COUPLING A BOLTZMANN PLASMA AND BCA SURFACE FOR THE KINETIC TREATMENT OF PLASMA-MATERIAL INTERACTIONS

BY:

SHANE KENILEY

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Nuclear, Plasma, and Radiological Engineering
in the Graduate College of the
University of Illinois in Urbana-Champaign, 2017

Urbana, Illinois

Master's Committee:

Assistant Professor Davide Curreli, Adviser
Professor Rizwan Uddin

Abstract

A near-wall plasma and material surface form a volatile region involving surface erosion, impurity ionization, and redeposition, creating a far-from-equilibrium system of mutually interacting plasma and impurity species. As impurity recycling is expected to play a major role in the long-term performance of plasma-facing components in magnetic fusion devices, modeling of the plasma-surface interface is required to predict the behavior of both the material surface and the near-wall plasma. In this work, a method of simulating plasma-material interactions by dynamically coupling a continuum Boltzmann plasma model to a Monte-Carlo surface model is presented. The model is based on a multi-species Boltzmann solver for the plasma using finite difference methods. Von Neumann stability analysis of the Runge-Kutta time discretization with upwind-biased numerical schemes are detailed up to fourth-order accuracy, and the errors associated with each scheme are quantified. A modification to the classical binary-collision approximation code TRIDYN is utilized to model the surface, which was treated as a boundary condition to the plasma model.

The Boltzmann solver calculates the ion energy-angle distribution and density of ions striking the surface that are needed as input to the BCA code, and density estimation is used to reconstruct a velocity distribution to be passed back to the Boltzmann solver. Both plasma ions and impurities are treated as Boltzmann kinetic species, allowing high resolution even at very disparate densities, particle fluxes, drift velocities, and energy fluxes. The plasma model is shown to be capable of resolving features of Landau damping with matching theoretical and calculated damping rates of 0.1534, and the two-stream instability is shown to have an energy peak at 18 $t\omega_p$. Convergence of the plasma sheath problem is established utilizing the fourth-order upwind finite difference method. Numerical density estimation techniques are applied to construct velocity distributions from discrete data samples provided by TRIDYN, and a sputtered particle sample size of 1000 is shown to constrain the mean integrated squared error of the density-estimated velocity distribution to

$O(10^{-1})$. As a proof-of-concept of the coupling method, an example calculation of a helium plasma facing a beryllium wall is reported in both unmagnetized and magnetized conditions, recording the evolution of the phase spaces of ions, neutrals, and material impurities in the near-wall region at nominal ITER conditions.

To my parents, for everything.

Acknowledgements

The list of people I should thank for helping me get this far would be of comparable length to this thesis, so for the sake of brevity I will only name a few.

I would first like to thank my parents, Martin and Nancy, for encouraging and supporting every interest I have ever had to the best of their abilities. Their patience, support, and love is the only reason I am here today. I also need to thank my brother, Noah, for being my confidant, and for pushing me forward when I need encouragement. I am lucky to have such a supportive and close-knit extended family; they are far too numerous to include, but I am indebted to and thankful for all of them.

My advisor, Professor Davide Curreli, is an amazing teacher and mentor. He has consistently pushed me to improve every aspect of my work, and his continued belief and trust does not go unnoticed. I appreciate the opportunity and guidance he has given me more than I can express in words. I would also like to thank Professor Rizwan Uddin for his input as a part of this Master's committee. Special thanks goes to my colleague Rinat, for all of his help on the work that went into this thesis.

It would be remiss of me to ignore my friends: Jim and Kelley, for blatantly ignoring my asocial tendencies and for being better friends than I ever thought I deserved; Gianluca, for being my (second) brother; Rachael and Ryland, for being a part of my life; and to all graduate students that I share my time with, for keeping me sane (and for the steady supply of coffee).

This thesis would not have been possible without funding through the Scientific Discovery through Advanced Computing (SciDAC) project on Plasma-Surface Interactions, under Award No. DE-SC00-08875.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Plasma-Material Interactions in Magnetic Fusion Devices

A fusion device capable of generating more power than it consumes has numerous potential benefits over current power sources: it avoids the pollution and ecological problems of fossil fuels along with the radiation and public perception issues with nuclear fission devices. Indeed, the concept is of such importance that creating a commercially viable fusion reactor has been listed as one of the grand engineering challenges of the 21st century. According to the EU roadmap for fusion energy, heat exhaust through the interaction of the fusion plasma and the surrounding materials is listed as the leading challenge toward obtaining a fusion reactor. A tokamak - which is the most widely researched candidate for a commercial fusion reactor - confines a plasma that is hotter than the surface of the sun, causing enormous temperature and density gradients that will force a portion of the plasma to interact with the wall. Plasma-facing components (PFCs) of a magnetically confined fusion device are required to withstand such intense heat loads over a time scale of months to years without requiring extensive maintenance. Structural integrity is not the only concern, however - it has been shown that the response of materials under ion bombardment can influence the stability and structure of the plasma itself. Plasma impacting a surface will cause sputtering, and sputtered particles that enter the plasma bulk are capable of cooling the plasma. Furthermore, the plasma-material interaction (PMI) process can affect hydrogen retention and wall erosion, affecting both plasma stability and PFC lifetime [1, 2].

Three coupled regions are considered in the study of PMI in magnetic fusion devices: (1) the plasma edge and scrape-off layer (SOL) region, (2) the near-surface region including the plasma sheath and the first surface layers, and (3) the material bulk [3], which together form a dynamically evolving environment involving a multitude of coupled phenomena between the plasma and the

material wall. The plasma sheath which forms near a surface acts as the boundary between the bulk plasma and the material and determines the energy-angle distribution of particles impacting the wall, which is directly related to the erosion of the material surface: sputtering, retention, and other surface modification processes depend on the energy, angle, flux, and species of particles impacting the surface. Impurity transport further influences the near-wall region through collisions with plasma species. For many divertor conditions approximately $\sim$5% of sputtered material enters the plasma bulk, while the remaining $\sim$95% of the material that is sputtered from the wall will be ionized close to the surface and recycled, creating a new surface layer with thermo-mechanical properties that differ from the original ordered lattice [4]. The near-wall plasma thus forms a far-from-equilibrium system in which the surface is continuously eroded, redeposited, and reconstituted. Significant research effort has been made to find sheath conditions which optimize the longevity of a plasma-facing component (PFC) while being compatible with bulk plasma properties [5].

1.2 Plasma and Surface Modeling

Modeling plasma-material interactions is limited by the largely disparate time- and length-scales which, due to their strong coupling must be considered simultaneously. Relevant plasma processes occur over mm of length and μs in time, while surface interactions take place on the order of an Angstrom and ns time-scales. Accurately simulating PMI is thus not only a matter of implementing the necessary physics, but also developing techniques for integrating physical models that necessarily operate on different scales. The multiscale modeling approach addresses this issue by utilizing a separate model for each region: kinetic plasma simulations may be used to predict the ion energy-angle distribution (IEAD) striking the surface, while atomistic models treat surface modification (sputtering, reflection, defect formation).

Kinetic Plasma Simulation Methods

Kinetic descriptions of the near-wall plasma have been achieved previously with both particle-in-cells (PICs) and continuum Boltzmann methods. PIC codes use a statistical approach to describe the phase space by tracking a finite number of macroparticles, N_p. These methods are currently the most widely used kinetic plasma codes, and they have been developed for fusion applications

in magnetized conditions in the divertor region of tokamaks [6, 7, 8]. VPIC is a three-dimensional PIC code that has been used to model laser-plasma interaction for inertial confinement fusion [9]. Gyrokinetic PIC codes, which track the guiding centers of particles rather than the full orbit, have also been applied to fusion applications [10].

While PIC models are well-established and have been shown to be scalable, they are inherently subject to numerical noise that can obscure features of the near-wall plasma. The noise can be controlled by increasing N_p, but it will be diminished only by a factor of $\sqrt{N_p}$. In contrast, continuum kinetic solvers directly discretize the Boltzmann equation (or its collisionless form, the Vlasov equation) in phase space with a numerical mesh. They are more computationally expensive than equivalent PIC models and are subject to the Courant-Friedrichs-Lewy (CFL) stability condtion, which describes an upper limit on the number of gridpoints data can move in a single iteration, but they have two key benefits: they avoid the numerical noise problem of PIC methods, and they have the possibility of using high-order schemes, which can achieve greater accuracy on a coarse mesh than a low-order scheme on a fine mesh and can thus reduce computational cost.

Several Eulerian schemes have been studied with the Boltzmann-Poisson system. The discontinuous Galerkin scheme, which is a finite element approach, has shown good agreement with established fluid plasma models [11]. The semi-Lagrangian method has been widely used, but such methods include a problem of sharp oscillations around steep gradients that must be filtered [12]. Previous applications of the semi-Lagrangian method are also restricted to second-order in space due to their treatment of integrated masses and interpolation of densities [13, 14]. However, high-order semi-Lagrangian schemes have also been achieved by combining the method with a weighted essentially non-oscillatory (WENO) reconstruction [15]. Another Eulerian method is the finite difference approach, which has previously been combined with the semi-Lagrangian method with accurate results [16]. When treated with a WENO scheme, finite differences have been shown to achieve machine-precision accuracy of Landau damping [15].

Sputtering and Erosion Models

The erosion process itself is traditionally modeled using either Molecular Dynamics (MD) or binary collision approximation (BCA) methods. The advantage of BCA over MD is mainly computational, since the ion-material interaction is reduced from a multi-body problem to a Monte Carlo sequence of binary collisions, allowing faster computations and straightforward parallelization. Despite the lower physics content of BCA with respect to MD, such family of codes has been successful in describing sputtering processes. The ACAT code, which models atomic collisions in amorphous targets with a BCA-based method, has been used to calculate sputtering yield and angular distributions of sputtered particles, obtaining good agreement with experimental measurements [17]. TRIDYN is a BCA code that models ballistic effects on amorphous multi-component targets, including dynamic composition and surface thermal relaxation [18]. Density functional theory (DFT) models such as SIESTA [19] can calculate interaction forces between surface particles, which may be used in cases where the simple interaction potentials utilized in the MD and BCA codes are insufficient.

Impurity Transport and Surface Response

To treat the material reponse of the PMI system, a wide variety of methods have been developed to focus on each individual phenomena occuring at the surface interface. Codes that treat each phenomenon include the REDEP/WBC code package for erosion and redeposition,[20] UEDGE-BOUT codes for fluid edge parameters,[21] and HEIGHTS for multi-physics transient response,[22] which are all reviewed in addition to other notable codes by Brooks *et al.* [23]. Kirschner *et al.* developed the 3D ERO code, which is a Monte Carlo tracker of the sputtered impurities in a given background. The code requires as an input the plasma background, the sheath model, and the surface erosion data (calculated for example using a BCA or MD code) though it does not consider any modification to the plasma background due to the material impurities [24].

1.3 Thesis Objectives

A robust model of PMI must be able to incorporate a kinetic description of the plasma, surface response, and impurity transport. As the majority of impurities are expected to be ionized a short distance from the wall and return to the surface, tracking impurities through the near-wall plasma and including collisional effects such as ionization and charge exchange is also required. This thesis is thus aimed at developing a method of coupling a multi-species finite-difference Boltzmann solver for the plasma, providing a fully-kinetic description of all the species in partially-ionized conditions, and a material code based on a BCA including dynamic composition, fractal-TRIDYN. The goals of the thesis are thus threefold:

1. Develop a continuum model of the near-wall Boltzmann-Poisson plasma
2. Couple the continuum model to a Monte-Carlo description of the plasma-facing surface
3. Ensure the statistical validity of the coupling approach in order to prepare the model for future applications

Combined, the Boltzmann-BCA method will be able to provide a kinetic description of both plasma and impurities, allowing for the analysis of their coupled effects on sheath structure, material response, and recycling.

1.4 Thesis Overview

With these goals in mind, the theory of both the plasma and surface from the standpoint of the Boltzmann transport equation is first developed to provide a definition of the near-wall plasma and to establish the necessary exchange of information between the two regions. The finite difference approach to solving the Boltzmann equation for plasmas, including a method of quantifying the accuracy of different finite difference methods, is established in Chapter 3. A convergence analysis of the finite difference method applied to a near-wall plasma is applied in order to ensure that the ion velocity distributions at the material surface are accurately constructed. Chapter 4 describes the method of dynamically coupling the multi-species Boltzmann-Poisson plasma treated with

finite differences to the BCA model describing the plasma-facing surface. Included are examples of calculation under divertor-relevant conditions, intended as a proof-of-concept of the coupling approach. The model is capable of providing a dynamical kinetic description of the near-wall physics, including the sheath and presheath structure, the distribution functions of eroded material, and the redeposition processes occurring during plasma bombardment.

