STRUCTURE PRESERVING AND FAST SPECTRAL METHODS FOR KINETIC EQUATIONS

by

Xiaodong Huang

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Department of Mathematics

West Lafayette, Indiana

December 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Jingwei Hu, Chair

Department of Mathematics

Dr. Haizhao Yang, Co-chair

Department of Mathematics

Dr. Jie Shen

Department of Mathematics

Dr. Cory Hauck

Oak Ridge National Laboratory

Approved by:

Dr. Plamen D.Stefanov

To my parents.

ACKNOWLEDGMENTS

Throughout the writing of this dissertation I have received a great deal of support and assistance.

My deepest gratitude is to my supervisor, Professor Jingwei Hu. Without her guidance and encouragement, I would not be able to overcome the problems and difficulties in my research. She has taught me how to ask questions, how to solve problems, and how to think as an applied mathematician. Her thoughts have deeply influenced me in the past four years.

I would like to acknowledge my colleagues from my internship at Oak Ridge National Lab for their wonderful collaboration. I would particularly like to my supervisor at Oak Ridge National Lab, Professor Cory Hauck. Thank for his support and for all of the opportunities I was given to further my research.

I am grateful to the tremendous support and encouragement I received from other professors at Purdue University, especially from Professor Haizhao Yang and Professor Jie Shen.

Last but not least, I would like to thank my parents for their wise counsel and sympathetic ear. Without their support, my achievements would not be possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This dissertation consists of three research projects of kinetic models: a structure-preserving scheme for Poisson-Nernst-Planck equations and two efficient spectral methods for multi-dimensional Boltzmann equation.

The Poisson-Nernst-Planck (PNP) equations is widely used to describe the dynamics of ion transport in ion channels. We introduce a structure-preserving semi-implicit finite difference scheme for the PNP equations in a bounded domain. A general boundary condition for the Poisson equation is considered. The fully discrete scheme is shown to satisfy the following properties: mass conservation, unconditional positivity, and energy dissipation (hence preserving the steady-state).

Numerical approximation of the Boltzmann equation presents a challenging problem due to its high-dimensional, nonlinear, and nonlocal collision operator. Among the deterministic methods, the Fourier-Galerkin spectral method stands out for its relative high accuracy and possibility of being accelerated by the fast Fourier transform. In this dissertation, we studied the state of the art in the fast Fourier method and discussed its limitation. Next, we proposed a new approach to implement the Fourier method, which can resolve those issues.

However, the Fourier method requires a domain truncation which is unphysical since the collision operator is defined in whole space $\mathbb{R}^d$. In the last part of this dissertation, we introduce a Petrov-Galerkin spectral method for the Boltzmann equation in the unbounded domain. The basis functions (both test and trial functions) are carefully chosen mapped Chebyshev functions to obtain desired convergence and conservation properties. Furthermore, thanks to the close relationship of the Chebyshev functions and the Fourier cosine series, we can construct a fast algorithm with the help of the non-uniform fast Fourier transform (NUFFT).

1. INTRODUCTION

1.1 Overview

In multi-scale modeling, kinetic theory serves as a basic building block that bridges microscopic particle models and macroscopic continuum models. By tracking the probability density function, kinetic equations describe the non-equilibrium dynamics of the complex particle systems and have been widely used in disparate fields such as rarefied gas dynamics [1], plasma physics [2], nuclear reactor modeling [3], chemistry [4], biology, and socioeconomics [5].

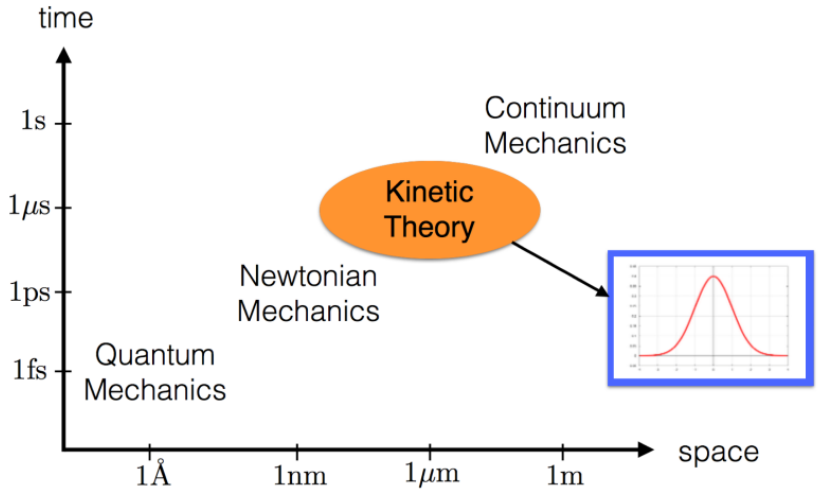

Figure 1.1. Role of kinetic theory in multiscale modeling hierarchy.

As presented in Fig 1.1, the kinetic models lie between the macroscale models and the quantum models. In kinetic models, the gases would be studied as a larger number of particles following some ideal assumptions of the interactions between them. The probability density function is used to describe the thermodynamics behavior of the system that we are interested in. In many situations, this allows us to think about the physical phenomenon with more detailed information that macroscopic models cannot capture. In the past century, the Boltzmann equation [6], [7] and its variants (such as BGK [8], Landau [9] and Fokker-Planck [10] models) played a key role in the development of kinetic theory. On the mathematical side, these models usually involve some complicated nonlinear operators which make the

numerical approximation to be a challenging task. For numerical schemes solving the kinetic equations, the following criteria are commonly used to verify their validity.

1. **structure preserving**: The solutions of kinetic models must follow some basic physical laws, such as positivity, mass conservative, momentum preserving, and entropy decay. It is not easy to guarantee all these properties at the discrete level.

2. **accuracy and efficiency**: Compared to the macroscale models, the numerical approximation for the kinetic model would be quite expensive especially in the multidimensional setting. For a real-world application, it is quite important to design a fast algorithm with a balance of accuracy and efficiency.

In this dissertation, we study the numerical approximation of the kinetic models. In chapter 2, we start with a coupled continuum model describing the dynamics of ion transport in membrane channels–the Poisson-Nernst-Planck equations, which is also a Fokker-Planck type system. A structure preserving semi-implicit finite difference scheme will be developed for this model. In chapter 3 and 4, we study the deterministic numerical approximation of the Boltzmann equation, one of the fundamental equations in kinetic theory. We focus on developing accurate and efficient spectral methods for solving the Boltzmann equation. The following of this chapter provides the prerequisite knowledge of Poisson-Nernst-Planck equations and Boltzmann equation, along with the main results in this dissertation.

1.2 Poisson-Nernst-Planck equations

In Poisson-Nernst-Planck (PNP) equations, the ions satisfy the Nernst-Planck equation:

$$\partial_t c^{(\mathrm{i})} = \nabla \cdot \left\{ D^{(\mathrm{i})} \left(\nabla c^{(\mathrm{i})} + \frac{z_{\mathrm{i}} \mathrm{e}}{k_B T} c^{(\mathrm{i})} \nabla \psi \right) \right\}, \quad \mathrm{i} = 1, \ldots, m, \tag{1.1}$$

where $c^{(\mathrm{i})} = c^{(\mathrm{i})}(t, \boldsymbol{x})$ is the local concentration of the i-th ion species, $D^{(\mathrm{i})} = D^{(\mathrm{i})}(\boldsymbol{x})$ is the diffusion coefficient, z_{i} is the valence of the ion, e is the unit charge of a proton, k_B is the

Boltzmann's constant, T is the absolute temperature, and $\psi = \psi(t, \boldsymbol{x})$ is the electrostatic potential related to ion concentrations via the Poisson equation:

$$-\nabla \cdot (\epsilon \nabla \psi) = \sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}} \mathrm{e}\, c^{(\mathrm{i})} + \rho, \tag{1.2}$$

where $\epsilon = \epsilon(\boldsymbol{x})$ is the permittivity of the electrolyte and $\rho = \rho(\boldsymbol{x})$ is the permanent charge density of the system. The PNP equations (1.1) and (1.2) are usually posed in a bounded domain with proper boundary and initial conditions (see Section 2.1 for details). Although termed by Eisenberg et. al [11], [12] in the 1990s to study the ion channels, the PNP equations have a long history in the broader context to describe charge transport, where they are often called drift-diffusion-Poisson equations, see for instance in semiconductor modeling [13]. For a review of recent development of more generalized PNP equations and related models, the readers are referred to [14].

For completeness, let us mention a few analytical works related to the well-posedness and long-time behavior of the PNP equations. Using a generalization of the Hopf-Cole variable transformation, the existence of a global classical solution and convergence to stationary solution was proved in [15] for a simplified 1D single species PNP model. In [16], the weak solution of a multi-D single species PNP model was studied and the well-posedness locally in time was proved. This result is improved to the two species case in [17], where the global in time existence of the solution was obtained. The long time asymptotic behavior with exponential convergence to steady states was obtained in [18], [19].

Solutions to the PNP equations satisfy a few important physical properties: mass conservation, positivity, energy dissipation, etc. When designing numerical methods, it would be desirable to maintain the same properties at the discrete level, preferably with a mild constraint on time step Δt and spatial size Δx, so that the long time simulation can be done accurately and efficiently.

Searching the literature, there have been numerous studies in recent years devoted to numerical simulation of the PNP equations. Many of them also aim to preserve the structure of the solutions. Without being exhaustive, we mention a few closely related works. Among the explicit methods, the finite difference scheme in [20] is able to preserve the positivity

under a parabolic CFL condition ($\Delta t = O(\Delta x^2)$), and the energy decay can be shown for the semi-discrete scheme (time is continuous). Later a DG version is developed in [21], where the positivity and fully discrete energy decay can be achieved still under a parabolic CFL condition. Among the implicit methods, the finite difference scheme in [22] obtains second order in time using a combination of the trapezoidal rule and backward differentiation formula. The scheme is positive, however, under a parabolic CFL condition and an additional constraint on spatial size. An energy-preserving version is recently presented in [23], where the energy decay rate is shown to be consistent up to $O(\Delta x^2+\Delta t^2)$. Finally, the finite element method in [24] employs the fully implicit backward Euler scheme to obtain the discrete energy decay. We mention that this time discretization only works for certain boundary conditions and would not work (or require extra conditions) for the general boundaries we considered in this paper (see Remark 3). From the above discussions, we can see that it is very difficult to obtain both unconditional positivity and discrete energy decay and that generally requires one to go from explicit to implicit schemes.

In chapter 2, we develop a semi-implicit finite difference scheme for the PNP equations that is first order in time and second-order in space. Our main contribution in this work is the *time discretization*, which is inspired by the recent work [25]. For generality, we consider an inhomogeneous Robin type boundary condition for the Poisson equation which includes Dirichlet and Neumann boundaries as subcases. The fully discrete scheme is proved to be mass conservative, unconditionally positive and energy dissipative. As a result of fully discrete energy decay, the numerical solution would converge to the solution of the (time independent) Poisson-Boltzmann equation, i.e., the scheme is steady-state preserving. To solve the nonlinear system resulting from the semi-implicit time discretization, we propose a simple fixed point iteration. Although we are not able to prove the convergence of the iterative scheme, we demonstrate numerically its fast convergence using a series of examples. Moreover, we provide rigorous proof of the solvability of the semi-discrete scheme (space is continuous). To the best of our knowledge, this is the first numerical method for the PNP equations that achieves simultaneously unconditional positivity and fully discrete energy decay, and works for a large class of boundary conditions.

1.3 Boltzmann equation

The complete Boltzmann equation includes both particle transport and collisions which are often treated separately by operator splitting,

$$\partial_t f + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f = Q(f, f), \quad t > 0, \quad \boldsymbol{x} \in \Omega \subset \mathbb{R}^d, \quad \boldsymbol{v} \in \mathbb{R}^d, \quad d \geq 2, \tag{1.3}$$

where $f = f(t, \boldsymbol{x}, \boldsymbol{v})$ is the probability density function of time t, location $\boldsymbol{x}$ and velocity $\boldsymbol{v}$, and $Q(f, f)$ is the collision operator.

Since the collision part is the main difficulty when numerically solving the equation, we focus on the following spatially homogeneous Boltzmann equation in multi-dimensional setting:

$$\partial_t f = Q(f, f), \quad t > 0, \quad \boldsymbol{v} \in \mathbb{R}^d, \quad d = 2, 3, \tag{1.4}$$

where $f = f(t, \boldsymbol{v})$ is the probability density function of time t and velocity $\boldsymbol{v}$, and $Q(f, f)$ is the collision operator whose bilinear form is given by

$$Q(g, f)(\boldsymbol{v}) = \int_{\mathbb{R}^d} \int_{S^{d-1}} \mathcal{B}(\boldsymbol{v}, \boldsymbol{v}_*, \boldsymbol{\sigma}) \left[g(\boldsymbol{v}'_*) f(\boldsymbol{v}') - g(\boldsymbol{v}_*) f(\boldsymbol{v}) \right] \mathrm{d}\boldsymbol{\sigma} \, \mathrm{d}\boldsymbol{v}_*, \tag{1.5}$$

where the post-collisional velocities $(\boldsymbol{v}', \boldsymbol{v}'_*)$ are defined in terms of the pre-collisional velocities $(\boldsymbol{v}, \boldsymbol{v}_*)$ as

$$\begin{cases} \boldsymbol{v}' = \frac{1}{2}(\boldsymbol{v} + \boldsymbol{v}_*) + \frac{1}{2}|\boldsymbol{v} - \boldsymbol{v}_*|\boldsymbol{\sigma}, \\ \boldsymbol{v}'_* = \frac{1}{2}(\boldsymbol{v} + \boldsymbol{v}_*) - \frac{1}{2}|\boldsymbol{v} - \boldsymbol{v}_*|\boldsymbol{\sigma}, \end{cases} \tag{1.6}$$

with $\boldsymbol{\sigma}$ being a vector over the unit sphere S^{d-1}. The collision kernel $\mathcal{B}$ takes the form

$$\mathcal{B}(\boldsymbol{v}, \boldsymbol{v}_*, \boldsymbol{\sigma}) = B(|\boldsymbol{v} - \boldsymbol{v}_*|, \cos\theta), \quad \cos\theta = \left\langle \frac{\boldsymbol{v} - \boldsymbol{v}_*}{|\boldsymbol{v} - \boldsymbol{v}_*|}, \boldsymbol{\sigma} \right\rangle, \tag{1.7}$$

i.e., it is a function depending only on the relative velocity $|\boldsymbol{v}-\boldsymbol{v}_*|$ and cosine of the scattering angle. The collision operator $Q(f,f)$ satisfies many important physical properties, including conservation of mass, momentum, and energy:

$$\int_{\mathbb{R}^d} Q(f,f)\ \mathrm{d}\boldsymbol{v} = \int_{\mathbb{R}^d} Q(f,f)\boldsymbol{v}\ \mathrm{d}\boldsymbol{v} = \int_{\mathbb{R}^d} Q(f,f)|\boldsymbol{v}|^2\ \mathrm{d}\boldsymbol{v} = 0, \tag{1.8}$$

and the Boltzmann's H-theorem:

$$\int_{\mathbb{R}^d} Q(f,f)\log f\ \mathrm{d}\boldsymbol{v} \le 0. \tag{1.9}$$

In the physically relevant case ($d=3$), the collision operator is a five-fold quadratic integral whose numerical approximation can be extremely challenging. The stochastic methods, such as the direct simulation Monte Carlo (DSMC) methods proposed by Nanbu [26] and Bird [27], have been historically popular due to their simplicity and efficiency. However, like any Monte Carlo method, they suffer from slow convergence and high statistical noise, especially for low-speed and unsteady flows. In the past two decades, the deterministic methods have undergone extensive development largely due to the advance in computing powers, see [28] for a recent review.

Among the deterministic methods for the Boltzmann equation, the Fourier-Galerkin spectral method stands out for its relatively high accuracy and the possibility of being accelerated by the fast Fourier transform (see, for instance, [29]–[32] for major algorithmic development and [33]–[36] for stability and convergence analysis). There are some other spectral methods [37]–[41] for the Boltzmann equation that use other orthogonal polynomial bases in $\mathbb{R}^d$.

In chapter 3, we study a fast Fourier spectral method [42] developed recently where a low-rank approximation was introduced for the speedup. Following a similar idea, we propose a new fast Fourier spectral method solving the spatially homogeneous Boltzmann equation. These two algorithms are compared in the numerical tests of the 3D variable hard sphere (VHS) molecule model. The new approach is able to get better accuracy in approximation to the high-frequency weight where the former method could not approximate very well.

Although being a method with reasonable efficiency and accuracy tradeoff, the Fourier spectral method requires a domain truncation which is unphysical since the original collision operator is defined in the whole space $\mathbb{R}^d$. This truncation changes the structure of the equation and often comes with an accuracy loss.

In chapter 4, we develop a Petrov-Galerkin spectral method for the Boltzmann equation (1.4) using mapped Chebyshev functions in $\mathbb{R}^d$. This is inspired by the recent work [43] where a spectral method was introduced for the 1D inelastic Boltzmann equation[1]. Both the test functions and trial functions are carefully chosen to obtain desired approximation properties. Furthermore, thanks to the close relationship of the Chebyshev functions and the Fourier cosine series, we are able to construct a fast algorithm with the help of the non-uniform fast Fourier transform (NUFFT). This speedup is critical as the direct implementation of the proposed method would require excessive storage for precomputation and significant online computational cost that soon become a bottleneck for larger N (the number of spectral modes). Extensive numerical tests in 2D and 3D are performed to demonstrate the accuracy and efficiency of the proposed method. In particular, the comparison with the Fourier spectral method in [32] indeed confirms the better approximation properties of the proposed method. Up to our knowledge, our method is the first with a fast implementation and a consistency analysis.

[1]↑Unlike the inelastic Boltzmann equation which has a non-trivial solution in 1D, the classical Boltzmann equation (1.4) must be considered at least for $d \geq 2$.

2. A STRUCTURE PRESERVING SCHEME FOR POISSON-NERST-PLANCK EQUATIONS

In this chapter, we propose a novel semi-implicit finite difference scheme for the PNP equations which will preserving the physical features in discrete level. We also discuss the solvability of the semi-discrete scheme and introduce a fixed point iteration to solve the fully discrete scheme. The numerical tests are performed in both 1D and 2D to verify the accuracy and structure preserving properties.

In Section 2.1, we give a brief introduction of the PNP equations in a bounded domain along with the basic properties. In Section 2.2, we describe in detail the fully discrete scheme in 1D and prove its properties: mass conservation, unconditional positivity, and energy dissipation. In addition, we prove the solvability of the semi-discrete scheme and propose a simple fixed point iteration to solve the fully discrete scheme. Extension to 2D is also discussed. Numerical examples are provided in Section 2.3 to demonstrate the convergence and properties of the proposed scheme. Concluding remarks are given in Section 2.4.

2.1 The PNP equations: initial boundary value problem and basic properties

In this section, we describe the initial boundary value problem of the PNP equations and summarize its basic properties.

2.1.1 Non-dimensionalization

To begin with, we first non-dimensionalize the equations (1.1) and (1.2) by introducing the following rescaled quantities:

$$\hat{c}^{(\mathrm{i})} = \frac{c^{(\mathrm{i})}}{c_0}, \quad \hat{\psi} = \frac{\psi}{\psi_0}, \quad \hat{\rho} = \frac{\rho}{\mathrm{e}c_0}, \quad \hat{\boldsymbol{x}} = \frac{\boldsymbol{x}}{x_0}, \quad \hat{D}^{(\mathrm{i})} = \frac{D^{(\mathrm{i})}}{D_0}, \quad \hat{t} = \frac{t}{x_0^2/D_0}, \quad \hat{\epsilon} = \frac{\epsilon}{\epsilon_0}, \tag{2.1}$$

where $c_0, \psi_0, \dots$ are the characteristic values of the corresponding quantities. Then (1.1) and (1.2) can be rewritten as

$$\frac{D_0}{x_0^2}\partial_{\hat{t}}\hat{c}^{(\mathrm{i})} = \frac{D_0}{x_0^2}\nabla\cdot\left\{\hat{D}^{(\mathrm{i})}\left(\nabla\hat{c}^{(\mathrm{i})} + \psi_0\frac{z_{\mathrm{i}}\mathrm{e}}{k_BT}\hat{c}^{(\mathrm{i})}\nabla\hat{\psi}\right)\right\}, \tag{2.2}$$

$$-\frac{\psi_0\epsilon_0}{x_0^2}\nabla\cdot\left(\hat{\epsilon}\nabla\hat{\psi}\right) = \mathrm{e}c_0\left(\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}\hat{c}^{(\mathrm{i})} + \hat{\rho}\right). \tag{2.3}$$

Define

$$\chi_1 := \frac{\mathrm{e}\psi_0}{k_BT}, \quad \chi_2 := \frac{\mathrm{e}c_0x_0^2}{\psi_0\epsilon_0}, \tag{2.4}$$

we obtain the non-dimensionalized PNP equations as (dropping $\hat{}$ for simplicity):

$$\partial_t c^{(\mathrm{i})} = \nabla\cdot\left(D^{(\mathrm{i})}\left(\nabla c^{(\mathrm{i})} + \chi_1 z_{\mathrm{i}}c^{(\mathrm{i})}\nabla\psi\right)\right), \tag{2.5}$$

$$-\nabla\cdot(\epsilon\nabla\psi) = \chi_2\left(\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}c^{(\mathrm{i})} + \rho\right). \tag{2.6}$$

For more physical background of these dimensionless parameters, we refer the interested reader to section 2.3 in article [22].

2.1.2 Initial and boundary value problem

When the PNP equations are imposed in a connected bounded domain $\Omega \subset \mathbb{R}^d$, proper initial and boundary conditions need to be supplemented.

For the Nernst-Planck equation (2.5), the initial condition is given by

$$c^{(\mathrm{i})}(0, \boldsymbol{x}) = c^{(\mathrm{i}),0}(\boldsymbol{x}), \quad \boldsymbol{x}\in\Omega, \quad \mathrm{i} = 1, \dots, m, \tag{2.7}$$

and the initial value of ψ is given by solving the Poisson equation (2.6) subject to (2.7).

For the boundary, one usually assumes the no-flux boundary condition for the Nernst-Planck equation, i.e.,

$$D^{(\mathrm{i})}\left(\nabla c^{(\mathrm{i})} + \chi_1 z_{\mathrm{i}}c^{(\mathrm{i})}\nabla\psi\right)\cdot\mathbf{n} = 0, \quad \boldsymbol{x}\in\partial\Omega, \quad t\geq 0, \quad \mathrm{i} = 1, \dots, m, \tag{2.8}$$

where $\mathbf{n}$ is the unit outward normal at the boundary point $\boldsymbol{x} \in \partial\Omega$. Boundary condition for the Poisson equation can be various. Here we consider a general boundary condition:

$$\alpha\psi + \beta\frac{\partial\psi}{\partial\mathbf{n}} = f, \quad \boldsymbol{x} \in \partial\Omega, \quad t \geq 0, \tag{2.9}$$

where α, β are some constants and $f = f(\boldsymbol{x})$ is a given function on $\partial\Omega$. Note that

- when $\alpha \neq 0, \beta \neq 0$, (2.9) is the Robin boundary condition;
- when $\alpha \neq 0, \beta = 0$, (2.9) reduces to the Dirichlet boundary condition;
- when $\alpha = 0, \beta \neq 0$, (2.9) reduces to the Neumann boundary condition. Solution to the Neumann problem can only be unique up to a constant. Also the following compatibility condition is required:

$$\chi_2 \int_\Omega \left(\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}} c^{(\mathrm{i}),0} + \rho \right) \mathrm{d}\boldsymbol{x} + \frac{1}{\beta} \int_{\partial\Omega} \epsilon f \, \mathrm{d}\mathbf{s} = 0. \tag{2.10}$$

Putting everything together, we have the following initial boundary value problem for the PNP equations:

$$\begin{cases} \partial_t c^{(\mathrm{i})} = \nabla \cdot \left(D^{(\mathrm{i})} \left(\nabla c^{(\mathrm{i})} + \chi_1 z_{\mathrm{i}} c^{(\mathrm{i})} \nabla\psi \right) \right), & \boldsymbol{x} \in \Omega, \quad t \geq 0, \quad \mathrm{i} = 1, \ldots, m, & (2.11) \\ c^{(\mathrm{i})}(0, \boldsymbol{x}) = c^{(\mathrm{i}),0}(\boldsymbol{x}), & \boldsymbol{x} \in \Omega, \quad \mathrm{i} = 1, \ldots, m, & (2.12) \\ D^{(\mathrm{i})} \left(\nabla c^{(\mathrm{i})} + \chi_1 z_{\mathrm{i}} c^{(\mathrm{i})} \nabla\psi \right) \cdot \mathbf{n} = 0, & \boldsymbol{x} \in \partial\Omega, \quad t \geq 0, \quad \mathrm{i} = 1, \ldots, m, & (2.13) \\ -\nabla \cdot (\epsilon \nabla\psi) = \chi_2 \left(\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}} c^{(\mathrm{i})} + \rho \right), & \boldsymbol{x} \in \Omega, \quad t \geq 0, & (2.14) \\ \alpha\psi + \beta\dfrac{\partial\psi}{\partial\mathbf{n}} = f, & \boldsymbol{x} \in \partial\Omega, \quad t \geq 0. & (2.15) \end{cases}$$

2.1.3 Basic properties

Here we list a few important properties of the problem (2.11)–(2.15), which will serve as a guidance in designing numerical schemes.

1. Mass conservation:

$$\int_\Omega c^{(\mathrm{i})}(t,\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_\Omega c^{(\mathrm{i}),0}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \quad \forall t>0,\ \mathrm{i}=1,\dots,m. \tag{2.16}$$

2. Positivity:

$$c^{(\mathrm{i}),0}(\boldsymbol{x}) \geq 0 \ \Rightarrow\ c^{(\mathrm{i})}(t,\boldsymbol{x}) \geq 0, \quad \forall t>0,\ \boldsymbol{x}\in\Omega,\ \mathrm{i}=1,\dots,m. \tag{2.17}$$

3. Energy dissipation:

$$\frac{\mathrm{d}\tilde{E}}{\mathrm{d}t} = -\sum_{\mathrm{i}=1}^{m}\int_\Omega D^{(\mathrm{i})}c^{(\mathrm{i})}\left|\nabla\Big(\log c^{(\mathrm{i})}+\chi_1 z_{\mathrm{i}}\psi\Big)\right|^2\mathrm{d}\boldsymbol{x} + \frac{\chi_1}{2\chi_2}\int_{\partial\Omega}\epsilon\left(\psi\frac{\partial\psi_t}{\partial\mathbf{n}} - \psi_t\frac{\partial\psi}{\partial\mathbf{n}}\right)\mathrm{d}\mathbf{s}, \tag{2.18}$$

where the free energy $\tilde{E}$ is defined as

$$\tilde{E} = \int_\Omega \sum_{\mathrm{i}=1}^{m}\Big(c^{(\mathrm{i})}\log c^{(\mathrm{i})}\Big)\,\mathrm{d}\boldsymbol{x} + \frac{\chi_1}{2}\int_\Omega\left(\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}c^{(\mathrm{i})}+\rho\right)\psi\,\mathrm{d}\boldsymbol{x}. \tag{2.19}$$

Note that using the boundary condition (2.15) and f does not depend on time, the last term on the right hand side of (2.18) can be written equivalently as

$$\frac{\chi_1}{2\chi_2}\int_{\partial\Omega}\epsilon\left(\psi\frac{\partial\psi_t}{\partial\mathbf{n}} - \psi_t\frac{\partial\psi}{\partial\mathbf{n}}\right)\mathrm{d}\mathbf{s} = \begin{cases} \dfrac{\chi_1}{2\chi_2\alpha}\displaystyle\int_{\partial\Omega}\epsilon f\frac{\partial\psi_t}{\partial\mathbf{n}}\,\mathrm{d}\mathbf{s}, & \text{if } \alpha\neq 0,\\[2ex] -\dfrac{\chi_1}{2\chi_2\beta}\displaystyle\int_{\partial\Omega}\epsilon f\psi_t\,\mathrm{d}\mathbf{s}, & \text{if } \beta\neq 0. \end{cases} \tag{2.20}$$

Therefore, to make (2.18) dissipative, one can choose

$$E = \tilde{E} + \begin{cases} -\dfrac{\chi_1}{2\chi_2\alpha}\displaystyle\int_{\partial\Omega}\epsilon f\frac{\partial\psi}{\partial\mathbf{n}}\,\mathrm{d}\mathbf{s}, & \text{if } \alpha\neq 0,\\[2ex] \dfrac{\chi_1}{2\chi_2\beta}\displaystyle\int_{\partial\Omega}\epsilon f\psi\,\mathrm{d}\mathbf{s}, & \text{if } \beta\neq 0. \end{cases} \tag{2.21}$$

Then one has

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -\sum_{\mathrm{i}=1}^{m}\int_\Omega D^{(\mathrm{i})}c^{(\mathrm{i})}\left|\nabla\Big(\log c^{(\mathrm{i})}+\chi_1 z_{\mathrm{i}}\psi\Big)\right|^2\mathrm{d}\boldsymbol{x} \leq 0. \tag{2.22}$$

4. Steady state: the energy dissipation implies that the steady state of the system is achieved when

$$\nabla\left(\log c^{(\mathrm{i}),\infty}+\chi_1 z_{\mathrm{i}}\psi^{\infty}\right)=0,\quad \mathrm{i}=1,\ldots,m, \tag{2.23}$$

which integrates to the equilibrium

$$c^{(\mathrm{i}),\infty}=\lambda_{\mathrm{i}}\mathrm{e}^{-\chi_1 z_{\mathrm{i}}\psi^{\infty}},\quad \text{with } \lambda_{\mathrm{i}}=\frac{\int_\Omega c^{(\mathrm{i}),0}\,\mathrm{d}\boldsymbol{x}}{\int_\Omega \mathrm{e}^{-\chi_1 z_{\mathrm{i}}\psi^{\infty}}\,\mathrm{d}\boldsymbol{x}}. \tag{2.24}$$

Substituting $c^{(\mathrm{i}),\infty}$ into the Poisson equation (2.14) leads to

$$-\nabla\cdot(\epsilon\nabla\psi^{\infty})=\chi_2\left(\sum_{\mathrm{i}=1}^{m}\lambda_{\mathrm{i}}z_{\mathrm{i}}\mathrm{e}^{-\chi_1 z_{\mathrm{i}}\psi^{\infty}}+\rho\right),\quad \boldsymbol{x}\in\Omega, \tag{2.25}$$

which together with the boundary condition (2.15) constitute the (nonlinear) Poisson-Boltzmann equation.

2.2 Numerical schemes

In this section, we describe the proposed numerical scheme for the initial boundary value problem (2.11)–(2.15). For simplicity, we assume $\chi_1=\chi_2=1$ in the following.

Before going into detail, we first summarize the key ingredients in our method.

- The **first ingredient** is to reformulate the Nernst-Planck equation (2.11) as

$$\partial_t c^{(\mathrm{i})}=\nabla\cdot\left(D^{(\mathrm{i})}M^{(\mathrm{i})}\nabla\left(\frac{c^{(\mathrm{i})}}{M^{(\mathrm{i})}}\right)\right),\quad \text{where } M^{(\mathrm{i})}=\mathrm{e}^{-z_{\mathrm{i}}\psi}. \tag{2.26}$$

Accordingly, the no-flux boundary condition (2.13) becomes

$$\nabla\left(\frac{c^{(\mathrm{i})}}{M^{(\mathrm{i})}}\right)\cdot\mathbf{n}=0. \tag{2.27}$$

Note that this is the Scharfetter-Gummel transform widely used in semiconductor community [44].

- The **second ingredient** is the spatial discretization. As both (2.26) and the Poisson equation (2.14) are diffusive type equations, it is simple and natural to use the central finite difference.

- The **third ingredient** (which is our main contribution) is a semi-implicit time discretization

$$\begin{cases} \dfrac{c^{(\mathrm{i}),n+1}-c^{(\mathrm{i}),n}}{\Delta t} = \nabla\cdot\left(D^{(\mathrm{i})}M^{(\mathrm{i}),*}\nabla\left(\dfrac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)\right), \\ -\nabla\cdot(\epsilon\nabla\psi^{n+1}) = \displaystyle\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}c^{(\mathrm{i}),n+1}+\rho, \end{cases} \tag{2.28}$$

  where $M^{(\mathrm{i}),*}=\mathrm{e}^{-z_{\mathrm{i}}\psi^*}$ and the potential ψ^* is chosen as

$$\psi^* = \frac{\psi^n+\psi^{n+1}}{2}. \tag{2.29}$$

2.2.1 Fully discrete scheme in 1D

We now describe in detail the proposed scheme in 1D. Assume the domain $\Omega=[a,b]$, then the PNP system reads

$$\begin{cases} \partial_t c^{(\mathrm{i})} = \left(D^{(\mathrm{i})}M^{(\mathrm{i})}\left(\dfrac{c^{(\mathrm{i})}}{M^{(\mathrm{i})}}\right)_x\right)_x, & x\in[a,b],\quad t\geq 0,\quad \mathrm{i}=1,\dots,m, & (2.30)\\ c^{(\mathrm{i})}(0,x)=c^{(\mathrm{i}),0}(x), & x\in[a,b],\quad \mathrm{i}=1,\dots,m, & (2.31)\\ \left(\dfrac{c^{(\mathrm{i})}}{M^{(\mathrm{i})}}\right)_x(t,a)=\left(\dfrac{c^{(\mathrm{i})}}{M^{(\mathrm{i})}}\right)_x(t,b)=0, & t\geq 0,\quad \mathrm{i}=1,\dots,m, & (2.32)\\ -(\epsilon\psi_x)_x=\displaystyle\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}c^{(\mathrm{i})}+\rho, & x\in[a,b],\quad t\geq 0, & (2.33)\\ \alpha\psi(t,a)-\beta\psi_x(t,a)=f_a, & t\geq 0, & (2.34)\\ \alpha\psi(t,b)+\beta\psi_x(t,b)=f_b, & t\geq 0. & (2.35) \end{cases}$$

We partition the interval $[a,b]$ into N uniform cells with mesh size $\Delta x=(b-a)/N$. The cell centers $x_{\mathrm{j}}=a+(\mathrm{j}-1/2)\Delta x$, $\mathrm{j}=1,\dots,N$ are chosen as the grid points; and the cell interfaces are given by $x_{\mathrm{j}+1/2}=a+\mathrm{j}\Delta x$, $\mathrm{j}=0,\dots,N$ (note that $x_{1/2}=a, x_{N+1/2}=b$). Let $t_n=n\Delta t$ be the discrete time step and we denote the numerical approximation of a function $u(t,x)$ at (t_n,x_{j}) by u^n_{j}.

