *J Comput Chem* **2010,** *31* (8), 1689-1698.

12. (a) Holst, M.; Kozack, R. E.; Saied, F.; Subramaniam, S., *J Biomol Struct Dyn* **1994,** *11* (6), 1437-1445; (b) Holst, M.; Kozack, R. E.; Saied, F.; Subramaniam, S., *Proteins* **1994,** *18* (3), 231-245.

13. Alcouffe, R. E.; Brandt, A.; Dendy, J. E.; Painter, J. W., *Siam Journal on Scientific and Statistical Computing* **1981,** *2* (4), 430-454.

14. (a) Sharp, K. A., *Current Opinion in Structural Biology* **1994,** *4* (2), 234-239; (b) Roux, B.; Simonson, T., *Biophysical Chemistry* **1999,** *78* (1-2), 1-20; (c) Im, W.; Chen, J. H.; Brooks, C. L., Peptide and protein folding and conformational equilibria: Theoretical treatment of electrostatics and hydrogen bonding with implicit solvent models. In *Peptide Solvation and H-Bonds*, Elsevier Academic Press Inc: San Diego, 2006; Vol. 72, pp 173-198; (d) Gilson, M. K., *Current Opinion in Structural Biology* **1995,** *5* (2), 216-223.

15. Hummer, G.; Pratt, L. R.; Garcia, A. E., *J Phys Chem-Us* **1996,** *100* (4), 1206-1215.

16. Sharp, K. A.; Honig, B., *J. Phys. Chem.* **1990,** *94* (19), 7684-7692.

17. Hackbusch, W., *Multi-grid methods and applications*. Springer-Verlag Berlin: 1985; Vol. 4.

18. (a) Koutis, I.; Miller, G. L.; Tolliver, D., *Comput Vis Image Und* **2011,** *115* (12), 1638-1646; (b) Koutis, I.; Miller, G. L.; Peng, R., *Ann Ieee Symp Found* **2011**, 590-598; (c) Blelloch, G. E.; Gupta, A.; Koutis, I.; Miller, G. L.; Peng, R.; Tangwongsan, K., *Spaa 11: Proceedings of the Twenty-Third Annual Symposium on Parallelism in Algorithms and Architectures* **2011**, 13-22.

19. Livne, O. E.; Brandt, A., *Siam J Sci Comput* **2012,** *34* (4), B499-B522.

20. Zhang, K.; Zhang, J.; Gao, Z.-G.; Zhang, D.; Zhu, L.; Han, G. W.; Moss, S. M.; Paoletta, S.; Kiselev, E.; Lu, W., *Nature* **2014,** *509* (7498), 115-118.

21. Madej, B. D.; Dickson, C. J.; Skjevik, A. A.; Betz, R. M.; Walker, R. C.; Teigen, K., *Abstr Pap Am Chem S* **2014,** *248*.

22. Biro, E.; Akkerman, J. W. N.; Hoek, F. J.; Gorter, G.; Pronk, L. M.; Sturk, A.; Nieuwland, R., *J Thromb Haemost* **2005,** *3* (12), 2754-2763.

23. Jo, S.; Kim, T.; Im, W., *Plos One* **2007,** *2* (9), e880.

24. Webb, B.; Sali, A., *Current protocols in bioinformatics* **2014**, 5.6. 1-5.6. 32.

25. Case, D.; Babin, V.; Berryman, J.; Betz, R.; Cai, Q.; Cerutti, D.; Cheatham Iii, T.; Darden, T.; Duke, R.; Gohlke, H., **2014**.

26. Cai, Q.; Hsieh, M.-J.; Wang, J.; Luo, R., *Journal of Chemical Theory and Computation* **2010,** *6* (1), 203-211.

27. Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A., *J Am Chem Soc* **1996,** *118* (9), 2309-2309.

28. (a) Ye, X.; Cai, Q.; Yang, W.; Luo, R., *Biophysical Journal* **2009,** *97* (2), 554-562; (b) Botello-Smith, W. M.; Liu, X.; Cai, Q.; Li, Z.; Zhao, H.; Luo, R., *Chemical Physics Letters* **2012**.

29. Tan, C.; Tan, Y.-H.; Luo, R., *Journal of Physical Chemistry B* **2007,** *111* (42), 12263-12274.

30. Ye, X.; Wang, J.; Luo, R., *Journal of Chemical Theory and Computation* **2010,** *6* (4), 1157-1169.

31. Miller, B. R.; McGee, T. D.; Swails, J. M.; Homeyer, N.; Gohlke, H.; Roitberg, A. E., *J Chem Theory Comput* **2012,** *8* (9), 3314-3321.

Chapter 3: Applications of MMPBSA to Membrane Proteins II: Design and Analysis of Protocol for Protein-Ligand Binding Free Energy Calculation

Introduction

Membrane proteins provide a range of important functions, such as being cell receptors, signaling proteins, transmembrane channels, and more. Their roles as receptors and channels make them particularly relevant as candidates for drug targets. Study of membrane proteins, however, is more complicated than globular proteins. Particularly, the presence of the membrane complicates structural studies, both experimentally and computationally. The presence of the membrane makes it more difficult to employ experimental techniques such as NMR and X-ray crystallography on membrane proteins. For instance, the signal from the membrane must be disentangled from that of the protein when using NMR and membrane proteins are notoriously difficult to crystallize. For computational studies, modeling of the membrane becomes an important consideration.

Inclusion of solvent effects in computational studies of biological systems is quite important. It is relatively common knowledge that solvent-solute interactions provide the primary driving for producing and maintaining properly folded structures of proteins.[1] Inclusion of solvent into a computational model or simulation can generally be classified into one of two different avenues, explicit and implicit solvation. In explicit solvation each atom and / or molecule of the solvent is modeled individually. While this is generally agreed to be the most accurate method, one is often not interested in the properties of the solvent itself, but rather the interest is in the behavior it induces upon the solute. Unfortunately, accurately capturing statistically meaningful characteristics requires either sampling from ensembles of trajectories, or, over very long simulation times in cases where an appeal to ergodicity is valid. Implicit solvents provide an attractive alternative wherein the effects of the solvent are modeled as a continuum.[1b, 1c, 2] While the fine grain details of individual solvent-solute particle interactions are lost, the relevant ensemble / statistically averaged effects may still be captured when a properly parameterized model is used. In addition, since the individual solvent molecules are no longer modeled directly, there will be far fewer particles to simulate.

In the case of membrane proteins, the membrane must also be included when modeling solvation effects.[3] In general, the molecules that make up a lipid membrane are much more complex than water or other small organic solvents. This increases both the computational expense of their inclusion as well as the amount of simulation time / replicas needed to accurately capture averaged / ensemble interactions. Thus, there has been much effort into development and testing of implicit membrane solvent models. However, just as with explicit solvation, inclusion of an implicit membrane complicates implementation of implicit solvent models.

One of the key features to consider in implicit solvent models is the modeling of electrostatic interactions. This is most typically accomplished by employing the Poisson-Boltzmann Equation (PBE).[4] The solvent is modeled as a region (or regions in the case of membrane models) of high dielectric constant, and the solute is modeled as a region of low dielectric constant under the Poisson Equation. The effect of charged ions in the solvent region may also be included by modeling it as a mobile charge density that obeys a Boltzmann distribution. In its full form, this yields a three-dimensional non-linear elliptical partial differential equation:

$$\nabla \cdot \varepsilon \nabla \phi = -4\pi\rho_0 - 4\pi \sum_i e z_i c_i \exp\left(-e z_i \phi / k_B T\right) \tag{3.1}$$

where ε is the dielectric constant distribution, φ is the electrostatic potential distribution, ρ is the charge density of the solute (usually modeled as a set of discrete point charges), c_i is the concentration of the ith solvent ion species in bulk, *e* is the absolute charge of an electron, z_i is the valence for the ith ion, k_B is Boltzmann's constant, T is the temperature, and ∇ is the spatial gradient operator.

In cases where ion concentration is relatively low (a few hundred millimolar or less) the second term on the right hand side, which corresponds to the mobile ion density distribution, may be reduced to a linear response term yielding:

$$\nabla \cdot \varepsilon \nabla \phi = -4\pi\rho_0 + \varepsilon_v \kappa^2 \phi \tag{3.2}$$

where $\kappa^2 = \frac{8\pi e^2 I}{\varepsilon_v k_B T}$. Here *v* denotes the solvent, I represents the ionic strength of the solution, and is computed as $I = z^2 c$. This is often referred to as the linear PBE.

Even in its simplified linear form, solution of the PBE is a non-trivial endeavor. Due to its complexity, there is no general closed form solution; and, with the exception of very simplified geometries, a numerical solution must be sought. The finite-difference method,[3a, 4f, 4o, 4p, 5] finite-element method,[6] and boundary-element method[7] are the commonly used numerical solution methods. The PBE-based solvent models have widely biological applications. For example, they have been applied to prediction of pKa values for ionizable groups in biomolecules,[8] solvation free energies,[9] binding free energies,[10] and protein folding and design.[11]

One such exception is the case of a single point charge in a spherical dielectric cavity interacting with an outside charge distribution. Indeed, this case is a major part of the basis for the Generalized Born Equation, which is a quite popular and widely used approximation to full numerical solution of the PBE. While the underlying assumptions make such methods less accurate than numerical solution of the full PBE, there is a significant speed gain, which can be particularly important for applications which required repeated calculations of electrostatic energy, such as implicit solvent based molecular dynamics simulations.[2k, 11d, 12]

In order to apply PBE or GB frameworks to implicit membrane solvent models, an additional solvent dielectric region must be added. The appropriate dielectric constant of the membrane region is generally thought to be quite low[13] relative to the solvent dielectric which is typically set to be between 60 (mimicking spc water models) and 80 (typical for tip3p models). Early membrane permeation experiments,[13d] indicate that the dielectric constant for membranes varies as a function of depth. As discussed in early development of GB implicit models,[13a, 13c, 14] the interior of the membrane which contains mostly hydrophobic hydrocarbon chains, can be modeled as a region of very low dielectric constant, typically in the range of about 1 to 4 with a value of 2 seeming to be most common. Adding this region means that the dielectric constant exterior to the solute is no longer homogenous. Moreover, in the case of numerical PBE solvers, this discontinuity complicates boundary conditions since it extends to the edge of the computational grid. Additionally, there is some evidence to support that more than one additional region may be needed to properly model the dielectric profile of a lipid membrane implicitly. The upper layer of lipid membranes, which consists of charged and polar head groups, exhibits very high dielectric constant. Various dielectric constant profiles were explored during the creation of GB implicit membrane models[13c] and it was demonstrated that a simple two dielectric constant model can reproduce proper electrostatic free energies relatively well by modeling the membrane as a slab like region with uniform dielectric constant of about 2. More accurate results could be attained using models with 3 or more layers, however, beyond 3 layers there did not appear to be any significant improvement in results.[13c]

