A NEW ADI METHOD FOR THE POISSON-BOLTZMANN EQUATION WITH A TWO-COMPONENT REGULARIZATION

by

SHEIK AHMED ULLAH

SHAN ZHAO, COMMITTEE CHAIR
DAVID HALPERN
BRENDAN AMES
MUHAMMAD A. R. SHARIF
MOJDEH RASOULZADEH

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Mathematics
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2019

ABSTRACT

The Poisson Boltzmann equation (PBE) is a well-established implicit solvent continuum model for the electrostatic analysis of solvated biomolecules. The solution for the nonlinear PBE is still a challenge due to its strong singularity by the source terms, dielectrically distinct regions, and exponential nonlinear terms. In this dissertation, a new alternating direction implicit method (ADI) is proposed for solving the nonlinear PBE using a two-component regularization. This scheme inherits all the advantages of the two-component regularization and the time-dependent PBE with ADI method while possess a novel approach to combine them. A modified version of the one-dimensional ghost fluid method (GFM) has been introduced to incorporate the nonzero jump condition into a new ADI method to propose GFM-ADI method. It produced better results in terms of spatial accuracy and stability compared to the previous ADI method and simpler to implement by circumventing the work necessary to apply the MIB method with the regularization for a 3D problem. Moreover, the stability of the GFM-ADI method has been significantly improved in comparing with the non-regularized ADI method, so that stable protein simulations can be carried out with a pretty large time step size. Two locally one-dimensional (LOD) methods have also been developed for the time-dependent regularized PBE, which are unconditionally stable. Finally, for numerical validation, we have evaluated the solvation free energy for a collection of 24 proteins with various sizes and the salt effect on the protein-protein binding energy of the complex 1beb.

DEDICATION

ACKNOWLEDGMENTS

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Solvated biomolecules and the electrostatic interaction with its solvent is critical to study various important biological process such as protein-drug binding site analysis, DNA recognition, protein folding and protein ligand bonding. In the past few decades with the development of numerical methods and computational techniques analyzing the functions and dynamics of bimolecular solvation is more practical and effective. However, imitating these interactions are still computationally expensive with biological significance. Here we are considering the Poisson-Boltzmann Equation (PBE) to describe the electrostatic potential generated by a low dielectric medium inside a protein molecule with embedded atomic charges solvated in a high dielectric medium with dissolved ions. The analytical solution of PBE is only available for some simple geometry such as a sphere or a cylinder. Efficiency and accuracy becomes a critical issue to solve PBE for biophysical models with complex geometry and a macro-molecule containing tens of thousands to millions of atoms (having partial charges).

In our mathematical model for the electrostatic analysis, PBE is a nonlinear elliptic equation on multiple domains with discontinuous dielectric coefficients separated by the solute-solvent interface or molecular surface. The difficulties with solving PBE arises from non-linearity, discontinuous dielectric coefficients, non-smoothness of the solution and singularities in the source term due to the atomic charges. Effects of non-linearity becomes significant with strong ionic presence [35].

For the non-linearity, two different approaches have been developed in the literature. The usual approach is to discretize the nonlinear PBE into an algebraic system using finite difference or finite element methods and then solve it by a nonlinear algebraic system such as (1) nonlinear relaxation method [17], [27], (2) nonlinear conjugate gradient method [25] or (3) inexact Newton method [16]. The other approach has been introduced recently and its based on the pseudo-transient continuation approach [29, 32, 41]. This approach converts time independent nonlinear PBE into a time dependent form by introducing pseudo-time derivative. The solution to the original boundary problem is then retrieved from the steady state solution of the time dependent PBE. The main advantage of introducing pseudo-time derivative is to be able to split time dependent PBE into linear and nonlinear subsystems to circumvent the blow up and overflow problem due to exponentially large term involving hyperbolic sine function.

To convert this three dimensional(3D) PBE into a set of multiple independent one-dimensional(1D) systems, the alternating direction implicit (ADI) methods in [5–7, 10] the locally one dimensional method (LOD) in [35] have been used in the literature. Specially in [10] Douglas-Rachford ADI scheme has been used to split the linear subsystem with the 3D laplacian operator into three more sub systems with one dimensional 2nd order derivatives. Altogether this method remains 2nd order of accuracy in time but conditionally stable. Later in [35] the LOD method appeared as an unconditionally stable method with reduced accuracy compared to ADI methods in [10]. Even though these 1D subsystems produced by these ADI and LOD methods are tridiagonal and can be efficiently solved by using the Thomas Algorithm [33], it lacks the treatment of the jump conditions at the interface which reduces the accuracy near the interface. Also the numerical error for these pseudo transient methods has been observed often to be dominated by the the singularity at the center of the atoms.

Even though the MIB solvers were pretty efficient to overcome the difficulty due to the jump conditions at the interface, the efficient treatment of charge singularity still remained challenging. It motivated many authors to develop different regularization methods in [1, 2, 9, 12, 14, 36, 45] to reduce the loss of accuracy due to the singularity. In these methods, the potential function has decomposed into a singular component plus one or two other components to break down the PBE into a system of partial differential equations (PDEs) containing a poisson equation with the singular term plus one or two other equations. Thus the singular component can be handled separately using the analytical solution for the poisson equation as Coulomb potentials and or Green's functions. So far these type of regularization methods have never been used with the pseudo transient methods like ADI or LOD.

In an attempt to maintain the efficiency and stability of the ADI methods while restoring the accuracy to the second order near the interface, several interface schemes has been developed in [20–23]. Then as a continuation of this approach recently matched ADI (mADI) method was developed in [44] and [20] to combine the MIB method (for interface treatment) with ADI. But it was mainly focused on the parabolic equations and only 2D condition was considered for the numerical validation.

In this paper our goal was to develop a new approach to solve the PBE combining the regularization, the pseudo-transient continuation and the interface treatment so that both nonlinearity and singularity are treated. For the regularization we have chosen the two component regularization used in [12]. But it changes the jump condition to be non zero which introduces the necessity of interface treatments. Otherwise the standard central difference becomes divergent. Then motivated by the mADI method we have introduced a modified version of the Ghost Fluid Method (GFM) in [4] as the interface treatment. Compared to mADI, GFM is simpler to apply in a pseudo-continuation approach. Altogether GFM-ADI method improved the accuracy and efficiency of the ADI method to solve the non-liner PB equation. Generally it more robust than ADI method but still fails to converge in some special cases. Then to continue the search of a more stable method for our regularized pseudo continuation approach we replaced ADI scheme by LOD method to propose GFM-LODCN and GFM-LODIE method. These other two methods are combining LOD with Crank-Nicolson (CN) and Implicit Euler (IE) to discretize the pseudo time derivative. All of the three methods produced more accurate results efficiently than their predecessors. GFM-LODIE has found to be most robust while GFM-ADI to be most accurate.

CHAPTER 2

PROTEIN DATA FILE PREPARATION

To generate the molecular surface as the interface in our computational domain we had to use the 3D structural data for proteins and nucleic acids available in the Protein Data Bank(PDB). PDB contains freely accessible data on the internet submitted by biologists and biochemists from around the world. These data are usually acquired by X-ray crystallography, NMR spectroscopy, or, cryo-electron microscopy. Each molecule is represented by 4 letter. Among several types of data file available from the PDB we focused on *.pdb* type files available from the website of RCSB [26], one of the member organization of the PDB.

Record Type	Columns	Data	Justification	Data Type
ATOM	1-4	"ATOM"		character
	7-11	Atom serial number	right	integer
	13-16	Atom name	left	character
	17	Alternate location indicator		character
	18-20	Residue name	right	character
	22	Chain identifier		character
	23-26	Residue sequence number	right	integer
	27	Code for insertions of residues		character
	31-38	X orthogonal Å coordinate	right	real (8.3)
	39-46	Y orthogonal Å coordinate	right	real (8.3)
	47-54	Z orthogonal Å coordinate	right	real (8.3)
	55-60	Occupancy	right	real (6.2)
	61-66	Temperature factor	right	real (6.2)
	73-76	Segment identifier	left	character
	77-78	Element symbol	right	character

Table 2.1: Protein Data file (*.pdb* type) format

In a *.pdb* file for a particular protein, ATOM record type (the rows starting with the word "ATOM") contains the information about each atom as shown in Table 2.1.

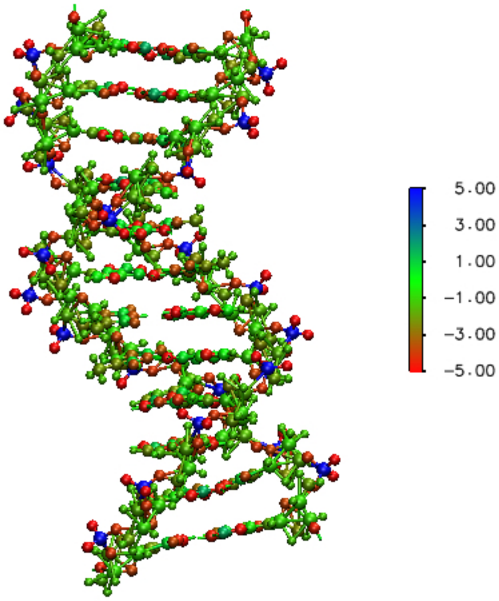


Figure 2.1: Structure of a DNA (*1bna*) constructed from its *.pqr* file showing atomic charges in color for each atoms.

These raw information are not readily usable for our proposed solver. A Python script has been developed to follow the following steps,

- Reading *.pdb* file RCSB website: Python module *"requests"* has been used to read and parse protein data from the web from the web address 'https://files.rcsb.org/view/' to write a local *.pdb* file.

- Converting local *.pdb* file to *.pqr* file: The PDB2PQR software [18] has been used to calculate the *.pqr* file. These *.pqr* files contains ATOM record type similar to *.pdb* files where the occupancy columns (55-60) and Temperature factor columns (61-66) have been replaced by the charge and radius.

Finally in *.pqr* files we have the X, Y and Z co-ordinate, the charge of the atomic center and the radius for each individual atom.

CHAPTER 3

POISSON-BOLTZMANN EQUATION AND ADI-MSMS

3.1 Poisson-Boltzmann Equation

We are considering the Poisson-Boltzmann Equation (PBE) as the governing equation for a solute macro molecule immersed in an aqueous solvent environment illustrated in Figure 3.1. Our computational domain $\Omega \in \mathbb{R}^3$ is separated into two regions Ω^- and Ω^+ by the molecular surface Γ, which is an arbitrarily shaped dielectric interface. Here Ω^- is the molecular region with dielectric constant ϵ^- and Ω^+ is the solvent region with dielectric constant ϵ^+. The cubic shape boundary of $\Omega = \Omega^- \cup \Omega^+$ is denoted by $\partial\Omega$. The charges at the center of each atom. inside Γ, have been distributed as the partial charges to assign to the nearest grid points. The charges outside Γ are actually mobile ions which are described by the Boltzmann distribution.

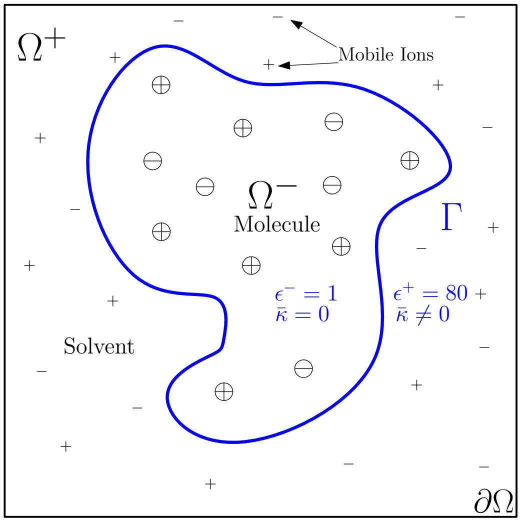


Figure 3.1: Poisson-Boltzmann Model

The electrostatic interaction of this solute-solvent system for $\mathbf{r} \in \mathbb{R}^3$ is governed by the nonlinear Poisson-Boltzmann Equation (PBE) as,

$$-\nabla.(\epsilon(\mathbf{r})\nabla\phi(\mathbf{r})) + \bar{\kappa}^2(\mathbf{r})\sinh(\phi(\mathbf{r})) = \rho(\mathbf{r}) \tag{3.1}$$

with the boundary condition,

$$\phi_b(\mathbf{r}) = \frac{e_c^2}{k_B T}\sum_{i=1}^{N_c}\frac{q_i e^{-|\mathbf{r}-\mathbf{r}_i|\sqrt{\frac{\bar{\kappa}^2}{\epsilon^+}}}}{\epsilon^+|\mathbf{r}-\mathbf{r}_i|} \tag{3.2}$$

where the singular source $\rho(\mathbf{r})$ term is defined as,

$$\rho(\mathbf{r}) = 4\pi\frac{e_c^2}{k_B T}\sum_{i=1}^{N_c} q_i\delta(\mathbf{r}-\mathbf{r}_i) \tag{3.3}$$

There are two conditions on Γ needed to be satisfied from the dielectric theory for the potential ϕ and flux density $\epsilon\phi_{\mathbf{n}}$,

$$[\phi]_\Gamma = 0 \text{ and } [\epsilon\phi_n]_\Gamma = 0 \tag{3.4}$$

Here $\mathbf{n} = (n_x, n_y, n_z)$ is the outer normal direction on the interface Γ and $\phi_{\mathbf{n}} = \frac{\partial\phi}{\partial\mathbf{n}}$ is the directional derivative in $\mathbf{n}$. The notation $[f]_\Gamma = f^+ - f^-$ represent the difference of the functional value across the interface Γ.The dielectric constant ϵ is piecewise such that, $\epsilon(\mathbf{r}) = \epsilon^-$ for $\mathbf{r} \in \Omega^-$ and $\epsilon(\mathbf{r}) = \epsilon^+$ for $\mathbf{r} \in \Omega^+$. Here N_c is the total number of atoms in the solute molecule, k_B is the Boltzmann constant, e_c is the fundamental charge and q_i, in the same unit as e_c is the partial charge on the ith atom of the solute molecule located at position $\mathbf{r}_j$. The Debey-Huckel parameter $\bar{\kappa}^2 = \left(\frac{2N_A e_c^2}{100k_b T}\right)I_s = 8.486902807\text{Å}^{-2}I_s$ from [15] for $\mathbf{r} \in \Omega^-$ and $\bar{\kappa} = 0$ for $\mathbf{r} \in \Omega^+$. Here N_A is Avogadro's Number and I_s is the molar ionic strength. We have converted the dimensionless electrostatic potential u to the units kcal/mol/e_c at the room temperature T by multiplying it by 0.592183 as in [15].