We first discretize the Nernst-Planck equation (2.30) in space by a second-order central difference scheme:

$$\partial_t c_{\mathrm{j}}^{(\mathrm{i})} = \frac{1}{\Delta x^2}\left(D_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})}\overline{M}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})}\hat{g}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})} - D_{\mathrm{j}-\frac{1}{2}}^{(\mathrm{i})}\overline{M}_{\mathrm{j}-\frac{1}{2}}^{(\mathrm{i})}\hat{g}_{\mathrm{j}-\frac{1}{2}}^{(\mathrm{i})}\right), \quad \mathrm{j}=1,\cdots,N, \tag{2.36}$$

where $\hat{g}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})}$ is defined by

$$\hat{g}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})} = \left(\frac{c^{(\mathrm{i})}}{M^{(\mathrm{i})}}\right)_{\mathrm{j}+1} - \left(\frac{c^{(\mathrm{i})}}{M^{(\mathrm{i})}}\right)_{\mathrm{j}}, \quad \mathrm{j}=1,\ldots,N-1. \tag{2.37}$$

At the boundary ($\mathrm{j}=0,N$), due to the no-flux boundary condition (2.32), we set

$$\hat{g}_{\frac{1}{2}}^{(\mathrm{i})} = \hat{g}_{N+\frac{1}{2}}^{(\mathrm{i})} = 0. \tag{2.38}$$

$D_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})}$ is the value of the diffusion coefficient $D^{(\mathrm{i})}$ at $x_{\mathrm{j}+\frac{1}{2}}$. $\overline{M}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})}$ is an approximation to $M^{(\mathrm{i})}$ at $x_{\mathrm{j}+\frac{1}{2}}$ and we take

$$\overline{M}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})} = \frac{M_{\mathrm{j}}^{(\mathrm{i})} + M_{\mathrm{j}+1}^{(\mathrm{i})}}{2}, \quad M_{\mathrm{j}}^{(\mathrm{i})} = \mathrm{e}^{-z_{\mathrm{i}}\psi_{\mathrm{j}}}, \quad \mathrm{j}=1,\ldots,N-1. \tag{2.39}$$

Remark 1. *We remark that the choice of $\overline{M}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})}$ is not unique. As long as it is a second order, positive approximation to $M^{(\mathrm{i})}$ at $x_{\mathrm{j}+\frac{1}{2}}$, all the properties derived in Section 2.2.1 can be carried over.*

For the Poisson equation (2.33), we also use the central difference scheme:

$$-\frac{1}{\Delta x^2}\left(\epsilon_{\mathrm{j}+\frac{1}{2}}\hat{\psi}_{\mathrm{j}+\frac{1}{2}} - \epsilon_{\mathrm{j}-\frac{1}{2}}\hat{\psi}_{\mathrm{j}-\frac{1}{2}}\right) = \sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}} c_{\mathrm{j}}^{(\mathrm{i})} + \rho_{\mathrm{j}}, \quad \mathrm{j}=1,\cdots,N, \tag{2.40}$$

where $\epsilon_{\mathrm{j}+\frac{1}{2}}$ is the value of the permittivity ϵ at $x_{\mathrm{j}+\frac{1}{2}}$, and $\hat{\psi}_{\mathrm{j}+\frac{1}{2}}$ is defined by

$$\hat{\psi}_{\mathrm{j}+\frac{1}{2}} = \psi_{\mathrm{j}+1} - \psi_{\mathrm{j}}, \quad \mathrm{j}=0,\ldots,N. \tag{2.41}$$

To obtain ψ_0 and ψ_{N+1}, note that the boundary condition (2.34) (2.35) can be discretized as

$$\alpha\frac{\psi_1+\psi_0}{2}-\beta\frac{\psi_1-\psi_0}{\Delta x}=f_a,\quad \alpha\frac{\psi_{N+1}+\psi_N}{2}+\beta\frac{\psi_{N+1}-\psi_N}{\Delta x}=f_b, \tag{2.42}$$

using which we can represent

$$\hat{\psi}_{\frac{1}{2}}:=\psi_1-\psi_0=\frac{2\alpha\Delta x}{\alpha\Delta x+2\beta}\psi_1-\frac{2\Delta x}{\alpha\Delta x+2\beta}f_a, \tag{2.43}$$

$$\hat{\psi}_{N+\frac{1}{2}}:=\psi_{N+1}-\psi_N=-\frac{2\alpha\Delta x}{\alpha\Delta x+2\beta}\psi_N+\frac{2\Delta x}{\alpha\Delta x+2\beta}f_b. \tag{2.44}$$

Remark 2. *$\hat{\psi}_{\frac{1}{2}}$ and $\hat{\psi}_{N+\frac{1}{2}}$ may not be well-defined in the case of Robin boundary (when $\alpha\neq 0$, $\beta\neq 0$). In this case, we assume $\Delta x\neq -2\beta/\alpha$.*

For brevity, we write the scheme (2.40) in a matrix vector multiplication form:

$$P\boldsymbol{\Psi}=\mathbf{h}, \tag{2.45}$$

where

$$P=\begin{pmatrix} p_{1,1} & -\epsilon_{\frac{3}{2}} & & & \\ -\epsilon_{\frac{3}{2}} & (\epsilon_{\frac{3}{2}}+\epsilon_{\frac{5}{2}}) & -\epsilon_{\frac{5}{2}} & & \\ & \ddots & \ddots & \ddots & \\ & & -\epsilon_{N-\frac{3}{2}} & (\epsilon_{N-\frac{3}{2}}+\epsilon_{N-\frac{1}{2}}) & -\epsilon_{N-\frac{1}{2}} \\ & & & -\epsilon_{N-\frac{1}{2}} & p_{N,N} \end{pmatrix},\Psi=\begin{pmatrix}\psi_1\\ \psi_2\\ \vdots\\ \psi_N\end{pmatrix},\mathbf{h}=\begin{pmatrix}h_1\\ h_2\\ \vdots\\ h_N\end{pmatrix}, \tag{2.46}$$

with

$$\begin{cases} p_{1,1}=\dfrac{2\alpha\Delta x}{\alpha\Delta x+2\beta}\epsilon_{\frac{1}{2}}+\epsilon_{\frac{3}{2}},\\ p_{N,N}=\epsilon_{N-\frac{1}{2}}+\dfrac{2\alpha\Delta x}{\alpha\Delta x+2\beta}\epsilon_{N+\frac{1}{2}}, \end{cases} \quad \begin{cases} h_1=\Delta x^2\left(\sum\limits_{\mathrm{i}=1}^{m}z_{\mathrm{i}}c_1^{(\mathrm{i})}+\rho_1\right)+\dfrac{2\Delta x}{\alpha\Delta x+2\beta}\epsilon_{\frac{1}{2}}f_a,\\ h_{\mathrm{j}}=\Delta x^2\left(\sum\limits_{\mathrm{i}=1}^{m}z_{\mathrm{i}}c_{\mathrm{j}}^{(\mathrm{i})}+\rho_{\mathrm{j}}\right),\quad \mathrm{j}=2,\ldots,N-1,\\ h_N=\Delta x^2\left(\sum\limits_{\mathrm{i}=1}^{m}z_{\mathrm{i}}c_N^{(\mathrm{i})}+\rho_N\right)+\dfrac{2\Delta x}{\alpha\Delta x+2\beta}\epsilon_{N+\frac{1}{2}}f_b. \end{cases}$$

Now let us add the time discretization as outlined in (2.28). Define

$$M_{\mathrm{j}}^{(\mathrm{i}),*} = \mathrm{e}^{-z_{\mathrm{i}}\psi_{\mathrm{j}}^*}, \quad \psi_{\mathrm{j}}^* = \frac{\psi_{\mathrm{j}}^n + \psi_{\mathrm{j}}^{n+1}}{2}, \tag{2.47}$$

then (2.36) with time discretization reads

$$\begin{aligned}\frac{c_{\mathrm{j}}^{(\mathrm{i}),n+1} - c_{\mathrm{j}}^{(\mathrm{i}),n}}{\Delta t} = \frac{1}{\Delta x^2}\Bigg\{ & D^{(\mathrm{i})}_{\mathrm{j}+\frac{1}{2}} \overline{M}^{(\mathrm{i}),*}_{\mathrm{j}+\frac{1}{2}} \left[\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_{\mathrm{j}+1} - \left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_{\mathrm{j}} \right] \\ & - D^{(\mathrm{i})}_{\mathrm{j}-\frac{1}{2}} \overline{M}^{(\mathrm{i}),*}_{\mathrm{j}-\frac{1}{2}} \left[\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_{\mathrm{j}} - \left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_{\mathrm{j}-1} \right] \Bigg\}, \quad \mathrm{j} = 2, \ldots, N-1; \end{aligned} \tag{2.48}$$

and for $\mathrm{j} = 1$ and N:

$$\frac{c_1^{(\mathrm{i}),n+1} - c_1^{(\mathrm{i}),n}}{\Delta t} = \frac{1}{\Delta x^2}\left\{ D^{(\mathrm{i})}_{\frac{3}{2}} \overline{M}^{(\mathrm{i}),*}_{\frac{3}{2}} \left[\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_2 - \left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_1 \right] \right\}, \tag{2.49}$$

$$\frac{c_N^{(\mathrm{i}),n+1} - c_N^{(\mathrm{i}),n}}{\Delta t} = \frac{1}{\Delta x^2}\left\{ -D^{(\mathrm{i})}_{N-\frac{1}{2}} \overline{M}^{(\mathrm{i}),*}_{N-\frac{1}{2}} \left[\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_N - \left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_{N-1} \right] \right\}. \tag{2.50}$$

Rearranging terms in (2.48) yields

$$\begin{aligned} & \left[M_{\mathrm{j}}^{(\mathrm{i}),*} + \frac{\Delta t}{\Delta x^2} \left(D^{(\mathrm{i})}_{\mathrm{j}+\frac{1}{2}} \overline{M}^{(\mathrm{i}),*}_{\mathrm{j}+\frac{1}{2}} + D^{(\mathrm{i})}_{\mathrm{j}-\frac{1}{2}} \overline{M}^{(\mathrm{i}),*}_{\mathrm{j}-\frac{1}{2}} \right) \right] \left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_{\mathrm{j}} \\ & - \frac{\Delta t}{\Delta x^2} D^{(\mathrm{i})}_{\mathrm{j}+\frac{1}{2}} \overline{M}^{(\mathrm{i}),*}_{\mathrm{j}+\frac{1}{2}} \left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_{\mathrm{j}+1} - \frac{\Delta t}{\Delta x^2} D^{(\mathrm{i})}_{\mathrm{j}-\frac{1}{2}} \overline{M}^{(\mathrm{i}),*}_{\mathrm{j}-\frac{1}{2}} \left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_{\mathrm{j}-1} = c_{\mathrm{j}}^{(\mathrm{i}),n}, \quad \mathrm{j} = 2, \ldots, N-1. \end{aligned} \tag{2.51}$$

Similarly, (2.49) (2.50) become

$$\left[M_1^{(\mathrm{i}),*} + \frac{\Delta t}{\Delta x^2} D^{(\mathrm{i})}_{\frac{3}{2}} \overline{M}^{(\mathrm{i}),*}_{\frac{3}{2}} \right] \left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_1 - \frac{\Delta t}{\Delta x^2} D^{(\mathrm{i})}_{\frac{3}{2}} \overline{M}^{(\mathrm{i}),*}_{\frac{3}{2}} \left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_2 = c_1^{(\mathrm{i}),n}, \tag{2.52}$$

$$\left[M_N^{(\mathrm{i}),*} + \frac{\Delta t}{\Delta x^2} D^{(\mathrm{i})}_{N-\frac{1}{2}} \overline{M}^{(\mathrm{i}),*}_{N-\frac{1}{2}} \right] \left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_N - \frac{\Delta t}{\Delta x^2} D^{(\mathrm{i})}_{N-\frac{1}{2}} \overline{M}^{(\mathrm{i}),*}_{N-\frac{1}{2}} \left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_{N-1} = c_N^{(\mathrm{i}),n}. \tag{2.53}$$

The schemes (2.51)-(2.53) can be written in a matrix vector multiplication form:

$$A^{(\mathrm{i})} \mathbf{g}^{(\mathrm{i})} = \mathbf{c}^{(\mathrm{i}),n}, \tag{2.54}$$

if we define

$$A^{(\mathrm{i})}=\begin{pmatrix} a_{1,1}^{(\mathrm{i})} & a_{1,2}^{(\mathrm{i})} & & \\ a_{2,1}^{(\mathrm{i})} & a_{2,2}^{(\mathrm{i})} & a_{2,3}^{(\mathrm{i})} & \\ & \ddots & \ddots & \ddots \\ & & a_{N,N-1}^{(\mathrm{i})} & a_{N,N}^{(\mathrm{i})} \end{pmatrix}, \quad \mathbf{g}^{(\mathrm{i})}=\begin{pmatrix} c_1^{(\mathrm{i}),n+1}/M_1^{(\mathrm{i}),*} \\ c_2^{(\mathrm{i}),n+1}/M_2^{(\mathrm{i}),*} \\ \vdots \\ c_N^{(\mathrm{i}),n+1}/M_N^{(\mathrm{i}),*} \end{pmatrix}, \quad \mathbf{c}^{(\mathrm{i}),n}=\begin{pmatrix} c_1^{(\mathrm{i}),n} \\ c_2^{(\mathrm{i}),n} \\ \vdots \\ c_N^{(\mathrm{i}),n} \end{pmatrix}, \tag{2.55}$$

where the entries of the matrix $A^{(\mathrm{i})}$ are given by

$$\begin{cases} a_{1,1}^{(\mathrm{i})}=M_1^{(\mathrm{i}),*}+\dfrac{\Delta t}{\Delta x^2}D_{\frac{3}{2}}^{(\mathrm{i})}\overline{M}_{\frac{3}{2}}^{(\mathrm{i}),*}, \\ a_{N,N}^{(\mathrm{i})}=M_N^{(\mathrm{i}),*}+\dfrac{\Delta t}{\Delta x^2}D_{N-\frac{1}{2}}^{(\mathrm{i})}\overline{M}_{N-\frac{1}{2}}^{(\mathrm{i}),*}, \end{cases} \quad \begin{cases} a_{\mathrm{j},\mathrm{j}}^{(\mathrm{i})}=M_{\mathrm{j}}^{(\mathrm{i}),*}+\dfrac{\Delta t}{\Delta x^2}\left(D_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})}\overline{M}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i}),*}+D_{\mathrm{j}-\frac{1}{2}}^{(\mathrm{i})}\overline{M}_{\mathrm{j}-\frac{1}{2}}^{(\mathrm{i}),*}\right), \quad \mathrm{j}=2,\dots,N-1 \\ a_{\mathrm{j},\mathrm{j}-1}^{(\mathrm{i})}=-\dfrac{\Delta t}{\Delta x^2}D_{\mathrm{j}-\frac{1}{2}}^{(\mathrm{i})}\overline{M}_{\mathrm{j}-\frac{1}{2}}^{(\mathrm{i}),*}, \quad \mathrm{j}=2,\dots,N, \\ a_{\mathrm{j},\mathrm{j}+1}^{(\mathrm{i})}=-\dfrac{\Delta t}{\Delta x^2}D_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})}\overline{M}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i}),*}, \quad \mathrm{j}=1,\dots,N-1; \end{cases}$$

Therefore, together with the system (2.45), we obtain the following fully discrete scheme for the PNP system:

$$\begin{cases} A^{(\mathrm{i})}\left(\mathbf{M}^{(\mathrm{i}),*}\right)\mathbf{g}^{(\mathrm{i})}\left(\mathbf{c}^{(\mathrm{i}),n+1},\mathbf{M}^{(\mathrm{i}),*}\right)=\mathbf{c}^{(\mathrm{i}),n}, \\ P\boldsymbol{\Psi}^{n+1}=\mathbf{h}\left(\mathbf{c}^{(\mathrm{i}),n+1}\right), \end{cases} \tag{2.56}$$

where with a little abuse of notations, the dependence of vectors is indicated.

We state the following lemma which will be useful later.

Lemma 2.2.1. *The matrix $A^{(\mathrm{i})}\left(\mathbf{M}^{(\mathrm{i}),*}\right)$ as defined in (2.55) is symmetric positive definite and strictly diagonally dominant, provided $\overline{M}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i}),*}$ is a second-order, positive approximation to $M^{(\mathrm{i}),*}$ at $x_{\mathrm{j}+\frac{1}{2}}$. In particular, the choice*

$$\overline{M}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i}),*}=\frac{M_{\mathrm{j}}^{(\mathrm{i}),*}+M_{\mathrm{j}+1}^{(\mathrm{i}),*}}{2} \tag{2.57}$$

suffices.

Proof. By definition, $M_{\mathrm{j}}^{(\mathrm{i}),*} = \mathrm{e}^{-z_{\mathrm{i}}\psi_{\mathrm{j}}^*} > 0$, and $\overline{M}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i}),*}$ is required to be positive, then the entries of $A^{(\mathrm{i})}$ satisfy

$$a_{\mathrm{j},\mathrm{j}}^{(\mathrm{i})} > 0, \quad a_{\mathrm{j},k}^{(\mathrm{i})} \leq 0, \quad \mathrm{j} \neq k. \tag{2.58}$$

Furthermore,

$$\left|a_{\mathrm{j},\mathrm{j}}^{(\mathrm{i})}\right| > \sum_{\mathrm{j}\neq k} \left|a_{\mathrm{j},k}^{(\mathrm{i})}\right|. \tag{2.59}$$

Hence the conclusion is immediate. □

Properties of the fully discrete scheme

In this section, we prove the properties of the fully discrete scheme (2.56). These are parallel to the theoretical properties listed in Section 2.1.3.

Define the total mass of the i-th ion species at time step t_n as

$$C_{\Delta}^{(\mathrm{i})}(t_n) = \Delta x \sum_{\mathrm{j}=1}^{N} c_{\mathrm{j}}^{(\mathrm{i}),n}. \tag{2.60}$$

Then we have

Theorem 2.1. (Mass conservation) *The fully discrete scheme (2.56) is always mass conservative for each ion species:*

$$C_{\Delta}^{(\mathrm{i})}(t_n) = C_{\Delta}^{(\mathrm{i})}(t_{n+1}), \quad \mathrm{i} = 1, \ldots, m. \tag{2.61}$$

Proof. Using (2.48), (2.49) and (2.50), it is easy to see

$$C_{\Delta}^{(\mathrm{i})}(t_{n+1}) - C_{\Delta}^{(\mathrm{i})}(t_n) = \Delta x \sum_{\mathrm{j}=1}^{N} \left(c_{\mathrm{j}}^{(\mathrm{i}),n+1} - c_{\mathrm{j}}^{(\mathrm{i}),n}\right) = 0. \tag{2.62}$$

This proves the numerical mass conservation. □

Theorem 2.2. (Positivity preserving) *The fully discrete scheme (2.56) is unconditionally positivity-preserving, i.e., if* $c_{\mathrm{j}}^{(\mathrm{i}),n} \geq 0$ *for all* $\mathrm{j} = 1, \ldots, N$, *then*

$$c_{\mathrm{j}}^{(\mathrm{i}),n+1} \geq 0, \quad \mathrm{j} = 1, \ldots, N, \tag{2.63}$$

for each species $\mathrm{i} = 1, \ldots, m$.

Proof. Lemma 2.2.1 implies that the matrix $A^{(\mathrm{i})}\left(\mathbf{M}^{(\mathrm{i}),*}\right)$ in the scheme (2.56) is a M-matrix, i.e., it is inverse positive ($(A^{(\mathrm{i})})^{-1}$ exists and each entry of $(A^{(\mathrm{i})})^{-1}$ is non-negative). Therefore, if $c_{\mathrm{j}}^{(\mathrm{i}),n} \geq 0$, by solving the first linear system in (2.56), we have $g_{\mathrm{j}}^{(\mathrm{i})} \geq 0$. Since $M_{\mathrm{j}}^{(\mathrm{i}),*} > 0$, then $c_{\mathrm{j}}^{(\mathrm{i}),n+1} = g_{\mathrm{j}}^{(\mathrm{i})} M_{\mathrm{j}}^{(\mathrm{i}),*} \geq 0$. □

Define the discrete free energy at time step t_n as

$$E_\Delta(t_n) = \Delta x \sum_{\mathrm{j}=1}^{N} \sum_{\mathrm{i}=1}^{m} c_{\mathrm{j}}^{(\mathrm{i}),n} \log c_{\mathrm{j}}^{(\mathrm{i}),n} + \frac{\Delta x}{2} \sum_{\mathrm{j}=1}^{N} \left(\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}} c_{\mathrm{j}}^{(\mathrm{i}),n} + \rho_{\mathrm{j}} \right) \psi_{\mathrm{j}}^n + \frac{\epsilon_{\frac{1}{2}} f_a \psi_1^n + \epsilon_{N+\frac{1}{2}} f_b \psi_N^n}{\alpha \Delta x + 2\beta}, \tag{2.64}$$

where we assume $\Delta x \neq -2\beta/\alpha$ when both α and β are nonzero. Then we have

Theorem 2.3. (Energy dissipation) *The fully discrete scheme (2.56) is unconditionally energy-dissipative:*

$$\begin{aligned}
&E_\Delta(t_{n+1}) - E_\Delta(t_n) \\
\leq& -\frac{\Delta t}{\Delta x} \sum_{\mathrm{j}=1}^{N-1} \sum_{\mathrm{i}=1}^{m} D_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})} \overline{M}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i}),*} \left[\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_{\mathrm{j}+1} - \left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_{\mathrm{j}} \right] \left[\log \left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_{\mathrm{j}+1} - \log \left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}} \right)_{\mathrm{j}} \right] \\
\leq& 0.
\end{aligned} \tag{2.65}$$

Proof. Using the definition (2.64), we have

$$
\begin{aligned}
&E_\Delta(t_{n+1}) - E_\Delta(t_n) \\
=&\Delta x \sum_{\mathrm{j}=1}^{N}\sum_{\mathrm{i}=1}^{m}\left(c_{\mathrm{j}}^{(\mathrm{i}),n+1}\log c_{\mathrm{j}}^{(\mathrm{i}),n+1} - c_{\mathrm{j}}^{(\mathrm{i}),n}\log c_{\mathrm{j}}^{(\mathrm{i}),n}\right) + \frac{\Delta x}{2}\sum_{\mathrm{j}=1}^{N}\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}\left(c_{\mathrm{j}}^{(\mathrm{i}),n+1}\psi_{\mathrm{j}}^{n+1} - c_{\mathrm{j}}^{(\mathrm{i}),n}\psi_{\mathrm{j}}^{n}\right) \\
&+ \frac{\Delta x}{2}\sum_{\mathrm{j}=1}^{N}\rho_{\mathrm{j}}\left(\psi_{\mathrm{j}}^{n+1} - \psi_{\mathrm{j}}^{n}\right) + \frac{\epsilon_{\frac{1}{2}} f_a(\psi_1^{n+1} - \psi_1^n) + \epsilon_{N+\frac{1}{2}} f_b(\psi_N^{n+1} - \psi_N^n)}{\alpha\Delta x + 2\beta} \\
=&\Delta x \sum_{\mathrm{j}=1}^{N}\sum_{\mathrm{i}=1}^{m}\left[c_{\mathrm{j}}^{(\mathrm{i}),n+1}\log c_{\mathrm{j}}^{(\mathrm{i}),n+1} - c_{\mathrm{j}}^{(\mathrm{i}),n}\log c_{\mathrm{j}}^{(\mathrm{i}),n} + \left(c_{\mathrm{j}}^{(\mathrm{i}),n} - c_{\mathrm{j}}^{(\mathrm{i}),n+1}\right)\log c_{\mathrm{j}}^{(\mathrm{i}),n+1}\right] \\
&+ \Delta x \sum_{\mathrm{j}=1}^{N}\sum_{\mathrm{i}=1}^{m}\left(c_{\mathrm{j}}^{(\mathrm{i}),n+1} - c_{\mathrm{j}}^{(\mathrm{i}),n}\right)\left(\log c_{\mathrm{j}}^{(\mathrm{i}),n+1} + z_{\mathrm{i}}\psi_{\mathrm{j}}^{*}\right) \\
&+ \Delta x \sum_{\mathrm{j}=1}^{N}\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}\left[\frac{1}{2}\left(c_{\mathrm{j}}^{(\mathrm{i}),n+1}\psi_{\mathrm{j}}^{n+1} - c_{\mathrm{j}}^{(\mathrm{i}),n}\psi_{\mathrm{j}}^{n}\right) + \left(c_{\mathrm{j}}^{(\mathrm{i}),n} - c_{\mathrm{j}}^{(\mathrm{i}),n+1}\right)\psi_{\mathrm{j}}^{*}\right] \\
&+ \frac{\Delta x}{2}\sum_{\mathrm{j}=1}^{N}\rho_{\mathrm{j}}(\psi_{\mathrm{j}}^{n+1} - \psi_{\mathrm{j}}^{n}) + \frac{\epsilon_{\frac{1}{2}} f_a(\psi_1^{n+1} - \psi_1^n) + \epsilon_{N+\frac{1}{2}} f_b(\psi_N^{n+1} - \psi_N^n)}{\alpha\Delta x + 2\beta} \\
=&I + II + III,
\end{aligned}
\tag{2.66}
$$

where the three parts are defined as:

$$
\begin{aligned}
I :=& \Delta x \sum_{\mathrm{j}=1}^{N}\sum_{\mathrm{i}=1}^{m}\left[c_{\mathrm{j}}^{(\mathrm{i}),n+1}\log c_{\mathrm{j}}^{(\mathrm{i}),n+1} - c_{\mathrm{j}}^{(\mathrm{i}),n}\log c_{\mathrm{j}}^{(\mathrm{i}),n} + \left(c_{\mathrm{j}}^{(\mathrm{i}),n} - c_{\mathrm{j}}^{(\mathrm{i}),n+1}\right)\log c_{\mathrm{j}}^{(\mathrm{i}),n+1}\right], \\
II :=& \Delta x \sum_{\mathrm{j}=1}^{N}\sum_{\mathrm{i}=1}^{m}\left(c_{\mathrm{j}}^{(\mathrm{i}),n+1} - c_{\mathrm{j}}^{(\mathrm{i}),n}\right)\left(\log c_{\mathrm{j}}^{(\mathrm{i}),n+1} + z_{\mathrm{i}}\psi_{\mathrm{j}}^{*}\right), \\
III :=& \Delta x \sum_{\mathrm{j}=1}^{N}\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}\left[\frac{1}{2}\left(c_{\mathrm{j}}^{(\mathrm{i}),n+1}\psi_{\mathrm{j}}^{n+1} - c_{\mathrm{j}}^{(\mathrm{i}),n}\psi_{\mathrm{j}}^{n}\right) + \left(c_{\mathrm{j}}^{(\mathrm{i}),n} - c_{\mathrm{j}}^{(\mathrm{i}),n+1}\right)\psi_{\mathrm{j}}^{*}\right] \\
&+ \frac{\Delta x}{2}\sum_{\mathrm{j}=1}^{N}\rho_{\mathrm{j}}(\psi_{\mathrm{j}}^{n+1} - \psi_{\mathrm{j}}^{n}) + \frac{\epsilon_{\frac{1}{2}} f_a(\psi_1^{n+1} - \psi_1^n) + \epsilon_{N+\frac{1}{2}} f_b(\psi_N^{n+1} - \psi_N^n)}{\alpha\Delta x + 2\beta}.
\end{aligned}
\tag{2.67}
$$

For part I,

$$
\begin{aligned}
I =& \Delta x \sum_{\mathrm{j}=1}^{N}\sum_{\mathrm{i}=1}^{m} c_{\mathrm{j}}^{(\mathrm{i}),n}\left(\log c_{\mathrm{j}}^{(\mathrm{i}),n+1} - \log c_{\mathrm{j}}^{(\mathrm{i}),n}\right) = \Delta x \sum_{\mathrm{j}=1}^{N}\sum_{\mathrm{i}=1}^{m} c_{\mathrm{j}}^{(\mathrm{i}),n}\log\frac{c_{\mathrm{j}}^{(\mathrm{i}),n+1}}{c_{\mathrm{j}}^{(\mathrm{i}),n}} \\
\le& \Delta x \sum_{\mathrm{j}=1}^{N}\sum_{\mathrm{i}=1}^{m} c_{\mathrm{j}}^{(\mathrm{i}),n}\left(\frac{c_{\mathrm{j}}^{(\mathrm{i}),n+1}}{c_{\mathrm{j}}^{(\mathrm{i}),n}} - 1\right) = 0,
\end{aligned}
\tag{2.68}
$$

where $\log x \le x-1$ $(x>0)$ is used in the inequality and mass conservation is used in the last equality.

For part II,

$$
\begin{aligned}
II =& \Delta x \sum_{\mathrm{j}=1}^{N}\sum_{\mathrm{i}=1}^{m}\left(c_{\mathrm{j}}^{(\mathrm{i}),n+1}-c_{\mathrm{j}}^{(\mathrm{i}),n}\right)\left(\log c_{\mathrm{j}}^{(\mathrm{i}),n+1}-\log M_{\mathrm{j}}^{(\mathrm{i}),*}\right)=\Delta x\sum_{\mathrm{j}=1}^{N}\sum_{\mathrm{i}=1}^{m}\left(c_{\mathrm{j}}^{(\mathrm{i}),n+1}-c_{\mathrm{j}}^{(\mathrm{i}),n}\right)\log\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}}\\
=&\frac{\Delta t}{\Delta x}\sum_{\mathrm{j}=1}^{N-1}\sum_{\mathrm{i}=1}^{m}D_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})}\overline{M}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i}),*}\left[\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}+1}-\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}}\right]\log\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}}\\
&-\frac{\Delta t}{\Delta x}\sum_{\mathrm{j}=2}^{N}\sum_{\mathrm{i}=1}^{m}D_{\mathrm{j}-\frac{1}{2}}^{(\mathrm{i})}\overline{M}_{\mathrm{j}-\frac{1}{2}}^{(\mathrm{i}),*}\left[\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}}-\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}-1}\right]\log\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}}\\
=&\frac{\Delta t}{\Delta x}\sum_{\mathrm{j}=1}^{N-1}\sum_{\mathrm{i}=1}^{m}D_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})}\overline{M}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i}),*}\left[\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}+1}-\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}}\right]\log\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}}\\
&-\frac{\Delta t}{\Delta x}\sum_{\mathrm{j}=1}^{N-1}\sum_{\mathrm{i}=1}^{m}D_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})}\overline{M}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i}),*}\left[\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}+1}-\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}}\right]\log\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}+1}\\
=&-\frac{\Delta t}{\Delta x}\sum_{\mathrm{j}=1}^{N-1}\sum_{\mathrm{i}=1}^{m}D_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i})}\overline{M}_{\mathrm{j}+\frac{1}{2}}^{(\mathrm{i}),*}\left[\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}+1}-\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}}\right]\left[\log\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}+1}-\log\left(\frac{c^{(\mathrm{i}),n+1}}{M^{(\mathrm{i}),*}}\right)_{\mathrm{j}}\right]\\
\le&0,
\end{aligned}
\tag{2.69}
$$

where $M^{(\mathrm{i}),*}=\mathrm{e}^{-z_{\mathrm{j}}\psi_{\mathrm{j}}^{*}}$ is used in the first equality and the schemes (2.48)–(2.50) are used in the third equality. Using log is a non-decreasing function, we obtained the last inequality.

For part III,

$$\begin{aligned}
III =&\Delta x \sum_{\mathrm{j}=1}^{N}\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}\left[\frac{1}{2}\left(c_{\mathrm{j}}^{(\mathrm{i}),n+1}-c_{\mathrm{j}}^{(\mathrm{i}),n}\right)\left(\psi_{\mathrm{j}}^{n+1}+\psi_{\mathrm{j}}^{n}\right)+\left(c_{\mathrm{j}}^{(\mathrm{i}),n}-c_{\mathrm{j}}^{(\mathrm{i}),n+1}\right)\psi_{\mathrm{j}}^{*}\right]\\
&+\frac{\Delta x}{2}\sum_{\mathrm{j}=1}^{N}\left[\left(\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}c_{\mathrm{j}}^{(\mathrm{i}),n}+\rho_{\mathrm{j}}\right)\psi_{\mathrm{j}}^{n+1}-\left(\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}c_{\mathrm{j}}^{(\mathrm{i}),n+1}+\rho_{\mathrm{j}}\right)\psi_{\mathrm{j}}^{n}\right]\\
&+\frac{\epsilon_{\frac{1}{2}}f_a(\psi_1^{n+1}-\psi_1^{n})+\epsilon_{N+\frac{1}{2}}f_b(\psi_N^{n+1}-\psi_N^{n})}{\alpha\Delta x+2\beta}\\
=&\frac{\Delta x}{2}\sum_{\mathrm{j}=1}^{N}\left[\left(\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}c_{\mathrm{j}}^{(\mathrm{i}),n}+\rho_{\mathrm{j}}\right)\psi_{\mathrm{j}}^{n+1}-\left(\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}c_{\mathrm{j}}^{(\mathrm{i}),n+1}+\rho_{\mathrm{j}}\right)\psi_{\mathrm{j}}^{n}\right]\\
&+\frac{\epsilon_{\frac{1}{2}}f_a(\psi_1^{n+1}-\psi_1^{n})+\epsilon_{N+\frac{1}{2}}f_b(\psi_N^{n+1}-\psi_N^{n})}{\alpha\Delta x+2\beta}\\
=&-\frac{1}{2\Delta x}\sum_{\mathrm{j}=1}^{N}\left[\left(\epsilon_{\mathrm{j}+\frac{1}{2}}\hat{\psi}_{\mathrm{j}+\frac{1}{2}}^{n}-\epsilon_{\mathrm{j}-\frac{1}{2}}\hat{\psi}_{\mathrm{j}-\frac{1}{2}}^{n}\right)\psi_{\mathrm{j}}^{n+1}-\left(\epsilon_{\mathrm{j}+\frac{1}{2}}\hat{\psi}_{\mathrm{j}+\frac{1}{2}}^{n+1}-\epsilon_{\mathrm{j}-\frac{1}{2}}\hat{\psi}_{\mathrm{j}-\frac{1}{2}}^{n+1}\right)\psi_{\mathrm{j}}^{n}\right]\\
&+\frac{\epsilon_{\frac{1}{2}}f_a(\psi_1^{n+1}-\psi_1^{n})+\epsilon_{N+\frac{1}{2}}f_b(\psi_N^{n+1}-\psi_N^{n})}{\alpha\Delta x+2\beta}\\
=&0,
\end{aligned}\tag{2.70}$$

where $\psi_{\mathrm{j}}^{*}=\frac{\psi_{\mathrm{j}}^{n}+\psi_{\mathrm{j}}^{n+1}}{2}$ is used to obtain the second equality and the scheme (2.40) is used in the third one. The last equality is obtained by using the following formula. For a sequence $\{\phi_{\mathrm{j}}\}_{\mathrm{j}=1}^{N}$, one has

$$\begin{aligned}
&\sum_{\mathrm{j}=1}^{N}\phi_{\mathrm{j}}\left(\epsilon_{\mathrm{j}+\frac{1}{2}}\hat{\psi}_{\mathrm{j}+\frac{1}{2}}-\epsilon_{\mathrm{j}-\frac{1}{2}}\hat{\psi}_{\mathrm{j}-\frac{1}{2}}\right)=\left(\phi_N\epsilon_{N+\frac{1}{2}}\hat{\psi}_{N+\frac{1}{2}}-\phi_1\epsilon_{\frac{1}{2}}\hat{\psi}_{\frac{1}{2}}\right)-\sum_{\mathrm{j}=1}^{N-1}\epsilon_{\mathrm{j}+\frac{1}{2}}\hat{\psi}_{\mathrm{j}+\frac{1}{2}}(\phi_{\mathrm{j}+1}-\phi_{\mathrm{j}})\\
=&-\frac{2\alpha\Delta x}{\alpha\Delta x+2\beta}\left(\epsilon_{\frac{1}{2}}\phi_1\psi_1+\epsilon_{N+\frac{1}{2}}\phi_N\psi_N\right)-\sum_{\mathrm{j}=1}^{N-1}\epsilon_{\mathrm{j}+\frac{1}{2}}(\psi_{\mathrm{j}+1}-\psi_{\mathrm{j}})(\phi_{\mathrm{j}+1}-\phi_{\mathrm{j}})\\
&+\frac{2\Delta x}{\alpha\Delta x+2\beta}\left(\epsilon_{\frac{1}{2}}\phi_1 f_a+\epsilon_{N+\frac{1}{2}}\phi_N f_b\right),
\end{aligned}\tag{2.71}$$

where the summation by parts is used in the first equality; (2.41), (2.43) and (2.44) are used in the second equality.

Combing parts I, II, and III, the theorem is proved. □

Remark 3. *If one considers the fully implicit time discretization, i.e., $\psi_{\mathrm{j}}^{*} = \psi_{\mathrm{j}}^{n+1}$, a similar calculation as above would also show the energy decay property, but based on the additional assumption that α and β have the same sign.*

For instance, the energy estimate of the semi-discrete scheme cloud be presented as,

$$E(t_{n+1}) - E(t_n) \leq T_B - \Delta t \sum_{\mathrm{i}=1}^{m} \int_{\Omega} c^{(\mathrm{i}),n+1} \left| \nabla \left(\log c^{(\mathrm{i}),n+1} + z_{\mathrm{i}} \psi^{*} \right) \right|^2 \mathrm{d}\boldsymbol{x}, \tag{2.72}$$

where the boundary term is given as

$$T_B = \begin{cases} -\int_{\partial\Omega} \epsilon \dfrac{\beta}{2\alpha} \left(\dfrac{\partial \psi^{n+1}}{\partial \mathbf{n}} - \dfrac{\partial \psi^{n}}{\partial \mathbf{n}} \right)^2 \mathrm{d}\mathbf{s}, & \text{if } \alpha \neq 0, \\ -\int_{\partial\Omega} \epsilon \dfrac{\alpha}{2\beta} \left(\psi^{n+1} - \psi^{n} \right)^2 \mathrm{d}\mathbf{s}, & \text{if } \beta \neq 0. \end{cases} \tag{2.73}$$

This is exactly what proposed in [24]. We point out that α and β come from the physical boundary condition and their signs are not definite. Therefore, our semi-implicit time discretization is more general and works for a larger class of boundary conditions.

As a consequence of the fully discrete energy decay, we have the following

Theorem 2.4. (Steady-state preserving) *Assume the discrete energy $E_{\Delta}(t_n)$ is bounded from below, the fully discrete scheme (2.56) is steady-state preserving, i.e., for fixed Δx, when time step $n \to \infty$, the numerical solutions $c_{\mathrm{j}}^{(\mathrm{i}),\infty}$ and $\psi_{\mathrm{j}}^{\infty}$ become the (second order) numerical solutions to the limiting Poisson-Boltzmann equation*

$$\begin{cases} -(\epsilon \psi_x^{\infty})_x = \sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}} c^{(\mathrm{i}),\infty} + \rho, & x \in [a,b], \\ \alpha \psi^{\infty}(a) - \beta \psi_x^{\infty}(a) = f_a, \quad \alpha \psi^{\infty}(b) + \beta \psi_x^{\infty}(b) = f_b, \end{cases} \tag{2.74}$$

where

$$c^{(\mathrm{i}),\infty} = \lambda_{\mathrm{i}} \mathrm{e}^{-z_{\mathrm{i}} \psi^{\infty}}, \quad \lambda_{\mathrm{i}} = \frac{\int_a^b c^{(\mathrm{i}),0} \, \mathrm{d}x}{\int_a^b \mathrm{e}^{-z_{\mathrm{i}} \psi^{\infty}} \, \mathrm{d}x}. \tag{2.75}$$

Proof. Since the discrete energy sequence $\{E_\Delta(t_n)\}$ is monotonically decreasing and bounded from below, the limit $\lim_{n\to\infty} E_\Delta(t_n) = E_\Delta(t_\infty)$ exists. Taking $n \to \infty$ in (2.65), we have

$$c_{\mathrm{j}}^{(\mathrm{i}),\infty} = \lambda_{\mathrm{i}} M_{\mathrm{j}}^{(\mathrm{i}),\infty} = \lambda_{\mathrm{i}} \mathrm{e}^{-z_{\mathrm{i}}\psi_{\mathrm{j}}^{\infty}}, \quad \text{for all } \mathrm{i} = 1, \ldots, m,\ \mathrm{j} = 1, \ldots, N, \tag{2.76}$$

where λ_{i} is some constant depending only on i and can be obtained by

$$\lambda_{\mathrm{i}} = \frac{\sum_{\mathrm{j}=1}^{N} c_{\mathrm{j}}^{(\mathrm{i}),\infty}}{\sum_{\mathrm{j}=1}^{N} \mathrm{e}^{-z_{\mathrm{i}}\psi_{\mathrm{j}}^{\infty}}} = \frac{\sum_{\mathrm{j}=1}^{N} c_{\mathrm{j}}^{(\mathrm{i}),0}}{\sum_{\mathrm{j}=1}^{N} \mathrm{e}^{-z_{\mathrm{i}}\psi_{\mathrm{j}}^{\infty}}}, \tag{2.77}$$

where we used the mass conservation. Finally substituting $c_{\mathrm{j}}^{(\mathrm{i}),\infty}$ into the system (2.45), we have

$$P\boldsymbol{\Psi}^{\infty} = \mathbf{h}\left(\mathbf{c}^{(\mathrm{i}),\infty}\right), \tag{2.78}$$

which is a second order finite difference discretization to the limiting Poisson-Boltzmann equation (2.74). □

Fixed point iteration to solve the fully discrete scheme

The system (2.56) is implicit and fully coupled. To solve it, we propose a simple fixed point iteration. The following algorithm describes how the iterations are performed at time step t_n to compute the solutions $c_{\mathrm{j}}^{(\mathrm{i}),n+1}$ and ψ_{j}^{n+1} ($\mathrm{i} = 1, \ldots, m$; $\mathrm{j} = 1, \ldots, N$) at time step t_{n+1}.

Algorithm 1 Fixed point iteration to solve the system (2.56)

1: **procedure** Given $c_{\mathrm{j}}^{(\mathrm{i}),n}$, ψ_{j}^{n} ▷ concentration and potential at time t_n

2: $l = 0$, $c_{\mathrm{j}}^{(\mathrm{i}),(0)} \leftarrow c_{\mathrm{j}}^{(\mathrm{i}),n}$, $\psi_{\mathrm{j}}^{(0)} \leftarrow \psi_{\mathrm{j}}^{n}$. ▷ initial guess

3: Define

$$M_{\mathrm{j}}^{(\mathrm{i}),(l)} = \mathrm{e}^{-z_{\mathrm{i}}\psi_{\mathrm{j}}^{(l)}}, \quad \psi_{\mathrm{j}}^{(l)} = \frac{\psi_{\mathrm{j}}^{n} + \psi_{\mathrm{j}}^{(l)}}{2}, \tag{2.79}$$

and accordingly the matrix $A^{(\mathrm{i})}\left(\mathbf{M}^{(\mathrm{i}),(l)}\right)$. Solve the Nernst-Planck equation

$$A^{(\mathrm{i})}\left(\mathbf{M}^{(\mathrm{i}),(l)}\right)\mathbf{g}^{(\mathrm{i})}\left(\mathbf{c}^{(\mathrm{i}),(l+1)}, \mathbf{M}^{(\mathrm{i}),(l)}\right) = \mathbf{c}^{(\mathrm{i}),n} \tag{2.80}$$

to obtain $g_{\mathrm{j}}^{(\mathrm{i})}$. Then $c_{\mathrm{j}}^{(\mathrm{i}),(l+1)}$ is computed by

$$c_{\mathrm{j}}^{(\mathrm{i}),(l+1)} = g_{\mathrm{j}}^{(\mathrm{i})} M_{\mathrm{j}}^{(\mathrm{i}),(l)}. \tag{2.81}$$

4: Solve the Poisson equation

$$P\mathbf{\Psi}^{(l+1)} = \mathbf{h}\left(\mathbf{c}^{(\mathrm{i}),(l+1)}\right) \tag{2.82}$$

to obtain $\psi_{\mathrm{j}}^{(l+1)}$.

5: $l = l + 1$;

6: repeat Steps 3-5 until $\|c_{\mathrm{j}}^{(\mathrm{i}),(l+1)} - c_{\mathrm{j}}^{(\mathrm{i}),(l)}\| \leq \mathrm{tol}$ for all $1 \leq \mathrm{i} \leq m$.