One of the major applications of implicit solvent free energy calculation is in the prediction of protein ligand binding affinities. The AMBER15 suite currently provides the capability of performing such computations for globular proteins in implicit water using either the GBSA or PBSA formalism via the MM_PBSA module.[15] Implementation of implicit membrane under a PBE formalism is not new and is available under other PB packages, such as APBS[16] and Delphi.[17] With the implementation of an implicit membrane model into the PBSA framework,[4t] the implicit membrane model can be more readily interfaced with the existing MM_PBSA framework.[15]

A complete implementation of an implicit membrane under the PBSA framework requires implementation of appropriate membrane to protein non-polar interaction free energy terms. While development of these terms is still underway, it is not expected to impact binding energy calculations for protein-ligand systems in which the binding pocket is sequestered away from the membrane in the protein interior. Thus, such systems make good candidates for testing of the current electrostatic free energy calculations provided by the current implementation of the implicit membrane model within the PBSA framework.[1c, 4g, 4t]

While three-dimensional structures for globular proteins seem quite abundant, such data is less prevalent for membrane proteins. To conduct this study it was necessary to find a protein for which experimental binding affinities and structures of the associated protein ligand complexes are both available. Recently, a study on the (P2Y12) receptor was released,[18] which provides both crystal structures of the receptor bound to three different ligands, as well as experimental measurements of dissociation constants for the wild type and several select mutants. Thus, this study provides a good testing set for assessing the utility of the PBSA and MM_PBSA frameworks for computation of binding free energy for membrane proteins.

Methods

Preparation of P2Y12R Complex Models from Crystal Structures

Three separate crystal structures of P2Y12R, two complexed with agonist ligands[18] (2-methylthio-ATP (2MeSATP) and 2-methylthio-ADP (2MeSADP)) and one with the antagonist drug ligand (AZD), were downloaded from the protein databank. As was noted in the corresponding literature,[18] each of the crystal structures contained several small residue sequences for which no structure could be resolved.

The program Modeller[19] was used to generate homology models for residue sequences with missing structure data. These homology models were then merged into the crystal structures as follows. The molecular visualization program VMD[20] was used to locally align the generated homology model with the corresponding regions of the crystal structure using the Mulit-SEQ[21] module. The substructures of the locally aligned homology model were then exported as individual coordinate files and merged into the crystal structure files.

The ligands of each separate P2Y12R complex structure (2MeSADP, 2MeSATP, and AZD) were extracted to individual structure files for parameterization. The ANTECHAMBER[22] module of the AMBER molecular modeling and simulation suite[23] was used to generate force field parameters compatible with the AMBERff14SB force field used during subsequent simulation protocol.

The first step toward obtaining force field parameters for ligand structures using ANTECHAMBER is parameterization of the ligand atomic point charges. ANTECHAMBER provides an internal quantum mechanics facility for calculation of ligand atomic point charge parameters. Due to the complexity of the 2MeSATP and 2MeSADP systems, however, it was necessary to employ external quantum mechanics software to obtain the needed parameters. The SPARTAN 14[24] quantum mechanics, molecular modeling, and visualization package was used to perform the needed computations for each ligand. For the 2MeSATP and 2MeSADP ligands, the phosphorous and linking oxygen atoms of the phosphate groups were frozen prior to geometry optimization. The geometry of the AZD ligand was converged without restraints. Geometry optimization was performed using the *ab-initio* Hartree Fock model with a 6-31G* basis set. This basis set is consistent with the AM1-BCC[25] model employed by ANTECHAMBER, which employs a semi-empirical AM1 basis set along with post calculation correction terms to approximate the point charges provided by the HF 6-31G* level of theory. The electrostatic point charges and optimized structures were then exported in the mol2 format for subsequent generation of force field parameter libraries via the PARMCHK module of ANTECHAMBER.

Two important considerations that can greatly impact the magnitude of binding energies are the protonation states of side chains in the receptor (more pronounced nearby the binding site) and the parameterization and protonation state of the ligand. In situations where there are a large number of like charged ionizable side chains occupying the same geometric space within a protein, the electrostatic environment may differ significantly from that of bulk water. It is reasonable, then, to expect that this may affect the protonation state of the side chains. Structural visualization shows that the two most buried lysines in the binding pocket, LYS266 and LYS66; and one buried aspartic acid within the membrane bound region, ASP294. These were modeled in their respective neutral states. Other exposed and less buried bases/acids that interact with the ligands/water are modeled in their default charged states.

The parameterization of the 2MeSATP and 2MeSADP agonist ligand mimics was also examined. Both ATP and ADP are known to complex Mg^{2+} and are generally not found in fully ionized forms in solution at the neutral pH if not complexed with a cation. When Mg^{2+} is present, $MgATP^{2-}$ will typically be the most prevalent species at neutral pH.[26] When not complexed with a suitable cation, ATP is known to act as a weak base with pKa's of 3.93 and 7.03 for its fourth and third deprotonation processes, respectively.[26] In either case, it is clear that the ATP ligand complex, whether existing in a doubly protonated or Mg complexed state, should have a net charge of -2 unlike the -4 net charge of an unbound and fully deprotonated ATP ion as is often depicted in introductory textbooks. While explicit inclusion of cations would provide the most accurate representation, doing so would complicate implementation under the MMPBSA framework. Instead, quantum mechanics optimization for the ATP molecule was run with two additional protons bonded to two of the three oxygens present on the terminal phosphate group. The net electrostatic charge of the two protons was then computed. Similarly, at neutral pH the monoprotonated form of ADP is most prevalent in solution, this procedure was repeated to model ADP with a -2 net charge.

Preparation of Lipid Membrane System Models

The P2Y12 receptor is found embedded within platelet outer membranes. Proper modeling of its solvent environment should also include a membrane model as well. During simulation, this was accomplished with an explicit all-atom model, while during post processing binding affinity calculations, an implicit continuum model was employed. Construction of the explicit all-atom membrane model was accomplished using the CHARMM-membrane builder web server.[27] Construction was guided by composition data taken from the 2005 paper by Biro et. al.,[28] wherein phospholipid and cholesterol composition fractions were reported. The membrane was constructed to have POPC, POPS, and POPE in 3:2:3 ratios with a 2:5 cholesterol to lipid ratio. Sphingomylin lipids, while notably present, could not be included, since their force field parameters were not yet available. Nevertheless, their absence was not expected to have a significant impact given the relatively short timescales being simulated. The aqueous phase of the P2Y12 membrane protein system was modeled using an explicit all-atom approach with the TIP3P model along with sufficient potassium and chloride ions to mimic a roughly 150 milimolar KCl concentration. This assembly setup was repeated for each of the three protein-ligand complexes studied.

The constructed membrane systems were then loaded into the Leap program of the AMBER15 for generation of simulation force field topology and coordinate files. At this time, the corresponding D294N mutant structures were generated. This was accomplished by renaming the backbone atoms of the appropriate residue within the wild type membrane-protein structure file followed by deletion of the side-chain atoms. Modeling of the side-chain atoms for the mutant residue was accomplished automatically upon loading of the structure into the Leap program. Counterions were then added as necessary to neutralize any remaining net charge and the topology and initial coordinate files were generated.

MD Simulation Protocol

Prior to beginning molecular dynamics heating and density equilibrations, each system was first minimized using 500 steps of steepest descent followed by 500 steps of conjugate gradient optimization. All loops containing coordinates taken directly from crystal structure data were held fixed. Loops containing coordinates generated from homology modeling were left unrestrained, along with all solvent molecules including membrane lipids, waters, and ions. Minimization was performed using the MPI parallelized SANDER code from the AMBER15 suite. After minimization, systems were heated and density equilibrated. Heating was performed in two phases. In the first phase, systems were brought up to 100 K over 2500 time steps (5000 ps), under the NTV condition using a Langevin thermostat with a collision frequency of 1.0 per ps. This was followed by heating from 100 K to 303 K over 100 ps under the NTP condition using anisotropic pressure scaling with the Berendsen barostat with a pressure relaxation constant of 2.0 ps and target pressure of 1.0 atm. The thermostat settings were left identical to the first heating run. In both cases, a cutoff radius of 10.0 Angstroms was used when computing non-bonded interactions and the restraint setup matched that of the minimization phase. Heating simulations were run using the MPI parallelized sander program from AMBER15.

After initial heating was completed, it was necessary to allow the membrane density to equilibrate prior to beginning production runs. Density equilibration was performed over 10 identical 500 ps NTP simulations. This setup is necessary when utilizing AMBER's GPU accelerated pmemd program which does not allow for frequent updating of box size. The membrane density equilibration steps utilized the same barostat and thermostat settings as the second heating step, except that a constant temperature of 303 K was employed. As with heating and minimization, portions of the protein substructure containing coordinates from the crystal structure were held fixed. After equilibration, all restraints were removed and a 10 ns simulation was performed for the single trajectory method, while a total of 100 ns of simulation was performed for the multi-trajectory method. In order to take full advantage of the GPU accelerated code, a Monte-Carlo thermostat was employed instead of the Berendsen thermostat used during previous simulation phases. This thermostat has the advantage of not requiring the virial to be calculated at each time step, a process that cannot be parallelized well.

