UNIVERSITY OF CALIFORNIA,
IRVINE

Multi-Resolution Models for Biomolecular Application

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Biomedical Engineering

by

Li Xiao

Dissertation Committee:
Professor Ray Luo, Chair
Professor Abraham Lee
Professor Hongkai Zhao
DEDICATION

To

my parents and friends

in recognition of their worth

an apology

A feeling bears on itself the scars of its birth; it recollects as a subjective emotion its struggle for existence; it retains the impression of what might have been, but is not.

Li Xiao
Process and Reality

and hope

If the fool would persist in his folly he would become wise.

William Blake
“Proverbs of Hell”
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>vii</td>
</tr>
<tr>
<td>CURRICULUM VITAE</td>
<td>viii</td>
</tr>
<tr>
<td>ABSTRACT OF THE DISSERTATION</td>
<td>x</td>
</tr>
<tr>
<td>CHAPTER 1: Introduction</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>23</td>
</tr>
<tr>
<td>CHAPTER 2: Electrostatic forces in the Poisson-Boltzmann systems</td>
<td>29</td>
</tr>
<tr>
<td>References</td>
<td>63</td>
</tr>
<tr>
<td>CHAPTER 3: Charge Central Interpretation of the Full Nonlinear PB Equation: Implications for Accurate and Scalable Modeling of Solvation Interactions</td>
<td>69</td>
</tr>
<tr>
<td>References</td>
<td>113</td>
</tr>
<tr>
<td>CHAPTER 4: A Continuum Poisson-Boltzmann Model for Membrane Channel Proteins</td>
<td>119</td>
</tr>
<tr>
<td>References</td>
<td>146</td>
</tr>
<tr>
<td>References</td>
<td>176</td>
</tr>
<tr>
<td>Appendix: Supplemental Material</td>
<td>179</td>
</tr>
<tr>
<td>CHAPTER 6: Exploring a Multi-Scale Method for Dynamics Simulations in Continuum Solvent Model: Explicit Simulation of Continuum Solvent as an Incompressible Fluid</td>
<td>211</td>
</tr>
<tr>
<td>References</td>
<td>239</td>
</tr>
<tr>
<td>CHAPTER 7: Conclusions and Future Directions</td>
<td>245</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2-1</td>
<td>Deformation of the Disk Volume Element</td>
</tr>
<tr>
<td>2-2</td>
<td>The Force on a Small Volume</td>
</tr>
<tr>
<td>2-3</td>
<td>Singular Charge Enclosed by an Infinitely Small Spherical Volume</td>
</tr>
<tr>
<td>2-4</td>
<td>Decomposition of Electric Field on the Surface</td>
</tr>
<tr>
<td>2-5</td>
<td>Correlations Between the Atomic Dielectric Boundary Forces</td>
</tr>
<tr>
<td>3-1</td>
<td>Correlations of Grid Potentials</td>
</tr>
<tr>
<td>3-2</td>
<td>Correlations of Electrostatic Energies</td>
</tr>
<tr>
<td>3-3</td>
<td>Convergence of the Reaction Field Energies</td>
</tr>
<tr>
<td>3-4</td>
<td>Convergence of Salt-Related Energy Term</td>
</tr>
<tr>
<td>3-5</td>
<td>Correlations between full PBE Electrostatic Energies</td>
</tr>
<tr>
<td>3-6</td>
<td>CPU time for the pair wise charge view method and P3M method</td>
</tr>
<tr>
<td>3-7</td>
<td>Self-convergence of forces with charges mapped onto the surface</td>
</tr>
<tr>
<td>3-8</td>
<td>Consistency of forces with charges mapped onto the surface</td>
</tr>
<tr>
<td>3-9</td>
<td>Self-convergence of forces without charges mapped onto the surface</td>
</tr>
<tr>
<td>3-10</td>
<td>Consistency of forces without charges mapped onto the surface</td>
</tr>
<tr>
<td>4-1</td>
<td>Sequence alignment of KcsA and hERG by ClustalX version 2.1</td>
</tr>
<tr>
<td>4-2</td>
<td>Comparison of target and parent structures of hEGH and KcsA</td>
</tr>
<tr>
<td>4-3</td>
<td>Solvent-solute interface of the membrane model</td>
</tr>
<tr>
<td>4-4</td>
<td>Solvent-solute interface without turning on the depth-first search</td>
</tr>
<tr>
<td>4-5</td>
<td>Discrepancy between implicit and explicit water simulations</td>
</tr>
<tr>
<td>4-6</td>
<td>MMPBSA binding affinities compared with experimental measurements</td>
</tr>
<tr>
<td>Figure 5-1</td>
<td>Order Analysis of Max Error</td>
</tr>
<tr>
<td>Figure 5-2</td>
<td>Single Bubble Dynamics</td>
</tr>
<tr>
<td>Figure 5-3</td>
<td>Two Bubble Merging Dynamics</td>
</tr>
<tr>
<td>Figure 5-4</td>
<td>Three Bubble Merging Dynamics</td>
</tr>
<tr>
<td>Figure 5-5</td>
<td>Parallel Scaling of Three-Bubble Merging Dynamics</td>
</tr>
<tr>
<td>Figure 6-1</td>
<td>Volume Evolution and spherical contour of sodium ion</td>
</tr>
<tr>
<td>Figure 6-2</td>
<td>Volume Evolution and SES-like solute-solvent interface of NMA</td>
</tr>
<tr>
<td>Figure 6-3</td>
<td>Volume Evolution of Four Dimers</td>
</tr>
<tr>
<td>Figure 6-4</td>
<td>Time Evolution of Average Absolute Pressure Components</td>
</tr>
<tr>
<td>Figure 6-5</td>
<td>Outside View of FD/MD Surface and Surface from explicit MD</td>
</tr>
<tr>
<td>Figure 6-6</td>
<td>Inside View of FD/MD Surface and Surface from explicit MD</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1</td>
<td>Trend lines for the full PBE electrostatic solvation energy</td>
<td>94</td>
</tr>
<tr>
<td>3-2</td>
<td>Relative deviations in P3M reaction field forces with different cutoffs</td>
<td>98</td>
</tr>
<tr>
<td>3-3</td>
<td>Relative deviations in forces with charges mapped onto the surface</td>
<td>102</td>
</tr>
<tr>
<td>3-4</td>
<td>Relative deviations in forces without charges mapped onto the surface</td>
<td>105</td>
</tr>
<tr>
<td>3-5</td>
<td>Averaged net force for nucleic acid 420D and protein 1F81</td>
<td>106</td>
</tr>
<tr>
<td>4-1</td>
<td>Different edges of dielectric constants</td>
<td>129</td>
</tr>
<tr>
<td>4-2</td>
<td>The thickness of membrane and mprob thresholds</td>
<td>137</td>
</tr>
<tr>
<td>4-3</td>
<td>Discrepancies in the solvent accessible region</td>
<td>141</td>
</tr>
<tr>
<td>4-4</td>
<td>MMPBSA binding affinities in comparison with experiment.</td>
<td>143</td>
</tr>
<tr>
<td>4-5</td>
<td>Averaged CPU times in setting up MMPBSA</td>
<td>145</td>
</tr>
<tr>
<td>5-1</td>
<td>Average CPU Time per Step</td>
<td>175</td>
</tr>
<tr>
<td>6-1</td>
<td>Conversion factors between FD and MD engines</td>
<td>226</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

I wish to express my deepest thanks to my PhD advisor, Ray Luo, for being a passionate and understanding mentor and providing the highest quality instruction, guidance, encouragement and support. The research environment provided by Professor Luo encourages our development in relatively broad scientific skill-sets, in-depth thinking, and strong intra-group collaboration as well as with scientists outside the university. I would not have had such a rich graduate experience without him as a caring mentor and friend.

I would like to thank my dissertation committee, Prof Abraham Lee, Prof Hongkai Zhao, as well as my advancement committee members, Prof Frithjof Krujgel and Prof Hung D. Nguyen for providing useful advice to help my projects move along. I must also thank the entire labs of Sheryl Tsai for useful discussion in several interesting projects. In addition, I must thank Professor Zhilin Li, of North Carolina State University, for his counsel and his groups very close collaboration with me, which helped me to accomplish some fundamental mathematical framework in Chapter 5 and Chapter 6 of the thesis.

I would like to thank my fellow labmates through the years. First, I must thank Dr. Qin Cai for helping me quickly pick up the knowledge and skills needed in my research and as coauthor of Chapter 2 and 5 of this thesis as well as being a great friend and mentor to me. I must also thank Dr. Wesley Smith helping me quickly learned how to use Amber and Linux. I must thank JianXiong Diao, who has contributed to Chapter 4 of the thesis using his expertise in homology modeling and docking. I must also thank Ruxi Qi, Changhao Wang, Andrew Schaub, D’Artagnan Greene, Eleftherios Lambros and Vy Duong for being excellent coworkers and friends during the last few years.

Financial support was provided by the University of California, Irvine, NIH Grant GM093040 & GM079383.
CURRICULUM VITAE

Li Xiao
Email: lxiao4@uci.edu

EDUCATIONAL BACKGROUND

Bachelor of Science, Tsinghua University 2006-2010

Master of Arts in Psychology at University of Minnesota 2010-2012

Doctoral in Biomedical Engineering at the University of California, Irvine 2012-2017

RESEARCH EXPERIENCE

Computational biophysics at the University of California, Irvine 2012-2017
- Development of a multi-scale strategy based on the Navier-Stokes equation for biomolecular simulation in implicit solvent
- Theoretical analysis of electrostatic forces in the nonlinear Poisson-Boltzmann systems
- Development of a particle-particle particle mesh method to compute full electrostatic interactions for nonlinear Poisson-Boltzmann systems
- Development of an implicit membrane model for membrane channels

Cognitive modeling and simulation at University of Minnesota 2010-2012
- Modeling of auditory signal process and development of a pitch recognition model
- Development of a statistical mix-effect model with extracted data from human subjects

PUBLICATIONS

Peer-Reviewed Articles


**Software Packages**

Amber and AmberTools, 2016, University of California, San Francisco.

**PRESENTATIONS**

Li Xiao, A Charge Central Interpretation of the Full Nonlinear PB Equation, AMBER Meeting, March 18, 2016

Li Xiao, Multi-resolution model for biomolecular application, ACS Poster, March 13, 2016

Li Xiao, Multi-scale Simulation of Biomolecular System, Brainstorm Workshop on Biological Molecules, Cells, and Beyond, August 5, 2014, UCSD

**HONORS AND AWARDS**

National Excellent Self-financed Student Scholarship by China Government, 2016

Justine Lambert Prize for graduate paper competition, University of California, Irvine, 2015

Top 10 in the US, IEEExreme 8.0 worldwide programming competition, 2014

Silver Medal, final round of the National High School Physics Competition, China, 2004, 2005

Admission to the Gifted Young Student Program, Tsinghua University, Beijing, China, 2004
ABSTRACT OF THE DISSERTATION

Multi-Resolution Models for Biomolecular Application

By

Li Xiao

Doctor of Philosophy in Biomedical Engineering

University of California, Irvine, 2017

Professor Ray Luo, Chair

Atomistic simulation has become an important tool for studying the structures, dynamics, and functions of biomolecular systems. Nevertheless efficient atomistic simulation of large and complex biomolecular systems is still one of the remaining challenges in computational molecular biology. The computational challenges in atomistic simulation of biomolecular systems are direct consequences of their high dimensionalities. Indeed biomolecules are highly complex molecular machines with thousands to millions of atoms. What further complicates the picture is the need to realistically treat the interactions between biomolecules and their surrounding water molecules that are ubiquitous and paramount important for their structures, dynamics, and functions.

Since most particles in biomolecular simulations are to represent water molecules solvating the target biomolecules, an implicit treatment of water molecules allows greatly increased simulation efficiency. Indeed, implicit solvation offers a unique opportunity for more efficient simulations without the loss of atomic-level resolution for biomolecules. Advance in implicit solvation, coupled with developments in sampling algorithms, classical force fields, and quantum
approximations, will prove useful to the larger biomedical community in a broad range of studies of biomolecular structures, dynamics and functions.

The objective of the dissertation is to develop multi-scale models for biomolecule simulation with implicit treatment of water. Specifically: 1) a math framework to compute electrostatic forces in implicit solvent model is derived; 2) a charge central interpolation method is implemented to efficiently compute electrostatic forces with high accuracy; 3) a multi-scale model with automatic pore regions detection is applied on membrane system to compute solvation energy; 4) a physical model of multi-scale model is proposed and a fluid dynamics algorithm solver is developed; and 5) the fluid dynamics algorithm is incorporated with the Amber molecular mechanics simulation engine to conduct atomistic simulations of biomolecules in the continuum solvent models.
CHAPTER 1

INTRODUCTION

1.1 Background

Solvation interaction is one of the essential determinants of the structures and functions in proteins and nucleic acids, and is crucial in their accurate modeling [1-15]. Due to the high computational overhead in the explicit treatments of solvent molecules, continuum models of solvation interactions, specifically the electrostatic solvation modeling that is based upon the Poisson-Boltzmann equation (PBE), are widely used in the studies of biomolecules [1-15]. Indeed continuum electrostatics modeling represents a physically effective approach that makes it possible to account for a number of phenomena involving solvent electrostatic effects on the functional analyses of biomolecules [1-15]. These are facilitated by numerical solutions of the PBE that can be obtained routinely for biomolecules and their complexes [3, 16-21]. Among the numerical solution methods, finite-difference method [22-34], finite-element method [35-43] and boundary-element method [44-59] are mostly used.

A disadvantage of the numerical continuum electrostatics methods is that it is conceptually difficult to incorporate them into molecular mechanics programs, mainly because of the problem of assigning forces related to the dielectric boundary (“dielectric stress”) to individual atoms [30, 35, 60-68]. Other problems include the convergence of the energy and forces with respect to the resolution of the solute-solvent boundary and charge representation [69-71]. Thus in most practical applications, the PBE is only solved for a few fixed conformations of a solute or solute complex. This limits the application of PBE in more elaborative modeling of biomolecules. Many efforts are invested to develop methods to compute electrostatic forces [30, 35, 49, 60-68, 72-79].
This review focuses on published efforts to adapt numerical PBE methods to molecular mechanics simulations, i.e. developmental efforts to interpolate solvation forces. The first to follow is the fundamental works to define the electrostatic free energy and Maxwell stressor tensor for systems modeled by PBE. Given these theoretical preparations, three different strategies are presented to derive algorithms suitable for numerical PBE solutions, including the variational approach, the stress tensor approach, and the shape derivative approach. Different strategies are also reviewed to improve the accuracy and convergence of the dielectric boundary force interpolation, a notoriously difficult problem. We end this review with a brief overview of different numerical procedures on both reaction field and dielectric boundary force interpolations.

1.2 Electrostatic free energy for the Poisson-Boltzmann systems

The PBE is based on the more fundamental equation in electrostatics, the Poisson equation is

\[ \nabla \cdot (\varepsilon(\mathbf{r}) \nabla \varphi(\mathbf{r})) + 4\pi \rho(\mathbf{r}) = 0, \]

where \( \varphi \) is the potential, \( \rho \) is the charge density, \( \varepsilon \) is the dielectric constant, and \( \mathbf{r} \) is the position vector.

In 1990, Sharp and Honig [80] proposed a method to define the electrostatic free energy based on eqn (1) for PBE systems. In their method, it was assumed that the amount of the fixed charge is constant. They also required that the chemical free energy of each type of ion is uniform throughout the solution at equilibrium.
\[
\mu_i(\mathbf{r}) = \mu_i^\circ + k_B T \ln a_i(\mathbf{r}) + z_i e \varphi(\mathbf{r}) = \mu_i^\circ + k_B T \ln a_i^b,
\]

where \(\mu_i^\circ\) is the standard-state chemical potential, \(e\) is the proton charge, \(k_B\) is the Boltzmann constant, \(T\) is the absolute temperature, \(z_i\) is the valence of the ion, \(a_i\) is the activity, and \(a_i^b\) is the activity of the ion in the bulk solution, i.e., far from the colloid where the potential is zero. Notice here that the activity satisfies the Boltzmann expression

\[
a_i(\mathbf{r}) / a_i^b = \exp(-\frac{z_i e \varphi(\mathbf{r})}{kT}).
\]

Given the above distribution and also the fact that \(a_i(\mathbf{r})\) is proportional to the ion concentration \(c_i(\mathbf{r})\), they expressed \(\rho(\mathbf{r})\) as a function of \(\varphi\) and \(c_i(\mathbf{r})\). Eqn (1) can then be rewritten as

\[
\nabla \cdot (\varepsilon(\mathbf{r}) \nabla \varphi(\mathbf{r})) + f(\varphi(\mathbf{r})) = 0,
\]

where \(f(\varphi(\mathbf{r})) = 4\pi \sum_i z_i e c_i \exp(-\frac{z_i e \varphi(\mathbf{r})}{kB T}) + 4\pi \rho^f(\mathbf{r})\) with \(c_i\) being the bulk concentration of ion type \(i\).

Following the Euler-Lagrange scheme [81], they designated a function \(F(x,y,z,\varphi,\varphi_x,\varphi_y,\varphi_z)\) to represent the free energy density, so that eqn (4) can be shown to satisfy

\[
\frac{\partial F}{\partial \varphi} - \left[ \frac{\partial (\partial F / \partial \varphi_x)}{\partial x} + \frac{\partial (\partial F / \partial \varphi_y)}{\partial y} + \frac{\partial (\partial F / \partial \varphi_z)}{\partial z} \right] = 0
\]

and
The free energy density can be obtained by solving eqn (5) and eqn (6) as

$$ G = \int F \, dv = \int (\rho^f \varphi - \Delta \Pi - \mathbf{E} \cdot \mathbf{D} / 2) \, dv, $$

(7)

where \( \Delta \Pi = k_B T \sum_i c_i (e^{-q_i \varphi / k_B T} - 1) \) is the excess osmotic pressure of the mobile ion concentration, \( \mathbf{E} \) is the electric field, \( \mathbf{D} \) is the electric displacement vector, and \( \rho^f \) is the fixed charge density.

Almost at the same time, Reiner and Radke [82] proposed another strategy to derive the electrostatic free energy for PBE systems using the variational principle. They started from the same fundamental equation, eqn (1). Given that the ion density satisfies

$$ \nabla \cdot (\varepsilon(r) \nabla \varphi(r)) + 4\pi \sum_{i=1}^{N} z_i e c_i \exp(-z_i e \varphi(r) / k_B T) = 0, $$

(8)

They defined a potential \( \varphi(r) \) dependent function \( \Gamma \) in a domain of interest subject to \( \Gamma[\varphi(r), r] = f(r) \), with a boundary condition \( \Gamma_s[\varphi(r^s), r] = f_s(r^s) \). It was shown that a unique solution \( \varphi^*(r) \) can be specified given this condition [83]. After obtaining \( \varphi^*(r) \) subject to the boundary condition, it is possible to identify a non-linear scalar function \( I \) of the potential and its gradient that satisfies a local stationary condition

$$ \delta I(\varphi, \nabla \varphi, r) / \delta \varphi(r) = 0; \varphi(r) = \varphi^*(r). $$

(9)

By setting \( f(r) = 0 \) and function \( \Gamma \) as the left hand side of the PBE, i.e.
\[ \Gamma[\varphi(r), r] = \nabla \cdot (\varepsilon \nabla \varphi(r)) + \sum_{i}^{N} z_i e_c \exp\left(-\frac{z_i e \varphi(r)}{k_B T}\right), \]  

(10)

yielded the expression for the non-linear scalar function, I, which can be used to define
the PBE electrostatic free energy. Their definition is consistent with that of Sharp and Honig as
shown in eqn (7).

### 1.3 Maxwell stress tensor for the Poisson-Boltzmann systems

In classical electrostatics, electrostatic forces can be computed via the Maxwell stress
tensor [84]. The Maxwell stress tensor can be written in the form of a rank two tensor

\[
P = \begin{bmatrix}
T_{xx} & T_{xy} & T_{xz} \\
T_{yx} & T_{yy} & T_{yz} \\
T_{zx} & T_{zy} & T_{zz}
\end{bmatrix}.
\]  

(11)

Considering a surface with area \( A \) and normal vector \( n \), the definition of stress tensor
shows that \( F = \int P \cdot n dA \), and \(-P \cdot n = -(T_{ij} n_j, T_{ij} n_j, T_{ij} n_j)^T\) is simply the pressure acted upon
the upper surface. Here Einstein’s summation convention is used, i.e. \( T_{ij} n_j = \sum_j T_{ij} n_j \).

Laudau and Lifshitz [84] proposed a method in deriving the Maxwell stress tensor based
on the electrostatic free energy. In their model, they considered a thin and small disk of dielectric
with small area \( A \) and thickness \( h \), and with \( h \ll \sqrt{A} \). Given the small dimension of the volume
element, they assumed its uniform composition, density, temperature, and electrostatic properties
such as $\varepsilon$ and $\lambda$. They also assumed that charge density $\rho'$, potential $\varphi$ and electric field $E$ are all smooth variables.

Considering the upper surface of the disk with area $A$ and normal vector $n$, the definition of stress tensor shows that $F = \int P \cdot n dA$. Thus $-P \cdot n = -(T_{ij} n_j, T_{ij} n_j, T_{ij} n_j)^T$ is simply the pressure acting upon the upper surface, where $T_{ij} n_j = \sum_j T_{ij} n_j$. The goal in the derivation is to impose a virtual deformation of the thin disk along a small virtual displacement $s$ ($s \ll h$) whose direction is arbitrary to compute the associated pressure via the virtual displacement method. By computing the variation of free energy before and after deformation, using the principle $-\delta G = F \cdot s A$, they obtained the stress tensor in the form as

$$T_{ij} = \frac{\varepsilon E_i E_j}{4\pi} - \rho \frac{\partial g}{\partial \varphi} \delta_{ij} + g \delta_{ij}, \quad (12)$$

where $g$ is the free energy density, $\rho$ is the charge density, $E_i$ and $D_j$ are the components of the electric field and the electric displacement, respectively.

Recently following the Landaau and Lifshitz framework [84], Li et al. [85] presented a theoretical study to derive the Maxwell stress tensor for the full nonlinear PBE system. Different from the Poisson system, the electrostatic free energy in the PBE system nonlinearly depends on the potential. In the Laudau and Lifshitz framework, the variation of the potential is in linear relationship with the displacement, hence there is a uniform variation of electric field. However, in the nonlinear PBE system, the PBE (eqn (1)) is not preserved if the variation of electric field $E$ is the same as that in a Poisson system. To overcome the difficulty, they added a correction term on $E$ to satisfy the PBE, and then proved that this correction term is infinitely small.
comparing with the absolute variation of $E$. They also proved that the variations of energy and charge density are the dominant terms, consistent with the Laudau and Lifshitz framework. Therefore only variations of energy and charge density need to be considered to compute the variation of electrostatic free energy. Finally the Maxwell stress tensor is

$$ T_{ij} = \frac{1}{4\pi} \varepsilon E_i E_j - \frac{1}{8\pi} \varepsilon E^2 \delta_{ij} - \Delta \Pi \lambda \delta_{ij}, \quad (13) $$

where $\Delta \Pi = k_b T \sum_i c_i (e^{-q_i \phi / k_b T} - 1)$. Note also that the tensor defined is symmetric, i.e. $T_{ij} = T_{ji}$.

1.4 Derivation of forces for the Poisson-Boltzmann systems

1.4.1 Free energy variation approach

Gilson et. al. presented a variational approach to derive the total electrostatic force [64]. The starting point of their approach is the electrostatic free energy in eqn (7)

$$ G(\phi, \rho^f, \varepsilon, \lambda) = \int_V (\rho^f \phi - \frac{1}{2} \varepsilon \nabla \phi \cdot \nabla \phi - k_b T \sum_i^N [c_i (e^{-q_i \phi / k_b T} - 1)] \lambda) dV. \quad (14) $$

Variation of the free energy can be written in terms of the variations of $\rho^f, \varepsilon, \phi, \lambda$ as

$$ \delta G = \int_V (\phi \delta \rho^f - \frac{1}{2} \nabla \phi \cdot \nabla \delta \varepsilon - k_b T \sum_i^N [c_i (e^{-q_i \phi / k_b T} - 1)] \delta \lambda + $$

$$ \rho^f \delta \phi - \varepsilon \nabla \phi \cdot \nabla (\delta \phi) + \lambda \sum_i^N q_i c_i e^{-q_i \phi / k_b T} \delta \phi) dV. \quad (15) $$

Assuming that the system is very large so that all variables vanish at the system boundary and also that $\varepsilon$ and $\lambda$ only depend on position, which do not change upon the variation of $\rho^f$ and
\( \mathbf{E} \), the variation of free energy can be expressed as

\[
-\delta G = \int_V \mathbf{s} \cdot \{ \rho \mathbf{f} \mathbf{E} - \frac{1}{2} E^2 \nabla \phi - k_B T \sum_i [(e^{-q_i \phi/k_B T} - 1) c_i \nabla \lambda] \} dV,
\]

(16)

where \( \mathbf{s} = \mathbf{s}(\mathbf{r}) \) is an arbitrary infinitesimal displacement of the system. Expressing the variation of free energy in terms of the virtual work principle leads to the total force as

\[
-\delta G = \int_V \mathbf{f}(\mathbf{r}) \cdot \mathbf{s}(\mathbf{r}) dV. 
\]

(17)

The force density is finally obtained as

\[
\mathbf{f} = \rho \mathbf{f} \mathbf{E} - \frac{1}{2} E^2 \nabla \phi - \frac{1}{2} k_B T \sum_i [c_i (e^{-q_i \phi/k_B T} - 1)] \nabla \lambda.
\]

(18)

In a separate effort, Che et. al. presented an alternative derivation of dielectric boundary force for the classical two-dielectric model via the variational principle [65]. Given the assumption that the normal surface field contributes predominantly in the dielectric boundary force (DBF), they showed that the DBF could be formulated as

\[
\mathbf{f}_{\text{DBF}} = -\frac{1}{8\pi} \left( \frac{1}{\varepsilon_i} - \frac{1}{\varepsilon_o} \right) |\mathbf{D}_n|^2 \mathbf{n},
\]

(19)

where \( \mathbf{D}_n \) represents the continuous normal dielectric displacement vector on the solute/solvent dielectric interface.

### 1.4.2 Stress tensor approach

Li et. al.’s recent work showed an alternative way to compute the electrostatic forces based on the stress tensor [85]. After obtaining the Maxwell stress tensor \( \mathbf{P} \) in the form of eqn
(13), the force can be computed as $\mathbf{P} \cdot \mathbf{n} da$ for any area element $da$ with a normal direction of $\mathbf{n}$.

Considering an ideal classical fluid, the natural first step is the computation of force density $\mathbf{f} dv$ for a volume element $dv$. The net force felt by the volume element is the total force acting upon it on its enclosing surface. This procedure suggests that the concept of divergence of the stress tensor can be introduced, which is $\lim_{dv \to 0} \oint \mathbf{P} \cdot \mathbf{n} da$, as is the case for the divergence of a vector field. It is straightforward to show that the divergence theorem also holds for the stress tensor [84]

$$\int \nabla \cdot \mathbf{P} dv = \oint \mathbf{P} \cdot \mathbf{n} da.$$  \hspace{1cm} (20)

Two consequences in eqn (20) concern us here: (1) the force density is simply the divergence of the stress tensor, i.e.

$$\mathbf{f} = \nabla \cdot \mathbf{P} = \rho' \mathbf{E} - \frac{1}{8\pi} \mathbf{E}^2 \nabla \varepsilon - \Delta \Pi \nabla \lambda;$$  \hspace{1cm} (21)

and (2) total force can not only be computed by the differential form, i.e. through the computation of force density (eqn (21)) followed by volume integration, but also by the integral form, i.e. through the surface integral of the stress tensor, and they are consistent according to eqn (20), i.e. when the divergence theorem holds. Since the divergence is undefined in regions with discontinuity, the differential form can only be used in regions without singularity, i.e. without jump of dielectric constant and without singular charge sources (for example the point charges represented as delta functions in the PBE). However, for systems/regions with singularity, only the integral form can be applied. In contrast the variational approach shown above is limited because it only holds in regions without singularity [64].
Based on the integral approach, Li et. al. [85] presented the derivation of total electrostatic forces throughout a system with singularities. For a solute volume element with singularity, they divided the volume element into a smaller spherical region containing the singular charge and the rest where there is no singular charge. And then they obtained the analytical expression for the electrostatic force in the small solute volume element based on the Maxwell stress tensor. The force density can then be universally written as

\[ f = \rho E \] (22)

in the solute region whether the charge density is singular or not.

The computation of the ionic boundary force (IDF) at the stern Layer can also be addressed using the integral approach. At the Stern layer, mobile ions exist only on one side though the dielectric constants on both sides are the same. The classical method models the Stern layer with a step function of \( \lambda \) that changes from 0 to 1. Given the availability of Maxwell stress tension, the ionic boundary force (IDF) caused by the excess osmotic pressure is simply the difference in pressure, expressed as

\[ f_{IDF} = (P_o - P_i) \cdot n = -\nabla \Pi, \] (23)

where \( P_o, P_i \) are the corresponding stress tensors on outside and inside of the Stern layer from eqn (21).

The integral approach was also applied in the computation of the dielectric boundary force (DBF) for the piece-wise constant classical dielectric model of biomolecules by Cai et. al. [68]. Their derivation shows
\[ \mathbf{f}_{DBF} = (\mathbf{P}_o - \mathbf{P}_i) \cdot \mathbf{n} = \frac{1}{4\pi} \left[ \left( \varepsilon_o E_{on}^2 - \frac{1}{2} \varepsilon_o \mathbf{E}_o^2 \right) - \left( \varepsilon_i E_{in}^2 - \frac{1}{2} \varepsilon_i \mathbf{E}_i^2 \right) \right] \mathbf{n}, \tag{24} \]

where \( \mathbf{n} \) is the outward-directed normal unit vector of the molecular surface, and \( \mathbf{P}_i \) and \( \mathbf{P}_o \) are the stress tensors on the surfaces parallel to the dielectric interface inside and outside of the solute. \( \varepsilon_o \) and \( \varepsilon_i \) are the dielectric constants of the solvent and the solute, respectively, \( \mathbf{E}_o, \mathbf{E}_i \) are the electric fields on the two sides of the solute, respectively, and \( E_{in}, E_{on} \), are the electric field components on the \( \mathbf{n} \) direction of \( \mathbf{E}_o, \mathbf{E}_i \), respectively. Interestingly, eqn (24) is consistent with a much older expression proposed by Davis and McCammon [60]. They pointed out without proof that the DBF surface density can be written as

\[ \mathbf{f}_{DBF} = -\frac{1}{8\pi} \left( \varepsilon_o - \varepsilon_i \right) \left( \mathbf{E}_o \cdot \mathbf{E}_i \right) \mathbf{n}. \tag{25} \]

### 1.4.3 Shape derivative approach

Starting from a completely different direction, Li and co-workers [67] presented a shape-derivative approach to derive the DBF. For a given solvation system with a fixed charge density and dielectric coefficient, any given possible dielectric boundary \( \Gamma \) determines the electrostatic potential \( \psi_\Gamma \) as the unique solution to the nonlinear PBE, which in turn determines the electrostatic free energy \( G[\Gamma] \) as in eqn (7). Using the notion and method of shape derivatives, they gave a precise definition of the normal component of the DBF as

\[ \delta_{\Gamma,y} G[\Gamma] = \frac{d}{dt} G[\Gamma_t(V)] \big|_{t=0}, \tag{26} \]

where \( \Gamma_t(V) = \{ x(t,X) : X \in \Gamma \} \), \( x(t,X) \) is the solution map of the dynamical system. Following this framework, Li and coworkers obtained the DBF as [67]
\[
\mathbf{f}_{DBF} = \frac{1}{4\pi} \left[ \varepsilon_o |\nabla \varphi_o \cdot \mathbf{n}|^2 - \frac{1}{2} \varepsilon_o |\nabla \varphi_o|^2 - \varepsilon_i |\nabla \varphi_i \cdot \mathbf{n}|^2 + \frac{1}{2} \varepsilon_i |\nabla \varphi_i|^2 \right].
\] (27)

Apparently eqn (27) is equivalent to eqn (24) and also to eqn (25).

### 1.5 Efforts in improving the dielectric boundary force calculations

Further improvements are possible to enhance the numerical convergence and stability of the particularly difficult calculation of dielectric boundary force. For DBF formulations derived from the differential/variational approaches, Cai et. al. [66] proposed a strategy to exploit the concept of boundary polarization charge in the DBF calculation as

\[
f_{DBF} = \frac{1}{2} \rho_{pol} \left| \mathbf{D} \cdot \mathbf{n} \right|^2,
\] (28)

where \( \rho_{pol} \) is the boundary polarization charge density, \( \mathbf{D} \) is the electric displacement vector, and \( D_n \) is the normal component of \( \mathbf{D} \). This “charge-based” method works for dielectric models that are smooth-transitioned, i.e., when differential or variational approaches were utilized in the fundamental DBF formulations. Their test results demonstrate high level of agreement between the new formulation and the analytical method as well as the virtual work principle. Later, Cai et. al. [68] used the integral approach to obtain the dielectric boundary force as eqn (24). They further pointed out that a charge-based strategy could also be proposed for the piece-wise constant dielectric treatment. Briefly the dielectric boundary force can be written as

\[
f_{DBF} = \frac{1}{2} \sigma_{pol} \frac{\varepsilon_i E_i \cdot E_o}{E_{on}} \mathbf{n} = \frac{1}{2} \sigma_{pol} \frac{\mathbf{D}_i \cdot \mathbf{D}_o}{D_{on}} \mathbf{n},
\] (29)
where $\sigma_{pol}$ is the bound polarized surface charge density, and $D_o$ and $D_i$ are the corresponding electric displacements of the solvent and solute side, respectively. An interesting observation is the similarity of eqn (29) and the charge-based approach for the smooth-transition dielectric treatment eqn (28) [66]. Of course, volume density and integration should be used in the smooth-transitioned dielectric model because there is no longer a sharp interface between the solvent and solute. However, the basic operation is still the same where the polarization charges and electric displacements are needed in the region of non-uniform dielectric constant.

The numerical tests show that the charge-based formulation offers much better consistency between the results at all grid spacings, as demonstrated by the fact that the slope is always very close to 1 and the deviation from analytical values is smaller than the method as in eqn (18)/(21) or eqn (24). Furthermore, the numerical uncertainties of the atomic forces by the charge-based method are also smaller, suggesting less significant grid dependence. The mean total electrostatic force by the charge-based formulation is also closer to zero, but its fluctuation is on the same order as that by the method as in eqn (24).

Further improvement can also be achieved by noting that the tangential components of the electric field are extremely small and hard to compute accurately with finite grid spacing because of the large jump in the dielectric constant between the solute and the solvent (typically, in water the dielectric constant is about 80, and in molecules it is set to be 1 - 4). Thus after dropping off the tangential component in eqn (28), we have

$$f_{DBF} = \frac{1}{2} \rho_{pol} \frac{|D|^2}{D_n} n = \frac{1}{2} \rho_{pol} D_n n.$$  (30)
Numerical tests show that the correlations between the so-called normal field approximation and total field method are quite good for high-resolution numerical calculations. Hence use of the normal component of the field alone is a reasonable choice given the more difficult task of interpreting the total field, or interpreting the extremely small tangential field. This is particularly difficult for typical coarse grid resolution used in biomolecular applications. Following the same logics, the charge-based integral formulation, eqn (29), can also be approximated as

$$f_{DBF} = \frac{1}{2} \rho_{pol} (D_o \cdot n) \cdot (D_i \cdot n)$$

Equation (31)

$$= \frac{1}{2} \rho_{pol} D_{on} n = \frac{1}{2} \rho_{pol} D_{in} n.$$

The boundary element method is another approach to incorporate the continuum electrostatics into molecular mechanics simulations [49, 72-75, 86, 87]. The DBF calculation in this method using a polarization charge method was first described by Zauhar [62], who showed that the DBF can be calculated as

$$f_{DBF} = \left[ 2\pi \sigma^2 \varepsilon + \frac{1}{8\pi} (\varepsilon_o - \varepsilon_i) \| E_o \|^2 \right] n,$$

Equation (32)

where $\sigma$ is the surface polarization charge density. This expression was derived from eqn (25). The use of surface polarization charge density makes it straightforward in the boundary element method, where the Poisson’s equation can be solved through the iteration of the surface polarization charge density. Cortis et. al. also tried to compute the DBF via the Maxwell stress tensor for their finite element method, leading to the same formulation as that of Zauhar [35].
1.6 Numerical point-charge force interpolations

In this review, we focus on numerical interpolations of forces for the widely used finite-difference schemes. Apparently other discretization schemes require the same careful considerations to achieve the best numerical accuracy and precision. In a finite-difference scheme all atomic charges have to be mapped onto grid points during discretization before potential and electric field can be computed. Thus charge assignment needs to be considered along with force interpolation for optimal numerical performance. In the following we review several typical methods in charge assignment and force interpolation.

1.6.1 NGP procedure

The nearest grid point (NGP) force calculation scheme assigns the full charge from a given particle to its nearest grid point and takes as the value of the force on the particle the mesh-computed value at the nearest grid point. The errors in the inter-particle forces arise solely from the crudeness of the charge assignment and force interpolation scheme. Apparently no other procedure is simpler and more efficient than the NGP procedure [88].

1.6.2 CIC procedure

A better approximation to the force, but one that is more costly in terms of number of arithmetic operations per particle per time step, can be obtained by replacing NGP assignment by a scheme involving all nearest neighbors. This more accurate scheme is known as “cloud-in-cell” (CIC) scheme [88].

Consider a 1D system of two particles with opposite charges. Particle 2 lies on the mesh point at zero and particle 1 lies on the mesh point p and p+1. To compute the force on particle 1, instead of taking the NGP value, we took a linear combination of the two nearest values as
\[ F(x) = \beta F(x_p) + (1 - \beta)F(x_{p+1}). \] (33)

For the above two particles system, it was proven that \( \beta = \frac{x_{p+1} - x}{x_{p+1} - x_p} \) can be chosen to make the error equal to zero in the 1D case. For the 3D case, a general weight function can be proposed following the same argument as

\[
W(x - x_p, y - y_p, z - z_p) = \begin{cases} 
(1 - \frac{|x - x_p|}{h})(1 - \frac{|y - y_p|}{h})(1 - \frac{|z - z_p|}{h}), \\
\max(|x - x_p|, |y - y_p|, |z - z_p|) \leq h \\
0, \text{otherwise}
\end{cases}
\] (34)

where \( h \) is the grid spacing. Thus eqn (33) can be written in the more general form for 3D cases as [88]

\[
F(x, y, z) = \sum_p W(x - x_p, y - y_p, z - z_p)E(x_p, y_p, z_p)q_\alpha, \] (35)

where the sum \( p \) is taken over all the adjacent grid points, \((x_\alpha, y_\alpha, z_\alpha)\) and \( q_\alpha \) are the position and charge of particle \( \alpha \), and \( E(x_p, y_p, z_p) \) is the interpolated electric field at point \( p \).

A similar procedure should also be used for charge assignment to achieve the best performance. Consider the same 1D system where Particle 1 is at a position between mesh points \( p \) and \( p+1 \), but now particle 2 lies at \( x \) between mesh points 0 and 1. If NGP charge assignment were used, then the force on particle 1 due to particle 2 would be insensitive to the displacement of particle 2 in the range of \( x \in [0, h/2] \) and \( x \in [h/2, h] \), but would change discontinuously at
\( x = \frac{h}{2} \). In addition system momentum should also be conserved, i.e., the inter-particle forces between a pair of particles supposed to be equal and opposite. To deal with both issues, the CIC system assigns the charge \(-q\) to the density function as follows

\[
\rho(x_0) = \frac{(-q)}{h} \beta \rho(x_1) = \frac{(-q)}{h} (1 - \beta),
\]

where \( \beta = \frac{x_{p+1} - x}{x_{p+1} - x_p} \), to cancel out the discontinuous effect as well as to minimize the force interpolation error. More generally, for 3D cases, the charge assignment is proposed as [88]

\[
\rho(x_p, y_p, z_p) = \frac{1}{h^3} \sum W(x_\alpha - x_p, y_\alpha - y_p, z_\alpha - z_p) q_\alpha,
\]

which sums over all the charged particles within the adjacent cubic of grid \( p \), \((x_\alpha, y_\alpha, z_\alpha)\) and \( q_\alpha \) are the position and charge of particle \( \alpha \) and \( W(x_\alpha - x_p, y_\alpha - y_p, z_\alpha - z_p) \) is the same weighting function in eqn (34).

The CIC scheme not only gives a much smoother force than the NGP scheme, but also reduces the amplitude of fluctuations in the inter-particle forces as the particles are displaced with respect to the mesh. In addition the residue errors in numerical forces are spatially more localized in the CIC scheme than in the NGP scheme.

1.6.3 Im et al’s approach for force interpolation
Im et. al. [30] explored to improve the electrostatic force assignment for finite-difference methods, they used the same charge assignment method as the CIC procedure, but proposed a different method to interpret the numerical forces.

For atom $\alpha$ at position $(x_\alpha, y_\alpha, z_\alpha)$, the x-component of reaction force on the atom can be written as

$$F_{x\alpha}^{RF} = -\sum_p (\phi_s - \phi_v) q_\alpha \left( \frac{\partial w(x_\alpha - x_p)}{\partial x} \right) w(y_\alpha - y_p) w(z_\alpha - z_p),$$

(38)

where $w(t) = (1 - |t|/h)$ is a weighting function – the 1D case of the weighting function $W$ in eqn (34). The derivative of the weighting function thus satisfies

$$\frac{\partial w(x_\alpha - x_p)}{\partial x_\alpha} = \begin{cases} h^{-1}, & x_\alpha < x_p \\ 0, & x_\alpha = x_p \\ -h^{-1}, & x_\alpha > x_p \end{cases}$$

(39)

The summation is over all the nearest grid point $p$ of atom $\alpha$ and $q_\alpha$ is the charge on the atom. Similar expression applies to $F_{y\alpha}^{RF}$ and $F_{z\alpha}^{RF}$. Here $\phi_s$ and $\phi_v$ are solvated and in vacuo potentials (to compute the reaction field forces). The scheme is better for interpolation since we do not need to interpolate the electric field. Nevertheless, its performance is not as good as the CIC scheme when the grid space is not small enough.

1.6.4 Least square interpolation

To obtain potential or electrostatic field at any position $(x_0, y_0, z_0)$, the general one-sided least-square interpolation method can also be used to compute both potential and field within the solute [89] [90]. Briefly a function of the form
\[ f(x, y, z) = a_0 + a_1(x - x_0) + a_2(y - y_0) + a_3(z - z_0) \\
+ a_4(x - x_0)^2 + a_5(y - y_0)^2 + a_6(z - z_0)^2 \\
+ a_7(x - x_0)(y - y_0) + a_8(y - y_0)(z - z_0) + a_9(z - z_0)(x - x_0) \]  

(40)

is fitted using the potentials of \( N_p \) (\( \geq 19 \) nearest grid points of an atom in the solute region. The coefficients \( a_i, i = 1, \ldots, 10 \) are determined by minimizing

\[ d = \sum_{p}^{N_p} [\varphi(x_p, y_p, z_p) - f(x_p, y_p, z_p)]^2, \]

(41)

so that the potential and gradient of potential at position \( x_0, y_0, z_0 \) is given by the following relation

\[ \varphi(x_0, y_0, z_0) = f(x_0, y_0, z_0) = a_0 \]
\[ E_x = \varphi_x(x_0, y_0, z_0) = f_x(x_0, y_0, z_0) = a_1 \]
\[ E_y = \varphi_y(x_0, y_0, z_0) = f_y(x_0, y_0, z_0) = a_2 \]
\[ E_z = \varphi_z(x_0, y_0, z_0) = f_z(x_0, y_0, z_0) = a_3 \]

(42)

Since the algorithm contains all the second order terms, it can achieve second order accuracy when doing the computation. Nevertheless, the extra cost over the CIC method is also noticeable.

