

Copyright  
by  
Shun-Chuan Chen  
2009

The Dissertation Committee for Shun-Chuan Chen  
certifies that this is the approved version of the following dissertation:

**Speed and Accuracy Tradeoffs in Molecular  
Electrostatic Computation**

Committee:

---

Chandrajit Bajaj, Supervisor

---

Okan Arikan

---

Inderjit Dhillon

---

Peter Rossky

---

Lexing Ying

**Speed and Accuracy Tradeoffs in Molecular  
Electrostatic Computation**

by

**Shun-Chuan Chen, M.S.**

**DISSERTATION**

Presented to the Faculty of the Graduate School of  
The University of Texas at Austin  
in Partial Fulfillment  
of the Requirements  
for the Degree of

**DOCTOR OF PHILOSOPHY**

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2009

To my loving family.

## Acknowledgments

I would like to thank the many people who have helped to make this dissertation possible. First, I would like to thank my advisors, Chandrajit Bajaj, who played an indispensable role in my intellectual development. I am truly grateful for his constant guidance, encouragement and patience along the difficult trail of finishing my doctoral study. I would also like to thank the rest of my committee, Okan Arikan, Inderjit Dhillon, Peter Rossky and Lexing Ying, for their commitment and support. This dissertation could not have been made a reality without them. I also offer sincere thanks to all of my fellow students in Computational Visualization Center, for understanding and supporting me as they climbed this mountain with me. My gratitude also extends to my lovely friends in Texas and in Taiwan for their invaluable friendships. I would especially like to thank Wenqi Zhao, for always being helpful in good and bad times.

I would like to thank my loving family - my dear parents, my sister, my brother-in-law Edward and my sister-in-law Grace, for their constant care and belief in my achievement. I dedicate my deepest gratitude and love to my fabulous wife, Hsuan-Wei Michelle Chen, who is the meaning of my life. Her love, support, encouragement and humor have made me alive throughout these rough and ready years. Finally, thank You my awesome God, for Your

mercy, grace and guidance. You are worthy Lord of all glory and all praise.

Hook 'em Horns!

Shun-Chuan (Albert)

# Speed and Accuracy Tradeoffs in Molecular Electrostatic Computation

Publication No. \_\_\_\_\_

Shun-Chuan Chen, Ph.D.  
The University of Texas at Austin, 2009

Supervisor: Chandrajit Bajaj

In this study, we consider electrostatics contributed from the molecules in the ionic solution. It plays a significant role in determining the binding affinity of molecules and drugs. We develop the overall framework of computing electrostatic properties for three-dimensional molecular structures, including potential, energy, and forces. These properties are derived from Poisson-Boltzmann equation, a partial differential equation that describes the electrostatic behavior of molecules in ionic solutions.

In order to compute these properties, we derived new boundary integral equations and designed a boundary element algorithm based on the linear time fast multipole method for solving the linearized Poisson-Boltzmann equation. Meanwhile, a higher-order parametric formulation called algebraic spline model is used for accurate approximation of the unknown solution of the linearized Poisson-Boltzmann equation. Based on algebraic spline model,

we represent the normal derivative of electrostatic potential by surrounding electrostatic potential. This representation guarantees the consistent relation between electrostatic potential and its normal derivative. In addition, accurate numerical solution and fast computation for electrostatic energy and forces are also discussed. In addition, we described our hierarchical modeling and parameter optimization of molecular structures. Based on this technique, we can control the scalability of molecular models for electrostatic computation. The numerical test and experimental results show that the proposed techniques offer an efficient and accurate solution for solving the electrostatic problem of molecules.

# Table of Contents

<b>Acknowledgments</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>List of Tables</b>	<b>xii</b>
<b>List of Figures</b>	<b>xv</b>
<b>Chapter 1. Background and Contribution</b>	<b>1</b>
1.1 Introduction . . . . .	2
1.2 Implicit Continuum Model and the Poisson-Boltzmann Equation	8
1.3 Basic molecular shapes . . . . .	12
1.4 Boundary conditions . . . . .	13
1.5 Theory and derivation . . . . .	16
1.5.1 Derivation of polarization energy . . . . .	17
1.5.2 Derivation of polarization forces . . . . .	20
1.5.3 Derivation of boundary integral equations . . . . .	23
1.5.4 The derivation of interior and exterior electrostatic potential . . . . .	26
1.5.5 New Boundary integral equations . . . . .	30
<b>Chapter 2. Efficient and Consistent Computation for Electrostatic Potential and its Gradient</b>	<b>33</b>
2.1 Implementation . . . . .	34
2.1.1 The implementation of electrostatic potential and its gradient . . . . .	35
2.1.1.1 The surface integral computation using higher order boundary elements ASMS . . . . .	35
2.1.1.2 The implementation of solving Juffer's Poisson-Boltzmann boundary integral equations . . . . .	37

2.1.1.3	The implementation of computing electrostatic potential and its gradient . . . . .	40
2.1.1.4	The implementation of solving enhanced Poisson-Boltzmann boundary integral equations . . . . .	41
2.2	The consistent representation of normal derivative of electrostatic potential . . . . .	43
2.3	Framework and computational steps . . . . .	48
2.3.1	Complexity Analysis . . . . .	51
2.4	Experimental results . . . . .	55
2.4.1	Analytical numerical evaluation . . . . .	55
2.4.2	Poisson-Boltzmann electrostatic potential . . . . .	59
<b>Chapter 3. Fast Computation of Electrostatic Energy and Forces</b>		<b>64</b>
3.1	Dielectric and ionic boundary model . . . . .	65
3.1.1	Im's model . . . . .	65
3.1.2	Bajaj and Zhao's model . . . . .	66
3.2	Implementation . . . . .	68
3.2.1	The implementation of electrostatic energy . . . . .	68
3.2.2	The implementation of electrostatic force . . . . .	69
3.3	Experimental Results . . . . .	73
3.3.1	Complexity Analysis . . . . .	73
3.3.2	Poisson-Boltzmann electrostatic solvation free energies . . . . .	73
3.3.2.1	A unit sphere with single positive charge . . . . .	73
3.3.2.2	A list of ligand-receptor complexes . . . . .	74
3.3.3	Poisson-Boltzmann Electrostatic forces . . . . .	77
3.3.4	interior and exterior electrostatic potential . . . . .	80
<b>Chapter 4. Hierarchical Molecular Modeling</b>		<b>83</b>
4.1	Background . . . . .	84
4.2	Hierarchical molecular models . . . . .	85
4.3	Experimental results . . . . .	89
4.3.1	Molecular surface comparison . . . . .	90
4.3.2	Poisson-Boltzmann electrostatic free energy . . . . .	91
4.3.3	Poisson-Boltzmann electrostatic potential . . . . .	94
4.3.4	Poisson-Boltzmann electrostatic solvation force . . . . .	97

<b>Bibliography</b>	<b>103</b>
<b>Vita</b>	<b>115</b>

## List of Tables

2.1	The notations for complexity analysis of electrostatic computation. . . . .	52
2.2	The results of analytical experiments computed using PB BEM solver with different number of A-patches for 213 molecules; column 1 is the number of triangles (* is the average number of A-patches of the original triangular mesh of 213 molecular surfaces); column 2 is the type of evaluation method of matrix-vector product; column 3 is the average relative errors of potential $\phi$ and $\tilde{\phi}$ ; column 4 is the number of iterations for the convergence; column 5 is the computation time in seconds. . .	57
2.3	The results of analytical experiments computed using PB BEM solver with different number of A-patches for 213 proteins; column 1 is the number of triangles (* is the average number of A-patches of the original triangular mesh of 213 molecular surfaces); column 2 is the type of numerical solution of boundary element method; column 3 is the average relative errors of potential $\phi$ and $\tilde{\phi}$ ; column 4 is the average relative errors of normal derivative of potential $\frac{\partial\phi}{\partial n}$ and $\frac{\partial\tilde{\phi}}{\partial n}$ ; column 5 is the number of iterations for the convergence; column 6 is the computation time in seconds. . . . .	58
2.4	Average experimental results of PB electrostatic potential computation for 213 proteins (71 sets of ligand-receptor complexes); column 1 is the numerical method; column 2 is the name of the solver; column 3 is the number of grids for FDM and number of A-patches for BEM; column 4 is the inverse discretization length scale of each grid or A-patches; column 5 is the correlation of electrostatic potential to FDM with $193^3$ grids. . . . .	61
2.5	The comparison of electrostatic potential and gradients of electrostatic potential computed using Juffer's boundary integral equation and enhanced boundary integral equations. The correlation is computed according to the results computed using finite difference method (Delphi II) with full Coulombic boundary condition. . . . .	63

2.6	The comparison of electrostatic potential and gradients of electrostatic potential computed using Juffer's boundary integral equation and enhanced boundary integral equations. The correlation is computed according to the results computed using finite difference method (Delphi II) with zero boundary condition. . . . .	63
3.1	PB electrostatic free energy of a unit sphere with single charge computed using different numerical method; column 1 is the numerical method; column 2 is the name of the solver; column 3 is the number of grids for FDM and number of A-patches for BEM; column 4 is the electrostatic free energy $G_{pol}$ ( $kcal/mol$ ); column 5 is the relative error of electrostatic free energy $G_{pol}$ . As a reference, the exact electrostatic free energy is $-80.9 kcal/mol$ with the interior and exterior dielectric constant 2 and 80; column 6 is computational time in seconds. . . . .	74
3.2	The statistics of the experiments including the average, maximum and minimum of the number of iterations, compute time, compute time per iterations and the correlation with Delphi FDM( $193^3$ grids) of our BEM with different number of A-patches for 213 molecules (71 sets of ligands, receptors and ligand-receptor complexes). (* the average number of A-patches of the original triangular mesh of 213 molecular surfaces) . . . .	75
4.1	The coefficients of implicit solvent model which is used for PB boundary element method. . . . .	89
4.2	The comparison of molecular surfaces of a set of proteins. Each term from left to right is the result of AA model, 5-bead CG model and 2-bead CG model; column 1 is PDB id of the protein where (l) and (r) indicate the ligand and receptor of the complex protein; column 2 shows the number of atoms or beads in the AA/5-bead CG/2-bead CG models, respectively; column 3 shows the Hausdorff distance $\hat{A}$ between the molecular surfaces of the 5-bead CG/2-bead CG to the molecular surfaces of the AA models; column 4 shows the surface area $\hat{A}^2$ of the AA/5-bead CG/2-bead CG models, respectively. The last column shows the relative error of the surface area of the 5-bead CG/2-bead CG models which are computed as $(\text{area of the CG model} - \text{area of the AA model})/(\text{area of the AA model})$ . . . .	90

4.3 The experimental results of a set of proteins. Each term from left to right is the result of AA model, 5-bead CG model and 2-bead CG model; column 1 is PDB id of the protein where (l) and (r) indicate the ligand and receptor of the complex protein; column 2 is number of atoms of AA model and number of beads of 5-bead CG and 2-bead CG model; column 3 is the number of A-patches of AA, 5-bead CG and 2-bead CG models; column 4 is electrostatic free energy  $G_{pol}$  (*kcal/mol*), 5-bead CG model is a good approximation; column 5 is the relative error of electrostatic free energy computed using 5-bead CG and 2-bead CG models to that computed using AA model; column 6 is computation time of PB boundary element method (*seconds*). 94

## List of Figures

1.1	The example of a four-atom molecule. Each sphere represents an atom. $\mathbf{x}_k$ represents the center of $k^{th}$ atom. The boundary surrounding the spheres is its molecular surface $\Gamma$ . . . . .	10
1.2	Basic molecular shapes: van der Waal surface (VWS), solvent-excluded surface (SES) and solvent-accessible surface (SAS). . .	12
1.3	The example of a four-atom molecule. Each sphere represents an atom. $\mathbf{x}_k$ represents the center of $k^{th}$ atom. The boundary surrounding the spheres is its molecular surface $\Gamma$ and the outer boundary of the volume region $\Omega$ is $\partial\Omega$ . . . . .	14
2.1	The example of boundary element decomposition of a four-atom molecule: $\mathbf{x}_k$ is the center of $k^{th}$ atom, $\mathbf{x}_i$ and $\mathbf{y}_j$ are the points on the elements $\Gamma_i$ and $\Gamma_j$ of the surface $\Gamma$ and $\mathbf{n}_i^{(x)}$ and $\mathbf{n}_j^{(y)}$ are their normal vectors. . . . .	34
2.2	A prism scaffold of triangular element $\mathbf{v}_i\mathbf{v}_j\mathbf{v}_k$ . . . . .	36
2.3	Transform the unknown functions $\phi$ and $\frac{\partial\phi}{\partial\mathbf{n}}$ to new unknown functions $\phi$ on two surface layers. . . . .	43
2.4	The representation of a point $\mathbf{x}$ on a algebraic patch $\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3$ . . .	44
2.5	The representation of electrostatic potential $\phi(\mathbf{x})$ at a point $\mathbf{x}$ on a algebraic patch $\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3$ . . . . .	46
2.6	The computational steps of the evaluation of matrix-vector product at each iteration of GMRES linear solver. . . . .	52
2.7	The PB electrostatic potential on molecular surface of nuclear transport factor 2 (PDB id: 1A2K) with different resolutions. (e) shows the difference of PB electrostatic potential between BEM and FDM. The color is going from red (potential of $-3.8 k_bT/e_c$ ) to blue (potential of $+3.8 k_bT/e_c$ ). . . . .	59
2.8	The comparison of electrostatic potential between the molecular surface of Bovine Chymotrypsinogen*A (PDB id: 1CGI) with different resolutions (the correlation in parentheses) where the average number of A-patches of the original surface is 63444.81. . . . .	60

3.1	Dielectric function $\epsilon$ and ionic boundary function $\lambda$ defined using Im's volume exclusion function with the window size $w = 0.2$	65
3.2	The quadrature points of the shell integral are points within a spherical shell around the atom. . . . .	71
3.3	The shell region of the molecular surface $\Gamma$ . . . . .	71
3.4	The comparison of electrostatic free energy ( $kcal/mol$ ) of 213 proteins (71 sets of ligand-receptor complexes) between BEM and FDM with $193^3$ grids with the correlation in parentheses. . . . .	76
3.5	The comparison of per-atom energy (KJ/mol) of Bovine Chymotrypsinogen*A (PDB id: 1CGI) between FDM with $193^3$ grids and BEM with 10000 A-patches. Each point indicates the electrostatic solvation free energy of an atom. . . . .	77
3.6	The relation of per-atom electrostatic force ( $kcal/mol \cdot \text{\AA}$ ) of Bovine Chymotrypsinogen*A (PDB id: 1CGI) computed using BEM or FDM where blue,pink,yellow dots indicate x,y,z-dimensional values of forces; (a) BEM with 28908 A-patches vs FDM with $193^3$ grids, the correlations at $x, y, z$ dimensions are (0.8108,0.8449,0.8198); (b) BEM with 28908 A-patches vs FDM with $256^3$ grids, the correlations at $x, y, z$ dimensions are (0.8118,0.8629,0.8317). . . . .	78
3.7	The inner product of unit normal vector and PB electrostatic forces on the molecular surface of nuclear transport factor 2 (PDB id: 1A2K) and Bovine Chymotrypsinogen*A (PDB id: 1CGI) with different resolutions. The color is going from blue ( $\geq 3.8 kcal/mol \cdot \text{\AA}$ ) to red ( $\leq -3.8 kcal/mol \cdot \text{\AA}$ ). . . . .	79
3.8	The PB electrostatic potential $\phi$ of Bovine Chymotrypsinogen*A (PDB id: 1CGI). The color is going from red ( $\phi < -3.8 k_bT/e_c$ ) to blue ( $\phi > 3.8 k_bT/e_c$ ). (a) the molecular surface; (b) the molecular surface colored by $\phi$ ; (c) $\phi$ distributed in a volume $64 \times 64 \times 64 \text{\AA}^3$ with the molecule on the center; (d) with the molecular surface colored by $\phi$ ; (e),(f) the cutting slices of the volume. . . . .	81
3.9	The PB reation field electrostatic potential $\phi_{rf} = \phi_{sol} - \phi_{air}$ of (a) nuclear transport factor 2 (PDB id: 1A2K) and (b) Bovine Chymotrypsinogen*A (PDB id: 1CGI). The color is going from red ( $\phi_{rf} < -3.8 k_bT/e_c$ ) to blue ( $\phi_{rf} > 3.8 k_bT/e_c$ ). . . . .	82
4.1	An example that the atoms of an amino-acid are clustered into 5 beads and 2 beads. . . . .	85

4.2	An example that all atoms are grouped into 2 beads. Their new centers $x'_1, x'_2$ and radius $r'_1, r'_2$ are computed. . . . .	87
4.3	(a) The Atomic model of protein 1AK4 (2260 atoms) and its SES surface. (b) The CG model (290 atoms) of 1AK4 and its SES surface. Each amino acid is represented by two beads, one for the backbone and one for the side chain. The Hausdorff distance between them is 1.998 Å. . . . .	88
4.4	The relation of per-bead electrostatic energy ( $kcal/mol$ ) between AA molecular model and CG molecular models of the protein (PDB id: 1CGI); (a) AA model vs 5-bead CG model, the correlation is 0.9950; (b) AA model vs 2-bead CG model, the correlation is 0.9135. . . . .	93
4.5	The PB electrostatic electrostatic free energy ( $kcal/mol$ ) during the charge optimizing process of 5-bead CG model of each protein. All of them are convergent. . . . .	95
4.6	The relation of per-bead electrostatic potential ( $k_bT/e_c$ ) between AA molecular model and CG molecular models of the protein (PDB id: 1CGI); (a) AA model vs 5-bead CG model, the correlation is 0.9950; (b) AA model vs 2-bead CG model, the correlation is 0.9059. . . . .	96
4.7	The PB electrostatic potential on molecular surface with different resolutions (atomic and coarse-grained resolution). The visual errors between AA and all CG models are small. The color is going from red (potential of $-3.8 k_bT/e_c$ ) to blue (potential of $+3.8 k_bT/e_c$ ). . . . .	98
4.8	The relation of per-bead electrostatic force ( $kcal/mol \cdot \text{Å}$ ) between AA molecular model and CG molecular models of the protein (PDB id: 1CGI) where blue, pink, yellow dots indicate x, y, z-dimensional values of forces; (a) AA model vs 5-bead CG model, the correlations at $x, y, z$ dimensions are 0.9223, 0.9117, 0.9448; (b) AA model vs 2-bead CG model, the correlations at $x, y, z$ dimensions are 0.8224, 0.6773, 0.8472. . . . .	100
4.9	The inner product of unit normal vector and PB electrostatic forces on the molecular surface with different resolutions (atomic and coarse-grained resolution). The visual error between AA and 5-bead CG models is small but not between AA and 2-bead CG model. The color is going from blue ( $\geq 7.6 kcal/mol \cdot \text{Å}$ ) to red ( $\leq -7.6 kcal/mol \cdot \text{Å}$ ). . . . .	101

# Chapter 1

## Background and Contribution

## 1.1 Introduction

Accurate and effective computational approaches for atomistic simulation of bio-molecules are significant topics of current computational biological research. Different biological activities such as drug design or molecular trajectory simulation can be performed based on numerical solutions of solvation energy [62].

The Molecular Energetics and Force Calculation Problem:

The total free energy of a molecule in solution includes two different parts, molecular mechanical energy  $E_{\text{MM}}$  and solvation energy  $G_{\text{sol}}$ . The molecular mechanical energy  $E_{\text{MM}}$  is measured by the following equation [52].

$$\begin{aligned} E_{\text{MM}} &= \sum_{\text{bonds}} k_b (r - r_{eq})^2 + \sum_{\text{bond angles}} k_\theta (\theta - \theta_{eq})^2 \\ &+ \sum_{\text{torsion angles}} k_\phi (1 - \cos[n(\phi - \phi_{eq})]) \\ &+ \sum_i \sum_{j < i} \left[ \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} + \frac{q_i q_j}{\epsilon r_{ij}^2} \right] \end{aligned}$$

The first three sums represent bonded interaction: covalent bonds, valence bonds, and torsions around bonds [30, 33]. The last two terms are Lennard-Jones energy and Coulomb energy.

On the other hand, the effect of the solvation free energy  $G_{\text{sol}}$  is used when a molecule is in the solution. The solvation free energy is the sum of the work to form a cavity in the solvent  $G_{\text{cav}}$ , van der Waals interaction energy  $G_{\text{vdw}}$ , and the electrostatic solvation free energy change due to the solvation  $G_{\text{pol}}$  (also known as polarization energy) [39, 40, 44, 66, 70].

$$G_{\text{sol}} = G_{\text{cav}} + G_{\text{vdw}} + G_{\text{pol}},$$

The electrostatic solvation energy is the change in the electrostatic energy due to the induced polarization in the solvent, and so the electrostatic component of the solvation energy can be described through the contribution of atomic charges and mobile ion charges with the effect of the dielectric medium. The derivation of electrostatic solvation energy will further be described in the following section.

A number of applications involve the computation of electrostatic solvation energy. For example, the binding effect of a drug (molecule 1) and its target (molecule 2) can be measured by the following binding energy relation.

$$\Delta G_{bind} = G_{complex} - (G_{molecule1} + G_{molecule2}),$$

which shows that the binding energy is the difference between the solvation energy of the complex of two molecules (e.g. target and drug) and the sum of the solvation energy of the two individual molecules.

Furthermore, the electrostatic forces of molecules can be computed to simulate their activity. The electrostatic forces are the derivative of electrostatic solvation energy due to the atomic centers. The accuracy and computational cost of electrostatic force computation directly affect the simulation results. In order to compute the electrostatic force, the electric field of proteins themselves and the influence of their dielectric and ionic environment should be considered.

**Background and Significance:** Considerable research efforts have been devoted to calculating binding solvation energy and forces in the past two decades.

Based on the solvent model used these different theoretical approaches can be divided into two broad categories: explicit and implicit.

Explicit solvent models adopt a microscopic treatment of both solvent and solute (molecule). Explicit approaches sample the solute-solvent space by molecular dynamics or Monte Carlo techniques which involve a large number of ions, water molecules and molecular atoms. This requires considerable computational effort for calculating the potential functions is needed and explicit solutions are often not practical especially for large domains. [77]

Implicit solvent models treat the solvent as a featureless dielectric material and adopt a semi-microscopic representation of the solute. The effects of the solvent are modeled in terms of dielectric and ionic physical properties. As a result, the computational cost is reduced in comparison with explicit solutions. Implicit continuum electrostatics approaches using the Poisson-Boltzmann (PB) equation are now widely used and have been successfully used to obtain good approximations.

#### **Main Contributions:**

In this thesis, we use implicit solvent models and Poisson-Boltzmann equation to compute the electrostatic properties including the electrostatic potential, the gradient of electrostatic potential, electrostatic energy and forces. The main contributions of this thesis are the development of efficient and consistent solution and algorithm for electrostatic computation of molecules. We discuss and analyze the tradeoff between accuracy and speed for each different

parts of electrostatic computation. Those contributions include:

- New consistent and efficient approaches for solving linear Poisson-Boltzmann equation
  - **New Poisson-Boltzmann boundary integral equations** The derivation of boundary integral equations is the first step for solving Poisson-Boltzmann equation using boundary element method. Zauhar and Morgan [79] first formulated the problem for zero ionic strength as an integral equation on the surface. Juffer [48] reformulated the boundary integral equations in such a way that no singularity in the surface integral appears. However, both formulations are treated with satisfying the regularity conditions at infinity. In this thesis, we defined a new boundary integral equation that produces a linear Poisson-Boltzmann system with respect to finite volume boundary condition. The boundary integral equations avoid the singularity from atomic charge density function.
  - **Accurate surface integral using curved boundary elements** In traditional boundary element method, the boundary integral equations are evaluated based on shape functions over the linear triangular boundary elements. In this thesis, higher-order curved boundary elements, algebraic spline model [8], are applied for evaluating the integral over the molecular surface by Gaussian quadrature computation [81]. Using less curved boundary elements, we could still get

more accurate numerical results than that using linear triangular elements.

- Consistent representation of normal derivative of electrostatic potential In the pervious PB boundary integral equations, electrostatic potential and its normal derivative on the molecular surface are treated as two independent unknown functions. We present a consistent parametric formulation of the normal derivative of electrostatic potential based on the algebraic spline model. The corresponding results are in our manuscript [14].
  - Iterative linear solver with linear-time evaluation of matrix-vector production The boundary integral equations derived from PB equation are evaluated by the Gaussian quadrature over whole boundary elements of the molecular surface. This linear system is dense and large. The size of the coefficient matrix of the linear system is  $O(L^2)$  where  $L$  is the number of boundary elements. Instead of explicitly storing the matrix into the memory, we evaluate the linear system by using GMRES iterative method with KiFMM [78]. KiFMM is a linear-time evaluation method for computing matrix-vector production of the linear system.
- Efficient computation of electrostatic energy and forces
    - Fast evaluation of the integral equations of electrostatic energy The fast evaluation of electrostatic potential and gradient of electrostatic

potential is applied for accurate computation of electrostatic energy using fast summation method.

- **Fast evaluation of the integral equations of electrostatic forces** The derivative of electrostatic energy at the center of an atom is the electrostatic force acting on an atom  $\alpha$ . We used shell integral with electrostatic potential and its gradient to accurately evaluate the electrostatic forces. Fast summation method is also applied for computational speed-up.
- **Coarse-grained description for molecular electrostatic computation**
  - **Hierarchical molecular modeling** In order to control the scalability of the molecular models, the hierarchical clustering is applied to group the atoms of a molecule into pseudo-atoms which are also called coarse-graining beads [15].
  - **Parameter Optimization** After the clustering process, the locations, radius and charges of new pseudo-atoms (coarse-grained beads) have to be determined. We designed a solution to optimize the force-field parameters for improving the efficiency of electrostatic computation [15].

For each contribution, we defined the problem, developed the solution and algorithm and then analyzed the solution and experimental results. All the details are presented in the following chapters.

## 1.2 Implicit Continuum Model and the Poisson-Boltzmann Equation

A molecule is defined as a stable group of at least two atoms in a definite arrangement held together by very strong chemical covalent bonds. For a molecule embedded in an ionic solution, we separated the domain  $\Omega$  into interior  $\Omega_{int}$  and exterior regions  $\Omega_{ext} = \Omega - \Omega_{int}$  by the molecular surface  $\Gamma$  [51].

The continuum model of a molecule in the solvent is then defined by these two regions and used for numerical computation of solvation electrostatic computation. Two important coefficients of a continuum model are dielectric and ionic strength. The dielectric coefficient  $\epsilon(\mathbf{x})$  and ion boundary function  $\lambda(\mathbf{x})$  at position  $\mathbf{x}$  depends on which region  $\mathbf{x}$  belongs to.

$$\epsilon(\mathbf{x}) = \begin{cases} \epsilon_I, & \mathbf{x} \in \Omega_{int}, \\ \epsilon_{II}, & \mathbf{x} \in \Omega_{ext}. \end{cases}$$
$$\lambda(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \Omega_{int}, \\ 1, & \mathbf{x} \in \Omega_{ext}. \end{cases}$$

where  $\epsilon_I$  and  $\epsilon_{II}$  are dielectric constants

Based on the continuum model, the electrostatic potential in the interior and exterior of a molecule is governed by Poisson equation.

$$\nabla \cdot (\epsilon(\mathbf{x}) \nabla \phi(\mathbf{x})) = \rho_c(\mathbf{x}) + \rho_b(\mathbf{x})$$

$$\begin{aligned}
\nabla \cdot (\epsilon(\mathbf{x}) \nabla \phi(\mathbf{x})) &= \rho_c(\mathbf{x}) + \rho_b(\mathbf{x}) \\
\rho_c(\mathbf{x}) &= -4\pi \sum_{k=1}^{n_c} q_k \delta(\mathbf{x} - \mathbf{x}_k) \\
\rho_b(\mathbf{x}) &= \lambda(\mathbf{x}) \sum_i e_c z_i c_i e^{-e_c z_i \phi(\mathbf{x}) / k_B T} \\
&\text{if the solvent only contains ions with 1 and } -1 \text{ charges,} \\
&= \bar{\kappa}^2 \lambda(\mathbf{x}) \frac{k_B T}{e_c} \sinh\left(\frac{e_c \phi(\mathbf{x})}{k_B T}\right)
\end{aligned}$$

where each notation in the equation is defined as follows.

$\delta(\mathbf{x}) = \begin{cases} +\infty, & \mathbf{x} = 0 \\ 0, & \mathbf{x} \neq 0 \end{cases}$	Dirac delta function
$q_k, \mathbf{x}_k$	charge and position of the atom $k$
$n_c$	number of atoms
$\bar{\kappa} = \sqrt{\frac{8\pi e_e^2 \epsilon_{II} I}{k_B T}}$	modified Debye-Huckel parameter
$e_c$	charge of an electron
$k_B$	Boltzmann's constant
$T$	absolute temperature
$I = \frac{1}{2} \sum_i c_i z_i^2$	ionic strength
$c_i$	concentration of $i^{th}$ ionic species
$z_i$	charge of $i^{th}$ ionic species

This modified Poisson equation is also called the Poisson-Boltzmann equation. The Poisson-Boltzmann (PB) equation has been used for modeling the electrostatic properties of macromolecules. The equation represents the electrostatic potential of macromolecules in the solvent which is described as a continuum model with implicit ionic representation. Lots of efforts are devoted to compute accurate electrostatic potential for 3D atomistic structure of macromolecules. The charge density  $\rho_c(\mathbf{x})$  in PB equation is explicitly determined by atomic charges of a molecule and the charge density  $\rho_b(\mathbf{x})$  is implicitly approximated by Boltzmann distribution of ionic charges.

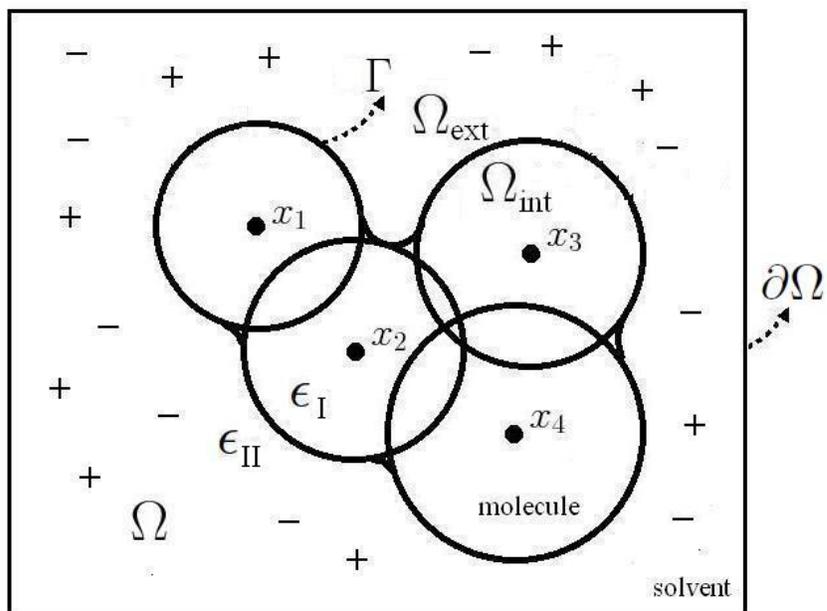


Figure 1.1: The example of a four-atom molecule. Each sphere represents an atom.  $x_k$  represents the center of  $k^{\text{th}}$  atom. The boundary surrounding the spheres is its molecular surface  $\Gamma$

Figure 1.1 shows an example of a four-atom molecule where each sphere represents an atom and the boundary surrounding the spheres is the molecular surface  $\Gamma$ .  $\mathbf{x}_k$  is the center of the  $k^{th}$  atom.

The linearized PB equation approximated from linearizing the full PB equation is widely used and believed as an efficient approximation for the regular solvation electrostatic problem. [57][21][76]

$$\nabla \cdot (\epsilon(\mathbf{x})\nabla\phi(\mathbf{x})) = \rho_c(\mathbf{x}) + \rho_b^L(\mathbf{x})$$

where  $\rho_b^L(\mathbf{x}) = \bar{\kappa}^2\lambda(\mathbf{x})\phi(\mathbf{x})$  is the first term of Taylor expansion of  $\rho_b(\mathbf{x})$ .

By solving this equation, we can obtain the electrostatic potential  $\phi(\mathbf{x})$  over the entire region. Since it is often difficult to directly solve PB equation for this kind of complex molecule-solvent systems, several computer programs have been created to solve it numerically. We also developed a boundary element solver with fast multipole method to numerically solve the linearized PB equation.

### 1.3 Basic molecular shapes

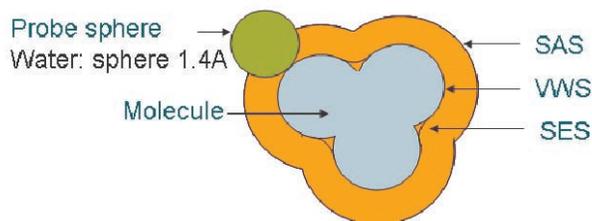


Figure 1.2: Basic molecular shapes: van der Waal surface (VWS), solvent-excluded surface (SES) and solvent-accessible surface (SAS).