Binding Free Energy Calculations

Binding free energies were computed using the SANDER/PBSA module of AMBER15. Each of the trajectories was post processed using the ANTE-MMPBSA module in order to generate trajectories containing coordinates of the receptor-ligand complex, receptor only, and ligand only with all solvent, membrane, and ion atoms removed. The first 100 frames from each of these trajectories were then processed using SANDER/PBSA to compute molecular mechanics potential energies and solvation free energies. The energy data for all frames of the complex, receptor only, and ligand only were extracted and were subject to statistical analysis. The binding free energy is then computed as the difference between the complex free energy and the sum of receptor and ligand free energies as:

$$\triangle G_{binding} = G_{complex} - G_{receptor} - G_{ligand} \tag{3.3}$$

Here, G represents the average free energy over the entire associated trajectory, computed as:

$$G = U + G_{PB} + G_{NP} \tag{3.4}$$

Where U is the potential energy of the system computed according to the bonded and van der Waals terms of the simulation forcefield, G_{PB} is the energy computed according to the Poisson-Boltzmann equation, and G_{NP} is the free energy contribution from the non polar solvation free energy term, computed as a function of the solute surface / volume.

These binding free energies were then compared against the experimental results.[18] The experimental dissociation constants, K_d, were converted to appropriate binding free energies using the formula:

$$\Delta G = +RTln(K_d) \tag{3.5}$$

where R is the gas constant, T is temperature, and K_d is the experimentally determined dissociation coefficient.

In order to assess the utility of the various solvation model parameters, these computations were repeated to test various relevant computational parameters, including selection of solute dielectric constant, membrane versus globular solvent model, boundary conditions, membrane dielectric constant, and non-polar solvent model. To simplify the discussion of various key parameters influencing the performance of the MMPBSA binding free energies, a pre-screening using 10 frames was conducted by exhausting all combinations of parameters discussed below. Thus the scanning of each parameter was conducted by setting the other parameters at their optimum values.

Implicit Polar Solvation Model

Neither the implicit membrane solvation model nor the periodic boundary condition options available under PBSA in AMBER have been previously tested for use in binding affinity applications. Two sets of tests were run to test the effect of periodic versus free boundary under water only implicit solvent, and implicit membrane versus water only solvent under periodic boundary conditions. For the latter, a dielectric constant of 1.0 was used for the membrane. A protein dielectric constant of 4.0 was used in both cases.

Implicit Non-polar Solvation Model

While electrostatic interactions play a major role in PBE-based implicit solvent models, various non-polar interactions, such as cavity surface tension and dispersion must also be accounted for. The PBSA module of AMBER currently provides two options for computing non-polar solvation energy terms. The first method is to simply use a linear function of the solvent accessible surface area / volume.[29] The second, more sophisticated method employs surface integration to decompose the non-polar contribution into separate cavity term and dispersion terms.[30] To test the relative effectiveness of these two terms, two sets of computations were run using both options (inp=1 or 2) respectively with all other parameters set at optimal values.

Protein Dielectric Constant

At the neutral pH, the P2Y12R and all associated ligands except AZJ are non-neutral in solution. In cases of neutral receptor and ligand systems, protein dielectric is typically assigned to a relatively low value, such as 1 or 2. Charged systems may require a higher dielectric constant to be assigned to the solute in order to compensate the lack of conformational sampling in the typical MMPBSA calculations. Choice of protein dielectric constant has a significant impact upon the electrostatic energy, since a high value will tend to dampen electrostatic interactions. This effect is expected to be particularly noticeable for large proteins with a high net charge. To test the effect of protein dielectric constant upon binding prediction efficacy, a series of calculations was performed with protein dielectric constants of 1.0, 2.0, 4.0, 6.0, 8.0, 12.0, 16.0, and 20.0 with all other parameters held at optimal values using the solvent excluded molecular surface description.

Membrane Dielectric Constant

Modeling of an implicit membrane under the PBE framework requires implementing an additional solvent region. Since the interior of a lipid membrane is well known to be highly hydrophobic and non-polar, it is generally agreed that the interior dielectric constant can be assumed to be quite low. Proper modeling of the dielectric profile of implicit membranes has been a subject of much previous work; and varying multiple dielectric schemes have been suggested such as the two-, three-, or seven-dielectric models proposed in Feig et. al.[3d] Such schemes may indeed be warranted when one examines the results of FEP or PMF studies.[3d, 13b] In this study, only the simpler and relatively common two-dielectric model was considered.[31] Multi-dielectric models will be implemented and examined in a future work. The effectiveness of various membrane dielectric constants for the purpose of estimating binding free energies was tested for dielectric constants of 1, 2, 4, and 7 with all other parameters set to optimal values.

Multi-trajectory Method

Thus far, the methodology employed revolves around generating a single all-atom trajectory for each system. The corresponding complex, receptor, and ligand trajectories were produced by extracting their substructure coordinates from the single all-atom trajectory. Inherent to this single-trajectory method is an underlying assumption that the conformational distributions of the ligand and receptor do not vary significantly between the bound and unbound states. In the case of relatively rigid ligands this may hold true for the ligand structure. Similarly the distribution of energetically favorable conformations of the receptor may or may not change upon binding depending upon the nature of the binding site.

In the case of the P2Y12 receptor, it is known that the binding domain contains a “hinge-like” structure that caps the binding pocket. While this structure was well defined for the crystal structures of the normal Me2SADP ligand, the structure was not well resolved in the case of the AZJ ligand. This evidence is sufficient to hint that the conformational distribution of the bound and unbound receptor may vary between bound and unbound states for some ligands. Because of this uncertainty, a slightly more advanced methodology, involving simulations of unbound receptor and ligand systems, was also developed and evaluated.

For receptor, the molecular dynamics protocol was identical to that used for the complex except that the ligand was removed from the crystal structure. Simulation of the ligand structures requires an aqueous environment. Addition of solvent molecules and ions proceeded in a manner consistent with the receptor and complex system setups. All simulation settings were identical except that 1) isotropic pressure scaling was used instead of the semi-isotropic scaling which was needed previously to stabilize membrane density, and 2) there was no need for the membrane density equilibration steps.

In order to provide proper results, it is necessary that the receptor sample an appropriate portion of its unbound conformational space during simulation. Since there is no crystal structure for the unbound conformation it is important to ensure that the receptor has had sufficient simulation time to allow the needed conformational changes to occur. Computation of backbone RMSD provides a useful metric to assess whether or not this has occurred. Since the ligand binding pocket is the region that is most closely interacting with the ligand, the RMSD of residues lining this region was monitored. Similarly, it is possible to visually gauge conformation changes and flexibility from an image of overlaid frames taken from the trajectory at regular intervals near the end of the simulation.

Assessment Metrics

The accuracy of the computed binding affinities was assessed by calculating the RMSD of the calculated versus the experimental values. The Pearson correlation coefficient, Spearman rank correlation coefficient, the slope and associated p-value of the linear regression between calculated and measured values were also analyzed. These were computed for both the absolute binding free energies (ΔG) and relative binding free energies (ΔΔG) between different receptors (wild type and mutant D294N) and different ligands (Me2SATP, Me2SADP, and AZJ). These metrics were then used to assess the effects of computation parameters including choice of non-polar solvation models, protein dielectric constants, and membrane dielectric constants.

Additional Computational Details

In each PBSA calculation a grid spacing of 0.5 Å was used with a grid to solute dimension ratio (fillratio) of 1.5. The geometric multigrid solver option was employed with a convergence threshold of 1.0 x 10^{-3} and electrostatic focusing turned off.[4f, 4o, 4p, 32] Choice of implicit solvation model and boundary conditions were investigated in a previous paper,[32] where it was shown that the implicit membrane model with periodic boundary conditions yielded improved results. Thus, the implicit membrane model (memopt1) and periodic boundary conditions (bcopt10) were employed for all PBSA computations on structures containing one of the P2Y12R structures; while implicit water (memopt0) and periodic boundary conditions (bcopt10) were employed for computations involving only the unbound small organic ligand structures.

The solvation system physical constants were set up as follows. The membrane is modeled as a solid slab of 40 Å. The water relative dielectric constant (epsout) is fixed at 80.0. The protein dielectric constant (epsin) of 4.0 was found to agree with experiment the best if it was not changed. The membrane dielectric constant (epsmem) of 1.0 was found to be the best value for MMPBSA calculations if it was not changed. The modeling of non-polar solvation is also important as discussed above and is set to be option 2 (inp2) if it was not changed. The water phase ionic strength (istrng) is set to be 150 mM. Weighted harmonic averaging was employed to assign dielectric constants for boundary grid edges to reduce grid dependence (smoothopt1) and charge singularity was not removed (bcopt5).[4i, 4s] Charges and radii were assigned using the same parameters as the simulation topology files.[23]

Results

In our first publication documenting our development of a MMPBSA method for membrane proteins,[32] we demonstrated the effect of implicit membrane model versus homogeneous implicit water model for the P2Y12R systems presented here. Our analysis shows that the use of the implicit membrane model markedly improves the correlation between experimental and computed binding free energies over the homogenous implicit water model.[32] This clearly shows the potential utility of the implicit membrane solvent model for use in future MMPBSA studies of membrane proteins. In this study, we intend to develop an automated procedure to set up and execute required MMPBSA calculations; and also, we seek to further investigate the effects of various relevant parameters, particularly non-polar solvation model, protein dielectric constant, and membrane dielectric constant, upon the accuracy and efficacy of MMPBSA binding free energy calculations.

Non-polar Solvent Models

There are currently two models available for computing non-polar solvation energy terms: the classical method that models the non-polar solvation free energy as a single linear term in proportion to the solvent accessible surface area, and the more advanced method that decomposes the nonpolar solvation free energy into separate (cavity and dispersion) terms. Since there were only two options, the dataset lends itself well to visualization and the results for both the raw binding energies and binding energy differences were plotted in Figure 3.2 and Figure 3.1, respectively. Visual inspection of the regression lines, however, does not provide a great deal of immediately useful insight, thus metrics for the correlations were computed and compiled in Table 3.1. Although the changes are somewhat less pronounced than for variations in protein or membrane dielectric constant to be discussed below, it is quite clear that the new method (inp2) yields improved results in all areas except for the Spearman rank R correlation. Nevertheless, these results provide a good indication that inp2 yields improved utility for binding energy computations.