### 1.7 Numerical dielectric-boundary force interpolations

The interpolation of dielectric boundary forces is more involving because the forces act upon the molecular surface that separates the low dielectric molecular interior from the high dielectric exterior. The key for higher accuracy and precision is thus how to handle the jump in the dielectric constant. The following methods were proposed in the past to discretize the various algorithms presented in sections IV and V.
1.7.1 Discretization of field based formulations

Gilson et. al. presented an efficient approach to discretize the original variational formulation (see eqn (18)) given that the harmonic averaging treatment is used for the interface dielectric smoothing. In the following we use the treatment of fractional x-edges to illustrate the basic idea [64]. The derivative of dielectric constant can be expressed as

$$\nabla \varepsilon = -\varepsilon^2 (\varepsilon_i - \varepsilon_o) \left( \frac{\Delta x}{h \Delta x} \mathbf{e}_x + \frac{\Delta y}{h \Delta x} \mathbf{e}_y + \frac{\Delta z}{h \Delta x} \mathbf{e}_z \right) = -\frac{\varepsilon^2 (\varepsilon_i - \varepsilon_o)}{\varepsilon_i \varepsilon_o} \frac{|\mathbf{r}|}{h \Delta x} \hat{\mathbf{r}}, \quad (43)$$

where $\mathbf{r}$ is the distance between the intersection point and the center of the atom/probe, $h$ is the grid length. The DBF term in eqn (18) can then be rewritten in discretized form as

$$f_{DBF} = -\frac{1}{8\pi} \nabla \varepsilon h^3 E_x^2 = \frac{1}{8\pi} \frac{\varepsilon^2 (\varepsilon_i - \varepsilon_o)}{\varepsilon_i \varepsilon_o} h^2 E_x^2 \frac{|\mathbf{r}|}{\Delta x} \hat{\mathbf{r}} = \frac{1}{8\pi} \frac{\varepsilon_i - \varepsilon_o}{\varepsilon_i \varepsilon_o} h^2 D_x^2 \frac{1}{\cos \theta} \hat{\mathbf{r}}, \quad (44)$$

where $E_x$ is the electric field in the x direction at the intersection point and $D_x$ is the x component of $\mathbf{D}_n$. Here $\cos \theta = \frac{d'}{|\mathbf{r}|}$ is used, $d'$ is the x-distance between the intersection point and the center of the atom/probe. Thus we have $D_x = \mathbf{D}_n \cos \theta$. Discretized forms in the y/z-directions can also be worked out similarly. It should be pointed out that the division by $\cos \theta$ in eqn (44) is a potential source of numerical instability since division by floating point number very close to zero is often observed in molecular simulations.

Following Gilson et. al.’s variational approach, Im et. al. proposed an equivalent expression to calculate the DBF force as [30]
where $\phi$ is the total electrostatic potential, $r$ represents the distance between a point of the interface and the center of an atom. Apparently both eqn (18) and (45) require smoothly varying dielectric models since $\nabla \varepsilon$ has to be finite, i.e. $\varepsilon$ has to be designed to change from $\varepsilon_i$ to $\varepsilon_o$ sufficiently smoothly for stable numerical performance [91].

To discretize eqn (45), the authors used the divergence theorem to eliminate the divergence operator, so that the total solute DBF can be discretized as

$$f_{DBF} = \left[ \frac{1}{8\pi} \phi \nabla \cdot \left( \frac{\partial \varepsilon}{\partial r} \nabla \phi \right) \right] n,$$

where $\phi$ is the total electrostatic potential, $r$ represents the distance between a point of the interface and the center of an atom. Apparently both eqn (18) and (45) require smoothly varying dielectric models since $\nabla \varepsilon$ has to be finite, i.e. $\varepsilon$ has to be designed to change from $\varepsilon_i$ to $\varepsilon_o$ sufficiently smoothly for stable numerical performance [91].
law with the grid potentials. Given this numerical representation of polarization charges and eqn (30), the total DBF in the solution system, $F_{DBF}$, can be computed numerically as

$$F_{DBF} = \int f_{DBF} \, dv = \frac{1}{2} \sum_{i=1}^{nbnd} q_{i}^{pol} D_{n,i} n_{i} = \frac{1}{2} \sum_{i=1}^{nbnd} \varepsilon q_{i}^{pol} E_{n,i} n_{i},$$

(47)

where nbnd is the number of the solute/solvent dielectric boundary grid points where the polarization charges are located. Of course the polarization charges are zero on all other grid points where there is no dielectric variation. $q_{i}^{pol}$, $D_{n,i}$, $E_{n,i}$, and $n_{i}$ are, respectively, the polarization charge, normal electric displacement, normal surface field, and surface normal unit vector at dielectric boundary grid point $i$. It is apparent that the normal field approximations of the two charge-based formulations are identical after inspecting eqn (30) and eqn (31). Thus eqn (47) is applicable to both charge based formulations.
References


23


CHAPTER 2
Electrostatic Forces in the Poisson-Boltzmann Systems

2.1 Summary

Continuum modeling of electrostatic interactions based upon numerical solutions of the Poisson-Boltzmann equation has been widely used in structural and functional analyses of biomolecules. A limitation of the numerical strategies is that it is conceptually difficult to incorporate these types of models into molecular mechanics simulations, mainly because of the issue in assigning atomic forces. In this theoretical study, we first derived the Maxwell stress tensor for molecular systems obeying the full nonlinear Poisson-Boltzmann equation. We further derived formulations of analytical electrostatic forces given the Maxwell stress tensor and discussed the relations of the formulations with those published in the literature. We showed that the formulations derived from the Maxwell stress tensor require a weaker condition for its validity, applicable to nonlinear Poisson-Boltzmann systems with a finite number of singularities such as atomic point charges and the existence of discontinuous dielectric as in the widely used classical piece-wise constant dielectric models.

2.2 Introduction

Solvation interaction is one of the essential determinants of the structures and functions in proteins and nucleic acids, and is crucial in their accurate modeling [1-15]. Due to the high
computational overhead in the explicit treatments of solvent molecules, continuum models of solvation interactions, specifically the electrostatic solvation modeling that is based upon the Poisson-Boltzmann equation (PBE), are widely used in the studies of biomolecules [1-15]. Indeed continuum electrostatics represents an effective and physically sound approach that makes it possible to account for a number of phenomena involving solvent electrostatic effects on the functional analyses of biomolecules [1-15]. These efforts have been facilitated by numerical solutions of the PBE that can be obtained routinely for biomolecules and their complexes [3, 16-21]. Among the numerical solution methods, finite-difference method (FDM) [22-34], finite-element method (FEM) [35-43] and boundary-element method (BEM) [44-59] are mostly used.

A disadvantage of the numerical continuum electrostatics methods is that it is conceptually difficult to incorporate them into molecular mechanics programs, mainly because of the problem of assigning forces related to the dielectric boundary (“dielectric stress”) to individual atoms [30, 35, 60-68]. Other problems are the convergence of the energy and forces with respect to the resolution of the solute-solvent boundary and charge representation [69-71]. Thus in most practical applications, the PBE is only solved for a few fixed conformations of a solute or solute complex. This limits the application of PBE in more elaborative modeling of biomolecules.

Many efforts have been invested to develop methods to compute electrostatic forces [30, 35, 49, 60-68, 72-79]. The “virtual work” method is apparently the defining benchmark for all analytical methods. In the “virtual work” method the electrostatic energy $G$ is recalculated for a
small displacement $d$ of each atom in the $x, y,$ and $z$ directions, respectively. The numerical force is then $-\Delta G / d$ for each direction. The limitation of this approach, however, is that at least four full numerical calculations are required in order to calculate each force vector. Apparently, this is only realistic for molecules treated as rigid bodies. In addition, the numerical forces, defined as the negative finite-difference derivatives, are very difficult to converge when the electrostatic energies are computed numerically due to the cancellation of significant digits in the subtraction of two large numbers. Thus analytical calculation of solvation forces is necessary for practical applications.

For the classical abrupt-transitioned two-dielectric models, multiple strategies have been proposed by Davis and McCammon [60], Che et al. [65], Li et al. [67], and most recently Cai et al. [68]. These formulations were derived following different strategies and were found to be consistent as to be discussed below. For the smooth-transitioned dielectric models, we have the ground-breaking strategy by Gilson et al. [64]. Subsequent works by Im et al. [30] and Cai et al. [66] were shown to be consistent with that of Gilson et al. [64], though different strategies were proposed to enhance numerical stability and convergence in the later works. The numerical methods derived from these formulations are mostly adapted for the numerical solutions by the FDM. Another promising approach to incorporate the PBE electrostatics into molecular mechanics is the BEM. The force calculation in a BEM calculation was first described by Zauhar [62]. Cortis et al. [35] also tried to compute the solvation force for their FEM calculations, leading to the same formulation as that of Zauhar [62].

These pioneer efforts in calculating the solvation forces for the numerical PBE methods
laid the foundation to develop more robust methods that eventually will lead to routine application of the numerical PBE methods to biomolecular simulations. Interestingly, there is yet no theoretical analysis on how to derive the Maxwell stress tensor, a crucial step in formulating the analytical solvation forces, even if the concept has been used in the solvation force derivations in the past. This study presents a theoretical framework of deriving the Maxwell stress tensor based on the full nonlinear PBE. In addition, it offers a systematic derivation of the formulations of electrostatic forces for systems with and without singularities that exist as point charges, abrupt-transitioned dielectric interface, and ionic interface. The theoretical study further discusses the relations between the Maxwell stress tensor-based formulations and the existing methods in the literature, and highlights the benefits of various numerical strategies proposed in the literature.

2.3 Derivation of Maxwell stress tensor for the full Poisson-Boltzmann systems

In the following we focus on the full PBE for systems with continuum mobile ions

$$\nabla \cdot (\varepsilon \nabla \phi) = -4\pi \rho^f - 4\pi \sum_i q_i c_i \exp(-q_i \phi / k_B T) \lambda$$

(1)

where $\varepsilon$ is the dielectric constant, $\phi$ is the potential, $\rho^f$ is the fixed charge distribution, $q_i$ is the charge of ion type $i$, $c_i$ is the bulk number density of ion type $i$, $\lambda$ is the ion exclusion function, $k_B$ is the Boltzmann constant and $T$ is the absolute temperature. Introducing the
electric displacement vector $\mathbf{D} = -\varepsilon \nabla \varphi$, eqn (1) becomes

$$\nabla \cdot \mathbf{D} = 4\pi \rho^f + 4\pi \sum_i q_i c_i \exp\left(-q_i \varphi / k_B T\right) \lambda$$  \hspace{1cm} (2)

The total electrostatic free energy of a Poisson-Boltzmann system can be written as [76]

$$G = \int \left( \rho^f \varphi - \frac{1}{8\pi} \mathbf{D} \cdot \mathbf{E} - \Delta \Pi \lambda \right) dv$$

$$\Delta \Pi = k_B T \sum_i c_i (e^{-q_i \varphi / k_B T} - 1)$$  \hspace{1cm} (3)

where $\mathbf{E}$ is the electric field.

In classical electrostatics, electrostatic forces can be computed via the variational approach, for example as in Ref. [64], or via the Maxwell stress tensor [80]. Though both strategies are not trivial to follow, here we follow the strategy based on the Maxwell stress tensor. The advantage of this strategy is that it is straightforward once the Maxwell stress tensor is defined for the problem at hand. As will be shown below, no matter what strategy is followed, the general formulation for total electrostatic forces is the same, at least for solution systems without any singularity.

Given the Maxwell stress tensor in the form of a rank two tensor

$$\mathbf{P} = \begin{bmatrix} T_{xx} & T_{xy} & T_{xz} \\ T_{yx} & T_{yy} & T_{yz} \\ T_{zx} & T_{zy} & T_{zz} \end{bmatrix}$$  \hspace{1cm} (4)

We first proceed to compute the nine components $T_{ij}$ from the electrostatic free energy for the ionic solution system obeying the PBE (eqn(1)).
Following Landau’s notation [80], with no loss of generality, consider a thin and small disk of dielectric with small area $A$ and thickness $h$, and with $h \ll \sqrt{A}$. Given the small dimension of the volume element, we can assume its uniform composition, density, temperature, and electrostatic properties such as $\varepsilon$ and $\lambda$. We further assume that charge density $\rho'$, potential $\varphi$ and electric field $E$ are all smooth variables within the volume element so that the electrostatic free energy density $g$ is also smoothly changed.

Consider the upper surface of the disk with area $A$ and normal vector $n$. The definition of stress tensor shows that $F = \int P \cdot n dA$, and hence $-P \cdot n = -(T_{ij} n_j, T_{ij} n_j, T_{ij} n_j)^T$ is simply the pressure acted upon the upper surface. Here Einstein’s summation convention is used, i.e. $T_{ij} n_j = \sum_j T_{ij} n_j$. Our goal is to impose a virtual deformation of the thin disk along a small virtual displacement $s$ ($s \ll h$) whose direction is arbitrary to compute the associated pressure via the virtual displacement method. Suppose the deformation of the disk is homogeneous so that the surface area of the deformed thick disk remains the same, $A$. Homogeneity also means that if a disk layer is located at $z$ to the bottom surface, its displacement is $\frac{z}{h} s$ after the deformation. (See Figure 2-1 before and after the deformation.)
Figure 2-1 Deformation of the disk volume element in deriving the Maxwell stress tensor.

We further assume that the virtual deformation process is isothermal to focus on the electrostatic properties. Upon the virtual deformation the pressure does work \(-T_ij n_j A_s^i\). Again here the Einstein’s summation convention is introduced, \(T_ij n_j A_s^i = \sum_i \sum_j T_ij n_j A_s^i\). On the other hand, the work done upon deforming the disk is equal to the decrease in electrostatic free energy contained in the volume element. Introducing \(\bar{g}\) to denote the average electrostatic free energy density \((\bar{g})\), the total electrostatic free energy within the volume element can be written as \(\bar{g}Ah\).

Thus we have

\[
T_ij n_j A_s^i = \delta(\bar{g}Ah) = A\delta(\bar{g})h + A\bar{g}\delta h
\]

(5)

Removing the constant surface area

\[
T_ij n_j s_i = h\delta(\bar{g}) + \bar{g}\delta h.
\]

(6)

According to eqn (3), \(g = \rho' \varphi - \frac{D \cdot E}{8\pi} - \Delta \Pi \lambda\) [76], variation of the electrostatic free energy density anywhere within the volume element gives

\[
\delta g = \varphi \delta \rho' - \frac{1}{8\pi} E^2 \delta \varepsilon - kT \sum_i (\varepsilon^e - 1)c_i \delta \lambda + \rho' \delta \varphi - \frac{1}{4\pi} D \cdot \delta E + \lambda \sum_i q_i c_i e^{\frac{\varphi}{kT}} \delta \varphi
\]

(7)

The change of \(\rho'\) is related to the change in the thickness of the thin disk by \(\delta \rho' = -\frac{\rho' \delta h}{h}\) due to the assumption of the homogeneous deformation.
The variation in $E$ is computed based on the following approximation on the potential distribution. The homogenous deformation process implies that the potential distribution in each disk layer remains the same. This is only true in a charge-free parallel capacity field [80] and is an approximation since the change in the charge density apparently leads to a change in potential if the PB equation is satisfied. However, as shown in Appendix A.1 and A.3, the change in potential is a higher-order small value compared with other changes if the PB equation is preserved.

Thus at a given point $(r)$ within a disk layer there appears to be a point originally at $r - u$ within another disk layer. Here $u$ is the vector of the displacement of the disk layer. This observation leads to the following relation between the potential and field as

$$
\delta \phi = \varphi(r - u) - \varphi(r) = -u \cdot \nabla \varphi = u \cdot E,
$$

where $E$ is the electric field in the undeformed volume element. Given $E \neq 0$ and $u = \frac{zs}{h}$ due to the uniform deformation, we have

$$
\delta \phi = \frac{z(s \cdot E)}{h}
$$

where $z$ is the distance of the layer from the lower disk surface. Taking the derivative of $\delta \phi$ over $z$ gives

$$
\delta E = \frac{n(s \cdot E)}{h}
$$

Finally $\varepsilon$ and $\lambda$ are the variables that only depend on the position and do not change upon the variation of $\rho^f$ and $E$ so that they do not cause any change in the free energy $g$ at any given position within the volume element.
With all above preparations and also considering the fact that \( \delta \phi \) is a higher order small value compared to \( \delta E \) and \( \delta \rho^f \) as shown in Appendix A.2 and A.3, eqn (7) can be simplified as

\[
\delta g = \phi \delta \rho^f - \frac{1}{4\pi} \mathbf{D} \cdot \delta \mathbf{E} \quad (10)
\]

after eliminating all the terms with \( \delta \epsilon \), \( \delta \lambda \), and \( \delta \phi \) because they either do not or only make a higher order small change when comparing with the variations in \( \delta \rho^f \) and \( \delta E \). Note that \( \rho^f \) is the free-charge density. In contrast, variation of the classical electrostatics free energy density, with the same assumptions, leads to

\[
\delta g = \frac{1}{2} \phi \delta \rho = \frac{1}{4\pi} \mathbf{D} \cdot \delta \mathbf{E} = \phi \delta \rho - \frac{1}{4\pi} \mathbf{D} \cdot \delta \mathbf{E},
\]

with \( \rho \) being the total charge density. Thus, if the classical electrostatics is used to model a solution system, \( \rho \) should include both the free (atomic) charges and the mobile ion charges. Thus the variation of energy of eqn (10) is different from the standard statement in the classical electrostatics. The extra term is responsible for the entropy change from the ionic concentration change upon polarization [81].

Substituting (9) in (10), we have

\[
\delta g = \phi \delta \rho^f - \frac{1}{4\pi} \left( \mathbf{D} \cdot \mathbf{n} \right) (s \cdot \mathbf{E}) \quad (11)
\]

Given \( \tilde{g} \to g \) and \( \delta(\tilde{g}) \to \delta g \) in the limit for the dimension of the volume element going to zero, and \( \delta \rho^f = -\frac{\rho^f \delta h}{h} \), the Maxwell stress tensor element in eqn (6) can be computed as

\[
T_{ij} n_j s_i = \frac{1}{4\pi} (\mathbf{D} \cdot \mathbf{n})(\mathbf{E} \cdot \mathbf{s}) - \rho^f \phi (s \cdot \mathbf{n}) + g(s \cdot \mathbf{n})
\]

\[
= \left( \frac{\varepsilon \mathbf{E} \cdot \mathbf{E}}{4\pi} - \frac{1}{8\pi} \varepsilon \mathbf{E}^2 \delta_{ij} - \Delta \Pi \lambda \delta_{ij} \right) n_j s_i \quad (12)
\]
Here we have also used the fact that \( \delta h = \mathbf{s} \cdot \mathbf{n} \). In summary

\[
T_{ij} = \frac{1}{4\pi} e E_i E_j - \frac{1}{8\pi} e E^2 \delta_{ij} - \Delta \Pi \lambda \delta_{ij}.
\]

(13)

where \( \Delta \Pi = k T \sum_i c_i (e^{-q_i \phi/kT} - 1) \). Note also that the tensor defined is symmetric, i.e. \( T_{ij} = T_{ji} \).

2.4 Computation of electrostatic forces – General Method

The force can be computed as \( \mathbf{P} \cdot \mathbf{n} da \) for any area element \( da \) with a normal direction of \( \mathbf{n} \). Since we are dealing with an ideal classical fluid, the natural first step is the computation of force density \( \mathbf{f} dv \) for a volume element \( dv \). This is readily available as \( \oint \mathbf{P} \cdot \mathbf{n} da \) once \( \mathbf{P} \) is known, namely, the net force felt by the volume element is the total force acting upon it on its enclosing surface. This procedure suggests that the concept of divergence of the stress tensor can be introduced, which is \( \lim_{dv \to 0} \frac{\oint \mathbf{P} \cdot \mathbf{n} da}{dv} \), as is the case for the divergence of a vector field. It is straightforward to show that the divergence theorem also holds for the stress tensor [80]

\[
\int \nabla \cdot \mathbf{P} dv = \oint \mathbf{P} \cdot \mathbf{n} da.
\]

(14)

There are two consequences in eqn (14) concerning us here. (1) The force density is simply the divergence of the stress tensor, i.e.

\[
\mathbf{f} = \nabla \cdot \mathbf{P}.
\]

(15)

(2) Total force can be computed either by the differential form, i.e. through the computation of force density [eqn (15)] followed by volume integration, or by the integral form, i.e. through the
surface integral of the stress tensor, and they are consistent according to eqn (14), i.e. when the divergence theorem holds. Since the divergence is undefined in regions with discontinuity, the differential form can only be used in regions without singularity, i.e. without jump of dielectric constant and without singular charge sources (for example the point charges represented as delta functions in the PB equation). However, for systems/regions with singularity, only the integral form can be applied. As a side note, the variational approach is also limited because it only holds in regions without singularity [64]. In the following, we shall first start our derivation on systems without any singularity.

2.5 Total electrostatic forces for systems without singularity

Based on the general discussion in section III, the force density is simply the divergence of the Maxwell stress tensor [80]. The electrostatic force density $f$ can then be expressed as

$$f = \rho f E - \frac{1}{8\pi} E^2 \nabla \varepsilon - \Delta \Pi \nabla \lambda.$$  \hfill (15)

with a simple derivation shown in Appendix A.4. The first term represents the free-charge force, the second term represents the bound-charge force due to the variation in dielectric constant, and the third term represents the pressure due to the presence of mobile ions. This is formally consistent with the formulation derived from the variational strategy by Gilson et. al. from the total electrostatic free energy (eqn. (3)) [64]. However, it is worth noting the different condition from the variational approach by Gilson et. al. that requires the smoothness in charge density, potential, and electric field throughout the solvated system of interest: the differential approach
based on the stress tensor only requires the smoothness in charge density, potential, and electric field within a local region where the divergence operator is to be applied.

Cai et al showed that the second term, termed dielectric boundary force, can be reformulated with the explicit presence of the bound polarization charge [66]

\[
\mathbf{f}_{\text{bd}}^{\text{die}} = -\frac{1}{8\pi} |\mathbf{E}|^2 \nabla \varepsilon = \frac{1}{2} \rho_{\text{pol}} \frac{|\mathbf{D}|^2}{\mathbf{D} \cdot \mathbf{n}}
\]  

(16)

where \( \rho_{\text{pol}} \) is the polarized charge density, \( \mathbf{n} \) is the normal direction of the boundary surface and \( \mathbf{D} \) is the electric displacement within the polarized area. Their numerical tests on realistic biomolecules show that the atomic solvation forces are more stable and converge faster using the so-called charge-based method. Of course both formulations are consistent with the “virtual work” principle as they demonstrated in the tested systems.

### 2.6 Total electrostatic forces in systems with singularity

If singularity exists in a solution system, eqn (15) cannot be used to compute electrostatic forces. We shall turn to the integral form, which is more fundamental and does not require the divergence of a singular Maxwell stress tensor. In general, we consider four situations, i.e. the solute region without singularity, the solute region with singularity, the dielectric interface region, and finally the mobile ion term at the Stern layer.

#### 2.6.1 Solute region with and without singularity
To obtain a detailed force distribution on each atom of the solute molecule, we first divide the solute volume into small rectangular elements, $\Delta v$ (see Figure 2-2). Apparently any rectangular volume element is much smaller than each atom, but it is still an order-of-magnitude larger than the disk-like element shown in Figure 2-1 utilized to derive the expression of the Maxwell stress tensor. As shown in Appendix A.5, for a solute region without singularity, the electrostatic force for each volume element $\Delta v$ is

$$ \mathbf{F} = \rho' \mathbf{E} \quad (17) $$

![Figure 2-2 The force on a small volume in the continuum solute.](image)

For a solute volume element with singularity, we further divide the volume element into a smaller spherical region containing the singular charge and the rest where there is no singular charge (see Figure 2-3). We can then obtain the analytical expression for the electrostatic force in the small rectangular volume element based on the Maxwell stress tensor. As shown and
summarized in Appendix A.6, the force density can then be universally written as

\[ f = \rho f E \]  

(18)

in the solute region whether the charge density is singular or not.

**Figure 2-3** Singular charge enclosed by an infinitely small spherical volume \( S' \) with radius \( \delta \).

**Figure 2-4** Decomposition of \( E' \) on the surface of the chosen small spherical volume \( S' \) in Figure 2-3

2.6.2 The dielectric interface region
Cai et al [68] utilized the integral formulation based on the Maxwell stress to compute the force in the dielectric interface region for the piece-wise constant classical dielectric models of biomolecules. Their derivation shows that

\[ f^\text{diele} = (P_o - P_i) \cdot n = \frac{1}{4\pi} \left[ (\varepsilon_o E^2_{on} - \frac{1}{2} \varepsilon_o E^2_o) - (\varepsilon_i E^2_{in} - \frac{1}{2} \varepsilon_i E^2_i) \right] n \]  

(19)

where \( n \) is the outward-directed normal unit vector of the molecular surface, and \( P_i \) and \( P_o \) are the stress tensors on the surfaces parallel to the dielectric interface inside and outside of the solute. \( E_o, E_i \) are the electric field on the two sides of the solute, and \( E_{in}, E_{on} \), respectively, are the electric field components on the \( n \) direction of \( E_o, E_i \).

They further pointed out that a charge-based strategy could also be proposed for the piece-wise constant dielectric treatment. Briefly the dielectric boundary force can be written as

\[ f^\text{diele} = \frac{1}{2} \sigma^\text{pol} \frac{\varepsilon_o E_o \cdot E_o}{E_{on} n} = \frac{1}{2} \sigma^\text{pol} \frac{D_o \cdot D_o}{D_{on} n} \]  

(20)

where \( \sigma^\text{pol} \) is the bound polarized surface charge density, and \( D_o \) and \( D_i \) are the corresponding electric displacements of the solvent and solute side, respectively.

An interesting observation is the similarity of eqn (20) and the charge-based approach for the smooth-transition dielectric treatment eqn (16) [66]. Of course, volume density and integration should be used in the smooth-transition dielectric model because there is no longer a sharp interface between the solvent and solute. However, the basic operation is still the same where the polarization charges and electric displacements are needed in the region of non-uniform dielectric constant.

The numerical tests show that the charge-based formulation offers much better
consistency between the results at all grid spacings, as demonstrated by the fact that the slope is always very close to 1 and the deviation from analytical values is smaller than the method as in eqn (19). Furthermore, the numerical uncertainties of the atomic forces by the charge-based method are also smaller, suggesting less significant grid dependence. The mean total electrostatic force by the charge-based formulation is also closer to zero, but its fluctuation is on the same order as that by the method as in eqn (19).

2.6.3 The ionic interface region

At the Stern Layer, mobile ions exist only on one side though the dielectric constants on both sides are the same. The classical method models the Stern layer with a step function of $\lambda$ that changes from 0 to 1. Given the availability of Maxwell stress tension, the boundary force caused by the excess osmotic pressure is the same as that caused by the jump of dielectric constant. We have

$$f_{\text{ion}} = (P_o - P_i) \cdot n = -\nabla \Pi$$

(21)

where $P_o, P_i$ are the corresponding stress tensors on outside and inside of the Stern layer from eqn (13). Here we have used the notation that $\lambda$ is 1 or 0 on the two sides of the layer. The simplicity of the derivation demonstrates the clean physics in the Maxwell stress tensor approach.

2.7 Connection and comparison with the Gilson et al. strategy
As discussed above, the field-based methods and the charge-based methods in the computation of dielectric boundary forces are mathematically consistent with the original method proposed by Gilson et al. Furthermore, by taking the limit of infinite thin transition zone in the smooth-transitioned dielectric model, the smooth-transitioned and abrupt-transitioned formulations can be shown to be consistent with each other mathematically, implying a high level of internal consistency between the integral and differential approaches based on the Maxwell stress tensor.

We expect all algorithms become numerically indistinguishable in the limit of zero grid spacing in the finite-difference method when the harmonic averaging method is used to smooth the transition between the solute and solvent dielectrics (chosen to be 1 and 80, respectively, in this test). Here we used a poly-alanine alpha-helix (eight residues) as an example to demonstrate their consistency numerically at the tested fine grid spacing of 1/16 Å. The finite difference box was set to be 1.5 times the dimension of the model helix due to the extremely large memory usage requirement of the finite-grid spacing. No electrostatic focusing was applied to maintain the highest possible quality in the numerical solution. The convergence of the numerical solver is reached when the relative residue is less than $10^{-9}$. The default Amber charges and continuum radii were used in the numerical calculations [82].

The correlations among the four methods are shown in Figure 2-5. It can be seen that the linear correlation coefficients between the other three methods and the Gilson et al. method are very high. Worth noting is the much higher agreement between the two methods based on the same smooth-transitioned dielectric treatment, i.e. the Gilson et al. method (labeled as “Smooth
Field” in the figure) and the charge-based method (labeled as “Smooth Charge”). The numerical test demonstrates that the four methods are numerically consistent, in agreement with our theoretical discussion above.
Figure 2-5 Correlations between the atomic dielectric boundary forces for a model helix computed by the smooth field method (i.e. the Gilson et al. method) and those computed by the smooth charge view method, the abrupt charge view method, and abrupt field method, respectively. Top: correlation between the smooth charge view method and the smooth field
method; regression line \( y = -0.005 + 0.999x, r^2 = 0.9999 \).

Middle: correlation between the abrupt charge view method and the smooth field method; regression line \( y = -0.036 + 0.957x, r^2 = 0.9989 \).

Bottom: correlation between the abrupt field method and the smooth field method; regression line \( y = -0.022 + 0.974x, r^2 = 0.9996 \).

All computations were at the grid spacing of 1/16 Å. Unit: kcal/(mol·Å).

2.8 Discussion

Much community-wide effort has been devoted to the calculation of the solvation forces in the numerical PBE methods [1, 30, 35, 49, 60-68, 72-80]. Due to the difficulty in computation of dielectric boundary forces, most of the previous reports were focused on this aspect of the force calculation.

For the abrupt-transitioned dielectric models, Davis and McCammon proposed a formulation by examining the integration of the Maxwell stress tensor of a Poisson system, the dielectric boundary forces surface density was shown as [60]

\[
f_{DBF} = -\frac{1}{8\pi}(\varepsilon_o - \varepsilon_i)(E_o \cdot E_i) \hat{n}
\]  

(22)

Integration of this quantity over the surface yields the total dielectric boundary force for the molecule. Cai et al.’s dielectric boundary forces formulation [68] on the abrupt-transitioned dielectric models is also derived from the Maxwell stress tensor. It can be shown that their method is consistent with the Davis and McCammon method: eqn (19) can be transformed into eqn (22) by Davis and McCammon [60] given the jump conditions [68].
Che et al. revisited the dielectric boundary forces calculation through a variational strategy in the classical abrupt-transitioned dielectric models [65]. Given the assumption that the normal surface field contributes predominantly to dielectric boundary force, they showed that the dielectric boundary force can be formulated as

$$f_{DBF} = -\frac{1}{8\pi} \left( \frac{1}{\epsilon_i} - \frac{1}{\epsilon_o} \right) |\epsilon \nabla \varphi|^2 \hat{n}$$  \hspace{1cm} (23)

where $\epsilon \nabla \varphi$ represents the continuous normal dielectric displacement vector on the solute/solvent dielectric interface. This formulation was later updated by Li et al. in their second paper on deriving the dielectric boundary forces from the variation of the electrostatic free energy with respect to the location change of the dielectric boundary to [67]

$$f_{DBF} = \frac{1}{4\pi} \left[ \epsilon_o |\nabla \varphi_o \cdot \hat{n}|^2 - \frac{1}{2} \epsilon_o |\nabla \varphi_o|^2 - \epsilon_i |\nabla \varphi_i \cdot \hat{n}|^2 + \frac{1}{2} \epsilon_i |\nabla \varphi_i|^2 \right]$$  \hspace{1cm} (24)

This formulation can be shown to be consistent with eqn (22) and eqn (19).

For the smooth-transitioned dielectric models, Gilson et al. presented a ground-breaking variational approach for the dielectric boundary forces [64], and it was further tailored into a numerical algorithm for the FDM. Their expression for the dielectric boundary forces can be expressed as

$$f_{DBF} = -\frac{1}{8\pi} |\mathbf{E}|^2 \nabla \epsilon \, ,$$  \hspace{1cm} (25)

Note this is consistent with eqn (15) as derived by Cai et al. based on the Maxwell stress tensor discussed above [66]. Im et al. proposed a method equivalent to eqn (25) as
\[ f_{DBF} = \left[ \frac{1}{8\pi} \varphi \nabla \cdot \left( \frac{\partial \varepsilon}{\partial r} \nabla \varphi \right) \right] \hat{n}, \]  

(26)

where \( r \) represents the atomic coordinates [30]. Apparently both eqn (25) and (26) require smoothly varying dielectric models since \( \nabla \varepsilon \) has to be finite, i.e. \( \varepsilon \) has to be designed to change from \( \varepsilon_i \) to \( \varepsilon_o \) sufficiently smoothly for stable numerical performance [83]. This would exclude the abrupt-transitioned dielectric models where \( \nabla \varepsilon \) is infinite.

The BEM is another promising approach to incorporate the continuum electrostatics into molecular mechanics simulations [49, 72-75, 84, 85]. The dielectric boundary forces calculation in the BEM using a polarization charge method was first described by Zauhar [62], who showed that the dielectric boundary forces can be calculated as

\[ f_{DBF} = -\left[ 2\pi \sigma^2 \varepsilon_i + \frac{1}{8\pi} (\varepsilon_o - \varepsilon_i) |E_o|^2 \right] \hat{n} \]

(27)

This expression was derived from eqn (22). The use of surface polarization charge density makes it straightforward in the BEM, where the Poisson’s equation can be solved through the iteration of the surface polarization charge density. Cortis et al. also tried to compute the dielectric boundary forces via the Maxwell stress tensor for their FEM, leading to the same formulation as that of Zauhar [35]. Of course, it is also consistent with eqn (20) as proposed by Cai et al. [68].

### 2.9 Appendices

#### 2.A.1 Derivation of \( \delta E \) upon the variation of the volume element
As shown in eqn (8)

$$\delta \varphi = \frac{z(s \cdot E)}{h}$$  \hspace{1cm} (28)

Taking the derivative, we obtain

$$\delta E = -\delta \nabla \varphi = -\frac{n(s \cdot E)}{h} - \frac{\nabla(s \cdot E)}{h} z$$  \hspace{1cm} (29)

Given that $E$ is a smooth variable, $\nabla(s \cdot E)$ in equation (29) is well defined and bounded.

Define $k = \frac{|\nabla(s \cdot E)|}{|s \cdot E|}$. Then we have

$$|\nabla(s \cdot E)| = |k(s \cdot E)|.$$  \hspace{1cm} (30)

Since $h$ can be arbitrarily small, it is always possible to choose $h$ so that $k \ll \frac{1}{h}$ and

$$|k(s \cdot E)| \ll \left| \frac{(s \cdot E)}{h} \right| = \left| \frac{(s \cdot E)n}{h} \right|.$$  \hspace{1cm} (31)

The condition of $z < h$ leads to

$$\left| \frac{\nabla(s \cdot E)}{h} z \right| < |\nabla(s \cdot E)|.$$  \hspace{1cm} (32)

Combining eqns (30), (31), and (32), we have

$$\left| \frac{\nabla(s \cdot E)}{h} z \right| \ll \left| \frac{(s \cdot E)n}{h} \right|,$$  \hspace{1cm} (33)

which is denoted as $\left| \frac{\nabla(s \cdot E)}{h} z \right| \sim o \left( \frac{(s \cdot E)n}{h} \right)$, as $h \to 0$. Here the little $o$ notation means

$$\lim_{h \to 0} \left| o \left( \frac{(s \cdot E)n}{h} \right) \right| = 0.$$  \hspace{1cm} (34)

Thus eqn (29) can be written as

\[ \lim_{h \to 0} \left| o \left( \frac{(s \cdot E)n}{h} \right) \right| = 0. \]
\[ \delta E = -\frac{n(s \cdot E)}{h} + o\left[ \frac{n(s \cdot E)}{h} \right]. \tag{34} \]

2.A.2 Relative orders among \( \delta \phi \), \( \delta E \), and \( \delta \rho' \) upon the variation of the volume element

We are now ready to prove that \( \delta \phi \) is a higher order small value than \( \delta E \), and \( \delta \rho' \) is the same order small value as \( \delta E \). Dot production by \( n \) on both side of eqn (34) gives

\[ n \cdot \delta E = -\frac{(s \cdot E)}{h} + o\left[ \frac{(s \cdot E)}{h} \right]. \tag{35} \]

Comparing (35) and \( \delta \phi = \frac{z(s \cdot E)}{h} \), (28) and also noting \( z < h \), we have

\[ |\delta \phi| < |hn \cdot \delta E| \sim o(n \cdot \delta E|, \text{ as } h \to 0. \tag{36} \]

Here the little \( o \) notation means \( \lim_{h \to 0} o\left[ n \cdot \delta E \right]/|n \cdot \delta E| = 0 \). Thus \( \delta \phi \) is a higher order small value than \( \delta E \).

Next given \( \delta h = s \cdot n \), we have

\[ |\delta \rho'| = \left| \frac{\rho'(s \cdot n)}{h} \right| = \left| \frac{\rho'(s \cdot n)}{h} \right|. \tag{36} \]

Comparison of eqns (36) with (35) shows that \( \delta \rho' \) is a same order small value as \( \delta E \) since they both contains small value \( s \) in the numerator and small value \( h \) in the denominator while both \( \rho' \) and \( E \) are finite.

Similarly from (28) we have

\[ |\delta \phi| = \left| \frac{(s \cdot E)z}{h} \right|. \tag{37} \]

Since \( z \) is a small value but both \( \rho' \) and \( E \) are finite, \( \delta \phi \) is one order smaller than \( \delta \rho' \).
as given in (36).

2.A.3 Preservation of nonlinear-PB equation upon the variation of the volume element

We have shown in (34) that the leading term of $\delta \mathbf{E}$ is

$$\delta \mathbf{E} = -\frac{n(s \cdot \mathbf{E})}{h}.$$  \hfill (38)

Apparently any change in the field must also satisfy the governing PB equation

$$\nabla \cdot \mathbf{\varepsilon} \mathbf{E} = -4\pi \rho' - 4\pi \rho^{\prime\prime}.$$  \hfill (39)

Indeed a potential violation of the PB equation is possible during the deformation process as discussed in the following. However, any deviations from the PB equation are only higher order small variations than that in (38) and its comparable variations.

Given that $\varepsilon$ and $\lambda$ do not change at the same position, $\delta \lambda = 0$ and $\delta \varepsilon = 0$ during the deformation process. The PB equation

$$\nabla \cdot \mathbf{\varepsilon} \mathbf{E} = -4\pi \rho' - 4\pi \sum_i q_i c_i \exp(-q_i \phi / k_B T) \lambda$$  \hfill (40)

can be varied as

$$\nabla \cdot \mathbf{\varepsilon} \delta \mathbf{E} = -4\pi \delta \rho' + 4\pi \sum_i q_i c_i / k_B T \exp(-q_i \phi / k_B T) \lambda \delta \phi$$  \hfill (41)

after dropping the zero variations. Since we only consider a very small volume element with infinite small changes in all smooth variables in the variational analysis, only the lowest order small terms need to be retained. All higher order small terms can be eliminated during the derivation.

After eliminating of $\delta \phi$ in eqn (41), we have
\[ \nabla \cdot (\varepsilon \delta \mathbf{E}) = 4\pi \rho f \frac{\delta h}{h} \]  

(42)

Substitution of eqn (38), we have

\[ \nabla \cdot (\varepsilon \delta \mathbf{E}) = -\frac{\partial}{\partial n} [\varepsilon (s \cdot \mathbf{E})] \frac{\delta h}{h} \]  

(43)

Next we show that eqn (42) can be satisfied by making an infinitely small adjustment of \( \delta \mathbf{E} \). Introducing \( \frac{\partial}{\partial n} [\varepsilon (s \cdot \mathbf{E})] \) to denote the average value of the gradient within the volume element, we change \( \delta \mathbf{E} \) to \( \delta \mathbf{E} + \frac{\partial}{\partial n} [\varepsilon (s \cdot \mathbf{E})] \frac{\delta h}{\varepsilon h} \), so that

\[ \delta \mathbf{E} = -n (s \cdot \mathbf{E}) \frac{\partial}{\partial n} [\varepsilon (s \cdot \mathbf{E})] \frac{\delta h}{h} + \frac{\partial}{\partial n} [\varepsilon (s \cdot \mathbf{E})] \frac{\delta h}{\varepsilon h} z + 4\pi \rho f \frac{\delta h}{h} \]  

(44)

And then

\[ \nabla \cdot \varepsilon \delta \mathbf{E} = - \frac{\partial}{\partial n} [\varepsilon (s \cdot \mathbf{E})] \frac{\delta h}{h} + \frac{\partial}{\partial n} [\varepsilon (s \cdot \mathbf{E})] \frac{\delta h}{h} + 4\pi \rho f \frac{\delta h}{h} \]  

(44)

Notice that \( \varepsilon, \mathbf{E} \) are both smoothly changed and hence we can expect that there is only a very small difference between \( \frac{\partial}{\partial n} [\varepsilon (s \cdot \mathbf{E})] \) and \( \frac{\partial}{\partial n} [\varepsilon (s \cdot \mathbf{E})] \). Also give

\[ n \cdot \delta h = s \cdot n, \text{ so that} \]  

\[ \left| 4\pi \rho f \frac{\delta h}{h} \right| \sim \left| 4\pi \rho f \frac{s \cdot n}{h} \right| \quad \left| \frac{\partial}{\partial n} [\varepsilon (s \cdot \mathbf{E})] \right| \gg \left| \frac{\partial}{\partial n} [\varepsilon (s \cdot \mathbf{E})] \right| + \frac{\partial}{\partial n} [\varepsilon (s \cdot \mathbf{E})] \]  

(45)

Thus we can eliminate the first two terms on the right side of eqn (44) and eqn (42) is satisfied.
Finally $z < h$ is a very small value, so that the second term and the third term on the right hand side of equation (44) are one order smaller compare to the first term. In summary we only need to modify the original variation of $\mathbf{E}$ with a higher order small value to preserve the PB equation. The same argument can also be applied for derivation with $\delta \phi$ as the leading small value.

