

PSEUDO-TRANSIENT GHOST FLUID METHODS 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

Copyright Sheik Ahmed Ullah 2019
ALL RIGHTS RESERVED

ABSTRACT

The Poisson Boltzmann equation (PBE) is a well-established implicit solvent continuum model for the electrostatic analysis of solvated biomolecules. The numerical solution of 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 pseudo-time solution of the PBE while possesses a novel approach to combine them. A modified Ghost Fluid Method (GFM) has been introduced to incorporate the nonzero jump condition into the ADI framework to construct a new GFM-ADI method. It produced better results in terms of spatial accuracy and stability compared to the existing PBE-ADI methods and it is simpler to implement by circumventing the work necessary to apply the rigorous 3D interface treatments with the regularization. Moreover, the stability of the GFM-ADI method has been significantly improved in comparing with the non-regularized ADI method, so that stable and efficient 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 protein complexes.

DEDICATION

To my loving parents and all the teachers in my life who inspired, helped and cared for me to pursue my interests in mathematics.

INDEX

ACKNOWLEDGMENTS

I wish to express my most sincere gratitude and appreciation to Professor Shan Zhao for his guidance, patience and encouragement. Without his help it would have been difficult for me to be focused to the right direction with all the patience and perseverance necessary before scholarly success. He has always been by my side as a local guardian and it meant a lot to me. His funding from the NSF grant DMS-1812930 titled "*A regularized Poisson Boltzmann model for fast computation of the ensemble average polar solvation energy*" was also a great support for the last part of my Ph. D. program.

I would also like to express my gratitude toward my committee members for their time and support: the external member from the Department of Aerospace Engineering and Mechanics at the University of Alabama, Dr. Muhammad Ali Rob Sharif and the local members from the Department of Mathematics at the University of Alabama, Dr. Brendan Ames, Dr. Mojdeh Rasoulzadeh and Professor David Halpern. I want to specially thank Professor Halpern for his help and guidance to get my access to the High Performance Computing facility from UAHPC and introducing me to the Summer School at the MSRI, UC Berkeley. I wish to express my indebtedness to Professor David Cruz-Uribe for his continuous support to arrange travel funds from the Graduate School, and the College of Arts and Sciences.

I want to take this occasion to thank Dr. Weihua Geng from the Southern Methodists University and Professor Emil Alexov form the Clemson University for their help and support for my research projects on different occasions. Besides Professor Zhao, Dr. Geng is the second most important person who helped and guided me a lot to explore different areas on my research topic.

I also want to recognize Dr. Khanh Dinh, David Neal, Dr. Xuan He, Dr. Timothy Homan and Summer Atkins for their friendship. Dr. Tania Hazra, Arum lee, Siwen Wang

and Hongsong Feng were more than just group mates and always helpful. My former roommate and a graduate student in the Department of Mechanical Engineering at UA, Md Abu Horaira Banna helped me to start using UAHPC which was an important skill to complete a major portion of the Numerical Validations for the proposed methods in this dissertation.

Finally I want to thank my family for supporting me mentally through this program. My parents were always there to encourage and support my early mathematics education. I am really grateful for all the emotional support I got from my wife Elizabeth Brezovich and a lot of encouragement from her father Professor Ivan Brezovich.

CONTENTS

ABSTRACT	ii
DEDICATION	iii
ACKNOWLEDGMENTS	v
LIST OF TABLES	x
LIST OF FIGURES	xi
1 INTRODUCTION	1
1.1 Outlines of this dissertation	4
2 PROTEIN DATA FILE PREPARATION	7
3 POISSON-BOLTZMANN EQUATION AND ADI-MSMS	10
3.1 Poisson-Boltzmann Equation	10
3.2 Molecular surfaces	12
3.3 ADI Method for pseudo transient PBE	13
3.4 Cleaning ADI-MSMS package to develop REG-GFM-MSMS package	15
4 REGULARIZED TIME DEPENDENT PBE	18
4.1 Two-component regularization for singular sources	18

4.2	Pseudo-transient approach for the RPBE	20
4.3	Operator Splitting for the GFM-ADI method	21
4.4	Operator Splitting for the GFM-LODCN method	23
4.5	Operator Splitting for the GFM-LODIE method	25
4.6	Solvation energy from Electrostatic potential	25
5	A NEW DERIVATION FOR GHOST FLUID METHOD	27
5.1	One dimension	28
5.2	Two dimensions	30
5.3	Three dimensions	32
5.4	Corner point for the modified GFM method	34
5.5	Simplicity of the modified GFM	35
6	NUMERICAL VALIDATION	36
6.1	Kirkwood Sphere with analytical solution	36
6.2	Biological applications	41
6.2.1	Protein Crambin	41
6.2.2	Solvation energy of 24 proteins	46
6.2.3	Binding energy of 2a9x	50
6.2.4	Salt effect on the binding affinity	52

7 CONCLUSIONS	54
REFERENCES	55
APPENDIX	60

LIST OF TABLES

2.1 Protein Data file (<i>.pdb</i> type) format	8
6.1 Spatial Convergence test for the krikwood sphere with $\Delta t = 0.001$ and $T_{end} = 10$. Comparable analytical value of the Solvation Energy is $E_{sol} =$ -81.97820845	39
6.2 Stability test by the Solvation Energy (<i>kcal/mol</i>) of <i>1cbn</i> for $h = 0.5$, $T_{end} = 10^4 * \Delta t$	42
6.3 Impact of domain size to the solvation energy of <i>1cbn</i> calculated by the GFM-ADI method.	45
6.4 Solvation energies (<i>kcal/mol</i>) of 24 Proteins considering $\Delta t = 0.001$ for ADI and $\Delta t = 0.05$ for GFM-ADI, GFM-LODCN, GFM-LODIE. Stop- ping condition is set to be either $T_{end} = 50$ or $\Delta E_{sol}^n < 10^{-4}$	47
6.5 Binding energy of <i>2a9x</i>	50
6.6 Comparison of binding affinities of the protein complexes	52

LIST OF FIGURES

2.1 Structure of a DNA (<i>1bna</i>) constructed from its <i>.pqr</i> file showing atomic charges in color for each atoms.	9
3.1 Poisson-Boltzmann Model	10
3.2 2D diagram of the Van der Waals (VdW) Surface (green), Solvent Accessible Surface (SAS) (blue), Solvent Excluded Surface (SES) (red) [30]. . .	12
3.3 Molecular Surface of the protein <i>1ajj</i> generated by MSMS	13
5.1 Irregular grid points marked as colored disks with blue and orange color for the corner points.	27
5.2 1D GFM. Red and Blue colored points are the irregular points.	28
5.3 2D GFM	31
6.1 Stability test for the Krikwood Sphere with $T_{end} = 10^4 * \Delta t$	38
6.2 Temporal Convergence test for the krikwood sphere with $h = 0.125$ and $T_{end} = 10$	40
6.3 Spatial convergence for <i>1cbn</i> with $\Delta t = 0.001$ and $T_{end} = 10$	43
6.4 Steady State Analysis by the Solvation Energy (<i>kcal/mol</i>) of <i>1cbn</i> with $h = 0.5$, $\Delta t = 0.05$ as $t \rightarrow \infty$. (a) GFM-ADI, (b) GFM-LODCN, (c) GFM-LODIE, (d) Solvation Energy difference $\Delta E_{sol}^n(\phi^n)$	44
6.5 Electrostatic potential for <i>1cbn</i> using $\Delta t = 0.05$ and $h = 0.5$	46
6.6 (a) Solvation energy of 24 proteins. (b) Difference in Solvation energy compared to rMIB method	48
6.7 CPU time for the Solvation Energy Calculation of 24 proteins	49
6.8 Electrostatic potential for <i>2a9x</i> using GFM-ADI for $\Delta t = 0.05$ and $h = 0.5$	51
6.9 The salt dependence of the binding affinities	53

CHAPTER 1

INTRODUCTION

Solvated biomolecules and their electrostatic interaction with the surrounding solvent are critical to the studies of various important biological processes 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 powers, the electrostatic analysis of functions and dynamics of bimolecular solvation has become 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 are still critical issues in numerical solution of the PBE for biophysical models with complex geometry, especially for macromolecules containing tens of thousands to millions of atoms.

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 nonlinearity, discontinuous dielectric coefficients, non-smoothness of the solution and singularities in the source term due to the atomic charges. Effects of nonlinearity becomes significant with strong ionic presence [41].

For the nonlinearity, 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 nonlinear relaxation method [19], [32], nonlinear conjugate gradient method [28] or inexact Newton method [18]. The other approach has been introduced recently based on the pseudo-transient continuation idea [34, 36, 48]. This approach converts time independent nonlinear PBE into a time dependent form by introducing a 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.

In pseudo-time methods, the time dependent PBE has to be solved until steady state. To maintain the efficiency, a large time increment for Δt is required. This is why the existing pseudo-time methods usually adopted an implicit scheme in time stepping. Moreover, to convert the three dimensional(3D) PBE into a set of multiple independent one-dimensional(1D) systems, the alternating direction implicit (ADI) methods in [9,12], the locally one dimensional (LOD) method in [41] have been introduced in the literature. Specially in [12] Douglas-Rachford ADI scheme has been used to split the linear subsystem with the 3D laplacian operator into three sub-systems with one dimensional 2nd order derivatives. Altogether this method has 1st order of accuracy in time but a quite severe stability condition,. Later in [41] the LOD method was introduced as an unconditionally stable method with reduced accuracy compared to ADI methods in [12]. Even though these 1D subsystems produced by these ADI and LOD methods are tridiagonal and can be efficiently solved by using the Thomas Algorithm [37], they lack the treatment of the jump conditions at the interface which reduces the spatial 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.

Besides strong nonlinearity, the numerical treatment of charge singularity is an-

other challenge for the PBE. At atom centers, both charge source and potential solution blow up, and the conventional discretization is doomed to be inaccurate. This motivated many authors to develop different regularization methods in [1, 3, 11, 13, 16, 42, 54] to reduce the loss of accuracy due to the singularity. In these methods, the potential function is 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 or Green’s functions. So far these type of regularization methods have never been used with the ADI or LOD type pseudo transient methods.

The dielectric interface is also crucial in numerical discretization of the PBE as it defines the boundary for the solute and solvent regions. Across a geometrically complex dielectric interface, or molecular surface, the potential solution is continuous, but its normal derivative is discontinuous. For un-regularized PBE, the standard finite difference method is still convergent and degenerated to first order convergence in space. However, the situation becomes worse for regularized PBE, because now both potential and its flux will be discontinuous across the interface. The standard finite difference solution will diverge in this case, if no interface treatment is imposed. For this reason, the regularized PBE is usually solved by some special interface schemes, such as matched interface and boundary (MIB) method [2, 11, 45, 46, 51–53]. We note that regularization methods have been applied with the finite element type pseudo transient methods in [6], in which the interface jump conditions can be built in the variational formulations. However, these methods are usually inefficient by solving a large linear system iteratively at each time step.

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 have been developed for solving the diffusion equationin [25, 26]. Then as a continuation

of this approach recently matched ADI (mADI) method was developed in [50] and [24] to combine the MIB method (for interface treatment) with ADI. But these ADI methods were mainly focused on the parabolic equations and have never been applied to the PBE. In fact, the mADI [50] could become cumbersome in treating complicated interface, like the molecular surface in protein studies.

In this study 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 properly treated. For the regularization we have chosen the two component regularization developed in [13] which is the simplest and most accurate regularization method. But it changes the jump condition to be non zero which introduces the necessity of interface treatments. Otherwise the standard central finite difference becomes divergent. Then motivated by the mADI method we have generalised the Ghost Fluid Method (GFM) developed in [8] as the interface treatment for the present study. 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 nonlinear PBE. Generally it is more robust than ADI method but still has a time stability constraint when the time step size is too huge. Then to continue the search of a more stable method for our regularized pseudo continuation approach we replaced ADI scheme by LOD formulation to propose GFM-LODCN and GFM-LODIE methods. These 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 and efficient results than their predecessors. In particular GFM-LODIE has found to be most robust while GFM-ADI to be most accurate.

1.1 Outlines of this dissertation

There are six main parts in this dissertation. The first part discusses the Protein data file preparations necessary for the numerical algorithm developed to calculate the electrostatic potential and the solvation energy by the PBE. The second part introduces

the Poisson-Boltzmann model and discusses the ADI method [12] to give an analytical background of the previous ADI methods, molecular surfaces and coding packages. The third part discusses the two component regularization and introduces three pseudo transient methods GFM-ADI, GFM-LODCN and GFM-LODIE. The fourth part introduces the a new GFM method to incorporate with the proposed pseudo transient methods. The last part validates the proposed method for a benchmark problem and examines the application the newly proposed methods to the PBE model for the real proteins. Below is a breakdown of the following chapters in greater details:

Chapter 2 starts with a description of the Protein Data Bank and its different file formats. A detail description of *.pdb* file is included with the process to convert them to get the *.pqr* file. The types of inputs like *x*, *y* and *z* coordinate, the Van der Walls radius and the charges for the numerical algorithms are discussed in detail.