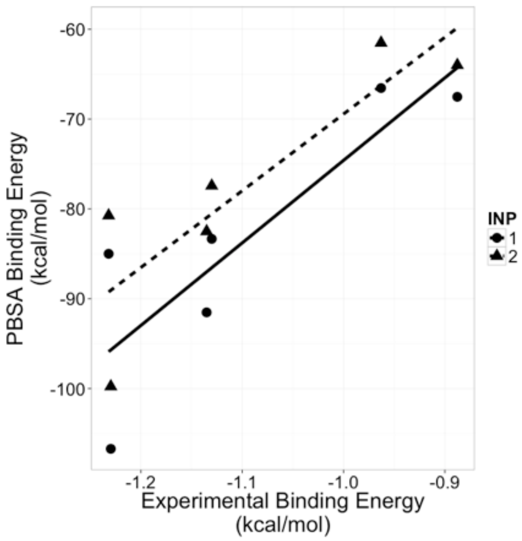

Figure 3.1: Binding energy (ΔG) correlation for P2Y12R using optimized parameters

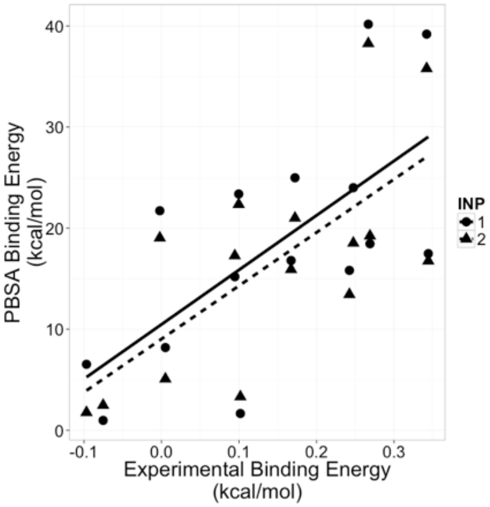

Figure 3.2: Binding energy difference (ΔΔG) correlation for P2Y12R using optimized parameters

Table 3.1: Effect of Non-Polar solvation term model

	ΔG					ΔΔG				
INP	RMSD	slope	p-value	R	Rank R	RMSD	slope	p-value	R	Rank R
1	83.481	91.891	0.029	0.736	0.771	21.26	53.93	0.007	0.67	0.62
2	77.594	85.436	0.025	0.754	0.771	19.52	52.71	0.004	0.69	0.57

Table 3.1 also shows that the RMSD between experimental and computationally values was lowered for both absolute and relative in free energies by using a cavity dispersion decomposition method (inp2). With the classical method (inp1), the RMSD's were 79.14 and 18.89 kcal/mol, respectively, while inp2 yielded 73.22 and 17.15 kcal/mol, respectively. The same trend holds for the p-value of the correlation; inp1 yielded p-values of .036 and .014 while inp2 yielded p-values of .033 and .010. Similarly, inp1 yielded R values of .71 and .62 while inp2 yielded R values of .72 and .64. While these differences are relatively small, particularly given the relatively limited test set size, testing nevertheless consistently shows that the advanced cavity/dispersion method, inp2, yielded better results, consistent with our finding in globular protein ligand binding calculation by the MMPBSA method (manuscript in preparation).

Protein Dielectric Constant Comparison

The most notable impact of solute dielectric constant selection is upon the RMSD, shown in Table 3.2 which decreases monotonically with increasing solute dielectric constant. While this would suggest choosing as large of a dielectric constant as possible, the correlation between calculated and experimental results becomes very poor as becomes evident upon examination of the correlation coefficients and the p-value of the slope for linear regressions.

Examination of linear regression statistics, particularly the p-value of the slope coefficient, shown in Table 3.2 sheds more light on selection of an optimal solute dielectric constant parameter. Here, slope, like RMSD appears to decrease monotonically with increasing solute dielectric constant. In this case, however, a slope of exactly 1 is the most desirable value, since that would correspond to the case where computed binding free energies, ΔG, and relative binding free energy, ΔΔG, would exactly match experimental values. Examination of the results reveals that there is, apparently, a systematic overestimation in binding free energies in most cases as is very common in most MMPBSA studies. However, care should be taken in using the slope too liberally as a scoring metric. It is important to also consider the probability of attaining the same slope if the data were actually random. This is given by the p-value. On inspection, it appears that a value of 4 for the protein dielectric yields optimal results. However, it is somewhat dangerous to use p-value alone as a scoring function. Pearson and Spearman correlation coefficients were also examined. For all metrics, it is again clear that a dielectric constant of 4 yields the strongest correlation, which is consistent with the observed trend in the p-value. In summary a protein dielectric constant of 4 is optimal when using MMPBSA for ranking or scoring of the tested membrane protein-ligand binding system.

Table 3.2: Effect of protein dielectric constant

	ΔG					ΔΔG				
EPSIN	RMSD	slope	p-value	R	Rank R	RMSD	slope	p-value	R	Rank R
1	89.365	214.062	0.038	0.699	0.886	51.09	136.15	0.014	0.62	0.58
2	84.209	138.452	0.027	0.746	0.886	31.90	88.23	0.005	0.68	0.55
4	77.594	85.436	0.025	0.754	0.771	19.52	52.71	0.004	0.69	0.57
6	74.096	62.738	0.030	0.733	0.771	14.50	37.11	0.007	0.66	0.49
8	71.904	50.064	0.035	0.712	0.600	11.70	28.77	0.012	0.63	0.50
12	69.326	35.602	0.051	0.654	0.600	8.65	19.08	0.039	0.54	0.51
16	67.829	27.724	0.068	0.605	0.543	6.98	14.40	0.075	0.47	0.49
20	66.880	22.705	0.093	0.546	0.543	6.00	11.23	0.137	0.40	0.38

Membrane Dielectric Constant Comparison

According to previous computational studies[13b-d] the lipid membranes exhibits a dielectric profile that varies as a function of distance from the membrane center. This value is typically quite low near the interior where membrane composition is primarily long hydrophobic hydrocarbon chains. Closer to the interface, in the region linking the hydrophilic tails to the hydrophilic head groups, the dielectric constant rises to a moderate value around 7.0 or so. Lastly, in the region composed of hydrophilic head groups, the dielectric constant is quite high, perhaps exceeding even that of the bulk water environment.

Currently, the implicit membrane model implemented in PBSA allows for only a single membrane region. While this may be extended with relative ease in the future, this study focuses on the single-dielectric protocol. In doing so, it was reasonable to examine choices ranging from 1 to 7 as this was consistent with the range of dielectric constants suggested for the dielectric profile of the membrane interior.

From examination of the data shown in Table 3.3 it is evident that the correlation between experimental and computation results degrades quickly as the membrane dielectric constant is increased. As in the protein dielectric scanning, the RMSD decreases as membrane dielectric increased. However, the loss of correlation, as measured by the p-value and R values, is much more dramatic than that in the scanning of protein dielectric. This result is somewhat surprising, given that the membrane region is not in direct contact with the ligand or the binding pocket.

Table 3.3: Effect of membrane dielectric constant

	ΔG					ΔΔG				
EPSMEM	RMSD	slope	p-value	R	Rank R	RMSD	slope	p-value	R	Rank R
1	77.594	85.436	0.025	0.754	0.771	19.52	52.71	0.004	0.69	0.57
2	73.220	73.411	0.033	0.720	0.771	17.15	42.60	0.010	0.64	0.50
4	68.976	60.670	0.048	0.664	0.771	14.74	31.34	0.035	0.55	0.47
7	65.884	51.218	0.068	0.607	0.771	13.00	22.77	0.093	0.45	0.36

It is widely known that long-range electrostatic interactions play important roles in protein binding interactions. Our data here further demonstrates that this remains to be the case for membrane proteins. Our analysis shows that a relatively low membrane dielectric should be used when employing MMPBSA as a ranking or scoring metric. While there seems to be a competing trend between the accuracy of magnitude of computed energies and the strength of ranking correlation, our results indicate that a low membrane dielectric constant is best since the correlation apparently degrades rapidly when membrane dielectric is increased. Furthermore, a low membrane dielectric is consistent with previous literature studies which investigated dielectric profiles of lipid membranes, wherein the membrane interior was shown to exhibit a very low dielectric.[13b]

Multi-trajectory Method

The multi-trajectory method, unlike the single-trajectory method, requires a much longer simulation period in order for the computed free energies to converge. However, the exact simulation time required is apparently system dependent. As can be seen in Figure 3.3, it is clear that the conformational free energies are still not converged by the end of the 100 ns production run. Unlike the single-trajectory method, where exact cancelation of various bonded energy terms is guaranteed, the multi-trajectory method only yields convergence of such terms after sufficient statistical averaging has occurred, even in cases where no significant change in protein or ligand conformation occurs upon binding.