Chapter 2

A Boltzmann-Poisson Model of the Plasma–Material Interface

This chapter establishes a two-region Boltzmann-Poisson model of the interface layer between a plasma and a material surface, either solid or liquid. Impacting ions and surface response are considered as boundary conditions between the regions. An overview of the kinetic treatment of plasmas is included with electric fields considered self-consistently through the Poisson equation, which allows systems far from equilibrium to be considered. Particle collisions in the plasma are treated with a BGK collision operator; collisions within the material surface are handled with a dedicated operator handling ion-matter interaction in a binary-collision approximation (BCA). Material surface response to plasma bombardment is considered through the concept of stopping power.

2.1 Transport Equation for Plasma-Material System

Consider a volume of phase space $(\vec{\boldsymbol{x}}, \vec{\boldsymbol{v}})$ populated with N particles of species s in which the density of particles is sufficiently large that in an infinitesimally small region, $N_s(\Delta\vec{\boldsymbol{x}}, \Delta\vec{\boldsymbol{v}}) >> 1$. In this region, the average distribution of particles in space and velocity at time t may be described by the distribution function, $f_s(\vec{\boldsymbol{x}}, \vec{\boldsymbol{v}}, t)$. The evolution of such a distribution function over the configuration space and through time t is governed by the Boltzmann equation [25]:

$$\frac{\partial f_s}{\partial t} + \frac{\vec{\boldsymbol{p}}}{m_s} \cdot \nabla_{\vec{\boldsymbol{x}}} f_s + \vec{\boldsymbol{F_s}} \cdot \nabla_{\vec{\boldsymbol{v}}} f_s = \left(\frac{\partial f_s}{\partial t}\right)_{coll} \tag{2.1}$$

where $\vec{\boldsymbol{p}} = (p_x, p_y, p_z)$ is the three-dimensional momentum vector, m_s is mass, and $\vec{\boldsymbol{F_s}}$ denotes the forces acting on each species. The form of $\vec{\boldsymbol{F_s}}$ must be prescribed depending on the dynamics of the particles. As both plasma and surface particles primarily experience electromagnetic interactions,

they are governed by the Newton-Lorentz force,

$$\vec{\boldsymbol{F}} = q_s(\vec{\boldsymbol{E}} + \vec{\boldsymbol{v}} \times \vec{\boldsymbol{B}}) \tag{2.2}$$

with electric field $\vec{\boldsymbol{E}}$ and magnetic field $\vec{\boldsymbol{B}}$, which are described by the Maxwell equations.

The right-hand side of equation 2.1 describes collisions affecting the considered species. In this thesis, the operator will treated to describe collisional processes occurring at the plasma-material interface. Boltzmann considered collisions under the assumption of molecular chaos, which describes the interaction between two previously-uncorrelated particles with an integral over phase space:

$$\left(\frac{\partial f}{\partial t}\right)_{coll} = \int\int v_r \left(\frac{d\sigma(v_r)}{d\Omega}\right)(f_1' f_2' - f_1 f_2) d\Omega d^3\vec{\boldsymbol{v}}_1 d^3\vec{\boldsymbol{v}}_2 \quad \vec{\boldsymbol{x}} \in [I, II] \tag{2.3}$$

where v_r is the relative velocity, $|v_1 - v_2|$ and $\frac{d\sigma(v_r)}{d\Omega}$ is the differential cross section of the collision. Note that the collision integral in this form encompasses elastic collisions, $|v_1 - v_2| = |v_1' - v_2'|$. Eq. 2.3 is valid for both plasmas and for particles traveling through surfaces under the assumption that collisions are binary events, which is known as the binary collision approximation [26]. Combined, Equations 2.1 - 2.3 create a kinetic description of a plasma-material system (figure 2.1).

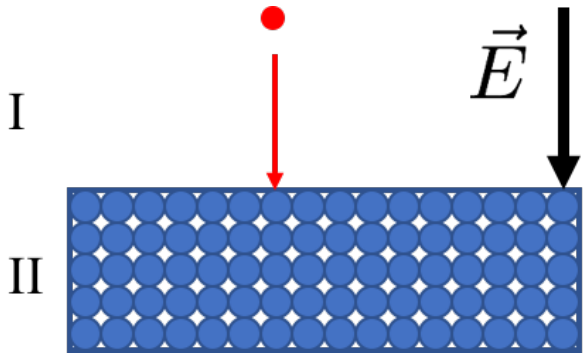


Figure 2.1: Simple diagram of plasma-material interaction system. Region I contains the plasma and region II is the plasma-facing surface composed of a dense lattice of atoms.

2.2 Kinetic Plasma Collision Model

Long-Range Interaction

In the non-relativistic, non-magnetic limit, the Boltzmann-Maxwell system shown in equations 2.1 may be simplified to a Boltzmann-Poisson system in which the electric field is calculated by taking the derivative of the potential ϕ, $\vec{\boldsymbol{E}} = -\vec{\boldsymbol{\nabla}}\phi$, and ϕ is obtained from Poisson's equation:

$$\epsilon_0 \nabla^2 \phi = -\sum_s q_s n_s \tag{2.4}$$

thus treating long-range Coulomb interactions self-consistently [27]. In equation 2.4, n_s is the number density of species s,

$$n_s = \int f_s(\vec{\boldsymbol{x}}, \vec{\boldsymbol{v}}) d\vec{\boldsymbol{v}} \tag{2.5}$$

and the sum takes place over all charged species.

In the most general case Poisson's equation requires kinetic treatment of all ionic species and electrons, but simplifications can be made to treat ions or electrons exclusively. On the short timescale of electron motion, ions may be considered a stationary background density that enforces quasineutrality. If ions must be considered instead, the problem may be reduced by considering the electrons as a fluid. The forces acting on electrons in dynamic equilibrium without a magnetic field are then reduced to pressure and electric forces:

$$\begin{aligned} \vec{\boldsymbol{F}}_{pressure} &= -kT_e \vec{\boldsymbol{\nabla}} n_e \\ \vec{\boldsymbol{F}}_{electric} &= q n_e \vec{\boldsymbol{\nabla}} \phi \end{aligned} \tag{2.6}$$

Integrating yields the Boltzmann relation, $n_e(\phi_2) = n_e(\phi_1) e^{q(\phi_2 - \phi_1)/kT_e}$, and the Poisson equation is transformed into the nonlinear problem:

$$\nabla^2 \phi = \frac{q}{\epsilon_0} \left(n_0 \exp \frac{q\phi}{kT_e} - n_i \right) \tag{2.7}$$

Short-Range Collisions

In Vlasov's description of a plasma, particles interact only through the self-consistently generated electric field through Poisson's equation and other collisional contributions are ignored. Particle-based interaction such as elastic scattering events, electron-impact collisions, and charge exchange between ions and neutrals are incorporated explicitly through equation 2.3. The integral can be simplified under the assumption that the net result of molecular collisions in a plasma is the thermalization of non-equilibrium distribution functions, which is known as the Bhatnagar-Gross-Krook (BGK) model [28]. The BGK collision operator takes the form,

$$\left(\frac{\partial f_s}{\partial t}\right)_{coll} = \nu_{bgk}(f_0 - f(\vec{\boldsymbol{x}}, \vec{\boldsymbol{v}}, t)) \tag{2.8}$$

where ν_{bgk} is the collision frequency and f_0 is the Maxwellian distribution. The BGK model solves the linearized kinetic equation by assuming perturbations are small, and particle distributions are driven back toward Maxwellian conditions on timescales of ν_{bgk}^{-1}. Since ν_{bgk} is a constant, there is no single value that will accurately account for all collisional processes. However, order of magnitude estimates are possible by considering the most prevalent physical processes in the plasma system under consideration by means of an effective collision frequency [29].

2.3 Region II - Surface

Under ion bombardment, ions moving through a solid are constantly subject to multiple physical processes that cause them to disperse their energy by interacting with atoms through both electronic processes, such as excitation, and nuclear collisions. Both types of interaction create collision cascades, and the effect on the surface can be dramatic: sputtering, backscattering, trapping of incident particles, adsorption and desorption, secondary electron emission, and changes in surface morphology all potentially occur under irradiation of a surface, which can all have an affect on the material's behavior and structural integrity [30].

Sputtering occurs when a surface atom gains enough backward momentum to overcome the surface binding energy of the lattice, E_b, and is ejected back into the incident plasma, with the number of surface atoms sputtered per incident particle is defined as the sputtering yield, SY.

SY is fundamentally linked to the ion energy-angle distribution (IEAD) at the wall, for which a detailed description does not exist [8]. The dependence of surface morphology on the IEAD has lead to the development of multiple approaches to treating ion-surface interactions. The collection by Behrisch concerning sputtering of single-element solids and Lieberman's text on material processing contain background information on the underlying physics of surface irradiation [30, 31]. Bringa and Johnson wrote about surface modification physics with a focus on astrophysical applications [32]. Johnson also wrote an extensive overview of atomic and molecular collisions that details the many approximations and empirical formulae that have been developed to describe stopping power [26]. The text compiled by Giannuzzi and Stevie, *Introduction to Focused Ion Beams*, has a chapter dedicated to ion-surface interactions that provides a description of the ion range in solids along with an overview of BCA modeling [33].

Computational and analytical approaches toward ion-surface interactions have also been described. Eckstein's book on computational methods details both BCA and classical dynamics (MD) simulations [34]. Christel takes an analytical approach using a form of Eq. 2.1, allowing for a calculation of ion and recoil range distributions [35]. Notably, Christel's analysis is possible due to previous empirical and analytical descriptions of stopping power and ion ranges developed by Lindhard and Scharff, which was later developed into the Lindhard-Scharff-Schiott (LSS) theory [36, 37].

As described by Bringa *et al.*, stopping power is defined as the energy dissipated along an ion's path as it travels through a surface lattice. This process can be understood by describing the rate of change of the energy: [32]

$$\frac{dE}{dt} = \vec{\boldsymbol{v}} \cdot \vec{\boldsymbol{F}}$$
$$\frac{dE}{dt} = -\frac{dx}{dt} F$$

The stopping power can then be defined as a force acting on particles:

$$-\frac{dE}{dx} = F \tag{2.9}$$

Energy dissipation occurs as an ion impacts a surface by both collide with surface atoms and electronically interacting through long- and short-range forces. Bohr concluded that this loss of energy could thus be split into two separate definitions: nuclear stopping power to describe ions colliding with lattice atoms, and electronic stopping to treat the ion's interaction with the electron gas [38]. Johnson described this energy loss in terms of stopping cross sections, $S(E)$ [26]. Using the definition of the mean free path, $\lambda = (N\sigma)^{-1}$, where N is the density of the medium and σ is the generic collision cross section, stopping power can be defined further. In an inelastic collision, for example, one can define the electronic energy loss per unit length:

$$\left(\frac{dE}{dx}\right)_e = \sum_c Q_c(N\sigma_c) \tag{2.10}$$

In this case, Q_f is the internal energy loss of some electronic collision process c. The elastic stopping power can be similarly described in terms of transferring energy to nuclei in a nuclear collision:

$$\left(\frac{dE}{dx}\right)_n = \int_0^\infty T(N\tfrac{d\sigma}{dT}dT) \tag{2.11}$$

The subscripts in Eqs. 2.10 and 2.11 refer to electronic and nuclear stopping, respectively. Combining the definition of stopping as energy dissipation with Eqs. 2.10 and 2.11, the total stopping power $S(E)$ can be split into electronic and nuclear stopping:

$$S(E) = S_n(E) + S_e(E) \tag{2.12}$$

and the energy loss can then be defined:

$$\left(\frac{dE}{dx}\right) = N(S_e + S_n) \tag{2.13}$$

Described in this way, the problem of ion-surface interactions is simplified into describing elec-

tronic and nuclear stopping processes independently. Nuclear stopping events in this approximation are defined by a sequence of binary collisions governed by the collision integral, Eq. 2.3. Due to the complexity of the ion-surface interaction process, stopping power is frequently described by empirical fits to data, as described by Ziegler [39]. The stopping power and collision integral (Eqs. 2.9 and 2.3, respectively), and more specifically their treatment with the binary collision approximation (BCA), are reviewed in the following sections.