Chapter 3 reviews the PB model with the Poisson-Boltzmann Equation which will be the problem we will be the center of the discussion in the rest of the dissertation. Different type of Molecular surfaces have been reviewed to explain our choice of SES surface to be generated by the MSMS software. A detail description of the previous ADI method [12] has been incorporated. At the end of the chapter the development of the software package REG-GFM-MSMS from ADI-MSMS has been described.

Chapter 4 reviews the two-component regularization and its incorporation with the pseudo-transient approaches. Three types of operator splitting methods have been proposed in this chapter to the solve the PBE. An analytical background of the calculation of the solvation energy from the solution of the PBE has been discussed.

Chapter 5 introduces a modified version of the Ghost Fluid Method (GFM) and its detail derivation. It also covers a background on other GFM methods.

Chapter 6 examines the numerical validation for the proposed three methods in chapter 4 for the kirkwood sphere problem and other biological problems. For the benchmark problem several tests have been performed to test the stability, the spatial

convergence and the temporal convergence. Similar tests have performed to calculate the solvation energy for a collection 24 proteins to calculate their solvation energies. To calculate other type of biological feature of proteins, the binding energy of HIV viral replication has been performed. As the last test the salt effects on the binding energy of several proteins has been calculated to compare with the experimental datas available.

Chapter 7 finally summarizes the findings made in the this dissertation and proposes some opportunities for the future work.

CHAPTER 2

PROTEIN DATA FILE PREPARATION

In order to calculate electrostatic potential and solvation free energy of real proteins, we have 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 a unique name with four symbols, while each symbol could be either a letter or a number. Among several types of data file available from the PDB we focused on *.pdb* type files available from the website of RCSB [29], one of the member organization of the PDB. Our goal here is to identify all atomic details of proteins, including coordinate (x,y,z) of all atom centers, their van-der Walls radii, and partial charges assigned to each atom.

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. 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: The 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 [20] has been used to calculate the *.pqr* file. These *.pqr* files contain ATOM record types similar to *.pdb* files where the occupancy columns (55-60) and Temperature factor columns (61-66) have been replaced by the charge and radius.

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

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.

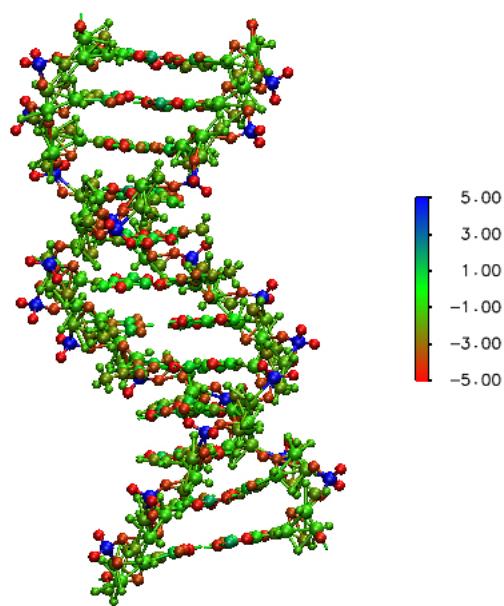


Figure 2.1: Structure of a DNA (*1bna*) constructed from its *.pqr* file showing atomic charges in color for each atoms.

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.

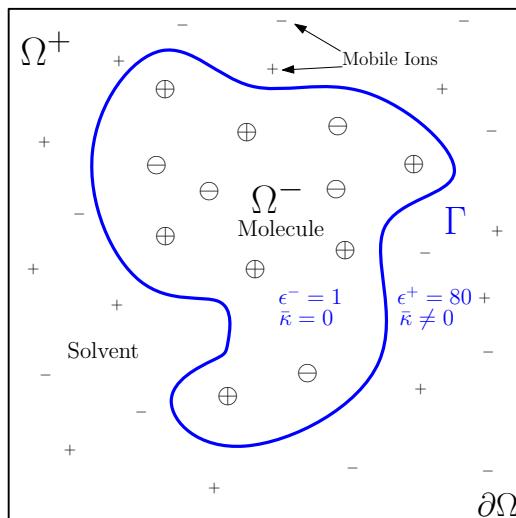


Figure 3.1: Poisson-Boltzmann Model

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,

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

We have used the following approximated analytical solution [18] as the boundary condition,

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

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

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

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

$$[\phi]_{\Gamma} = 0 \text{ and } [\epsilon\phi_{\mathbf{n}}]_{\Gamma} = 0 \quad (3.4)$$

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 i th atom of the solute molecule located at position \mathbf{r}_j . The Debey-Huckel parameter $\bar{\kappa}^2 = \left(\frac{2N_A e_c^2}{100 k_B T}\right) I_s = 8.486902807 \text{ \AA}^{-2} I_s$ from [17] for $\mathbf{r} \in \Omega^-$ and $\bar{\kappa} = 0$ for $\mathbf{r} \in \Omega^+$. Here N_A is the 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 [17].

3.2 Molecular surfaces

Molecular surface plays an important role to solve the PBE equation by setting the interface between Ω^- and Ω^+ . The location of the centers of the atoms in a molecule and their Van der Waals radius have used to construct the following types of molecular surface in the literature.

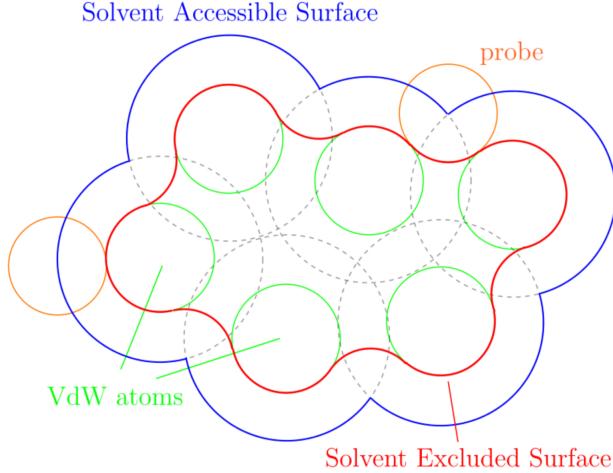


Figure 3.2: 2D diagram of the Van der Waals (VdW) Surface (green), Solvent Accessible Surface (SAS) (blue), Solvent Excluded Surface (SES) (red) [30].

Van der Waals (VdW) surface: The simplest type of molecular surface can be thought of a union of atomic spheres known as an der Waals balls using their Van der Waals radius. The surface generated by this process is called as Van der Waals (VdW) Surface as shown in Figure 3.2. These types of surfaces are often not suitable for PB models due to their singularities at the intersections of the spheres.

SAS and SES surface: Two other kinds of molecular surfaces that are commonly used for PB models are the Solvent Accessible Surface (SAS) and the Solvent Excluded Surface (SES) introduced by Lee and Richards in 1970's [21, 31]. In their approach, the solvent molecules surrounding a solute molecule are reduced to spherical probes [39]. The SAS is defined as the path of the center of the solvent probe when it roles over the solute molecule. That is, SAS is the surface enclosing the region in which

the *center* of the solvent probe can not enter. On the other hand, SES is defined as the surface enclosing the region in which the *surface* of the solvent probe can not enter. The SAS surface might have the same type of singularity but SES is known for its smoothness and called “the smooth molecular surface” or “the Connolly surface”, due to Connolly’s fundamental work [5]. Besides these sharp interfaces there are other soft interfaces like the Gaussian interface has been considered for PBE [15].

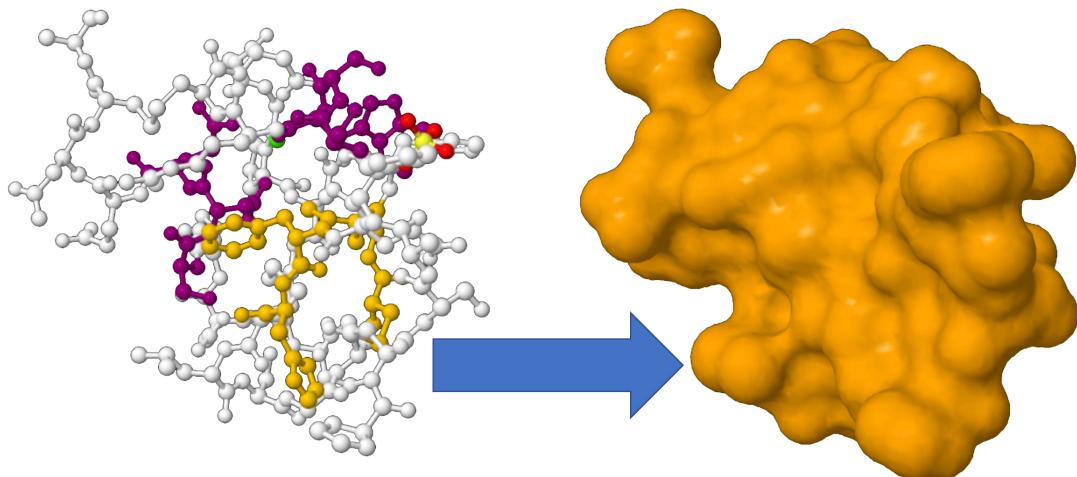


Figure 3.3: Molecular Surface of the protein *1ajj* generated by MSMS

For our computations we used the *reduced surface* developed by Sanner and his team [33] at the Molecular Graphics Lab oratory (TSRI) along with the MSMS (Michel Sanner’s Molecular Surface) algorithm for computing an approximate analytical representation of the SES . 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 create the molecular surface as shown in Figure 3.3 to decide whether a grid point is inside or outside the solute molecule.

3.3 ADI Method for pseudo transient PBE

The pseudo-transient variation to the PBE has become popular approach for solving the nonlinear PBE [34,36,48,49]. In this approach equation (3.1) is converted to the following time dependent PBE (TPBE),

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

Here only ϕ is time dependent, while all other functions in (3.5) i.e. ϵ, ρ and $\bar{\kappa}^2$ are time independent. 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 $\mathbf{U}^n = \phi_{ijk}^n$ for $i = 1, \dots, N_x, j = 1, \dots$, and $N_y, k = 1, \dots, N_z$ denote all the nodal values of ϕ at the time level t_n . Two stages of operator splitting schemes has been used later for updating ϕ^n at time level t_n to ϕ^{n+1} at time level $t_{n+1} = t_n + \Delta t$.

At the first stage equation (3.5) has been separated into the following two equations using a first order time splitting,

$$\frac{\partial w}{\partial t} = -\bar{\kappa}^2 \sinh(w) \text{ with } \mathbf{W}^n = \mathbf{U}^n \text{ and } t \in [t_n, t_{n+1}] \quad (3.6)$$

$$\frac{\partial v}{\partial t} = \nabla \cdot (\epsilon \nabla v) + \rho(\mathbf{r}) \text{ with } \mathbf{V}^n = \mathbf{W}^{n+1} \text{ and } t \in [t_n, t_{n+1}] \quad (3.7)$$

Where $\mathbf{U}^{n+1} = \mathbf{V}^{n+1}$. Then the following analytical solution (3.8) has been used for the equation (3.6) in [12].

$$\mathbf{W}^{n+1} = \ln \left(\frac{\cosh(\frac{1}{2}\bar{\kappa}^2 \Delta t) + \exp(-\mathbf{W}^n) \sinh(\frac{1}{2}\bar{\kappa}^2 \Delta t)}{\exp(-\mathbf{W}^n) \cosh(\frac{1}{2}\bar{\kappa}^2 \Delta t) + \sinh(\frac{1}{2}\bar{\kappa}^2 \Delta t)} \right) \quad (3.8)$$

Which is helpful to avoid the difficulty due to the nonlinear term with $\sinh(.)$ in (3.1). Here the right hand side of equation (3.8) is just a function of W and Δt which allows to rewrite the equation (3.8) as $\mathbf{W}^{n+1} = F(\mathbf{W}^n, \Delta t)$ to facilitate the following discussion. Then for the discretization of equation (3.7), Backward-Euler integration in time and central differencing in space results in,

$$v_{i,j,k}^{n+1} = v_{i,j,k}^n + \Delta t (\Delta_x^2 + \Delta_y^2 + \Delta_z^2) v_{i,j,k}^{n+1} + \Delta t \rho(\mathbf{r}) \quad (3.9)$$

Where Δ_x , Δ_y and Δ_z are the central finite difference operators in x , y and z directions defined as,

$$\begin{aligned}\Delta_x^2(v_{i,j,k}^n) &= \frac{1}{h^2} \left(\epsilon_{i+\frac{1}{2},j,k}(v_{i+1,j,k}^n - v_{i,j,k}^n) + \epsilon_{i-\frac{1}{2},j,k}(v_{i-1,j,k}^n - v_{i,j,k}^n) \right) \\ \Delta_y^2(v_{i,j,k}^n) &= \frac{1}{h^2} \left(\epsilon_{i,j+\frac{1}{2},k}(v_{i,j+1,k}^n - v_{i,j,k}^n) + \epsilon_{i,j-\frac{1}{2},k}(v_{i,j-1,k}^n - v_{i,j,k}^n) \right) \\ \Delta_z^2(v_{i,j,k}^n) &= \frac{1}{h^2} \left(\epsilon_{i,j,k+\frac{1}{2}}(v_{i,j,k+1}^n - v_{i,j,k}^n) + \epsilon_{i,j,k-\frac{1}{2}}(v_{i,j,k-1}^n - v_{i,j,k}^n) \right)\end{aligned}\quad (3.10)$$

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.

