A new computationally facile analytical approximation of electrostatic potential suitable for macromolecules.

John Carroll Gordon

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science

Alexey Onufriev, Chair
T.M. Murali
Adrian Sandu
Lenwood S. Heath

February 1, 2007
Blacksburg, Virginia

Keywords: Electrostatic Potential, ALPB, GEM, Poisson-Boltzmann, Virus, Macromolecule

Abstract

John Carroll Gordon

A new computationally facile analytical approximation of electrostatic potential suitable for macromolecules.

The electrostatic properties of a molecule are often essential in determining its behavior; as such, the ability to approximate these electrostatic potentials computationally is often essential to obtaining a full understanding of how these molecules function. An approximate, analytical solution to the (linearized) Poisson-Boltzmann equation is proposed that is suitable for realistic biomolecules of virtually any size. A comparison with accepted numerical approaches on a large test set of biomolecular structures shows that the proposed method is considerably less expensive computationally, yet accurate enough to be considered as a possible alternative. The utility of the approach is demonstrated by computing and analyzing the electrostatic potential generated by full capsid of the tobacco ringspot virus (half a million atoms) at atomic resolution. The details of the potential distribution on the molecular surface sheds light on the mechanism behind the high selectivity of the capsid to the viral RNA. These results are generated with the modest computational power of a desktop PC. The applicability of the analytical approximation as an initial guess for traditional numerical methods as a means of improving the convergence of iterative solutions is investigated and found to be quite promising.

Dedication

This work is dedicated to my family and Dr. Alexey Onufriev. I am certain that my life would be fundamentally worse without their support, encouragement, and advice.

Acknowledgments

I would like to thank Dr. Onufriev for his fatherly advice and interest in me as a graduate student at Virginia Tech. His guidance was invaluable during a number of scary moments. I would also like to thank Dr. Murali for his support and instruction during my time here. I would like to thank Dr. Heath for his advice on writing papers (which I have taken to heart) and his excellent critical eye. I couldn't forget Ginger Clayton, who saved me from myself a number of times (always with a smile). Of course Andrew Fenley, Jon Myers, and Jory Zmuda helped me too many times and in too many ways to count.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The utility of electrostatic potential for gaining understanding of the function of proteins[2] and nucleic acids[3] has long been established. Electrostatic interactions are often a key factor determining properties of biomolecules,[2,4–7] including functions such as: catalytic activity,[8,9] ligand binding,[10,11] complex formation,[12] proton transport,[13] and structural stability.[14,15] In-depth studies of electrostatics-based phenomena in large molecular systems require the ability to compute the potentials and fields efficiently and accurately on and below the atomic scale (approximately 2 Åand smaller).

A text-book example is the function of the enzyme *acetylcholine esterase*, which is a key enzyme involved in the transmission of nerve impulses across synapses (nerve junctions). This signal is passed by a mediator molecule *acetylcholine* that is steered into the enzyme's active site by electrostatic forces. Electrostatic steering contributes to the rapid reaction rate required for this enzyme to function in the context of a mechanism for rapid impulse transmission. Deciphering the underlying molecular mechanism was only possible by computing a detailed picture of the electrostatic field and potential generated by the enzyme. Figure 1.1 demonstrates the electrostatic surface potential of acetylcholine esterase as computed by `DelPhi` and displayed by `GRASP`.[1,16,17] The acetylcholine molecule is represented by the small green molecule inside the red pocket (here colored red because its electrostatic potential is negative). The negative electrostatic potential of the pocket attracts the positively charged ligand to improve the rate of uptake. The importance of electrostatic potential is not unique to acetylcholine esterase; electrostatic steering has been shown to play a significant role in a broad range of neurological functions, such as: presynaptic vesicle-cell membrane fusion,[18] norepinephrine uptake,[19] and uptake of drugs such as cocaine by dopamine receptors.[20]

Electrostatic forces are the result of attractions and repulsions between positive and negative charges. These forces are particularly powerful at the atomic scale, and apply even to neutrally charged molecules (where the charge distribution may be uneven in some sense but neutral overall). The Poisson-Boltzmann (PB) formulation for determining electrostatic

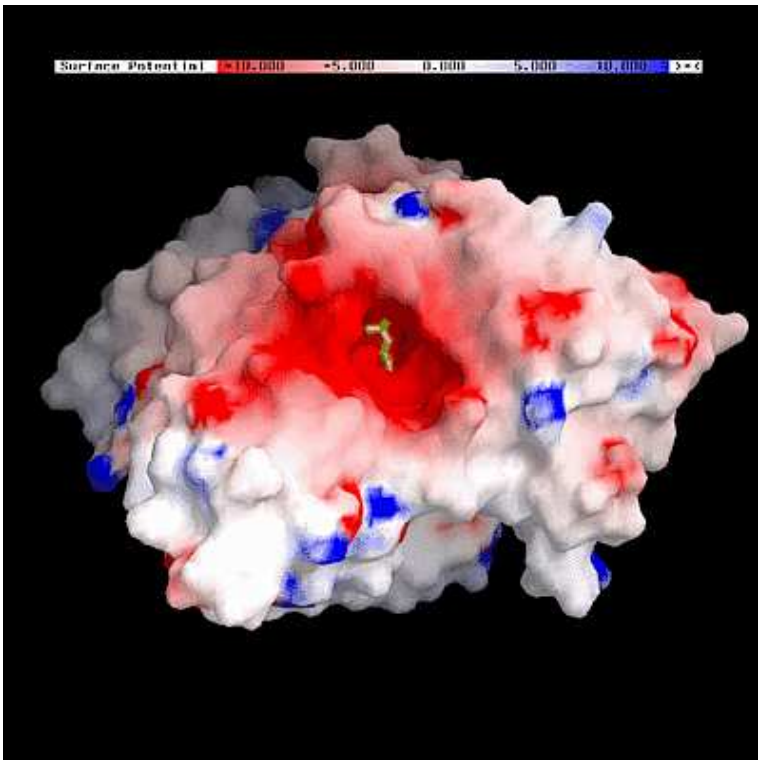

Figure 1.1: Acetylcholine esterase bound to an acetylcholine mediator molecule.[1]

potential at a point in space is:

$$\begin{aligned} \nabla\epsilon(\vec{r})\nabla\phi(\vec{r}) + \kappa^{-2}(\vec{r})sinh[\phi(\vec{r})] &= -4\pi\rho(\vec{r}) \\ \lim_{\vec{r}\to\infty}\phi(\vec{r}) &= 0 \end{aligned} \tag{1.1}$$

where $\vec{r}$ is the position in space, $\rho(\vec{r})$ is the charge density at $\vec{r}$, κ is the Debye-Hückel screening parameter, $\phi(\vec{r})$ represents the electrostatic potential at position $\vec{r}$, and ϵ is the dielectric coefficient.[21]

Equation (1.1) is called the *nonlinear Poisson-Boltzmann* equation. Within the implicit solvent model (in which solvent molecules are treated as a continuous dielectric environment rather than as individual molecules in calculations), the nonlinear Poisson-Boltzmann equation is considered to be the most accurate description of electrostatic potential. However, this equation is very difficult to solve for a given charge distribution.

$$\begin{aligned} \nabla\epsilon(\vec{r})\nabla\phi(\vec{r}) &= -4\pi\rho(\vec{r}) \\ \lim_{\vec{r}\to\infty}\phi(\vec{r}) &= 0 \end{aligned} \tag{1.2}$$

Equation (1.2) is Poisson's equation, the prototypical equation defining how electrostatic potential behaves in a vacuum. In Poisson's equation electrostatic potential is linear, in that

increasing a charge by some factor increases the potential generated by that charge by the same factor ($\phi(k\rho) = k\phi(\rho)$ where k is a constant). It is also additive, in that the sum of the potential generated by two charge distributions is the same as the potential generated by the sum of the two charge distributions ($\phi(\rho_1 + \rho_2) = \phi(\rho_1) + \phi(\rho_2)$).

It can be seen that the nonlinear PB equation does not have these properties due to the exponential term $sinh[\phi(\vec{r})]$. These properties are both physically and computationally desirable because they are believed to be properties of electrostatic potential in general and because they dramatically improve the computability of the problem. For these reasons, a linear approximation of the PB problem is typically used in practice as a basis for approximating electrostatic potentials.

This paper focuses on the more practical *linearized Poisson-Boltzmann* (LPB) equation. The linear Poisson-Boltzmann equation is centered around using the Taylor series expansion of the exponential term in Equation (1.1) to produce:

$$\begin{aligned} \nabla\epsilon(\vec{r})\nabla\phi(\vec{r}) &= -4\pi\rho(\vec{r}) + \kappa^2(\vec{r})\epsilon(\vec{r})\phi(\vec{r}) \\ \lim_{\vec{r}\to\infty} \phi(\vec{r}) &= 0 \end{aligned} \tag{1.3}$$

The linearization of the second term is the result of expanding the exponential $e^{-Z_A\frac{\phi(x)}{kT}}$ as $1 - Z_A c_A \frac{\phi(x)}{kT} + \frac{1}{2}[Z_A^2 c_A \frac{\phi(x)}{kT}] - \cdots$ and ignoring all terms of order 2 and higher. Z_A is the valence of mobile ion species A, c_A is the concentration of mobile ion species A, and all mobile ions are assumed to follow a Boltzmann distribution throughout the medium. The resulting term $\kappa^2(\vec{r})$ is equal to $\frac{8\pi e^2 I}{\epsilon(\vec{r})kT}$ where $I = \frac{1}{2}\left[Z_A^2 c_A + Z_B^2 c_B + \cdots\right]$, e is the charge of a proton, and kT is a thermal unit.[22] This approximation is valid only when $-Z_A\frac{\phi(x)}{kT} \ll 1$, which is found to be valid for most practical cases. This linear equation has all the desirable properties of Poisson's equation.

Traditionally, methods based upon numerical solutions of the Poisson-Boltzmann equation (herein referred to as the NPB approach) have been used to compute the electrostatic potential of biological structures. However the use of the NPB methodology to study electrostatic properties of bio-molecules becomes problematic as ever larger, high-resolution structures become available to structural biologists through the advances of X-ray crystallography and other imaging techniques. For example, a recent NPB-based study of the *ribosomal complex* – a structure of nearly 100,000 atoms – required sophisticated parallel computations on 343 CPUs of the Blue Horizon supercomputer[23] for an unpublished amount of time. Viral capsids, which are typically much larger, are expected to present an even greater challenge to the traditional numerical approach. In this thesis, a radically different approach is used to compute and analyze the electrostatic potential of the tobacco ringspot viral capsid. The method is based on a closed form analytical solution of the Poisson-Boltzmann equation. Extensive comparisons with the numerical reference reveal that the method is accurate enough for practical purposes; its computational efficiency allows us to perform the study of the full viral capsid made up of half a million atoms on a desktop PC. The method is derived

in Chapter 4 and validated in Chapter 5. The method is conceptually facile to implement, computationally facile to execute, and sufficiently accurate to be used in many applications.

Chapter 2

Related Methods

2.1 Numerical Poisson-Boltzmann (NPB) Frameworks

Here follows a terse explanation of the general steps involved in solving partial differential equations using numerical methods as a means of providing some understanding of their limitations and strengths. The methods are first derived for example problems up to the point that the problem becomes a system of simultaneous equations. An iterative method for solving these sparse matrix equations is then explained in order to complete the derivation and to focus on the primary bottleneck involved in storing and computing these approximations.

2.1.1 Finite Difference Methods

Finite difference methods are useful for approximating the solution of differential equations that are otherwise intransigent or overwhelmingly difficult to solve. However, the methods themselves are easiest to present within the context of solving a simple problem. Poisson's equation (Equation (2.2)) is presented in one dimension as a means of focusing on the method. Dirichlet boundary conditions are used to represent the values of u at both edges of the solution. Equation (2.2) represents a common class of diffusion equations; u represents an unknown function of an independent variable x, ρ represents a source term related to a derivative of u, and k is an arbitrary terminal boundary value for x in order to use Dirichlet boundaries on both sides of the domain for the sake of symmetry in the matrix solution.

$$\frac{\partial^2 u}{\partial x^2} = -\rho(x) \tag{2.1}$$

$$u(0) = 0$$
$$u(k) = 0 \tag{2.2}$$

The key approximation used in the finite difference method is to estimate the gradients by a *differencing operator* (for example a backward difference operator would be $\frac{\partial f(x)}{\partial x} \simeq \frac{f(x)-f(x-\Delta)}{\Delta}$)[22,24–26] on a uniform distribution of finite sample points $x_i = x_0 + i * \Delta$ where Δ is the uniform spacing value.

Equation (2.2) can be transformed to Equation (2.3) under a simple first order difference operator with second order convergence.

$$u(x_{i+1}) - 2u(x_i) + u(x_{i-1}) = -\Delta^2 \rho(x_i) \tag{2.3}$$

The differencing operator determines the order of the approximation being used in the finite difference method. There are various differencing schemes with different orders of accuracy including: forward differencing, backward differencing, and midpoint methods. These differencing approximations determine the slope of a secant line about some point p. The secant line, geometrically, represents the slope between two points. So a forward differencing operator would provide the slope between p and $p + h$, a backward differencing operator would provide the slope between $p - h$ and p where h is an arbitrary value. For a straight line, the slope is constant, so the forward and backward differencing operators estimate the tangent exactly and both evaluate to the same value. For this reason, they are viewed as first order differencing operators.

For second order functions, the slope uniformly changes in a linear fashion so secant approximations centered about p approximate the tangent at p exactly in the sense that it linearly interpolates a linear derivative. For simple linear functions with constant derivatives, linear interpolants still exactly represent the slope at p, so centered differencing operators capture the slopes of both second order and first order equations. Therefore, centered differencing operators result in second order approximations.

As the order of approximation improves, the support (elements in each row not equal to 0) of the final matrix generally increases and so the amount of work required to solve the set of simultaneous equations also increases. Therefore, selection of the differencing operator is fundamental to the accuracy and time complexity of the overall method and should be done with great care.

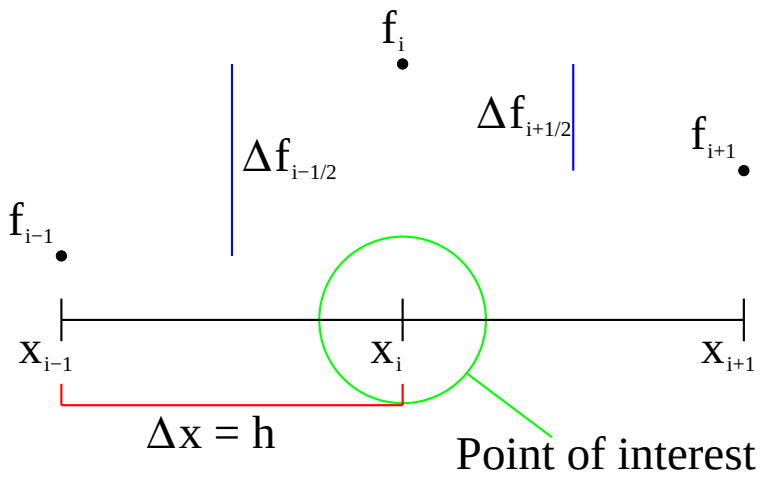

Figure 2.1: Example finite element with approximations in one dimension.

It is useful to decompose the problem to a finite sub-domain to fully understand the tessellation of interactions inherent in the solution. Figure 2.1 demonstrates a decomposition of the problem into a single point-centered sub-domain in one dimension. The f values represent approximations of the true solution while the Δf values (blue lines) represent approximations of the change in f between two points (uniformly scaled by $\frac{1}{\Delta x}$ to obtain approximations of the derivative). The Δf values are indexed with $i \pm \frac{1}{2}$ to indicate that these derivatives are approximated at the half distance between i and its adjacent points. It can be seen that the distance between both the derivative estimates and the function sample points are h and that x_i interacts with the two adjacent sample points x_{i+1} and x_{i-1}.

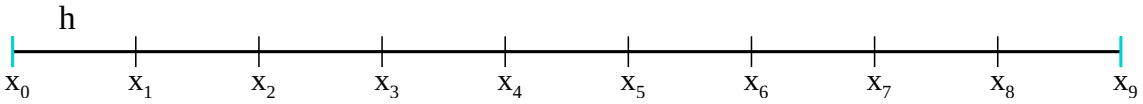

Figure 2.2: Sample discretization of the given 1-dimensional problem.

The next step is to construct a uniform mesh of the domain such as that in Figure 2.2. The x_i values represent equidistant sample points along the domain where instances of Equation (2.3) would be solved. Adjacent elements contain interdependencies generated through the estimation of the partial derivative. The edges (x_0 and x_9) represent boundary values, which have different fundamental equations describing them (such as $u(x_0) = 0$) and require special treatment. For our example, we are setting $h = \frac{k}{9}$ to use a fixed number of elements and to keep the resulting set of linear equations small and presentable.

Equation (2.4) represents a matrix form of Equation (2.3). The matrix is square and symmetric because special care is taken at the boundaries (seen in the matrix as rows with only two terms in them).

$$\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = -\Delta^2 \begin{bmatrix} \rho_1 - x_0 \\ \rho_2 \\ \rho_3 \\ \rho_4 \\ \rho_5 \\ \rho_6 \\ \rho_7 \\ \rho_8 - x_9 \end{bmatrix} \tag{2.4}$$

Equation (2.4) is by no means representative of the form of two and three dimensional problem. With each additional dimension, two interactions are incurred at some increased distance in the matrix due to the numbering. In this case, for a domain with g elements along an edge, two dimensional problems incur two additional terms g elements away in a row column numbering and three dimensional problems require consideration of two dimensional terms as well as two interactions g^2 elements away in the same element numbering system. The tri-diagonal nature of the one-dimensional problem, then, is unique to 1 dimensional problems.

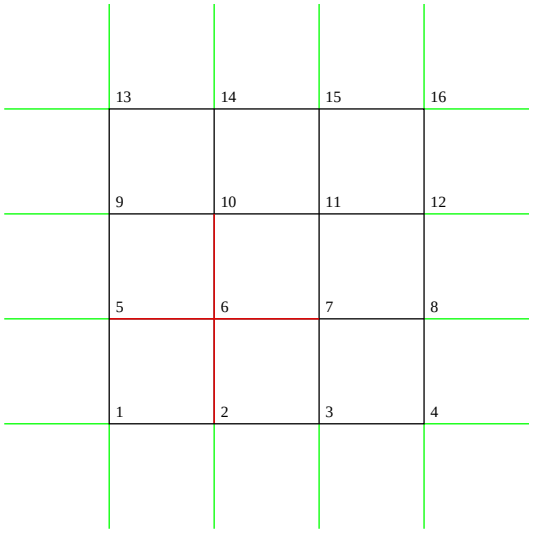

Figure 2.3: Sample discretization of a simple two-dimensional domain.