The basic idea of constructing an implicit solvent model is to divide the open region by the interface of a molecule. As is well known, there are typically three types of molecular surfaces [64], the van der Waals surface (VWS), the solvent-accessible surface (SAS) and the solvent-excluded surface (SES) ([31]) or sometimes called the Lee-Richards surface ([50]). The molecular shapes of three-atom example are shown in Figure 1.2. The van der Waals surface is defined from the van der Waals radii of the atoms, which is the boundary of the region formed by the union of all the atoms. The SAS introduced by Lee-Richards is defined to be the locus of the center of the rolling spherical water molecule which makes contact with the VWS ([63]). Hence the SAS is an inflated VWS with a probe radius. The SES is the solvent surface inside of which the probe never intrudes.

In other words, SES is the offset surface of SAS in the inward direction with the solvent probe radius as the offset radius. This kind of surface can be represented by alpha shapes [28, 29, 53], which has been extensively

used for molecular surface modeling [1], cavity and pocket recognition ([54]). Molecular surface always referred to as SES could be represented analytically as a patch complex of spheres and tori ([32]) or as a patch complex of spheres and quadratic hyperboloids ([38]) or as parametric B-spline or NURBS ([6, 7]) or as level sets of summation of Gaussian atomic electron density functions ([27, 37, 42]).

Since the molecular surface also acts as a dielectric interface for electrostatic and polarization energy and force computations, the molecular surface should be at least  $C^1$  smooth and not too inflated or deflated. We use the level set of a tri-cubic B-spline function to approximate the molecular surface, which is almost  $C^2$  smooth [10, 11].

## 1.4 Boundary conditions

We expect the electrostatic potential  $\phi(\mathbf{r})$  to be continuous at the boundary  $\Gamma$  between the regions  $\Omega_{int}$  and  $\Omega_{ext}$ , as well as the dielectric coefficient times the normal derivative of the electrostatic potential,  $\epsilon \frac{\partial \phi(\mathbf{r})}{\partial \mathbf{n}} = \epsilon \nabla \phi(\mathbf{r}) \cdot \mathbf{n}$ , where  $\mathbf{n}$  is the unit outward normal vector at  $\mathbf{r}$  on the boundary  $\Gamma$ . Therefore, in  $\Gamma$ , the following surface boundary condition must be true that:

$$\phi_I(\mathbf{x})|_{\mathbf{x} \in \Gamma} = \phi_{II}(\mathbf{x})|_{\mathbf{x} \in \Gamma} \quad (1.1)$$

$$\epsilon_I \frac{\partial \phi_I}{\partial \mathbf{n}}(\mathbf{x})|_{\mathbf{x} \in \Gamma} = \epsilon_{II} \frac{\partial \phi_{II}}{\partial \mathbf{n}}(\mathbf{x})|_{\mathbf{x} \in \Gamma} \quad (1.2)$$

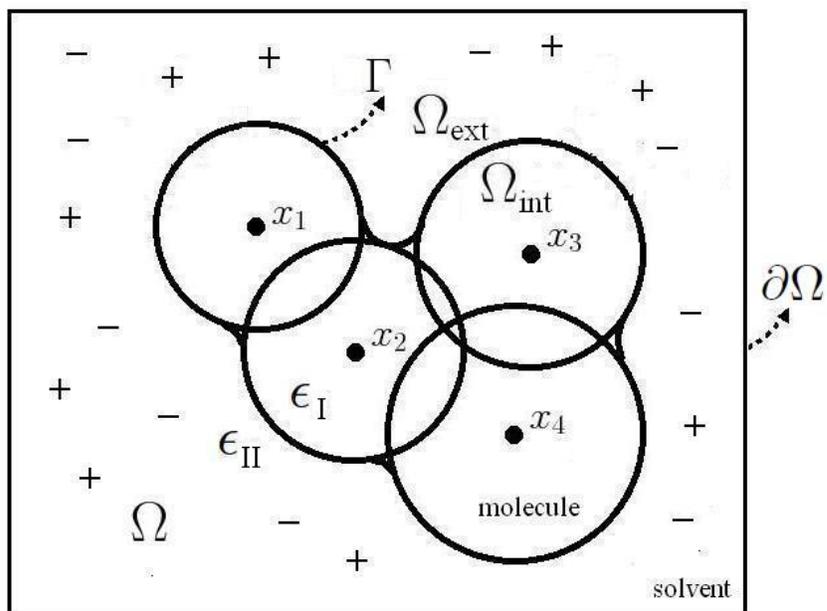


Figure 1.3: The example of a four-atom molecule. Each sphere represents an atom.  $x_k$  represents the center of  $k^{\text{th}}$  atom. The boundary surrounding the spheres is its molecular surface  $\Gamma$  and the outer boundary of the volume region  $\Omega$  is  $\partial\Omega$ .

In Figure 1.3, the rectangle containing the molecule represents the domain boundary  $\partial\Omega$  which is a cube in three-dimension. In addition to surface boundary conditions (1.1) and (1.2) defined on the molecular surface  $\Gamma$ , domain boundary condition is defined on the domain boundary  $\partial\Omega$  such that  $\phi = g$  on the boundary  $\partial\Omega$  where  $g$  is an arbitrary user-defined function. For example, the most common domain boundary definition is full Coulombic boundary condition:

$$g(\mathbf{x}) = \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_{II}} \frac{e^{-\bar{\kappa}\lambda(\mathbf{x})\|\mathbf{x}-\mathbf{x}_k\|}}{\|\mathbf{x}-\mathbf{x}_k\|} \quad (1.3)$$

This boundary condition is one of the possible boundary conditions taken to be induced by a known analytical solution to the simplification of the linearized PB equation. When the boundary  $\partial\Omega$  is far from the molecule, such analytical solution provides an accurate boundary condition approximation for the general PB equation on a truncation of  $\mathbb{R}^3$ .

## 1.5 Theory and derivation

Finite difference method (FDM), finite element method (FEM) and boundary element method (BEM) are three types of numerical solvers widely used to solve the PB equation.

FDM employs a box of three dimensional cubic grids where the molecule and surrounding solvent are contained. The electrostatic potentials are approximately solved on these grid points based on the PB equation. [45, 61, 65]

The idea of FEM is the approximation of partial differential equations in variational form over the space. FEM employs robust and various discretization of three dimensional space. The approximate solution of the PB equation is solved over these discrete elements while some iterative solution strategies, like inexact Newton methods and multilevel algorithms, are often applied for accurate and efficient numerical solution. [16, 24, 34, 35, 46].

Since R.J. Zauhar and R.S. Morgan introduced a BEM paper on continuum electrostatic of biological systems [79], in the past two decades, scientists have made contributions to improve and extend the BEM solution and performance. Some of these works focus on overcoming the difficulties of BEM which typically gives rise to fully populated matrices with numerous singular and hypersingular surface integrals. These works include the implementation of accelerating techniques for numerous singular and hypersingular surface integral operations [2, 20, 21, 23, 49, 56, 76] and, the analysis of and strategies for conditioning the linear system. [2, 23, 55, 56]

Other works make contributions to the methodological generalization including the solution from single molecule to multiple molecules [57, 82], the solution from the two-region case to the multiple-region case [2], and the extensive method for solving nonlinear PB equation. [22, 75]

We used boundary element method for solving PB equation and compute electrostatic energy and force with the corresponding PB electrostatic potential.

The following sections are the derivations of electrostatic energy, forces and Juffer’s well-conditioned boundary integral equations based on infinite-volume boundary condition. We then derived the new well-conditioned boundary integral equations with consideration of both surface and domain boundary conditions. The further implementation and analysis will be discussed in the following chapters.

### 1.5.1 Derivation of polarization energy

K.A. Sharp [66] derived the electrostatic solvation free energy from Poisson-Boltzmann equation using variation principle.

The theorem of variation principle states that

**Theorem 1.5.1.** *A multi-dimensional generalization comes from considering a function on  $x, y, z$ . If  $\Omega$  is the domain of  $x, y, z$*

$$G = \int_{\Omega} g(\phi, x, y, z, \frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial z})d\Omega$$

is extremized only if  $\phi$  satisfies the partial differential equation

$$\frac{\partial g}{\partial \phi} - \left( \frac{\partial}{\partial x} \frac{\partial g}{\partial \frac{\partial \phi}{\partial x}} + \frac{\partial}{\partial y} \frac{\partial g}{\partial \frac{\partial \phi}{\partial y}} + \frac{\partial}{\partial z} \frac{\partial g}{\partial \frac{\partial \phi}{\partial z}} \right) = 0$$

In order to get the electrostatic free energy of the PB equation, we design a undetermined function  $g(\phi, x, y, z, \frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}, \frac{\partial \phi}{\partial z})$  such that the following lemma derived from the theorem (1.5.1) holds:

**Lemma 1.5.2.** *For any PDE that can be put in Euler-Lagrange form,  $g$  is a unique function whose integral over the domain  $\Omega$  of the independent variables  $x, y, z$ ,  $G = \int_{\Omega} g d\Omega$  is minimized by the solution of PDE.*

We write PB equation in a form of Euler-Lagrange equation using  $F$  which satisfies

$$\frac{\partial g}{\partial \phi} - \left( \frac{\partial}{\partial x} \frac{\partial g}{\partial \frac{\partial \phi}{\partial x}} + \frac{\partial}{\partial y} \frac{\partial g}{\partial \frac{\partial \phi}{\partial y}} + \frac{\partial}{\partial z} \frac{\partial g}{\partial \frac{\partial \phi}{\partial z}} \right) = 0$$

where

$$\rho_c(\phi) + \rho_b(\phi) = \frac{\partial g}{\partial \phi}$$

and

$$-\nabla(\epsilon(\mathbf{x})\nabla\phi(\mathbf{x})) = \frac{\partial}{\partial x} \frac{\partial g}{\partial \frac{\partial \phi}{\partial x}} + \frac{\partial}{\partial y} \frac{\partial g}{\partial \frac{\partial \phi}{\partial y}} + \frac{\partial}{\partial z} \frac{\partial g}{\partial \frac{\partial \phi}{\partial z}}$$

Since the condition of the equilibrium is the the free energy of the system be at a minimum, we identify

$$G = \int_{\Omega} g(\phi, \epsilon, \lambda) d\Omega$$

as the electrostatic energy described by the PB equation where

$$g(\phi, \epsilon, \lambda) = \begin{aligned} & -\frac{\epsilon(\mathbf{x})}{2} (\nabla\phi(\mathbf{x}) \cdot \nabla\phi(\mathbf{x})) + 4\pi\phi(\mathbf{x}) \sum_{k=1}^{n_c} q_k \delta(\mathbf{x} - \mathbf{x}_k) \\ & - k_B T \sum_i [z_i c_i e^{-z_i \phi(\mathbf{x})/k_B T}] \lambda(\mathbf{x}) \end{aligned} \quad (1.4)$$

Inspection of the equation (1.4) shows that, in fact, any function  $C_1 g + C_0$  is also a solution. To conform to the usual electrostatic convention we put  $C_1 = \frac{1}{4\pi}$ . The constant  $C_0$  is determined from the requirement that  $G = 0$  when  $\phi = 0$  anywhere such that

$$C_0 = -\frac{1}{4\pi} g|_{\phi(\mathbf{x})=0} = -\frac{k_B T}{4\pi} \sum_i z_i c_i \lambda(\mathbf{x}).$$

Because the equation (1.4) is composed of three terms, the electrostatic potential  $G$  is also divided and defined into three energetic terms.

- the fixed charge energy

$$G^{QF} = \int_{\Omega} \phi(\mathbf{x}) \sum_{k=1}^{n_c} q_k \delta(\mathbf{x} - \mathbf{x}_k) d\Omega = \frac{1}{2} \sum_{k=1}^{n_c} \phi(\mathbf{x}_k) q_k \quad (2.1.1)$$

- the dielectric boundary energy

$$G^{DB} = - \int_{\Omega} \frac{\epsilon(\mathbf{x})}{8\pi} (\nabla \phi(\mathbf{x}) \cdot \nabla \phi(\mathbf{x})) d\Omega \quad (2.1.2)$$

- the ionic boundary energy

$$G^{IB} = -\frac{k_B T}{4\pi} \int_{\Omega} \sum_i [z_i c_i (e^{-z_i \phi(\mathbf{x})/k_B T} - 1)] \lambda(\mathbf{x}) d\Omega \quad (2.1.3)$$

In the fixed charge energy  $G^{QF}$ , the electrostatic potential  $\phi(\mathbf{x}_k)$  is the electrostatic potential due to all point charges except the one  $q_k$  at  $\mathbf{x}_k$  [43].

If we can solve PB equation, we could use the PB electrostatic potential to calculate each term the electrostatic energy  $G = G^{QF} + G^{DB} + G^{IB}$ .

We can apply the linearization process of PB equation on the ionic boundary energy  $G^{IB}$  if the solvent only contains 1 and  $-1$  charges with concentration  $c$ .

$$\begin{aligned}
G^{IB} &= -\frac{k_B T}{4\pi} \int_{\Omega} \sum_i [z_i c_i (e^{-z_i \phi(\mathbf{x})/k_B T} - 1)] \lambda(\mathbf{x}) d\Omega \\
&= -\frac{k_B T}{4\pi} \int_{\Omega} c [e^{\phi(\mathbf{x})/k_B T} - e^{-\phi(\mathbf{x})/k_B T} - 2] \lambda(\mathbf{x}) d\Omega \\
&\approx -\frac{ck_B T}{2\pi} \int_{\Omega} [\phi(\mathbf{x})/k_B T - 1] \lambda(\mathbf{x}) d\Omega \\
&= -\int_{\Omega} [\frac{c}{2\pi} \phi(\mathbf{x}) - \frac{ck_B T}{2\pi}] \lambda(\mathbf{x}) d\Omega
\end{aligned}$$

### 1.5.2 Derivation of polarization forces

Gilson et al [40] derive the electrostatic force from the atomic derivative of the polarization energy  $G$ . The electrostatic force acting on an atom  $\alpha$  of the molecule can be written as the atomic derivative of polarization energy [47].

$$F_{\alpha} = -\frac{\partial}{\partial \mathbf{x}_{\alpha}} G = -\frac{\partial}{\partial \mathbf{x}_{\alpha}} (G^{QF} + G^{DB} + G^{IB})$$

where

$$\begin{aligned}
-\frac{\partial}{\partial \mathbf{x}_{\alpha}} G^{QF} &= \frac{1}{4\pi} \int_{\Omega} \left( \phi(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_{\alpha}} \rho_c(\mathbf{x}) + \frac{\partial}{\partial \mathbf{x}_{\alpha}} \phi(\mathbf{x}) \rho_c(\mathbf{x}) \right) d\Omega \\
-\frac{\partial}{\partial \mathbf{x}_{\alpha}} G^{DB} &= \frac{1}{4\pi} \int_{\Omega} \left[ \frac{\partial}{\partial \mathbf{x}_{\alpha}} \epsilon(\mathbf{x}) (\nabla \phi(\mathbf{x}) \cdot \nabla \phi(\mathbf{x})) + \epsilon(\mathbf{x}) \nabla \phi(\mathbf{x}) \cdot \nabla \left( \frac{\partial}{\partial \mathbf{x}_{\alpha}} \phi(\mathbf{x}) \right) \right] d\Omega \\
-\frac{\partial}{\partial \mathbf{x}_{\alpha}} G^{IB} &= \frac{1}{4\pi} \int_{\Omega} \left\{ k_B T \sum_i [z_i c_i (e^{-z_i \phi(\mathbf{x})/k_B T} - 1)] \frac{\partial}{\partial \mathbf{x}_{\alpha}} \lambda(\mathbf{x}) \right\} d\Omega + \\
&\quad \int_{\Omega} \left\{ k_B T \sum_i [z_i c_i (e^{-z_i \phi(\mathbf{x})/k_B T} - 1)] \lambda(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_{\alpha}} \phi(\mathbf{x}) \right\} d\Omega
\end{aligned}$$

we then integrate the second term in  $-\frac{\partial}{\partial \mathbf{x}_{\alpha}} G^{DB}$  by parts

$$\begin{aligned}
\int_{\Omega} \epsilon(\mathbf{x}) \nabla \phi(\mathbf{x}) \cdot \nabla \left( \frac{\partial}{\partial \mathbf{x}_{\alpha}} \phi(\mathbf{x}) \right) d\Omega &= \int_{\Omega} \nabla \cdot \left( \epsilon(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_{\alpha}} \phi(\mathbf{x}) \cdot \nabla \phi(\mathbf{x}) \right) d\Omega - \\
&\quad \int_{\Omega} \left( \frac{\partial}{\partial \mathbf{x}_{\alpha}} \phi(\mathbf{x}) \right) \nabla \cdot \left( \epsilon(\mathbf{x}) \nabla \phi(\mathbf{x}) \right) d\Omega
\end{aligned}$$

and apply the divergence theorem to convert the first integral on the right-hand side into a vanishing surface integral at infinity.

$$\int_{\Omega} \nabla \cdot \left( \epsilon(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_{\alpha}} \phi(\mathbf{x}) \cdot \nabla \phi(\mathbf{x}) \right) d\Omega = \int_{\partial\Omega} \epsilon(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_{\alpha}} \phi(\mathbf{x}) \cdot \nabla \phi(\mathbf{x}) d\partial\Omega = 0$$

Therefore, we combine  $-\frac{\partial}{\partial \mathbf{x}_{\alpha}} G^{QF}$ ,  $-\frac{\partial}{\partial \mathbf{x}_{\alpha}} G^{DB}$  and  $-\frac{\partial}{\partial \mathbf{x}_{\alpha}} G^{IB}$  together to write the electrostatic force  $F_{\alpha}$

$$F_{\alpha} = \frac{1}{4\pi} \int_{\Omega} \left[ \phi(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_{\alpha}} \rho_c(\mathbf{x}) + \frac{\partial}{\partial \mathbf{x}_{\alpha}} \epsilon(\mathbf{x}) (\nabla \phi(\mathbf{x}) \cdot \nabla \phi(\mathbf{x})) + k_B T \sum_i [z_i c_i (e^{-z_i \phi(\mathbf{x})/k_B T} - 1)] \frac{\partial}{\partial \mathbf{x}_{\alpha}} \lambda(\mathbf{x}) \right] + \frac{\partial}{\partial \mathbf{x}_{\alpha}} \phi(\mathbf{x}) \left[ -\rho_c(\mathbf{x}) + \nabla \cdot (\epsilon(\mathbf{x}) \nabla \phi(\mathbf{x})) + k_B T \sum_i [z_i c_i (e^{-z_i \phi(\mathbf{x})/k_B T} - 1)] \lambda(\mathbf{x}) \right] d\Omega$$

In the previous section, we announce that the potential function  $\phi(\mathbf{x})$  solved from PB equation minimizes the energy  $G$ . This vanishes the second term (PB equation in the brace) on the right hand side. Therefore, the atomic derivative of  $G$  can be written as

$$F_{\alpha} = \frac{1}{4\pi} \int_{\Omega} \left( \phi(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_{\alpha}} \rho_c(\mathbf{x}) \right) d\Omega + \frac{1}{4\pi} \int_{\Omega} \left[ \frac{\partial}{\partial \mathbf{x}_{\alpha}} \epsilon(\mathbf{x}) (\nabla \phi(\mathbf{x}) \cdot \nabla \phi(\mathbf{x})) \right] d\Omega + \int_{\Omega} \left[ k_B T \sum_i [z_i c_i (e^{-z_i \phi(\mathbf{x})/k_B T} - 1)] \frac{\partial}{\partial \mathbf{x}_{\alpha}} \lambda(\mathbf{x}) \right] d\Omega \quad (1.5)$$

These three terms are derived from the atomic derivative of  $G^{QF}$  (2.1.1),  $G^{DB}$  (2.1.2) and  $G^{IB}$  (2.1.3), so the electrostatic force acting on an atom  $\alpha$  can also be separated into three terms.

- the fixed charge force

$$F_{\alpha}^{QF} = \frac{1}{4\pi} \int_{\Omega} \phi(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_{\alpha}} \rho_c(\mathbf{x}) d\Omega \quad (2.2.1)$$

- the dielectric boundary force

$$F_{\alpha}^{DB} = \int_{\Omega} \left( \frac{1}{8\pi} \nabla \phi(\mathbf{x}) \cdot \nabla \phi(\mathbf{x}) \right) \frac{\partial}{\partial \mathbf{x}_{\alpha}} \epsilon(\mathbf{x}) d\Omega \quad (2.2.2)$$

- the ionic boundary force

$$F_\alpha^{IB} = \int_\Omega \frac{k_B T}{4\pi} \sum_i [z_i c_i (e^{-z_i \phi(\mathbf{x})/k_B T} - 1) \frac{\partial}{\partial \mathbf{x}_\alpha} \lambda(\mathbf{x})] d\Omega \quad (2.2.3)$$

where the charge density term is  $\rho_c(\mathbf{x}) = \sum_{k=1}^{n_c} \rho_k(\|\mathbf{x} - \mathbf{x}_k\|)$  and  $\rho_k(\|\mathbf{x} - \mathbf{x}_k\|) = -4\pi q_k \delta(\mathbf{x} - \mathbf{x}_k)$  [47]. Its atomic derivative is

$$\frac{\partial}{\partial \mathbf{x}_\alpha} \rho_c(\mathbf{x}) = -\nabla \rho_\alpha(\|\mathbf{x} - \mathbf{x}_\alpha\|).$$

The fixed charge force (2.2.1) can be simplified by integrating it by parts as follows.

$$\begin{aligned} F_\alpha^{QF} &= \frac{1}{4\pi} \int_\Omega \phi(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_\alpha} \rho_c(\mathbf{x}) d\Omega \\ &= -\frac{1}{4\pi} \int_\Omega \phi(\mathbf{x}) \nabla \rho_\alpha(\|\mathbf{x} - \mathbf{x}_\alpha\|) d\Omega \\ &= \frac{1}{4\pi} \int_\Omega \nabla \phi(\mathbf{x}) \rho_\alpha(\|\mathbf{x} - \mathbf{x}_\alpha\|) d\Omega - \frac{1}{4\pi} \int_\Omega \nabla [\phi(\mathbf{x}) \rho_\alpha(\|\mathbf{x} - \mathbf{x}_\alpha\|)] d\Omega \\ &= \frac{1}{4\pi} \int_\Omega \nabla \phi(\mathbf{x}) \rho_\alpha(\|\mathbf{x} - \mathbf{x}_\alpha\|) d\Omega - \frac{1}{4\pi} \int_{\delta\Omega} \mathbf{n} \cdot [\phi(\mathbf{x}) \rho_\alpha(\|\mathbf{x} - \mathbf{x}_\alpha\|)] d\delta\Omega \\ &= \frac{1}{4\pi} \int_\Omega \nabla \phi(\mathbf{x}) \rho_\alpha(\|\mathbf{x} - \mathbf{x}_\alpha\|) d\Omega \\ &= -q_\alpha \nabla \phi(\mathbf{x}_\alpha) \end{aligned}$$

In the fixed charge energy  $F_\alpha^{QF}$ , the gradient of electrostatic potential  $\nabla \phi(\mathbf{x}_\alpha)$  is due to all point charges except the one  $q_\alpha$  at  $\mathbf{x}_\alpha$  [43].

We can also apply the linearization process of PB equation on the ionic boundary force (2.2.3) as we did for the ionic boundary energy  $G^{IB}$  if the solvent only contains 1 and  $-1$  charges with concentration  $c$ .

$$\begin{aligned} F_\alpha^{IB} &= \int_\Omega \frac{k_B T}{4\pi} \sum_i [z_i c_i (e^{-z_i \phi(\mathbf{x})/k_B T} - 1) \frac{\partial}{\partial \mathbf{x}_\alpha} \lambda(\mathbf{x})] d\Omega \\ &= \int_\Omega \frac{k_B T}{4\pi} c [e^{\phi(\mathbf{x})/k_B T} - e^{-\phi(\mathbf{x})/k_B T} - 2] \frac{\partial}{\partial \mathbf{x}_\alpha} \lambda(\mathbf{x}) d\Omega \\ &\approx \int_\Omega c \frac{k_B T}{2\pi} [\phi(\mathbf{x})/k_B T - 1] \frac{\partial}{\partial \mathbf{x}_\alpha} \lambda(\mathbf{x}) d\Omega \\ &= \int_\Omega [ \frac{c}{2\pi} \phi(\mathbf{x}) - \frac{ck_B T}{2\pi} ] \frac{\partial}{\partial \mathbf{x}_\alpha} \lambda(\mathbf{x}) d\Omega \end{aligned}$$

The electrostatic force  $F_\alpha$  acting on an atom  $\alpha$  of the molecule is the combination of  $F_\alpha^{QF}$ ,  $F_\alpha^{DB}$  and  $F_\alpha^{IB}$ .

$$F_\alpha = F_\alpha^{QF} + F_\alpha^{DB} + F_\alpha^{IB} \quad (1.6)$$

In fact, for an atom of a molecule which doesn't form part of a dielectric or ionic boundary:  $F_\alpha = F_\alpha^{QF}$  and which forms part of a dielectric or ionic boundary:  $F_\alpha = F_\alpha^{QF} + F_\alpha^{DB} + F_\alpha^{IB}$ .

### 1.5.3 Derivation of boundary integral equations

The following boundary integral equations are derived by A.H. Juffer and other researchers [48]. From now on,  $r^-$  will be a point inside the surface  $\Gamma$ ,  $r^+$  will be a point outside  $\Gamma$  and  $r$  will be on  $\Gamma$ .

$$\text{Region } \Omega_{int}: \nabla^2 \phi_I(\mathbf{x}) = \frac{1}{\epsilon_I} \rho_c(\mathbf{x}) = -4\pi \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \delta(\mathbf{x} - \mathbf{x}_k)$$

We apply Green's second identity to this partial differential equation,

$$\int_{\Omega} [\psi \nabla^2 \phi - \phi \nabla^2 \psi] dV = \int_{\Gamma} \left[ \psi \frac{\partial \phi}{\partial \mathbf{n}} - \phi \frac{\partial \psi}{\partial \mathbf{n}} \right] d\Gamma$$

where  $\mathbf{n}$  is the outward unit normal at the point  $y$  on the surface  $\Gamma$  and  $\psi = G_I$  is a fundamental solution of the Poisson equation such that  $G_I(\mathbf{x}, y) = \frac{1}{4\pi \|\mathbf{x} - y\|}$  and  $\nabla^2 G_I(\mathbf{x}, y) = -\delta(\mathbf{x} - y)$ ,

$$\begin{aligned} & \int_{\Omega} \left[ G_I(r^-, y) \left( -4\pi \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \delta(y - \mathbf{x}_k) \right) - \phi_I(y) (-\delta(r^- - y)) \right] d\Omega \\ &= \phi_I(r^-) - \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(r^-, \mathbf{x}_k) \\ &= \int_{\Gamma} \left[ G_I(r^-, y) \frac{\partial \phi_I}{\partial \mathbf{n}}(y) - \phi_I(y) \frac{\partial G_I}{\partial \mathbf{n}}(r^-, y) \right] dy \end{aligned} \quad (1.7)$$

where  $\frac{\partial G_I(\mathbf{x}, y)}{\partial \mathbf{n}} = \frac{-\cos \theta}{4\pi \|\mathbf{x} - y\|^2}$  with  $\cos \theta = \frac{\mathbf{n} \cdot (\mathbf{x} - y)}{\|\mathbf{x} - y\|}$ .

If  $\mathbf{r}^- \rightarrow \mathbf{r}$  on the surface,

$$\phi_I(\mathbf{r}) = \lim_{\mathbf{r}^- \rightarrow \mathbf{r}} \phi_I(\mathbf{r}^-) = \int_{\Gamma} \left[ G_I(\mathbf{r}, \mathbf{y}) \frac{\partial \phi_I}{\partial \mathbf{n}}(\mathbf{y}) \right] dy - \lim_{\mathbf{x} \rightarrow \mathbf{r}} \int_{\Gamma} \left[ \phi_I(\mathbf{y}) \frac{\partial G_I}{\partial \mathbf{n}}(\mathbf{x}, \mathbf{y}) \right] dy + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{r}, \mathbf{x}_k)$$

.

Finally, we got a boundary integral equation from Poisson equation of the region  $\Omega_{int}$ .

$$\frac{1}{2} \phi_I(\mathbf{r}) = \oint_{\Gamma} \left[ G_I(\mathbf{r}, \mathbf{y}) \frac{\partial \phi_I}{\partial \mathbf{n}}(\mathbf{y}) - \phi_I(\mathbf{y}) \frac{\partial G_I}{\partial \mathbf{n}}(\mathbf{r}, \mathbf{y}) \right] dy + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{r}, \mathbf{x}_k) \quad (1.8)$$

where  $\oint$  is the principal value integral.

Region  $\Omega_{ext}$ :  $\nabla^2 \phi_{II}(\mathbf{x}) = \kappa^2 \phi_{II}(\mathbf{x})$ , we apply Green's second identity to this partial differential equation, where  $\psi = G_{II}$  is a fundamental solution of PB equation such that  $G_{II}(\mathbf{x}, \mathbf{y}) = e^{-\kappa \|\mathbf{x}-\mathbf{y}\|} G_I(\mathbf{x}, \mathbf{y}) = \frac{e^{-\kappa \|\mathbf{x}-\mathbf{y}\|}}{4\pi \|\mathbf{x}-\mathbf{y}\|}$  and  $(\nabla^2 - \kappa^2)G_{II}(\mathbf{x}, \mathbf{y}) = -\delta(\mathbf{x} - \mathbf{y})$ ,

$$\int_{\Omega} [G_{II}(\mathbf{r}^+, \mathbf{y}) (\kappa^2 \phi_{II}(\mathbf{y})) - \phi_{II}(\mathbf{y}) (\kappa^2 G_{II}(\mathbf{r}^+, \mathbf{y}) - \delta(\mathbf{r}^+ - \mathbf{y}))] d\Omega = -\phi_{II}(\mathbf{r}^+) = \int_{\Gamma} [G_{II}(\mathbf{r}^+, \mathbf{y}) \frac{\partial \phi_{II}}{\partial \mathbf{n}}(\mathbf{y}) - \phi_{II}(\mathbf{y}) \frac{\partial G_{II}}{\partial \mathbf{n}}(\mathbf{r}^+, \mathbf{y})] dy \quad (1.9)$$

where  $\frac{\partial G_{II}(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}} = \frac{-e^{-\kappa \|\mathbf{x}-\mathbf{y}\|} (1.0 + \kappa \|\mathbf{x}-\mathbf{y}\|) \cos \theta}{4\pi \|\mathbf{x}-\mathbf{y}\|^2}$

If  $\mathbf{r}^+ \rightarrow \mathbf{r}$  on the surface, we got a boundary integral equation from PB equation of the region  $\Omega_{ext}$ ,

$$\frac{1}{2} \phi_{II}(\mathbf{r}) = \oint_{\Gamma} \left[ -G_{II}(\mathbf{r}, \mathbf{y}) \frac{\partial \phi_{II}}{\partial \mathbf{n}}(\mathbf{y}) + \phi_{II}(\mathbf{y}) \frac{\partial G_{II}}{\partial \mathbf{n}}(\mathbf{r}, \mathbf{y}) \right] dy \quad (1.10)$$

where  $\oint$  is the principal value integral.

Note that  $G_I(x, y)$  and  $\frac{\partial G_I}{\partial \mathbf{n}}(x, y)$  are weakly singular in  $y$ , but they are still integrable over the surface  $\Gamma$  and the same holds for  $G_{II}(r, y)$  and  $\frac{\partial G_{II}}{\partial \mathbf{n}}(r, y)$ .