For the second stage in operator splitting, the following Douglas-Rachford type ADI scheme has been introduced for the equation (3.9) in [12].

$$\begin{aligned}(1 - \Delta t \Delta_x^2) v_{i,j,k}^* &= [1 + \Delta t (\Delta_y^2 + \Delta_z^2)] v_{i,j,k}^n + \Delta t \rho(\mathbf{r}) \\ (1 - \Delta t \Delta_y^2) v_{i,j,k}^{**} &= v_{i,j,k}^* - \Delta t \Delta_y^2 (v_{i,j,k}^n) \\ (1 - \Delta t \Delta_z^2) v_{i,j,k}^{n+1} &= v_{i,j,k}^{**} - \Delta t \Delta_z^2 (v_{i,j,k}^n)\end{aligned}\quad (3.11)$$

The ADI methods discussed in [12] 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 will observe in the chapter 6, these two issues reduced the accuracy and stability of the ADI method significantly.

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

To simulate the ADI Methods in [12] and the LOD Methods in [41], Professor Shan Zhao, Dr. Weihua Geng form the Southern Methodists University 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 a sub-packages called FISHPACK90 which 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-order and fourth-order finite difference approximations to the elliptic Partial Differential Equations. The ADI-MSMS package has also been used to 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.

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 with 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 solver package REG-GFM-MSMS to incorporate all three of our proposed numerical methods to solve the PBE.

CHAPTER 4

REGULARIZED TIME DEPENDENT PBE

4.1 Two-component regularization for singular sources

To avoid the difficulty due to the PBE for the vacuum case Cai, Wang and Zhao proposed a two component regularization in [1]. For this regularization the electrostatic potential ϕ has been considered as the addition of the coulomb component ϕ_c and the reaction field component ϕ_{RF} . Here ϕ_c satisfies the following Poission's equation,

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

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

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

Now the PBE from equation (3.1) can be rewritten as,

$$-\nabla \cdot (\epsilon \nabla (\phi_C(\mathbf{r}))) - \nabla \cdot (\epsilon \nabla (\phi_{RF}(\mathbf{r}))) + \bar{\kappa}^2(\mathbf{r}) \sinh(\phi_C(\mathbf{r}) + \phi_{RF}(\mathbf{r})) = \rho(\mathbf{r}), \quad (4.3)$$

The reaction field component ϕ_{RF} can be physically interpreted as the electrostatic field generated by the charges induced by replacing the solvent around the solute molecule. It changes the dielectric constant in Ω^+ form ϵ^+ to ϵ^- [1]. Now substituting (4.1) into (4.3) we have,

$$-\nabla(\epsilon^+ \phi_{RF}(\mathbf{r})) + \bar{\kappa}^2 \sinh(\phi_C(\mathbf{r}) + \phi_{RF}(\mathbf{r})) = 0 \text{ in } \Omega^- \cup \Omega^+ \quad (4.4)$$

More explicitly ϕ_{RF} satisfies the following elliptic interface problem as described in [3],

$$-\nabla \cdot (\epsilon^- \nabla \phi_{RF}) = 0 \text{ in } \Omega^- \quad (4.5)$$

$$-\nabla(\epsilon^+ \phi_{RF}) + \bar{\kappa}^2 \sinh(\phi_C + \phi_{RF}) = 0 \text{ in } \Omega^+ \quad (4.6)$$

$$[\phi_{RF}] = 0 \text{ on } \Gamma \quad (4.7)$$

$$\left[\epsilon \frac{\partial \phi_{RF}}{\partial n} \right] = (\epsilon^+ - \epsilon^-) \frac{\partial G}{\partial n} \text{ on } \Gamma \quad (4.8)$$

$$\phi_{RF} = \phi_b - G \text{ on } \partial\Omega \quad (4.9)$$

Now there is no singular term on the right hand side but we still have several numerical difficulties left to address. In [16] it has been reported that ϕ_C and ϕ_{RF} have different signs and their magnitude is much larger than ϕ . Here a relatively small error in ϕ_{RF} will produce a relatively larger error in ϕ given that the ϕ_C is analytically calculated. Sometimes this amplifying factor [16] can be as large as $(\epsilon^+/\epsilon^- - 1)$. In our case this factor is about 79 by taking $\epsilon^+ = 80$ and $\epsilon^- = 1$. Another problem is, the calculation of ϕ_C is necessary for all N grid points in Ω^+ at a computational cost $O(N^2)$ which is very expensive for large N .

These difficulties motivated us to consider a newer version of the two component regularization proposed by Luo and his collaborators in [1]. They proposed to solve for the whole original solution ϕ in Ω^+ instead of the reaction component ϕ_{RF} only. Then to make the required adjustments to fit this regularization approach with finite difference and finite element methods, Professor Zhao and Dr. Geng proposed a new interface problem [13] with discontinuous flux jumps for the regularized potential $\tilde{\phi}$. In particular they defined it as,

$$\tilde{\phi} = \begin{cases} \phi_{RF} & \text{in } \Omega^- \\ \phi_C + \phi_{RF} & \text{in } \Omega^+ \end{cases} \quad (4.10)$$

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

$$\phi^+ = \phi_{RF}^- + \phi_C^-, \text{ and } \epsilon^+ \frac{\partial \phi^+}{\partial n} = \epsilon^- \frac{\partial \phi_{RF}^-}{\partial n} + \epsilon^- \frac{\partial \phi_C^-}{\partial n} \text{ on } \Gamma \quad (4.11)$$

Hence, the regularized PBE (RPBE) with the corresponding interface and the boundary conditions takes the following form,

$$-\nabla \cdot (\epsilon^- \nabla \tilde{\phi}) = 0 \text{ in } \Omega^- \quad (4.12)$$

$$-\nabla \cdot (\epsilon^+ \nabla \tilde{\phi}) + \bar{\kappa}^2 \sinh(\tilde{\phi}) = 0 \text{ in } \Omega^+ \quad (4.13)$$

$$[\tilde{\phi}] = G \text{ on } \Gamma \quad (4.14)$$

$$\left[\epsilon \frac{\partial \tilde{\phi}}{\partial n} \right] = \epsilon^- \frac{\partial G}{\partial n} \text{ on } \Gamma \quad (4.15)$$

$$\tilde{\phi} = \phi_b \text{ on } \partial\Omega \quad (4.16)$$

Now we can summarize from equation (4.12) to equation (4.16) as,

$$-\nabla \cdot (\epsilon \nabla \tilde{\phi}) + \bar{\kappa}^2 \sinh(\tilde{\phi}) = 0 \text{ in } \Omega^- \cup \Omega^+ \quad (4.17)$$

Then in [13] Professor Zhao and Dr. Geng proposed to numerically solve the regularized PB interface problem (RPBE) given in (4.12)-(4.16) for $\tilde{\phi}$ and finally recover the original solution as $\phi = \tilde{\phi}$ in Ω^+ and $\phi = \tilde{\phi} + G$ in Ω^- . Here the Green's function G can be calculated analytically from (4.2). We note that in jump conditions (4.14) and (4.15) both solution and its flux are discontinuous.

4.2 Pseudo-transient approach for the RPBE

As the ADI method discussed in section 3.3, a pseudo-time derivative has been introduced in the RPBE to solve the PBE as,

$$\frac{\partial \tilde{\phi}(\mathbf{r}, t)}{\partial t} = \nabla \cdot (\epsilon(\mathbf{r}) \nabla \tilde{\phi}(\mathbf{r}, t)) - \bar{\kappa}^2(\mathbf{r}) \sinh(\tilde{\phi}(\mathbf{r}, t)) \text{ in } \Omega^- \cup \Omega^+ \quad (4.18)$$

Here the time independent RPBE in (4.17) has been converted in to a time dependent regularized PBE (TRPBE) in (4.18). Unlike TPBE in (3.5), TRPBE does not have any singular term $\rho(\mathbf{r})$.

As the initial condition, we used the electrostatic potential solved from a linearized PBE [48] or trivially $\tilde{\phi} = 0$. Then numerically integrate (4.18) for a sufficiently long period to get the steady state solution as the solution of the original regularized PBE (4.17). Here the sign on the right hand side of equation (4.18) has been considered as the reverse of the equation(4.17) to ensure the numerical stability.

But, there are still some challenges left for the numerical integration of the TRPBE (4.18) because of the requirement of long time integration, when explicit time stepping methods are usually not efficient [34, 36, 48, 49]. Hence we are employing a semi-implicit time splitting method [34, 36], which have been commonly used to solve the TPBE (3.5) 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 $\mathbf{u}^n = \tilde{\phi}_{ijk}^n$ for $i = 1, \dots, N_x, j = 1, \dots$, and $N_y, k = 1, \dots, N_z$ denote all the nodal values of $\tilde{\phi}$ at the time level t_n . Two stages of operator splitting schemes have been used later for updating $\mathbf{u}^n = \tilde{\phi}^n$ at time level t_n to $\mathbf{u}^{n+1} = \tilde{\phi}^{n+1}$ at time level $t_{n+1} = t_n + \Delta t$. In these two stages at each time step we will develop several types of operator splitting schemes for updating \mathbf{u}^n .

4.3 Operator Splitting for the GFM-ADI method

In this scheme at the first stage, TRPBE in (4.18) will be solved by a first order time splitting into the two following equations similar to the ADI method discussed in

section (3.3),

$$\frac{\partial w}{\partial t} = -\bar{\kappa}^2 \sinh(w) \text{ with } \mathbf{w}^n = \mathbf{u}^n \text{ and } t \in [t_n, t_{n+1}] \quad (4.19)$$

$$\frac{\partial v}{\partial t} = \nabla \cdot (\epsilon \nabla v) \text{ with } \mathbf{v}^n = \mathbf{w}^{n+1} \text{ and } t \in [t_n, t_{n+1}] \quad (4.20)$$

For the first equation (4.19) we will use the analytical solution $\mathbf{W}^{n+1} = F(\mathbf{W}^n, \Delta t)$ defined in (3.8). Then for the temporal discretization of the equation (4.20) we have used the Backward-Euler integration in time to get,

$$v_{i,j,k}^{n+1} = v_{i,j,k}^n + \Delta t (\delta_x^2 + \delta_y^2 + \delta_z^2) v_{i,j,k}^{n+1} \quad (4.21)$$

Now we can not use the definition of Δ_x^2, Δ_y^2 and Δ_z^2 in (3.11) for δ_x^2, δ_y^2 and δ_z^2 , since both the potential function $\tilde{\phi}$ and it's flux are discontinuous on the interface Γ as in equations (4.14) and (4.15). So we define δ_x^2, δ_y^2 and δ_z^2 as the central finite difference operators in

$$(4.22) \text{ in the } x, y \text{ and } z \text{ directions, respectively with } \epsilon_{i,j,k} = \begin{cases} \epsilon^- & \text{if } x_{i,j,k} \in \Omega^- \cup \Gamma \\ \epsilon^+ & \text{if } x_{i,j,k} \in \Omega^+ \end{cases} \text{ as,}$$

$$\begin{aligned} \delta_x^2 (v_{i,j,k}^n) &= \frac{\epsilon_{i,j,k}}{h^2} (v_{i-1,j,k}^n - 2v_{i,j,k}^n + v_{i+1,j,k}^n) \\ \delta_y^2 (v_{i,j,k}^n) &= \frac{\epsilon_{i,j,k}}{h^2} (v_{i,j-1,k}^n - 2v_{i,j,k}^n + v_{i,j+1,k}^n) \\ \delta_z^2 (v_{i,j,k}^n) &= \frac{\epsilon_{i,j,k}}{h^2} (v_{i,j,k-1}^n - 2v_{i,j,k}^n + v_{i,j,k+1}^n) \end{aligned} \quad (4.22)$$

Even though these three point stencils for δ_x^2, δ_y^2 and δ_z^2 give a huge advantage for the regular grid points, it posses a difficulty for the points adjacent to the interface. We will discuss more about this difficulty later in Chapter 5.