2.3.1 Electronic Stopping

The concept of electronic stopping was developed to describe the interaction of ions with the electron gas as it travels through a surface lattice. Electronic interaction occurs through both local and nonlocal processes, where local energy loss is caused by electronic excitations and nonlocal losses are continuous effects on a moving atom due to electronic interaction with the lattice. Fermi and Teller first derived velocity-dependent stopping, inferring that a free electron gas will act analoguous to a viscous medium for moving ions [40]. This estimation allows electronic stopping to be treated as a frictional force. However, low energy incident ions will experience different forces than high energy ions. Experimental data exists for high energy electronic stopping which allows for accurate empirical fits, but little data exists for low energy collisions. Theoretically the local density approximation will reduce to Newtonian motion through a viscous fluid at the low energy limit, which is the reasoning behind the fits used today [34]. Based on Eq. 2.10, the cross section can be described by any electron-particle collision. Johnson used this to make the observation that at ion velocities that are larger than the mean orbital velocity of the electron, the collision is simply described by a Rutherford cross section [26]:

$$\sigma_f = \frac{2\pi}{m_e v^2}\left(\frac{Z_A e^2}{Q_f}\right)^2 dQ_f \tag{2.14}$$

This would result in the following electronic stopping cross section:

$$S_e = \frac{2\pi}{m_e v^2}(Z_A e^2)^2 \sum_j N_j ln\left(\frac{2m_e v^2}{(Q_j)_{min}}\right) \tag{2.15}$$

For a more general estimation of the cross section, Bethe and Born provided a quantum me-

chanical calculation of the cross section:

$$\sigma_f = \frac{2\pi(Z_A e^2)^2}{m_e v^2} Z_B \frac{dQ_f}{Q_f^2} |F_f(Q)|^2 \tag{2.16}$$

Where $Q = \Delta\vec{\boldsymbol{p}}/2m_e$ and $F_f(Q)$ is the interaction matrix element of the surface atom considered in the collision. This leads to the common Bethe formula, which is described in detail by Ziegler [41]:

$$S_e = \frac{4\pi e^4 Z_1^2 Z_2}{mv^2} \times [ln(\frac{2mv^2}{\Phi}) + ln(\frac{1}{1-\beta^2}) - \beta^2 - \frac{C}{Z_2}] \tag{2.17}$$

In this equation, $\beta = v/c$, m is the electron mass, and c is the velocity of light in a vacuum. C and Z_2 are constant corrections to the mean excitation potential Φ:

$$ln(\Phi) = \sum_n f_n ln(E_n) \tag{2.18}$$

Here, E_n and f_n are energy transitions and oscillator strengths of the target atom. A description of Φ for all elements was developed using wave functions and has been shown to be in good agreement with experimental data [42].

Low energy collisions can be more difficult to describe without empirical data. BCA collisions, which were assumed for the bulk of this analysis, give inaccurate results at low energy. In this regime collisions such as charge exchange begin playing a larger role, and at particularly low collision energies multibody effects must be taken into account. Lindhard and Scharff used the assumption mentioned previously, that Newtonian motion occurs at low energies, to provide a continuous electronic energy loss description of ion-solid interactions at low energy: [36]

$$S_e(E) = 1.212 \frac{Z_a^{7/6} Z_b}{(Z_a^{2/3} + Z_b^{2/3})^{3/2}} \tag{2.19}$$

The values Z_a and Z_b are the atomic numbers of the interacting ions and surface, respectively. The constant term was determined from experimental fits to data. With the energy cross section for electronic stopping, an approximate frictional force on impacting ions can be developed:

$$\vec{\boldsymbol{F}} = -N \cdot S_e(E)\hat{v} \tag{2.20}$$

where $\hat{v}$ is the velocity unit vector.

Alternatively, electronic stopping has also been modeled with the local density approximation, which treats ion-target interactions as a density-averaged free electron gas. As described by Ziegler, electronic stopping in this approximation is described by the volume integral [41]:

$$S_e = \int I(v,\rho)|Z_1^*(v)|^2 \rho d^3\vec{\boldsymbol{x}} \tag{2.21}$$

where $I(v,\rho)$ is the interaction function, v is the velocity, ρ is the free electron gas density, and Z_1^* is the effective charge of the ion with atomic number Z_1. The interaction function was described by Lindhard under the simplifications that the lattice is comprised of an nonrelativistic electron gas of uniform density, where electrons can be described by plane waves. Ions traveling through the lattice under this model are treated as a perturbation of the electron gas. This study lead to the following description of the interaction function [43]:

$$I = \frac{4\pi e^2}{mv^2}\frac{i}{\pi\omega_0^2}\int_0^\infty \frac{dk}{k}\int_{-kv}^{kv} \omega d\omega \Big[\frac{1}{\epsilon^l(k,\omega)} - 1\Big] \tag{2.22}$$

with a longitudinal dielectric constant ϵ^l:

$$\epsilon^l(k,\omega) = 1 + \frac{2m^2\omega_0^2}{\hbar^2 k^2}\sum_n \frac{f(E_n)}{N} \tag{2.23}$$

In Eqs. 2.22 and 2.23, Calculations of $I(v,\rho)$ are detailed extensively by Ziegler *et al.*. However, for the purposes of this work the description by Eq. 2.19-2.20 will be used.

2.3.2 Nuclear Stopping

Nuclear stopping, $S_n(E)$, is a general term describing atomic collisions between impacting ions and the surface lattice. As explained by Ziegler, at energies above 200 keV/amu nuclear stopping is a small contribution to the total energy lost. At lower energies the distance of closest approach of an incident ion A is close enough to the target atom, B, that the interaction takes place within a single

unit cell. This interaction is shown in Figure 2.2. In this regime only two particles interact with each other at any given time, making the BCA model a valid approximation. Treating collisions as a sequence of independent binary events is precisely the motivation behind the collision integral shown in Eq. 2.3, which makes the integral a natural choice to model nuclear collisions.

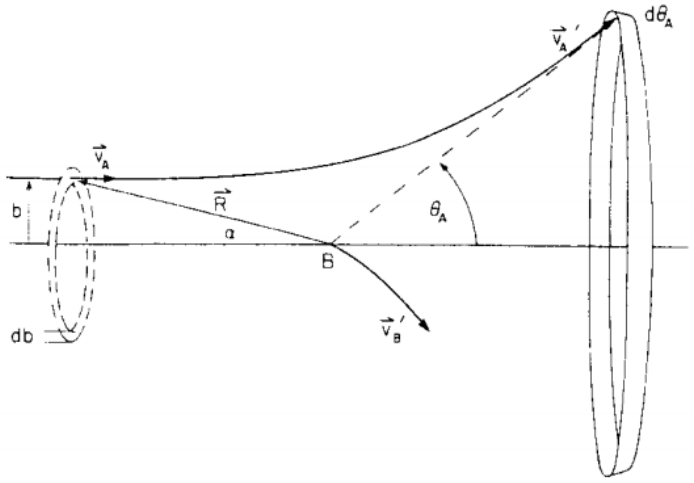


Figure 2.2: Diagram describing the scattering of incident particle A by surface atom B, where b is the impact parameter and θ_A is the scattering angle. Figure courtesy of [26].

From Fig. 2.2 the differential cross section can be described. In the center-of-mass system, the energy is described as

$$E_c = \frac{m_B}{m_A + m_B} E \tag{2.24}$$

and by momentum and energy conservation, the asymptotic scattering angles can be found:

$$tan(\theta_A) = \frac{sin(\vartheta)}{\frac{m_A}{m_B} + cos(\vartheta)} \tag{2.25}$$

$$\Phi = \frac{\pi - \vartheta}{2} \tag{2.26}$$

The recoil energy of the target atom can be described:

$$T = \gamma E sin^2(\frac{\vartheta}{2}) \tag{2.27}$$

where γ is the reduced energy factor:

$$\gamma = \frac{4 m_A m_B}{(m_A + m_B)^2} \tag{2.28}$$

The term ϑ is known as the asymptotic scattering angle, which can be found as a function of the impact parameter b.

$$\vartheta = \pi - 2p \int_0^{R^-1_{min}} \frac{d(\frac{1}{R})}{\sqrt{1 - \frac{V(r)}{E_c} - \frac{b^2}{R^2_{min}}}} \tag{2.29}$$

Finally, the differential cross section can be calculated in terms of the scattering angle:

$$\frac{d\sigma}{d\omega} = \left| \frac{2\pi b db}{2\pi sin(\vartheta) d\vartheta} \right| \tag{2.30}$$

To convert from center-of-mass coordinates back into the lab frame, a transformation is applied:

$$\frac{d\sigma}{d\Omega} = \frac{d\sigma}{d\omega} \frac{((\frac{m_1}{m_2} + cos(\vartheta))^2 + sin(\vartheta))^{3/2}}{1 + \frac{m_1}{m_2} cos(\vartheta)} \tag{2.31}$$

Exact calculation of Eq. 2.31 depends very strongly on the interaction potential, $V(r)$.

2.3.3 Interaction Potentials

Fundamentally, the interaction potential describes the behavior of both short- and long-rage interactions. By far the simplest form of the potential is the pair potential, which considers interaction between two particles at a time. These are only a small subset of interaction potentials and are only valid for a dense, monatomic surface, while molecules will not be described accurately. A non-exhaustive but extensive summary of interaction potentials, including the pair potentials shown here, was compiled by Graves and Pascal [44].

Under the pair interaction assumption, the interaction potential is simply a sum of the two interaction possibilities previously discussed, namely electronic and nuclear repulsion [26]:

$$V_j(R) = \frac{Z_A Z_B e^2}{R} + (\epsilon_j(R) - \epsilon_j) \tag{2.32}$$

where $\epsilon_j(R)$ is the averaged electronic energy, the first term covers the nuclear interaction, and ϵ_j is the total electronic energy of the two atoms. In the short-range limit, which is necessary for BCA to be accurate, nuclear repulsion is dominant. In this case the potential takes the form of a screened Coulomb potential:

$$V(r) = \frac{Z_1 Z_2 e^2}{4\pi\epsilon_0 r}\phi(\frac{r}{a}) \tag{2.33}$$

where ϕ is the screening function, which can be chosen to suit the problem being solved. In ϕ, a is the screening length, which is described by Eckstein [34]:

$$a = \Big(\frac{9\pi^2}{128}\Big)^{(1/3)} a_B Z_{12}^{-1/3} \tag{2.34}$$

where a_B is the Bohr radius. The screening function can be approximated as [45]:

$$\begin{aligned} \phi(\frac{r}{a}) &= \sum_{i=1}^{n} c_i e^{-d_i \frac{r}{a}} \\ \sum_{i=1}^{n} c_i &= \phi(0) = 1 \end{aligned} \tag{2.35}$$

The Moliere, Kr-C, and ZBL potentials[46] are all widely used to approximate Eq. 2.35. The coefficients of these potentials are tabulated in Table 2.1.

The screening length is often approximated in different ways. The most well-known are Firsov's:

$$a_F = 0.88534 a_B (Z_1^{1/2} + Z_2^{1/2})^{-2/3} \tag{2.36}$$

Lindhard's:

$$a_F = 0.88534 a_B (Z_1^{2/3} + Z_2^{2/3})^{-1/2} \tag{2.37}$$

and Robinson's, which is used in the case that $Z_A = Z_B$ and is measured in angstroms:

$$a_R = 0.0750 \tag{2.38}$$

Empirical fits are often used for the screening function and thus the interaction potentials, and extensive tabulated data has been compiled [34, 44].

It should be noted that BCA models fail at very low energies, $O(10eV)$. The distance of closest approach becomes large enough at such low energies that it can be in a neighboring unit cell, at which point it would be interacting with different particles. At this point many-body collisions become relevant, breaking the main tenant of BCA.