The two dimensional domain in Figure 2.3 represents a simple two-dimensional discretization. The red lines represent the dependencies of node 6, the green lines represent dependencies upon a Dirichlet boundary condition which (in this example) has been separated over to the other side of the equation to simplify it. For this particular example with 16 internal elements, the matrix component of the equation representing these interdependent linear equations is:

$$
\begin{bmatrix}
-4 & 1 & & & 1 & & & & & & & & & & & \\
1 & -4 & 1 & & & 1 & & & & & & & & & & \\
 & 1 & -4 & 1 & & & 1 & & & & & & & & & \\
 & & 1 & -4 & 1 & & & 1 & & & & & & & & \\
1 & & & 1 & -4 & 1 & & & 1 & & & & & & & \\
 & 1 & & & 1 & -4 & 1 & & & 1 & & & & & & \\
 & & 1 & & & 1 & -4 & 1 & & & 1 & & & & & \\
 & & & 1 & & & 1 & -4 & 1 & & & 1 & & & & \\
 & & & & 1 & & & 1 & -4 & 1 & & & 1 & & & \\
 & & & & & 1 & & & 1 & -4 & 1 & & & 1 & & \\
 & & & & & & 1 & & & 1 & -4 & 1 & & & 1 & \\
 & & & & & & & 1 & & & 1 & -4 & 1 & & & \\
 & & & & & & & & 1 & & & 1 & -4 & 1 & & \\
 & & & & & & & & & 1 & & & 1 & -4 & 1 & \\
 & & & & & & & & & & 1 & & & 1 & -4 &
\end{bmatrix}
\tag{2.5}
$$

2.1.2 Finite Element Methods

This entire section can be seen as a broad overview of Galerkin's method for solving partial differential equations, however there are much better resources available for advanced study.[21,23,27–31] Here Galerkin's method is used as an example to solve a typical finite element problem as a means of touching on the steps involved. Equation (2.6) is used as an example problem to explore the finite element approach to solving partial differential equations where p and u are unknown functions, Ω is a mathematical symbol representing the domain, and $\partial\Omega$ represents the surface.

$$\begin{aligned} -\nabla[p\nabla u] &= f(x,y) \\ u(x,y)=0, u_n(x,y)=0 &: (x,y)\in\partial\Omega \end{aligned} \tag{2.6}$$

First, Equation (2.6) is multiplied by a *test function* v of the same order and in the same domain as u and integrated as a means of estimating the strain energy (which should be minimized) of the approximation. This results in Equation (2.7) which is called the *weak formulation* of the problem.

$$\iint_\Omega -v\nabla[p\nabla u] = \iint_\Omega f(x,y)v \tag{2.7}$$

Or after applying the "product rule" for gradient operators as well as Green's theorem:

$$\iint_\Omega [\nabla v\cdot(p\nabla u)]\partial x\partial y - \int_{\partial\Omega} vpu_n\partial s = \iint_\Omega vf\partial x\partial y \tag{2.8}$$

The surface integral evaluates to 0 after applying boundary conditions, so Equation (2.8) becomes:

$$\iint_\Omega [\nabla v\cdot(p\nabla u)]\partial x\partial y = \iint_\Omega vf\partial x\partial y \tag{2.9}$$

The next step is to discretize the domain. A uniform grid spacing of h will be used here with square elements similar to that in Figure 2.2. For a uniform square discretization, each element will have four adjacent elements with whom it must share a boundary. Figure 2.4 contains an example numbering of the vertices on one element.

The next step is to construct a set of simple basis functions that must (in this case) have defined first derivatives. A *hat function* is a standard linear interpolant which mathematically represents the derivative of a delta function and the integral of a step function simultaneously. Equation (2.10) represents a typical hat function of width w centered at k with a peak value of z.

$$F(x) = \begin{cases} 0 \text{ if } x > k+w \\ 0 \text{ if } x \le k-w \\ z(x+w-k)/w \text{ if } k-w < x \le k \\ z - z(x-k)/w \text{ if } k < x \le k+w \end{cases} \tag{2.10}$$

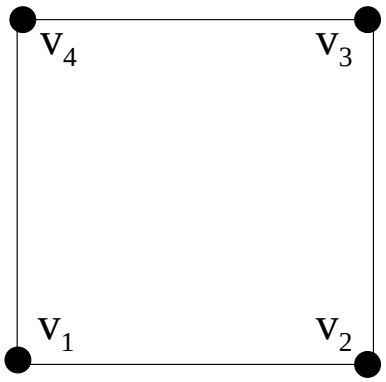

Figure 2.4: Example vertex numbering for a canonical square element

A simple hat function will satisfy this requirement. Equation (2.11) is an example set of first order basis functions that could be used for this problem.

$$\begin{aligned} N_1(x,y) &= \frac{h-x}{h} * \frac{h-y}{h} \\ N_2(x,y) &= \frac{x}{h} * \frac{h-y}{h} \\ N_3(x,y) &= \frac{x}{h} * \frac{y}{h} \\ N_4(x,y) &= \frac{h-x}{h} * \frac{y}{h} \end{aligned} \tag{2.11}$$

The partial derivatives of N defined in Equation (2.11) are:

$$\begin{aligned}
\frac{\partial N_1(x,y)}{\partial x} &= \frac{-1}{h} * \frac{h-y}{h} \\
\frac{\partial N_1(x,y)}{\partial y} &= \frac{h-x}{h} * \frac{-1}{h} \\
\frac{\partial N_2(x,y)}{\partial x} &= \frac{1}{h} * \frac{h-y}{h} \\
\frac{\partial N_2(x,y)}{\partial y} &= \frac{x}{h} * \frac{-1}{h} \\
\frac{\partial N_3(x,y)}{\partial x} &= \frac{1}{h} * \frac{y}{h} \\
\frac{\partial N_3(x,y)}{\partial y} &= \frac{x}{h} * \frac{1}{h} \\
\frac{\partial N_4(x,y)}{\partial x} &= \frac{-1}{h} * \frac{y}{h} \\
\frac{\partial N_4(x,y)}{\partial y} &= \frac{h-x}{h} * \frac{1}{h}
\end{aligned} \tag{2.12}$$

Note that the N_k functions are designed to be 1 at vertex k and 0 elsewhere, also these particular N functions are derived for a canonical element of dimension h with v_1 at the origin. To transform an arbitrary element whose v_1 vertex is at (x_0, y_0), one would simply subtract x_0 from the x values in the functions and y_0 from the y values, this is simply a translation function because elements are uniformly sized due to the uniform spacing. The next step is to approximate u and v by U and V defined in Equation (2.13):

$$\begin{aligned}
u \simeq U &= c_1 N_1 + c_2 N_2 + c_3 N_3 + c_4 N_4 \\
v \simeq V &= N_1 + N_2 + N_3 + N_4
\end{aligned} \tag{2.13}$$

The lack of constants to solve for in V can be attributed to a uniform weight factor in the test function, which implies that each point in the equation is equally important. Constants would exist in the V vector if some regions in the space were more or less important than the rest.

If U and V are represented as vectors:

$$\begin{aligned} V &= \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} \\ U &= [N_1, N_2, N_3, N_4] \\ C &= \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} \end{aligned}$$

$$\iint_\Omega p \nabla V \nabla U C = \iint_\Omega V f \partial x \partial y \tag{2.14}$$

f is typically approximated in the same space as U by $F = N_1 * f_1 * N_2 * f_2 + N_3 * f_3 + N_4 * f_4$. Where f_i is a discrete sample of f at vertex i.

The integrated outer product of ∇U and ∇V with the consideration of F:

$$P_j = \frac{1}{h^2} * \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} \iint_\Omega F * N_1 \partial x \partial y \\ \iint_\Omega F * N_2 \partial x \partial y \\ \iint_\Omega F * N_3 \partial x \partial y \\ \iint_\Omega F * N_4 \partial x \partial y \end{bmatrix} \tag{2.15}$$

Equation (2.15) defines the linear equations for one element. The form of the final matrix containing the solution for the whole domain will depend on the ordering of the elements, as there are interactions with neighboring elements that share vertices. The final matrix problem will be the sum of all P_j, which will be a sparse linear system. If the elements and vertices are ordered correctly, then the system will be diagonally dominant and banded. The C array will consist of sums of different unknown c_i values from different elements. The last step in solving this problem is to solve the matrix equation $AC = F$.

2.1.3 Solving the sparse linear systems

Both finite element and finite difference methods require the solution of a large set of linear equations in the matrix equation $Ax = b$ where A is sparse, positive definite, and diagonally dominant. It is standard practice to approach this problem with the goal of quickly and approximately solving the problem, and so relaxation methods are commonly used to solve this set of equations.[22,24,32] Successive Overrelaxation is an iterative method for estimating the solution to large, sparse matrix problems suited for applications of NPB methods to large molecules. The method approximates the solution to Equation (2.16) for x without

the expense of inverting A by progressively improving a guess of x until the error becomes acceptably small.

$$Ax = b \tag{2.16}$$

The general method is to start by decomposing A into C and E such that $A = C - E$ resulting in Equation (2.17), this technique is called "splitting".

$$\begin{aligned} (C - E)x &= b \\ Cx &= b + Ex \end{aligned} \tag{2.17}$$

Equation (2.17) is recognizably Gauss-Seidel iteration, where $Cx_i = b + Ex_{i-1}$.[24,33] A typical matrix implementation of the SOR method follows the form described in Equation (2.18).

$$\begin{aligned} x_{(k)} &= x^{(i-1)} - w(L + D)^{-1} * \xi_{(i-1)} \\ \xi_{(k)} &= Ax - b \end{aligned} \tag{2.18}$$

where D is the diagonal portion of A, L is the lower triangular portion of A, U is the upper triangular portion of A, $x_{(i-1)}$ is the approximation from the last iteration, x_i is the current approximation of x being computed, $\xi_{(i-1)}$ is the residual from the previous approximation, and $0 < w \leq 2$ is a weighting factor affecting the convergence of the method.[24,34] Equation (2.18) is repeated until $\xi_i < conv$ where $conv$ is some required convergence factor or allowable error in the approximation.

2.1.4 NPB Computational Requirements

The solution of the algebraic system $Ax = b$ dominates the computational and memory requirements of finite difference and finite element methods. Here the computational and storage costs of the SOR algorithm are analyzed as an example iterative solution.

The storage required to represent the matrix problem $Ax = b$ depends upon the representation of the matrix A. Naive implementations would store the zero values in memory, resulting in N^2 storage costs. However, in these cases A is typically a sparse diagonally dominant matrix and so sparse representations will require only $O(N)$ storage if zero values are not stored.

The number of iterations required for SOR to reduce the initial error in x with N elements by a factor of 10^p can be bounded by $\frac{1}{3}p\sqrt{N}$ if an optimal w is used.[24] Each iteration requires $O(N)$ computations, so the approximate solution requires $O(pN^{1.5})$ computations to complete.

The analysis above applies to the abstraction that results from an initial problem. However, it is useful to *directly* couple these computational requirements to the initial problem. The complexity can be evaluated in terms of the length of one edge L of the q dimensional problem with uniform grid spacing h. Trivially $N = (L/h)^q$. By using this definition in

conjunction with the analysis in terms of N, we can derive the computational and memory requirements. Clearly as the dimensionality of the problem increases the computational and memory requirements increase exponentially. In the case of PB solvers, the problem is 3 dimensional, and so in terms of the length of an edge of the domain it costs $O(L/h)^3$ memory to store and $O(p(L/h)^{4.5})$ operations to reduce the error by a factor of 10^p. It is perhaps more intuitive, however, to view this problem in terms of its specific input – a three-dimensional structure made up of M atoms. Each atom has some minimum excluded volume about their centers within which other atoms cannot penetrate. A simple volume argument can be used to show that for the volume of the cubic region $N >= M$ with a correlation ranging from $N \approx M^3$ for a linear arrangement of atoms to $N \approx M$ for a globular arrangement of atoms. Most biological molecules are globular in some sense and so the best approximation in this range for biological molecules in particular is $N \approx M$. However, for the sake of argument, analysis will be performed for both the upper and lower bounds of N relative to M though the distance from the molecule to the edge of the finite domain will be ignored because at most it is a linear multiple of M and drops out of any asymptotic analysis. In the worst case for a cubic region, the molecule is linear such as a short string of DNA. In these cases, $N \approx M^3$ for a cubic lattice so the memory required to store the discretization will be $O(M^3)$ and the number of computations required will be $O(M^{4.5})$. The vast majority of molecules of interest, however, are enzymes which are typically globular proteins, so in these cases $N \approx M$. So in most cases, the memory required to store a cubic discretization for molecular studies will be $O(M)$ and the computational cost will be $O(M^{1.5})$ where M is the number of atoms. While not all finite element or finite difference methods use cubic volumes with uniform grid spacing and a rectangular discretization, many NPB solvers do use this discretization.

2.1.5 NPB Strengths and Weaknesses

NPB methods have two primary strengths: accuracy and error approximation. They have a large body of foundational research in Finite Difference and Finite Element methodologies and error estimation available for their approximations during and after calculation to provide users with a hard upper bound on the error in their potential estimates. These bounds are useful for quality control purposes to determine whether or not a particular NPB approximation should be used for further calculations.

NPB methods suffer from a lack of scalability in terms of the number of points of interest. The method requires $O(M^{1.5})$ computations where M is the number of atoms even for a single point of interest. The entire domain must be solved regardless of the number of points of interest. In most cases, this means that more points are computed and stored than requested. Because of the inherent storage requirements of the methods, this requirement can preclude the calculation of electrostatic regions around very large molecules for even small regions of interest such as binding sites, the molecular surface, or other significant regions. It should also be noted that while the memory required in these implementations

is $O(M)$, it appears in practice to have a high $(40-60)$ prefactor that can dramatically influence the computability of large structures. Finally, for cubic volume decompositions, the method does not scale well with the number of atoms in pseudo-linear molecules.

2.2 The Generailzed Born (GB) framework

The Generalized Born equation is a simple analytical formula used to compute the solvation energy of a molecule. Energy is related to potential in that electrostatic energy is the product of a potential and a charge at a given point in space, therefore the GB equation approximates electrostatic potential in a sense.

The modern GB framework is actually quite a bit more complicated than that which will be presented here. There are multiple ways to approximate the solvation energy that are consistent with the GB methodology. Here, a very simple approach is presented with little detail about the underlying complexities involved in implementing a GB solver. The following citations have much more information about existing modern GB models and their derivations, implementations, and applications.[35–41]

In order to explain the GB framework, the electrostatic contribution to solvation free energy must be defined. Equation (2.19) represents the standard representation of electrostatic energy as a function of potential ($\phi(r)$) and charge concentration ($q(r)$).

$$G^{tot} = \int \phi(r)^{tot} q(r) \tag{2.19}$$

Equation (2.20) represents the relationship between total energy G^{tot}, the energy due to charge interactions in vacuum G^{vac} obtainable through the Coulomb equation ($\frac{q}{d}$), and the energy due to polarization of mobile ions in the solvent dielectric environment G^{pol}.

$$G^{tot} = G^{pol} + G^{vac} \tag{2.20}$$

Equation (2.19) translates to Equation (2.21) for a set of discrete point charges.

$$G^{tot} = \sum_{i=1}^{N} q_i \phi_i^{tot} \tag{2.21}$$

Upon inspection of Equation (2.21), it is clear that only potentials for the discrete set of point charges (e.g. atomic centers) are required to calculate G^{tot}. The Generalized Born approximation estimates the difference in energy of a molecule when it is brought from a vacuum to the external dielectric environment ΔG^{pol}, from Equation (2.20) is $G^{tot} - G^{vac}$. Since energy is the product of a potential with a charge in space, Equation (2.22) can be used to model the change in energy as a difference in potentials at a position in space multiplied by the charge at that point.

$$\Delta G^{pol} = \frac{1}{2}\sum_{i=1}^{N} q_i \left(\phi^{tot} - \phi^{vac}\right) \tag{2.22}$$

The Generalized Born formulation approximates Equation (2.22) as Equation (2.23) where the first term is the well-known Born formula and the second term is the pairwise Coulomb potential contribution of other charges in the molecule, r_{ij} is the distance between atoms i and j, ϵ_w is the ratio of the interior dielectric constant divided by the exterior dielectric constant, and a is the radius of the sphere in which the charge is embedded (typically this corresponds to the Born radius of the atom in question).

$$\Delta G^{pol} = \sum_{i=1}^{N} \frac{q_i^2}{2a_i}\left(\frac{1}{\epsilon_w} - 1\right) + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N} \frac{q_i q_j}{r_{ij}}\left(\frac{1}{\epsilon_w} - 1\right) \tag{2.23}$$

From Equation (2.22) we can see that the potential *at atom centers* can be derived from Equation (2.23) by dividing by the charge of the given atom – resulting in Equation (2.24).

$$\phi_i^{pol} = \frac{\Delta G^{pol}}{q_i} = \frac{q_i}{2a_i}\left(\frac{1}{\epsilon_w} - 1\right) + \sum_{j=1, j\neq i}^{N} \frac{q_j}{r_{ij}}\left(\frac{1}{\epsilon_w} - 1\right) \tag{2.24}$$

2.2.1 GB Computational Requirements

GB quite trivially requires $O(M)$ storage and $O(Mk)$ computations to compute the electrostatic potentials at k atom centers for a molecule consisting of M atoms. To compute solvation energy, all atom centers must have their potential contributions calculated so it is $O(M^2)$. The $O(M)$ storage term is strictly to store the atom positions and charges. For any given calculation, the sum can be computed and then written to disk directly because the potential estimate at any given point is disjoint from the rest of the set.

2.2.2 GB Strengths and Weaknesses

The primary strength of GB lies in its scalability with molecular size. It does not need to store the entire set of points of interest at one time because the solutions are disjoint. Therefore each term can be computed and then stored to disk. Its computational complexity is consistently $O(N^2)$ with regardless of molecular shape due to its analytical nature which can be advantageous when considering a large set of arbitrarily shaped molecules where storing a discretization can be impossible.

Electrostatic potential, by definition is a continuous function of space. It should be noted that the Generalized Born equation is not suitable for calculation of electrostatic potential

because *it defines potential only at the centers of atoms*. This potential is suitable for calculation of free energy, but not for computation of electrostatic fields surrounding the molecule. Therefore, it is rare to see the GB formula used outside of the context of free energies – the purpose for which it was designed.

Chapter 3

Motivation

Electrostatic forces are the strongest in nature. As such, electrostatic potentials are considered in every aspect of molecular modeling including but not limited to: quantum calculations,[42,43] molecular dynamics simulations,[44–48] and rational drug design.[49,50] It is clear that the GB methodology is not capable of determining electrostatic potential beyond the centers of atoms. NPB methodologies accurately approximate electrostatic potential throughout space, but at too great a cost to feasibly study many large system. So there is great call for a computationally facile, consistently scalable approximation of electrostatic potential with low memory requirements.