Using the boundary conditions (1.1) and (1.2), and adding (1.8) and  $\epsilon = \frac{\epsilon_{II}}{\epsilon_I}$  (1.10), we obtained the first integral equation:

$$\begin{aligned} \frac{1}{2}(1 + \epsilon)\phi_I(r) &= \int_{\Gamma} [G_I(r, y) - G_{II}(r, y)] \frac{\partial \phi}{\partial \mathbf{n}}(y) dy + \int_{\Gamma} \left[ \epsilon \frac{\partial G_{II}}{\partial \mathbf{n}}(r, y) - \frac{\partial G_I}{\partial \mathbf{n}}(r, y) \right] \phi_I(y) dy \\ &\quad + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(r, \mathbf{x}_k) \end{aligned} \quad (1.11)$$

In order to derive the second integral equation, the normal derivatives in (1.7) and (1.9) are considered. Let  $\mathbf{n}_0$  be the outward unit normal at the point  $r$  on the surface  $\Gamma$ . The normal derivative  $\frac{\partial}{\partial \mathbf{n}_0} = \mathbf{n}_0 \cdot \nabla_{r^\pm}$ , we find

$$\begin{aligned} \frac{\partial \phi_I}{\partial \mathbf{n}_0}(r^-) &= \mathbf{n}_0 \cdot \nabla_{r^-} \phi_I(r^-) \\ &= \mathbf{n}_0 \cdot \left( \frac{\partial \phi_I}{\partial r_x^-}, \frac{\partial \phi_I}{\partial r_y^-}, \frac{\partial \phi_I}{\partial r_z^-} \right)^T \\ &= \int_{\Gamma} \left[ \left( \mathbf{n}_{0x} \frac{\partial G_I}{\partial r_x^-} + \mathbf{n}_{0y} \frac{\partial G_I}{\partial r_y^-} + \mathbf{n}_{0z} \frac{\partial G_I}{\partial r_z^-} \right) (r^-, y) \frac{\partial \phi_I}{\partial \mathbf{n}}(y) \right] dy \\ &\quad - \int_{\Gamma} \left[ \phi_I(y) \left( \mathbf{n}_{0x} \frac{\partial^2 G_I}{\partial r_x^- \partial \mathbf{n}} + \mathbf{n}_{0y} \frac{\partial^2 G_I}{\partial r_y^- \partial \mathbf{n}} + \mathbf{n}_{0z} \frac{\partial^2 G_I}{\partial r_z^- \partial \mathbf{n}} \right) (r^-, y) \right] dy \\ &\quad + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \left( \mathbf{n}_{0x} \frac{\partial G_I}{\partial r_x^-} + \mathbf{n}_{0y} \frac{\partial G_I}{\partial r_y^-} + \mathbf{n}_{0z} \frac{\partial G_I}{\partial r_z^-} \right) (r^-, \mathbf{x}_k) \\ &= \int_{\Gamma} \left[ \frac{\partial G_I}{\partial \mathbf{n}_0}(r^-, y) \frac{\partial \phi_I}{\partial \mathbf{n}}(y) - \phi_I(y) \frac{\partial^2 G_I}{\partial \mathbf{n}_0 \partial \mathbf{n}}(r^-, y) \right] dy + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \frac{\partial G_I}{\partial \mathbf{n}_0}(r^-, \mathbf{x}_k) \end{aligned}$$

and the same holds for the normal derivative of (1.9)

$$\frac{\partial \phi_{II}}{\partial \mathbf{n}_0}(r^+) = \int_{\Gamma} \left[ -\frac{\partial G_{II}}{\partial \mathbf{n}_0}(r^+, y) \frac{\partial \phi_{II}}{\partial \mathbf{n}}(y) + \phi_{II}(y) \frac{\partial^2 G_{II}}{\partial \mathbf{n}_0 \partial \mathbf{n}}(r^+, y) \right] dy$$

where

$$\begin{aligned} \frac{\partial G_I(x, y)}{\partial \mathbf{n}_0} &= \frac{\cos \theta}{4\pi \|x-y\|^2} \\ \frac{\partial G_{II}(x, y)}{\partial \mathbf{n}_0} &= \frac{e^{-\kappa \|x-y\|} (1.0 + \kappa \|x-y\|) \cos \theta}{4\pi \|x-y\|^2} \\ \frac{\partial^2 G_I(x, y)}{\partial \mathbf{n}_0 \partial \mathbf{n}} &= \frac{(\mathbf{n}_0 \cdot \mathbf{n}) - 3 \cos \theta_0 \cos \theta}{4\pi \|x-y\|^3} \\ \frac{\partial^2 G_{II}(x, y)}{\partial \mathbf{n}_0 \partial \mathbf{n}} &= e^{-\kappa \|x-y\|} (1.0 + \kappa \|x-y\|) \frac{G_I(x, y)}{\partial \mathbf{n}_0 \partial \mathbf{n}} - \frac{\kappa^2 e^{-\kappa \|x-y\|}}{4\pi \|x-y\|} \cos \theta_0 \cos \theta \end{aligned}$$

with  $\cos \theta_0 = \frac{\mathbf{n}_0 \cdot (\mathbf{x} - \mathbf{y})}{\|\mathbf{x} - \mathbf{y}\|}$ .

Take the limits for  $\mathbf{r}^- \rightarrow \mathbf{r}$  and  $\mathbf{r}^+ \rightarrow \mathbf{r}$  on the surface in these two equations;

$$\begin{aligned} \lim_{\mathbf{r}^- \rightarrow \mathbf{r}} \frac{\partial \phi_I}{\partial \mathbf{n}_0}(\mathbf{r}^-) = \frac{\partial \phi_I}{\partial \mathbf{n}}(\mathbf{r}) &= \int_{\Gamma} \left[ \frac{\partial G_I}{\partial \mathbf{n}_0}(\mathbf{r}, \mathbf{y}) \frac{\partial \phi_I}{\partial \mathbf{n}}(\mathbf{y}) - \phi_I(\mathbf{y}) \frac{\partial^2 G_I}{\partial \mathbf{n}_0 \partial \mathbf{n}}(\mathbf{r}, \mathbf{y}) \right] d\mathbf{y} + \frac{1}{2} \frac{\partial \phi_I}{\partial \mathbf{n}_0}(\mathbf{r}) \\ &+ \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \frac{\partial G_I}{\partial \mathbf{n}}(\mathbf{r}, \mathbf{x}_k) \end{aligned} \quad (1.12)$$

and

$$\lim_{\mathbf{r}^+ \rightarrow \mathbf{r}} \frac{\partial \phi_{II}}{\partial \mathbf{n}_0}(\mathbf{r}^+) = \frac{\partial \phi_{II}}{\partial \mathbf{n}}(\mathbf{r}) = \int_{\Gamma} \left[ -\frac{\partial G_{II}}{\partial \mathbf{n}_0}(\mathbf{r}, \mathbf{y}) \frac{\partial \phi_{II}}{\partial \mathbf{n}}(\mathbf{y}) + \phi_{II}(\mathbf{y}) \frac{\partial^2 G_{II}}{\partial \mathbf{n}_0 \partial \mathbf{n}}(\mathbf{r}, \mathbf{y}) \right] d\mathbf{y} + \frac{1}{2} \frac{\partial \phi_{II}}{\partial \mathbf{n}}(\mathbf{r}) \quad (1.13)$$

Using the boundary conditions (1.1) and (1.2) and adding (1.12) and  $\frac{1}{\epsilon}$  (1.13), we obtained the second integral equation:

$$\begin{aligned} \frac{1}{2} \left( 1 + \frac{1}{\epsilon} \right) \frac{\partial \phi_I}{\partial \mathbf{n}}(\mathbf{r}) &= \int_{\Gamma} \left[ \left\{ \frac{\partial G_I}{\partial \mathbf{n}_0}(\mathbf{r}, \mathbf{y}) - \frac{\partial G_{II}}{\partial \mathbf{n}_0}(\mathbf{r}, \mathbf{y}) / \epsilon \right\} \frac{\partial \phi_I}{\partial \mathbf{n}}(\mathbf{y}) - \phi_I(\mathbf{y}) \left\{ \frac{\partial^2 G_{II}}{\partial \mathbf{n}_0 \partial \mathbf{n}}(\mathbf{r}, \mathbf{y}) - \frac{\partial^2 G_I}{\partial \mathbf{n}_0 \partial \mathbf{n}}(\mathbf{r}, \mathbf{y}) \right\} \right] d\mathbf{y} \\ &+ \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \frac{\partial G_I}{\partial \mathbf{n}}(\mathbf{r}, \mathbf{x}_k) \end{aligned} \quad (1.14)$$

This is the final pair of boundary integral equations (1.11) and (1.14) which  $\phi_I(\mathbf{r})$  and  $\frac{\partial \phi_I}{\partial \mathbf{n}}(\mathbf{r})$  on the surface  $\Gamma$  can be solved.

In the special case  $\kappa = 0$ , we have  $G_I(\mathbf{r}, \mathbf{y}) = G_{II}(\mathbf{r}, \mathbf{y})$ ; the boundary integral equations contains only  $\phi_I$  as an unknown function.

#### 1.5.4 The derivation of interior and exterior electrostatic potential

Juffer and other researchers [48] also derived the interior and exterior electrostatic potential from PB equation. Using PB equation in the region  $\Omega_{ext}$   $\nabla^2 \phi_{II}(\mathbf{x}) = \kappa^2 \phi_{II}(\mathbf{x})$ , taking an interior point  $\mathbf{r}^-$  and applying Green's

second identity on the volume outside  $\Gamma$  yields

$$\int_{\Gamma} \left[ G_{II}(\mathbf{r}^-, y) \frac{\partial \phi_{II}}{\partial \mathbf{n}}(y) - \frac{\partial G_{II}}{\partial \mathbf{n}}(\mathbf{r}^-, y) \phi_{II}(y) \right] dy = 0$$

Using the boundary conditions (1.1) and (1.2) on the surface  $\Gamma$ , we can rewrite this equations as

$$\int_{\Gamma} \left[ G_{II}(\mathbf{r}^-, y) \frac{\partial \phi_I}{\partial \mathbf{n}}(y) / \epsilon - \frac{\partial G_{II}}{\partial \mathbf{n}}(\mathbf{r}^-, y) \phi_I(y) \right] dy = 0 \quad (1.15)$$

Subtracting  $\epsilon$  times (1.15) from (1.7) results in

$$\begin{aligned} \phi_I(\mathbf{r}^-) &= \int_{\Gamma} [G_I(\mathbf{r}^-, y) - G_{II}(\mathbf{r}^-, y)] \frac{\partial \phi}{\partial \mathbf{n}}(y) dy + \int_{\Gamma} \left[ \epsilon \frac{\partial G_{II}}{\partial \mathbf{n}}(\mathbf{r}^-, y) - \frac{\partial G_I}{\partial \mathbf{n}}(\mathbf{r}^-, y) \right] \phi_I(y) dy \\ &\quad + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{r}^-, \mathbf{x}_k) \end{aligned} \quad (1.16)$$

We wish to derive a similar equation for the potential outside  $\Gamma$ . Using the Poisson equation in the region  $\Omega_{int}$   $\nabla^2 \phi_I(\mathbf{x}) = \rho_c(\mathbf{x}) = -4\pi \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \delta(\mathbf{x} - \mathbf{x}_k)$ , taking an exterior point  $\mathbf{r}^+$  and applying Green's second identity on the volume inside  $\Gamma$  yields

$$\int_{\Gamma} \left[ \frac{\partial G_I}{\partial \mathbf{n}}(\mathbf{r}^+, y) \phi_I(y) - G_I(\mathbf{r}^+, y) \frac{\partial \phi_I}{\partial \mathbf{n}}(y) \right] dy = \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{r}^+, \mathbf{x}_k) \quad (1.17)$$

Using the boundary conditions (1.1) and (1.2) on the surface  $\Gamma$ , we can derive the equation (1.18) from (1.9)

$$\epsilon \phi_{II}(\mathbf{r}^+) = \int_{\Gamma} \left[ -G_{II}(\mathbf{r}^+, y) \frac{\partial \phi_I}{\partial \mathbf{n}}(y) + \epsilon \phi_I(y) \frac{\partial G_{II}}{\partial \mathbf{n}}(\mathbf{r}^+, y) \right] dy \quad (1.18)$$

Adding (1.17) and (1.18) yields

$$\begin{aligned} \epsilon \phi_{II}(\mathbf{r}^+) &= \int_{\Gamma} [G_I(\mathbf{r}^+, y) - G_{II}(\mathbf{r}^+, y)] \frac{\partial \phi_I}{\partial \mathbf{n}}(y) dy + \int_{\Gamma} \left[ \epsilon \frac{\partial G_{II}}{\partial \mathbf{n}}(\mathbf{r}^+, y) - \frac{\partial G_I}{\partial \mathbf{n}}(\mathbf{r}^+, y) \right] \phi_I(y) dy \\ &\quad + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{r}^+, \mathbf{x}_k) \end{aligned} \quad (1.19)$$

The gradient of electrostatic potential  $\nabla_{\mathbf{r}^-}\phi(\mathbf{r}^-)$  and  $\nabla_{\mathbf{r}^+}\phi(\mathbf{r}^+)$  can also be derived from (1.16) and (1.19) such that

$$\begin{aligned}\nabla_{\mathbf{r}^-}\phi_I(\mathbf{r}^-) &= \int_{\Gamma} [\nabla_{\mathbf{r}^-}G_I(\mathbf{r}^-, y) - \nabla_{\mathbf{r}^-}G_{II}(\mathbf{r}^-, y)] \frac{\partial\phi}{\partial\mathbf{n}}(y)dy \\ &+ \int_{\Gamma} \left[ \epsilon \nabla_{\mathbf{r}^-} \frac{\partial G_{II}}{\partial\mathbf{n}}(\mathbf{r}^-, y) - \nabla_{\mathbf{r}^-} \frac{\partial G_I}{\partial\mathbf{n}}(\mathbf{r}^-, y) \right] \phi_I(y)dy \\ &+ \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \nabla_{\mathbf{r}^-} G_I(\mathbf{r}^-, \mathbf{x}_k)\end{aligned}\quad (1.20)$$

and

$$\begin{aligned}\epsilon \nabla_{\mathbf{r}^+}\phi_{II}(\mathbf{r}^+) &= \int_{\Gamma} [\nabla_{\mathbf{r}^+}G_I(\mathbf{r}^+, y) - \nabla_{\mathbf{r}^+}G_{II}(\mathbf{r}^+, y)] \frac{\partial\phi_I}{\partial\mathbf{n}}(y)dy \\ &+ \int_{\Gamma} \left[ \epsilon \nabla_{\mathbf{r}^+} \frac{\partial G_{II}}{\partial\mathbf{n}}(\mathbf{r}^+, y) - \nabla_{\mathbf{r}^+} \frac{\partial G_I}{\partial\mathbf{n}}(\mathbf{r}^+, y) \right] \phi_I(y)dy \\ &+ \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \nabla_{\mathbf{r}^+} G_I(\mathbf{r}^+, \mathbf{x}_k)\end{aligned}\quad (1.21)$$

where

$$\nabla_{\mathbf{r}}G_I(\mathbf{r}, \mathbf{x}_k) = \nabla_{\mathbf{r}}\left(\frac{1}{4\pi\|\mathbf{r} - \mathbf{x}_k\|}\right) = \frac{\mathbf{x}_k - \mathbf{r}}{4\pi\|\mathbf{r} - \mathbf{x}_k\|^3}$$

For computing the electrostatic energy  $G_{pol}$  and force  $F_{pol}$ , one always has to compute the electrostatic potential and gradient of electrostatic potential at the center  $\mathbf{x}_\alpha$  of an atom  $\alpha$ . In  $\Omega_{int}$ , interior electrostatic potential (1.16) and its gradient (1.20) are both singular at this position. Under the assumption that the charge of each atom is treated as point charges, the electrostatic potential  $\phi$  and its gradient  $\nabla\phi$  computed for  $G_{pol}$  and  $F_{pol}$  will be computed due to all the point charges except the atom itself. Under this assumption, we can compute the electrostatic potential and its gradient without the singularity in the nonintegral term  $\sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{x}_\alpha, \mathbf{x}_k)$

- Interior electrostatic potential for  $\mathbf{r} \in \Omega_{int}$

$$\phi_I(\mathbf{r}) = \begin{cases} \int_{\Gamma} [G_I(\mathbf{r}, y) - G_{II}(\mathbf{r}, y)] \frac{\partial\phi}{\partial\mathbf{n}}(y)dy + \int_{\Gamma} \left[ \epsilon \frac{\partial G_{II}}{\partial\mathbf{n}}(\mathbf{r}, y) - \frac{\partial G_I}{\partial\mathbf{n}}(\mathbf{r}, y) \right] \phi_I(y)dy \\ \quad + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{r}, \mathbf{x}_k), & \text{if } \mathbf{r} \notin \{\mathbf{x}_k\}_{k=1}^{n_c}, \\ \int_{\Gamma} [G_I(\mathbf{r}, y) - G_{II}(\mathbf{r}, y)] \frac{\partial\phi}{\partial\mathbf{n}}(y)dy + \int_{\Gamma} \left[ \epsilon \frac{\partial G_{II}}{\partial\mathbf{n}}(\mathbf{r}, y) - \frac{\partial G_I}{\partial\mathbf{n}}(\mathbf{r}, y) \right] \phi_I(y)dy \\ \quad + \sum_{k=1, k \neq \alpha}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{r}, \mathbf{x}_k), & \text{if } \mathbf{r} = \mathbf{x}_\alpha, \alpha = 1, \dots, k. \end{cases}$$

- Exterior electrostatic potential for  $\mathbf{r} \in \Omega_{ext}$

$$\begin{aligned} \epsilon\phi_{II}(\mathbf{r}) &= \int_{\Gamma} [G_I(\mathbf{r}, \mathbf{y}) - G_{II}(\mathbf{r}, \mathbf{y})] \frac{\partial\phi_I}{\partial\mathbf{n}}(\mathbf{y})d\mathbf{y} + \int_{\Gamma} \left[ \epsilon \frac{\partial G_{II}}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) - \frac{\partial G_I}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) \right] \phi_I(\mathbf{y})d\mathbf{y} \\ &\quad + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{r}, \mathbf{x}_k) \end{aligned}$$

- Interior gradient of electrostatic potential for  $\mathbf{r} \in \Omega_{int}$

$$\nabla_{\mathbf{r}}\phi_I(\mathbf{r}) = \begin{cases} \int_{\Gamma} [\nabla_{\mathbf{r}}G_I(\mathbf{r}, \mathbf{y}) - \nabla_{\mathbf{r}}G_{II}(\mathbf{r}, \mathbf{y})] \frac{\partial\phi}{\partial\mathbf{n}}(\mathbf{y})d\mathbf{y} \\ + \int_{\Gamma} \left[ \epsilon \nabla_{\mathbf{r}} \frac{\partial G_{II}}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) - \nabla_{\mathbf{r}} \frac{\partial G_I}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) \right] \phi_I(\mathbf{y})d\mathbf{y} \\ + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \nabla_{\mathbf{r}}G_I(\mathbf{r}, \mathbf{x}_k), & \text{if } \mathbf{r} \notin \{\mathbf{x}_k\}_{k=1}^{n_c}, \\ \int_{\Gamma} [\nabla_{\mathbf{r}}G_I(\mathbf{r}, \mathbf{y}) - \nabla_{\mathbf{r}}G_{II}(\mathbf{r}, \mathbf{y})] \frac{\partial\phi}{\partial\mathbf{n}}(\mathbf{y})d\mathbf{y} \\ + \int_{\Gamma} \left[ \epsilon \nabla_{\mathbf{r}} \frac{\partial G_{II}}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) - \nabla_{\mathbf{r}} \frac{\partial G_I}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) \right] \phi_I(\mathbf{y})d\mathbf{y} \\ + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \nabla_{\mathbf{r}}G_I(\mathbf{r}, \mathbf{x}_k), & \text{if } \mathbf{r} = \mathbf{x}_{\alpha}, \alpha = 1, \dots, k. \end{cases}$$

- Exterior gradient of electrostatic potential for  $\mathbf{r} \in \Omega_{ext}$

$$\begin{aligned} \epsilon\nabla_{\mathbf{r}}\phi_{II}(\mathbf{r}) &= \int_{\Gamma} [\nabla_{\mathbf{r}}G_I(\mathbf{r}, \mathbf{y}) - \nabla_{\mathbf{r}}G_{II}(\mathbf{r}, \mathbf{y})] \frac{\partial\phi_I}{\partial\mathbf{n}}(\mathbf{y})d\mathbf{y} \\ &\quad + \int_{\Gamma} \left[ \epsilon \nabla_{\mathbf{r}} \frac{\partial G_{II}}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) - \nabla_{\mathbf{r}} \frac{\partial G_I}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) \right] \phi_I(\mathbf{y})d\mathbf{y} \\ &\quad + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \nabla_{\mathbf{r}}G_I(\mathbf{r}, \mathbf{x}_k) \end{aligned}$$

### 1.5.5 New Boundary integral equations

Juffer derived the derivative boundary integral equations which satisfy the regularity condition at infinity, i.e.,  $|r|\phi(r)$  and  $|r|^2\nabla\phi(r)$  are bounded for  $|r|$  tending to infinity. Instead of dealing with the regularity condition, we reformulate the boundary integral equation 1.9 with additional outer domain boundary condition.

In  $\Omega_{II}$ , the electrostatic potential is  $\nabla^2\phi_{II}(x) = \rho_b(x)/\epsilon_{II}$ . We apply Green's second identity to this partial differential equation,

$$\int_{\Omega_{II}} [\psi\nabla^2\phi - \phi\nabla^2\psi] d\Omega = \int_{\partial\Omega} \left[ \psi\frac{\partial\phi}{\partial n} - \phi\frac{\partial\psi}{\partial n} \right] d\partial\Omega + \int_{\Gamma} \left[ \psi\frac{\partial\phi}{\partial n} - \phi\frac{\partial\psi}{\partial n} \right] d\Gamma$$

where  $n$  is the outward unit normal at the point  $y$  on the boundary of  $\Omega_{II}$ ,  $\partial\Omega$  and  $\Gamma$ .  $\psi = G_{II}$  is a fundamental solution of PB equation such that  $G_{II}(x, y) = e^{-\kappa\|x-y\|}G_I(x, y) = \frac{e^{-\kappa\|x-y\|}}{4\pi\|x-y\|}$  and  $(\nabla^2 - \kappa^2)G_{II}(x, y) = -\delta(x - y)$ ,

$$\begin{aligned} & \int_{\Omega} [G_{II}(r^+, y) (\kappa^2\phi_{II}(y)) - \phi_{II}(y) (\kappa^2G_{II}(r^+, y) - \delta(r^+ - y))] d\Omega \\ & = -\phi_{II}(r^+) = H_{\partial\Omega}(r^+) + \int_{\Gamma} [G_{II}(r^+, y)\frac{\partial\phi_{II}}{\partial n}(y) - \phi_{II}(y)\frac{\partial G_{II}}{\partial n}(r^+, y)] dy \end{aligned} \quad (1.22)$$

where  $\frac{\partial G_{II}(x, y)}{\partial n} = \frac{-e^{-\kappa\|x-y\|}(1.0+\kappa\|x-y\|)\cos\theta}{4\pi\|x-y\|^2}$

where

$$H_{\partial\Omega}(r^+) = \int_{\partial\Omega} \left[ g(y)\frac{\partial G_{II}}{\partial n}(r^+, y) - G_{II}(r^+, y)\frac{\partial g}{\partial n}(y) \right] dy$$

We then apply Juffer's reformulation steps to get a well-conditioned

boundary integral equations as follows.

$$\begin{aligned} \frac{1}{2}(1 + \epsilon)\phi_I(\mathbf{r}) &= \int_{\Gamma} [G_I(\mathbf{r}, \mathbf{y}) - G_{II}(\mathbf{r}, \mathbf{y})] \frac{\partial\phi}{\partial\mathbf{n}}(\mathbf{y})d\mathbf{y} + \int_{\Gamma} \left[ \epsilon \frac{\partial G_{II}}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) - \frac{\partial G_I}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) \right] \phi_I(\mathbf{y})d\mathbf{y} \\ &\quad + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{r}, \mathbf{x}_k) + \epsilon H_{\partial\Omega}(\mathbf{r}) \end{aligned} \quad (1.23)$$

and

$$\begin{aligned} \frac{1}{2}\left(1 + \frac{1}{\epsilon}\right) \frac{\partial\phi_I}{\partial\mathbf{n}}(\mathbf{r}) &= \int_{\Gamma} \left[ \left\{ \frac{\partial G_I}{\partial\mathbf{n}_0}(\mathbf{r}, \mathbf{y}) - \frac{\partial G_{II}}{\partial\mathbf{n}_0}(\mathbf{r}, \mathbf{y})/\epsilon \right\} \frac{\partial\phi_I}{\partial\mathbf{n}}(\mathbf{y}) - \phi_I(\mathbf{y}) \left\{ \frac{\partial^2 G_{II}}{\partial\mathbf{n}_0\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) - \frac{\partial^2 G_I}{\partial\mathbf{n}_0\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) \right\} \right] d\mathbf{y} \\ &\quad + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \frac{\partial G_I}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{x}_k) + \frac{1}{\epsilon} \frac{\partial H_{\partial\Omega}}{\partial\mathbf{n}}(\mathbf{r}) \end{aligned} \quad (1.24)$$

where

$$\frac{\partial H_{\partial\Omega}}{\partial\mathbf{n}}(\mathbf{r}) = \int_{\partial\Omega} \left[ g(\mathbf{y}) \frac{\partial^2 G_{II}}{\partial\mathbf{n}\partial\mathbf{n}_0}(\mathbf{r}, \mathbf{y}) - \frac{\partial G_{II}}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) \frac{\partial g}{\partial\mathbf{n}}(\mathbf{y}) \right] d\mathbf{y}$$

The interior and exterior electrostatic potential and gradient of electrostatic potential are then defined.

- Interior electrostatic potential for  $\mathbf{r} \in \Omega_{int}$

$$\phi_I(\mathbf{r}) = \begin{cases} \int_{\Gamma} [G_I(\mathbf{r}, \mathbf{y}) - G_{II}(\mathbf{r}, \mathbf{y})] \frac{\partial\phi}{\partial\mathbf{n}}(\mathbf{y})d\mathbf{y} + \int_{\Gamma} \left[ \epsilon \frac{\partial G_{II}}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) - \frac{\partial G_I}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) \right] \phi_I(\mathbf{y})d\mathbf{y} \\ \quad + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{r}, \mathbf{x}_k), & \text{if } \mathbf{r} \notin \{\mathbf{x}_k\}_{k=1}^{n_c}, \\ \int_{\Gamma} [G_I(\mathbf{r}, \mathbf{y}) - G_{II}(\mathbf{r}, \mathbf{y})] \frac{\partial\phi}{\partial\mathbf{n}}(\mathbf{y})d\mathbf{y} + \int_{\Gamma} \left[ \epsilon \frac{\partial G_{II}}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) - \frac{\partial G_I}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) \right] \phi_I(\mathbf{y})d\mathbf{y} \\ \quad + \sum_{k=1, k \neq \alpha}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{r}, \mathbf{x}_k), & \text{if } \mathbf{r} = \mathbf{x}_\alpha, \alpha = 1, \dots, k. \end{cases}$$

- Exterior electrostatic potential for  $\mathbf{r} \in \Omega_{ext}$

$$\begin{aligned} \frac{\epsilon_{II}}{\epsilon_I} \phi(\mathbf{r}) &= \int_{\Gamma} [G_0(\mathbf{r}, \mathbf{y}) - G_\kappa(\mathbf{r}, \mathbf{y})] \frac{\partial\phi}{\partial\mathbf{n}}(\mathbf{y})d\mathbf{y} + \int_{\Gamma} \left[ \frac{\epsilon_{II}}{\epsilon_I} \frac{\partial G_\kappa}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) - \frac{\partial G_0}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) \right] \phi(\mathbf{y})d\mathbf{y} \\ &\quad + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_0(\mathbf{r}, \mathbf{x}_k) + \frac{\epsilon_{II}}{\epsilon_I} H_{\partial\Omega}(\mathbf{r}) \end{aligned}$$

- Interior gradient of electrostatic potential for  $\mathbf{r} \in \Omega_{int}$

$$\nabla_{\mathbf{r}} \phi_I(\mathbf{r}) = \begin{cases} \int_{\Gamma} [\nabla_{\mathbf{r}} G_I(\mathbf{r}, \mathbf{y}) - \nabla_{\mathbf{r}} G_{II}(\mathbf{r}, \mathbf{y})] \frac{\partial\phi}{\partial\mathbf{n}}(\mathbf{y})d\mathbf{y} \\ \quad + \int_{\Gamma} \left[ \epsilon \nabla_{\mathbf{r}} \frac{\partial G_{II}}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) - \nabla_{\mathbf{r}} \frac{\partial G_I}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) \right] \phi_I(\mathbf{y})d\mathbf{y} \\ \quad + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \nabla_{\mathbf{r}} G_I(\mathbf{r}, \mathbf{x}_k), & \text{if } \mathbf{r} \notin \{\mathbf{x}_k\}_{k=1}^{n_c}, \\ \int_{\Gamma} [\nabla_{\mathbf{r}} G_I(\mathbf{r}, \mathbf{y}) - \nabla_{\mathbf{r}} G_{II}(\mathbf{r}, \mathbf{y})] \frac{\partial\phi}{\partial\mathbf{n}}(\mathbf{y})d\mathbf{y} \\ \quad + \int_{\Gamma} \left[ \epsilon \nabla_{\mathbf{r}} \frac{\partial G_{II}}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) - \nabla_{\mathbf{r}} \frac{\partial G_I}{\partial\mathbf{n}}(\mathbf{r}, \mathbf{y}) \right] \phi_I(\mathbf{y})d\mathbf{y} \\ \quad + \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \nabla_{\mathbf{r}} G_I(\mathbf{r}, \mathbf{x}_k), & \text{if } \mathbf{r} = \mathbf{x}_\alpha, \alpha = 1, \dots, k. \end{cases}$$

- Exterior gradient of electrostatic potential for  $r \in \Omega_{ext}$

$$\begin{aligned}
\frac{\epsilon_{II}}{\epsilon_I} \nabla_r \phi(r) &= \int_{\Gamma} [\nabla_r G_0(r, y) - \nabla_r G_{\kappa}(r, y)] \frac{\partial \phi}{\partial n}(y) dy \\
&+ \int_{\Gamma} \left[ \frac{\epsilon_{II}}{\epsilon_I} \nabla_r \frac{\partial G_{\kappa}}{\partial n}(r, y) - \nabla_r \frac{\partial G_0}{\partial n}(r, y) \right] \phi(y) dy \\
&+ \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \nabla_r G_0(r, x_k) + \frac{\epsilon_{II}}{\epsilon_I} \nabla_r H_{\partial\Omega}(r)
\end{aligned}$$

## Chapter 2

### Efficient and Consistent Computation for Electrostatic Potential and its Gradient

## 2.1 Implementation

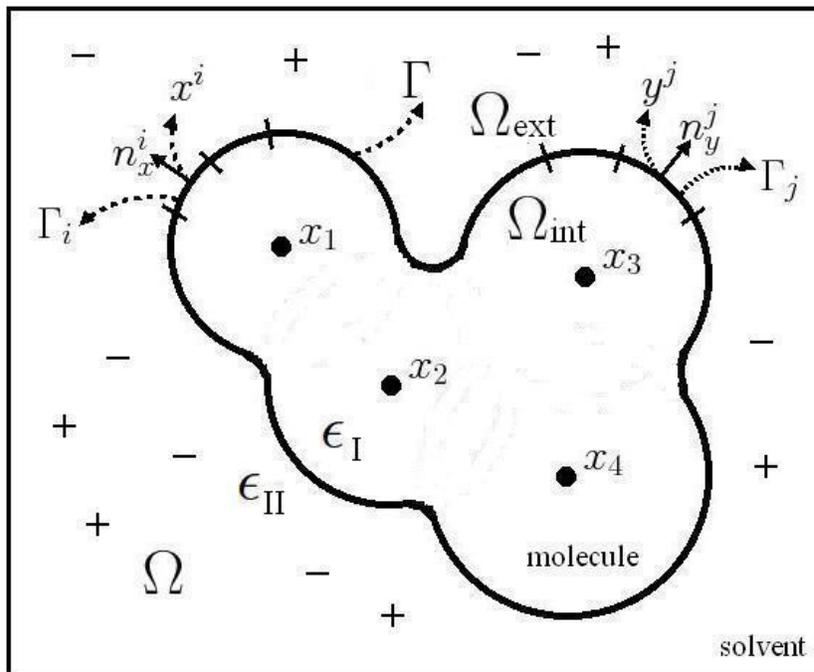


Figure 2.1: The example of boundary element decomposition of a four-atom molecule:  $x_k$  is the center of  $k^{th}$  atom,  $x_i$  and  $y_j$  are the points on the elements  $\Gamma_i$  and  $\Gamma_j$  of the surface  $\Gamma$  and  $n_i^{(x)}$  and  $n_j^{(y)}$  are their normal vectors.

Figure 2.1 shows an example of boundary element decomposition of a four-atom molecule. Each sphere represents an atom with its center  $x_k$  and the boundary surrounding the spheres is the molecular surface  $\Gamma$ . In order to numerically solve PB equation and energies and forces derived from PB equation, we discretized  $\Gamma$  into boundary elements  $\Gamma_i$ ,  $i = 1, \dots, L$  where  $x_i$  represents a point on an element  $\Gamma_i$  and  $y_j$  represents a point on another element  $\Gamma_j$ . Their normal vectors are written as  $n_i^{(x)}$  and  $n_j^{(y)}$ . These notations are used to

formulate the numerical solutions of PB boundary integral equations, energies and forces.

In this section, we first introduce the dielectric and ionic boundary model and their derivatives. Then, the numerical formulation of electrostatic energy and forces are described. Finally, we explain our methodology for solving PB boundary integral equations with higher order surface integral.