Now for the second stage of the operator splitting, a first order Douglas-Rachford type ADI scheme has been used to decompose the diffusion equation (4.21) in x, y and

z directions as,

$$(1 - \Delta t \delta_x^2) v_{i,j,k}^* = [1 + \Delta t (\delta_y^2 + \delta_z^2)] v_{i,j,k}^n \quad (4.23)$$

$$(1 - \Delta t \delta_y^2) v_{i,j,k}^{**} = v_{i,j,k}^* - \Delta t \delta_y^2 (v_{i,j,k}^n) \quad (4.24)$$

$$(1 - \Delta t \delta_z^2) v_{i,j,k}^{n+1} = v_{i,j,k}^{**} - \Delta t \delta_z^2 (v_{i,j,k}^n) \quad (4.25)$$

Where v^* and v^{**} are two intermediate values to create three tridiagonal one dimensional system. Here, the three dimensional linear algebraic system in equation (4.21) has been decomposed into several one dimensional linear algebraic systems in (4.23), (4.24) and (4.25). The finite difference matrix for each one these linear equations has a tridiagonal structure. These three tridiagonal systems are much more efficient to solve than one non-structured system in (4.21). Then by eliminating $v_{i,j,k}^*$ and $v_{i,j,k}^{**}$ and solving for $v_{i,j,k}^{n+1}$ in (4.21) we get,

$$\begin{aligned} v_{i,j,k}^{n+1} &= v_{i,j,k}^n + \Delta t (\delta_x^2 + \delta_y^2 + \delta_z^2) v_{i,j,k}^{n+1} - \Delta t^2 (\delta_x^2 \delta_y^2 + \delta_x^2 \delta_z^2 + \delta_z^2 \delta_y^2) (v_{i,j,k}^{n+1} - v_{i,j,k}^n) \\ &\quad + \Delta t^3 \delta_x^2 \delta_y^2 \delta_z^2 (v_{i,j,k}^{n+1} - v_{i,j,k}^n) \end{aligned} \quad (4.26)$$

Hence the Douglas-Rachford scheme (4.21) is a higher order perturbation of the Implicit-Euler method. Since both (4.19) and (4.20) are first order in time this proposed GFM-ADI method is of first order accuracy in time. For the boundary conditions we use the same Dirichlet boundary boundary values from equation 3.2 for v , v^* and v^{**} for u . The entire time integration here is fully implicit.

4.4 Operator Splitting for the GFM-LODCN method

In this operator splitting scheme at each time step from t_n to t_{n+1} , the TRPBE (4.18) will be solved by a second order time splitting methods into the following three

equations. [47],

$$\frac{\partial w}{\partial t} = -\frac{1}{2}\bar{\kappa}^2 \sinh(w) \text{ with } \mathbf{w}^n = \mathbf{u}^n \text{ and } t \in [t_n, t_{n+1}] \quad (4.27)$$

$$\frac{\partial v}{\partial t} = \nabla \cdot (\epsilon \nabla v) \text{ with } \mathbf{v}^n = \mathbf{w}^{n+1} \text{ and } t \in [t_n, t_{n+1}] \quad (4.28)$$

$$\frac{\partial \tilde{w}}{\partial t} = -\frac{1}{2}\bar{\kappa}^2 \sinh(\tilde{w}) \text{ with } \tilde{\mathbf{w}}^n = \mathbf{v}^{n+1} \text{ and } t \in [t_n, t_{n+1}] \quad (4.29)$$

We then have $\mathbf{u}^{n+1} = \tilde{\mathbf{w}}^{n+1}$. Similar to the GFM-ADI method in section 4.3, at the first stage we will use an analytical integration for the nonlinear equations 4.27 and 4.29. Symbolically, we have $\mathbf{w}^{n+1} = F(\mathbf{w}^n, \frac{\Delta t}{2})$ and $\tilde{\mathbf{w}}^{n+1} = F(\tilde{\mathbf{w}}^n, \frac{\Delta t}{2})$, where F is defined in equation (3.8).

Then for the second stage, we have proposed another multiplicative operator splitting scheme called as Locally One Dimensional (LOD) scheme to solve the diffusion equation (4.28). This types of fractional step methods were first developed by Russian mathematicians in [7, 43, 44]. The discretization of equation (4.28) using Crank-Nicolson integration in time and central differencing in space results in,

$$\left(1 - \frac{\Delta t}{2}(\delta_x^2 + \delta_y^2 + \delta_z^2)\right) v_{i,j,k}^{n+1} = \left(1 + \frac{\Delta t}{2}(\delta_x^2 + \delta_y^2 + \delta_z^2)\right) v_{i,j,k}^n \quad (4.30)$$

Which can be decomposed into x, y and z directions to give the LOD scheme for (4.28) as,

$$\begin{aligned} \left(1 - \frac{\Delta t}{2}\delta_x^2\right) v_{i,j,k}^* &= \left(1 + \frac{\Delta t}{2}\delta_x^2\right) v_{i,j,k}^n \\ \left(1 - \frac{\Delta t}{2}\delta_y^2\right) v_{i,j,k}^{**} &= \left(1 + \frac{\Delta t}{2}\delta_y^2\right) v_{i,j,k}^* \\ \left(1 - \frac{\Delta t}{2}\delta_z^2\right) v_{i,j,k}^{n+1} &= \left(1 + \frac{\Delta t}{2}\delta_z^2\right) v_{i,j,k}^{**} \end{aligned} \quad (4.31)$$

Similar to the GFM-ADI method described in Section 4.3, the tridiagonal systems in (4.31) can be efficiently solved by the Thomas algorithm.

4.5 Operator Splitting for the GFM-LODIE method

For this method at each time step from t_n to t_{n+1} , the t TRPBE (4.18) will be solved in two stages similar to the GFM-ADI scheme described in the section 4.3. The first stage is exactly similar to the GFM-ADI method. The first order time splitting will be used for TRPBE (4.18) to generate equation (4.19) and (4.20). Equation (4.19) will be solved analytically by the function $\mathbf{w}^{n+1} = F(\mathbf{w}^n, \Delta t)$ defined in equation (3.8).

Then for the second stage, we have applied the LOD scheme described in the section 4.4 to decompose equation (4.21) into x, y and z direction as,

$$\begin{aligned}(1 - \Delta t \delta_x^2) v_{i,j,k}^* &= v_{i,j,k}^n \\ (1 - \Delta t \delta_y^2) v_{i,j,k}^{**} &= v_{i,j,k}^* \\ (1 - \Delta t \delta_z^2) v_{i,j,k}^{n+1} &= v_{i,j,k}^{**}\end{aligned}\tag{4.32}$$

Which will solved agin individually by the Thomas Algorithm like the previous sections.

4.6 Solvation energy from Electrostatic potential

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

$$\Delta G_{\text{ele}} = E_{\text{cou}} + E_{\text{sol}}\tag{4.33}$$

$$E_{\text{cou}} = \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} \frac{q_i q_j}{\epsilon^- d_{i,j}}, i \neq j\tag{4.34}$$

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 [35] described the calculation electrostatic free energy

by,

$$\Delta G_{\text{ele}} = \int_{\mathbb{R}^3} \left(\phi\rho + \Delta\Pi - \frac{1}{2}\epsilon|\mathbf{E}|^2 \right) d\mathbf{r} \quad (4.35)$$

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 chapter 6 we omitted the energy components related to the mobile ion pressure and the electrostatic stress to report the solvation energy E_{sol} as,

$$E_{\text{sol}} = \Delta G_{\text{ele}} - E_{\text{cou}} \approx \sum_{i=1}^{N_c} q_i \phi_{RF}(\mathbf{r}_i) \quad (4.36)$$

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 [10, 14, 35] for more details.

CHAPTER 5

A NEW DERIVATION FOR GHOST FLUID METHOD

At the points adjacent to the interface Γ , the central difference operators defined in (4.22)-(4.22) is not applicable. These points are defined as the irregular points where at-least one of its adjacent points is on the other side of the interface Γ . On the irregular points at least one of the three point stencils for δ_x^2, δ_y^2 and δ_z^2 is across the interface. Here the information about the function v is not available for the side point which is on the other side of the interface. 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 this chapter using the fictitious points (or ghost points) and the jump conditions in (4.14) and (4.15).

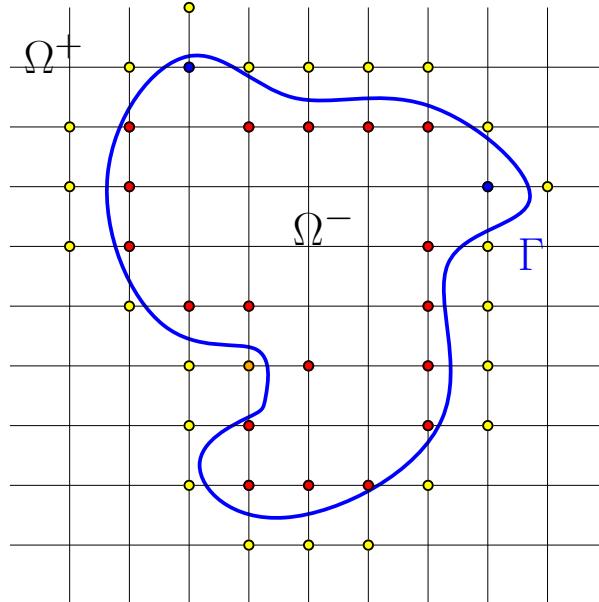


Figure 5.1: Irregular grid points marked as colored disks with blue and orange color for the corner points.

Ghost Fluid Method(GFM) is a sharpe interface technique introduced in [8] 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 [27] to solve elliptic equations with variable coefficients. But in contrast to their method [27], 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

For an one dimensional representation of the proposed GFM schemes we try to evaluate the finite difference operator δ_{xx} at the irregular points where the interface is at x_Γ between x_i and x_{i+1} as the following figure.

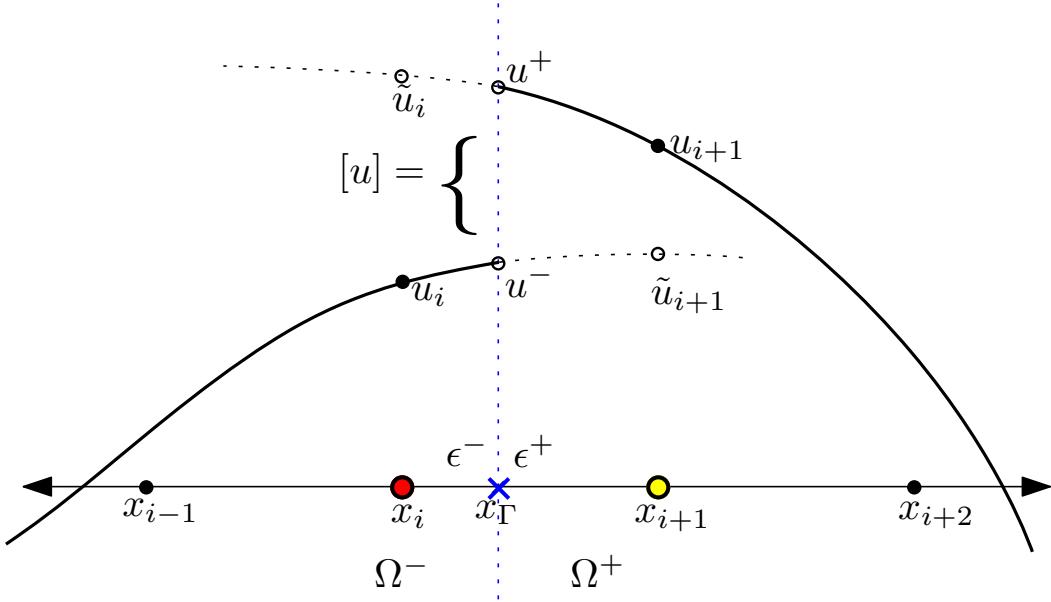


Figure 5.2: 1D GFM. Red and Blue colored points are the irregular points.

Here u satisfies the PBE as $u = \begin{cases} u_{in} & \text{in } \Omega^- \\ u_{out} & \text{in } \Omega^+ \end{cases}$ and 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 equations,

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

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 5.2, we have $\tilde{u}_i \approx u_{out}(x_i)$ and $\tilde{u}_{i+1} \approx u_{in}(x_{i+1})$. 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+1} .