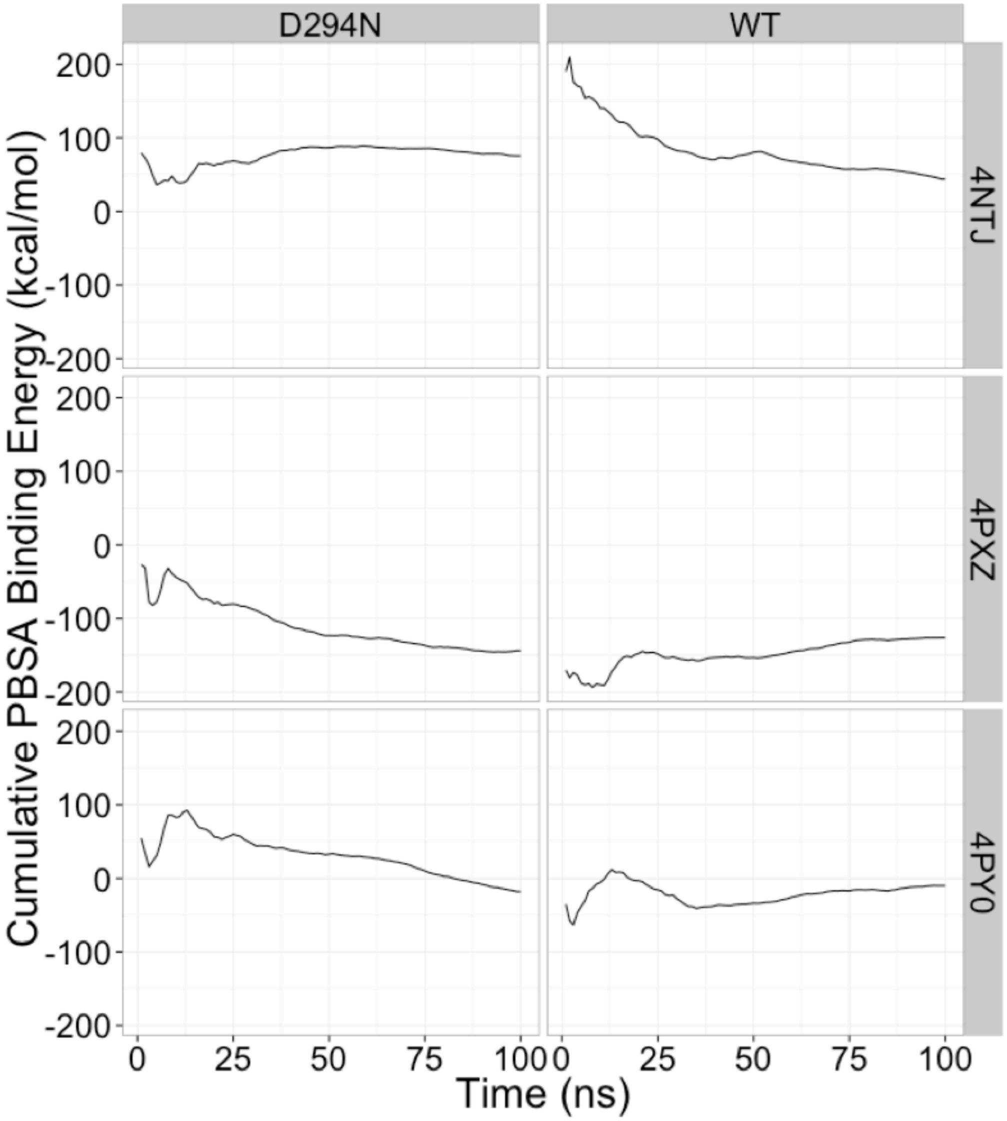

Figure 3.3: Cumulative PBSA binding free energies

The data shows that additional sampling is required for the tested systems, probably due to ongoing conformational relaxation throughout the 100ns simulated. This hypothesis is motivated by discussion in the article where the original crystal structures were reported.[18] It was noted that there was a noticeable conformational change in the one of the helices surrounding the binding pocket. The helix is observed to be straight in the unbound conformation. Upon bindin with the natural agonist ligands, the helix bends inward, toward the bound ligand. When binding with the antagonist drug ligand, however, the helix is forced outward instead.[18]

To investigate this phenomenon, and determine whether or not the suggested conformational changes had indeed been completed, multi-frame overlay renderings of binding pocket residues were generated to allow visual inspection of the difference in conformational flexibility of the binding pocket and neighboring residues for the ligand-protein complex and the unbound receptor, as shown in Figure 3.4 (2MeSADP structures) and Figure 3.5 (AZD structures). While the binding pocket has not opened for the unbound receptor, as in the antagonist drug complex, it is clear that the snapshots for the unbound receptor show much greater conformational flexibility than the 2MeSADP bound complex. As expected, the difference is most pronounced for the hinge-like capping loop of the binding pocket and the helical side loop reported in the literature.

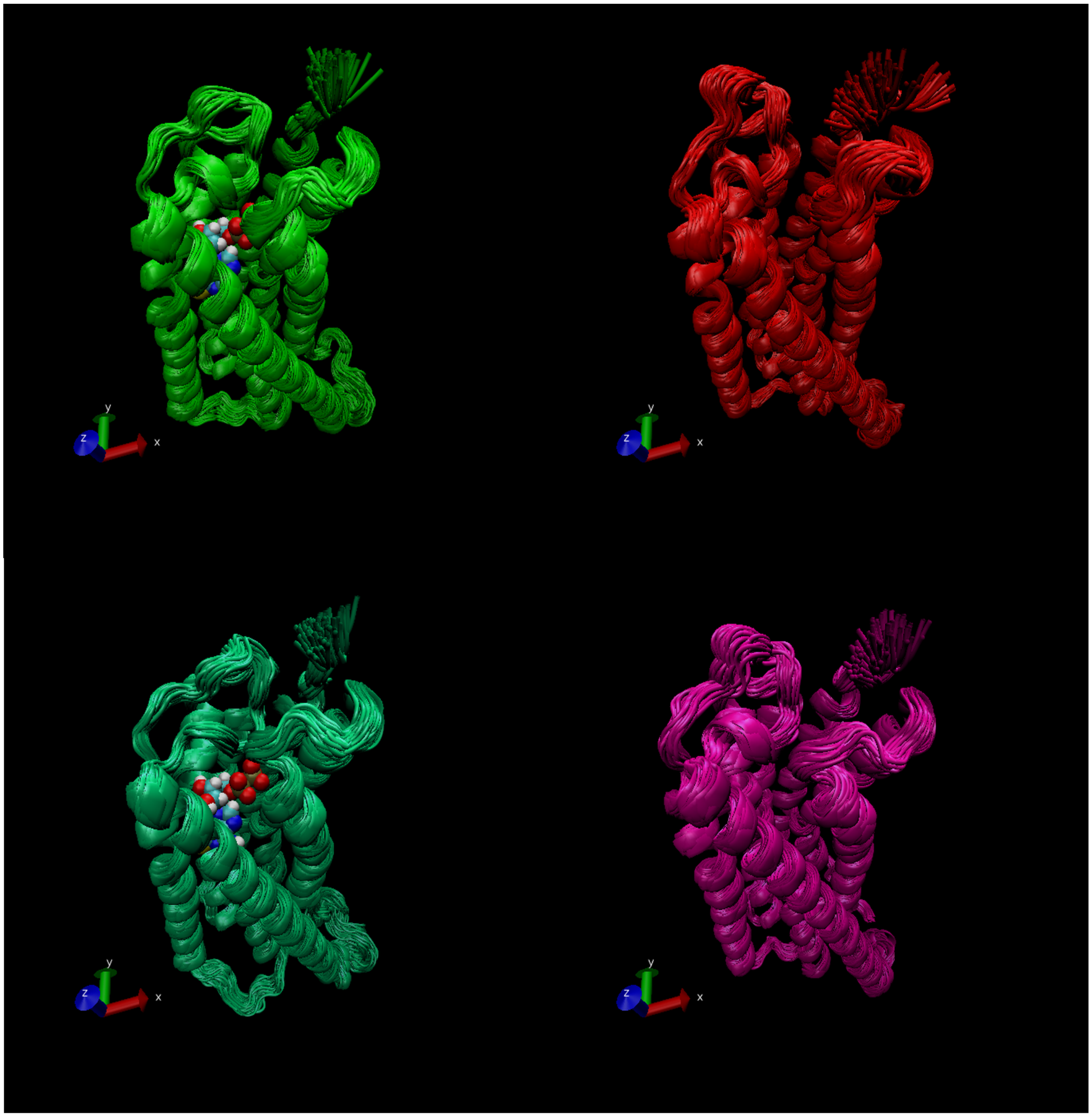

Figure 3.4: Comparison of multi-frame overlays for P2Y12R – 4PXZ

Showing the first frame of each .1 ns for the last 10 ns of the simulation. Left: Complexes of 4PXZ wild type (top) and mD294N mutant (bottom) with 2MeSATP agonist ligand. Right: Receptors with ligand removed for wild type (top) and mD294N mutant (bottom).

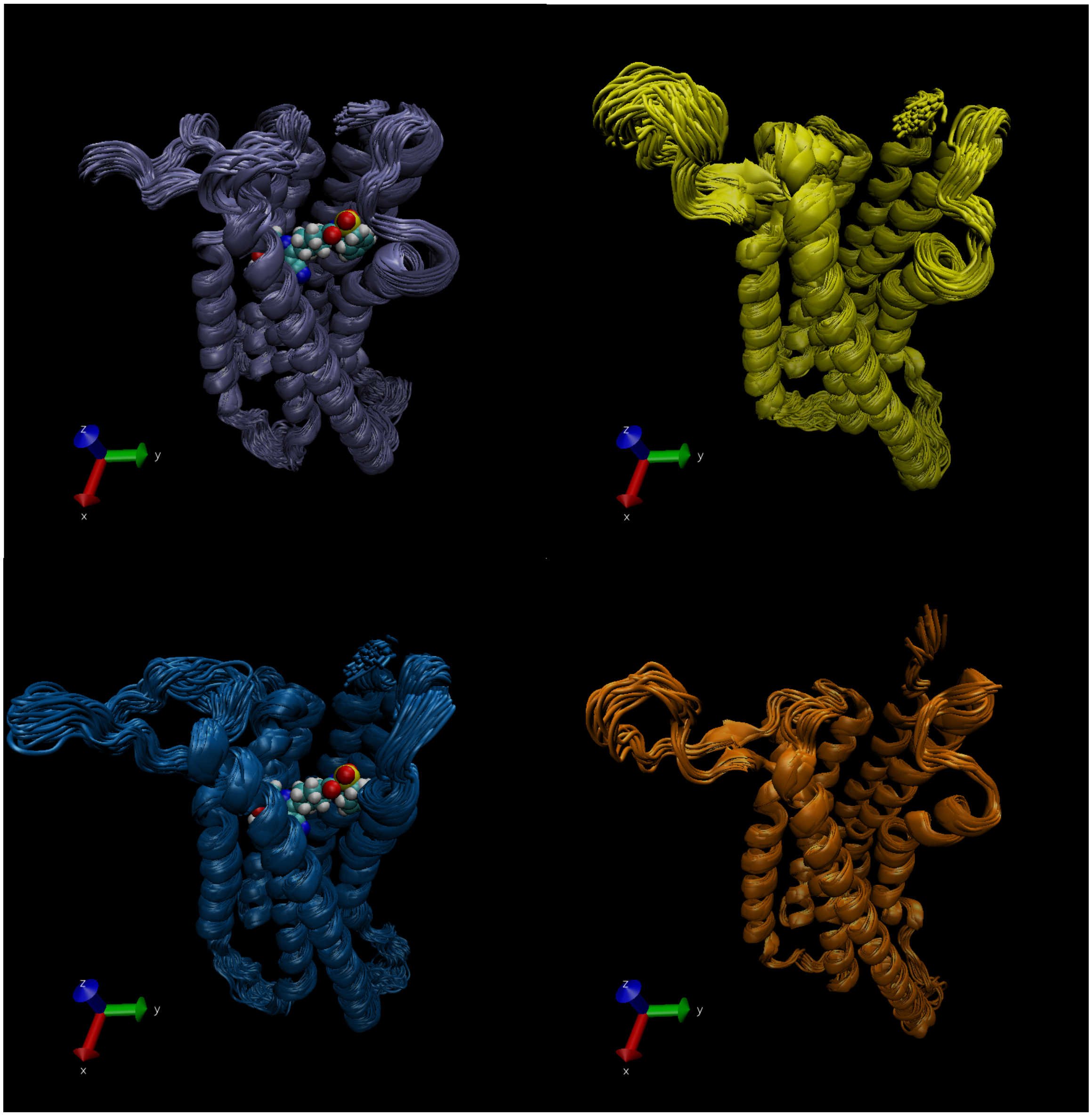

Figure 3.5: Comparison of multi-frame overlays for P2Y12R – 4NTJ

Showing the first frame of each .1 ns for the last 10 ns of the simulation. Left: Complexes of 4NTJ wild type (top) and mD294N mutant (bottom) with AZJ antagonist drug ligand. Right: Receptors with ligand removed for wild type (top) and mD294N mutant (bottom) .