3.2 ADI Method for pseudo transient PBE

Using pseudo transient PBE introducing a pseudo-transient variation has become popular approach for solving the nonlinear PB equation [29, 32, 42, 43]. In this approach equation (3.1) is converted to the following time dependent PBE (TPBE),

$$\frac{\partial \phi}{\partial t}(\mathbf{r},t) = \nabla.(\epsilon(\mathbf{r})\nabla\phi(\mathbf{r},t)) - \bar{\kappa}^2(\mathbf{r})\sinh(\phi(\mathbf{r},t)) + \rho(\mathbf{r}) \tag{3.5}$$

In [11] equation (3.5) has later been separated in two separate equation using a first order time splitting like the following equations,

$$\begin{aligned} \frac{\partial w}{\partial t} &= -\bar{\kappa}^2 \sinh(w) \text{ with } W^n = \phi^n \text{ and } t \in [t_n, t_{n+1}] & (3.6) \\ \frac{\partial v}{\partial t} &= \nabla.(\epsilon\nabla v) \text{ with } V^n = W^{n+1} \text{ and } t \in [t_n, t_{n+1}] & (3.7) \end{aligned}$$

Where $V^{n+1} = \phi^{n+1}$. After using an analytical solution for equation (3.6) a Douglas-Rachford type ADI scheme has been introduced in [11]. The discretization of equation (3.7) using backward-Euler integration in time and central differencing in space results in,

$$v_{i,j,k}^{n+1} = v_{i,j,k}^{n} + \Delta t\left(\delta_x^2 + \delta_y^2 + \delta_z^2\right) v_{i,j,k}^{n+1} + \Delta t \rho(\mathbf{r}) \tag{3.8}$$

Where δ_x, δ_y and δ_z are the central finite difference operators in x, y and z directions as,

$$\begin{aligned} \delta_x^2\left(v_{i,j,k}^n\right) &= \frac{1}{h^2}\left(\epsilon_{i+\frac{1}{2},j,k}(v_{i+1,j,k}^n - v_{i,j,k}^n) + \epsilon_{i-\frac{1}{2},j,k}(v_{i-1,j,k}^n - v_{i,j,k}^n)\right) \\ \delta_y^2\left(v_{i,j,k}^n\right) &= \frac{1}{h^2}\left(\epsilon_{i,j+\frac{1}{2},k}(v_{i,j+1,k}^n - v_{i,j,k}^n) + \epsilon_{i,j-\frac{1}{2},k}(v_{i,j-1,k}^n - v_{i,j,k}^n)\right) \\ \delta_z^2\left(v_{i,j,k}^n\right) &= \frac{1}{h^2}\left(\epsilon_{i,j,k+\frac{1}{2}}(v_{i,j,k+1}^n - v_{i,j,k}^n) + \epsilon_{i,j,k-\frac{1}{2}}(v_{i,j,k-1}^n - v_{i,j,k}^n)\right) \end{aligned} \tag{3.9}$$

Here the ϵ-halfs ($\epsilon_{i+\frac{1}{2},j,k}, \epsilon_{i,j+\frac{1}{2},k}$ and $\epsilon_{i,j,k+\frac{1}{2}}$) are the average of the dielectric constants value at two adjacent grid point in x, y and z directions. Then the following Douglas-Rachford ADI scheme has been used in [11]

$$\begin{aligned}
\left(1-\Delta t\delta_x^2\right) v_{i,j,k}^{*} &= \left[1+\Delta t\left(\delta_y^2+\delta_z^2\right)\right] v_{i,j,k}^{n} + \Delta t\rho(\mathbf{r}) \\
\left(1-\Delta t\delta_y^2\right) v_{i,j,k}^{**} &= v_{i,j,k}^{*} - \Delta t\delta_y^2\left(v_{i,j,k}^{n}\right) \\
\left(1-\Delta t\delta_z^2\right) v_{i,j,k}^{n+1} &= v_{i,j,k}^{**} - \Delta t\delta_z^2\left(v_{i,j,k}^{n}\right)
\end{aligned} \tag{3.10}$$

The ADI methods discussed in [11] are promising tool to solve the nonlinear PBE but it is not focused on the issues due to the singularity at the source terms and the jump conditions on the interface. As we observed in the chapter 6, these two issues reduced the accuracy and stability of the ADI method significantly.

3.3 Cleaning ADI-MSMS package to develop REG-GFM-MSMS package

To simulate the ADI Methods in [11] and the LOD Methods in [35], Professor Shan Zhao, Dr. Weihua Geng form the Southern Methodists University and Leighton Wilson developed a coding package named ADI-MSMS. This package uses two types of protein data files for a particular protein as inputs and calculated the electrostatic potential ϕ and the solvation energy as the outputs. The input files were generated from the *.pdb* and *.pqr* file. This package used the following two sub-packages developed by others,

- **FISHPACK90:** This sub-package was originally developed by the NCAR (National Center for Atmospheric Research) back in the late 70's and revised later in 90's. It is a collection of FORTRAN programs and subroutines that solves second- and fourth-order finite difference approximations to the elliptic Partial Differential Equations (PDEs). In ADI-MSMS package its been used to the solve the Poisson Equation for a special case of the PBE when the protein molecule is considered to be in vacuum instead of surrounded by the solvent.

- **MSMS:** It is a program [28] written in the programming language C to compute the molecular surface. It was developed by Michel Sanner and his team at the Molecular Graphics Laboratory (TSRI). It uses the input files for the ADI-MSMS package and produces two files with extensions *.vert* and *.face*. Later these output files from the MSMS package are used to decide whether a grid point is inside or outside or on the molecular interface.

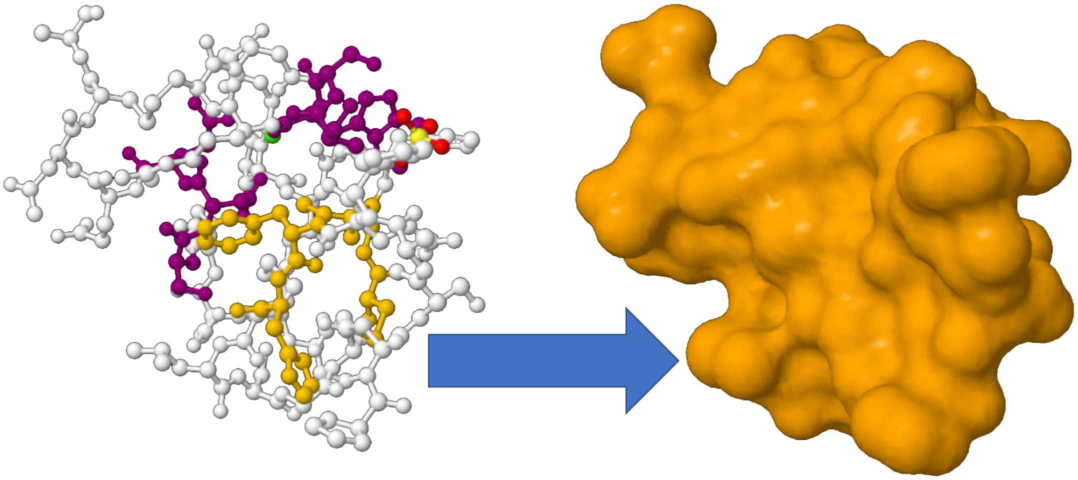


Figure 3.2: Molecular Surface of the protein *1ajj* generated by MSMS

Though this ADI-MSMS package served well for the previous works for the ADI and LOD methods, it was time to do a major revision. When we started with this ADI-MSMS package we had to fix the following issues to prepare it to replace the Numerical methods by our proposed methods to solve the PBE,

- FISHPACK90 got updated a long time ago and it was failing to comply with the new FORTRAN compilers. Also the subsidiary FFT(Fast Fourier Transform) routines were passing arguments of one type and using them as another. So we made the following adjustments.

    1. In the subroutine *POISOLVE3D* the variables *BDXS, BDXF, BDYS, BDYF, BDZS, BDZF* were declared as real variables but used later as two dimensional

arrays. We fixed this issues by declaring them in proper dimension and size.

2. The variable $IFAC$ was declared as integer but it was meant to be a real variable.

3. In the module *fish* the arrays *rew* and *cxw* were declared as pointers but this type of declarations became obsolete for the new FORTRAN compilers. We updated them to be declared as allocatable.

- Using two different type of files as inputs for the ADI-MSMS package was requiring an extra manual step outside the package to prepare the protein data to be usable. We updated the subroutine *readin* to read the necessary data directly from the *.pqr* file. It was a more user friendly process and better in terms of data management for the whole package.

- Unlike the ADI or LOD methods we needed the exact location of the points where the molecular surface was intersecting with the gridlines. We had to add extra features in the package to extract that information from the output files produced by the MSMS package.

After cleaning the bugs form ADI-MSMS and adding the new features we developed a new coding package REG-GFM-MSMS to incorporate all three of our proposed numerical methods to solve the PBE.

CHAPTER 4

PSEUDO-TIME SOLUTION OF REGULARIZED PBE

4.1 Two-component regularization for singular sources

To avoid the difficulty due to the source term and the work for solving the Laplace equation we consider a two component regularization proposed by Cai, Wang and Zhao [1]. For this regularization the electrostatic potential ϕ has been considered as the addition of the coulmb component ϕ_c and the reaction field component ϕ_{RF} as $\phi = \phi_c + \phi_{RF}$. Here ϕ_c satisfies the following Poission's equation,

$$\begin{cases} -\epsilon^- \Delta\phi_C(r) = \rho(r) \text{ in } \mathbb{R}^3 \\ \phi_C(r) = 0. \text{ as } |r| \to \infty \end{cases} \tag{4.1}$$

which gives us the analytical solution as the Green's function G for ϕ_C as,

$$G(r) = \frac{e_c^2}{k_B T} \sum_{i=1}^{N_c} \frac{q_i}{\epsilon^- |r - r_i|} \tag{4.2}$$

Then Luo and his collaborators [1] proposed to solve for the original solution ϕ in Ω^+ instead of the reaction component ϕ_{RF}. Now to make the required adjustments to fit this regularization approach [1] with finite difference and finite element method Zhao and Geng [12] proposed a new elliptic problem with discontinuous function flux jumps for the two-component regularization. In particular they defined the regularized potential as,

$$\tilde{\phi} \;=\; \begin{cases} \phi_{RF} \text{ in } \Omega^- \\ \\ \phi_C + \phi_{RF} \text{ in } \Omega^+ \end{cases} \tag{4.3}$$

The jump conditions for $\tilde{\phi}$ were derived from ϕ and the definition $\phi = \phi_C + \phi_{RF}$ as,

$$\phi^+ = \phi^-_{RF} + \phi^-_C, \text{ and } \epsilon^+ \frac{\partial \phi^+}{\partial n} = \epsilon^- \frac{\partial \phi^-_{RF}}{\partial n} + \epsilon^- \frac{\partial \phi^-_C}{\partial n} \text{ on } \Gamma \tag{4.4}$$

Thus, the regularized PB equation of $\tilde{\phi}$ with corresponding interface and boundary conditions are given as.

$$-\nabla.(\epsilon^- \nabla \tilde{\phi}) \;=\; 0 \text{ in } \Omega^- \tag{4.5}$$

$$-\nabla.(\epsilon^+ \nabla \tilde{\phi}) + \bar{\kappa}^2 \sinh(\tilde{\phi}) \;=\; 0 \text{ in } \Omega^+ \tag{4.6}$$

$$\left[\tilde{\phi}\right] \;=\; G \text{ on } \Gamma \tag{4.7}$$

$$\left[\epsilon \frac{\partial \tilde{\phi}}{\partial n}\right] \;=\; \epsilon^- \frac{\partial G}{\partial n} \text{ on } \Gamma \tag{4.8}$$

$$\tilde{\phi} \;=\; \phi_b \text{ on } \partial\Omega \tag{4.9}$$

Now from (4.6) and (4.9) we can summarize that,

$$-\nabla.(\epsilon \nabla \tilde{\phi}) + \bar{\kappa}^2 \sinh(\tilde{\phi}) \;=\; 0 \text{ in } \Omega^- \cup \Omega^+ \tag{4.10}$$

where $\tilde{\phi}$ satisfies the PB equation without the source term. Then Zhao and Geng [12] proposed to numerically solve the regularized PB interface problem (RPB) given in (4.6)-(4.9) and to finally recover the original solution as $\phi = \tilde{\phi}$ in Ω^+ and

$\phi = \tilde{\phi} + G$ in Ω^-, Where the Green's function G is analytically given.