Table 2.1: Expansion of screening function for different potentials.[45]

n	**Kr-C**	**Moliere**	**ZBL**
c_1	0.190945	0.35	0.02817
c_2	0.473674	0.55	0.28022
c_3	0.335381	0.10	0.50986
c_4	-	-	0.18175
d_1	0.278544	0.3	0.20162
d_2	0.637174	1.2	0.40290
d_3	1.919249	6.0	0.94229
d_4	-	-	3.1998

Chapter 3

Discretization of the Boltzmann-Poisson problem using Continuum Methods

The following chapter develops a continuum method of solving the Boltzmann equation based on an upwind finite-volume discretization of the phase space with a high-order Strang splitting of the Vlasov operator. First the finite difference approach to solving derivatives will be reviewed, focusing on upwind-biased methods due to their frequent use in computational fluid dynamics and other hyperbolic problems [47]. Then the methods are applied to the Boltzmann equation derived in the previous chapter by means of operator splitting, which transforms the hyperbolic portion of the integro-differential problem into a system of one-dimensional wave equations that can individually be solved using finite differences.

Numerical solutions to partial differential equations naturally introduce error and stability concerns to the problem. Discretizing a continuous domain into a finite grid can coarsely sample a solution, while the finite difference method introduces error on the treatment of derivatives. In order to verify that finite differences are capable of accurately solving the Boltzmann equation, it is necessary to both understand the limits of the method and quantify any errors that are introduced. With this in mind, von Neumann analysis is performed in section 3.2 to ensure that the presented numerical schemes will remain stable, and a method of quantifying the errors that propagate through the numerical approximation is described in section 3.3. The analysis establishes a direct comparison of numerical schemes, which provides a quantitative understanding of the errors one can expect in the solution of a partial differential equation.

Finally, verification tests of the Vlasov-Poisson system solved with the upwind-biased finite difference approach are presented, verifying the method's capability of resolving well-known features in a plasma. The tests include linear and non-linear Landau damping, two-stream instability, and a steady-state plasma sheath formation. In particular, the last test is performed in order to prove that the method is able to robustly converge to steady state.

3.1 Finite Difference Treatment of the Boltzmann Equation

Numerically integrating Eqs. 3.14 and 3.15 require discretizing the phase space, $(\vec{\boldsymbol{x}}, \vec{\boldsymbol{v}})$. Discretization in its simplest form is splitting each dimension, including time, into a structured grid with uniform spacing: $[\Delta x, \Delta v, \Delta t]$. The spatial and velocity dimensions, which are treated identically with finite differences, are targeted first.

3.1.1 Spatial Discretization

A scalar distribution, u, that is discretized will be divided among N_x uniformly-spaced gridpoits, $u_j, u_{j+1}, ..., u_{j_{N_x}}$, where each j corresponds to a spatial grid location, x_j. Finite difference methods provide a way of approximating the derivative of the discretized distribution u through Taylor series expansions. The first-order derivative $\frac{\partial u}{\partial x}$ can be Taylor expanded about a point, x_j:

$$u(x_j + \Delta x) = u(x_j) + \Delta x u'(x_j) + \frac{(\Delta x)^2}{2\,!} u''(x_j) + O(\Delta x^3)$$

And rearranging to solve for the first derivative gives the form:

$$u'(x_j) = \frac{u(x_j + \Delta x)}{\Delta x} - \frac{\Delta x}{2\,!} u''(x_j) + O(\Delta x^3)$$

Similarly, if the point is evaluated around $x_j - \Delta x$, the derivative is approximated as:

$$u'(x_j) = \frac{u(x_j - \Delta x)}{\Delta x} + \frac{\Delta x}{2\,!} u''(x_j) - O(\Delta x^3)$$

The simplest finite difference methods are those for which only the first term in the Taylor expansion is retained. With a grid spacing of Δx, the first derivative for each case becomes

$$\begin{aligned} \left(\frac{\partial u}{\partial x}\right)_j &\approx \frac{u_{j+1} - u_j}{\Delta x} \\ \left(\frac{\partial u}{\partial x}\right)_j &\approx \frac{u_j - u_{j-1}}{\Delta x} \end{aligned} \tag{3.1}$$

which are the forward difference and backward difference schemes, respecitvely. Since the scheme retains only the first term in the Taylor expansion, both schemes are first-order accurate in space. A higher-order approximation can be derived by subtracting the Taylor expansions of the

forward- and backward-cases:

$$u(x_{j+1}) - u(x_{j-1}) = 2\Delta x u'(x_j) + 2\frac{(\Delta x)^3}{3\,!}u'''(x_j) + O(\Delta x^5) \tag{3.2}$$

which, when rearranged to solve for the first derivative and retaining only the lowest-order terms, yields the central difference scheme:

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u(x_{i+1}) - u(x_{i-1})}{2\Delta x} \tag{3.3}$$

This scheme is second-order accurate. To approximate the second derivative, consider Eq. 3.3 centered about $\Delta x/2$ rather than Δx. Applying the same derivative again then yields the form

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_j = \frac{u(x_{j+1}) - 2u(x_j) + u(x_{j-1})}{\Delta x^2} \tag{3.4}$$

The forward-difference approximation of the second derivative is similarly found:

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_j = \frac{u(x_{j+2}) - 2u(x_{j+1}) + u(x_j)}{\Delta x^2} \tag{3.5}$$

Note that central difference methods have no inherent directionality; each gridpoint is influenced by the value of the field next to it, whereas the forward- and backward- schemes have a biased directional dependence. For problems which are inherently directional, such as the one-way wave equation describing the linear advection of a scalar field with flow speed c, $\frac{\partial u}{\partial t} = -c\frac{\partial u}{\partial x}$, it is convenient to exploit the directionality of the problem by applying an upwind-biased numerical scheme. Upwind schemes are widely used in computational fluid dynamics (CFD), and it has been found that the highest potential accuracy per gridpoint of a finite differencing stencil is possible with upwind-biased methods [48]. They will thus be the focus of the present work.

The first-order upwind scheme is simply the forward or backward difference scheme, depending on the local flow direction:

$$\begin{aligned} \frac{\partial u_j^n}{\partial t} &= -c\frac{u_j^n - u_{j-1}^n}{\Delta x} \qquad \langle c > 0 \rangle \\ \frac{\partial u_j^n}{\partial t} &= -c\frac{u_{j+1}^n - u_j^n}{\Delta x} \qquad \langle c < 0 \rangle \end{aligned} \tag{3.6}$$

The method remains first order accurate in this form, but it is easily extended into higher order, which can be accomplished either through including increasingly more terms of the Taylor series or by calculating a polynomial fit of the derivative. The second-order upwind method takes the form:

$$\begin{aligned} \frac{\partial u_j^n}{\partial t} &= -c\frac{3u_j^n - 4u_{j-1}^n + u_{j-2}^n}{2\Delta x} \qquad \langle c > 0 \rangle \\ \frac{\partial u_j^n}{\partial t} &= -c\frac{-u_{j+2}^n + 4u_{j+1}^n - 3u_j^n}{2\Delta x} \qquad \langle c < 0 \rangle \end{aligned} \tag{3.7}$$

the third-order scheme is:

$$\begin{aligned} \frac{\partial u_j^n}{\partial t} &= -c\frac{2u_{j+1}^n + 3u_j^n - 6u_{j-1}^n + u_{j-2}^n}{6\Delta x} \qquad \langle c > 0 \rangle \\ \frac{\partial u_j^n}{\partial t} &= -c\frac{-u_{j+2}^n + 6u_{j+1}^n - 3u_j^n - 2u_{j-1}^n}{6\Delta x} \qquad \langle c < 0 \rangle \end{aligned} \tag{3.8}$$

and the fourth-order scheme is:

$$\begin{aligned} \frac{\partial u_j^n}{\partial t} &= -c\frac{3u_{j+1}^n + 10u_j^n - 18u_{j-1}^n + 6u_{j-2}^n - u_{j-3}^n}{12\Delta x} \qquad \langle c > 0 \rangle \\ \frac{\partial u_j^n}{\partial t} &= -c\frac{u_{j+3} - 6u_{j+2}^n + 18u_{j+1}^n - 10u_j^n - 3u_{j-1}^n}{12\Delta x} \qquad \langle c < 0 \rangle \end{aligned} \tag{3.9}$$

Higher order schemes retain higher spatial accuracy by including increasingly more terms in the expansion. However, it is not necessarily sufficient to select the highest order scheme one can find; while the magnitude of errors in a solution will decrease as the order of the approximation increases, the behavior of the numerical solution can also change, as illustrated in Sections 3.2 and 3.3.

3.1.2 Time discretization

The time derivative can also be discretized with Taylor expansions. To first order, $\frac{\partial u}{\partial t}$ can be approximated by the Euler method, which is analoguous to the space discretization in Eq. 3.1:

$$\left(\frac{\partial u}{\partial t}\right)_{n+1} \approx \frac{u_{n+1} - u_n}{\Delta t} \tag{3.10}$$

If this equation is applied to find future time levels in terms of the current time level, this is known as an explicit scheme. In this case the advection equation will take the form:

$$u_{n+1} = u_n + \Delta t F(t_n, x_n) \tag{3.11}$$

where $F(t_n, x_n)$ designates a discretization of the spatial derivative. This is the simplest approach, but it includes a stability condition on Δt. It is also possible to solve the equation implicitly, which involves a more expensive calculation per time iteration but has the benefit of having no constraint on Δt:

$$u_{n+1} = u_n + \Delta t F(t_{n+1}, x_{n+1}) \tag{3.12}$$

However, this work will focus exclusively on explicit methods.

Higher order time approximations can be used to achieve greater temporal accuracy and stability than the forward Euler scheme, the most common example of which are the set of Runge-Kutta algorithms [49]. The four step, fourth-order scheme (RK4) is defined as:

$$\begin{aligned} u^{(1)} &= u^n + \tfrac{\Delta t}{2} R^{(n)} \\ u^{(2)} &= u^n + \tfrac{\Delta t}{2} R^{(1)} \\ u^{(3)} &= u^n + \Delta t R^{(2)} \\ u^{n+1} &= u^n + \tfrac{\Delta t}{2}(R^{(n)} + 2R^{(1)} + 2R^{(2)} + R^{(3)}) \end{aligned} \tag{3.13}$$

Higher order extensions require increasingly more steps to calculate, creating a more costly tradeoff between computational complexity and accuracy; for example, the RK5 scheme provides fifth-order time accuracy but requires six calculations to solve for a single timestep.

3.1.3 Operator Splitting

To solve Eq. 2.1 with the finite difference methods, it must first be converted from a multidimensional partial differential equation to a problem that can be described by one-dimensional finite differences. By making the assumption that the electromagnetic field is held constant while integrating over a single timestep, the force $\vec{\boldsymbol{F}}$ becomes time-independent during space- or velocity-differentiation [50]. The Boltzmann equation then involves two constant-coefficient differential operators, which allows operator splitting to be applied:

$$\frac{\partial f_s}{\partial t} + \vec{\boldsymbol{v}} \cdot \frac{\partial f_s}{\partial \vec{\boldsymbol{x}}} = 0 \tag{3.14}$$

$$\frac{\partial f_s}{\partial t} + \vec{\boldsymbol{a}} \cdot \frac{\partial f_s}{\partial \vec{\boldsymbol{v}}} = \Big(\frac{\partial f_s}{\partial t}\Big)_C \tag{3.15}$$

However, direct operator splitting is only first order accurate in time [51]. Strang splitting can be applied to increase the accuracy, which further splits the advection equations into half-timesteps [52]. In one spatial dimension and one velocity dimension (1D1V), Strang splitting of Eqs. 3.14 and 3.15 is represented as:

$$F(\Delta x, \tfrac{\Delta t}{2}) F(\Delta v, \Delta t) F(\Delta x, \tfrac{\Delta t}{2}) \tag{3.16}$$

where F denotes a general numerical routine. Since the force $\vec{\boldsymbol{F}}$ is found by solving for the electric field, the splitting algorithm developed by Cheng and Knorr can be used, whereby the Poisson equation is solved after the first spatial half-step [12]. The Laplacian operator in the one-dimensional Poisson equation can be treated simply with Eq. 3.4.

$$\frac{\phi(x_{j+1}) - 2\phi(x_j) + \phi(x_{j-1})}{\Delta x^2} = \frac{-\rho}{\epsilon_0} \tag{3.17}$$

Poisson's equation can be cast into matrix form, $A\vec{\boldsymbol{x}} = \vec{\boldsymbol{b}}$, where $\vec{\boldsymbol{x}}$ stores the values of ϕ at each spatial location, $\vec{\boldsymbol{b}}$ contains the charge density information, and the matrix A contains the coefficients of the numerical stencil:

$$A = \begin{bmatrix} 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ 0 & 0 & 1 & -2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix} \tag{3.18}$$

As described, Poisson's equation is solved by inverting the matrix, $\vec{\boldsymbol{x}} = A^{-1}\vec{\boldsymbol{b}}$, and the electric field is then found from the gradient of the potential, $\vec{\boldsymbol{E}} = -\vec{\boldsymbol{\nabla}}\phi$, to which any of the finite

difference approximations defined above can be applied. The Poisson solver's order of accuracy can be increased by replacing the 2nd order scheme in A with a higher-order method.