It is apparent that the helix in question (as seen on the right front of the proteins in Figure 3.4 and Figure 3.5) had not yet returned to a straightened conformation. In the case of the wild type 4PXZ structure, the helix is less bent in the bound structure, indicating that the conformational change has not completed. In the other receptor structures, there was little to no change observed. Finally, an examination of the structural RMSD relative to the averaged structure over the last 2 ns of simulation, as shown in Figure 3.6 confirms that the structures had not yet reached a full equilibrium state. This is particularly pronounced for the antagonist drug complex and receptor structures. While this is somewhat encouraging, it is not known whether the required conformational changes occur on timescales readily attainable under the computational architectures employed in this study. Nevertheless, the intention of our analysis here was to show that it is important to study both energetic and structural dependent over simulation time to study the convergence of conformational free energies before MMPBSA can be applied in the multiple-trajectory set up as implemented.

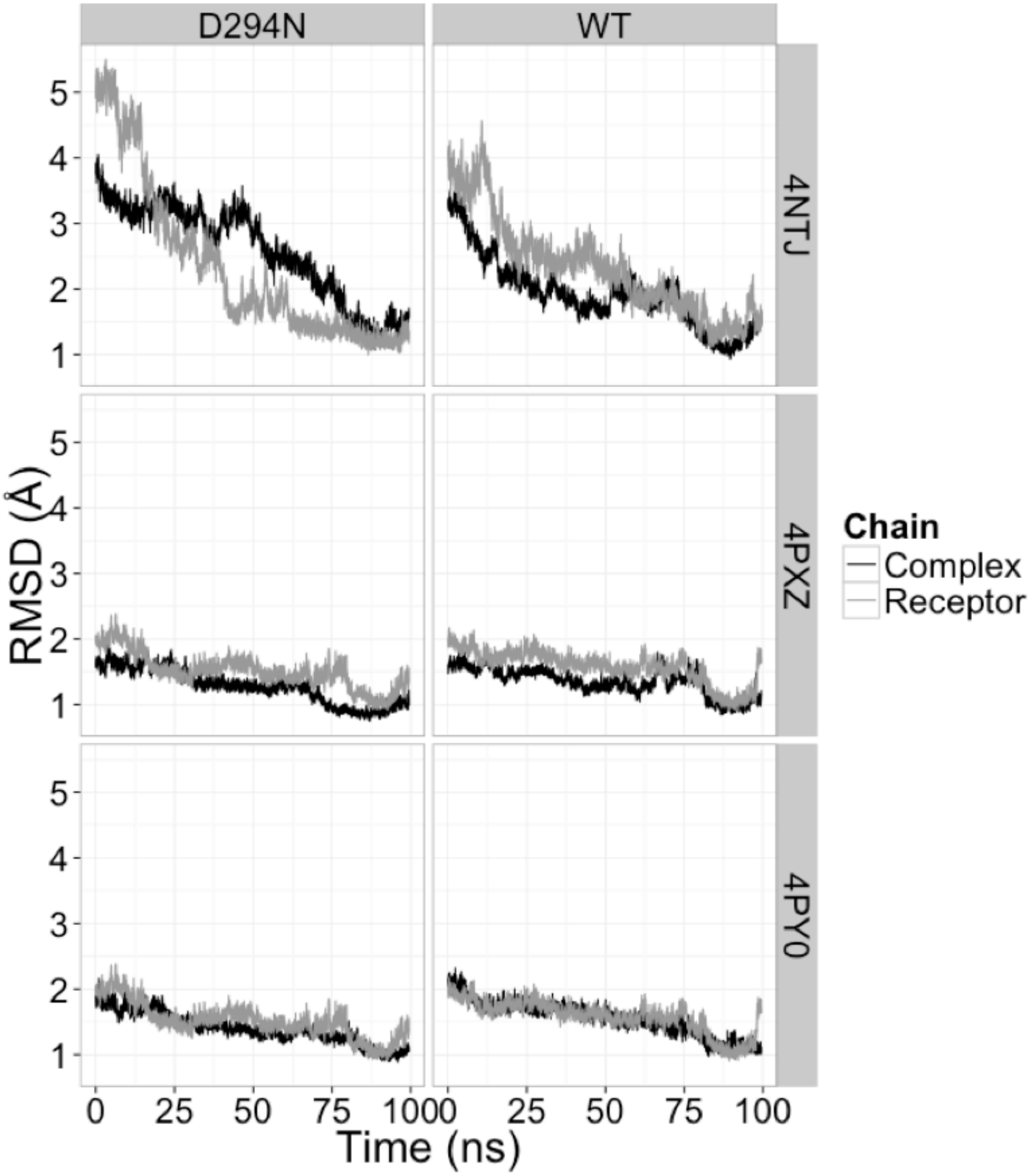

Figure 3.6: Structural RMSD plots for P2Y12R production simulations.

Conclusions

We have developed an automatic data processing protocol for membrane protein-ligand binding free energy estimation. Given the automated procedure, a detailed further investigation of parameters influencing the accuracy of MMPBSA results was conducted, in compliment to our previous study that use of an implicit membrane model can indeed yield improved correlation between experimental and computed binding energy differences. The automated procedure will facilitated future studies of membrane protein ligand binding systems. The reported optimization procedure also illustrates the various details that should be considered when selecting parameters whilst applying the MMPBSA method in studies of membrane protein-ligand binding systems.

Testing of the non-polar solvent term model indicates that the surface integration method (inp2) yields improved results all around as compared with the simpler linear model (inp1). These results provide a clear-cut indication that inp2 is preferred when performing binding free energy calculations for membrane protein-ligand systems.

Investigation of protein and membrane dielectric constants leads to less clear-cut results. In both cases, the deviation between calculated and experimental binding energies decreases as the dielectric constant increases. At high dielectric constants, however, the correlation also becomes extremely poor. In the case of the protein dielectric constant, a value of 4.0 to 6.0 is apparently an optimal choice with respect to the strength of the correlation achieved. Testing of membrane dielectric parameter did not reveal the same pattern. In this case correlation seems to deteriorate as dielectric constant is increased while the RMSD improves. The rate of degradation in correlation does appear to be somewhat more pronounced for changes in membrane dielectric than for changes in protein dielectric, particularly for membrane dielectric constants above 2.

This coincides nicely with previous literature investigations into membrane dielectric profiles, which indicate that the dielectric constant near the interior of lipid membranes is quite low.[13b] Currently, the implicit membrane modeling facilities in PBSA allow for a one-dielectric membrane model, although additional dielectrics can be implemented and will be made available in the future.