### **2.1.1 The implementation of electrostatic potential and its gradient**

#### **2.1.1.1 The surface integral computation using higher order boundary elements ASMS**

In order to calculate PB energy and force, PB boundary integral equations are solved to obtain  $\phi$  and  $\nabla\phi$ . In the boundary integral equations derived from PB equation, one important issue is to evaluate the surface integral accurately and efficiently. The numerical computation of the surface integral depends on the parametrization of triangular elements. Here, we compare the evaluation of the integrals of kernel functions between planar linear elements and higher-order algebraic elements. We do the parametrization on the triangulation of the molecular surface. The triangulation of the surface is composed of the vertices  $V = \{v_i\}_{i=1}^P$  with their unit normal vectors  $\{n_i\}_{i=1}^P$  and the triangular elements  $\Gamma = \{\Gamma_j | \Gamma_j = v_{j1}v_{j2}v_{j3} \text{ where } j = 1, \dots, L \text{ and } v_{j1}, v_{j2}, v_{j3} \in V\}$ .

Algebraic patches or A-patches are a kind of low degree algebraic surface finite elements with dual implicit and rational parametric representations

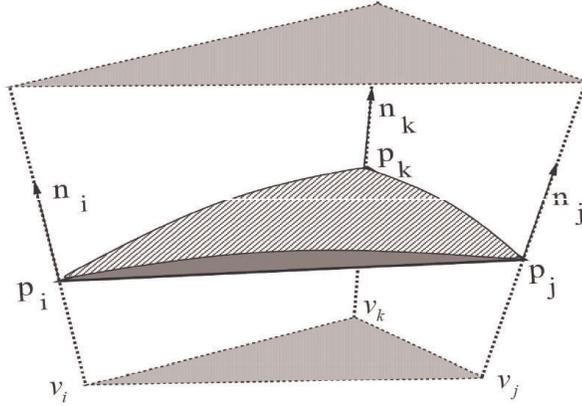


Figure 2.2: A prism scaffold of triangular element  $v_i v_j v_k$

[8]. The A-patch element is defined within a prism scaffold as shown in Figure 2.2. For some triangle element  $\Gamma_j = v_{j1} v_{j2} v_{j3}$  of a triangulation of the molecular surface, the A-patch  $\{\bar{\Gamma}_j\}_{j=1}^L$  is defined on this prism.

$$v_{jl}(\lambda) = v_{jl} + \lambda n_{jl}, \quad l = 1, 2, 3$$

where the prism is defined by

$$D(\Gamma_j) := \{y : y = b_1 v_{j1}(\lambda) + b_2 v_{j2}(\lambda) + b_3 v_{j3}(\lambda), 0 \leq \lambda \leq 1\}$$

where  $(b_1, b_2, b_3)$  are the barycentric coordinates of points in  $v_{j1} v_{j2} v_{j3}$ .

According to the definition of algebraic patches, we define an implicit function over the prism  $D(\Gamma_j)$  in Bernstein-Bezier spline form.

$$F_d(b_1, b_2, b_3, \lambda) = \sum_{i+j+k=d} b_{ijk}(\lambda) B_{ijk}^d(b_1, b_2, b_3)$$

$$B_{ijk}^d(b_1, b_2, b_3) = \frac{d!}{i!j!k!} b_1^i b_2^j b_3^k$$

which is also called the algebraic spline model. The details of the parametrization of algebraic spline model are in [81]. The molecular surface  $\Gamma$  can be approximated by the zero contour of the implicit function  $F_d$ :

$$\{(b_1, b_2, b_3, \lambda) : F_d(b_1, b_2, b_3, \lambda) = 0\}$$

Now, given the barycentric coordinates  $(b_1, b_2, b_3)^T$  on the triangle, the parametric form of the position  $y_j$  on the A-patch element  $\bar{\Gamma}_j = v_{j1}v_{j2}v_{j3}$  is

$$\begin{bmatrix} y_j \\ 1 \end{bmatrix} = \begin{bmatrix} v_{j1}(\lambda) & v_{j2}(\lambda) & v_{j3}(\lambda) \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

As what we did for linear element, we can also approximate those integrals using Gaussian quadrature on the A-patches [81],

$$\int_{\bar{\Gamma}} G(x, y) f(y) dy = \sum_{j=1}^L \int_{\bar{\Gamma}_j} G(x, y_j) f(y_j) dy_j = \sum_{j=1}^L \sum_{m=1}^M W_m G(x, y_{jm}) f(y_{jm}) J(\bar{\Gamma}_j)$$

where  $x$  is the evaluation point.  $\bar{\Gamma}_j$  is the zero contour of the cubic Bezier basis over  $j^{th}$  triangle where  $W_m$  and  $y_{jm} = b_{m1}v_{j1}(\lambda_{jm}) + b_{m2}v_{j2}(\lambda_{jm}) + b_{m3}v_{j3}(\lambda_{jm})$  are the  $m^{th}$  weight and points of Gaussian quadrature on this patch  $\bar{\Gamma}_j$ . The Jacobian weight  $J(\bar{\Gamma}_j)$  is described in the appendix as the area of the patch.

### 2.1.1.2 The implementation of solving Juffer's Poisson-Boltzmann boundary integral equations

We apply Gaussian quadrature on the A-patches to numerically compute (1.11) and (1.14) which are also called derivative boundary integral equa-

tions (dBIEs) is

$$\begin{aligned}
& \frac{1}{2} \left(1 + \frac{\epsilon_{II}}{\epsilon_I}\right) \phi(\mathbf{x}_i) \\
& + \sum_{j=1}^L \sum_{m=1}^M \phi(y_{jm}) W_m \left( \frac{\partial G_I}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{x}_i, y_{jm}) - \frac{\epsilon_{II}}{\epsilon_I} \frac{\partial G_{II}}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{x}_i, y_{jm}) \right) J(y_{jm}) \\
& - \sum_{j=1}^L \sum_{m=1}^M \frac{\partial \phi}{\partial \mathbf{n}_{jm}^{(y)}}(y_{jm}) W_m \left( G_I(\mathbf{x}_i, y_{jm}) - G_{II}(\mathbf{x}_i, y_{jm}) \right) J(y_{jm}) \\
& = \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{x}_i, \mathbf{x}_k) \\
& \frac{1}{2} \left(1 + \frac{\epsilon_I}{\epsilon_{II}}\right) \frac{\partial \phi}{\partial \mathbf{n}_i^x}(\mathbf{x}_i) \\
& - \sum_{j=1}^L \sum_{m=1}^M \phi(y_{jm}) W_m \left( \frac{\partial^2 G_I}{\partial \mathbf{n}_{jm}^{(y)} \partial \mathbf{n}_i^x}(\mathbf{x}_i, y_{jm}) - \frac{\partial^2 G_{II}}{\partial \mathbf{n}_{jm}^{(y)} \partial \mathbf{n}_i^x}(\mathbf{x}_i, y_{jm}) \right) J(y_{jm}) \\
& + \sum_{j=1}^L \sum_{m=1}^M \frac{\partial \phi}{\partial \mathbf{n}_{jm}^{(x)}}(y_{jm}) W_m \left( \frac{\partial G_I}{\partial \mathbf{n}_i^x}(\mathbf{x}_i, y_{jm}) - \frac{\epsilon_I}{\epsilon_{II}} \frac{\partial G_{II}}{\partial \mathbf{n}_i^x}(\mathbf{x}_i, y_{jm}) \right) J(y_{jm}) \\
& = \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \frac{G_I}{\mathbf{n}_i^x}(\mathbf{x}_i, \mathbf{x}_k).
\end{aligned}$$

where  $L$  is the number of patches,  $y_{jm} = \sum_{i=1}^3 b_{mi} \mathbf{v}_{ji}(\lambda_{jm})$  represents the  $m^{\text{th}}$  integration point on the element  $\Gamma_j$ .

$$\begin{aligned}
G_I(\mathbf{x}_i, y_{jm}) &= \frac{1}{4\pi \|\mathbf{x}_i - y_{jm}\|} \\
G_{II}(\mathbf{x}_i, y_{jm}) &= \frac{e^{-\kappa \|\mathbf{x}_i - y_{jm}\|}}{4\pi \|\mathbf{x}_i - y_{jm}\|} \\
\frac{\partial G_I(\mathbf{x}_i, y_{jm})}{\partial \mathbf{n}_{jm}^{(y)}} &= \frac{-(\mathbf{x}_i - y_{jm}) \cdot \mathbf{n}_{jm}^{(y)}}{4\pi \|\mathbf{x}_i - y_{jm}\|^3} \\
\frac{\partial G_{II}(\mathbf{x}_i, y_{jm})}{\partial \mathbf{n}_{jm}^{(y)}} &= \frac{-e^{-\kappa \|\mathbf{x}_i - y_{jm}\|} (1 + \kappa \|\mathbf{x}_i - y_{jm}\|) (\mathbf{x}_i - y_{jm}) \cdot \mathbf{n}_{jm}^{(y)}}{4\pi \|\mathbf{x}_i - y_{jm}\|^3},
\end{aligned}$$

The boundary integral equations are treated as a linear system For better elaboration, we can write dBIEs in the following matrix form.

$$\begin{bmatrix} \frac{1}{2} \left(1 + \epsilon\right) I + \frac{\partial G_{II}}{\partial \mathbf{n}^{(y)}} - \epsilon \frac{\partial G_I}{\partial \mathbf{n}^{(y)}} & G_I - G_{II} \\ \frac{\partial^2 G_I}{\partial \mathbf{n}^{(y)} \partial \mathbf{n}^{(x)}} - \frac{\partial^2 G_{II}}{\partial \mathbf{n}^{(y)} \partial \mathbf{n}^{(x)}} & \frac{1}{2} \left(1 + \frac{1}{\epsilon}\right) I + \frac{\partial G_{II}}{\partial \mathbf{n}^{(x)}} - \frac{\epsilon_I}{\epsilon_{II}} \frac{\partial G_{II}}{\partial \mathbf{n}^{(x)}} \end{bmatrix} \begin{bmatrix} \phi \\ \frac{\partial \phi}{\partial \mathbf{n}^{(y)}} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_{I,k} \\ \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \frac{\partial G_{I,k}}{\partial \mathbf{n}^{(x)}} \end{bmatrix} \quad (2.1)$$

where

- $\phi_j$  and  $\left(\frac{\partial \phi}{\partial \mathbf{n}}\right)_j$  are the  $j^{\text{th}}$  unknown electrostatic potential and its normal derivative at some point  $y_j$  on the patch  $\bar{\Gamma}_j$

- $I$  is the identity operator so that  $I_{ij}\phi_j = \phi_j$
- The operators compute the potential at the point  $\mathbf{x}_i$  due to the patch  $\bar{\Gamma}_j$

$$\begin{aligned}
\left(\frac{\partial G_I}{\partial \mathbf{n}^{(y)}}\right)_{ij} \phi_j &= \int_{\bar{\Gamma}_j} \frac{\partial G_I}{\partial \mathbf{n}_j^{(y)}}(\mathbf{x}_i, y_j) \phi(y_j) dy_j = \sum_{m=1}^M W_m \frac{\partial G_I}{\partial \mathbf{n}_{j_m}^{(y)}}(\mathbf{x}_i, y_{j_m}) \phi(y_{j_m}) J(\bar{\Gamma}_j), \\
(G_I)_{ij} \left(\frac{\partial \phi}{\partial \mathbf{n}^{(y)}}\right)_j &= \int_{\bar{\Gamma}_j} G_I(\mathbf{x}_i, y_j) \frac{\partial \phi(y_j)}{\partial \mathbf{n}_j^{(y)}} dy_j = \sum_{m=1}^M W_m G_I(\mathbf{x}_i, y_{j_m}) \frac{\partial \phi(y_{j_m})}{\partial \mathbf{n}_{j_m}^{(y)}} J(\bar{\Gamma}_j), \\
\left(\frac{\partial G_{II}}{\partial \mathbf{n}^{(y)}}\right)_{ij} \phi_j &= \int_{\bar{\Gamma}_j} \frac{\partial G_{II}}{\partial \mathbf{n}_j^{(y)}}(\mathbf{x}_i, y_j) \phi(y_j) dy_j = \sum_{m=1}^M W_m \frac{\partial G_{II}}{\partial \mathbf{n}_{j_m}^{(y)}}(\mathbf{x}_i, y_{j_m}) \phi(y_{j_m}) J(\bar{\Gamma}_j), \\
(G_{II})_{ij} \left(\frac{\partial \phi}{\partial \mathbf{n}^{(y)}}\right)_j &= \int_{\bar{\Gamma}_j} G_{II}(\mathbf{x}_i, y_j) \frac{\partial \phi(y_j)}{\partial \mathbf{n}_j^{(y)}} dy_j = \sum_{m=1}^M W_m G_{II}(\mathbf{x}_i, y_{j_m}) \frac{\partial \phi(y_{j_m})}{\partial \mathbf{n}_{j_m}^{(y)}} J(\bar{\Gamma}_j), \\
\left(\frac{\partial G_I}{\partial \mathbf{n}^{(x)}}\right)_{ij} \left(\frac{\partial \phi}{\partial \mathbf{n}^{(y)}}\right)_j &= \int_{\bar{\Gamma}_j} \frac{\partial G_I}{\partial \mathbf{n}_i^{(x)}}(\mathbf{x}_i, y_j) \frac{\partial \phi(y_j)}{\partial \mathbf{n}_j^{(y)}} dy_j = \sum_{m=1}^M W_m \frac{\partial G_I}{\partial \mathbf{n}_i^{(x)}}(\mathbf{x}_i, y_{j_m}) \frac{\partial \phi(y_{j_m})}{\partial \mathbf{n}_{j_m}^{(y)}} J(\bar{\Gamma}_j), \\
\left(\frac{\partial^2 G_I}{\partial \mathbf{n}^{(x)} \partial \mathbf{n}^{(y)}}\right)_{ij} \phi_j &= \int_{\bar{\Gamma}_j} \frac{\partial^2 G_I}{\partial \mathbf{n}_i^{(x)} \partial \mathbf{n}_j^{(y)}}(\mathbf{x}_i, y_j) \phi(y_j) dy_j = \sum_{m=1}^M W_m \frac{\partial^2 G_I}{\partial \mathbf{n}_i^{(x)} \partial \mathbf{n}_{j_m}^{(y)}}(\mathbf{x}_i, y_{j_m}) \phi(y_{j_m}) J(\bar{\Gamma}_j), \\
\left(\frac{\partial G_{II}}{\partial \mathbf{n}^{(x)}}\right)_{ij} \left(\frac{\partial \phi}{\partial \mathbf{n}^{(y)}}\right)_j &= \int_{\bar{\Gamma}_j} \frac{\partial G_{II}}{\partial \mathbf{n}_i^{(x)}}(\mathbf{x}_i, y_j) \frac{\partial \phi(y_j)}{\partial \mathbf{n}_j^{(y)}} dy_j = \sum_{m=1}^M W_m \frac{\partial G_{II}}{\partial \mathbf{n}_i^{(x)}}(\mathbf{x}_i, y_{j_m}) \frac{\partial \phi(y_{j_m})}{\partial \mathbf{n}_{j_m}^{(y)}} J(\bar{\Gamma}_j), \\
\left(\frac{\partial^2 G_{II}}{\partial \mathbf{n}^{(x)} \partial \mathbf{n}^{(y)}}\right)_{ij} \phi_j &= \int_{\bar{\Gamma}_j} \frac{\partial^2 G_{II}}{\partial \mathbf{n}_i^{(x)} \partial \mathbf{n}_j^{(y)}}(\mathbf{x}_i, y_j) \phi(y_j) dy_j = \sum_{m=1}^M W_m \frac{\partial^2 G_{II}}{\partial \mathbf{n}_i^{(x)} \partial \mathbf{n}_{j_m}^{(y)}}(\mathbf{x}_i, y_{j_m}) \phi(y_{j_m}) J(\bar{\Gamma}_j), \\
(G_{I,k})_i &= G_I(\mathbf{x}_i, \mathbf{x}_k), \\
\left(\frac{\partial G_{I,k}}{\partial \mathbf{n}^{(x)}}\right)_i &= \frac{\partial G_I}{\partial \mathbf{n}_i^{(x)}}(\mathbf{x}_i, \mathbf{x}_k).
\end{aligned}$$

where  $\mathbf{x}_i$  is a point on the patch  $\bar{\Gamma}_i$  and  $y_{j_m}$  is  $m^{\text{th}}$  Gaussian quadrature point on the patch  $\bar{\Gamma}_j$ .

In our previous manuscript [14], we presented our solution for efficiently solving the linear system (2.1) is solved using generalized minimal residual iterative method (GMRES) with fast multipole algorithm for speeding up the evaluation of matrix-vector production.

### 2.1.1.3 The implementation of computing electrostatic potential and its gradient

We apply Gaussian quadrature on the A-patches to compute the interior electrostatic potential (1.16) at a point  $\mathbf{r}^-$ .

$$\begin{aligned}\phi(\mathbf{r}^-) &= \sum_{j=1}^L \sum_{m=1}^M \frac{\partial \phi}{\partial \mathbf{n}_{jm}^{(y)}}(y_{jm}) W_m (G_I(\mathbf{r}^-, y_{jm}) - G_{II}(\mathbf{r}^-, y_{jm})) J(y_{jm}) \\ &+ \sum_{j=1}^L \sum_{m=1}^M \phi(y_{jm}) W_m \left( \frac{\epsilon_{II}}{\epsilon_I} \frac{\partial G_{II}}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{r}^-, y_{jm}) - \frac{\partial G_I}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{r}^-, y_{jm}) \right) J(y_{jm}) \\ &+ \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{r}^-, \mathbf{x}_k)\end{aligned}$$

and the exterior electrostatic potential (1.19) at a point  $\mathbf{r}^+$

$$\begin{aligned}\frac{\epsilon_{II}}{\epsilon_I} \phi(\mathbf{r}^+) &= \sum_{j=1}^L \sum_{m=1}^M \frac{\partial \phi}{\partial \mathbf{n}_{jm}^{(y)}}(y_{jm}) W_m (G_I(\mathbf{r}^+, y_{jm}) - G_{II}(\mathbf{r}^+, y_{jm})) J(y_{jm}) \\ &+ \sum_{j=1}^L \sum_{m=1}^M \phi(y_{jm}) W_m \left( \frac{\epsilon_{II}}{\epsilon_I} \frac{\partial G_{II}}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{r}^+, y_{jm}) - \frac{\partial G_I}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{r}^+, y_{jm}) \right) J(y_{jm}) \\ &+ \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{r}^+, \mathbf{x}_k)\end{aligned}$$

The same process is applied for the gradient of electrostatic potential (1.20) at an interior point  $\mathbf{r}^-$

$$\begin{aligned}\nabla_{\mathbf{r}^-} \phi(\mathbf{r}^-) &= \sum_{j=1}^L \sum_{m=1}^M \frac{\partial \phi}{\partial \mathbf{n}_{jm}^{(y)}}(y_{jm}) W_m (\nabla_{\mathbf{r}^-} G_I(\mathbf{r}^-, y_{jm}) - \nabla_{\mathbf{r}^-} G_{II}(\mathbf{r}^-, y_{jm})) J(y_{jm}) \\ &+ \sum_{j=1}^L \sum_{m=1}^M \phi(y_{jm}) W_m \left( \frac{\epsilon_{II}}{\epsilon_I} \nabla_{\mathbf{r}^-} \frac{\partial G_{II}}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{r}^-, y_{jm}) - \nabla_{\mathbf{r}^-} \frac{\partial G_I}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{r}^-, y_{jm}) \right) J(y_{jm}) \\ &+ \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \nabla_{\mathbf{r}^-} G_I(\mathbf{r}^-, \mathbf{x}_k)\end{aligned}$$

and the gradient of electrostatic potential (1.21) at an exterior point  $\mathbf{r}^+$ .

$$\begin{aligned}
\frac{\epsilon_{II}}{\epsilon_I} \nabla_{\mathbf{r}^+} \phi(\mathbf{r}^+) &= \sum_{j=1}^L \sum_{m=1}^M \frac{\partial \phi}{\partial \mathbf{n}_{jm}^{(y)}}(y_{jm}) W_m \left( \nabla_{\mathbf{r}^+} G_I(\mathbf{r}^+, y_{jm}) - \nabla_{\mathbf{r}^+} G_{II}(\mathbf{r}^+, y_{jm}) \right) J(y_{jm}) \\
&+ \sum_{j=1}^L \sum_{m=1}^M \phi(y_{jm}) W_m \left( \frac{\epsilon_{II}}{\epsilon_I} \nabla_{\mathbf{r}^+} \frac{\partial G_{II}}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{r}^+, y_{jm}) - \nabla_{\mathbf{r}^+} \frac{\partial G_I}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{r}^+, y_{jm}) \right) J(y_{jm}) \\
&+ \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \nabla_{\mathbf{r}^+} G_I(\mathbf{r}^+, \mathbf{x}_k)
\end{aligned}$$

For computing  $G_{pol}$  and  $F_{pol}$ , the electrostatic potential  $\phi$  at the center  $\mathbf{x}_\alpha$  of an atom  $\alpha$  is

$$\begin{aligned}
\phi(\mathbf{x}_\alpha) &= \sum_{j=1}^L \sum_{m=1}^M \frac{\partial \phi}{\partial \mathbf{n}_{jm}^{(y)}}(y_{jm}) W_m \left( G_I(\mathbf{x}_\alpha, y_{jm}) - G_{II}(\mathbf{x}_\alpha, y_{jm}) \right) J(y_{jm}) \\
&+ \sum_{j=1}^L \sum_{m=1}^M \phi(y_{jm}) W_m \left( \frac{\epsilon_{II}}{\epsilon_I} \frac{\partial G_{II}}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{x}_\alpha, y_{jm}) - \frac{\partial G_I}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{x}_\alpha, y_{jm}) \right) J(y_{jm}) \\
&+ \sum_{k=1, k \neq \alpha}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{x}_\alpha, \mathbf{x}_k),
\end{aligned}$$

and the gradient of electrostatic potential  $\nabla \phi$  is

$$\begin{aligned}
\nabla_{\mathbf{x}_\alpha} \phi(\mathbf{x}_\alpha) &= \sum_{j=1}^L \sum_{m=1}^M \frac{\partial \phi}{\partial \mathbf{n}_{jm}^{(y)}}(y_{jm}) W_m \left( \nabla_{\mathbf{x}_\alpha} G_I(\mathbf{x}_\alpha, y_{jm}) - \nabla_{\mathbf{x}_\alpha} G_{II}(\mathbf{x}_\alpha, y_{jm}) \right) J(y_{jm}) \\
&+ \sum_{j=1}^L \sum_{m=1}^M \phi(y_{jm}) W_m \left( \frac{\epsilon_{II}}{\epsilon_I} \nabla_{\mathbf{x}_\alpha} \frac{\partial G_{II}}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{x}_\alpha, y_{jm}) - \nabla_{\mathbf{x}_\alpha} \frac{\partial G_I}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{x}_\alpha, y_{jm}) \right) J(y_{jm}) \\
&+ \sum_{k=1, k \neq \alpha}^{n_c} \frac{q_k}{\epsilon_I} \nabla_{\mathbf{x}_\alpha} G_I(\mathbf{x}_\alpha, \mathbf{x}_k)
\end{aligned}$$

#### 2.1.1.4 The implementation of solving enhanced Poisson-Boltzmann boundary integral equations

We apply Gaussian quadrature on the A-patches to numerically compute (1.23) and (1.24) which are also called finite-volume derivative boundary

integral equations (fdBIEs) is

$$\begin{aligned}
& \frac{1}{2}(1 + \frac{\epsilon_{II}}{\epsilon_I})\phi(\mathbf{x}_i) \\
& + \sum_{j=1}^L \sum_{m=1}^M \phi(y_{jm}) W_m \frac{\epsilon_{II}}{\epsilon_I} \frac{\partial G_{II}}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{x}_i, y_{jm}) - \left( \frac{\partial G_I}{\partial \mathbf{n}_{jm}^{(y)}}(\mathbf{x}_i, y_{jm}) \right) J(y_{jm}) \\
& - \sum_{j=1}^L \sum_{m=1}^M \frac{\partial \phi}{\partial \mathbf{n}_{jm}^{(y)}}(y_{jm}) W_m (G_{II}(\mathbf{x}_i, y_{jm}) - G_I(\mathbf{x}_i, y_{jm})) J(y_{jm}) \\
& = \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_I(\mathbf{x}_i, \mathbf{x}_k) + \frac{\epsilon_I}{\epsilon_I} H_{\partial\Omega}(\mathbf{x}_i) \\
& \frac{1}{2}(1 - \frac{\epsilon_I}{\epsilon_{II}}) \frac{\partial \phi}{\partial \mathbf{n}_i^x}(\mathbf{x}_i) \\
& - \sum_{j=1}^L \sum_{m=1}^M \phi(y_{jm}) W_m \left( \frac{\partial^2 G_{II}}{\partial \mathbf{n}_{jm}^{(y)} \partial \mathbf{n}_i^x}(\mathbf{x}_i, y_{jm}) - \frac{\partial^2 G_I}{\partial \mathbf{n}_{jm}^{(y)} \partial \mathbf{n}_i^x}(\mathbf{x}_i, y_{jm}) \right) J(y_{jm}) \\
& + \sum_{j=1}^L \sum_{m=1}^M \frac{\partial \phi}{\partial \mathbf{n}_{jm}^{(y)}}(y_{jm}) W_m \left( \frac{\epsilon_I}{\epsilon_{II}} \frac{\partial G_{II}}{\partial \mathbf{n}_i^{(x)}}(\mathbf{x}_i, y_{jm}) - \frac{\partial G_I}{\partial \mathbf{n}_i^{(x)}}(\mathbf{x}_i, y_{jm}) \right) J(y_{jm}) \\
& = \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \frac{G_I}{\mathbf{n}_i^x}(\mathbf{x}_i, \mathbf{x}_k) + \frac{\epsilon_I}{\epsilon_{II}} \frac{\partial H_{\partial\Omega}}{\partial \mathbf{n}_i^x}(\mathbf{x}_i).
\end{aligned}$$

where  $L$  is the number of patches,  $y_{jm} = \sum_{i=1}^3 b_{mi} v_{ji}(\lambda_{jm})$  represents the  $m^{th}$  integration point on the element  $\Gamma_j$ .

$$\begin{aligned}
G_I(\mathbf{x}_i, y_{jm}) &= \frac{1}{4\pi \|\mathbf{x}_i - y_{jm}\|} \\
G_{II}(\mathbf{x}_i, y_{jm}) &= \frac{e^{-\kappa \|\mathbf{x}_i - y_{jm}\|}}{4\pi \|\mathbf{x}_i - y_{jm}\|} \\
\frac{\partial G_I(\mathbf{x}_i, y_{jm})}{\partial \mathbf{n}_{jm}^{(y)}} &= \frac{-(\mathbf{x}_i - y_{jm}) \cdot \mathbf{n}_{jm}^{(y)}}{4\pi \|\mathbf{x}_i - y_{jm}\|^3} \\
\frac{\partial G_{II}(\mathbf{x}_i, y_{jm})}{\partial \mathbf{n}_{jm}^{(y)}} &= \frac{-e^{-\kappa \|\mathbf{x}_i - y_{jm}\|} (1 + \kappa \|\mathbf{x}_i - y_{jm}\|) (\mathbf{x}_i - y_{jm}) \cdot \mathbf{n}_{jm}^{(y)}}{4\pi \|\mathbf{x}_i - y_{jm}\|^3},
\end{aligned}$$

The boundary integral equations are treated as a linear system. For better elaboration, we can write fdBIEs in the following matrix form.

$$\begin{aligned}
\frac{1}{2}(1 + \frac{\epsilon_{II}}{\epsilon_I})\phi(\mathbf{r}) &= \int_{\Gamma} [G_0(\mathbf{r}, y) - G_{\kappa}(\mathbf{r}, y)] \frac{\partial \phi}{\partial \mathbf{n}}(y) dy \\
&+ \int_{\Gamma} \left[ \frac{\epsilon_{II}}{\epsilon_I} \frac{\partial G_{\kappa}}{\partial \mathbf{n}}(\mathbf{r}, y) - \frac{\partial G_0}{\partial \mathbf{n}}(\mathbf{r}, y) \right] \phi(y) dy \\
&+ \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} G_0(\mathbf{r}, \mathbf{x}_k) + \frac{\epsilon_{II}}{\epsilon_I} H_{\partial\Omega}(\mathbf{r}) \\
\frac{1}{2}(1 + \frac{\epsilon_I}{\epsilon_{II}}) \frac{\partial \phi}{\partial \mathbf{n}}(\mathbf{r}) &= \int_{\Gamma} \left[ \frac{\partial G_0}{\partial \mathbf{n}}(\mathbf{r}, y) - \frac{\epsilon_I}{\epsilon_{II}} \frac{\partial G_{\kappa}}{\partial \mathbf{n}}(\mathbf{r}, y) \right] \frac{\partial \phi}{\partial \mathbf{n}_0}(y) dy \\
&- \int_{\Gamma} \left[ \frac{\partial^2 G_{\kappa}}{\partial \mathbf{n}_0 \partial \mathbf{n}}(\mathbf{r}, y) - \frac{\partial^2 G_0}{\partial \mathbf{n}_0 \partial \mathbf{n}}(\mathbf{r}, y) \right] \phi(y) dy \\
&+ \sum_{k=1}^{n_c} \frac{q_k}{\epsilon_I} \frac{\partial G_0}{\partial \mathbf{n}}(\mathbf{r}, \mathbf{x}_k) + \frac{\epsilon_I}{\epsilon_{II}} \frac{\partial H_{\partial\Omega}}{\partial \mathbf{n}}(\mathbf{r})
\end{aligned} \tag{2.2}$$

where

$$\begin{aligned} (H_{\partial\Omega})_i &= H_{\partial\Omega}(\mathbf{x}_i) = \int_{\partial\Omega} \left[ G_I(\mathbf{x}_i, \mathbf{y}) \frac{\partial g}{\partial \mathbf{n}}(\mathbf{y}) - g(\mathbf{y}) \frac{\partial G_I}{\partial \mathbf{n}}(\mathbf{x}_i, \mathbf{y}) \right] d\mathbf{y} \\ &= |\partial\Omega| \sum_p \left[ G_I(\mathbf{x}_i, \mathbf{y}_p) \frac{\partial g}{\partial \mathbf{n}}(\mathbf{y}_p) - g(\mathbf{y}_p) \frac{\partial G_I}{\partial \mathbf{n}}(\mathbf{x}_i, \mathbf{y}_p) \right] \end{aligned}$$

with the grid points  $\{\mathbf{y}_p\}_{p=1}^{N_{grid}}$  on the outer boundary  $\partial\Omega$  and  $|\partial\Omega|$  as its area. All other notations are equivalent to the derivative boundary integral equations

As we implemented the solution of Juffer's boundary integral equations, we also applied GMRES with fast multipole method to solve the linear system (2.2).

## 2.2 The consistent representation of normal derivative of electrostatic potential

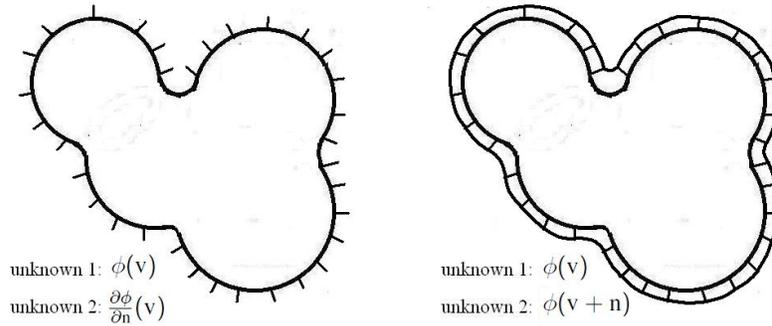


Figure 2.3: Transform the unknown functions  $\phi$  and  $\frac{\partial\phi}{\partial\mathbf{n}}$  to new unknown functions  $\phi$  on two surface layers.

In Juffer's boundary integral equations or the new boundary integral equations we derived, the normal derivatives of electrostatic potential are treated as an independent unknown function to the electrostatic potential as shown in Figure 2.3. The normal derivative of electrostatic potential computed

through solving this solution do not guarantee to be consistent with the corresponding electrostatic potential. We applied a consistent derivation for normal derivative of electrostatic potential  $\frac{\partial\phi}{\partial n}$  for maintaining the consistent relation between electrostatic potential and its normal derivative on the boundary element. This representation is available to be applied to any boundary integral equations we described in the previous sections.