7: **return** $c_{\mathrm{j}}^{(\mathrm{i}),n+1} \leftarrow c_{\mathrm{j}}^{(\mathrm{i}),(l+1)}$, $\psi_{\mathrm{j}}^{n+1} \leftarrow \psi_{\mathrm{j}}^{(l+1)}$. ▷ concentration and potential at time t_{n+1}

8: **end procedure**

Note that in each iteration, we need to solve two linear systems (2.80) and (2.82). Both of them can be solved efficiently using sparse linear solvers. Furthermore, the matrix $A^{(\mathrm{i})}\left(\mathbf{M}^{(\mathrm{i}),(l)}\right)$ is a M-matrix (by Lemma 2.2.1), hence the solution $c^{\mathrm{i},(l)}$ obtained in internal steps is guaranteed to be positive. For the Poisson equation, special care is needed for the Neumann boundary condition since the solution is unique up to a constant. Here we choose one solution by setting $\psi_1 = 0$.

The above fixed point iteration is just one strategy to solve the nonlinear system and our numerical experiments show that it generally converges in several steps (less than 10). One could also use Newton's method to achieve potentially faster convergence. We leave the convergence studies of different iterative methods to future work (see [45] for a related study). Nonetheless, to better understand the proposed time discretization, we do provide in this work a proof of the solvability of the semi-discrete scheme (2.28).

Solvability of the semi-discrete scheme

To prove the solvability of the semi-discrete scheme (2.28), we consider $D^{(\mathrm{i})} = \epsilon = 1$ for simplicity and rewrite it as

$$\begin{cases} \dfrac{c^{(\mathrm{i}),n+1} - c^{(\mathrm{i}),n}}{\Delta t} = \Delta c^{(\mathrm{i}),n+1} + \dfrac{1}{2}\nabla \cdot \left(z_{\mathrm{i}} c^{(\mathrm{i}),n+1} \nabla \left(\psi^n + \psi^{n+1} \right) \right), \\ -\Delta \psi^{n+1} = \displaystyle\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}} c^{(\mathrm{i}),n+1} + \rho. \end{cases} \tag{2.83}$$

The boundary condition is given as

$$\begin{cases} \left(\nabla c^{(\mathrm{i}),n+1} + \dfrac{1}{2} z_{\mathrm{i}} c^{(\mathrm{i}),n+1} \nabla \left(\psi^n + \psi^{n+1} \right) \right) \cdot \mathbf{n} = 0, \\ \nabla \psi^{n+1} \cdot \mathbf{n} = 0. \end{cases} \tag{2.84}$$

Note that the homogeneous Neumann boundary condition is assumed for the Poisson equation in our analysis, which is a bit less general than what we considered for the rest of the paper.

Definition 2.2.1. *Given* $\left(\{c^{(\mathrm{i}),n}\}_{\mathrm{i}=1}^m, \psi^n \right) \in H^1(\Omega)$, *we say that* $\left(\{c^{(\mathrm{i}),n+1}\}_{\mathrm{i}=1}^m, \psi^{n+1} \right) \in H^1(\Omega)$ *is a weak solution of (2.83)-(2.84), if it satisfies*

$$\begin{cases} \dfrac{1}{\Delta t} \displaystyle\int_\Omega \left(c^{(\mathrm{i}),n+1} - c^{(\mathrm{i}),n} \right) \phi \, \mathrm{d}\boldsymbol{x} + \int_\Omega \nabla c^{(\mathrm{i}),n+1} \cdot \nabla \phi \, \mathrm{d}\boldsymbol{x} = -\dfrac{1}{2} \displaystyle\int_\Omega z_{\mathrm{i}} c^{(\mathrm{i}),n+1} \nabla \left(\psi^n + \psi^{n+1} \right) \cdot \nabla \phi \, \mathrm{d}\boldsymbol{x}, \\ \displaystyle\int_\Omega \nabla \psi^{n+1} \cdot \nabla \phi \, \mathrm{d}\boldsymbol{x} = \int_\Omega \left(\sum_{\mathrm{i}=1}^m z_{\mathrm{i}} c^{(\mathrm{i}),n+1} + \rho \right) \phi \, \mathrm{d}\boldsymbol{x}, \end{cases} \tag{2.85}$$

for all test function $\phi \in H^1(\Omega)$.

We now state the solvability theorem for problem (2.83)-(2.84).

Theorem 2.5. *Let Ω be a bounded, open subset of $\mathbb{R}^d (d \leq 3)$, and $\partial\Omega$ is C^1. Then the semi-discrete scheme (2.83)-(2.84) has a weak solution $\left(\{c^{(\mathrm{i}),n+1}\}_{\mathrm{i}=1}^m, \psi^{n+1}\right)$, when Δt is sufficient small.*

The proof of this theorem is provided in the Appendix, which follows a similar line of the well-posedness theory for the PNP equations [16], [17], [46].

2.2.2 Fully discrete scheme in 2D

The extension of the 1D scheme to multi-D in the rectangular domain is straightforward. Here for completeness, we briefly present the scheme in 2D.

Consider the domain $\Omega = [a, b] \times [c, d]$, then the 2D PNP system reads

$$
\begin{cases}
\partial_t c^{(\mathrm{i})} = \left(D^{(\mathrm{i})} M^{(\mathrm{i})} \left(\dfrac{c^{(\mathrm{i})}}{M^{(\mathrm{i})}} \right)_x \right)_x + \left(D^{(\mathrm{i})} M^{(\mathrm{i})} \left(\dfrac{c^{(\mathrm{i})}}{M^{(\mathrm{i})}} \right)_y \right)_y, & (x,y) \in \Omega, t \geq 0, \quad (2.86) \\
c^{(\mathrm{i})}(0,x,y) = c^{(\mathrm{i}),0}(x,y), & (x,y) \in \Omega, \quad (2.87) \\
\left(\dfrac{c^{(\mathrm{i})}}{M^{(\mathrm{i})}} \right)_x (t,a,y) = \left(\dfrac{c^{(\mathrm{i})}}{M^{(\mathrm{i})}} \right)_x (t,b,y) = 0, & y \in [c,d], t \geq 0, \quad (2.88) \\
\left(\dfrac{c^{(\mathrm{i})}}{M^{(\mathrm{i})}} \right)_y (t,x,c) = \left(\dfrac{c^{(\mathrm{i})}}{M^{(\mathrm{i})}} \right)_y (t,x,d) = 0, & x \in [a,b], t \geq 0, \quad (2.89) \\
-(\epsilon\psi_x)_x - (\epsilon\psi_y)_y = \sum_{\mathrm{i}=1}^m z_{\mathrm{i}} c^{(\mathrm{i})} + \rho, & (x,y) \in \Omega, t \geq 0, \quad (2.90) \\
\alpha\psi(t,a,y) - \beta\psi_x(t,a,y) = f_a,\ \alpha\psi(t,b,y) + \beta\psi_x(t,b,y) = f_b, & y \in [c,d], t \geq 0, \quad (2.91) \\
\alpha\psi(t,x,c) - \beta\psi_y(t,x,c) = f_c,\ \alpha\psi(t,x,d) + \beta\psi_y(t,x,d) = f_d, & x \in [a,b], t \geq 0. \quad (2.92)
\end{cases}
$$

We partition Ω into N_x and N_y uniform cells in each dimension with mesh size $\Delta x = (b-a)/N_x, \Delta y = (d-c)/N_y$, respectively. The interior grid points are chosen as $(a + (\mathrm{j} - 1/2)\Delta x, c + (k - 1/2)\Delta y)$, $\mathrm{j} = 1, \ldots, N_x$, $k = 1, \ldots, N_y$, and the numerical approximation of a function $u(t,x,y)$ at this point and time step t_n is denoted by $u^n_{\mathrm{j},k}$. Cell interface values are defined similarly as in 1D.

The fully discrete scheme for the Nernst-Planck equation (2.86) is given as follows:

$$\begin{aligned}\frac{c_{\mathrm{j},k}^{(\mathrm{i}),n+1}-c_{\mathrm{j},k}^{(\mathrm{i}),n}}{\Delta t}=&\frac{1}{\Delta x^2}\left[D_{\mathrm{j}+\frac{1}{2},k}^{(\mathrm{i})}\overline{M}_{\mathrm{j}+\frac{1}{2},k}^{(\mathrm{i}),*}\hat{g}_{\mathrm{j}+\frac{1}{2},k}^{(\mathrm{i}),n+1}-D_{\mathrm{j}-\frac{1}{2},k}^{(\mathrm{i})}\overline{M}_{\mathrm{j}-\frac{1}{2},k}^{(\mathrm{i}),*}\hat{g}_{\mathrm{j}-\frac{1}{2},k}^{(\mathrm{i}),n+1}\right]\\&+\frac{1}{\Delta y^2}\left[D_{\mathrm{j},k+\frac{1}{2}}^{(\mathrm{i})}\overline{M}_{\mathrm{j},k+\frac{1}{2}}^{(\mathrm{i}),*}\hat{g}_{\mathrm{j},k+\frac{1}{2}}^{(\mathrm{i}),n+1}-D_{\mathrm{j},k-\frac{1}{2}}^{(\mathrm{i})}\overline{M}_{\mathrm{j},k-\frac{1}{2}}^{(\mathrm{i}),*}\hat{g}_{\mathrm{j},k-\frac{1}{2}}^{(\mathrm{i}),n+1}\right],\end{aligned}\tag{2.93}$$

where

$$\hat{g}_{\mathrm{j}+\frac{1}{2},k}^{(\mathrm{i}),n+1}=\frac{c_{\mathrm{j}+1,k}^{(\mathrm{i}),n+1}}{M_{\mathrm{j}+1,k}^{(\mathrm{i}),*}}-\frac{c_{\mathrm{j},k}^{(\mathrm{i}),n+1}}{M_{\mathrm{j},k}^{(\mathrm{i}),*}},\quad \hat{g}_{\mathrm{j},k+\frac{1}{2}}^{(\mathrm{i}),n+1}=\frac{c_{\mathrm{j},k+1}^{(\mathrm{i}),n+1}}{M_{\mathrm{j},k+1}^{(\mathrm{i}),*}}-\frac{c_{\mathrm{j},k}^{(\mathrm{i}),n+1}}{M_{\mathrm{j},k}^{(\mathrm{i}),*}},\tag{2.94}$$

$$\overline{M}_{\mathrm{j}+\frac{1}{2},k}^{(\mathrm{i}),*}=\frac{1}{2}\left(M_{\mathrm{j},k}^{(\mathrm{i}),*}+M_{\mathrm{j}+1,k}^{(\mathrm{i}),*}\right),\quad \overline{M}_{\mathrm{j},k+\frac{1}{2}}^{(\mathrm{i}),*}=\frac{1}{2}\left(M_{\mathrm{j},k}^{(\mathrm{i}),*}+M_{\mathrm{j},k+1}^{(\mathrm{i}),*}\right),\tag{2.95}$$

and

$$M_{\mathrm{j},k}^{(\mathrm{i}),*}=\mathrm{e}^{-z_{\mathrm{i}}\psi_{\mathrm{j},k}^{*}},\quad \psi_{\mathrm{j},k}^{*}=\frac{1}{2}\left(\psi_{\mathrm{j},k}^{n}+\psi_{\mathrm{j},k}^{n+1}\right).\tag{2.96}$$

At the boundary

$$\hat{g}_{\frac{1}{2},k}^{(\mathrm{i}),n+1}=\hat{g}_{N_x+\frac{1}{2},k}^{(\mathrm{i}),n+1}=0,\quad \hat{g}_{\mathrm{j},\frac{1}{2}}^{(\mathrm{i}),n+1}=\hat{g}_{\mathrm{j},N_y+\frac{1}{2}}^{(\mathrm{i}),n+1}=0.\tag{2.97}$$

For the Poisson equation (2.90), the scheme is given as

$$\begin{aligned}\sum_{\mathrm{i}=1}^{m}z_{\mathrm{i}}c_{\mathrm{j},k}^{(\mathrm{i}),n+1}+\rho_{\mathrm{j},k}=&-\frac{1}{\Delta x^2}\left[\epsilon_{\mathrm{j}-\frac{1}{2},k}\psi_{\mathrm{j}-1,k}^{n+1}-(\epsilon_{\mathrm{j}-\frac{1}{2},k}+\epsilon_{\mathrm{j}+\frac{1}{2},k})\psi_{\mathrm{j},k}^{n+1}+\epsilon_{\mathrm{j}+\frac{1}{2},k}\psi_{\mathrm{j}+1,k}^{n+1}\right]\\&-\frac{1}{\Delta y^2}\left[\epsilon_{\mathrm{j},k-\frac{1}{2}}\psi_{\mathrm{j},k-1}^{n+1}-(\epsilon_{\mathrm{j},k-\frac{1}{2}}+\epsilon_{\mathrm{j},k+\frac{1}{2}})\psi_{\mathrm{j},k}^{n+1}+\epsilon_{\mathrm{j},k+\frac{1}{2}}\psi_{\mathrm{j},k+1}^{n+1}\right],\end{aligned}\tag{2.98}$$

where the boundary terms are defined through

$$\alpha\frac{\psi^{n+1}_{1,k}+\psi^{n+1}_{0,k}}{2}-\beta\frac{\psi^{n+1}_{1,k}-\psi^{n+1}_{0,k}}{\Delta x}=f_a,\qquad \alpha\frac{\psi^{n+1}_{N_x+1,k}+\psi^{n+1}_{N_x,k}}{2}+\beta\frac{\psi^{n+1}_{N_x+1,k}-\psi^{n+1}_{N_x,k}}{\Delta x}=f_b, \tag{2.99}$$

$$\alpha\frac{\psi^{n+1}_{\mathrm{j},1}+\psi^{n+1}_{\mathrm{j},0}}{2}-\beta\frac{\psi^{n+1}_{\mathrm{j},1}-\psi^{n+1}_{\mathrm{j},0}}{\Delta y}=f_c,\qquad \alpha\frac{\psi^{n+1}_{\mathrm{j},N_y+1}+\psi^{n+1}_{\mathrm{j},N_y}}{2}+\beta\frac{\psi^{n+1}_{\mathrm{j},N_y+1}-\psi^{n+1}_{\mathrm{j},N_y}}{\Delta y}=f_d. \tag{2.100}$$

For the 2D scheme, we can also show the following properties: mass conservation, positivity preserving, and energy dissipation, which we give without proof.

Theorem 2.6. (Mass conservation) *The fully discrete scheme (2.93) (2.98) is always mass conservative:*

$$C^{(\mathrm{i})}_{\Delta}(t_n)=C^{(\mathrm{i})}_{\Delta}(t_{n+1}),\quad \mathrm{i}=1,\dots,m. \tag{2.101}$$

where

$$C^{(\mathrm{i})}_{\Delta}(t_n)=\Delta x\Delta y\sum_{\mathrm{j}=1}^{N_x}\sum_{k=1}^{N_y}c^{(\mathrm{i}),n}_{\mathrm{j},k} \tag{2.102}$$

is the total mass of the i*th ion species at* t_n*.*

Theorem 2.7. (Positivity preserving) *The fully discrete scheme (2.93) (2.98) is unconditionally positivity-preserving, i.e., if* $c^{(\mathrm{i}),n}_{\mathrm{j},k}\geq 0$ *for all* $\mathrm{j}=1,\dots,N_x,\ k=1,\dots,N_y$*, then*

$$c^{(\mathrm{i}),n+1}_{\mathrm{j},k}\geq 0,\quad \mathrm{j}=1,\dots,N_x,\ k=1,\dots,N_y, \tag{2.103}$$

for each $\mathrm{i}=1,\dots,m$*.*

Theorem 2.8. (Energy dissipation) *The fully discrete scheme (2.93) (2.98) is unconditionally energy-dissipative:*

$$E_{\Delta}(t_{n+1})\leq E_{\Delta}(t_n), \tag{2.104}$$

where the discrete free energy at t_n is defined as

$$\begin{aligned} E_\Delta(t_n) =& \Delta x \Delta y \sum_{\mathrm{i}=1}^{m} \sum_{\mathrm{j}=1}^{N_x} \sum_{k=1}^{N_y} c_{\mathrm{j},k}^{(\mathrm{i}),n} \log c_{\mathrm{j},k}^{(\mathrm{i}),n} + \frac{\Delta x \Delta y}{2} \sum_{\mathrm{j}=1}^{N_x} \sum_{k=1}^{N_y} \psi_{\mathrm{j},k}^n \left(\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}} c_{\mathrm{j},k}^{(\mathrm{i}),n} + \rho_{\mathrm{j},k} \right) \\ &+ \sum_{k=1}^{N_y} \frac{\epsilon_{\frac{1}{2},k} f_a \psi_{1,k}^n + \epsilon_{N_x+\frac{1}{2},k} f_b \psi_{N_x,k}^n}{\alpha \Delta x + 2\beta} + \sum_{\mathrm{j}=1}^{N_x} \frac{\epsilon_{\mathrm{j},\frac{1}{2}} f_c \psi_{\mathrm{j},1}^n + \epsilon_{\mathrm{j},N_y+\frac{1}{2}} f_d \psi_{\mathrm{j},N_y}^n}{\alpha \Delta y + 2\beta}. \end{aligned} \tag{2.105}$$

2.3 Numerical examples

In this section, we perform several numerical tests to demonstrate the convergence and properties of the proposed scheme. We will consider both 1D and 2D examples, and in particular, a practical example with physical parameters specifically suited toward the modeling of ion channels. The tolerance for fixed point iteration will be chosen as tol $= 10^{-8}$ for all the tests except the tolerance test in section 4.2.

2.3.1 Accuracy test: manufactured solution

We first examine the accuracy of our scheme using a manufactured solution. Consider the following 1D single-species PNP system with a source term

$$\begin{cases} \partial_t c = \partial_x \left(\partial_x c + c \partial_x \psi \right) + h, & x \in [0,1], \quad t \geq 0, \\ c(0,x) = x^2 (1-x)^2, & x \in [0,1], \\ -\partial_{xx} \psi = c, & x \in [0,1], \quad t \geq 0, \\ \psi(t,0) = 0, \quad \psi(t,1) = -\frac{1}{60} \mathrm{e}^{-t}, & t \geq 0, \\ \partial_x c + c \partial_x \psi = 0, & x = 0,1, \quad t \geq 0, \end{cases} \tag{2.106}$$

where h is given by

$$h(t,x) = \left(\frac{9}{5} x^8 - \frac{36}{5} x^7 + \frac{161}{15} x^6 - 7x^5 + \frac{5}{3} x^4 \right) \mathrm{e}^{-2t} - \left(x^4 - 2x^3 + 13x^2 - 12x + 2 \right) \mathrm{e}^{-t}.$$

For this system, one can construct the exact solution as

$$c(t,x) = x^2(1-x)^2\mathrm{e}^{-t}, \quad \psi(t,x) = -\left(\frac{1}{30}x^6 - \frac{1}{10}x^5 + \frac{1}{12}x^4\right)\mathrm{e}^{-t}. \tag{2.107}$$

We verify the order of the proposed scheme in both space and time. The results are shown in Table 2.1 and Table 2.2, where the errors of a numerical solution u_{j}^n is computed as

$$\| u^{\mathrm{num}} - u^{\mathrm{ext}} \|_{l^\infty} := \max_{\mathrm{j}} |u_{\mathrm{j}}^n - u^{\mathrm{ext}}(t_n, x_{\mathrm{j}})|, \quad \| u^{\mathrm{num}} - u^{\mathrm{ext}} \|_{l^2} := \left(\Delta x \sum_{\mathrm{j}} |u_{\mathrm{j}}^n - u^{\mathrm{ext}}(t_n, x_{\mathrm{j}})|^2\right)^{1/2} \tag{2.108}$$

at time $t_n = 0.5$. These results imply that our scheme can achieve the first order accuracy in time and the second order in space.

Table 2.1. Table of errors with different time step sizes Δt. This test is performed with fixed spatial mesh $\Delta x = 0.001$ and tolerance $\mathrm{tol} = 10^{-8}$.

error	$\|c^{\mathrm{error}}\|_{l^\infty}$	$\|c^{\mathrm{error}}\|_{l^2}$	$\|\psi^{\mathrm{error}}\|_{l^\infty}$	$\|\psi^{\mathrm{error}}\|_{l^2}$
$\Delta t = 1/10$	2.7880e-03	1.6698e-03	1.0106e-03	4.7973e-04
$\Delta t = 1/20$	1.3984e-03	8.3752e-04	5.0512e-04	2.3949e-04
$\Delta t = 1/40$	7.0048e-04	4.1952e-04	2.5254e-04	1.1965e-04
$\Delta t = 1/80$	3.5072e-04	2.1005e-04	1.2627e-04	5.9794e-05
$\Delta t = 1/160$	1.7564e-04	1.0519e-04	6.3133e-05	2.9880e-05

Table 2.2. Table of errors with different spatial mesh sizes Δx. This test is performed with fixed time step $\Delta t = 0.0001$ and tolerance $\mathrm{tol} = 10^{-8}$.

error	$\|c^{\mathrm{error}}\|_{l^\infty}$	$\|c^{\mathrm{error}}\|_{l^2}$	$\|\psi^{\mathrm{error}}\|_{l^\infty}$	$\|\psi^{\mathrm{error}}\|_{l^2}$
$\Delta x = 1/10$	4.1718e-03	3.9332e-03	5.3634e-04	3.9158e-04
$\Delta x = 1/20$	1.0469e-03	9.8417e-04	1.3417e-04	9.6947e-05
$\Delta x = 1/40$	2.6394e-04	2.4686e-04	3.3355e-05	2.3963e-05
$\Delta x = 1/80$	6.8095e-05	6.2541e-05	8.1313e-06	5.7674e-06
$\Delta x = 1/160$	1.9127e-05	1.6495e-05	1.8431e-06	1.2613e-06

2.3.2 1D multiple species

Next we apply our scheme to solve the 1D two-species PNP system (2.30)-(2.35) and verify its properties. Two different tests are performed:

Case 1) The Dirichlet boundary value problem in domain $[-1,1]$ with $D^{(1)} = D^{(2)} = \epsilon = 1$, $z_1 = 1, z_2 = -1, \rho = 0$, the initial and boundary conditions are chosen as

$$\begin{cases} c^{(1)}(0,x) = 2 - x^2, \quad c^{(2)}(0,x) = x^2, \\ \psi(t,-1) = -1, \quad \psi(t,1) = 1. \end{cases} \tag{2.109}$$

Case 2) The Neumann boundary value problem in domain $[0,1]$ with $D^{(1)} = D^{(2)} = \epsilon = 1$, $z_1 = 1, z_2 = -2, \rho = x$, the initial and boundary conditions are chosen as

$$\begin{cases} c^{(1)}(0,x) = 2 + x + \sin(2\pi x), \quad c^{(2)}(0,x) = 1 + x, \\ \partial_x \psi(t,0) = \partial_x \psi(t,1) = 0. \end{cases} \tag{2.110}$$

Figure 2.1 shows the time evolution of the ion concentrations $c^{(1)}$, $c^{(2)}$ and the electrostatic potential ψ. One can see that the proposed scheme works well with a large time step and spatial mesh size in both cases.

To verify the energy dissipation and mass conservation, we plot in Figure 2.2 the time evolution of the discrete free energy $E_\Delta(t_n)$ and the total mass $C_\Delta^{(1)}(t_n)$, $C_\Delta^{(2)}(t_n)$. Although not shown here, the positivity of the ion concentrations is also checked and no negative values are detected.

Next, we check how the tolerance threshold in the fixed point iteration will affect the accuracy. The time step and spatial mesh size are fixed in this test as above. The numerical solution with tolerance tol $= 10^{-10}$ is chosen as a reference solution. For Case 1), we check the maximum error for the ion concentrations and electrostatic potential $\{c^{(1)}, c^{(2)}, \psi\}$ at time $t = 00.2$. For Case 2), the numerical error is inspected at time $t = 0.01$. The results in Table 2.3 and Table 2.4 indicate that our scheme will achieve better accuracy with lower tolerance.

In [19], the exponential convergence towards the steady states was proved for the PNP system

$$W(t) \leq W(0)\mathrm{e}^{-\lambda t}, \qquad \text{for } \lambda = \lambda(\Omega), \tag{2.111}$$

where $W(t)$ is the entropy functional defined as $W(t) = \tilde{E}(t) - \tilde{E}(t_\infty)$. Here we try to verify such a property for our numerical solution. In Figure 2.3, the discrete entropy functional $W_\Delta(t_n) = E_\Delta(t_n) - E_\Delta(t_\infty)$ is plotted for both cases, where the exponential convergence is evident.

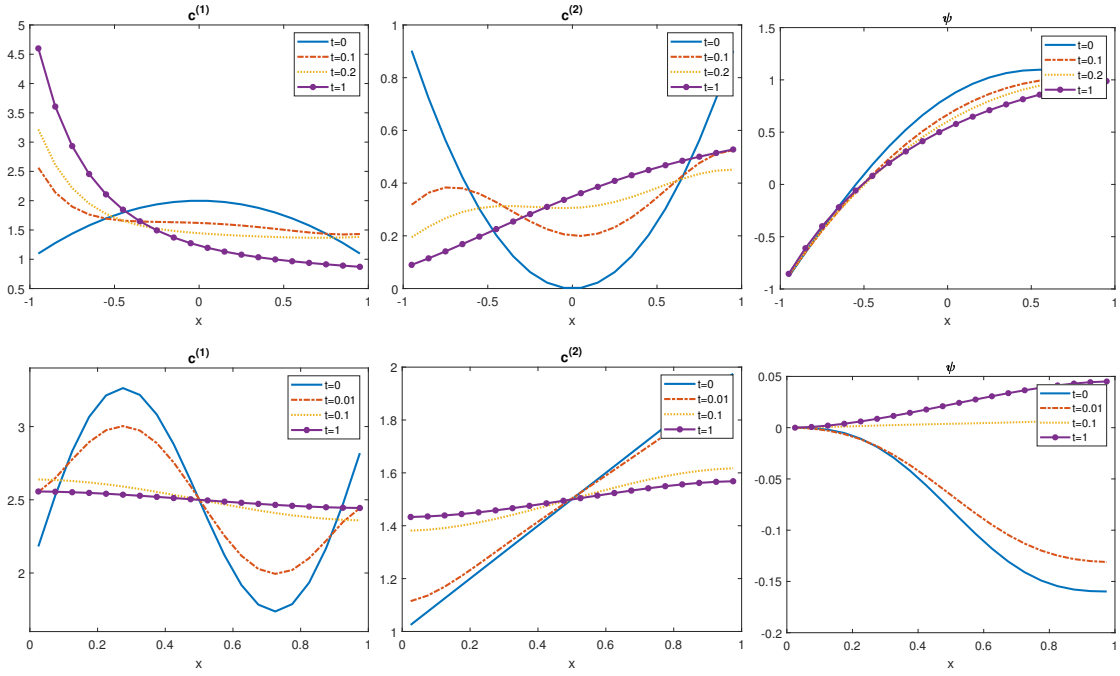

Figure 2.1. Time evolution of the ion concentrations $c^{(1)}$, $c^{(2)}$ and the potential ψ. Top row: Case 1). Bottom row: Case 2). Time step and spatial mesh size are chosen as $\Delta t = 0.05$(for Case 1), 0.01(for Case 2), $\Delta x = 0.05$.

Finally, to demonstrate the convergence of the fixed point iteration, we record the number of iterations at each time step in Figure 2.4. We can see that the method converges in less than 10 iterations, and this number decreases as the solution approaches the steady state.

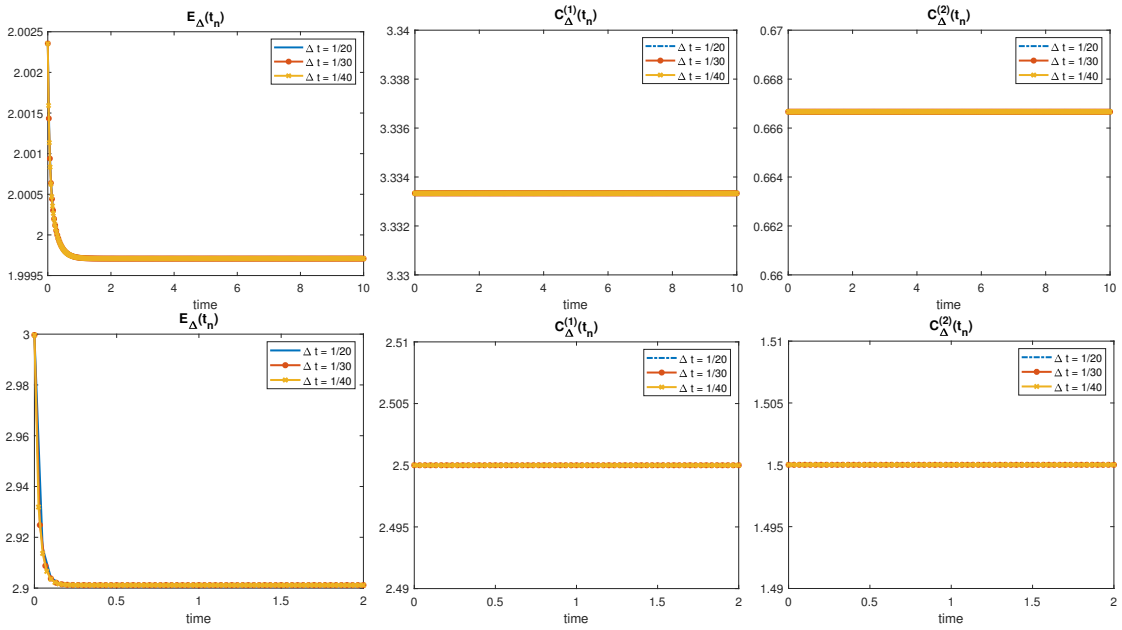

Figure 2.2. Time evolution of the discrete energy $E_\Delta(t_n)$ and the total mass $C_\Delta^{(1)}(t_n)$, $C_\Delta^{(2)}(t_n)$. Top row: Case 1). Bottom row: Case 2). Spatial mesh size is fixed at $\Delta x = 0.001$. Different time steps are chosen as indicated in the figures.

Table 2.3. Table of errors for Case 1) at time $t = 0.2$ with different tolerance $\text{tol} = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$. This test is performed with fixed time step $\Delta t = 0.05$ and spatial mesh size $\Delta x = 10^{-2}$.

error	$\|c^{(1)}\|_{l^\infty}$	$\|c^{(2)}\|_{l^\infty}$	$\|\psi\|_{l^\infty}$
$\text{tol} = 10^{-2}$	2.7232e-04	7.0131e-05	5.2871e-05
$\text{tol} = 10^{-3}$	2.0099e-05	3.5110e-06	3.5359e-06
$\text{tol} = 10^{-4}$	6.9042e-07	1.0994e-07	1.1688e-07
$\text{tol} = 10^{-5}$	3.8027e-07	6.2615e-08	5.6451e-08

Table 2.4. Table of errors for Case 2) at time $t = 0.1$ with different tolerance $\text{tol} = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$. This test is performed with fixed time step $\Delta t = 0.01$ and spatial mesh size $\Delta x = 10^{-2}$.

error	$\|c^{(1)}\|_{l^\infty}$	$\|c^{(2)}\|_{l^\infty}$	$\|\psi\|_{l^\infty}$
$\text{tol} = 10^{-2}$	1.5739e-04	1.9239e-04	1.0790e-04
$\text{tol} = 10^{-3}$	5.9846e-05	7.4411e-05	4.0508e-05
$\text{tol} = 10^{-4}$	5.4075e-06	6.8027e-06	3.7434e-06
$\text{tol} = 10^{-5}$	3.7434e-06	2.9617e-07	2.9617e-07

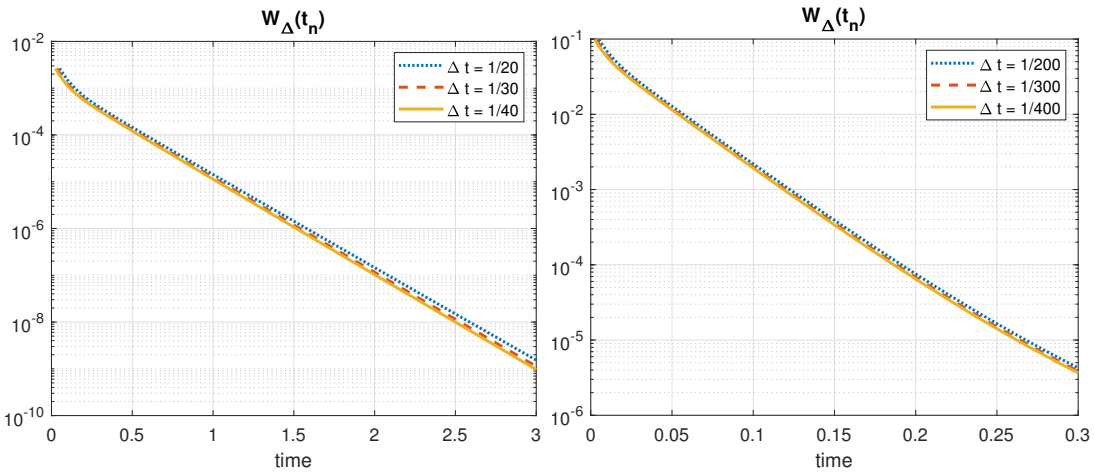

Figure 2.3. Time evolution of the discrete entropy functional $W_\Delta(t)$ in semi-log plot. Left: Case 1). Right: Case 2). Spatial mesh size is fixed at $\Delta x = 0.001$. Different time steps are chosen as indicated in the figures. For both cases, we consider the numerical solution at $t = 5$ as the steady state.

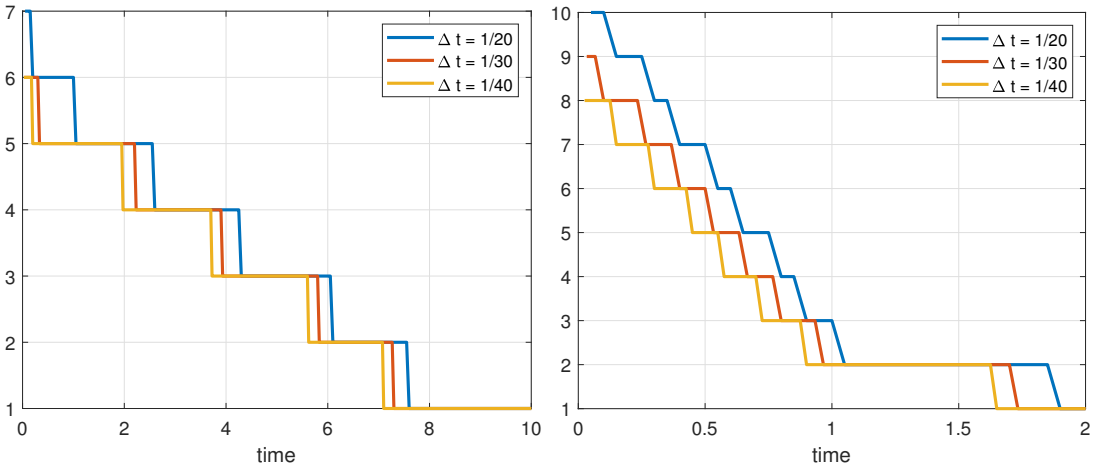

Figure 2.4. Number of fixed point iterations needed at each time step, the convergence tolerance is set as $\max_j |c_j^{(i),(l+1)} - c_j^{(i),(l)}| \leq 10^{-8}$). Left: Case 1). Right: Case 2). Spatial mesh size is fixed at $\Delta x = 0.001$. Different time steps are chosen as indicated in the figures.

2.3.3 2D single species

We now apply our scheme to solve the 2D single-species PNP system (2.86)-(2.92). Let $\Omega = [0,1] \times [0,1]$ be the computational domain and $D^{(1)} = \epsilon = z_1 = 1$, $\rho = 0$. Two different boundary and initial conditions are considered:

Case 1) $c(0,x,y) = 4,\ \alpha = 0,\ \beta = 1,\ f_a = f_b = f_c = f_d = -1;$

Case 2) $c(0,x,y) = 2,\ \alpha = 0,\ \beta = 1,\ f_a = f_b = -1,\ f_c = f_d = 0.$

Note that the compatibility condition is satisfied in both settings.

The time evolution of the ion concentration in both cases are shown Figure 2.5 and Figure 2.6, respectively. The energy dissipation is demonstrated in Figure 2.7. Finally, the positivity of the ion concentration is also checked and no negative values are detected.

2.3.4 KcsA model with Space-Dependent diffusion coefficients

In the ion channels, the values of the diffusion coefficients depend on the ion species and channels. They only affect the rate of evolution of the system. In this section, we apply our scheme on a simplified KcsA model with space-dependent diffusion coefficients [47] to verify the impact of diffusion coefficients.

We consider the KcsA model in domain $[-1,1]$ with $\epsilon = 1,\ z_1 = 1, z_2 = -1, \rho = 0$, the initial and boundary conditions are chosen as

$$\begin{cases} c^{(1)}(0,x) = 2 - x^2, \quad c^{(2)}(0,x) = x^2, \\ \psi(t,-1) = -1, \quad \psi(t,1) = 1. \end{cases} \tag{2.112}$$

Then we seperate the domain into three regions:

a) channel outside(CO): $0.7 \le |x| \le 1$;

b) selectivity filter (SF): $-0.1 < x < 0.7$;

c) intracellular (IC): $-0.7 < x < -0.1$.

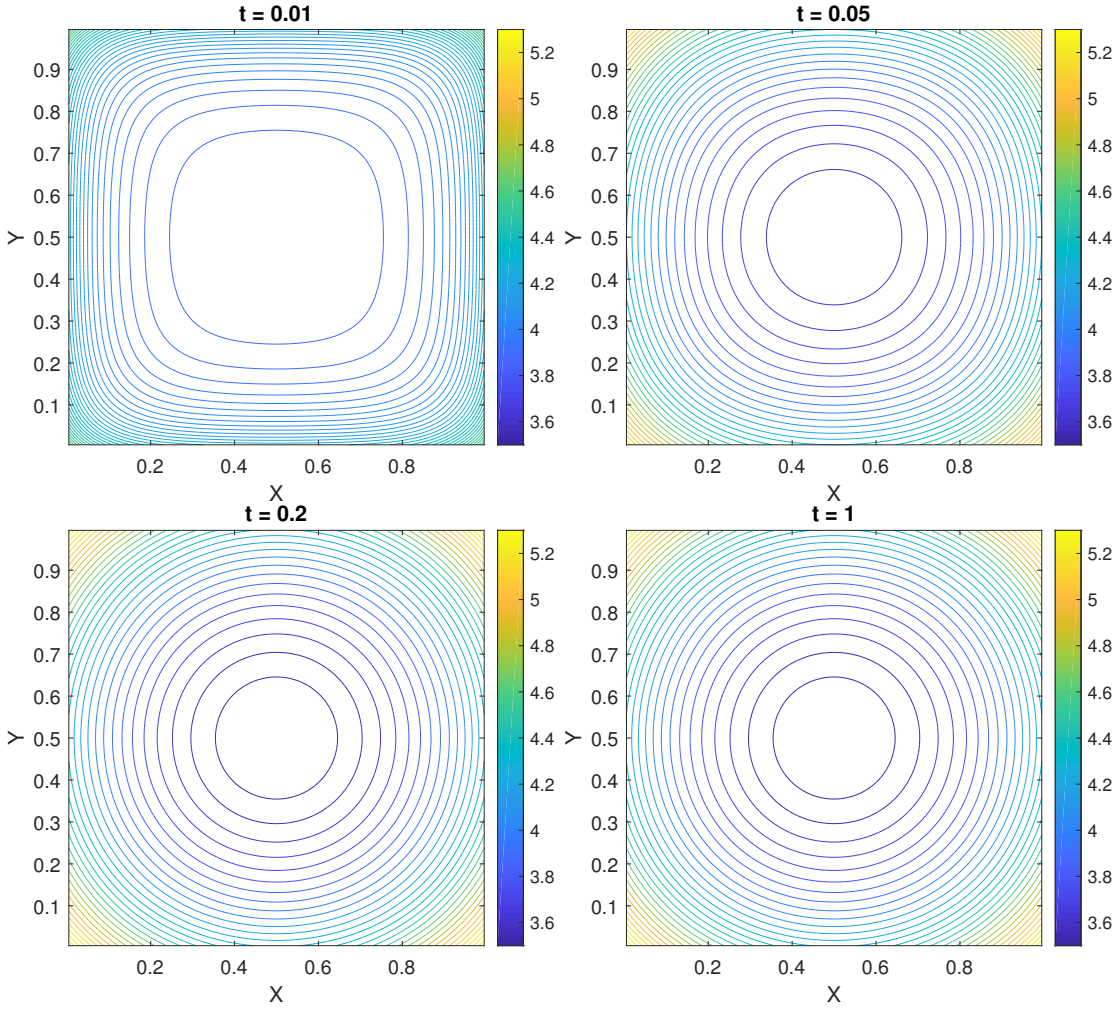

Figure 2.5. Case 1: Time evolution (contour plot) of the ion concentration c. Time step and spatial mesh size are chosen as $\Delta x = 0.01$ and $\Delta t = 0.01$.