Eq. 3.16 is only valid for a case with no magnetic fields, however; in the electromagnetic case the problem is extended into 1D3V. By taking advantage of the cross product in the Lorentz force, Eq. 2.2:

$$\vec{\boldsymbol{v}} \times \vec{\boldsymbol{B}} = \begin{bmatrix} v_y B_z - v_z B_y \\ v_z B_x - v_x B_z \\ v_x B_y - v_y B_x \end{bmatrix}$$

the differential operator can be written as:

$$(\vec{\boldsymbol{v}} \times \vec{\boldsymbol{B}}) \cdot \vec{\boldsymbol{\nabla}}_{\boldsymbol{v}} = (v_y B_z - v_z B_y)\frac{\partial}{\partial v_x} + (v_z B_x - v_x B_z)\frac{\partial}{\partial v_y} + (v_x B_y - v_y B_x)\frac{\partial}{\partial v_z}$$

Since the coefficient of ∂v_i does not include v_i, the equation remains a constant coefficient problem and Strang splitting can be applied in an extended form:

$$F(\Delta x, \tfrac{\Delta t}{2})F(\Delta v_x, \tfrac{\Delta t}{2})F(\Delta v_y, \tfrac{\Delta t}{2})F(\Delta v_z, \Delta t)F(\Delta v_y, \tfrac{\Delta t}{2})F(\Delta v_x, \tfrac{\Delta t}{2})F(\Delta x, \tfrac{\Delta t}{2}) \tag{3.19}$$

In both 1D1V and 1D3V, and indeed up to the full 3D3V phase space, the Boltzmann-Poisson system is thus reduced to a sequence of one-dimensional advection equations that may be individually solved with finite difference methods.

3.2 Stability Analysis

Finite difference schemes are subject to a stability condition based on the relationship between the flow speed and the discretization in space and time. Error is introduced into finite differences through both the truncation of higher-order terms in the numerical stencil and the discretization of space. "Stability" in this sense refers to the behavior of the errors as a numerical solution is marched through time. The solution is unstable if errors become amplified as a solution proceeds.

Von Neumann stability analysis can be used to find the stability condition for a given numerical

scheme in terms of the Courant-Friedrichs-Lewy (CFL) value, $\alpha = \frac{c\Delta t}{\Delta x}$. The method expands the scalar distribution into its Fourier components, $f_j^n = \tilde{f}^n e^{ikj\Delta x}$, which allows error to be described in terms of the amplitude and phase behavior in Fourier space. If errors are damped or remain constant as the solution proceeds, the scheme is considered stable; the scheme will be unstable if the error amplifies. Thus the stability condition requires that every Fourier component remains bounded over time, which provides an upper limit on the amplification factor, $\frac{\tilde{f}^{n+1}}{\tilde{f}^n} = \lambda$:

$$|\lambda| \leq 1$$

where $|\lambda|$ is defined as:

$$|\lambda| = \sqrt{(Re(\lambda))^2 + (Im(\lambda))^2} \tag{3.20}$$

An amplification factor less than one suggests that waves are dissipated as the scalar field f propagates, and the scheme will remain stable. Conversely, if $|\lambda| > 1$ errors are amplified over each timestep and the solution will not converge. Calculating the behavior of the amplitude with respect to α thus enables the stability of a numerical scheme to be controlled by refining or relaxing Δt and Δx as necessary. The relative phase error of a scheme, which describes how the Fourier modes propagate relative to the solution, can also be calculated from von Neumann's analysis:

$$\frac{\phi}{\phi_e} = \frac{\tan^{-1}(\frac{Im(\lambda)}{Re(\lambda)})}{-\beta\alpha} \tag{3.21}$$

where $\beta = k\Delta x$ is the phase angle.

For steady state solutions the stability limit is sufficient - waves are stationary, and as long as stability is ensured the accuracy of the solution depends entirely on the spatial resolution. For unsteady solutions or for simulations which require many timesteps to reach steady state, however, understanding the behavior of waves in a solution is vital [53]. An additional factor that needs consideration is whether or not the λ and $\frac{\phi}{\phi_e}$ depend heavily on the CFL value. If they do, a nonuniform flow field or a variable grid size can allow errors of varying magnitude and type to propagate throughout the simulation domain inconsistently. Von Neumann analysis is therefore beneficial not only for finding a stability condition, but also for predicting whether a solution will

exhibit dissipation and dispersion.

To perform the von Neumann stability analysis on a numerical scheme using the RK4 time discretization, the four step method is first converted into a single-step discretization. In this form the scalar field is converted into a Fourier component in Eq. 3.6 and the amplification factor is found. The von Neumann analysis of the Runge-Kutta algorithm, including the conversion to a single-step method and the form of the amplification factors, were previously found by Zha [53]. The forward Euler method is already in a single-step form and can be analyzed directly. Table 3.1 contains the CFL condition for each numerical scheme, while Figures 3.1-3.4 display the amplification factor and relative phase differences as a function of CFL numbers.

3.2.1 First Order Upwind Scheme

The forward Euler, first-order upwind scheme takes the form:

$$f_j^{n+1} = f_j^n - \alpha(f_j^n - f_{j-1}^n)$$

Von Neumann analysis is applied by substituting $f_j^n = \tilde{f}^n e^{ikj\Delta x}$ and simplifying to find the amplification factor:

$$\tilde{f}^{n+1} e^{ikj\Delta x} = \tilde{f}^n e^{ikj\Delta x} - \alpha(\tilde{f}^n e^{ikj\Delta x} - \tilde{f}^n e^{ikj\Delta x} e^{-ik\Delta x}$$

$$\tilde{f}^{n+1} = \tilde{f}^n - \alpha(\tilde{f}^n - \tilde{f}^n e^{-ik\Delta x})$$

$$\tilde{f}^{n+1} = \tilde{f}^n[1 - \alpha(1 - e^{ik\Delta x})]$$

The amplification factor is defined as the term modifying the Fourier component $\tilde{f}^n$:

$$\lambda = 1 - \alpha(1 - \cos k\Delta x + i \sin k\Delta x)$$

which is the amplification factor of the first-order upwind, forward Euler scheme, and is plotted in Figure 3.1a.

The RK4-first-order upwind scheme converted into a single-step process becomes:

$$f_j^{n+1} = f_j^n - \alpha(u_j^n - u_{j-1}^n) - \frac{\alpha^2}{2}(u_j^n - 2u_{j-1}^n + u_{j-2}^n) - \frac{\alpha^3}{6}(u_j^n - 3u_{j-1}^n + 3u_{j-2}^n - u_{j-3}^n) + \frac{\alpha^4}{24}(u_j^n - 4u_{j-1}^n + 6u_{j-2}^n - 4u_{j-3}^n + u_{j-4}^n)$$

and with von Neumann analysis, the amplification factor takes the form:

$$\lambda = 1 - \alpha(1 - \cos\beta + i\sin\beta) + \frac{\alpha^2}{2}(1 - 2\cos\beta + 2i\sin\beta + \cos 2\beta + i\sin 2\beta) - \frac{\alpha^3}{6}(1 - 3\cos\beta + 3i\sin\beta + 3\cos 2\beta + 3i\sin 2\beta - \cos 3\beta + i\sin 3\beta) + \frac{\alpha^4}{24}(1 - 4\cos\beta + 4i\sin\beta + 6\cos 2\beta + 6i\sin 2\beta - 4\cos 3\beta + 4i\sin 3\beta + \cos 4\beta + i\sin 4\beta)$$

Figures 3.1a and 3.2a present the magnitude of the amplification factor as a function of β for a range of CFL numbers for the forward Euler and RK4 schemes, and figures 3.3a and 3.4a plot the the relative phase error. At a low CFL number the amplification factor remains close to 1, while at higher CFL values the amplification factor drops rapidly as $k\Delta x$ increases. Figure 3.4a suggests that the phase error can be either leading and lagging, depending on the CFL condition and frequency.

3.2.2 Second Order Upwind Scheme

Following the same method as above, von Neumann analysis of Eqs. 3.7 with both the Forward Euler and RK4 schemes is performed for the second order upwind scheme. The Euler method yields:

$$u_j^{n+1} = u_j^n - \frac{\alpha}{2}(3u_j^n - 4u_{j-1}^n + u_{j-2}^n)$$

And the amplification factor is then:

$$\lambda = 1 - \frac{\alpha}{2}(3 - 4\cos k\Delta x + 4i\sin k\Delta x + \cos 2k\Delta x - i\sin 2k\Delta x)$$

For the RK4 analysis, the single-level scheme takes the form [53]:

$$
\begin{aligned}
u_j^{n+1} = u_j^n \quad &- \quad \frac{\alpha}{2}(3u_j^n - 4u_{j-1}^n + u_{j-2}^n) \\
&+ \quad \frac{\alpha^2}{8}(9u_j^n - 24u_{j-1}^n + 22u_{j-2}^n - 8u_{j-3}^n + u_{j-4}^n) \\
&- \quad \frac{\alpha^3}{48}(27u_j^n - 108u_{j-1}^n + 171u_{j-2}^n - 136u_{j-3}^n + 57u_{j-4}^n - 12u_{j-5}^n + u_{j-6}^n) \\
&+ \quad \frac{\alpha^4}{384}(81u_j^n - 432u_{j-1}^n + 972u_{j-2}^n - 1200u_{j-3}^n + 886u_{j-4}^n - 400u_{j-5}^n \\
&+ \quad 108u_{j-6}^n - 16u_{j-7}^n + u_{j-8}^n)
\end{aligned}
$$

The amplification factor is shown in the Appendix. Figures 3.1b and 3.2b show that there is no CFL number that results in a dissipation-free solution. However, using the RK4 scheme improves the stability of the scheme, allowing a larger CFL factor to be used. Figures 3.3b and 3.4b both show that the second-order upwind scheme has leading phase error for most of the $k\Delta x$ range ($\frac{\phi}{\phi_e} < 1$). At high frequencies the errors begin to scatter.

3.2.3 Third Order Upwind Scheme

The third order upwind forward Euler method yields

$$u_j^{n+1} = u_j^n - \frac{\alpha}{6}(2u_{j+1}^n + 3u_j^n - 6u_{j-1}^n + u_{j-2}^n)$$

with an amplification factor of:

$$\lambda = 1 - \frac{\alpha}{6}[2\cos k\Delta x + 2i\sin k\Delta x + 3 - 6\cos k\Delta x + 6i\sin k\Delta x + \cos 2k\Delta x - i\sin 2k\Delta x]$$

The single-level scheme of the RK4-third order upwind scheme is [53]:

$$
\begin{aligned}
f_j^{n+1} = f_j^n \quad &- \quad \frac{\alpha}{6}(2f_{j+1}^n + 3f_j^n - 6f_{j-1}^n + f_{j-2}^n) \\
&+ \quad \frac{\alpha^2}{72}\left(4f_{j+2}^n + 12f_{j+1}^n - 15f_j^n - 32f_{j-1}^n + 42f_{j-2}^n - 12f_{j-3}^n + f_{j-4}^n\right) \\
&- \quad \frac{\alpha^3}{1296}(8f_{j+3}^n + 36f_{j+2}^n - 18f_{j+1}^n - 177f_j^n + 90f_{j-1}^n + 279f_{j-2}^n - 318f_{j-3}^n \\
&+ \quad 117f_{j-4}^n - 18f_{j-5}^n + f_{j-6}^n) \\
&+ \quad \frac{\alpha^4}{31104}(16f_{j+4}^n + 96f_{j+3}^n + 24f_{j+2}^n - 616f_{j+1}^n - 207f_j^n + 1872f_{j-1}^n - 516f_{j-2}^n \\
&- \quad 2304f_{j-3}^n + 2502f_{j-4}^n - 1072f_{j-5}^n + 228f_{j-6}^n - 24f_{j-7}^n + f_{j-8}^n)
\end{aligned}
$$

with the amplification factor given in the Appendix. The stability condition of the third-order upwind scheme is notably higher than either the first- or second-order schemes, at $\alpha = 1.74$. Dissipation exists for all CFL values, and as with the second-order upwind method, the dissipation and dispersion errors are both scattered for high-frequency waves. The scheme is thus insensitive to the CFL value.