References

1. (a) Perutz, M. F., *Science* **1978,** *201* (4362), 1187-1191; (b) Davis, M. E.; Mccammon, J. A., *Chem Rev* **1990,** *90* (3), 509-521; (c) Honig, B.; Nicholls, A., *Science* **1995,** *268* (5214), 1144-1149.
2. (a) Honig, B.; Sharp, K.; Yang, A. S., *J. Phys. Chem.* **1993,** *97* (6), 1101-1109; (b) Beglov, D.; Roux, B., *Journal of Chemical Physics* **1996,** *104* (21), 8678-8689; (c) Cramer, C. J.; Truhlar, D. G., *Chemical Reviews* **1999,** *99* (8), 2161-2200; (d) Bashford, D.; Case, D. A., *Annual Review Of Physical Chemistry* **2000,** *51*, 129-152; (e) Baker, N. A., *Curr. Opin. Struct. Biol.* **2005,** *15* (2), 137-143; (f) Chen, J. H.; Im, W. P.; Brooks, C. L., *Journal of the American Chemical Society* **2006,** *128* (11), 3728-3736; (g) Feig, M.; Chocholousova, J.; Tanizaki, S., *Theoretical Chemistry Accounts* **2006,** *116* (1-3), 194-205; (h) Koehl, P., *Curr. Opin. Struct. Biol.* **2006,** *16* (2), 142-151; (i) Im, W.; Chen, J. H.; Brooks, C. L., *Peptide Solvation and H-Bonds* **2006,** *72*, 173-+; (j) Lu, B. Z.; Zhou, Y. C.; Holst, M. J.; McCammon, J. A., *Communications in Computational Physics* **2008,** *3* (5), 973-1009; (k) Wang, J.; Tan, C. H.; Tan, Y. H.; Lu, Q.; Luo, R., *Communications in Computational Physics* **2008,** *3* (5), 1010-1031; (l) Altman, M. D.; Bardhan, J. P.; White, J. K.; Tidor, B., *Journal of Computational Chemistry* **2009,** *30* (1), 132-153; (m) Cai, Q.; Wang, J.; Hsieh, M.-J.; Ye, X.; Luo, R., Chapter Six - Poisson–Boltzmann Implicit Solvation Models. In *Annual Reports in Computational Chemistry*, Ralph, A. W., Ed. Elsevier: 2012; Vol. Volume 8, pp 149-162; (n) Xiao, L.; Wang, C.; Luo, R., *Journal of Theoretical and Computational Chemistry* **2014,** *13* (03), 1430001; (o) Botello-Smith, W. M.; Cai, Q.; Luo, R., *Journal of Theoretical and Computational Chemistry* **2014,** *13* (03), 1440008.
3. (a) Forsten, K. E.; Kozack, R. E.; Lauffenburger, D. A.; Subramaniam, S., *J. Phys. Chem.* **1994,** *98* (21), 5580-5586; (b) Spassov, V. Z., Yan, L., and Szalma, S. , *J Phys. Chem. B* **2002,** *106* (8726-38); (c) Im, W., Feigh, M., and Brooks III, C. L. , *Biophysical Journal* **2003,** *85*, 2900-18; (d) Tanizaki, S.; Feig, M., *Journal of Chemical Physics* **2005,** *122* (12); (e) Tanizaki, S.; Feig, M., *Journal of Physical Chemistry B* **2006,** *110* (1), 548-556; (f) Callenberg, K. M., Choudhary, O. P., de Forest, G. L., Gohara, D. W., Baker, N. A., and Grabe, M. , *PLoS One* **2010,** *5* (9), 1-11.
4. (a) Warwicker, J.; Watson, H. C., *J Mol Biol* **1982,** *157* (4), 671-679; (b) Bashford, D.; Karplus, M., *Biochemistry* **1990,** *29* (44), 10219-10225; (c) Jeancharles, A.; Nicholls, A.; Sharp, K.; Honig, B.; Tempczyk, A.; Hendrickson, T. F.; Still, W. C., *Journal of the American Chemical Society* **1991,** *113* (4), 1454-1455; (d) Gilson, M. K., *Curr. Opin. Struct. Biol.* **1995,** *5* (2), 216-223; (e) Edinger, S. R.; Cortis, C.; Shenkin, P. S.; Friesner, R. A., *J. Phys. Chem. B* **1997,** *101* (7), 1190-1197; (f) Luo, R.; David, L.; Gilson, M. K., *Journal of Computational Chemistry* **2002,** *23* (13), 1244-1253; (g) Lu, Q.; Luo, R., *Journal of Chemical Physics* **2003,** *119* (21), 11035-11047; (h) Tan, C.; Yang, L.; Luo, R., *Journal of Physical Chemistry B* **2006,** *110* (37), 18680-18687; (i) Cai, Q.; Wang, J.; Zhao, H.-K.; Luo, R., *Journal of Chemical Physics* **2009,** *130* (14); (j) Wang, J.; Cai, Q.; Li, Z.-L.; Zhao, H.-K.; Luo, R., *Chemical Physics Letters* **2009,** *468* (4-6), 112-118; (k) Ye, X.; Cai, Q.; Yang, W.; Luo, R., *Biophysical Journal* **2009,** *97* (2), 554-562; (l) Ye, X.; Wang, J.; Luo, R., *Journal of Chemical Theory and Computation* **2010,** *6* (4), 1157-1169; (m) Luo, R.; Moult, J.; Gilson, M. K., *Journal of Physical Chemistry B* **1997,** *101* (51), 11226-11236; (n) Wang, J.; Tan, C.; Chanco, E.; Luo, R., *Physical Chemistry Chemical Physics* **2010,** *12* (5), 1194-1202; (o) Wang, J.; Luo, R., *Journal of Computational Chemistry* **2010,** *31* (8), 1689-1698; (p) Cai, Q.; Hsieh, M.-J.; Wang, J.; Luo, R., *Journal of Chemical Theory and Computation* **2010,** *6* (1), 203-211; (q) Hsieh, M. J.; Luo, R., *Journal of Molecular Modeling* **2011,** *17* (8), 1985-1996; (r) Cai, Q.; Ye, X.; Wang, J.; Luo, R., *Journal of Chemical Theory and Computation* **2011,** *7* (11), 3608-3619; (s) Wang, J.; Cai, Q.; Xiang, Y.; Luo, R., *Journal of*

Chemical Theory and Computation **2012,** *8* (8), 2741-2751; (t) Botello-Smith, W. M.; Liu, X.; Cai, Q.; Li, Z.; Zhao, H.; Luo, R., *Chemical Physics Letters* **2012**; (u) Liu, X.; Wang, C.; Wang, J.; Li, Z.; Zhao, H.; Luo, R., *Physical Chemistry Chemical Physics* **2013**; (v) Wang, C.; Wang, J.; Cai, Q.; Li, Z. L.; Zhao, H.; Luo, R., *Computational and Theoretical Chemistry* **2013,** *1024*, 34-44.

5. (a) Klapper, I.; Hagstrom, R.; Fine, R.; Sharp, K.; Honig, B., *Proteins Structure Function and Genetics* **1986,** *1* (1), 47-59; (b) Davis, M. E.; McCammon, J. A., *Journal of Computational Chemistry* **1989,** *10* (3), 386-391; (c) Nicholls, A.; Honig, B., *Journal of Computational Chemistry* **1991,** *12* (4), 435-445; (d) Luty, B. A.; Davis, M. E.; McCammon, J. A., *Journal of Computational Chemistry* **1992,** *13* (9), 1114-1118; (e) Holst, M.; Saied, F., *Journal of Computational Chemistry* **1993,** *14* (1), 105-113; (f) Holst, M. J.; Saied, F., *Journal of Computational Chemistry* **1995,** *16* (3), 337-364; (g) Bashford, D., *Lecture Notes in Computer Science* **1997,** *1343*, 233-240; (h) Im, W.; Beglov, D.; Roux, B., *Comput. Phys. Commun.* **1998,** *111* (1-3), 59-75; (i) Rocchia, W.; Alexov, E.; Honig, B., *J. Phys. Chem. B* **2001,** *105* (28), 6507-6514.

6. (a) Cortis, C. M.; Friesner, R. A., *Journal of Computational Chemistry* **1997,** *18* (13), 1591-1608; (b) Holst, M.; Baker, N.; Wang, F., *Journal of Computational Chemistry* **2000,** *21* (15), 1319-1342; (c) Baker, N.; Holst, M.; Wang, F., *Journal of Computational Chemistry* **2000,** *21* (15), 1343-1352; (d) Shestakov, A. I.; Milovich, J. L.; Noy, A., *Journal of Colloid and Interface Science* **2002,** *247* (1), 62-79; (e) Chen, L.; Holst, M. J.; Xu, J. C., *Siam Journal on Numerical Analysis* **2007,** *45*, 2298-2320; (f) Xie, D.; Zhou, S., *BIT Numerical Mathematics* **2007,** *47* (4), 853-871; (g) Wang, J.; Cieplak, P.; Li, J.; Wang, J.; Cai, Q.; Hsieh, M.; Lei, H.; Luo, R.; Duan, Y., *Journal of Physical Chemistry B* **2011,** *115* (12), 3100-3111; (h) Lu, B.; Holst, M. J.; McCammon, J. A.; Zhou, Y. C., *J Comput Phys* **2010,** *229* (19), 6979-6994; (i) Bond, S. D.; Chaudhry, J. H.; Cyr, E. C.; Olson, L. N., *Journal of Computational Chemistry* **2010,** *31* (8), 1625-1635.

7. (a) Miertus, S.; Scrocco, E.; Tomasi, J., *Chemical Physics* **1981,** *55* (1), 117-129; (b) Hoshi, H.; Sakurai, M.; Inoue, Y.; Chujo, R., *Journal of Chemical Physics* **1987,** *87* (2), 1107-1115; (c) Zauhar, R. J.; Morgan, R. S., *Journal of Computational Chemistry* **1988,** *9* (2), 171-187; (d) Rashin, A. A., *J. Phys. Chem.* **1990,** *94* (5), 1725-1733; (e) Yoon, B. J.; Lenhoff, A. M., *Journal of Computational Chemistry* **1990,** *11* (9), 1080-1086; (f) Juffer, A. H.; Botta, E. F. F.; Vankeulen, B. A. M.; Vanderploeg, A.; Berendsen, H. J. C., *J Comput Phys* **1991,** *97* (1), 144-171; (g) Zhou, H. X., *Biophys J* **1993,** *65* (2), 955-963; (h) Bharadwaj, R.; Windemuth, A.; Sridharan, S.; Honig, B.; Nicholls, A., *Journal of Computational Chemistry* **1995,** *16* (7), 898-913; (i) Purisima, E. O.; Nilar, S. H., *Journal of Computational Chemistry* **1995,** *16* (6), 681-689; (j) Liang, J.; Subramaniam, S., *Biophysical Journal* **1997,** *73* (4), 1830-1841; (k) Vorobjev, Y. N.; Scheraga, H. A., *Journal of Computational Chemistry* **1997,** *18* (4), 569-583; (l) Totrov, M.; Abagyan, R., *Biopolymers* **2001,** *60* (2), 124-133; (m) Boschitsch, A. H.; Fenley, M. O.; Zhou, H. X., *J. Phys. Chem. B* **2002,** *106* (10), 2741-2754; (n) Lu, B. Z.; Cheng, X. L.; Huang, J. F.; McCammon, J. A., *Proceedings of the National Academy of Sciences of the United States of America* **2006,** *103* (51), 19314-19319; (o) Lu, B.; Cheng, X.; Huang, J.; McCammon, J. A., *Journal of Chemical Theory and Computation* **2009,** *5* (6), 1692-1699; (p) Bajaj, C.; Chen, S.-C.; Rand, A., *Siam Journal on Scientific Computing* **2011,** *33* (2), 826-848.

8. (a) Georgescu, R. E.; Alexov, E. G.; Gunner, M. R., *Biophys J* **2002,** *83* (4), 1731-1748; (b) Nielsen, J. E.; McCammon, J. A., *Protein Sci* **2003,** *12* (2), 313-326; (c) Warwicker, J., *Protein Sci* **2004,** *13* (10), 2793-2805; (d) Tang, C. L.; Alexov, E.; Pyle, A. M.; Honig, B., *J Mol Biol* **2007,** *366* (5), 1475-1496.