At the frontiers of molecular size, NPB calculations are often impossible or require the use of sophisticated machinery such as a supercomputer to contain the discretization of the domain. As such, an analytical approximation suitable for electrostatics calculations everywhere in space would significantly benefit scientific exploration of the frontiers of molecular size by escaping the large memory requirements inherent in discrete methods to approximate the solution numerically.

In many cases, electrostatic potential is relevant only for significant regions in the domain near a molecule such as the molecular surface, atom centers, or binding sites. For average molecules, these types of regions tend to comprise a very small percentage of the molecular volume, let alone the volume of the domain necessary to accurately approximate the boundary conditions. In these cases, being able to directly calculate electrostatic potentials in interesting sub-domains would provide a significant computational advantage.

Because of the high computational and storage costs of volumetric integral approaches to numerical approximations of electrostatic potentials, some work has been done to approximate electrostatic potential using a boundary integral approach.[51–53] The boundary integral approach uses significantly fewer elements to approximate the solution near the molecular surface fundamentally by constructing a discretization of the surface and solving a transformed partial differential equation there. This approach suffers less from the scalability

issues that volumetric discretization are known to have. However, for a small set of points of interest, the method still requires the solution of the entire surface – meaning that it does not scale down to one point of interest. The method also suffers from similar problems to the GB model in that the problem is solved at the molecular surface and as points are sampled farther and farther from the surface the accuracy deteriorates due to interpolation error.

If electrostatics calculations are ever to be brought to the desktop computer for any molecule of current research interest, they must be made capable of scaling down in their computational complexity and storage costs with the number of points of interest. Analytical approximations are ideal candidates for this particular purpose because they tend to be defined everywhere in space and solutions require little memory and computational time to approximate. Molecular imaging techniques are now beginning to determine the structures of more and more large complexes at the atomic scale, and the tools that are traditionally used to evaluate the properties of molecules must adapt to the change in scale in order to further understand the properties of these complexes. As science moves from evaluating electrostatic potential at the atomic scale for nanomolecules toward approximating electrostatic potential for microcomplexes it is clear that even for supercomputers, a simple analytical approximation will be needed to analyze the electrostatic properties of these structures and determine how they function.

Chapter 4

Derivation of the analytical model

4.1 Problem set up

Recall that the electrostatic potential $\phi(\vec{r})$ in and around a biomolecule can be computed as the solution of the linearized Poisson-Boltzmann (PB) equation:

$$\begin{aligned} \nabla\epsilon(\vec{r})\nabla\phi(\vec{r}) &= -4\pi\rho(\mathbf{r}) + \kappa^2\epsilon(\mathbf{r})\phi(\mathbf{r}). \\ \lim_{\vec{r}\to\infty}\phi(\vec{r}) &= 0 \end{aligned} \tag{4.1}$$

Where ρ is the (fixed) charge distribution, the dielectric environment is given by the distance-dependent function $\epsilon(\mathbf{r})$ and the effects of monovalent salt enter via the Debye-Hückel screening length of κ^{-1}. One typically assumes a step-function transition between the solvent and solute dielectric environments. In addition, mobile ions are assumed to exist only outside of the so called ion exclusion radius. Under these assumptions, it is more convenient to solve Equation (4.1) separately in the corresponding three regions defined in Figure 4.1. Appropriate continuity requirements are then applied at the manifolds of the regions to obtain the unique, physically correct solution. There are two boundaries at finite distances inherent to the system. The first boundary is the solvent excluded surface (molecular surface) between the low dielectric region of the molecular interior and the high dielectric solvent. This boundary represents the interface between the two dielectrics ϵ_{in} and ϵ_{out}. The second boundary is set a distance b out from the molecular surface, where b is the ion exclusion or Stern radius.

The Poisson-Boltzmann (PB) equation is solved separately in each region defined in Figure 4.1, and the additive constants are chosen to meet the proper continuity requirements at the boundaries.

The fixed charges exist only in region I, and so the corresponding PB equation is:

$$\nabla^2\phi^i{}_I = -\frac{q_i}{\epsilon_{in}}\frac{1}{|\vec{\mathbf{r}} - r_i\hat{\mathbf{e}}_z|} \tag{4.2}$$

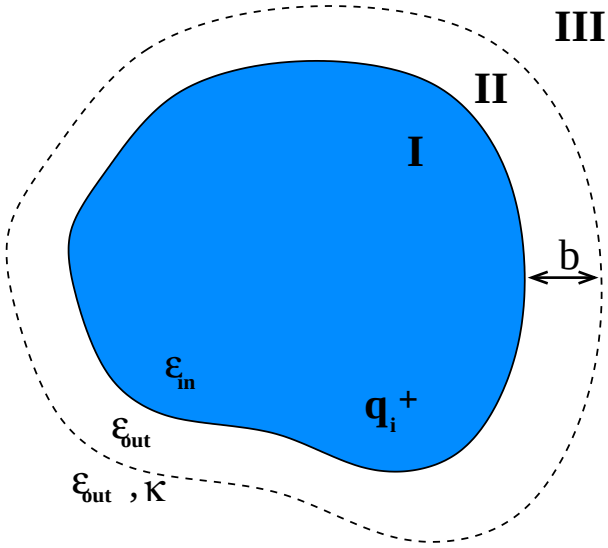

Figure 4.1: The three regions determining PB boundary conditions.

where the point charge density $\rho = q_i\delta(\mathbf{r} - r_i\hat{\mathbf{e}}_z)$ is placed on the z-axis at position r_i.

In region II:

$$\nabla^2\phi^i_{II} = 0 \tag{4.3}$$

In region III:

$$\nabla^2\phi^i{}_{III} = \kappa^2\phi^i{}_{III} \tag{4.4}$$

4.2 The no salt limit

4.2.1 Boundary Values and Geometry Definition

An analytical, closed-form solution of Equation (4.1) is desirable, but for an arbitrary charge distribution inside the molecule it is only obtainable for simple, symmetric shapes. The applicability of such solutions to realistic bio-molecular shapes is not guaranteed *a-priori*, but the early success of this general philosophy is encouraging. As shown in Refs.,[54,55] the approach – termed the *ALPB* in that work – is capable of providing more accurate approximations for biomolecular solvation energy than the famous generalized Born (GB) model. Here, the *exact*, Kirkwood-like[56] infinite-series solution of the PB Equation (4.1) for an arbitrary charge q_i inside a spherical molecule is also used as a starting point. The $\kappa = 0$ case is presented first for clarity. Salt effects will be considered later.

In the $\kappa = 0$ case, there are only two distinct regions I and II, and so $\phi_{II} = \phi_{III}$. These two regions in the spherically symmetric case are: $0 \le r \le A$ and $A \le r < \infty$. placed on the

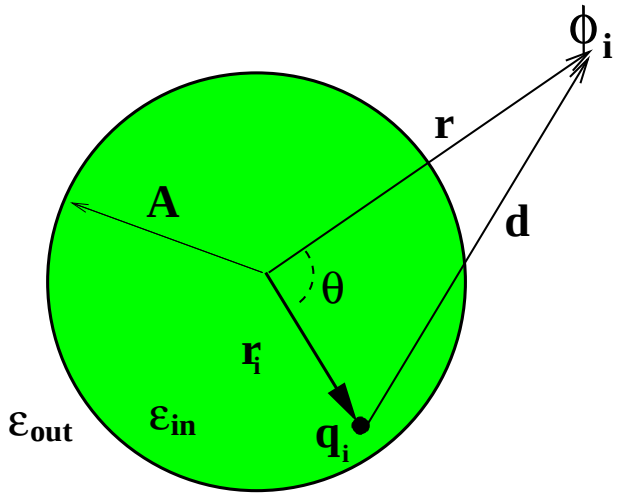

Figure 4.2: Geometric parameters of interest on a sphere

z-axis. The solution of the Poisson equation for region I, Equation (4.2), is the sum of the Coulomb's potential due to the point charge q_i and the reaction field part. Due to azimuthal symmetry (charge q_i is assumed to be placed on the z-axis) the solution depends only on the angle θ through Legendre polynomials $P_l(\cos\theta)$:

$$\phi_I^i = \frac{q_i}{\epsilon_{in}} \frac{1}{|\vec{\mathbf{r}} - r_i \hat{\mathbf{e}}_z|} + \sum_{l=0}^{\infty} \bar{A}_l r^l P_l(\cos\theta) \tag{4.5}$$

Using the following definitions:

$$\begin{aligned} if \quad & r_i > r, \ then \ r_i = r_> \ and \ r = r_< \\ if \quad & r_i < r, \ then \ r_i = r_< \ and \ r = r_>, \end{aligned} \tag{4.6}$$

and the well-known identity,[57]

$$\frac{q_i}{\epsilon_{in}} \frac{1}{|\vec{\mathbf{r}} - r_i \hat{\mathbf{e}}_z|} = \frac{q_i}{\epsilon_{in}} \sum_{l=0}^{\infty} \frac{{r_<}^l}{{r_>}^{l+1}} P_l(\cos\theta) \tag{4.7}$$

the equation for region I is:

$$\phi_I^i = \frac{q_i}{\epsilon_{in}} \sum_{l=0}^{\infty} \frac{{r_<}^l}{{r_>}^{l+1}} P_l(\cos\theta) + \sum_{l=0}^{\infty} \bar{A}_l r^l P_l(\cos\theta) \tag{4.8}$$

No fixed charges are present in region II, which gives:

$$\phi_{II}^i = \sum_{l=0}^{\infty} \frac{\bar{B}_l}{r^{l+1}} P_l(\cos\theta) \tag{4.9}$$

where $\bar{A}$ and $\bar{B}$ are constants determined by the continuity conditions at the boundary $r = A$: $\phi_I(A) = \phi_{II}(A)$ and $\epsilon_{in}\frac{\partial\phi_I}{\partial r}\mid_A = \epsilon_{out}\frac{\partial\phi_{II}}{\partial r}\mid_A$.

The first boundary condition gives:

$$\frac{q_i}{\epsilon_{in}}\sum_{l=0}^{\infty}\frac{{r_i}^l}{A^{l+1}}P_l(\cos\theta) + \sum_{l=0}^{\infty}\bar{A}_l A^l P_l(\cos\theta) = \sum_{l=0}^{\infty}\frac{\bar{B}_l}{A^{l+1}}P_l(\cos\theta) \tag{4.10}$$

Every term above has a Legendre Polynomial dependence in the summation. Applying the orthogonality of the Legendre Polynomial , the equality simplifies to a relation between $\bar{A}_l$ and $\bar{B}_l$.

$$\int_{-1}^{1} P_l(x)P_{l'}(x)dx \;=\; \frac{2}{2l+1}\delta_{ll'} \tag{4.11}$$

or, after integration

$$\bar{A}_l = \frac{1}{A^{2l+1}}(\bar{B}_l - \frac{q_i}{\epsilon_{in}}(r_i)^l) \tag{4.12}$$

The second boundary condition equates the normal components of the electric displacement fields of the two regions.

$$\begin{aligned} -\epsilon_{out}\sum_{l=0}^{\infty}(l+1)\frac{\bar{B}_l}{A^{l+2}}P_l(\cos\theta) \;&=\; \epsilon_{in}[\sum_{l=0}^{\infty} l\bar{A}_l A^{l-1}P_l(\cos\theta) \\ &-\; \frac{q_i}{\epsilon_{in}}\sum_{l=0}^{\infty}(l+1)\frac{{r_i}^l}{A^{l+2}}P_l(\cos\theta)] \end{aligned} \tag{4.13}$$

The orthogonality relation between the Legendre Polynomials is used again to simplify Equation (4.13) thus providing the second relationship between $\bar{A}_l$ and $\bar{B}_l$.

$$\bar{B}_l = \frac{\epsilon_{in}}{\epsilon_{out}}\left[\frac{q_i}{\epsilon_{in}}{r_i}^l - \frac{l}{l+1}A^{2l+1}\bar{A}_l\right] \tag{4.14}$$

Equations (4.12 and 4.14) are solved simultaneously to give independent expressions for $\bar{A}_l$ and $\bar{B}_l$:

$$\bar{A}_l = \frac{q_i}{A^{2l+1}}{r_i}^l\left(\frac{1}{\epsilon_{out}} - \frac{1}{\epsilon_{in}}\right)\frac{1}{1+\frac{l}{l+1}\beta} \tag{4.15}$$

$$\bar{B}_l = q_i{r_i}^l\left(\frac{1}{\epsilon_{out}} - \frac{1}{\epsilon_{in}}\right)\frac{1}{1+\frac{l}{l+1}\beta} + \frac{q_i}{\epsilon_{in}}{r_i}^l \tag{4.16}$$

4.2.2 Region I

Recall that the equation for region I is:

$$\phi_I^i = \frac{q_i}{\epsilon_{in}}\sum_{l=0}^{\infty}\frac{{r_<}^l}{{r_>}^{l+1}}P_l(\cos\theta) + \sum_{l=0}^{\infty}\bar{A}_l r^l P_l(\cos\theta) \tag{4.17}$$

Let $t = \frac{r_<}{r_>}$ then the equation for region I becomes:

$$\phi_I^i = \frac{q_i}{r_> \epsilon_{in}} \sum_{l=0}^{\infty} t^l P_l(\cos\theta) + \sum_{l=0}^{\infty} \bar{A}_l r^l P_l(\cos\theta) \tag{4.18}$$

The following identity can now be used, and will be reused quite often in these derivations:

Identity

$$\sum_{l=0}^{\infty} z^l P_l(\cos\theta) \;=\; \frac{1}{\sqrt{1+z^2-2z\cos\theta}} \tag{4.19}$$

By applying Equation (4.19), Equation (4.18) becomes:

$$\phi_I^i = \frac{q_i}{r_> \epsilon_{in}} \frac{1}{\sqrt{1+t^2-2t\cos\theta}} + \sum_{l=0}^{\infty} \bar{A}_l r^l P_l(\cos\theta) \tag{4.20}$$

Figure 4.2 represents the geometry definition and defines $cos\theta = \frac{r_<^2 + r_>^2 - d^2}{r_< r_>}$. By replacing $\cos\theta$ with this identity and simplifying, the potential in region I, ${\phi_I}^i$ becomes:

$${\phi_I}^i = \frac{q_i}{\epsilon_{in} d} + \sum_{l=0}^{\infty} \left[\frac{q_i r_i^l r^l}{A^{2l+1}} \left(\frac{1}{\epsilon_{out}} - \frac{1}{\epsilon_{in}} \right) \frac{1}{1 + \frac{l}{l+1}\beta} \right] P_l \cos\theta \tag{4.21}$$

Note that $r_>$ and $r_<$ have both fallen out of the equation. Factoring out constants and simplifying yields:

$${\phi_I}^i = \frac{q_i}{\epsilon_{in} d} + \left(\frac{1}{\epsilon_{out}} - \frac{1}{\epsilon_{in}} \right) \frac{q_i}{A} \sum_{l=0}^{\infty} \left[\left(\frac{r_i r}{A^2} \right)^l \frac{1}{1 + \frac{l}{l+1}\beta} \right] P_l \cos\theta \tag{4.22}$$

To simplify the equation, let $s = \left(\frac{r_i r}{A^2}\right)$. Then

$${\phi_I}^i = \frac{q_i}{\epsilon_{in} d} + \left(\frac{1}{\epsilon_{out}} - \frac{1}{\epsilon_{in}} \right) \frac{q_i}{A} \sum_{l=0}^{\infty} \left[s^l \frac{1}{1 + \frac{l}{l+1}\beta} \right] P_l \cos\theta \tag{4.23}$$

This is an exact expression for the spherical case, but still not a closed-form solution. The key next step is to approximate $\frac{l}{l+1} \approx const = \alpha$ for all $l > 0$ in the first of the above infinite sums. Thus,

$$\begin{aligned} \sum_{l=0}^{\infty} \left[\frac{1}{1 + \frac{l}{l+1}\beta} \right] z^l P_l(\cos\theta) \;&\approx\; 1 + \frac{1}{1+\alpha\beta} \sum_{l=1}^{\infty} z^l P_l(\cos\theta) \\ &\approx\; \left[1 - \frac{1}{1+\alpha\beta} \right] + \left[\frac{1}{1+\alpha\beta} \right] \sum_{l=0}^{\infty} z^l P_l(\cos\theta) \end{aligned} \tag{4.24}$$

Where z (in this context) is s.

It was shown earlier that $\alpha = \frac{32(3\ln 2 - 2)}{3\pi^2 - 28} - 1 \approx 0.580127$ results in a fairly accurate approximation for this infinite sum.[54,55] Applying Equations (4.24 and 4.19) to Equation (4.23), one obtains:

$$\phi_I^i = \frac{q_i}{\epsilon_{in} d} + \frac{q_i}{A}\left(\frac{1}{\epsilon_{out}} - \frac{1}{\epsilon_{in}}\right)\frac{1}{1+\alpha\beta}\left[\frac{1}{\sqrt{1+s^2-2s\cos\theta}} + \alpha\beta\right] \tag{4.25}$$

Applying the identity for s, and simplifying yields Equation (4.26) for region I:

$$\phi_I^i = \frac{q_i}{\epsilon_{in} d} - \frac{q_i}{A}\left(\frac{1}{\epsilon_{in}} - \frac{1}{\epsilon_{out}}\right)\frac{1}{1+\alpha\beta}\left[\frac{A^2}{\sqrt{\left(A^2 - r_i^2\right)\left(A^2 - r^2\right) + A^2 d^2}} + \alpha\beta\right] \tag{4.26}$$

4.2.3 Region II

The potential in region II, ${\phi_{II}}^i$ is:

$${\phi_{II}}^i = \frac{q_i}{r}\sum_{l=0}^{\infty}\left(\frac{r_i}{r}\right)^l\left[\frac{1}{\epsilon_{in}} - \left(\frac{1}{\epsilon_{in}} - \frac{1}{\epsilon_{out}}\right)\frac{1}{1+\frac{l}{l+1}\beta}\right]P_l\left(\cos\theta\right) \tag{4.27}$$

To simplify the notations, let $t = \frac{r_i}{r}$. Then

$${\phi_{II}}^i = -\frac{q_i}{r}\left(\frac{1}{\epsilon_{in}} - \frac{1}{\epsilon_{out}}\right)\sum_{l=0}^{\infty}\left[\frac{1}{1+\frac{l}{l+1}\beta}\right]t^l P_l\left(\cos\theta\right) \;+\; \frac{q_i}{r}\frac{1}{\epsilon_{in}}\sum_{l=0}^{\infty}t^l P_l\left(\cos\theta\right) \tag{4.28}$$

Applying Equations (4.24 and 4.19) to Equation (4.28): yields the following *closed form* approximate expression for ϕ_{II}:

$$\begin{aligned}{\phi_{II}}^i \;&=\; -\frac{q_i}{r}\left(\frac{1}{\epsilon_{in}} - \frac{1}{\epsilon_{out}}\right)\frac{1}{1+\alpha\beta}\left[\frac{1}{\sqrt{1+t^2-2t\cos\theta}} + \alpha\beta\right] \\ &+\; \frac{q_i}{r}\frac{1}{\epsilon_{in}}\frac{1}{\sqrt{1+t^2-2t\cos\theta}}\end{aligned} \tag{4.29}$$

Figure 4.2 represents the geometry of the system and defines $\cos\theta = \frac{{r_i}^2 + r^2 - d^2}{2 r_i r}$, with d being the distance from the charge to test point. Therefore:

$$\frac{1}{\sqrt{1+t^2-2t\cos\theta}} = \frac{r}{d} \tag{4.30}$$

and:

$$t\cos\theta = \frac{{r_i}^2 + r^2 - d^2}{2r^2} \tag{4.31}$$

The substitutions reduce Equation (4.29) to

$$\phi^i_{II} = -\frac{q_i}{r}\left(\frac{1}{\epsilon_{in}} - \frac{1}{\epsilon_{out}}\right)\frac{1}{1+\alpha\beta}\left[\frac{r}{d} \; + \; \alpha\beta\right] \; + \; \frac{q_i}{d}\frac{1}{\epsilon_{in}} \tag{4.32}$$

Since β is dependent on both ϵ_{in} and ϵ_{out}, a two variable dependence is employed by letting $\epsilon_{in} = \beta\epsilon_{out}$. After simplification,

$$\phi^i_{II} = \frac{q_i}{\epsilon_{out}}\frac{1}{1+\alpha\beta}\left[\frac{1+\alpha}{d} - \frac{\alpha(1-\beta)}{r}\right] \tag{4.33}$$

The above formula approximates the electrostatic potential anywhere within or around the spherical molecule. Its accuracy relative to the exact solution of the PB equation is discussed in Chapter 5. To use the above formula for realistic non-spherical structures one needs a meaningful definition of r, which is the distance to the sphere's center in the spherical case. For non-spherical molecules $r = A + p$ is used, where A is the effective electrostatic size (radius) of the molecule defined earlier,[54,55] and p is the distance from the test point to the molecular surface.