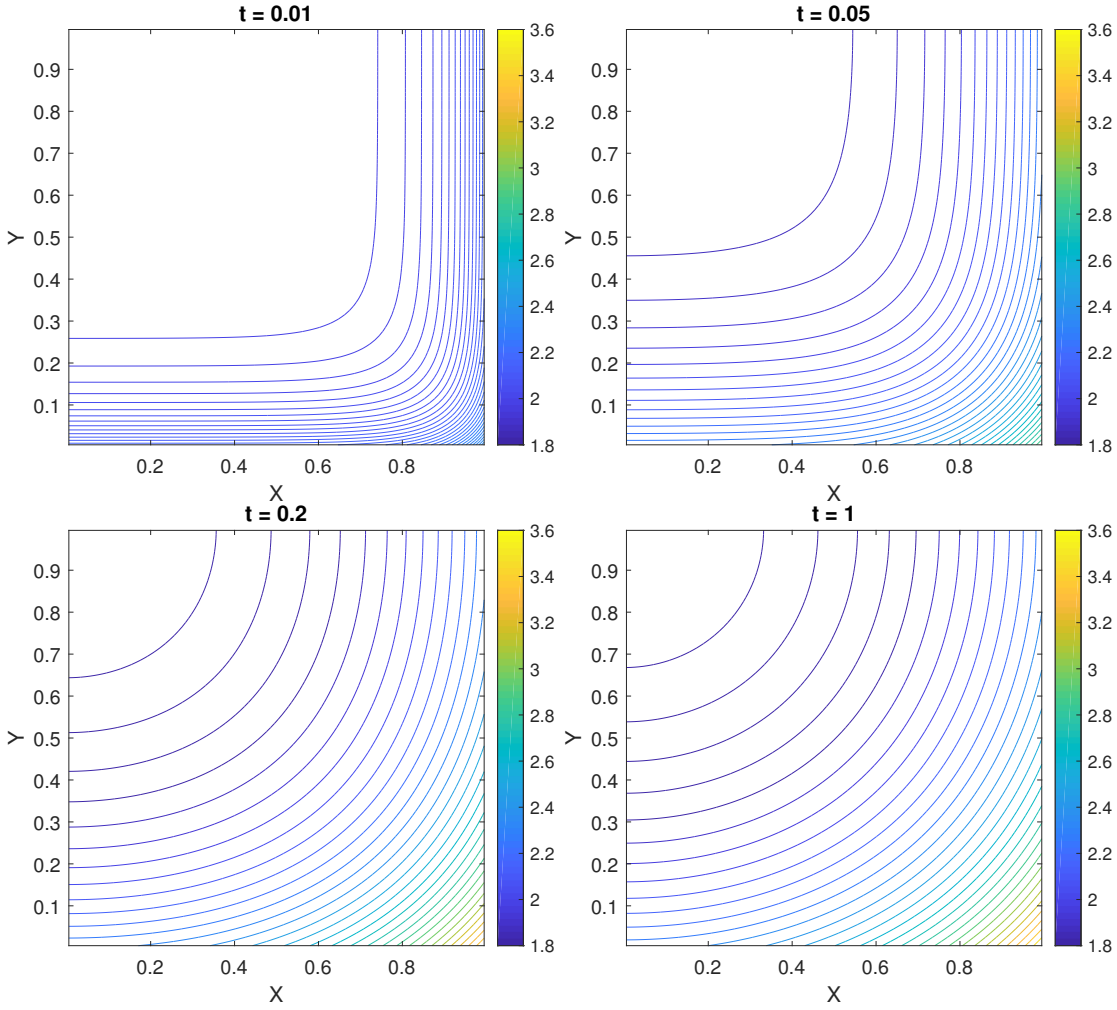

Figure 2.6. Case 2: Time evolution (contour plot) of the ion concentration c. Time step and spatial mesh size are chosen as $\Delta x = 0.01$ and $\Delta t = 0.01$.

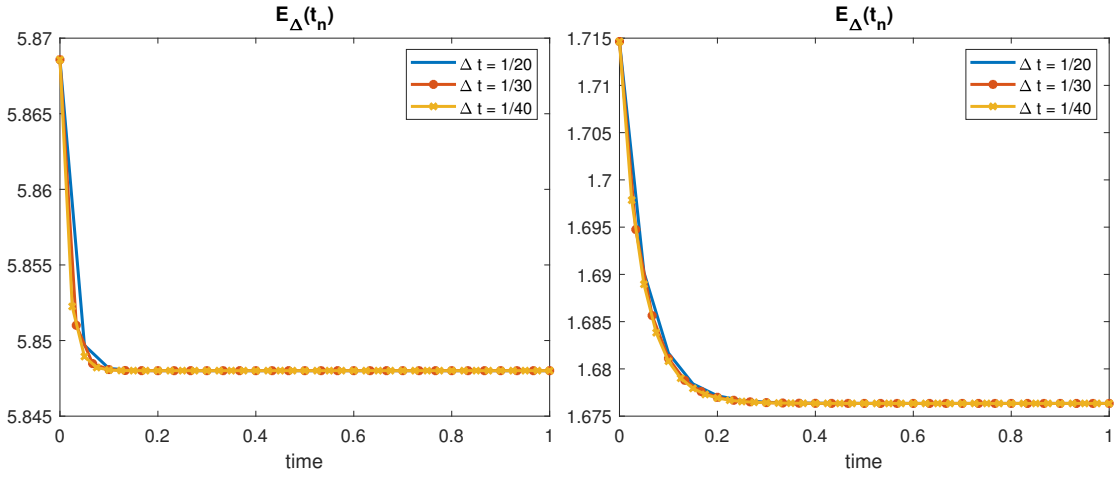

Figure 2.7. Time evolution of the discrete free energy $E_\Delta(t_n)$. Left: Case 1). Right: Case 2). Spatial mesh size is fixed at $\Delta x = 0.01$. Different time steps are chosen as indicated in the figures.

Outside the channel the diffusion coefficients are set as $D^{(1)} = D^{(2)} = 1$. For this model, we test three diffusion coefficient profiles:

i) $D^{(\mathrm{i})}$ is reduced to 40% in SF;

ii) $D^{(\mathrm{i})}$ is reduced to 40% in SF and 80% in IC;

iii) $D^{(\mathrm{i})}$ is reduced to 40% both in SF and IC.

The time evolution of the ion concentrations and energy are presented in Fig 2.8 and Fig 2.9. These three systems do converge to the same steady state at different rates.

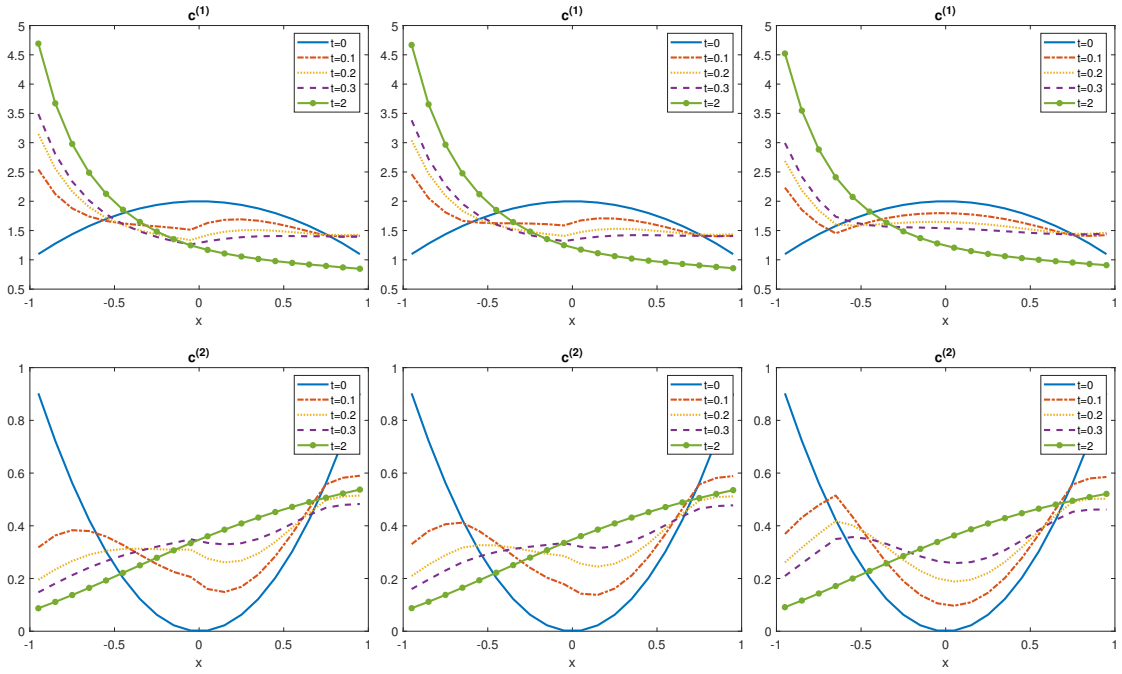

Figure 2.8. Time evolution of the ion concentrations (KcsA). First column: case i). Second row: case ii). Third column: case iii). Time step and spatial mesh size are chosen as $\Delta t = 0.05$ and $\Delta x = 0.05$.

2.3.5 Gouy-Chapman model

In this final test, we simulate the so-called Gouy-Chapman model widely used to describe the double layer structure in ion channels.

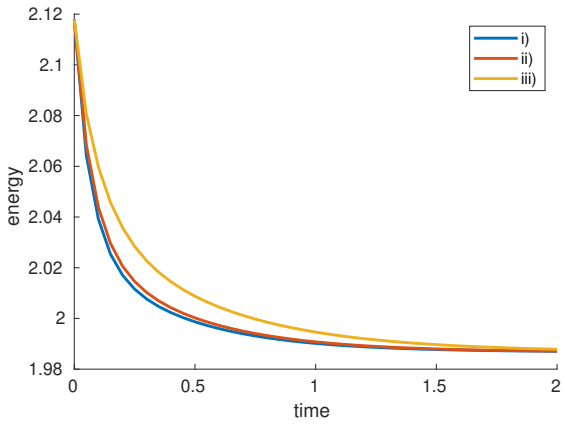

Figure 2.9. Time evolution of the energy (KcsA). Time step and spatial mesh size are chosen as $\Delta t = 0.05$ and $\Delta x = 0.05$.

We consider the 1D two-species PNP system (2.30)-(2.35) in domain $[-1, 1]$ with $D^{(1)} = D^{(2)} = \epsilon = 1$, $z_1 = 1$, $z_2 = -1$, and $\rho = 0$, which is taken from the work [22]. The dimensionless parameters χ_1 and χ_2 are chosen as $\chi_1 = 3.1, \chi_2 = 125.4$. A uniform initial condition is assumed $c^{(\mathrm{i})}(x, 0) = 1, \mathrm{i} = 1, 2$ for all $-1 \leq x \leq 1$ and the boundary condition for the Poisson equation is given by $\alpha = 1, \beta = 4.63 \times 10^{-5}, f_a = 1, f_b = -1$.

Figure 2.10 shows the time evolution of the ion concentrations and the electrostatic potential. Beginning with the linear profile, the electrostatic potential becomes zero in the bulk region (away from the boundary) and increases drastically in the diffuse layers (close to the boundary) at the steady state. Notice that the presence of diffuse layers requires a small spatial mesh size in numerical simulations. The solution will be far away from the thin layer solution if the mesh size is large, for example, $\Delta x > 0.05$.

2.4 Conclusion

We have introduced a semi-implicit finite difference scheme for the PNP equations in a bounded domain. A general boundary condition for the Poisson equation which includes (nonhomogeneous) Dirichlet, Neumann, and Robin boundaries as subcases were considered. The proposed scheme is first order in time and second order in space. The fully discrete scheme was proved to be mass conservative, unconditionally positive and energy dissipative (hence preserving the steady state). The solvability of the semi-discrete scheme was investigated and a fixed point iteration was proposed to solve the fully discrete scheme. Numerical examples were presented to demonstrate the accuracy and efficiency of the proposed scheme. Note that the fixed point iteration employed in this work is not necessarily the best method to solve the implicit scheme. We will investigate different iterative methods such as Newton's method in future work. Also, it would be interesting and challenging to develop a high order in time scheme which preserves the same properties as the first order one.

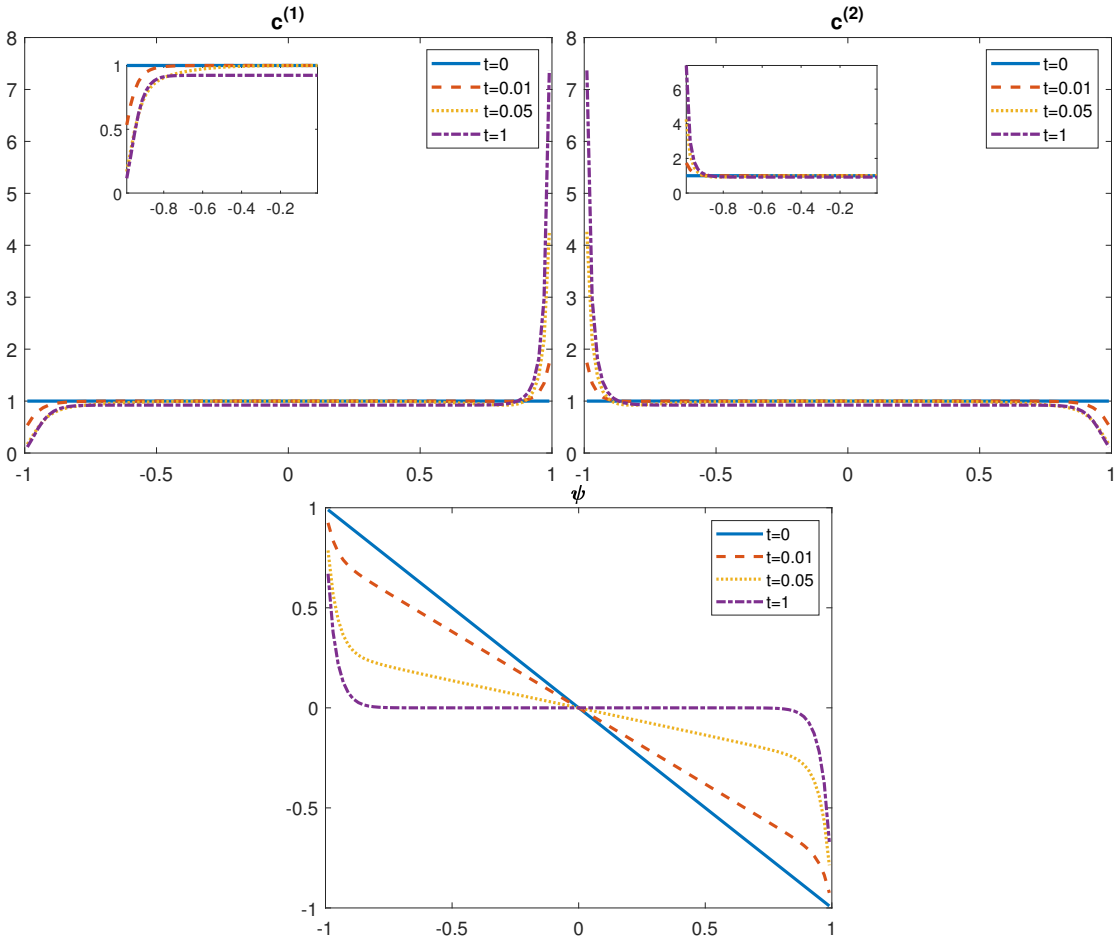

Figure 2.10. Time evolution of the ion concentrations and the electrostatic potential in Gouy-Chapman model. Time step and spatial mesh size are chosen as $\Delta t = 0.00125$ and $\Delta x = 0.02$.

3. A FAST FOURIER-GALERKIN SPECTRAL METHOD FOR BOLTZMANN EQUATION

In this chapter, we study the fast Fourier method proposed in [42] which uses a special approximation form of the weight term $G(l,m)$. In section 3.1, this fast algorithm and its limitation in the evaluation of the weight term will be discussed in detail. Following the similar idea, we propose a new approach using a different approximation form of the weight term $G(l,m)$ in section 3.2,. Numerical examples will be provided to verify the efficiency and accuracy of this new fast Fourier method in Section 3.3. This chapter is concluded in Section 3.4.

3.1 The fast Fourier spectral method for Boltzmann equation

For the Boltzmann equation:

$$\frac{\partial f}{\partial t} + \boldsymbol{v}\cdot\nabla_x f = \mathcal{Q}(f,f), \quad t>0,\ \boldsymbol{x},\boldsymbol{v}\in\mathbb{R}^3, \tag{3.1}$$

we consider the $\boldsymbol{\sigma}$-representation of the nonlinear Boltzmann collision operator

$$\mathcal{Q}(f,f)(\boldsymbol{v}) = \int_{\mathbb{R}^3}\int_{S^2} B(|\boldsymbol{v}-\boldsymbol{v}_*|, \boldsymbol{\sigma}\cdot(\widehat{\boldsymbol{v}-\boldsymbol{v}_*}))[f(\boldsymbol{v}')f(\boldsymbol{v}_*') - f(\boldsymbol{v})f(\boldsymbol{v}_*)]\,\mathrm{d}\boldsymbol{\sigma}\,\mathrm{d}\boldsymbol{v}_*. \tag{3.2}$$

Here the post-collision velocity $(\boldsymbol{v}', \boldsymbol{v}_*')$ can be expressed in terms of $(\boldsymbol{v}, \boldsymbol{v}_*)$:

$$\boldsymbol{v}' = \frac{\boldsymbol{v}+\boldsymbol{v}_*}{2} + \frac{|\boldsymbol{v}-\boldsymbol{v}_*|}{2}\boldsymbol{\sigma}, \quad \boldsymbol{v}_*' = \frac{\boldsymbol{v}+\boldsymbol{v}_*}{2} - \frac{|\boldsymbol{v}-\boldsymbol{v}_*|}{2}\boldsymbol{\sigma}, \tag{3.3}$$

where $\boldsymbol{\sigma}$ is the unit vector along the direction of $\boldsymbol{v}'-\boldsymbol{v}_*'$.

By a change of variable $\boldsymbol{g} = \boldsymbol{v}-\boldsymbol{v}_*$, the collision operator will be written as

$$\mathcal{Q}(f,f)(\boldsymbol{v}) = \int_{\mathbb{R}^3}\int_{S^2} B(|\boldsymbol{g}|, \boldsymbol{\sigma}\cdot\widehat{\boldsymbol{g}})[f(\boldsymbol{v}')f(\boldsymbol{v}_*') - f(\boldsymbol{v})f(\boldsymbol{v}-\boldsymbol{g})]\,\mathrm{d}\boldsymbol{\sigma}\,\mathrm{d}\boldsymbol{g}, \tag{3.4}$$

with

$$\boldsymbol{v}' = \boldsymbol{v} - \frac{\boldsymbol{g}}{2} + \frac{|\boldsymbol{g}|}{2}\boldsymbol{\sigma}, \quad \boldsymbol{v}_*' = \boldsymbol{v} - \frac{\boldsymbol{g}}{2} - \frac{|\boldsymbol{g}|}{2}\boldsymbol{\sigma}. \tag{3.5}$$

Assume that f has a compact support in velocity $\boldsymbol{v}$: $\mathrm{supp}_v f \approx \mathcal{B}_S$, where $\mathcal{B}_S$ is a ball centered at the origin with radius S. We truncate the infinite integral in $\boldsymbol{g}$ to a larger ball $\mathcal{B}_R$ with radius $R = 2S$:

$$\mathcal{Q}(f,f)(\boldsymbol{v}) \approx \mathcal{Q}_R(f,f)(\boldsymbol{v}) = \int_{\mathcal{B}_R}\int_{S^2} B(|\boldsymbol{g}|, \boldsymbol{\sigma}\cdot\widehat{\boldsymbol{g}})[f(\boldsymbol{v}')f(\boldsymbol{v}'_*) - f(\boldsymbol{v})f(\boldsymbol{v}-\boldsymbol{g})]\,\mathrm{d}\boldsymbol{\sigma}\,\mathrm{d}\boldsymbol{g}. \quad (3.6)$$

In the Fourier spectral method, we choose a computational domain as $\mathcal{D}_L = [-L, L]^3$, such that f is approximated by a truncated Fourier series:

$$f(\boldsymbol{v}) \approx f_N(\boldsymbol{v}) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{f}_k \mathrm{e}^{\mathbf{i}\frac{\pi}{L}k\cdot\boldsymbol{v}}, \quad \hat{f}_k = \frac{1}{(2L)^3}\int_{\mathcal{D}_L} f(\boldsymbol{v})\mathrm{e}^{-\mathbf{i}\frac{\pi}{L}k\cdot\boldsymbol{v}}\mathrm{d}\boldsymbol{v}, \quad (3.7)$$

where L is chosen as $L \geq \frac{3+\sqrt{2}}{2}S$ to avoid aliasing (see [48] for detail).

Substituting (3.7) into the collision operator (3.6), the k-th coefficient of the Fourier expansion is given as:

$$\begin{aligned}
\hat{\mathcal{Q}}_k &:= \frac{1}{(2L)^3}\int_{\mathcal{D}_L} \mathcal{Q}(f_N, f_N)(\boldsymbol{v})\mathrm{e}^{-\mathbf{i}\frac{\pi}{L}k\cdot\boldsymbol{v}}\mathrm{d}\boldsymbol{v} \\
&= \sum_{l,m=-\frac{N}{2}}^{\frac{N}{2}-1}\left(\frac{1}{(2L)^3}\int_{\mathcal{D}_L}\mathcal{Q}\left(\mathrm{e}^{\mathbf{i}\frac{\pi}{L}l\cdot\boldsymbol{v}}, \mathrm{e}^{\mathbf{i}\frac{\pi}{L}m\cdot\boldsymbol{v}}\right)\mathrm{e}^{-\mathbf{i}\frac{\pi}{L}k\cdot\boldsymbol{v}}\right)\hat{f}_l\hat{f}_m \\
&= \sum_{l+m=k}[G(l.m) - G(m,m)]\,\hat{f}_l\hat{f}_m, \quad (3.8)
\end{aligned}$$

for $-\frac{N}{2} \leq l, m, k \leq \frac{N}{2}-1$, and weight term $G(l,m)$ is given as

$$\begin{aligned}
G(l,m) &= \frac{1}{(2L)^3}\int_{\mathcal{D}_L}\mathcal{Q}\left(\mathrm{e}^{\mathbf{i}\frac{\pi}{L}l\cdot\boldsymbol{v}}, \mathrm{e}^{\mathbf{i}\frac{\pi}{L}m\cdot\boldsymbol{v}}\right)\mathrm{e}^{-\mathbf{i}\frac{\pi}{L}k\cdot\boldsymbol{v}} \\
&= \int_{\mathcal{B}_R}\int_{S^2} B(|\boldsymbol{g}|, \boldsymbol{\sigma}\cdot\widehat{\boldsymbol{g}})\mathrm{e}^{-\mathbf{i}\frac{\pi}{L}\frac{l+m}{2}\cdot\boldsymbol{g}}\mathrm{e}^{\mathbf{i}\frac{\pi}{L}|g|\frac{l-m}{2}\cdot\boldsymbol{\sigma}}\mathrm{d}\boldsymbol{\sigma}\,\mathrm{d}\boldsymbol{g}. \quad (3.9)
\end{aligned}$$

Let $\hat{\mathcal{Q}}_k = \hat{\mathcal{Q}}_k^+ - \hat{\mathcal{Q}}_k^-$, where the gain and loss part read as

$$\hat{\mathcal{Q}}_k^+ = \sum_{l+m=k} G(l,m)\hat{f}_l\hat{f}_m, \quad \hat{\mathcal{Q}}_k^- = \sum_{l+m=k} G(m,m)\hat{f}_l\hat{f}_m. \quad (3.10)$$

Since the loss term $\hat{\mathcal{Q}}_k^-$ is a convolution of functions $G(m,m)\hat{f}_m$ and $\hat{f}_l$, it can be computed via FFT in $O(N^3 \log N)$ operations.

In [42], an approximation form has been developed for the weight

$$G(l,m) \approx \sum_{p=1}^{N_p} \alpha_p(l+m)\beta_p(l)\gamma_p(m), \tag{3.11}$$

such that one can express the gain term $\hat{\mathcal{Q}}_k^+$ as a convolution

$$\hat{\mathcal{Q}}_k^+ \approx \sum_{p=1}^{N_p} \alpha_p(k) \sum_{l+m=k} \left(\beta_p(l)\hat{f}_l\right)\left(\gamma_p(m)\hat{f}_m\right). \tag{3.12}$$

The total cost is reduced to $O(N_p N^3 \log N)$ via FFT for a single evaluation of $\hat{\mathcal{Q}}_k^+$.

To get a suitable low-rank approximation (3.11), the weight $G(l,m)$ is written as

$$G(l,m) = \int_0^R \int_{S^2} F(l+m,\rho,\boldsymbol{\sigma}) \mathrm{e}^{\mathbf{i}\frac{\pi}{L}\rho\frac{l}{2}\cdot\boldsymbol{\sigma}} \mathrm{e}^{-\mathbf{i}\frac{\pi}{L}\rho\frac{m}{2}\cdot\boldsymbol{\sigma}} \mathrm{d}\boldsymbol{\sigma}\ \mathrm{d}\rho, \tag{3.13}$$

where

$$F(l+m,\rho,\boldsymbol{\sigma}) = \rho^2 \int_{S^2} B(\rho,\boldsymbol{\sigma}\cdot\hat{\boldsymbol{g}}) \mathrm{e}^{-\mathbf{i}\frac{\pi}{L}\rho\frac{l+m}{2}\cdot\hat{\boldsymbol{g}}}\ \mathrm{d}\hat{\boldsymbol{g}}. \tag{3.14}$$

The integral in the radial direction and sphere are computed using a numerical quadrature

$$G(l,m) \approx \sum_{\rho,\boldsymbol{\sigma}} w_\rho w_{\boldsymbol{\sigma}} F(l+m,\rho,\boldsymbol{\sigma}) \mathrm{e}^{\mathbf{i}\frac{\pi}{L}\rho\frac{l-m}{2}\cdot\boldsymbol{\sigma}}, \tag{3.15}$$

where ρ and $\boldsymbol{\sigma}$ are the Gauss-Legendre and spherical quadrature point, w_ρ and $w_{\boldsymbol{\sigma}}$ represent the corresponding quadrature weights.

In the radial direction, the number (N_ρ) of quadrature points in ρ must be $O(N)$ because of the oscillation of the integrand. For the integration on the sphere, we use $N_{\boldsymbol{\sigma}}$ quadrature points, which is much less than N^2 (confirmed by numerical tests). Therefore, we have $N_p = N_\rho N_{\boldsymbol{\sigma}} \ll N^3$, so that the total cost will be $O(N_p N^3 \log N) \ll O(N^6)$.

Eventually, we get the fast approximation for gain term

$$\widehat{\mathcal{Q}}_k^+ \approx \sum_{\rho,\boldsymbol{\sigma}} \omega_\rho \omega_{\boldsymbol{\sigma}} F(k,\rho,\boldsymbol{\sigma}) \sum_{\substack{l,m=-\frac{N}{2} \\ l+m=k}}^{\frac{N}{2}-1} \left[\mathrm{e}^{\mathrm{i}\frac{\pi}{L}\rho\frac{l}{2}\cdot\boldsymbol{\sigma}} \hat{f}_l\right] \left[\mathrm{e}^{-\mathrm{i}\frac{\pi}{L}\rho\frac{m}{2}\cdot\boldsymbol{\sigma}} \hat{f}_m\right], \tag{3.16}$$

and loss term

$$\hat{\mathcal{Q}}_k^- = \sum_{l+m=k} G(m,m)\hat{f}_l\hat{f}_m. \tag{3.17}$$

3.1.1 Limitation of the current algorithm

We consider the model of 3D VHS molecules where $B = |\boldsymbol{g}|^\gamma$ $(0 \le \gamma \le 1)$. The weight $G(l,m)$ is given in a simple form

$$G(l,m) = 16\boldsymbol{\pi}^2 \int_0^R \rho^{\gamma+2} \operatorname{sinc}\left(\frac{\boldsymbol{\pi}}{L}\rho\frac{|l+m|}{2}\right) \operatorname{sinc}\left(\frac{\boldsymbol{\pi}}{L}\rho\frac{|l-m|}{2}\right) \mathrm{d}\rho, \tag{3.18}$$

where $\operatorname{sinc}(x) = \frac{\sin(x)}{x}$.

In the fast algorithm above, we approximate $\operatorname{sinc}\left(\frac{\pi}{L}\rho\frac{|l-m|}{2}\right)$ with Lebedev quadrature

$$G(l,m) \approx \int_0^R F(l+m,\rho) \sum_{\mathrm{j}=1}^{N_{\boldsymbol{\sigma}}} \omega_{\mathrm{j}}^{(\boldsymbol{\sigma})} \exp(\mathbf{i}\frac{\boldsymbol{\pi}}{L}\rho\frac{l}{2}\cdot\boldsymbol{\sigma}_{\mathrm{j}}) \exp(\mathbf{i}\frac{\boldsymbol{\pi}}{L}\rho\frac{-m}{2}\cdot\boldsymbol{\sigma}_{\mathrm{j}})\mathrm{d}\rho, \tag{3.19}$$

where

$$F(l+m,\rho) = 4\boldsymbol{\pi}\rho^{\gamma+2} \operatorname{sinc}\left(\frac{\boldsymbol{\pi}}{L}\rho\frac{|l+m|}{2}\right). \tag{3.20}$$

However, there is some limitation to this approach. Due to the oscillatory nature of the integrand, one might need a large number of quadrature points to approximate the weight $G(l,m)$ accurately.

In the following implementation, we compute the weights $G(l,m)$ for all index in two approaches,

- the direct Fourier method, G^{direct} is evaluated via (3.18) with $N_\rho = N$ quadrature points in radial direction.

- the fast Fourier method, G^{fast} is evaluated via (3.19) with $N_\rho = N$ quadrature points in radial direction and $N_{\boldsymbol{\sigma}}$ Lebedev quadrature points on the unit sphere.

The parameters in Fourier spectral method are chosen as $\gamma = 0$, $S = 5.0$, $R = 2S$ and $L = \frac{3+\sqrt{2}}{2}S$. The relative error is defined as

$$\frac{\| G^{\text{direct}} - G^{\text{fast}} \|_\infty}{\| G^{\text{direct}} \|_\infty}. \tag{3.21}$$

The following table shows that it is hard to get a good approximation to the weight $G(l,m)$ when a small number of spherical quadrature points ($N_{\boldsymbol{\sigma}}$) are applied in the fast algorithm.

Table 3.1. Relative error in l_∞ norm for $N_{\boldsymbol{\sigma}} = 14, 38, 74$, here we choose $N = 16$.

Quadrature pts N_σ	14	38	74
Relative error	0.5254	0.3015	0.2649

3.2 The new approach for fast algorithm

In this section, we propose a different decomposition for $\text{sinc}\left(\frac{\pi}{L}\rho\frac{|l-m|}{2}\right)$ which leads to a different low-rank approximation to $G(l,m)$ in the form of (3.11).

First, by change of variable, we consider

$$u(s) := \text{sinc}(\sqrt{s}) = \text{sinc}\left(\frac{\boldsymbol{\pi}}{L}\rho\frac{|l-m|}{2}\right), \tag{3.22}$$

where s is defined as

$$s = \frac{\rho^2\boldsymbol{\pi}^2}{4L^2}|l-m|^2 \in [0, P_N], \ P_N = \frac{R^2\boldsymbol{\pi}^2}{4L^2}dN^2. \tag{3.23}$$

For example, if the parameters are chosen as $S = 5.0$, $R = 2S$ and $L = \frac{3+\sqrt{2}}{2}S$, we will get $P_N \approx 6224.07$. The profile of $u(s)$ function is presented in Fig 3.1. One can see that $u(s)$ function has strong oscillations close to the origin. To guarantee the accuracy in the evalua-

tion of collision kernel, one need to approximat $u(s)$ well for small $|s|$, which corresponding to sinc $\left(\frac{\pi}{L}\rho\frac{|l-m|}{2}\right)$ in low frequency region.

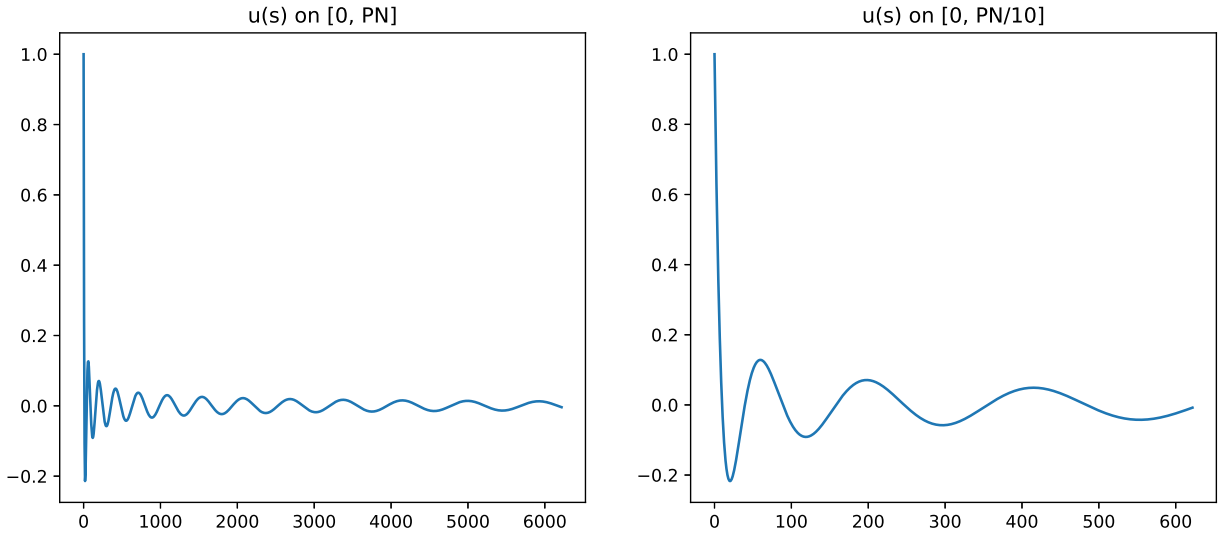

Figure 3.1. Left: $u(s)$ function on $[0, P_N]$; Right: $u(s)$ on $[0, P_N/10]$.

So that, an extension of $u(s)$ function is needed to over come the numerical challenge caused by the oscillations. Notice that hyperbolic sine function satisfies that

$$\sinh(z) = -\mathbf{i}\sin(\mathbf{i}z). \tag{3.24}$$

Then, for $\{s < 0 \mid s \in \mathbb{R}\} \subset \mathbb{C}$, we define the extension function $\widetilde{u}(s)$ as

$$\widetilde{u}(s) = \operatorname{sinc}(\sqrt{s}) = \frac{\sin(\sqrt{s})}{\sqrt{s}} = \frac{\sin(\mathbf{i}\sqrt{|s|})}{\mathbf{i}\sqrt{|s|}} = \frac{\sinh(\sqrt{|s|})}{\sqrt{|s|}}. \tag{3.25}$$

We extend $u(s)$ to the domain $[a, b]$ and define the extension as

$$u(s) \approx \widetilde{u}(s) := \begin{cases} \frac{\sin(\sqrt{s})}{\sqrt{s}}, & 0 \le s \le b, \\ \frac{\sinh(\sqrt{|s|})}{\sqrt{|s|}}, & a \le s < 0, \end{cases} \tag{3.26}$$

which will guarantee the continuity at the origin. The range of domain $[a, b]$ will be decided later.

We wish the target function will have some periodicity on domain $[a, b]$ since it will be approximated by Fourier basis. Therefore, we multiply $\widetilde{u}$ with a damping function and define the target function as

$$u^{\text{new}}(s) := f^{\text{damp}}(s) \times \widetilde{u}(s), \quad a \leq s \leq b, \tag{3.27}$$

where

$$f^{\text{damp}}(s) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{s-\mu}{50}\right)\right]\frac{1}{2}\left[1 + \text{erf}\left(\frac{\nu - s}{50}\right)\right], \tag{3.28}$$

with μ, ν are the parameters to be decided.

For example, let the domain $[a, b]$ and parameters μ, ν to be

$$\begin{cases} a = -320, \quad b = P_N \approx 6224.07, \\ \mu = -120, \quad \nu = 5000. \end{cases}$$

The profile of $\widetilde{u}$, f^{damp} and u^{new} functions are presented in Fig 3.2.

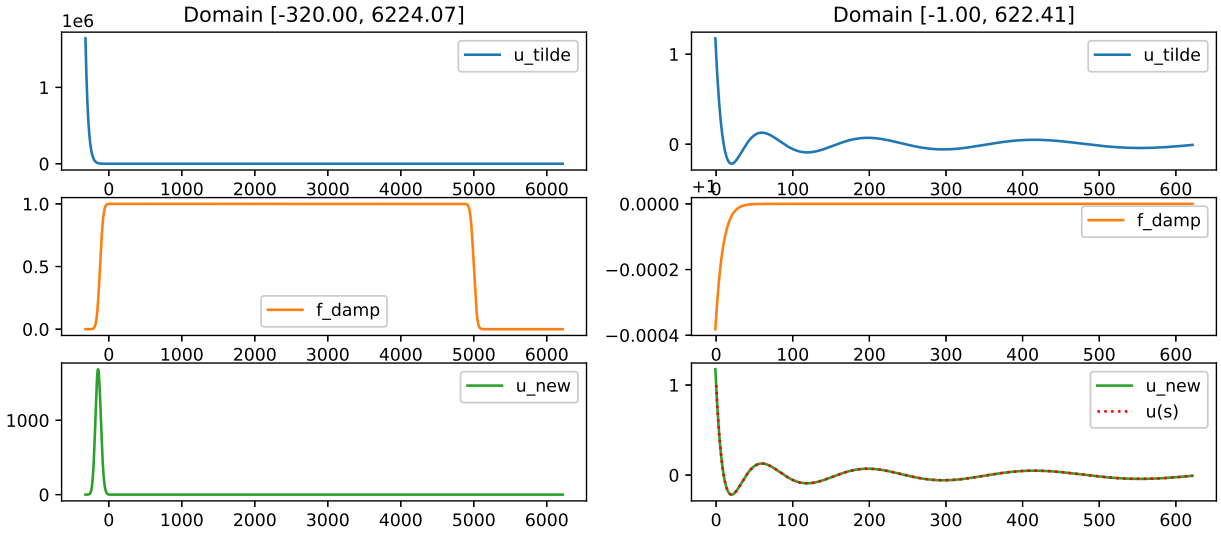

Figure 3.2. Left: profile of $\widetilde{u}$, f^{damp} and u^{new} functions on the whole domain $[a, b]$; Right: profile of functions on $[-1, P_N/10]$ which is close to the origin.