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

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

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

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

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

$$\begin{aligned}\delta_x^2(u_i) &= \frac{1}{h^2} (a_1 u_{i-1} + b_1 u_i + c_1 u_{i+1}) + \frac{\epsilon^-}{h^2} \left(e_1 \cdot [u] + f_1 \cdot \left[\epsilon \frac{\partial u}{\partial x} \right] \right) \\ \delta_x^2(u_{i+1}) &= \frac{1}{h^2} (a_2 u_i - b_2 u_{i+1} + c_2 u_{i+2}) + \frac{\epsilon^+}{h^2} \left(e_2 \cdot [u] + f_2 \cdot \left[\epsilon \frac{\partial u}{\partial x} \right] \right)\end{aligned}\quad (5.4)$$

Where,

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

Here the second 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 keeping it tridiagonal 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 the two dimensional PB model we need to evaluate δ_{xx} and δ_{yy} , where the similar derivation of the 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,

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

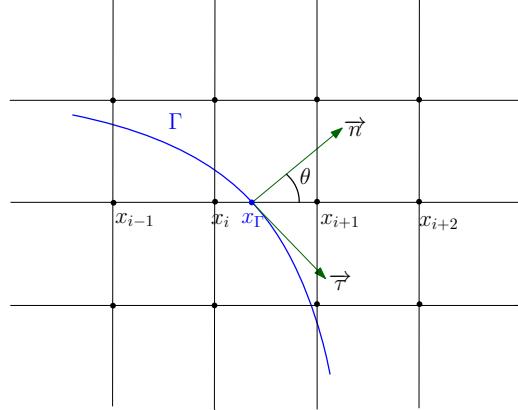


Figure 5.3: 2D GFM

where θ is the angle between the normal direction and positive x direction, see Figure 5.3. Then equations in 5.6 can be solved for $[\epsilon \frac{\partial u}{\partial x}]$ and $[\epsilon \frac{\partial u}{\partial y}]$ as,

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

To simplify the jump-conditions Liu, Fedkiw and Kang [27] smeared out the tangential derivative considering $[\epsilon \frac{\partial u}{\partial \tau}] = 0$ to get the following equations,

$$\begin{aligned} \left[\epsilon \frac{\partial u}{\partial x} \right] &\approx \cos \theta \left[\epsilon \frac{\partial u}{\partial n} \right] \\ \left[\epsilon \frac{\partial u}{\partial y} \right] &\approx \sin \theta \left[\epsilon \frac{\partial u}{\partial n} \right] \end{aligned} \quad (5.8)$$

Which are in general not true but $[\epsilon \frac{\partial u}{\partial x}]$ and $[\epsilon \frac{\partial u}{\partial y}]$ in (5.8) satisfies (5.6) for $[\epsilon \frac{\partial u}{\partial n}]$. It allows us to replace the unknown quantities as $[\epsilon \frac{\partial u}{\partial x}]$ and $[\epsilon \frac{\partial u}{\partial y}]$ by the known quantities $[\epsilon \frac{\partial u}{\partial n}]$ and $[\epsilon \frac{\partial u}{\partial \tau}]$. This process used to simplify the jump condition in [27] 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 $[\epsilon \frac{\partial u}{\partial x}]$ 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 $[\epsilon \frac{\partial u}{\partial n}] = \epsilon^- \frac{\partial G}{\partial n}$ and $[\frac{\partial u}{\partial \tau}] = \frac{\partial G}{\partial \tau}$.

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

Similarly it can be shown that $\left[\epsilon \frac{\partial u}{\partial y} \right] \approx \epsilon^- \frac{\partial G}{\partial y}$. Like the original GFM, our new GFM also omits some tangential information. Thus, two methods will have a local first order of truncation error. Nevertheless, in our new method, the normal direction of the complicated molecular surface is not required, so that the implementation of modified GFM is much simpler than the standard GFM.

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,

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

where \mathbf{P} is the transformation matrix

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

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,

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

Now along η axis,

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

Similarly along ζ axis,

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

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

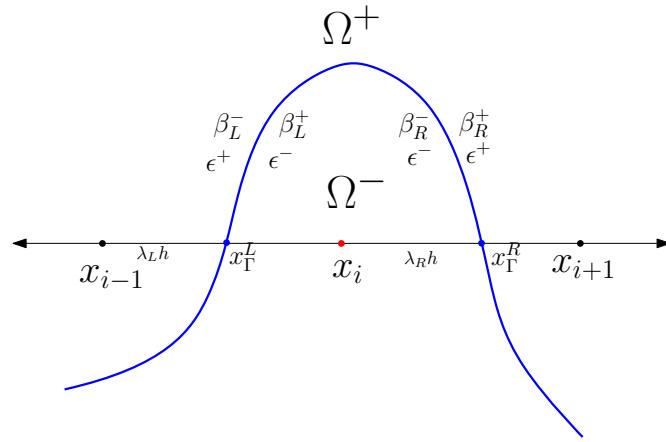
$$\begin{aligned}
\left[\epsilon \frac{\partial u}{\partial x} \right] &= \epsilon^- \left(\sin \psi \cos \theta \frac{\partial G}{\partial \xi} - \sin \theta \frac{\partial G}{\partial \eta} - \cos \psi \cos \theta \frac{\partial G}{\partial \zeta} \right) \\
&\quad - \sin \theta (\epsilon^+ - \epsilon^-)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

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 modified 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} .

$$\tilde{u}_{i-1} = A_1 u_{i-1} + B_1 u_i + C_1 G_L - D_1 \epsilon^- \frac{\partial G_L}{\partial x} \quad (5.16)$$

$$\tilde{u}_{i+1} = A_2 u_{i+1} + B_2 u_i - C_2 G_R - D_2 \epsilon^- \frac{\partial G_R}{\partial x} \quad (5.17)$$

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 Simplicity of the modified GFM

Altogether this proposed modified GFM is much more easier to program compared to the MIB method used in [40]. Unlike MIB method, modified GFM doesn't need to generate local coordinates in non-orthogonal tangential directions. We are also avoiding the tensor product decomposition of jump conditions used in mADI [40] by the approximate jump conditions proposed in equation (5.14) and (5.14). At the end we are only using the three points on the finite difference stencil and it's not necessary to consider any other auxiliary point to calculate any approximation of tangential derivatives like mADI does.

We have also simplified the step to get the axial direction jump conditions from the normal direction jump conditions. For the original GFM method in [27], Liu, Fedkiw and Kang needed the angle θ between the normal direction and the axial directions for this process which is not required for the modified GFM method. This reduces the required amount of information about the molecular surface.

CHAPTER 6

NUMERICAL VALIDATION

To validate our proposed algorithm we are providing numerical simulations in this chapter. At first we have solved the nonlinear PBE for the 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 PBE and solved it to compare with the analytical result.

6.1 Krikwood Sphere with analytical solution

The analytical solutions of the PBE are available only for simple geometries such as spheres. We have chosen the benchmark problem known as the Krikwood sphere, where a sphere has a charge at its center. This problem has the following analytical solution (6.1) with the source term (6.2) as described in [12].

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

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

where $\varepsilon = \epsilon^{+}/\epsilon^{-}$ and $R = 2$ is the radius of the sphere and $\bar{\kappa} = 1$. We have a unit charge $1e_c$ located at the center of the sphere at the co-ordinate $(0, 0, 0)$. Altogether this sphere is comparable to a protein molecule located inside our computational domain set by the boundaries as $x, y, z = [-7, 7]$. Here the dielectric constants are set as $\epsilon^{+} = 80$ outside the sphere and $\epsilon^{-} = 1$ inside the sphere. The ionic strength has been used as $I_s = 0.01$ for all the tests discussed in this section for the Krikwood sphere problem.

It can be shown that this analytical solution in (6.1) with the source term defined in (6.2) will satisfy the nonlinear PBE in (3.1) together with the jump condition defined in (3.4). Both of the major challenges like the singularity and the non-smoothness are present in this benchmark problem. For an example the singularity is generating from the source term in (6.2) for $\|\mathbf{r}\| > R$. Also the non-smoothness is present due to the jump condition (3.4). Both of these difficulties gives similar kind of challenge present in the nonlinear PBE 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.

$$L_\infty = \max_{i,j,k} |\phi_{\text{exact}} - \phi_{\text{num}}|, L_2 = \sqrt{\frac{\sum_{i,j,k} |\phi_{\text{exact}} - \phi_{\text{num}}|^2}{N}} \quad (6.3)$$

Where ϕ_{true} is the analytical solution and ϕ_{num} is the numerical solution representing the electrostatic potential for the nonlinear PBE. 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: One way to test the stability of the numerical methods to solve the PBE is, to run the time loops for a lot of iterations to check if it diverges or blows-up. In this study we chose to run the whole process for 10^4 times for different choices of time step size Δt for each one of our proposed methods along with the previous ADI method [12]. Our goal here is to check if these numerical methods are unconditionally stable or not for different choices of Δt and if its conditionally stable then what is the best choice for Δt . We focused on the variation of only Δt and used finer grids such as $h = 0.5$ and $h = 0.25$ to avoid the difficulty due to the larger grid spacing. We have found all three of our methods to be stable for $\Delta t \in [0.001, 5]$ for this benchmark problem. To illustrate this we considered the sampling for Δt such as $\Delta t \in t_{\text{set}} = \{0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$ and the stopping time $T_{\text{end}} = 10^4 * \Delta t$, so that enough accumulations of numerical errors are experienced.

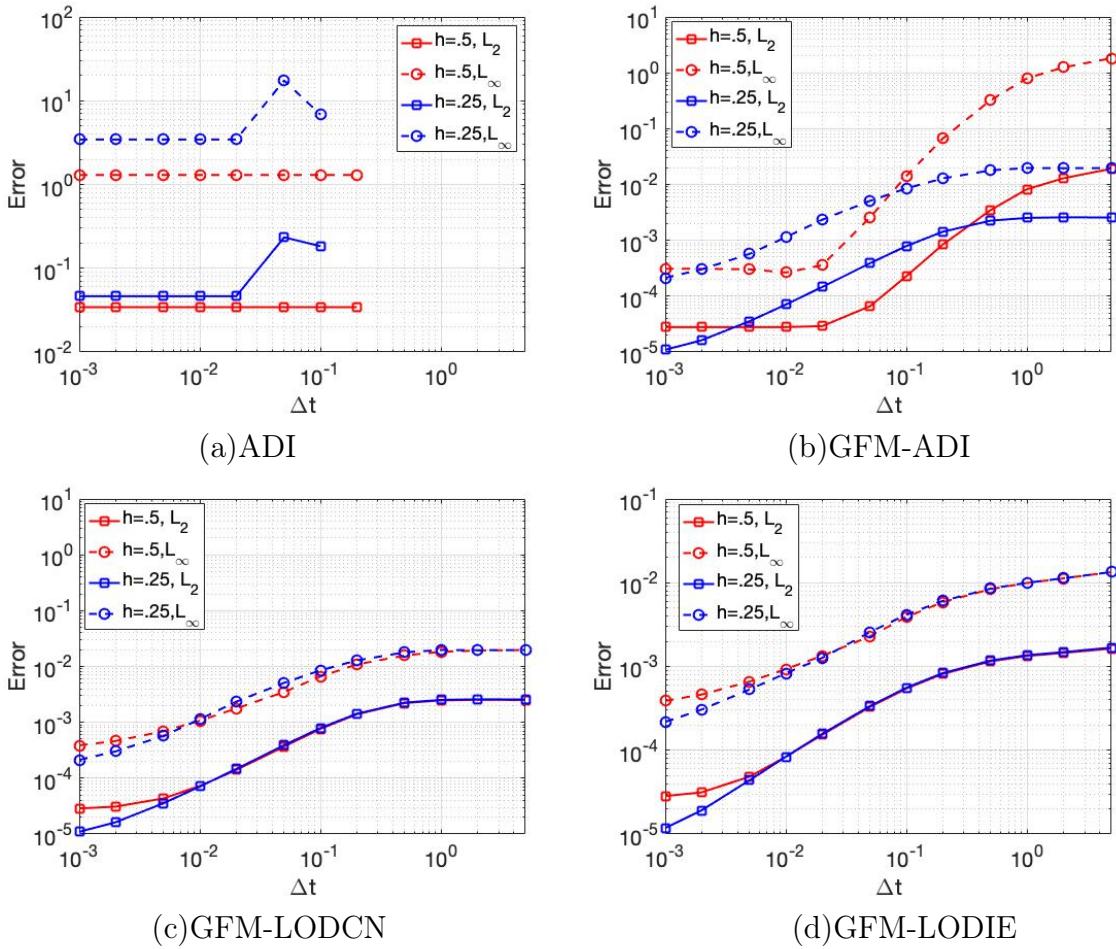


Figure 6.1: Stability test for the Krikwood Sphere with $T_{end} = 10^4 * \Delta t$.

We observed in Figure 6.1 (b), (c) and (d) that both L_2 and L_∞ error to be finite for all $\Delta t \in t_{set}$. 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. In comparison the ADI method [12] in Figure 6.1 (a), blows-up for $\Delta t > 0.2$ in both cases. Also both of the error line being almost flat (constant) for $\Delta t < 0.02$ in Figure 6.1 (a) indicates that, the L_2 and L_∞ error have been polluted and dominated by the singularities at the center of the sphere and not reducing when Δt becomes smaller like the other methods do. This observation again emphasizes the importance of regularization to avoid the singularity of the PBE.

Spatial Convergence: In this study, we investigated the order of accuracy for

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 $\Delta t = 0.001$ and $T_{\text{end}} = 10$. Comparable analytical value of the Solvation Energy is $E_{\text{sol}} = -81.97820845$.

the spacial convergence in terms of L_2 and L_∞ error using equation (6.3) in Table 6.1 for the Krikwood Sphere. The time step size 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. In this experiment, the stopping time has been set to $T_{\text{end}} = 10$ and found to be long enough to produce acceptable results. We have also computed the solvation energy E_{sol} for the nonlinear PBE in (3.1) with the source term defined in (3.3). 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. In comparison the accuracy of the ADI method decreases instead of increasing as the grid spacing becomes finer. The major reason behind this is the singularity at the center being one of the grid points. The actual difference between the numerical solution and the analytical solution is ∞ since the term $\frac{1}{||\mathbf{r}||}$ in the analytical solution becomes ∞ at the center of the sphere. So we avoided this point to calculate our L_2 and L_∞ errors showed in Table 6.1. But still, it effects the neighboring points error and ultimately dominates the errors in all other points to keep L_2 an L_∞ errors larger.