4.3 Introducing monovalent salt dependence

4.3.1 Boundary Values and Geometry Definitions

Equation (4.33) behaves as the sum of two point charge potentials proportional to $\frac{1}{d}$ and $\frac{1}{r}$ respectively. This realization guides the process of introducing an explicit salt dependence into Equation (4.33). This will be done within the Debye-Hückel limit. As before, the equations for the case of perfect spherical symmetry are derived, and then tested on realistic structures.

4.3.2 Region III

A point charge potential in the presence of an evenly distributed ionic solution has the form of a Yukawa potential: $\sim \frac{e^{-\kappa r}}{r}$, where κ^{-1} is the Debye screening length. Therefore, it is natural to try the following ansatz:

$$\phi^i_{III} = \bar{C}\frac{e^{-\kappa r}}{r} + \bar{D}\frac{e^{-\kappa d}}{d} \tag{4.34}$$

Where $\bar{C}$ and $\bar{D}$ are constants to be determined.

There are now three unknown constants whose values can be determined by matching the boundary conditions and by considering the behavior of Equation (4.34) in the limiting

cases where the exact solution of the corresponding linearized PB equation is known. A combination of both methods is used. Namely, if the charge q_i is located at the center of the sphere, and thus the center of the spherical coordinate system, then the exact solution of Equation (4.4) in region III must equal:

$$\phi^i_{III}(d=r) = \frac{q_i}{\epsilon_{out}} \frac{1}{1+\kappa(A+b)} \frac{e^{-\kappa(r-A-b)}}{r} \tag{4.35}$$

which gives us one equation of the three needed. Next, the continuity of the tangential components of the electric field are used at the boundary: $\nabla\phi^i_{II}|_{A+b} = \nabla\phi^i_{III}|_{A+b}$ which yields two separate relations, one for each tangential coordinate variable. Due to the simplicity of the θ-component, one uses:

$$\frac{\partial\phi^i_{II}}{r^2 sin(\theta)\partial\theta}|_{A+b} = \frac{\partial\phi^i_{III}}{r^2 sin(\theta)\partial\theta}|_{A+b} \tag{4.36}$$

The continuity of the potential itself gives $\phi_{II}|_{A+b} = \phi_{III}|_{A+b}$. Using the three constraints above yields the following independent expressions for the constants in Equations (4.41 and 4.34):

$$\bar{C} = -\frac{q_i}{\epsilon_{out}} \frac{\alpha(1-\beta)}{1+\alpha\beta} \frac{e^{\kappa\acute{r}}}{1+\kappa\acute{r}} \tag{4.37}$$

$$\bar{D} = \frac{q_i}{\epsilon_{out}} \frac{1+\alpha}{1+\alpha\beta} \frac{e^{\kappa\acute{d}}}{1+\kappa\acute{d}} \tag{4.38}$$

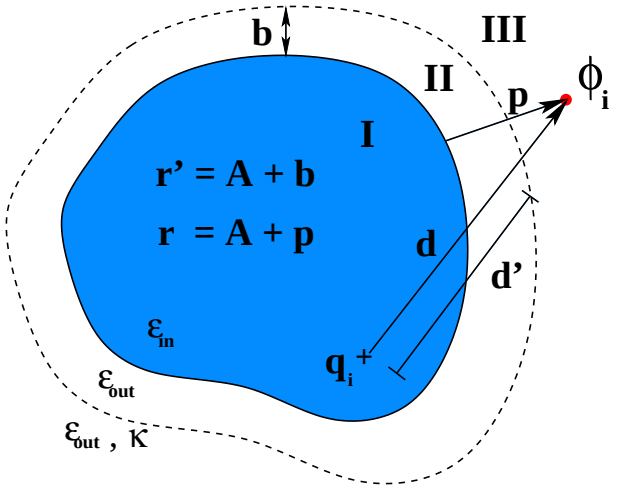

Figure 4.3: Geometric parameters of interest

Figure 4.3 represents the geometry of the system with salt and defines $\acute{r} = A + b$ and $\acute{d} = \sqrt{r_i{}^2 + (A+b)^2 - 2r_i(A+b)cos(\theta)}$. Using these representations, the the final analytical

form for the potential in region III (solvent with mobile ions) is:

$$\phi^i_{III} = \frac{q_i}{\epsilon_{out}} \frac{1}{1+\alpha\beta} \left[\frac{1+\alpha}{1+\kappa\acute{d}} \frac{e^{-\kappa(d-\acute{d})}}{d} - \frac{\alpha(1-\beta)}{1+\kappa\acute{r}} \frac{e^{-\kappa(r-\acute{r})}}{r} \right] \tag{4.39}$$

4.3.3 Region II

In region II, where no salt penetrates, the solution of the Poisson equation looks exactly like Equation (4.33), except that a yet unknown constant, $\bar{E}$, is now added to ensure the continuity of the potential at the boundary between regions II and III.

$$\bar{E} = \frac{q_i}{\epsilon_{out}} \frac{1}{1+\alpha\beta} \left[\frac{1+\alpha}{\acute{d}} \left(\frac{1}{1+\kappa\acute{d}} - 1 \right) - \frac{\alpha(1-\beta)}{\acute{r}} \left(\frac{1}{1+\kappa\acute{r}} - 1 \right) \right] \tag{4.40}$$

$$\phi^i_{II} = \frac{q_i}{\epsilon_{out}} \frac{1}{1+\alpha\beta} \left[\frac{1+\alpha}{d} - \frac{\alpha(1-\beta)}{r} \right] + \bar{E} \tag{4.41}$$

4.3.4 Region I

To ensure continuity at the boundary between regions I and II, $\bar{E}$ as is included in region II is added to produce the form in Equation (4.42):

$$\phi^i_I = \frac{q_i}{\epsilon_{in} d} - \frac{q_i}{A} \left(\frac{1}{\epsilon_{in}} - \frac{1}{\epsilon_{out}} \right) \frac{1}{1+\alpha\beta} \left[\frac{A^2}{\sqrt{(A^2 - r_i^2)(A^2 - r^2) + A^2 d^2}} + \alpha\beta \right] + \bar{E} \tag{4.42}$$

Equation (4.39), together with Equations (4.41 and 4.42) and the additive constant from Equation (4.40) define the electrostatic potential at every point in space outside of the molecule. There are four parameters with the unit of length, r, d, d' and r' that enter the equation. The geometric interpretation of d' and r' is shown in Figure 4.3. Note that Equation (4.39) is a sum of scaled Yukawa potentials and therefore is a solution of the PB equation. Of course, it is only an approximate solution even for a spherical case. At first glance, the existence of such a solution, not equal to the exact one, may appear to contradict the uniqueness theorem for the PB equation. However, notice that the approximate solution satisfies only some but not all of the boundary conditions. In fact, a detailed analysis (not presented here) shows that the solution based on the simple ansatz Equations (4.34 and 4.41) can not simultaneously satisfy all of the boundary conditions, which resolves the apparent paradox. An extensive testing of the approximate solution on realistic structures is presented in Chapter 5.

4.4 ALPB Implementation (GEM) and Features

4.4.1 GEM Implementation

In addition to formulating the approximate analytical model for computing electrostatic potential for macromolecules, we have constructed a sandbox for exploring electrostatic phenomena at the molecular scale called GEM. It is structured to be highly available to developers and users alike. Functionality is accessible through linking to C libraries, command line parameters, and through the visual interface. The implementation is decomposed into three major libraries: file I/O, calculation, and visualization. The package is built and configured under the GNU tools "autoconf" and "automake" to aid in cross platform development and distribution.

The file I/O library contains algorithms to read: PQR files, MSMS surface files, AVS phi maps from `MEAD`, and `DelPhi` phi maps. It also contains algorithms to write: xyzr files, tga image files, AVS phi maps in `MEAD` format, and `DelPhi` phi maps. Correctly reading AVS phi maps and `DelPhi` phi maps is not a trivial task in and of itself. Having the ability to read and write these formats available in an open-source library will significantly ease further development of open source algorithms intended to interact with or replace `MEAD` or `DelPhi` in a given scenario.

The calculations library contains numerous algorithms to perform useful calculations such as: approximating the analytical potential described in this paper, estimating the electrostatic radius of a molecule,[54] extrapolating bond information from inter-atomic distances and electrostatic radii, determining the color of atoms by their type, sorting spatial locations by radix, and sampling a uniform cubic lattice. All of these methods are of fundamental importance to the computation and display of electrostatic potential or the display of molecular structure in general.

The visualization library contains two major components: dialogs and drawing. The dialogs component contains a set of novel dialogs developed to aid in the design of further user interactions including an extensible modified open file dialog, a specialized status dialog that makes use of Motif timer callbacks to provide pseudo-threads while monitoring the status of long processes, and a "tellUser" dialog to provide a simple interface for a notification system.

These libraries have been leveraged to produce four example programs: gem – the electrostatics calculation and viewing program, gridmath – a simple program to apply various unary operators (*, +, -, /) to all members of two uniform grid files, printgrid – a simple program to produce human-readable grid files from standard binary files, and diffgrids – a program to compute the RMSD error and max error between two uniform grid files for the purposes of testing the accuracy of various approximations against a baseline.

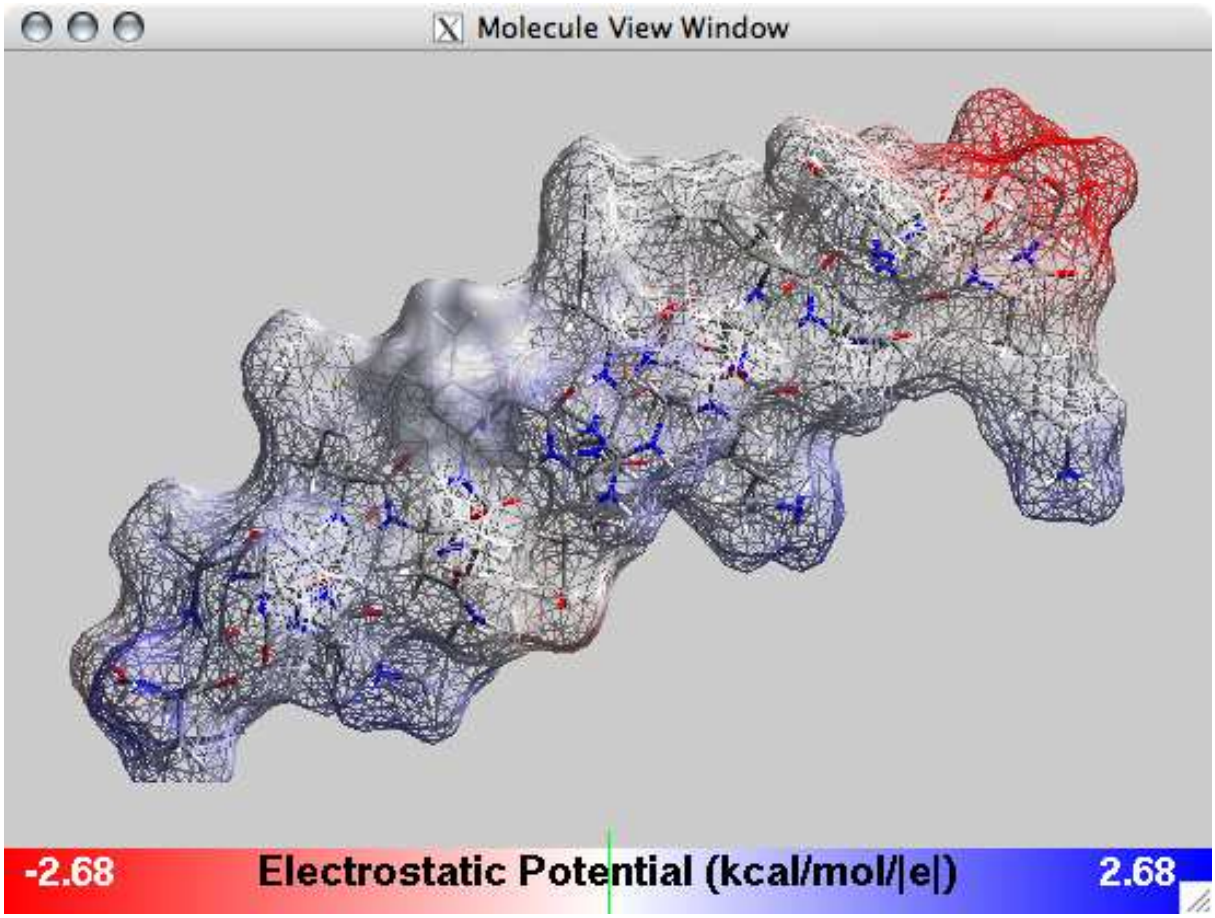

Figure 4.4: A screen shot demonstrating some view features of GEM.

4.4.2 GEM Computational Requirements

In order to compute P points of interest about a molecule consisting of N atoms, GEM requires $O(N)$ memory. There is no need to continue to store points after they are calculated, and so they can be output directly as the algorithm calculates the potential at the next point. This attractive feature sets it apart from packages based on NPB methodology where the entire domain must be solved in order to provide approximations of even 1 point of interest. The freedom from this limitation is a crucial practical advantage when analyzing the electrostatic properties of such molecules. As an example, the RAM required by GEM to store the potential map of the surface of the TRSV virus consisting of $651,544$ surface grid points is only 30 Mb. This is an insignificant overhead for even the modest desktop computer. The corresponding requirements are orders of magnitude larger for the NPB solutions. For example, in order to store a finite mesh (at a typical resolution of 0.25 ångströms per grid point) of floating point values for a molecule of the size of TRSV virus,

about 1200^3 $(1,440,000,000)$ separate grid points would be needed, requiring a minimum of nearly 13 GB of memory assuming 8 byte double representation of electrostatic potential per mesh point.

Due to the additivity of the electrostatic potential, GEM must compute the contributions from each charge in the molecule to each point of interest; without any further approximations or optimizations its time complexity is $O(MP)$ where M is the number of atoms in the molecule and P is the number of points of interest. The algorithm scales well with the number of points of interest or the number of charges in the molecule. If $P \simeq M$ then GEM requires between $O(M^2)$ and $O(M^4)$ computations to approximate the potential field in the volume depending upon the shape of the molecule.

4.5 GEM Strengths and Weaknesses

GEM has similar strengths to the GB method with the added strength of being capable of approximating electrostatic potential everywhere in space. Its low memory requirements are particularly advantageous when computing electrostatic potential for macromolecules where conventional computers are incapable of containing a suitable discretization for NPB methods. Its computational scalability extends to smaller molecules belonging to large search domains for high throughput computational screening.

The primary weakness of GEM is the unbounded error in its approximations. The vast majority of approximations fall within thermal noise but that does not guarantee a suitably accurate solution in every case. Ideally, this error will be bounded by further research or a hybrid numerical/analytical solution will be developed to limit and determine the error while retaining the computational facility of the analytical approach presented in this work.

Chapter 5

Validation of the analytical approach

The analytical approximation is validated through three primary methods: testing against the exact solution for a perfect sphere, visually examining results relative to numerical solutions on a variety of molecular shapes, and finally analysis of individual vertex errors for a set of 580 small biomolecules relative to available NPB approximations.

5.1 Testing against the exact solution on a sphere

The *exact* solution of the PB equation for spherical geometry is used, given by Equation (4.28), to test the accuracy of the approximate solution directly. The existence of an exact expression for ϕ also allows us to estimate the accuracy of the numerical PB procedures used here as reference. A partial sum of the first $N = 200$ terms in the infinite series in Equation (4.28) is taken to represent the exact solution; the partial sum converges to machine precision when N is approximately 100 for the spherical geometry used.

In Figure 5.1, the exact error of Equation (4.33) is represented by the red line; the blue line represents the exact error of `DelPhi`,[2,32] a representative numerical solution to the PB equation. The approximate analytical expression Equation (4.33) is quite accurate on a sphere. In fact, it is more accurate than the NPB solution for most points on the surface of the sphere.

The 0.25 Å resolution used for the spherical test case provides insight into the accuracy of the ϕ_{NPB} being used as a reference. The NPB solutions based on 0.5 Å grids used here as a reference for the realistic bio-molecular test are expected to be *less* accurate than 0.25 Å solutions used in the spherical case because as the resolution of the discretization increases the accuracy of numerical approximations increases. It is therefore probable that in some cases, the disagreement between the analytical model and the NPB reference is due to inaccuracy in the reference as opposed to the inherent inaccuracies of the proposed method.