Finally, we consider the Fourier expansion of $u^{\text{new}}(s)$ on the interval $[a,b]$,

$$u^{\text{new}}(s) \approx \sum_{n=-M/2}^{M/2-1} \widehat{u}_n \cdot \mathrm{e}^{\mathbf{i}2\pi n\frac{s-a}{b-a}}, \quad \widehat{u}_n := \frac{1}{b-a}\int_a^b u(t)\cdot \mathrm{e}^{-\mathbf{i}\frac{2\pi n}{b-a}t}\,\mathrm{d}t. \tag{3.29}$$

Those coefficients will be computed by FFT in the numerical implementation.

Using the identity

$$|l-m|^2 = 2l^2 + 2m^2 - |l+m|^2, \tag{3.30}$$

we can get the decomposition of n-th Fourier basis

$$\begin{aligned} \mathrm{e}^{\mathbf{i}2\pi n\frac{s-a}{b-a}} &= \mathrm{e}^{\mathbf{i}2\pi n\frac{-a}{b-a}}\exp(\mathbf{i}\frac{2\pi n}{b-a}\frac{\rho^2\pi^2}{4L^2}|l-m|^2) \\ &= \mathrm{e}^{\mathbf{i}2\pi n\frac{-a}{b-a}}\frac{[h_n(\rho,|l|)h_n(\rho,|m|)]^2}{h_n(\rho,|l+m|)}, \end{aligned} \tag{3.31}$$

where h_n is defined as

$$h_n(\rho,|k|) \;:=\; \exp(\mathbf{i}\frac{\pi^3 n\rho^2}{2L^2(b-a)}|k|^2), \tag{3.32}$$

for $0\le\rho\le R, -\frac{M}{2}\le n\le\frac{M}{2}-1$ and $-\frac{N}{2}\le k\le\frac{N}{2}-1$.

To summarize, the sinc term will be numerically approximated as follows

$$\begin{aligned} \mathrm{sinc}\left(\frac{\pi}{L}\rho\frac{|l-m|}{2}\right) &\xrightarrow{\text{Change of variable (3.22)}} && u(s)=\mathrm{sinc}(\sqrt{s}), && s\in[0,P_N] \\ &\xrightarrow{\text{Domain extension (3.26)}} && u(s)\approx\widetilde{u}(s), && s\in[a,b], \\ &\xrightarrow{\text{Damping function (3.27)}} && \widetilde{u}(s)\approx u^{\text{new}}(s), && s\in[a,b], \\ &\xrightarrow{\text{Fourier approx (3.29)}} && u^{\text{new}}(s)\approx\textstyle\sum\widehat{u}_n\cdot\mathrm{e}^{\mathbf{i}2\pi n\frac{s-a}{b-a}}, && s\in[a,b]. \end{aligned}$$

Therefore, we get the expansion of sinc term as

$$\mathrm{sinc}\left(\frac{\pi}{L}\rho\frac{|l-m|}{2}\right) \approx \sum_{n=-M/2}^{M/2-1}\widehat{u}_n\cdot\mathrm{e}^{\mathbf{i}2\pi n\frac{-a}{b-a}}\frac{[h_n(\rho,|l|)h_n(\rho,|m|)]^2}{h_n(\rho,|l+m|)}. \tag{3.33}$$

In the new approach, the weight $G(l,m)$ will be approximated by

$$\begin{aligned} & G(l,m) \\ \approx \; & 16\pi^2 \sum_{n=-M/2}^{M/2-1} \widehat{u}_n \mathrm{e}^{\mathbf{i}2\pi n\frac{-a}{b-a}} \int_0^R \rho^{\gamma+2} \frac{\operatorname{sinc}\left(\frac{\pi}{L}\rho\frac{|l+m|}{2}\right)}{h_n(\rho,|l+m|)} \left[h_n(\rho,|l|)h_n(\rho,|m|)\right]^2 \mathrm{d}\rho. \end{aligned} \tag{3.34}$$

where ρ_{i} is the Gauss-Legendre quadrature points on $[0,R]$ and $\omega_{\mathrm{i}}^{(\rho)}$ represents the corresponding quadrature weights.

Eventually, the fast approximation for gain term reads as

$$\widehat{\mathcal{Q}}_k^+ \approx \sum_{\rho,n} \omega_\rho F_n'(k,\rho) \sum_{\substack{l,m=-\frac{N}{2} \\ l+m=k}}^{\frac{N}{2}-1} \left(\left[h_n(\rho,|l|)\right]^2 \hat{f}_l\right) \left(\left[h_n(\rho,|m|)\right]^2 \hat{f}_m\right), \tag{3.35}$$

where

$$F_n'(k,\rho) = 16\pi^2 \widehat{u}_n \rho^{\gamma+2} \frac{\operatorname{sinc}\left(\frac{\pi}{L}\rho\frac{|k|}{2}\right)}{h_n(\rho,|k|)} \mathrm{e}^{\mathbf{i}2\pi n\frac{-a}{b-a}}. \tag{3.36}$$

In this new approach, we use M Fourier basis in the approximation of u^{new} function, which is less than N^2. Therefore, we have $N_p' = N_\rho M \ll N^3$, so that the total cost will be $O(N_p' N^3 \log N) \ll O(N^6)$, which is less than the direct Fourier method.

3.2.1 The parameters (a,b,μ,ν) in new method

One should notice that these parameters need to be treated carefully in the new approach:

1. **The domain $[a,b]$ of extension function $\widetilde{u}$:** First, it must cover $[0,P_N]$. This interval should not be too large nor too small. The larger $[a,b]$ is, the more Fourier bases are required to ensure a good approximation of u^{new} function. If domain $[a,b]$ is very small, meaning that a is close to 0, it would lead to difficulties in treating the oscillations close to the origin.

2. **The parameters μ,ν in the damping function f^{damp}:** Notice that the f^{damp} function is used to guarantee the periodicity of u^{new} on interval $[a,b]$. That means μ

and ν cannot be too close to the boundary a and b so that f^{damp} and u^{new} will vanish on the boundary. We also notice that u^{new} has a large peak on $[a, 0]$. This peak will jeopardize our approximation around the origin if it's too close to the origin.

Therefore, it's very important to select suitable domain $[a, b]$ and parameters μ, ν in th enew approach.

3.3 Numerical examples

In this section, we apply the fast algorithms in computing the weight $G(l, m)$ and solving $Q(f, f)$ in the Boltzmann equation. We consider the 3D VHS molecule model and let $S = 5.0$ and $L \approx 11.04$ in the Fourier spectral method. We let $N = 32$ and apply $N_\rho = 32$ Gauss-Legendre quadrature points in radial direction.

In this section we compare three fast Fourier methods for numerical evaluation:

1. **the direct spectral method,** G^{direct} is computed via (3.18) and $Q^{\text{direct}}(f)$ is computed via (3.10) directly. They will be used as reference solution.

2. **the fast algorithm,** G^{fast} is computed via (3.19) using M_{sph} spherical quadrature points on the unit sphere. $Q^{\text{fast}}(f)$ is computed via the fast evaluation (3.16) and (3.17).

3. **the new approach,** G^{new} is computed via (3.34) using M Fourier approximation of $u^{\text{new}}(s)$ on interval $[a, b]$. $Q^{\text{new}}(f)$ is computed via the fast evaluation (3.35) and (3.17).

As mentioned in the last section, the parameters (a, b, μ, ν) would affect the performance of new method dramatically. For comparison, we consider four sets of parameters in the following numerical tests

$$(A): \begin{cases} a = -120, \quad b = P_N \approx 6224.07, \\ \mu = -20, \quad \nu = 5000, \end{cases} \tag{3.37}$$

$$(B): \begin{cases} a = -200, \quad b = P_N \approx 6224.07, \\ \mu = -50, \quad \nu = 5000, \end{cases} \tag{3.38}$$

$$(C): \begin{cases} a = -320, \quad b = P_N \approx 6224.07, \\ \mu = -120, \quad \nu = 5000, \end{cases} \tag{3.39}$$

and

$$(D): \begin{cases} a = -500, \quad b = P_N \approx 6224.07, \\ \mu = -200, \quad \nu = 5000. \end{cases} \tag{3.40}$$

The profile of u^{new} functions in different parameters are plotted in Fig 3.3.

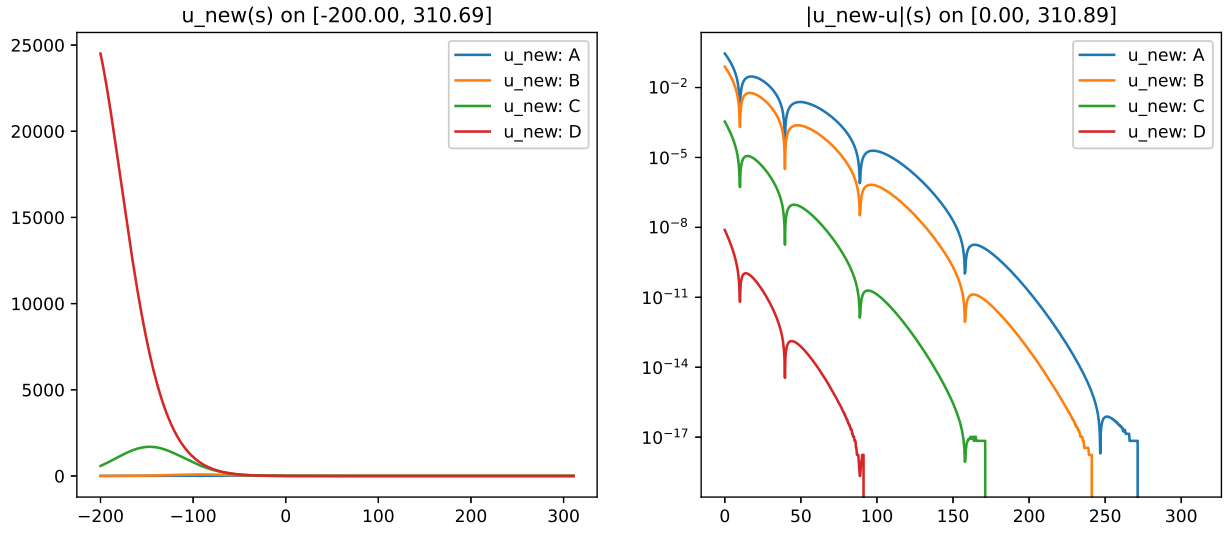

Figure 3.3. The profile of u^{new} function with different parameters. Left: u^{new} on the interval $[-200, P_N/20]$. Right: absolute error $|u^{\text{new}} - u|(s)$ on interval $[0, P_N/20]$.

In the following numerical tests, we will check the fast algorithm with different spherical quadrature and the new approach with different parameters:

- the fast algorithm using Lebedev quadrature on the unit sphere.

- the fast algorithm using Symmetric Spherical Designs (SSD) on the unit sphere.
- the new approach using Fourier approximation of $u^{\text{even}} = \text{sinc}(\sqrt{|s|})$ on the domain $s \in [-P_N, P_N]$.
- the new approach using Fourier approximation of u^{new} with four different sets of parameters (a, b, μ, ν).

3.3.1 Approximation of weight $G(l, m)$

We introduce three sets of index (l, m) for the weight $G(l, m)$.

(1) **global testing points**: a group of $M^{\text{test}} = 10026$ randomly chosen index

$$T_{\text{global}} = \left\{ (l^\alpha, m^\alpha) : -\frac{N}{2} \le l^\alpha + m^\alpha \le \frac{N}{2} - 1 \right\}. \tag{3.41}$$

Here we use the python package `sobol_seq` to generate quasi-random index l^α and m^α in T_{global}. It's used to evaluate the approximation of $G(l, m)$ in both low and high frequency region.

(2) **center testing points**: $M^{\text{test}} = 729$ sampling points given by

$$T_{\text{center}} = \{ (l_x, 0, l_z, 0, m_y, 0) : -4 \le l_x, l_z, m_y \le 4 \}. \tag{3.42}$$

It's used to evaluate the approximation of $G(l, m)$ in low frequency region.

(3) **non-center testing points**: $M^{\text{test}} = 729$ sampling points given by

$$T_{\text{non-center}} = \left\{ (l_x, 0, l_z, 0, m_y, 0) : \begin{cases} -16 \le l_x, l_z, m_y \le -12, \\ \text{or } 12 \le l_x, l_z, m_y \le 15. \end{cases} \right\}. \tag{3.43}$$

It's used to evaluate the approximation of $G(l, m)$ in high frequency region.

For the two fast approaches, we check their mean absolute error

$$\mathrm{MAE} = \frac{1}{M^{\mathrm{test}}} \sum_{1 \le \alpha \le M^{\mathrm{test}}} |G^*(l^\alpha, m^\alpha) - G^{\mathrm{direct}}(l^\alpha, m^\alpha)|, \tag{3.44}$$

on these three sets of sampling points, where G^* represents the numerical evaluation G^{fast} or G^{new}. The numerical results are shown in Fig 3.4 and Table 3.2. We can observe that:

- For T_{center}, the fast algorithm always performs better than the new approach. Since the integrands on the unit sphere

$$\exp\left(\mathrm{i}\frac{\boldsymbol{\pi}}{L}\rho\frac{l-m}{2}\cdot\sigma\right) \tag{3.45}$$

  don't involve strong oscillation when l, m lie in the low-frequency region. It is easy for spherical quadrature to achieve good accuracy on those integrals.

- For T_{global} and $T_{\mathrm{non-center}}$, when $M \le 200$ the fast algorithm still provide the best approximation. As M increases, the numerical errors decay much faster in the new approach with parameter sets (C) and (D). Eventually, the new approach could provide a better approximation when $M \ge 400$. That means when approximating weight $G(l, m)$ in the high-frequency region, it will cost a large number of spherical quadrature points to resolve the strong oscillations in the integrals. Meanwhile, the new approach can always guarantee accuracy with a suitable choice of parameters and adequate Fourier bases. Therefore, the new approach can approximate high-frequency weight well with much less computation cost.

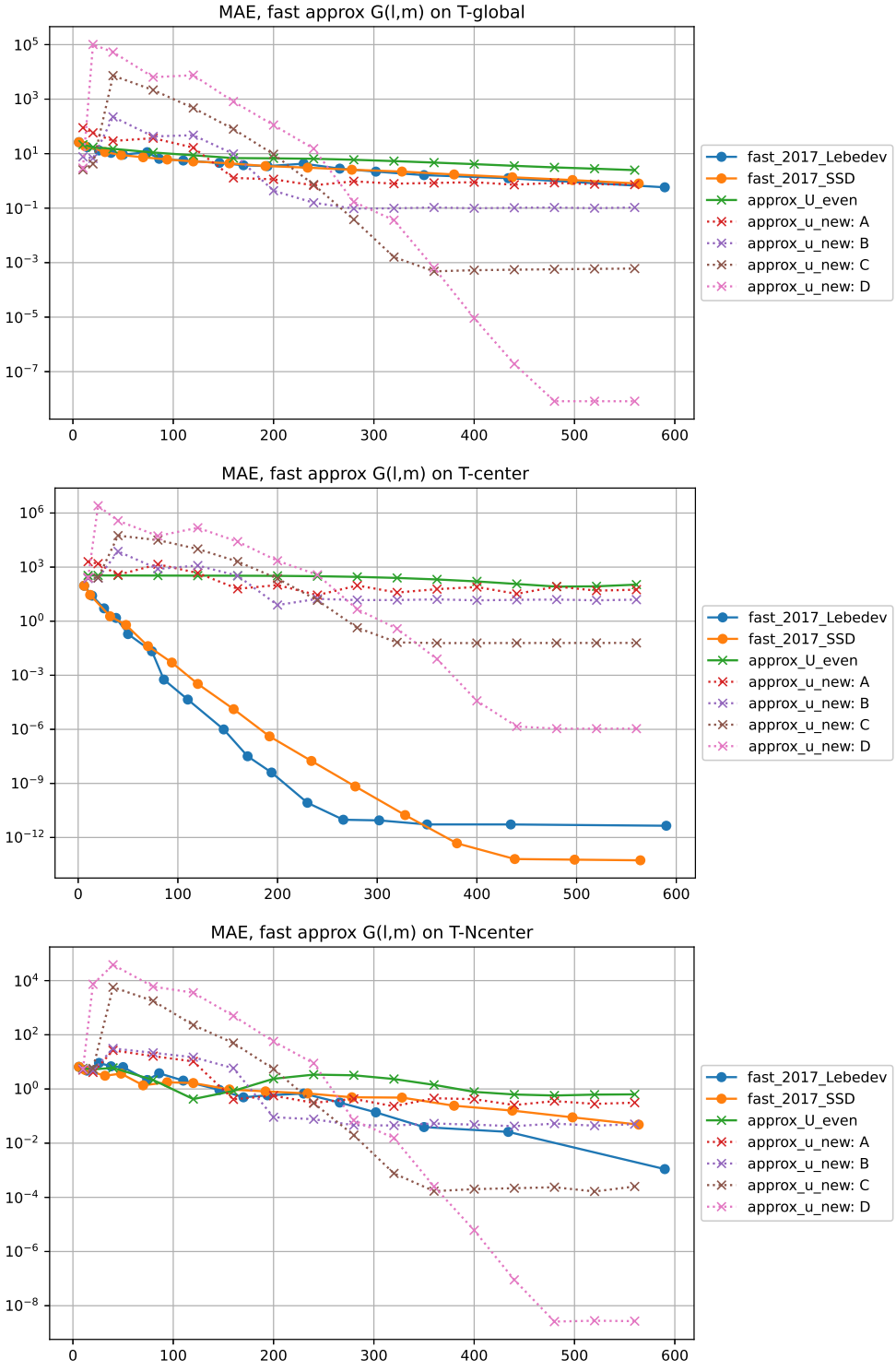

Figure 3.4. MAE of the different fast decomposition in estimating the weight $G(l, m)$. Error is computed on T_{global}, T_{center} and $T_{\text{non-center}}$.

Table 3.2. The scale of MAE on T_{gloabl} for Fourier methods.

	$M \leq 200$	$200 < M < 300$	$M \geq 300$
2017 fast Lebedev	$\mathcal{O}(10)$	$\mathcal{O}(1)$	$\mathcal{O}(1) \to \mathcal{O}(10^{-1})$
2017 fast SSD	$\mathcal{O}(10)$	$\mathcal{O}(1)$	$\mathcal{O}(1) \to \mathcal{O}(10^{-1})$
new: u^{even}	$\mathcal{O}(10)$	$\mathcal{O}(10)$	$\mathcal{O}(1)$
new: u^{new}-(A)	$\mathcal{O}(10^2) \to \mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
new: u^{new}-(B)	$\mathcal{O}(1) \to \mathcal{O}(10^3) \to \mathcal{O}(1)$	$\mathcal{O}(1) \to \mathcal{O}(10^{-1})$	$\mathcal{O}(10^{-1})$
new: u^{new}-(C)	$\mathcal{O}(1) \to \mathcal{O}(10^4) \to \mathcal{O}(10)$	$\mathcal{O}(10) \to \mathcal{O}(10^{-2})$	$\mathcal{O}(10^{-3})$
new: u^{new}-(D)	$\mathcal{O}(1) \to \mathcal{O}(10^5) \to \mathcal{O}(10^2)$	$\mathcal{O}(10^2) \to \mathcal{O}(10^{-1})$	$\mathcal{O}(10^{-1}) \to \mathcal{O}(10^{-8})$

3.3.2 Solving $Q(f)$ in Boltzmann equation

We now test the spectral methods for solving the Boltzmann equation. Let us consider five different initial f, which are defined as

1. **BKW solution:** Consider the 3D BKW solution:

$$f_{\text{BKW}}(t, v) = \frac{1}{2(2\pi K)^{3/2}} \exp\left(-\frac{v^2}{2K}\right)\left(\frac{5K-3}{K} + \frac{1-K}{K^2}v^2\right), \tag{3.46}$$

   where $K(t) = 1 - \exp(-t/6)$.

   For the BKW model, we know the exact $Q(f)$ as

$$Q(f) = \left\{\left(-\frac{3}{2K} + \frac{v^2}{2K^2}\right) f + \frac{1}{2(2\pi K)^{3/2}} \exp\left(-\frac{v^2}{2K}\right)\left(\frac{3}{K^2} + \frac{K-2}{K^3}v^2\right)\right\} K', \tag{3.47}$$

   with $K' = \exp(-t/6)/6$, which will be used as reference solution.

2. **two Gaussian initial:** Consider the initial condition

$$f_\alpha(v) = \frac{\rho_1}{(2\pi T_1)^{3/2}} \exp\left(-\frac{(v-V_1)^2}{2T_1}\right) + \frac{\rho_2}{(2\pi T_2)^{3/2}} \exp\left(-\frac{(v-V_2)^2}{2T_2}\right), \tag{3.48}$$

   with $\rho_1 = \rho_2 = 1/2$, $T_1 = T_2 = 1$ and $V_1 = (x_1, y_1, z_1) = (-2, 2, 0)$, $V_2 = (x_2, y_2, z_2) = (2, 0, 0)$.

3. **two Gaussian initial:** Consider the initial condition

$$f_\beta(v) = \frac{\rho_1}{(2\pi T_1)^{3/2}} \exp\left(-\frac{(v-V_1)^2}{2T_1}\right) + \frac{\rho_2}{(2\pi T_2)^{3/2}} \exp\left(-\frac{(v-V_2)^2}{2T_2}\right), \tag{3.49}$$

with $\rho_1 = 0.7, \rho_2 = 0.3$, $T_1 = T_2 = 1$ and $V_1 = (x_1, y_1, z_1) = (-2, 1, 0)$, $V_2 = (x_2, y_2, z_2) = (0, 0, -1)$.

4. **dis-continuous initial:** Consider the initial condition

$$f_{\text{discon}}(v) = \begin{cases} \frac{\rho_1}{(2\pi T_1)^{3/2}} \exp\left(-\frac{v^2}{2T_1}\right), & v_x \geq 0, \\ \frac{\rho_2}{(2\pi T_2)^{3/2}} \exp\left(-\frac{v^2}{2T_2}\right), & v_x < 0, \end{cases} \tag{3.50}$$

where $\rho_1 = 0.8$, $\rho_2 = 0.5$ and $T_1 = T_2 = 1$.

5. **random initial:** Consider a random initial f_{rand} generated by python package `numpy.random` .

Notice that only the BKW solution has an exact solution that can be used as a reference solution. For the other four initial conditions f, the numerical solutions will be compared with the direct spectral method. The numerical results are shown in Fig 3.5-3.9. One can observe that:

- For f_{BKW}, all the Fourier methods can achieve $\mathcal{O}(1\text{e}-5)$ accuracy quickly. That is exactly the approximation error $\| Q^{\text{ext}}(f) - Q^{\text{direct}}(f) \|_{L^\infty}$ in the direct spectral method.

- For the two Gaussian initial and discontinuous initial, the fast algorithm performs better than the new approach. That is due to the fact that fast algorithm could approximate the weight $G(l, m)$ very well for index l, m in the low-frequency region. This will give the fast algorithm an edge since the Fourier coefficients of initial condition f_α and f_β decay very fast in the high-frequency region.

- For the random initial f_{rand}, the Fourier coefficients don't decay like the smooth initial. The fast algorithm still provides the best accuracy when $M \leq 200$. However, the new approach with parameter set (D) could get better accuracy for $M \geq 300$.

- Overall, one can see that the scale of $\| Q^{\text{ext}}(f) - Q^{\text{num}}(f) \|_{L^\infty}$ is much larger than $\| Q^{\text{direct}}(f) - Q^{\text{num}}(f) \|_{L^\infty}$.

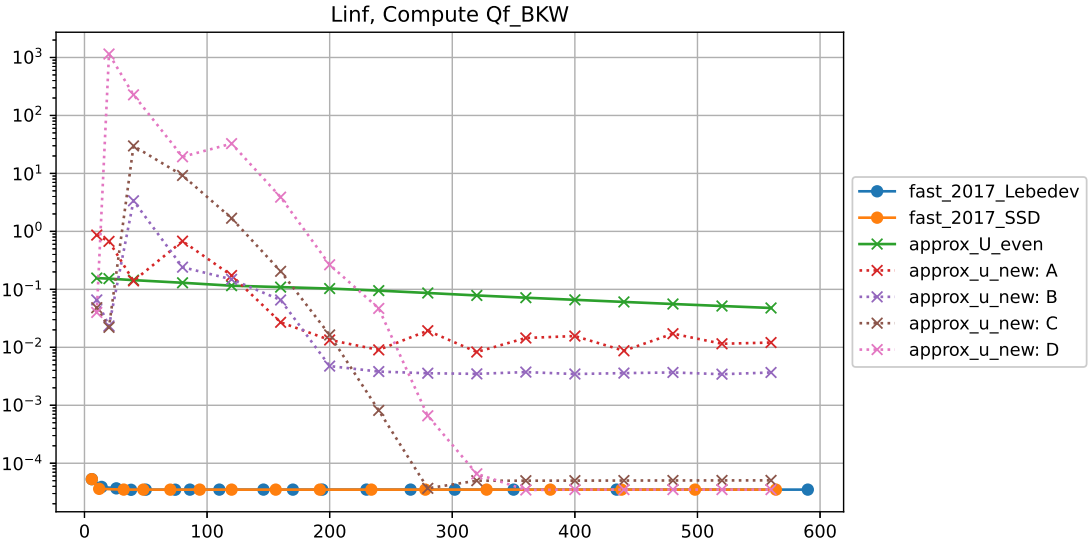

Figure 3.5. (f_{BKW}) Error $\| Q^{\text{ext}}(f) - Q^{\text{num}}(f) \|_{L^\infty}$ of the different fast approaches. The x-axis corresponds to number of spherical quadrature points (M_{sph}) in the fast algorithm and number of Fourier basis (M) in approximation to u^{new}.

3.4 Conclusion

We studied the recent development in fast Fourier spectral methods solving the Boltzmann equation. By using a different decomposition of the weight term $G(l, m)$, we can propose a new approach to speed up the Fourier method. Several numerical examples of 3D VHS molecule models have been presented for a comparison between different Fourier methods. In certain situations, the new approach did gain some improvement in numerical accuracy and efficiency.

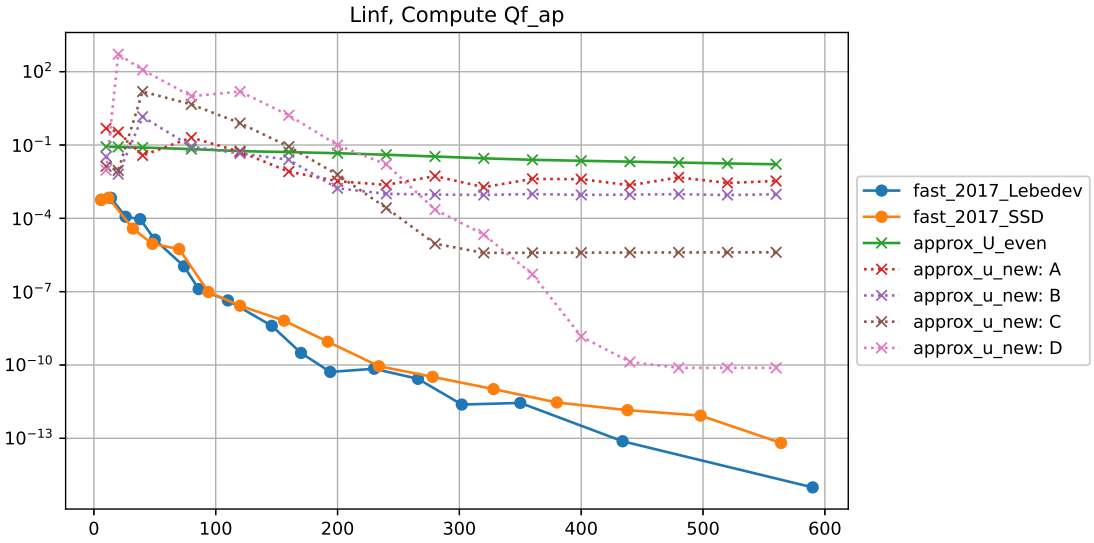

Figure 3.6. (f_α) Error $\| Q^{\text{direct}}(f) - Q^{\text{num}}(f) \|_{L^\infty}$ of the different fast approaches. The x-axis corresponds to number of spherical quadrature points (M_{sph}) in the fast algorithm and number of Fourier basis (M) in approximation to u^{new}.

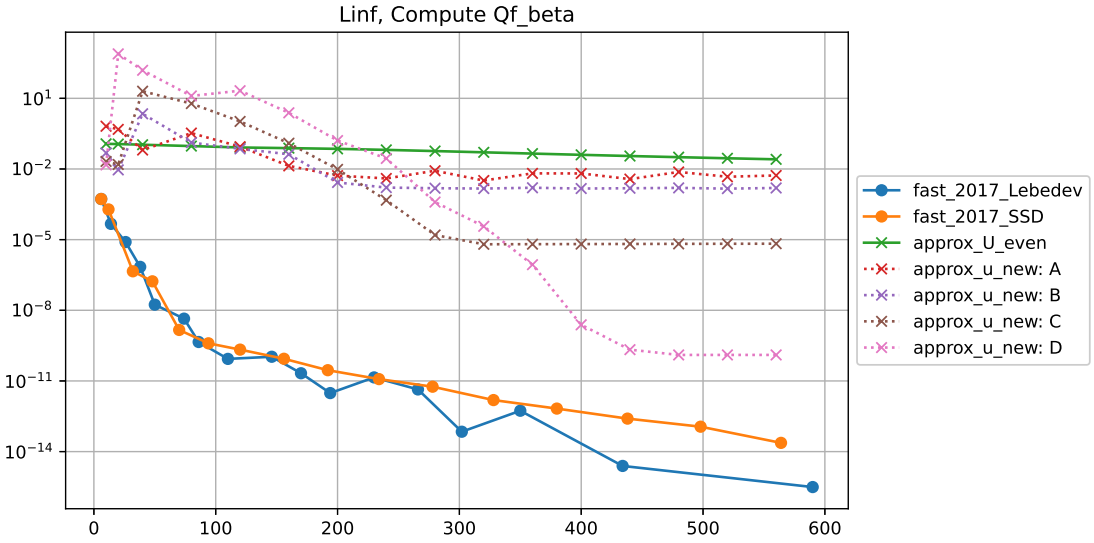

Figure 3.7. (f_β) Error $\| Q^{\text{direct}}(f) - Q^{\text{num}}(f) \|_{L^\infty}$ of the different fast approaches. The x-axis corresponds to number of spherical quadrature points (M_{sph}) in the fast algorithm and number of Fourier basis (M) in approximation to u^{new}.

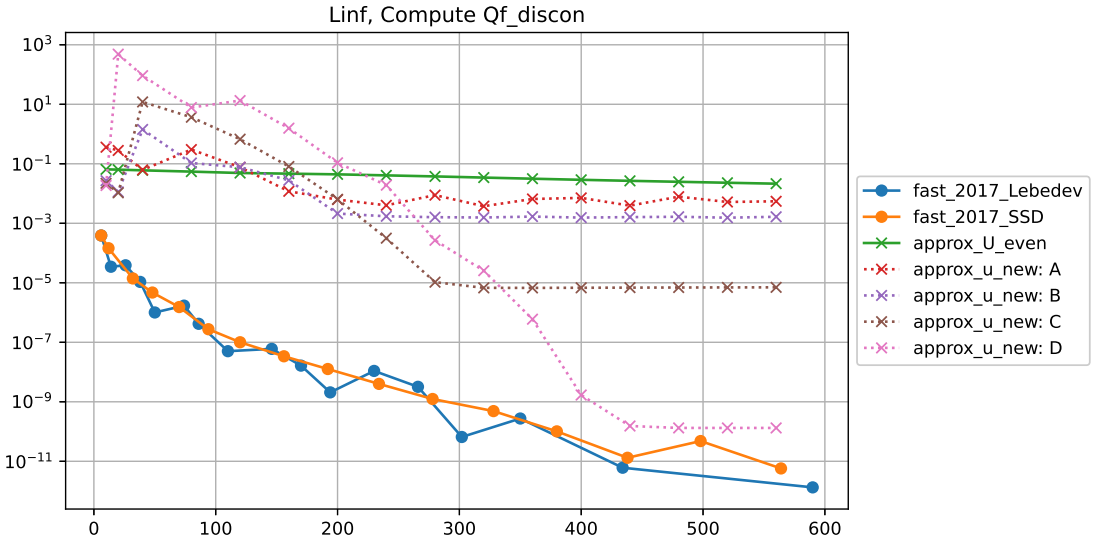

Figure 3.8. (f_{discon}) Error $\| Q^{\text{direct}}(f) - Q^{\text{num}}(f) \|_{L^\infty}$ of the different fast approaches. The x-axis corresponds to number of spherical quadrature points (M_{sph}) in the fast algorithm and number of Fourier basis (M) in approximation to u^{new}.

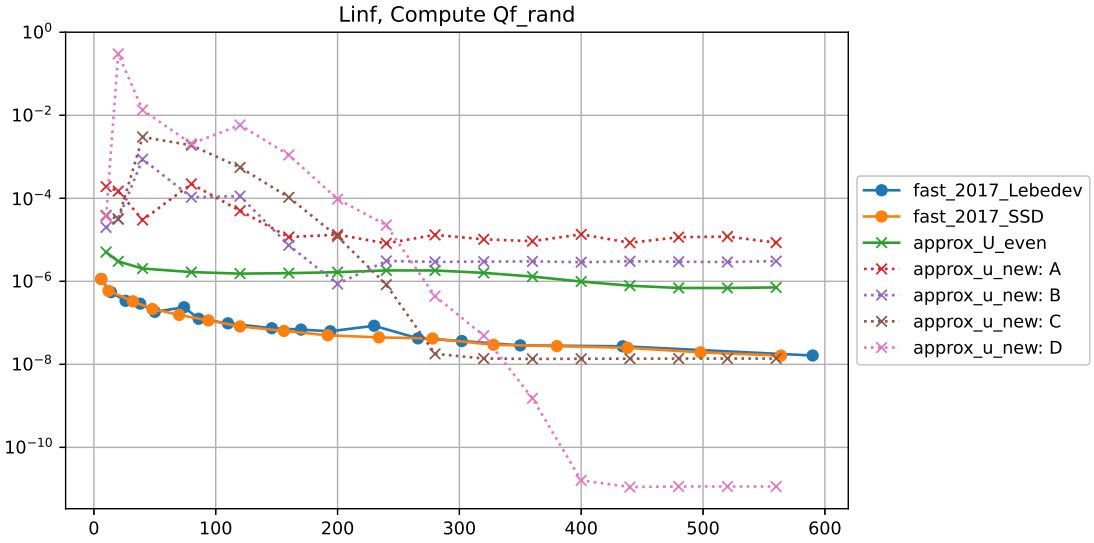

Figure 3.9. (f_{rand}) Error $\| Q^{\text{direct}}(f) - Q^{\text{num}}(f) \|_{L^\infty}$ of the different fast approaches. The x-axis corresponds to number of spherical quadrature points (M_{sph}) in the fast algorithm and number of Fourier basis (M) in approximation to u^{new}.

4. A FAST PETROV-GALERKIN SPECTRAL METHOD FOR BOLTZMANN EQUATION

In this chapter, we propose a Petrov-Galerkin spectral method for the spatially inhomogeneous Boltzmann equation

$$\partial_t f = Q(f, f), \quad t > 0, \quad \boldsymbol{v} \in \mathbb{R}^d, \quad d \geq 2. \tag{4.1}$$

In Section 4.1, we introduce the mapped Chebyshev functions in $\mathbb{R}^d$ along with their approximation properties. In Section 4.2, we construct the Petrov-Galerkin spectral method for the Boltzmann equation using the mapped Chebyshev functions as trial and test functions. The approximation properties for the collision operator and moments are proved as well. The numerical realization including the fast algorithm is described in detail in Section 4.3. In Section 4.4, several numerical tests in 2D and 3D are performed to demonstrate the accuracy and efficiency of the proposed method. This chapter is concluded in Section 4.5.

4.1 Multi-dimensional mapped Chebyshev functions

In this section, we introduce the mapped Chebyshev functions in $\mathbb{R}^d$ and discuss their approximation properties. These functions are extension of the one-dimensional mapped Chebyshev functions introduced in [43] based on tensor product formulation [49], [50]. Later in Section 4.2, they will serve as the trial functions and test functions in the Petrov-Galerkin spectral method for the Boltzmann equation.

4.1.1 Mapped Chebyshev functions in $\mathbb{R}^d$

To define the mapped Chebyshev functions in $\mathbb{R}^d$, we start with the one-dimensional Chebyshev polynomials on the interval $I = (-1, 1)$:

$$T_0(\xi) = 1, \quad T_1(\xi) = \xi, \quad T_{k+1}(\xi) = 2\xi T_k(\xi) - T_{k-1}(\xi), \quad k \geq 1. \tag{4.2}$$

Define the inner product $(\cdot,\cdot)_\omega$ as

$$(F,G)_\omega := \int_I F(\xi)G(\xi)\omega(\xi)\ \mathrm{d}\xi, \quad \omega(\xi) = (1-\xi^2)^{-\frac{1}{2}}, \tag{4.3}$$

then $\{T_k(\xi)\}_{k\geq 0}$ satisfy the orthogonality condition

$$(T_k, T_l)_\omega = c_k\delta_{k,l}, \quad \forall\ k,l \geq 0, \tag{4.4}$$

where $c_0 = \boldsymbol{\pi}$ and $c_k = \boldsymbol{\pi}/2$ for $k \geq 1$.