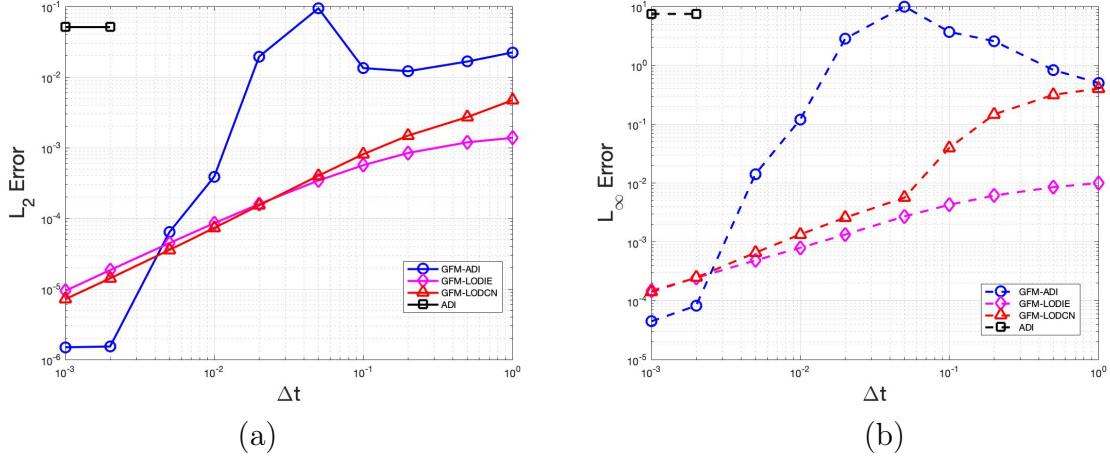


Figure 6.2: Temporal Convergence test for the krikwood sphere with $h = 0.125$ and $T_{end} = 10$.

Temporal Convergence: Finally we investigate the temporal convergence of all three of our proposed methods for the Krikwood Sphere in Figure 6.2 along with the ADI method. Similar to the spatial convergence test we have chosen the fixed stopping time to be $T_{end} = 10$ and $\Delta t \in \{0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1\}$. As we have observed in Figure 6.2, GFM-ADI is the most accurate method for the smaller Δt but as Δt increases GFM-LODCN and GFM-LODIE performs better than GFM-ADI in terms of both L_2 and L_∞ error. For the larger Δt GFM-LODIE method remains the most accurate. In comparison the ADI method represented by the black lines in Figure 6.2 blows-up just after $\Delta t = 0.002$. This indicates that, we can choose larger time step

sizes Δt for the newly proposed methods compared to ADI method which will eventually require less time to reach the steady state solution.

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 (Kirkwood sphere) with $\Delta t = [0.001, 5]$, 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 [38]. It is found in the seeds of *Crambe abyssinica* and have local anesthetic activity in a lobster walking leg axon (J. Marquis [38]). We used the MSMS package to generate the molecular surface for this protein using the crystallographic data recorded at 130K as reported in [38]. 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 (4.36) has been discretized further to calculate the solvation energy using equation (4.36) 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 T_{end} 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 $I_s = 0.15$.

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: Stability test by the Solvation Energy (*kcal/mol*) of 1cbn for $h = 0.5$, $T_{end} = 10^4 * \Delta t$.

Stability Analysis: 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 (≈ -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 loses 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.

Altogether 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 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.

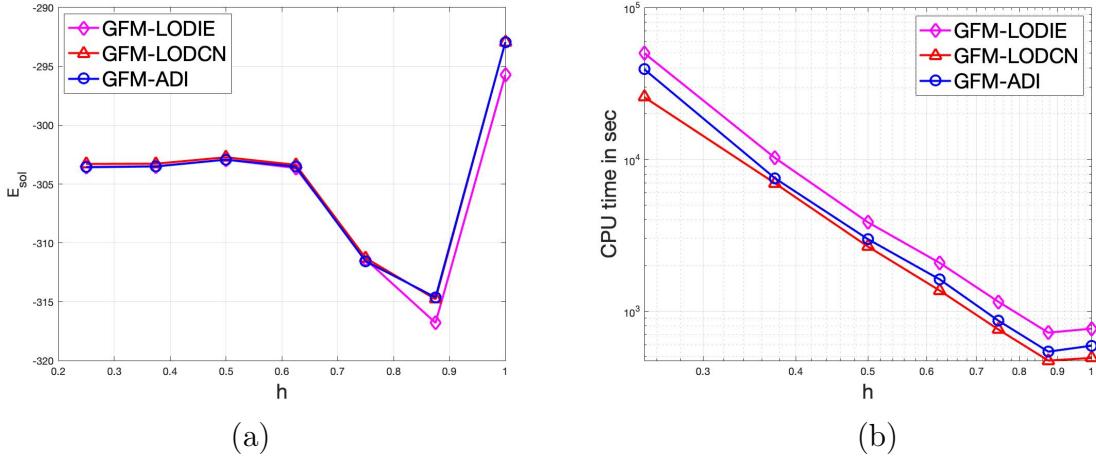


Figure 6.3: Spatial convergence for *1cbn* with $\Delta t = 0.001$ and $T_{\text{end}} = 10$.

Spatial Convergence: In Figure 6.3 we have investigated how the change of grid spacing h effects the solvation energy of real proteins like *1cbn*. We kept the time increment small and fixed to $\Delta t = 0.001$ with the stopping time $T_{\text{end}} = 10$. From Figure 6.3 (b) it is clear that the required CPU time increases dramatically when the grid spacing h decreases. So we need to find the value of h as large as possible without losing too much accuracy. We have also observed in Figure 6.3 (a) that for $h < 0.6$ the solvation energy doesn't change that much indicating that $h = 0.5$ might be the optimum choice for the solvation energy of proteins.

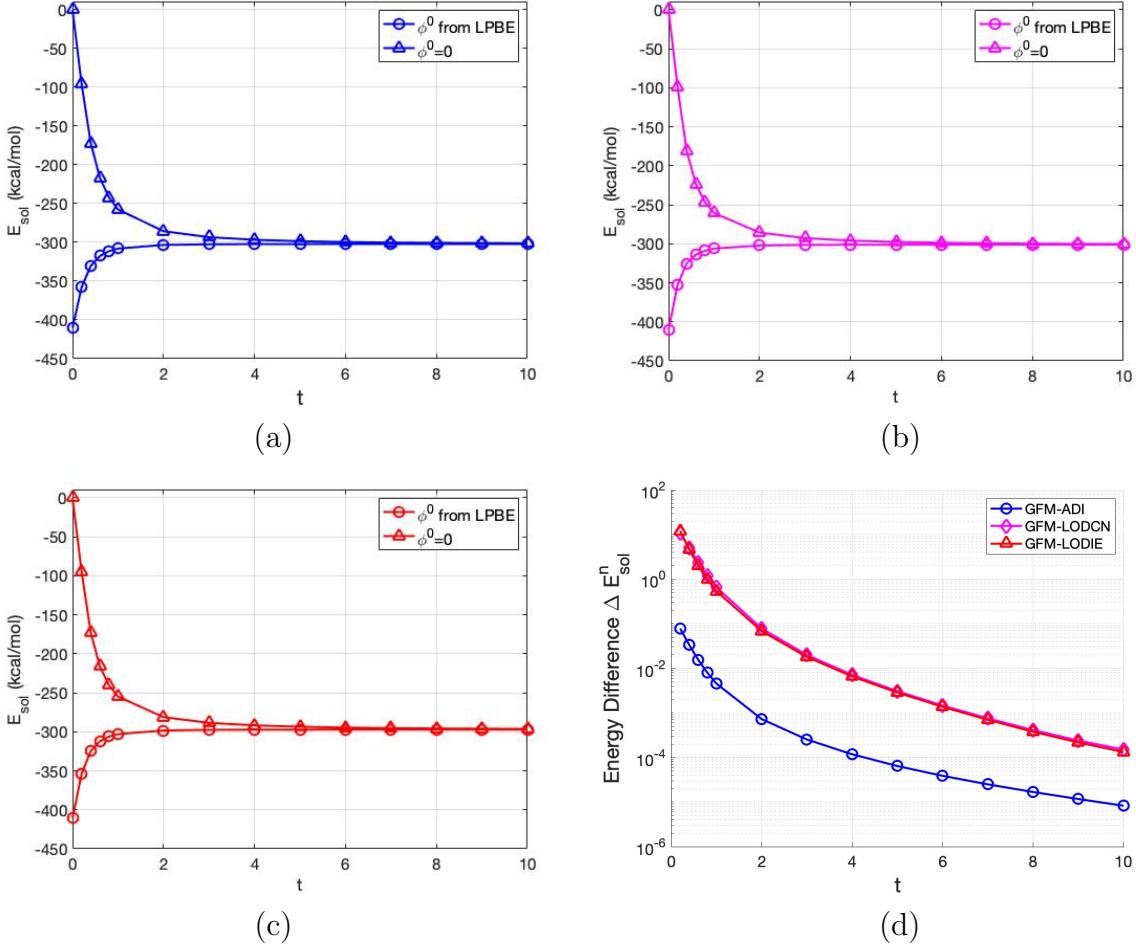


Figure 6.4: Steady State Analysis by the Solvation Energy (*kcal/mol*) of $1cbn$ with $h = 0.5$, $\Delta t = 0.05$ as $t \rightarrow \infty$. (a) GFM-ADI, (b) GFM-LODCN, (c) GFM-LODIE, (d) Solvation Energy difference $\Delta E_{sol}^n(\phi^n)$.

Steady State as $t \rightarrow \infty$: The pseudo transient methods proposed in this dissertation usually approaches to a steady state for $t \rightarrow \infty$. But with a nonlinear operator there is always a question that the steady state solution produced by the pseudo-transient method is unique or not. Hence, we have tried two different type of initial condition to set ϕ^0 at $t = 0$ to check if the proposed methods reach to the same steady state solution. For the fist type of the initial condition we just set $\phi^0 = 0$. Then for the second type we solve a linearized PBE (LPBE) [48] and use the solution to initialize ϕ^n as $\phi^0 =$ "the solution of the LPBE". We fixed grid spacing $h = 0.5$ and $\Delta t = 0.05$ and run the iterations until $t = 10$. As we can see in Figure 6.4 (a), (b) and (c), all of our pro-

posed methods reach to the same steady state solution for both type of initial condition demonstrating its uniqueness. It is also noticeable in Figure 6.4 (a), (b) and (c), that initializing ϕ^n from LPBE is better than initializing ϕ^n as zero since for LPBE solution as the initial condition, ϕ^n reaches to the steady state earlier. In Figure 6.4(d) we reported the calculated solvation energy difference ΔE_{sol}^n at the n-th time step defined as,

$$\Delta E_{sol}^n(\phi^n) = |E_{sol}^n(\phi^n) - E_{sol}^{n-1}(\phi^{n-1})| \quad (6.4)$$

As we observe in Figure 6.4(b), this energy difference ΔE_{sol}^n is reducing over time and it can be another tool to identify the steady state set the stopping time T_{end} as we have used in Table 6.4.

X	Y	Z	E_{sol}
[-10,32]	[-9,29]	[-15,28]	-302.0594
[-13,35]	[-12,32]	[-18,31]	-302.0594
[-16,38]	[-14,35]	[-20,34]	-302.0626
[-19,41]	[-17,38]	[-23,36]	-302.0656

Table 6.3: Impact of domain size to the solvation energy of 1cbn calculated by the GFM-ADI method.

Finally we investigate the effect of the size of the computational domain on the calculation of the solvation energy. As we have observed in Table 6.3 that changing the length of the cuboidal domain in x, y and z direction up to 18Å doesn't change the solvation energy for 1cbn significantly. So we set the computational domain as small as possible keeping the whole molecule inside plus a fixed distance from furthest atom from the coordinate origin.

Electrostatic potential on the molecular surface: In Figure 6.5 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.