2.A.4 Derivation of the electrostatic force density

Based on the general discussion in section III, the force density is simply the divergence of the Maxwell stress tensor. [80]

$$f = \nabla \cdot \mathbf{P} = \frac{\partial}{\partial x}(\mathbf{e}_x \cdot \mathbf{P}) + \frac{\partial}{\partial y}(\mathbf{e}_y \cdot \mathbf{P}) + \frac{\partial}{\partial z}(\mathbf{e}_z \cdot \mathbf{P})$$

(46)

We rewrite eqn (13) into the following form

$$T_{ij} = \frac{1}{4\pi} E_i D_j - \frac{1}{8\pi} \varepsilon E^2 \delta_{ij} - \Delta \Pi \lambda \delta_{ij}$$

(47)

where $\Delta \Pi = kT \sum_i c_i (e^{-q_i \phi/kT} - 1)$. So along the direction of $\mathbf{e}_x$, the force component is

$$f_x = \frac{\partial T_{xx}}{\partial x} + \frac{\partial T_{xy}}{\partial y} + \frac{\partial T_{xz}}{\partial z}$$

$$= \frac{1}{4\pi}\left[-\frac{1}{2} \varepsilon \frac{\partial}{\partial x} E^2 + \sum_i \frac{\partial (E_i D_i)}{\partial i} - \frac{E^2 \partial \varepsilon}{8\pi} \frac{\partial \lambda}{\partial x} \right] - \Delta \Pi \frac{\partial \lambda}{\partial x} + \sum_j q_j c_j e^{-q_j \phi/kT} \lambda \frac{\partial \phi}{\partial x}$$

$$= \frac{1}{4\pi}\left[ -\varepsilon E_i \frac{\partial E_i}{\partial x} + \sum_i D_i \frac{\partial E_i}{\partial i} + E_x \sum_i \frac{\partial D_i}{\partial i} \right] - \frac{E^2 \partial \varepsilon}{8\pi} \frac{\partial \lambda}{\partial x} - \Delta \Pi \frac{\partial \lambda}{\partial x} - \sum_j q_j c_j e^{-q_j \phi/kT} \lambda E_x$$

(48)

$$= \frac{1}{4\pi} \sum_i D_i \left( \frac{\partial E_i}{\partial i} - \frac{\partial E_i}{\partial x} \right) - \frac{E^2 \partial \varepsilon}{8\pi} \frac{\partial \lambda}{\partial x} - \Delta \Pi \frac{\partial \lambda}{\partial x} + \frac{1}{4\pi} \left( -4\pi \sum_j q_j c_j e^{-q_j \phi/kT} \lambda + \nabla \cdot \mathbf{D} \right) E_x$$

Where $i = x, y, z$. Since there is no changing magnetic field, we have $\nabla \times \mathbf{E} = \mathbf{0}$. Thus the first term on the right side of eqn (48) equals to zero. Given the notation in eqn (2), we obtain
\[ f_x = \rho f E_x - \frac{E^2}{8\pi} \frac{\partial \epsilon}{\partial x} - \Delta \Pi \frac{\partial \lambda}{\partial x} \]  

(49)

Following the same way, we obtain

\[ f_y = \rho f E_y - \frac{E^2}{8\pi} \frac{\partial \epsilon}{\partial y} - \Delta \Pi \frac{\partial \lambda}{\partial y} \]  

(50)

\[ f_z = \rho f E_z - \frac{E^2}{8\pi} \frac{\partial \epsilon}{\partial z} - \Delta \Pi \frac{\partial \lambda}{\partial z} \]

Thus the force density \( f \) is

\[ f = \rho f \mathbf{E} - \frac{1}{8\pi} \mathbf{E}^2 \nabla \epsilon - \Delta \Pi \nabla \lambda \]  

(51)

2.A.5 Solute region without singularity

For a volume element in the solute region, shown in Figure 2-2, the force acting on the left surface perpendicular to the \( x \) axis is written as

\[ F_x(x) = P(x) \Delta y \Delta z \]

\[ = \left[ \frac{1}{4\pi} (\epsilon E_x^2 - \frac{1}{2} \epsilon \mathbf{E}^2) \mathbf{e}_x + \frac{1}{4\pi} (\epsilon \mathbf{E}_y \mathbf{E}_x) \mathbf{e}_y + \frac{1}{4\pi} (\epsilon \mathbf{E}_z \mathbf{E}_x) \mathbf{e}_z \right] \Delta y \Delta z \]  

(52)

\[ = \left[ \frac{1}{8\pi} (\epsilon E_x^2 - \epsilon E_y^2 - \epsilon E_z^2) \mathbf{e}_x + \frac{1}{4\pi} (\epsilon \mathbf{E}_y \mathbf{E}_x) \mathbf{e}_y + \frac{1}{4\pi} (\epsilon \mathbf{E}_z \mathbf{E}_x) \mathbf{e}_z \right] \Delta y \Delta z \]

Similarly the force acting on the right surface perpendicular to the \( x \) axis is
Due to the absence of singularity, the Taylor expansion can be used. The difference of \( F_x(x) \) and \( F_x(x + \Delta x) \) is the force acting on the surface perpendicular to the \( x \) direction. With \( \Delta x \Delta y \Delta z = dv \), we have

\[
F_x(x + \Delta x) = \left[ \frac{1}{8\pi} \left( \varepsilon_i E_x^2(x + \Delta x) - \varepsilon_i E_y^2(x + \Delta x) - \varepsilon_i E_z^2(x + \Delta x) \right) \mathbf{e}_x + \right.
\]

\[
\frac{1}{4\pi} \left[ \varepsilon_i E_y(x + \Delta x) E_z(x + \Delta x) \mathbf{e}_y + \right.
\]

\[
\frac{1}{4\pi} \left[ \varepsilon_i E_z(x + \Delta x) E_z(x + \Delta x) \mathbf{e}_z \right] \Delta y \Delta z
\]

\[
= \left[ \frac{1}{8\pi} \varepsilon_i [(E_x + \frac{\partial E_x}{\partial x} \Delta x)^2 - (E_y + \frac{\partial E_y}{\partial x} \Delta x)^2 - (E_z + \frac{\partial E_z}{\partial x} \Delta x)^2] \mathbf{e}_x + \right.
\]

\[
\frac{1}{4\pi} \varepsilon_i [(E_y + \frac{\partial E_y}{\partial x} \Delta x)(E_y + \frac{\partial E_y}{\partial x} \Delta x)] \mathbf{e}_y + \right.
\]

\[
\frac{1}{4\pi} \varepsilon_i [(E_z + \frac{\partial E_z}{\partial x} \Delta x)(E_z + \frac{\partial E_z}{\partial x} \Delta x)] \mathbf{e}_z \Delta y \Delta z
\]

Similarly, the force acting on the surfaces perpendicular to the \( y \) direction and the \( z \) direction is

\[
F_y(y + \Delta y) = \frac{1}{4\pi} \varepsilon_i \left[ \left( \frac{\partial E_y}{\partial y} E_y + \frac{\partial E_y}{\partial y} E_y \right) \mathbf{e}_x + \right.
\]

\[
\left( \frac{\partial E_x}{\partial y} E_x - \frac{\partial E_x}{\partial y} E_y - \frac{\partial E_z}{\partial y} E_z \right) \mathbf{e}_y + \right.
\]

\[
\left( \frac{\partial E_y}{\partial y} E_y + \frac{\partial E_y}{\partial y} E_y \right) \mathbf{e}_z \right] dv
\]

\[
F_z(z + \Delta z) = \frac{1}{4\pi} \varepsilon_i \left[ \left( \frac{\partial E_z}{\partial z} E_z + \frac{\partial E_z}{\partial z} E_z \right) \mathbf{e}_x + \right.
\]

\[
\left( \frac{\partial E_x}{\partial z} E_x - \frac{\partial E_x}{\partial z} E_z - \frac{\partial E_y}{\partial z} E_y \right) \mathbf{e}_y + \right.
\]

\[
\left( \frac{\partial E_z}{\partial z} E_z + \frac{\partial E_z}{\partial z} E_z \right) \mathbf{e}_z \right] dv
\]
\( F_z(z) - F_z(z + \Delta z) = \frac{1}{4\pi} \varepsilon_i \left[ \left( \frac{\partial E_x}{\partial z} E_z + \frac{\partial E_y}{\partial z} E_z \right) e_x + \left( \frac{\partial E_x}{\partial z} E_z + \frac{\partial E_y}{\partial z} E_x \right) e_y + \left( \frac{\partial E_x}{\partial z} E_z - \frac{\partial E_x}{\partial z} E_x - \frac{\partial E_y}{\partial z} E_y \right) e_z \right] dv \)  

(56)

Summation of the \( e_z \) components of \( F_x(x) - F_x(x + \Delta x) \), \( F_y(y) - F_y(y + \Delta y) \), and \( F_z(z) - F_z(z + \Delta z) \) gives us the force acting on the volume element in the \( x \) direction

\[ F_x = \frac{1}{4\pi} \varepsilon_i \left( \frac{\partial E_x}{\partial x} E_x - \frac{\partial E_y}{\partial x} E_y - \frac{\partial E_z}{\partial x} E_z + \frac{\partial E_x}{\partial y} E_x + \frac{\partial E_y}{\partial y} E_y + \frac{\partial E_z}{\partial y} E_z + \frac{\partial E_x}{\partial z} E_x + \frac{\partial E_y}{\partial z} E_y + \frac{\partial E_z}{\partial z} E_z \right) dv \]  

(57)

Due to \( \nabla \times \mathbf{E} = 0 \), we have

\[
\left\{ \begin{array}{l}
\frac{\partial E_x}{\partial y} E_z = \frac{\partial E_x}{\partial z} E_z \\
\frac{\partial E_x}{\partial z} E_y = \frac{\partial E_x}{\partial y} E_z \\
\end{array} \right.
\]

(58)

With eqn (58), eqn (57) can be simplified as

\[ F_x = \frac{1}{4\pi} \varepsilon_i \left( \frac{\partial E_x}{\partial x} E_x + \frac{\partial E_y}{\partial y} E_y + \frac{\partial E_z}{\partial z} E_z \right) dv \]  

(59)

Similarly it can be shown that

\[ F_y = \frac{1}{4\pi} \varepsilon_i \left( \frac{\partial E_x}{\partial x} E_y + \frac{\partial E_y}{\partial y} E_y + \frac{\partial E_z}{\partial z} E_y \right) dv \]  

(60) \[
F_z = \frac{1}{4\pi} \varepsilon_i \left( \frac{\partial E_x}{\partial x} E_z + \frac{\partial E_y}{\partial y} E_z + \frac{\partial E_z}{\partial z} E_z \right) dv \]

(61)

With eqn (59), (60) and (61), the total force acting the volume element can be written as
\[ F = F_x \mathbf{e}_x + F_y \mathbf{e}_y + F_z \mathbf{e}_z \]
\[ = \frac{1}{4\pi} \varepsilon_i \left( \frac{\partial E_x}{\partial x} + \frac{\partial E_y}{\partial y} + \frac{\partial E_z}{\partial z} \right) (E_x \mathbf{e}_x + E_y \mathbf{e}_y + E_z \mathbf{e}_z) dv \]
\[ = \frac{1}{4\pi} \varepsilon_i \nabla \cdot \mathbf{E} dv \]
\[ = \rho/\mathbf{E} dv \] (62)

2.A.6 Solute region with singularity

In elementary physics the total electrostatic force is presented as \( q \mathbf{E} \) given a singular charge of \( q \delta(x - x') \) in an external field. Apparently the electric field here does not include the singular self field, i.e. it is the field from all other fixed/mobile charges and polarized bound charges. This expression can be obtained with the force formulation based on the Maxwell stress tensor though it is not trivial. The difficulty exists because the stress tensor is a function of total electric field that contains the singular self field.

To start we note that for any singular charge source in an arbitrary solute region enclosed by \( S \), we can always isolate the singular charge in a small spherical volume element enclosed by \( S' \) with the charge in the center, as shown in Figure 2-3. Since there is no more singularity in the region enclosed by surface \( S \) and \( S' \), we know the total force acting on the \( S \) and \( S' \) surfaces is
\[ \int \rho' \mathbf{E} dv \] from eqn (62). Here, we have assumed that a smooth charge source, \( \rho' \), still exists in the region. Now the question is the total force acting upon the small spherical volume element enclosed by \( S' \). We cannot follow the previous strategy based on the Taylor expansion to compute the total force because the total electric field is singular at the center of \( S' \).

We proceed by separating the total electric field into two parts, \( \mathbf{E} = \mathbf{E}' + \mathbf{E}'' \). Here \( \mathbf{E}' \) is
the singular self field, the Coulombic field of the singular charge source; and $\mathbf{E}'$ is the regular interaction field, including the Coulombic field of all other charge sources and the reaction field. We further exploit the fact that the singular Coulombic field $\mathbf{E}'$ is spherically symmetrical to make the derivation manageable within a few pages. Indeed, the small spherical volume element was also chosen to exploit the symmetry. Since $\mathbf{E}'$ only has the normal component on $S'$, the force on $S'$ is also spherically symmetrical if there is no other field in the system. Thus self Coulombic force is zero, as expected.

Next we first assume that $\mathbf{E}'$ is a constant within $S'$ to obtain the 0th order approximation for the total force. It will become clear later that this is already a very good approximation because the charge density $[q\delta(x-x')]$ is infinitely large at $x'$. Without loss of generality, we set $\mathbf{E}'$ to be along the $z$-axis (Figure 2-4). And since the charge density $\rho'$, potential $\varphi$, and electric field $\mathbf{E}$ is smooth on the surface of $S'$, the Maxwell stress tensor in the local coordinate system of $(n, t)$, is

$$
\begin{bmatrix}
E_n^2 - \frac{1}{2}(E_n^2 + E_t^2) & E_n E_i \\
E_i E_n & E_t^2 - \frac{1}{2}(E_n^2 + E_t^2)
\end{bmatrix}
$$

(63)

Note that the contribution from the tangential force density is zero. And the force density on the surface element is

$$
P \cdot n = \begin{bmatrix}
\frac{1}{2}(E_n^2 - E_t^2), & E_i E_n
\end{bmatrix}
$$

(64)

In general both the normal and tangential force density components are nonzero as follows.
\[
F_n = \frac{1}{2}(E_S + E_r \cos \theta)^2 - \frac{1}{2}(E_r')^2 \sin \theta
\]
\[
F_i = E_r' \sin \theta (E_S + E_r' \cos \theta)
\]
(65)

with \( E_n = E_s + E_r' \sin \theta \) and \( E_i = E_r' \cos \theta \), where \( E_s \) is the electric field induced by the singularity, and \( E_r' \) is the mode of the approximated constant field \( E_r' \).

The force from \( F_n \) in the z-direction is
\[
F_n^Z = F_n \cos \theta
\]
\[
= r^2 \int_0^{2\pi} \int_0^{\pi} \sin \theta \cos \theta \left[ E_S^2 + \frac{1}{2} (E_r')^2 (\cos^2 \theta - \sin^2 \theta) + E_S E_r' \cos \theta \right] d\theta \]
(66)
\[
= \frac{4}{3} \pi r^2 E_S E_r'
\]

And similarly we can obtain the force from \( F_i \) in the z-direction
\[
F_i^Z = F_i \sin \theta
\]
\[
= r^2 \int_0^{2\pi} \int_0^{\pi} \sin \theta \sin \theta (E_S + E_r' \cos \theta) d\theta \]
(67)
\[
= \frac{8}{3} \pi r^2 E_S E_r'
\]

Finally the total force from \( F_n \) and \( F_i \) in the z-direction are
\[
F^Z = F_n^Z + F_i^Z
\]
\[
= 4\pi r^2 E_S E_r'
\]
(68)

Here the Gauss law for the singular point charge is used. Similarly we can obtain the total forces in the x- and y-direction and these are identically zero, apparently due to symmetry. Thus the total force is \( qE_r' \) that is along the direction of \( E_r' \). This result holds regardless of the radius of the spherical volume element due to the existence of the singular charge source.

With variable \( E_r' \), the derivation is more complicated when the Taylor expansion of \( E_r' \) at the center has to be used to represent \( E_r' \) on \( S' \). Nevertheless, if we represent \( E_r' \) as
\[ \mathbf{E}' = \mathbf{E}'(x') + O(\delta, \theta, \phi) \]  

(69)

where \( \lim_{\delta \to 0} O(\delta, \theta, \phi) = 0 \) collectively represents the first-order variations (as in Taylor expansions) of \( \mathbf{E}' \) on \( S' \) with respect to \( \mathbf{E}' \) at center \( x' \). It is straightforward to show that substitution of eqn (69) into eqn (68) does not make the finite term \( qE \) go away even if all terms containing \( O(\delta, \theta, \varphi) \) goes to 0 in the limit of \( \delta \to 0 \). This is because of the singular charge density of \( q\delta(x-x') \). In summary, our discussion shows that the total force of the spherical volume element can still be expressed as \( \int \rho' \mathbf{E} dv \) with the understanding that \( \rho' \) can be singular and \( \delta \to 0 \) does not contain the singular self field. Thus the force density can be universally written as

\[ \int \rho' \mathbf{E} dv \]  

(70)

in the solute region whether the charge density is singular or not.
References


30. W. Im, D. Beglov, and B. Roux, Continuum Solvation Model: Computation of


44. S. Miertus, E. Scrocco, and J. Tomasi, Electrostatic Interaction of a Solute with a Continuum - a Direct Utilization of Abinitio Molecular Potentials for the Prevision of


74. 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. Journal of


CHAPTER 3

A Charge Central Interpretation of the Full Nonlinear PB Equation:
Implications to Accurate and Scalable Modeling of Solvation Interactions

3.1 Summary

Continuum solvation modeling based upon the Poisson-Boltzmann equation (PBE) is widely used in structural and functional analysis of biomolecules. In this work we proposed a charge-central interpretation of the full nonlinear PBE electrostatic interactions. The validity of the charge-central framework, as formulated as a vacuum Poisson equation with effective charges, was first demonstrated by reproducing both electrostatic potentials and energies from the original solvated full nonlinear PBE. There are at least two benefits when the charge-central framework is applied. Firstly the convergence analyses show that the use of polarization charges allows a much faster converging numerical procedure for electrostatic energy and forces calculation for the full nonlinear PBE. Secondly the formulation of the solvated electrostatic interactions as effective charges in vacuum allows scalable algorithms to be deployed for large biomolecular systems. Here we exploited the charge-central interpretation and developed a particle-particle particle-mesh (P3M) strategy for the full nonlinear PB systems. We also studied the accuracy and convergence of solvation forces with the charge-view and the P3M methods. It is interesting to note that the convergences of both the charge-view and the P3M methods are
more rapid than the original full nonlinear PB method. Given the developments and validations documented here, we are working to adapt the P3M treatment of the full nonlinear PB model to molecular dynamics simulations.

### 3.2 Introduction

Life as we know it occurs in water so that inclusion of water is crucial to model the structures and functions of biomolecules accurately.\(^1\)\(^-\)\(^9\) Since the cost of explicitly including water molecules in computer models is very high, approximating water interactions implicitly or in a continuum manner is widely used in computational studies.\(^1\)\(^-\)\(^8\) A key component in such approaches is the electrostatic solvation modeling based on the Poisson-Boltzmann equation (PBE). A number of pioneer works have been published to study the solvation electrostatic effects in biomolecular functions.\(^1\)\(^-\)\(^8\), \(^20\) In these studies, numerical solution of the PBE is crucial because of the irregular shapes of biomolecules.\(^3\), \(^21\)\(^-\)\(^26\) Currently, the finite-difference method,\(^27\)\(^-\)\(^42\) finite-element method\(^43\)\(^-\)\(^51\) and boundary-element method\(^52\)\(^-\)\(^68\) are the most widely used numerical methods.

Due to its numerical nature, there are difficulties to incorporate PBE methods into molecular mechanics simulations, such as interpolating electrostatic forces, revmoing singular atomic charges,\(^35\), \(^43\), \(^69\)\(^-\)\(^79\) and achieving faster numerical convergence.\(^80\)\(^-\)\(^82\) Currently the numerical PBE methods are only applied in situations where a few fixed conformations are
involved, limiting the potentials of the methods. Efforts have been invested to improve these numerical issues of the PBE methods\textsuperscript{35, 43, 57, 69-79, 83-90} in order to facilitate the incorporation of the model in molecular mechanics simulations. The “virtual work” method, which computes forces according to the numerical derivative of potential energy, is apparently the benchmark for all analytical methods, but it is only realistic when molecules are treated as rigid bodies. Thus analytical calculation of solvation forces is necessary for in most situations. Two broadly different schemes have been developed to interpolate solvation forces. For the classical abrupt-transitioned two-dielectric models, multiple strategies have been proposed by Davis and McCammon,\textsuperscript{69} Che \textit{et al.},\textsuperscript{74} Bo \textit{et al.},\textsuperscript{76} Cai \textit{et al.}\textsuperscript{77} and most recently Li \textit{et al.}\textsuperscript{79}. These formulations were derived following different strategies and were found to be consistent. For the smooth-transitioned dielectric models, we have the ground-breaking strategy by Gilson \textit{et al.}\textsuperscript{73} Subsequent works by Im \textit{et al.}\textsuperscript{35} and Cai \textit{et al.}\textsuperscript{75} were shown to be consistent with that of Gilson \textit{et al.}\textsuperscript{73} though different strategies were proposed to enhance numerical stability and convergence in the later works. The numerical methods derived from these formulations are mostly adapted for the numerical solutions by the finite-difference method (FDM). Boundary-element method (BEM) is another promising approach to incorporate the PBE electrostatics into molecular mechanics simulations. The force calculation in a BEM calculation was first described by Zauhar.\textsuperscript{71} In addition, Cortis \textit{et al.}\textsuperscript{43} explored to compute the solvation force for their finite-element method (FEM) calculations, leading to the same formulation as that
of Zauhar.\textsuperscript{71}

In this study we address this issue in the context of full nonlinear PBE, which is useful in modeling of highly charged systems, i.e. nucleic acids or nucleic acid-binding proteins or ligands. Specifically we explored a charge central interpretation of the full PBE potentials and energies, which is to use the effective charge density to compute all electrostatic potentials, energies, and forces. There are at least two benefits in developing this strategy. Firstly the use of polarization charges allows a much faster converging numerical procedure for electrostatic energy and forces calculation for the Poisson’s equation and linear PBE as we have shown,\textsuperscript{75, 77, 82} and also for the full nonlinear PBE as shown below. Secondly the formulation of the solvated electrostatic interactions as effective charges in vacuum allows scalable algorithms to be deployed for large biomolecular systems. For example, we have explored to adapt the particle-particle particle-mesh (P3M) strategy for fully solvated electrostatic interactions as modeled by the full nonlinear PBE. P3M is a typical method for accurate and efficient calculation of Coulombic interactions of biomolecular systems.\textsuperscript{91-93} Apparently P3M cannot be used directly in calculating energy and forces based on the nonlinear PBE because there are heterogeneous dielectrics in the solute and solvent regions. In the following we first present the charge-central strategy to model the full nonlinear PBE potentials, energies, and forces. This is followed by numerical implementations and how to use P3M to balance accuracy and efficiency in applying nonlinear PBE to complex molecular systems.
3.3 Theory and Computational Details

3.3.1 Effective charge interpretation of Poisson-Boltzmann equation

The full nonlinear PBE for systems with continuum mobile ions can be expressed as

$$\nabla \cdot (\varepsilon \nabla \phi) = -4\pi \rho^f - 4\pi \sum_i e_i c_i \exp\left(-e_i \phi / k_B T\right) \lambda$$

(1)

Where $\phi$ is the potential, $\varepsilon$ is the dielectric constant, $e_i$ is the charge of ion type $i$, $c_i$ is the bulk number density of ion type $i$, $\lambda$ is the ion exclusion function, $k_B$ is the Boltzmann constant and $T$ is the absolute temperature. Following our developments for the Poisson equation\(^{75, 77, 82}\), the solution of the PBE can be cast into a vacuum Poisson equation with effective charges only, as shown in detail in Appendix 1. Briefly, consider a solute molecule with dielectric constant $\varepsilon_i$ surrounded by a solvent medium with dielectric constant $\varepsilon_o$, the solution of the original PBE satisfies a vacuum Poisson equation in the form of

$$\nabla^2 \phi = -4\pi \left( \frac{\rho^f}{\varepsilon_i} + \frac{\rho^m}{\varepsilon_o} + \rho^{pol} \right),$$

(2)

where

$$\phi = \phi^f_c + \phi^m_c + \phi_{RF}$$
$$\varepsilon_i \nabla^2 \phi^f_c = -4\pi \rho^f$$
$$\varepsilon_o \nabla^2 \phi^m_c = -4\pi \rho^m$$
$$\nabla^2 \phi_{RF} = -4\pi \rho^{pol}$$
$$\rho^m = \sum_i e_i c_i \exp\left(-e_i \phi / k_B T\right) \lambda$$

(3)

Here $\phi_c$ is the Coulomb potential generated by solute charges (denoted by superscript $f$) or
mobile ion charges (denoted by superscript \( m \)) in their respective uniform dielectric media throughout the whole space and \( \phi_{RF} \) is the total reaction field potential.

It is worth pointing out that the equivalence of eqn (2) to eqn (1) is not based on the superposition principle, which does not hold for nonlinear partial differential equations in general. In addition the effective source terms \( \rho^m \) and \( \rho^{pol} \) cannot be known before solving eqn (1) via standard numerical procedures. A more rigorous presentation is also possible via the integral form utilizing the Green’s theorem. However, it is more physically intuitive by following the discussion presented here.

There are at least two benefits when the charge-central framework is applied. Firstly the use of polarization charges allows a much faster converging numerical procedure for electrostatic energy and forces calculation for the full nonlinear PBE as shown below. Secondly many efficient numerical algorithms developed to speed up vacuum electrostatic/Coulombic field calculation can now be applied to the full PBE systems if the effective charge sources can be obtained, which will be discussed below.

3.3.2 Total electrostatic energy and forces of Poisson-Boltzmann systems

The total electrostatic energy of a Poisson-Boltzmann system can be written as

\[
G = \int \left( \rho \phi - \frac{1}{8\pi} \mathbf{D} \cdot \mathbf{E} - \Delta \Pi \lambda \right) dV
\]

\[
\Delta \Pi = kT \sum_i c_i (e^{-\phi_i/kT} - 1)
\]

Substitution of the PBE into eqn (4) leads to
\[ G = \int \left( \frac{\rho \phi}{2} - \frac{\rho^m \phi}{2} - \Delta \Pi \lambda \right) dv. \]  \hspace{1cm} (5)

Eqn (5) suggests that \( G \) be decomposed into three parts as \( G = G^f + G^m + G^{\Pi} \). Here \( G^{\Pi} \) is the entropic term due to the excess osmotic pressure, apparently not due to charge-charge interactions so that no further treatment is attempted below. In contrast, \( G^f \) and \( G^m \) can be reformulated according to the effective charge view presented above.

Substitution of \( \phi = \phi_C^f + \phi_C^m + \phi_{RF} \) into the first term of eqn (5) leads to

\[ G^f = \frac{1}{2} \int \rho^f (\phi_C^f + \phi_C^m + \phi_{RF}) dv \]  \hspace{1cm} (6)

Apparently \( G^f \) can be decomposed into three parts:

\[
\begin{align*}
G^f &= G_C^f + G_m^f + G_{pol}^f \\
G_C^f &= \frac{1}{2} \int \rho^f \phi_C^f dv \\
G_m^f &= \frac{1}{2} \int \rho^f \phi_C^m dv \\
G_{pol}^f &= \frac{1}{2} \int \rho^f \phi_{RF} dv
\end{align*}
\]  \hspace{1cm} (7)

Thus the electrostatic energy associated with the fixed charges on the atoms is due to interactions with other atomic charges, mobile ion charges, and overall induced polarization charges.

Similarly the second term of eqn (5) can be written as follows upon the substitution of \( \phi = \phi_C^f + \phi_C^m + \phi_{RF} \)
\[ G^m = G^m_c + G^m_m + G^m_{pol} \]
\[ G^m_c = -\frac{1}{2} \int \rho^m \phi^m_c \, dv = -\frac{e_o}{2\epsilon_i} \int \rho^f \phi^m_c \, dv \]
\[ G^m_m = -\frac{1}{2} \int \rho^m \phi^m_m \, dv \]
\[ G^m_{pol} = -\frac{1}{2} \int \rho^m \phi^m_{RF} \, dv \]

(8)

Thus the electrostatic energy associated with the mobile ion charges is due to interactions with atomic charges, other mobile ion charges, and overall induced polarization charges. Overall the conclusions in eqn (7) and (8) are consistent with the Coulomb’s law once the effective charge view of the PBE system is used.

We now turn to the formulation of electrostatic forces. Given the discussion in Refs 75, 77, the force density is the divergence of the Maxwell stress tensor for systems without singularities or discontinuities

\[ \nabla \cdot \mathbf{P} = \frac{\partial}{\partial x} (i \cdot \mathbf{P}) + \frac{\partial}{\partial y} (j \cdot \mathbf{P}) + \frac{\partial}{\partial z} (k \cdot \mathbf{P}) \]

\[ = \rho^f \mathbf{E} - \frac{1}{8\pi} \mathbf{E}^2 \nabla \epsilon - \Delta^\Pi \nabla \lambda \]

(9)

which is consistent with the formulation derived by the variational strategy by Gilson et. al. 73

Relying on an integral approach, Li et al. derived the total electrostatic forces throughout a system with singularities. For the solute region with or without singularity, it can be shown that the force density is universally

\[ \mathbf{f} = \rho^f \mathbf{E} \]

(10)

The ionic boundary force (IBF) can also be computed using the integral approach, which is
simply the difference in ionic pressure, expressed as

\[ f_{\text{IDF}} = (P_o - P_i) \cdot n = -\Delta \Pi, \]  

(11)

where \( P_o \) and \( P_i \) are the corresponding outside and inside stress tensors, respectively, on the surfaces parallel the Stern layer. The integral approach can also be applied to compute the dielectric boundary force \((DBF)\) for the piece-wise constant dielectric model as

\[ f_{\text{DBF}} = (P_o - P_i) \cdot n = \frac{1}{4\pi} \left[ (\varepsilon_o E_{o\text{an}}^2 - \frac{1}{2} \varepsilon_o E_o^2) - (\varepsilon_i E_{i\text{an}}^2 - \frac{1}{2} \varepsilon_i E_i^2) \right] n, \]  

(12)

where \( n \) is the outward-directed normal unit vector of the molecular surface, and \( P_i \) and \( P_o \) are the inside and outside stress tensors, respectively, on the surfaces parallel to the dielectric interface. \( E_o \) and \( E_i \) are the electric fields on the two sides of the solute, respectively, and \( E_{i\text{an}} \) and \( E_{o\text{an}} \), are the electric field components on the \( n \) direction of \( E_o \) and \( E_i \), respectively. This agrees with the conclusion by Davis and McCammon.\(^{69}\)

We next turn to the electrostatic forces on atoms based on the charge-central interpretation. Similar to the expression of electrostatic energies, it is the summation of the Coulomb and reaction field forces, so that it can also be written as the pairwise charge-charge interaction format. For the same reason as in the discussion of energies, \( \int f dv \) (eqn (10)) can be decomposed into three parts as following, given \( E = E_C^f + E_m^o + E_{RF} \)

\[ \int \rho^f E dv = F_C^f + F_m^f + F_{pol}^f \]

\[ F_C^f = \int \rho^f E_C^f dv \]

\[ F_m^f = \int \rho^f E_m^o dv \]

\[ F_{pol}^f = \int \rho^f E_{RF} dv \]  

(13)
This shows that both Coulomb and reaction field forces can be grouped into interactions with atomic charges, mobile ion charges, and overall induced polarization charges. In summary the conclusions in eqn (13) are consistent with the Coulomb’s law once the effective charge view of the Poisson-Boltzmann system is used.

Cai et al. proposed a new DBF formulation based on the concept of boundary polarization charges as

\[ f_{DBF} = \frac{1}{2} \rho_{pol} \frac{|D|^2}{D_n} n \]  \hspace{1cm} (14)

where \( \rho_{pol} \) is the boundary polarization charge density, \( D \) is the electric displacement vector, and \( D_n \) is the normal component of \( D \). This “charge-central” method was designed for smooth-transitioned dielectric models. Nevertheless, the charge-central method appears to be in the highly similar mathematical form for the abrupt-transitional dielectric models.\(^7\) Due to the typical high value of dielectric constant of water versus that of the solute in molecular mechanics force fields, the tangential surface field is often extremely small when compared with the normal surface field. When we apply this normal field approximation, \( D = D_n \). Thus the DBF force for both smooth-transitioned and abrupt-transitioned dielectric models can be approximated as

\[ f_{DBF} = \frac{1}{2} \rho_{pol} D_n n \]  \hspace{1cm} (15)

The ionic boundary force (IBF) is simply as

\[ f_{IBF} = -\Delta \Pi \cdot n \]  \hspace{1cm} (16)
where \( \Delta \Pi_i = kT \sum_i c_i (e^{-q_i \phi / kT} - 1) \) is the osmotic pressure outside of the Stern layer.

### 3.3.3 Numerical calculation of electrostatic energy and forces

**Discretized Charges** According to eqn (2), the effective charges are composed of \( \rho^f \), \( \rho^m \) and \( \rho^{pol} \). \( \rho^f \) is simply atomic charges. Since it is usually singular point charges, we denote it as \( q^f \). \( \rho^m = \sum_i q_i c_i \exp(-q_i \phi / k_B T) \lambda_i \) is the ionic charge density according to eqn (3). Finally \( \rho^{pol} \) is the polarization charge density at the dielectric boundaries, which can be computed as \( \rho^{pol} = \frac{1}{4\pi} \nabla \cdot E \) for the smooth-transition dielectric treatment, assuming there is no atomic charges in any dielectric boundary region. For the abrupt-transition dielectric treatment, the surface charge density can be computed as \( \sigma^{pol} = \frac{1}{4\pi} \left( E_{o\xi} - E_{i\xi} \right) \), where \( E_{o\xi}, E_{i\xi} \) are the normal component of the electric field on the solvent and solute side, respectively.

Next we introduce \( Q^f \), \( Q^m \) and \( Q^{pol} \) to denote the charges mapped onto the grid from \( \rho^f \), \( \rho^m \) and \( \rho^{pol} \), respectively. The following notations are introduced to accommodate the use of different potentials in different regions. \( \Omega_i \) is used to denote the solute grids with all of their six neighbor grids also within the solute region. \( \Omega_o \) is used to denote the solvent grids with all of their six neighbor grids within the solvent region. \( \Gamma_i \) is used to denote the solute grids with one or more of their six neighbor grids in the solvent region. \( \Gamma_o \) is used to denote the solvent grids with one or more of their six neighbor grids in the solute region.

1. \( Q^f \) is obtained by a standard tri-linear mapping of atomic charges.
\[
Q_{i,j,k}^f = Q^f(x_i, y_j, z_k) = \sum W(x_\alpha - x_i, y_\alpha - y_j, z_\alpha - z_k) q_\alpha^f
\]

which is a sum over all the charged particles within the adjacent cubic grid \((i, j, k)\), \((x_\alpha, y_\alpha, z_\alpha)\) and \(q_\alpha^f\) are the position and charge of atom \(\alpha\) and \(W(x_\alpha - x_i, y_\alpha - y_j, z_\alpha - z_k)\) is defined as

\[
W(x-x_i, y-y_j, z-z_k) = \begin{cases} 
\left(1 - \frac{|x-x_i|}{h}\right)\left(1 - \frac{|y-y_j|}{h}\right)\left(1 - \frac{|z-z_k|}{h}\right), \\
0, & \text{max}(|x-x_i|,|y-y_j|,|z-z_k|) \leq h \\
\text{otherwise}
\end{cases}
\]

(2) \(Q^n\) is directly derived from the charge density from the grid point

\[
Q_{i,j,k}^n = \rho^n h^3 = kT h^3 \sum e_i e_j (e^{-e_i \phi_{i,j,k}/kT} - 1)
\]

(3) \(Q^{pol}\) represents the induced charge on grid points nearby dielectric boundaries defined by eqn (3), which can be obtained as

\[
Q_{i,j,k}^{pol} = \frac{h}{4\pi} (6 \phi_{i,j,k+1} - (\phi_{i-1,j,k} + \phi_{i-1,j,k+1} + \phi_{i,j-1,k} + \phi_{i,j-1,k+1} + \phi_{i,j,k-1} + \phi_{i,j,k+1})) - Q_{i,j,k}^f / \varepsilon_i - Q_{i,j,k}^n / \varepsilon_o
\]

where the subtraction of atomic charges and mobile ion charges in the last two terms is necessary because it is possible to observe mapped atomic charges/mobile ion charges on boundary grid points in the process of the finite-difference discretization.

**Finite-Difference Energies** We now turn to numerical computation of electrostatic energies via the finite-difference method, i.e. the particle-mesh method. For the sake of clear presentation, we assume all individual potentials \((\phi^f, \phi^n, \phi^{RF})\) on the grid are already known. The detailed procedures are presented in Computational Details below. According to eqn (5), the total electrostatic energy can be computed as
As discussed in section B, the total electrostatic energy can be computed in an alternative way as a charge-view method if we focus on each of the first two terms in eqn (21).

For the first term, i.e. energies due to interactions with the fixed atomic charges, our analysis in Section B shows that $G^f = G_C^f + G_m^f + G_{pol}^f$. Since $\phi_C^f$ satisfies eqn eqn (3). Coulombic energy $G_C^f$ can be computed as

$$G_C^f = \frac{1}{2} \epsilon_i \sum_{i,j,k}^f \sum_{i',j',k'}^f \frac{Q_{i,j,k}^f}{\Delta x, \Delta y, \Delta z} Q_{i',j',k'}^f (\Delta x, \Delta y, \Delta z) g_{i,i',j,j',k,k'}^f$$

where $\epsilon_i$ is the solute interior dielectric constant, $Q_{i,j,k}^f$ and $Q_{i',j',k'}^f$ are grid charges interpolated by atoms on grid $(i, j, k)$ and $(i', j', k')$, respectively, $(\Delta x, \Delta y, \Delta z)$ is the distance vector between two grid charges in the grid unit and

$$g_{i,i',j,j',k,k'}^f = \frac{1}{\pi^2} \int_{0}^{\pi} \int_{0}^{\pi} \int_{0}^{\pi} \frac{\cos(w \Delta x) \cos(u \Delta y) \cos(v \Delta z)}{\sin^2(w/2) + \sin^2(u/2) + \sin^2(v/2)} \times dw du dv$$

is the finite-difference Green’s function as shown by Luty and McCammon. According to eqn (7) and eqn (8), $G_m^f$ can be computed together with the term $G_C^m$ in $G_m^m$, i.e. $G_m^f + G_m^m = (1 - \epsilon_0) G_m^f$, and $\phi_C^m$ satisfies eqn (3). $G_m^f + G_C^m$ can then be computed in the following way.
\[ G'_m + G'_C = \begin{cases} 
\frac{1}{2} \left( \frac{1}{\varepsilon_0} - \frac{1}{\varepsilon_i} \right) \sum Q'_{i,j,k} \phi'^m_{i,j,k} & \text{particle mesh} \\
\frac{1}{2} \left( \frac{1}{\varepsilon_0} - \frac{1}{\varepsilon_i} \right) \sum \sum Q'_{i,j,k} g(\Delta x, \Delta y, \Delta z) Q''_{i,j,k} & \text{charge view} 
\end{cases} \]  \tag{23}

As for \( G'_{pol} \), we have

\[ G'_{pol} = \begin{cases} 
\frac{1}{2} \sum Q'_{i,j,k} \phi'^{RF}_{i,j,k} & \text{particle mesh} \\
\frac{1}{2} \sum \sum Q'_{i,j,k} g(\Delta x, \Delta y, \Delta z) Q''_{i,j,k} & \text{charge view} 
\end{cases} \]  \tag{24}

since \( \phi'^{RF} \) satisfies eqn (3).

For the second term, i.e. energy due to interactions with the mobile ionic charges, our analysis in Section B shows that \( G'' = G'_m + G''_m + G''_{pol} \). Note that \( G''_C \), the atom-ion interaction energy, has been taken care of in eqn (23) along with energies associated with atomic charges. Since \( \phi'^m_C \) satisfies eqn(3), \( G''_m \) can be computed in the following way

\[ G''_m = \begin{cases} 
-\frac{1}{2\varepsilon_0} \sum Q''_{i,j,k} \phi'^m_{i,j,k} & \text{particle mesh} \\
-\frac{1}{2\varepsilon_i} \sum \sum Q''_{i,j,k} g(\Delta x, \Delta y, \Delta z) Q''_{i,j,k} & \text{charge view} 
\end{cases} \]  \tag{25}

As for \( G''_{pol} \), we have

\[ G''_{pol} = \begin{cases} 
-\frac{1}{2} \sum Q''_{i,j,k} \phi'^{RF}_{i,j,k} & \text{particle mesh} \\
-\frac{1}{2} \sum \sum Q''_{i,j,k} g(\Delta x, \Delta y, \Delta z) Q''_{i,j,k} & \text{charge view} 
\end{cases} \]  \tag{26}
since $\phi^{RF}$ satisfies eqn (3).

Thus the total electrostatic energy in the Poisson-Boltzmann’s equation is finally written in pairwise charge-charge interactions except $G^{\Pi}$. In summary the total electrostatic energy is given by

$$G = G_C^f + G_m^f + G_{pol}^f + G_C^m + G_m^m + G_{pol}^m + G^{\Pi}$$

(27)

after computing all energy terms in eqns (22) – (26).

**Finite-Difference Forces** As shown above, the total energy can be decomposed into several charge-charge interaction terms by using effective charges, and each term can then be computed by either particle mesh or charge view methods. The same strategy can also be used to compute the electrostatic forces.

As shown in eqn (13), since $E_C^f$ satisfies eqn (3), it can be computed in the following way

$$F_{C,i,j,k}^f = \begin{cases} 
\frac{Q_{i,j,k}^f E_{i,j,k}^f}{\varepsilon_i} & \text{particle mesh} \\
\frac{1}{\varepsilon_i} \sum_{(i,j,k)\neq(i',j',k')} Q_{i',j',k'}^f G(\Delta x, \Delta y, \Delta z) Q_{i,j,k}^f & \text{charge view}
\end{cases}$$

(28)

where $\varepsilon_i$ is the solute interior dielectric constant, $Q_{i,j,k}^f$ and $Q_{i',j',k'}^f$ are grid charges interpolated by atoms on grid $(i,j,k)$ and $(i',j',k')$, respectively, $G(\Delta x, \Delta y, \Delta z)$ is the finite-difference gradient of $g(\Delta x, \Delta y, \Delta z)$. As for $F_{m}^f$, since $E_C^m$ satisfies eqn (3), we have
\[
F_{m,j,k}^f = \begin{cases} 
\frac{Q_{i,j,k}^f E_{i,j,k}^m}{\varepsilon_o} & \text{particle mesh} \\
\frac{1}{\varepsilon_o} \sum_{i',j',k'} Q_{i',j',k'}^m G(\Delta x, \Delta y, \Delta z) Q_{i,j,k}^f & \text{charge view}
\end{cases}
\]  

(29)

where \( \varepsilon_o \) is the solvent exterior dielectric constant. As for \( F_{pol}^f \), since \( E_{RF} \) satisfies eqn (3) we have

\[
F_{pol,j,k}^f = \begin{cases} 
Q_{i,j,k}^f E_{i,j,k}^{BF} & \text{particle mesh} \\
\sum_{i',j',k'} Q_{i',j',k'}^{pol} G(\Delta x, \Delta y, \Delta z) Q_{i,j,k}^f & \text{charge view}
\end{cases}
\]  

(30)

Combining eqn (28) to (30), the total electrostatic force on grid \((i,j,k)\) is given by

\[
F_{i,j,k}^f = F_{c,j,k}^f + F_{m,j,k}^f + F_{pol,j,k}^f
\]  

(31)

Finally the grid force is then interpreted onto the atoms as

\[
F_{i,j,k}^f(x_\alpha, y_\alpha, z_\alpha) = \sum W(x_\alpha - x_i, y_\alpha - y_j, z_\alpha - z_k) F_{i,j,k}^f
\]  

(32)

Finally, according to eqn (15) and eqn(16), the dielectric boundary forces can be computed as

\[
F_{DBF,i,j,k} = \frac{1}{2} Q_{i,j,k}^{pol} D_{n,i,j,k} \mathbf{n} = \frac{1}{2} Q_{i,j,k}^{pol} E_{n,i,j,k} \mathbf{n}
\]  

(33)

and the ionic boundary force (IBF) is simply discretized as

\[
F_{IBF,i,j,k} = kT \sum_{l,j,k} c_{l,j,k} (e^{-q_{l,j,k} \phi_{l,j,k} / kT} - 1) \mathbf{n}
\]  

(34)

at grid edges flanked by one ion excluded grid node and one ion occupied grid node.