We then introduce a one-to-one mapping $\xi \to v(\xi)$ (its inverse is denoted as $v \to \xi(v)$) from I to $\mathbb{R}$ such that

$$\frac{\mathrm{d}v}{\mathrm{d}\xi} = \frac{S}{(1-\xi^2)^{1+\frac{r}{2}}} := \frac{\omega(\xi)}{[\mu(\xi)]^2}, \quad v(\pm 1) = \pm\infty, \tag{4.5}$$

where $S > 0$ is a scaling parameter, $r \geq 0$ is the tail parameter, and the function μ is given by

$$\mu(\xi) = \frac{(1-\xi^2)^{\frac{1+r}{4}}}{\sqrt{S}}. \tag{4.6}$$

With this mapping we define two sets of mapped Chebyshev functions in $\mathbb{R}$ as

$$\widetilde{T}_k(v) := \frac{[\mu(\xi(v))]^4}{\sqrt{c_k}} T_k(\xi(v)), \quad \widehat{T}_k(v) := \frac{[\mu(\xi(v))]^{-2}}{\sqrt{c_k}} T_k(\xi(v)). \tag{4.7}$$

Define the inner product $(\cdot,\cdot)_{\mathbb{R}}$ as

$$(f,g)_{\mathbb{R}} := \int_{\mathbb{R}} f(v)g(v)\ \mathrm{d}v, \tag{4.8}$$

then it is easy to check that $\{\widetilde{T}_k(v)\}_{k\geq 0}$ and $\{\widehat{T}_k(v)\}_{k\geq 0}$ satisfy the orthonormal condition:

$$(\widetilde{T}_k, \widehat{T}_l)_{\mathbb{R}} = \delta_{k,l}, \quad \forall\ k,l \geq 0. \tag{4.9}$$

Remark 4. *Two one-to-one mappings between I and $\mathbb{R}$ often used in practice are*

- **logarithmic mapping** ($r = 0$)*:*

$$v = \frac{S}{2} \ln\left(\frac{1+\xi}{1-\xi}\right), \quad \xi = \tanh\left(\frac{v}{S}\right), \quad \mu(\xi) = \frac{1}{\sqrt{S}}(1-\xi^2)^{\frac{1}{4}}, \tag{4.10}$$

- **algebraic mapping** ($r = 1$)*:*

$$v = \frac{S\xi}{\sqrt{1-\xi^2}}, \quad \xi = \frac{v}{\sqrt{S^2+v^2}}, \quad \mu(\xi) = \frac{1}{\sqrt{S}}(1-\xi^2)^{\frac{1}{2}}. \tag{4.11}$$

In the multi-dimensional case, we denote the multi-vector as $\boldsymbol{v} = (v_1, \ldots, v_d)$ and multi-index as $\boldsymbol{k} = (k_1, \ldots, k_d)$, where k_{j} is a non-negative integer for each $\mathrm{j} = 1, \ldots, d$; $0 \le \boldsymbol{k} \le N$ means $0 \le k_{\mathrm{j}} \le N$ for each $\mathrm{j} = 1, \ldots, d$. We define the mapped Chebyshev functions in $\mathbb{R}^d$ using (4.7) via the tensor product as

$$\widetilde{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v}) := \prod_{\mathrm{j}=1}^{d} \widetilde{T}_{k_{\mathrm{j}}}(v_{\mathrm{j}}), \quad \widehat{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v}) := \prod_{\mathrm{j}=1}^{d} \widehat{T}_{k_{\mathrm{j}}}(v_{\mathrm{j}}). \tag{4.12}$$

The inner products $(\cdot,\cdot)_{\boldsymbol{\omega}}$ in $I^d = (-1,1)^d$ and $(\cdot,\cdot)_{\mathbb{R}^d}$ in $\mathbb{R}^d$ are defined, respectively, by

$$(F,G)_{\boldsymbol{\omega}} := \int_{I^d} F(\boldsymbol{\xi})G(\boldsymbol{\xi})\boldsymbol{\omega}(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}, \quad (f,g)_{\mathbb{R}^d} := \int_{\mathbb{R}^d} f(\boldsymbol{v})g(\boldsymbol{v})\,\mathrm{d}\boldsymbol{v}, \tag{4.13}$$

with the weight function $\boldsymbol{\omega}(\boldsymbol{\xi}) := \prod_{\mathrm{j}=1}^{d} \omega(\xi_{\mathrm{j}})$. Then we have

$$\left(\widetilde{\boldsymbol{T}}_{\boldsymbol{k}}, \widehat{\boldsymbol{T}}_{\boldsymbol{l}}\right)_{\mathbb{R}^d} = \prod_{\mathrm{j}=1}^{d} \left(\widetilde{T}_{k_{\mathrm{j}}}, \widehat{T}_{l_{\mathrm{j}}}\right)_{\mathbf{R}} = \prod_{\mathrm{j}=1}^{d} \delta_{k_{\mathrm{j}},l_{\mathrm{j}}} =: \boldsymbol{\delta}_{\boldsymbol{k},\boldsymbol{l}}. \tag{4.14}$$

Therefore, any d-variate function $f(\boldsymbol{v})$ can be expanded using $\{\widetilde{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v})\}_{\boldsymbol{k}\ge 0}$ as

$$f(\boldsymbol{v}) = \sum_{\boldsymbol{k}\ge 0} \widetilde{f}_{\boldsymbol{k}} \widetilde{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v}) = \sum_{\boldsymbol{k}\ge 0} \widetilde{f}_{\boldsymbol{k}} \frac{[\boldsymbol{\mu}(\boldsymbol{\xi})]^4}{\sqrt{\boldsymbol{c}_{\boldsymbol{k}}}} \boldsymbol{T}_{\boldsymbol{k}}(\boldsymbol{\xi}), \tag{4.15}$$

and the expansion coefficients $\{\widetilde{f}_{\boldsymbol{k}}\}_{\boldsymbol{k}\ge 0}$ are determined by

$$\widetilde{f}_{\boldsymbol{k}} = \left(f, \widehat{\boldsymbol{T}}_{\boldsymbol{k}}\right)_{\mathbb{R}^d} = \frac{1}{\sqrt{\boldsymbol{c}_{\boldsymbol{k}}}} \left([\boldsymbol{\mu}(\boldsymbol{\xi})]^{-4} f(\boldsymbol{v}(\boldsymbol{\xi})), \boldsymbol{T}_{\boldsymbol{k}}(\boldsymbol{\xi})\right)_{\boldsymbol{\omega}}, \tag{4.16}$$

where

$$\boldsymbol{\mu}(\boldsymbol{\xi}) := \prod_{\mathrm{j}=1}^{d} \mu(\xi_{\mathrm{j}}), \quad \boldsymbol{c_k} := \prod_{\mathrm{j}=1}^{d} c_{k_{\mathrm{j}}}, \quad \boldsymbol{T_k}(\boldsymbol{\xi}) := \prod_{\mathrm{j}=1}^{d} T_{k_{\mathrm{j}}}(\xi_{\mathrm{j}}), \tag{4.17}$$

and $\boldsymbol{v}(\boldsymbol{\xi})$ is the mapping from I^d to $\mathbb{R}^d$ such that each component ξ_{j} is mapped to v_{j} via the 1D mapping (4.5). The inverse mapping $\boldsymbol{\xi}(\boldsymbol{v})$ is understood similarly.

In Section 4.2, we will introduce the Petrov-Galerkin spectral method for the Boltzmann equation in $\mathbb{R}^d$, where the trial function space and test function space are chosen, respectively, as

$$\widetilde{\mathbb{T}}_N^d := \{\widetilde{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v})\}_{0\le \boldsymbol{k}\le N}, \quad \widehat{\mathbb{T}}_N^d := \{\widehat{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v})\}_{0\le \boldsymbol{k}\le N}. \tag{4.18}$$

The choice of these functions is motivated by their decay/growth properties at large $|\boldsymbol{v}|$. The following result is a straightforward extension of the 1D result in [43].

Lemma 4.1.1. *For any $\boldsymbol{k} \ge 0$ and $|\boldsymbol{v}| \gg 1$, we have*

$$\left|\widetilde{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v})\right| \sim \begin{cases} \mathrm{e}^{-\frac{2}{S}(\sum_{\mathrm{j}=1}^{d}|v_{\mathrm{j}}|)}, & r=0, \\ \prod_{\mathrm{j}=1}^{d}|v_{\mathrm{j}}|^{-4}, & r=1; \end{cases} \quad \left|\widehat{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v})\right| \sim \begin{cases} \mathrm{e}^{\frac{1}{S}(\sum_{\mathrm{j}=1}^{d}|v_{\mathrm{j}}|)}, & r=0, \\ \prod_{\mathrm{j}=1}^{d}|v_{\mathrm{j}}|^{2}, & r=1, \end{cases} \tag{4.19}$$

where $r = 0$ corresponds to the logarithmic mapping (4.10) and $r = 1$ to the algebraic mapping (4.11).

4.1.2 Approximation properties

We describe below some approximation properties of the mapped Chebyshev functions in $\mathbb{R}^d$.

For a function $f(\boldsymbol{v})$ defined in $\mathbb{R}^d$, the transform $\boldsymbol{v}(\boldsymbol{\xi})$ maps it to a function in I^d. Hence, we introduce the linked function pair (f, F) such that $f(\boldsymbol{v}) = f(\boldsymbol{v}(\boldsymbol{\xi})) \equiv F(\boldsymbol{\xi})$. In addition, we introduce another function pair $(\widehat{f}^{\alpha}, \widehat{F}^{\alpha})$ as

$$\widehat{f}^{\alpha}(\boldsymbol{v}) := f(\boldsymbol{v})[\boldsymbol{\mu}(\boldsymbol{\xi}(\boldsymbol{v}))]^{-\alpha} = F(\boldsymbol{\xi})[\boldsymbol{\mu}(\boldsymbol{\xi})]^{-\alpha} =: \widehat{F}^{\alpha}(\boldsymbol{\xi}). \tag{4.20}$$

We define the approximation space in $\mathbb{R}^d$ with a parameter α as

$$\mathbb{V}_N^{\alpha,d}(\mathbb{R}^d) := \text{span}\left\{\boldsymbol{T}_{\boldsymbol{k}}^{\alpha}(\boldsymbol{v}) := [\boldsymbol{\mu}(\boldsymbol{\xi}(\boldsymbol{v}))]^{\alpha}\boldsymbol{T}_{\boldsymbol{k}}(\boldsymbol{\xi}(\boldsymbol{v})),\ 0 \le \boldsymbol{k} \le N\right\}. \tag{4.21}$$

Therefore, the trial function space $\widetilde{\mathbb{T}}_N^d$ and test function space $\widehat{\mathbb{T}}_N^d$ introduced in the previous section correspond to $\mathbb{V}_N^{4,d}$ and $\mathbb{V}_N^{-2,d}$, respectively.

In the following, the L^2 space with a given weight $\boldsymbol{w}$ is equipped with norm

$$\| f \|_{L^2_{\boldsymbol{w}}(I^d)} = \left(\int_{I^d} |f(\boldsymbol{\xi})|^2 \boldsymbol{w}(\boldsymbol{\xi})\ \mathrm{d}\boldsymbol{\xi}\right)^{1/2} \quad \text{or} \quad \| f \|_{L^2_{\boldsymbol{w}}(\mathbb{R}^d)} = \left(\int_{\mathbb{R}^d} |f(\boldsymbol{v})|^2 \boldsymbol{w}(\boldsymbol{v})\ \mathrm{d}\boldsymbol{v}\right)^{1/2}, \tag{4.22}$$

depending on the domain of interest.

Let $\mathbb{P}_N^d(I^d)$ denote the set of d-variate polynomials in I^d with degree $\le N$ in each direction, and $\Pi_N^d : L^2_{\boldsymbol{\omega}}(I^d) \to \mathbb{P}_N^d(I^d)$ be the Chebyshev orthogonal projection operator such that

$$\left(\Pi_N^d F - F, \phi\right)_{\boldsymbol{\omega}} = 0, \quad \forall \phi \in \mathbb{P}_N^d(I^d). \tag{4.23}$$

Then we define another projection operator $\boldsymbol{\pi}_N^{\alpha,d} : L^2_{\boldsymbol{\mu}^{2-2\alpha}}(\mathbb{R}^d) \to \mathbb{V}_N^{\alpha,d}(\mathbb{R}^d)$ by

$$\boldsymbol{\pi}_N^{\alpha,d} f := \boldsymbol{\mu}^{\alpha}\Pi_N^d(F\boldsymbol{\mu}^{-\alpha}) = \boldsymbol{\mu}^{\alpha}\Pi_N^d \widehat{F}^{\alpha}. \tag{4.24}$$

One can verify using the definition that

$$\begin{aligned}
\left(\boldsymbol{\pi}_N^{\alpha,d} f - f, \boldsymbol{\mu}^{2-2\alpha}\boldsymbol{T}_{\boldsymbol{k}}^{\alpha}\right)_{\mathbb{R}^d} &= \int_{\mathbb{R}^d} (\boldsymbol{\pi}_N^{\alpha,d} f - f)\boldsymbol{\mu}^{2-\alpha}\boldsymbol{T}_{\boldsymbol{k}}(\boldsymbol{\xi}(\boldsymbol{v}))\,\mathrm{d}\boldsymbol{v} \\
&= \int_{I^d} \left[\boldsymbol{\mu}^{\alpha}\Pi_N^d \widehat{F}^{\alpha} - \boldsymbol{\mu}^{\alpha}\widehat{F}^{\alpha}\right] \boldsymbol{T}_{\boldsymbol{k}}(\boldsymbol{\xi})\boldsymbol{\mu}^{2-\alpha}\frac{\boldsymbol{\omega}(\boldsymbol{\xi})}{\boldsymbol{\mu}^2}\,\mathrm{d}\boldsymbol{\xi} \\
&= \left(\Pi_N^d \widehat{F}^{\alpha} - \widehat{F}^{\alpha}, \boldsymbol{T}_{\boldsymbol{k}}\right)_{\boldsymbol{\omega}} = 0, \quad \forall\ 0 \le \boldsymbol{k} \le N.
\end{aligned} \tag{4.25}$$

Next, we introduce the function space $\boldsymbol{B}_{\alpha}^m(\mathbb{R}^d)$ equipped with the norm

$$\| f \|_{\boldsymbol{B}_{\alpha}^m(\mathbb{R}^d)} = \left(\sum_{0 \le \boldsymbol{k} \le m} \| \boldsymbol{D}_{\alpha,\boldsymbol{v}}^{\boldsymbol{k}} f \|^2_{L^2_{\boldsymbol{\varpi}^{\boldsymbol{k}+\frac{1+r}{2}\mathbf{1}}}(\mathbb{R}^d)}\right)^{1/2}, \tag{4.26}$$

and semi-norm

$$|f|_{\boldsymbol{B}^m_\alpha(\mathbb{R}^d)} = \left(\sum_{\mathrm{j}=1}^{d} \| D^m_{\alpha,v_\mathrm{j}} f \|^2_{L^2_{\boldsymbol{\varpi}^{m\mathbf{e}_\mathrm{j}+\frac{1+r}{2}\mathbf{1}}}(\mathbb{R}^d)} \right)^{1/2}, \tag{4.27}$$

where $\mathbf{1}$ is an all-one vector, $\mathbf{e}_\mathrm{j} = (0, \ldots, 1, \ldots, 0)$ with 1 in the j-th position and 0 elsewhere, and

$$\boldsymbol{D}^{\boldsymbol{k}}_{\alpha,\boldsymbol{v}} f := D^{k_1}_{\alpha,v_1} \cdots D^{k_d}_{\alpha,v_d} f, \quad \boldsymbol{\varpi}^{\boldsymbol{k}} := \prod_{\mathrm{j}=1}^{d} (1 - \xi(v_\mathrm{j})^2)^{k_\mathrm{j}}, \tag{4.28}$$

with

$$D^{k_\mathrm{j}}_{\alpha,v_\mathrm{j}} f := \underbrace{a(v_\mathrm{j}) \frac{\partial}{\partial v_\mathrm{j}} \left(a(v_\mathrm{j}) \frac{\partial}{\partial v_\mathrm{j}} \left(\ldots \left(a(v_\mathrm{j}) \frac{\partial \widehat{f}^\alpha}{\partial v_\mathrm{j}} \right) \ldots \right) \right)}_{k_\mathrm{j} \text{ times derivatives}} = \frac{\partial^{k_\mathrm{j}} \widehat{F}^\alpha}{\partial \xi_\mathrm{j}}, \tag{4.29}$$

where $a(v_\mathrm{j}) := \frac{\mathrm{d}v_\mathrm{j}}{\mathrm{d}\xi_\mathrm{j}}$ is determined by the mapping.

We have the following approximation result.

Theorem 1.1. *Let $\alpha \in \mathbf{R}$, $r \geq 0$. If $f \in \boldsymbol{B}^m_\alpha(\mathbb{R}^d)$, we have*

$$\| \boldsymbol{\pi}^{\alpha,d}_N f - f \|_{L^2_{\boldsymbol{\mu}^{2-2\alpha}}(\mathbb{R}^d)} \leq C N^{-m} |f|_{\boldsymbol{B}^m_\alpha(\mathbb{R}^d)}. \tag{4.30}$$

Proof. Note that

$$\begin{aligned} \| \boldsymbol{\pi}^{\alpha,d}_N f - f \|^2_{L^2_{\boldsymbol{\mu}^{2-2\alpha}}(\mathbb{R}^d)} &= \int_{\mathbb{R}^d} (\boldsymbol{\pi}^{\alpha,d}_N f - f)^2 \boldsymbol{\mu}^{2-2\alpha} \, \mathrm{d}\boldsymbol{v} \\ &= \int_{I^d} \left[\boldsymbol{\mu}^\alpha \Pi^d_N \widehat{F}^\alpha - \boldsymbol{\mu}^\alpha \widehat{F}^\alpha \right]^2 \boldsymbol{\mu}^{2-2\alpha} \frac{\boldsymbol{\omega}(\boldsymbol{\xi})}{\boldsymbol{\mu}^2} \, \mathrm{d}\boldsymbol{\xi} \\ &= \| \Pi^d_N \widehat{F}^\alpha - \widehat{F}^\alpha \|^2_{L^2_{\boldsymbol{\omega}}(I^d)}. \end{aligned}$$

By the multi-variate (full tensor product) Chebyshev approximation result (Theorem 2.1 in [51]), we know

$$\| \Pi^d_N \widehat{F}^\alpha - \widehat{F}^\alpha \|_{L^2_{\boldsymbol{\omega}}(I^d)} \leq C N^{-m} \left(\sum_{\mathrm{j}=1}^{d} \| \partial^m_{\xi_\mathrm{j}} \widehat{F}^\alpha \|^2_{L^2_{\boldsymbol{\varpi}^{m\mathbf{e}_\mathrm{j}-\frac{1}{2}\mathbf{1}}}(I^d)} \right)^{1/2}.$$

Hence,

$$\begin{aligned}
&\| \boldsymbol{\pi}_N^{\alpha,d} f - f \|_{L^2_{\boldsymbol{\mu}^{2-2\alpha}}(\mathbb{R}^d)} = \| \Pi_N^d \widehat{F}^\alpha - \widehat{F}^\alpha \|_{L^2_{\boldsymbol{\omega}}(I^d)} \\
\leq\ & CN^{-m} \left(\sum_{\mathrm{j}=1}^{d} \| \partial_{\xi_\mathrm{j}}^m \widehat{F}^\alpha \|^2_{L^2_{\boldsymbol{\varpi}^{m\mathbf{e}_\mathrm{j} - \frac{1}{2}\mathbf{1}}}(I^d)} \right)^{1/2} \\
\leq\ & CN^{-m} \left(\sum_{\mathrm{j}=1}^{d} \| D_{\alpha,v_\mathrm{j}}^m f \|^2_{L^2_{\boldsymbol{\varpi}^{m\mathbf{e}_\mathrm{j} + \frac{1+r}{2}\mathbf{1}}}(\mathbb{R}^d)} \right)^{1/2} \\
=\ & CN^{-m} |f|_{\boldsymbol{B}^m_\alpha(\mathbb{R}^d)}.
\end{aligned}$$

□

4.2 A Petrov-Galerkin spectral method for the Boltzmann equation

We consider the initial value problem

$$\begin{cases} \partial_t f(t, \boldsymbol{v}) = Q(f, f), \quad t > 0, \quad \boldsymbol{v} \in \mathbb{R}^d, \\ f(0, \boldsymbol{v}) = f^0(\boldsymbol{v}), \end{cases} \tag{4.31}$$

where $Q(f, f)$, in a strong form, is given by (1.5). To construct the Petrov-Galerkin spectral method, the following weak form of the collision operator is more convenient:

$$(Q(f,f), \phi)_{\mathbb{R}^d} = \int_{\mathbb{R}^d} Q(f,f)(\boldsymbol{v})\phi(\boldsymbol{v})\, \mathrm{d}\boldsymbol{v} = \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\int_{S^{d-1}} \mathcal{B}(\boldsymbol{v}, \boldsymbol{v}_*, \boldsymbol{\sigma}) f(\boldsymbol{v}) f(\boldsymbol{v}_*) [\phi(\boldsymbol{v}') - \phi(\boldsymbol{v})]\, \mathrm{d}\boldsymbol{\sigma}\, \mathrm{d}\boldsymbol{v}\, \mathrm{d}\boldsymbol{v}_*, \tag{4.32}$$

where $\phi(\boldsymbol{v})$ is a test function.

We look for an approximation of f in the trial function space $\widetilde{\mathbb{T}}_N^d$ as

$$f(t, \boldsymbol{v}) \approx f_N(t, \boldsymbol{v}) = \sum_{0 \leq \boldsymbol{k} \leq N} \widetilde{f}_{\boldsymbol{k}}(t) \widetilde{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v}) \in \widetilde{\mathbb{T}}_N^d. \tag{4.33}$$

Substituting f_N into (4.31) and requiring the residue of the equation to be orthogonal to the test function space $\widehat{\mathbb{T}}_N^d$, we obtain

$$\left(\partial_t f_N - Q(f_N, f_N), \widehat{\boldsymbol{T}}_{\boldsymbol{k}}\right)_{\mathbb{R}^d} = 0 \quad \text{for all} \quad \widehat{\boldsymbol{T}}_{\boldsymbol{k}} \in \widehat{\mathbb{T}}_N^d. \tag{4.34}$$

By the orthogonality condition (4.14), we find that the coefficients $\{\widetilde{f}_{\boldsymbol{k}}(t)\}$ satisfy the following ODE system

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\widetilde{f}_{\boldsymbol{k}}(t) = \mathcal{Q}_{\boldsymbol{k}}^N, \\ \widetilde{f}_{\boldsymbol{k}}(0) = \widetilde{f}_{\boldsymbol{k}}^0, \end{cases} \qquad 0 \le \boldsymbol{k} \le N, \tag{4.35}$$

where

$$\begin{aligned} \mathcal{Q}_{\boldsymbol{k}}^N &:= \left(Q(f_N, f_N), \widehat{\boldsymbol{T}}_{\boldsymbol{k}}\right)_{\mathbb{R}^d} \\ &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\int_{S^{d-1}} \mathcal{B}(\boldsymbol{v}, \boldsymbol{v}_*, \boldsymbol{\sigma}) f_N(\boldsymbol{v}) f_N(\boldsymbol{v}_*) \left[\widehat{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v}') - \widehat{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v})\right] \mathrm{d}\boldsymbol{\sigma}\, \mathrm{d}\boldsymbol{v}\, \mathrm{d}\boldsymbol{v}_*, \end{aligned} \tag{4.36}$$

and

$$\widetilde{f}_{\boldsymbol{k}}^0 := \left(f^0, \widehat{\boldsymbol{T}}_{\boldsymbol{k}}\right)_{\mathbb{R}^d} = \frac{1}{\sqrt{\boldsymbol{c}_{\boldsymbol{k}}}} \left([\boldsymbol{\mu}(\boldsymbol{\xi})]^{-4} f^0(\boldsymbol{v}(\boldsymbol{\xi})), \boldsymbol{T}_{\boldsymbol{k}}(\boldsymbol{\xi})\right)_{\boldsymbol{\omega}}. \tag{4.37}$$

Note that we used the weak form (4.32) in (4.36).

Remark 5. *An equivalent way of writing the ODE system (4.35) is*

$$\begin{cases} \partial_t f_N(t, \boldsymbol{v}) = \boldsymbol{\pi}_N^{4,d} Q(f_N, f_N), \\ f_N(0, \boldsymbol{v}) = \boldsymbol{\pi}_N^{4,d} f^0(\boldsymbol{v}), \end{cases} \tag{4.38}$$

where $\boldsymbol{\pi}_N^{4,d}$ is the projection operator defined in (4.24) (with $\alpha = 4$). Indeed, for any $f \in L^2_{\boldsymbol{\mu}^{-6}}(\mathbb{R}^d)$,

$$\boldsymbol{\pi}_N^{4,d} f = \sum_{0 \le \boldsymbol{k} \le N} \left(f, \widehat{\boldsymbol{T}}_{\boldsymbol{k}}\right)_{\mathbb{R}^d} \widetilde{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v}) \in \mathbb{V}_N^{4,d}(\mathbb{R}^d) = \widetilde{\mathbb{T}}_N^d. \tag{4.39}$$

4.2.1 Approximation property for the collision operator

In this subsection, we establish a consistency result of the spectral approximation for the collision operator. We will show that if f and $Q(f,f)$ have certain regularity, the proposed approximation of the collision operator $\boldsymbol{\pi}_N^{4,d} Q(\boldsymbol{\pi}_N^{4,d} f, \boldsymbol{\pi}_N^{4,d} f)$ enjoys spectral accuracy. We will only prove this result under the algebraic mapping (4.11) with $S = 1$, that is in 1D,

$$v = \frac{\xi}{\sqrt{1-\xi^2}}, \quad \xi = \frac{v}{\sqrt{1+v^2}}, \quad \mu = \sqrt{1-\xi^2} = \frac{1}{\sqrt{1+v^2}}. \tag{4.40}$$

The reason of this choice is strongly motivated by the existing regularity result of the Boltzmann collision operator under an exponentially weighted Lebesgue norm:

$$\| f \|_{\mathcal{L}^p_k(\mathbb{R}^d)} = \left(\int_{\mathbb{R}^d} |f(\boldsymbol{v})|^p (1+|\boldsymbol{v}|^2)^{kp/2} \, \mathrm{d}\boldsymbol{v} \right)^{1/p}, \quad k \in \mathbf{R}, \ 1 \le p < \infty. \tag{4.41}$$

Specifically, we write the collision operator (1.5) as $Q(g,f) = Q^+(g,f) - Q^-(g,f)$, where the gain part and loss part are given by

$$\begin{aligned} Q^+(g,f)(\boldsymbol{v}) &= \int_{\mathbb{R}^d} \int_{S^{d-1}} \mathcal{B}(\boldsymbol{v}, \boldsymbol{v}_*, \boldsymbol{\sigma}) g(\boldsymbol{v}'_*) f(\boldsymbol{v}') \, \mathrm{d}\boldsymbol{\sigma} \, \mathrm{d}\boldsymbol{v}_*, \\ Q^-(g,f)(\boldsymbol{v}) &= \int_{\mathbb{R}^d} \int_{S^{d-1}} \mathcal{B}(\boldsymbol{v}, \boldsymbol{v}_*, \boldsymbol{\sigma}) g(\boldsymbol{v}_*) f(\boldsymbol{v}) \, \mathrm{d}\boldsymbol{\sigma} \, \mathrm{d}\boldsymbol{v}_*. \end{aligned} \tag{4.42}$$

Then we have the following regularity result for the gain operator $Q^+(g,f)$ established in [52].

Theorem 2.1 (Theorem 2.1, [52]). *Let $k, \eta \in \mathbb{R}$, $1 \le p < \infty$, and let the collision kernel $\mathcal{B}$ satisfy certain cut-off assumption[1]. Then the following estimate holds*

$$\| Q^+(g,f) \|_{\mathcal{L}^p_\eta(\mathbb{R}^d)} \le C_{k,\eta,p}(\mathcal{B}) \| g \|_{\mathcal{L}^1_{|k+\eta|+|\eta|}(\mathbb{R}^d)} \| f \|_{\mathcal{L}^p_{k+\eta}(\mathbb{R}^d)}, \tag{4.43}$$

where $C_{k,\eta,p}(\mathcal{B})$ is a constant that depends only on the kernel $\mathcal{B}$ and k, η and p.

[1] To avoid technicality, we do not spell out the condition here and only mention that most of the collision kernels used in numerical simulations satisfy this assumption.

To obtain a similar estimate for the loss operator $Q^-(g,f)$, we restrict ourselves to the variable hard sphere (VHS) collision model [53]. Note that this kernel falls into the assumption in Theorem 2.1. We have the following result.

Proposition 2.1. *Let $\eta \in \mathbb{R}$, $1 \le p < \infty$, and let the collision kernel takes the form $\mathcal{B} = C_\lambda |\boldsymbol{v} - \boldsymbol{v}_*|^\lambda$, where $0 \le \lambda \le 1$ and C_λ is a positive constant. Then the following estimate holds*

$$\| Q^-(g,f) \|_{\mathcal{L}^p_\eta(\mathbb{R}^d)} \le C_\lambda \| g \|_{\mathcal{L}^1_\lambda(\mathbb{R}^d)} \| f \|_{L^p_{\lambda+\eta}(\mathbb{R}^d)} . \tag{4.44}$$

Proof. Note that

$$|\boldsymbol{v} - \boldsymbol{v}_*| \le |\boldsymbol{v}| + |\boldsymbol{v}_*| = (|\boldsymbol{v}|^2 + |\boldsymbol{v}_*|^2 + 2|\boldsymbol{v}||\boldsymbol{v}_*|)^{1/2} \le \left(1 + |\boldsymbol{v}|^2\right)^{1/2} \left(1 + |\boldsymbol{v}_*|^2\right)^{1/2} .$$

Then

$$\begin{aligned} Q^-(g,f)(\boldsymbol{v}) &= C_\lambda f(\boldsymbol{v}) \int_{\mathbb{R}^d} g(\boldsymbol{v}_*) |\boldsymbol{v} - \boldsymbol{v}_*|^\lambda \, \mathrm{d}\boldsymbol{v}_* \\ &\le C_\lambda f(\boldsymbol{v}) \left(1 + |\boldsymbol{v}|^2\right)^{\lambda/2} \left[\int_{\mathbb{R}^d} g(\boldsymbol{v}_*) \left(1 + |\boldsymbol{v}_*|^2\right)^{\lambda/2} \mathrm{d}\boldsymbol{v}_* \right] \\ &= C_\lambda f(\boldsymbol{v}) \left(1 + |\boldsymbol{v}|^2\right)^{\lambda/2} \| g \|_{L^1_\lambda(\mathbb{R}^d)} . \end{aligned}$$

Therefore, for any $\eta \in \mathbf{R}$, $1 \le p < \infty$,

$$\| Q^-(g,f) \|_{\mathcal{L}^p_\eta(\mathbb{R}^d)} \le C_\lambda \| g \|_{\mathcal{L}^1_\lambda(\mathbb{R}^d)} \| f \|_{L^p_{\lambda+\eta}(\mathbb{R}^d)} .$$

□

Combining the previous two results, we can obtain the following theorem.

Theorem 2.2. *Let the collision kernel takes the form $\mathcal{B} = C_\lambda |\boldsymbol{v} - \boldsymbol{v}_*|^\lambda$, where $0 \le \lambda \le 1$ and C_λ is a positive constant. Then the collision operator $Q(g,f)$ satisfies*

$$\| Q(g,f) \|_{\mathcal{L}^2_{3d}(\mathbb{R}^d)} \le C_d(\mathcal{B}) \| g \|_{\mathcal{L}^1_{\lambda+6d}(\mathbb{R}^d)} \| f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)}, \tag{4.45}$$

where $C_d(\mathcal{B})$ is a constant that depends only on the kernel $\mathcal{B}$ and the dimension d.

Proof. Choosing $k = \lambda$, $\eta = 3d$, $p = 2$ in (4.43), we have

$$\| Q^+(g,f) \|_{\mathcal{L}^2_{3d}(\mathbb{R}^d)} \leq C_{\lambda,3d,2}(\mathcal{B}) \| g \|_{\mathcal{L}^1_{\lambda+6d}(\mathbb{R}^d)} \| f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)} .$$

Choosing $\eta = 3d$, $p = 2$ in (4.44), we have

$$\| Q^-(g,f) \|_{\mathcal{L}^2_{3d}(\mathbb{R}^d)} \leq C_\lambda \| g \|_{\mathcal{L}^1_{\lambda}(\mathbb{R}^d)} \| f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)} .$$

Combining both, we obtain

$$\| Q(g,f) \|_{\mathcal{L}^2_{3d}(\mathbb{R}^d)} \leq C_d(\mathcal{B}) \| g \|_{\mathcal{L}^1_{\lambda+6d}(\mathbb{R}^d)} \| f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)} .$$

□

Before we proceed to the consistency proof, we need the following lemmas.

Lemma 4.2.1. *Under the algebraic mapping (4.11) with $S = 1$, we have*

$$\| f \|_{\mathcal{L}^2_{\eta}(\mathbb{R}^d)} \leq \| f \|_{L^2_{\boldsymbol{\mu}^{-2\eta}}(\mathbb{R}^d)} \leq \| f \|_{\mathcal{L}^2_{d\eta}(\mathbb{R}^d)} \quad \text{for any } \eta \geq 0. \tag{4.46}$$

Proof. Note that

$$1 + \sum_{\mathrm{j}=1}^{d} |v_{\mathrm{j}}|^2 \leq \prod_{\mathrm{j}=1}^{d} (1 + |v_{\mathrm{j}}|^2) \leq (1 + \sum_{\mathrm{j}=1}^{d} |v_{\mathrm{j}}|^2)^d.$$

Then we have

$$\begin{aligned}
\| f \|_{L^2_{\boldsymbol{\mu}^{-2\eta}}(\mathbb{R}^d)} &= \left(\int_{\mathbb{R}^d} |f(\boldsymbol{v})|^2 \boldsymbol{\mu}^{-2\eta} \, \mathrm{d}\boldsymbol{v} \right)^{1/2} \\
&= \left(\int_{\mathbb{R}^d} |f(\boldsymbol{v})|^2 \prod_{\mathrm{j}=1}^{d} (1 + |v_{\mathrm{j}}|^2)^{\eta} \, \mathrm{d}\boldsymbol{v} \right)^{1/2} \\
&\geq \left(\int_{\mathbb{R}^d} |f(\boldsymbol{v})|^2 \left(1 + \sum_{\mathrm{j}=1}^{d} |v_{\mathrm{j}}|^2 \right)^{\eta} \, \mathrm{d}\boldsymbol{v} \right)^{1/2} = \| f \|_{\mathcal{L}^2_{\eta}(\mathbb{R}^d)} .
\end{aligned}$$

Also,

$$\begin{aligned}\| f \|_{L^2_{\boldsymbol{\mu}^{-2\eta}}(\mathbb{R}^d)} &= \left(\int_{\mathbb{R}^d}|f(\boldsymbol{v})|^2\prod_{\mathrm{j}=1}^{d}(1+|v_{\mathrm{j}}|^2)^{\eta}\,\mathrm{d}\boldsymbol{v}\right)^{1/2}\\ &\le \left(\int_{\mathbb{R}^d}|f(\boldsymbol{v})|^2\left(1+\sum_{\mathrm{j}=1}^{d}|v_{\mathrm{j}}|^2\right)^{d\eta}\,\mathrm{d}\boldsymbol{v}\right)^{1/2} =\| f \|_{\mathcal{L}^2_{d\eta}(\mathbb{R}^d)}\,.\end{aligned}$$

□

Lemma 4.2.2. *For any $\eta \ge 0$, there exist $\epsilon > 0$ and $C_\epsilon > 0$ such that*

$$\| f \|_{\mathcal{L}^1_\eta(\mathbb{R}^d)} \le C_\epsilon \| f \|_{\mathcal{L}^2_{\eta+\frac{1+\epsilon}{2}}(\mathbb{R}^d)}\,. \tag{4.47}$$

Proof. Note that

$$\begin{aligned}\| f \|^2_{\mathcal{L}^1_\eta(\mathbb{R}^d)} &= \left(\int_{\mathbb{R}^d}|f(\boldsymbol{v})|(1+|\boldsymbol{v}|^2)^{\frac{\eta}{2}}\,\mathrm{d}\boldsymbol{v}\right)^2\\ &\le \int_{\mathbb{R}^d}|f(\boldsymbol{v})|^2(1+|\boldsymbol{v}|^2)^{\eta+\frac{1+\epsilon}{2}}\,\mathrm{d}\boldsymbol{v}\int_{\mathbb{R}^d}(1+|\boldsymbol{v}|^2)^{-\frac{1+\epsilon}{2}}\,\mathrm{d}\boldsymbol{v}\\ &\le C_\epsilon\int_{\mathbb{R}^d}|f(\boldsymbol{v})|^2(1+|\boldsymbol{v}|^2)^{\eta+\frac{1+\epsilon}{2}}\,\mathrm{d}\boldsymbol{v}\\ &= C_\epsilon \| f \|^2_{\mathcal{L}^2_{\eta+\frac{1+\epsilon}{2}}(\mathbb{R}^d)},\end{aligned}$$

where we used the Cauchy-Schwarz inequality. □

We are ready to present a consistency result.

Theorem 2.3. *Let the collision kernel takes the form $\mathcal{B} = C_\lambda|\boldsymbol{v}-\boldsymbol{v}_*|^\lambda$, where $0 \le \lambda \le 1$ and C_λ is a positive constant. Then under the algebraic mapping (4.11) with $S = 1$, we have*

$$\begin{aligned}&\| Q(f,f) - \boldsymbol{\pi}_N^{4,d}Q(\boldsymbol{\pi}_N^{4,d}f, \boldsymbol{\pi}_N^{4,d}f) \|_{\mathcal{L}^2_3(\mathbb{R}^d)}\\ \le& C_{d,\epsilon}(\mathcal{B})N^{-m}\left(|f|_{\mathbf{B}^m_{\lambda+6d+\frac{3+\epsilon}{2}}(\mathbb{R}^d)}\| f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)} + |f|_{\mathbf{B}^m_{\lambda+3d+1}(\mathbb{R}^d)}\| f \|_{\mathcal{L}^1_{\lambda+6d}(\mathbb{R}^d)} + |Q(f,f)|_{\boldsymbol{B}^m_4(\mathbb{R}^d)}\right),\end{aligned} \tag{4.48}$$

where m is a positive integer, d is the dimension, $\epsilon > 0$ is a constant, and $C_{d,\epsilon}(\mathcal{B})$ is a constant depending only on the kernel $\mathcal{B}$, d and ϵ.