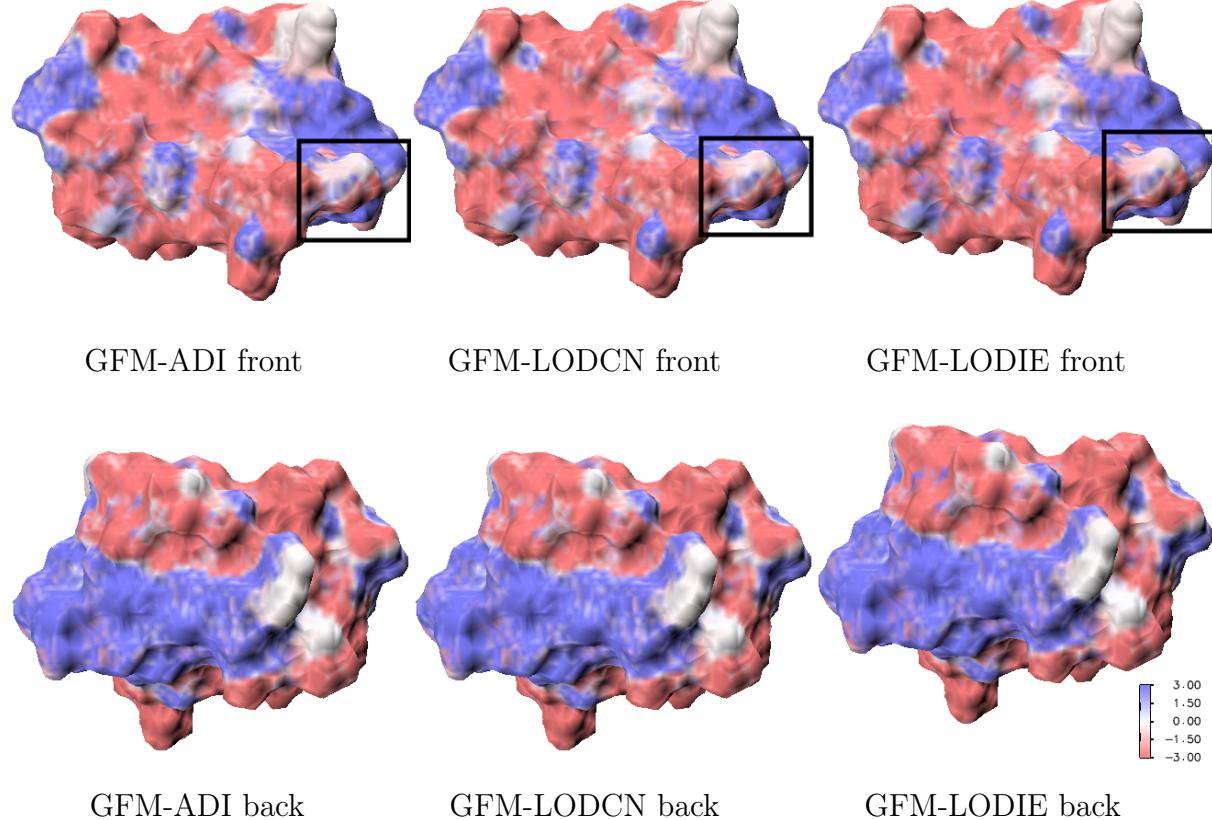


Figure 6.5: Electrostatic potential for 1cbn using $\Delta t = 0.05$ and $h = 0.5$.

6.2.2 Solvation energy of 24 proteins

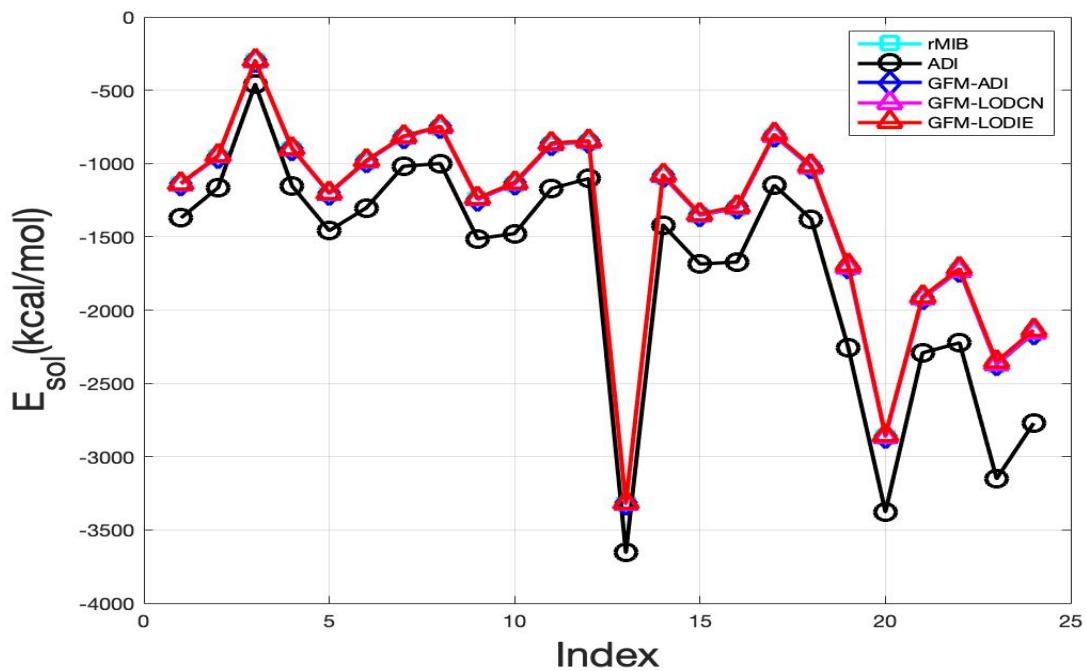
Next we solved the nonlinear PBE and computed solvation energy for a collection of 24 proteins as in [11, 13]. The dielectric constants, Ionic strength are same as our study for the protein Crambin(1cbn) as $\epsilon^- = 1$, $\epsilon^+ = 80$ and $I_s = 0.15$.

Solvation energies for all three of our proposed schemes have been compared with the rMIB and MIB schemes in Table 6.4 and Figure 6.6(a). The results from our proposed schemes have been observed to be very close to the results from rMIB and MIB schemes

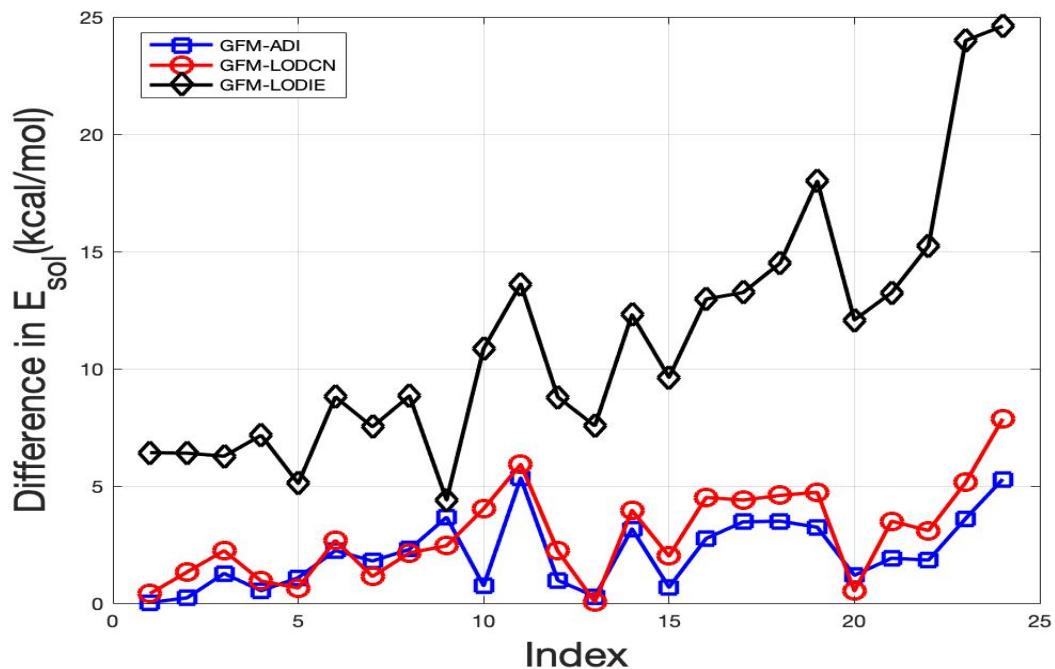
PDB	No. of Atoms	rMIB	ADI	GFM-ADI	GFM-LODCN	GFM-LODIE
1ajj	519	-1139.48	-1371.10	-1139.45	-1139.07	-1133.06
2erl	573	-952.36	-1165.28	-952.13	-951.03	-945.96
1cbn	648	-303.33	-459.51	-302.06	-301.09	-297.06
1vii	596	-902.31	-1154.67	-901.78	-901.36	-895.15
1fca	729	-1204.44	-1458.16	-1205.53	-1205.09	-1199.33
1bbl	576	-988.40	-1302.49	-986.15	-985.72	-979.60
2pd ^e	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
1fd	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 6.4: 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. Stopping condition is set to be either $T_{end} = 50$ or $\Delta E_{sol}^n < 10^{-4}$.

while solving nonlinear PBE instead of linear PBE. As we have identified in previous sections 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 6.4 also confirms that if GFM-ADI fails to converge for any protein then GFM-LODCN or GFM-LODIE can be used due to the fact that they are more stable and the results are not that much different than GFM-ADI.



(a)



(b)

Figure 6.6: (a) Solvation energy of 24 proteins. (b) Difference in Solvation energy compared to rMIB method

It is noticeable in Figure 6.6 that the solvation energies for GFM-ADI, GFM-LODCN and GFM-LODIE are very close to each other. But the solvation energies from ADI Methods are always 200 to 700 *kcal/mol* lower than other proposed methods. This is an indicator that a major systematic difficulty has been avoided to improve the accuracy in these newly proposed methods in this dissertation. We have also reported the difference in calculated solvation energy for our proposed methods with the rMIB method in Figure 6.6. Figure 6.7 reports the required time for these methods highlighting the fact that ADI method takes around 10 times more CPU time compared to the newly proposed methods. A major reason behind that is the necessity of Δt to be as small as 0.001 for ADI method to keep it stable and accurate enough for all 24 proteins tested here. For the newly proposed methods we were able to use $\Delta t = 0.05$ requiring less time to produce solvation energies more accurate than ADI method.

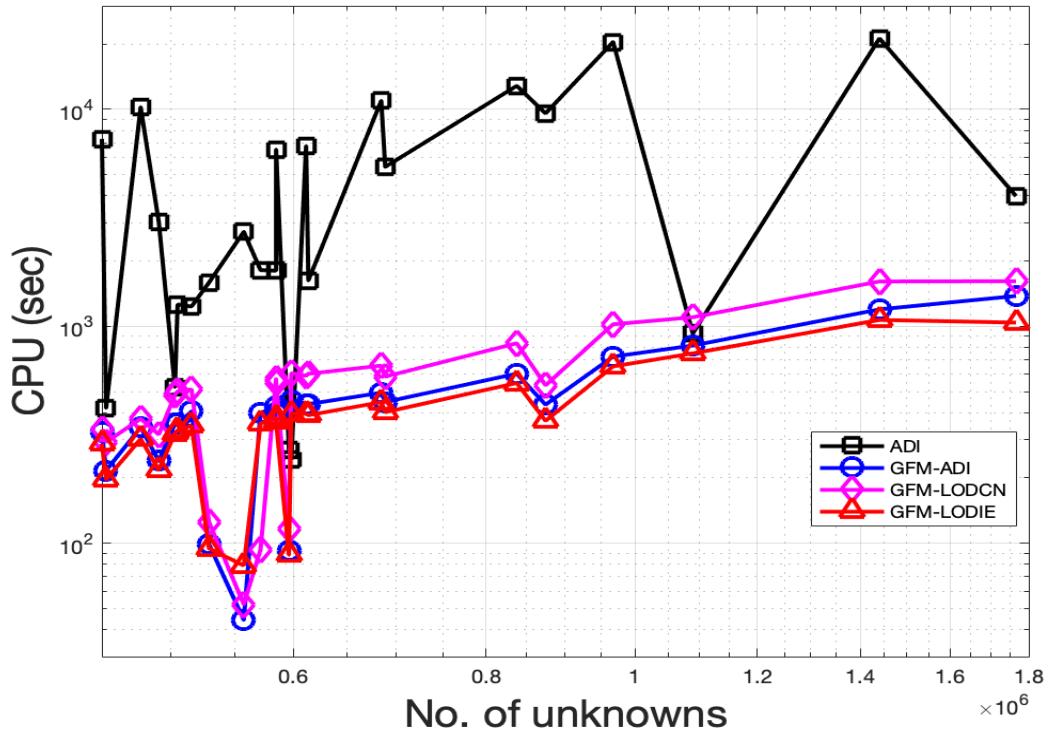


Figure 6.7: CPU time for the Solvation Energy Calculation of 24 proteins

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 [22]. 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.

h	$E_{\text{sol}}^{\text{complex}}$	$E_{\text{sol}}^{\text{protein}}$	$E_{\text{sol}}^{\text{RNA}}$	$E_{\text{bind}}^{\text{complex}}$
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 6.5: Binding energy of 2a9x

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

$$E_{\text{bind}}^{\text{AB}} = \Delta G_{\text{ele}}^{\text{AB}} - \Delta G_{\text{ele}}^{\text{A}} - \Delta G_{\text{ele}}^{\text{B}} = [E_{\text{sol}}^{\text{AB}} + E_{\text{cou}}^{\text{AB}}] - [E_{\text{sol}}^{\text{A}} + E_{\text{cou}}^{\text{A}}] - [E_{\text{sol}}^{\text{B}} + E_{\text{cou}}^{\text{B}}] \quad (6.5)$$

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 (4.33).