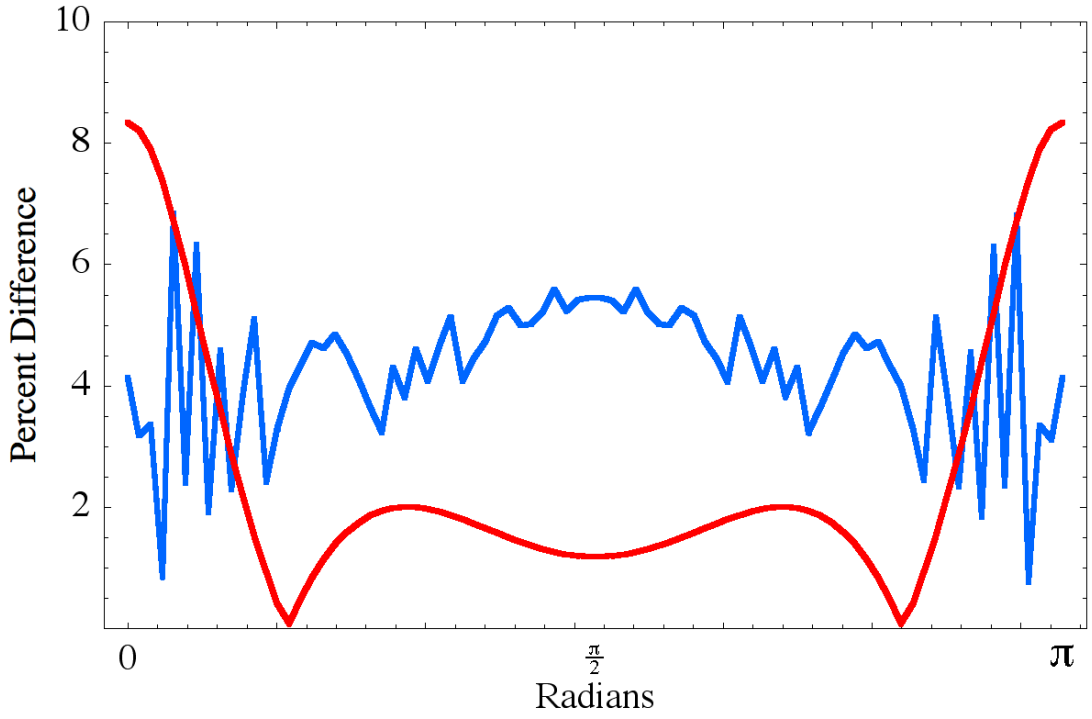

Figure 5.1: Percent error of NPB and GEM potential estimates on a sphere.

The use of 0.25 Å grid spacing would result in prohibitively large memory requirements for a substantial subset of the 580 molecule test set.

5.2 Testing against NPB on realistic biomolecular shapes

Figure 5.2 contains images of the electrostatic potential computed at the surface of various biomolecules for which the analytical solutions based on symmetric shapes are expected to deviate most from the NPB reference described in Appendix D. Right column: analytical potential. Left column: numerical reference. A continuous color scale is used to represent the potential, from red (-1.8 kCal/mol/$|e|$) to white (0) to blue ($+1.8$ kCal/mol/$|e|$). The structures used are: the Alzheimer's disease amyloid A4 peptide (top), human apolioprotein C-II (middle), the lysozyme (bottom).

No *exact* solutions of the PB equation are available for realistic biomolecular shapes; therefore the accepted approximate numerical solutions are used here to test the analytical approximations on a set of 580 representative biomolecules listed in Appendix A.[58] The error is estimated as $(\phi_{GEM} - \phi_{NPB})$ over a combined total of $9,384,884$ points sampled as described in Appendix F. For comparison, the "error" distribution $(\phi_{NPB} - \phi_{NPB})$ is also calculated

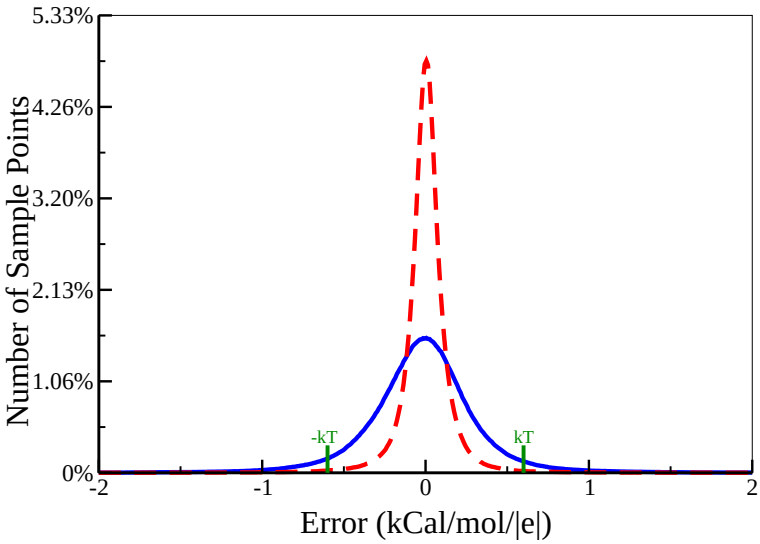

Figure 5.2: Absolute vertex error distribution curves between ϕ_{GEM} ϕ_{NPB}.

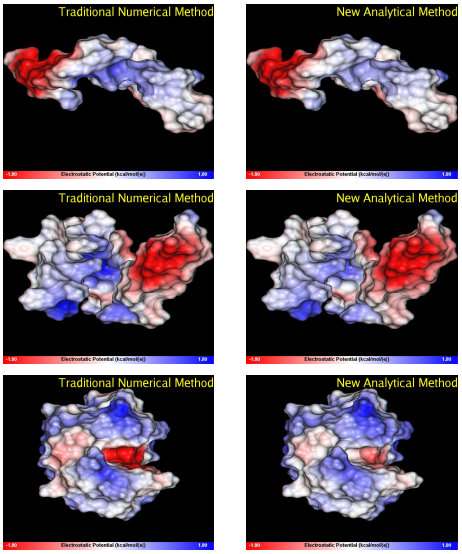

Figure 5.2: Electrostatic potential computed at the surface of various biomolecules.

for two popular finite-difference PB solvers `DelPhi`[2,32] (the baseline NPB approximation used for estimating the error of the analytical approximation) and `MEAD`[59] which may serve as a practical scale to which the $(\phi_{GEM} - \phi_{NPB})$ deviations can be compared. To obtain probability distributions, errors are categorized into 1000 equidistant ranges of width 0.005 kCal/mol/$|e|$ ranging from -2.0 to 2.0 kCal/mol/$|e|$ and the number of points in each classification are divided by the total number of sample points in the set. The standard deviations of the distributions are 0.41 for $(\phi_{GEM} - \phi_{NPB})$ and 0.19 for $(\phi_{NPB} - \phi_{NPB})$.

Figure 5.2 demonstrates the strong overall agreement between ϕ_{GEM} and ϕ_{NPB} for the biomolecular sample set. The blue curve represents the distribution of $(\phi_{GEM} - \phi_{NPB})$ while the red curve represents the distribution of $(\phi_{NPB} - \phi_{NPB})$ which should ideally be 0 but is not in practice. Both of these quantities are computed at every vertex point about the triangulated solvent excluded surface of each of the 580 representative biomolecules used in the test set. The vast majority (91.92%) of vertex errors fall within thermal noise (kT per unit charge). Note that ϕ_{NPB}, ϕ_{GEM}, and ϕ_{exact} all asymptotically approach 0 as the distance to the surface (and consequently distance to charge sources) approaches ∞, so the above error is expected to decrease as one steps further away from the molecular surface. In this sense, the absolute error shown likely corresponds to an upper bound on the error everywhere in the solvent space.

It is possible in principle that the distribution of average errors per molecule would be fundamentally different from the distribution of errors in Figure 5.2 because this classification was discarded during the consideration of absolute vertex error on a per vertex basis. Therefore, all 8.08% of the errors outside of thermal noise might feasibly belong to a class of molecules

whose characteristics defeat the analytical approximation. The distribution of the average magnitude of vertex errors per molecule is determined as a means of capturing the relationship between molecular characteristics (captured by the identities of molecules) and error. The average absolute vertex error on a per molecule basis is computed as:

$$Err_j = \sum_{i=0}^{n_j} \frac{|\phi_{GEM}^{(i)} - \phi_{NPB}^{(i)}|}{n_j} \tag{5.1}$$

where j is the structure index, n_j is the number of vertex points for structure j, and i is the vertex index. As seen in Figures 5.3 and 5.3, molecular identity plays little role in the determination of vertex error as can be seen by the fact that almost all average magnitudes of error fall within $0.41 kCal/mol/|e|$, the standard deviation of the overall vertex error curve. Figure 5.3 also offers some insight into what kinds of vertex errors can be expected on average for a random realistic biomolecule. Given that Figure 5.2 indicates that 91.92% of vertices fall within thermal noise, it is reassuring to see that all average vertex errors for all molecules also fall within thermal noise.

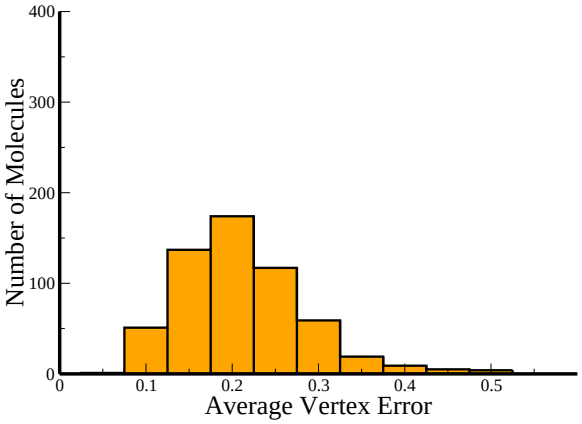

Figure 5.3: Average surface potential differences between PHI_{NPB} and PHI_{GEM}.

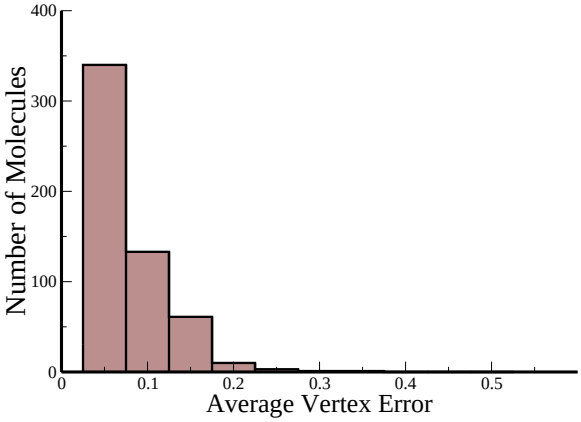

Figure 5.3: Average surface potential differences between $PHI_{NPB}^{(}1)$ and $PHI_{NPB}^{(}2)$.

The reasonable performance of the analytical approach to compute the electrostatic potential around realistic biomolecules is not completely unexpected; after all, success of the use of simple shapes in a related problem – deriving approximate expressions for biomolecular solvation energy – has had a long history.[55,56,60] Figure 5.2 demonstrates the noteworthy agreement between the analytical electrostatic potential introduced here and the NPB reference for particularly irregular molecular shapes. For some of the more spherical molecules the analytical solutions might even be more accurate than the corresponding approximate numerical solutions obtained with commonly used parameter settings. This is because for a perfect sphere, the analytical approach is indeed more accurate than the NPB approach at grid resolution of 0.25 Å. Therefore, some of the deviations between ϕ_{GEM} and ϕ_{NPB} seen

in Figure 5.2 may be due to inaccuracies of ϕ_{NPB} itself. Finally, one should also consider these errors within the already approximate implicit solvent PB framework being solved. Recent studies show that the differences between explicit and implicit solvent representations themselves are not negligible.[61]

Chapter 6

Applications

6.1 Surface Potential of the TRSV Viral Capsid

The Tobacco Ringspot Virus (TRSV) belongs to the Comoviridae family of the Genus Nepovirus. The TRSV satellite sequence was determined in 1969 and brought forth the concept of a virus having its own parasite that is dependent on the host virus for encapsidation.[62] The TRSV virus is believed to represent a very simple precursor to the nepovirus, picornavirus, and comovirus families because its capsid is made of a single protein subunit, it has no lipid coat, and its polyproteins have no cleavage sites.[63] Despite its considerable evolutionary age, there are only 2 known satellite sequences.[64] This is especially puzzling considering the observed mutation rate of the satellite RNA.[65] These observations suggest the presence of a powerful and specific selection agent in the TRSV life cycle, likely the capsid itself. Further experimental evidence suggests that the mechanism is structure-based: circular satellite RNA with the same sequence as the native one is rejected by the capsid.[66,67] The precise mechanism underlying the selectivity of the TRSV capsid for its RNA is still unknown. Since electrostatics plays a major role in protein - nucleic acid interactions, taking these effects into account is expected to be critical for solving the puzzle.

The capsid serves a dual purpose, one for the exterior and one for interior. The outside interacts with the environment during the various stages of the virus' life cycle. As the virion moves from the vertical vector to the cytoplasm of a tobacco plant cell to the plant sap, it experiences multiple pH values which uniformly changes the outside electrostatic potential. The inside of the capsid has a repeated area of distinct, positive electrostatic potential. These areas are located at the center of a 5 protomer subunit (pentamer) and could serve as an RNA binding site. Details about where the capsid structure is obtained can be found in Appendix A.

6.1.1 The Outer Surface

The entire TRSV capsid is protonated for a broad range of pH values as described in Appendix B. The pH for which the overall charge of the capsid is computed to be 0 (isoelectric point) is 7.15. The resulting PQR structure files differ by the placement of Hydrogen atoms (positive charges) at protonation sites throughout the surface of the molecule. The molecular surface of each structure is triangulated by the program `MSMS` with a resolution of 2.5 Å per vertex; then electrostatic potential is computed by the analytical approximation presented in this work 3 Å outside the molecular surface to determine if and how pH affects the surface potential of this massive structure.

Figure 6.1 is a pictographic representation of the outer surface of the TRSV viral capsid color-coded according to the computed electrostatic potential on its surface. The computations are performed at a constant salt concentration (0.15 M) and three different pH values shown under each structure. A continuous color scale is used, from red (corresponding to -4.68 $kcal/mol/|e|$), to blue (+4.68 $kcal/mol/|e|$). The regions of zero potential are shown in white. The arrow points to an outcropping that corresponds to the center of the pentamer shown in Figure 6.2.

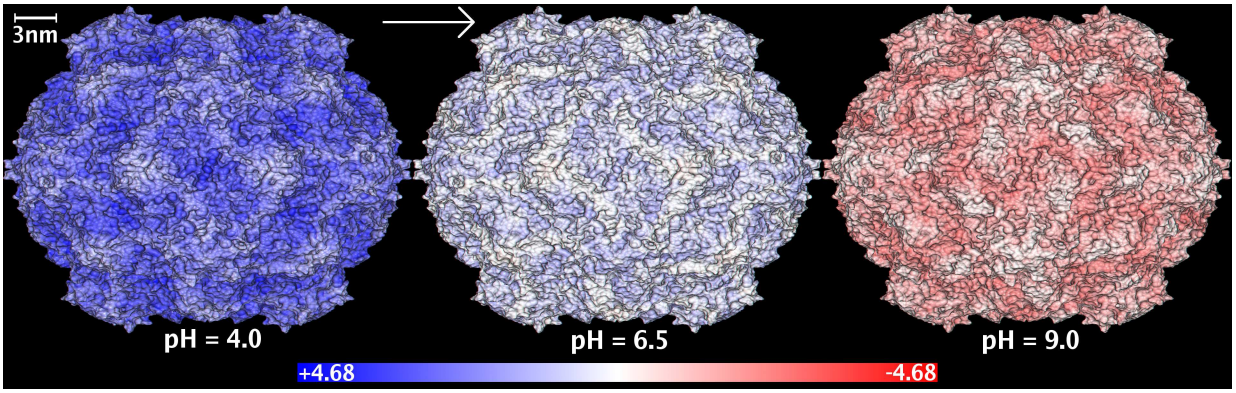

Figure 6.1: Electrostatic potential around the outer surface of the TRSV viral capsid.

Infection usually occurs through the vertical vector: *Xiphinema americanum*.[68,69] The nematodes acquire the virions during feeding on an infected plant. These virions become caught in the stylet extension, anterior esophageal lumen, and esophageal bulb.[70,71] From these regions, the virions are deposited in a healthy plant cell during a later feeding.[68] The release of the virions from these regions in the nematode is proposed to initiate through a pH change due to salivation by the nematode.[71,72] The absence of strong electrostatic repulsion in the capsid leading to its structural stability in the neutral pH range makes sense biologically; the virion is known to use the sap of a healthy tobacco plant of pH 6.2 as a means for circulating through the plant in attempt to find other mechanically damaged cells to infect.[73] The build up of a fairly uniform negative charge across the capsid at high pH,

Figure 6.1 (right panel), diminishes its stability due to Coulomb repulsion. This is consistent with the theorized swelling of the capsid at pH greater than 8.0.[69] In living cells, swelling might be the mechanism causing the virion to release its RNA in cell compartments that have high pH.

6.1.2 The Inner Surface

Figure 6.2 is a representation of the inner surface of the pentamer subunit color-coded according to the computed electrostatic potential. The computations are performed at three different pH values shown under each structure with a constant salt concentration of 0.15 M. A continuous color scale is used from red (corresponding to -4.00 $kcal/mol/|e|$) to blue (+4.00 $kcal/mol/|e|$). The regions of zero potential are shown in white. The proposed RNA binding pocket is seen as a bright blue spot in the center of the structure which remains distinct throughout the entire pH range. The primary source of this region of intense positive potential is a "ring" of ten arginines. Each protomer of the pentamer provides two sequential arginines (residues 453 and 454) which are in close proximity to each other in the pentamer structure.

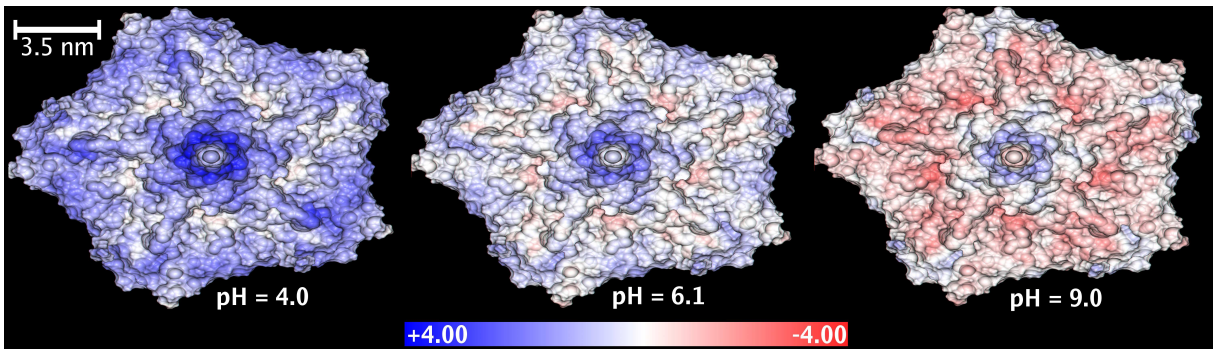

Figure 6.2: Electrostatic potential around the inner surface of the TRSV capsid.