Proof. By the triangle inequality

$$\begin{aligned} & \| Q(f,f) - \pi_N^{4,d} Q(\pi_N^{4,d} f, \pi_N^{4,d} f) \|_{\mathcal{L}_3^2(\mathbb{R}^d)} \\ \leq & \| Q(f,f) - \pi_N^{4,d} Q(f,f) \|_{\mathcal{L}_3^2(\mathbb{R}^d)} + \| \pi_N^{4,d} Q(f,f) - \pi_N^{4,d} Q(\pi_N^{4,d} f, \pi_N^{4,d} f) \|_{\mathcal{L}_3^2(\mathbb{R}^d)} . \end{aligned}$$

For the first term, by Lemma 4.2.1 and Theorem 1.1, we have

$$\| Q(f,f) - \pi_N^{4,d} Q(f,f) \|_{\mathcal{L}_3^2(\mathbb{R}^d)} \leq \| Q(f,f) - \pi_N^{4,d} Q(f,f) \|_{L^2_{\boldsymbol{\mu}^{-6}}(\mathbb{R}^d)} \leq C N^{-m} |Q(f,f)|_{\boldsymbol{B}_4^m(\mathbb{R}^d)}.$$

For the second term, using again Lemma 4.2.1, Theorem 1.1, and Lemma 4.2.2, Theorem 2.2, we have

$$\begin{aligned} & \| \pi_N^{4,d} Q(f,f) - \pi_N^{4,d} Q(\pi_N^{4,d} f, \pi_N^{4,d} f) \|_{\mathcal{L}_3^2(\mathbb{R}^d)} \leq \| \pi_N^{4,d} Q(f,f) - \pi_N^{4,d} Q(\pi_N^{4,d} f, \pi_N^{4,d} f) \|_{L^2_{\boldsymbol{\mu}^{-6}}(\mathbb{R}^d)} \\ \leq & \| Q(f,f) - Q(\pi_N^{4,d} f, \pi_N^{4,d} f) \|_{L^2_{\boldsymbol{\mu}^{-6}}(\mathbb{R}^d)} \leq \| Q(f,f) - Q(\pi_N^{4,d} f, \pi_N^{4,d} f) \|_{\mathcal{L}_{3d}^2(\mathbb{R}^d)} \\ \leq & \| Q(f - \pi_N^{4,d} f, f) \|_{\mathcal{L}_{3d}^2(\mathbb{R}^d)} + \| Q(\pi_N^{4,d} f, f - \pi_N^{4,d} f) \|_{\mathcal{L}_{3d}^2(\mathbb{R}^d)} \\ \leq & C_d(\mathcal{B}) \left(\| f - \pi_N^{4,d} f \|_{\mathcal{L}^1_{\lambda+6d}(\mathbb{R}^d)} \| f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)} + \| \pi_N^{4,d} f \|_{\mathcal{L}^1_{\lambda+6d}(\mathbb{R}^d)} \| f - \pi_N^{4,d} f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)} \right) \\ \leq & C_d(\mathcal{B}) \left(\| f - \pi_N^{4,d} f \|_{\mathcal{L}^1_{\lambda+6d}(\mathbb{R}^d)} (\| f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)} + \| f - \pi_N^{4,d} f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)}) + \| f \|_{\mathcal{L}^1_{\lambda+6d}(\mathbb{R}^d)} \| f - \pi_N^{4,d} f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)} \right. \\ \leq & C_{d,\epsilon}(\mathcal{B}) \| f - \pi_N^{4,d} f \|_{\mathcal{L}^2_{\lambda+6d+\frac{1+\epsilon}{2}}(\mathbb{R}^d)} \left(\| f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)} + \| f - \pi_N^{4,d} f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)} \right) \\ & + C_d(\mathcal{B}) \| f \|_{\mathcal{L}^1_{\lambda+6d}(\mathbb{R}^d)} \| f - \pi_N^{4,d} f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)} \\ \leq & C_{d,\epsilon}(\mathcal{B}) \| f - \pi_N^{4,d} f \|_{L^2_{\boldsymbol{\mu}^{-2(\lambda+6d+\frac{1+\epsilon}{2})}}(\mathbb{R}^d)} (\| f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)} + \| f - \pi_N^{4,d} f \|_{L^2_{\boldsymbol{\mu}^{-2(\lambda+3d)}}(\mathbb{R}^d)}) \\ & + C_d(\mathcal{B}) \| f \|_{\mathcal{L}^1_{\lambda+6d}(\mathbb{R}^d)} \| f - \pi_N^{4,d} f \|_{L^2_{\boldsymbol{\mu}^{-2(\lambda+3d)}}(\mathbb{R}^d)} \\ \leq & C_{d,\epsilon}(\mathcal{B}) N^{-m} |f|_{\mathbf{B}^m_{\lambda+6d+\frac{3+\epsilon}{2}}(\mathbb{R}^d)} \left(\| f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)} + C N^{-m} |f|_{\mathbf{B}^m_{\lambda+3d+1}(\mathbb{R}^d)} \right) \\ & + C_d(\mathcal{B}) N^{-m} \| f \|_{\mathcal{L}^1_{\lambda+6d}(\mathbb{R}^d)} |f|_{\mathbf{B}^m_{\lambda+3d+1}(\mathbb{R}^d)} \\ \leq & C_{d,\epsilon}(\mathcal{B}) N^{-m} \left(|f|_{\mathbf{B}^m_{\lambda+6d+\frac{3+\epsilon}{2}}(\mathbb{R}^d)} \| f \|_{\mathcal{L}^2_{\lambda+3d}(\mathbb{R}^d)} + |f|_{\mathbf{B}^m_{\lambda+3d+1}(\mathbb{R}^d)} \| f \|_{\mathcal{L}^1_{\lambda+6d}(\mathbb{R}^d)} \right). \end{aligned}$$

Combining the above inequalities, we arrive at the desired result. □

4.2.2 Approximation property for the moments

For the Boltzmann equation, the moments or macroscopic observables are important physical quantities. Still, under the algebraic mapping (4.11), we can show that the spectral method (4.35) preserves mass and energy.

Theorem 2.4. *If using the algebraic mapping (4.11) with $N \geq 2$, the spectral method (4.35) preserves mass and energy, i.e., $\rho(t)$ and $E(t)$ defined by*

$$\rho(t) := \int_{\mathbb{R}^d} f_N(t, \boldsymbol{v}) \, \mathrm{d}\boldsymbol{v}, \quad E(t) = \int_{\mathbb{R}^d} f_N(t, \boldsymbol{v}) |\boldsymbol{v}|^2 \, \mathrm{d}\boldsymbol{v} \tag{4.49}$$

remain constant in time. Furthermore,

$$\rho(t) \equiv \int_{\mathbb{R}^d} f^0(\boldsymbol{v}) \, \mathrm{d}\boldsymbol{v}, \quad E(t) \equiv \int_{\mathbb{R}^d} f^0(\boldsymbol{v}) |\boldsymbol{v}|^2 \, \mathrm{d}\boldsymbol{v}. \tag{4.50}$$

Proof. In 1D, the first few Chebyshev polynomials read

$$T_0(\xi) = 1, \quad T_1(\xi) = \xi, \quad T_2(\xi) = 2\xi^2 - 1.$$

With the algebraic mapping (4.11), we have

$$T_0(\xi(v)) = 1, \quad T_1(\xi(v)) = \frac{v}{\sqrt{v^2 + S^2}}, \quad T_2(\xi(v)) = \frac{v^2 - S^2}{v^2 + S^2}, \quad \mu(\xi(v)) = \frac{\sqrt{S}}{\sqrt{v^2 + S^2}}.$$

Then

$$\widehat{T}_k(v) = \frac{[\mu(\xi(v))]^{-2}}{\sqrt{c_k}} T_k(\xi(v)) = \frac{v^2 + S^2}{\sqrt{c_k} S} T_k(\xi(v)).$$

Specifically,

$$\widehat{T}_0(v) = \frac{v^2 + S^2}{\sqrt{c_0} S}, \quad \widehat{T}_1(v) = \frac{v\sqrt{v^2 + S^2}}{\sqrt{c_1} S}, \quad \widehat{T}_2(v) = \frac{v^2 - S^2}{\sqrt{c_2} S}.$$

Therefore,

$$1 = \frac{\sqrt{c_0}}{2S} \widehat{T}_0(v) - \frac{\sqrt{c_2}}{2S} \widehat{T}_2(v), \quad v^2 = \frac{\sqrt{c_0} S}{2} \widehat{T}_0(v) + \frac{\sqrt{c_2} S}{2} \widehat{T}_2(v).$$

Hence we can replace $(\widehat{T}_0(v), \widehat{T}_2(v))$ by $(1, v^2)$ as basis functions, namely,

$$\widehat{\mathbb{T}}_N^1 = \text{span}\{1, \widehat{T}_1, v^2, \widehat{T}_3, \widehat{T}_4, \cdots, \widehat{T}_N\}.$$

In d dimensions, it is easy to see

$$1,\ v_1^2,\ v_2^2,\ \ldots,\ v_d^2 \in \widehat{\mathbb{T}}_N^d \quad \text{for } N \geq 2.$$

In other words, we have shown that $1, |\boldsymbol{v}|^2 \in \widehat{\mathbb{T}}_N^d$ for $N \geq 2$.

On the other hand, by (4.49) and (4.34), we have

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\rho(t) =& (\partial_t f_N(t, \boldsymbol{v}), 1)_{\mathbb{R}^d} = (Q(f_N, f_N), 1)_{\mathbb{R}^d} = 0; \\
\frac{\mathrm{d}}{\mathrm{d}t}E(t) =& (\partial_t f_N(t, \boldsymbol{v}), |\boldsymbol{v}|^2)_{\mathbb{R}^d} = \left(Q(f_N, f_N), |\boldsymbol{v}|^2\right)_{\mathbb{R}^d} = 0,
\end{aligned}$$

where in the last equality we used the conservation property (1.8) of the collision operator.

It remains to show

$$\int_{\mathbb{R}^d} f_N(0, \boldsymbol{v})\ \mathrm{d}\boldsymbol{v} = \int_{\mathbb{R}^d} f^0(\boldsymbol{v})\ \mathrm{d}\boldsymbol{v}, \quad \int_{\mathbb{R}^d} f_N(0, \boldsymbol{v})|\boldsymbol{v}|^2\ \mathrm{d}\boldsymbol{v} = \int_{\mathbb{R}^d} f^0(\boldsymbol{v})|\boldsymbol{v}|^2\ \mathrm{d}\boldsymbol{v}.$$

Note that $f_N(0, \boldsymbol{v}) = \boldsymbol{\pi}^{4,d} f^0$, it suffices to show

$$(\boldsymbol{\pi}^{4,d} f^0 - f^0, 1)_{\mathbb{R}^d} = (\boldsymbol{\pi}^{4,d} f^0 - f^0, |\boldsymbol{v}|^2)_{\mathbb{R}^d} = 0,$$

which is true by (4.25) (with $\alpha = 4$). □

4.3 Numerical realization

To implement the proposed spectral method, one needs to solve the ODE system (4.35). For time discretization, one can just use the explicit Runge-Kutta methods. Hence, the key is the efficient evaluation of $\mathcal{Q}_{\boldsymbol{k}}^N$ as defined in (4.36).

In this section, we introduce two algorithms to compute $\mathcal{Q}_{\boldsymbol{k}}^N$. The first one is a direct algorithm that treats $\mathcal{Q}_{\boldsymbol{k}}^N$ as a matrix/tensor-vector multiplication. Since the weight ma-

trix/tensor does not depend on the numerical solution f_N, it can be precomputed and stored for repeated use. This approach is simple but will soon meet a bottleneck when N increases since the memory requirement as well as the online computational cost can get extremely high. To alleviate this, we propose a fast algorithm, where the key idea is to recognize the gain term of the collision operator as a non-uniform discrete Fourier cosine transform to be accelerated by the non-uniform FFT (NUFFT). Note that this is possible because we are using the mapped Chebyshev functions as a basis, which is related to the Fourier cosine series.

4.3.1 A direct algorithm

To derive the direct algorithm, we substitute (4.33) into (4.36) to obtain

$$\begin{aligned} \mathcal{Q}_{\boldsymbol{k}}^N &= \sum_{0\le \mathbf{i},\mathbf{j}\le N} \widetilde{f}_{\mathbf{i}}\widetilde{f}_{\mathbf{j}} \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\int_{S^{d-1}} \mathcal{B}(\boldsymbol{v},\boldsymbol{v}_*,\boldsymbol{\sigma})\, \widetilde{\boldsymbol{T}}_{\mathbf{i}}(\boldsymbol{v})\widetilde{\boldsymbol{T}}_{\mathbf{j}}(\boldsymbol{v}_*)[\widehat{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v}') - \widehat{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v})]\,\mathrm{d}\boldsymbol{\sigma}\,\mathrm{d}\boldsymbol{v}\,\mathrm{d}\boldsymbol{v}_* \\ &= \sum_{0\le \mathbf{i},\mathbf{j}\le N} \widetilde{f}_{\mathbf{i}}\widetilde{f}_{\mathbf{j}} \left[\widetilde{I}_1(\mathbf{i},\mathbf{j},\boldsymbol{k}) - \widetilde{I}_2(\mathbf{i},\mathbf{j},\boldsymbol{k})\right], \quad 0\le \boldsymbol{k}\le N, \end{aligned} \tag{4.51}$$

where

$$\widetilde{I}_1(\mathbf{i},\mathbf{j},\boldsymbol{k}) := \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\int_{S^{d-1}} \mathcal{B}(\boldsymbol{v},\boldsymbol{v}_*,\boldsymbol{\sigma})\, \widetilde{\boldsymbol{T}}_{\mathbf{i}}(\boldsymbol{v})\widetilde{\boldsymbol{T}}_{\mathbf{j}}(\boldsymbol{v}_*)\widehat{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v}')\,\mathrm{d}\boldsymbol{\sigma}\,\mathrm{d}\boldsymbol{v}\,\mathrm{d}\boldsymbol{v}_*, \tag{4.52}$$

$$\widetilde{I}_2(\mathbf{i},\mathbf{j},\boldsymbol{k}) := \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\int_{S^{d-1}} \mathcal{B}(\boldsymbol{v},\boldsymbol{v}_*,\boldsymbol{\sigma})\, \widetilde{\boldsymbol{T}}_{\mathbf{i}}(\boldsymbol{v})\widetilde{\boldsymbol{T}}_{\mathbf{j}}(\boldsymbol{v}_*)\widehat{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v})\,\mathrm{d}\boldsymbol{\sigma}\,\mathrm{d}\boldsymbol{v}\,\mathrm{d}\boldsymbol{v}_*. \tag{4.53}$$

Since the tensors $\widetilde{I}_1(\mathbf{i},\mathbf{j},\boldsymbol{k})$ and $\widetilde{I}_2(\mathbf{i},\mathbf{j},\boldsymbol{k})$ do not depend on coefficients $\{\widetilde{f}_{\boldsymbol{k}}\}_{0\le\boldsymbol{k}\le N}$, a straightforward way to evaluate $\mathcal{Q}_{\boldsymbol{k}}^N$ is to precompute $\widetilde{I}_1(\mathbf{i},\mathbf{j},\boldsymbol{k})$ and $\widetilde{I}_2(\mathbf{i},\mathbf{j},\boldsymbol{k})$, and then evaluate the sum in (4.51) directly in the online computation. This is what we refer to as the direct algorithm. We observe that this algorithm requires $\mathcal{O}(N^{3d})$ memory to store the tensors $\widetilde{I}_1$ and $\widetilde{I}_2$; and to evaluate (4.51), it requires $\mathcal{O}(N^{3d})$ operations. Both the memory requirement and online computational cost can be quite demanding especially for $d=3$ and large N.

We give some details on how to approximate $\widetilde{I}_1(\mathbf{i},\mathbf{j},\boldsymbol{k})$ and $\widetilde{I}_2(\mathbf{i},\mathbf{j},\boldsymbol{k})$, though this step can be completed in advance and does not take the actual computational time. We first perform

a change of variables $(\boldsymbol{v}, \boldsymbol{v}_*) \to (\boldsymbol{v}(\boldsymbol{\xi}), \boldsymbol{v}_*(\boldsymbol{\eta}))$ to transform the integrals of $(\boldsymbol{v}, \boldsymbol{v}_*) \in \mathbb{R}^d \times \mathbb{R}^d$ into integrals of $(\boldsymbol{\xi}, \boldsymbol{\eta}) \in I^d \times I^d$, using the mapping introduced in Section 4.1.1:

$$\widetilde{I}_1(\mathbf{i}, \mathbf{j}, \boldsymbol{k}) = \int_{I^d} \int_{I^d} G_{\boldsymbol{k}}(\boldsymbol{\xi}, \boldsymbol{\eta}) \frac{\boldsymbol{T}_{\mathbf{i}}(\boldsymbol{\xi}) \boldsymbol{T}_{\mathbf{j}}(\boldsymbol{\eta})}{\sqrt{\boldsymbol{c}_{\mathbf{i}} \boldsymbol{c}_{\mathbf{j}}}} \boldsymbol{\omega}(\boldsymbol{\xi}) \boldsymbol{\omega}(\boldsymbol{\eta}) \, \mathrm{d}\boldsymbol{\xi} \, \mathrm{d}\boldsymbol{\eta}, \tag{4.54}$$

$$\widetilde{I}_2(\mathbf{i}, \mathbf{j}, \boldsymbol{k}) = \int_{I^d} \int_{I^d} L_{\boldsymbol{k}}(\boldsymbol{\xi}, \boldsymbol{\eta}) \frac{\boldsymbol{T}_{\mathbf{i}}(\boldsymbol{\xi}) \boldsymbol{T}_{\mathbf{j}}(\boldsymbol{\eta})}{\sqrt{\boldsymbol{c}_{\mathbf{i}} \boldsymbol{c}_{\mathbf{j}}}} \boldsymbol{\omega}(\boldsymbol{\xi}) \boldsymbol{\omega}(\boldsymbol{\eta}) \, \mathrm{d}\boldsymbol{\xi} \, \mathrm{d}\boldsymbol{\eta}, \tag{4.55}$$

where

$$G_{\boldsymbol{k}}(\boldsymbol{\xi}, \boldsymbol{\eta}) := [\boldsymbol{\mu}(\boldsymbol{\xi}) \boldsymbol{\mu}(\boldsymbol{\eta})]^2 \int_{S^{d-1}} \mathcal{B}(\boldsymbol{v}(\boldsymbol{\xi}), \boldsymbol{v}_*(\boldsymbol{\eta}), \boldsymbol{\sigma}) \frac{\boldsymbol{T}_{\boldsymbol{k}} \left(\boldsymbol{\zeta}_{(\boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\sigma})} \right)}{\sqrt{\boldsymbol{c}_{\boldsymbol{k}}} [\boldsymbol{\mu} \left(\boldsymbol{\zeta}_{(\boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\sigma})} \right)]^2} \, \mathrm{d}\boldsymbol{\sigma}, \tag{4.56}$$

$$L_{\boldsymbol{k}}(\boldsymbol{\xi}, \boldsymbol{\eta}) := \frac{\boldsymbol{T}_{\boldsymbol{k}}(\boldsymbol{\xi}) \left[\boldsymbol{\mu}(\boldsymbol{\eta}) \right]^2}{\sqrt{\boldsymbol{c}_{\boldsymbol{k}}}} \int_{S^{d-1}} \mathcal{B}(\boldsymbol{v}(\boldsymbol{\xi}), \boldsymbol{v}_*(\boldsymbol{\eta}), \boldsymbol{\sigma}) \, \mathrm{d}\boldsymbol{\sigma}. \tag{4.57}$$

Notice that in (4.56), $\boldsymbol{\zeta}_{(\boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\sigma})} \in I^d$ is the value transformed from

$$\boldsymbol{v}' = \frac{1}{2}(\boldsymbol{v}(\boldsymbol{\xi}) + \boldsymbol{v}_*(\boldsymbol{\eta})) + \frac{1}{2}|\boldsymbol{v}(\boldsymbol{\xi}) - \boldsymbol{v}_*(\boldsymbol{\eta})| \boldsymbol{\sigma} \in \mathbb{R}^d$$

under the same mapping. To approximate the above integrals in $\boldsymbol{\xi}$, $\boldsymbol{\eta}$, and $\boldsymbol{\sigma}$, we choose $M_{\boldsymbol{v}}$ Chebyshev-Gauss-Lobatto quadrature points in each dimension of I^d for both $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$; and $M_{\boldsymbol{\sigma}}$ quadrature points on the unit sphere S^{d-1} (for $d = 2$, this can be the uniform points in polar angle; for $d = 3$, this can be the Lebedev quadrature [54]). Therefore, for each fixed index $\boldsymbol{k}$, (4.54) and (4.55) are forward Chebyshev transforms of the functions $G_{\boldsymbol{k}}(\boldsymbol{\xi}, \boldsymbol{\eta})$ and $L_{\boldsymbol{k}}(\boldsymbol{\xi}, \boldsymbol{\eta})$, respectively. Thus, they can be evaluated efficiently using the fast Chebyshev transform.

4.3.2 A fast algorithm

To introduce the fast algorithm, we take the original form (4.36) and split $\mathcal{Q}_{\boldsymbol{k}}^N$ into a gain term and a loss term as $\mathcal{Q}_{\boldsymbol{k}}^N = \mathcal{Q}_{\boldsymbol{k}}^{N,+} - \mathcal{Q}_{\boldsymbol{k}}^{N,-}$, where

$$\mathcal{Q}_{\boldsymbol{k}}^{N,+} = \int_{\mathbb{R}^d} \left(\int_{\mathbb{R}^d} \int_{S^{d-1}} \mathcal{B}(\boldsymbol{v}, \boldsymbol{v}_*, \boldsymbol{\sigma}) f_N(\boldsymbol{v}_*) \widehat{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v}') \, \mathrm{d}\boldsymbol{\sigma} \, \mathrm{d}\boldsymbol{v}_* \right) f_N(\boldsymbol{v}) \, \mathrm{d}\boldsymbol{v}, \tag{4.58}$$

$$\mathcal{Q}_{\boldsymbol{k}}^{N,-} = \int_{\mathbb{R}^d} \left(\int_{\mathbb{R}^d} \int_{S^{d-1}} \mathcal{B}(\boldsymbol{v}, \boldsymbol{v}_*, \boldsymbol{\sigma}) f_N(\boldsymbol{v}_*) \, \mathrm{d}\boldsymbol{\sigma} \, \mathrm{d}\boldsymbol{v}_* \right) f_N(\boldsymbol{v}) \widehat{\boldsymbol{T}}_{\boldsymbol{k}}(\boldsymbol{v}) \, \mathrm{d}\boldsymbol{v}. \tag{4.59}$$

We propose to evaluate $\mathcal{Q}_{\boldsymbol{k}}^{N,+}$ and $\mathcal{Q}_{\boldsymbol{k}}^{N,-}$ following the above expressions. To this end, given the coefficients $\{\widetilde{f}_{\boldsymbol{k}}\}_{0 \le \boldsymbol{k} \le N}$ at each time step, we first reconstruct f_N as in (4.33) at $M_{\boldsymbol{v}}$ Chebyshev-Gauss-Lobatto quadrature points in each dimension of $\boldsymbol{v}$ (for an accurate approximation we choose $M_{\boldsymbol{v}} = N + 2$). This can be achieved by the fast Chebyshev transform in $\mathcal{O}(M_{\boldsymbol{v}}^d \log M_{\boldsymbol{v}})$ operations.

To evaluate the gain term $\mathcal{Q}_{\boldsymbol{k}}^{N,+}$, we change the integrals of $(\boldsymbol{v}, \boldsymbol{v}_*) \in \mathbb{R}^d \times \mathbb{R}^d$ into integrals of $(\boldsymbol{\xi}, \boldsymbol{\eta}) \in I^d \times I^d$ in (4.58) (similarly as in the previous subsection for $\widetilde{I}_1(\mathbf{i}, \mathbf{j}, \boldsymbol{k})$):

$$\begin{aligned} \mathcal{Q}_{\boldsymbol{k}}^{N,+} &= \int_{I^d} \left(\int_{I^d} \int_{S^{d-1}} \mathcal{B}(\boldsymbol{v}(\boldsymbol{\xi}), \boldsymbol{v}_*(\boldsymbol{\eta}), \boldsymbol{\sigma}) f_N(\boldsymbol{v}_*(\boldsymbol{\eta})) \frac{\boldsymbol{T}_{\boldsymbol{k}}\left(\boldsymbol{\zeta}_{(\xi,\eta,\sigma)}\right)}{\sqrt{\boldsymbol{c}_{\boldsymbol{k}}} \left[\boldsymbol{\mu}\left(\boldsymbol{\zeta}_{(\xi,\eta,\sigma)}\right)\right]^2} \frac{\boldsymbol{\omega}(\boldsymbol{\eta})}{[\boldsymbol{\mu}(\boldsymbol{\eta})]^2} \, \mathrm{d}\boldsymbol{\sigma} \, \mathrm{d}\boldsymbol{\eta} \right) f_N(\boldsymbol{v}(\boldsymbol{\xi})) \frac{\boldsymbol{\omega}(\boldsymbol{\xi})}{[\boldsymbol{\mu}(\boldsymbol{\xi})]^2} \, \mathrm{d}\boldsymbol{\xi} \\ &= \int_{I^d} \left(\int_{S^{d-1}} F_{\boldsymbol{k}}(\boldsymbol{\sigma}, \boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\sigma} \right) f_N(\boldsymbol{v}(\boldsymbol{\xi})) \frac{\boldsymbol{\omega}(\boldsymbol{\xi})}{[\boldsymbol{\mu}(\boldsymbol{\xi})]^2} \, \mathrm{d}\boldsymbol{\xi}, \end{aligned} \tag{4.60}$$

where

$$F_{\boldsymbol{k}}(\boldsymbol{\sigma}, \boldsymbol{\xi}) := \int_{I^d} \mathcal{B}(\boldsymbol{v}(\boldsymbol{\xi}), \boldsymbol{v}_*(\boldsymbol{\eta}), \boldsymbol{\sigma}) f_N(\boldsymbol{v}_*(\boldsymbol{\eta})) \frac{\boldsymbol{T}_{\boldsymbol{k}}\left(\boldsymbol{\zeta}_{(\xi,\eta,\sigma)}\right)}{\sqrt{\boldsymbol{c}_{\boldsymbol{k}}} \left[\boldsymbol{\mu}\left(\boldsymbol{\zeta}_{(\xi,\eta,\sigma)}\right)\right]^2} \frac{\boldsymbol{\omega}(\boldsymbol{\eta})}{[\boldsymbol{\mu}(\boldsymbol{\eta})]^2} \, \mathrm{d}\boldsymbol{\eta}. \tag{4.61}$$

Suppose $M_{\boldsymbol{v}}$ quadrature points are used in each dimension of $\boldsymbol{v}$ and $\boldsymbol{v}_*$ and $M_{\boldsymbol{\sigma}}$ points are used on the sphere S^{d-1}, a direct evaluation of (4.61) would require $\mathcal{O}(M_{\boldsymbol{\sigma}} M_{\boldsymbol{v}}^{2d} N^d)$ operations. Given $F_{\boldsymbol{k}}(\boldsymbol{\sigma}, \boldsymbol{\xi})$, a direct evaluation of (4.60) would take $\mathcal{O}(M_{\boldsymbol{\sigma}} M_{\boldsymbol{v}}^d N^d)$ operations. Therefore, the major bottleneck is to compute $F_{\boldsymbol{k}}(\boldsymbol{\sigma}, \boldsymbol{\xi})$, which is prohibitively expensive without a fast algorithm. Our main idea is to recognize (4.61) as a non-uniform discrete

Fourier cosine transform so it can be evaluated by the non-uniform fast Fourier transform (NUFFT). We will see that the total complexity to evaluate $F_{\boldsymbol{k}}(\boldsymbol{\sigma},\boldsymbol{\xi})$ can be brought down to $\mathcal{O}(M_{\boldsymbol{\sigma}}M_v^{2d}|\log\epsilon| + M_{\boldsymbol{\sigma}}M_v^d N^d \log N)$, where ϵ is the requested precision in the NUFFT algorithm.

Applying the Chebyshev-Gauss-Lobatto quadrature $(\boldsymbol{\eta}_{\mathbf{j}}, w_{\mathbf{j}})_{1\le\mathbf{j}\le M_v}$, (4.61) becomes

$$\begin{aligned} F_{\boldsymbol{k}}(\boldsymbol{\sigma},\boldsymbol{\xi}) &\approx \sum_{1\le\mathbf{j}\le M_v} w_{\mathbf{j}} \frac{\mathcal{B}(\boldsymbol{v}(\boldsymbol{\xi}),\boldsymbol{v}_*(\boldsymbol{\eta}_{\mathbf{j}}),\boldsymbol{\sigma}) f_N(\boldsymbol{v}_*(\boldsymbol{\eta}_{\mathbf{j}}))}{\sqrt{c_{\boldsymbol{k}}}[\boldsymbol{\mu}(\boldsymbol{\zeta}_{(\boldsymbol{\xi},\boldsymbol{\eta}_{\mathbf{j}},\boldsymbol{\sigma})})]^2[\boldsymbol{\mu}(\boldsymbol{\eta}_{\mathbf{j}})]^2} \boldsymbol{T}_{\boldsymbol{k}}(\boldsymbol{\zeta}_{(\boldsymbol{\xi},\boldsymbol{\eta}_{\mathbf{j}},\boldsymbol{\sigma})}) \\ &= \sum_{1\le\mathbf{j}\le M_v} w_{\mathbf{j}} \frac{\mathcal{B}(\boldsymbol{v}(\boldsymbol{\xi}),\boldsymbol{v}_*(\boldsymbol{\eta}_{\mathbf{j}}),\boldsymbol{\sigma}) f_N(\boldsymbol{v}_*(\boldsymbol{\eta}_{\mathbf{j}}))}{\sqrt{c_{\boldsymbol{k}}}[\boldsymbol{\mu}(\boldsymbol{\zeta}_{(\boldsymbol{\xi},\boldsymbol{\eta}_{\mathbf{j}},\boldsymbol{\sigma})})]^2[\boldsymbol{\mu}(\boldsymbol{\eta}_{\mathbf{j}})]^2} \prod_{l=1}^{d} \cos\left(k_l \arccos(\boldsymbol{\zeta}_{(\boldsymbol{\xi},\boldsymbol{\eta}_{\mathbf{j}},\boldsymbol{\sigma}),l})\right) \\ &= \frac{1}{\sqrt{c_{\boldsymbol{k}}}} \sum_{1\le\mathbf{j}\le M_v} q_{\mathbf{j}} \prod_{l=1}^{d} \cos\left(k_l \boldsymbol{z}_{\mathbf{j},l}\right), \end{aligned} \tag{4.62}$$

where $\boldsymbol{\zeta}_{(\boldsymbol{\xi},\boldsymbol{\eta}_{\mathbf{j}},\boldsymbol{\sigma}),l}$ is the l-th component of $\boldsymbol{\zeta}_{(\boldsymbol{\xi},\boldsymbol{\eta}_{\mathbf{j}},\boldsymbol{\sigma})}$, and

$$q_{\mathbf{j}} := w_{\mathbf{j}} \frac{\mathcal{B}(\boldsymbol{v}(\boldsymbol{\xi}),\boldsymbol{v}_*(\boldsymbol{\eta}_{\mathbf{j}}),\boldsymbol{\sigma}) f_N(\boldsymbol{v}_*(\boldsymbol{\eta}_{\mathbf{j}}))}{[\boldsymbol{\mu}(\boldsymbol{\zeta}_{(\boldsymbol{\xi},\boldsymbol{\eta}_{\mathbf{j}},\boldsymbol{\sigma})})]^2[\boldsymbol{\mu}(\boldsymbol{\eta}_{\mathbf{j}})]^2}, \quad \boldsymbol{z}_{\mathbf{j},l} := \arccos(\boldsymbol{\zeta}_{(\boldsymbol{\xi},\boldsymbol{\eta}_{\mathbf{j}},\boldsymbol{\sigma}),l}). \tag{4.63}$$

Note that $q_{\mathbf{j}}$ and $\boldsymbol{z}_{\mathbf{j}}$ depend on $\boldsymbol{\sigma}$ and $\boldsymbol{\xi}$. (4.62) is almost like a non-uniform discrete Fourier cosine transform. Indeed, we propose to evaluate $F_{\boldsymbol{k}}(\boldsymbol{\sigma},\boldsymbol{\xi})$ as follows.

For each fixed $\boldsymbol{\sigma}$ and $\boldsymbol{\xi}$, we compute

$$\widetilde{F}_{\boldsymbol{K}} := \sum_{1\le\mathbf{j}\le M_v} q_{\mathbf{j}} \exp(\mathbf{i}\boldsymbol{K}\cdot\boldsymbol{z}_{\mathbf{j}}), \quad -N \le \boldsymbol{K} \le N, \tag{4.64}$$

which is a non-uniform discrete Fourier transform mapping non-uniform samples $\boldsymbol{z}_{\mathbf{j}} \in [0,\pi]^d$ into frequencies $\boldsymbol{K} \in [-N, N]^d$. This can be done efficiently using the NUFFT. In recent years, various NUFFT algorithms have been developed. In our numerical realization, we employ an efficient algorithm called FINUFFT [55]. The general idea is to apply an interpolation between non-uniform samples and an equispaced grid, and then perform the uniform FFT on the new grid. This algorithm only costs $\mathcal{O}(M_v^d|\log\epsilon| + N^d \log N)$ operations in computing (4.64) with the requested precision ϵ. Once we obtain $\widetilde{F}_{\boldsymbol{K}}$, $F_{\boldsymbol{k}}(\boldsymbol{\sigma},\boldsymbol{\xi})$ can be retrieved as

- in 2D

$$F_{\boldsymbol{k}}(\boldsymbol{\sigma},\boldsymbol{\xi}) = \frac{1}{\sqrt{c_{\boldsymbol{k}}}}\frac{1}{2}\mathbf{Re}\left(\widetilde{F}_{(k_1,k_2)} + \widetilde{F}_{(-k_1,k_2)}\right); \tag{4.65}$$

- in 3D

$$F_{\boldsymbol{k}}(\boldsymbol{\sigma},\boldsymbol{\xi}) = \frac{1}{\sqrt{c_{\boldsymbol{k}}}}\frac{1}{4}\mathbf{Re}\left(\widetilde{F}_{(k_1,k_2,k_3)} + \widetilde{F}_{(-k_1,k_2,k_3)} + \widetilde{F}_{(k_1,-k_2,k_3)} + \widetilde{F}_{(k_1,k_2,-k_3)}\right). \tag{4.66}$$

This procedure needs to be repeated for every $\boldsymbol{\sigma}$ and $\boldsymbol{\xi}$, hence the overall computational cost for getting $F_{\boldsymbol{k}}(\boldsymbol{\sigma},\boldsymbol{\xi})$ is $\mathcal{O}(M_{\boldsymbol{\sigma}}M_{\boldsymbol{v}}^{2d}|\log\epsilon| + M_{\boldsymbol{\sigma}}M_{\boldsymbol{v}}^{d}N^d\log N)$.

To evaluate the loss term $\mathcal{Q}_{\boldsymbol{k}}^{N,-}$, we change the integrals of $(\boldsymbol{v},\boldsymbol{v}_*) \in \mathbb{R}^d \times \mathbb{R}^d$ into integrals of $(\boldsymbol{\xi},\boldsymbol{\eta}) \in I^d \times I^d$ in (4.59) (similarly as in the previous subsection for $\widetilde{I}_2(\mathbf{i},\mathbf{j},\boldsymbol{k})$):

$$\mathcal{Q}_{\boldsymbol{k}}^{N,-} = \int_{I^d}\left(\int_{I^d}\int_{S^{d-1}} \mathcal{B}(\boldsymbol{v}(\boldsymbol{\xi}),\boldsymbol{v}_*(\boldsymbol{\eta}),\boldsymbol{\sigma}) f_N(\boldsymbol{v}_*(\boldsymbol{\eta}))\frac{\boldsymbol{\omega}(\boldsymbol{\eta})}{[\boldsymbol{\mu}(\boldsymbol{\eta})]^2}\,\mathrm{d}\boldsymbol{\sigma}\,\mathrm{d}\boldsymbol{\eta}\right) f_N(\boldsymbol{v}(\boldsymbol{\xi}))\,\frac{\boldsymbol{T}_{\boldsymbol{k}}(\boldsymbol{\xi})\boldsymbol{\omega}(\boldsymbol{\xi})}{\sqrt{c_{\boldsymbol{k}}}\,[\boldsymbol{\mu}(\boldsymbol{\xi})]^4}\,\mathrm{d}\boldsymbol{\xi}. \tag{4.67}$$

Then one can just evaluate the terms in the parentheses directly with complexity $\mathcal{O}(M_{\boldsymbol{v}}^{2d}M_{\boldsymbol{\sigma}})$. The outer integral in $\boldsymbol{\xi}$ can be viewed as the Chebyshev transform of some function thus can be evaluated efficiently by the fast Chebyshev transform in $\mathcal{O}(M_{\boldsymbol{v}}^d \log M_{\boldsymbol{v}})$. In particular, if we consider the Maxwell kernel, i.e., $\mathcal{B}(\boldsymbol{v},\boldsymbol{v}_*,\boldsymbol{\sigma}) \equiv$ constant, terms in the parentheses only requires $\mathcal{O}(M_{\boldsymbol{v}}^d)$ complexity.

4.3.3 Comparison of direct and fast algorithms

To summarize, we list the storage requirement and (online) computational complexity for both the direct and fast algorithms in Table 1. Note that we only list the dominant complexity for each term. It is clear that the main cost of the fast algorithm comes from evaluating the gain term. Compared with the direct algorithm, the fast algorithm is generally faster as $M_{\boldsymbol{\sigma}}$ can be chosen much smaller than N^d in practice (see Section 4.4). Most importantly, the fast algorithm does not require any precomputation with excessive storage requirement and everything can be computed on the fly.

Table 4.1. Storage requirement and (online) computational cost of the direct and fast algorithms. N is the number of spectral modes in each dimension of $\boldsymbol{v}$; $M_{\boldsymbol{v}} = \mathcal{O}(N)$ is the number of quadrature points in each dimension; $M_{\boldsymbol{\sigma}} \ll N^d$ is the number of quadrature points on the sphere S^{d-1}; and ϵ is the requested precision in the NUFFT algorithm. The proposed fast algorithm does not require extra storage other than that storing the computational target, e.g., the gain and loss terms.

	direct algorithm		fast algorithm
	storage	(online) operation	(online) operation
gain term	$\mathcal{O}(N^{3d})$	$\mathcal{O}(N^{3d})$	$\mathcal{O}(M_{\boldsymbol{\sigma}} M_{\boldsymbol{v}}^{2d} \lvert\log \epsilon\rvert + M_{\boldsymbol{\sigma}} M_{\boldsymbol{v}}^{d} N^d \log N)$
loss term	$\mathcal{O}(N^{3d})$	$\mathcal{O}(N^{3d})$	$\mathcal{O}(M_{\boldsymbol{\sigma}} M_{\boldsymbol{v}}^{2d})$

4.4 Numerical examples

In this section, we perform extensive numerical tests to demonstrate the accuracy and efficiency of the proposed Petrov-Galerkin spectral method in both 2D and 3D.

Recall that the main motivation of the current work is to obtain better accuracy by considering approximations in an unbounded domain. To illustrate this point, we will compare three methods to solve the Boltzmann equation:

(1) Fast Fourier-Galerkin spectral method proposed in [42]: this method can achieve a good accuracy-efficiency tradeoff among the current deterministic methods for the Boltzmann equation. However, it requires the truncation of the domain to $[-L, L]^d$, where L is often chosen empirically such that the solution is close to zero at the boundary.

(2) Fast Chebyshev-0 method: the method proposed in this chapter using the logarithmic mapping (4.10), where $r = 0$ and the scaling parameter S needs to be properly chosen.

(3) Fast Chebyshev-1 method: the method proposed in this chapter using the algebraic mapping (4.11), where $r = 1$ and the scaling parameter S needs to be properly chosen.

In all three methods, the choice of truncation or mapping/scaling parameters has a great impact on numerical accuracy. In the following tests, we first determine L in the Fourier spectral method. Then for the two Chebyshev methods, we propose an adaptive strategy to determine the scaling parameter S: for example, in 1D, the Chebyshev-Gauss-Lobatto quadrature points on the interval $[-1, 1]$ are given by

$$-1 = \xi_1 < \xi_2 < \ldots < \xi_N = 1. \tag{4.68}$$

We choose S such that the two quadrature points ξ_2 and ξ_{N-1} are mapped to the boundary of $[-L, L]$, i.e.,

$$v(\xi_1) = -\infty, \quad v(\xi_2) = -L, \quad v(\xi_{N-1}) = L, \quad v(\xi_N) = -\infty. \tag{4.69}$$

Note that this S is adaptive in the sense that different N will correspond to different S.

4.4.1 2D examples

2D BKW solution

We consider first the 2D BKW solution. This is one of the few known analytical solutions to the Boltzmann equation and a perfect example to verify the accuracy of a numerical method.