In Table 6.5 we have found the bidding energies calculated by GFM-ADI to be very close to the results from rMIB method reported in [13]. This shows the ability of

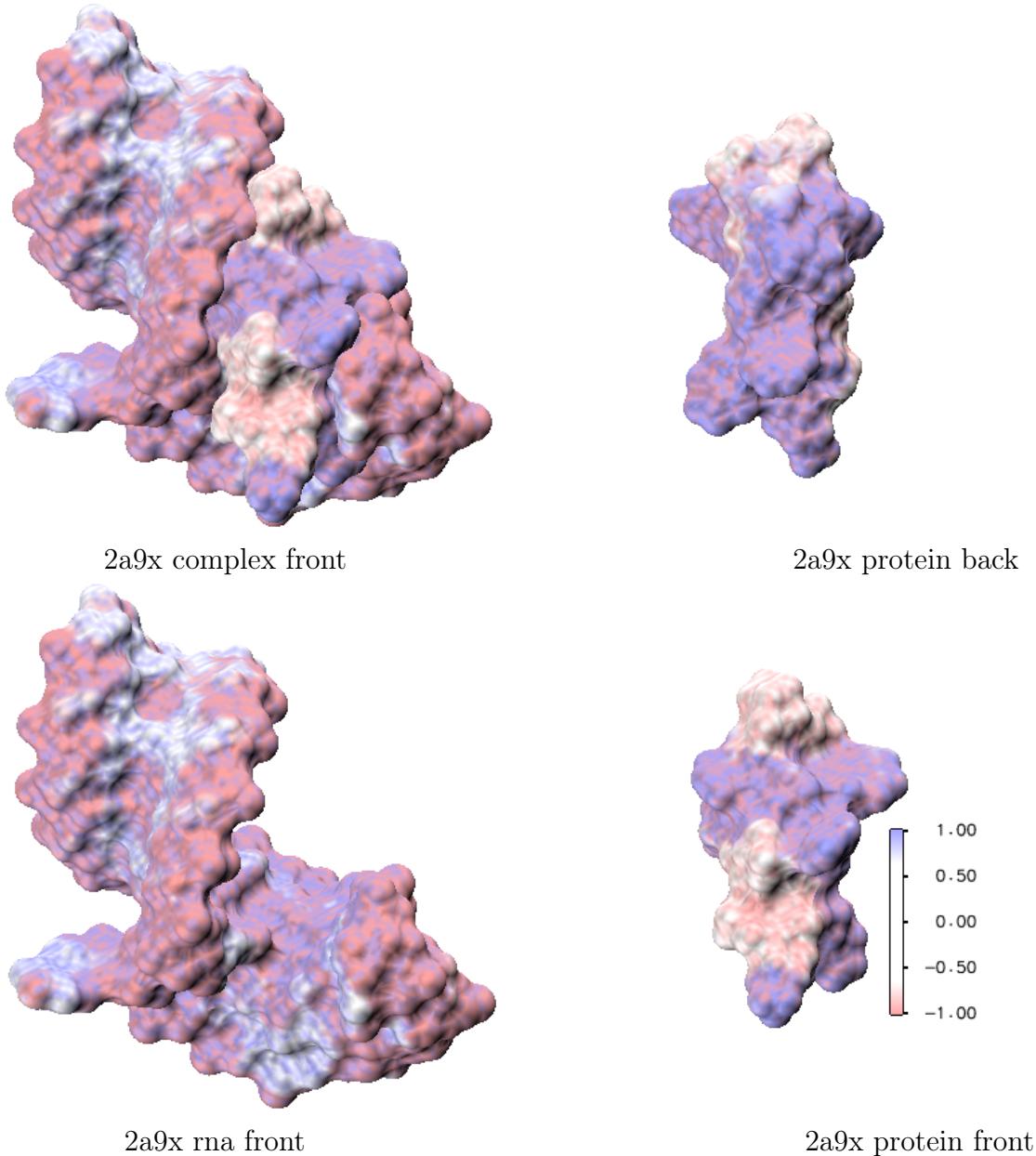


Figure 6.8: Electrostatic potential for 2a9x using GFM-ADI for $\Delta t = 0.05$ and $h = 0.5$

GFM-ADI method to calculates other type of energies for large protein-ligand complexes like 2a9x. Later in Figure 6.8 the electrostatic potential of 2a9x and its monomers have been visualized on their surface in a color scale ranging from -1 to 1 . These type of visualizations help to provide optimized parameters for the molecular mechanics calculations of the drug candidate that will influence binding affinity [23].

6.2.4 Salt effect on the binding affinity

The nonlinear PBE 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 performance of the proposed GFM-ADI scheme for the evaluation of the salt effect in the protein-protein binding of the complex Lactoglobulin dimer(A-B) (PDB ID 1beb) and E9Dnase-Im9(10)(B-A)(PDB ID 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.

$$\begin{aligned}\Delta E_{\text{bind}}(I_s) &= E_{\text{bind}}(I_s) - E_{\text{bind}}(0) \\ &= [E_{\text{sol}}^{\text{AB}}(I_s) - E_{\text{sol}}^{\text{AB}}(0)] - [E_{\text{sol}}^{\text{A}}(I_s) - E_{\text{sol}}^{\text{A}}(0)] - [E_{\text{sol}}^{\text{B}}(I_s) - E_{\text{sol}}^{\text{B}}(0)] \quad (6.6)\end{aligned}$$

PDB	Charges			Slope ratios		
	AB	A	B	Experimental	GFM-ADI	LFLPB
1beb	+26	+13	+13	-1.62	-1.82	-2.02
1emv	-3	-8	+5	2.17	0.52	2.4
1brs	-4	+2	-6	0.96	0.09	0.67

Table 6.6: Comparison of binding affinities of the protein complexes

For this study we have used the same model parameters used earlier for 24 proteins. In Figure 6.9 we reported the calculated binding free energy with the experimental

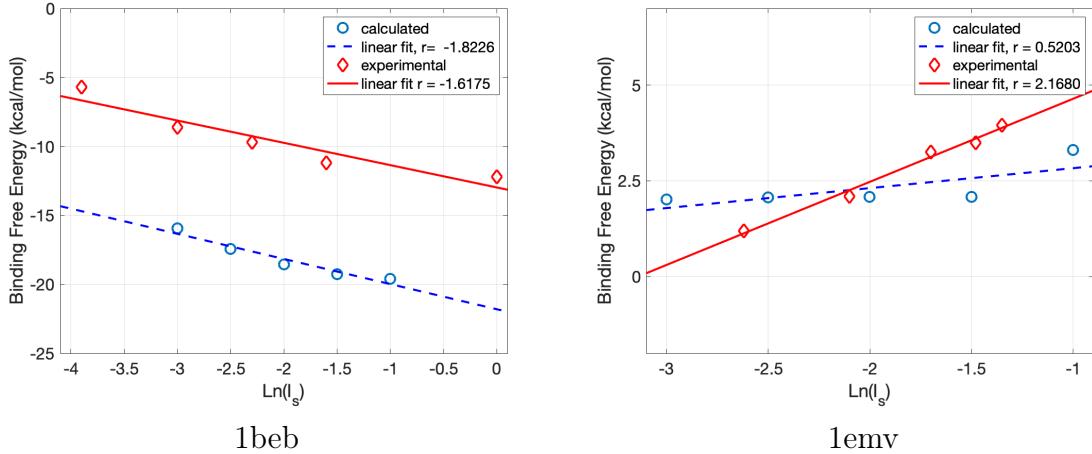


Figure 6.9: The salt dependence of the binding affinities

results. The slope ratio or the binding affinity is calculated and reported in Table (6.6) as in [48]. The results attained by the Lagrangian formulation linearized PB (LFLPB) model [4] are also given in Table (6.6) for comparison. For 1beb the binding affinity calculated by GFM-ADI scheme sufficiently close to experimental data and better than LFLPB. For 1emv and 1brs 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 and the error in the mathematical modeling, i.e., the nonlinear Poisson Boltzmann electrostatic analysis is not enough for the calculation of the binding affinities. Other biophysical models have to be used in order to produce a better slope estimation.

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 [2, 11, 45, 46, 51–53] 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 (4.23) (4.24) and (4.25) 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 [27] 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 [12]. 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 [41]. 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 [41] 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 [10].

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] D. Chen, Z. Chen, C. Chen, W. Geng, and G. W. Wei. MIBPB: A software package for electrostatic analysis. *Journal of Computational Chemistry*, 32(4):756–770, 2011.
- [3] 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.
- [4] Z. Chen, N. Baker, and G. Wei. Differential geometry based solvation model ii: Lagrangian formulation. *Journal of mathematical biology*, 63(6):1139–1200, 2011.
- [5] M. L. Connolly. Analytical molecular surface calculation. *Journal of Applied Crystallography*, 16(5):548–558, Oct 1983.
- [6] W. Deng, J. Xu, and S. Zhao. On developing stable finite element methods for pseudo-time simulation of biomolecular electrostatics. *Journal of Computational and Applied Mathematics*, 330:456 – 474, 2018.
- [7] E. D’Yakonov. Difference schemes with split operators for multidimensional unsteady problems (english translation). *USSR Comp. Math.*, 3:518–607, 1963.
- [8] 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.
- [9] 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.
- [10] W. Geng and G. Wei. Multiscale molecular dynamics using the matched interface and boundary method. *Journal of Computational Physics*, 230(2):435 – 457, 2011.
- [11] W. Geng, S. Yu, and G. Wei. Treatment of charge singularities in implicit solvent models. *The Journal of Chemical Physics*, 127(11):114106, 2007.
- [12] W. Geng and S. Zhao. Fully implicit ADI schemes for solving the nonlinear Poisson-Boltzmann equation. *Molecular Based Mathematical Biology*, 1:109–123, 2013.
- [13] 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.
- [14] 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.

- [15] T. Hazra, S. Ahmed Ullah, S. Wang, E. Alexov, and S. Zhao. A super-gaussian poisson–boltzmann model for electrostatic free energy calculation: smooth dielectric distribution for protein cavities and in both water and vacuum states. *Journal of Mathematical Biology*, Apr 2019.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] B. Lee and F. Richards. The interpretation of protein structures: Estimation of static accessibility. *Journal of Molecular Biology*, 55(3):379 – IN4, 1971.
- [22] 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.
- [23] R. A. Lewis. *Chapter 4 The Development of Molecular Modelling Programs: The Use and Limitations of Physical Models*. The Royal Society of Chemistry, 2012.
- [24] C. Li and S. Zhao. A matched Peaceman–Rachford ADI method for solving parabolic interface problems. *Applied Mathematics and Computation*, 299:28–44, 2017.
- [25] 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.
- [26] 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.

- [27] 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.
- [28] 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.
- [29] 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.
- [30] C. Quan. *Mathematical methods for implicit solvation models in quantum chemistry*. PhD thesis, 2018.
- [31] F. M. Richards. Areas, volumes, packing, and protein structure. *Annual Review of Biophysics and Bioengineering*, 6(1):151–176, 1977. PMID: 326146.
- [32] 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.
- [33] M. F. Sanner, A. J. Olson, and J.-C. Spehner. Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320.
- [34] 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.
- [35] 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.
- [36] 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, 2002.
- [37] J. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. SIAM, 2004.
- [38] M. M. Teeter, S. Roe, and N. H. Heo. Atomic resolution (0.83 \AA) crystal structure of the hydrophobic protein crambin at 130 k. *Journal of Molecular Biology*, 230(1):292 – 311, 1993.
- [39] J. Tomasi, B. Mennucci, and R. Cammi. Quantum mechanical continuum solvation models. *Chemical reviews*, 105 8:2999–3093, 2005.

- [40] Z. Wei, C. Li, and S. Zhao. A spatially second order alternating direction implicit (ADI) method for solving three dimensional parabolic interface problems. *Computers and Mathematics with Applications*, 2017.
- [41] 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.
- [42] 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.
- [43] 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.
- [44] N. Yanenko. *The method of fractional steps (English translation; originally published in Russian)*. Springer-Verlag, Berlin, Heidelberg, and New York, 1967.
- [45] S. Yu, W. Geng, and G. W. Wei. Treatment of geometric singularities in implicit solvent models. *The Journal of Chemical Physics*, 126(24):244108, 2007.
- [46] S. Yu and G. Wei. Three-dimensional matched interface and boundary (mib) method for treating geometric singularities. *Journal of Computational Physics*, 227(1):602 – 632, 2007.
- [47] 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.
- [48] 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.
- [49] S. Zhao. Operator splitting adi schemes for pseudo-time coupled nonlinear solvation simulations. *Journal of Computational Physics*, 257:1000 – 1021, 2014.
- [50] S. Zhao. A Matched Alternating Direction Implicit (ADI) Method for Solving the Heat Equation with Interfaces. *Journal of Scientific Computing*, 2015.
- [51] S. Zhao and G. Wei. High-order fdtd methods via derivative matching for maxwell’s equations with material interfaces. *Journal of Computational Physics*, 200(1):60 – 103, 2004.
- [52] Y. Zhou and G. Wei. On the fictitious-domain and interpolation formulations of the matched interface and boundary (mib) method. *Journal of Computational Physics*, 219(1):228 – 246, 2006.

- [53] Y. Zhou, S. Zhao, M. Feig, and G. Wei. High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. *Journal of Computational Physics*, 213(1):1 – 30, 2006.
- [54] 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.