3.3.4 P3M implementation in calculations of energy and forces
In most current PBE implementations, the particle-mesh method is used to compute the system electrostatic energies, at least for the reaction field energies, i.e. the solvation electrostatic energies. Since particle-mesh methods involve mapping a system of particles onto grid nodes, the procedure apparently introduces discretization error in energy calculations, particularly for interaction energies between particles/charges that are close to each other. Besides the apparent issue of discretizing atomic charges onto a finite-difference grid (i.e. displace of charges from their analytical positions), another major reason behind the discretization error is due to the difference between analytical Green’s function and finite-difference Green’s function\(^8\) as we analyzed previously.\(^9\) A simple solution to reduce the discretization error is to use the analytic Green’s function \((1/r)\) to replace finite-difference Green’s function \((g(\Delta x,\Delta y,\Delta z))\) and to keep the particle charges at their original positions for eqn (22)-(26) and eqn (28)-(30). However, a brute-force pairwise summation approach is highly inefficient except for very simple/small systems due to the scaling of pairwise sums. This is the same problem that the particle-simulation community has addressed in the past. On the other hand, the difference becomes extremely small when the charges are separated by long distance (in terms of grid spacing). This motivated us to introduce the particle–particle particle–mesh (P3M) strategy method, initially in the computation of Coulomb interactions in the Poisson equation.\(^9\) Given the theoretical and numerical developments presented above, we are in the position to extend the method to the total electrostatic energy (i.e. both Coulomb and reaction
field energies) in the full nonlinear PBE method.

According to eqn (27), the full PBE electrostatic energy can be split into six terms with each formulated as a pairwise summation of relevant effective charges, or its equivalent finite-difference (FD) summation. This is equivalent to the original full PBE finite-difference electrostatic energy, eqn (21). Apparently the FD electrostatic energy (denoted as $G^{FD}$ below to highlight it is its particle-mesh nature) is an accurate approximation of pairwise summation at long distance, but not at short distance. Thus to reduce the FD discretization error, we can replace the particle-mesh method with the particle-particle method for short-range interactions, with a predefined cutoff distance.\(^\text{93}\) This suggests that a P3M strategy can be used in computing the pairwise sums in eqn (27) to balance accuracy and efficiency. Given our use of pairwise sums of effective charge interactions, $G^{FD}$, can be split into three parts as

$$G^{FD} = G^{FD}_{\text{self}} + G^{FD}_{\text{short}} + G^{FD}_{\text{long}} \quad (35)$$

Here $G^{FD}_{\text{self}}$ is the self-energy, i.e., the energy due to interactions of grid charges within the grid charges of one single atom, a pure artifact of the FD approach and must be removed.\(^\text{93}\) $G^{FD}_{\text{short}}$ is the electrostatic energy for short-range interactions, and $G^{FD}_{\text{long}}$ is the electrostatic energy for long-range interactions. The partition of these two groups of interactions apparently needs a cutoff distance ($R_{\text{cut}}$).

The FD self-energy and short-range FD energy can be computed by summing up all relevant pairwise interactions, i.e. $G_{i,j,k/l,j',k'}$, among all grid charges involved, as
\[ G_{i,j,k|i',j',k'}(\Delta x, \Delta y, \Delta z) = f Q_{i,j,k}g(\Delta x, \Delta y, \Delta z)Q_{i',j',k'}, \]  

(36)

where \( Q_{i,j,k} \) is the grid charge at grid \((i,j,k)\), and \( Q_{i',j',k'} \) is the grid charge at grid \((i',j',k')\). \( f \) is the coefficient defined in eqn (22)-(26) depending on if \( Q \) is atomic charge, polarization charge, or ionic charge. Here the distance \((r)\) between any pair of charges satisfies

\[ r = \sqrt{(x - x')^2 + (y - y')^2 + (z - z')^2} \leq R_{cut} \]  

(37)

\((x,y,z)\) and \((x',y',z')\) are the coordinates of two charges at \((i,j,k)\) and \((i',j',k')\). When grid charges are from the same atom, eqn (36) gives the FD self energy.

Therefore the full PBE electrostatic energy, \( G \), can be computed via the P3M strategy by subtracting the FD self energy \( (G_{FD}^{FD} - G_{FD}^{self}) \) and FD short-range energy \( (G_{FD}^{FD} - G_{FD}^{short}) \) from the FD electrostatic energy and adding back the analytical short-range energy \( (G_{FD}^{short} + G_{FD}^{short}) \) by using the Coulomb’s law for the short-range charge-charge pairs

\[ G = G^{FD} - G^{FD}_{self} - G^{FD}_{short} + G^{ana}_{short} \]

\[ G^{FD}_{self} + G^{FD}_{short} = \sum_{i,j,k} \sum_{i',j',k'} f Q_{i,j,k}g(\Delta x, \Delta y, \Delta z)Q_{i',j',k'} \]

\[ G^{ana}_{short} = \frac{1}{2\varepsilon_i} \sum_{l} \sum_{n\neq l} \frac{q_l q_n}{r_{ln}} + \frac{1}{2\varepsilon_i} \sum_{l} \sum_{i,j,k} \frac{q_l Q^m_{i,j,k}}{r_{li,j,k'}} + \frac{1}{2} \sum_{l} \sum_{i,j,k} \frac{q_l Q^pol_{i,j,k}}{r_{li,j,k}} + \frac{1}{2\varepsilon_o} \sum_{i,j,k} \sum_{i',j',k'} Q^m_{i,j,k}Q^m_{i',j',k'} \sqrt{r_{ii',jj',kk'}} |Q^pol_{i,j,k}Q^pol_{i',j',k'} - \frac{1}{2} \sum_{i,j,k} \sum_{i',j',k'} Q^m_{i,j,k}Q^pol_{i',j',k'} \sqrt{r_{ii',jj',kk'}} | 

\]

(38)

where the double summation in \( G^{ana}_{short} \) includes only pairs within the preset cutoff distance \( R_{cut} \); \( q_l \) and \( q_n \) are atomic charges of atom \( l \) and \( n \), respectively; \( \varepsilon_i \) is the solute interior dielectric constant and \( \varepsilon_o \) is the solvent dielectric constant; and \( r \) is the distance between charge positions defined in eqn (37).

87
It is worth mentioning that for a pair longer than the cutoff the relative error between $g(\Delta x, \Delta y, \Delta z)$ and $1/\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$ is often not trivially small. For example, if we choose a cutoff distance as 14 grids, the relative error between the two is $1.1 \times 10^{-3}$. However, since pairs in shorter distance contribute the dominant part of energy and forces, the errors are usually much smaller than the error as shown here. Otherwise, a cutoff distance much longer would have to be used and lead to a very inefficient P3M method.

3.3.5 Treatment of salt-related terms

According to eqn (6) to (26), the total electrostatic energy in the full Poisson-Boltzmann’s equation can be written as

$$G_{\text{tot}} = G_C^f + G_m^f + G_{\text{pol}}^f + G_C^m + G_m^m + G_{\text{pol}}^m + G^\Pi$$  \hspace{1cm} (39)

The first six terms on the right-hand side are all pairwise charge-charge interactions except the last term, $G^\Pi$. Now considering the salt-related term $G_m^f + G_C^m + G_m^m + G_{\text{pol}}^m + G^\Pi$, Notice that computation of $G_m^m$ with the charge view method needs a pair-wise summation of all grid points (N), so that the time scales quadratically with the grid points as $O(N^2)$, which is very computationally demanding, while with the original finite-difference method (particle mesh method) the time scales only linearly with the grid nodes as $O(N)$. As shown later, salt-related energy is only a small portion of the total electrostatic energy and it also converges very rapidly with reducing grid spacing. Therefore in this development, the original finite-difference approach is used for the ionic energy terms for all tested methods. Only $G_C^f$ and $G_{\text{pol}}^f$ are treated.
with P3M. Eqn (38) finally becomes

\[
G = G^{FD} - G^{FD}_{self} - G^{FD}_{short} + G^{ana}_{short}
\]

\[
G^{FD}_{self} + G^{FD}_{short} = \frac{1}{2\varepsilon} \sum_i \sum_{\ell} \sum_{k,l,k} Q^f_{i,j,k} g(\Delta x, \Delta y, \Delta z) Q^f_{i,j,k} + \frac{1}{2} \sum_{i,j,k} \sum_{i,j,k} Q^f_{i,j,k} g(\Delta x, \Delta y, \Delta z) Q^{pol}_{i,j,k}
\]

\[
G^{ana}_{short} = \frac{1}{2\varepsilon} \sum_i \sum_{\ell} \sum_{k,l,k} q_i q_n r_{ln} + \frac{1}{2} \sum_{i,j,k} \sum_{i,j,k} q_i Q^{pol}_{i,j,k}
\]

Finally, according to eqn (13) and eqn (28)-(31), the force term \( F = \int \rho^f E \ dv \) can also be computed by the P3M strategy, the force on a given atom \( l \) is

\[
F^f_l = F^{FD}_{short,l} + F^{FD}_{short,l} + F^{ana}_{short,l}
\]

\[
F^{FD}_{short,l} = \frac{1}{\varepsilon} \sum_i \sum_{\ell} \sum_{k,l,k} Q^f_{i,j,k} G(\Delta x, \Delta y, \Delta z) Q^f_{i,j,k} + \sum_{i,j,k} \sum_{i,j,k} Q^f_{i,j,k} G(\Delta x, \Delta y, \Delta z) Q^{pol}_{i,j,k}
\]

\[
F^{ana}_{short,l} = \frac{1}{\varepsilon} \sum_i \sum_{\ell} \sum_{k,l,k} q_i q_n r_{ln} + \sum_{i,j,k} \sum_{i,j,k} q_i Q^{pol}_{i,j,k}
\]

Here in \( F^{FD}_{short,l} \), \((i,j,k)\) is looping over all the grid charges maps from atom \( l \), and \( F^{FD}_{short,l} \) and \( F^{ana}_{short,l} \) include only pairs within the preset cutoff distance \( R_{cut} \).

### 3.4 Results and Discussion

#### 3.4.1 Consistency of charge-view equation and original full nonlinear PB equation

In the following we first demonstrate that the vacuum Poisson equation in the effective charges, eqn (2) and eqn (17)-(20), can reproduce the potential from the original full PBE, Figure 3-1 shows the correlation between the charge-view potentials computed by eqn (2) and
eqn (17)-(20) and the potential computed by eqn (1) for a single polyAT DNA with or without salt. Particularly, the full PB or Poisson equation was first solved for the polyAT system with given dielectric constants distribution and ion concentrations. The effective charges on all grid grids were then computed according to eqn (17)-(20). We then used the effective grid charges to compute the potential distribution by solving the Poisson equation in vacuum as eqn (2). Apparently the same finite-volume discretization and numerical solver was used. In this comparison, we purposely used different combination of solute/solvent dielectrics to highlight that the formulation is independent of the exact dielectrics used in modeling the original system in the full PB equation. Overall it can be seen that the charge-view method agrees with the original full PB equation, with relative error < $10^{-9}$ for all grid potentials.

Figure 3-1 Correlations between grid potentials (kcal/mol-e) computed by the charge view
and the full Poisson or PBE methods. A polyAT DNA was modeled with the full PB or the Poisson equations. The charge view potentials were computed with Poisson with effective charges. The grid spacing is 0.5 Angstrom. Top: Poisson equation. The inside relative dielectric constant is 1.0 (epsin=1.0) and outside relative dielectric constant is 80.0 (epsout=80.0). The root-mean-squared (rms) relative deviation is 9.8×10⁻¹¹. Bottom: PB equation with ionic strength of 1M (istrng=1000). The inside relative dielectric constant is 4.0 (epsin=4.0), and the outside relative dielectric constant is 80.0 (epsout=80.0). The rms relative deviation is 1.0×10⁻⁹.

3.4.2 Consistency between pairwise charge-view energies and the full nonlinear PB energies

Given that the charge-view equation reproduces the full PBE potential, we next investigated the possibility to reproduce the full PBE electrostatic energy (eqn (21)) with the pairwise summation as outlined by eqn (22)-(26) with the charge-view framework. Figure 3-2 shows the correlation between the finite-difference charge-view energies and the original finite-difference electrostatics energies using both Poisson equation and full PBE for a large set of nucleic acid PDB structures. Overall the charge-view energies reproduce both Poisson and full PBE energies very well with relative errors < 2×10⁻⁷.
Figure 3-2 Correlations between electrostatic energies (kcal/mol) computed by the charge view and the full Poisson or PBE methods. A test set of 283 different nucleic acids was modeled with the full PB or the Poisson equations. The charge view energies were computed with the finite-difference Green’s function as eqn (22) – (26). The grid spacing is 1.0 Angstrom. No cutoff is used. Left: Energies from the two methods. Right: Relative Error between this two methods. Top: Poisson equation with the same set up as in Figure 1. The rms relative deviation is $4.0 \times 10^{-8}$. Bottom: PB equation with the same set up as in Figure 1. The rms relative deviation is $1.7 \times 10^{-7}$.

3.4.3 Convergence of charge-view energies

Next we tested the convergence behavior of the pairwise charge-view strategy comparing with the original full PBE for total electrostatic energy calculation. Here we focus on the convergence behavior of the solvation free energy only since it is the component most susceptible to discretization errors. As shown in Figure 3-3, the charge-view and the full PBE energies are compared in two different situations: (1) the polarization charges are mapped onto the molecular surface (top) and (2) the polarization charges remain on the grid points. And we
also used a nonlinear function \( y = a + bh^c \) to fit the convergence data. Thus \( y|_{h=0} \) is regarded as the “converged” energy. Table 3-1 shows the fitted coefficients for different strategies. For the original full PBE energies, the convergent value (in kcal/mol) is \(-7.7915\) with relative RMS error of (in kcal/mol) \(0.0011\).

![Graph showing convergence of reaction field energies](image)

**Figure 3-3 Convergence of the reaction field energies (kcal/mol) versus grid spacing (Ångstrom).** The AT base dimer was modeled with the full PBE (epsout=80.0, epsin=4.0, istrng=1000). Reaction field energies are computed with the full PBE (black), charge-view (red) strategies. Top: the polarization charges are mapped onto the molecular surface. Bottom: the polarization charges remain on the boundary grid points. For the full PBE and the charge-view methods, the convergence trend lines are fitted with respect to grid spacing in the form of \( y = a + bx^c \). Since the deviation of the field view and charge view without charge mapping (bottom) are large at course grids, only data with grid spacing equal or less than 0.3 Å were used when fitting those curves. See Table 3-1 for all fitted parameters.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>( r^2 )</th>
<th>RMSrD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full PBE</td>
<td>-7.7915</td>
<td>-1.9369</td>
<td>1.4284</td>
<td>0.9966</td>
<td>0.0011</td>
</tr>
<tr>
<td>Charge view (map)</td>
<td>-7.7935</td>
<td>0.8683</td>
<td>3.5828</td>
<td>0.9768</td>
<td>0.00062</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>Charge view (nomap)</td>
<td>-7.7838</td>
<td>-1.6670</td>
<td>1.3078</td>
<td>0.9976</td>
<td>0.00087</td>
</tr>
</tbody>
</table>

**Table 3-1** Fitted nonlinear trend lines for the full PBE electrostatic solvation energy (kcal/mol) versus grid spacing using different methods. The nonlinear model to be fitted is \( y = a + bx^c \) where \( x \) is the grid spacing. RMSrD: root-mean squared relative residue between the fitted values and input values. map: polarization charges are mapped onto the atomic surface. nomap: polarization charges are not mapped onto the atomic surface.

In the top figure, the polarization charges are mapped onto the atomic surface, the fitted “converged” energy by the charge-view method is -7.7935 kcal/mol, which is within the RMS error of the converged energy for the full PBE method, indicating that the charge-view method and the full PBE method converge to the same value within fitting uncertainty. In addition, it is apparent that the convergence curve of the charge-view energy is almost flat, indicating that this method offers a good estimation of the full PBE energy at tested coarse grid spacing. This supports our conclusion that the use of polarization charges allows a much faster converging numerical procedure for electrostatic energy calculation as we have shown for the Poisson’s equation and linear PB equation.\(^75, 77, 82\)

In the bottom figure, the polarization charges are set to remain at the grid points. It can be seen that the fitted “converged” energy by the charge-view method is -7.7838 kcal/mol, which is also within the RMS error of the converged energy for the FDPB method. However, the charge-view energy also converges similarly as the full PBE energy, so that its energy is no longer a good approximation of the converged energy at coarse grid spacing if the polarization charges remain at the grid points.
3.4.4 Convergence and treatment of ionic energy terms

The full PBE electrostatic energy is composed of the reaction field energy and the salt-related energy. As discussed in the subsection E of Theory and Computational Details, salt-related energy is only a small portion of the total electrostatic energy and it also converges very rapidly. We studied a typical-sized nucleic acid (PDBID: 420D), a 1026-atom solute. The convergence of salt-related energy of 420D by the full PBE method is shown in Figure 3-4. The data was first fitted by nonlinear curve $y = a + bh^c$, given a predicted energy $y=1.865$ kcal/mol at $h=0$, which is less than $10^{-3}$ of the total electrostatic solvation free energy of $4.7 \times 10^3$ kcal/mol. Figure 3-4 also shows that the salt-related energy converges very well at typical grid spacing values (1/16 to 1/2 Angstrom). The convergence error is $<10^{-4}$ of the total electrostatic solvation free energy at grid spacing values less than 1/2 Angstrom and is $<10^{-5}$ at grid spacing values less than 1/4 Angstrom.
Figure 3-4 Convergence of salt-related energy term (kcal/mol) versus grid spacing (Ångstrom). The nucleic acid 420D was modeled with the full PBE (epsout=80.0, epsin=1.0, istrng=1000). The salt energy is the sum of the ionic energy term and the entropy term in the full PB electrostatic free energy. Top: The salt energy versus grid spacing. The convergence trend lines are fitted with respect to grid spacing in the form of $y = a + bx^c$. Bottom: The deviation between the salt energies and predicted energy at $h=0$ according to the curve fitting. The deviation is relative and is respect to predicted total solvation free energy. On the other hand, as shown in eqn (25), calculation of the ionic energy with the charge view method needs a pair-wise summation of all grid points (N), scaling quadratically with the number of grid points, $O(N^2)$, which is very computationally demanding, while the original finite-difference full PBE method (particle mesh) scales only linearly with the number of grid points, $O(N)$. Therefore in the development of an efficient P3M method, the original finite-difference approach is used for the salt-related terms.

3.4.5 Accuracy and Efficiency of P3M method

As presented in Theory and Computational Details (subsection D), the essence of the P3M strategy is to use the more accurate charge-based treatments of energy and forces for short-range interactions and keep the long-ranged interactions from the finite-difference full
PBE treatment. Our discussion in Theory and Computational Details (subsection E) and the testing data above show that the ionic terms can generally be treated as in the finite-difference full PBE treatment without introducing much error, so the charge-based method is only applied to Coulomb and reaction field interactions to balance accuracy and efficiency.

The pairwise cutoff for the short-range interactions in the P3M strategy apparently influences the accuracy of energy and forces calculation. In general, a larger cutoff leads to more accurate results but is more computationally demanding. Worth noting is that the total electrostatic energy converges very rapidly: the relative error is already less than $10^{-4}$ when a very short cutoff distance of six grids is used. This is already within the convergence errors of the pairwise charge view method at typical coarse grid spacing used, i.e. 1/2 or 1/4 Angstrom as shown in Figure 3-3 and Table 3-1. For example, at 1/4 Angstrom, the charge view energy is -7.7962 kcal/mol, the relative error is $6 \times 10^{-4}$ compares to the fitted value of the full PB equation, the relative error is larger when grid spacing is 1/2 Angstrom.

However the accuracy of atomic forces is more sensitive to cutoff. Table 3-2 summarizes the accuracy of atomic reaction forces for the tested nucleic acid 420D. The errors in forces by the P3M strategy with different cutoffs were analyzed with the charge-view method set as the benchmark. The grid spacing was set as 1/2 Angstrom and 1/4 Angstrom. It is clear that the errors of forces decrease when the cutoff distance increases. And the force errors at the 1/4 Angstrom grid spacing are also smaller than those at the 1/2 Angstrom grid spacing. In the
following analysis, the short-range cutoff distance is set as 14 grid spacing, with which the RMS error of the forces is $< 5 \times 10^{-4}$.

<table>
<thead>
<tr>
<th>Cutoff</th>
<th>RMSrD force components (0.50 Å)</th>
<th>RMSrD force components (0.25 Å)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>$2.7 \times 10^{-3}$</td>
<td>$1.1 \times 10^{-3}$</td>
</tr>
<tr>
<td>8</td>
<td>$1.3 \times 10^{-3}$</td>
<td>$5.6 \times 10^{-4}$</td>
</tr>
<tr>
<td>10</td>
<td>$9.4 \times 10^{-4}$</td>
<td>$4.3 \times 10^{-4}$</td>
</tr>
<tr>
<td>12</td>
<td>$6.2 \times 10^{-4}$</td>
<td>$3.2 \times 10^{-4}$</td>
</tr>
<tr>
<td>14</td>
<td>$4.4 \times 10^{-4}$</td>
<td>$2.4 \times 10^{-4}$</td>
</tr>
<tr>
<td>16</td>
<td>$3.4 \times 10^{-4}$</td>
<td>$1.9 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

**Table 3-2** Root-mean-squared relative deviations in P3M reaction field forces on DNA 420D with absolute values greater than 1 kcal/mol-Å for the P3M strategy with different cutoff (in grid spacing) compared to the charge view method. Both grid spacing of 0.25 Å and grid spacing of 0.50 Å are computed. Polarization charges are mapped onto the atomic surface.

Next the consistency of the P3M method and the pairwise charge-based method was systematically validated with a large set of PDB structures of nucleic acids. Figure 3-5 shows the electrostatic energy correlation between the charge view method and the P3M method for non-linear PB equations. It can be seen that these two sets of data are highly consistent with each other, with relative error less than $5 \times 10^{-4}$. 

![Electrostatic Energy Correlation](image)
Figure 3-5 Correlations between full PBE electrostatic energies (kcal/mol) computed by the charge view and the P3M methods. A test set of 283 different nucleic acids was modeled with the full PBE (epsout=80.0, epsin=4.0, istrng=1000). The charge view energies were computed with the analytical Green’s function, i.e. Coulomb’s law. P3M cutoff is 14. The rms relative deviation is $3.3 \times 10^{-4}$.

Finally a timing analysis was conducted on the tested nucleic acids and is shown in Figure 3-6, which plots the correlation of CPU times of energy/force computation versus the system sizes (no. of atoms). It is clear that the P3M time scales with system size much better than the pairwise charge view strategy, with the difference becomes significantly larger when the no. of atoms approaching 1,000’s.

![Figure 3-6](image.png)

Figure 3-6 CPU time versus no. of solute atoms for the pairwise charge view method and the P3M method. The same test set of nucleic acids was modeled with the full PB equation (epsout=80.0, epsin=4.0, istrng=1000) as in Figure 3-5. The x-axis is the number of atoms in each nucleic acids, the y-axis is the cost of CPU time for computing the energies. The trend lines are fitted functions in the form of $y = ax^2 + bx + c$.

3.4.6 Accuracy and convergence of electrostatic forces

The accuracy and convergence of electrostatic forces are important issues if we want to use the new method for dynamics simulations. A comparison of the original PBE method,
pairwise charge view method, and the P3M method is shown in Figure 3-7 – Figure 3-10 and Table 3-3 – Table 3-4. We first studied the performance of the new methods with all polarization charges mapped to molecular surface as shown to improve the convergence of energies (Figure 3-3).

Figure 3-7 plots the correlations of forces computed at different grid spacing values with the benchmark data set obtained at the finest grid spacing used (1/16 Angstrom). It is clear that the correlation becomes better as the grid spacing decreases. Table 3-3 shows the RMS relative errors for the dominant force components (here chosen to be larger than 1 kcal/mol-Å) at tested grid spacing values with respect to the benchmark data set. The analysis shows that the RMS relative error becomes significantly smaller for all three tested strategies as the grid spacing is reduced. As the grid spacing reduces from 0.5 Angstrom to 0.125 Angstrom, the RMS relative error for the original full PB method is reduced from 0.084 to 0.0092, while for the charge view method and P3M method, the error is reduced from 0.025 to 0.0023. It is also interesting to notice that the RMS relative errors of the both the charge-view method and the P3M method are significantly smaller than those of the original finite-difference method at all tested grid spacing values (1/8 to 1/2 Angstrom). Finally, the consistency among the three different methods at 1/16 Angstroms is also shown in Figure 3-8 and Table 3-3. The RMS relative error between the charge view method and the full PBE method is 0.0053. This is similar to the convergence error of the full PBE energy (Figure 3-3) at 1/16 Angstrom, around 0.0047, assuming the charge-view
method converges much earlier as shown in Figure 3-3 (top panel). The RMS relative error between the charge view method and the P3M method is 0.0002, which is consistent with the accuracy of the P3M method for energies as shown in Figure 3-5.

![Figure 3-7](image)

**Figure 3-7 Self-convergence of reaction field forces (kcal/mol-Å).** The AT base dimer was modeled with the full PBE. (epsout=80.0, epsin=4.0, istrng=1000). Left: field-view method. Center: charge-view method. Right: P3M method. The polarization charges are mapped onto the molecular surface. See Table 3-4 for correlation analyses.
Figure 3-8 Consistency of reaction field forces (kcal/mol-Å) among different force interpretation methods at the tested fine grid spacing. Polarization charges are mapped onto the molecular surface. (epsout=80.0, epsin=4.0, istrng=1000). Top: The full PBE versus charge view. The rms relative deviation is 0.0052 for nontrivial force components (absolute values > 1 kcal/mol-Å). Bottom: P3M versus charge view. The rms relative deviation is 0.0002 for nontrivial force components. The fine grid spacing is set at 1/16 Angstrom as in Figure 3-7.

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>RMSrD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0625 full PBE</td>
<td>0.5 full PBE</td>
<td>0.084</td>
</tr>
<tr>
<td>0.0625 full PBE</td>
<td>0.25 full PBE</td>
<td>0.038</td>
</tr>
<tr>
<td>0.0625 full PBE</td>
<td>0.125 full PBE</td>
<td>0.0092</td>
</tr>
<tr>
<td>0.0625 charge view</td>
<td>0.5 charge view</td>
<td>0.025</td>
</tr>
<tr>
<td>0.0625 charge view</td>
<td>0.25 charge view</td>
<td>0.0077</td>
</tr>
<tr>
<td>0.0625 charge view</td>
<td>0.125 charge view</td>
<td>0.0023</td>
</tr>
<tr>
<td>0.0625 P3M</td>
<td>0.5 P3M</td>
<td>0.025</td>
</tr>
<tr>
<td>0.0625 P3M</td>
<td>0.25 P3M</td>
<td>0.0076</td>
</tr>
<tr>
<td></td>
<td>0.0625 P3M</td>
<td>0.125 P3M</td>
</tr>
<tr>
<td>----------------</td>
<td>------------</td>
<td>-----------</td>
</tr>
<tr>
<td>0.0625 charge view</td>
<td>0.0625 full PB</td>
<td>0.0053</td>
</tr>
<tr>
<td>0.0625 charge view</td>
<td>0.0625 P3M</td>
<td>0.00020</td>
</tr>
</tbody>
</table>

**Table 3-3** Root-mean-squared relative deviations in reaction field forces with absolute values greater than 1 kcal/mol-Å between strategy y and strategy x. Polarization charges are mapped onto the atomic surface.

In Figure 3-9 and 4-10 and Table 3-4 the same analysis was conducted but with the polarization charges set to remain on the grid points for charge-view and P3M methods. Figure 3-9 shows that the correlation becomes better as the grid spacing decreases, similar to Figure 3-7. Table 3-4 shows that for the charge view method and P3M method, the RMS relative errors at all grid spacing values are larger than those with the polarization charges mapped onto the molecular surface (Table 3-3). Table 3-4 also shows that the consistency among the three methods is higher at the finest grid spacing (1/16 Angstrom) when charges are set to remain at the grid points. This is also in agreement with the convergence trend of energies as presented in Figure 3-3 (bottom panel). Note too the benefits of mapping charges remain to be high at finer grid spacing tested, i.e. 1/4 and 1/8 Angstrom. In summary the two sets of convergence data support the practice of mapping polarization charges onto the molecular surface.
Figure 3-9 Self-convergence of reaction field forces (kcal/mol-Å). Same as Figure 3-7, except that the polarization charges remain on the boundary grid points. See Table 3-5 for correlation analyses.
Figure 3-10 Consistency of reaction field forces (kcal/mol-Å) among different force interpretation methods at tested fine grid spacing. Same as Figure 3-8, except that the polarization charges remain on the boundary grid points. Top: the rms relative deviation is 0.0002. Bottom: the rms relative deviation is 0.0002.

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>RMSrD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0625 full PBE</td>
<td>0.5 full PBE</td>
<td>0.084</td>
</tr>
<tr>
<td>0.0625 full PBE</td>
<td>0.25 full PBE</td>
<td>0.038</td>
</tr>
<tr>
<td>0.0625 full PBE</td>
<td>0.125 full PBE</td>
<td>0.0092</td>
</tr>
<tr>
<td>0.0625 charge view</td>
<td>0.5 charge view</td>
<td>0.078</td>
</tr>
<tr>
<td>0.0625 charge view</td>
<td>0.25 charge view</td>
<td>0.035</td>
</tr>
<tr>
<td>0.0625 charge view</td>
<td>0.125 charge view</td>
<td>0.0090</td>
</tr>
<tr>
<td>0.0625 P3M</td>
<td>0.5 P3M</td>
<td>0.077</td>
</tr>
<tr>
<td>0.0625 P3M</td>
<td>0.25 P3M</td>
<td>0.035</td>
</tr>
<tr>
<td>0.0625 P3M</td>
<td>0.125 P3M</td>
<td>0.0092</td>
</tr>
<tr>
<td>0.0625 charge view</td>
<td>0.0625 full PB</td>
<td>0.00024</td>
</tr>
</tbody>
</table>
Table 3-4 Root-mean-squared relative deviations in reaction field forces with absolute values greater than 1 kcal/mol-Å between strategy y and strategy x. Polarization charges are not mapped onto the atomic surface.

3.4.7 Net force analysis

In molecular dynamics simulation, the net system force is supposed to be zero given the use of a conservative force field. However, due to numerical error, the net force is in general not zero. Thus this is a simple initial test to assess whether the force interpolation is accurate enough before it is tested in the complex molecular dynamics programs.

Here the analysis was conducted on the 1026-atom nucleic acid (420D) and also a 1389-atom protein (1F81), to demonstrate the performance of the P3M method. As shown in Table 3-5, the net force is first computed and averaged over each atom, and then the unsigned average of the force components over all the atoms are also computed as references. As shown in Table 3-5, the averaged net force components are always less than $10^{-3}$ of the unsigned average force components. This should be viewed in the context of typical continuum dynamics simulations, where Langevin thermostat is often used. Given a small collision frequency (1 ps$^{-1}$) often used in continuum dynamics simulations, a time step of 0.001 ps, and 300K simulation temperature, the average random forces are ~1.7 kcal/mol-Å on the lightest hydrogen atoms. Thus the nonzero net force plays a less significant role in biasing the dynamics simulations.

<table>
<thead>
<tr>
<th>Biomolecules</th>
<th>420D</th>
<th>1F81</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average net force x</td>
<td>$-3.2\times10^{-4}$</td>
<td>$-2.1\times10^{-4}$</td>
</tr>
<tr>
<td>Average unsigned force x</td>
<td>3.4</td>
<td>2.5</td>
</tr>
<tr>
<td>Average net force y</td>
<td>$-6.2\times10^{-4}$</td>
<td>$-7.7\times10^{-4}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>Average unsigned force y</td>
<td>3.7</td>
<td>2.5</td>
</tr>
<tr>
<td>Average net force z</td>
<td>-9.1×10^{-4}</td>
<td>-9.3×10^{-5}</td>
</tr>
<tr>
<td>Average unsigned force z</td>
<td>2.7</td>
<td>2.5</td>
</tr>
</tbody>
</table>

**Table 3-5** Averaged net force and unsigned average of force components in x, y, z directions for nucleic acid 420D and protein 1F81. The unit is kcal/mol-Å.

3.5 Appendices

3.A.1 Effective charge interpretation of Poisson-Boltzmann equation

Before exploring the general Poisson-Boltzmann’s equation, it is instructive to discuss the more fundamental Poisson’s equation

\[ \nabla \cdot (\varepsilon \nabla \phi) = -4\pi \rho' \]

(42)

It is well known that \( \phi \) can be split into Coulombic potential and reaction field potential, i.e., \( \phi = \phi_C + \phi_{RF} \) in eqn (42). To see how this is possible, suppose a solute molecule with dielectric constant \( \varepsilon_i \) is surrounded by a solvent medium with a uniform dielectric constant \( \varepsilon_o \), the Coulombic potential is defined as

\[ \varepsilon_i \nabla^2 \phi_C = -4\pi \rho' \]

(43)

which means that the Coulombic potential (\( \phi_C \)) is generated by atomic charges in a uniform medium with dielectric constant \( \varepsilon_i \) throughout the whole space. Next introduce its associated electric displacement \( D_C = -\varepsilon_i \nabla \phi_C = \varepsilon_i E_C \). Given \( D_C = E_C + 4\pi \mathbf{P}_c \), where \( \mathbf{P}_c \) is the
polarization vector in the uniform dielectric of $\varepsilon_i$. It is clear that eqn (43) can be rewritten as

$$\nabla \cdot \mathbf{E}_C + 4\pi \nabla \cdot \mathbf{P}_C = 4\pi \rho_f$$  

(44)

With the Coulombic field ($\mathbf{E}_C$) so defined, the reaction field is simply the electrostatic field generated by the charges induced by transferring the environment surrounding the solute from $\varepsilon_o$ to $\varepsilon_i$. To obtain an equation for the reaction field potential, we first reformulate eqn (42) with the help of the electric displacement vector for the inhomogeneous dielectric $\mathbf{D} = -\varepsilon \nabla \phi = \varepsilon \mathbf{E}$

$$\nabla \cdot (\varepsilon \mathbf{E}) = \nabla \cdot \mathbf{D} = 4\pi \rho_f$$  

(45)

Now define $\mathbf{E}_{RF}$ and $\mathbf{P}_{RF}$ so that they satisfy the following relation between the total electrostatic field/polarization and the Coulombic field/polarization:

$$\mathbf{E} = \mathbf{E}_C + \mathbf{E}_{RF}$$
$$\mathbf{P} = \mathbf{P}_C + \mathbf{P}_{RF}$$  

(46)

Thus $\mathbf{E}_{RF}$ and $\mathbf{P}_{RF}$ are the reaction field and polarization, respectively, induced by the inhomogeneous dielectric with respect to the homogeneous dielectric as in eqn (44). Substitution of $\mathbf{D} = \mathbf{E} + 4\pi \mathbf{P}$ into eqn (42), we have

$$\nabla \cdot \mathbf{E}_C + \nabla \cdot \mathbf{E}_{RF} + 4\pi \nabla \cdot \mathbf{P}_C + 4\pi \nabla \cdot \mathbf{P}_{RF} = 4\pi \rho_f$$  

(47)

Given eqn (47), it can be simplified to

$$\nabla \cdot \mathbf{E}_{RF} = -4\pi \nabla \cdot \mathbf{P}_{RF}$$  

(48)

Thus the reaction field potential satisfies
\[ \nabla^2 \phi_{RF} = \nabla \cdot \mathbf{E}_{RF} = -4\pi \rho_{pol} \]

Given \( \rho_{pol} = -\nabla \cdot \mathbf{P}_{RF} \) \hfill (49)

Given \( \phi = \phi_c + \phi_{RF} \) and eqn (43) and (49), the Poisson’s eqn (42) can then be reformulated as

\[ \nabla^2 \phi = -4\pi \left( \frac{\rho^f}{\varepsilon_i} + \rho_{pol} \right) \] \hfill (50)

Thus the total electrostatic potential can be viewed as the summation of two vacuum Coulombic potentials: \( \phi_c \) from effective charge source \( \frac{\rho^f}{\varepsilon_i} \) and reaction field potential \( \phi_{RF} \) from effective charge source \( \rho_{pol} \). In reformulating the Poisson’s equation this way, it is possible to explore alternative strategies in solving the equation. Of course, this requires an efficient way to compute the polarization charges, for example, as shown in one of our recent works. \(^9^4\)

Now consider the more general Poisson-Boltzmann equation for systems with continuum mobile ions

\[ \nabla \cdot (\varepsilon_i \nabla \phi) = -4\pi \rho^f - 4\pi \sum_i e_i c_i \exp\left(-\frac{e_i \phi}{k_B T}\right) \lambda \] \hfill (51)

where \( e_i \) is the charge of ion type \( i \), \( c_i \) is the bulk number density of ion type \( i \), \( \lambda \) is the ion exclusion function, \( k_B \) is the Boltzmann constant and \( T \) is the absolute temperature. Following the development in the Poisson’s equation, what is needed here is a decomposition of total electrostatic potential when the continuum ion terms present.

To build upon the previous analysis of Poisson’s equation, let us assume that the final state of the system is reached first by charging up the solute without any continuum ions, then
by releasing the continuum ions. Thus, the final electrostatic potential for the 
Poisson-Boltzmann’s equation (φ) can be regarded as a perturbation to the electrostatic 
potential for the Poisson’s equation (φ_{PE}), i.e. φ = φ_{PE} + φ^m, where φ^m is used to denote the 
perturbation due to the continuum ions. It is straightforward to show that φ^m satisfies the 
following equation

\[ \nabla \cdot (\varepsilon \nabla \phi^m) = -4\pi \rho^m \]

given \( \rho^m = \sum_i e_i c_i \exp(-e_i \phi / k_b T) \lambda_i \) (52)

by substituting φ = φ_{PE} + φ^m into the Poisson-Boltzmann’s equation. eqn shows that φ^m can be 
viewed as a potential caused by the charge distribution \( \rho^m \) in the dielectric environment just as 
φ_{PE} is caused by \( \rho^f \) in the same dielectric environment.

Similar to the treatment of potential by \( \rho^f \), we shall decompose \( \rho^m \) into a Coulombic 
field and reaction field, \( \phi^m = \phi^C + \phi_{RF}^m \). However, the difference from \( \rho^f \) is that \( \rho^m \) is 
immersed in the homogenous dielectric of solvent (\( \varepsilon_o \)), but not that of solute (\( \varepsilon_i \)). Therefore the 
equation for \( \phi^C \) is

\[ \varepsilon_o \nabla^2 \phi^m_C = -\varepsilon_o \nabla \cdot E^m_C = -4\pi \rho^m \] (53)

Similar to \( \phi_{RF} \), \( \phi_{RF}^m \) can be viewed as generated by a polarization charge density,
\( \rho^{pol.m} \).

\[ \nabla^2 \phi_{RF} = -4\pi \rho^{pol.m} \]
\[ \rho^{pol.m} = -\nabla \cdot P_{RF} \] (54)

Therefore the perturbation potential \( \phi^m = \phi^C + \phi_{RF}^m \) due to the continuum ions can be regarded
as being caused by the charge distributions of $\frac{\rho^m}{\varepsilon_o}$ and $\rho^{pol,m}$

$$\nabla^2 \phi^m = -4\pi \left( \frac{\rho^m}{\varepsilon_o} + \rho^{pol,m} \right),$$ \hspace{1cm} (55)

Now it is time to merge what we have derived in the treatment of Poisson’s equation and the perturbation due to the continuum ions. eqn (51) and (55) thus give us

$$\nabla^2 \phi = -4\pi \left( \frac{\rho^f}{\varepsilon_i} + \frac{\rho^m}{\varepsilon_o} + \rho^{pol} \right),$$ \hspace{1cm} (56)

where we have merged $\rho^{pol,m}$ into $\rho^{pol}$. Here $\phi_{RF}$ satisfies the revised relation

$$\nabla^2 \phi_{RF} = -\nabla \cdot \mathbf{E}_{RF} = -4\pi \rho^{pol}$$ \hspace{1cm} (57)

3.A.2 High precision finite-difference Green’s function values

The finite-difference Green’s function needed for self-energy and short-range Coulombic energy was previously precomputed based on the work of Luty and McCammon.\textsuperscript{87} Due to limited computational resources, only function values up to 20-grid separation in each dimension were precomputed. In this study, finite-difference Green’s function values were computed in a brute-force manner by solving a vacuum Coulomb field of a point charge of 1 unit charge with the standard finite-difference method. The charge was positioned at the center of the finite-difference grids, and the boundary potential was set analytically according to the Coulomb law. As the grid dimension increases, potentials on grid nodes close to the center converge to the values of the finite-difference Green’s function. Table 3-A1 shows the maximum relative error of Green’s function as the grid dimension increases. The error analysis
shows that 8 digits of accuracy can be achieved when $\Delta x, \Delta y, \Delta z \leq 20$, and the error of the function values when $\Delta x, \Delta y, \Delta z \leq 40$ is also very close, less than $1.4 \times 10^{-8}$.

The updated finite difference Green’s function values along with the documented algorithms are incorporated in the latest Amber simulation package to be released in the spring of 2016.

<table>
<thead>
<tr>
<th>Grid dimension</th>
<th>Max error $\Delta x, \Delta y, \Delta z \leq 40$</th>
<th>Max error $\Delta x, \Delta y, \Delta z \leq 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>481×481×481</td>
<td>$1.1 \times 10^{-7}$</td>
<td>$5.7 \times 10^{-8}$</td>
</tr>
<tr>
<td>601×601×601</td>
<td>$5.1 \times 10^{-8}$</td>
<td>$2.5 \times 10^{-8}$</td>
</tr>
<tr>
<td>721×721×721</td>
<td>$2.6 \times 10^{-8}$</td>
<td>$1.3 \times 10^{-8}$</td>
</tr>
<tr>
<td>841×841×841</td>
<td>$1.4 \times 10^{-8}$</td>
<td>$7.3 \times 10^{-9}$</td>
</tr>
<tr>
<td>961×961×961</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

**Table 3-A1**: Maximum convergence error of finite-difference Green’s function values when $\Delta x$, $\Delta y$, $\Delta z \leq 40$ or $\Delta x$, $\Delta y$, $\Delta z \leq 20$. 
References


CHAPTER 4

A Continuum Poisson-Boltzmann Model for Membrane Channel Proteins

4.1 Summary

Membrane proteins constitute a large portion of the human proteome and perform a variety of important functions as membrane receptors, transport proteins, enzymes, signaling proteins, and more. The computational studies of membrane proteins are usually much more complicated than for globular proteins. Here we propose a new continuum model for Poisson-Boltzmann calculations of membrane channel proteins. Major improvements of our model compared to the continuum slab model are as follows: 1) The highly different accessibility in the membrane and water regions can be addressed automatically, 2) The water pores/channels can be automatically labeled, and 3) The mapping of membrane, water, and protein dielectric constants throughout the simulation box is done automatically. The new continuum membrane model was then optimized (by adjusting the membrane probe and the slab thickness) to best reproduce the distributions of buried water molecules in the membrane region as sampled in explicit water simulations. We also found that the widely adopted water probe of 1.4 Å for globular proteins is a very reasonable default value for membrane protein simulations. It gives an overall minimum number of inconsistencies between the continuum and explicit representations of water distributions in membrane channel proteins, at least in the water accessible pore/channel regions that we have focused on. Finally, we validated the new membrane model by carrying out binding affinity calculations for a potassium channel, and we observed good agreement with experiment.