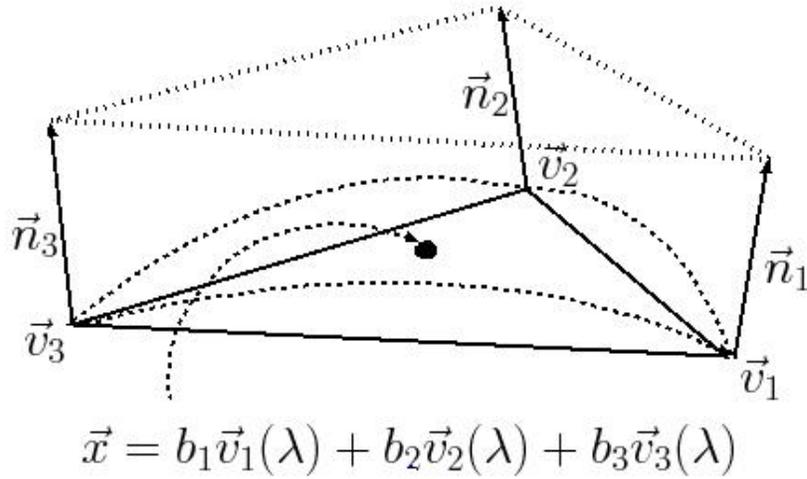


Figure 2.4: The representation of a point  $x$  on a algebraic patch  $v_1v_2v_3$ .

In Figure 2.4, we know that the parametric form of any position  $x$  on a A-patch  $\Gamma$  of a triangulation  $v_1v_2v_3$  is written as

$$x = b_1v_1(\lambda) + b_2v_2(\lambda) + (1 - b_1 - b_2)v_3(\lambda), x \in \Gamma \quad (2.3)$$

If we write the parameters as a vector  $b = (b_1, b_2, \lambda)$ , the normal derivative of the electrostatic potential can be written in the following form using the chain

rule

$$\begin{aligned}
\frac{\partial \phi}{\partial \mathbf{n}}(\mathbf{x}) &= \mathbf{n} \cdot \nabla \phi(\mathbf{x}) = \mathbf{n} \cdot \begin{bmatrix} \frac{\partial \phi}{\partial x_1}(\mathbf{x}) \\ \frac{\partial \phi}{\partial x_2}(\mathbf{x}) \\ \frac{\partial \phi}{\partial x_3}(\mathbf{x}) \end{bmatrix} \\
&= [n_1^{(x)}, n_2^{(x)}, n_3^{(x)}] \cdot \begin{bmatrix} \frac{\partial b_1}{\partial x_1} & \frac{\partial b_2}{\partial x_1} & \frac{\partial \lambda}{\partial x_1} \\ \frac{\partial b_1}{\partial x_2} & \frac{\partial b_2}{\partial x_2} & \frac{\partial \lambda}{\partial x_2} \\ \frac{\partial b_1}{\partial x_3} & \frac{\partial b_2}{\partial x_3} & \frac{\partial \lambda}{\partial x_3} \end{bmatrix} \begin{bmatrix} \frac{\partial \phi}{\partial b_1}(\mathbf{x}) \\ \frac{\partial \phi}{\partial b_2}(\mathbf{x}) \\ \frac{\partial \phi}{\partial \lambda}(\mathbf{x}) \end{bmatrix}
\end{aligned} \tag{2.4}$$

where  $\mathbf{x} = (x_1, x_2, x_3)^T$  and  $\mathbf{n}(\mathbf{x}) = (n_1^{(x)}, n_2^{(x)}, n_3^{(x)})^T$  is the unit vector of the normal at  $\mathbf{x}$ .

We can derive each term in the equation (2.4) in terms of the parametric parameters of the algebraic spline model. First,

$$\begin{aligned}
\begin{bmatrix} \frac{\partial b_1}{\partial x_1} \\ \frac{\partial b_1}{\partial x_2} \\ \frac{\partial b_1}{\partial x_3} \end{bmatrix} &= (\mathbf{v}_1 - \mathbf{v}_3) + \lambda(\mathbf{n}_1 - \mathbf{n}_3) \\
\begin{bmatrix} \frac{\partial b_2}{\partial x_1} \\ \frac{\partial b_2}{\partial x_2} \\ \frac{\partial b_2}{\partial x_3} \end{bmatrix} &= (\mathbf{v}_2 - \mathbf{v}_3) + \lambda(\mathbf{n}_2 - \mathbf{n}_3) \\
\begin{bmatrix} \frac{\partial \lambda}{\partial x_1} \\ \frac{\partial \lambda}{\partial x_2} \\ \frac{\partial \lambda}{\partial x_3} \end{bmatrix} &= b_1 \mathbf{n}_1 + b_2 \mathbf{n}_2 + b_3 \mathbf{n}_3
\end{aligned} \tag{2.5}$$

where  $\mathbf{v}_i = (v_{i1}, v_{i2}, v_{i3})^T$  and its unit normal vector  $\mathbf{n}_i = (n_{i1}, n_{i2}, n_{i3})^T$  for  $i = 1, 2, 3$ . Then, we approximate the electrostatic potential function in the region by

$$\begin{aligned}
\phi(\mathbf{x}) &= b_1((1 - \lambda)\phi(\mathbf{v}_1) + \lambda\phi(\mathbf{v}_1 + \mathbf{n}_1)) + b_2((1 - \lambda)\phi(\mathbf{v}_2) + \lambda\phi(\mathbf{v}_2 + \mathbf{n}_2)) \\
&\quad + (1 - b_1 - b_2)((1 - \lambda)\phi(\mathbf{v}_3) + \lambda\phi(\mathbf{v}_3 + \mathbf{n}_3)),
\end{aligned} \tag{2.6}$$

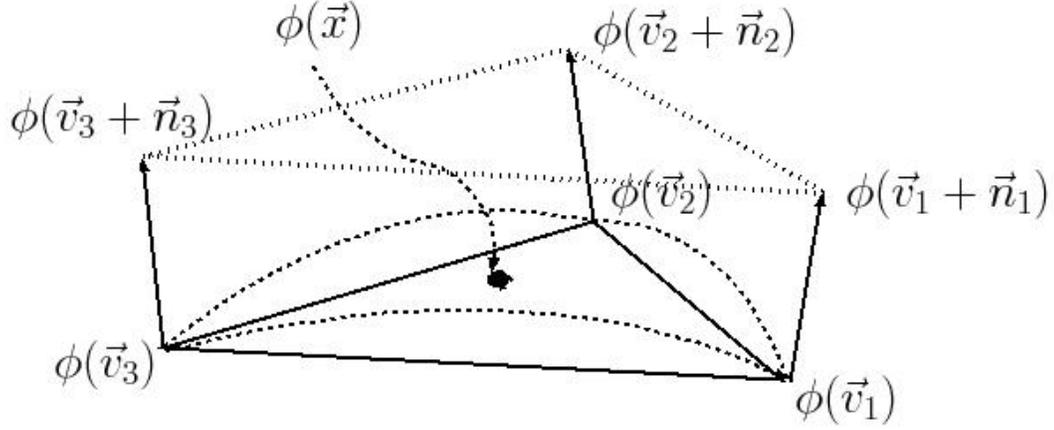


Figure 2.5: The representation of electrostatic potential  $\phi(\mathbf{x})$  at a point  $\mathbf{x}$  on a algebraic patch  $v_1v_2v_3$ .

and the derivative of the potential to the coordinate  $b$  becomes

$$\begin{aligned}
 \frac{\partial \phi}{\partial b_1}(\mathbf{x}) &= ((1 - \lambda)(\phi(v_1) - \phi(v_3)) + \lambda(\phi(v_1 + n_1) - \phi(v_3 + n_3))) \\
 \frac{\partial \phi}{\partial b_2}(\mathbf{x}) &= ((1 - \lambda)(\phi(v_2) - \phi(v_3)) + \lambda(\phi(v_2 + n_2) - \phi(v_3 + n_3))) \\
 \frac{\partial \phi}{\partial \lambda}(\mathbf{x}) &= b_1(\phi(v_1 + n_1) - \phi(v_1)) + b_2(\phi(v_2 + n_2) - \phi(v_2)) + b_3(\phi(v_3 + n_3) - \phi(v_3))
 \end{aligned}
 \tag{2.7}$$

Finally, we can get the normal derivative of electrostatic potential in the equation (2.4) in this A-patch by combining the above two equations (2.5) and (2.7).

The unknown electrostatic potential and its normal derivatives at the Gaussian quadrature points, in the matrix form of PB BIEs can be derived from the parametric representation of the electrostatic potential at the positions of the vertices and the vertices with a displacement of its unit normal. Here, the parametric form of a Gaussian point  $y_{jm}$  on the element  $\bar{\Gamma}_j$  is

$$y_{jm} = b_{m1}v_{j1}(\lambda_{jm}) + b_{m2}v_{j2}(\lambda_{jm}) + (1 - b_{m1} - b_{m2})v_{j3}(\lambda_{jm})$$

and the electrostatic potential  $\phi(y_{jm})$  and normal derivative of electrostatic potential  $\frac{\partial\phi}{\partial n_{jm}^{(y)}}(y_{jm})$  at this point are computed using the equation (2.4) with the equations (2.5) and (2.7).

## 2.3 Framework and computational steps

In this paper, we developed a platform of data structures and routines of a 3D boundary element solver, called PB-CFMM (Poisson-Boltzmann - curved fast multipole method). We implemented all the above methodologies for solving the PB electrostatic problem in PB-CFMM and it is callable from TexMol [5]. As we described in the above section, the input of the solver is the 3D atomic structure and the triangular mesh of the target molecule. Its properties including electrostatic potential, electrostatic free energy and forces are computed.

Here, PETSc (Portable, Extensible Toolkit for Scientific Computation) is used for the solution of the PB linear system [17]. It supports matrix-free Krylov iterative method (e.g. GMRES, CG) which do not require explicit storage of the matrix. The explicit matrix is replaced by a user-defined evaluation of matrix vector production. Here, we use kernel independent fast multipole method, KiFMM, to do linear-time evaluation [78].

The computational steps for the solution of the PB electrostatic problem are concluded in the following list.

**Structure preparation** Prepare structures for continuum electrostatic calculations using “PDB2PQR”. The main task of “PDB2PQR” assigning charge and radius parameters to the atomic PDB structure [36]. Since many biomolecular structures in the Protein Data Bank do not contain hydrogen atoms and a fraction of heavy atoms, this software also checks

and rebuilds those missing hydrogen and heavy atoms to biomolecular structures based on standard amino acid topologies.

**Molecular surface extraction** Extract the molecular surface from the level set computed through geometric flow evolution [9].

**Triangular mesh generation** Compute high-qualified linear triangular boundary elements using octree-based dual contouring method [80].

$C^1$  **A-spline modeling** Compute the cubic algebraic spline over the triangular elements.

**Numerical solution** Compute electrostatic potential by solving the PB equation using our boundary element solver "PB-CFMM" with the fast summation method using "KiFMM" [78].

- construct KiFMM models for PB kernels on the algebraic spline model,
- solve the linear system using GMRES iterative method with KiFMM.

**Post-processing** compute electrostatic free energy and forces using electrostatic potential.

In this paper, we solve the linear Poisson-Boltzmann system using the iterative method, GMRES with the initialization of electrostatic potential using the coulombic equation. The relative residual tolerance is  $10^{-7}$  and number of Gaussian quadrature points per triangle is 7. The detailed computational steps for solving boundary integral equations (2.1) or (2.2) are as follows.

- Input: atomic charges  $\{q_k\}_{k=1}^{n_c}$  and centers  $\{x_k\}_{k=1}^{n_c}$  of a molecule  $M$  and the triangulation  $\{\Gamma_j\}_{j=1}^L$  of molecular surface  $\Gamma$
- Output: PB electrostatic potential on vertices of the molecular surface  $\Gamma$
- Parametrize the molecular surface
  - Generate triangular A-patches  $\{\bar{\Gamma}_j\}_{j=1}^L$  on triangles  $\{\Gamma_j\}_{j=1}^L$ .
  - Compute the Gaussian quadrature points  $\{y_{jm}\}_{m=1}^M$  for all triangular A-patches  $\{\bar{\Gamma}_j\}_{j=1}^L$ .
- Initialize the electrostatic potential  $\{\phi^0(x_i)|i = 1, \dots, V\}$  on the vertices  $\{x_i\}_{i=1}^V$  of the molecular surface using Coulomb potential  $\phi^0(x_i) = \sum_{k=1}^{n_c} \frac{q_k}{4\pi\epsilon_i \|x_i - x_k\|}$
- Initialize data structure of fast multipole method for each kernel in boundary integral equations (2.1) or (2.2) based on
  - all kernel functions  $G_I, G_{II}$  and so on,
  - source points: Gaussian quadrature points  $\{y_{jm}\}_{m=1}^M$  on the molecular surface  $\Gamma$
  - target points: vertices of the triangular molecular surface:  $\{x_i\}_{i=1}^V$
- compute electrostatic potential  $\phi$  by iteratively solving BIEs (2.1) or (2.2) using GMRES. At  $n^{th}$  iteration of GMRES

- compute  $\{\phi^n(y_{jm})|j = 1, \dots, L; m = 1, \dots, M\}$  on the Gaussian quadrature points of the molecular surface  $\Gamma$  by interpolation of  $\{\phi^n(x_i)|i = 1, \dots, V\}$  using algebraic spline model
- compute weighted electrostatic potential  $\{\hat{\phi}^n(y_{jm}) = w_m \phi^n(y_{jm}) J(\bar{\Gamma}_j)|j = 1, \dots, L; m = 1, \dots, M\}$  where Gaussian weights  $\{W_m\}_{m=1}^M$  and Jacobian of the A-patch  $\{J(\bar{\Gamma}_j)|j = 1, \dots, L\}$ .
- evaluate the left hand side of the equation (2.1) or (2.2)
  - \* apply the fast multipole method of some kernel  $G$  to compute the Gaussian quadrature (summation)  $\sum_{j=1}^L \sum_{m=1}^M G(x_i, y_{jm}) \hat{\phi}^n(y_{jm})$  for all target points  $\{x_i\}_{i=1}^V$
- check if the ratio of residual and the norm of the right hand side of (2.1) or (2.2) is smaller than a tolerance  $\tau$ . If yes, break from the GMRES loop.
- update the electrostatic potential  $\phi^n$  to  $\phi^{n+1}$  using Arnoldi iteration in GMRES.

Figure (2.6) shows the computational steps of the evaluation of matrix-vector product at each iteration of GMRES linear solver.

### 2.3.1 Complexity Analysis

The number of the boundary elements  $L$  of molecular surface  $\Gamma$ , the number of vertices  $V$  of the boundary elements and atom/bead size  $n_c$  of the molecular model determine the time and space complexity of the PB linear

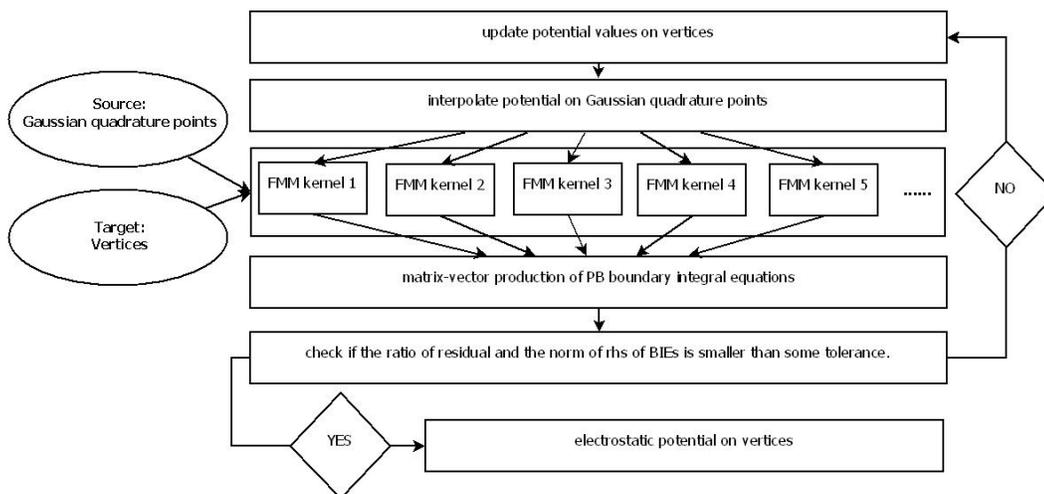


Figure 2.6: The computational steps of the evaluation of matrix-vector product at each iteration of GMRES linear solver.

$L$	# boundary elements of molecular surface $\Gamma$
$V$	# vertices of molecular surface $\Gamma$
$M$	# Gaussian quadrature points per boundary element (constant)
$n_c$	# atoms of molecule
$K$	# of kernels in boundary integral equations
$N_{iter}$	# of iterations for GMRES convergence

Table 2.1: The notations for complexity analysis of electrostatic computation.

system. Before GMRES computation, we compute the right hand side of the linear system (2.1). It contains two summations of two different kernels with the same source and target points. The sources points are the atoms of the molecular model and the target points are the vertices of the triangulation. With the constant number of coefficient for KiFMM, we can compute the right hand side of the linear system with the time complexity  $O(V + n_c)$ . The results are then stored in a vector with size  $2V$ .

At each iteration of GMRES, we use KiFMM to evaluate the matrix-vector product in the linear system (2.1). The source points are the Gaussian quadrature points on the surfaces which is the product of the number of the boundary elements times the number of quadrature points per boundary element. The target points are the vertices of the boundary elements. With the constant number of coefficient for KiFMM, this evaluation is done with the time complexity  $O(ML + V)$  where  $M$  is a constant number of quadrature points per boundary element. In the following experiments, we take 7 quadrature points per boundary element,  $M = 7$ . In the linear system (2.1), different kernel functions are considered. All of them are derived from  $G_I$  and  $G_{II}$  and the number of kernel functions are  $K = 16$  because the normal derivative of kernel functions along the target normals are required to be computed as the inner product of three distinct kernels. For example,

$$\frac{\partial G}{\partial \mathbf{n}^{(x)}}(\mathbf{x}, \mathbf{y}) = \mathbf{n}^{(x)} \cdot \nabla G(\mathbf{x}, \mathbf{y}) = \mathbf{n}^{(x)} \cdot \left( \frac{\partial G}{\partial x}(\mathbf{x}, \mathbf{y}), \frac{\partial G}{\partial y}(\mathbf{x}, \mathbf{y}), \frac{\partial G}{\partial z}(\mathbf{x}, \mathbf{y}) \right).$$

If we evaluate each kernel sequentially, with  $K$  number of kernels, the time complexity for each iteration of GMRES is  $O(K(L + V))$ . We improved this

by using the multicore speed-up to evaluate the matrix-vector production of each kernel in parallel, so the time complexity is  $O(L + V)$

Therefore, if we assume the number of iterations of GMRES is  $N_{iter}$ , the time complexity of the overall evaluation of the linear system is  $O(N_{iter}(L + V) + (V + n_c))$ . Actually, the number of iterations of GMRES is determined by the condition numbers of linear system. Enhanced boundary integral equation is a modified version of Juffer's boundary integral equations. If we set the domain boundary condition to be zero boundary condition such that  $g = 0$ , enhanced boundary integral equations is equivalent to Juffer's boundary integral equations. J. Liang and S. Subramaniam have proved that Juffer's boundary integral equations are well-conditioned [55]. Therefore, enhanced boundary integral equations which do not change the left hand side of the boundary integral equations are also well-conditioned.

The convergence of GMRES is determined by the condition number of the matrix operators. In principle, the Poisson equation leads to a second kind Fredholm type integral equation, characterized by a well-conditioned compact integral operator [18]. Poisson-Boltzmann equation is still a Poisson equation where the charge density function is contributed by atom charges and mobile ion charges. In our case, the derivative boundary integral equations is also a second kind Fredholm type integral equation. In the following experiments, we can observe that the number of iterations to convergence is consistently low even when the number of boundary elements increases. In the following cases, when we set the tolerance of residual to be  $10^{-6}$ , the maximum number

of iteration to converge is less than 100 which also indicates that the derivative boundary integral equation is well-conditioned.

We gathered 71 sets of ligand-receptor protein complexes (ligand,receptor,ligand-receptor complex) from RCSB protein data bank (PDB). These are used for the evaluation of the PB electrostatic computation.

## 2.4 Experimental results

The first experiment is an analytical numerical error evaluation with a given potential function. This experiment is applied for understanding the reliability and efficiency of our PB solution. In the second experiment, we compute and compare real electrostatic results of these proteins between our boundary element solvers and DelPhi II finite difference solver [60, 67]. Then, we study the performance of our system by controlling different effective factors. All experiments are done on a linux machine with Dual Core AMD Opteron processor 280 with 4 GB memory. We discussed and analyze the experimental results in the following experiments.

### 2.4.1 Analytical numerical evaluation

In the first experiment, we evaluate the efficiency and accuracy of numerical computation of electrostatic potentials and their normal derivatives using regular or consistent PB boundary element solvers with fast matrix-vector product evaluation. The numerical test is done with the assumption that electrostatic potential is given as an exponential function  $\tilde{\phi}(\mathbf{x}) = e^{-\|\mathbf{x}\|^2}$

and the normal derivative of potential as the normal derivative of this exponential function  $\frac{\partial \tilde{\phi}(\mathbf{x})}{\partial \mathbf{n}(\mathbf{x})} = -2e^{-\|\mathbf{x}\|^2}(\mathbf{x} \cdot \mathbf{n}(\mathbf{x}))$ . We calculate  $Q(\mathbf{x})$  and  $R(\mathbf{x})$  on the vertices of the triangular meshes by evaluating the left hand side of dBIEs (1.11) and (1.14).

$$\begin{aligned} Q(\mathbf{x}) &= \frac{1}{2} \left(1 + \frac{\epsilon_E}{\epsilon_I}\right) \tilde{\phi}(\mathbf{x}) + \int_{\Gamma} \left( \frac{\partial G_0(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} - \frac{\epsilon_E}{\epsilon_I} \frac{\partial G_{\kappa}(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} \right) \tilde{\phi}(\mathbf{y}) d\mathbf{y} \\ &\quad - \int_{\Gamma} (G_0(\mathbf{x}, \mathbf{y}) - G_{\kappa}(\mathbf{x}, \mathbf{y})) \frac{\partial \tilde{\phi}(\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} d\mathbf{y} \\ R(\mathbf{x}) &= \frac{1}{2} \left(1 + \frac{\epsilon_I}{\epsilon_E}\right) \frac{\partial \phi(\mathbf{x})}{\partial \mathbf{n}(\mathbf{x})} + \int_{\Gamma} \left( \frac{\partial^2 G_0(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}(\mathbf{x}) \partial \mathbf{n}(\mathbf{y})} - \frac{\partial^2 G_{\kappa}(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}(\mathbf{x}) \partial \mathbf{n}(\mathbf{y})} \right) \phi(\mathbf{y}) d\mathbf{y} \\ &\quad - \int_{\Gamma} \left( \frac{\partial G_0(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}(\mathbf{x})} - \frac{\epsilon_I}{\epsilon_E} \frac{\partial G_{\kappa}(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}(\mathbf{x})} \right) \frac{\partial \phi(\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} d\mathbf{y} \end{aligned}$$

We evaluate the electrostatic potential and its normal derivative on the vertices of triangulation computed using our boundary element solver by the relative errors

$$\frac{\sqrt{\sum_{i=1}^P |\phi(\mathbf{v}_i) - \tilde{\phi}(\mathbf{v}_i)|^2}}{\sqrt{\sum_{i=1}^P |\phi(\mathbf{v}_i)|^2}}$$

and

$$\frac{\sqrt{\sum_{i=1}^P \left| \frac{\partial \phi}{\partial \mathbf{n}_i}(\mathbf{v}_i) - \frac{\partial \tilde{\phi}}{\partial \mathbf{n}_i}(\mathbf{v}_i) \right|^2}}{\sqrt{\sum_{i=1}^P \left| \frac{\partial \phi}{\partial \mathbf{n}_i}(\mathbf{v}_i) \right|^2}}.$$

Table 2.2 shows the average relative error of potential and compute time of the evaluations of whole proteins.

In this experiment, we observe that our fast boundary element solver is much more efficient than the direct solver because fast multipole methods are linear-time algorithms with high accuracy. With triangular meshes in different resolutions, small relative errors of KiFMM indicate that our fast multipole method works well in solving the PB linear system.

On the other hand, the normal derivative of potential on the molecular surface is taken as unknown in the original derivative boundary integral

# of A-patches	evaluation method	relative error ( $\phi$ )	# of iterations	compute time (seconds)
2000	direct	$6.380 \times 10^{-7}$	35.21	165.063
	KiFMM	$6.379 \times 10^{-7}$	35.21	60.858
5000	direct	$9.472 \times 10^{-7}$	40.88	1237.451
	KiFMM	$1.309 \times 10^{-6}$	41.72	216.232
10000	direct	$2.424 \times 10^{-7}$	46.71	5423.711
	KiFMM	$2.635 \times 10^{-7}$	46.83	528.605
63444.81*	KiFMM	$4.678 \times 10^{-7}$	38.41	3012.344

Table 2.2: The results of analytical experiments computed using PB BEM solver with different number of A-patches for 213 molecules; column 1 is the number of triangles (\* is the average number of A-patches of the original triangular mesh of 213 molecular surfaces); column 2 is the type of evaluation method of matrix-vector product; column 3 is the average relative errors of potential  $\phi$  and  $\tilde{\phi}$ ; column 4 is the number of iterations for the convergence; column 5 is the computation time in seconds.

equations. In our paper, we used the parametric formulation of the algebraic spline model to derive the normal derivative of potential. Here, we compute the potential and normal derivative of potential using regular or consistent numerical methods and compare the relative errors and computation time in Table 2.3.

The relative errors of potential are similar in both numerical solutions but those of the normal derivative of potential are not. The normal derivative of potential computed using the parametric formulation is more accurate than that computed using the regular solution. This indicates that the relation between potential and normal derivative of potential is accurate and consistent when we used our parametric formulation (2.4).

# of A-patches	numerical method	relative error ( $\phi$ )	relative error ( $\frac{\partial\phi}{\partial n}$ )	# of iterations	compute time (s)
2000	regular	$6.379 \times 10^{-7}$	$1.442 \times 10^{-3}$	35.21	60.858
	consistent	$5.208 \times 10^{-7}$	$1.533 \times 10^{-7}$	36.17	62.388
5000	regular	$1.309 \times 10^{-6}$	$9.454 \times 10^{-4}$	41.72	216.232
	consistent	$9.081 \times 10^{-7}$	$5.900 \times 10^{-7}$	45.83	258.544
10000	regular	$2.635 \times 10^{-7}$	$2.850 \times 10^{-3}$	46.83	528.605
	consistent	$2.769 \times 10^{-7}$	$3.669 \times 10^{-7}$	45.82	492.002
63444.81*	regular	$4.678 \times 10^{-7}$	$1.498 \times 10^{-3}$	38.41	3012.344
	consistent	$4.921 \times 10^{-7}$	$4.944 \times 10^{-7}$	39.09	3107.15

Table 2.3: The results of analytical experiments computed using PB BEM solver with different number of A-patches for 213 proteins; column 1 is the number of triangles (\* is the average number of A-patches of the original triangular mesh of 213 molecular surfaces); column 2 is the type of numerical solution of boundary element method; column 3 is the average relative errors of potential  $\phi$  and  $\tilde{\phi}$ ; column 4 is the average relative errors of normal derivative of potential  $\frac{\partial\phi}{\partial n}$  and  $\frac{\partial\tilde{\phi}}{\partial n}$ ; column 5 is the number of iterations for the convergence; column 6 is the computation time in seconds.

## 2.4.2 Poisson-Boltzmann electrostatic potential

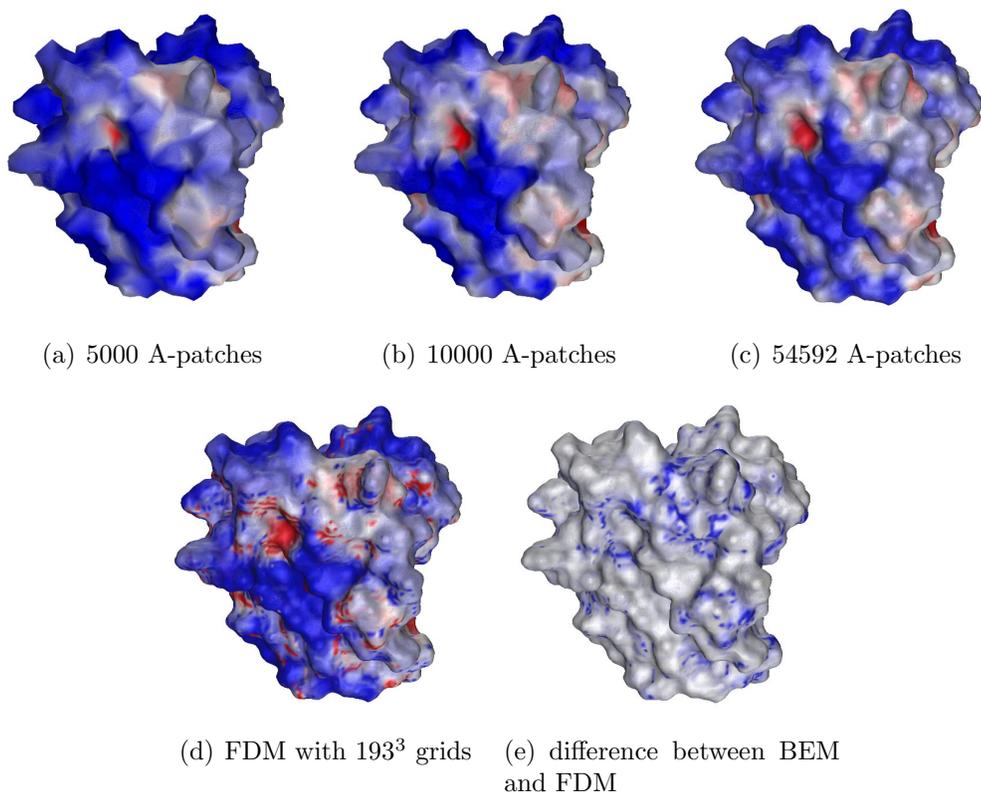


Figure 2.7: The PB electrostatic potential on molecular surface of nuclear transport factor 2 (PDB id: 1A2K) with different resolutions. (e) shows the difference of PB electrostatic potential between BEM and FDM. The color is going from red (potential of  $-3.8 k_b T / e_c$ ) to blue (potential of  $+3.8 k_b T / e_c$ ).

We also compute the real PB electrostatic potential for all the proteins. In the following experiments, we consider to use the same domain boundary condition, full Coulombic boundary condition, in both boundary element solver and The state of the art finite difference solver, Delphi II. In Figure 2.7, we show the electrostatic potential on the molecular surface of an example in the

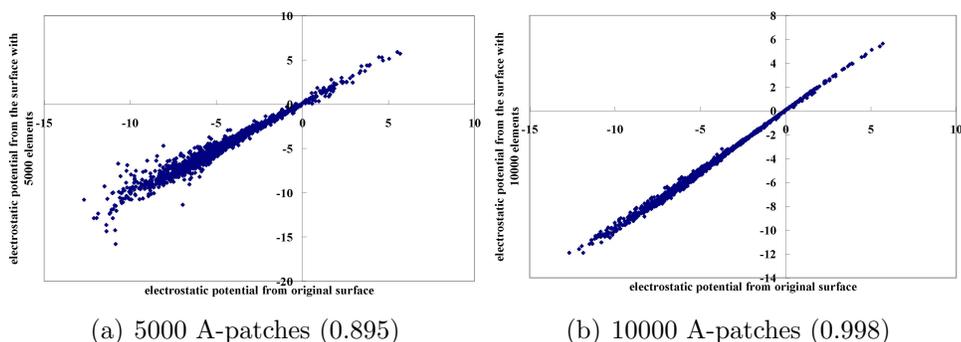


Figure 2.8: The comparison of electrostatic potential between the molecular surface of Bovine Chymotrypsinogen\*A (PDB id: 1CGI) with different resolutions (the correlation in parentheses) where the average number of A-patches of the original surface is 63444.81.

protein list. PB electrostatic potential is computed with different numbers of A-patches. The color of the surface represents the electrostatic potential on the molecular surface, going from red (potential of  $-3.8 k_bT/e_c$ ) to blue (potential of  $+3.8 k_bT/e_c$ ) and white is neutral potential. The distribution of electrostatic potential computed using the triangular A-spline models with different resolution are almost the same. The same results can be observed in Figure 2.8 which represents the different of electrostatic potential of a protein (PDB id: 1CGI) computed using A-spline models with different resolutions. The number of A-patches of its original surface is 54592. The correlation of the results computed from the original surface and decimated surface with 10000 A-patches is up to 0.998. It indicates that we can get a similar result using only 1/5 of A-patches. However, if we just use 5000 A-patches, they are not enough to represent the details of the molecular surface and the correlation

numerical method	solver name	# of grids/ # of A-patches	inverse discretization length scale (1.0/Å)	correlation ( $\phi$ )
FDM	Delphi II	65 <sup>3</sup>	0.333	0.965
FDM	Delphi II	129 <sup>3</sup>	0.667	0.977
FDM	Delphi II	193 <sup>3</sup>	1.000	-
BEM	PB-CFMM	5000	0.367	0.944
BEM	PB-CFMM	10000	0.732	0.968
BEM	PB-CFMM	63444.81*	5.301	0.981

Table 2.4: Average experimental results of PB electrostatic potential computation for 213 proteins (71 sets of ligand-receptor complexes); column 1 is the numerical method; column 2 is the name of the solver; column 3 is the number of grids for FDM and number of A-patches for BEM; column 4 is the inverse discretization length scale of each grid or A-patches; column 5 is the correlation of electrostatic potential to FDM with 193<sup>3</sup> grids.

becomes 0.895.