3.2.4 Fourth Order Upwind Scheme

The Forward Euler method applied to the fourth-order upwind scheme takes the form:

$$
u_j^{n+1} = u_j^n - \frac{\alpha}{12}(3u_{j+1}^n + 10u_j^n - 18u_{j-1}^n + 6u_{j-2}^n - u_{j-3}^n)
$$

with an amplification factor of:

$$
\begin{aligned}
\lambda = 1 - \frac{\alpha}{12}[3\cos k\Delta x \quad &+ \quad 3i\sin k\Delta x + 10 - 18\cos k\Delta x + 18i\sin k\Delta x + 6\cos 2k\Delta x \\
&- \quad 6i\sin 2k\Delta x - \cos 3k\Delta x + i\sin 3k\Delta x]
\end{aligned}
$$

The single-level scheme of the RK4-fourth order upwind scheme is:

$$
\begin{aligned}
u_j^{n+1} = u_j^n \quad &- \quad \frac{\alpha}{12}(3u_{j+1}^n + 10u_j^n - 18u_{j-1}^n + 6u_{j-2}^n - u_{j-3}^n) \\
&+ \quad \frac{\alpha^2}{144}(9u_{j+2}^n + 60u_{j+1}^n - 8uf_j^n - 324u_{j-1}^n + 438u_{j-2}^n - 236u_{j-3}^n \\
&+ \quad 72u_{j-4}^n - 12u_{j-5}^n + u_{j-6}^n) \\
&- \quad \frac{\alpha^3}{1728}(27u_{j+3}^n + 270u_{j+2}^n + 414u_{j+1}^n - 2078u_j^n - 1431u_{j-1}^n + 9396u_{j-2}^n \\
&- \quad 11964u_{j-3}^n + 788u_{j-4}^n - 3267u_{j-5}^n + 894u_{j-6}^n - 162u_{j-7}^n + 18u_{j-8}^n - u_{j-9}^n) \\
&+ \quad \frac{\alpha^4}{20736}(81u_{j+4}^n + 1080u_{j+3}^n + 3456u_{j+2}^n - 6792u_{j+1}^n - 30932u_j^n + 53496u_{j-1}^n \\
&+ \quad 70944u_{j-2}^n - 271624u_{j-3}^n + 342198u_{j-4}^n - 253080u_{j-5}^n + 126528u_{j-6}^n \\
&- \quad 45144u_{j-7}^n + 11724u_{j-8}^n - 2200u_{j-9}^n + 288u_{j-10}^n - 24u_{j-11}^n + u_{j-12}^n)
\end{aligned}
$$

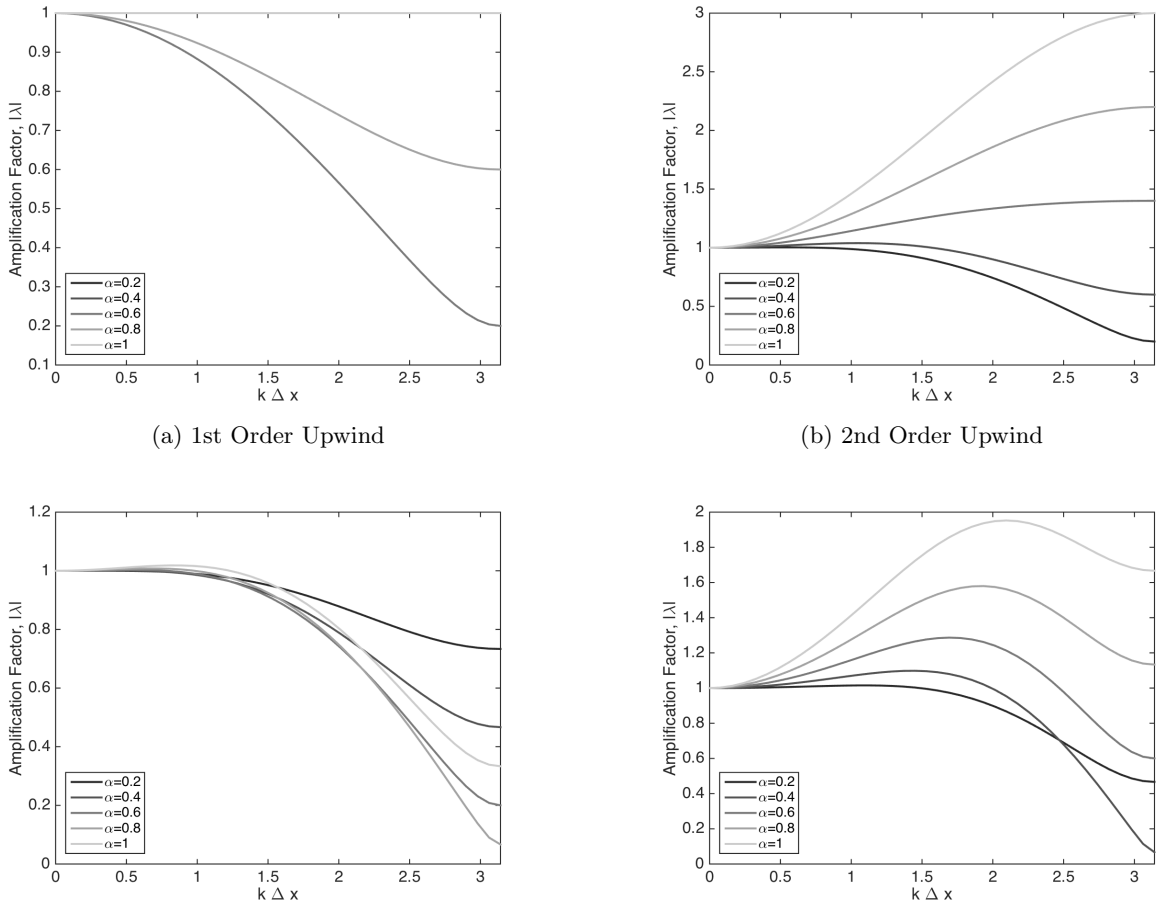
Note that the stability condition for the fourth-order upwind scheme is lower than the third-order method. While the magnitudes are different, the errors shown in Figures 3.1d, 3.2d,3.3d, and 3.4d exhibit behavior similar to the second-order scheme. Both include largely leading phase error, for example.

3.2.5 Stability Discussion

Table 3.1: Maximum value of α for stability of selected numerical schemes.

Space Discretization	**Forward Euler**	**RK4**
First-Order Upwind	1.00	0.46
Second-Order Upwind	0.62	0.69
Third-Order Upwind	0.50	1.74
Fourth-Order Upwind	0.07	1.05

The stability condition is prohibitively restrictive for the forward Euler fourth-order scheme. In addition, the amplification factor in the forward Euler case falls below 1.0 much faster than for the RK4 method. The RK4 scheme creates a sharp increase in the stability condition, allowing for high-order schemes to be used. By comparing the figures, it is evident that the magnitudes of both the amplification factor and relative phase error are noticeably diminished using the RK4 scheme



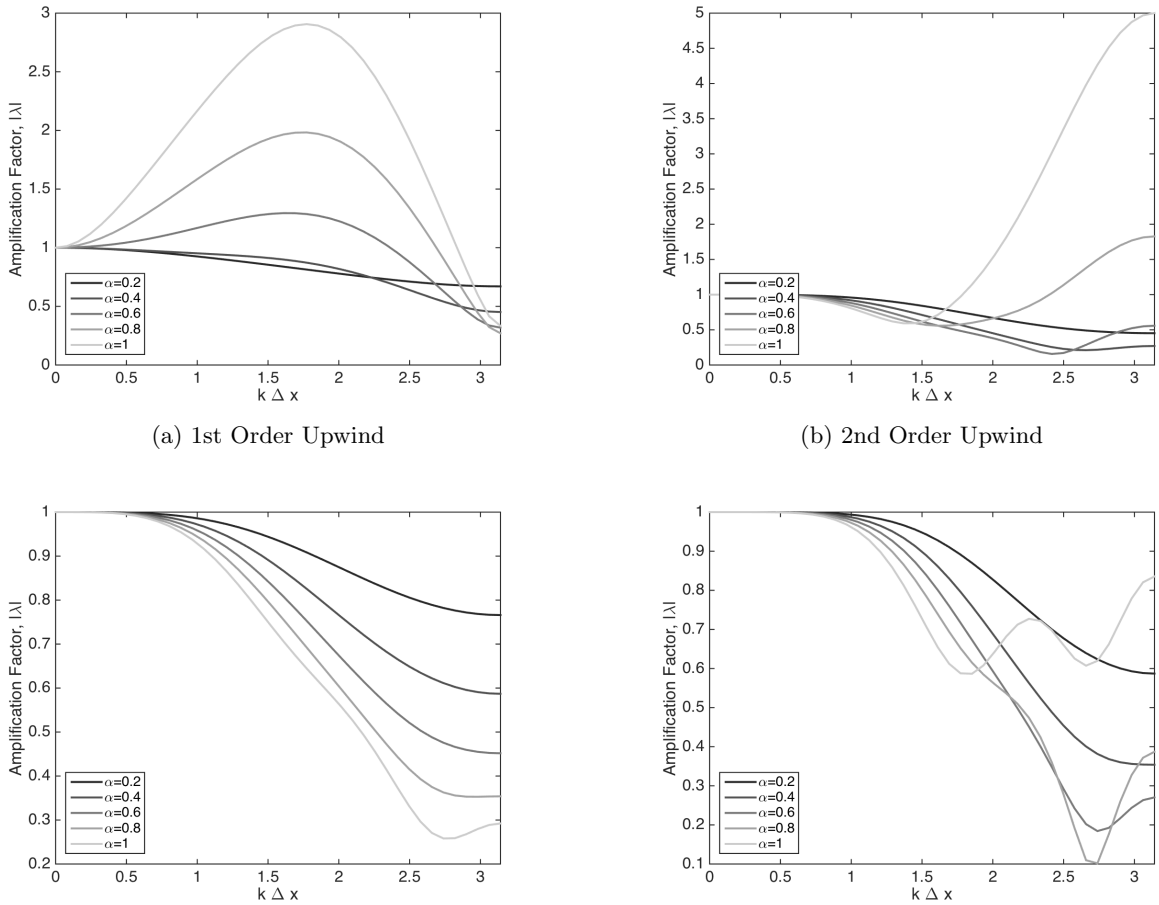
(a) 1st Order Upwind

(b) 2nd Order Upwind

(c) 3rd Order Upwind

(d) 4th Order Upwind

Figure 3.1: Forward Euler amplification factor for a range of CFL values for the first-, second-, third-, and fourth-order upwind schemes.



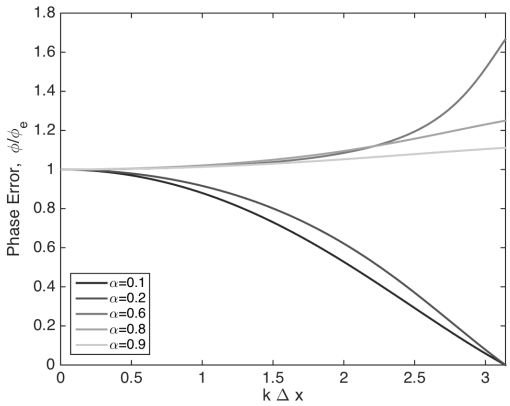
(a) 1st Order Upwind

(b) 2nd Order Upwind

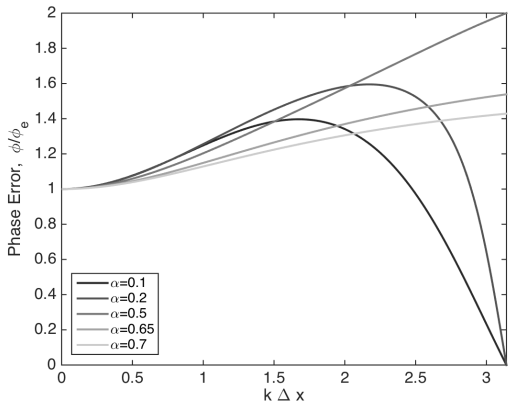
(c) 3rd Order Upwind

(d) 4th Order Upwind

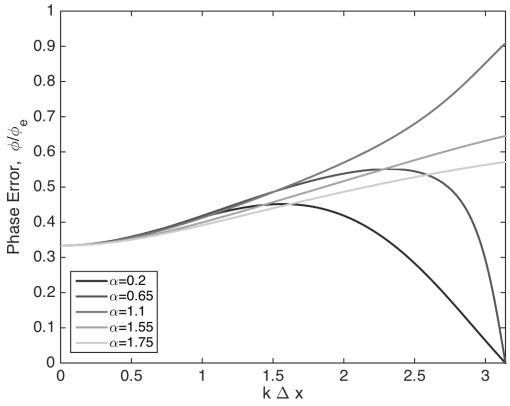
Figure 3.2: 4th-Order Runge-Kutta amplification factor for a range of CFL values for the first-, second-, third-, and fourth-order upwind schemes.