4.2 Introduction
Membrane proteins constitute a large portion of the human proteome and perform a variety of important functions as membrane receptors, transport proteins, enzymes, and signaling proteins. These important proteins have become primary drug targets in modern medicine: over 60% of all drugs target these proteins. However, the study of membrane proteins is usually much more complicated than that of globular proteins, both experimentally and computationally. For experimental studies, the difficulty of obtaining a high-resolution structure is an obstacle, especially for studies that involve proteins found in humans. For computational studies, modeling of the membrane environment is also an important consideration.

Since most biomolecular systems exist in an aqueous environment, it is important to account for solvent effects. There are two ways to include solvent effects in a computational simulation: explicit and implicit solvation. In explicit solvation modeling, each solvent atom is modeled explicitly. Although this is the most accurate method, what we are interested in is often not the properties of the solvent itself, but rather its influence on the solute molecules. In addition, accurately capturing the solvent influence in a statistically meaningful way requires sampling either from an ensemble of trajectories or from a single very long trajectory, which is very computationally demanding. Implicit solvation modeling provides an attractive alternative wherein the solvent molecules are collectively modeled as a continuum. In implicit solvent models, although the details of individual solvent atoms are lost, the relevant important statistically averaged effects can still be preserved by design. Since solvent molecules typically constitute the major portion of molecules for an explicit solvent simulation, implicit solvent modeling can lead to much more efficient simulations. In addition to water, membrane molecules should also be included when modeling solvation effects, and implicit membrane modeling has also been developed.
A key issue in developing implicit solvent models is the modeling of electrostatic interactions. The Poisson-Boltzmann equation (PBE) has been established as a fundamental equation to model continuum electrostatic interactions. The solvent molecules are modeled as a continuum with a high dielectric constant, and the solute atoms are modeled as a continuum with a low dielectric constant and buried atomic charges. The effect of charged ions in the solvent region is included by adding mobile charge density terms that obey Boltzmann distributions. The potential of the full system is then governed by the partial differential equation:

\[ \nabla \cdot \varepsilon \nabla \phi = -4\pi \rho_0 - 4\pi \sum_i e z_i c_i \lambda \exp(-e z_i \phi / k_B T) \]

where \( \nabla \) is the spatial gradient operator, \( \varepsilon \) is the dielectric constant distribution, \( \phi \) is the electrostatic potential distribution, \( \rho_0 \) is the charge density of the solute (usually modeled as a set of discrete point charges), \( c_i \) is the concentration of the \( i \)th solvent ion species in bulk, \( e \) is the absolute charge of an electron, \( z_i \) is the valence for the \( i \)th ion, \( k_B \) is Boltzmann’s constant, \( T \) is the temperature, and \( \lambda \) is the Stern layer masking function, which is 0 within or 1 outside of the Stern layer.

The PBE is a non-linear elliptical partial differential equation. There is no closed form solution, and thus, numerical methods are required. Efficient numerical PBE-based solvent models have been widely used to study biological processes including predicting pKa values, computing solvation and binding free energies, and protein folding. Predicting protein-ligand binding affinities is one of the major applications for implicit solvent free energy calculations. In the Amber software package, MMPBSA is the module performing such calculations. Implicit membrane modeling has also been applied and developed in binding free energy calculations. There are a noticeable number of pioneer works that implement
implicit membrane modeling in several PB packages, such as APBS, Delphi, PBEQ, and PBSA. All of them add the membrane as a slab with a relatively low dielectric constant that is embedded in water for PBE calculations.

Our previous work implemented an implicit membrane model into the PBE framework. The implicit membrane model can be readily interfaced with the existing MMPBSA program to perform binding free energy calculations of several protein structures embedded in a membrane. However, a problem arises when those membrane proteins contain a pore or a gated channel, since the region of the channel is usually permeable and should be composed of water. Therefore, a simple slab-like membrane setup may cause problems if the membrane protein contains pore- or channel-like region(s). Similar to the approaches adopted in the community, we dealt with this issue by manually defining the pore region as a cylinder, and we then set the dielectric constant within the cylindrical region as that of water if it was not occupied by protein atoms. The limitation of this method is that, for every snapshot of a trajectory, we need to visualize and locate the cylinder by hand, which is neither efficient nor practical given the large number of snapshots that must be processed for converged calculations.

In this work, we propose a new continuum membrane model for PBE calculations of biomolecules. Major improvements from the standard continuum slab model are the following: 1) a two-step, two-probe initial grid labeling procedure was adopted to address highly different accessibility in the membrane region and water region; 2) a depth-first search algorithm was introduced to detect the water pores/channels automatically based on the initial grid labels; and 3) an explicit solvent MD simulation was exploited to fine tune the slab model, i.e. its exact location and thickness, to best reproduce the solvent accessibility and the water accessible channel. This procedure follows our basic algorithm proposed for globular proteins, and adds
little overall overhead in the application of linear finite-difference PBE solvers to typical membrane proteins.

4.3 Methods

The Poisson-Boltzmann equation (Eqn (1)) is widely used in capturing electrostatic energy and forces in implicit solvent modeling. For systems with dilute ion concentrations, the second term on the right-hand side is usually linearized, giving the simpler form:

$$\nabla \cdot \varepsilon \nabla \phi = -4\pi \rho_0 + \lambda \kappa^2 \phi$$  \hspace{1cm} (2)

where $\kappa^2 = 4\pi \sum_i c_i e_i^2 z_i^2 / k_B T$.

The finite-difference method $^{22,43,71-82,84,87}$ is one of the most popular methods used in the numerical implementation of the PBE. In a typical procedure, the rectangular grid covering the solution system is first defined. Next, the atomic point charges are mapped onto the grid points with a predefined assignment function. Third, the dielectric constant distribution is mapped to the grid edges. The discretized linear system is then turned to a linear solver to solve for potentials on the grid points, which can be expressed as:

$$\begin{bmatrix}
\varepsilon_x(i,j,k)\phi(i+1,j,k) + \varepsilon_y(i-1,j,k)\phi(i-1,j,k) \\
\varepsilon_x(i,j,k)\phi(i,j+1,k) + \varepsilon_y(i,j-1,k)\phi(i,j-1,k) \\
\varepsilon_z(i,j,k)\phi(i,j,k+1) + \varepsilon_z(i,j,k-1)\phi(i,j,k-1) \\
\varepsilon_x(i-1,j,k) + \varepsilon_y(i,j,k) + \varepsilon_x(i,j-1,k) \\
\varepsilon_y(i,j,k) + \varepsilon_z(i,j,k-1) + \varepsilon_z(i,j,k)
\end{bmatrix} \phi(i,j,k) = -\frac{4\pi \rho(i,j,k)}{h} + h^2 \lambda(i,j,k) \kappa^2 \phi(i,j,k)$$  \hspace{1cm} (3)

Here $\varepsilon_x, \varepsilon_y, \text{and} \ \varepsilon_z$ represent the dielectric constants for grid edges along the x, y, and z directions respectively, and $h$ represents the grid spacing.
This study focuses on how to set up linear PBE applications for membrane systems. A major issue is the presence of the membrane and its influence on the dielectric constant distribution. In globular proteins, the solvent excluded surface (SES)\textsuperscript{43, 126-130} is often used as a boundary separating the high dielectric water exterior and the low dielectric protein interior. The presence of the membrane introduces at least a third region. In this study, we adopt the uniform membrane dielectric model, though our procedure can be easily extended to accommodate another often used depth-dependent membrane dielectric model.

The first step is to introduce a membrane region to the existing solvent excluded surface procedure with minimum invasion to the program and minimum efficiency lost. The SES is the most common surface definition used to describe the dielectric interface between the two piece-wise dielectric constants. In fact, comparative analysis of PB-based solvent models and TIP3P solvent models have shown that the SES definition is reasonable in the calculation of reaction field energies and electrostatic potentials of mean force fields.\textsuperscript{131-133} Here, we follow the idea from Rocchia \textit{et al.}\textsuperscript{130} and Wang \textit{et al.}\textsuperscript{43} of mapping the SES to a finite-difference grid. While keeping the variables used to label the solvent and solute regions, we also introduce a new variable to label the membrane region. Considering the membrane molecules are usually larger than solvent molecules, we use two different solvent probe radii to set up the membrane and solvent regions. And finally, we assign the dielectric constant on each region and the interface.

\textbf{4.3.1 Grid point labeling}

Our general strategy is to model the membrane as a second continuum solvent of finite region, i.e. a slab located at a user specified position. The essence of the algorithm is to determine both the membrane accessibility and water accessibility around a molecular solute.
Assisted with both sets of accessibility data, the presence of water channels or water pores within the membrane region can then be identified in the next step. Due to the much larger size of lipid molecules, a separate solvent probe (mprob) must be used to determine the membrane accessibility. This is apparently much larger than the water probe (dprob). The influence of both probes on reproducing the solvent accessible surface of a membrane protein is presented in Results and Discussion.

In Amber/PBSA, an integer array insas is used to label whether the grid point is outside the solute region (insas<0) or inside the solute region (insas>0) for fast mapping of solvent accessibility information. This labeling scheme has been extended to map all commonly used surfaces, SES, SAS, VDW, and DEN in recent Amber and AmberTools releases. To minimize the interference to existing procedures and maximize efficiency, a separate integer array inmem is used to label whether a grid point is inside the membrane (inmem>0) or outside the membrane (inmem=0). Specifically, the grid labeling algorithm can be summarized as the following five steps:

0. Initialize insas of all grid points as “-4”, i.e. in the bulk solvent and salt region, and inmem of all grid points as “0”, i.e. outside the membrane region.
1. Using mprob as the solvent probe radius, label insas of all grid points as “-3” if within the stern layer; “-2” if within the solvent accessible surface layer; “-1” if within the reentry region but outside the SES; “1” if within the reentry region but inside the SES; “2” if inside the VDW surface.
2. Add a slab perpendicular to the z-axis as the membrane region at the specified location. Label inmem of the membrane-region grid points with insas<0 as “1”.
3. Apply the depth-first search algorithm to detect any possible membrane accessible grid point that is not connected to the bulk membrane. If so, relabel its inmem as “0”.
4. For each grid point with (inmem=0) within the slab, if it has a neighbor with (inmem=1) within the distance cutoff of memmaxd, relabel its inmem as “2”.
5. Using dprob as the solvent probe radius, relabel insas of all grid points as “-3” if within the stern layer; “-2” if within
the solvent accessible surface layer; “-1” if within the reentry region outside the SES; “1” if within the reentry region inside the SES; “2” if inside the VDW surface.

A few explanations are in order here. First, \textit{inmem} is determined in Step 2 through Step 4, so that its value is controlled by both the \textit{mprob}-generated \textit{insas} and the depth-first search algorithm. Second, a new variable (\textit{memmaxd}) is introduced in Step 4. Since \textit{mprob} is usually much larger than \textit{dprob}, there exists a thin layer of grid points with \textit{insas}>0 and \textit{inmem}=0 between the membrane region and the protein region. If the grid labels are set this way, these grid points would be labeled as water in a later processing stage of our method, thus leading to an artificial layer of water between the protein and membrane. To resolve this issue, a cutoff distance of \textit{memmaxd} is introduced to represent the maximum difference between the SES surfaces generated by \textit{mprob} and \textit{dprob}. This is estimated to be \textit{mprob−dprob} assuming maximum reentry by \textit{dprob}. Thus Step 4 changes the \textit{inmem} labels of the grid points from 0 (\textit{mprob} inaccessible) to 2 (\textit{mprob} accessible) if they are \textit{memmaxd} inside the \textit{mprob}-generated SES. The correction effectively removes the artificial layer of water between the protein and the membrane. Here the revised \textit{inmem} values are set to be “2” so these grid points would not interfere with the subsequent search. Note too that this correction does not change the protein interior definition, which is defined with the water \textit{dprob}. Nevertheless, it does have the effect of pushing back the potential buried water pockets, if any, from the protein-membrane interface.

In summary, the three different regions that are readily available for further processing after the grid-labeling step are:

1. Solute region: \textit{insas}(i,j,k)>0
2. Membrane region: \textit{insas}(i,j,k)<0 and \textit{inmem}(i,j,k)>0
3. Solvent region: \textit{insas}(i,j,k)<0 and \textit{inmem}(i,j,k)=0

\subsection*{4.3.2 Membrane pore/channel detection}
Step 3 in the above general grid-labeling algorithm is meant to identify pore- or channel-like water-accessible water pockets within a user-specified membrane region. Given the convention that the membrane is parallel with the $xy$ plane, the membrane region can be mathematically defined to be all grid points within $[z_{\text{min}}, z_{\text{max}}]$. Thus the method starts by initializing all grid points that are defined as solvent ($\text{insas}<0$) within $[z_{\text{min}}, z_{\text{max}}]$ as $\text{inmem}=1$. Next the recursive depth-first search algorithm is used to traverse all grid points to see whether they are connected or not. Our goal of using the algorithm is to walk and label recursively all grid points in the non-protein regions within $[z_{\text{min}}, z_{\text{max}}]$. Upon completion, all grid points that are not connected to the membrane region (i.e. the pore region) are labeled back as the water region ($\text{inmem}=0$). To facilitate the bookkeeping of the search, a variable $\text{kzone}$ is introduced to label the different regions: the protein region ($\text{kzone}=0$), the membrane region ($\text{kzone}=1$), and the water regions ($\text{kzone}>1$). Since the search starts from the edge of the membrane slab, the first region found is always the membrane region ($\text{kzone}=1$), and the rest are the water regions or the protein region. In general multiple $\text{kzone}$ values are assigned because most water-accessible regions are not connected. The algorithm can be summarized as shown below:

```plaintext
nzone = 0; kzone = -1
for k = zmin:zmax
  for j, i = 1:n
    if kzone(i,j,k) != -1 then
      cycle
    end if
    if insas(i,j,k) > 0 then
      kzone(i,j,k) = 0
    else
      nzone = nzone + 1
      kzone(i,j,k) = nzone
      call walk(i,j,k,kzone,nzone)
    end if
  end
end

recursive subroutine walk(i,j,k,kzone,nzone)
```
kzone(i,j,k) = nzone
if (kzone(i+1,j,k) == -1 .and. insas(i+1,j,k)<0)
call walk(i+1,j,k,kzone,nzone)
if (kzone(i-1,j,k) == -1 .and. insas(i-1,j,k)<0)
call walk(i-1,j,k,kzone,nzone)
if (kzone(i,j+1,k) == -1 .and. insas(i,j+1,k)<0)
call walk(i,j+1,k,kzone,nzone)
if (kzone(i,j-1,k) == -1 .and. insas(i,j-1,k)<0)
call walk(i,j-1,k,kzone,nzone)
if (k+1<=zmax .and. kzone(i,j,k+1) == -1 .and. insas(i,j,k+1)<0)
call walk(i,j,k+1,kzone,nzone)
if (k-1>=zmin .and. kzone(i,j,k-1) == -1 .and. insas(i,j,k-1)<0)
call walk(i,j,k-1,kzone,nzone)
end recursive subroutine walk

In this way, all grid points with kzone > 1 are water accessible, and inmem of these grid points are set back to 0, i.e. membrane inaccessible.

4.3.3 Mapping solvent/membrane accessibility to dielectric constants

In this study, we adopted a three-dielectric model to model the membrane-protein electrostatics. The dielectric constants for the three different regions are denoted as \( \varepsilon_{in} \) (solute), \( \varepsilon_{out} \) (solvent) and \( \varepsilon_{mem} \) (membrane), respectively.

The next step is to map the grid labeling information into the dielectric constants at the midpoints on all grid edges. The general principle, to be consistent with Wang et al.,\(^{43}\) is that the dielectric constant of a grid edge should be equal to the dielectric constant in the region where the two flanking grid points reside. When the two neighboring grid points belong to different dielectric regions, the weighted harmonic averaging (WHA) method is used to calculate the “fractional” dielectric constant based on the precise intersection point where the molecular surface cut the grid edge\(^{43}\). Specifically the dielectric constant is assigned as:

\[
\varepsilon = \frac{1}{\frac{a}{\varepsilon_1} + \frac{1-a}{\varepsilon_2}},
\]

(4)
where \( a \) denotes the fraction of the grid edge in region 1. Eqn (4) is applied on three different kinds of interfaces:

\[
\begin{align*}
\epsilon_1 &= \epsilon_{in}, \epsilon_2 = \epsilon_{out} \quad \text{solute and solvent interface} \\
\epsilon_1 &= \epsilon_{in}, \epsilon_2 = \epsilon_{mem} \quad \text{solute and membrane interface} \\
\epsilon_1 &= \epsilon_{out}, \epsilon_2 = \epsilon_{mem} \quad \text{solute and membrane interface}
\end{align*}
\]

The assignment of dielectric constants on the solute and solvent interface is the same as Wang et al.\(^{43}\). We now consider the grid edge between \((i,j,k)\) and \((i+1,j,k)\); the grid edge can be classified according to the rules in Table 4-1. The procedure of assigning the dielectric constants on the membrane related region and interface is as follows, for each of the x-, y-, and z-edges, respectively.

<table>
<thead>
<tr>
<th>insas(i,j,k)</th>
<th>insas(i+1,j,k)</th>
<th>inmem(i,j,k)</th>
<th>inmem(i+1,j,k)</th>
<th>region</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;0</td>
<td>&gt;0</td>
<td>&gt;0</td>
<td>&gt;0</td>
<td>inside solute</td>
</tr>
<tr>
<td>&gt;0</td>
<td>&gt;0</td>
<td>&gt;0</td>
<td>=0</td>
<td>inside solute</td>
</tr>
<tr>
<td>&gt;0</td>
<td>&gt;0</td>
<td>=0</td>
<td>&gt;0</td>
<td>inside solute</td>
</tr>
<tr>
<td>&gt;0</td>
<td>&lt;0</td>
<td>&gt;0</td>
<td>&gt;0</td>
<td>solute and membrane interface</td>
</tr>
<tr>
<td>&gt;0</td>
<td>&lt;0</td>
<td>&gt;0</td>
<td>=0</td>
<td>solute and solvent interface</td>
</tr>
<tr>
<td>&gt;0</td>
<td>&lt;0</td>
<td>=0</td>
<td>&gt;0</td>
<td>solute and membrane interface</td>
</tr>
<tr>
<td>&lt;0</td>
<td>&gt;0</td>
<td>&gt;0</td>
<td>&gt;0</td>
<td>solute and membrane interface</td>
</tr>
<tr>
<td>&lt;0</td>
<td>&gt;0</td>
<td>=0</td>
<td>&gt;0</td>
<td>solute and solvent interface</td>
</tr>
<tr>
<td>&lt;0</td>
<td>&gt;0</td>
<td>=0</td>
<td>=0</td>
<td>solute and solvent interface</td>
</tr>
<tr>
<td>&lt;0</td>
<td>&lt;0</td>
<td>&gt;0</td>
<td>&gt;0</td>
<td>inside membrane</td>
</tr>
<tr>
<td>&lt;0</td>
<td>&lt;0</td>
<td>&gt;0</td>
<td>=0</td>
<td>inside solvent</td>
</tr>
<tr>
<td>&lt;0</td>
<td>&lt;0</td>
<td>=0</td>
<td>&gt;0</td>
<td>inside solvent</td>
</tr>
<tr>
<td>&lt;0</td>
<td>&lt;0</td>
<td>=0</td>
<td>=0</td>
<td>inside solvent</td>
</tr>
</tbody>
</table>

Table 4-1 Different edges of dielectric constants defined by adjacent values of insas and inmem.

For x-edges, fractional membrane edges are only possible with the membrane- solute interface, so that the following pseudo code can be added to the existing dielectric mapping procedure:
If \((\text{inmem}(i,j,k)>0 \text{ or } \text{inmem}(i+1,j,k)>0)\) then
  If \((\text{insas}(i,j,k)>0 \text{ and } \text{insas}(i+1,j,k)>0)\) then
    \(\varepsilon_x(i,j,k) = \varepsilon_{in}\)
  // grid edge in solute
  else if ((\text{insas}(i,j,k)>0 \text{ and } \text{inmem}(i+1,j,k)>0) \text{ or } \text{(insas}(i+1,j,k)>0 \text{ and } \text{inmem}(i,j,k)>0) ) then
    \(\varepsilon_x(i,j,k) = \frac{1}{a + \frac{1-a}{\varepsilon_{in} \varepsilon_{mem}}}\)
  // grid edge between membrane and solute
  else if (\text{insas}(i,j,k)>0 \text{ or } \text{insas}(i+1,j,k)>0) then
    \(\varepsilon_x(i,j,k) = \frac{1}{a + \frac{1-a}{\varepsilon_{in} \varepsilon_{out}}}\)
  // grid edge between solvent and solute
  else if (\text{inmem}(i,j,k)>0 \text{ and } \text{inmem}(i+1,j,k)>0) then
    \(\varepsilon_x(i,j,k) = \varepsilon_{mem}\)
  // grid edge in membrane
  end if
end if

Here \(a\) is the fraction of grid edge in the solute region. The algorithm along the \(y\)-axis is similar to the \(x\)-axis, as follows:

If \((\text{inmem}(i,j,k)>0 \text{ or } \text{inmem}(i,j+1,k)>0)\) then
  If \((\text{insas}(i,j,k)>0 \text{ and } \text{insas}(i,j+1,k)>0)\) then
    \(\varepsilon_y(i,j,k) = \varepsilon_{in}\)
  // grid edge in solute
  else if ((\text{insas}(i,j,k)>0 \text{ and } \text{inmem}(i,j+1,k)>0) \text{ or } \text{(insas}(i,j+1,k)>0 \text{ and } \text{inmem}(i,j,k)>0) ) then
    \(\varepsilon_y(i,j,k) = \frac{1}{a + \frac{1-a}{\varepsilon_{in} \varepsilon_{mem}}}\)
  // grid edge between membrane and solute
  else if (\text{insas}(i,j,k)>0 \text{ or } \text{insas}(i,j+1,k)>0) then
    \(\varepsilon_y(i,j,k) = \frac{1}{a + \frac{1-a}{\varepsilon_{in} \varepsilon_{out}}}\)
  // grid edge between solvent and solute
  else if (\text{inmem}(i,j,k)>0 \text{ and } \text{inmem}(i,j+1,k)>0) then
    \(\varepsilon_y(i,j,k) = \varepsilon_{mem}\)
  // grid edge in membrane
  end if
end if
For the dielectric constant mapping along the z-axis, it also involves the solvent-membrane interface; the algorithm should also take care of this, as follows:

\[
\begin{align*}
\text{If } \left( \text{inmem}(i,j,k) > 0 \text{ or } \text{inmem}(i,j,k+1) > 0 \right) \\
\text{If } (\text{insas}(i,j,k) > 0 \text{ and } \text{insas}(i,j,k+1) > 0) \text{ then} \\
\varepsilon_z(i,j,k) = \varepsilon_{in} \\
\text{else if } ((\text{insas}(i,j,k) > 0 \text{ and } \text{inmem}(i,j,k+1) > 0) \text{ or } (\text{insas}(i,j,k+1) > 0 \text{ and } \text{inmem}(i,j,k) > 0)) \text{ then} \\
\varepsilon_z(i,j,k) = \frac{1}{\frac{a}{\varepsilon_{in}} + \frac{1-a}{\varepsilon_{mem}}} \\
\text{// grid edge between membrane and solute} \\
\text{else if } (\text{insas}(i,j,k) > 0 \text{ or } \text{insas}(i,j,k+1) > 0) \text{ then} \\
\varepsilon_z(i,j,k) = \frac{1}{\frac{a}{\varepsilon_{in}} + \frac{1-a}{\varepsilon_{out}}} \\
\text{// grid edge between solvent and solute} \\
\text{else if } (\text{inmem}(i,j,k) > 0 \text{ and } \text{inmem}(i,j,k+1) > 0) \text{ then} \\
\varepsilon_z(i,j,k) = \varepsilon_{mem} \\
\text{else if } \text{(grid edge is cross the slab)} \\
\varepsilon_z(i,j,k) = \frac{1}{\frac{a}{\varepsilon_{out}} + \frac{1-a}{\varepsilon_{mem}}} \\
\text{// grid edge cross the slab, now } a \text{ is the fraction of grid edge in the solvent region} \\
\end{align*}
\]

Finally, all edges in the water are assigned the dielectric constant of water, all edges in the membrane are assigned the dielectric constant of membrane, and all edges in the protein interior are assigned the dielectric constant of protein. For all the edges crossing different regions, i.e. between any two of water, membrane, or protein, weighted harmonic averages between the two corresponding dielectric constants are assigned.

### 4.3.4 Protein and complex structure preparation

To calibrate the new continuum membrane model for channel detection, we simulated three channel proteins with crystal structures: 1K4C\textsuperscript{136}, a KcsA potassium channel; 5CFB\textsuperscript{137}, an
alpha1 GlyR Glycine receptor; and 5HCJ, a prokaryotic pentameric ligand-gated ion channel. To demonstrate the feasibility of the new continuum membrane model in data intensive binding affinity calculations, we chose the HERG K\(^+\) channel protein, given its importance in drug discovery and availability of high-quality experimental data.

A homology model of HERG K\(^+\) channel was built based on the X-ray crystal structure of KcsA (PDB ID: 1K4C) using MODELLER (version 9.15) with the default setting. The amino acid sequence of HERG K\(^+\) channel was directly extracted from the Swiss-Prot database (accession number: Q12809 and entry: KCNH2_HUMAN). Sequence alignment was generated using CLUSTALX (version 2.1), showing a good match in helices S5, S6, and the pore region, with identity about 44% (Figure 4-1). After automatic model building and loop refinement, candidate models were evaluated based on the DOPE score from MODELLER. The final homology model of the HERG K\(^+\) channel is shown in Figure 4-2, which is found to be highly consistent with a previously reported model based on a different procedure.

**Figure 4-1** Sequence alignment of KcsA and hERG by ClustalX version 2.1. The identified S5 helix, S6 helix, amphipathic helix and pore helix are labeled above the sequence. Asterisks (*): conserved amino acid residues; colons (•): conserved substitutions; dots (.): semi-conserved substitutions.
**Figure 4-2** Comparison of target and parent structures, showing the secondary structure elements in homology models of hEGH (red) and KcsA (blue). The plot shows three orientations of the aligned structure. Top: side view with the binding pocket on the top. Bottom left: viewed from the binding pocket/extracellular side. Bottom right: viewed from the intracellular side.

Initial complex structures of the HERG $K^+$ channel with its inhibitors were generated with the SURFLEX-DOCK program in Sybyl-X (version 1.3). Ten different inhibitors with experimental binding affinities\(^{139}\) were chosen to assess the quality of the MMPBSA procedure,
including astemizole (AST), sertindole (SER), pimozide (PIM), droperidol (DRO), terfenadine (TE0, TE1), domperidone (DOM), loratadine (LOR), mizolaatine (MIZ), perhexilene (PE0, PE1) and amitriptyline (AMI). The terfenadine and perhexilene are chiral molecules with two enantiomers, so both enantiomers were used in the docking.

4.3.5 Molecular dynamics simulation

The protein was first inserted into a membrane layer using the CHARMM-GUI lipid builder148-152. Lipid DPPC was used for the membrane layer with a lipid to water ratio of 29. The solvated membrane system first underwent a 10,000-step energy minimization using a 5,000-step steepest descent followed by a 5,000-step conjugated gradient. The main chain atoms for the protein were then restrained with a force constant of 2 kcal/mol-Å². Subsequently, a 5 ps MD simulation was conducted to heat the system from 0 to 100K followed by a 100 ps MD simulation to heat the system from 100K to 310K. This was then followed with a 5 ns simulation for equilibration. Finally, production MD was run for 50 ns.

4.3.6 MMPBSA calculations of binding affinities

Binding free energies were computed using a revised MMPBSA module124 of Amber 16 or AmberTools 2016134, 153, 154. The production run trajectory was post-processed with CPPTRAJ155 in order to remove the solvent, membrane, and counter ions from the receptor-ligand complex. Snapshots from the last 10 ns of the production run were processed to compute molecular mechanics potential energies and solvation free energies in the MMPBSA procedure. The binding free energy for the protein-ligand complex was computed as the difference between the complex free energy and the sum of the receptor and ligand free energies, as outlined in our
previous work. The electrostatic solvation free energies were calculated using the linearized PBE model as implemented in PBSA. The non-electrostatic solvation free energies were calculated using either the classical model or the modern model as documented previously.

### 4.3.7 Additional computational details

In each PBSA calculation, a finite-difference grid spacing of 0.5 Å was used for MMPBSA calculations, which was found to be sufficient due to MD sampling and the approximate nature of the binding affinity calculation. Production snapshots up to 10ns were found to be sufficient to converge the averaging process used in MMPBSA calculations of these membrane protein ligand complexes. The periodic geometric multigrid solver option was employed with a convergence threshold of $1.0 \times 10^{-3}$, and electrostatic focusing was turned off due to the presence of the membrane. The use of a periodic boundary also allowed a somewhat small fill ratio (i.e. the ratio of the finite-difference box dimension over the solute dimension) of 1.25 to be used in these calculations. The solvation system physical constants were set up as follows. The membrane was modeled as a solid slab as simulated in the explicit water MD trajectories. The water relative dielectric constant was set at 80.0. The membrane dielectric constant was set to be 7.0. And the protein dielectric constant was set to be 20.0 due to the presence of charged ligand molecules. The water phase ionic strength was set to be 150 mM. The lower dielectric region within the molecular solutes was defined with the classical solvent excluded surface model using a water solvent probe and a membrane solvent probe to be optimized as described in Results and Discussion. The default weighted harmonic averaging was
employed to assign dielectric constants for boundary grid edges to reduce grid dependency. Charges and radii were assigned as in the simulation topology files.

4.4 Results and Discussion

4.4.1 Optimization of the new slab membrane model

Given the automatic procedure in place to identify water channels/pores with the depth-first search method, we further optimized the membrane probe value and the slab membrane model (i.e. its thickness) to best reproduce the distributions of buried water molecules in the membrane region as sampled in explicit water MD simulations. Three different membrane proteins with channels were utilized in this optimization: 1K4C, 5HCJ, and 5CFB.

Three different slab definitions were evaluated to set up the continuum membrane model, i.e. the inner and outer faces are chosen to be positioned at (1) the average z-coordinates of nitrogen atoms of the lipid head groups; (2) the average z-coordinates of the phosphorus atoms of the lipid head groups; (3) the average z-coordinates of both nitrogen and phosphorus atoms in the lipid head groups. Here the average z-coordinates are computed from the explicit-water MD simulations.

Next, mprob values were scanned from 1.4 Å upwards to 3.0 Å with an increment of 0.1 Å. The smallest mprob value with which these known channels can be displayed was recorded as the mprob threshold, as shown in Table 4-2 for all three slab membrane definitions. It should be pointed out that a small mprob produces excessive membrane accessibility in the protein interior so that it is more likely for the buried membrane pockets to be connected to the bulk membrane. Excessive membrane accessibility can also be lessened by reducing the membrane
thickness, as in the use of phosphorus atoms to define the boundaries of the continuum membrane. Indeed, our analysis showed this setup caused the least penetration of the continuum membrane into the protein interior, so the smallest mprob (2.7 Å) was needed to capture the water channels/pores for all three tested proteins.

Figure 4-3 shows the rendering of water-channels/pores of the three tested membrane proteins with the optimized mprob. The advantage of the optimal mprob over the default solvent probe of 1.4 Å is apparent by comparing the renderings generated with the two probes. For all the channel proteins, the new model automatically detects the water channels/pores. Figure 4-4 further shows the benefit of the depth-first-search feature that is a must in the new slab membrane model. Without it, it is apparent that none of the water channels/pores can be identified for any of the tested proteins, even using the larger probe for the membrane region.

<table>
<thead>
<tr>
<th>Protein</th>
<th>mthick (Å)</th>
<th>mcenter(Å)</th>
<th>mprob (Å)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mthick=</td>
<td>N⁺–N⁻</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1K4C</td>
<td>39.24</td>
<td>-1.10</td>
<td>&gt;2.2</td>
</tr>
<tr>
<td>5CFB</td>
<td>40.72</td>
<td>64.95</td>
<td>&gt;1.7</td>
</tr>
<tr>
<td>5HCJ</td>
<td>41.11</td>
<td>68.60</td>
<td>&gt;3.0</td>
</tr>
<tr>
<td>mthick=</td>
<td>P⁺–P⁻</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1K4C</td>
<td>36.13</td>
<td>-0.97</td>
<td>&gt;2.2</td>
</tr>
<tr>
<td>5CFB</td>
<td>37.27</td>
<td>64.97</td>
<td>&gt;1.6</td>
</tr>
<tr>
<td>5HCJ</td>
<td>37.89</td>
<td>68.40</td>
<td>&gt;2.7</td>
</tr>
<tr>
<td>mthick=</td>
<td>N⁺P⁺– N⁻P⁻</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1K4C</td>
<td>37.69</td>
<td>-1.04</td>
<td>&gt;2.2</td>
</tr>
<tr>
<td>5CFB</td>
<td>39.00</td>
<td>64.96</td>
<td>&gt;1.7</td>
</tr>
<tr>
<td>5HCJ</td>
<td>39.50</td>
<td>68.50</td>
<td>&gt;3.0</td>
</tr>
</tbody>
</table>

Table 4-2 The thickness of membrane and mprob thresholds based on the different criterion measured from MD simulations. Here the mprob threshold is the minimum value with which the channel is visible with the SES approach. (Top) mthick=|N⁺–N⁻|: The thickness of the membrane slab is defined as the z-distance between the average head group nitrogen atoms of the lipid molecules. (Middle) mthick=|P⁺–P⁻|: The thickness of the membrane slab is defined as the z-distance between the average head group phosphorus atoms of the lipid molecules. (Bottom) mthick=|N⁺P⁺– N⁻P⁻|: the membrane slab is defined as the z-distance between the average head group centers (i.e. the means of nitrogen and phosphorus atoms) of the lipid molecules. The membrane center locations were then computed as the mean of the upper and lower bounds.
Figure 4-3 Solvent-solute interface determined with the new continuum membrane model. Left: mprob is set to be 1.4 Angstroms, the default value of the solvent probe. Right: mprob is set to be 2.7 Angstroms, the optimized value of the membrane probe. Three proteins are tested: 1K4C (top); 5CFB (middle); 5HCJ (bottom).
Figure 4-4 Same as Figure 4-3, except without turning on the depth-first search in the pore region detection. Three proteins are tested: 1K4C (top); 5CFB (middle); 5HJC (bottom).
4.4.2 Impact of the water solvent probe upon agreement with an explicit solvent simulation

It is worth pointing that the agreement of the continuum membrane model also depends on how we model the water accessible region. The standard practice has been to consider the finite size of the water molecule with a predefined probe radius, often taken as 1.4 Å. The probe is then used to compute the solvent excluded surface used as the interface separating the protein interior from the water region. It is apparent that the size of the water accessible pores/channels would depend on how large the water probe is defined. Thus, it is interesting to analyze how well the widely used water probe performs in the context of membrane channel proteins.

This analysis was conducted in the following manner. The distributions of water molecules (in the water pore/channel regions) in explicit water MD simulations were sampled every 50 ps over the course of a 5 ns production run. Note the protein atoms were all restrained to the reference structure after equilibration since the focus was on the water distribution. A total of 100 frames worth of water sampling were collected and were combined into one snapshot for visualization. This water distribution map was used as a reference to evaluate how the hard sphere SES surface behaves with one single adjustable parameter, i.e. the water solvent probe (dprob in Amber/PBSA).

The same three membrane channel proteins were analyzed to address this question. Specifically, the counts for the following disagreements/mismatches were recorded: (1) absence of explicit water molecules in the continuum water accessible regions; and (2) presence of explicit water molecules in the continuum water inaccessible regions. The overall summary of both mismatches is reported in Table 4-3. Sample mismatches are shown in Figure 4-5. It is interesting to note that the standard value of water solvent probe of 1.4 Å is a very reasonable default value, which gives an overall minimum number of inconsistencies between the
continuum and explicit representations of water distributions in the tested membrane channel proteins, at least in the water accessible pore/channel regions that we have focused on.

<table>
<thead>
<tr>
<th>Protein</th>
<th>dprob (Å)</th>
<th>No. solvent region w/o water molecules</th>
<th>No. water molecules in non-solvent region</th>
</tr>
</thead>
<tbody>
<tr>
<td>1K4C</td>
<td>1.2</td>
<td>23</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>18</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>15</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>1.6</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>5CFB</td>
<td>1.2</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>17</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>14</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>10</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>1.6</td>
<td>7</td>
<td>22</td>
</tr>
<tr>
<td>5HCJ</td>
<td>1.2</td>
<td>22</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>9</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>1.6</td>
<td>7</td>
<td>23</td>
</tr>
</tbody>
</table>

Table 4-3 Discrepancies in the solvent accessible region between explicit water MD simulations and the membrane PBSA calculations. Two types of discrepancies were recorded: (1) how many continuum solvent pockets do not have water molecules; and (2) how many explicit water molecules are observed in the non-continuum solvent pockets defined in the membrane PBSA calculation. The water probe (dprob) was scanned from 1.2 Å to 1.6 Å for three different proteins: 1K4C, 5CFB, 5HCJ. The membrane setup has been optimized according to the values given in Table II. For each protein, the samples of water molecules were taken from a 5ns equilibrium MD simulation with all protein atoms restrained to the initial structure, which was obtained from the last snapshot of the unconstrained normal MD, which is also the reference for the water sampling run. The listed values are the averages of 100 snapshots evenly selected from the 5ns MD simulation.
**Figure 4-5** Discrepancy between implicit and explicit water simulations. The protein surface of 1K4C (blue) is overlaid with a bond representation and sampled water positions (yellow). Left: a solvent region defined by the PBSA model but with no explicit water. Right: explicit water is detected in a region where no solvent is defined in the PBSA model.

It is instructive to point out that the inconsistency between the two representations may be due to the setup of the explicit water MD simulation and also to the limitations of MD sampling of water distributions. First, it is well known that isolated water cavities exist in the protein interior, which are disconnected from the bulk water. Unless crystal water molecules were observed and retained in the initial setup of the MD simulations, these isolated cavities are most likely modeled as water free due to the default closeness tolerance used in the placement of explicit water molecules when building the topology files. This issue would lead to the type (1) mismatches described above.

Second, although protein atoms were restrained during the MD simulations, they are not as inflexible as frozen hard spheres as in the case of the continuum solvent model that must use a single mean structure as input. Their motions allow minor structural changes, leading to the
opening and closing of buried water cavities. If the mean structure happens to correspond to a closed form, the continuum model would not capture the water accessible cavity.

Finally, the protein atom cavity radii that were used to present the size of each atom were chosen to be best for energetics and/or stability of the MD simulations. These may or may not be optimal to quantify water accessibility in the protein interior. This points to future efforts to model the protein-water interface more self-consistently based on the consistent energy model as defined by the protein-water force field used in both explicit and implicit simulations.

### 4.4.3 MMPBSA calculations of binding affinities

Finally, as an illustration of our new continuum membrane model, we conducted a set of binding free energy calculations of ten different ligands independently bound to a potassium channel protein. The computed binding affinities and experimental IC$_{50}$ values are summarized in Table 4-4. The correlation analysis between computation and experiment is shown in Figure 4-6. Both the classical and modern nonpolar solvent models (INP=1 and INP=2 respectively) were tested, and the correlations for these two methods are similar, which is consistent with what we expected. Overall good correlations with experiment were observed: with correlation coefficients of 0.79 for INP=1 and 0.73 for INP=2 (due to the smaller range of the data).

<table>
<thead>
<tr>
<th>Name</th>
<th>RTln(IC50)</th>
<th>mthick</th>
<th>mccenter</th>
<th>MMPBSA (INP=1)</th>
<th>MMPBSA (INP=2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amitriptyline</td>
<td>-7.08</td>
<td>36.086</td>
<td>-10.383</td>
<td>-38.23</td>
<td>-9.84</td>
</tr>
<tr>
<td>Perhexline (PE0)</td>
<td>-7.23</td>
<td>36.661</td>
<td>-1.620</td>
<td>-40.73</td>
<td>-12.44</td>
</tr>
<tr>
<td>Perhexline (PE1)</td>
<td>-7.23</td>
<td>36.473</td>
<td>-5.681</td>
<td>-41.96</td>
<td>-13.63</td>
</tr>
<tr>
<td>Mizolastine</td>
<td>-9.15</td>
<td>36.260</td>
<td>1.360</td>
<td>-61.25</td>
<td>-24.59</td>
</tr>
<tr>
<td>Loratadine</td>
<td>-9.58</td>
<td>36.126</td>
<td>-0.969</td>
<td>-52.48</td>
<td>-18.64</td>
</tr>
<tr>
<td>Domperidone</td>
<td>-9.62</td>
<td>35.979</td>
<td>0.707</td>
<td>-59.53</td>
<td>-20.69</td>
</tr>
<tr>
<td>Terfenadine (TE0)</td>
<td>-9.82</td>
<td>36.616</td>
<td>-2.319</td>
<td>-69.67</td>
<td>-29.00</td>
</tr>
<tr>
<td></td>
<td>MMPBSA Binding Affinity (kcal/mol)</td>
<td>Experimental IC50 (kcal/mol)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------</td>
<td>-------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terfenadine (TE1)</td>
<td>-9.82</td>
<td>37.216</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-0.611</td>
<td>-72.91</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-30.56</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Droperidol</td>
<td>-10.61</td>
<td>36.459</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.308</td>
<td>-59.04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-24.78</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pimozide</td>
<td>-10.97</td>
<td>36.910</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.190</td>
<td>-60.04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-20.09</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sertindole</td>
<td>-11.13</td>
<td>36.438</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.161</td>
<td>-62.91</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-23.89</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Astemizole</td>
<td>-12.81</td>
<td>34.133</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-0.043</td>
<td>-67.64</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-26.21</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 4-4** MMPBSA binding affinities (kcal/mol) in comparison with experiment (IC50). The slab membrane geometry (the thickness and z-center in Å) compiled from the explicit solvent MD simulation are also shown for each complex.

**Figure 4-6** MMPBSA binding affinities compared with experimental measurements. Binding affinities are in kcal/mol. Top: MMPBSA was computed with the classical nonpolar solvent model (INP=1), the correlation coefficient is 0.79. Bottom: MMPBSA was computed with the modern nonpolar solvent model (INP=2), the correlation coefficient is 0.73.

**4.4.4 Timing analysis**
Finally, we conducted a timing analysis of the new membrane model. Table 4-5 summarizes the average CPU times over 100 frames that are used for setting up the dielectric grids with or without the membrane model in the MMPBSA calculation of the receptor. We can see the average time for the surface calculation increases by more than four times; this is mainly because two separate SES calls are made, once with the water probe and once with the membrane probe. Furthermore, the SES calculation with the much larger membrane probe is behind the much higher cost in the total SES time due to the longer non-bonded list and much more overlaps among larger probe-augmented atomic volumes. In addition, the grid-labeling step is also about three times slower, though not a significant portion of the overall CPU cost. Finally, the mapping from grid labels to dielectric constants changes little due to the virtually linear nature of the algorithm. Overall the numerical finite-difference PB calculations are about 10% slower with the new continuum membrane model for these typical membrane protein-ligand binding calculations.