4.2 Pseudo-transient continuation approach for the PBE

As the pseudo-transient continuation approach, an indirect method discussed in [29, 31, 42] introduced a pseudo-time derivative to solve the PB equation as,

$$\begin{aligned} \frac{\partial u}{\partial t} &= \nabla.(\epsilon \nabla u) - \bar{\kappa}^2 \sinh(u) \text{ in } \Omega^- \cup \Omega^+ && (4.11) \\ [u] &= G, \text{ and } \left[\epsilon \frac{\partial u}{\partial n}\right] = \epsilon^- \frac{\partial G}{\partial n} \text{ on } \Gamma && (4.12) \\ u &= \phi_b \text{ on } \partial\Omega && (4.13) \end{aligned}$$

Here the time independent regularized PB equation in (4.10) has been converted in to a time dependent regularized PB (TDRPB) in (4.11). Our goal is to first specify the initial condition, which could be the electrostatic potential solved from a linearized PB [42] equation or trivially $u = 0$ and then numerically integrate (4.11) for a sufficiently long period to get the steady state solution as the solution of the original regularized PB (4.10). Here the sign on the right hand side of eq. (4.11) has been considered as the reverse of the eq.(4.10) to ensure the numerical stability.

But, there are difficulties in the numerical integration of the TDRPB equation (4.11) because of the requirement of long time integration, when explicit time stepping methods are usually not efficient [29], [31], [42], [43]. Hence we are employing a semi-implicit time splitting method [29], [31] which have been commonly used to solve the TDPB in the literature.

Let us consider a uniform mesh with a grid spacing h in all x, y and z directions having N_x, N_y and N_z as the number of the grid points in each direction. We assume the vector $U^n = \tilde{\phi}^n_{ijk}$ for $i = 1, ..., N_x, j = 1, ..., N_y, k = 1, ...N_z$ denote all the nodal values of $\tilde{\phi}$ at the time level t_n. We will develop several schemes for updating U^n at time level t_n to U^{n+1} at time level $t_{n+1} = t_n + \Delta t$. In this time dependent RPB we will consider only U as time dependent, while all other functions in (4.11) i.e. ϵ and $\bar{\kappa}^2$ are time independent.

4.3 ADI method with Implicit-Euler method (GFM-ADI)

In this scheme at each time step from t_n to t_{n+1}, the time dependent equation TDRPB in (4.11) will be solved by a first order time splitting in two stages,

$$\begin{aligned} \frac{\partial w}{\partial t} &= -\bar{\kappa}^2 \sinh(w) \text{ with } W^n = U^n \text{ and } t \in [t_n, t_{n+1}] \qquad (4.14) \\ \frac{\partial v}{\partial t} &= \nabla.(\epsilon \nabla v) \text{ with } V^n = W^{n+1} \text{ and } t \in [t_n, t_{n+1}] \qquad (4.15) \end{aligned}$$

Altogether we have $U^{n+1} = V^{n+1}$. And for $\frac{\partial w}{\partial t} = -\bar{\kappa}^2 \sinh(w)$ we have the analytical solution as,

$$W^{n+1} = \ln\left(\frac{\cosh(\frac{1}{2}\bar{\kappa}^2\Delta t) + \exp(-W^n)\sinh(\frac{1}{2}\bar{\kappa}^2\Delta t)}{\exp(-W^n)\cosh(\frac{1}{2}\bar{\kappa}^2\Delta t) + \sinh(\frac{1}{2}\bar{\kappa}^2\Delta t)}\right) \qquad (4.16)$$

Which will help us to avoid to the difficulty due to the non-linear term with sinh(.) in (3.1). Here the right hand side of equation (4.16) is just a function of W and Δt which allows us to rewrite the equation (4.16) as $W^{n+1} = F(W^n, \Delta t)$ to facilitate the following discussion. For the temporal discretization of the equation (4.15) we have used Backward-Euler integration in time to get,

$$v_{i,j,k}^{n+1} = v_{i,j,k}^{n} + \Delta t \left(\delta_x^2 + \delta_y^2 + \delta_z^2\right) v_{i,j,k}^{n+1} \qquad (4.17)$$

Where δ_x^2, δ_y^2 and δ_z^2 are the central finite difference operators defined in (4.18) for the x, y and z directions, respectively with $\epsilon_{i,j,k} = \begin{cases} \epsilon^- \text{ if } x_{i,j,k} \in \Omega^- \cup \Gamma \\ \epsilon^+ \text{ if } x_{i,j,k} \in \Omega^+ \end{cases}$.

$$
\begin{aligned}
\delta_x^2\left(v_{i,j,k}^n\right) &= \frac{\epsilon_{i,j,k}}{h^2}\left(v_{i-1,j,k}^n - 2v_{i,j,k}^n + v_{i+1,j,k}^n\right) \\
\delta_y^2\left(v_{i,j,k}^n\right) &= \frac{\epsilon_{i,j,k}}{h^2}\left(v_{i,j-1,k}^n - 2v_{i,j,k}^n + v_{i,j+1,k}^n\right) \\
\delta_z^2\left(v_{i,j,k}^n\right) &= \frac{\epsilon_{i,j,k}}{h^2}\left(v_{i,j,k-1}^n - 2v_{i,j,k}^n + v_{i,j,k+1}^n\right)
\end{aligned} \tag{4.18}
$$

But at the points near the interface the central difference operators defined in (4.18)-(4.18) will not be applicable. These points are defined as the irregular points where at-least one of its adjacent points is on the other side of the boundary. On the irregular points one of the three point stencils for δ_x^2, δ_y^2 and δ_z^2 will be on the other side of the interface where the information about the function v is not available. A non-standard finite difference formula is necessary on the irregular points to discretize δ_x^2, δ_y^2 and δ_z^2. In this regard a modified version of the Ghost Fluid Method has been proposed in Section **??** using the fictitious points (or ghost points) and the jump conditions in (4.13) rigorously.

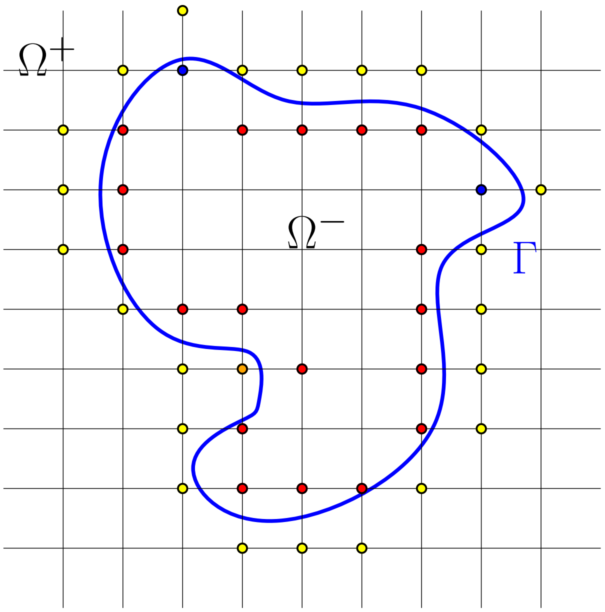


Figure 4.1: Irregular grid points marked as colored disks with blue and orange color for corner points.

Then a first order Douglas-Rachford type ADI scheme has been used to decompose the diffusion equation (4.17) in x, y and z directions as,

$$\begin{aligned} \left(1 - \Delta t \delta_x^2\right) v_{i,j,k}^{*} &= \left[1 + \Delta t \left(\delta_y^2 + \delta_z^2\right)\right] v_{i,j,k}^{n} && (4.19) \\ \left(1 - \Delta t \delta_y^2\right) v_{i,j,k}^{**} &= v_{i,j,k}^{*} - \Delta t \delta_y^2 \left(v_{i,j,k}^{n}\right) && (4.20) \\ \left(1 - \Delta t \delta_z^2\right) v_{i,j,k}^{n+1} &= v_{i,j,k}^{**} - \Delta t \delta_z^2 \left(v_{i,j,k}^{n}\right) && (4.21) \end{aligned}$$

Where v^* and v^{**} are two intermediate values to create three tridiagonal one dimensional system. Here, the three dimensional linear algebraic system in equation (4.17) has been decomposed into several one dimensional linear algebraic systems in (4.19), (4.20) and (4.21). Each one these equations has a tridiagonal structure. These three tridiagonal systems are much more efficient to solve than one non-structured system in (4.17). Then by eliminating $v_{i,j,k}^{*}$ and $v_{i,j,k}^{**}$ and solving for $v_{i,j,k}^{n+1}$ in (4.17) we get,

$$\begin{aligned} v_{i,j,k}^{n+1} = v_{i,j,k}^{n} + \Delta t \left(\delta_x^2 + \delta_y^2 + \delta_z^2\right) v_{i,j,k}^{n+1} - \Delta t^2 \left(\delta_x^2 \delta_y^2 + \delta_x^2 \delta_y^2 + \delta_z^2 \delta_y^2\right) \left(v_{i,j,k}^{n+1} - v_{i,j,k}^{n}\right) \\ + \Delta t^3 \delta_x^2 \delta_y^2 \delta_z^2 \left(v_{i,j,k}^{n+1} - v_{i,j,k}^{n}\right) \end{aligned} \qquad (4.22)$$

Hence the Douglas-Rachford scheme (4.17) is a higher order perturbation of the Implicit-Euler method. Since both (4.14) and (4.15) are first order in time this proposed ADI-IE-GFM scheme is of first order accuracy in time. As the boundary conditions the same Dirichlet boundary boundary values are assumed for v, v^* and v^{**} for u. The entire time integration here is fully implicit.

4.4 LOD with Crank-Nicolson method(GFM-LODCN)

In this scheme at each time step from t_n to t_{n+1}, the time dependent non-linear equation TDPBE will be solved by a second order time splitting methods in three stages

[39],

$$
\begin{aligned}
\frac{\partial w}{\partial t} &= -\frac{1}{2}\bar{\kappa}^2 \sinh(w) \text{ with } W^n = U^n \text{ and } t \in [t_n, t_{n+1}] && (4.23)\\
\frac{\partial v}{\partial t} &= \nabla.(\epsilon \nabla v) \text{ with } V^n = W^{n+1} \text{ and } t \in [t_n, t_{n+1}] && (4.24)\\
\frac{\partial \tilde{w}}{\partial t} &= -\frac{1}{2}\bar{\kappa}^2 \sinh(\tilde{w}) \text{ with } \tilde{W}^n = V^{n+1} \text{ and } t \in [t_n, t_{n+1}] && (4.25)
\end{aligned}
$$

We then have $U^{n+1} = \tilde{W}^{n+1}$. As in the subsection **??** we will use an analytical integration in the first and the last stage of this scheme. Symbolically, we have $W^{n+1} = F(W^n, \frac{\Delta t}{2})$ and $\tilde{W}^{n+1} = F(\tilde{W}^n, \frac{\Delta t}{2})$, where F is defined as in Eq.(4.16).

Then we have proposed another multiplicative operator splitting scheme called as Locally One Dimensional (LOD) scheme to solve the diffusion equation (4.24). This types of fractional step methods were first developed by Russian mathematicians in [3], [37], [38]. The discretization of (4.24) using Crank-Nicolson integration in time and central differencing in space results in,

$$
\left(1 - \frac{\Delta t}{2}(\delta_x^2 + \delta_y^2 + \delta_z^2)\right) v_{i,j,k}^{n+1} = \left(1 + \frac{\Delta t}{2}(\delta_x^2 + \delta_y^2 + \delta_z^2)\right) v_{i,j,k}^{n} \qquad (4.26)
$$

Which can be decomposed in to x, y and z directions to give a LOD scheme for (4.24),

$$
\begin{aligned}
\left(1 - \frac{\Delta t}{2}\delta_x^2\right) v_{i,j,k}^{*} &= \left(1 + \frac{\Delta t}{2}\delta_x^2\right) v_{i,j,k}^{n}\\
\left(1 - \frac{\Delta t}{2}\delta_y^2\right) v_{i,j,k}^{**} &= \left(1 + \frac{\Delta t}{2}\delta_y^2\right) v_{i,j,k}^{*} \qquad (4.27)\\
\left(1 - \frac{\Delta t}{2}\delta_z^2\right) v_{i,j,k}^{n+1} &= \left(1 + \frac{\Delta t}{2}\delta_z^2\right) v_{i,j,k}^{**}
\end{aligned}
$$

Similarly as the GFM-ADI scheme described in subsection **??** the tridiagonal systems in (4.27) can be efficiently solved by the Thomas algorithm.

4.5 LOD with Implicit-Euler method (GFM-LODIE)

For this scheme at each time step from t_n to t_{n+1}, the time dependent equation TDRPB in (4.11) will be solved by the first order time splitting similar to the GFM-ADI scheme described in the subsection **??** in two stages. To follow the same steps in the first stage Eq. (4.14) will be solved analytically by the function $W^{n+1} = F(W^n, \Delta t)$ in Eq. (4.16) and in the second stage for the temporal discretization of the Eq. (4.15) we have used Implicit Euler method to get Eq.(4.17).

Then we have applied the LOD scheme described in the subsection 4.4 to decompose Eq. (4.17) in to x, y and z direction as,

$$\begin{aligned} (1 - \Delta t \delta_x^2) v_{i,j,k}^{*} &= v_{i,j,k}^{n} \\ (1 - \Delta t \delta_y^2) v_{i,j,k}^{**} &= v_{i,j,k}^{*} \\ (1 - \Delta t \delta_z^2) v_{i,j,k}^{n+1} &= v_{i,j,k}^{**} \end{aligned} \tag{4.28}$$

Which will solved individually by the Thomas Algorithm like the previous subsections.

CHAPTER 5

A NEW DERIVATION FOR GHOST FLUID METHOD

Ghost Fluid Method(GFM) is a sharpe interface technique introduced in [4] to treat the two-face contact discontinuities in the Euler equations. It extends values across the interface into an artificial fluid (ghost fluid), inducing the jump conditions at the interface. This GFM method was later extended in [24] to solve elliptic equations with variable coefficients. But in contrast to their method [24] , the jump conditions are incorporated into the numerical discretization in such a way that the symmetry of the finite difference stencil is preserved. Which makes it compatible with most standard solvers. The flux jump has been decomposed in each axis direction treating the problem dimension by dimension. As a result this extended GFM becomes only first order accurate.