9. (a) Shivakumar, D.; Deng, Y. Q.; Roux, B., *J. Chem. Theory Comput.* **2009,** *5* (4), 919-930; (b) Nicholls, A.; Mobley, D. L.; Guthrie, J. P.; Chodera, J. D.; Bayly, C. I.; Cooper, M. D.; Pande, V. S., *Journal of Medicinal Chemistry* **2008,** *51* (4), 769-779.

10. (a) Swanson, J. M. J.; Henchman, R. H.; McCammon, J. A., *Biophys J* **2004,** *86* (1), 67-74; (b) Bertonati, C.; Honig, B.; Alexov, E., *Biophysical Journal* **2007,** *92* (6), 1891-1899; (c) Brice, A. R.; Dominy, B. N., *J Comput Chem* **2011,** *32* (7), 1431-1440; (d) Luo, R.; Gilson, H. S. R.; Potter, M. J.; Gilson, M. K., *Biophysical Journal* **2001,** *80* (1), 140-148; (e) David, L.; Luo, R.; Head, M. S.; Gilson, M. K., *Journal of Physical Chemistry B* **1999,** *103* (6), 1031-1044.

11. (a) Marshall, S. A.; Vizcarra, C. L.; Mayo, S. L., *Protein Sci* **2005,** *14* (5), 1293-1304; (b) Hsieh, M. J.; Luo, R., *Proteins-Structure Function and Bioinformatics* **2004,** *56* (3), 475-486; (c) Wen, E. Z.; Luo, R., *Journal of Chemical Physics* **2004,** *121* (5), 2412-2421; (d) Wen, E. Z.; Hsieh, M. J.; Kollman, P. A.; Luo, R., *Journal of Molecular Graphics & Modelling* **2004,** *22* (5), 415-424; (e) Lwin, T. Z.; Luo, R., *Journal of Chemical Physics* **2005,** *123* (19); (f) Lwin, T. Z.; Zhou, R. H.; Luo, R., *Journal of Chemical Physics* **2006,** *124* (3); (g) Lwin, T. Z.; Luo, R., *Protein Science* **2006,** *15* (11), 2642-2655; (h) Tan, Y.-H.; Luo, R., *Journal of Physical Chemistry B* **2008,** *112* (6), 1875-1883; (i) Tan, Y.; Luo, R., *BMC Biophysics* **2009,** *2* (1), 5.

12. (a) Wang, J.; Cai, Q.; Li, Z. L.; Zhao, H. K.; Luo, R., *Chemical Physics Letters* **2009,** *468* (4-6), 112-118; (b) Morozov, A. V.; Kortemme, T.; Baker, D., *J. Phys. Chem. B* **2003,** *107* (9), 2075-2090; (c) Wagoner, J.; Baker, N. A., *Journal of Computational Chemistry* **2004,** *25* (13), 1623-1629.

13. (a) Im, W.; Feig, M.; Brooks, C. L., *Biophysical Journal* **2003,** *85*, 2900-2918; (b) Nymeyer, H.; Zhou, H.-X., *Biophysical Journal* **2008,** *94*, 1185-1193; (c) Tanizaki, S.; Feig, M., *The Journal of chemical physics* **2005,** *122*, 124706; (d) Stern, H. A.; Feller, S. E., *The Journal of Chemical Physics* **2003,** *118*, 3401.
14. Ulmschneider, M. B.; Ulmschneider, J. P.; Sansom, M. S.; Di Nola, A., *Biophysical Journal* **2007,** *92* (7), 2338-2349.
15. (a) Srinivasan, J.; Cheatham, T. E.; Cieplak, P.; Kollman, P. A.; Case, D. A., *J Am Chem Soc* **1998,** *120* (37), 9401-9409; (b) Kollman, P. A.; Massova, I.; Reyes, C.; Kuhn, B.; Huo, S. H.; Chong, L.; Lee, M.; Lee, T.; Duan, Y.; Wang, W.; Donini, O.; Cieplak, P.; Srinivasan, J.; Case, D. A.; Cheatham, T. E., *Accounts Chem Res* **2000,** *33* (12), 889-897; (c) Gohlke, H.; Case, D. A., *Journal of Computational Chemistry* **2004,** *25* (2), 238-250; (d) Yang, T. Y.; Wu, J. C.; Yan, C. L.; Wang, Y. F.; Luo, R.; Gonzales, M. B.; Dalby, K. N.; Ren, P. Y., *Proteins-Structure Function and Bioinformatics* **2011,** *79* (6), 1940-1951; (e) Miller III, B. R.; McGee Jr, T. D.; Swails, J. M.; Homeyer, N.; Gohlke, H.; Roitberg, A. E., *J. Chem. Theory Comput.* **2012,** *8* (9), 3314-3321; (f) Miller, B. R.; McGee, T. D.; Swails, J. M.; Homeyer, N.; Gohlke, H.; Roitberg, A. E., *J Chem Theory Comput* **2012,** *8* (9), 3314-3321.
16. Callenberg, K. M.; Choudhary, O. P.; de Forest, G. L.; Gohara, D. W.; Baker, N. A.; Grabe, M., *Plos One* **2010,** *5* (9).
17. (a) Li, C.; Li, L.; Zhang, J.; Alexov, E., *Journal of Computational Chemistry* **2012,** *33* (24), 1960-1966; (b) Li, L.; Li, C.; Sarkar, S.; Zhang, J.; Witham, S.; Zhang, Z.; Wang, L.; Smith, N.; Petukh, M.; Alexov, E., *BMC biophysics* **2012,** *5* (1), 9.
18. Zhang, K.; Zhang, J.; Gao, Z.-G.; Zhang, D.; Zhu, L.; Han, G. W.; Moss, S. M.; Paoletta, S.; Kiselev, E.; Lu, W., *Nature* **2014,** *509* (7498), 115-118.
19. Webb, B.; Sali, A., *Current protocols in bioinformatics* **2014**, 5.6. 1-5.6. 32.
20. Humphrey, W.; Dalke, A.; Schulten, K., *Journal of Molecular Graphics & Modelling* **1996,** *14* (1), 33-38.
21. Roberts, E.; Eargle, J.; Wright, D.; Luthey-Schulten, Z., *BMC bioinformatics* **2006,** *7* (1), 382.
22. Wang, J.; Wang, W.; Kollman, P. A.; Case, D. A., *J. Am. Chem. Soc* **2001,** *222*, U403.
23. Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J., *Journal of Computational Chemistry* **2005,** *26* (16), 1668-1688.
24. *Spartan'14*, Wavefunction, Inc.: Irvine, CA, 2014.
25. Jakalian, A.; Jack, D. B.; Bayly, C. I., *Journal of computational chemistry* **2002,** *23* (16), 1623-1641.
26. Storer, A. C.; Cornish-Bowden, A., *Biochem. J* **1976,** *159*, 1-5.
27. Jo, S.; Kim, T.; Im, W., *Plos One* **2007,** *2* (9), e880.
28. Biró, É.; Akkerman, J. W. N.; Hoek, F. J.; Gorter, G.; Pronk, L. M.; Sturk, A.; Nieuwland, R., *Journal of Thrombosis and Haemostasis* **2005,** *3* (12), 2754-2763.
29. Eisenberg, D.; McLachlan, A. D., **1986**.
30. Tan, C.; Tan, Y.-H.; Luo, R., *The Journal of Physical Chemistry B* **2007,** *111* (42), 12263-12274.
31. (a) Hadi-Alijanvand, H.; Rouhani, M., *The Journal of Physical Chemistry B* **2015,** *119*, 6113-6128; (b) Supunyabut, C.; Fuklang, S.; Sompornpisut, P., *Journal of Molecular Graphics and Modelling* **2015,** *59*, 81-91; (c) Eddy, M. T.; Andreas, L.; Teijido, O.; Su, Y.; Clark, L.; Noskov, S. Y.; Wagner, G.; Rostovtseva, T. K.; Griffin, R. G., *Biochemistry* **2015,** *54*, 994-1005.
32. Smith, W.; Luo, R., *Journal of chemical information and modeling* **2015**.

CHAPTER 4: Summary and Conclusion

Continuum electrostatics modeling of solvation based on the Poisson-Boltzmann (PB) equation has gained wide acceptance in biomolecular applications such as energetic analysis and structural visualization. Recently much attention has been given to inclusion of implicit membrane into existing continuum Poisson-Boltzmann solvent models to extend their applications to membrane systems, which are important computational research topics due to their roles in rational drug design.

In this dissertation, we first developed a continuum membrane model under the numerical PB framework for applications such as protein-ligand binding affinity and docking pose predictions. Both the level set method based on a revised density function and the classical geometric method based on the solvent excluded surface were developed for biomolecular applications. In both strategies, the membrane region is modeled as a slab region of a specified dielectric constant to model its influence in electrostatics calculations.

Apparently, inclusion of an implicit membrane complicates numerical solutions of the underlining Poisson-Boltzmann equation due to the dielectric inhomogeneity on the boundary surfaces of a computation grid. This can be alleviated by the use of the periodic boundary condition, a common practice in electrostatic computations in particle simulations. The conjugate gradient and successive over-relaxation methods are relatively straightforward to be adapted to periodic calculations, but their convergence rates are quite low, limiting their applications to free energy simulations that require a large number of conformations to be processed. To accelerate convergence, the Incomplete

Cholesky preconditioning and the geometric multi-grid methods have been extended to incorporate periodicity for biomolecular applications. Impressive convergence behaviors were found as in the previous applications of these numerical methods to tested biomolecules and (Molecular Mechanics Poisson-Boltzmann Surface Area) MMPBSA calculations. This lays foundations for sophisticated models with variable dielectric treatments and second-order accurate modeling of solvation interactions.

Successful application of the PB solvent models requires careful calibration of the solvation parameters. Extensive testing and validation is also important to ensure accuracy in their applications. Limitation in the continuum modeling of solvation is also a known issue in certain biomolecular applications. Growing interest in membrane systems has further spurred developmental efforts to allow inclusion of membrane in the PB solvent models. Despite their past successes due to careful parameterization, algorithm development, and parallel implementation, there is still much to be done to improve their transferability from the small molecular systems upon which they were developed and validated to complex macromolecular systems as advances in technology continue to push forward, providing ever greater computational resources to researchers to study more interesting biological systems of higher complexity.