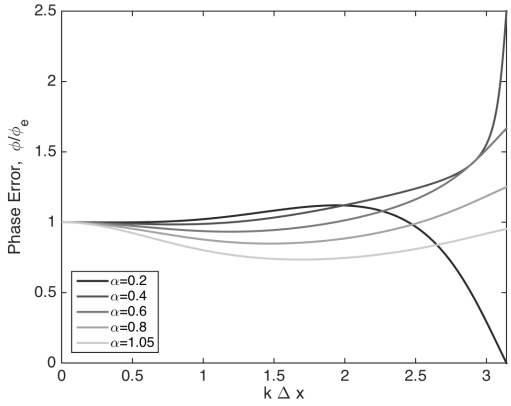
(a) 1st Order Upwind



(b) 2nd Order Upwind

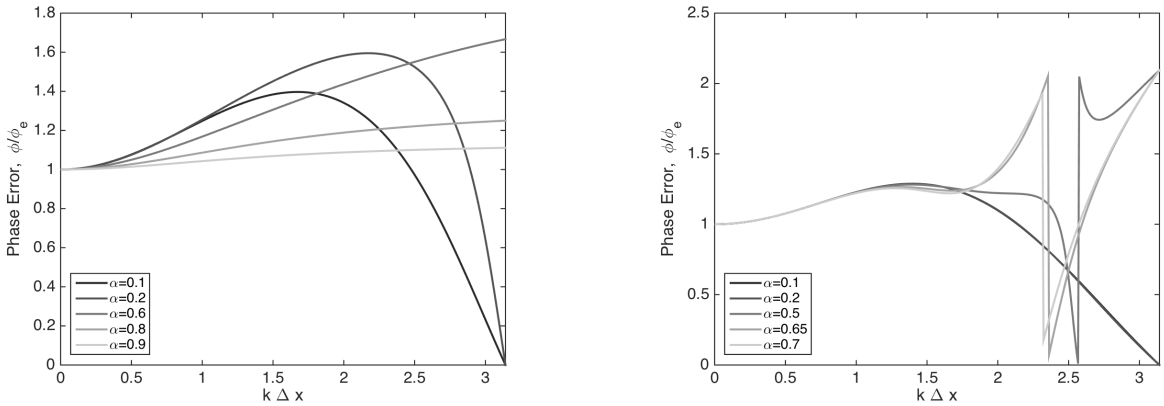


(c) 3rd Order Upwind



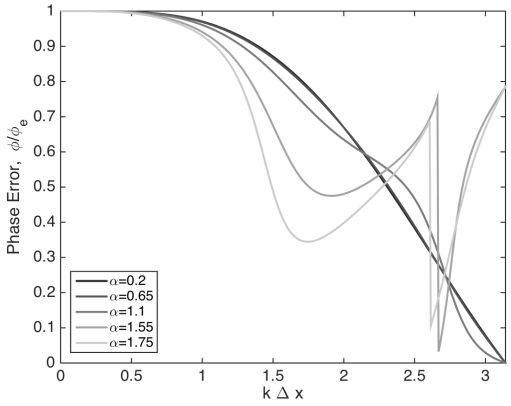
(d) 4th Order Upwind

Figure 3.3: Forward Euler phase error for a range of CFL values for the first-, second-, third-, and fourth-order upwind schemes.

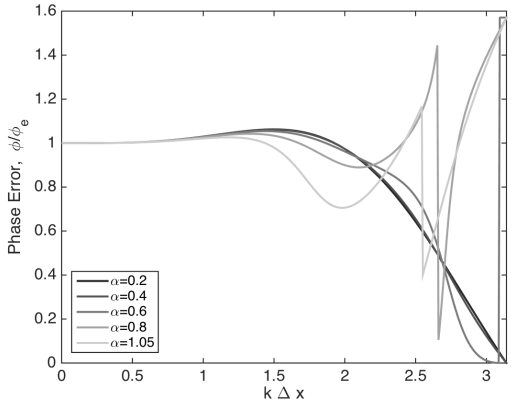


(a) 1st Order Upwind

(b) 2nd Order Upwind



(c) 3rd Order Upwind



(d) 4th Order Upwind

Figure 3.4: 4th-Order Runge-Kutta phase error for a range of CFL values for the first-, second-, third-, and fourth-order upwind schemes.

as well. Notably, the amplification factors and phase errors of the even-ordered schemes exhibit similar behavior. Both second- and fourth-order upwind methods are characterized by a leading phase error, a similar trend in amplifcation factor, and a more restrictive stability condition when compared to the odd-ordered schemes.

The amplification factor and phase error plots also reveal beneficial behavior of upwind-biased differencing. While dissipation and dispersion exist for all CFL factors, the types of errors that propagate are largely insensitive to CFL [53]. For example, in Figure 3.4, the phase errors scatter at high frequency for all higher-order schemes. This result means that time-dependent errors behave relatively consistently across domains even with variable flow speeds and grid sizes. In order to ensure accurate solutions utilizing upwind schemes, the exact value of the CFL factor is irrelevant as long as the stability criteria is met at every grid point.

3.3 Error Analysis

The above discussion provides a method for discretizing solutions with Taylor expansions and for ensuring the resulting approximations are stable, but error analysis on the numerical schemes is necessary to ensure the accuracy of a solution. Amplification factors less than or greater than unity cause dissipation error, the smearing of a wave's amplitude that can damp out the maxima and minima of a solution and spread waves across a wider area. Solutions can also exhibit dispersion error, which is characterized by small oscillations in a numerical solution which create artificial maxima and minima. Takacs *et al.* developed a method for quantifying the errors associated with a numerical scheme by comparing the numerical solution to the analytical solution [54]. Under Takacs's method, the total error is defined as:

$$E_{TOT} = \frac{1}{M}\sum_{j}(f_T - f_D)^2 \tag{3.22}$$

where M is the total number of grid points and j is the grid index. By definition, E_{TOT} is the sum of the dissipation and dispersion errors, E_{DISS} and E_{DISP}, which are calculated as shown:

$$E_{DISS} = [\sigma(f_T) - \sigma(f_D)]^2 + (\bar{f}_T - \bar{f}_D)^2 \tag{3.23}$$

$$E_{DISP} = 2(1-p)\sigma(f_T)\sigma(f_D) \tag{3.24}$$

where σ describes the variance of each solution, $\bar{f}$ is the mean of the solution, and p is the correlation factor between the two solutions, which is defined as

$$p = \frac{\sum(f_D - \bar{f}_D)(f_T - \bar{f}_T)}{\sqrt{\sum(f_D - \bar{f}_D)^2 \sum(f_T - \bar{f}_T)^2}} \tag{3.25}$$

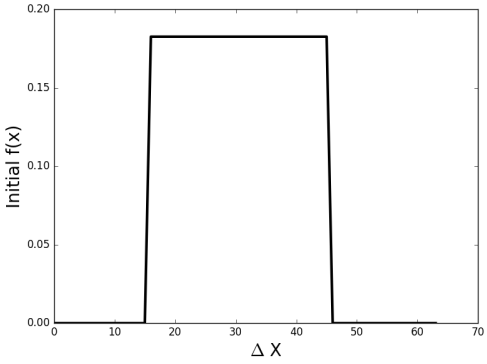
Since quantifying the error under Takacs's method requires an analytical solution, it is convenient to test the efficacy of numerical methods in uniform flow fields that do not deform the solution. The analytical solution under such flow fields is simply the initial condition itself, providing a straightforward method of quantizing numerical effects.

3.3.1 1D Advection

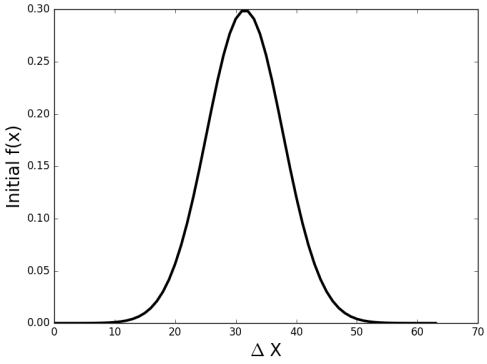
The accuracy of the numerical schemes discussed above applied to the problem of the one-dimensional linear advection of a wave was analyzed with two cases: a step-function and a Maxwellian distribution (figures 3.5a and 3.5b). In both cases the problem was set up with 64 uniformly spaced grid points, and periodic boundary conditions were imposed through the use of "ghost" cells. The error was quantified with Eqs. 3.22-3.24. The true solution, f_T, was defined as the initial condition. In order to compare directly to f_T, each distribtion was simulated through one full cycle with a CFL factor of $\alpha = 0.4$. Note that the RK4 method was used for all tests.

After one period, the solutions of the step function and Maxwellian distribution are shown in Figures 3.5c and 3.5d, respectively, and the errors are quantized in Table 3.2. The first-order uwpind scheme is heavily dissipative, losing much of the definition of the step function and the peak amplitude dropping by 10%. The second-order scheme decreases the disspation error by an order of magnitude, but the dispersion error is larger. The third-order scheme further decreases the dissipation and dispersion error by a factor of 2, and the wave more closely approximates the step function after one cycle, with a similar result found with the fourth-order method.

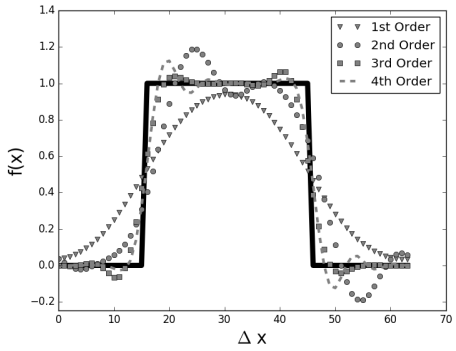
A case more relevant to the Boltzmann equation is the propagation of a Maxwellian distribution, which is shown in figure 3.5d. The errors follow a similar trend as before - as the order of the scheme



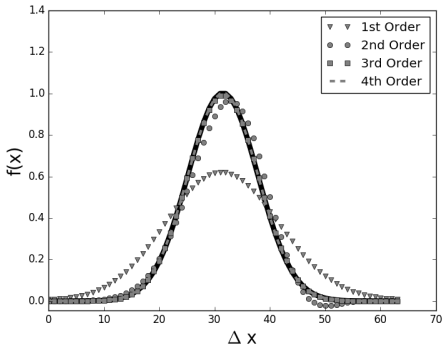
(a) Initial step function.



(b) Initial Maxwellian distribution.



(c) Step function numerical results - single period.



(d) Maxwellian numerical results - single period.

Figure 3.5: Initial conditions of the one-dimensional numerical tests: step function (a) and Maxwellian distribution (b). Numerical solutions for the step (c) and Maxwellian (d) functions after one rotation through periodic boundary conditions. $\Delta x = 64$. In (c) and (d), the solid line is the initial condition.

increases, the total error quickly decreases. The third- and fourth-order upwind schemes remain close approximations of the initial condition. The second-order scheme induces a leading phase error, seen in both figures 3.5c and 3.5d, while the third-order scheme has a more symmetrical dispersion. Additionally, finite difference methods can poorly resolve sharp gradients, as in the edges of the step function. Indeed, Table 3.2 shows that increasing the order of the scheme only marginally decreases the total error in the sytem for the step function, while the same schemes decrease the total error by more than an order of magnitude in the Maxwellian distribution test.