5.1 One dimension

In order to overcome the difficulty due to the jump-condition on the interface Γ, a modified version of Ghost Fluid Method(GFM) has been used. For a simple 1D model when we try to evaluate by the finite difference operator δ_{xx} at irregular points where the interface is at x_Γ and between x_i and x_{i+1} as the following figure.

Here u is the unknown function defined as solution of the PBE as $u = \begin{cases} u_{in} \text{ in } \Omega^- \\ u_{out} \text{ in } \Omega^+ \end{cases}$

Then we define $u^+ = u_{out}(x_\Gamma), u^- = u_{in}(x_\Gamma), h = x_{i+1} - x_i$ and $\lambda = \frac{x_\Gamma - x_i}{h}$ to get following relations,

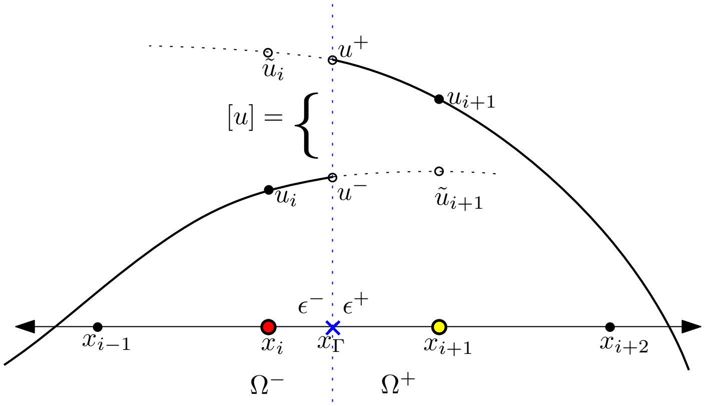


Figure 5.1: 1D GFM

$$
\begin{aligned}
&x_\Gamma = x_i + \lambda h \\
&u^- = u_i(1-\lambda) + \tilde{u}_{i+1}\lambda \text{ and } u^+ = \tilde{u}_i(1-\lambda) + u_{i+1}\lambda \\
&[u] = u^+ - u^- \text{ and } \left[\epsilon \frac{\partial u}{\partial x}\right] = \epsilon^+ u_x^+ - \epsilon^- u_x^-
\end{aligned}
\tag{5.1}
$$

Where $\tilde{u_i}$ and $\tilde{u_{i+1}}$ are the fictitious values(or ghost values) of u at the irregular points extending u_{in} or u_{out} on the other side of the interface. In particular for figure 5.1, we have $\tilde{u}_i = u_{out}(x_i)$ and $\tilde{u}_{i+1} = u_{in}(x_{i+1})$. Here u_{out} has been extended to the point x_i for $\tilde{u_i}$ and u_{in} has been extended to the point x_{i+1} for $\tilde{u_i}$.

Our purpose here is to apply the non-standard finite difference operator δ_x^2 at x_i and x_{i+1} to get,

$$\delta_x^2\left(u_i\right) = \frac{\epsilon^-}{h^2}\left(u_{i-1} - 2u_i + \tilde{u}_{i+1}\right) \text{ and } \delta_x^2\left(u_{i+1}\right) \quad = \frac{\epsilon^+}{h^2}\left(\tilde{u}_i - 2u_{i+1} + u_{i+2}\right) \tag{5.2}$$

Now solving above equations in (5.1) for $\tilde{u}_i$ and $\tilde{u}_{i+1}$ we get,

$$\begin{aligned}
\tilde{u}_i &= \frac{\epsilon^-}{\epsilon^+\lambda+\epsilon^-(1-\lambda)}u_i + \frac{\lambda(\epsilon^+-\epsilon^-)}{\epsilon^+\lambda+\epsilon^-(1-\lambda)}u_{i+1} \\
&+ \frac{\epsilon^-}{\epsilon^+\lambda+\epsilon^-(1-\lambda)}[u] - \frac{h\lambda}{\epsilon^+\lambda+\epsilon^-(1-\lambda)}\left[\epsilon\frac{\partial u}{\partial x}\right] \\
\tilde{u}_{i+1} &= \frac{(\epsilon^--\epsilon^+)(1-\lambda)}{\epsilon^+\lambda+\epsilon^-(1-\lambda)}u_i + \frac{\epsilon^+}{\epsilon^+\lambda+\epsilon^-(1-\lambda)}u_{i+1} \\
&- \frac{\epsilon^+}{\epsilon^+\lambda+\epsilon^-(1-\lambda)}[u] - \frac{h(1-\lambda)}{\epsilon^+\lambda+\epsilon^-(1-\lambda)}\left[\epsilon\frac{\partial u}{\partial x}\right]
\end{aligned} \tag{5.3}$$

Then substituting equation (5.3) into equation (5.2) we get,

$$\begin{aligned}
\delta_x^2\left(u_i\right) &= \frac{1}{h^2}\left(a_1u_{i-1}+b_1u_i+c_1u_{i+1}\right) + \frac{\epsilon^-}{h^2}\left(e_1.[u]+f_1.\left[\epsilon\frac{\partial u}{\partial x}\right]\right) \\
\delta_x^2\left(u_{i+1}\right) &= \frac{1}{h^2}\left(a_2u_i-b_2u_{i+1}+c_2u_{i+2}\right) + \frac{\epsilon^+}{h^2}\left(e_2.[u]+f_2.\left[\epsilon\frac{\partial u}{\partial x}\right]\right)
\end{aligned} \tag{5.4}$$

Where,

$$\begin{aligned}
&d = \epsilon^+\lambda+\epsilon^-(1-\lambda) \\
&a_1 = \epsilon^-, b_1 = -\epsilon^-\left(1+\frac{\epsilon^+}{d}\right), c_1 = \frac{\epsilon^-\epsilon^+}{d}, e_1 = \frac{\epsilon^-}{d}, f_1 = \frac{h\lambda}{d} \\
&a_2 = \frac{\epsilon^+\epsilon^-}{d}, b_2 = -\epsilon^+\left(1+\frac{\epsilon^-}{d}\right), c_2 = \epsilon^+, e_2 = -\frac{\epsilon^+}{d}, f_2 = \frac{h(1-\lambda)}{d}
\end{aligned} \tag{5.5}$$

Here the 2nd terms of the equations in (5.4) are considered to be known. Only the coefficients in the 1st terms of (5.4) contribute to the finite difference operator matrix to make it diagonally dominant(as $|b_1|-a_1-c_1=0$ and $|b_2|-a_2-c_2=0$) and symmetric (as $a_2=c_1$).

5.2 Two dimensions

For two dimensional PB model we need to evaluate δ_{xx} and δ_{yy}, where the similar derivation of equations in (5.4) can be used to calculate δ_{yy} in irregular points. But in this case, $\left[\epsilon\frac{\partial u}{\partial x}\right]$ and $\left[\epsilon\frac{\partial u}{\partial y}\right]$ is not known, while $\left[\epsilon\frac{\partial u}{\partial n}\right]$ is given. Now to get $\left[\epsilon\frac{\partial u}{\partial x}\right]$ and $\left[\epsilon\frac{\partial u}{\partial y}\right]$ in terms of $\left[\epsilon\frac{\partial u}{\partial n}\right]$ and $\left[\epsilon\frac{\partial u}{\partial \tau}\right]$ we have the following relations,

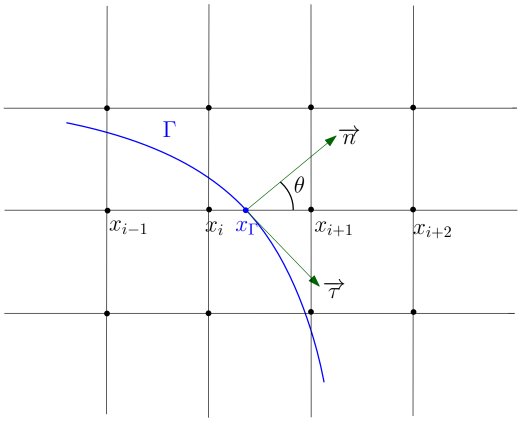


Figure 5.2: 2D GFM

$$\begin{aligned}\left[\epsilon\frac{\partial u}{\partial n}\right] &= \cos\theta\left[\epsilon\frac{\partial u}{\partial x}\right] + sin\theta\left[\epsilon\frac{\partial u}{\partial y}\right]\\ \left[\epsilon\frac{\partial u}{\partial \tau}\right] &= sin\theta\left[\epsilon\frac{\partial u}{\partial x}\right] - \cos\theta\left[\epsilon\frac{\partial u}{\partial y}\right]\end{aligned} \tag{5.6}$$

Which can be solved for $\left[\epsilon\frac{\partial u}{\partial x}\right]$ and $\left[\epsilon\frac{\partial u}{\partial y}\right]$ as,

$$\begin{aligned}\left[\epsilon\frac{\partial u}{\partial x}\right] &= \cos\theta\left[\epsilon\frac{\partial u}{\partial n}\right] + sin\theta\left[\epsilon\frac{\partial u}{\partial \tau}\right]\\ \left[\epsilon\frac{\partial u}{\partial y}\right] &= sin\theta\left[\epsilon\frac{\partial u}{\partial n}\right] - \cos\theta\left[\epsilon\frac{\partial u}{\partial \tau}\right]\end{aligned} \tag{5.7}$$

To simplify the jump-conditions Liu, Fedkiw and Kang [24] smeared out the tangential derivative considering $\left[\epsilon\frac{\partial u}{\partial \tau}\right] = 0$ to get the following equations,

$$\begin{aligned}\left[\epsilon\frac{\partial u}{\partial x}\right] &= \cos\theta\left[\epsilon\frac{\partial u}{\partial n}\right]\\ \left[\epsilon\frac{\partial u}{\partial y}\right] &= \sin\theta\left[\epsilon\frac{\partial u}{\partial n}\right]\end{aligned} \tag{5.8}$$

Which are in general not true but $\left[\epsilon\frac{\partial u}{\partial x}\right]$ and $\left[\epsilon\frac{\partial u}{\partial y}\right]$ in (5.8) and (5.8) satisfies (5.6) for $\left[\epsilon\frac{\partial u}{\partial n}\right]$. It allows us to replace the unknown quantities as $\left[\epsilon\frac{\partial u}{\partial x}\right]$ and $\left[\epsilon\frac{\partial u}{\partial y}\right]$ by a known quantity $\left[\epsilon\frac{\partial u}{\partial n}\right]$ and $\left[\epsilon\frac{\partial u}{\partial \tau}\right]$. The process used to simplify the jump condition in [24] still requires the normal direction θ. In our proposed GFM scheme we have considered $\frac{\partial u^+}{\partial \tau} = 0$ to get the following derivation for $\left[\epsilon\frac{\partial u}{\partial x}\right]$ as,

$$
\begin{aligned}
\left[\epsilon\frac{\partial u}{\partial x}\right] &= \cos\theta\left[\epsilon\frac{\partial u}{\partial n}\right] + \sin\theta\left[\epsilon\frac{\partial u}{\partial \tau}\right] \\
&= \cos\theta\left[\epsilon\frac{\partial u}{\partial n}\right] + \sin\theta\left(\epsilon^+\frac{\partial u^+}{\partial \tau} - \epsilon^-\frac{\partial u^-}{\partial \tau}\right) \\
&= \cos\theta\left[\epsilon\frac{\partial u}{\partial n}\right] + \sin\theta\left(\epsilon^+\frac{\partial u^+}{\partial \tau} - \epsilon^-\frac{\partial u^+}{\partial \tau} + \epsilon^-\frac{\partial u^+}{\partial \tau} - \epsilon^-\frac{\partial u^-}{\partial \tau}\right) \\
&= \cos\theta\left[\epsilon\frac{\partial u}{\partial n}\right] + \sin\theta\left(\epsilon^-\left(\frac{\partial u^+}{\partial \tau} - \frac{\partial u^-}{\partial \tau}\right) + (\epsilon^+ - \epsilon^-)\frac{\partial u^+}{\partial \tau}\right) \\
&= \cos\theta\left[\epsilon\frac{\partial u}{\partial n}\right] + \sin\theta\left(\epsilon^-\left[\frac{\partial u}{\partial \tau}\right] + (\epsilon^+ - \epsilon^-)\frac{\partial u^+}{\partial \tau}\right)
\end{aligned}
$$

Now by the jump conditions we have $\left[\epsilon\frac{\partial u}{\partial n}\right] = \epsilon^-\frac{\partial G}{\partial n}$ and $\left[\frac{\partial u}{\partial \tau}\right] = \frac{\partial G}{\partial \tau}$.

$$
\begin{aligned}
\text{Thus } \left[\epsilon\frac{\partial u}{\partial x}\right] &= \cos\theta(\epsilon^-\frac{\partial G}{\partial n}) + \sin\theta.\epsilon^-\frac{\partial G}{\partial \tau} + \sin\theta(\epsilon^+ - \epsilon^-)\frac{\partial u^+}{\partial \tau} \\
&\approx \epsilon^-(\cos\theta\frac{\partial G}{\partial n} + \sin\theta\frac{\partial G}{\partial \tau}) = \epsilon^-\frac{\partial G}{\partial x}, \text{ consedering } \frac{\partial u^+}{\partial \tau} = 0 \\
\therefore \left[\epsilon\frac{\partial u}{\partial x}\right] &\approx \epsilon^-\frac{\partial G}{\partial x}
\end{aligned}
$$

Similarly it can be shown that $\left[\epsilon\frac{\partial u}{\partial y}\right] \approx \epsilon^-\frac{\partial G}{\partial y}$.