The pocket resembles a narrowing dome: near the surface it is approximately 50 Å wide, it narrows deeper in to a more cylindrical shape with a diameter of roughly 20 Å. The entire site from top to bottom is roughly 40 Å deep. This pocket might represent the RNA binding site and play a key role in the observed high selectivity of the TRSV capsid for its RNA. The positively charged arginine ring attracts RNA; geometry determines which RNAs are structurally compatible with it. Namely, there are 3 sequence-dissimilar RNA particles involved in TRSV infection: RNA-1, RNA-2, and the 359 or 360 nt short satellite sequence RNA-s. Mature TRSV capsids are known to contain almost exclusively either one of these 3 RNA particles or nothing; small changes in the RNA sequence are known to preclude the corresponding particle to be captured by the capsid.[66,67] Moreover, it has been shown experimentally that circular RNA with the sequence identical to RNA-s fails to be

encapsidated.[66,67] This peculiar phenomenon provides the key support for the hypothesis that the determining factor for the strong selectivity for the RNA encapsidation is structural in nature.

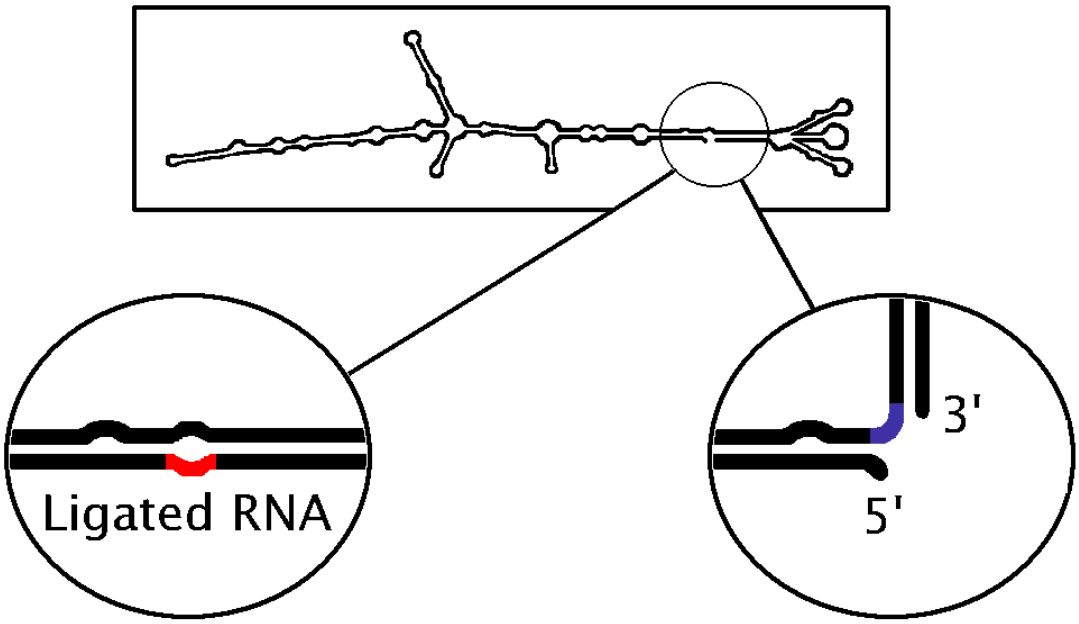

Figure 6.3: The predicted secondary structure of TRSV satellite RNA in two conformations.

The evidence comes from an analysis of structural models of the native and circular RNA-s. While both RNA-1 RNA-2 are too large (7514 nt[74] and 3929 nt[75] respectively) for the available theoretical tools to make confident predictions of their 3D or even secondary structures, the 359 nt long RNA-s sequence is short enough for its secondary structure to be computed with confidence.[76] To explain why the circular form of RNA-s may be structurally incompatible with the RNA binding pocket described above, the differences between the structures of RNA-s in its native (unligated) and circular forms produced by ligation of its 3' and 5' ends are of particular interest. As seen in Figure 6.3, the 3' and 5' ends of RNA-s are adjacent, which suggests that no re-arrangement of the secondary structure occurs upon ligation that forms the circular RNA-s. Therefore, the differences between the native and circular RNA-s may only come from their corresponding 3D structures. Indeed, the native RNA-s is likely to bend around the single-stranded section (marked by blue in the bottom right of Figure 6.3) opposite to the break between its 3' and 5' ends. A wedge-like local 3D structure can form that can fit into the binding pocket in Figure 6.2; the $\sim$ 20 Å diameter of the narrow part of the "dome" is just enough to accept the 1 or 2 unpaired bases on the 5' end of RNA-s. This is in contrast to the ligated RNA-s structure where this same section (marked by red in the bottom left of Figure 6.3) becomes double-stranded with much less flexibility – the corresponding 3D structure is likely to remain "straight" around this section. Fitting it into

the binding pocket will be much less favorable energetically than in the case of the native RNA-s.

6.2 Using GEM to Improve the Performance of NPB solvers

All finite element and finite difference methods for solving the Linear Poisson-Boltzmann equation have one fundamental problem. The Dirichlet boundary condition $\phi(\infty) = 0$ is defined at an infinite distance from the source charge, and the elements must eventually reach the boundary. This means that, strictly speaking, the boundary of the finite field must reach infinity to converge upon the exact solution. As a means of addressing this problem, NPB solvers make use of some simple analytical solution such as q/d where q is the charge and d is the distance from the point to the charge to approximate electrostatic potential at a non-infinite distance from the molecular surface. Initialization of the boundaries by such an approximation is a weakness of all NPB methods, and is simply a byproduct of the need to define a finite domain. Coarser approximations of electrostatic potential require larger distances from the molecular surface to the boundary of the discretization, and so they require more memory and computational time to approximate electrostatic potential for the same molecule. Clearly, a highly accurate analytical approximation of electrostatic potential could significantly decrease the ratio V/M of the volume of the approximation relative to the volume of the molecule for which the approximation is computed without introducing large errors or convergence issues due to inaccurate boundary values. The analytical approximation presented in this work could feasibly be used by NPB solvers as an initialization of boundary potentials or even to initialize the entire volume for use with iterative matrix solutions to improve the rate of convergence.

Most large scale iterative methods make use of some intelligent choice for an initial guess of the solution to improve convergence of the iterative matrix solution to the problem. While it may make sense for generalized PDE and ODE solvers to use best average case initializers, specialized applications like NPB solvers may benefit more from the use of a specialized initializer using domain-specific knowledge to formulate an initial guess in these specific cases. One ideal such specialized initial approximation might be a rapid analytical approximation of electrostatic potential. Though an initializer can not improve the memory required to contain the discretization, they may improve the computational cost to iteratively solve these problems.

One application of the analytical approximation presented in this work may be as an initial guess for iterative approximate solutions. GEM generates a reasonable approximation to the PB problem, therefore, it should be an ideal initial guess to improve convergence for an iterative method like SOR. Revisiting the SOR analysis from Chapter 2.1.4, the total number of iterations (each consisting of O(N) operations) to *reduce the error* by a factor

of p is $\frac{1}{3}p\sqrt{N}$ where N is the cardinality of the unknown vector. Therefore, by reducing p significantly, we might also greatly improve the time complexity of this computation.

6.2.1 Conceptual validation on a spherical geometry

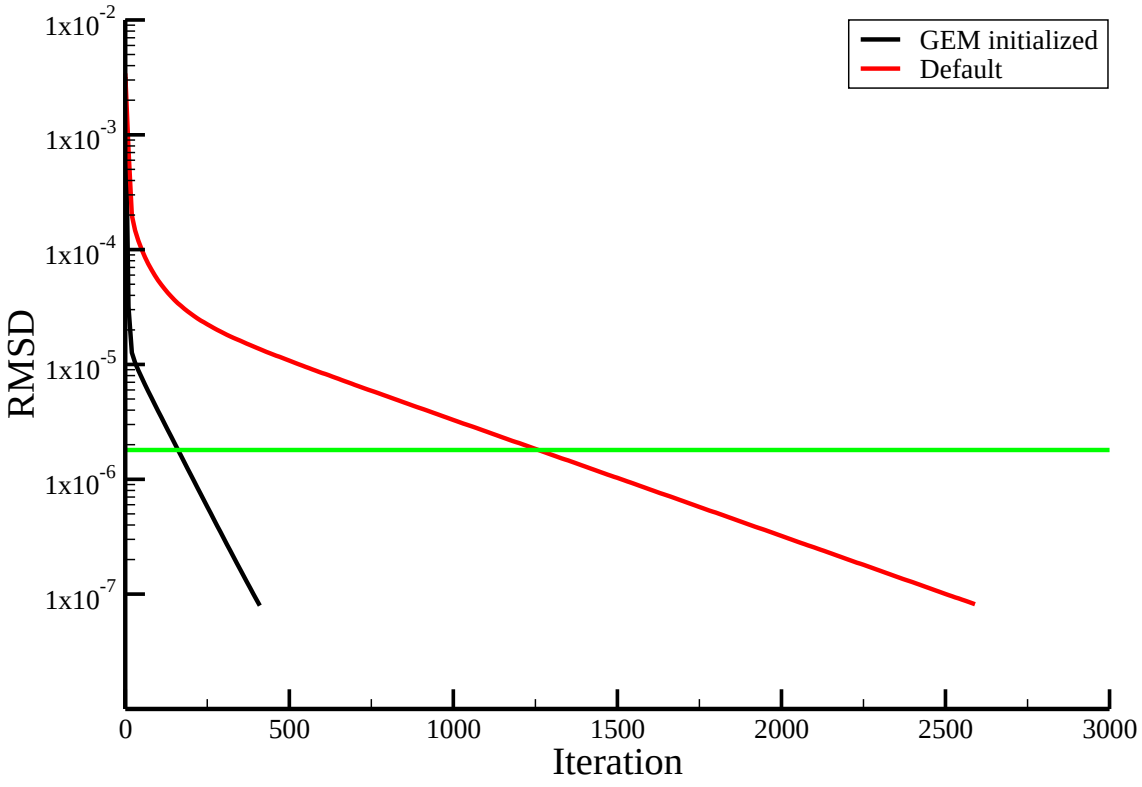

Figure 6.4: Relative rates of convergence for the spherical model.

`MEAD` is a popular NPB solver that employs the finite difference method. It is an ideal candidate for experimentation into boundary and mesh initializations for the following reasons: it is open source, it employs Successive Overrelaxation to solve its matrix equation, and it allows the input of an initial guess to the solution which is used as a means of approximating the results of using an internal pre-conditioner based on the analytical model presented here.

The spherical model demonstrated in Figure 5.1 is used for which it has been shown that the GEM model generates more accurate potentials than even a fully converged `MEAD` approximation. It is clear that the accuracy of the initial guess of an iterative method affects the time required to converge (if the initial guess is the correct answer then it would require only 0 iterations to discover and converge). However, it is not necessarily true that the GEM solution will converge very rapidly because of the differences in molecular representations used by the two methods enumerated in Appendix G.

Initializing `MEAD` with GEM electrostatic potentials dramatically improves the performance of the SOR component of the algorithm. Figure 6.4 demonstrates the differences in RMSD convergence for GEM initialized electrostatic potentials versus MEAD running with no explicit initialization, achieved by using a Debye initialization at the boundary of the solution. The black line represents the RMSD observed when `MEAD` is explicitly initialized with the GEM solution. The green line is thermal noise / 10^3 converted to raw potential units used by MEAD, which represents one example physically relevant convergence criterion. For this test, a domain of 120 ångströms with elements equally spaced .25 ångströms apart was used (241 x 241 x 241 total elements) to provide an accurate formulation of the problem.

One alternative way of viewing the improvement experienced by using GEM as an initializer is to measure the ratio of work required to converge to within a given error bound. In this case, rather than viewing the problem in terms of how much accuracy can be achieved given a fixed number of iterations we view the problem as how many iterations are required to achieve a given accuracy. Because computational cost is of such importance in so many areas of biological research regarding electrostatic potential, viewing the problem in terms of computational cost is appropriate. In this case, a performance efficiency ratio can be used to view the performance speed improvement experienced by using GEM initializations relative to using a default initialization. Figure 6.5 plots the performance improvements experienced on the spherical test case with a grid resolution of .5 ångströms. The performance improvement for a given RMSD convergence factor is measured by Equation (6.1).

$$perf_{RMSD} = \frac{NPB\ default}{GEM\ initialized} \tag{6.1}$$

Where $NPB\ default$ is the number of iterations required to achieve the given RMSD by default and $GEM\ initialized$ is the number of iterations required to achieve the given RMSD by using GEM as an initial guess. Three candidate convergence values were selected based on thermal noise (kT): $\frac{kT}{10}$, $\frac{kT}{100}$, and $\frac{kT}{1000}$. These values represent viable, increasingly conservative choices for error in the potential estimations relative to the uncertainty inherent in sampling the physical state of the molecule to obtain the structure.

At less than $\frac{kT}{10}$, the gem initialized solution provides infinite performance improvement (as it requires 0 iterations to converge). However, as the required accuracy is increased, the performance improvement decreases as one would expect due to the way iterates are formed.

6.2.2 Stepping toward reality: two intersecting spheres

There are multiple levels of complexity involved in stepping away from the simple spherical model toward real biomolecules including the increased complexity in the charge distribution and the increased complexity of the surface. Since NPB methods discretize both the surface

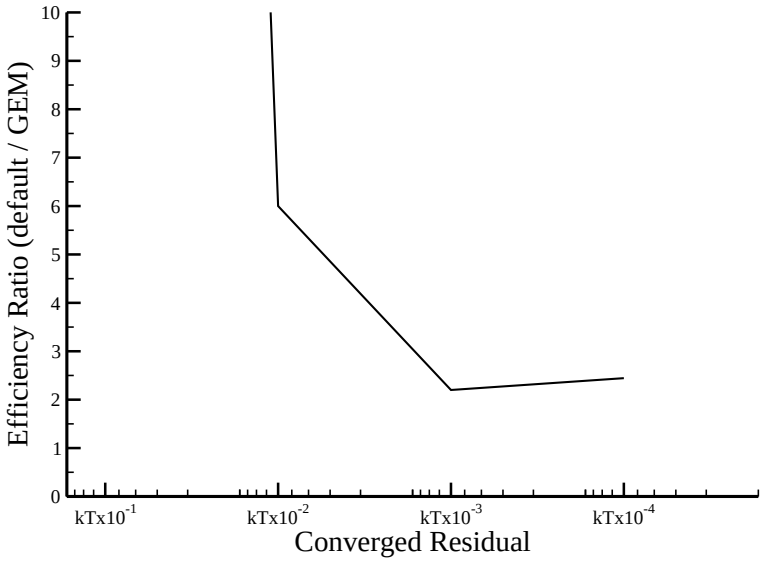

Figure 6.5: Speed improvements observed in the spherical test case.

and the charge distribution in such a manner as to fundamentally modify the problem, it is useful to view what effects (if any) this might have on using GEM as a preconditioner for NPB methods. Under ideal conditions, of course, these effects could be negated by running GEM on the discretized problem rather than the free space problem. Two intersecting spheres offer the ability to view the effects of adding complexity to the surface without adding complexity to the charge distribution. In this way, the effects of discretizing the surface and charge distribution can be determined as the problem increases in complexity.

Figure 6.6 demonstrates the performance improvements experienced by using GEM as a preconditioner for the intersecting sphere model at .5 ångström resolution. The performance improvement ranges from 3 to 4.19 and is not a monotonic function. Since the charges and placements are similar to the spherical test case, the charge distribution has not increased in complexity relative to the simple spherical case, however the surface representation has increased in complexity. One immediate observation of the problem is that performance improvement is no longer highest where required accuracy is lowest. It is clear, at this point, that surface complexity plays a significant role in determining how well GEM initializations perform as it has changed very little and the resulting curve has changed dramatically.

6.2.3 Conceptual validation on a real bio-molecule test set

The performance improvement experienced in Figure 6.4 may be unique to the spherical geometry upon which GEM was derived. Therefore, similar analysis is performed on 19 molecules from the biological test set as an additional step toward validating the applicability

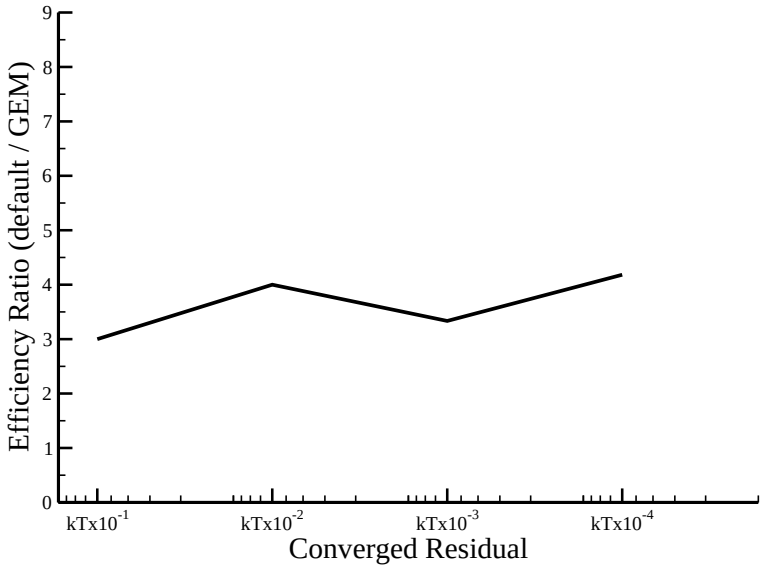

Figure 6.6: Speed improvements observed in the more complicated ideal test case.

of GEM as a pre-conditioner. The structures used for this validation are listed in Appendix A. The average performance improvement for a given RMSD convergence factor is measured by Equation (6.1).

$$average\ perf_{RMSD} = \frac{1}{19}\sum_{i=1}^{19} perf_{RMSD}^{(i)} \tag{6.2}$$

Where $perf_{RMSD}^{(i)}$ is defined in Equation (6.1).

Figure 6.7 demonstrates the minimum, maximum, and average performance improvements experienced by using GEM as a preconditioner for the 19 biomolecule test set. Of primary interest is the average efficiency ratio; however, the minimum and maximum efficiency ratios experienced are somewhat enlightening in that they provide measures of how much the performance improvement changes by molecule.

The average efficiency ratio improves with desired accuracy up to $10^{-3}\ kT$ where it begins to break down. At this point, there are any number of possible explanations for this effect including effects of Chebyshev acceleration or long-term effects of representational differences.