<table>
<thead>
<tr>
<th></th>
<th>Globular Protein Setup</th>
<th>Membrane Protein Setup</th>
</tr>
</thead>
<tbody>
<tr>
<td>SES Calculations (s)</td>
<td>3.76</td>
<td>15.63</td>
</tr>
<tr>
<td>Grid Labeling (s)</td>
<td>1.61</td>
<td>4.79</td>
</tr>
<tr>
<td>EPS Mapping (s)</td>
<td>0.21</td>
<td>0.22</td>
</tr>
</tbody>
</table>

Table 4-5 Average CPU times (in seconds) used in setting up the dielectric grid for 100 snapshots in the MMPBSA calculation of the receptor. The membrane-free set up was run using memopt=0, and the membrane setup was run using memopt=1 in Amber/PBSA.
References


18. Altman, M. D.; Bardhan, J. P.; White, J. K.; Tidor, B., Accurate Solution of Multi-Region Continuum Biomolecule Electrostatic Problems Using the Linearized Poisson-Boltzmann


144. Österberg, F.; Åqvist, J., Exploring blocker binding to a homology model of the open hERG K+ channel using docking and molecular dynamics methods. *FEBS letters* 2005, 579, 2939-2944.


CHAPTER 5

A Multi-Scale Method for Dynamics Simulation in Continuum Solvent Models

I: Finite-Difference Algorithm for Navier-Stokes Equation

5.1 Summary

A multi-scale framework is proposed for more realistic molecular dynamics simulations in continuum solvent models by coupling a molecular mechanics treatment of solute with a fluid mechanics treatment of solvent. This chapter reports our initial efforts to formulate the physical concepts necessary for coupling the two mechanics and develop a 3D numerical algorithm to simulate the solvent fluid via the Navier-Stokes equation. The numerical algorithm was validated with multiple test cases. The validation shows that the algorithm is effective and stable, with observed accuracy consistent with our design.

5.2 Introduction

To model a biomolecular system, there are different levels of approximation. The most accurate approach is the high level quantum description [1], which explicitly accounts for the electronic structures in the molecules. However the computational cost of this approach is high for large and complex biomolecular systems. The most efficient approach is the coarse-grained description [2], which averages atomic structures and properties. The intermediate approach is the atom-based description, which preserves the atomic details in molecular structures and models the effects of electronic structure through a few empirical energy terms [3]. On the other
hand, continuum description is widely used for solvents to capture average solvation properties [4].

Since each level of description has its own advantages and limitations, multi-resolution or multi-scale modeling has also been used to study complex molecular systems [2]. In this type of strategies, researchers aim to calculate material properties or system behavior at one resolution/level using information or models from different resolutions/levels. At each level a particular approach is used for the description of a system of interest [2]. For example, the QM/MM (quantum mechanics/molecular mechanics) model is widely used [5, 6].

Another popular multi-scale model is the MM/CM (molecular mechanics/continuum mechanics) model [4]. Langevin and Brownian dynamics are often used to model the motion of particles in the MM/CM model [7]. The development of Langevin/Brownian dynamics dates back to Einstein’s original work, in which the collective motion of particles in water is modeled by the diffusion theory [8]

$$\frac{\partial \rho}{\partial t} = D \frac{\partial^2 \rho}{\partial x^2}. \quad (1)$$

Here $D$ is the mass diffusivity, $\rho$ is the particle density. The average distance after a period of time $t$ is given by

$$\overline{x^2} = 2Dt. \quad (2)$$

and the mass diffusivity is

$$D = \frac{k_B T}{m\gamma}. \quad (3)$$

Here $k_B$ is the Boltzmann constant, $T$ is the temperature, $m$ is the particle mass, $\gamma$ is the particle viscosity coefficient. Eqn (3) indicates that the viscous force is proportional to the particle
velocity. Subsequently Langevin proposed a random force term to study the Brownian motion of a single particle in water [9], which can be summarized as the Langevin equation

\[ m \frac{d^2 x}{dt^2} = -\nabla U(x) - \gamma \frac{dx}{dt} + \sqrt{2\gamma k_B T m} R(t), \]  

where the first term on the right-hand side of the equation is the force due to the particle interaction potential, the second term is the viscosity force in proportion to the particle velocity, and the third term is a stochastic noise term representing the effect of the collisions between the particle and the molecules of the solvent fluid. The solution of the Langevin equation is consistent with eqn (2) when there is no interaction potential force. Note that constant \( \gamma \) in the viscous force term is modeled as being proportional to the particle velocity as indicated by Eisterin’s previous study [8]. In general, it depends on the geometry of the particle, i.e., shape and size, and also depends on how the particle is exposed to the solvent. Thus, it is an approximation to assume all atoms are the same size and shape, and are all exposed to the solvent for simplicity when the Langevin equation is used in MM/CM simulations. Brownian dynamics is also widely used to describe long-time inter-molecular motions [10-12]. It is a simplified version of Langevin dynamics where no acceleration of collective motion is assumed to take place [7]

\[ 0 = -\nabla U(x) - \gamma \frac{dx}{dt} + \sqrt{2\gamma k_B T m} R(t). \]

The constant \( \gamma \) in Langevin/Brownian dynamics algorithms can be obtained via the Stokes equation from a more universal constant of bulk solvent viscosity that is solute independent [13-15]. The Stokes equation is a special case of the Navier-Stokes equation for low Reynolds number and steady state incompressible flow. Brune and Kim modeled proteins as rigid bodies immersed in a viscous incompressible fluid [13]. They computed the motility tensors
distributed on the protein surface, which can be further incorporated into diffusion coefficient (including both diffusional and rotational). An alternative way adopted by Delatorre et. al. can also provide many interesting physical properties [15]. Hubbard and Douglas [14] presented a simple way to compute the viscosity by extending an analogy between hydrodynamics and electrostatics. They showed that the translational friction coefficient $\gamma$ in eqn (4) and eqn (5) has a simple relationship with the bulk solvent viscosity $\eta_0$ and the solute capacity $C$ as

$$\gamma = 6\pi\eta_0C.$$  

Intrinsic viscosity is another important physical property for solvated biomolecular systems and can be obtained by solving the Stokes equation. Douglas and Garboczi [16] found a simple expression of the intrinsic viscosity in term of polarizability, which can be calculated directly according to molecular geometry. Along a similar direction, Zhou et. al presented an interesting method to calculate the translational friction and intrinsic viscosity, and found a good agreement with experiment [17]. Interestingly, viscosity also plays an important role in the stability of solute-solvent interface, which is recently demonstrated by a model with cylindrical solute-solvent interface [18].

Limitations of the Langevin/Brownian dynamics algorithms are obvious. One is that the solute molecules are assumed to be rigid bodies in most studies, while the conformations of biomolecules are quite flexible. Another one is that all atoms of biomolecules are assumed to be 100% surrounded by the solvent, i.e. the solvent dragging force and the stochastic term are applied to all atoms, which is not a good approximation since buried atoms only interact with solute atoms. Besides, all the particles are unrealistically assumed to be of the same size. These approximations often lead to incorrect dynamics or sampling of biomolecular conformations. Finally an important feature lacking in many Langevin and Brownian dynamics studies is the
hydrodynamic effect, a must in the studies of many biochemical processes that are diffusion-dominant. Indeed the hydrodynamic effect plays an important role in determining the rate constants of diffusion-controlled reactions, and it cannot be derived from an intermolecular potential [19]. To date only a few approximated strategies are available to estimate the hydrodynamic effect [20, 21]. These models all rely on the ground-breaking algorithm proposed by Ermak and McCammon [10].

This chapter reports our exploration of a more physical approach in simulating dynamics of biomolecular systems in continuum solvent models. Specifically, the Navier-Stokes equation or the Stokes equation is used to simulate the continuum solvent and is coupled to the Newtonian equation for the particle-based biomolecular solute. In this first attempt, we formulated the necessary physical model to couple molecular mechanics and fluid mechanics. We also developed and validated a new three-dimensional numerical algorithm for the Navier-Stokes equation before our next step to couple it to the Newtonian equation of a molecular solute.

5.3 Theoretical Model

5.3.1 Physical model

We proceed by first defining a Hamiltonian for the entire system. Its degrees of freedom are atomic positions (x) and their velocities (v) for the solute molecular dynamics (MD) region; and fluid element displacements (y) and their velocities (u) for the solvent fluid dynamics (FD) region. For the MD region, all-atom molecular mechanics force fields, either polarizable [22-25] or nonpolarizable [26-28], can be used. Reduced models can certainly be used as well [29, 30]. For the FD region, the solvent fluid can be modeled as an incompressible viscous fluid.
Equations of motion for all the relevant variables in the system are obtained by taking the appropriate derivatives of this Hamiltonian. All variables can then be updated in lock step as a function of time with time integration. Thus the entire time history of the system can be obtained numerically given an appropriate set of initial conditions and boundary conditions.

Conceptually, the Hamiltonian can be written as

\[ H = H_{MD}(x,p_x) + H_{FD}(y,p_y) + H_{MD/FD}(x,p_x;y,p_y). \]  

(7)

where \( p_x \) is the momentum of molecule region and \( p_y \) is the momentum of fluid region.

\( H_{MD} \) is the Hamiltonian of for the MD region that is modeled by molecular mechanics

\[ H_{MD} = U + K, \]

where \( U \) is the force field potential energy and \( K \) is the kinetic energy. \( H_{FD} \) represents the Hamiltonian for the incompressible solvent fluid. There have been different strategies adopted to formulate a Hamiltonian for the Navier-Stokes equation [31, 32]. Here we follow the formulation from [32] by explicitly considering the constraints of the mass and momentum conservations, and the first and second laws of thermodynamics. Briefly, for an adiabatic system, \( H_{FD} \) can be written as

\[ H_{FD} = \int \left[ \frac{1}{2} \rho u^2 + U_{\text{int}}(\rho,s) \right] dV, \]  

(8)

where \( U_{\text{int}} \) is the internal energy density, \( \rho \) is the fluid density, and \( s \) is the entropy. The coupling Hamiltonian, \( H_{MD/FD} = U_{\text{ele}} + U_{\text{vdw}} + U_{\text{hse}} \), consists of three terms. \( U_{\text{ele}} \) is the Poisson-Boltzmann electrostatic solvation energy [33-36]. The nonelectrostatic solvation energy is modeled as two components: the van del Waals component \( U_{\text{vdw}} \) and the hard sphere entropy/cavity component \( U_{\text{hse}} \) [37-41].

It should be pointed out that non-Hamiltonian systems are also widely used in biomolecular simulations, for which additional terms are used to generate specific statistical
ensembles, e.g. the constant temperature and constant pressure ensemble. Among those methods, extended Lagrangian, with artificial coordinates and velocities added into a Lagrangian [42], is widely used. Meanwhile the fluid region governed by the Navier-Stokes equation should also incorporate the thermal fluctuations into its macroscopic hydrodynamics. A common way is to use a stochastic and temperature-dependent noise term in the Navier-Stokes equation as proposed by Landau and Lifshitz [43].

Next we proceed to derive the dynamics equation by first setting $\mathbf{q} = (x, y)$ as the position vector of the system and $\mathbf{p} = (p_x, p_y)$ as the momentum vector of the system. The familiar Newtonian dynamics equation can be derived from the Hamiltonian equation

$$\dot{p} = -\frac{\partial H}{\partial \mathbf{q}}. \quad (9)$$

In the molecule dynamics region, the equation of motion can be expressed symbolically as

$$\dot{\mathbf{p}}_x = -\frac{\partial H_{\text{MD}}}{\partial x} - \frac{\partial H_{\text{MD/FD}}}{\partial x}. \quad (10)$$

$-\frac{\partial H_{\text{MD}}}{\partial x}$ represents the usual force field terms in molecule dynamics simulations. Since $U_{\text{hue}}$ does not depend on atomic positions, the coupling force terms that the atoms feel are only those of electrostatics and van der Waals in nature, i.e., $-\frac{\partial H_{\text{MD/FD}}}{\partial x} = -\frac{\partial U_{\text{ele}}}{\partial x} - \frac{\partial U_{\text{vdw}}}{\partial x}$. An interesting property is that the electrostatic forces $-\frac{\partial U_{\text{ele}}}{\partial x}$ are simply $q\mathbf{E}$ forces, where $q$'s are “free” charges, i.e. atomic point charges in molecular mechanics models [35]. $-\frac{\partial U_{\text{vdw}}}{\partial x}$ are the van der Waals forces from the solvent molecules modeled as continuum [39].
In the fluid dynamics region, as it is modeled as a continuum, the partial derivative is equivalent to the variation at each volume element

$$\vec{p}_y = -\frac{\delta H_{FD}}{\delta y} - \frac{\delta H_{MD/FD}}{\delta y}. \quad (11)$$

As discussed in Supplemental Material S.1, given a small fluid volume element at position $y$ and volume $V$, we impose a variation $\delta y$ on the element. The derivative of the Hamiltonian gives

$$-\frac{\delta H_{FD}}{\delta y} = -\frac{\partial \sigma_{ij}}{\partial y_j} V, \quad (12)$$

where $\sigma_{ij} = -p \delta_{ij} + \mu \left( \frac{\partial (u \cdot y_i)}{\partial y_j} + \frac{\partial (u \cdot y_j)}{\partial y_i} \right)$ is the stress tensor of the fluid [44]. The only significant derivative of $H_{MD/FD}$ is the van der Waals force, which is termed as the external force density ($F$) on the fluid element, i.e.

$$-\frac{\delta H_{MD/FD}}{\delta y} = -\frac{\delta U_{vdw}}{\delta y} = FV. \quad (13)$$

The change of momentum of the fluid volume element is

$$\vec{p}_y = \rho V \frac{\partial \vec{u}}{\partial t} + \rho V (u \cdot \nabla) \vec{u}. \quad (14)$$

Combining eqns (11) and (12) – (14) and dropping the element volume leads to

$$\rho \frac{\partial \vec{u}}{\partial t} + \rho (u \cdot \nabla) \vec{u} = \frac{\partial \sigma_{ij}}{\partial y_j} + F. \quad (15)$$

Given the conservation of volume/mass for the given volume element, i.e. $\nabla \cdot \vec{u} = 0$, the incompressible Navier-Stokes equation is expressed as
\[\rho \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \mu \Delta \mathbf{u} + \mathbf{F} \]
\[\nabla \cdot \mathbf{u} = 0,\]  
(16)

where \( p \) is pressure and \( \mu \) is viscosity.

As shown in S.2, the interface conditions can also be obtained using the variational principle. Introducing the local coordinate system, which consists of one normal direction (\( \mathbf{n} \)) and two tangential directions (\( \mathbf{t}, \mathbf{\tau} \)) at a certain point on the interface, i.e.,

\[
\begin{align*}
\mathbf{n} &= \cos \alpha_1 \mathbf{i} + \cos \alpha_2 \mathbf{j} + \cos \alpha_3 \mathbf{k} \\
\mathbf{t} &= \cos \beta_1 \mathbf{i} + \cos \beta_2 \mathbf{j} + \cos \beta_3 \mathbf{k} \\
\mathbf{\tau} &= \cos \gamma_1 \mathbf{i} + \cos \gamma_2 \mathbf{j} + \cos \gamma_3 \mathbf{k}.
\end{align*}
\]  
(17)

The interface conditions can be summarized as

\[ -p + p_g - \gamma \kappa + 2\mu \frac{\partial \mathbf{u} \cdot \mathbf{n}}{\partial \mathbf{n}} + f_{\text{dielec}} = 0 \]
\[ \frac{\partial (\mathbf{u} \cdot \mathbf{n})}{\partial t} + \frac{\partial (\mathbf{u} \cdot \mathbf{t})}{\partial \mathbf{n}} = 0 \quad \text{on } \partial \Omega \]  
\[ \frac{\partial (\mathbf{u} \cdot \mathbf{n})}{\partial \tau} + \frac{\partial (\mathbf{u} \cdot \mathbf{\tau})}{\partial \mathbf{n}} = 0,\]  
(18)

where \( \kappa \) is the curvature, \( \gamma \) is surface tension coefficient and \( p_g \) is internal pressure. \( f_{\text{dielec}} \) is the dielectric boundary electrostatic force, which is in the normal direction of the interface [35].

5.3.2 Mathematical model for the fluid dynamic region

In the fluid domain of the physical model, we need to solve a 3D incompressible Navier-Stokes equation (eqn(16)). The fluid domain is typically contained in a rectangular box, with one or multiple interior domain(s) excluded due to the presence of solute molecules or a different phase of fluids, e.g., air bubbles. The conditions at the outer boundary of the rectangular \( R \) are
\[
\begin{align*}
\mathbf{u}|_{y=y_{\text{min}}} &= 0, & \mathbf{u}|_{x=x_{\text{max}}} &= 0 \\
\frac{\partial \mathbf{u}}{\partial y}|_{y=y_{\text{min}}} &= 0, & \frac{\partial \mathbf{u}}{\partial y}|_{y=y_{\text{max}}} &= 0 \\
\mathbf{u}|_{z=z_{\text{min}}} &= 0, & \mathbf{u}|_{z=z_{\text{max}}} &= 0 \\
p &= 0,
\end{align*}
\]

which represents a pipe flow in the \( y \) direction. Inside the domain \( R \) there are the cavities, the union of which is denoted by \( \Omega \). For the sake of simplicity in developing the fluid dynamics algorithm, the interior \( \Omega \), which is occupied by biomolecules or another phase of fluid, is modeled as ideal gas with pressure \( p_g = \lambda / V_g \) in this initial development. Here \( V_g \) is the volume of the air bubble, \( \lambda \) is a gas constant. Eqn (18) can also be written in component form as

\[
\begin{align*}
\mathbf{u}_n \cos \alpha_1 + v_n \cos \alpha_2 + w_n \cos \alpha_3 &= \frac{1}{2\mu} (p - p_g + \gamma\kappa) \\
\mathbf{u}_n \cos \beta_1 + v_n \cos \beta_2 + w_n \cos \beta_3 &= -u_t \cos \alpha_1 - v_t \cos \alpha_2 - w_t \cos \alpha_3 \\
\mathbf{u}_n \cos \gamma_1 + v_n \cos \gamma_2 + w_n \cos \gamma_3 &= -u_t \cos \alpha_1 - v_t \cos \alpha_2 - w_t \cos \alpha_3.
\end{align*}
\]

where \( \mathbf{u} = u\mathbf{i} + v\mathbf{j} + w\mathbf{k} \), and the subscript of the velocity denotes the derivative with respect to the designated direction.

### 5.4 Algorithms

The solution for the above mathematical problem consists of three steps at each time step. The first step is to solve the nonlinear equation for velocity. The backward Euler method instead of the Crank-Nicholson method was adopted to achieve better stability [45]. To decouple velocity and pressure, we used a semi-implicit method [46], i.e. the nonlinear convection term and pressure term are treated explicitly and the viscosity term is treated implicitly when updating velocity. The second step is to compute pressure, which can be obtained by solving a simple
Poisson equation [44]. The third step is to update the interface implicitly via the level set method [47].

### 5.4.1 Solver for velocity and pressure

As discussed in S.4 [45, 46], with the introduction of the backward Euler method and the second order semi-implicit integration scheme, the velocity solver is equivalent to the Helmholtz equation

\[
\Delta u^{k+1} - \frac{3}{2\mu \Delta t} u^{k+1} =
\begin{cases}
\frac{1}{\mu} \left( (u \cdot \nabla u)^{k+1} + \nabla \tilde{p}^{k+1} + \frac{-4u^k + u^{k-1}}{2\Delta t} - F^{k+1} \right), & \text{outside} \\
\frac{-4u^k + u^{k-1}}{2\mu \Delta t}, & \text{inside}
\end{cases}
\]

where

\[
\tilde{p}^{k+1} = 2p^k - p^{k-1},
\]

\[
(u \cdot \nabla u)^{k+1} = 2(u \cdot \nabla u)^k - (u \cdot \nabla u)^{k-1}.
\]  

As shown in S.5, given the updated velocity, the pressure solver is simplified to a Poisson equation as

\[
\nabla^2 p^{k+1} = -\nabla \cdot ((u^{k+1} \cdot \nabla) u^{k+1}) + \nabla \cdot F^{k+1}.
\]  

### 5.4.2 Updating the domain interface

As shown in S.6-S.8, the interface is represented implicitly by a level set function [47]. The level set function is initialized as a signed distance to the interface. Given a moving interface with velocity \( \mathbf{u} \), the level set function can be updated by the following equation

\[
\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0.
\]
To enforce the level set as a signed distance function, the reinitializing algorithm is applied after each update.

5.4.3 Augmented strategy

The augmented strategy is adopted to enforce the boundary conditions when solving pressure or velocity. We set up the augmented variables $G$ as the jump conditions of $u_n$ and $p_n$ when solving $u$ and $p$, respectively [45]. Since the degrees of freedom for $G$ is much smaller than that of the entire set of grid points after discretization, we followed the strategy by solving $G$ first, and then plugging it back to update the velocity or pressure. As shown in S.9-S.12, the equations for pressure or velocity together with its boundary conditions lead to two linear equations for pressure or velocity and its augmented variable $G$. Then $G$ is solved by combining the two equations, symbolically, as

$$EG = F,$$  \hspace{1cm} (25)

where $E$ and $F$ are large dense matrices in practice, which makes the linear system computationally challenging to solve. S.9 outlines detailed procedure on how to obtain these matrices before its solution.

5.5 Results and Discussion

Several test cases were presented to illustrate the proposed physical model and the effectiveness and accuracy of the associated numerical algorithm. The numerical algorithm is first validated with an analytical system of a 3D single spherical bubble in the fluid. Next we show the oscillation dynamics of the single bubble under different solvent viscosity values. Finally we illustrate the more complex dynamics of multiple air bubble merging processes to
assess the numerical procedures’ robustness in handling processes that involve the molecular association.

5.5.1 Accuracy Analysis

Our discretization procedure is based on a first-order interpolation scheme and hence first order accuracy is expected for both velocity and pressure. This was demonstrated with a grid refinement analysis that leads to a quantitative estimation for the order of the accuracy of our numerical method. The analytical solution for the single sphere test case is given in S.13.

Several simulations are performed using different grid sizes. The time step of the integration was set to be in proportional to the grid size as in previous developments [45]. The error is computed as

$$\|E_u\|_\infty = \max_{\text{outside}} \left\{ \sqrt{|U_{ijk} - u(x_i, y_j, z_k)|^2 + |v_{ijk} - v(x_i, y_j, z_k)|^2 + |w_{ijk} - w(x_i, y_j, z_k)|^2} \right\},$$

$$\|E_p\|_\infty = \max_{\text{outside}} \left\{ |P_{ijk} - p(x_i, y_j, z_k)| \right\},$$

which is the difference between the analytical solution and the numerical solution in the fluid after one time step. Figure 5-1 shows the linear regression of the error data in log plot, indicating that the first order accuracy is indeed satisfied: order of 1.30 for velocity and 1.58 for pressure.
Figure 5.1 Order analysis of max errors in velocity and pressure for the numerical procedures used in solving NSE for the analytical test case. Both axes are in the log scale, so the regression slope represents the order of decreasing error with decreasing grid spacing $h$.

5.5.2 Single Bubble Dynamics

Since our interests are in realistic biomolecular systems, the radius of the bubble at equilibrium is chosen to be 5nm, about the radii of typical proteins. The default water parameters are also chosen to be those under room temperature (300K): the viscosity coefficient $\mu = 8.5 \times 10^{-4} Pa\cdot s$, the surface tension coefficient $\gamma = 7.2 \times 10^{-2} N/m$ and the bulk density $\rho = 1.0 \times 10^3 kg/m^3$. The simulation time step is 1ps, which is found to be sufficient for the accuracy and stability for the dynamics simulated.
The simulation was initiated from a non-equilibrium state and would proceed to the equilibrium state within finite time due to the presence of viscosity in the solvent fluid. Here the equilibrium state is reached when the air bubble radius reaches 5nm when the solvent velocity is zero and the solvent pressure is at equilibrium with the environment. Specifically, initial velocity is set as \( u_0 = v_0 = w_0 = 0 \), initial pressure is computed according to eqn (23), and the initial radius of the air bubble is set as 4nm and 6nm, respectively. The grid size is set as 0.625nm. Two different groups of tests were performed, the first test was conducted with the default bulk water viscosity and the second test was done with an artificially low viscosity at 20\% of the default value. The time evolution of the bubble volume is plotted in Figure 5-2.

Figure II shows that the average volume gradually approaches the equilibrium volume in both cases, i.e. the spherical volume with a radius of 5nm. The oscillation dynamics with the default water viscosity is apparently over-damped at the nanometer scale as simulated. Under the low viscosity condition, the fluid flows more freely and the oscillation is apparent before reaching the equilibrium.

Note that there is around 1\% error in the final equilibrium volume with respect to the analytical value, due to the numerical error in this simulation. It is also interesting to note that the error of the equilibrium volume is less for the low-viscosity simulation, which is in part due to the fact that the simulation takes less time to reach the equilibrium state.
Figure 5-2 Time evolutions (in ps) of the volume (in nm$^3$) of the tested single bubble with normal water (up) and low viscous fluid (bottom), respectively. Initial radius of the air bubble is $r_0 = 4\, nm$ and $r_0 = 6\, nm$, respectively. The equilibrium position is at $r = 5\, nm$.

5.5.3 Multiple Bubbles Merging Dynamics

We next studied whether the model is capable of handling the merging of multiple air bubbles due to its relevance to the molecular association. Here we tested two air bubbles and three air bubbles that are placed close to each other, and expect merging of air bubbles once their boundaries touch each other while expanding. Here the simulation box is chosen as 30nm×30nm×30nm and the grid size is set as 1nm. The time integration step remains at 1ps. During our simulations, the air bubbles first expand, and then merge together after touching with each other. The merged large bubble gradually evolves into a sphere and finally the flow becomes equilibrium. The error in the equilibrium volume is about $\sim 2\%$ when comparing with the analytical value based on mass conservation. The snapshots are provided in Figure 5-3 and Figure 5-4 for two bubbles and three bubbles, respectively. The trajectory and velocity flow animations are also provided as Supplemental Material.
Figure 5-3 Snapshots of two-bubble merging dynamics. Simulation is plotted in a box of 20nm×20nm×20nm. The bubble surface is represented as dots.
Figure 5-4 Snapshots of three-bubble merging dynamics. Simulation is plotted in a box of $30nm \times 30nm \times 30nm$. The bubble surface is represented as dots.
5.5.4 Timing Analysis of Simulations

In our algorithm, at each time step the velocity and pressure are first solved and then the level set function is updated. To solve the velocity or pressure, the augmented variables were introduced to impose the interface conditions. In particular, the linear system for the augmented variables, eqn (25), is first solved, and is then plugged back to update the velocity or pressure. As shown in S.9, solving eqn (25) consists of three steps: compute the right-hand side term $F$, compute the coefficient matrix $E$ column by column and then solve the linear equation. Since the column-based calculations are highly independent, the matrix is distributed to different processors and solved it in parallel. Also the parallel linear system solver from Scalapack [48] is adopted to solve the linear equation $EG = F$. All other computations in the numerical algorithm take little time and are implemented in the serial mode. Table 5-1 lists the average time per step of the three air bubbles merging dynamics as the system approaches the equilibrium state. The data shows that computing coefficient matrix and solving the linear equation cost most of the time in the single-thread run and their computation time is greatly reduced when the program is run in parallel with 32 threads. The scaling of the parallel program is illustrated in Figure 5-5, which plots the speedup versus the number of processors tested. The analysis shows that the parallel efficiency is quite reasonable up to 16 threads, but it starts to decay when more nodes are used. Further optimization is clearly possible and will be pursued in our next step.
Table 5-1: Average CPU time per step for the three-bubble merging dynamics. The timing data were collected for ten steps when the simulation is close to equilibrium. The total CPU time is divided into three parts, those for computing the coefficient matrix in eqn (25), solving the linear equation, and other nonparallel procedures. The CPU time for other nonparallel procedures also includes the waiting time for slower threads to finish so it is longer in a parallel run. Note that all CPU times are rounded so they may or may not add up exactly to the total values.

<table>
<thead>
<tr>
<th></th>
<th>32 threads</th>
<th>1 thread</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute coefficient matrix(s)</td>
<td>7.3</td>
<td>149.7</td>
</tr>
<tr>
<td>Solve linear equation (s)</td>
<td>8.1</td>
<td>42.3</td>
</tr>
<tr>
<td>Other nonparallel procedures (s)</td>
<td>4.6</td>
<td>3.3</td>
</tr>
<tr>
<td>Total CPU time (s)</td>
<td>20.0</td>
<td>195.3</td>
</tr>
</tbody>
</table>

Figure 5-5 Parallel scaling of three-bubble merging dynamics. The numbers of processes were chosen as 1, 2, 4, 8, 16, and 32. The speed up ratio is defined as the CPU time for the single-processor run over the CPU time for the tested multiple-processor run. See Table 1 for further detail.
References

7. L.E. Reichl, A Modern Course in Statistical Physics, Verlag Chemie, Wiler-VCH.


Appendix: Supplemental Material

S.1 Derivation of dynamics equations

Given the Hamiltonian defined as

\[ H = H_{MD}(x,p_x) + H_{FD}(y,p_y) + H_{MD/FD}(x,p_x;y,p_y), \]  

(S.1)

where \( x \) represents the coordinates of all solute particles and \( y \) represents the coordinates of all solvent fluid elements, we proceed to derive the dynamics equation by first setting \( q = (x,y) \) as the position vector of the system and \( p = (p_x,p_y) \) as the momentum vector of the system. The familiar Newtonian dynamics can be derived from the Hamilton’s equation

\[ \dot{p} = -\frac{\partial H}{\partial q}. \]  

(S.2)

Here we have adopted the convention that \( p \) and \( q \) represent the moment and position vectors of each particle/element, respectively.

In the molecule dynamics region, the equation of motion for an atom at position vector \( x \) can be expressed symbolically as

\[ \dot{p}_x = -\frac{\partial H_{MD}}{\partial x} - \frac{\partial H_{MD/FD}}{\partial x}. \]  

(S.3)

\[ -\frac{\partial H_{MD}}{\partial x} \] represents the usual force field terms in molecule dynamics simulations. The coupling Hamiltonian has three terms, \( U_{ele} + U_{vdw} + U_{hse} \). Since \( U_{hse} \) does not depend on atomic positions, the coupling force terms that the atoms feel are only those of electrostatics and van del Waals in nature, i.e.,

\[ -\frac{\partial H_{MD/FD}}{\partial x} = -\frac{\partial U_{ele}}{\partial x} - \frac{\partial U_{vdw}}{\partial x}. \]

It is interesting to note that the electrostatic forces
\[-\frac{\partial U_{\text{ele}}}{\partial x}\] are simply the \(qE\) forces, where \(q\)'s are the “free” charges, i.e. atomic point charges in a force field model [1]. \(-\frac{\partial U_{\text{vdw}}}{\partial x}\) are the van del Waals forces from the solvent molecules modeled as continuum [2].

In the fluid dynamics region, consider a small fluid volume element at position \(y\), with volume \(V\) and velocity \(u\). The equation of motion of the fluid element is

\[
\mathbf{p}_y = -\frac{\partial H_{\text{FD}}}{\partial y} - \frac{\partial H_{\text{MD/FD}}}{\partial y}.
\] (S.4)

As shown below, the variational principle will be applied on this element. The partial derivative can also be written in the variational form as

\[
\mathbf{p}_y = -\frac{\delta H_{\text{FD}}}{\delta y} \bigg|_u - \frac{\delta H_{\text{MD/FD}}}{\delta y} \bigg|_u.
\] (S.5)

Here the subscript \(u\) denotes that it is fixed during the variation. Notice here \(y = y(a,t)\) is the Lagrangian coordinate of the volume element, which is fixed on the fluid element and \(a\) is introduced here to denote the actual spatial position [3].

Let us first focus on the variation of \(H_{\text{FD}}\) that is given in the text as

\[
H_{\text{FD}} = \int \left[ \frac{1}{2} \rho u^2 + U_{\text{in}}(\rho,s) \right] dV.
\] (S.6)

For a small fluid volume element at position \(y\) and volume \(V\), we impose a variation \(\delta y\) on the element within time \(dt\), and proceed to compute the variation of \(H_{\text{FD}}\). The process is assumed to be very rapid, i.e., \(\frac{\delta y}{dt} \gg 0\). Since the fluid is incompressible, \(\nabla \cdot \delta y = 0\), the work done to the environment is

\[
dW = \int p \delta y \cdot dS = \int \nabla \cdot (p \delta y) dV = \int \nabla p \cdot \delta y dV + \int p \nabla \cdot \delta y dV = \int \nabla p \cdot \delta y dV, \] (S.7)
where vector $dS$ denotes the surface element of the element with the direction along the normal direction of the surface. Given that the first and second laws of thermodynamics still hold, the internal energy variation can be expressed as

$$\int \delta U_{\text{int}} dV = \int (T \delta s) dV - dW = \int (T \delta s) dV - \int \nabla p \cdot \delta y dV. \quad (S.8)$$

The entropy constraint also gives

$$\int (T \delta s) dV = \int f_{\text{vis}} \cdot \delta y dV - dt \int \nabla \cdot q dV, \quad (S.9)$$

where $q$ is the heat flux and $f_{\text{vis}}$ is the viscous force density. Substitution of eqn (S.9) into eqn (S.8) and the fact that the term involving $dt$ can be ignored as $\frac{\delta y}{dt} \gg 0$ give the variation of the internal energy as

$$\int \delta U_{\text{int}} dV = \int (f_{\text{vis}} \cdot \delta y) dV - \int \nabla p \cdot \delta y dV = \int (f \cdot \delta y) dV. \quad (S.10)$$

where the total force density $f = f_{\text{vis}} - \nabla p = \frac{\partial \sigma_{ij}}{\partial y_j}$ is introduced and

$$\sigma_{ij} = -p \delta_{ij} + \mu \left( \frac{\partial (u \cdot y_j)}{\partial y_j} + \frac{\partial (u \cdot y_i)}{\partial y_i} \right)$$

is the stress tensor of the fluid [4]. Given the assumption that the force density is uniform within the volume element, substitution of eqn (S.6) and eqn (S.10) into the variation of $H_{FD}$ gives

$$- \frac{\delta H_{FD}}{\delta y} \bigg|_u = - \int \frac{\delta U_{\text{int}} dV}{\delta y} = - fV = - \frac{\partial \sigma_{ij}}{\partial y_j} V. \quad (S.11)$$

The variation of $H_{MD/FD}$ is presented next. In Poisson-Boltzmann systems with mobile ions, there is an ionic force term at the Stern layer [1], but it is usually much smaller than other force terms, and is often ignored. If it were not ignored, the ionic force would act upon relevant volume elements. $U_{hse}$ only depends on the interface boundary so that it does not change under
the variation of the volume element. Thus the only significant derivative of $H_{MD/FD}$ is the van der Waals force, which can be treated as the “external force” density ($F$) on the fluid element, i.e.

$$-\delta H_{MD/FD} \bigg|_u = -\delta U_{vde} = FV. \quad (S.12)$$

Finally, the change of momentum of the fluid volume element is

$$\dot{p}_y = \frac{d(\rhoVu)}{dt} = \rhoV \frac{\partial u}{\partial t} + \sum_i \rho V \frac{\partial u}{\partial a_i} \frac{\partial a_i}{\partial t} = \rhoV \frac{\partial u}{\partial t} + \rhoV (u \cdot \nabla)u. \quad (S.13)$$

Combination of eqns (S.5) and (S.11) – (S.13) gives

$$\rhoV \frac{\partial u}{\partial t} + \rhoV (u \cdot \nabla)u = \frac{\partial \sigma_{ij}}{\partial y_j} V + FV$$

$$\Rightarrow \rho \frac{\partial u}{\partial t} + \rho(u \cdot \nabla)u = \frac{\partial \sigma_{ij}}{\partial y_j} + F. \quad (S.14)$$

Including the conservation of volume/mass for the given volume element, i.e. $\nabla \cdot u = 0$, the incompressible Navier-Stokes equation can be expressed as

$$\rho \left( \frac{\partial u}{\partial t} + (u \cdot \nabla)u \right) = -\nabla p + \mu \Delta u + F$$

$$\nabla \cdot u = 0, \quad (S.15)$$

where $p$ is pressure and $\mu$ is viscosity.

### S.2 Derivation of interface conditions

To obtain the interface conditions, an infinitely small fluid disk element $\epsilon$ is introduced with small area $A$ and thickness $h$, and with $h \ll \sqrt{A}$. The disk surfaces are parallel to the boundary interface and one side of the surface is in the molecule dynamics region. Given a variation of the disk position with $\delta r_\epsilon$, 

182
\[
\dot{\mathbf{p}}_e = -\frac{\delta H_{FD}}{\delta \mathbf{r}_e} = \frac{\delta H_{FD/MD}}{\delta \mathbf{r}_e}.
\]  
(S.16)

On the left-hand side
\[
\dot{\mathbf{p}}_e = \rho A \dot{\mathbf{u}}. 
\]  
(S.17)

On the right-hand side, we first introduce the local coordinate system, which consists of one normal direction (\( \mathbf{n} \)) and two tangential directions (\( \mathbf{t}, \mathbf{\tau} \)) at a certain point on the interface, i.e.,
\[
\begin{align*}
\mathbf{n} &= \cos \alpha_1 \mathbf{i} + \cos \alpha_2 \mathbf{j} + \cos \alpha_3 \mathbf{k} \\
\mathbf{t} &= \cos \beta_1 \mathbf{i} + \cos \beta_2 \mathbf{j} + \cos \beta_3 \mathbf{k} \\
\mathbf{\tau} &= \cos \gamma_1 \mathbf{i} + \cos \gamma_2 \mathbf{j} + \cos \gamma_3 \mathbf{k}
\end{align*}
\]  
(S.18)

Stress \(-\frac{\delta H_{FD}}{\delta \mathbf{r}_e}\) only exerts on the disk surface in the fluid region, so that
\[
-\frac{\delta H_{FD}}{\delta \mathbf{r}_e} = \sigma_{\text{pol}} \cdot \mathbf{n} A = (-p + 2\mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \cdot \mathbf{n}) \mathbf{n} + \mu (\frac{\partial (\mathbf{u} \cdot \mathbf{n})}{\partial \mathbf{t}} + \frac{\partial (\mathbf{u} \cdot \mathbf{t})}{\partial \mathbf{n}}) \mathbf{t} + \mu (\frac{\partial (\mathbf{u} \cdot \mathbf{n})}{\partial \mathbf{\tau}} + \frac{\partial (\mathbf{u} \cdot \mathbf{\tau})}{\partial \mathbf{n}}) \mathbf{\tau} A. 
\]  
(S.19)

The last term of eqn (S.16) can be worked out as
\[
-\frac{\delta H_{MD/FD}}{\delta \mathbf{r}_e} = -\frac{\delta U_{\text{ele}}}{\delta \mathbf{r}_e} - \frac{\delta U_{\text{hse}}}{\delta \mathbf{r}_e} - \frac{\delta U_{\text{vdw}}}{\delta \mathbf{r}_e}, 
\]  
(S.20)

where \(-\frac{\partial U_{\text{ele}}}{\partial \mathbf{r}_e} = f_{\text{dielec}} A = \frac{1}{2} \sigma_{\text{pol}} \frac{D_i D_o}{D_{\text{en}}} n A\) is the dielectric boundary electrostatic force [1]. The term \(-\frac{\partial U_{\text{hse}}}{\partial \mathbf{r}_e} = (p_g - \gamma \kappa) n A\) is the pressure and surface tension from the hard sphere entropy term, where \(\kappa\) is the curvature. The van der Waals force, \(-\frac{\partial U_{\text{vdw}}}{\delta \mathbf{r}_e}\), proportional to the volume of element \(Ah\), can be ignored when comparing to the electrostatic forces as \(h\) is infinitely small.

Combining eqns (S.16) and (S.17) and the terms calculated above, the interface conditions can be summarized as
\[-p + p_s - \gamma \kappa + 2\mu \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{n} + f_{\text{dielec}} = 0 \]
\[\frac{\partial (\mathbf{u} \cdot \mathbf{n})}{\partial t} + \frac{\partial (\mathbf{u} \cdot \mathbf{t})}{\partial n} = 0\quad \text{on } \partial \Omega \quad (S.21)\]

\[\frac{\partial (\mathbf{u} \cdot \mathbf{n})}{\partial \tau} + \frac{\partial (\mathbf{u} \cdot \tau)}{\partial n} = 0,\]

**S.3 Solution of the Navier-Stokes Equation: An overview**

The mathematical problem arising from the above physical model is a typical free surface problem, for which many numerical algorithms exist. However the computational efficiency is crucial in our choice of different algorithms because dynamics simulations are often very time-consuming. The first consideration is how to discretize the studied region. The grid may be fixed [5-11] or moving [12-14] and for a fixed grid it may be structured [5-8] or unstructured [9-11], or structured in the fluid and unstructured at the interface [15, 16]. In comparison with a moving grid, the fixed grid does not need a moving operation in the dynamic process. An unstructured grid may simplify the analysis near the interface. However the generation of the unstructured grid is more complex and slower than that of a structured grid. In addition, the unstructured grid needs regeneration when large deformation or topology change happens. The fixed structured grid is used in this study.

The second consideration is how to track the dynamics of the free surface. There are three major kinds of methods based on the fixed structured grid. The first method is Marker And Cell (MAC) method which was originally developed by Harlow and Welch in 1965 [5]. In this method, virtual marker particles are placed in the structured cells and move with the fluid to track the movement of the surface. This method is successful at describing the detail of the fluid. However it has some limitations. A limitation of this method is that additional memory and CPU
time are required for the movement of virtual marker particle. The second limitation is that a special method is needed to compute the normal direction on the surface. The second method is volume of fluid (VOF) method. Hirt and Nichols first adopted this method to study free surface problems [6]. VOF uses the fractional volume of every structured cell to detect the surface. As opposed to MAC, VOF saves much memory and CPU time and can conserve the mass. However, the reconstruction of the surface in VOF and the computation of geometric quantities are not straightforward. The third method is level set method introduced by Osher and Sethian [7]. In this method, the surface is defined by a contour line of a real function defined on the whole region (when discretized, the function is defined at grid points). This method cannot conserve the mass well and is not good at describing the fluid with a sharp shape. However, one advantage of this method is that it is natural to calculate the geometric quantities of the surface. And this method can easily handle the change of topology. To learn more about the detail and the development of these methods, please see reviews [17] for MAC, [18] for VOF and [19] for level set. A coupled level set and VOF method (CLSVOF) has also emerged to adopt the advantages of both methods [20, 21]. For the consideration of efficiency, the level set method is introduced. Another reason is that the surface in our research is supposed to be second-order smooth, which avoids the disadvantage of level set.