5.3 Three dimensions

Consider that the interface Γ intersects the grid line in the x direction at a point (i_Γ, j, k) which is located between (i, j, k) and $(i+1, j, k)$, We therefore have two irregular grid points, (i, j, k) and $(i+1, j, k)$. The fictitious values $\tilde{u}_{i,j,k}$ and $\tilde{u}_{i+1,j,k}$ are to be determined. To use one of the jump conditions which is defined in the normal direction of the interface point, it is convenient to introduce a local coordinates (ξ, η, ζ) such that ξ is along the normal direction and η is in the xy plane. Then the coordinate transformation can be given as,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{P} \begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} \tag{5.9}$$

where $\mathbf{P}$ is the transformation matrix

$$\mathbf{P} = \begin{bmatrix} \sin\psi\cos\theta & -\sin\theta & -\cos\psi\cos\theta \\ \sin\psi\sin\theta & \cos\theta & -\cos\psi\sin\theta \\ \cos\psi & 0 & \sin\psi \end{bmatrix}. \tag{5.10}$$

Here θ and ψ are the azimuth and zenith angles with respect to the normal direction ξ. Then from equation (5.9) and (5.10) we have,

$$\left[\epsilon\frac{\partial u}{\partial x}\right] = \sin\psi\cos\theta\left[\epsilon\frac{\partial u}{\partial \xi}\right] - \sin\theta\left[\epsilon\frac{\partial u}{\partial \eta}\right] - \cos\psi\cos\theta\left[\epsilon\frac{\partial u}{\partial \zeta}\right] \tag{5.11}$$

Now along η axis,

$$\begin{aligned} \left[\epsilon\frac{\partial u}{\partial \eta}\right] &= \epsilon^+\frac{\partial u^+}{\partial \eta} - \epsilon^-\frac{\partial u^-}{\partial \eta} \\ &= \epsilon^+\frac{\partial u^+}{\partial \eta} - \epsilon^-\frac{\partial u^+}{\partial \eta} + \epsilon^-\frac{\partial u^+}{\partial \eta} - \epsilon^-\frac{\partial u^-}{\partial \eta} \\ &= \epsilon^-\left(\frac{\partial u^+}{\partial \eta} - \frac{\partial u^-}{\partial \eta}\right) + (\epsilon^+ - \epsilon^-)\frac{\partial u^+}{\partial \eta} \\ &= \epsilon^-\frac{\partial G}{\partial \eta} + (\epsilon^+ - \epsilon^-)\frac{\partial u^+}{\partial \eta}. \text{ since } \left(\frac{\partial u^+}{\partial \eta} - \frac{\partial u^-}{\partial \eta}\right) = \frac{\partial}{\partial \eta}[u] = \frac{\partial G}{\partial \eta}. \end{aligned} \tag{5.12}$$

Similarly along ζ axis,

$$\left[\epsilon\frac{\partial u}{\partial \zeta}\right] = \epsilon^{-}\frac{\partial G}{\partial \zeta} + (\epsilon^{+} - \epsilon^{-})\frac{\partial u^{+}}{\partial \zeta} \tag{5.13}$$

Then from equation (5.11), (5.12) and (5.13)we have,

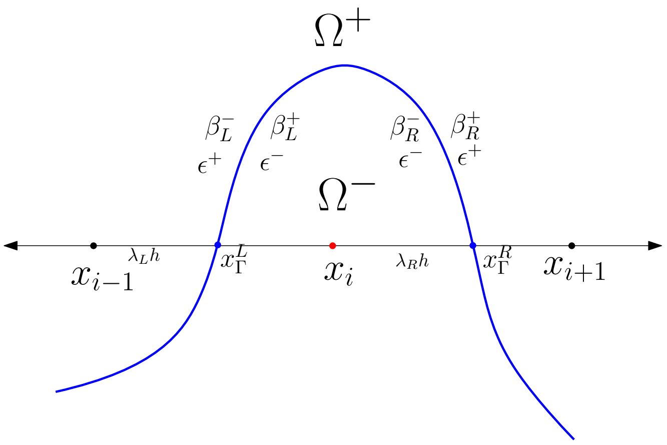
$$\begin{aligned}
\left[\epsilon\frac{\partial u}{\partial x}\right] &= \epsilon^{-}\left(\sin\psi\cos\theta\frac{\partial G}{\partial \xi} - \sin\theta\frac{\partial G}{\partial \eta} - \cos\psi\cos\theta\frac{\partial G}{\partial \zeta}\right) \\
&\quad - \sin\theta(\epsilon^{+} - \epsilon^{-})\frac{\partial u^{+}}{\partial \eta} - \cos\phi\cos\theta(\epsilon^{+} - \epsilon^{-})\frac{\partial u^{+}}{\partial \zeta} \\
&= \epsilon^{-}\frac{\partial G}{\partial x} - \sin\theta(\epsilon^{+} - \epsilon^{-})\frac{\partial u^{+}}{\partial \eta} - \cos\phi\cos\theta(\epsilon^{+} - \epsilon^{-})\frac{\partial u^{+}}{\partial \zeta} \\
&\approx \epsilon^{-}\frac{\partial G}{\partial x} \text{ considering } \frac{\partial u^{+}}{\partial \eta} = 0 \text{ and } \frac{\partial u^{+}}{\partial \zeta} = 0.
\end{aligned} \tag{5.14}$$

Similarly along x axis and y axis we have, Then from equation (5.11), (5.12) and (5.13)we have,

$$\left[\epsilon\frac{\partial u}{\partial y}\right] \approx \epsilon^{-}\frac{\partial G}{\partial y} \text{ and } \left[\epsilon\frac{\partial u}{\partial z}\right] \approx \epsilon^{-}\frac{\partial G}{\partial z} \tag{5.15}$$

5.4 Corner point for the new GFM method

We have a special situation for a corner point when the interface crosses the grid line for the same axis twice around x_i where x_{i-1} and x_{i+1} are on the other side of the interface.



Corner point case for the new GFM with $G_L = -G$ and $G_R = G$. Which gives the following equations for the fictitious points $\tilde{u}_{i-1}$ and $\tilde{u}_{i+1}$.

$$\begin{aligned} \tilde{u}_{i-1} &= A_1 u_{i-1} + B_1 u_i + C_1 G_L - D_1 \epsilon^- \frac{\partial G_L}{\partial x} && (5.16) \\ \tilde{u}_{i+1} &= A_2 u_{i+1} + B_2 u_i - C_2 G_R - D_2 \epsilon^- \frac{\partial G_R}{\partial x} && (5.17) \end{aligned}$$

Where,

$$\beta_L^- = \epsilon^+, \beta_L^+ = \epsilon^-, \beta_R^- = \epsilon^-, \beta_R^+ = \epsilon^+$$

$$\lambda_L = \frac{x_i - x_{i-1}}{h}, \lambda_R = \frac{x_{i+1} - x_i}{h}$$

$$F_1 = \beta_L^+ \lambda_L + \beta_L^- (1 - \lambda_L), F_2 = \beta_R^+ \lambda_R + \beta_R^- (1 - \lambda_R)$$

$$A_1 = \frac{\beta_L^-}{F_1}, B_1 = \frac{\lambda_L(\beta_L^+ - \beta_L^-)}{F_1}, C_1 = \frac{\beta_L^-}{F_1}, D_1 = \frac{h\lambda_L}{F_1}$$

$$A_2 = \frac{\beta_R^+}{F_2}, B_2 = \frac{(1 - \lambda_R)(\beta_R^- - \beta_R^+)}{F_2}, C_2 = \frac{\beta_R^+}{F_2}, D_2 = \frac{h(1 - \lambda_R)}{F_2}$$

5.5 Solvation energy from Electrostatic potential

The solvation energy can be defined as the energy released when the solute in free space is dissolved in solvent and it can be expressed in terms of the electrostatic free energy ΔG_{ele} and the coulomb energy E_{cou} as

$$\begin{aligned} \Delta G_{\text{ele}} &= E_{\text{cou}} + E_{\text{sol}} && (5.18) \\ E_{\text{cou}} &= \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} \frac{q_i q_j}{\epsilon^- d_{i,j}}, i \neq j && (5.19) \end{aligned}$$

where q_i and q_j are the charges at the center of the atoms and $d_{i,j}$'s are the distance between the i-th and j-the atom.

Then Sharp and Honig in [30] described the calculation electrostatic free energy by,

$$\Delta G_{\text{ele}} = \int_{\mathbb{R}^3} \left(\phi\rho + \Delta\Pi - \frac{1}{2}\epsilon|\mathbf{E}|^2 \right) d\mathbf{r} \tag{5.20}$$

where ϕ is the electrostatic potential, ρ is the fixed charge density represented as a smeared surface charge or as a collection of point charges, $\Delta\Pi$ is the excess osmotic pressure of the mobile ion cloud, and $\frac{1}{2}\epsilon|\mathbf{E}|^2$ is the electrostatic stress.

Now for the simplification of the numerical validations of our proposed schemes in the next chapter we omitted the energy components related to the mobile ion pressure and the electrostatic stress to report the solvation energy E_{sol} as,

$$E_{\text{sol}} = \Delta G_{\text{ele}} - E_{\text{cou}} \approx \sum_{i=1}^{N_c} q_i \phi_{RF}(\mathbf{r}_i) \tag{5.21}$$

Here the contribution of the mobile ion pressure and the electrostatic stress to the whole calculation is really small and computationally more challenging. The readers can refer to [8, 13, 30] for more details.

CHAPTER 6

NUMERICAL VALIDATION

To validate our proposed algorithm we are providing numerical simulations in this chapter. At first we have solved the nonlinear PB equation on a krikwood sphere and compared the numerical results with the analytical solution. Then we have considered a hypothetical protein molecule with just one atom for the PB equation and solved it to compare with the analytical result.

6.1 Krikwood Sphere with analytical solution

The analytical solutions of the PB equation are available only for simple geometries such as spheres. So we choose the case where we have a charge at the center of a sphere known as the krikwood sphere with the analytical solution based on the reference [10].

$$\phi(\mathbf{r}) = \begin{cases} \dfrac{1}{\varepsilon R} - \dfrac{1}{R} + \dfrac{1}{||\mathbf{r}||} & ||\mathbf{r}|| < R, \\ \dfrac{1}{\varepsilon ||\mathbf{r}||} & ||\mathbf{r}|| > R. \end{cases} \tag{6.1}$$

$$\rho(\mathbf{r}) = \begin{cases} 4\pi\epsilon^{-}\delta(\mathbf{r}) & ||\mathbf{r}|| < R, \\ \bar{\kappa}^2 \sinh(\dfrac{1}{\varepsilon||\mathbf{r}||}) & ||\mathbf{r}|| > R, \end{cases} \tag{6.2}$$

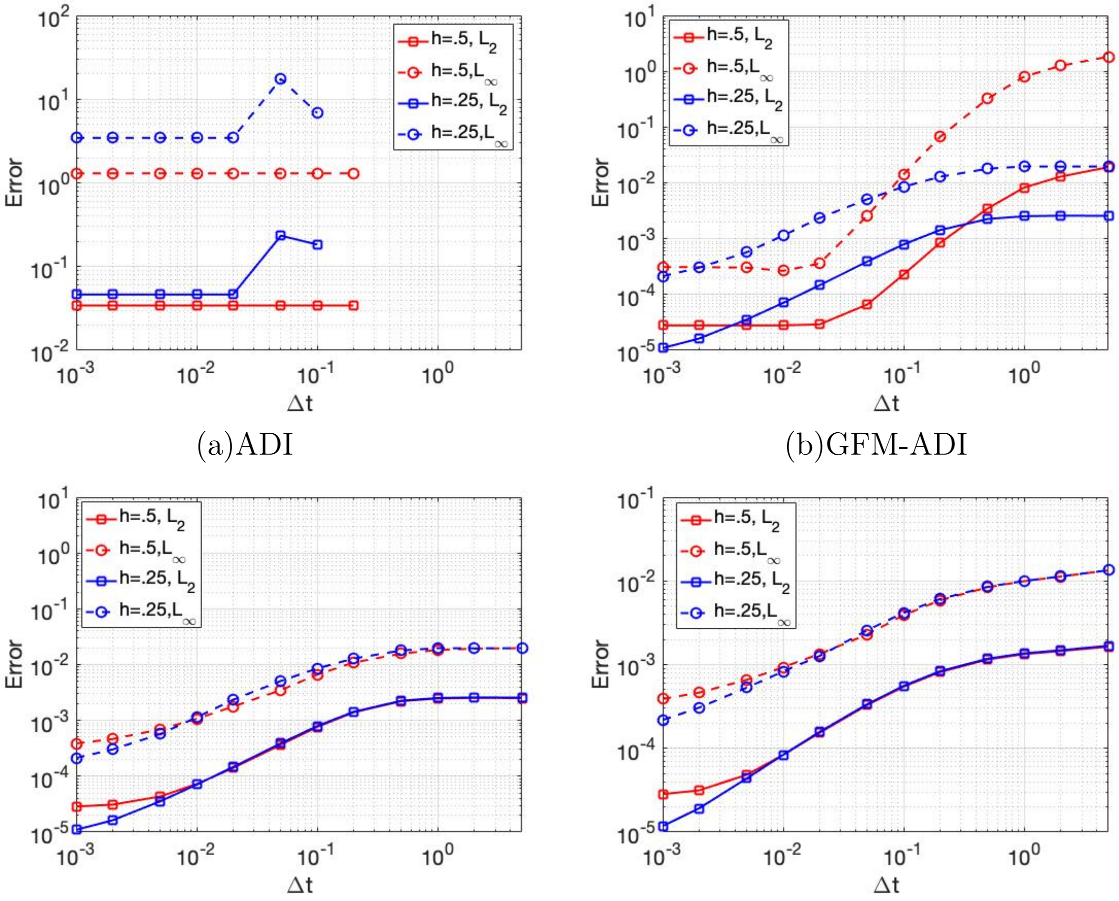
where $\varepsilon = \epsilon^{+}/\epsilon^{-}$ and $R = 2$Å is the radius of the sphere and $\bar{\kappa} = 1$. We have a unit charge $1e_c$ located at the center of the sphere. Our dielectric constants are $\epsilon^{+} = 80$ and $\epsilon^{-} = 1$ It can be shown that this analytical solution in (6.1) with the source term defined in (6.2) will satisfy the non-linear PB equation in (3.1) together with the jump condition defined in (3.4). Both the singularity and the non-smoothness are present in this analytical solution. The singularity is from the source term in (6.2)for $||\mathbf{r}|| > R$ and the non-smoothness comes from the jump condition (3.4). Both of these difficulties gives similar kind of challenge present in the non-linear PB equation to reduce the accuracy of the spatial discretization numerically. To use this benchmark problem to test the stability and the convergence in time and space we computed the L_2 error and L_∞ error using the following measures.

$$L_\infty = \max_{i,j,k} |\phi_{\text{exact}} - \phi_{\text{num}}|, L_2 = \sqrt{\frac{\sum_{i,j,k} |\phi_{\text{exact}} - \phi_{\text{num}}|^2}{N}}$$

Where ϕ_{true} is the analytical solution and ϕ_{num} is the numerical solution representing the electrostatic potential for the non-linear PB equation. For the L_2 error we have used $N = N_x \times N_y \times N_z$ as the total number of unknowns on the grid points.