When $d = 2$ and the collision kernel $\mathcal{B} \equiv 1/(2\pi)$, the following is a solution to the initial value problem (4.31):

$$f_{\mathrm{BKW}}(t, \boldsymbol{v}) = \frac{1}{2\pi K^2} \exp\left(-\frac{\boldsymbol{v}^2}{2K}\right)\left(2K - 1 + \frac{1-K}{2K}\boldsymbol{v}^2\right), \tag{4.70}$$

where $K = 1 - \exp(-t/8)/2$. By taking the time derivative of $f_{\mathrm{BKW}}(t, \boldsymbol{v})$, we can obtain the exact collision operator as

$$Q_{\mathrm{BKW}}(f) = \left\{\left(-\frac{2}{K} + \frac{\boldsymbol{v}^2}{2K^2}\right) f_{\mathrm{BKW}} + \frac{1}{2\pi K^2} \exp\left(-\frac{\boldsymbol{v}^2}{2K}\right)\left(2 - \frac{1}{2K^2}\boldsymbol{v}^2\right)\right\} K', \tag{4.71}$$

where $K' = \exp(-t/8)/16$. This way we can apply the numerical method to compute $Q_{\mathrm{BKW}}(f)$ directly and check its error without worrying about the time discretization.

In the fast Fourier spectral method, we take $N_\rho = N$ quadrature points in the radial direction and $N_\theta = 8$ quadrature points on the unit circle (see [42] for more details). In the fast Chebyshev methods, we take $M_{\boldsymbol{v}} = N + 2$ quadrature points for each dimension of $(\boldsymbol{v}, \boldsymbol{v}_*)$ and $M_{\boldsymbol{\sigma}} = N$ quadrature points on the unit circle. The precision in NUFFT is selected as $\epsilon = 1\mathrm{e}-14$. The numerical error of $Q_{\mathrm{BKW}}(f)$ is estimated on a 200×200 uniform grid in the rectangular domain $[-6.3, 6.3]^2$ at time $t = 2$.

Test 01: In this test, we examine thoroughly the numerical errors concerning different truncation parameters L in the Fourier method, and scaling parameters S in the Chebyshev methods. The L^2 errors of $Q_{\mathrm{BKW}}(f)$ for three methods are presented in Figure 4.1. It is obvious that the accuracy is not good when L and S are too small or too large. When L and S are chosen appropriately, the accuracy is close to the machine precision. In Table 4.2, we record the best accuracy for a given N of each method. One can see that the fast Chebyshev-0 method can always achieve the best accuracy.

Table 4.2. (2D BKW: Test 01) The L^2 error of $Q_{\mathrm{BKW}}(f)$ at time $t = 2$. The best accuracy for a given N of each method.

	fast Fourier	fast Chebyshev-0	fast Chebyshev-1
$N = 8$	8.9223e-03	2.2289e-03	4.1388e-03
$N = 16$	1.0989e-04	2.2033e-05	2.9713e-04
$N = 24$	3.2104e-06	1.9843e-07	1.5004e-05
$N = 32$	9.4720e-08	3.0082e-09	5.3946e-07
$N = 40$	1.9836e-09	2.5434e-11	1.6120e-08
$N = 48$	6.1797e-11	1.6255e-13	6.0320e-10
$N = 56$	1.4315e-12	2.1482e-14	2.5213e-11

Test 02: In this test, we fix the computational domain and examine the numerical errors concerning different N. In the Fourier method, we test $L = 8.83$ and $L = 13.24$. In the Chebyshev methods, we use the same L to select the scaling parameter S accordingly. The L^∞ errors of $Q_{\mathrm{BKW}}(f)$ for three methods are presented in Figure 4.2. Among these three methods, the fast Chebyshev-0 method can achieve the best accuracy when N is small. The fast Chebyshev-1 method doesn't provide a good approximation. This is because the

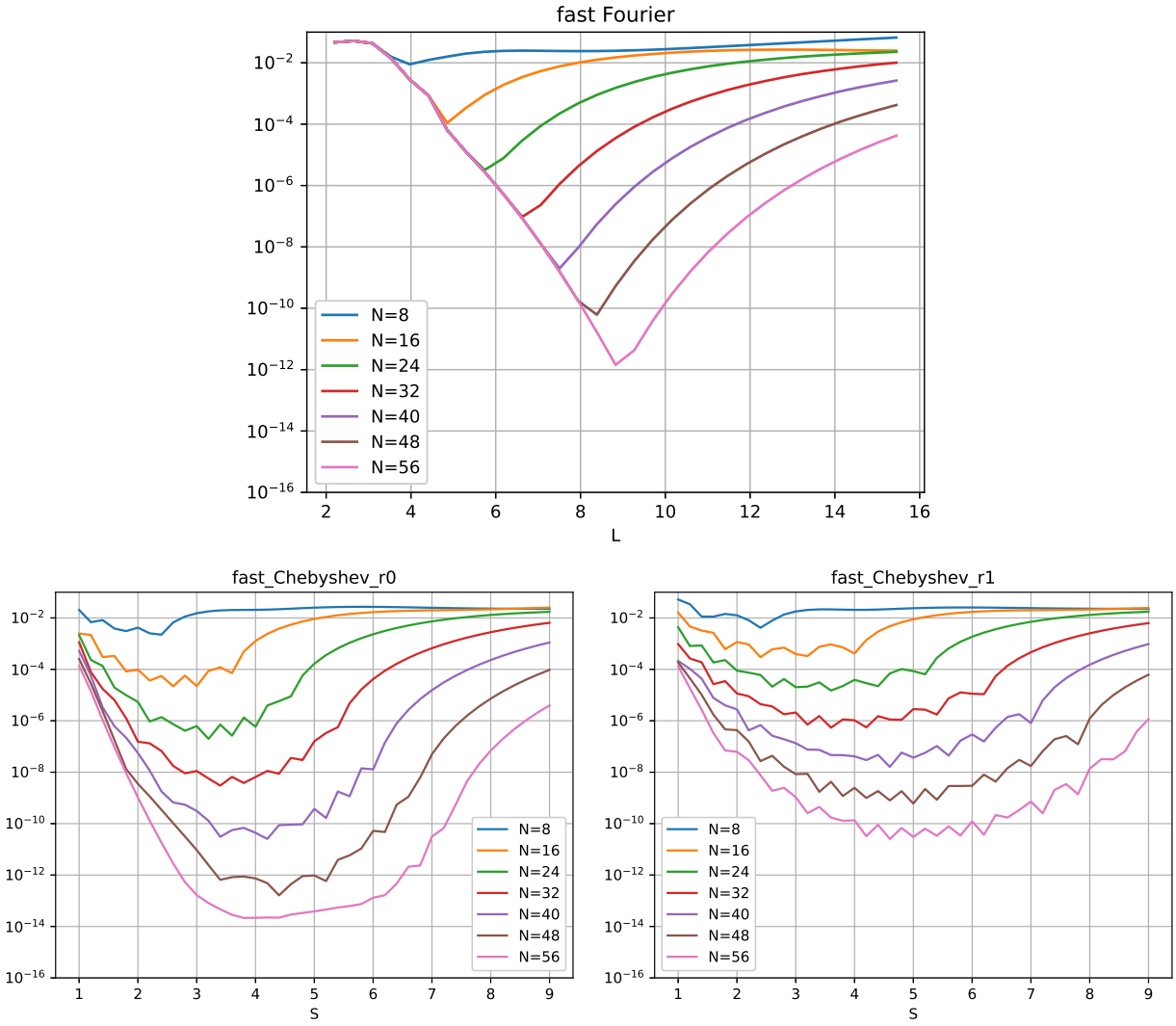

Figure 4.1. (2D BKW: Test 01) The L^2 error of $Q_{\mathrm{BKW}}(f)$ at time $t = 2$. Top: fast Fourier method. Bottom: fast Chebyshev methods.

quadrature points in Chebyshev-1 method are much more clustered near the origin compared to Chebyshev-0 method and apparently points located far away from the origin play an important role in the unbounded domain problem.

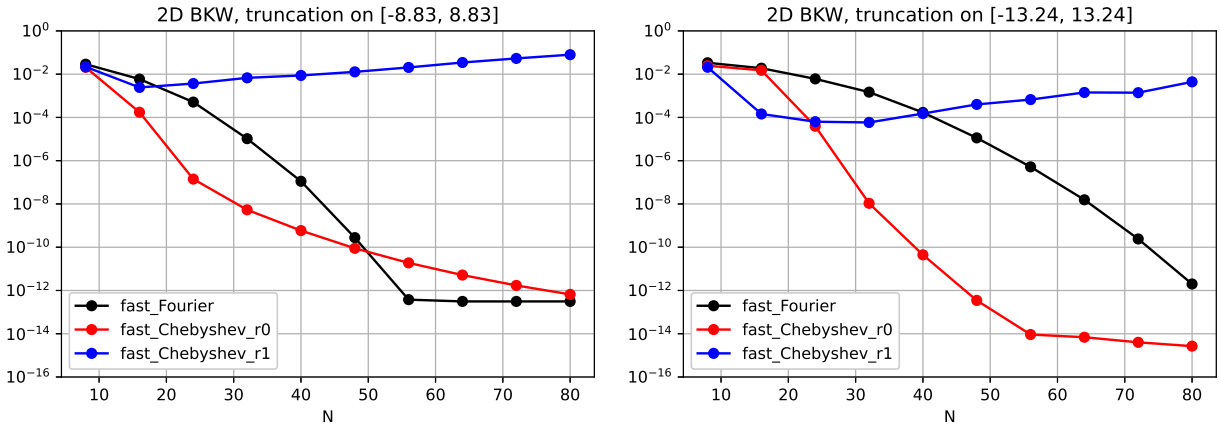

Figure 4.2. (2D BKW: Test 02) The L^∞ error of $Q_{\mathrm{BKW}}(f)$ at time $t = 2$. Left: $L = 8.83$; Right: $L = 13.24$.

Test 03: In this test, we examine the numerical errors of the Chebyshev methods with a fixed scaling parameter: $S = 4$ in the fast Chebyshev-0 method; $S = 5$ in the fast Chebyshev-1 method. These two values are selected based on the results in Figure 4.1. The L^∞ errors of $Q_{\mathrm{BKW}}(f)$ for both methods are presented in Figure 4.3. As a comparison, results of the fast Fourier method are also plotted. Again for small N, the fast Chebyshev-0 method provides the best accuracy among the three methods.

Test 04: In this test, we report the computational time of the direct (Chebyshev) algorithm and the fast (Chebyshev) algorithm. The computations were done on Intel(R) Core(TM) i7-6700 CPU in a single thread. Table 4.3 shows the running time of the direct and fast algorithms concerning different N. Note that the direct algorithm is left out when $N \geq 32$ due to the memory constraint.

Computing the moments

We next consider the time evolution problem and check the accuracy for moments approximation. Since the fast Chebyshev-0 method performs generally better than the fast

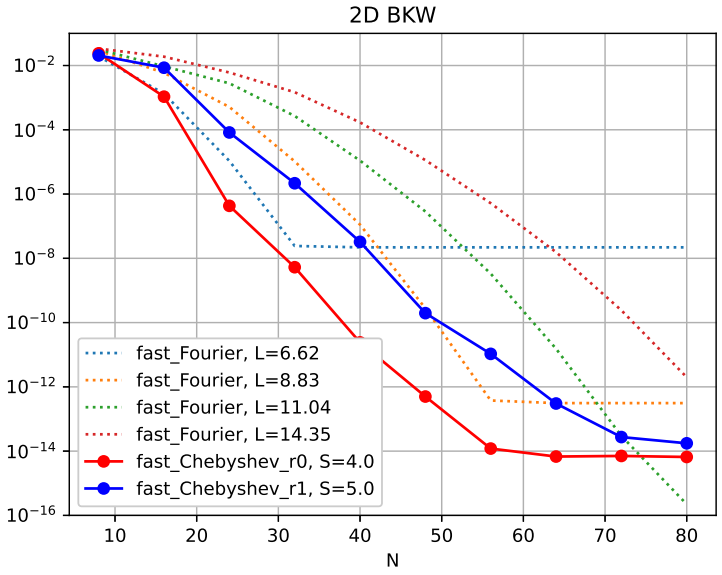

Figure 4.3. (2D BKW: Test 03) The L^∞ error of $Q_{\mathrm{BKW}}(f)$ at time $t = 2$.

	direct algorithm		fast algorithm
	online (sec)	precomputation (sec)	online (sec)
$N = 8$	0.0047	1.5956	0.194955
$N = 16$	0.1207	62.7991	2.036821
$N = 32$	-	-	24.779492
$N = 64$	-	-	4.937722e+02
$N = 128$	-	-	1.331576e+04

Table 4.3. (2D BKW: Test 04) Running time in second for a single evaluation of the gain term.

Chebyshev-1 method, we will restrict to the former in the following tests. The comparison with the fast Fourier method will still be considered.

In (4.31), we choose the collision kernel $\mathcal{B} \equiv 1/(2\pi)$ and the initial condition as

$$f^0(\boldsymbol{v}) = \frac{\rho_1}{2\pi T_1} \exp\left(-\frac{(\boldsymbol{v}-V_1)^2}{2T_1}\right) + \frac{\rho_2}{2\pi T_2} \exp\left(-\frac{(\boldsymbol{v}-V_2)^2}{2T_2}\right), \tag{4.72}$$

where $\rho_1 = \rho_2 = 1/2$, $T_1 = T_2 = 1$ and $V_1 = (x_1, y_1) = (-1, 2)$, $V_2 = (x_2, y_2) = (3, -3)$.

Then for the momentum flow and energy flow defined as

$$P_{\mathrm{ij}} = \int_{\mathbf{R}^2} f \boldsymbol{v}_{\mathrm{i}} \boldsymbol{v}_{\mathrm{j}} \,\mathrm{d}\boldsymbol{v}, \quad (\mathrm{i},\mathrm{j} = 1,2), \quad q_{\mathrm{i}} = \int_{\mathbf{R}^2} f \boldsymbol{v}_{\mathrm{i}} |\boldsymbol{v}|^2 \,\mathrm{d}\boldsymbol{v}, \quad (\mathrm{i} = 1,2), \tag{4.73}$$

we have their exact formulas

$$P_{11} = -\frac{9}{8}\mathrm{e}^{-t/2} + \frac{57}{8}, \quad P_{12} = P_{21} = -5\mathrm{e}^{-t/2} - \frac{1}{2}, \quad P_{22} = \frac{9}{8}\mathrm{e}^{-t/2} + \frac{51}{8}, \tag{4.74}$$

and

$$q_1 = \frac{1}{4}\left(11\mathrm{e}^{-t/2} + 103\right), \quad q_2 = -\frac{1}{8}\left(89\mathrm{e}^{-t/2} + 103\right). \tag{4.75}$$

In the fast Fourier method, we take $N_\rho = N$ quadrature points in the radial direction and $N_\theta = N$ quadrature points on the unit circle. The truncation domain $[-L, L]^2$ is selected as $L = 14.35$. In the fast Chebyshev-0 method, we take $M_{\boldsymbol{v}} = N + 2$ quadrature points for each dimension of $(\boldsymbol{v}, \boldsymbol{v}_*)$ and $M_{\boldsymbol{\sigma}} = N$ quadrature points on the unit circle. The precision in NUFFT is selected as $\epsilon = 1\mathrm{e}-14$. The scaling parameter S is adaptively chosen based on L. For both methods, we use the 4th-order Runge-Kutta method with $\Delta t = 0.02$ for time integration.

The absolute errors of the moments are presented in Figure 4.4–4.8. The fast Chebyshev-0 method clearly provides a better approximation in comparison to the Fourier method for fixed N.

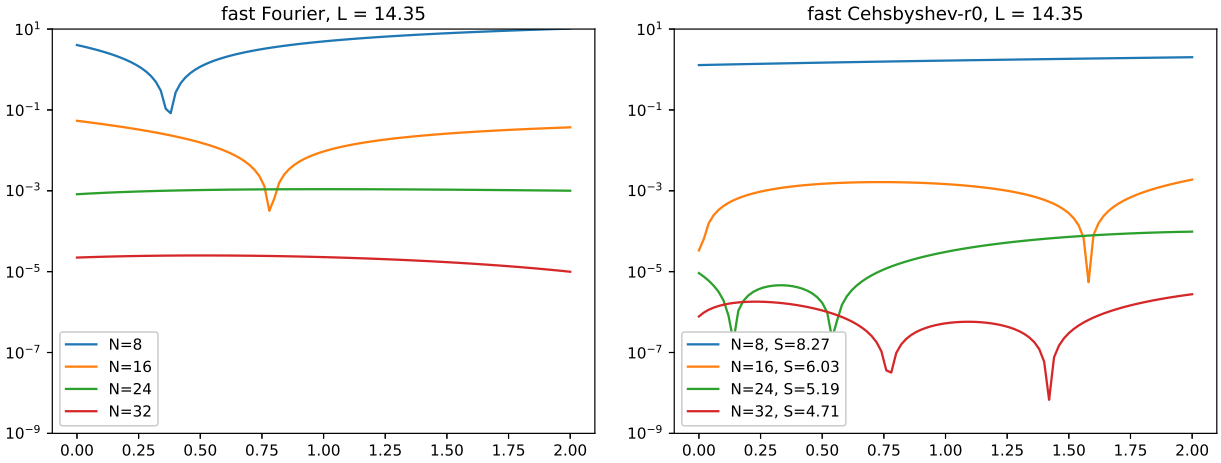

Figure 4.4. (2D moments) The time evolution for the absolute error of the momentum flow P_{11}. Left: the fast Fourier method. Right: the fast Chebyshev-0 method.

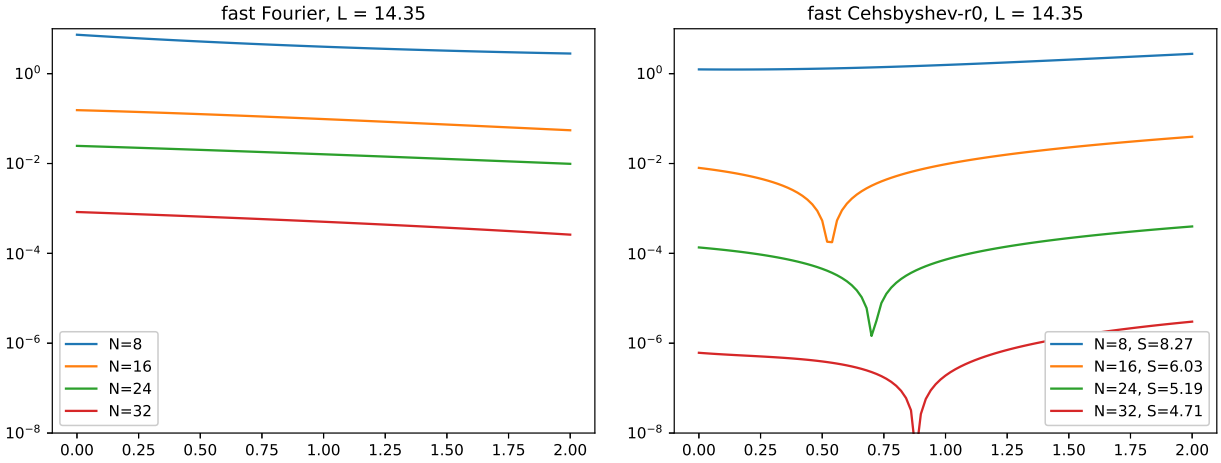

Figure 4.5. (2D moments) The time evolution for the absolute error of the momentum flow P_{12}. Left: the fast Fourier method. Right: the fast Chebyshev-0 method.

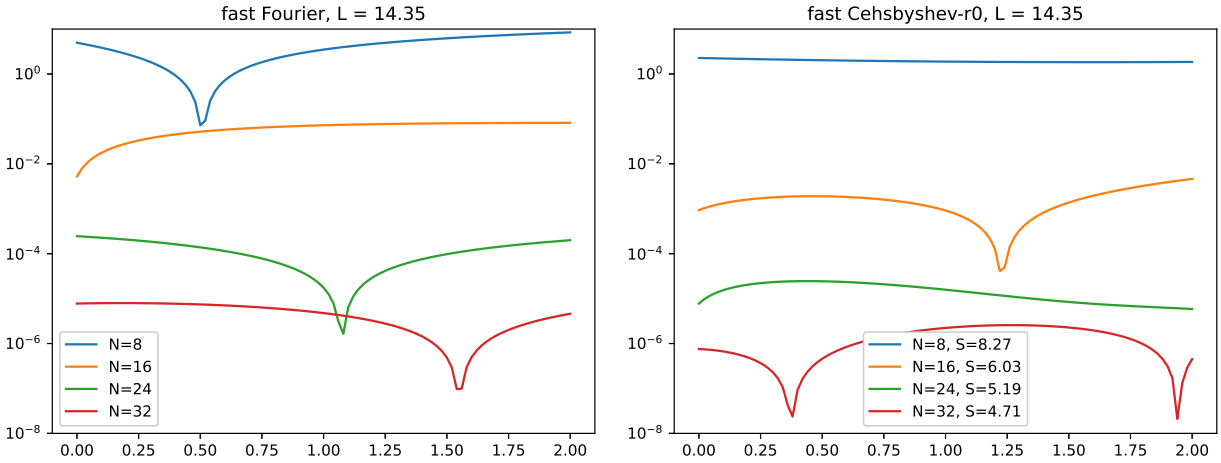

Figure 4.6. (2D moments) The time evolution for the absolute error of the momentum flow P_{22}. Left: the fast Fourier method. Right: the fast Chebyshev-0 method.

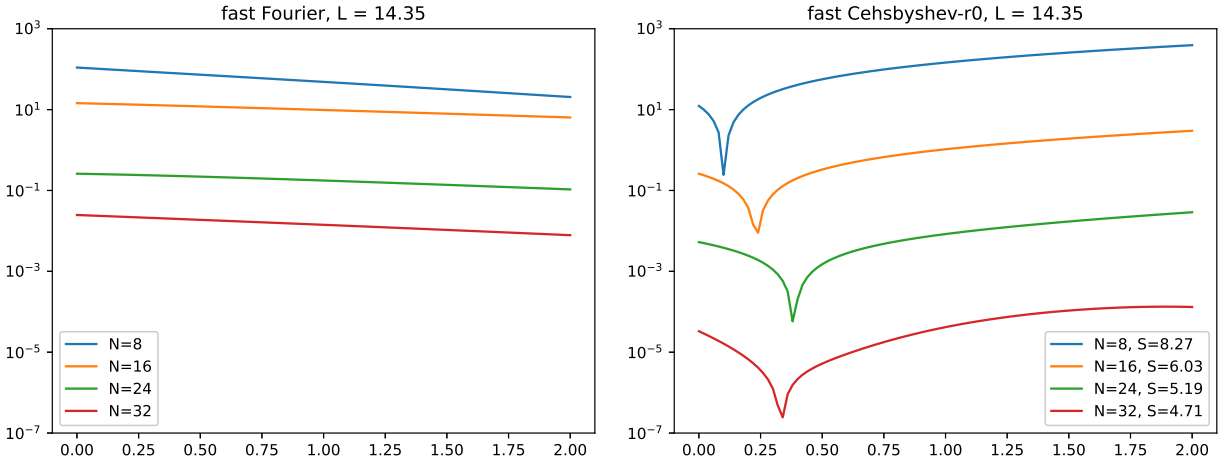

Figure 4.7. (2D moments) The time evolution for the absolute error of the momentum flow q_1. Left: the fast Fourier method. Right: the fast Chebyshev-0 method.

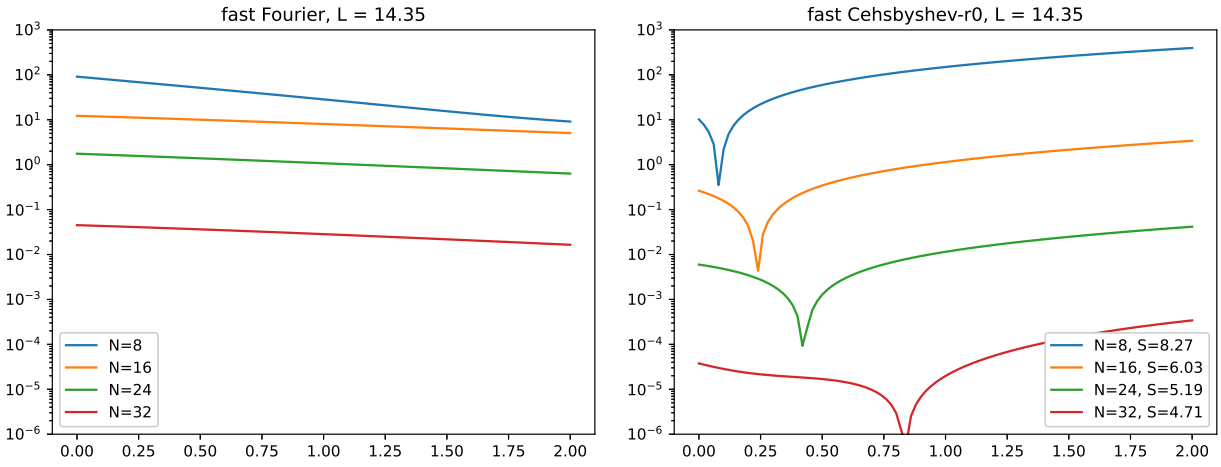

Figure 4.8. (2D moments) The time evolution for the absolute error of the momentum flow q_2. Left: the fast Fourier method. Right: the fast Chebyshev-0 method.

4.4.2 3D BKW solution

We finally consider the 3D BKW solution. When $d = 3$ and the collision kernel $\mathcal{B} \equiv 1/(4\pi)$, the following is a solution to the initial value problem (4.31):

$$f_{\mathrm{BKW}}(t, \boldsymbol{v}) = \frac{1}{2(2\pi K)^{3/2}} \exp\left(-\frac{\boldsymbol{v}^2}{2K}\right)\left(\frac{5K-3}{K} + \frac{1-K}{K^2}\boldsymbol{v}^2\right), \tag{4.76}$$

where $K = 1 - \exp(-t/6)$. As in 2D, we can obtain the exact collision operator as

$$Q_{\mathrm{BKW}}(f) = \left\{\left(-\frac{3}{2K} + \frac{\boldsymbol{v}^2}{2K^2}\right) f_{\mathrm{BKW}} + \frac{1}{2(2\pi K)^{3/2}} \exp\left(-\frac{\boldsymbol{v}^2}{2K}\right)\left(\frac{3}{K^2} + \frac{K-2}{K^3}\boldsymbol{v}^2\right)\right\} K', \tag{4.77}$$

with $K' = \exp(-t/6)/6$.

Here we again compare the fast Fourier spectral method with the fast Chebyshev-0 method. In the former, we take domain $L = 6.62$, $N_\rho = N$ quadrature points in the radial direction and $M_{\boldsymbol{\sigma}} = 38$ Lebedev quadrature points on the unit sphere. In the latter, we choose S adaptively based on L, $M_{\boldsymbol{v}} = N + 2$ quadrature points for each dimension of $(\boldsymbol{v}, \boldsymbol{v}_*)$ and $M_{\boldsymbol{\sigma}}$ Lebedev quadrature points on the unit sphere. The precision in NUFFT is selected as $\epsilon = 1\mathrm{e}-14$. The L^∞ error of $Q_{\mathrm{BKW}}(f)$ is estimated on a $30 \times 30 \times 30$ uniform grid in the rectangular domain $[-6.3, 6.3]^3$ at time $t = 6.5$.

Table 4.4. (3D BKW) The L^∞ error of $Q_{\mathrm{BKW}}(f)$ at time $t = 6.5$.

	fast Fourier ($M_{\boldsymbol{\sigma}} = 38$)	fast Chebyshev-0
$N = 12$	2.36e-03	1.61e-02 ($M_{\boldsymbol{\sigma}} = 14$)
$N = 16$	4.37e-04	2.72e-03 ($M_{\boldsymbol{\sigma}} = 38$)
$N = 20$	3.62e-05	3.08e-06 ($M_{\boldsymbol{\sigma}} = 86$)
$N = 24$	3.61e-06	3.10e-08 ($M_{\boldsymbol{\sigma}} = 146$)
$N = 28$	1.64e-07	1.58e-08 ($M_{\boldsymbol{\sigma}} = 170$)
$N = 32$	3.82e-08	7.16e-10 ($M_{\boldsymbol{\sigma}} = 230$)

The results are reported in Table 4.4. Unlike the Fourier method for which $M_{\boldsymbol{\sigma}} = 38$ is enough (we have tested that larger values of $M_{\boldsymbol{\sigma}}$ would not further increase the accuracy), we observe that more quadrature points on the sphere are needed to get the best accuracy in the Chebyshev method. As soon as $N \geq 20$, the Chebyshev method can always obtain better accuracy than the Fourier method.

4.5 Conclusion

We introduced a Petrov-Galerkin spectral method for the spatially homogeneous Boltzmann equation in multi-dimensions. The mapped Chebyshev functions in $\mathbb{R}^d$ were carefully chosen to serve as the trial functions and test functions in the approximation. In the case of the algebraic mapping, we established a consistency result for approximation of the collision operator as well as the conservation property for the moments. Thanks to the close relation between the Chebyshev functions and the Fourier cosine series, we proposed a fast algorithm to alleviate the memory constraint in the precomputation and accelerate the online computation in the direct implementation. Through a series of numerical examples in 2D and 3D, we demonstrated that the proposed method can provide better accuracy (at least one or two digits for small N) in comparison to the popular Fourier spectral method.

REFERENCES

[1] C. Cercignani, *Rarefied Gas Dynamics: From Basic Concepts to Actual Calculations*. Cambridge University Press, Cambridge, 2000.

[2] C. K. Birdsall and A. B. Langdon, *Plama Physics via Computer Simulation*. CRC Press, 2018.

[3] S. Chandrasekhar, *Radiative Transfer*. Dover Publications, 1960.

[4] V. Giovangigli, *Multicomponent Flow Modeling*. Springer Science & Business Media, 1999.

[5] G. Naldi, L. Pareschi, and G. Toscani, Eds., *Mathematical Modeling of Collective Behavior in Socio-Economic and Life Sciences*. Birkhäuser Basel, 2010.

[6] C. Cercignani, *The Boltzmann Equation and Its Applications*. Springer-Verlag, New York, 1988.

[7] C. Villani, "A review of mathematical topics in collisional kinetic theory," in *Handbook of Mathematical Fluid Mechanics*, S. Friedlander and D. Serre, Eds., vol. I, North-Holland, 2002, pp. 71–305.

[8] P. L. Bhatnagar, E. P. Gross, and M. Krook, "A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems," *Physical review*, vol. 94, no. 3, p. 511, 1954.

[9] L. D. Landau, "The transport equation in the case of Coulomb interactions," *Collected papers of LD Landau*, pp. 163–170, 1936.

[10] H. Risken, "Fokker-Planck equation," in *The Fokker-Planck Equation*, Springer, 1996, pp. 63–95.

[11] D. Chen and R. Eisenberg, "Poisson-Nernst-Planck (PNP) theory of open ionic channels," *Biophys. J.*, vol. 64, A22, 1993.

[12] R. Eisenberg, "Ion channels in biological membranes: Electrostatic analysis of a natural nanotube," *Contemp. Phys.*, vol. 39, p. 447, 1998.

[13] P. A. Markowich, C. Ringhofer, and C. Schmeiser, *Semiconductor Equations*. New York: Springer Verlag Wien, 1990.

[14] G.-W. Wei, Q. Zheng, Z. Chen, and K. Xia, "Variational multiscale models for charge transport," *SIAM Rev.*, vol. 54, pp. 699–754, 2012.

[15] A. Krzywicki and T. Nadzieja, "A nonstationary problem in the theory of electrolytes," *Quart. Appl. Math.*, vol. 50, pp. 105–107, 1992.

[16] P. Biler, "Existence and asymptotics of solutions for a parabolic-elliptic system with nonlinear no-flux boundary conditions," *Nonlinear Anal.*, vol. 19, pp. 1121–1136, 1992.

[17] P. Biler, W. Hebisch, and T. Nadzieja, "The Debye system: Existence and large time behavior of solutions," *Nonlinear Anal.*, vol. 23, pp. 1189–1209, 1994.

[18] A. Arnold, P. Markowich, and G. Toscani, "On large time asymptotics for drift-diffusion-Poisson systems," *Transport Theory Statist. Phys.*, vol. 29, pp. 571–581, 2000.

[19] P. Biler and J. Dolbeault, "Long time behavior of solutions to Nernst-Planck and Debye-Hückel drift-diffusion systems," *Ann. Henri Poincaré*, vol. 1, pp. 461–472, 2000.

[20] H. Liu and Z. Wang, "A free energy satisfying finite difference method for Poisson-Nernst-Planck equations," *J. Comput. Phys.*, vol. 268, pp. 363–376, 2014.

[21] H. Liu and Z. Wang, "A free energy satisfying discontinuous Galerkin method for one-dimensional Poisson-Nernst-Planck systems," *J. Comput. Phys.*, vol. 328, pp. 413–437, 2017.

[22] A. Flavell, M. Machen, B. Eisenberg, J. Kabre, C. Liu, and X. Li, "A conservative finite difference scheme for Poisson-Nernst-Planck equations," *J. Comput. Electron.*, vol. 13, pp. 235–249, 2014.

[23] A. Flavell, J. Kabre, and X. Li, "An energy-preserving discretization for the Poisson-Nernst-Planck equations," *J. Comput. Electron.*, vol. 16, pp. 431–441, 2017.

[24] M. Metti, J. Xu, and C. Liu, "Energetically stable discretizations for charge transport and electrokinetic models," *J. Comput. Phys.*, vol. 306, pp. 1–18, 2016.

[25] R. Bailo, J. Carrillo, and J. Hu, "Fully discrete positivity-preserving and energy-decaying schemes for aggregation-diffusion equations with a gradient flow structure," *Communications in Mathematical Sciences*, vol. 18, pp. 1259–1303, 2020.

[26] K. Nanbu, "Direct simulation scheme derived from the Boltzmann equation. I. Monocomponent gases," *J. Phys. Soc. Jpn.*, vol. 49, pp. 2042–2049, 1980.

[27] G. A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Clarendon Press, Oxford, 1994.

[28] G. Dimarco and L. Pareschi, "Numerical methods for kinetic equations," *Acta Numer.*, vol. 23, pp. 369–520, 2014.

[29] L. Pareschi and G. Russo, "Numerical solution of the Boltzmann equation I: Spectrally accurate approximation of the collision operator," *SIAM J. Numer. Anal.*, vol. 37, pp. 1217–1245, 2000.

[30] C. Mouhot and L. Pareschi, "Fast algorithms for computing the Boltzmann collision operator," *Math. Comp.*, vol. 75, pp. 1833–1852, 2006.

[31] I. Gamba and S. Tharkabhushanam, "Spectral-Lagrangian methods for collisional models of non-equilibrium statistical states," *J. Comput. Phys.*, vol. 228, pp. 2012–2036, 2009.

[32] I. Gamba, J. Haack, C. Hauck, and J. Hu, "A fast spectral method for the Boltzmann collision operator with general collision kernels," *SIAM J. Sci. Comput.*, vol. 39, B658–B674, 2017.

[33] L. Pareschi and G. Russo, "On the stability of spectral methods for the homogeneous Boltzmann equation," *Transport Theory Statist. Phys.*, vol. 29, pp. 431–447, 2000.

[34] F. Filbet and C. Mouhot, "Analysis of spectral methods for the homogeneous Boltzmann equation," *Trans. Amer. Math. Soc.*, vol. 363, pp. 1947–1980, 2011.

[35] R. Alonso, I. Gamba, and S. Tharkabhushanam, "Convergence and error estimates for the Lagrangian-based conservative spectral method for Boltzmann equations," *SIAM J. Numer. Anal.*, vol. 56, pp. 3534–3579, 2018.

[36] J. Hu, K. Qi, and T. Yang, "A new stability and convergence proof of the Fourier-Galerkin spectral method for the spatially homogeneous Boltzmann equation," *SIAM J. Numer. Anal.*, vol. 59, no. 2, pp. 613–633, 2021.

[37] E. Fonn, P. Grohs, and R. Hiptmair, "Polar spectral scheme for the spatially homogeneous Boltzmann equation," *Research Report*, vol. 2014, 2014.

[38] I. M. Gamba and S. Rjasanow, "Galerkin–Petrov approach for the Boltzmann equation," *Journal of Computational Physics*, vol. 366, pp. 341–365, 2018.

[39] G. Kitzler and J. Schöberl, "A polynomial spectral method for the spatially homogeneous Boltzmann equation," *SIAM Journal on Scientific Computing*, vol. 41, no. 1, B27–B49, 2019.

[40] Z. Hu and Z. Cai, "Burnett spectral method for high-speed rarefied gas flows," *SIAM J. Sci. Comput.*, vol. 42, B1193–B1226, 2020.

[41] Z. Hu, Z. Cai, and Y. Wang, "Numerical simulation of microflows using Hermite spectral methods," *SIAM J. Sci. Comput.*, vol. 42, B105–B134, 2020.

[42] I. M. Gamba, J. R. Haack, C. D. Hauck, and J. Hu, "A fast spectral method for the Boltzmann collision operator with general collision kernels," *SIAM Journal on Scientific Computing*, vol. 39, no. 4, B658–B674, 2017.

[43] J. Hu, J. Shen, and Y. Wang, "A Petrov-Galerkin spectral method for the inelastic Boltzmann equation using mapped Chebyshev functions," *Kinetic & Related Models*, vol. 13, no. 4, 2020.

[44] D. L. Scharfetter and H. K. Gummel, "Large signal analysis of a silicon read diode," *IEEE Transactions on Electron Devices*, vol. 16, pp. 64–67, 1969.

[45] A. Bousquet, X. Hu, M. Metti, and J. Xu, "Newton solvers for drift-diffusion and electrokinetic equations," *SIAM J. Sci. Comput.*, vol. 40, B982–B1006, 2018.

[46] M. Schmuck, "Analysis of the Navier-Stokes-Nernst-Planck-Poisson system," *Math. Models Methods Appl. Sci.*, vol. 19, pp. 993–1015, 2009.

[47] S. Furini, F. Zerbetto, and S. Cavalcanti, "Application of the Poisson-Nernst-Planck theory with space-dependent diffusion coefficients to KcsA," *Biophysical journal*, vol. 91, no. 9, pp. 3162–3169, 2006.

[48] L. Pareschi and G. Russo, "Numerical solution of the Boltzmann equation I: Spectrally accurate approximation of the collision operator," *SIAM journal on numerical analysis*, vol. 37, no. 4, pp. 1217–1245, 2000.

[49] J. Shen, T. Tang, and L.-L. Wang, *Spectral methods: algorithms, analysis and applications*. Springer Science & Business Media, 2011, vol. 41.

[50] J. Shen, L.-L. Wang, and H. Yu, "Approximations by orthonormal mapped Chebyshev functions for higher-dimensional problems in unbounded domains," *Journal of Computational and Applied Mathematics*, vol. 265, pp. 264–275, 2014.

[51] J. Shen and L.-L. Wang, "Sparse spectral approximations of high-dimensional problems based on hyperbolic cross," *SIAM Journal on Numerical Analysis*, vol. 48, no. 3, pp. 1087–1109, 2010.

[52] C. Mouhot and C. Villani, "Regularity theory for the spatially homogeneous Boltzmann equation with cut-off," *Arch. Rational Mech. Anal.*, vol. 173, pp. 169–212, 2004.

[53] G. A. Bird and J. Brady, *Molecular gas dynamics and the direct simulation of gas flows*. Clarendon press Oxford, 1994, vol. 42.

[54] V. Lebedev, "Quadratures on a sphere," *USSR Computational Mathematics and Mathematical Physics*, vol. 16, no. 2, pp. 10–24, 1976.

[55] A. H. Barnett, J. Magland, and L. af Klinteberg, "A parallel nonuniform fast Fourier transform library based on an "exponential of semicircle" kernel," *SIAM Journal on Scientific Computing*, vol. 41, no. 5, pp. C479–C504, 2019.