The third consideration is how to solve Navier-Stokes equations (NSE). Among Cartesian grid based finite-difference methods, one common approach is to use a stream function–vortices formulation, see for example, [18, 22, 23]. In these methods, the domain is embedded into a rectangular one so that the immersed interface method can be used to get first order accuracy for the vortices. But this formulation is often restricted to two-dimensional problems only. A second type of finite-difference methods is based on the projection method for small to medium
Reynolds number flow, see for example, [24, 25], because of its stability and accuracy. In these methods, the NSE can be solved by solving several Helmholtz/Poisson’s equations, and the boundary is treated as an immersed interface that exerts forces to the surrounding fluid. The force density is chosen so that the boundary condition is satisfied. The problem in finding such force strengths is an inverse problem and it is ill conditioned. In addition, the projection method is based on Helmholtz-Hodge decomposition. The projection method may not work well for open and traction boundary conditions [26]. Also, for the projection method with pressure increment, the solution relies on accurate initial condition of the pressure. There are similarities between local pressure boundary conditions and the local vortices boundary conditions. According to [27], The simplicity of the local pressure boundary condition approach and its easy adaption to more general flow settings make the resulting scheme an attractive alternative to more popular methods for solving NSE in the velocity-pressure formulation. The key idea in the local pressure boundary condition approach is to solve the pressure from the momentum equation. The Dirichlet boundary condition and the incompressibility condition are used to evaluate the Laplacian of the velocity along the boundary, which is the key to the stability. However, this approach does not directly apply to traction boundary conditions for which derivatives of the velocity and the pressure are coupled together. A direct interpolation of the velocity to get the Laplacian from the previous step is likely unstable.

In order to overcome the difficulty, many attempts have been made to utilize a Helmholtz/Poisson solver on irregular domains based on the augmented immersed interface method (IIM) in [18, 28, 29]. In an augmented immersed interface method, an augmented variable (often is one dimension lower than that of the solution) is introduced so that the IIM can be easily implemented. The solution then is a functional of the augmented variable and should
satisfy the boundary or interface condition, which is the augmented equation. In this paper we developed such a method for the incompressible NSE in 3D irregular domains. A zero level set of a Lipschitz continuous function is introduced to represent the boundary, which is more flexible for multi-connected domains. To enforce the free boundary condition at the interface between the gas and the fluid, the augmented method is adopted to solve the resultant nonsymmetrical linear system. At each time step, the linear system of equations for the augmented variables are solved and the pressure is obtained from a Poisson’s equation with the local pressure boundary condition for stability.

**S.4 Solution of velocity**

The NSE (eqn (S.15)) for velocity is first written symbolically as the following ordinary differential equation

$$\frac{du}{dt} = f(t, u)|_{p, F}.$$  \hspace{1cm} (S.22)

Note that both pressure $p$ and force $F$ are known variables, but the time ($t$) dependent function $f$ is unknown. Given the initial condition $u^0$ at $t_0$, a finite-difference time-integration method produces a sequence of $u^0, u^1, u^2, ...$, such that $u^k$ approximates $u_{t_0+k\Delta t}$, where $\Delta t$ is the step size.

The traditional projection method in solving the NSE is based on the Crank-Nicholson (trapezoidal) type of schemes [30]. The Crank-Nicholson method approximates the velocity with the following scheme

$$u^{k+1} - u^k = \Delta t \cdot [f(t_{k+1}, u^{k+1}) + f(t_k, u^k)] / 2.$$  \hspace{1cm} (S.23)

One of the obvious advantages of this approach is that the method can be second order accurate both in space and time. In addition only two time steps are involved in the time integration so
that it is relatively simple to implement. However, it is known that the Crank-Nicholson type
discretization is marginally stable so it is not ideal for non-linear problems or problems with
curved boundaries [31]. To achieve higher stability, a fully implicit discretization for the
diffusion term can be deployed. This is the idea behind the backward Euler method.

In the backward Euler method the approximation of eqn (S.22) is

$$u^{k+1} - u^k = \Delta t \cdot f(t_{k+1}, u^{k+1}).$$  \hspace{0.5cm} (S.24)

The finite-difference approximation of the NSE leads to

$$\frac{u^{k+1} - u^k}{\Delta t} + (u \cdot \nabla u)^{k+1} = -\nabla p^{k+1} + \mu \Delta u^{k+1} + F^{k+1}, \text{outside}$$

$$\frac{u^{k+1} - u^k}{\Delta t} = \mu \Delta u^{k+1}, \text{inside}$$  \hspace{0.5cm} (S.25)

where $p$ is the pressure. For the interior of the bubble, there is some freedom to solve the NSE
since it is modeled as an ideal gas. Nevertheless, there are two purposes in set up an artificial
equation for the velocity inside. 1) The velocity solution is used to set up the free boundary
condition. 2) The new equation in (S.25) is used to patch up a solution domain without a hole(s)
in the middle to facilitate the use of standard finite-difference solvers. Since the bubble is
modeled as an ideal gas, the pressure is set as constant $p_g$ inside the bubble, obeying the ideal
gas law. Of course, the external force $F$ is zero. Finally the nonlinear convection term $u \cdot \nabla u$ is
also set to zero for simplicity. These considerations lead to the simpler form for the interior of
the air bubble in the new equation of (S.25).

As shown in [26, 32], use of higher order semi-implicit model can lead to better accuracy
and stability. Following the scheme in [32], the pressure term and nonlinear convection term are
treated explicitly. Here explicit means these two terms in the equation are known and obtained
by the results from previous steps, so that explicit treatment of the pressure term allows the
decoupling of the computations of velocity and pressure. The viscosity term is treated implicitly, which means they are treated as unknown variables and needed updating at the current time step. The order of the semi-implicit scheme means how many previous steps are used to compute the current step.

For example, considering the pressure term, which is treated explicitly. For the first order approximation, only \( p^k \) is considered, hence \( p^{k+1} = p^k \). For the second order approximation, \( p^k \) and \( p^{k-1} \) are both considered to compute \( p^{k+1} \). The assumption that the function form between \( f \) and \( t \) is linear leads to \( \frac{p^{k+1} - p^k}{\Delta t} = \frac{p^k - p^{k-1}}{\Delta t} \), so that the pressure can be updated as \( p^{k+1} = 2p^k - p^{k-1} \).

Next consider the term \( \frac{du}{dt} \), which is treated implicitly. This means that \( u^{k+1} \) is used and the result from the previous steps is used to write the finite-difference equation as accurately as possible. In the first order approximation, only \( u^{k+1}, u^k \) are used, leading to a finite-difference expression simply as \( \frac{(u^{k+1} - u^k)}{\Delta t} \) to approximate \( \frac{du}{dt} \) at time \( k+1 \). For the second order approximation, \( u^{k+1}, u^k, u^{k-1} \) are used. To approximate \( \frac{du}{dt} \) at time \( k+1 \) expression \( \frac{(u^{k+1/2} - u^{k-1/2})}{\Delta t} \) is apparently better. Define \( f_{k+1} = u^{k+1} - u^k \) and \( f_k = u^k - u^{k-1} \) below to simply our discussion. To approximate \( f_{k+1/2} \) with \( f_{k+1}, f_k \) in the second-order, we assume that a linear function form exists between \( f \) and \( t \) so that \( \frac{f_{k+1/2} - f_{k+1}}{\Delta t / 2} = \frac{f_{k+1} - f_k}{\Delta t} \), i.e., \( f_{k+1/2} = \frac{3f_{k+1} - f_k}{2} \).

The assumption leads to
\[
\frac{\partial \mathbf{u}}{\partial t}
\bigg|_{t_{k+1}} = \frac{f_{i+1} - f_i}{\Delta t} = \frac{3u^{k+1} - 4u^k + u^{k-1}}{2\Delta t}. \quad (S.26)
\]

Thus the second order semi-implicit scheme can be shown as

\[
\begin{align*}
3u^{k+1} - 4u^k + u^{k-1} &= \frac{u \cdot \nabla u)^{k+1}}{2\Delta t} \quad \text{outside} \\
3u^{k+1} - 4u^k + u^{k-1} &= \mu \Delta u^{k+1} \quad \text{inside}
\end{align*}
\]

where

\[
p^{k+1} = 2p^k - p^{k-1}
\]

\[
(u \cdot \nabla u)^{k+1} = 2(u \cdot \nabla u)^k - (u \cdot \nabla u)^{k-1}
\]

Rearrangement of eqn (S.28) leads to a Helmholtz equation that can be solved with standard finite-difference algorithms

\[
\Delta u^{k+1} - \frac{3}{2\mu \Delta t} u^{k+1} =
\]

\[
\begin{cases}
\frac{1}{\mu} \left( (u \cdot \nabla u)^{k+1} + \nabla p^{k+1} + \frac{-4u^k + u^{k-1}}{2\Delta t} - F^{k+1} \right), & \text{outside} \\
\frac{-4u^k + u^{k-1}}{2\mu \Delta t}, & \text{inside}
\end{cases}
\]

(S.29)

Here vector \( \mathbf{u} \) has three components. i.e. \( \mathbf{u} = (u, v, w) \). Eqn (S.29) is solved subject to the boundary condition as in eqn (S.21). To ensure continuous mass transport, the velocity should also be continuous across the interface. Introducing \( [\mathbf{u}] = \mathbf{u}_{out} - \mathbf{u}_{in} \) to denote the jump condition across the interface, the jump condition can be written as

\[
[\mathbf{u}] = 0. \quad (S.30)
\]

Eqn. (S.29) is solved by allowing \( [\mathbf{u}] \) to take any value so that the boundary conditions, eqns (S.21) and (S.30), are satisfied via a self-consistent algorithm to be discussed in S.9 and S.10.
S.5 Solution of pressure

Taking the gradient of the first equation of eqn (S.15):

\[
\frac{\partial (\nabla \cdot \mathbf{u})}{\partial t} + \nabla \cdot ((\mathbf{u} \cdot \nabla) \mathbf{u}) = -\nabla \cdot \nabla p + \mu \nabla \cdot (\Delta \mathbf{u}) + \nabla \cdot \mathbf{F}.
\] (S.31)

The second equation of eqn (S.15) shows

\[
\nabla \cdot \mathbf{u} = 0.
\] (S.32)

Note also that

\[
\nabla \cdot (\Delta \mathbf{u}) = 0.
\] (S.33)

Since all components are zero after taking the divergence twice. Simplification of eqn (S.31) with eqn (S.32) and eqn (S.33) gives

\[
\Delta p^{k+1} = -\nabla \cdot (\mathbf{u}^{k+1} \cdot \nabla) \mathbf{u}^{k+1} + \nabla \cdot \mathbf{F}^{k+1}.
\] (S.34)

Eqn (S.34) is solved using the Dirichlet boundary condition eqn (S.21) at the inner interface. Of course only the first equation related to pressure is used. Note too that eqn (S.34) is only used to model the outside fluid region. To facilitate the use of the standard finite-difference solver, the equation of the region inside of the bubble is set to be

\[
\Delta p^{k+1} = 0.
\] (S.35)

To simplify the model, the jump condition is set as continuous across the interface

\[
[p] = 0, \partial \Omega
\] (S.36)

while \([p_n]\) is allowed to take any value so that boundary conditions, eqns (S.21) and (S.36), are satisfied via a self-consistent algorithm to be discussed in S.9 and S.11.
S.6 Representation of irregular domain interface

The level set function is used to represent the irregular domain interface and compute its geometrical characteristics, such as the local normal direction, the local curvature and so on. The normal direction can be computed by

\[ n = \frac{\nabla \phi}{|\nabla \phi|} = \frac{1}{\sqrt{\phi_x^2 + \phi_y^2 + \phi_z^2}} (\phi_x, \phi_y, \phi_z). \]  \hspace{1cm} (S.37)

The two tangential directions can be chosen arbitrarily. In this study, the two tangential directions are chosen as:

\[ t = \frac{1}{\sqrt{\phi_y^2 + \phi_z^2}} (\phi_y, -\phi_z, 0), \quad \text{if } \phi_y^2 \geq \phi_z^2 \]  \hspace{1cm} (S.38)

\[ \tau = \frac{1}{\sqrt{\phi_x \phi_y + \phi_y \phi_z + (\phi_x^2 + \phi_y^2)^2}} (\phi_x, \phi_y, \phi_z, -\phi_y^2 - \phi_x^2) \]

or

\[ t = \frac{1}{\sqrt{\phi_z^2 + \phi_x^2}} (\phi_z, 0, -\phi_x), \quad \text{if } \phi_y^2 < \phi_z^2 \]  \hspace{1cm} (S.39)

\[ \tau = \frac{1}{\sqrt{\phi_x \phi_z + \phi_x \phi_y + (\phi_x^2 + \phi_z^2)^2}} (-\phi_x, \phi_y, \phi_x^2 + \phi_z^2, -\phi_z, \phi_z) \]

The mean curvature \( \kappa \) is computed by

\[ \kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}. \]  \hspace{1cm} (S.40)
S.7 Update of moving interface

The update algorithm is based on the constraint that the zero level set of the level set function always coincides with the moving interface at all time, i.e. \( \frac{d\phi}{dt} = 0 \). This constraint leads to the following equation [33]

\[
\frac{d\phi}{dt} = \frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x} \frac{dx}{dt} + \frac{\partial \phi}{\partial y} \frac{dy}{dt} + \frac{\partial \phi}{\partial z} \frac{dz}{dt} = \frac{\partial \phi}{\partial t} + \mathbf{V} \cdot \nabla \phi = 0, \tag{S.41}
\]

which assumes that the interface moves with velocity \( \mathbf{V} \). Since the geometry of the moving interface is only dependent on the normal component of the velocity, (the tangential component only changes the parameterization), eqn (S.41) can be rewritten as

\[
\frac{\partial \phi}{\partial t} + \eta \mathbf{n} \cdot \nabla \phi = \frac{\partial \phi}{\partial t} + \eta \frac{\nabla \phi}{|\nabla \phi|} \cdot \nabla \phi = \frac{\partial \phi}{\partial t} + \eta |\nabla \phi| = 0, \tag{S.42}
\]

where \( \eta = \mathbf{V} \cdot \mathbf{n} \) is the component of velocity in the normal direction.

As discussed in [7], the most straightforward discretization method for eqn (S.42) relies on a central-difference scheme as proposed by Lax & Friedrichs [34]. However it suffers from excessive information (in our case: the level set function) diffusion on the interface for problems with computational domains limited to a finite region. In these problems, the central-difference scheme creates spurious wave at the interface because it does not utilize the direction of propagating characteristics. Instead the first-order Propagation of Surfaces under Curvature algorithms (PSC) by Osher & Sethian is adopted to accurately capture the formation of sharp gradients and cusps in the moving fronts and to better handle topological merging and breaking [7].

The discretization scheme can be better understood if assuming the \( x \)-direction as the
normal direction [33]. Given this simplification, eqn (S.42) can be rewritten as

$$\frac{\partial \phi}{\partial t} + \frac{a\phi_x}{|\phi_x|} \cdot \phi_x = \frac{\partial \phi}{\partial t} + a|\phi_x| = 0,$$

(S.43)

where $a$ is the velocity projection in the $x$-direction and $\phi_x$ is the gradient of $\phi$ in the $x$-direction.

Consider the gradients of $\phi$ on both sides of the interface, $\phi_x^+$, $\phi_x^-$. If $a\phi_x^+$ and $a\phi_x^-$ are both positive, the algorithm sets $\phi_x = \phi_x^+$. If $a\phi_x^-$ and $a\phi_x^+$ are both negative, it sets $\phi_x = \phi_x^-$. If $a\phi_x^+ \leq 0$ and $a\phi_x^- \geq 0$, it sets it as $\phi_x = 0$. Finally if $a\phi_x^- \geq 0$ and $a\phi_x^+ \leq 0$, it treats the shock by setting $\phi_x$ to either $\phi_x^-$ or $\phi_x^+$, depending on which gives the largest magnitude for $a\phi_x$. Combining those, $\phi_x$ can be approximated by the following equations

$$\phi_x^2 = \max(\max(\phi_x^-,0)^2, \min(\phi_x^+,0)^2), a > 0$$
$$\phi_x^2 = \max(\min(\phi_x^-,0)^2, \max(\phi_x^+,0)^2), a < 0$$

(S.44)

Given the above introduction, the algorithm to update the level set function on each grid can be obtained by generalizing to 3D as [33, 35]

$$\phi^{k+1} = \phi^k + \Delta t[\max(\eta,0)\nabla^+ + \min(\eta,0)\nabla^-].$$

(S.45)

where

$$\nabla^+ = [\max(D^{-\phi^k},0)^2 + \min(D^{+\phi^k},0)^2 +$$
$$\max(D^{-\phi^k},0)^2 + \min(D^{+\phi^k},0)^2 +$$
$$\max(D^{-\phi^k},0)^2 + \min(D^{+\phi^k},0)^2 +]^{1/2},$$

(S.46)

$$\nabla^- = [\min(D^{-\phi^k},0)^2 + \max(D^{+\phi^k},0)^2 +$$
$$\min(D^{-\phi^k},0)^2 + \max(D^{+\phi^k},0)^2 +$$
$$\min(D^{-\phi^k},0)^2 + \max(D^{+\phi^k},0)^2 +]^{1/2}.$$

(S.47)

Here $D^{-x/y/z}$ is the backward difference operator in the $x/y/z$ direction, $D^{+x/y/z}$ is the forward difference operator in the $x/y/z$ direction. In all these operations the standard finite-difference notation is used, for example,
\[ D_{ij}^{n+1} = \frac{\phi_{i+1,j,k} - \phi_{i,j,k}}{\Delta x}. \]  

(S.48)

**S.8 Reinitialization of level set function**

The finite-difference algorithm (S.45) shows that every level set is moving with the normal velocity component \( \eta \). However such time evolution of \( \phi \) generally does not guarantee that it remains as a signed distance function even if initially it is. For the sake of numerical accuracy and stability, we would like to keep it close to a distance function throughout the time integration via a reinitialization process. Specifically \( \phi \) is reinitialized by finding a new \( \phi \) with the same zero level set but with \( |\nabla \phi| = 1 \). This condition guarantees that the reinitialized function is a signed distance function [33]. As shown in [33], a typical reinitialization method is implemented by the following Hamilton-Jacobi equation

\[ \frac{\partial \phi}{\partial t} + \text{sign}(\phi)(|\nabla \phi| - 1) = 0. \]  

(S.49)

Here \( \text{sign}(\phi) = 0 \) is the sign function that keeps the zero level set unchanged and switch the direction when \( \phi \) changes sign; and \( t \) is a pseudo-time, i.e., iteration step. Its discretization is similar to the one for equation (S.42) (i.e. eqn (S.45) to eqn (S.47)). As shown in [36], the iteration is guaranteed to converge successfully. The method ensures that the level set function near zero level set is close to the distance function, which is required by other numerical treatments in this model.
S.9 Algorithm in solving the augmented variables

By setting up fictitious equations in the ideal gas region, we can afford to deploy a standard finite-difference solver to solve eqn (S.29) for velocity and solve eqn (S.34) for pressure in the entire rectangular domain [37]. The strategy is to set up augmented variables as the jump conditions of \( u_n \) and \( p_n \) when solving pressure or velocity. As shown below (see S.12), the immersed interface method is used to set up finite-difference equations for pressure and velocity, which is to add a correction term to the right-hand side of both equations [38]. These variables are updated in an iterative procedure. At each iteration step, the augmented variables, i.e. jump conditions, can be used to set up finite-difference equations to solve the equations in the entire rectangular domain. The outer loop over the augmented variables terminate only when the free boundary condition, eqn (S.21), is satisfied.

In this so-called augmented method, the coupled finite-difference equations can be expressed symbolically as

\[
AU + BG = F_1. \tag{S.50}
\]

Here \( U \) is the original variable (pressure or velocity) and \( G \) is the augmented variable; \( A \) and \( B \) are the coefficient matrices; and \( F_1 \) is the right-side term of eqn (S.29) or eqn (S.34). In addition, the free boundary condition, eqn (S.21), offers another constraint on the interface for both pressure and velocity. It is shown in S.10 and S.11 that the constraint leads to a second finite-difference equation on pressure or velocity symbolically as

\[
CU + DG = F_2. \tag{S.51}
\]
From eqn (S.50) and eqn (S.51), the Schur complement system for the augmented variable can be obtained by eliminating $U$ as

$$(D - CA^{-1}B)G = F_2 - CA^{-1}F_1.$$  \hfill (S.52)

Use of short-handed notion $E = D - CA^{-1}B$ to represent the coefficient matrix in (S.52) leads $EG = F$.

Eqn (S.52) is non-trivial to solve because it is quite involving to compute the required matrix-vector multiplications for both sides of the equation. However, as discussed in detail in [37], these matrix-vector multiplications can be obtained for free when an iterative procedure is utilized. To understand the details, note that our goal is to find a set of suitable augmented variables to satisfy the free boundary condition or in its matrix form, $CU + DG = F_2$. Thus it is natural to set the residue for the iterative procedure as

$$R(G) = CU + DG - F_2.$$  \hfill (S.53)

When $G = 0$,

$$U(G = 0) = A^{-1}F_1.$$  \hfill (S.54)

Combining eqn (S.53) and eqn (S.54)

$$F_2 - CA^{-1}F_1 = -R(G = 0).$$  \hfill (S.55)

which turns out to be the right-hand side term of eqn (S.52). On the other hand, the left-hand side of eqn (S.52) can be computed as

$$(D - CA^{-1}B)G = DG - CA^{-1}(F_1 - AU)$$

$$= (DG + CU(G)) - CA^{-1}F_1$$

$$= (DG + CU(G)) - CU(G = 0)$$

$$= (DG + CU(G) - F_2) - (D \cdot 0 + CU(G = 0) - F_2)$$

$$= R(G) - R(G = 0).$$

Thus the left-hand side is simply the update of the residue vector during the iterative procedure.
In this study we used the following three steps to solve eqn (S.52):

1. Setting augmented variables as zero, and obtain the right-hand side of eqn (S.52) by computing \(-R(G = 0)\).

2. Assuming there are \(N\) augmented variables and running \(N\) groups of computation. Each time, we set \(G = [0, \ldots, 0, c, 0, \ldots, 0]^T\), i.e., for \(j\)th time, the \(j\)th element of \(G\) is a constant \(c\) and other elements as zero. According to eqn (S.56), the \(j\)th column of the coefficient matrix is

\[
[(D - CA^{-1}B)]_{i:N,j} \cdot c = R(G = c) - R(G = 0), \quad (S.57)
\]

so that

\[
[(D - CA^{-1}B)]_{i:N,j} = (R(G = c) - R(G = 0)) / c. \quad (S.58)
\]

3. After obtain the coefficient matrix and the right-hand side term of eqn (S.52), the augmented variable \(G\) can be computed directly. \(G\) can then be plugged into eqn (S.50) to obtain the unknown variable.

In the next two subsections, we specifically address how the residue vectors can be computed for both velocity and pressure, respectively.

**S.10 Solution of velocity using augmented approach**

Given \(N\) irregular grid points, the velocity matrix for the irregular grid points is defined as

\[
V = \begin{bmatrix}
    u^1 & v^1 & w^1 \\
    \vdots & \ddots & \vdots \\
    u^N & v^N & w^N
\end{bmatrix}^T. \quad (S.59)
\]

The matrix for the augmented variables can also be written as
\[ \mathbf{G} = \begin{bmatrix} [u_n^i] & [v_n^i] & [w_n^i] \\ \vdots & \ddots & \vdots \\ [u_n^N] & [v_n^N] & [w_n^N] \end{bmatrix} \]  \quad (S.60)

where \([u_n^i], [v_n^i], [w_n^i]\) are the jump conditions of the normal derivatives of the three components of the velocity at the interface projection point of irregular grid point \(i\). Note too that we have also imposed the continuity condition across the interface while solving for velocity

\[
\begin{bmatrix} [u^i] & [v^i] & [w^i] \\ \vdots & \ddots & \vdots \\ [u^N] & [v^N] & [w^N] \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & 0 \end{bmatrix}^T. \quad (S.61)
\]

Give that velocity is continuous, the velocity gradient on the tangential directions are zero across the interface, i.e.

\[
\begin{bmatrix} [u_1^i] & [v_1^i] & [w_1^i] \\ \vdots & \ddots & \vdots \\ [u_1^N] & [v_1^N] & [w_1^N] \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & 0 \end{bmatrix}^T. \quad (S.62)
\]

and

\[
\begin{bmatrix} [u_t^i] & [v_t^i] & [w_t^i] \\ \vdots & \ddots & \vdots \\ [u_t^N] & [v_t^N] & [w_t^N] \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & 0 \end{bmatrix}^T. \quad (S.63)
\]

The Helmholtz equation for velocity

\[
\Delta \mathbf{u}^{k+1} - \frac{3}{2\mu\Delta t} \mathbf{u}^{k+1} = \begin{cases} 
\frac{1}{\mu}((\mathbf{u} \cdot \nabla \mathbf{u})^{k+1} + \nabla p^{k+1} + \frac{-4\mathbf{u}^k + \mathbf{u}^{k-1}}{2\Delta t} - \mathbf{F}^{k+1}), & \text{outside} \\
\frac{-4\mathbf{u}^k + \mathbf{u}^{k-1}}{2\mu\Delta t}, & \text{inside} 
\end{cases} \quad (S.64)
\]
which is discretized, one component at a time using the immersed interface method (see S.12 below), is discretized as

\[
\Delta u_{ijk}^{k+1} - \frac{3}{2\mu\Delta t} u_{ijk}^{k+1} = \begin{cases} \\
\frac{1}{\mu} ((u \cdot \nabla) u_{ijk}^{k+1} + \nabla p_{ijk}^{k+1} + \frac{-4u_{ijk}^{k} + u_{ijk}^{k-1}}{2\Delta t} - F_{ijk}^{k+1}) + C_{ijk}, & \text{outside} \\
\frac{-4u_{ijk}^{k} + u_{ijk}^{k-1}}{2\mu\Delta t} + C_{ijk}, & \text{inside} \end{cases} \quad (S.65)
\]

Given that the velocity is continuous across the interface, it can be shown that \( C = -BG \) where \( B \) is a coefficient matrix. Eqn (S.65) can then be expressed symbolically as the following form

\[ AV + BG = F. \quad (S.66) \]

where \( A \) and \( B \) are coefficient matrices.

As introduced previously, our goal is to satisfy the free boundary condition in the solution of eqn (S.66). Define the residue elements as

\[
\begin{align*}
  f(3i-2) &= u_n \cos \alpha_i + v_n \cos \alpha_2 + w_n \cos \alpha_3 - \frac{1}{2\mu} (p_i - p_g + T_{surf}) \bigg|_i, \\
  f(3i-1) &= u_n \cos \beta_i + v_n \cos \beta_2 + w_n \cos \beta_3 - (-u_t \cos \alpha_1 - v_t \cos \alpha_2 - w_t \cos \alpha_3) \bigg|_i, \\
  f(3i) &= u_n \cos \gamma_1 + v_n \cos \gamma_2 + w_n \cos \gamma_3 - (-u_t \cos \alpha_1 - v_t \cos \alpha_2 - w_t \cos \alpha_3) \bigg|_i.
\end{align*} \quad (S.67)
\]

where \( i \in (1,2,\ldots,N) \) denotes the irregular point index, so that the residue vector can be written as

\[ R = [f(1), f(2), \ldots, f(3*N)]^T. \quad (S.68) \]

Our goal is to solve for velocity so that \( R = [0,0,\ldots,0]^T \). Note that the residue can be expressed as a linear combination of \( V \) as well as jump conditions of \( V \) and \( V_n \) [29]. Since only
the normal derivatives of the jump conditions $G$ are nonzero according to eqns (S.62) and (S.63),
the residue can be expressed as the linear combination of $V$ and $G$, i.e.

$$R = EV + TG.$$ \hspace{1cm} (S.69)

where $E$ and $T$ are coefficient matrices.

**S.11 Solution of pressure using augmented approach**

The augmented variables in the solution of pressure can be expressed symbolically as

$$G = \left[\begin{array}{c}
p_n^1, p_n^2, \ldots, p_n^N 
\end{array}\right]^T,$$ \hspace{1cm} (S.70)

where $[p_n^i]$ is the jump condition of the normal derivative of pressure at the interface projection point of irregular grid point $i, i \in (1, 2, \ldots, N)$. Upon iterative update of $G$, the free boundary condition needs to be satisfied. That is

$$p_i = 2\mu (u_n \cos \alpha_1 + v_n \cos \alpha_2 + w_n \cos \alpha_3) + p_g - T_{surf}.$$ \hspace{1cm} (S.71)

In addition the continuity condition for pressure is also imposed across the interface

$$[p_1^1, p_2^2, \ldots, p_N^N] = [0, 0, \ldots, 0],$$ \hspace{1cm} (S.72)

so that the jump conditions for the tangential derivatives are also zero, i.e.

$$[p_1^1, p_2^2, \ldots, p_N^N] = [0, 0, \ldots, 0],$$

$$[p_1^1, p_2^2, \ldots, p_N^N] = [0, 0, \ldots, 0].$$ \hspace{1cm} (S.73)

The Poisson’s equation for pressure can be written symbolically as

$$\nabla^2 p = f.$$ \hspace{1cm} (S.74)

Given the interface conditions in eqns (S.70) and (S.72), the immersed interface method (see S.12 below) is used to discretize eqn (S.74) as

$$\nabla_h^2 p_{ijk} = F_{ijk} + C_{ijk}.$$ \hspace{1cm} (S.75)
Since pressure is continuous, the correction term in eqn (S.75) can be written symbolically as
\[ C = -BG, \]
where \( B \) is a coefficient matrix. Thus eqn (S.75) can be written as
\[ Ap + BG = F. \]  \hspace{1cm} (S.76)

For the iterative procedure the residue is defined as
\[ R(G) = [p_0^1 - p_i^1, p_0^2 - p_i^2, ..., p_0^N - p_i^N]^T, \]  \hspace{1cm} (S.77)
where \( p_0^i \) is the pressure at the interface projection point of irregular grid point \( i, i \in (1,2,...,N) \) and \( p_i^i \) is defined by eqn (S.71) for at the interface projection point of irregular point \( i \). Note that \( [p_0^1, p_0^2, ..., p_0^N]^T \) can be expressed as a linear combination of pressure and jump conditions of pressure and their derivatives. And also notice that only the normal direction of the gradient of pressure \( G \) is nonzero according to eqn (S.73). Given these preparations, we have the expression
\[ [p_0^1, p_0^2, ..., p_0^N]^T = Ep + TG \]  \hspace{1cm} (S.78)
and
\[ R(G) = Ep + TG - [p_i^1, p_i^2, ..., p_i^N]^T, \]  \hspace{1cm} (S.79)
where \( E \) and \( T \) are coefficient matrices.

**S.12 Immersed Interface Method**

Following the procedure in \([38]\), the finite-difference method is used to solve the Helmholtz and Poisson equations for velocity and pressure as described above. For a regular grid point where the boundary of the interface does not intersect any grid edge between the grid point and any of its neighboring grid points, both finite-difference functions can be expressed as
\[ \sum_{m} \gamma_{m} \phi(i + m, j + m, k + m) = f(i, j, k). \]  \hspace{1cm} (S.80)
Here $\phi$ is used to indicate the variable (pressure or velocity) to be interpreted and $f$ is the right-hand side of the original equation to be solved. For an irregular grid point where the boundary of interface intersects at least one grid edge between the grid point and its neighboring grid points, a correction term is added to maximize the satisfaction of the jump conditions on the interface as

$$\sum_{m}^{n} \gamma_m \phi(i + i_m, j + j_m, k + k_m) = f(i, j, k) + C(i, j, k). \quad (S.81)$$

Correction term $C(i,j,k)$ is obtained for each irregular grid point so that the local truncation error

$$T(i, j, k) = \sum_{m}^{n} \gamma_m \phi(i + i_m, j + j_m, k + k_m) - f(i, j, k) - C(i, j, k) \quad (S.82)$$

is minimized.

Specifically we expand $\phi(i + i_m, j + j_m, k + k_m)$ in the local coordinate system with a Taylor series with respect to the origin set as irregular grid point $(i,j,k)$ ’s projection point on the interface. Since the variable (pressure or velocity) is continuous across the interface, the following interface relations are used to simplify the Taylor expansion

\[
\begin{align*}
\phi^* &= \phi^- \\
\phi^*_n &= \phi^-_n + [\phi^-_n] \\
\phi^*_t &= \phi^-_t \\
\phi^*_zz &= \phi^-_zz - [\phi^-_n] \chi_{zt} \\
\phi^*_n &= \phi^-_n - [\phi^-_n] \chi_{zt} \\
\phi^*_zt &= \phi^-_zt - [\phi^-_n] \chi_{tt} \\
\phi^*_nt &= \phi^-_nt + \frac{\partial[\phi^-_n]}{\partial t} \\
\phi^*_nt &= \phi^-_nt + \frac{\partial[\phi^-_n]}{\partial \tau} \\
\phi^*_nn &= \phi^-_nn + (\chi_{nt} + \chi_{tt})[\phi^-_n].
\end{align*}
\]
where the subscript "t" represents the outside interface and the subscript "i" represents the inside interface, \( n \) is the normal direction, \( r, \tau \) are two orthogonal tangential directions, and \( n = \chi(t, \tau) \) is interface function expressed in the local coordinate system.

**First order IIM** Following the method in [29], only first order terms are considered in the Taylor expansion, the local truncation error (eqn (S.82)) can be written as

\[
T(i, j, k) = a_i \phi^- + a_j \phi_n^- + a_k \phi_r^- + a_\tau \phi_\tau^- - f^- \\
+ (\hat{T}(i, j, k) - C(i, j, k)) + O(h + \max_m |\gamma_m| h^1),
\]  

(S.84)

where \( a_i, i = 1, \ldots, 4 \), are the linear combinations of \( \gamma_m \), \( \hat{T}(i, j, k) \) is a linear combination of jump conditions and their surface derivatives in eqn (S.83). To minimize \( T(i, j, k) \), the identity

\[
a_i \phi^- + a_j \phi_n^- + a_k \phi_r^- + a_\tau \phi_\tau^- - f^- = 0.
\]  

(S.85)

should be satisfied first, which determines the value of \( \gamma_m \) [29]. After obtaining \( \gamma_m \), \( C(i, j, k) \) is calculated to cancel \( \hat{T}(i, j, k) \) to construct the new equation at the irregular grid point.

**Second order IIM** Following the method in [29], both the first order and second order terms are considered in the Taylor expansion, The local truncation error (eqn (S.82)) can be written as

\[
T(i, j, k) = a_i \phi^- + a_j \phi_n^- + a_k \phi_r^- + a_\tau \phi_\tau^- + a_{ij} \phi_i^+ + a_{jk} \phi_j^+ + a_{k\tau} \phi_k^+ + a_{\tau\tau} \phi_\tau^+ - f^- \\
+ (\hat{T}(i, j, k) - C(i, j, k)) + O(h + \max_m |\gamma_m| h^1),
\]  

(S.86)

where \( a_i, i = 1, \ldots, 10 \), are linear combinations of \( \gamma_m \), \( \hat{T}(i, j, k) \) is a linear combination of jump conditions and their surface derivatives in eqn (S.83). To minimize \( T(i, j, k) \), identity

\[
a_i \phi^- + a_j \phi_n^- + a_k \phi_r^- + a_\tau \phi_\tau^- + a_{ij} \phi_i^+ + a_{jk} \phi_j^+ + a_{k\tau} \phi_k^+ + a_{\tau\tau} \phi_\tau^+ - f^- = 0
\]  

(S.87)
should be satisfied first, which determines the value of $\gamma_m$ [29]. After obtaining $\gamma_m$, $C(i,j,k)$ is calculated to cancel $\hat{T}(i,j,k)$ to construct the new equation at the irregular grid point.

S.13 3D analytical test case

The simplest analytic solution for the mathematical model is that of a spherical air bubble (of radius $R$) in a central-symmetric velocity field without any external force, $\mathbf{F} = 0$. Given the incompressibility requirement ($\nabla \cdot \mathbf{u} = 0$), the velocity must be in the form of $\mathbf{u} = C \frac{\mathbf{r}}{r^3}$ in the fluid outside, where $C$ is a constant. On the other hand, pressure must satisfy the NSE (eqn (S.15)) given the velocity, with the zero boundary condition at infinity. Thus the general solutions of pressure and velocity can be expressed as

$$
\begin{align*}
u_1 &= \frac{C_x}{r^3} \\
v_1 &= \frac{C_y}{r^3} \\
w_1 &= \frac{C_z}{r^3} \\
p_1 &= -\frac{C^2}{2r^4} + D_1,
\end{align*}
$$

where $C$ and $D_1$ are integration constants and $C$ is a function of $R$.

For the air bubble inside, the solution does not need to satisfy the incompressibility requirement $\nabla \cdot \mathbf{u} = 0$ since it is only a pseudo area for the sake of solving NSE (eqn (S.15)). However, for simplicity, it is assumed that velocity changes linearly from 0 at the center to the interface. Thus the pseudo velocity and pressure inside is
\begin{align*}
  u_2 &= \frac{Cx}{R^5} \\
  v_2 &= \frac{Cy}{R^5} \\
  w_2 &= \frac{Cz}{R^3} \\
  p_2 &= -\frac{C^2 r^2}{2R^6} + D_2, \\
\end{align*}

where $D_2$ are the constant.

Since pressure satisfies the continuity requirement, $p_1 = p_2$ on the interface. Thus

\begin{equation}
  D_1 = D_2 \quad (S.90)
\end{equation}

from eqn (S.88) and eqn (S.89). In addition, if the volume of the air bubble is set as 1, $p_g = \frac{1}{R^3}$ at $r = R$.

For simplicity, the system is also assumed to reach equilibrium when $R = 1$, i.e., $u = v = w = 0$ when $R = 1$. Therefore the bubble expands when $R < 1$, and shrinks when $R > 1$.

Combining the free boundary condition with the general pressure solution in the liquid, eqn (S.89) leads to

\begin{equation}
  p_1 - p_g + \frac{2\gamma}{R} = 0 \\
  \Rightarrow D_2 - \frac{1}{R^3} + \frac{2\sigma}{R} = 0 \quad (S.91)
\end{equation}

at $r = R = 1$. For simplicity, the surface tension coefficient is further chosen as $\gamma = 1/2$. This leads to $D_2 = 0$. Eqn (S.90) further leads to

\begin{equation}
  D_1 = D_2 = 0. \quad (S.92)
\end{equation}
The constant $C$ is determined next. Apparently this is only possible when the system is not at equilibrium, $R \neq 1$. Starting from the free boundary condition and substituting the general solution for $p_1$, eqn (S.88) gives

$$-p_1 + p_g - T_{\text{surf}} + 2\mu \frac{\partial u}{\partial n} = 0$$

$$\Rightarrow p_1 - p_g + \frac{2\gamma}{R} + 4\frac{\mu C}{R^3} = 0$$

$$\Rightarrow -\frac{1}{R^3} - \frac{C^2}{2R^4} + \frac{2\gamma}{R} + 4\frac{\mu C}{R^3} = 0.$$  \hspace{1cm} (S.93)

Constant $C$ is thus $4\mu R - \sqrt{16\mu^2 R^2 - 2(R-R^3)}$.

In summary, the analytical solution for the analytical system is

$$u = \begin{cases} \frac{Cx}{r^3} & r > R \\ \frac{Cx}{R^3} & r < R \end{cases}$$

$$v = \begin{cases} \frac{Cy}{r^3} & r > R \\ \frac{Cy}{R^3} & r < R \end{cases}$$

$$w = \begin{cases} \frac{Cz}{r^3} & r > R \\ \frac{Cz}{R^3} & r < R \end{cases}$$

$$p = \begin{cases} -\frac{C^2}{2R^4} & r > R \\ -\frac{C^2r^2}{2R^6} & r < R \end{cases},$$

with constant $C = 4\mu R - \sqrt{16\mu^2 R^2 - 2(R-R^3)}$. The system reaches equilibrium when $R = 1$, $u = v = w = 0$, and $p = 0$. The analytical jump conditions on the interface are, accordingly,
\[
\begin{align*}
\frac{\partial u}{\partial n} &= \left( -\frac{2C}{r^3} \right) |_{r=R} = -\frac{3C}{R^3}, \\
\frac{\partial p}{\partial n} &= \left( \frac{2C^2}{r^5} \right) |_{r=R} - \left( -\frac{C^2r}{R^6} \right) |_{r=R} = \frac{3C^2}{R^5}.
\end{align*}
\]

(S.95)

S.14 References


CHAPTER 6

Exploring A Multi-Scale Method for Molecular Simulations in Continuum Solvent Model: Explicit Simulation of Continuum Solvent As An Incompressible Fluid

6.1 Summary

A multi-scale framework was recently proposed for more realistic molecular dynamics simulations in continuum solvent models by coupling a molecular mechanics treatment of solute with a fluid mechanics treatment of solvent, where we formulated the physical model and developed a numerical fluid dynamics integrator. In this study, we incorporated the fluid dynamics integrator with the Amber simulation engine to conduct atomistic simulations of biomolecules. At this stage of the development, only nonelectrostatic interactions, i.e., van del Waals and hydrophobic interactions are included in the multi-scale model. Nevertheless numerical challenges exist in accurately interpolating the highly nonlinear van del Waals term when solving the finite-difference fluid dynamics equations. We were able to bypass the challenge rigorously by merging the van del Waals potential and pressure together when solving the fluid dynamics equations and by considering its contribution in the free-boundary condition analytically. The multi-scale simulation engine was first validated by reproducing the solute-solvent interface of a single atom with analytical solution. Next, we performed the relaxation simulation of a restrained symmetrical monomer and observed a symmetrical solvent interface at equilibrium with detailed surface features resembling those found on the solvent excluded surface. Four typical small molecular complexes were then tested, both volume and force
balancing analysis showing that these simple complexes can reach equilibrium within the simulation time window. Finally, we studied the quality of the multi-scale solute-solvent interfaces for the four tested dimer complexes and found they agree well with the boundaries as sampled in the explicit water simulations.

6.2 Introduction

Atomistic simulation has become an important tool for studying the structures, dynamics, and functions of biomolecular systems. Nevertheless efficient atomistic simulation of large and complex biomolecular systems is still one of the remaining challenges in computational molecular biology. The computational challenges in atomistic simulation of biomolecular systems are direct consequences of their high dimensionalities. Indeed biomolecules are highly complex molecular machines with thousands to millions of atoms. What further complicates the picture is the need to realistically treat the interactions between biomolecules and their surrounding water molecules that are ubiquitous and paramount important for their structures, dynamics, and functions.

To appreciate these challenges, it is instructive to highlight the two bottlenecks in biomolecular simulations: (1) the cost of each energy evaluation that is determined by the number of particles in a mathematical model; and (2) the number of time steps of dynamics that it takes for sufficient coverage of different conformations. Hundreds of millions of time steps are routinely required in biomolecular simulations to draw statistically significant conclusions. It is often the case that more particles need more time steps for sufficient coverage. Thus the overall simulation cost usually scale exponentially with the number of particles in the mathematical model used in a simulation. Indeed many fundamental and interesting biomolecular processes
remain largely inaccessible to atomistic simulations when system sizes exceed more than a few hundred residues.

Since most particles in biomolecular simulations are to represent water molecules solvating the target biomolecules, an implicit treatment of water molecules allows greatly increased simulation efficiency. Indeed, implicit solvation offers a unique opportunity for more efficient simulations without the loss of atomic-level resolution for biomolecules $^{1-17}$. Advance in implicit solvation, coupled with developments in sampling algorithms, classical force fields, and quantum approximations, will prove useful to the larger biomedical community in a broad range of studies of biomolecular structures, dynamics and functions.

One class of implicit solvent models, the classical Poisson-Boltzmann (PB) solvent model, has become widely accepted in biomolecular applications after over 30 years of basic research and development. Efficient numerical PBE-based solvent models have been widely used to study biological processes including predicting pKa values $^{18-21}$, computing solvation and binding free energies $^{22-31}$, and protein folding $^{32-42}$. However, challenges remain to achieve more consistent, accurate, and robust analysis of biomolecules $^{43-60}$. The existing dielectric model based on molecular solvent excluded surface is a major hurdle for applications of the Poisson-Boltzmann solvent models. This dielectric model is ad-hoc, expensive, and numerically unstable due to its treatment of atoms as hard spheres in molecular simulations.