Figures 2.7 (c) and (d) show the surface electrostatic potential computed using our BEM solver and finite different solver, Delphi II. The distributions of their electrostatic potential are roughly the same. We then compute the difference between them, shown in Figure 2.7 (e). Blue color represents the magnitude of the difference of surface electrostatic potential. We can observe that the large difference occurs only in some small regions. In Table 2.4, we compute electrostatic potential at the points of 65<sup>3</sup> grids using BEM or FDM with different resolutions and compare the results by their correlation to the electrostatic potential computed by FDM with 193<sup>3</sup> grids. The inverse discretization length scale in the table is the average edge length of triangulation for BEM and distance between grid points for FDM. We can observe that electrostatic potential computed using BEM and FDM is highly correlated.

The electrostatic potential on the surface computed using enhanced boundary integral equations is very similar to that computed through Juffer's boundary integral equations ( the domain boundary condition  $g = 0$ ). The average correlation between them is 0.957. It indicates that the enhanced boundary integral equations follow Poisson-Boltzmann equation with interfacial boundary conditions. The only difference is that the formulation that different domain boundary conditions are considered.

The electrostatic potential computed on the boundary is highly correlated to the domain boundary formulation. The average correlation is up to 0.991. However, The correlation of electrostatic potential computed using the domain boundary conditions and Juffer's boundary integral equations is only 0.632.

In Table 2.5, we show the comparison of electrostatic potential and gradients of electrostatic potential computed using Juffer's boundary integral equation with different domain boundary conditions. Juffer BIE type indicates that we set the domain boundary condition to be zero  $g = 0$  and enhanced indicates that we used full Coulombic boundary condition. The correlation is computed according to the results computed using finite difference method (Delphi II) with full Coulombic boundary condition. We can observe that enhanced boundary integral equations with the same domain boundary condition perform better. On the other hand, if we set the domain boundary condition of finite difference method to be zero boundary condition  $g = 0$ , Juffer's solution gives a better performance as shown in Table 2.6.

# of A-patches	BIE type	correlation ( $\phi$ )	correlation ( $-\nabla\phi$ )	compute time (s)
5000	Juffer	0.814	0.707/0.735/0.775	120.9
	Enhanced	0.858	0.887/0.804/0.842	162.4
10000	Juffer	0.852	0.825/0.810/0.802	522.1
	Enhanced	0.954	0.933/0.907/0.947	598.9
20000	Juffer	0.887	0.840/0.835/0.857	934.1
	Enhanced	0.969	0.933/0.914/0.958	907.3

Table 2.5: The comparison of electrostatic potential and gradients of electrostatic potential computed using Juffer’s boundary integral equation and enhanced boundary integral equations. The correlation is computed according to the results computed using finite difference method (Delphi II) with full Coulombic boundary condition.

# of A-patches	BIE type	correlation ( $\phi$ )	correlation ( $-\nabla\phi$ )	compute time (s)
5000	Juffer	0.913	0.854/0.833/0.819	120.9
	Enhanced	0.841	0.792/0.801/0.743	162.4
10000	Juffer	0.940	0.877/0.843/0.832	522.1
	Enhanced	0.897	0.890/0.839/0.834	598.9
20000	Juffer	0.948	0.920/0.936/0.936	934.1
	Enhanced	0.904	0.864/0.891/0.877	907.3

Table 2.6: The comparison of electrostatic potential and gradients of electrostatic potential computed using Juffer’s boundary integral equation and enhanced boundary integral equations. The correlation is computed according to the results computed using finite difference method (Delphi II) with zero boundary condition.

## Chapter 3

# Fast Computation of Electrostatic Energy and Forces

### 3.1 Dielectric and ionic boundary model

#### 3.1.1 Im's model

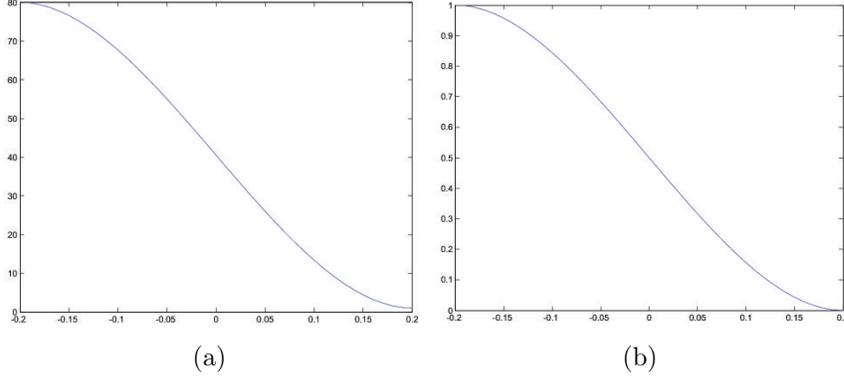


Figure 3.1: Dielectric function  $\epsilon$  and ionic boundary function  $\lambda$  defined using Im's volume exclusion function with the window size  $w = 0.2$

Im et al. defined  $\epsilon(\mathbf{x})$  and  $\lambda(\mathbf{x})$  by a volume exclusion function [47].

$$\epsilon(\mathbf{x}) = \epsilon_I + (\epsilon_{II} - \epsilon_I)H(\mathbf{x}; \{\mathbf{x}_k\}),$$

and

$$\lambda(\mathbf{x}) = H(\mathbf{x}; \{\mathbf{x}_k\}),$$

where the volume exclusion function is defined by the atomic centers  $\{\mathbf{x}_k\}_{k=1}^{n_c}$ ,

$$H(\mathbf{x}; \{\mathbf{x}_k\}) = \prod_{k=1}^{n_c} H_k(\|\mathbf{x} - \mathbf{x}_k\|),$$

and

$$H_k(r) = \begin{cases} 0, & r \leq r_k - w, \\ -\frac{(r-r_k+w)^3}{4w^3} + \frac{3(r-r_k+w)^2}{4w^2}, & r_k - w < r < r_k + w, \\ 1, & r \geq r_k + w. \end{cases}$$

where  $w$  is the windows size.

Figure 3.1 shows the dielectric function  $\epsilon$  and ionic boundary function  $\lambda$  defined using Im's model with the window size  $w = 0.2$ . We can observe that both functions are smooth and we can compute their derivative by solving the derivative of the volume exclusion function.

$$\frac{\partial H}{\partial \mathbf{x}_\alpha}(\mathbf{x}; \{\mathbf{x}_k\}) = \frac{\partial H_\alpha}{\partial \mathbf{x}_\alpha}(\|\mathbf{x} - \mathbf{x}_\alpha\|) \frac{H(\mathbf{x}; \{\mathbf{x}_k\})}{H_\alpha(\|\mathbf{x} - \mathbf{x}_\alpha\|)}$$

where

$$\frac{\partial H_\alpha(r)}{\partial \mathbf{x}_\alpha} = \begin{cases} 0, & r \leq r_\alpha - w, \\ \left(-\frac{3}{4w^3}(r - r_\alpha + w)^2 + \frac{3}{2w^2}(r - r_\alpha + w)\right) \frac{\mathbf{x}_\alpha - \mathbf{x}}{r}, & r_\alpha - w < r < r_\alpha + w, \\ 0, & r \geq r_\alpha + w. \end{cases}$$

and  $r = \|\mathbf{x} - \mathbf{x}_\alpha\|$ .

Therefore, the atomic derivative of  $\epsilon$  and  $\lambda$  will become

$$\frac{\partial}{\partial \mathbf{x}_\alpha} \epsilon(\mathbf{x}) = (\epsilon_{II} - \epsilon_I) \frac{\partial H_\alpha}{\partial \mathbf{x}_\alpha}(\mathbf{x}; \{\mathbf{x}_k\}),$$

and

$$\frac{\partial}{\partial \mathbf{x}_\alpha} \lambda(\mathbf{x}) = \frac{\partial H_\alpha}{\partial \mathbf{x}_\alpha}(\mathbf{x}; \{\mathbf{x}_k\}).$$

### 3.1.2 Bajaj and Zhao's model

Bajaj and Zhao defined  $\epsilon(\mathbf{x})$  and  $\lambda(\mathbf{x})$  using the inclusion-exclusion principle [12].

$$\epsilon(\mathbf{x}) = \epsilon_{II} + (\epsilon_I - \epsilon_{II}) \bar{V}(\mathbf{x}; \{\mathbf{x}_k\}),$$

and

$$\lambda(\mathbf{x}) = \bar{V}(\mathbf{x}; \{\mathbf{x}_k\}),$$

They define the complementary function  $V = 1 - \bar{V}$  and neglect the cases that more than four atoms overlap simultaneously.

$$V(\mathbf{x}) = \sum_i V_i - \sum_{i<j} V_i V_j + \sum_{i<j<k} V_i V_j V_k - \sum_{i<j<k<l} V_i V_j V_k V_l$$

where

$$V_i(\mathbf{x}) = \begin{cases} 1, & \|\mathbf{x} - \mathbf{x}_i\| \leq r_i, \\ \frac{2}{w^3}(\|\mathbf{x} - \mathbf{x}_i\| - r_i)^3 - \frac{3}{w^2}(\|\mathbf{x} - \mathbf{x}_i\| - r_i)^2 + 1, & r_i < \|\mathbf{x} - \mathbf{x}_i\| < r_i + w, \\ 0, & \|\mathbf{x} - \mathbf{x}_i\| \geq r_i + w. \end{cases}$$

By setting  $w = 1.4 \text{ \AA}$ , atoms  $i$  and  $j$  are disconnected iff  $V_i(\mathbf{x})V_j(\mathbf{x}) = 0$  for any  $\mathbf{x} \in \Omega$ . Within the Van der Waal surface of the molecule,  $\bar{V}$  is always 0; beyond the solvent accessible surface,  $\bar{V}$  is always 1 and in between  $0 < \bar{V} < 1$ .

The atomic derivative of  $\bar{V}(\mathbf{x}; \{\mathbf{x}_k\})$  can then be written as

$$\frac{\partial}{\partial \mathbf{x}_\alpha} \bar{V}(\mathbf{x}; \{\mathbf{x}_k\}) = -\frac{\partial V_\alpha}{\partial \mathbf{x}_\alpha} \left( 1 - \sum_j V_j + \sum_{j<k} V_j V_k - \sum_{j<k<l} V_j V_k V_l \right) = -\frac{\partial V_\alpha}{\partial \mathbf{x}_\alpha} g_\alpha$$

where  $j, k, l$  are the atoms overlapping with atom  $\alpha$ ,  $g_\alpha = 1 - \sum_j V_j + \sum_{j<k} V_j V_k - \sum_{j<k<l} V_j V_k V_l$  and

$$\frac{\partial V_i}{\partial \mathbf{x}_\alpha}(\mathbf{x}) = \begin{cases} 0, & \|\mathbf{x} - \mathbf{x}_i\| \leq r_\alpha, \\ \left( \frac{6}{w^3}(\|\mathbf{x} - \mathbf{x}_i\| - r_\alpha)^2 - \frac{6}{w^2}(\|\mathbf{x} - \mathbf{x}_i\| - r_i) \right) \frac{\mathbf{x}_\alpha - \mathbf{x}}{\|\mathbf{x} - \mathbf{x}_i\|}, & r_i < \|\mathbf{x} - \mathbf{x}_i\| < r_i + w, \\ 0, & \|\mathbf{x} - \mathbf{x}_i\| \geq r_i + w. \end{cases}$$

Therefore, the atomic derivative of  $\epsilon$  and  $\lambda$  will become

$$\frac{\partial}{\partial \mathbf{x}_\alpha} \epsilon(\mathbf{x}) = (\epsilon_I - \epsilon_{II}) \frac{\partial}{\partial \mathbf{x}_\alpha} \bar{V}(\mathbf{x}; \{\mathbf{x}_k\}),$$

and

$$\frac{\partial}{\partial \mathbf{x}_\alpha} \lambda(\mathbf{x}) = \frac{\partial}{\partial \mathbf{x}_\alpha} \bar{V}(\mathbf{x}; \{\mathbf{x}_k\}).$$

## 3.2 Implementation

### 3.2.1 The implementation of electrostatic energy

The fixed charge energy in the equation (2.1.1) can be computed directly using the interior electrostatic potential (1.16) computed using boundary element method.

$$\begin{aligned}
G^{QF} &= \frac{1}{2} \sum_{k=1}^{n_c} \phi(\mathbf{x}_k) q_k \\
&= \frac{1}{2} \sum_{k=1}^{n_c} q_k \left( \int_{\Gamma} [G_I(\mathbf{x}_k, \mathbf{y}) - G_{II}(\mathbf{x}_k, \mathbf{y})] \frac{\partial \phi}{\partial \mathbf{n}}(\mathbf{y}) d\mathbf{y} + \int_{\Gamma} \left[ \epsilon \frac{\partial G_{II}}{\partial \mathbf{n}}(\mathbf{x}_k, \mathbf{y}) - \frac{\partial G_I}{\partial \mathbf{n}}(\mathbf{x}_k, \mathbf{y}) \right] \phi_I(\mathbf{y}) d\mathbf{y} \right. \\
&\quad \left. + \sum_{l=1, l \neq k}^{n_c} \frac{q_l}{\epsilon_I} G_I(\mathbf{x}_k, \mathbf{x}_l) \right) \\
&= \frac{1}{2} \sum_{k=1}^{n_c} q_k \left( \sum_{j=1}^L \sum_{m=1}^M W_m [G_I(\mathbf{x}_k, \mathbf{y}_{jm}) - G_{II}(\mathbf{x}_k, \mathbf{y}_{jm})] \frac{\partial \phi}{\partial \mathbf{n}}(\mathbf{y}_{jm}) J(\bar{\Gamma}_j) \right. \\
&\quad \left. + \sum_{j=1}^L \sum_{m=1}^M W_m \left[ \epsilon \frac{\partial G_{II}}{\partial \mathbf{n}}(\mathbf{x}_k, \mathbf{y}_{jm}) - \frac{\partial G_I}{\partial \mathbf{n}}(\mathbf{x}_k, \mathbf{y}_{jm}) \right] \phi(\mathbf{y}_{jm}) J(\bar{\Gamma}_j) \right. \\
&\quad \left. + \sum_{l=1, l \neq k}^{n_c} \frac{q_l}{\epsilon_I} G_I(\mathbf{x}_k, \mathbf{x}_l) \right)
\end{aligned}$$

The computation of all the summations in the parentheses is done by the fast evaluation method, KiFMM.

The remaining two terms are the volume integral over whole region  $\Omega$

If we replace the dielectric function  $\epsilon$  by the smoothing cubic function derived by Bajaj and Zhao, the dielectric boundary energy is rewritten as

$$\begin{aligned}
G^{DB} &= - \int_{\Omega} \frac{\epsilon(\mathbf{x})}{8\pi} (\nabla \phi(\mathbf{x}) \cdot \nabla \phi(\mathbf{x})) d\Omega \\
&= - \frac{1}{8\pi} \int_{\Omega} (\nabla \phi(\mathbf{x}) \cdot \nabla \phi(\mathbf{x})) (\epsilon_{II} + (\epsilon_I - \epsilon_{II}) \bar{V}(\mathbf{x}; \{\mathbf{x}_k\})) d\Omega,
\end{aligned}$$

and the same process is applied for the ionic boundary energy,

$$\begin{aligned}
G^{IB} &= - \frac{k_B T}{4\pi} \int_{\Omega} \sum_i [c_i (e^{-z_i \phi(\mathbf{x})/k_B T} - 1)] \lambda(\mathbf{x}) d\Omega \\
&= - \frac{k_B T}{4\pi} \int_{\Omega} \sum_i [c_i (e^{-z_i \phi(\mathbf{x})/k_B T} - 1)] \bar{V}(\mathbf{x}; \{\mathbf{x}_k\}) d\Omega.
\end{aligned}$$

The equations (2.1.2) and (2.1.3) show that the integration domain of  $G^{DB}$  and  $G^{IB}$  includes interior, exterior and shell regions. The volume integrals are computed using the summation over the grid points.

$$\int_{\Omega} f(x) \bar{V}(x; \{x_k\}) d\Omega = \sum_{i,j,k} f(i, j, k) \bar{V}(i, j, k)$$

### 3.2.2 The implementation of electrostatic force

In Electrostatic force acting on an atom  $\alpha$  is composed of three terms, fixed charge force  $F_{\alpha}^{QF}$  (2.2.1), dielectric boundary force  $F_{\alpha}^{DB}$  (2.2.2) and ionic boundary force  $F_{\alpha}^{IB}$  (2.2.3) The fixed charge force acting on an atom  $\alpha$  is computed using the gradient of potential computed on the atom center  $x_{\alpha}$ .

$$F_{\alpha}^{QF} = -q_{\alpha} \nabla \phi(x_{\alpha})$$

The definition of the dielectric boundary force  $F_{\alpha}^{DB}$  and the ionic boundary force  $F_{\alpha}^{IB}$  acting on an atom  $\alpha$  are the volume integrals over the domain  $\Omega$ . Based on Bajaj and Zhao's model, we can rewrite the formulas by the integral over a regular spherical shell of the width  $w$  around the atom  $\alpha$ .

$$\begin{aligned} F_{\alpha}^{DB} &= \frac{1}{8\pi} \int_{\Omega} (\nabla \phi(y) \cdot \nabla \phi(y)) \frac{\partial}{\partial x_{\alpha}} \epsilon(y) dy \\ &= \frac{\epsilon_{II} - \epsilon_I}{8\pi} \int_{\|y-x_{\alpha}\|=r_{\alpha}}^{\|y-x_{\alpha}\|=r_{\alpha}+w} (\nabla \phi(y) \cdot \nabla \phi(y)) \frac{\partial V_{\alpha}}{\partial x_{\alpha}}(y) g_{\alpha}(y) dy \end{aligned} \quad (3.1)$$

$$\begin{aligned} F_{\alpha}^{IB} &= \frac{k_B T}{4\pi} \int_{\Omega} \sum_i [c_i (e^{-z_i \phi(y)/k_B T} - 1)] \frac{\partial}{\partial x_{\alpha}} \lambda(y) dy \\ &= -\frac{k_B T}{4\pi} \int_{\|y-x_{\alpha}\|=r_{\alpha}}^{\|y-x_{\alpha}\|=r_{\alpha}+w} \sum_i [c_i (e^{-z_i \phi(y)/k_B T} - 1)] \frac{\partial V_{\alpha}}{\partial x_{\alpha}}(y) g_{\alpha}(y) dy \end{aligned} \quad (3.2)$$

where  $\phi(y)$  and  $\nabla \phi(y)$ , inside the shell region  $r_{\alpha} < \|y - x_{\alpha}\| < r_{\alpha} + w$  are computed using the equations (1.19) and (1.21).

Then, we used the technique developed by C. Bajaj and W. Zhao to switch the integration domain to the spherical coordinate system for the cal-

ulation of shell integral [12].

$$\begin{cases} x = x_\alpha + (r_\alpha + r) \cos \theta \sin \omega \\ y = y_\alpha + (r_\alpha + r) \sin \theta \sin \omega \\ z = z_\alpha + (r_\alpha + r) \cos \omega \end{cases}$$

where  $(r, \theta, \omega) \in [0, w] \times [0, 2\pi] \times [0, \pi]$ . We sample  $r, \theta, \omega$  by using the 2, 6, 6 point Gaussian quadrature nodes in each dimension as shown in Figure 3.2. Therefore, for all the atoms in the molecules, they share the same set of sampling points.

We rewrite the shell integral in  $F_\alpha^{DB}$  (3.1) and  $F_\alpha^{IB}$  (3.2) in the Gaussian quadrature form.

$$\int_{\|y-x_\alpha\|=r_\alpha}^{\|y-x_\alpha\|=r_\alpha+w} f(y) \frac{\partial V_\alpha}{\partial x_\alpha}(y) g_\alpha(y) dy = \sum_r \sum_\theta \sum_\omega W(r, \theta, \omega) f(r, \theta, \omega) \frac{\partial V_\alpha}{\partial x_\alpha}(r, \theta, \omega) g_\alpha(r, \theta, \omega) \quad (3.3)$$

where the atomic derivative of the inclusion-exclusion function  $\frac{\partial V_\alpha}{\partial x_\alpha}(r, \theta, \omega)$  in the equation (3.3) is

$$\frac{\partial V_\alpha}{\partial x_\alpha}(r, \theta, \omega) = \begin{pmatrix} \left( \frac{6r^2}{w^3} - \frac{6r}{w^2} \right) \cos \theta \sin \omega \\ \left( \frac{6r^2}{w^3} - \frac{6r}{w^2} \right) \sin \theta \sin \omega \\ \left( \frac{6r^2}{w^3} - \frac{6r}{w^2} \right) \cos \omega \end{pmatrix}.$$

The bottleneck of solving dielectric boundary force and dielectric boundary force is computing the electrostatic potential and the gradient of electrostatic potential at the Gaussian quadrature points in the shell region. The Gaussian quadrature points which are not in the shell region of the molecular surface yield the inclusion-exclusion function to be zero. We only have to consider those Gaussian quadrature points in the shell region of the molecular surface as shown in Figure 3.3.

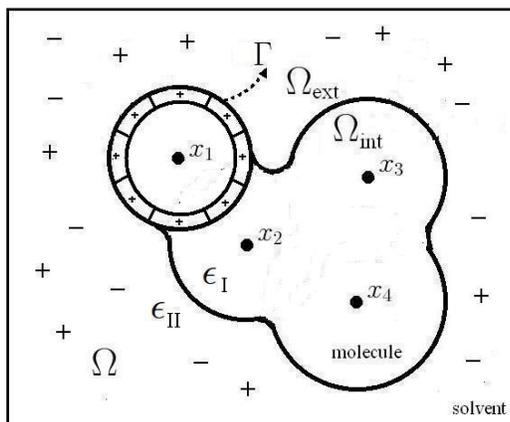


Figure 3.2: The quadrature points of the shell integral are points within a spherical shell around the atom.

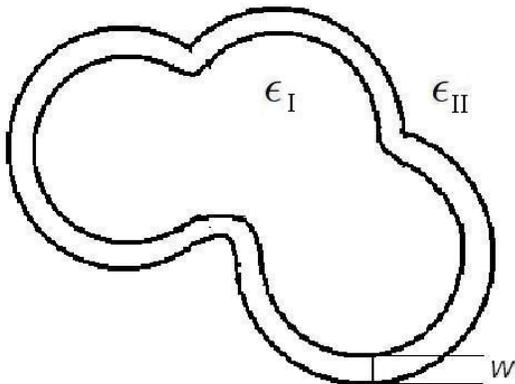


Figure 3.3: The shell region of the molecular surface  $\Gamma$

The electrostatic potential and the gradient of electrostatic potential in this shell region can then be computed using isoparametric form of the algebraic spline model. Therefore, instead of computing  $\phi$  and  $\nabla\phi$  using the boundary integral equations (1.16) and (1.20) , we can directly computed  $\phi$  and  $\nabla\phi$  using our isoparametric form of  $\phi$  and  $\nabla\phi$  such that

$$\phi(\mathbf{x}) = \mathbb{T}_1 \begin{bmatrix} \phi(\mathbf{v}_1) \\ \phi(\mathbf{v}_2) \\ \phi(\mathbf{v}_3) \\ \phi(\mathbf{v}_1 + \mathbf{n}_1) \\ \phi(\mathbf{v}_2 + \mathbf{n}_2) \\ \phi(\mathbf{v}_3 + \mathbf{n}_3) \end{bmatrix}$$

$$\nabla\phi(\mathbf{x}) = \begin{bmatrix} \mathbb{T}_{21} \\ \mathbb{T}_{22} \\ \mathbb{T}_{23} \end{bmatrix} \begin{bmatrix} \phi(\mathbf{v}_1) \\ \phi(\mathbf{v}_2) \\ \phi(\mathbf{v}_3) \\ \phi(\mathbf{v}_1 + \mathbf{n}_1) \\ \phi(\mathbf{v}_2 + \mathbf{n}_2) \\ \phi(\mathbf{v}_3 + \mathbf{n}_3) \end{bmatrix}$$

### 3.3 Experimental Results

#### 3.3.1 Complexity Analysis

Both electrostatic energy and forces are composed of three different terms. Fixed charge energy and fixed charge force is computed directly by evaluating  $\phi(\mathbf{x}_k)$  and  $\nabla\phi(\mathbf{x}_k)$  at the atomic centers  $\{\mathbf{x}_k\}_{k=1}^{n_c}$ . The computing time is to evaluate interior electrostatic potential (1.20) and interior gradient of electrostatic potential (1.16). The time complexity is  $O(n_c + L)$ . All these evaluations are done by kernel independent fast multipole method. To evaluate the dielectric boundary energy and ionic boundary energy, we have to evaluate the domain integral of electrostatic potential and inner product of gradient of electrostatic potential. Because we evaluate the volume integral over a regular grid of the domain  $\Omega$ . The time complexity is  $O(P + L)$  where  $P$  is the number of grid points we used to discretize the domain  $\Omega$ . The time complexity of computing remaining two terms of electrostatic forces is  $O(n_c + L)$  because the Gaussian quadrature points in the shell regions are proportional to the number of solvent-exposed atoms.

#### 3.3.2 Poisson-Boltzmann electrostatic solvation free energies

##### 3.3.2.1 A unit sphere with single positive charge

Only in some ideal cases, we can derive the electrostatic free energy analytically from the PB equation. To test the correctness of the PB solver, we compute the electrostatic free energy for a unit sphere with  $+1e$  single charge placed at its center and the results are shown in Table 3.1. In this

numerical method	solver name	# of grids/ # of A-patches	$G_{pol}$ ( <i>kcal/mol</i> )	relative error	compute time (seconds)
FDM	Delphi II	65 <sup>3</sup>	-82.943	2.523%	11.39
FDM	Delphi II	129 <sup>3</sup>	-82.228	1.642%	95.35
FDM	Delphi II	193 <sup>3</sup>	-82.144	1.244%	286.65
BEM	PB-CFMM	1436	-80.926	0.032%	4.63

Table 3.1: PB electrostatic free energy of a unit sphere with single charge computed using different numerical method; column 1 is the numerical method; column 2 is the name of the solver; column 3 is the number of grids for FDM and number of A-patches for BEM; column 4 is the electrostatic free energy  $G_{pol}$  (*kcal/mol*); column 5 is the relative error of electrostatic free energy  $G_{pol}$ . As a reference, the exact electrostatic free energy is  $-80.9$  *kcal/mol* with the interior and exterior dielectric constant 2 and 80; column 6 is computational time in seconds.

ideal case, the electrostatic free energy is  $-80.9$  *kcal/mol* with the interior and exterior dielectric constants 2 and 80. We can see that our BEM solution is more accurate and efficient than FDM in this case. The relative error of  $G_{pol}$  computed using BEM is much lower than that of FDM with any grid size. BEM also costs less computational time than FDM.

### 3.3.2.2 A list of ligand-receptor complexes

Using PB BEM solver, we compute electrostatic free energy for all proteins in the list of ligand-receptor complexes. In Table 3.2, we show the statistics of the PB computation using our BEM solution. We compute the average, maximum and minimum iteration number and compute time from the results of all 213 proteins ( $71 \times 3$ ). The average iteration number is smaller than 40 and not related to the number of A-patches. The computational

numerical method	BEM	BEM	BEM	BEM	FDM
# of A-patches/grids	2000	5000	10000	63444.81*	193 <sup>3</sup>
avg. # of iterations	35.21	41.72	37.14	28.86	-
max # of iterations	91	98	93	84	-
min # of iterations	13	15	19	12	-
avg. compute time (s)	60.86	216.23	418.96	1506.18	408.66
max compute time (s)	165.6	553.69	901.91	7578.36	2705.42
min compute time (s)	19.99	56.88	153.87	221.77	69.11
avg. compute time per iter (s)	2.19	6.71	13.40	61.46	-
avg. correlation of $G_{pol}$	0.852	0.927	0.948	0.960	-

Table 3.2: The statistics of the experiments including the average, maximum and minimum of the number of iterations, compute time, compute time per iterations and the correlation with Delphi FDM(193<sup>3</sup> grids) of our BEM with different number of A-patches for 213 molecules (71 sets of ligands, receptors and ligand-receptor complexes). (\* the average number of A-patches of the original triangular mesh of 213 molecular surfaces)

time includes the time of solving surface and per-atom electrostatic potential and computing electrostatic free energy. The evaluation time per iteration is linearly proportional to the number of A-patches since KiFMM is a linear time solver of fast matrix-vector product.

Meanwhile, we also observe the influence of the mesh quality to the convergence speed of iterative solution. We use average aspect ratio (twice of the ratio of the incircle radius to the circumcircle radius of a triangle) of a mesh to measure the quality of the mesh. After we compute an initial triangular mesh for the molecular surface of a protein, we applied a geometric flow algorithm to improve the quality of the mesh [80]. We observe that the average aspect ratio of a mesh goes from 0.326 to 0.430 after improving the mesh quality using geometric flow algorithm. At the same time, the average

number of iterations goes from 43.41 to 28.86. It indicates that better mesh quality will lead to faster convergence speed. The correlation of our BEM

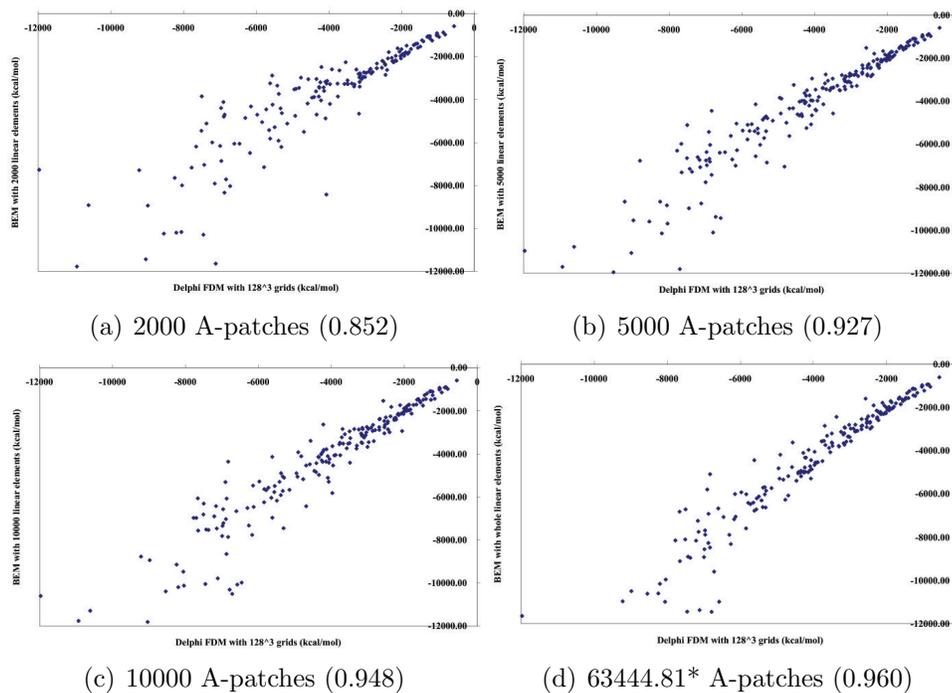


Figure 3.4: The comparison of electrostatic free energy ( $kcal/mol$ ) of 213 proteins (71 sets of ligand-receptor complexes) between BEM and FDM with  $193^3$  grids with the correlation in parentheses.

solver to Delphi II FDM solver with  $193^3$  grids are shown in Figure 3.4. Each point in the chart indicates PB electrostatic solvation free energy of a protein (ligand, receptor or their complex) computed using BEM or FDM. According to the value of correlation, we found that the more patches we used, the higher a correlation we obtained.