At this point, it is clear that as the complexity of the problem increases, the predictability of error decreases. This is likely related to the discretization process, though these effects are unknown at this point. It is interesting to see that the best biomolecular performance improvement is higher than the best spherical test improvement or dual sphere performance improvement. This is somewhat reassuring while simultaneously puzzling.

This analysis provides a computational bound requirement for GEM to be used as an initial guess to volumetric finite difference methods. In order for this pre-conditioner to break even

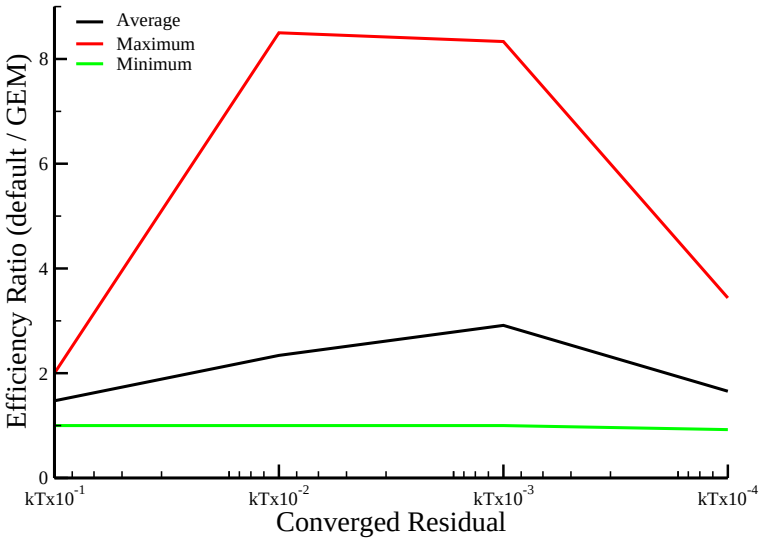

Figure 6.7: Performance improvements experienced in the biomolecular test.

on this particular set in terms of computational work, it must be capable of computing electrostatic potential throughout the volume in between $\frac{1}{3}$ and $\frac{2}{3}$ of the time required for the finite difference approximation to converge using SOR under current conditions (differing applications with differing representations of the surface and charge distribution). A formulation of the algorithm specifically for grid outputs may be capable of performing within this computational bound, but has not yet been explored. It must also be considered that this requirement may change dramatically if the GEM initializer is used on the exact same charge distribution and surface representation as the numerical algorithm.

Chapter 7

Discussion

There are fundamental differences in approach and result between the analytical methodology presented in this work and standard numerical approaches. These differences must be taken into account when selecting an approximation for a specific application. There are many uses for electrostatic potential and many sources of molecular structures. Many things should be considered when determining how electrostatic potential should be treated in a given case: the resolution of the structure being studied, the source of the partial charges in the structure, time constraints on the computation, and finally the domain of interest.

The resolution of molecular structures determined through molecular imaging techniques varies with the crystallization process and equipment used from less than 1 ångström to more than two ångströms. Near the molecular surface, the possible rearrangement of atoms can significantly alter the electrostatic potential and contribute a great deal to error at close proximities.

Partial charges are needed for any approximation of electrostatic potential, and are typically determined through use of databases containing expected partial charges for most residues. There could be significant variation in these partial charges due to thermal noise and throughout the proper functioning of these molecules and so again some error does exist in the partial charge assignments though it does not vary in a known manner and typically is approximated as thermal noise or 1 kT.

Some computations, while accurate, are too computationally intense to be feasible in a given pipeline. These specialized pipelines – such as docking determinations, folding problems, molecular dynamics, and high throughput drug screening – involve large highly dynamic problems that require approximations of electrostatic potential to be computed rapidly so as to enable these algorithms to run at a sufficient speed to be computationally and economically viable.

For reasonably small domains of interest relative to the size of the molecule in question, an analytical approximation can be computed much faster and with less memory overhead than

numerical approximations due to the independence of each sample point from the others. For these cases such as the interior and exterior surface area of the tobacco ringspot viral capsid, the domains of interest are significantly smaller than the volume of the structure and can be computed significantly faster and with fewer resources than standard volumetric finite difference methods.

Chapter 8

Summary

Various force-field methods of approximating the linearized Poisson-Boltzmann equation in the continuum dielectric model have been reviewed and an additional model has been derived for use in high-throughput situations to replace the GB model and broaden the spectrum of options available for electrostatic estimations for macromolecules. Some experimentation has been done with regard to using the new analytical method in conjunction with standard numerical methods as a means of investigating its applicability as a boundary initializer for volumetric finite difference approximations of electrostatic potential. The method has been tested and applied to one macromolecular example (a viral capsid) where it can be computed overnight on a desktop computer. This application of the analytical method derived in this work demonstrates the power of the method as well as its applicability to large molecules or complexes where numerical solutions would be prohibitively costly in terms of memory and computational time.

Appendix A

Structures

The Tobacco Ringspot Virus (TRSV) capsid is constructed from 60 identical protomers. The PDB file 1A6C contains the x-ray crystallographic coordinates of the single protomer at 3.50Å resolution; the transformation matrix given in the PDB file header is used to properly rotate and align each unit to form the complete capsid icosahedral structure.

The structures used to test the analytical electrostatic potential against the numerical PB reference are selected as follows. Start from the 600 representative biological molecules used for the testing purposes in earlier works.[55,58] Then numerical PB solvers `DelPhi`[2,32] and `MEAD`[59] with the default settings are used to generate the electrostatic potential on a $255 \times 255 \times 255$ cubic grid with 0.5 Å grid spacing. Finally, 20 of the 600 structures are excluded from the test set because either `DelPhi` or `MEAD` fail to output the potential map. For most of the failed cases the attempted calculation fails due to the requested memory exceeding the 1GB RAM capability of the test PC. Table A.1 contains the PDB codes of all the remaining molecules included in the test set for validating the analytical method. Codes listed in red were also used to evaluate the average speed increase experienced when using GEM as a pre-conditioner for NPB methods. However, in addition to the red codes listed in the table, BDNA, beta.bondi, lyso, and Mb.Hhelix were used to supplement the set with some small, non-spherical molecules.

Table A.1: The PDB codes of the 580 molecule test-set used to validate the GEM method.

Test PDB Codes											
2trx	7a3h	5gcn	4ull	4eug	3znf	3vub	3sil	3rpb	3phy	3msp	3mef
3lri	3crd	3chb	2vik	2u2f	2tps	2tmp	2sob	2rel	2ptl	2pth	2prf
2pcf	2orc	2olb	2nlr	2ncm	2mrb	2lis	2lfb	2jhb	2ifo	2ife	2if1
2hir	2hgf	2gva	2gcc	2gat	2fmr	2ezm	2ezk	2ezh	2end	2ctc	2bid
2alc	2a3d	1zto	1zta	1yui	1yub	1yua	1xpa	1xnb	1xna	1xbl	1whi
1wfb	1wdb	1vre	1vpu	1vgh	1uxc	1uwo	1utr	1urk	1ums	1u2f	1tsg
1trl	1tpm	1tof	1tns	1tle	1tfb	1tbn	1tbd	1tba	1tba	1swu	1svq
1svf	1suh	1ssn	1sro	1shc	1sgg	1scy	1sap	1rxr	1rrb	1rpr	1rot
1rip	1rie	1rge	1res	1rcs	1rch	1rax	1r2a	1qyp	1qu5	1qtw	1qtt
1qts	1qto	1qtn	1qtn	1qsv	1qry	1qqv	1qqi	1qqf	1qp6	1qop	1qnr
1qnd	1qn0	1qm9	1qlo	1ql0	1qks	1qkl	1qkf	1qk9	1qk7	1qk6	1qjo
1qhk	1qh4	1qgp	1qft	1qfr	1qfq	1qfd	1qdp	1qck	1qa5	1psm	1prr
1pou	1pon	1pnj	1pnb	1pnb	1pms	1pmc	1pls	1pir	1pih	1pfs	1pfl
1peh	1pcp	1pcn	1pce	1paa	1pa2	1om2	1olg	1oaa	1ntc	1ns1	1noe
1nls	1nkl	1ngr	1ngl	1neq	1nct	1ncs	1myf	1mut	1mun	1mro	1mro
1mnt	1mla	1mkn	1mkc	1mgs	1mfn	1lyp	1lre	1lea	1ksr	1krs	1koe
1kla	1kjs	1khm	1kdx	1jwe	1jun	1joy	1jli	1jhb	1jba	1ixh	1itf
1isu	1irs	1irp	1irl	1irf	1ioj	1inz	1imt	1il6	1ija	1iie	1ihv
1ica	1ibx	1iba	1i6w	1i5j	1i5h	1i5g	1i27	1i25	1i1s	1i0h	1hzy
1hzn	1hyw	1hyk	1hyi	1hx2	1hsq	1hs7	1hre	1hpw	1hp8	1hnr	1hks
1hhn	1hev	1hdo	1hcd	1hbw	1ha9	1h8c	1gyf	1gw3	1gp8	1gnc	1gio
1ghh	1ghc	1gh9	1ggw	1ge9	1gd0	1gab	1g9l	1g90	1g84	1g7e	1g7d
1g6s	1g6e	1g66	1g61	1g5v	1g4f	1g2h	1g26	1g25	1g1e	1fyj	1fyc
1fyb	1fwq	1fwp	1fwo	1fw9	1fvl	1fu9	1fsh	1fre	1fr3	1fqq	1fp0
1fo5	1fmh	1fm0	1fjn	1fjk	1fje	1fj2	1fgp	1fdm	1fd8	1fcy	1fct
1fbr	1faf	1fa4	1fa3	1f8p	1f81	1f5y	1f53	1f41	1f3r	1f3c	1f24
1f0z	1ezt	1ezo	1ezg	1eza	1exk	1exg	1exe	1eww	1ews	1ewi	1ev0
1euw	1esx	1es9	1erx	1erd	1erc	1eqo	1eq3	1ep0	1eo1	1eo0	1enw
1elk	1ekt	1ej5	1eiw	1eit	1eik	1ehx	1ehs	1ehj	1eh2	1egx	1ef4
1edx	1edv	1eds	1eci	1e8r	1e8l	1e88	1e7l	1e6u	1e6q	1e68	1e5u
1e5g	1e53	1e4u	1e3y	1e3t	1e2b	1e29	1e19	1e17	1e0z	1e0l	1e0h
1e0e	1e0a	1e01	1dz7	1dxz	1dx8	1dx7	1dx0	1dwm	1dvj	1dvh	1dv0
1duj	1du6	1du2	1dtv	1ds9	1ds1	1dro	1dqz	1dqc	1dqb	1dpu	1dp7
1dp3	1dny	1dmc	1dlx	1dl6	1dl0	1dj0	1dip	1dgq	1dgn	1dfs	1dfe
1def	1dec	1de3	1de1	1ddf	1dci	1dbf	1dbd	1daq	1d8v	1d8j	1d8b
1d7q	1d6g	1d1h	1d1d	1cz4	1cyu	1cye	1cx1	1cwx	1cww	1cw5	1cur
1cou	1coo	1cok	1co4	1cn2	1cmr	1cmo	1clh	1cl4	1ckv	1ck2	1chl
1chc	1cg7	1cfe	1cf4	1ce4	1cdq	1cdb	1ccm	1cch	1c9q	1c89	1c7u
1c7k	1c75	1c5e	1c55	1c4e	1c3y	1c2n	1c20	1c1k	1c1d	1c05	1c01
1bzk	1bzg	1byy	1byq	1bym	1byi	1by1	1bxo	1bxd	1bwx	1bw6	1bw3
1bvh	1bve	1buy	1buq	1bt7	1bsh	1brz	1brv	1br0	1bqv	1bpv	1bpr
1boe	1bo9	1bo0	1bnr	1bno	1bmx	1bmw	1bmr	1bm4	1blr	1blj	1bla
1bl1	1bku	1bkr	1bjx	1bj8	1bip	1bi6	1bhu	1bh4	1bgk	1bfm	1bds
1bdc	1bct	1bci	1bc6	1bby	1bbn	1bbi	1bbg	1bb8	1baq	1bal	1bak
1ba9	1b9u	1b91	1b8w	1b8o	1b6f	1b64	1b4r	1b22	1b1a	1b16	1az6
1ayj	1axj	1axh	1awj	1aw6	1aw0	1auz	1auu	1arb	1aq5	1aps	1apc
1ap8	1ap7	1ap0	1aoy	1ao8	1aml	1akp	1ak6	1ajy	1ajw	1aje	1aj3
1aiw	1ahl	1ah9	1ah2	1agg	1afo	1afh	1af8	1adr	1adn	1aci	1aca
1ac0	1a93									1a66	1a23

Appendix B

Protonating the TRSV Capsid

The standard continuum electrostatics methodology[77,78] is used to protonate the viral capsid. The full structure contains 4617 titratable groups – too many for the standard continuum solvent based approach. Therefore the number of titratable groups is reduced by generating a subsection of the capsid surface such that one protomer unit is completely surrounded by other protomers. This results in a nine protomer (enneamer) subsection of the surface with one unit in the center and eight units surrounding it. The enneamer contains 981 titratable sites, which is still too many for the standard continuum solvent based approach. Only the groups in the central unit are considered to be titratable in the calculations, the others are set in their standard protonation states. The total number of groups treated as titratable is therefore reduced to 125.

The `AMBER`[79] set of partial atomic charges is used here for the protein charges. For the protonated states of Asp and Glu, in which the correct location of the proton is not known a priori a "smeared charge" representation is used, in which the neutralizing positive charge is symmetrically distributed: 0.45 on each carbonyl oxygen atom and 0.1 on the carbon atom. The web server `H++`[80] is used to perform the calculations with the following settings: 0.15M monovalent salt concentration, internal dielectric 6, and external dielectric 80. The computed pK_as of the central unit are used to set its protonation state at each pH. The full capsid is then constructed from this protonated unit as described in Appendix A. The biologically relevant pH interval from 4 to 9 is divided into 100 equidistant points: for each pH value the full capsid is constructed in the corresponding protonation state.

Appendix C

Generating the Secondary Structure of TRSV Satellite RNA

The program MFOLD[81,82] is used with the default setting to generate the secondary structure of TRSV satellite RNA. Its sequence length is 359 nt which is within the confidence range of the MFOLD methodology. The lowest free energy conformation is chosen, with $\Delta G = -141.28\ kcal/mol$.

The sequence of the TRSV satellite RNA:[83]

```
  1: accggatgtgctttccggtctgatgagtccgtgaggacgaaacaggactg
 51: tcaggtggccgaaagccaccacgtaaactagtgaaccgtgctgcgtagcg
101: taggggtctgctacctcgttggaggtggagattgtagccttcgtgtgggc
151: gcggcggtgtagctagtcaaggcgtaccaggtaatataccacaacgtgtg
201: tttctctggttgacttctctgtttgttgtgtcattggttcccggatctcg
251: cattagcggcgacggggtattctcattcgacatggaagtttgagagaccg
301: cgcctctacactatgcgcggccggggcgaatccaaattgttctagcccga
351: taccctgtc
```

Appendix D

Generation of reference NPB electrostatic potential

The reference electrostatic potential around each of the test structures is computed using `DelPhi`[2,32] and `MEAD`[59] with a $255 \times 255 \times 255$ cubic box. The default `MEAD` and `DelPhi` convergence criteria are used in all cases.

For the "perfect sphere" test case, the external medium is assumed to be pure water with a dielectric constant of 80 and no mobile ions ($\kappa = 0$). The internal medium is assumed to have a dielectric constant of 1.

Biologically realistic conditions have been used for the 580 realistic biomolecular structures. The solvent is assumed to be a dielectric constant of 80, a salt content of 0.145M, and an ion exclusion radius of 2.0 Å. The internal medium is assumed to have dielectric constant of 4.

Appendix E

Generation of molecular surfaces

For each of the 580 bio-molecules in the test set described in Appendix A, the molecular surface is obtained through the program `MSMS`[84] using a probe radius of 2.0 and a triangle density of 3.0 triangles per Å. The molecular surface sets the boundary between the interior and exterior dielectric environments. The vertices that make up the `MSMS` molecular surface are then used as a basis for the sample points used to test the analytical formula against the NPB reference because electrostatic potential is largest near the surfaces of molecules (and so error is likely to be similarly larger).

Appendix F

Sampling points

The electrostatic potential estimations provided by numerical solvers right at the molecular surface may be sensitive to the details of the surface definition. To avoid the related artifact, sample points 1.5 Å away from the molecular surface (described in Appendix E) are chosen. That is sample points are obtained by projecting each `MSMS` surface vertex outwards, 1.5 Å along its surface normal.

For each sample point, two potential values are obtained: ϕ_{GEM} (the analytical approximation) and ϕ_{NPB} (the numerical approximation). The ϕ_{GEM} is calculated via Equations (4.41 and 4.39). The ϕ_{NPB} is taken to be ϕ_{NPB} of the nearest finite-difference grid point.

The probe radius used in `MSMS` must satisfy $probe > \mathbf{p} + R/2$ where $\mathbf{p}$ is the projection length, R is the grid resolution. This ensures that, inside regions of negative curvature, the sample points do not get projected back into the solute. The minimal such probe radius given grid grid resolution of 0.5 Å and projection length of 1.5 Å would be 1.75 Å; however, 2.0 Å is used as a means of partially addressing differences in the surface representation used by the reference NPB solvers and `MSMS`.

For hybridization experiments, a grid of ϕ values identical in format and orientation to the `MEAD` method are generated. This involves using a uniform spacing throughout the volume of the molecule and its surrounding solvent. A grid dimension of 40 Å and element spacing of .5 Å is used. This results in an 81x81x81 cubic volume of uniformly spaced approximations with 1 point in the exact center of the field.

Appendix G

Representational Differences Between GEM and MEAD

Both MEAD and GEM take as input a PQR file, which generally contains the physical locations, charges, and radii of all atoms making up a molecule. There are two fundamental differences in representation that can be a source of error in calculations using hybrid methods. First, the charge distributions are not treated in the exact same way. Second, the molecular surfaces are not treated in the exact same way.

The charge distribution, realistically, is continuous across the molecule and does not exist as a set of point charges located at individual locations in space. As part of the finite difference methodology ρ values are desired at every point in space. It is also desirable that ρ be as smooth as possible within the confines of its defined space. As a means of coping with this MEAD spreads the charge of a given atom across all grid points within the volume of a given atom. The resulting transformed set of point charges is smooth, but fundamentally different from the distribution provided by the PQR input file. GEM treats the charge distribution as a set of point charges defined exactly as they are given in the input file. These differences are particularly important near singularities (locations of point charges). Within the interior of the molecule, it is given that $\vec{r}$ will be within the radius of one or more atoms. Within GEM, ϕ approaches a singular value in a uniform manner relative to the centers of the input atoms. Within MEAD, ϕ approaches smaller singularities because $|\vec{r}|$ is bounded, and these singularities are more prevalent due to the smearing of the charges.