Stability test: At first for the stability of the non-linear PB equation to calculate the potential $\phi(\mathbf{r})$ for the krikwood sphere we considered the domain to be $[-3, 3]$ for $x, y$and z direction with the spherical radius to be $R = 1$ for the interface. In this study we used finer grid as $h = 0.125$ to avoid the difficulty due to the larger grid spacing and focused on the effect on the stability due to the changes in time increment Δt at each time step. We have found all three of our methods to be stable for $\Delta t = [0.001, 5]$. To illustrate this we considered the sampling for Δt as $\Delta t = \{0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$ and the stopping time T as $T = 100$ so that enough accumulations are experienced. We observed both L_2 and L_∞ error to be finite. For larger values of Δt as $\Delta t = 5$ the numerical errors might feel meaningless but as long as this errors remains to be finite, this demonstrates the stability of the underlying time integration.

GFM-ADI is much more stable than ADI. This makes GFM-ADI a very practical method, while the ADI method is impractical by requiring too small dt values. In some sense, the GFM-ADI method is even better than LOD methods. Although LOD methods



(a)ADI

(b)GFM-ADI

(c)GFM-LODCN

(d)GFM-LODIE

Figure 6.1: Stability test for the Krikwood Sphere with $x, y, z = [-7, 7]$, Radius $= 2$, TEND $= \Delta t * 10^4$, $I_s = 0.01$ and $\bar{\kappa} = 1$

h	L_2	Order	L_∞	Order	E_{sol}
ADI					
2	6.45E-03	N/A	3.82E-02	N/A	-92.699927
1	4.88E-03	0.40	7.61E-02	-0.99	-83.683388
1/2	3.43E-02	-2.81	1.30E+00	-4.09	-85.921222
1/4	4.63E-02	-0.43	3.44E+00	-1.41	-83.279725
1/8	5.11E-02	-0.14	7.49E+00	-1.12	-82.680633
GFM-ADI					
2	3.14E-04	N/A	1.82E-03	N/A	-81.742795
1	1.18E-04	1.41	8.28E-04	1.13	-82.132181
1/2	2.79E-05	2.08	3.10E-04	1.42	-82.063724
1/4	8.51E-06	1.71	1.23E-04	1.34	-82.051117
1/8	1.49E-06	2.51	4.47E-05	1.46	-82.046462
GFM-LODCN					
2	3.14E-04	N/A	1.82E-03	N/A	-81.742788
1	1.18E-04	1.41	8.89E-04	1.03	-82.132148
1/2	2.87E-05	2.04	3.87E-04	1.20	-82.063684
1/4	1.10E-05	1.39	2.13E-04	0.86	-82.051064
1/8	7.16E-06	0.62	1.41E-04	0.60	-82.046402
GFM-LODIE					
2	3.16E-04	N/A	1.84E-03	N/A	-81.738399
1	1.17E-04	1.43	8.85E-04	1.06	-82.123319
1/2	2.84E-05	2.05	3.89E-04	1.19	-82.055388
1/4	1.18E-05	1.27	2.18E-04	0.83	-82.043011
1/8	9.40E-06	0.33	1.48E-04	0.56	-82.046402

Table 6.1: Spatial convergence test for the krikwood sphere with $\epsilon^+ = 80$, $\epsilon^- = 1$, $\Delta t = 0.001$, TEND $= 10$, $I_s = 0.01$ and $\kappa = 1$. The centered charge of unit $1e_c$ is located at (0, 0, 0).

are unconditionally stable, energies are inaccurate for large dt values. For GFM-ADI, the critical dt value for stability tells us a safe zone, within this safe zone, the GFM-ADI is both accurate and efficient.

Spacial Convergence: In this study, we investigated the order of accuracy for the spacial convergence in Table 6.1 for the krikwood sphere. The time increment Δt has been kept fixed to 0.001 while reducing the grid spacing h from 2 to 1/8 in this process. Within this range of h we have noticed the accuracy of GFM-ADI to be nearly 2nd order while gradually reducing for GFM-LODCN and GFM-LODIE. Then we have computed the solvation energy E_{sol} using the non-linear PB equation in (3.1) with the source term defined in (3.3) for the similar setup for the krikwood sphere in 6.1. The solvation energy for this setup can also be computed analytically as -81.97820845. For all three of our proposed methods the solvation energies E_{sol} found to be very close to this analytical value as reported in Table 6.1.

Temporal Convergence:

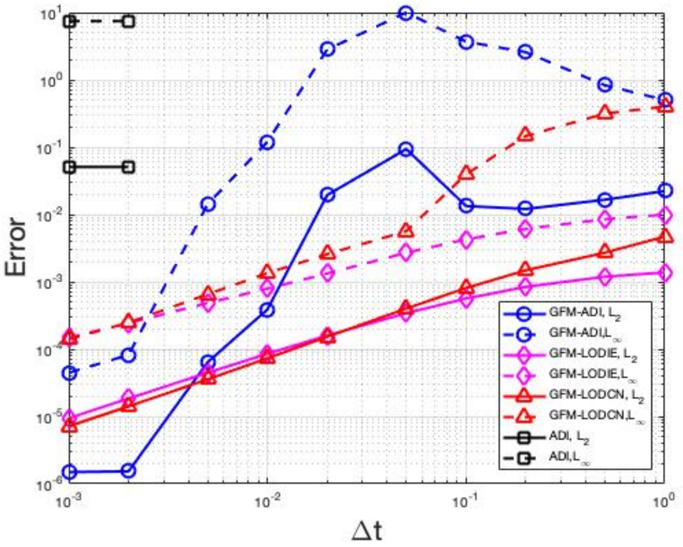


Figure 6.2: Temporal convergence test for the krikwood sphere with $h = 0.125$

6.2 Biological applications

In this section, we focus on exploring the stability and accuracy of GFM-ADI, GFM-LODCN and GFM-LODIE schemes by considering the solvation analysis of proteins. Even though all of our proposed schemes were found to be stable for the one atom case (Krikwood sphere) for $\Delta t = [0.001, 5]$, it is of great interest to see if these schemes are stable for real protein systems. We will compare all three methods in detail for a particular protein system and try to identify the best choice for the time increment Δt and the grid spacing h to get the most accurate result within a reasonable amount of time. The optimum Δt and h has been used later to calculate the solvation energy of 24 proteins.

6.2.1 Protein Crambin

To validate our proposed schemes with a real protein we have studied the hydrophobic protein Crambin (PDB ID : 1cbn). It is a 46 residue protein homologous to a membrane-active plant toxins [34]. It is found in the seeds of *Crambe abyssinicia* and have local anesthetic activity in a lobster walking leg axon (J. Marquis [34]). We used the MSMS package to generate the molecular surface for this protein using the crystallographic data recored at $130K$ as reported in [34]. At this step we used the probe radius as 1.4 the density as 10 for the MSMS package calculate the molecular surface.

For this study at first we reported the solvation energy of 1cbn calculated by all three of our proposed schemes in Table 6.2. After calculating electrostatic potential ϕ, equation (5.21) has been discretized further to calculate the solvation energy as,

$$E_{\text{sol}} = \sum_i \sum_j \sum_k Q(x_i, y_j, z_k)\phi_{RF}(x_i, y_j, z_k) \tag{6.3}$$

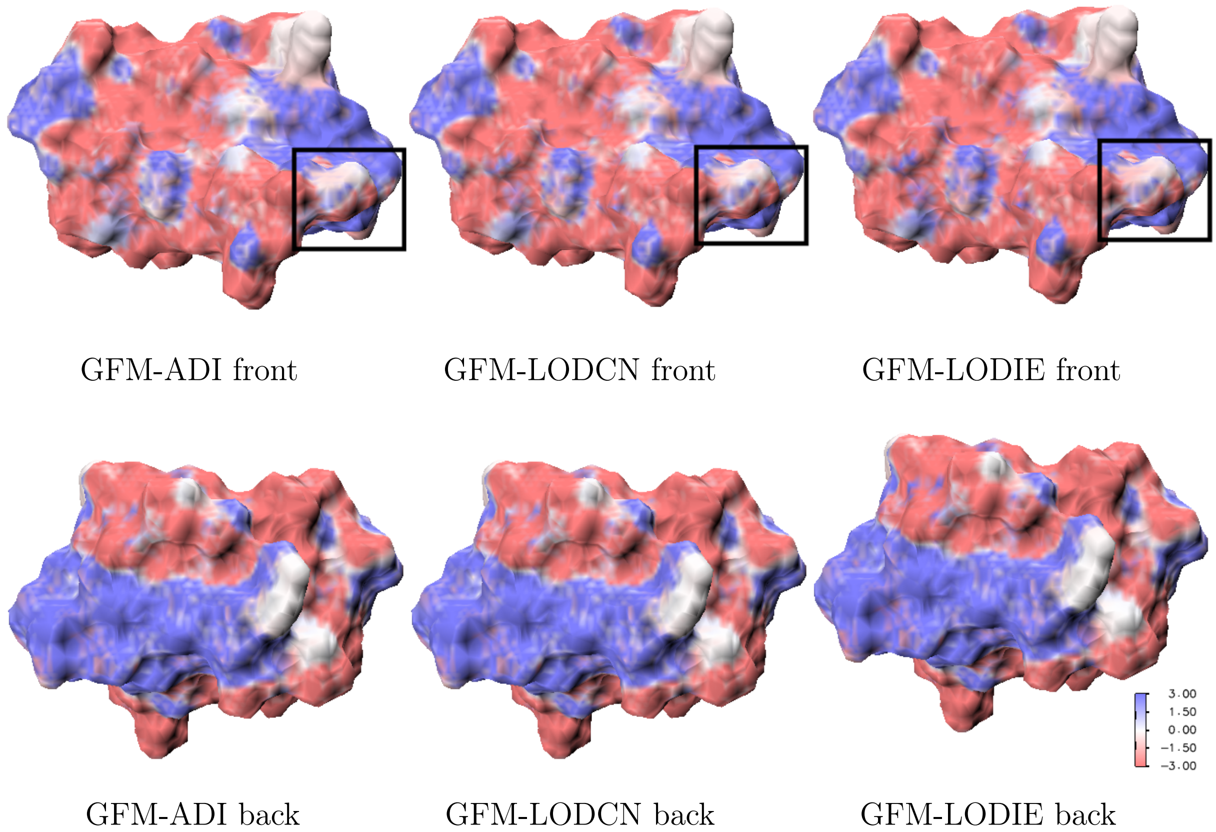
where Q is the trilinear interpolation of the singular charges q_i at the center of the atoms. The potential values are obtained by scaling our calculated dimensionless potentials with the constant 0.596163438 corresponding for the room temperature (300K). In all cases a uniform mesh size $h = 0.5$ and a large stopping time $TEND$ will be used to ensure that the steady state solution is reached. For the dielectric constant we have used $\epsilon^+ = 80$ for water as the solvent and $\epsilon^- = 1$ the region in side the protein. The ionic strength has been used as $I_s = 0.15$.

Here we tried to identify the value of Δt as large as possible without losing the adequate amount accuracy for a real protein like 1cbn. But in Table 6.2 as we have observed for $\Delta t > 0.5$, GFM-ADI diverges totally while other two schemes loses significant amount of accuracy. If we consider the solvation energy ($\approx$ -302 *kcal/mol*) for $\Delta t = 0.005$ to be accurate enough and compare all other solvation energies in Table 6.2, it can be observed that with the increase of Δt in all three proposed schemes looses accuracy but at a different rate. GFM-ADI scheme is usually more accurate while GFM-LODIE schemes more robust to the larger values of Δt. The performance of GFM-LODCN is roughly in between the other two schemes in terms of accuracy and stability. To be uniform with the studies for the other proteins in this paper we have used the optimum value for $\Delta t = 0.05$ and $h = 0.5$. For GFM-ADI, the critical value of Δ (in this case 0.5) for stability gives us a safe zone ($\Delta t < 0.5$), within this safe zone, the GFM-ADI is both accurate and efficient. In comparison for ADI method this critical value is $\Delta t = 0.005$ which results in taking unreasonably longer time to reach the steady state to produce any results.

Δt	ADI	GFM-ADI	GFM-LODCN	GFM-LODIE
0.001	-459.5742719854	-303.00657886	-303.00154088	-302.80556740
0.002	-458.1104685049	-302.99808443	-302.98279503	-302.61375402
0.005	-452.1116748705	-302.90138868	-302.83729044	-302.03341621
0.01	NaN	-302.85075766	-302.68957418	-301.31096348
0.02	NaN	-302.69227062	-302.33466174	-300.06953533
0.05	NaN	-302.04538862	-301.08386563	-297.04896758
0.1	NaN	-300.84648128	-298.86787256	-293.05472908
0.2	NaN	-298.63624081	-294.81110682	-286.88648904
0.5	NaN	-293.21633440	-284.83429816	-274.64462149
0.7	NaN	NaN	-279.04495504	-269.01031327
1	NaN	NaN	-271.29161057	-262.50478312
2	NaN	NaN	-251.54655749	-248.89416804
5	NaN	NaN	-218.29937773	-230.91124478

Table 6.2: Solvation Energy (*kcal/mol*) of 1cbn for $h = 0.5$ and the ionic strength $I_s = 0.15$

In Figure 6.3 we focus on the electrostatic potential on the surface of the protein 1cbn for all three of our prosed schemes. The difference is not that much noticeable unless we focus in the black squared region of the front side. This shows that, even though there are some differences in solvation energies calculated by the proposed three methods there is no significant difference for the electrostatic potentials for our choice for the optimum values of the parameters.