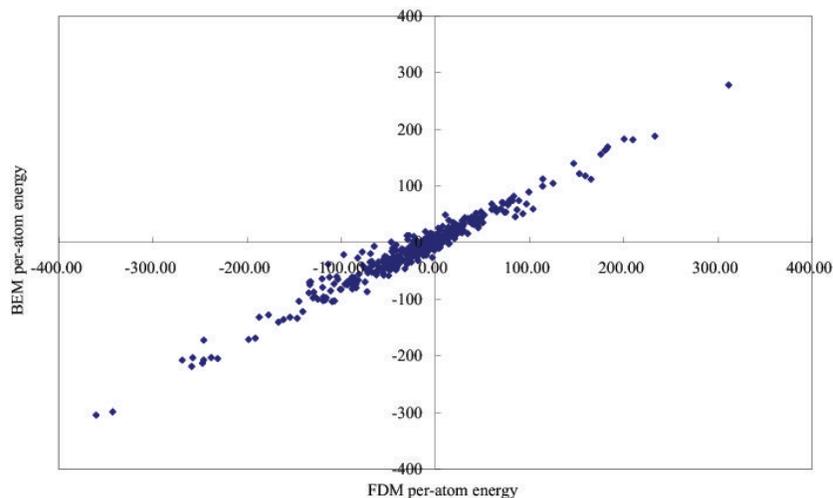


Figure 3.5: The comparison of per-atom energy (KJ/mol) of Bovine Chymotrypsinogen\*A (PDB id: 1CGI) between FDM with  $193^3$  grids and BEM with 10000 A-patches. Each point indicates the electrostatic solvation free energy of an atom.

### 3.3.3 Poisson-Boltzmann Electrostatic forces

Electrostatic force computation depends on the accurate evaluation of the gradient of electrostatic potential. It requires a very stable electrostatic potential computation. For FDM, we approximate the gradient of electrostatic potential at any specific point based on the electrostatic potential computed on each grid points. On the other hand, for BEM, we can compute the gradient of electrostatic potential at a point using potential computed along three different directions. We can observe the correlation of electrostatic forces between BEM and FDM of an example in Figure 3.6. The correlation becomes higher when the number of grids in FDM increases. It indicates that the electrostatic forces computed using FDM may converge to that computed using BEM. In Figure

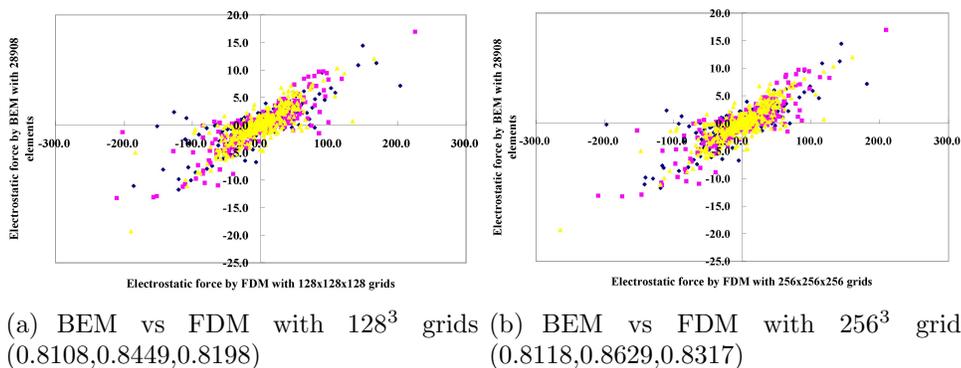
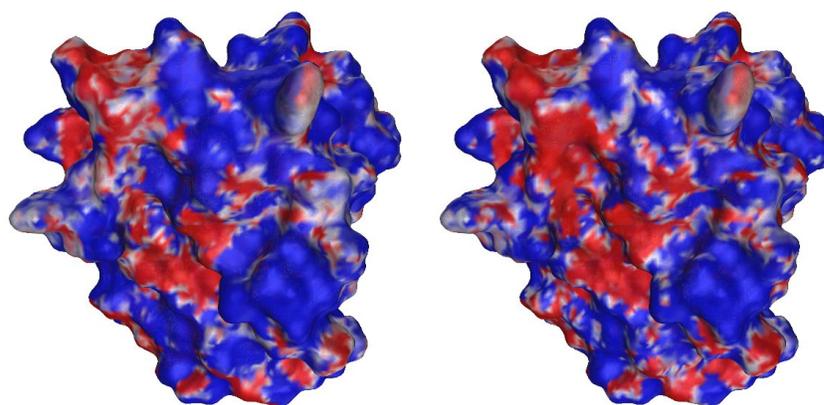


Figure 3.6: The relation of per-atom electrostatic force ( $kcal/mol \cdot \text{\AA}$ ) of Bovine Chymotrypsinogen\* A (PDB id: 1CGI) computed using BEM or FDM where blue, pink, yellow dots indicate x, y, z-dimensional values of forces; (a) BEM with 28908 A-patches vs FDM with  $193^3$  grids, the correlations at  $x, y, z$  dimensions are (0.8108,0.8449,0.8198); (b) BEM with 28908 A-patches vs FDM with  $256^3$  grids, the correlations at  $x, y, z$  dimensions are (0.8118,0.8629,0.8317).

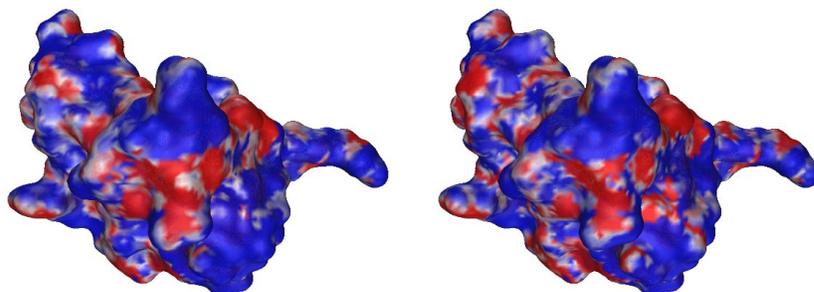
3.7, we show PB electrostatic forces of two protein examples (PDB id: 1A2K and 1CGI). The color of the molecular surface represents the inner product of the electrostatic forces and the unit surface normals. The outward force gives a positive inner product and negative otherwise. The color is going from blue ( $\geq 3.8 kcal/mol \cdot \text{\AA}$ ) to red ( $\leq -3.8 kcal/mol \cdot \text{\AA}$ ). We can see that the distribution of inward and outward forces computed using BEM and FDM are almost the same.

The electrostatic force computation depends on the accurate computation of the gradient of electrostatic potential and the approximation of the dielectric function and ionic boundary. In this part, we found that if we used the fast multipole method to compute the integrals of three electrostatic force



(a) BEM with 54592 A-patches

(b) FDM with  $256^3$  grids



(c) BEM with 29108 A-patches

(d) FDM with  $256^3$  grids

Figure 3.7: The inner product of unit normal vector and PB electrostatic forces on the molecular surface of nuclear transport factor 2 (PDB id: 1A2K) and Bovine Chymotrypsinogen\*A (PDB id: 1CGI) with different resolutions. The color is going from blue ( $\geq 3.8 \text{ kcal/mol}\cdot\text{\AA}$ ) to red ( $\leq -3.8 \text{ kcal/mol}\cdot\text{\AA}$ ).

terms, the numerical error will be amplified. Therefore, we still used direct computation to deal with force computation. On the other hand, in both BEM and FDM solutions, we used Im's volume exclusion function to approximate the derivatives of the dielectric function and ionic boundary function in  $F_{db}$  and  $F_{ib}$ . This approximate function is used for computing the  $\nabla\epsilon(x)$  term in dielectric boundary force and the  $\nabla\lambda(x)$  term in the ionic boundary force.

### 3.3.4 interior and exterior electrostatic potential

We compute the electrostatic potential  $\phi$  of proteins by solving dBIEs described from PB equation. In Figure 3.8 (b), we show the electrostatic potential on the molecular surface (Figure 3.8 (a)) of an example (PDB id: 1CGI) in the protein list. PB electrostatic potential is computed with different numbers of A-patches. The color of the surface represents the electrostatic potential on the molecular surface, going from red ( $\phi < -3.8 k_bT/e_c$ ) to blue ( $\phi > 3.8 k_bT/e_c$ ) and white is neutral potential.

In Figure 3.8 (c), we compute the PB electrostatic potential of the protein on the center of a volume  $64 \times 64 \times 64 \text{ \AA}^3$ . In Figure 3.8 (d), we add the molecular surface of the protein into this volume. The Figures 3.8 (e) and (f) are the cutting slices of this volume.

Figure 3.9 shows the reaction electrostatic potential  $\phi_{rf} = \phi_{sol} - \phi_{air}$  of (a)nuclear transport factor 2 (PDB id: 1A2K) and (b) Bovine Chymotrypsinogen\*A (PDB id: 1CGI).

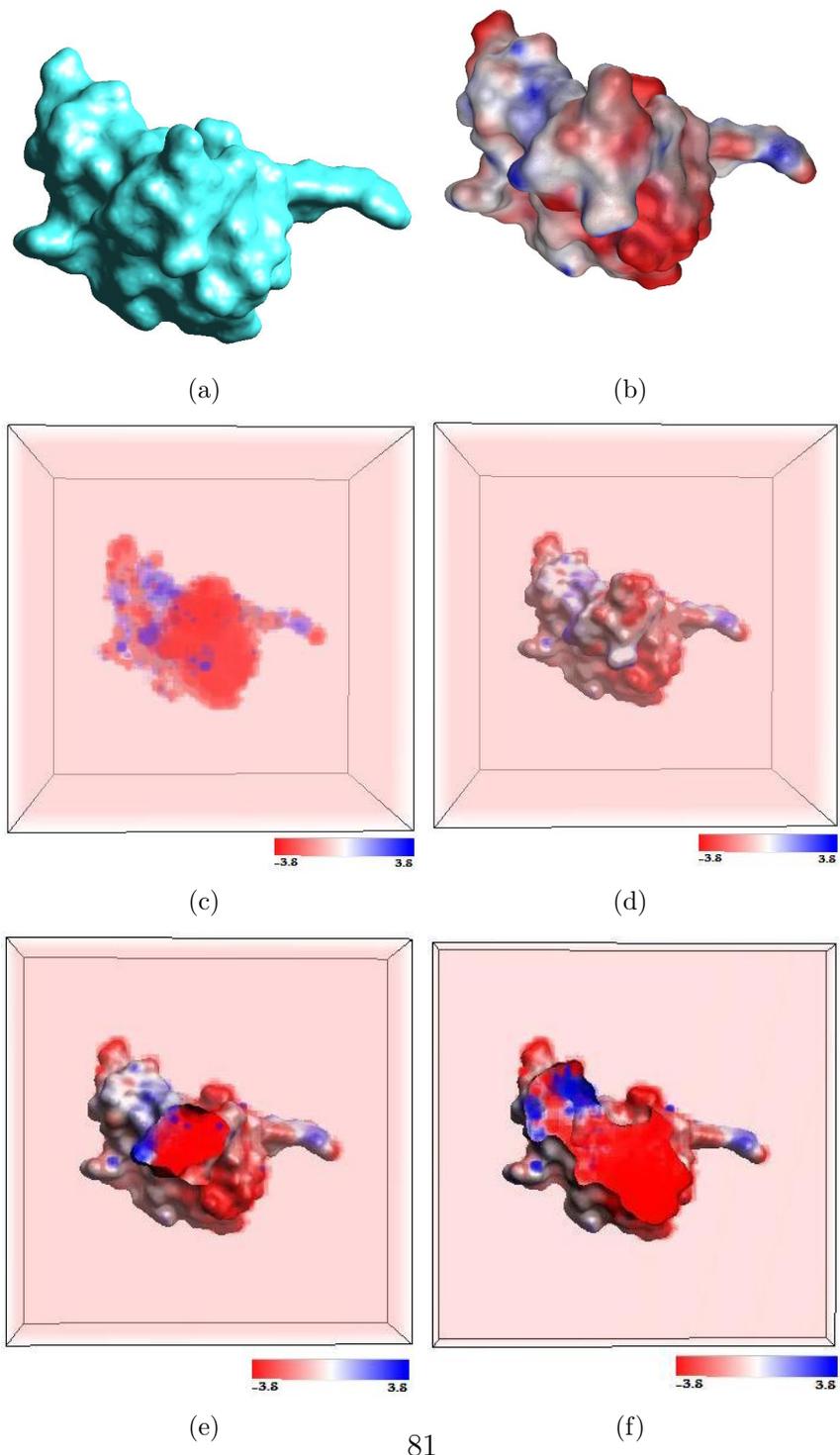


Figure 3.8: The PB electrostatic potential  $\phi$  of Bovine Chymotrypsinogen\*A (PDB id: 1CGI). The color is going from red ( $\phi < -3.8 k_bT/e_c$ ) to blue ( $\phi > 3.8 k_bT/e_c$ ). (a) the molecular surface; (b) the molecular surface colored by  $\phi$ ; (c)  $\phi$  distributed in a volume  $64 \times 64 \times 64 \text{ \AA}^3$  with the molecule on the center; (d) with the molecular surface colored by  $\phi$ ; (e),(f) the cutting slices of the volume.

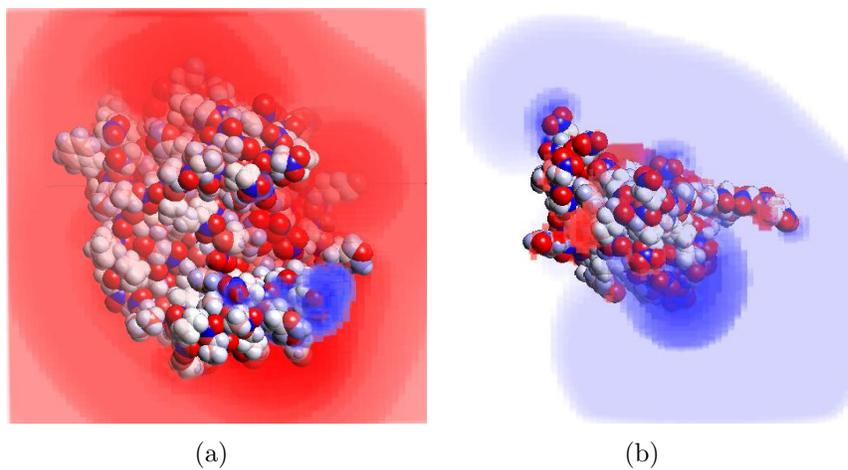


Figure 3.9: The PB reaction field electrostatic potential  $\phi_{rf} = \phi_{sol} - \phi_{air}$  of (a) nuclear transport factor 2 (PDB id: 1A2K) and (b) Bovine Chymotrypsinogen\*A (PDB id: 1CGI). The color is going from red ( $\phi_{rf} < -3.8 k_bT/e_c$ ) to blue ( $\phi_{rf} > 3.8 k_bT/e_c$ ).

# Chapter 4

## Hierarchical Molecular Modeling

## 4.1 Background

The 3-D molecular structure of a molecule with its charge distribution is called its molecular model. The most common and basic molecular model is the 3-D structure based on all the atoms of a molecule. However, because a bio-molecule is composed of thousands to millions of atoms, it makes the simulation computation costly. Therefore, Several important efforts are devoted to develop lower resolution models with reasonable accuracy. Such low-resolution reduced protein models are also called coarse-grained models. They are historically developed to handle the folding problems. The prototype of the non-lattice CG model was Smit and coworkers' model in which a lipid molecule is partitioned into a hydrophilic 'head' and a hydrophobic 'tail' and all the particles interact via a sort of simplified Lennard-Jones potential [71]. A systematic CG approach was developed by Shelley et al [68, 69] where the CG model is systematically parameterized to mimic the existing force fields or statistical mechanical properties obtained from the atomistic models of phospholipid. Marrink et al [59] improve the CG model of lipid by classifying the CG beads into different types according to properties such as hydrophobicity, hydrogen bonding capacity, and charge. We developed a hierarchical coarse-grained clustering to control and generate reduced molecular models especially for electrostatic computation.

## 4.2 Hierarchical molecular models

Several levels of CG representations of the proteins have also been developed. One of the earliest and simplest models is the  $G\bar{o}$  model which represents the polypeptide chain as a chain of  $C_\alpha$  atoms with attractive or repulsive non-bonded interactions only [74]. This model has been developed to add one more bead on each side chain (SC) [4, 25]. In the  $C_\alpha$ -SC-Pep model [26], an additional interaction center (Pep) is added on the backbone in the middle of the C-N peptide bond which strongly improves the orientation-dependent potentials. In [19] extended side chains (such as Arg, Lys, etc.) are represented by two beads in order to have CG beads of about the same size. A four-bead model is given in [72] in which each residue is explicitly represented by three heavy atoms on the backbone and one SC bead. Most recently a multi-resolution CG model is developed in [3] which allows coarser and finer representations in different parts of the macromolecule and therefore fixes the deficiency of assigning each CG bead to the same number of atoms. An ellipsoid CG model is presented in [41].

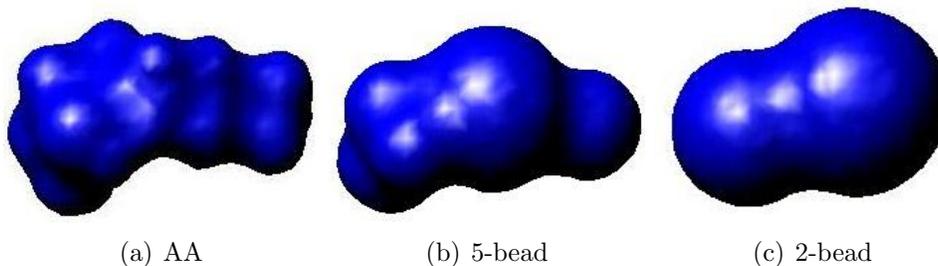


Figure 4.1: An example that the atoms of an amino-acid are clustered into 5 beads and 2 beads.

In our study, we generate the CG model in three steps without introducing too much error. First we build a hierarchical clustering of the atoms according to the hierarchy of the protein structure. In the hierarchy, from top to bottom, they are the tertiary structures, secondary structures, residues, backbone and side chains, functional groups, and atoms. According to this clustering, one can represent the atoms in a group as one CG bead. Since at the top levels, too much detail of the protein is lost, the coarsest CG model we build for our current energy and force calculation is one bead for each amino acid group. In Figure 4.1, you can see an example that the atoms of an amino-acid are clustered into 5 beads and 2 beads.

In the second step, we compute the new locations and sizes of the CG beads. Our goal is to let the new molecular surface of the CG model be as much close to the surface of the AA model as possible. Since the molecular surface can be approximated by the level set of the Gaussian density function, for the accuracy purpose, we individually find the centers  $(\{\vec{x}'_k\}_{k=1}^{n'_c})$  and the radius  $(\{r'_k\}_{k=1}^{n'_c})$  for each CG bead such that the Gaussian density function which is defined as  $\varrho(\vec{x}) = e^{-C_i(\|\vec{x}-\vec{x}'_k\|^2-(r'_k)^2)}$ , where  $C_i$  is a Gaussian decay rate, agrees with the density function  $G_i(\vec{x}) = \sum_{k=i_1}^{i_{M_i}} e^{-C_k(\|\vec{x}-\vec{x}_k\|^2-r_k^2)}$ , where atoms  $i_1 \dots i_{M_i}$  are grouped into bead  $i$ . This is done by solving the least square problem

$$\min \frac{1}{2} \sum_{j=1}^n [\varrho(\vec{x}_j) - G_i(\vec{x}_j)]^2,$$

where  $\vec{x}_j$  are sample points on the level set  $\{\vec{x} : G_i(\vec{x}) = 1\}$ . Figure 4.2 shows an example that all atoms are grouped into 2 beads. Their new centers  $\mathbf{x}'_1, \mathbf{x}'_2$

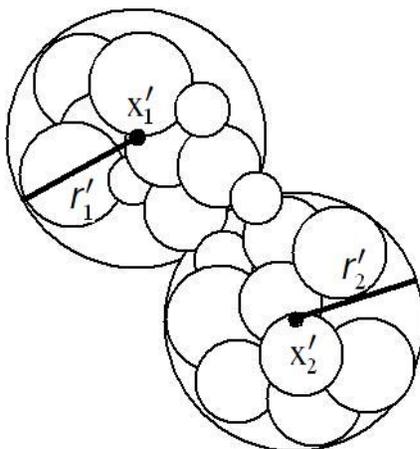


Figure 4.2: An example that all atoms are grouped into 2 beads. Their new centers  $x'_1, x'_2$  and radius  $r'_1, r'_2$  are computed.

and radius  $r'_1, r'_2$  are computed.

In the third step, we assign charges to the CG beads such that the electrostatic solvation energy of the CG model reproduces that of the atomic model. We use the GB energy function [73] as the objective function and the optimization is subject to the constraint that the total charge of the molecule does not change. This constrained nonlinear optimization problem is solved by using the Levenberg-Marquardt algorithm [58]. The details of the CG model generation are described in [13].

In Figure 4.3, we show an example of the atomic and the coarse-grained model and the corresponding molecular surface. In the CG model, each amino acid is split into two groups and therefore two beads, one for the backbone and the other for the side chain. The molecular surface in (a) and the molecular surface in (b) are very similar. The Hausdorff distance between them is 1.998

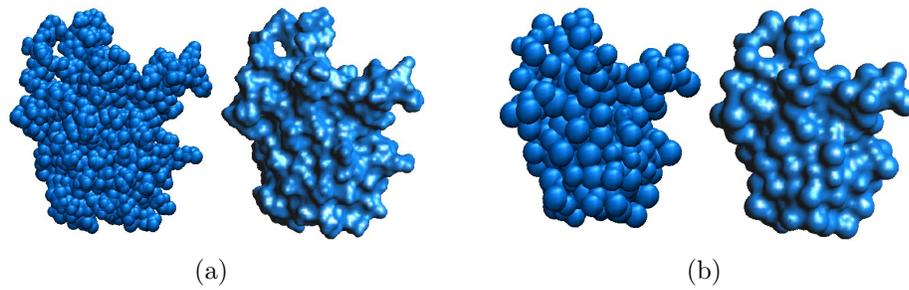


Figure 4.3: (a) The Atomic model of protein 1AK4 (2260 atoms) and its SES surface. (b) The CG model (290 atoms) of 1AK4 and its SES surface. Each amino acid is represented by two beads, one for the backbone and one for the side chain. The Hausdorff distance between them is 1.998 Å.

Å.

### 4.3 Experimental results

In the following experiments, we construct three different levels of molecular model, including AA model, 5-bead CG model (5 beads per residue; three for side chain and two for backbone), and 2-bead CG model (2 beads per residue; one for side chain and one for backbone), for these proteins. These molecular models are in hierarchy. Based on these molecular models, the molecular interfaces are generated. The coefficients of solute-solvent environment in our experiment are shown in Table 4.1. We compute PB electrostatic

temperature $T$	298.15 $K$
solvent dielectric $\epsilon_{II}$	80.0
solute dielectric $\epsilon_I$	1.0
probe radius	1.4 $\text{\AA}$
ion concentration $\iota$	0.05 $M$

Table 4.1: The coefficients of implicit solvent model which is used for PB boundary element method.

energy, potential and forces using different models. All experiments are done on a linux machine with Dual Core AMD Opteron processor 280 with 4 GB memory. The matrix-vector product is computed using kernel independent fast multipole method developed by Lexing Ying. [78] It is a highly accurate compression technique to evaluate the integral of Poisson-Boltzmann kernels over a set of points in linear time.

We discussed and estimate the error and computational time of different solutions in the following experiments.

### 4.3.1 Molecular surface comparison

PDB ID	# atoms	Hausdorff dist.	surface area	error of area
1AK4 (l)	2260 / 401 / 290	1.735 / 1.998	6915.46 / 6904.60 / 6817.75	0.16% / 1.41%
1AK4 (r)	2503 / 440 / 330	5.240 / 2.592	6156.69 / 6183.13 / 6011.58	0.43% / 2.36%
1AVX	2662 / 477 / 344	1.609 / 2.059	7446.88 / 7458.57 / 7242.72	0.16% / 2.74%
1AY7	2875 / 517 / 370	2.263 / 4.059	7659.91 / 7553.31 / 7424.50	1.39% / 3.07%
1AY7 (l)	1434 / 251 / 178	1.714 / 2.196	3934.07 / 3973.79 / 3914.01	1.01% / 0.51%
1AY7 (r)	1441 / 266 / 192	1.604 / 4.765	4525.75 / 4452.77 / 4485.75	1.61% / 0.88%
1B6C	1663 / 290 / 214	1.502 / 3.622	4836.82 / 4804.62 / 4668.27	0.67% / 3.48%
1BJ1	2986 / 544 / 376	1.630 / 3.663	8913.19 / 8899.72 / 8832.07	0.15% / 0.91%
1BUH	1190 / 205 / 140	2.651 / 2.017	3822.71 / 3876.85 / 3699.82	1.41% / 3.21%
1CGI	852 / 157 / 112	1.881 / 3.139	3186.14 / 3244.90 / 3081.38	1.84% / 3.29%
average	1987 / 355 / 255	2.183 / 3.011	5739.762 / 5735.226 / 5617.785	0.883% / 2.186%

Table 4.2: The comparison of molecular surfaces of a set of proteins. Each term from left to right is the result of AA model, 5-bead CG model and 2-bead CG model; column 1 is PDB id of the protein where (l) and (r) indicate the ligand and receptor of the complex protein; column 2 shows the number of atoms or beads in the AA/5-bead CG/2-bead CG models, respectively; column 3 shows the Hausdorff distance  $\text{\AA}$  between the molecular surfaces of the 5-bead CG/2-bead CG to the molecular surfaces of the AA models; column 4 shows the surface area  $\text{\AA}^2$  of the AA/5-bead CG/2-bead CG models, respectively. The last column shows the relative error of the surface area of the 5-bead CG/2-bead CG models which are computed as (area of the CG model - area of the AA model)/(area of the AA model).

Here, we evaluate the molecular surfaces by Hausdorff distance between the surfaces computed from CG model and AA model where all the surfaces are constructed using our geometric flow evolution technique.

The experimental results of those proteins are given in Table 4.2 where each line in the table indicates the model information and the experimental results of PB electrostatic computation of a protein computed using AA, 5-bead CG and 2-bead CG models. Here, we can observe that the number of beads of 5-bead CG model is reduced to less than 1/5 number of the original atoms and that of 2-bead CG model is reduced to about 1/8. Even through

the numbers of beads are quite few, we can still get a good approximate surface where average Hausdorff distances of 5-bead and 2-bead CG molecular surfaces to AA molecular surface are low (2.183 Å and 3.011 Å) and in most of cases, the area of 5-bead molecular surface is closer to the area of AA molecular surface than that of 2-bead molecular surface. Hausdorff distance of 5-bead CG molecular surface is smaller than that of 2-bead CG molecular surface in most of the cases. Because the boundary element method relies on surface integral computation, the better CG molecular surfaces approximate AA molecular surfaces, the more accurate the PB solution should be. In the following PB electrostatic experiments, we can observe the actual influence caused by these approximation errors.

### 4.3.2 Poisson-Boltzmann electrostatic free energy

The relation of electrostatic solvation free energy between AA model and different levels of CG models of proteins are also shown in Table 4.3.

The correlation of electrostatic solvation free energy between 5-bead CG and AA models is up to 0.99917. This result indicates that the evaluation accuracy of the electrostatic free solvation energy using 5-bead CG model is consistently satisfactory, while that of using coarser 2-bead CG model is not. The errors of PB energy computation of 2-bead CG model are from 0.016 to 0.385 in this set of proteins.

In addition, we can see a binding example (PDB id:1AY7), which is a microbial ribonuclease complex with Barstar. The binding electrostatic energy

of this case can be evaluated using the following equation.

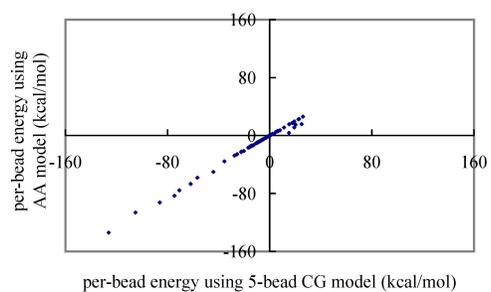
$$\Delta G_{bind} = G_{complex} - (G_{molecule1} + G_{molecule2})$$

Therefore, the binding electrostatic energy computed using AA model is  $-287.06$  *kcal/mol*. The negative energy indicates that the complex is a more stable structure than distinct molecules. If we used 5-bead and 2-bead CG model, the binding electrostatic energy are  $-293.19$  *kcal/mol* and  $-216.58$  *kcal/mol*. Both of them are also negative but 5-bead CG model is still more accurate than 2-bead CG model.

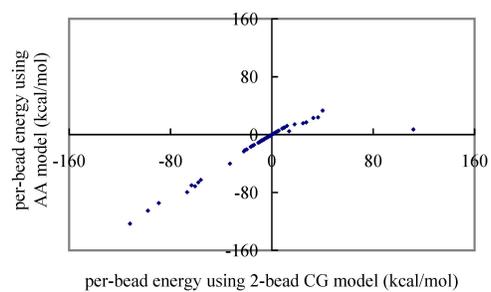
These results give us a message that there is a limitation of coarse-grained clustering. If we keep doing the clustering and make much coarser molecular model, the model cannot guarantee to be a good approximation of AA model. 5-bead CG model with radius and charge optimization is shown to be a great approximation for PB energetics.

Meanwhile, the reduction of the atom/bead number decreases the number of A-patch because the complexity of molecular surface becomes lower. The decrease of the number of patches improves the computation cost of electrostatic free energy computation.

Now, we take one of those proteins (PDB id: 1CGI) as an example to see the details of PB energetic results. Figure 4.4 (a) shows that the electrostatic solvation free energy of each CG bead of the protein computed using 5-bead CG model is highly related to the energy computed using AA model. The correlation of per-bead electrostatic free energy between AA model and CG



(a)



(b)

Figure 4.4: The relation of per-bead electrostatic energy ( $kcal/mol$ ) between AA molecular model and CG molecular models of the protein (PDB id: 1CGI); (a) AA model vs 5-bead CG model, the correlation is 0.9950; (b) AA model vs 2-bead CG model, the correlation is 0.9135.

PDB id	# atoms/beads	# A-patches	$G_{pol}$	relative error of $G_{pol}$	compute time
1AK4 (l)	2260/401/290	12829/10294/9438	-1638.17/-1560.44/-1790.65	-/0.047/0.093	682.85/446.40/417.88
1AK4 (r)	2503/440/330	11730/9562/8568	-1907.00/-1862.45/-1689.96	-/0.023/0.114	661.41/535.04/470.18
1AVX	2662/477/344	12468/10341/9547	-3349.88/-3209.52/-3220.84	-/0.041/0.039	614.96/437.93/402.25
1AY7	2875/517/370	13493/10602/9694	-3657.04/-3591.34/-3061.09	-/0.018/0.163	973.38/454.36/424.88
1AY7 (l)	1434/251/178	9049/7106/6461	-1601.60/-1576.64/-1576.64	-/0.016/0.016	438.89/303.32/204.92
1AY7 (r)	1441/266/192	10594/8548/5208	-1768.38/-1721.51/-1267.87	-/0.027/0.283	630.15/435.05/386.96
1B6C	1663/290/214	10062/8051/7411	-1342.30/-1300.59/-896.64	-/0.031/0.332	484.07/425.44/326.63
1BJ1	2986/544/376	13022/11167/10230	-3812.96/-3712.58/-3031.71	-/0.026/0.205	587.75/586.55/501.30
1BUH	1190/205/140	8996/6983/6188	-1456.83/-1510.51/-1494.61	-/0.037/0.026	675.80/438.34/244.29
1CGI	852/157/112	7277/6312/5497	-971.77/-931.69/-597.23	-/0.041/0.385	374.49/258.66/243.36
average	1987/355/255	10952/8897/7824	-2150.59/-2097.73/-1862.72	-/0.031/0.166	612.38/432.11/362.27

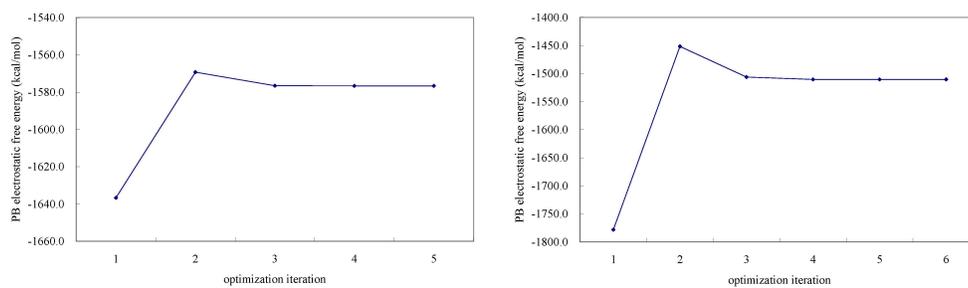
Table 4.3: The experimental results of a set of proteins. Each term from left to right is the result of AA model, 5-bead CG model and 2-bead CG model; column 1 is PDB id of the protein where (l) and (r) indicate the ligand and receptor of the complex protein; column 2 is number of atoms of AA model and number of beads of 5-bead CG and 2-bead CG model; column 3 is the number of A-patches of AA, 5-bead CG and 2-bead CG models; column 4 is electrostatic free energy  $G_{pol}$  ( $kcal/mol$ ), 5-bead CG model is a good approximation; column 5 is the relative error of electrostatic free energy computed using 5-bead CG and 2-bead CG models to that computed using AA model; column 6 is computation time of PB boundary element method (*seconds*).

model is 0.9950. Figure 4.4 (b) shows the same comparison between 2-bead CG model and AA model. The correlation is 0.9135. As expected, 5-bead CG model performs better than 2-bead CG model.