The solvent excluded surface of a molecule can be defined in a number of ways, and various treatments have been discussed for this problem. There are three dominant ways to define molecular surface right now, the union of the Vanderwall's radii of all atoms, the Connolly surface,[85] and the Gaussian surface.[86] Connolly surfaces are derived by simulating the "rolling" of a spherical probe about the surface of the molecule to determine which crevices are inaccessible to water. The Gaussian surface is defined as the sum of a set of Gaussian functions defining the density of the atoms, and a cut-off is used to determine the location

of the manifold. GEM makes use of the Connolly surface as provided by the popular program `MSMS`, and uses the definition provided directly. It is unknown how the reference NPB methods generate their molecular surfaces, but it is known that Finite difference methods operate at a fixed resolution in a cubic lattice. Points that are located within the molecule are flagged as interior and points that are located outside the molecule are flagged as exterior. The exact surface is less important in finite difference methods than the flagging of interior and exterior points, and determining if a point is on the surface is done by checking to see if the points on either side of it have a different classification regarding its location (interior or exterior). The surface is important in GEM, in that it is used to determine the depth of an atom in the molecule, it plays a role in determining mobile salt effects, and it determines where the sample point lies (region I, II, or III). Therefore, surface differences between GEM and `MEAD` can have a significant effect on calculations throughout space, but are most powerful near the surface itself where misclassification of a sample point would result in significantly different potential approximations.

Bibliography

[1] Radić, Z., Kirchhoff, D., Quinn, D., McCammon, J., and Taylor, P. Electrostatic influence on the kinetics of ligand binding to acetylcholinesterase. Journal of Biological Chemistry 272(37):23265–23277, Sept, 1997.

[2] Honig, B. and Nicholls, A. Classical electrostatics in biology and chemistry. Science 268:1144, 1995.

[3] Chin, K., Sharp, K. A., Honig, B., and Pyle, A. M. Calculating the electrostatic properties of rna provides new insights into molecular interactions and function. Nat Struct Biol 6(11):1055–1061, Nov, 1999.

[4] Perutz., M. Electrostatic effects in proteins. Science 201:1187–1191, 1978.

[5] Davis, M. E. and McCammon, J. A. Electrostatics in Biomolecular Structure and Dynamics. Chem. Rev. 90:509–521, 1990.

[6] Baker, N. A. and McCammon, J. A. Eelectrostaic Interactions. In *Structural Bioinformatics*. John Wiley & Sons, Inc., New York, , 2002.

[7] Warshel, A. and Åqvist, J. Electrostatic Energy and Macromolecular Function. Ann. Rev. Biophys. Biophys. Chem. 20:267–298, 1991.

[8] Warshel, A. Calculations of enzymatic reactions: Calculations of pk_a, proton transfer reactions, and general acit catalysis reactions in enzymes. Biochemistry 20:3167–3177, 1981.

[9] Fersht, A., Shi, J., Knill-Jones, J., Lowe, D., Wilkinson, A., Blow, D., Brick, P., Carter, P., Waye, M., and Winter, G. Hydrogen bonding and biological specificity analysed by protein engineering. Nature. 314:235–8, 1985.

[10] Szabo, G., Eisenman, G., McLaughlin, S., and Krasne, S. Ionic probes of membrane structures. in: Membrane structure and its biological applications. Ann. N.Y. Acad. Sci. 195:273–290, 1972.

[11] Douglas, T. and Ripoll, D. R. Calculated electrostatic gradients in recombinant human h-chain ferritin. Protein Sci 7(5):1083–1091, May, 1998.

[12] Sheinerman, F. B., Norel, R., and Honig, B. Electrostatic aspects of protein-protein interactions. Curr. Opin. Struct. Biol 10(2):153–9, 2000.

[13] Onufriev, A., Smondyrev, A., and Bashford, D. Proton affinity changes during unidirectional proton transport in the bacteriorhodopsin photocycle. J. Mol. Biol. 332:1183–1193, 2003.

[14] Yang, A.-S. and Honig, B. Electrostatic effects on protein stability. Curr. Opin. Struct. Biol. 2:40–45, 1992.

[15] Whitten, S. and Garcia-Moreno, B. ph dependence of stability of staphyococcal nuclease: Evidence of substantial electrostatic interactions in denatured state. Biochemistry 39:14292–14304, 2000.

[16] Felder, C., Botti, S., Lifson, S., Silman, I., and Sussman, J. External and internal electrostatic potentials of cholinesterase models. Journal of Molecular Graphics and Modelling 15(5):318–327, Oct, 1998.

[17] Botti, S., Felder, C., Lifson, S., Sussman, J., and I Silman, I. A modular treatment of molecular traffic through the active site of cholinesterase. Biophysical Journal 77(5):2430–2450, Nov, 1999.

[18] Trkanjec, Z. Electrostatic attraction: The driving force for the presynaptic vesicle-cell membrane fusion. Medical hypotheses 47:93–96, 1996.

[19] Han, X. and Jackson, M. Electrostatic interactions between the syntaxin membrane anchor and neurotransmitter passing through the fusion pore. Biophysical Journal 88(3):L20–L22, Jan, 2005.

[20] Lin, Z. and Uhl, G. Dopamine transporter mutants with cocaine resistance and normal dopamine uptake provide targets for cocaine antagonism. Molecular Pharmacology 61(4):885–891, Apr, 2002.

[21] Baker, N. A., Sept, D., Holst, M. J., and McCammon, J. A. The adaptive multilevel finite element solution of the Poisson–Boltzmann equation on massively parallel computers. IBM Journal of Research and Development 45(3/4):427–438, October, 2001.

[22] D, B. Macroscopic electrostatic models for protonation states in proteins. Fronteirs in bioscience 1:1082–1099, May, 2004.

[23] Baker, N. A., Sept, D., Joseph, S., Holst, M. J., and McCammon, J. A. Electrostatics of nanosystems: application to microtubules and the ribosome. Proc Natl Acad Sci U S A 98(18):10037–10041, Aug, 2001.

[24] Press, W., Flannery, B., Teukolsky, S., and Vetterling, W. Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, New York, NY, USA, , 1992.

[25] Bruccoleri, R., Novotny, J., Davis, M., and Sharp, K. Finite difference Poisson–Boltzmann electrostatic calculations: Increased accuracy achieved by harmonic dielectric smoothing and charge antialiasing. Journal of Computational Chemistry 18:268–276, February, 1997.

[26] Zhou, Z., Payne, P., Vasquez, M., Kuhn, N., and Levitt, M. Finite-difference solution of the Poisson–Boltzmann equation: Complete elimination of self-energy. Journal of Computational Chemistry 11:1344 – 1351, November, 1996.

[27] Flaherty, J. Finite element analysis lecture notes. Rensselaur Polytechnic Institute 1:1–300, January, 2000.

[28] Axelsson, O. and Barker, V. Finite Element Solution of Boundary Value Problems: Theory and Computation. Society for Industrial and Applied Mathematics, Philadelphi, PA, , 1934.

[29] Baker, N., Holst, M. J., and Wang, F. The adaptive multilevel finite element solution of the Poisson–Boltzmann equation ii. refinement at solvent-accessible surfaces in biomolecular systems. Journal of Computational Chemistry 21(15):1343 – 1352, June, 2000.

[30] Sayyed-Ahmad, A., K., T., and Ortoleva, P. Efficient solution technique for solving Poisson–Boltzmann equation. Journal of Computational Chemistry 25:1068–1074, 2004.

[31] Host, M., Baker, N., and Wang, F. Adaptive multilevel finite element solution of the Poisson–Boltzmann equation i: Algorithms and examples. Journal of Computational Chemistry 21:1319–1342, 2000.

[32] Nicholls, A. and Honig, B. A Rapid Finite Difference Algorithm, Utilizing Successive Over Relaxation to solve the Poisson–Boltzmann Equation. J. Comp. Chem. 12:435–445, 1991.

[33] Hageman, L. and Young, D. Applied Iterative Methods. Academic Press, New York, , 1981.

[34] Kahan, W. Gauss-seidel methods of solving large systems of linear equations. Doctoral Thesis, University of Toronto 0, 1958.

[35] Bashford, D. and Case, D. Generalized Born models of macromolecular solvation effects. Annu. Rev. Phys. Chem. 51:129–152, 2000.

[36] Tsui, V. and Case, D. Theory and applications of the generalized born model suitable for macromoelcules. Biopolymers 56:275–291, 2001.

[37] Onufriev, A., Bashford, D., and Case, D. Modification of the Generalized Born Model Suitable for Macromolecules. J. Phys. Chem. B 104:3712–3720, 2000.

[38] Tsui, V. and Case, D. Molecular dynamics simulations of nucleic acids using a generalized Born solvation model. J. Am. Chem. Soc. 122:2489–2498, 2000.

[39] Jayaram, B., Liu, Y., and Beveridge, D. A modification of the generalized Born theory for improved estimates of solvation energies and pK shifts. J. Chem. Phys. 109:1465–1470, 1998.

[40] Onufriev, A., Bashford, D., and Case, D. Effective born radii in the generalized born approximation: The importance of being perfect. Journal of Computational Chemistry 23:1297–1304, 2002.

[41] Onufriev, A., Bashford, D., and Case, D. Exploring native states and large-scale conformational changes with a generalized born model. Proteins 55:383–394, 2004.

[42] Casanova, J., Kent IV, D., William, A., III, G., and Roberts, J. Quantum-mechanical calculations of the stabilities of fluxional isomers of $C_4H_7^+$ in solution. PNAS 100(1):15–19, Jan, 2002.

[43] Mei, Y., Ji, C., and Zhang, J. A new quantum method for electrostatic solvation energy of protein. Journal of Chemical Physics 125(9):094906–094906–7, 2006.

[44] Eichinger, M., Grubmüller, H., Heller, H., and Tavan, P. Famusamm: An algorithm for rapid evaluation of electrostatic interactions in molecular dynamics simulations. J. Comp. Chem. 18:1729–1749, 1997.

[45] Neiderman, C. and Tavan, P. Fast version of the structure adapted multipole method - efficient calculation of electrastatic forces in protein dynamics. Molecular Simulation 17:57–66, 1996.

[46] Nelson, M., Humphrey, W., Gursoy, A., Dalke, A., L., K., Skeel, R., Schulten, K., and Kufrin, R. Mdscope – a visual computing environment for structural biology. Computational Physics Communications 91:111–134, 1995.

[47] Gargallo, R., Oliva, B., Querol, E., and Avilés, F. Effect of reaction field electrostatic term on the molecular dynamics simulation of the activation domain of procarboxypeptidase b. protein engineering design and selection 13(1):21–26, Jan, 2000.

[48] Patra, M., Karttunen, M., Hyvönen, M., Falck, E., Lindqvist, P., and Vattulainen, I. Molecular dynamics simulations of lipid bilayers: Major artifacts due to truncating electrostatic interactions. Biophysical Journal 84:3636–3645, 2003.

[49] Ramos, M. and Fernandes, P. Atomic-level rational drug design. Current Computer-Aided Drug Design 2(1):57–81, Mar, 2006.

[50] Hermann, T. and Westof, E. Rational drug design and high-throughput techniques for rna targets. Comb. Chem. and High Throughput Screening 3(3):219–234, Jun, 2000.

[51] Lu, B., Cheng, X., Huang, J., and McCammon, A. Order n algorithm for computation of electrostatic interactions in biomolecular systems. PNAS 103:19314–19319, Dec, 2006.

[52] Bordner, A. and Huber, G. Boundary element solution of the linear Poisson–Boltzmann equation and a multipole method for the rapid calculation of forces on macromolecules in solution. Journal of Computational Chemistry 24:353–367, Jan, 2003.

[53] Vorobjev, Y., Grant, J., and Scheraga, H. A combined iterative and boundary element approach for solution of the nonlinear Poisson–Boltzmann equation. J. Am. Chem. Soc. 114:3189–3196, 1992.

[54] Sigalov, G., Scheffel, P., and Onufriev, A. Incorporating variable dielectric environments into the generalized born model. J. Chem. Phys. 122(9):094511–094511, Mar, 2005.

[55] Sigalov, G., Fenley, A., and Onufriev, A. Analytical linearized Poisson–Boltzmann approach: Beyond the generalized born approximation. J. Chem. Phys. in press, 2006.

[56] Kirkwood, J. G. Theory of Solutions of Molecules Containing Widely Separated Charges with Special Application to Zwitterions. J. Chem. Phys. 2:351–361, 1934.

[57] Jackson, J. Classical Electrodynamics Third Edition. J. Wiley & Sons, New York, , 1999.

[58] Feig, M., Onufriev, A., Lee, M., Im, W., Case, D., and Brooks, C. Performance Comparison of Generalized Born and Poisson Methods in the Calculation of Electrostatic Solvation Energies for Protein Structures. J. Comp. Chem. 25:265–284, 2004.

[59] Bashford, D. An object-oriented programming suite for electrostatic effects in biological molecules. In *Scientific Computing in Object-Oriented Parallel Environments,* Ishikawa, Y., Oldehoeft, R. R., Reynders, J. V. W., and Tholburn, M., editors, volume 1343 of *Lecture Notes in Computer Science*, 233–240 (ISCOPE97Springer, Berlin, 1997).

[60] Havranek, J. J. and Harbury, P. B. Tanford–Kirkwood electrostatics for protein modeling. Proc. Natl. Acad. Sci. U. S.A. 96:11145–11150, 1999.

[61] Jessica M.J. Swanson and Stewart A. Adcock and J. Andrew McCammon. Optimized radii for Poisson–Boltzmann calculations using amber force field. J. Chem. Theor. Comp. 1:484–493, 2005.

[62] Schneider, I. R. Satellite-like particle of tobacco ringspot virus that resembles tobacco ringspot virus. Science 166(913):1627–1629, Dec, 1969.

[63] Chandrasekar, V. and Johnson, J. E. The structure of tobacco ringspot virus: a link in the evolution of icosahedral capsids in the picornavirus superfamily. Structure 6(2):157–171, Feb, 1998.

[64] Buzayan, J. M., McNinch, J. S., Schneider, I. R., and Bruening, G. A nucleotide sequence rearrangement distinguishes two isolates of satellite tobacco ringspot virus rna. Virology 160(1):95–99, Sep, 1987.

[65] Robaglia, C., Bruening, G., Haseloff, J., and Gerlach, W. L. Evolution and replication of tobacco ringspot virus satellite RNA mutants. The EMBO Journal 12:2969–2976, 1993.

[66] Passmore, B. and Bruening, G. Similar structure and reactivity of satellite tobacco ringspot virus rna obtained from infected tissue and by in vitro transcription. Virology 197:108–115, 1993.

[67] Singh, S., Rothnagel, R., Prasad, B., and Buckley, B. Expression of tobacco ringspot virus capsid protein and satellite rna in insect cells and three-dimensional structure of tobacco rignspot virus-like particles. Virology 213:472–481, 1995.

[68] Brown, D. J. F., Robertson, W. M., and Trudgill, D. L. Transmission of viruses by plant nematodes. Annual Reviews: Phytopathol 33:223–249, 1995.

[69] ICTVdB-Management. 18.0.3.0.027 tobacco ringspot virus. ICTVdB - The Universal Virus Database, version 4 http://www.ncbi.nlm.nih.gov/ICTVdb/ICTVdB/00.018.0.03.027.htm, 2002.

[70] Wang, S. and Gergerich, R. C. Immunofluorescent Localization of Tobacco Ringspot Nepovirus in the Vector Nematode *Xiphinema americanum*. American Phytopathological Society - Phytopathology 88:885–889, 1998.

[71] Wang, S., Gergerich, R. C., Wickizer, S. L., and Kim, K. S. Localization of Transmissible and Nontransmissible Viruses in the Vector Nematode *Xiphinema americanum*. American Phytopathological Society - Phytopathology 92:646–653, 2002.

[72] Harrison, B. D., Robertson, W. M., and Taylor, C. E. Specificity of retention and transmission of viruses by nematodes. J. Nematol. 6:155–164, 1974.

[73] Johnstone, G. R. and Wade, G. C. Therapy of Virus-Infected Plants by Heat Treatment. I Some Properties of Tomato Aspermy Virus and its Inactivation at 36C. Aust. J. Bot. 22:437–450, 1974.

[74] Zalloua, P. A., Buzayan, J. M., and Bruening, G. Chemical cleavage of 5'-linked protein from tobacco ringspot virus genomic rnas and characterization of the protein-rna linkage. Virology 219(1):1–8, May, 1996.

[75] Moon, J. S., Domier, L. L., and L, H. G. Nucleotide sequence of tobacco ringspot virus rna2. NCBI GenBank NC 005096, 2004.

[76] Mathews, D., Sabina, J., Zuker, M., and Turner, D. Expanded sequence dependence of thermodynamic parameters improves prediction of rna secondary structure. J Mol Biol. 5:911–940, 1999.

[77] Bashford, D. and Karplus, M. pka's of ionizable groups in proteins: atomic detail from a continuum electrostatic model. Biochemistry 29(44):10219–10225, 1990.

[78] Bashford, D. and Gerwert, K. Electrostatic calculations of the pka values of ionizable groups in bacteriorhodopsin. J. Mol. Biol. 224:473–486, 1992.

[79] Pearlman, D., Case, D., Caldwell, J., Ross, W., Cheatham III, T., DeBolt, S., Ferguson, D., Seibel, G., and Kollman., P. Amber, a computer program for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to elucidate the structures and energies of molecules. Comp. Phys. Commun. 91:1–41, 1995.

[80] Gordon, J., Myers, J., Folta, T., Shoja, V., Heath, L., and Onufriev, A. H++: a server for estimating pk_as and adding missing hydrogens to macromolecules. Nucleic Acids Research 33:368–371, 2005.

[81] Zuker, M. On finding all suboptimal foldings of an rna molecule. Science 244:48–52, 1989.

[82] Mathews, D., Sabina, J., and Turner, D. Expanded sequence dependence of thermodynamic parameters provides robust prediction of rna secondary structure. J. Mol. Biol. 288:911–940, 1999.

[83] Buzayan, J. M., Gerlach, W. L., Bruening, G. E., and Gould, P. K. A. R. Nucleotide sequence of satellite tobacco ringspot virus RNA and its relationship to multimeric forms. Virology 151:186–199, 1986.

[84] Sanner, M. F., Olson, A., and Spehner, J. Fast and robust computation of molecular surfaces. In *Proceedings of the eleventh annual symposium on Computational geometry*, 406–407. ACM Press, , 1995.

[85] Connolly, M. Analytical molecular surface calculation. Journal of Applied Crystallography 16:548–558, 1983.

[86] Grant, J. and Pickup, B. A gaussian description of molecular shape. J. Phys. Chem. 99:3503–3510, 1995.