GFM-ADI front GFM-LODCN front GFM-LODIE front

GFM-ADI back GFM-LODCN back GFM-LODIE back

Figure 6.3: Electrostatic potential for 1cbn using $dt = 0.05$ and $h = 0.5$.

6.2.2 Solvation energy of 24 proteins

Next we solved the nonlinear PB equation and computed solvation energy for a collection of 24 proteins as in [9, 12]. The dielectric constants , Ionic strength and all other parameters have been set similar to our study for the protein Crambin(1cbn).

Solvation energies for all three of our proposed schemes have been compared with the rMIB and MIB schemes in Table 6.3. The results from our proposed schemes have been observed to be very close to the results from rMIB and MIB schemes while solving nonlinear PB equation instead of linear PB equation. As we have identified in previous subsections GFM-ADI appeared to be more accurate than the other two of our proposed schemes and obtained the same level of accuracy as rMIB and MIB schemes. Table 6.3 also confirms that if GFM-ADI fails to converge for any protein then GFM-LODCN or

PDB	No. of Atoms	rMIB	ADI	GFM-ADI	GFM-LODCN	GFM-LODIE
1ajj	519	-1139.48	-1371.10	-1139.45	-1139.07	-1133.06
2erl	573	-952.36	-1165.28	-952.13	-951.03	-945.96
1cbn	648	-303.33	-459.51	-302.06	-301.09	-297.06
1vii	596	-902.31	-1154.67	-901.78	-901.36	-895.15
1fca	729	-1204.44	-1458.16	-1205.53	-1205.09	-1199.33
1bbl	576	-988.40	-1302.49	-986.15	-985.72	-979.60
2pde	667	-820.97	-1018.66	-819.17	-819.83	-813.45
1sh1	702	-753.99	-999.92	-751.69	-751.84	-745.15
1vjw	826	-1241.07	-1513.17	-1244.74	-1243.52	-1236.67
1uxc	809	-1139.25	-1478.20	-1138.50	-1135.22	-1128.39
1ptq	795	-873.32	-1170.00	-867.98	-867.40	-859.71
1bor	832	-853.47	-1102.40	-852.49	-851.24	-844.68
1fxd	824	-3321.39	-3653.81	-3321.68	-3321.34	-3313.83
1r69	997	-1088.62	-1419.35	-1085.45	-1084.66	-1076.32
1mbg	903	-1353.31	-1685.70	-1352.63	-1351.30	-1343.68
1bpi	898	-1304.37	-1672.02	-1301.61	-1299.86	-1291.40
1hpt	858	-812.49	-1147.42	-809.02	-808.09	-799.24
451c	1216	-1027.21	-1379.27	-1023.71	-1022.61	-1012.70
1svr	1435	-1711.11	-2257.80	-1707.87	-1706.38	-1693.11
1frd	1478	-2862.50	-3376.35	-2863.69	-2863.03	-2850.42
1a2s	1272	-1921.20	-2292.15	-1919.28	-1917.70	-1907.96
1neq	1187	-1731.71	-2223.08	-1729.87	-1728.61	-1716.47
1a63	2065	-2374.41	-3149.69	-2370.80	-2369.26	-2350.42
1a7m	2809	-2160.34	-2771.41	-2155.05	-2152.48	-2135.73

Table 6.3: Solvation energies ($kcal/mol$) of 24 Proteins considering $\Delta t = 0.001$ for ADI and $\Delta t = 0.05$ for GFM-ADI, GFM-LODCN, GFM-LODIE

GFM-LOD can also be used since they are more stable and the results are not that much different than GFM-ADI.

6.2.3 Binding energy of 2a9x

Binding energies play an important role in viral transcription and antiviral drug designs. In particular a better accuracy of the binding energy of the BIV Tat Protein and BIV TAR RNA in HIV viral replication can significantly help in search for the new antiviral drugs that repress the replication by blocking transactivation of viral RNA transcription [19]. In this section we will demonstrate the ability of GFM-ADI scheme to compute the binding energy of the BIV Tat Protein and BIV TAR RNA.

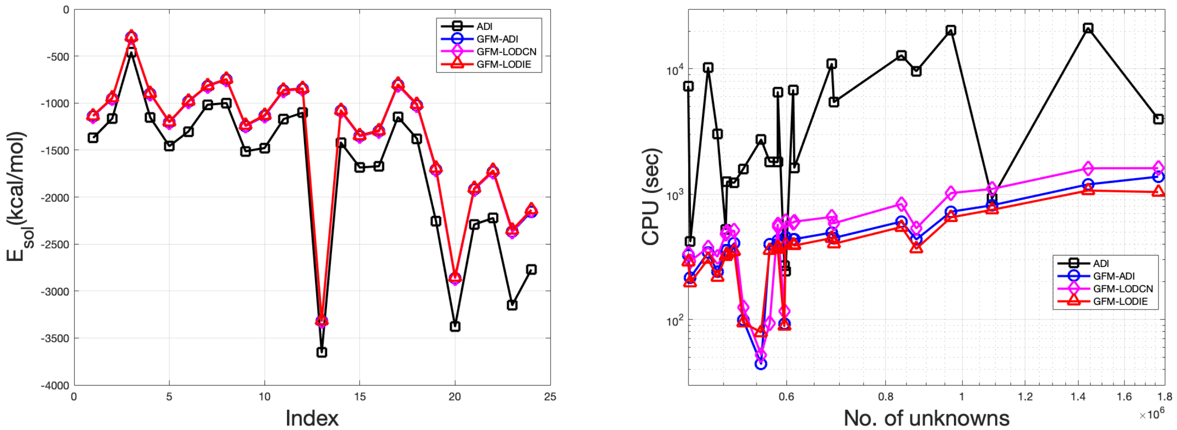


Figure 6.4: Difference in solvation energy(left) and CPU time (right) in solving nonlinear PB eqation for 24 protein.

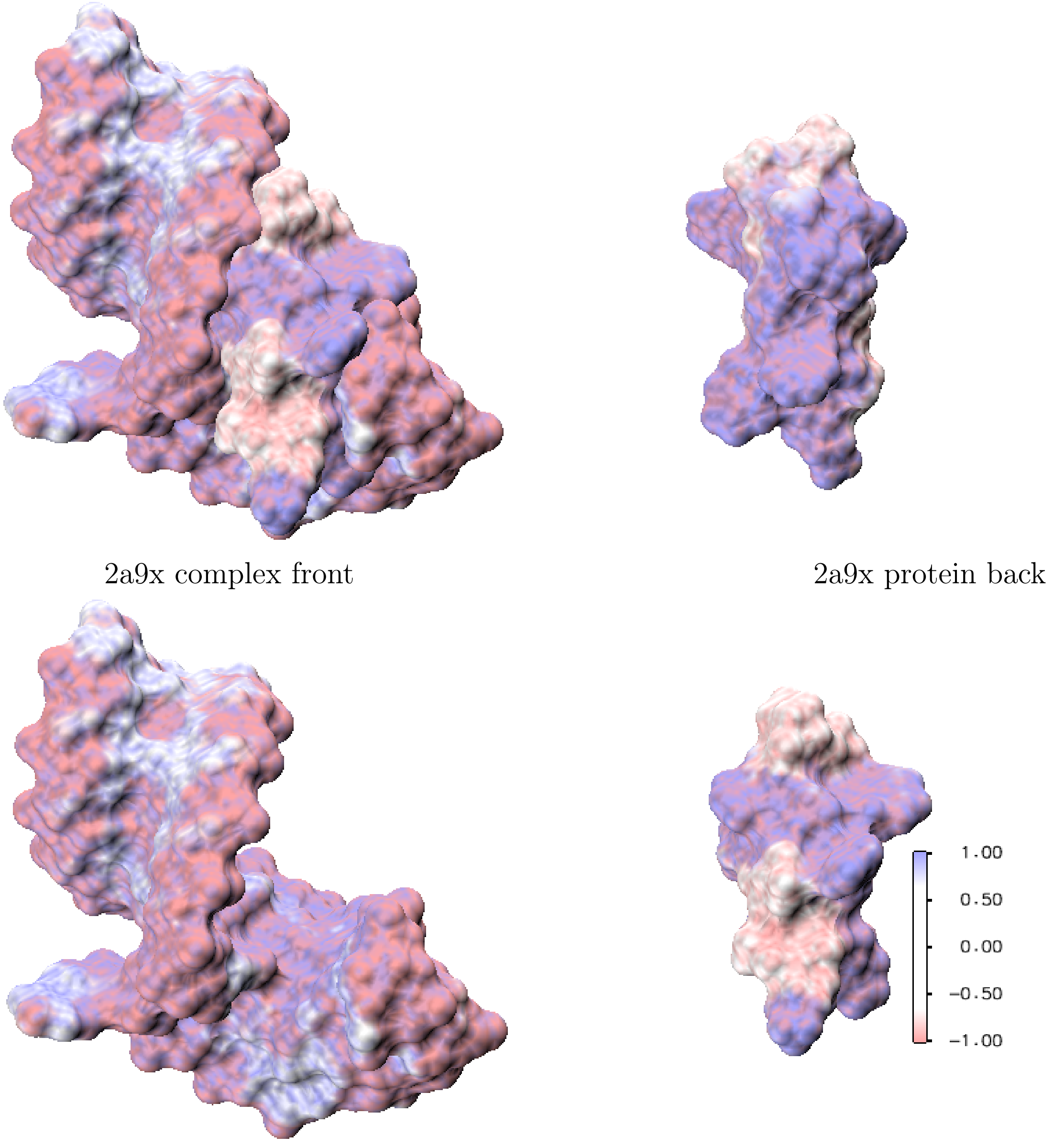
The electrostatic binding free energy can be calculated by the following formula based on the free energy cycle,

$$\begin{aligned} E^{\mathrm{AB}}_{\mathrm{bind}} &= \Delta G^{\mathrm{AB}}_{\mathrm{ele}} - \Delta G^{\mathrm{A}}_{\mathrm{ele}} - \Delta G^{\mathrm{B}}_{\mathrm{ele}} \\ &= [E^{\mathrm{AB}}_{\mathrm{sol}} + E^{\mathrm{AB}}_{\mathrm{cou}}] - [E^{\mathrm{A}}_{\mathrm{sol}} + E^{\mathrm{A}}_{\mathrm{cou}}] - [E^{\mathrm{B}}_{\mathrm{sol}} + E^{\mathrm{B}}_{\mathrm{cou}}] \qquad (6.4) \end{aligned}$$

where the free energies of the complex AB and its monomers A and B on the RHS can be calculated using the calculated solvation energies in equation (5.18).

h	$E^{\mathrm{complex}}_{\mathrm{sol}}$	$E^{\mathrm{protein}}_{\mathrm{sol}}$	$E^{\mathrm{RNA}}_{\mathrm{sol}}$	$E^{\mathrm{complex}}_{\mathrm{bind}}$
		rMIB		
1	-5816.38	-1021.94	-8893.39	383.60
1/2	-5821.22	-1025.86	-8898.54	387.84
1/4	-5823.39	-1026.27	-8900.52	388.05
		GFM-ADI		
1	-5834.10	-1027.14	-8915.17	392.84
1/2	-5824.82	-1025.98	-8905.59	391.38
1/4	-5841.62	-1026.40	-8916.69	386.12

Table 6.4: Binding energy of 2a9x



2a9x complex front

2a9x protein back

2a9x rna front

2a9x protein front

Figure 6.5: Electrostatic potential for 2a9x using GFM-ADI for $\Delta t = 0.05$ and $h = 0.5$

6.2.4 Salt effect on the binding affinity

The nonlinear PB equation is often used to describe the salt effects on the binding of ligands, peptides, and proteins to nucleic acids, membranes, and proteins. In this investigation we have tested the the performance of the proposed GFM-ADI scheme for the evaluation of the salt effect in the protein-protein binding of the complex 1beb and

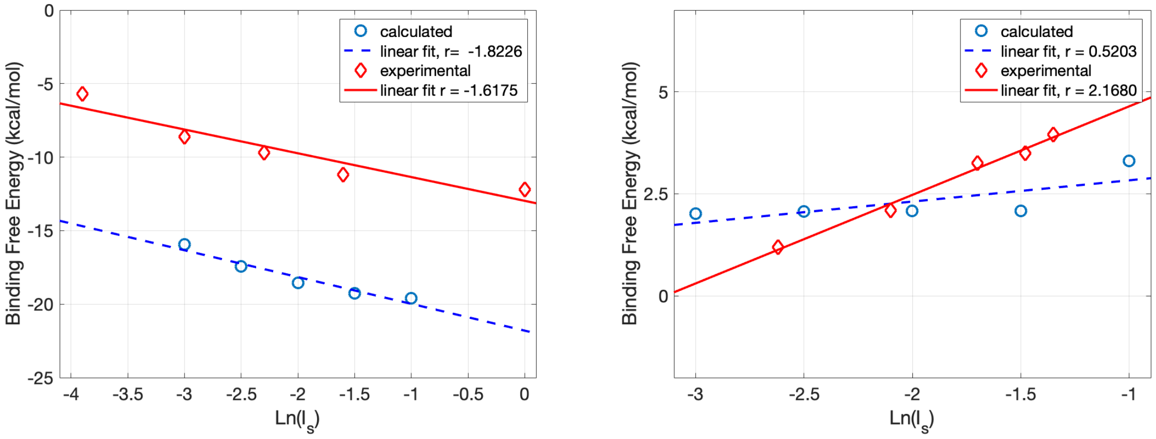
1emv. Physically, the binding affinity can be quantitatively represented based on the binding-free energies, which reflect the non-specific salt dependence of the formation of macro- molecular complexes. The binding affinity is then calculated as the slope ratio of the salt-dependent binding energy at certain salt strength I_s against the natural logarithm of I_s. The electrostatic binding-free energy can be further split into $E_{\text{cou}}(I_s)$'s as the salt-independent parts and $E_{\text{sol}}(I_s)$'s as the salt-dependent parts. The variation of the salt-dependent part of the binding-free energy $\Delta E_{\text{bind}}(I_s)$ can thus be calculated as the difference in $E_{\text{bind}}(I_s)$ for some nonzero salt strength and the zero salt concentration, because the salt independent parts gets simply cancelled. Altogether we have the following formula.