In Figure 4.5, we show the PB electrostatic solvation free energy at each iteration of the charge optimizing process of coarse-grained model. These results show that the optimization based on GB objective function also makes PB results convergent in each case.

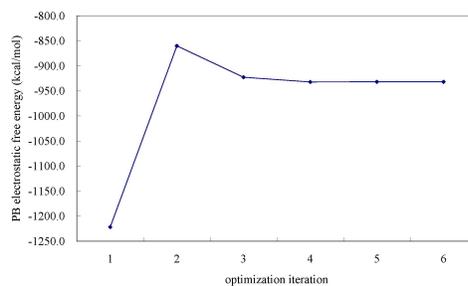
### 4.3.3 Poisson-Boltzmann electrostatic potential

Here, we compute electrostatic potential of proteins using AA, 5-bead and 2-bead CG models. We also take one of those proteins (PDB id: 1CGI) as



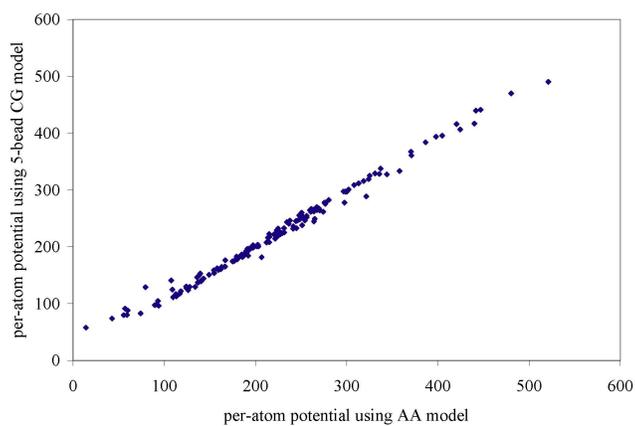
(a) PDB id: 1AY7

(b) PDB id: 1BUH

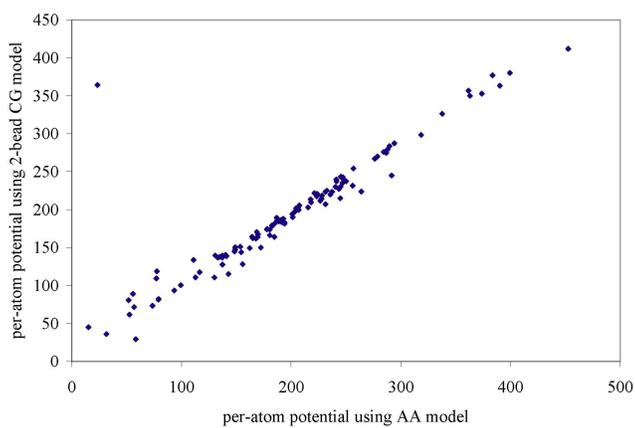


(c) PDB id: 1CGI

Figure 4.5: The PB electrostatic electrostatic free energy ( $kcal/mol$ ) during the charge optimizing process of 5-bead CG model of each protein. All of them are convergent.



(a)



(b)

Figure 4.6: The relation of per-bead electrostatic potential ( $k_bT/e_c$ ) between AA molecular model and CG molecular models of the protein (PDB id: 1CGI); (a) AA model vs 5-bead CG model, the correlation is 0.9950; (b) AA model vs 2-bead CG model, the correlation is 0.9059.

an example to see the details of PB electrostatic potential results. Figure 4.6 (a) shows that the reaction field electrostatic potential of each CG beads of the protein computed using the 5-bead CG molecular model is highly related to that computed using the AA molecular model. The correlation between them is 0.9950. The same experiment is applied for 2-bead CG model and its correlation is 0.9059.

We show examples in Figure 4.7. The color of the surface represents the electrostatic potential on the molecular surface, going from red (potential of  $-3.8 k_bT/e_c$ ) to blue (potential of  $+3.8 k_bT/e_c$ ) and white is neutral potential.

The surface electrostatic potential is widely used to evaluate several bio-molecular activities. Most of the protein docking problems, e.g. ligand binding, rely on the mapping of the surface electrostatic potential. As we mentioned in the introduction, the coarse-grained model is an important technique for reducing the degree of freedom of protein docking problem. Meanwhile, with a good approximation of the coarse-grained molecular surface, BEM can provide highly accurate PB surface electrostatic potential. From our observation, the distribution of electrostatic potential on the molecular surfaces in different levels are highly related. The parts of the AA molecular surface with highly positive or negative electrostatic potential will still hold in CG cases.

#### 4.3.4 Poisson-Boltzmann electrostatic solvation force

The simulation of a protein activity is determined by its electrostatic forces. The accuracy and computation cost of electrostatic force computation

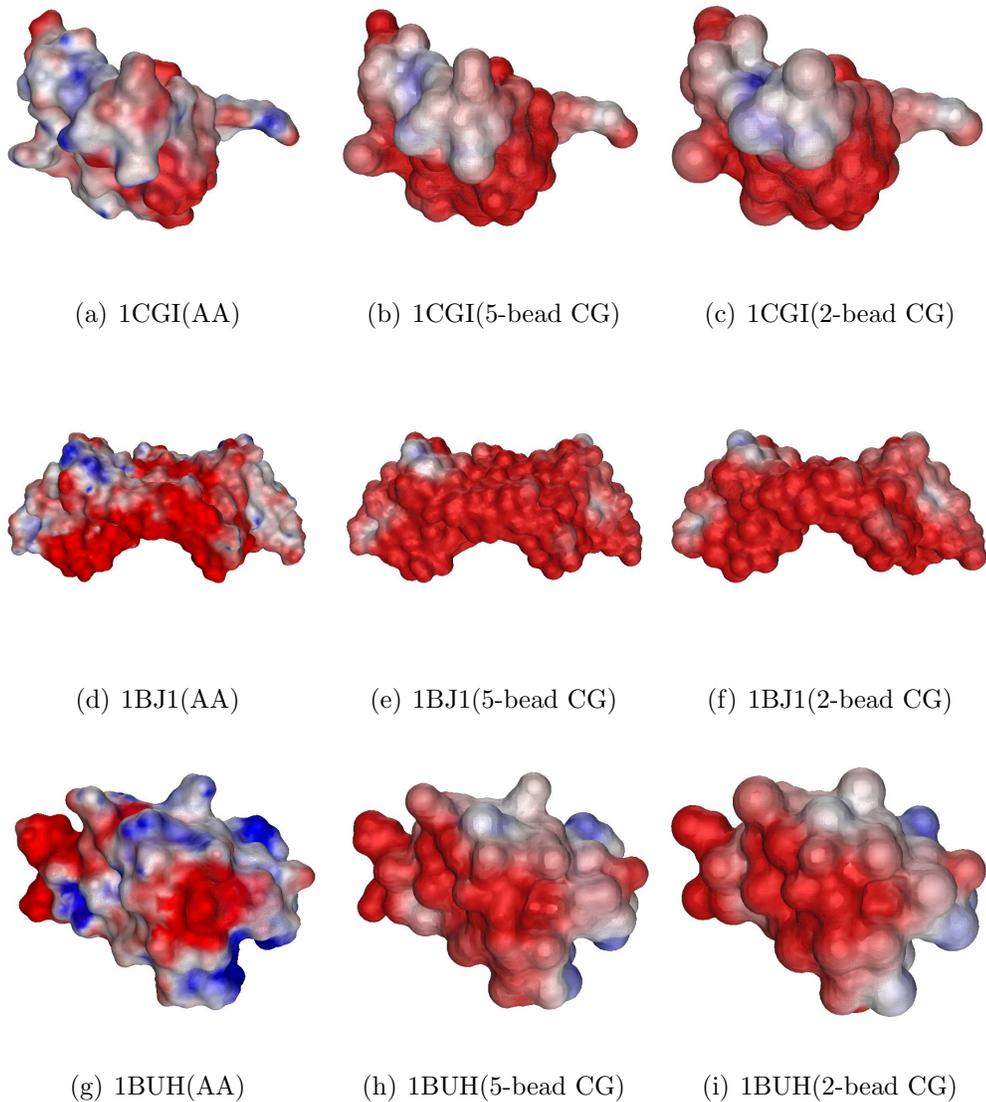
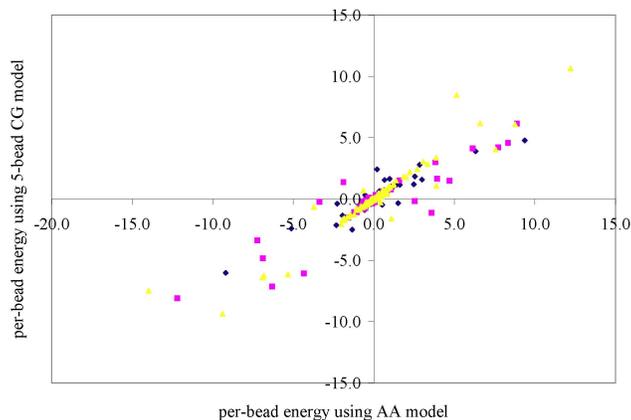


Figure 4.7: The PB electrostatic potential on molecular surface with different resolutions (atomic and coarse-grained resolution). The visual errors between AA and all CG models are small. The color is going from red (potential of  $-3.8 k_bT/e_c$ ) to blue (potential of  $+3.8 k_bT/e_c$ ).

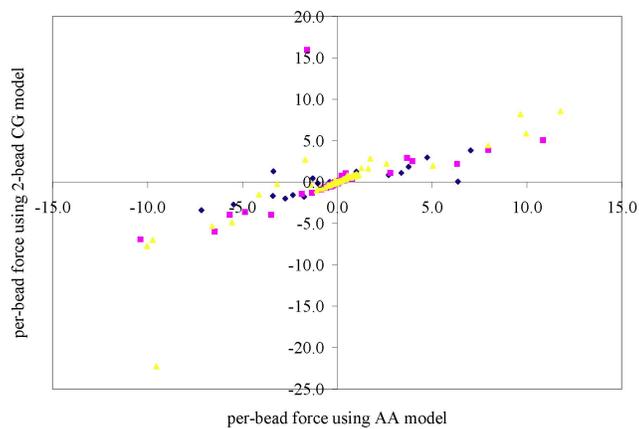
directly affect the simulation results. While coarse-grained model is widely applied to reduce the computational time, the accuracy of electrostatic force computation based on the reduced model is an important aspect.

In Figure 4.8 (a), we show the relation of per-bead electrostatic forces computed by AA model and 5-bead CG model for a protein (PDB id: 1CGI). The unit of the force is  $kcal/mol \cdot \text{\AA}$ . Blue, pink and yellow dots on the chart are the value of forces at  $x, y, z$  dimensions. The correlations between them are 0.9223, 0.9117 and 0.9448 at  $x, y, z$  dimensions. 5-bead CG model is a reasonably good approximate model for electrostatic force computation but doesn't give such high correlation as the per-atom energy in Figure 4.4 (a). The same experiment is done for 2-bead CG model and shown in Figure 4.8 (b). The correlations at  $x, y, z$  dimensions are 0.8224, 0.6773 and 0.8472. As we observed in the experiments of electrostatic energy and potential computation, the coarser our model is, the less accurate the results are. Here, the electrostatic force computation looks more sensitive to the resolution of molecular models than the electrostatic energy or potential computation. In this case, 2-bead CG model is definitely not a good approximation for electrostatic force computation.

In Figure 4.9, we show the electrostatic solvation force on the molecular surfaces of different proteins. The color of the molecular surface represents the inner product of the electrostatic force and the surface normal at the surface point. The outward force gives the positive inner product and negative otherwise. The color is going from blue ( $\geq 7.6 kcal/mol \cdot \text{\AA}$ ) to red ( $\leq -7.6$



(a)



(b)

Figure 4.8: The relation of per-bead electrostatic force ( $kcal/mol \cdot \text{\AA}$ ) between AA molecular model and CG molecular models of the protein (PDB id: 1CGI) where blue, pink, yellow dots indicate  $x, y, z$ -dimensional values of forces; (a) AA model vs 5-bead CG model, the correlations at  $x, y, z$  dimensions are 0.9223, 0.9117, 0.9448; (b) AA model vs 2-bead CG model, the correlations at  $x, y, z$  dimensions are 0.8224, 0.6773, 0.8472.

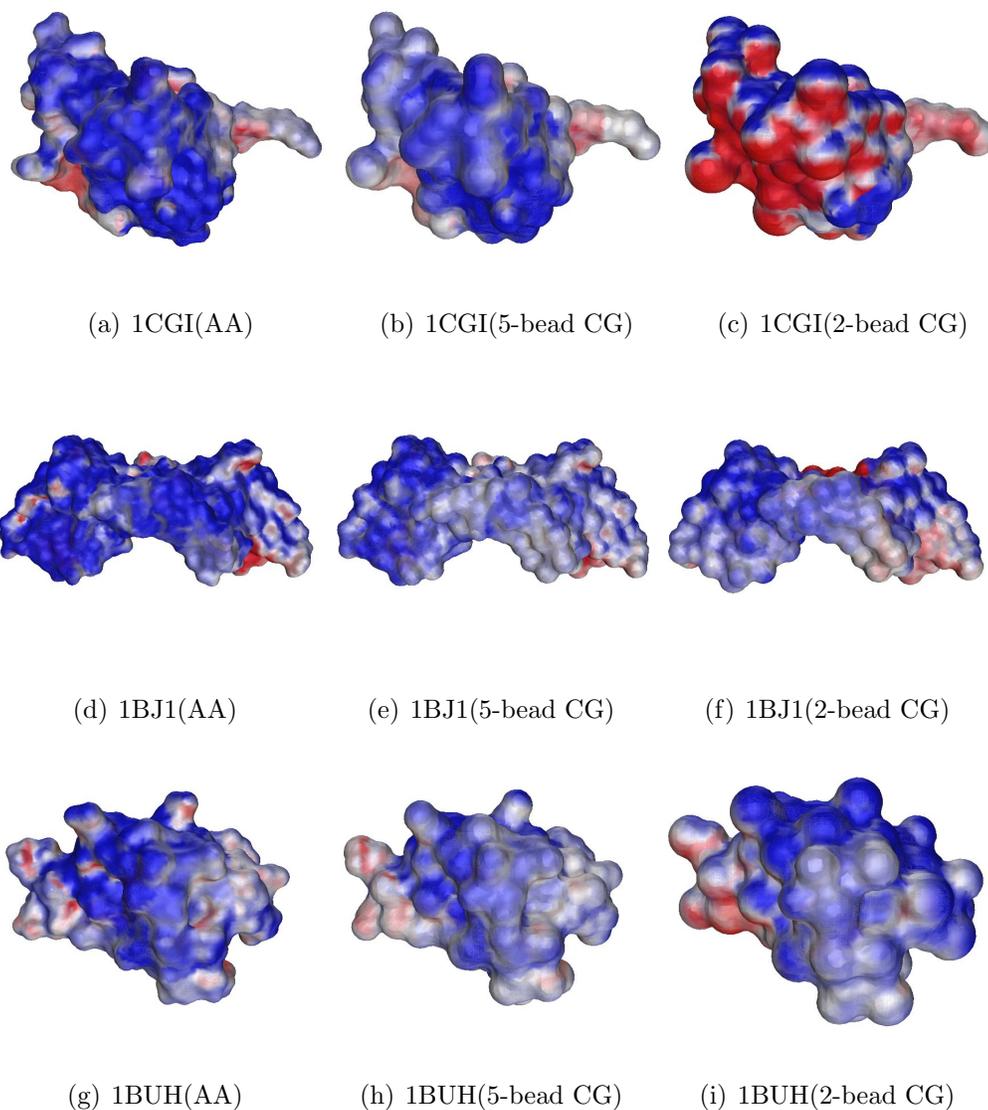


Figure 4.9: The inner product of unit normal vector and PB electrostatic forces on the molecular surface with different resolutions (atomic and coarse-grained resolution). The visual error between AA and 5-bead CG models is small but not between AA and 2-bead CG model. The color is going from blue ( $\geq 7.6 \text{ kcal/mol}\cdot\text{\AA}$ ) to red ( $\leq -7.6 \text{ kcal/mol}\cdot\text{\AA}$ ).

$kcal/mol \cdot \text{\AA}$ ). The distribution of inward and outward forces is almost the same in AA and 5-bead CG models but not in AA and 2-bead CG models.

## Bibliography

- [1] N. Akkiraju and H. Edelsbrunner. Triangulating the surface of a molecule. *Discr. Appl. Math.*, 71:5–22, 1996.
- [2] M. D. Altman, J. P. Bardhan, J. K. White, and B. Tidor. Accurate solution of multi-region continuum biomolecule electrostatic problems using the linearized poisson-boltzmann equation with curved boundary elements. *J. Comput Chem.*, 30:132–153, 2009.
- [3] A. Arkhipov, P. Freddolino, K. Imada, K. Namba, and K. Schulten. Coarse-grained molecular dynamics simulations of a rotating bacterial flagellum. *Biophys. J.*, 91:4589–4597, 2006.
- [4] I. Bahar and R. Jernigan. Inter-residue potentials in globular proteins and the dominance of highly specific hydrophilic interactions at close separation. *J. Mol. Biol.*, 266:195–214, 1997.
- [5] C. Bajaj, P. Djeu, V. Siddavanahalli, and A. Thane. Texmol: Interactive visual exploration of large flexible multi-component molecular complexes. In *Proceedings of the Annual IEEE Visualization Conference'04*, pages 243–250, Austin, Texas, October 2004.
- [6] C. Bajaj, H. Lee, R. Merkert, and V. Pascucci. Nurbs based b-rep models from macromolecules and their properties. In *Proceedings of*

*Fourth Symposium on Solid Modeling and Applications*, pages 217–228, 1997.

- [7] C. Bajaj, V. Pascucci, A. Shamir, R. Holt, and A. Netravali. Dynamic maintenance and visualization of molecular surfaces. *Discrete Applied Mathematics*, 127:23–51, 2003.
- [8] C. Bajaj and G. Xu. A-splines: Local interpolation and approximation using  $g_k$ -continuous piecewise real algebraic curves. *Computer Aided Geometric Design*, 16:557–578, 1999.
- [9] C. Bajaj and G. Xu. Smooth shell construction with mixed prism fat surfaces. *Brunett, G., Bieri, H., Farin, G. (eds.), Geometric Modeling Computing Supplement*, 14:19–35, 2001.
- [10] C. Bajaj, G. Xu, and Q. Zhang. Higher-order level-set method and its application in biomolecule surfaces construction. *Journal of Computer Science and Technology*, 23(6):1026–1036, 2008.
- [11] C. Bajaj, G. Xu, and Q. Zhang. A fast variational method for the construction of resolution adaptive  $c^2$ -smooth molecular surfaces. *Comput. Methods Appl. Mech. Engrg.*, page to appear, 2009.
- [12] C. Bajaj and W. Zhao. Fast molecular solvation energetics and force computation. *SIAM Journal on Scientific Computation*, 2009.
- [13] C. Bajaj, W. Zhao, G. Xu, and S. Chen. Error-bounded coarse-grained models for biomolecules. *ICES Report*, 2009.

- [14] Chandrajit Bajaj and Shun-Chuan Albert Chen. Efficient and accurate higher-order fast multipole boundary element method for poisson boltzmann electrostatics. *UTCS technical report, submitted to SIAM Journal on Scientific Computing*, 2009.
- [15] Chandrajit Bajaj, Shun-Chuan Albert Chen, Guoliang Xu, Qin Zhang, and Wenqi Zhao. Hierarchical molecular interfaces and solvation electrostatics. *Proceedings of the 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*, 2009.
- [16] N. Baker, M. Holst, and F. Wang. Adaptive multilevel finite element solution of the Poisson-Boltzmann equation II: refinement at solvent accessible surfaces in biomolecular systems. *J. Comput Chem.*, 21:1343–1352, 2000.
- [17] Satish Balay, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.
- [18] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA, 1994.
- [19] N. Basdevant, D. Borgis, and T. Ha-Duong. A coarse-grained protein-protein potential derived from an all-atom force field. *J. Phys. Chem.*

- B*, 111:9390–9399, 2007.
- [20] R. Bharadwaj, A. Windemuth, S. Sridharan, B. Honig, and A. Nicholls. The fast multipole boundary element method for molecular electrostatics: An optimal approach for large systems. *J. Comput Chem.*, 16:898, 1995.
- [21] A. J. Bordner and G. A. Huber. Boundary element solution of the linear Poisson-Boltzmann equation and a multipole method for the rapid calculation of forces on macromolecules in solution. *J. Comput Chem.*, 24:353–367, 2003.
- [22] A. H. Boschitsch and M. O. Fenley. Hybrid boundary element and finite difference method for solving the nonlinear Poisson-Boltzmann equation. *J. Comput Chem.*, 25:935–955, 2003.
- [23] A. H. Boschitsch, M. O. Fenley, and Huan-Xiang Zhou. Fast boundary element method for the linear Poisson-Boltzmann equation. *J. Phys. Chem.*, 106:2741–2754, 2002.
- [24] W.R. Bowen and A.O. Sharif. Adaptive finite element solution of the nonlinear Poisson-Boltzmann equation: A charged spherical particle at various distances from a charge cylindrical pore in a charged planar surface. *Journal of Colloid and Interface Science*, 187:363–374, 1997.
- [25] N.-V. Buchete, J. Straub, and D. Thirumalai. Anisotropic coarse-grained statistical potentials improve the ability to identify natively protein structures. *J. Chem. Phys.*, 118:7658–7671, 2003.

- [26] N.-V. Buchete, J. Straub, and D. Thirumalai. Orientational potentials extracted from protein structures improve native fold recognition. *Protein Sci.*, 13:862–874, 2004.
- [27] T. Can, C. Chen, and Y. Wang. Efficient molecular surface generation using level-set methods. *Journal of Molecular Graphics and Modelling*, 25(4):442–454, 2006.
- [28] F. Cazals and F. Proust. On the topology and the geometry of (voronoi) molecular interfaces. part i: Algorithms. Research Report 5346, INRIA, 2004.
- [29] F. Cazals and F. Proust. Revisiting the description of protein-protein interfaces. part i: Algorithms. *Unite de recherche INRIA Sophia Antipolis*, (5346), 2005.
- [30] T. E. Cheatham and P. A. Kollman. Molecular dynamics simulation of nucleic acid. *Annu. Rev. Phys. Chem.*, 51:435–471, 2000.
- [31] M. L. Connolly. Analytical molecular surface calculation. *J Appl Cryst*, 16(5):548–558, 1983.
- [32] M. L. Connolly. Solvent- accessible surfaces of proteins and nucleic acids. *Science*, 221:709, 1983.
- [33] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz Jr, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. Kollman. A

- second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J. Am. Chem. Soc.*, 117:5179–5197, 1995.
- [34] C.M. Cortis and R.A. Friesner. An automatic three-dimensional finite element mesh generation system for the Poisson-Boltzmann equation. *J. Comput Chem.*, 18:1570–1590, 1997.
- [35] C.M. Cortis and R.A. Friesner. Numerical solution of the Poisson-Boltzmann equation using tetrahedral finite element methods. *J. Comput Chem.*, 18:1591–1608, 1997.
- [36] T. Dolinsky, J. Nielsen, A. McCammon, and N. Baker. Pdb2pqr: an automated pipeline for the setup of poissonboltzmann electrostatics calculations. *Nucleic Acids Research*, 32:665–667, 2004.
- [37] B. S. Duncan and A. J. Olson. Shape analysis of molecular surfaces. *Biopolymers*, 33:231–238, 1993.
- [38] H. Edelsbrunner. Deformable smooth surface design. *Discrete and computational geometry*, 21:87–115, 1999.
- [39] D. Eisenberg and A. D. McLachlan. Solvation energy in protein folding and binding. *Nature (London)*, 319:199–203, 1986.
- [40] M. K. Gilson, M. E. Davis, B. A. Luty, and J.A. McCammon. Computation of electrostatic forces on solvated molecules using the poisson-boltzmann equation. *J. Phys. Chem.*, 97:3591–600, 1993.

- [41] P. A. Golubkov and P. Ren. Generalized coarse-grained model based on point multipole and Gay-Berne potentials. *J. Chem. Phys.*, 125:064103, 2006.
- [42] J. Grant and B. Pickup. A Gaussian description of molecular shape. *Journal of Phys. Chem.*, 99:3503–3510, 1995.
- [43] David Halliday, Robert Resnick, and Jearl Walker. *Electric Potential*. John Wiley and Sons, fifth edition, 1997.
- [44] R. B. Hermann. Theory of hydrophobic bonding. ii. correlation of hydrocarbon solubility in water with solvent cavity surface area. *J. Phys. Chem.*, 76:2754–2759, 1972.
- [45] M. Holst. *Multilevel Methods for the Poisson-Boltzmann Equation*. Ph.d. thesis, University of Illinois at Urbana-Champaign, 1993.
- [46] M. Holst, N. Baker, and F. Wang. Adaptive multilevel finite element solution of the Poisson-Boltzmann equation I: algorithm and examples. *J. Comput Chem.*, 21:1319–1342, 2000.
- [47] W. Im, D. Beglov, and B. Roux. Continuum solvation model: computation of electrostatic forces from numerical solutions to Poisson-Boltzmann equation. *Computer Physics Communications*, 111:59–75, 1997.
- [48] A.H. Juffer, E.F.F. Botta, B.A.M. van Keulen, A. van der Ploeg, and H.J.C Berensen. The electric potential of a macromolecule in a solvent: a fundamental approach. *J Chemical Physics*, 97:144–171, 1991.

- [49] S. S. Kuo, M. D. Altman, J. P. Bardhan, B. Tidor, and J. K. White. Fast methods for simulation of biomolecule electrostatics. *IEEE/ACM international conference on Computer-aided design*, pages 466–473, 2002.
- [50] B. Lee and F. Richards. The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol.*, 55(3):379–400, 1971.
- [51] B. Lee and F.M. Richards. The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol.*, 55(3):379–400, 1971.
- [52] M. Levitt, M. Hirshberg, R. Sharon, and V. Daggett. Potential energy function and parameters for simulations of the molecular dynamics of proteins and nucleic acids in solution. *Comp. Phys. Comm.*, 91:215–231, 1995.
- [53] J. Liang, H. Edelsbrunner, P. Fu, and P. V. Sudhakar. Analytical shape computation of macromolecules: I. molecular area and volume through alpha shape. *Proteins: Structure, function, and genetics*, 33:1–17, 1998.
- [54] J. Liang, H. Edelsbrunner, P. Fu, P. V. Sudhakar, and S. Subramaniam. Analytical Shape Computation of Macromolecules: II. Inaccessible Cavities in Protein. *Proteins: Structure, Function, and Genetics*, 33:18–29, 1998.
- [55] J. Liang and S. Subramaniam. Computation of molecular electrostatics with boundary element methods. *Biophysical Journal*, 73:1830–1841, 1997.

- [56] B. Z. Lu, X. Cheng, J. Huang, and J. A. McCammon. Order  $n$  algorithm for computation of electrostatic interactions in biomolecular systems. *Proc. Natl. Acad. Sci.*, 103:19314–19319, 2006.
- [57] B. Z. Lu, D. Q. Zhang, and J. A. McCammon. Computation of electrostatic forces between solvated molecules determined by the Poisson-Boltzmann equation using a boundary element method. *J Chemical Physics*, 122(21):214102, 2005.
- [58] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Indust. Appl. Math.*, 11:431–441, 1963.
- [59] S. Marrink, A. de Vries, and A. Mark. Coarse grained model for semi-quantitative lipid simulations. *J. Phys. Chem. B*, 108:750–760, 2004.
- [60] A. V. Morozov, T. Kortemme, and D. Baker. Evaluation of models of electrostatic interactions in proteins. *J. Phys. Chem. B*, 107:2075–2090, 2003.
- [61] A. Nicholls and B. Honig. A rapid finite difference algorithm, utilizing successive over-relaxation to solve the Poisson-Boltzmann equation. *J. Comput Chem.*, 12:435–445, 1991.
- [62] T. J. Perun and C. L. Propst. *Computer-Aided Drug Design: Methods and Applications*. Informa Healthcare, 1989.
- [63] F. Richards. Area, volumes, packing and protein structures. *Ann. Rev. Biophys. Bioeng.*, 6:151–176, 1977.

- [64] M. Sanner, A. Olson, and J. Spehner. Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers*, 38:305–320, 1996.
- [65] A Sayyed-Ahmad, K. Tuncay, and P.J. Ortoleva. Efficient solution technique for solving the Poisson-Boltzmann equation. *J. Comput Chem.*, 25:1068–1074, 2004.
- [66] K. A. Sharp and B. Honig. Calculating total electrostatic energies with the nonlinear poisson-boltzmann equation. *J. Phys. Chem.*, 94:7684–7692, 1990.
- [67] K.A. Sharp and B. Honig. Applications of the finite difference Poisson-Boltzman method to proteins and nucleic acids. *Structure and Methods: DNA Protein Complexes and Proteins*, 2:211–214, 1990.
- [68] J. Shelley, M. Shelley, R. Reeder, S. Bandyopadhyay, and M. Klein. A coarse grain model for phospholipid simulations. *J. Phys. Chem. B*, 105:4464–4470, 2005.
- [69] J. Shelley, M. Shelley, R. Reeder, S. Bandyopadhyay, P. Moore, and M. Klein. Simulations of phospholipids using a coarse grain model. *J. Phys. Chem. B*, 105:9785–9792, 2001.
- [70] T. Simonson and A. T. Bruenger. Solvation free energies estimated from macroscopic continuum theory: An accuracy assessment. *J. Phys. Chem.*, 98:4683 – 4694, 1994.

- [71] B. Smit, P. Hilbers, K. Esselink, L. Rupert, N. van Os, and A. Schlijper. Computer simulations of a water/oil interface in the presence of micelles. *Nature*, 348:624–625, 1990.
- [72] A. Smith and C. Hall.  $\alpha$ -Helix formation: Discontinuous molecular dynamics on an intermediate-resolution protein model. *Proteins: Structure, function, and genetics*, 44:344–360, 2001.
- [73] W. C. Still, A. Tempczyk, R. C. Hawley, and T. Hendrickson. Semianalytical treatment of solvation for molecular mechanics and dynamics. *J. Am. Chem. Soc.*, 112:6127–6129, 1990.
- [74] Y. Ueeda, H. Taketomi, and N. Gō. Studies on protein folding, unfolding and fluctuations by computer simulation. a three-dimensional lattice model of lysozyme. *Biopolymers*, 17:1531–1548, 1978.
- [75] Y. N. Vorobjev, J. A. Grant, and H. A. Scheraga. A combined iterative and boundary element approach for solution of the nonlinear Poisson-Boltzmann equation. *J. Am. Chem. Soc.*, 114:3189–3196, 1992.
- [76] Y. N. Vorobjev and H. A. Scheraga. A fast adaptive multigrid boundary element method for macromolecular electrostatic computations in a solvent. *J. Comput Chem.*, 18:569–583, 1996.
- [77] J. Wagoner and N. A. Baker. Solvation forces on biomolecular structures: A comparison of explicit solvent and poisson-boltzmann models. *J. Comput Chem.*, 25:1623–1629, 2004.

- [78] L. Ying, G. Biros, and D. Zorin. A kernel-independent adaptive fast multipole method in two and three dimensions. *Journal of Computational Physics*, 196(2):591–626, 2004.
- [79] R.J. Zauhar and R.S. Morgan. A new method for computing the macromolecular electric potential. *J. Mol. Biol.*, 186:815–820, 1985.
- [80] Yongjie Zhang, Guoliang Xu, and Chandrajit Bajaj. Quality meshing of implicit solvation models of biomolecular structures. *Computer Aided Geometric Design*, 23:510–530, 2006.
- [81] W. Zhao, G. Xu, and C. Bajaj. An algebraic spline model of molecular surfaces for energetic computations. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2009.
- [82] H. Zhou. Boundary element solution of macromolecular electrostatics: interaction energy between two proteins. *Biophys. J.*, 65:955–963, 1993.

# Vita

Shun-Chuan Chen was born on December 2, 1979 in Okayama, Japan to parents Neng-Chin Chen and Shu-Hui Lan. He moved back to Taiwan at the age of 4. After graduating from Taipei Private Yan Ping High School in 1998, he attended National Taiwan University, where he received his B. S. and M. S. degrees in Computer Science and Information Engineering in June 2002 and June 2004, respectively. In August 2004, he joined the doctoral program of Computer Sciences at the University of Texas at Austin.

Permanent address: 3517 North Hills Dr. Apt Y303  
Austin, Texas 78731

This dissertation was typeset with L<sup>A</sup>T<sub>E</sub>X<sup>†</sup> by the author.

---

<sup>†</sup>L<sup>A</sup>T<sub>E</sub>X is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's T<sub>E</sub>X Program.