In our previous study $^{61}$, we explored a multi-scale simulation strategy to explicitly simulate the continuum solvent/solute interface with the solvent fluid dynamics that is coupled to the solute molecular dynamics. This strategy (1) allows a self-consistent treatment of the solvation interactions, i.e. the dielectric interface automatically adjusts to local conformational and energetic fluctuations and is guaranteed to be at the system free energy minimum upon
equilibrium; (2) allows a “soft” and more physical dielectric interface for stable dynamics; (3)
eliminates atom-specific cavity radii that must be defined, dramatically reducing the freely
adjustable parameters of the continuum solvent treatment; (4) eliminates the expensive molecular
surface reconstruction step during dynamics; and (5) eliminates the difficult and expensive
molecular surface-to-atom mapping of dielectric boundary forces and hydrophobic boundary
forces, and applies these surface forces to the continuum solvent instead. In addition, a 3D
numerical algorithm was developed to simulate the implicit solvent via the Navier-Stokes
equation. It should be pointed out that the use of Navier-Stokes equation, instead of Stokes
equation that is sufficient for biomolecular processes of interest, is necessary for the lack of a
predefined solute-solvent interface, or in a “free boundary” problem. Our numerical
algorithm was validated with multiple model test cases, demonstrating its effectiveness and
numerical stability, with observed accuracy consistent with the designed numerical algorithm.

In this study, we intended to explore the feasibility of incorporating the fluid dynamics
algorithm into the Amber molecular mechanics simulation engine to assess the feasibility
and quality of the new multi-scale model for potential applications to biomolecular simulations.
At the current stage, we are mainly interested in equilibrium properties of the biomolecular
solute, and solvent hydrodynamics is not our consideration. Thus certain alterations of the
original model can be utilized to artificially accelerate the solvent relaxation process so that the
precious computing resources can be focused on sampling of solute conformations.

6.3 Theoretical Model

In the following we first review our physical model for easy understanding of the overall
approach. Next we briefly go over the fluid dynamics algorithm and procedure with a focus on
what has been revised from our previous study to adapt the method to atomistic molecular simulations. Finally computational details are presented for the numerical tests of specific molecular systems.

6.3.1 Physical model

Our basic model is derived from the Hamiltonian equation. A Hamiltonian for the entire system is thus defined first. Its degrees of freedom are atomic positions (x) and their velocities (v) for the solute molecular dynamics (MD) region; and fluid element displacements (y) and their velocities (u) for the solvent fluid dynamics (FD) region. For the MD region, all-atom molecular mechanics will be used. Molecular mechanics usually adopts a relatively simple potential energy function, or force field, for efficient computation. Many potential energy functions have been developed for biomolecular applications, such as Amber\textsuperscript{66-71}, CHARMM\textsuperscript{72-74} and OPLS\textsuperscript{75-77}. For FD region, an incompressible viscous fluid model is adopted.

The Hamiltonian is defined as

\[
H = H_{MD}(x, \alpha_x) + H_{FD}(y, \alpha_y) + H_{MD/FD}(x, \alpha_x, y, \alpha_y),
\]

where \(\alpha_x\) is the momentum of MD region and \(\alpha_y\) is the momentum of FD region. \(H_{MD}\) is the Hamiltonian of for the MD region modeled by molecular mechanics \(H_{MD} = U + K\), where \(U\) is the force field potential energy and \(K\) is the kinetic energy. \(H_{FD}\) represents the Hamiltonian for the incompressible solvent fluid. \(H_{MD/FD} = U_{ele} + U_{vdw} + U_{hse}\), consists of three terms. \(U_{ele}\) is the Poisson-Boltzmann electrostatic solvation energy\textsuperscript{78-81}. The nonelectrostatic solvation energy is modeled as two components: the van der Waals component \(U_{vdw}\) and the hard sphere entropy/cavity component \(U_{hse}\textsuperscript{82-86}\). Here \(U_{ele}\) is defined as

\[
U_{ele} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left( q_i q_j \right) \frac{1}{r_{ij}} e^{-\frac{r_{ij}}{\sigma_{ij}}}
\]
\[ U_{\text{ele}} = \int \left( \rho f \varphi - \frac{1}{8\pi} \mathbf{D} \cdot \mathbf{E} - \Delta \Pi \lambda \right) d\nu \]
\[ \Delta \Pi = kT \sum_i c_i \left( e^{-q_i \varphi / kT} - 1 \right) \]

and \( U_{\text{vdw}} \) and \( U_{\text{hse}} \) are defined as

\[ U_{\text{vdw}} = \sum_{a=1}^{N_s} \int \rho_{aw}(r_{aw}) u_{LJ}(r_{aw}) d\nu \]
\[ U_{\text{hse}} = \gamma \cdot \text{SAS} + c \]

Here the sum is over all solute atoms \((N_s)\), and the integration is over the solvent-occupied volume. \( \rho_{aw}(r_{aw}) \) is a solvent distribution function around solute ‘a’ at a given solute-solvent distance. \( u_{LJ}(r) = \frac{A}{r^{12}} - \frac{B}{r^6} \) is the force field Lennard Jones potential given the coefficient \(A, B\) for each atom. \( \gamma \) is the surface tension and \( c \) is an offset constant.

Now we proceed to derive the dynamics equation by first setting \( \beta = (x, y) \) as the position vector of the system and \( \alpha = (\alpha_x, \alpha_y) \) as the momentum vector of the system. The familiar Newtonian dynamics can be derived from the Hamilton’s equation

\[ \ddot{\alpha} = -\frac{\partial H}{\partial \beta} \]

Here we have adopted the convention that \( \alpha \) and \( \beta \) represent the moment and position vectors of each particle/element, respectively.

In the molecule dynamics region, the equation of motion for an atom at position vector \( x \) can be expressed symbolically as

\[ \dot{\alpha}_x = -\frac{\partial H_{\text{MD}}}{\partial x} - \frac{\partial H_{\text{MD/FD}}}{\partial x} \]
\[ -\frac{\partial H_{MD}}{\partial x} \] represents the usual force field terms in molecule dynamics simulations. The coupling Hamiltonian has three terms, \( U_{ele} + U_{vdw} + U_{hs} \). Since \( U_{hs} \) does not depend on atomic positions, the coupling force terms that the atoms feel are only those of electrostatics and van del Waals in nature, i.e.,

\[ -\frac{\partial H_{MD/FD}}{\partial y} = -\frac{\partial U_{ele}}{\partial x} - \frac{\partial U_{vdw}}{\partial x}. \]

It is interesting to note that the electrostatic forces \( -\frac{\partial U_{ele}}{\partial x} \) are simply the \( qE \) forces, where \( q's \) are the “free” charges, i.e. atomic point charges in a force field model \(^{87}\). \( -\frac{\partial U_{vdw}}{\partial x} \) are the van del Waals forces from the solvent molecules modeled as continuum \(^{84}\).

In the fluid dynamics region, consider a small fluid volume element at position \( y \), with volume \( V \) and velocity \( u \). The equation of motion of the fluid element is

\[ \dot{\alpha}_y = -\frac{\partial H_{FD}}{\partial y} - \frac{\partial H_{MD/FD}}{\partial y}. \] (6)

As shown below, the variational principle will be applied on this element. The partial derivative can also be written in the variational form as

\[ \dot{\alpha}_y = \frac{\delta H_{FD}}{\delta y} \bigg|_u - \frac{\delta H_{MD/FD}}{\delta y} \bigg|_u. \] (7)

Here the subscript \( u \) denotes that it is fixed during the variation. Notice here \( y = y(b,t) \) is the Lagrangian coordinate of the volume element, which is fixed on the fluid element and \( b \) is introduced here to denote the actual spatial position \(^{88}\).

Let us first focus on the variation of \( H_{FD} \), which has the form of

\[ H_{FD} = \int \left[ \frac{1}{2} \rho u^2 + U_{int}(\rho,s) \right] dV, \] (8)
where \( U_{\text{int}} \) is the internal energy density, \( \rho \) represents the fluid density, and \( s \) is entropy. For a small fluid volume element at position \( y \) and volume \( V \), we impose a variation \( \delta y \) on the element within time \( dt \), and proceed to compute the variation of \( H_{FD} \). The process is assumed to be very rapid, i.e., \( \frac{\delta y}{dt} \gg 0 \). Since the fluid is incompressible, \( \nabla \cdot \delta y = 0 \), the work done to the environment is

\[
dW = \int p \delta y \cdot dS = \int \nabla \cdot (p \delta y) dV = \int \nabla p \cdot \delta y dV = \int \nabla p \cdot \delta y dV,
\]

where \( p \) is pressure and vector \( dS \) denotes the surface element of the element with the direction along the normal direction of the surface. Given that the first and second laws of thermodynamics still hold, the internal energy variation can be expressed as

\[
\int \delta U_{\text{int}} dV = \int (T \delta s) dV - dW = \int (T \delta s) dV - \int \nabla p \cdot \delta y dV.
\]

The entropy constraint also gives

\[
\int (T \delta s) dV = \int f_{vis} \cdot \delta y dV - dt \int \nabla \cdot q dV,
\]

where \( q \) is the heat flux and \( f_{vis} \) is the viscous force density. Substitution of eqn (11) into eqn (10) and the fact that the term involving \( dt \) can be ignored as \( \frac{\delta y}{dt} \gg 0 \) given the variation of the internal energy as

\[
\int \delta U_{\text{int}} dV = \int (f_{vis} \cdot \delta y) dV - \int \nabla p \cdot \delta y dV = \int (f \cdot \delta y) dV,
\]

where the total force density \( f = f_{vis} - \nabla p = \frac{\partial \sigma_{ij}}{\partial y_j} \) is introduced and

\[
\sigma_{ij} = -p \delta y_i + \mu \left( \frac{\partial (u \cdot y_i)}{\partial y_j} + \frac{\partial (u \cdot y_j)}{\partial y_i} \right)
\]

is the stress tensor and \( \mu \) is the fluid viscosity constant. 

89
of the fluid. Given the assumption that the force density is uniform within the volume element, substitution of eqn (8) and eqn (12) into the variation of $H_{FD}$ gives

$$- \frac{\delta H_{FD}}{\delta y} \bigg|_u = - \int \delta U_{int} \frac{dV}{\delta y} = - FV = - \frac{\partial \sigma_{ij}}{\partial y_j} V. \tag{13}$$

The variation of $H_{MD/FD}$ is presented next. In Poisson-Boltzmann systems with mobile ions, there is an ionic force term at the Stern layer, but it is usually much smaller than other force terms, and is often ignored. If it were not ignored, the ionic force would act upon relevant volume elements. $U_{hse}$ only depends on the interface boundary so that it does not change under the variation of the volume element. Thus the only significant derivative of $H_{MD/FD}$ is the van der Waals force, which can be treated as the “external force” density ($F$) on the fluid element, i.e.

$$- \frac{\delta H_{MD/FD}}{\delta y} \bigg|_u = - \frac{\delta U_{vdw}}{\delta y} = FV. \tag{14}$$

Finally, the change of momentum of the fluid volume element is

$$\alpha_y = \frac{d(\rho V \mathbf{u})}{dt} = \rho V \frac{\partial \mathbf{u}}{\partial t} + \sum_i \rho V \frac{\partial \mathbf{u}}{\partial a_i} \frac{\partial a_i}{\partial t} = \rho V \frac{\partial \mathbf{u}}{\partial t} + \rho V (\mathbf{u} \cdot \nabla) \mathbf{u}. \tag{15}$$

Combination of eqns (6) and (13) –(15) gives

$$\rho V \frac{\partial \mathbf{u}}{\partial t} + \rho V (\mathbf{u} \cdot \nabla) \mathbf{u} = \frac{\partial \sigma_{ij}}{\partial y_j} V + FV$$

$$\Rightarrow \rho \frac{\partial \mathbf{u}}{\partial t} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} = \frac{\partial \sigma_{ij}}{\partial y_j} + F. \tag{16}$$

Including the conservation of volume/mass for the given volume element, i.e. $\nabla \cdot \mathbf{u} = 0$, the incompressible Navier-Stokes equation can be expressed as

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = - \nabla p + \mu \nabla^2 \mathbf{u} + F.$$

$$\nabla \cdot \mathbf{u} = 0$$
6.3.2 Derivation of interface conditions

To obtain the interface conditions, an infinitely small fluid disk element $\varepsilon$ is introduced with small area $A$ and thickness $h$, and with $h \ll \sqrt{A}$. The disk surfaces are parallel to the boundary interface and one side of the surface is in the molecule dynamics region. Given a variation of the disk position with $\delta \mathbf{r}_\varepsilon$,

$$
\dot{\alpha}_\varepsilon = - \frac{\delta H_{FD}}{\delta \mathbf{r}_\varepsilon} - \frac{\delta H_{FD,MD}}{\delta \mathbf{r}_\varepsilon}. \tag{18}
$$

On the left-hand side

$$
\dot{\alpha}_\varepsilon = \rho Ah \frac{d\mathbf{u}}{dt}. \tag{19}
$$

On the right-hand side, we first introduce the local coordinate system, which consists of one normal direction ($\mathbf{n}$) and two tangential directions ($\mathbf{t}, \mathbf{\tau}$) at a certain point on the interface, i.e.,

$$
\begin{align*}
\mathbf{n} &= \cos \alpha_1 \mathbf{i} + \cos \alpha_2 \mathbf{j} + \cos \alpha_3 \mathbf{k} \\
\mathbf{t} &= \cos \beta_1 \mathbf{i} + \cos \beta_2 \mathbf{j} + \cos \beta_3 \mathbf{k} \\
\mathbf{\tau} &= \cos \gamma_1 \mathbf{i} + \cos \gamma_2 \mathbf{j} + \cos \gamma_3 \mathbf{k}
\end{align*} \tag{20}
$$

Stress $-\frac{\delta H_{FD}}{\delta \mathbf{r}_\varepsilon}$ only exerts on the disk surface in the fluid region, so that

$$
-\frac{\delta H_{FD}}{\delta \mathbf{r}_\varepsilon} = \sigma_{ij} \cdot \mathbf{n} A = \left[ (-p + 2\mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \cdot \mathbf{n}) n + \mu \left( \frac{\partial (\mathbf{u} \cdot \mathbf{n})}{\partial \mathbf{t}} + \frac{\partial (\mathbf{u} \cdot \mathbf{t})}{\partial \mathbf{n}} \right) t + \mu \left( \frac{\partial (\mathbf{u} \cdot \mathbf{n})}{\partial \mathbf{\tau}} + \frac{\partial (\mathbf{u} \cdot \mathbf{\tau})}{\partial \mathbf{n}} \right) \mathbf{\tau} \right] A. \tag{21}
$$

The last term of eqn (18) can be worked out as

$$
-\frac{\delta H_{MD,FD}}{\delta \mathbf{r}_\varepsilon} = -\frac{\delta U_{el}}{\delta \mathbf{r}_\varepsilon} - \frac{\delta U_{bce}}{\delta \mathbf{r}_\varepsilon} - \frac{\delta U_{vdw}}{\delta \mathbf{r}_\varepsilon}, \tag{22}
$$
where \( \frac{-\partial U_{ele}}{\partial r_e} = f_{dielec}A = \frac{1}{2}\sigma_{pol} D_i \cdot D_e nA \) is the dielectric boundary electrostatic force \(^{87}\). The term \( -\frac{\partial U_{hs}}{\partial r_e} = -\gamma \kappa nA \) is the pressure and surface tension from the hard sphere entropy, aka the hydrophobic term \(^{84}\), where \( \kappa \) is the curvature. The van der Waals force, \( \frac{\delta U_{vdw}}{\delta r_e} \) \(^{84}\), proportional to the volume of element \( Ah \), can be ignored when comparing to the electrostatic forces and surface tension as \( h \) is infinitely small. Combining eqns (18) and (19) and the terms calculated above, the interface conditions can be summarized as

\[
-p + p_g - \gamma \kappa + f_{dielec} + 2\mu \frac{\partial u}{\partial n} \cdot n = 0
\]

\[
\frac{\partial (u \cdot n)}{\partial t} + \frac{\partial (u \cdot t)}{\partial n} = 0 \quad \text{on } \partial \Omega \quad (23)
\]

\[
\frac{\partial (u \cdot n)}{\partial \tau} + \frac{\partial (u \cdot \tau)}{\partial n} = 0,
\]

\[6.4 \quad \text{Numerical Algorithms}\]

We explored to implement the multi-scale model in numerical simulations with a strategy similar to those of the classical Car-Parrinello molecular dynamics (CPMD) model \(^{90}\), which can be regarded as a multi-scale model via coupling equations of motion for ions and electrons in two different mechanics. In CPMD electrons are treated as active degree of freedom, via fictitious dynamics variable, and the fictitious electron dynamics is coupled with ionic dynamics in the Berendsen heat bath to approach the Born-Oppenheimer surface. The CPMD model results in a conservative ionic dynamics that is extremely close to the Born-Oppenheimer surface.

Our approach is to couple equations of motion for solute atoms and continuum solvent. The solvent part is also treated by the fictitious dynamics variable, and since our model is based
on a finite-difference method, it is the fluid element. The fictitious fluid dynamics is modeled by
the incompressible Navier-Stokes (NS) equation. The fictitious fluid dynamics model is coupled
with all-atom molecular dynamics model in the Berendsen heat bath to approach the surface
provided by all-atom MD simulations at a preset temperature. In doing so, the changes to the
existing molecular mechanics simulation engine can be kept at the minimal and there is a very
clear boundary between the FD and MD simulation routines, facilitating the development of the
new model into a viable simulation engine for future biomolecular applications.

6.4.1 FD time integration

Our previous work has addressed the mathematical issues in solving fluid dynamics equations
numerically \(^{61, 62}\). After setting the water density to unity, the velocity can be solved by the
second-order semi-implicit backward Euler method as

$$\frac{3u^{k+1} - 4u^k + u^{k-1}}{2\Delta t} + (u \cdot \nabla u)^{k+1} = -\nabla p^{k+1} + \mu \Delta u^{k+1} + F^{k+1}, \text{outside}$$

$$\frac{3u^{k+1} - 4u^k + u^{k-1}}{2\Delta t} = \mu \Delta u^{k+1}, \text{inside}$$

where

$$p^{k+1} = 2p^k - p^{k-1}$$

$$(u \cdot \nabla u)^{k+1} = 2(u \cdot \nabla u)^k - (u \cdot \nabla u)^{k-1}.$$  \hspace{1cm} (25)

The pressure is solved by:

$$\Delta p^{k+1} = -\nabla \cdot ((u^{k+1} \cdot \nabla)u^{k+1}) + \nabla \cdot F^{k+1}.$$  \hspace{1cm} (26)

A new issue facing the application of the FD model to molecular simulation is the
presence of van der Waals force \((F)\), which has a large gradient nearby the interface because it
is too close to the solute atom centers. The large gradient is almost always challenging to address
with a finite-difference type of method. In this study, we overcome the issue by introducing a
variable \( p' \), where \( p' = p + \Gamma \) with \( \nabla \Gamma = -\mathbf{F} \) obtained analytically. Therefore, we can solve \( \nabla p' = \mu \Delta \mathbf{u} \) without computing the numerical gradient of the van der Waals potential. Specifically given \( p''(k+1) = p''(k) + \Gamma(k) \), the equivalent form of eqn (26) to be solved numerically as

\[
\Delta p''(k+1) = -\nabla \cdot (u''(k+1) \cdot \nabla u''(k+1))
\]  

(27)

Accordingly, eqn (24) is updated as

\[
\frac{3u''(k+1) - 4u''(k) + u''(k-1)}{2\Delta t} + (u \cdot \nabla)u''(k+1) = -\nabla p''(k+1) + \mu \Delta u''(k+1), \text{outside}
\]

\[
\frac{3u''(k+1) - 4u''(k) + u''(k-1)}{2\Delta t} = \mu \Delta u''(k+1), \text{inside}
\]

(28)

where \( p''(k+1) \) is taken as

\[
p''(k+1) = 2p''(k) - p''(k-1)
\]  

(29)

Finally the interface boundary condition eqn (23) becomes

\[
-p' + \Gamma + p_s - \gamma \kappa + f_{\text{dielec}} + 2\mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \cdot \mathbf{n} = 0
\]

\[
\frac{\partial (u \cdot \mathbf{n})}{\partial t} + \frac{\partial (u \cdot \mathbf{t})}{\partial \mathbf{n}} = 0
\]

\[
\frac{\partial (u \cdot \mathbf{n})}{\partial \tau} + \frac{\partial (u \cdot \mathbf{\tau})}{\partial \mathbf{n}} = 0,
\]  

(30)

At each time step, the \( p' \) is interpolated with the one-side least square fitting method.\(^1\) \( \Gamma \) is computed analytically for each interface point where the interface boundary condition eqn (30) is enforced. When doing so, we can completely avoid finite-difference operations involving van der Waals energy and forces.

As presented in our previous works, the remaining major mathematical challenge in solving these coupled partial differential equations is the presence of the free boundary condition eqn (23) that allows the solute-solvent interface to equilibrate according to our physical model.
To enforce the free boundary condition when solving pressure or velocity, we utilized the jump conditions of \( u_n \) and \( p_n \) as the augmented variables, respectively \(^{61,62,92}\). The considerations of augmented variables lead to extra correction terms on the right-hand side in eqns (24) and (26). After the correction, each velocity component solver is equivalent to a Helmholtz equation. Once the velocity is updated, the pressure solver is simplified to a Poisson equation. In this implementation, we utilized the MICCCG numerical solver to solve these linear differential equations \(^{93-96}\).

When solving the linear systems, the fluid domain is contained in a rectangular box, the conditions at the outer boundary of the rectangular \( R \) are

\[
\begin{align*}
    u|_{x=x_{\text{min}}} &= 0, & u|_{x=x_{\text{max}}} &= 0 \\
    \frac{\partial u}{\partial y}|_{y=y_{\text{min}}} &= 0, & \frac{\partial u}{\partial y}|_{y=y_{\text{max}}} &= 0 \\
    u|_{z=z_{\text{min}}} &= 0, & u|_{z=z_{\text{max}}} &= 0 \\
    p &= 0,
\end{align*}
\]

which represents a pipe flow in the \( y \) direction. The use of the boundary condition allows the mass conservation law to be preserved since the incompressible solvent fluid can go in and out of the simulation box freely.

### 6.4.2 FD/MD interface update

Once the fluid velocity field is known, the next step is to use it to update the solute/solvent interface. The equivalent step in the solute region is to update particle positions based on particle velocities. Numerically we use the level set method based on the finite-difference method \(^{97-99}\). In the level set method, a scalar function, i.e. the level set function, is used to represent the moving interface implicitly. The interface is located where the level set function is zero \((d=0)\), i.e. the zero level set \( \Gamma(t) = \{ y : d(y,t) = 0 \} \). Suppose that \( \Gamma(t) \) moves according to velocity \( \mathbf{v} \):
\[ \frac{\partial \Gamma(t)}{\partial t} = \mathbf{v}(\Gamma(t)), \]
where \( \mathbf{v} \) is known after the fluid dynamics equations are solved. Given the interface velocity, if we want the level set function \( (d) \) to satisfy \( \Gamma(t) = \{ y : d(y, t) = 0 \} \) after updating, we can impose the following equation upon \( d(y, t) \):

\[
\frac{\partial d}{\partial t} + \mathbf{v} \cdot \nabla d = 0
\]  

(32)

with the initial condition \( \Gamma(0) = \{ y : d(y, 0) = 0 \} \), i.e. the level set function initially set for the initial configuration in our case. Here the level set function was initially set as a signed distance function to the solvent accessible surface with a specified solvent probe.

6.4.3 Overview of the FD/MD numerical procedure

In our system, the atomic details for the solute region are preserved, and the solvent region is modeled as in 6.4.1. To simulate the solute particle dynamics, a standard MD engine with the leapfrog time integrator \(^{100}\) coupled to a heat bath is used. The temperature coupling is realized with the Berendsen thermostat, which has been widely used in molecular simulation community \(^{101}\). Once the heat bath is specified, the procedure of the FD/MD can be summarized into the following steps:

1. Input and initialize system parameters for solute atoms such as temperature, number of particles, time step, etc. Initialize initial positions and velocities of all solute atoms;
2. Initialize FD simulation box and grid points. Initialize velocity and pressure of fluid elements;
3. Compute energy and forces from the potential function of solute atoms;
4. Compute van del Waals forces and pressure between solute atoms and fluid atoms;
5. Use the particle MD engine to update new velocities and positions of solute atoms;
6. Use the FD engine to update new velocities and pressures of fluid elements;
7. Update new FD/MD interface;
8. Repeat steps 3-7.

Dynamics variables, such as position, velocity, pressure, and level set function, are periodically stored after step 7 as requested. These can be used as input to restart the FD/MD simulation as needed.

### 6.5 Other Computational Details

For the FD simulations, physical parameters of water are set as those at 300K with viscosity $\mu = 8.51 \times 10^{-4} \text{Pa} \cdot \text{s}$, density $\rho = 1.00 \times 10^3 \text{kg/m}^3$, and hydrophobic surface tension $\gamma = 8.94 \times 10^{-2} \text{ kcal/mol} \cdot \text{A}^2$, with the later optimized for biomolecules given the SAS molecular surface definition in a previous work for the Amber force fields. The water probe was set as 1.0 Å to set up the initial SAS surface. In the FD simulation programs, both water viscosity and density are often set as 1.0 in the internal unit. Thus proper interface between FD and MD simulation portions of our model require careful unit conversion. The details in deriving these conversion factors are given in Appendix, and the actual conversion factors are listed in Table 6-1.

<table>
<thead>
<tr>
<th>Variable</th>
<th>MD Unit</th>
<th>FD Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time ($t$)</td>
<td>1 ps</td>
<td>85.1</td>
</tr>
<tr>
<td>Density ($\rho$)</td>
<td>1 kg</td>
<td>$1.00 \times 10^{27}$</td>
</tr>
</tbody>
</table>
Table 6-1: Conversion factors between FD and MD engines.

The FD/MD multi-scale simulation engine was developed in a revised Amber 16 release 63-65. The Amber ff14 force field is used to generate the topology files and the TIP3P water model is used to model the water molecules. All atomic charges were set to be zero to focus on the nonelectrostatic interactions in this study. The simulations were conducted with bonds involving hydrogen constrained. Time step was set to be 0.002 ps for both fluid dynamics region and molecule dynamics region. The temperature coupling constant is 0.2 ps in the Berendsen’s thermostat to couple the temperature of the MD region, which is set to be 5 K to study the relaxation of the solute-solvent interface in this study.

Since the goal of the current development is to evaluate how well the MD/FD method reproduce the solvent interface, the MD region are restrained to focus on the FD simulation. Given that the external forces on the FD region are only van der Waals force and hydrophobic force, and they should be balanced each other at equilibrium. To speed up the relaxation, we explored both artificially increase the external force terms (by a factor of 10) or decrease the viscosity terms (by a factor of 10) to accelerate the relaxation towards equilibrium. It was found that the low viscosity runs did not relax as fast as the high force runs (data not shown). Nevertheless all alternatives will be further explored in a future study.

A single ion (Na+) , a single molecule n-methyl amine (NMA), and four typical small molecular complexes, adenine-thymine (AT), guanine-cytosine (GC), arginine-aspartic acid (RD) and lysine-aspartic acid (KD) were chosen to analyze the solute-solvent surface produced by the FD/MD method. In this stage of our development, the electrostatic interactions were turned off so only van der Waals and hydrophobic interactions of the solute molecules were considered.
though water molecules were not alternated. As a benchmark to evaluate the quality of the new multi-scale model, we conducted all-atom molecular mechanics simulations for the four tested dimer complexes to sample the solvent interface with explicit TIP3P water molecules. In these simulations, all molecules first underwent a 10,000-step energy minimization starting with a 5,000-step steepest descent followed by a 5,000-step conjugate gradient minimization. Then all solute atoms were restrained with a harmonic force constant of 50 kcal/mol-Å in all subsequent heating, equilibration, and production simulations. The molecular dynamics simulations were first heat up from 0K to 300K in 20 ps. This was then followed with a 10 ns simulation at the constant temperature of 300K and the constant pressure of 1 bar with the Berendsen heat and pressure baths. The water molecules sampled in the last 5 ns was used to analyze the solute-solvent surfaces.

6.6 Results and Discussion

6.6.1 Single atom relaxation: Reproduction of analytical solution

We first validated the FD/MD engine with a simple system with analytical solution: the solute-solvent interface of a single atom, given that the balance of hydrophobic force and van der Waals force would lead to a final equilibrium surface, a sphere with radius of $r_0$. The equilibrium can be analytically solved once the solvation free energy of the system is given as

$$G = \rho \int_{r_0}^{\infty} \left( \frac{A}{r^{12}} - \frac{B}{r^6} \right) 4\pi r^2 \, dr + \gamma 4\pi r^2$$

Starting from a given initial state, it is expected that the system converge to its free energy minimum if there is no energy barrier, which is the case here.
In this test, an Amber sodium ion solvated in TIP3P water was used as an illustration. With the specified surface tension and van der Waals parameters from Amber 14 force field, the gradient of the free energy can be expressed as

$$\frac{\partial G}{\partial r} = 4\pi \rho \left( -\frac{A}{r^{10}} + \frac{B}{r^6} + 2 \gamma r \right)$$

(34)

Given the values of $A = 4127 \text{kcal/mol} \cdot \text{Å}^{12}$, $B = 3.570 \text{kcal/mol} \cdot \text{Å}^6$, and $\gamma = 8.94 \times 10^{-2} \text{kcal/mol} \cdot \text{Å}^2$, the numerical solution shows that there is only one root for $\frac{\partial G}{\partial r} = 0$ when $r$ is positive, which gives the radius of the sphere to be 2.45 Å. It is also clear that $\frac{\partial G}{\partial r} < 0$ when $r$ approaches 0$^+$ and $\frac{\partial G}{\partial r} > 0$ when $r$ approaches infinity. Given that 1) the gradient changes from negative to positive as $r$ changes from 0$^+$ to $+\infty$, and 2) there is only one root for the gradient, it can be concluded that the gradient is negative when $r < 2.45$ Å, and positive when $r > 2.45$ Å. Thus free energy $G$ is monotonically decreasing when $r < 2.45$ Å, and monotonically increasing when $r > 2.45$ Å. This analysis shows that there is no energy barrier in the physically allowed range of $r$.

Therefore it is possible to use a simple steepest descent minimization or a low-temperature MD relaxation to reach the global minimum in the solvation free energy. Figure 6-1 plots the evolution of volume versus time for the tested low-temperature relaxation run. It is apparent that the volume of the solute-solvent interface quickly converges to a constant volume, consistent with our analysis above. The numerical volume agrees with the analytical solution with an error of $\sim 0.3\%$. Note also that the equilibrium volume is a spherical sphere for the single ion as expected.
**Figure 6-1:** (1) Time evolution of volume (Å³) in the restrained FD/MD simulation of sodium ion. (2) Spherical contour of solute-solvent interface when reaching the equilibrium.

### 6.6.2 Monomer relaxation: Symmetric interface

Next, we performed the low-temperature relaxation of NMA, a mirror-symmetrical monomer. As shown in Figure 6-2, the volume reaches the equilibrium value within 500 steps (1.0 ps). The contour plot shows the symmetrical monomer possesses a symmetrical interface at equilibrium. VMD visualization in 3D indicates that a detailed surface contour similar to that of the solvent excluded surface can be found (see supplementary materials).
Figure 6-2  (1) Time evolution of volume (Å³) in the restrained FD/MD simulation of NMA. (2) SES-like solute-solvent interface is observed when reaching the equilibrium.

6.6.3 Dimer relaxation

Four typical small molecular complexes, adenine-thymine (AT), guanine-cytosine (GC), arginine-aspartic acid (RD) and lysine-aspartic acid (KD) were tested to evaluate the performance of the FD/MD simulation method. As shown in Figure 6-3, the solute volumes reach the equilibrium values within 500 steps (1.0 ps) for all four dimers. Figure 6-4 presents the time evolutions of force balancing on the solute-solvent interface. It is clear that the numerical
solvent pressure and viscosity pressure decrease significantly and approach zero as time goes on. On the other hand, the hydrophobic (surface tension) pressure and the analytical van der Waals pressure become the dominant components, reaching steady values while balancing each other out. This is another evidence that the system approach equilibrium. Apparently the balance between hydrophobic and van der Waals components is not perfect, due to the presence of residual fluid flow nearby the solute. This issue will be addressed in our future refinement of the numerical algorithm to be discussed below.
Figure 6-3 Time evolutions of volume ($\text{Å}^3$) in the restrained FD/MD simulations of dimers: adenine-thymine (AT), guanine-cytosine (GC), arginine-aspartic acid (RD) and lysine-aspartic acid (KD).

![Graphs showing time evolutions of volume for AT, GC, RD, and KD dimers.]

Figure 6-4 Time evolutions of average absolute pressure components on the solute-solvent interface: adenine-thymine (AT), guanine-cytosine (GC), arginine-aspartic acid (RD) and lysine-aspartic acid (KD). (a) Numerical solvent pressure; (b) Hydrophobic pressure; (c) Viscosity pressure; (d) Analytical van der Waals pressure.

6.6.4 Comparison with explicit solvent simulations
Finally, a key issue in the current development of the FD/MD model is to see whether the model at least qualitatively agrees with explicit solvent MD simulations. Discrepancy is possible given that no optimization has been attempted. Therefore, it is interesting to analyze the solute-solvent interfaces as sampled by both the FD/MD model and the explicit solvent MD model.

This analysis was conducted in the following manner. The water molecules in explicit solvent MD simulations were sampled every 5 ps over the course of a 5 ns production run for each tested dimer with all solute atoms restrained in the initial position. A total of 1000 snapshots were collected for visualization. To facilitate visualization, water molecules beyond 3.0 Å distance from any solute atom were discarded. The water distribution maps were used as references to assess the solute-solvent surface sampled by the FD/MD simulation method. Figure 6-5 shows the distribution of water oxygen atoms and the FD/MD surface when viewed outside of the solute-solvent surface, and Figure 6-6 shows the distribution and surface when viewed inside of the solute-solvent surface. Overall, the FD/MD surfaces match very well with the solute-solvent boundaries as sampled in the explicit MD simulations for all four tested complexes. Note too there are a few places of discrepancies, which indicate that the parameters used in the FD/MD model needs to be optimized. VMD visualization in 3D further illustrates the agreement presented here (see supplementary materials).
Figure 6-5 FD/MD surfaces (white wireframe) and water molecules from explicit MD simulations (yellow dots) of four tested dimers: adenine-thymine (AT), guanine-cytosine (GC), arginine-aspartic acid (RD) and lysine-aspartic acid (KD). Here viewer stands outside of the surfaces.
Figure 6-6 FD/MD surfaces (white wireframe) and water molecules from explicit MD simulations (yellow dots) of four tested dimers: adenine-thymine (AT), guanine-cytosine (GC), arginine-aspartic acid (RD) and lysine-aspartic acid (KD). Here viewer stands inside of the surfaces.

6.6.5 Limitations of the model and future directions

There are clearly limitations in the proposed FD/MD model. The first limitation is that we artificially make both hydrophobic and van der Waals term 10 times higher to accelerate the solute-solvent interface relaxation because the focus of the current model is for equilibrium properties of the solute, but not the physically correct solvation relaxation process, which may be important if the FD/MD model is applied to study hydrodynamic properties due to the presence
of the molecular solute. Nevertheless, the artificial setting does not affect the converged solute-solvent interface because both hydrophobic and van der Waals pressure are simultaneously increased. Secondly, the finite-difference grid spacing used in the FD engine is 0.5 Å, which is widely used in biomolecular applications of a finite-difference method given a high enough resolution of molecular surface topology can be achieved. However, the relatively fine grid also leads to highly inefficient numerical procedure. To date we have not paid special attention to the numerical efficiency of our FD engine, and this will be a focus in our future development. The development and illustrations here mainly show that the FD/MD model is sound and it does produce physically meaningful observations consistent with the all-atom MD model, which is very promising.

As we pointed out in Other Computational Details the FD parameters for the water solvation process was from a previous study to optimize a related nonpolar solvent model. Apparently this is not optimal for the current FD/MD model. Our next step will be to investigate how to optimize the hydrophobic term and van der Waals term to best reproduce the all-atom explicit solvent MD model. In addition, we will also incorporate the electrostatic interaction as modeled by the Poisson-Boltzmann method to build a more realistic FD/MD model for biomolecular applications. To study fluid dynamic properties due to the presence of molecular solutes, we think the best strategy is to incorporate a coarse-grained molecular model instead of the all-atom model to make it a viable approach for systems with interesting hydrodynamic properties.
6.7 Appendix

Since FD programs can use any arbitrary length unit, we set 1 internal length unit as 1 Ångstrom. Given the length unit settled and the water density (\( \rho = 1.00 \times 10^3 \text{kg/m}^3 \)) set as 1 internal density unit, the internal mass unit can be computed to be equivalent to \( 1.00 \times 10^{-27} \) kg. Next we can utilize viscosity to compute the time conversion factor. Given the unit of viscosity \( \text{Pa} \cdot \text{s} = \text{kg/(m \cdot s)} \) we can use the mass conversion factor to derive the time conversion factor as follows

\[
1 \text{ internal viscosity unit} = 8.509 \times 10^{-4} \text{kg/(m \cdot s)}
\]
\[
= 8.509 \times 10^{-4} \times 1.00 \times 10^{27} / (10^{10} \times T)
\]

This leads to \( T = 8.51 \times 10^{13} \), which means \( 1 \text{ s} = 8.51 \times 10^{13} \) internal time unit. And thus we have \( 1 \text{ ps} = 10^{-12} \text{ s} = 85.1 \) internal time unit. The energy unit of 1 kcal/mol can be converted as

\[
1 \text{ kcal/mol} = 6.948 \times 10^{-21} \text{ J} = 6.948 \times 10^{-21} \text{ kg m}^2/\text{s}^2
\]
\[
= 6.948 \times 10^{-21} \times 10^{27} \times (10^{10})^2 / (8.509 \times 10^{13})^2 \text{ internal energy unit}
\]
\[
= 9.60 \times 10^{-2} \text{ internal energy unit}
\]
References


CHAPTER 7

Conclusions and Future Directions

To analyze biomolecular solvation, atomistic simulation is an ideal choice. Indeed, atomistic simulation has become an important tool for studying the structures, dynamics, and functions of biomolecular systems. Nevertheless, efficient atomistic simulation of large and complex biomolecular systems is still one of the remaining challenges in computational molecular biology. To appreciate the challenges, it is instructive to highlight the two bottlenecks in biomolecular simulations: (1) the cost of each energy evaluation that is determined by the number of particles in a mathematical model; and (2) the number of dynamics time steps that it takes for sufficient coverage of different configurations. Hundreds of millions of time steps are routinely required in biomolecular simulations to draw statistically significant conclusions. It is often the case that more particles need more time steps for sufficient coverage. Thus, the overall simulation cost usually scales exponentially with the number of particles in the mathematical model used in the simulation. Since most particles in biomolecular simulations are to represent water molecules solvating the target biomolecules, an implicit treatment of water molecules and mobile ions allows greatly increased simulation efficiency. The objective of the dissertation is to develop multi-scale models for biomolecule simulation with implicit treatment of water.

In Chapter 2, a theoretical work on how to compute electrostatic forces in the multi-scale model is conducted. Specifically, a Maxwell stress tensor for molecular systems obeying the full nonlinear Poisson-Boltzmann equation is derived. The formulations of analytical electrostatic forces given the Maxwell stress tensor is further derived and the relations of the formulations with those published in the literature is also discussed. It is shown that the formulations derived
from the Maxwell stress tensor require a weaker condition for its validity, applicable to nonlinear Poisson-Boltzmann systems with a finite number of singularities such as atomic point charges and the existence of discontinuous dielectric as in the widely used classical piece-wise constant dielectric models.

In Chapter 3, a charge-central interpretation of the full nonlinear PBE electrostatic interactions is proposed and a particle-particle particle-mesh (P3M) strategy for the full nonlinear PB systems is developed. The accuracy and convergence of solvation forces with the charge-view and the P3M methods is then studied. It is interesting to note that the convergences of both the charge-view and the P3M methods are more rapid than the original full nonlinear PB method. Given the developments and validations documented here.

In Chapter 4, A Depth-First Search algorithm is introduced on the implicit membrane model in the PBSA framework to detect the pore or channel gated regions automatically. The model is first validated by testing the depth-first search algorithm for pore region detection on three different channel proteins. The results show that for all the channel proteins, the program can automatically detect the pore region inside of the protein and label it as solvent. We then validate that use the average positions of phosphorus to set up our membrane positions and thickness, and use 2.7 Angstrom as our default values for mprob is a reasonable strategy. Given the criterion of setting up the membrane thickness and mprob value, we also found the value of dprob as 1.4 Angstrom is a reasonable default value gives the minimum number of inconsistencies compares to the all-atom MD results. Finally we test the binding energy of eleven different ligand docked with 1K4C, and do the correlation with RT\ln(\text{IC50}), demonstrating that our new program can perform the MMPBSA calculations with pore region detected automatically.
In Chapter 5, a multi-scale framework is proposed for more realistic molecular dynamics simulations in continuum solvent models by coupling a molecular mechanics treatment of solute with a fluid mechanics treatment of solvent. A 3D numerical algorithm is developed to simulate the solvent fluid via the Navier-Stokes equation. The numerical algorithm was validated with multiple dynamical test cases, which also demonstrates that the algorithm is effective and stable, with observed accuracy consistent with the designed numerical algorithm.

In Chapter 6, we incorporated the fluid dynamics integrator with the Amber simulation engine to conduct atomistic simulations of biomolecules. At this stage of the development, only nonelectrostatic interactions, i.e., van del Waals and hydrophobic interactions are included in the multi-scale model. Nevertheless numerical challenges exist in accurately interpolating the highly nonlinear van del Waals term when solving the finite-difference fluid dynamics equations. We were able to bypass the challenge rigorously by merging the van del Waals potential and pressure together when solving the fluid dynamics equations and by considering its contribution in the free-boundary condition analytically. The multi-scale simulation engine was first validated by reproducing the solute-solvent interface of a single atom with analytical solution. Next, we performed the relaxation simulation of a restrained symmetrical monomer and observed a symmetrical solvent interface at equilibrium with detailed surface features resembling those found on the solvent excluded surface. Four typical small molecular complexes were then tested, both volume and force balancing analysis showing that these simple complexes can reach equilibrium within the simulation time window. Finally, we studied the quality of the multi-scale solute-solvent interfaces for the four tested dimer complexes and found they agree well with the boundaries as sampled in the explicit water simulations.
Future efforts can be conducted to model the protein-water interface more self-consistently based on the consistent energy model as defined using the protein-water force field in both explicit and implicit simulations. Specifically, we can investigate how to optimize the hydrophobic term and van der Waals term to best reproduce all-atom MD model, and applied this model to our implicit membrane model as shown in chapter 5. In addition, we can incorporate the electrostatic interaction as modeled by the Poisson-Boltzmann model to build a more realistic FD/MD model for biomolecular applications, and adapt the P3M treatment of the full nonlinear PB model. To study fluid dynamic properties due to the presence of molecular solutes, we think the best strategy is to incorporate a coarse-grained molecular model instead of the all-atom model to make it a viable model for study systems with interesting hydrodynamic properties. Apparently efficiency is also an important issue and another future direction is to develop an more efficient fluid dynamics algorithm and port the new nonlinear PB model to the GPU platforms to further improve the simulation efficiency.