$$\begin{aligned}\Delta E_{\text{bind}}(I_s) &= E_{\text{bind}}(I_s) - E_{\text{bind}}(0) \\ &= [E^{\text{AB}}_{\text{sol}}(I_s) - E^{\text{AB}}_{\text{sol}}(0)] - [E^{\text{A}}_{\text{sol}}(I_s) - E^{\text{A}}_{\text{sol}}(0)] - [E^{\text{B}}_{\text{sol}}(I_s) - E^{\text{A}}_{\text{sol}}(0)] \quad (6.5)\end{aligned}$$

For this study we have used the same model parameters used earlier for 24 proteins. In Figure 6.6 we reported the calculated binding free energy with the experimental results.The slope ration or the binding affinity is calculated and reported in Table (6.5) as in [42]. The results attained by the Lagrangian formulation linearized PB (LFLPB) model [40] are also given in Table (6.5) for comparison. For 1beb the binding affinity calculated by GFM-ADI scheme sufficiently close to experimental data and better than LFLPB. For 1emv the results from GFM-ADI is not as good as those of the LFLPB model but qualitatively it agrees with the experimental observations; that is as the hetero-diemric complex, the binding-free energy increases when the ionic strength is larger. This is probably because the calculation of the binding affinity requires a physical cutoff to obtain two monomers A and B.

		Charges			Slope ratios		
Complex	PDB	AB	A	B	Experimental	GFM-ADI	LFLPB
Lactoglobulin dime(A-B)	1beb	+26	+13	+13	−1.62	−1.82	−2.02
E9Dnase-Im9(10)(B-A)	1emv	−3	−8	+5	2.17	0.52	2.4

Table 6.5: Comparison of binding affinities of the protein complexes 1emv and 1beb



1beb

1emv

Figure 6.6: The salt dependence of the binding affinities

CHAPTER 7

CONCLUSIONS

The pseudo-transient methods and regularization methods are popular methods to solve nonlinear PBE. Even though these two types were successful in circumventing different challenges to solve PBE, each one of them was lacking the advantages of the other one. When we tried to combine them we faced a new challenge due to the new jump conditions being nonzero. This forced us to find a way to apply interface treatments. The MIB method [ ] was a great choice for this type of interface treatments but it would have ruined the tridiagonal structure of the finite difference operator matrix of the 1D equations like (4.19) (4.20) and (4.21) in all of our proposed methods. Having tridiagonal structure for these three equations are very important since we have to solve all three of them at each time step. So for small time step size, it would take unreasonably long time for the whole system to reach the equilibrium state to produce the solution.

So we considered the GFM method [ ] [ ] for $L = 1$ which uses a three point stencil keeping the tridiagonal structure for the 1D equations in our proposed methods. But the original GFM method requires the jump conditions to be in axial directions while the regularized PBE has its jump condition in the normal direction. It motivated us to modify the original GFM method to able to use the normal direction jump condition by considering approximate jump conditions like (5.14) and (5.15).

In comparison with the existing pseudo-transient approaches the GFM-ADI proposed in this dissertation is much more stable than ADI in [11]. This makes GFM-ADI a very practical method, while the ADI method is impractical by requiring too small time step size. In some sense, the GFM-ADI method is even better than LOD methods in [35]. Although LOD methods are unconditionally stable, energies are inaccurate for large time steps. GFM-LODCN and GFMLOD are also producing more accurate results than its predecessor the LOD methods in [35] leaving an option for the cases when GFM-ADI fails to converge.

In future we have a plan to take the advantage of the simplicity of these proposed schemes to the other areas of molecular biophysics like the problem of computing the electrostatic field and forces for molecular dynamic simulations as in [8].

REFERENCES

[1] Q. Cai, J. Wang, H. K. Zhao, and R. Luo. On removal of charge singularity in poisson-boltzmann equation. *Journal of Chemical Physics*, 130(14), 2009.

[2] L. Chen, M. Holst, and J. Xu. The Finite Element Approximation of the Nonlinear Poisson-Boltzmann Equation. *SIAM Journal on Numerical Analysis*, 45(6):2298–2320, 2007.

[3] E. D'Yakonov. Difference schemes with split operators for multidimensional unsteady prob- lems (english translation). *USSR Comp. Math.*, 3:518–607, 1963.

[4] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method). *Journal of Computational Physics*, 152(2):457–492, 1999.

[5] F. Fogolari, A. Brigo, and H. Molinari. The Poisson-Boltzmann equation for biomolecular electrostatics: A tool for structural biology. *Journal of Molecular Recognition*, 15(6):377–392, 2002.

[6] W. Geng. A boundary integral Poisson-Boltzmann solvers package for solvated bimolecular simulations. *Computational and Mathematical Biophysics*, 3(1):54–69, 2015.

[7] W. Geng and R. Krasny. A treecode-accelerated boundary integral Poisson-Boltzmann solver for electrostatics of solvated biomolecules. *Journal of Computational Physics*, 247:62–78, 2013.

[8] W. Geng and G. Wei. Multiscale molecular dynamics using the matched interface and boundary method. *Journal of Computational Physics*, 230(2):435 – 457, 2011.

[9] W. Geng, S. Yu, and G. Wei. Treatment of charge singularities in implicit solvent models. *The Journal of Chemical Physics*, 127(11):114106, 2007.

[10] W. Geng and S. Zhao. Fully implicit ADI schemes for solving the nonlinear Poisson-Boltzmann equation. *Molecular Based Mathematical Biology*, 1:109–123, 2013.

[11] W. Geng and S. Zhao. Fully implicit ADI schemes for solving the nonlinear Poisson-Boltzmann equation. *Molecular Based Mathematical Biology*, 1, Jan. 2013.

[12] W. Geng and S. Zhao. A two-component Matched Interface and Boundary (MIB) regularization for charge singularity in implicit solvation. *Journal of Computational Physics*, 351:25–39, 2017.

[13] M. K. Gilson, M. E. Davis, B. A. Luty, and J. A. McCammon. Computation of electrostatic forces on solvated molecules using the poisson-boltzmann equation. *The Journal of Physical Chemistry*, 97(14):3591–3600, 1993.

[14] M. Holst, J. A. McCammon, Z. Yu, Y. Zhou, and Y. Zhu. Adaptive Finite Element Modeling Techniques for the Poisson-Boltzmann Equation. *Communications in computational physics*, m(1):28, 2010.

[15] M. J. Holst. *Multilevel Methods for the Poisson-Boltzmann Equation.* PhD thesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 1993. UMI Order No. GAX94-11653.

[16] M. J. Holst and F. Saied. Numerical solution of the nonlinear Poisson–Boltzmann equation: Developing more robust and efficient methods. *Journal of Computational Chemistry*, 16(3):337–364, 1995.

[17] W. Im, D. Beglov, and B. Roux. Continuum solvation model: Computation of electrostatic forces from numerical solutions to the poisson-boltzmann equation. *Computer Physics Communication*, 11(1-3):59–75, 1998.

[18] G. Klebe, H. Li, J. H. Jensen, J. E. Nielsen, P. Czodrowski, T. J. Dolinsky, and N. A. Baker. Pdb2pqr: expanding and upgrading automated preparation of biomolecular structures for molecular simulations. *Nucleic Acids Research*, 35(Suppl. 2):W522–W525, 07 2007.

[19] T. C. Leeper, Z. Athanassiou, R. L. Dias, J. A. Robinson, and G. Varani. TAR RNA recognition by a cyclic peptidomimetic of Tat protein. *Biochemistry*, 44(37):12362–12372, 2005.

[20] C. Li and S. Zhao. A matched Peaceman–Rachford ADI method for solving parabolic interface problems. *Applied Mathematics and Computation*, 299:28–44, 2017.

[21] Z. Li and A. Mayo. *ADI Methods for Heat Equations with Discontinuities Along an Arbitrary Interface.* IBM Thomas J. Watson Research Division, 1993.

[22] Z. Li and Y.-Q. Shen. A numerical method for solving heat equations involving interfaces. In *Electronic Journal of Differential Equations, Conf*, volume 3, pages 100–108, 1999.

[23] J. Liu and Z. Zheng. IIM-based ADI finite difference scheme for nonlinear convection-diffusion equations with interfaces. *Applied Mathematical Modelling*, 37(3):1196–1207, 2013.

[24] X.-D. Liu, R. P. Fedkiw, and M. Kang. A Boundary Condition Capturing Method for Poisson's Equation on Irregular Domains. *Journal of Computational Physics*, 160(1):151–178, 2000.

[25] B. A. Luty, M. E. Davis, and J. A. McCammon. Solving the finite-difference non-linear Poisson–Boltzmann equation. *Journal of Computational Chemistry*, 13(9):1114–1118, 1992.

[26] A. Prlic, B. Yukich, B. Beran, C. Zardecki, C. Bi, D. S. Goodsell, D. Dimitropoulos, G. B. Quinn, H. M. Berman, J. Young, J. D. Westbrook, M. Quesada, W. F. Bluhm, P. W. Rose, and P. E. Bourne. The rcsb protein data bank: redesigned web site and web services. *Nucleic Acids Research*, 39(Suppl. 1):D392–D401, 10 2010.

[27] W. Rocchia, E. Alexov, and B. Honig. Extending the applicability of the nonlinear Poisson-Boltzmann equation: Multiple dielectric constants and multivalent ions. *Journal of Physical Chemistry B*, 105(28):6507–6514, 2001.

[28] M. F. Sanner, A. J. Olson, and J.-C. Spehner. Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320.

[29] A. Sayyed-Ahmad, K. Tuncay, and P. J. Ortoleva. Efficient solution technique for solving the Poisson-Boltzmann equation. *Journal of Computational Chemistry*, 25(8):1068–1074, 2004.

[30] K. A. Sharp and B. Honig. Calculating total electrostatic energies with the nonlinear poisson-boltzmann equation. *The Journal of Physical Chemistry*, 94(19):7684–7692, 1990.

[31] A. Shestakov, J. Milovich, and A. Noy. Solution of the Nonlinear Poisson–Boltzmann Equation Using Pseudo-transient Continuation and the Finite Element Method. *Journal of Colloid and Interface Science*, 247(1):62–79, Mar. 2002.

[32] A. I. Shestakov, J. L. Milovich, and A. Noy. Solution of the nonlinear Poisson-Boltzmann equation using pseudo-transient continuation and the finite element method. *Journal of Colloid and Interface Science*, 247(1):62–79, 2002.

[33] J. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. SIAM, 2004.

[34] M. M. Teeter, S. Roe, and N. H. Heo. Atomic resolution (0·83 Å) crystal structure of the hydrophobic protein crambin at 130 k. *Journal of Molecular Biology*, 230(1):292 – 311, 1993.

[35] L. Wilson and S. Zhao. Unconditionally stable time splitting methods for the electrostatic analysis of solvated biomolecules. *International Journal of Numerical Analysis and Modeling*, 13(6):852–878, 2016.

[36] D. Xie. New solution decomposition and minimization schemes for poisson–boltzmann equation in calculation of biomolecular electrostatics. *Journal of Computational Physics*, 275:294 – 309, 2014.

[37] N. Yanenko. Convergence of the method of splitting for the heat conduction equations with variable coefficients (english translation). *USSR Comp. Math.*, 3:1094–1100, 1963.

[38] N. Yanenko. *The method of fractional steps (English translation; originally published in Russian)*. Springer-Verlag, Berlin, Heidelberg, and New York, 1967.

[39] S. Yu, S. Zhao, and G. W. Wei. Local spectral time splitting method for first- and second-order partial differential equations. *Journal of Computational Physics*, 206(2):727–780, 2005.

[40] Zhan, Baker, and W. GW. Differential geometry based solvation model ii: Lagrangian formulation. *Journal of mathematical biology*, 63(6):1139–1200, 2011.

[41] S. Zhao. Pseudo-time-coupled nonlinear models for biomolecular surface representation and solvation analysis. *International Journal for Numerical Methods in Biomedical Engineering*, 27(12):1964–1981, 2011.

[42] S. Zhao. Pseudo-time-coupled nonlinear models for biomolecular surface representation and solvation analysis. *International Journal for Numerical Methods in Biomedical Engineering*, 27(12):1964–1981, Dec. 2011.

[43] S. Zhao. Operator splitting adi schemes for pseudo-time coupled nonlinear solvation simulations. *Journal of Computational Physics*, 257:1000 – 1021, 2014.

[44] S. Zhao. A Matched Alternating Direction Implicit (ADI) Method for Solving the Heat Equation with Interfaces. *Journal of Scientific Computing*, 2015.

[45] Z. Zhou, P. Payne, M. Vasquez, N. Kuhn, and M. Levitt. Finite-difference solution of the poisson–boltzmann equation: Complete elimination of self-energy. *Journal of Computational Chemistry*, 17(11):1344–1351, 1996.