In figure 3.6a, the same test as in figure 3.5 was extended from one to 500 cycles with the CFL factor again held at $\alpha = 0.4$. As the first-order upwind scheme is dominated by strong damping, the result is expected: on long timescales the solution dissipates to a uniform non-zero value, and the

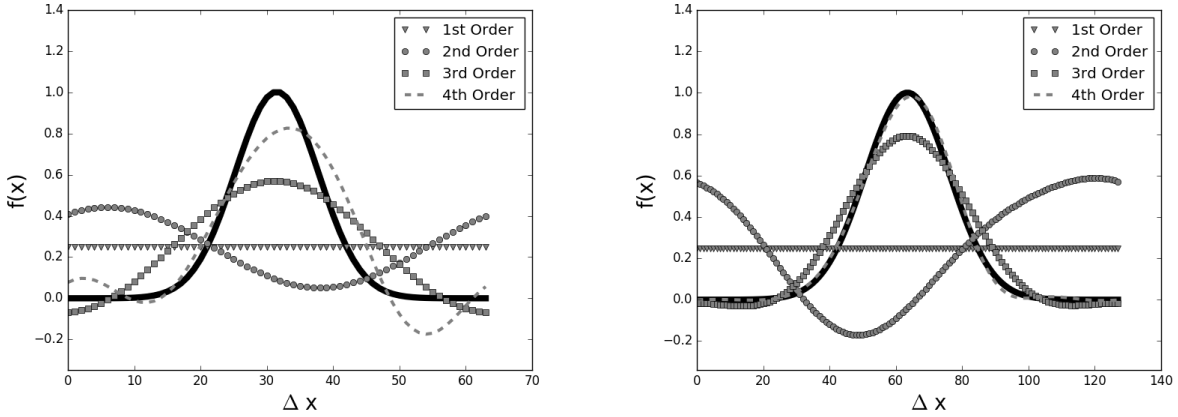
Table 3.2: Dissipation, dispersion, and total errors of the finite difference upwind schemes with $\Delta x = 64$ after one rotation (See Figure 3.5).

Step function			
Numerical Scheme	E_{DISS}	E_{DISP}	E_{TOT}
First-Order Upwind	2.76e−2	3.03e−2	5.79e−2
Second-Order Upwind	8.52e−4	3.46e−2	3.54e−2
Third-Order Upwind	4.72e−4	1.39e−2	1.43e−2
Fourth-Order Upwind	1.98e−4	1.21e−2	1.23e−2
Maxwellian function			
Numerical Scheme	E_{DISS}	E_{DISP}	E_{TOT}
First-Order Upwind	1.39e−2	8.26e−3	2.22e−2
Second-Order Upwind	1.25e−5	1.87e−3	1.88e−3
Third-Order Upwind	1.52e−6	9.76e−6	1.13e−5
Fourth-Order Upwind	1.54e−9	4.78e−7	4.79e−7

shape of the original distribution is lost. The second-order upwind case is dominated by dispersion errors, causing the solution to be completely out of phase. In contrast, the third-order upwind scheme is subject to small dispersion errors that cause local minima to appear both leading and lagging the solution, although the magnitude of these errors are much less than the second-order upwind case.

The similarities between even- and odd-ordered schemes are more readily apparent after 500 cycles. While the phase error is greatly diminished, the fourth-order scheme has a clear leading error that resembles the second-order solutions in figure 3.5. Likewise, the third-order solution has a clear dissipation error reminiscent of the first-order method.

Doubling the number of cells in the simulation creates a sharp decrease in the total error, Figure 3.6b. In particular, the high-order schemes are significantly improved, with the third- and fourth-order methods retaining negligible dispersion error. This is evidence of the convergence of the numerical schemes: as Δx and Δt approach 0, the Taylor approximation from which the schemes were derived approaches the true solution. Notably, the third-order upwind method yields a smaller total error at 3.38e−2 on the coarse grid than the second-order method shows on the refined grid, 3.06e−1. Indeed, this behavior is a defining characteristic of finite difference methods, the implications of which are briefly discussed in Section 3.3.3.



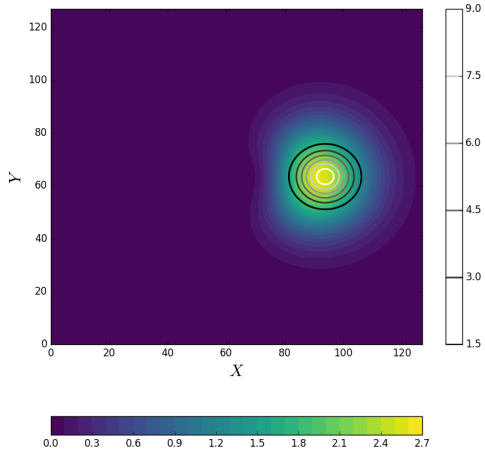
(a) $\Delta x = 64$ (b) $\Delta x = 128$

Figure 3.6: Numerical solutions for the Maxwellian distribution after 500 rotations through periodic boundary conditions. $\Delta x = 64$ (a) and $\Delta x = 128$ (b). The solid line is the initial condition.

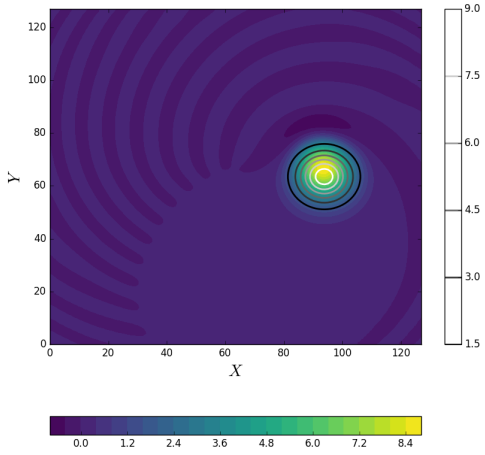
3.3.2 2D Rotation

To verify that the errors remain consistent in multiple dimensions, a similar test was performed on a two-dimensional Maxwellian. The x- and y-dimensions were uniformly spaced from -10 to 10 with $\Delta x, \Delta y = 128$, number of timesteps $Nt = 800$, and timestep size $\Delta t = \frac{\pi}{Nt}$, with open boundary conditions in both dimensions and a CFL factor of $\alpha \approx 0.5$. Operator splitting was applied to reduce the two-dimensional problem into two constant-coefficient wave equations, Eq. 3.14. A rotational field was initialized with $V_x = -2Y$ and $V_y = -2X$. Under this setup the conditions for operator splitting are satisfied, and the numerical advection can be performed on each dimension independently (Eq. 3.16). All simulations were again performed with a fourth-order Runge-Kutta time integration. The results are shown in Figure 3.7 for the first- through fourth-order upwind schemes. The errors calculated from Eqs. 3.22-3.24 are shown in Table 3.4.

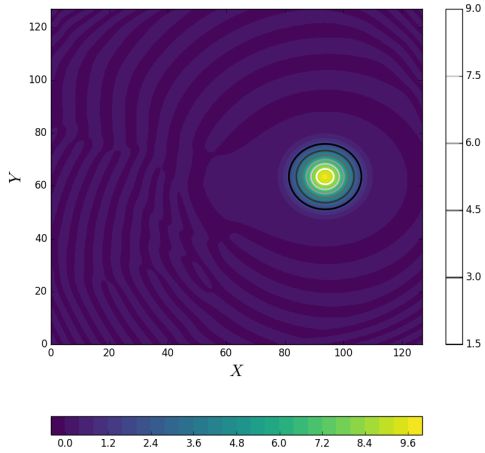
The errors reveal a similar trend to the one-dimensional test. After one full rotation, it is clear that the first order upwind method is the weakest of the three tests, with an error an order of magnitude larger than the others. The second-order upwind method is again dominated by dispersion error, while the third-order upwind scheme is again shown to preserve amplitude within 10% with negligible phase error. The fourth-order scheme performs better overall, with a reduction in the total error by a factor of 25 compared to third-order. Phase errors are clearly visible in the



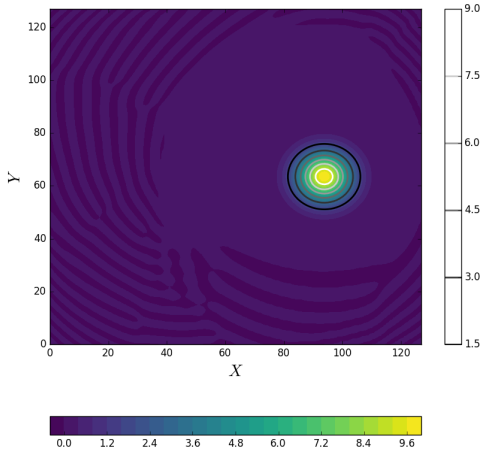
(a) 1st Order Upwind



(b) 2nd Order Upwind



(c) 3rd Order Upwind



(d) 4th Order Upwind

Figure 3.7: Two dimensional Maxwellian distribution in a rotational velocity field. The initial condition is shown as a line contour (right color scale), and the distribution after one full rotation is shown as a filled contour (bottom color scale).

Table 3.3: Dissipation, dispersion, and total errors of the finite difference upwind schemes with $\Delta x = 64$ after 500 rotations (See Figure 3.6).

$\Delta x = 64$			
Numerical Scheme	E_{DISS}	E_{DISP}	E_{TOT}
First-Order Upwind	1.14e−1	3.19e−16	1.14e−1
Second-Order Upwind	4.02e−2	1.57e−1	1.97e−1
Third-Order Upwind	1.26e−2	2.12e−2	3.38e−2
Fourth-Order Upwind	1.24e−4	1.32e−2	1.33e−2
$\Delta x = 128$			
Numerical Scheme	E_{DISS}	E_{DISP}	E_{TOT}
First-Order Upwind	1.14e−1	4.39e−16	1.14e−1
Second-Order Upwind	5.08e−3	3.01e−1	3.06e−1
Third-Order Upwind	1.60e−3	5.69e−3	7.49e−3
Fourth-Order Upwind	3.48e−7	3.85e−4	3.86e−4

higher-order plots while the first-order scheme retains only a single maximum, as in the 1D test case.

Table 3.4: Error measurements of numerical methods.

	E_{DISS}	E_{DISP}	E_{TOT}
First-Order Upwind	2.05e−1	1.42e−1	3.47e−1
Second-Order Upwind	5.01e−4	5.28e−2	5.32e−2
Third-Order Upwind	6.30e−5	3.19e−4	3.82e−4
Fourth-Order Upwind	6.48e−8	1.49e−5	1.50e−5

To measure the performance of the four numerical schemes in the two-dimensional case as grid refinement increases, the total error was recorded over an increasingly small Δx while holding the CFL factor at a constant value of $\alpha = 0.5$. The results are summarized in Table 3.3. The effect of utilizing a high-order derivative approximation is clear: the high-order schemes have greater accuracy on coarser grids than the lower-order methods have on a highly refined grid. For example, the first-order upwind scheme has a total error that reached $E_{TOT} \approx 0.1$ only at the highest tested refinement of $N_x, N_v = 512$, while the third-order method achieved lower error at $N_x, N_v = 64$.

Table 3.5: Total error for the first, second, third, and fourth-order upwind schemes with increasing resolution.

N_x, N_y	1st Order	2nd Order	3rd order	4th Order
64	5.10e−1	2.56e−1	1.31e−2	2.61e−3
128	3.47e−1	5.32e−2	3.82e−3	1.50e−5
256	1.99e−1	4.70e−3	6.73e−6	6.17e−8
512	9.20e−2	3.07e−4	1.06e−7	2.85e−10

3.3.3 Discussion

It is important to note that the error is always relative - doubling the spatial resolution will improve error, but increasing the length of the simulation through time will cause the error to increase proportionally if the solution is cyclical, as is the case for both rotational flow and periodic boundary conditions. For a simulation that pushes distributions toward open boundaries (like in a plasma sheath scenario), however, information will only travel from one side of the domain to the other at maximum. This fact simplifies error control: in a domain from $[-L, L]$, the maximum amount of error in a simulation can be defined by the amount of error a distribution accumulates after moving through $2L\Delta x$ spatial points, or equivalently the amount of error in a distribution being passed through a single cycle in periodic conditions.

Another consideration that must be taken into account is the presence of steep gradients in a distribution. As seen in Figure 3.5, the presented numerical schemes see a sharp increase in the total error if applied to the edge of a step function. Specifically, such regions tend to facilitate the creation of dispersion error, with artifical extrema that can be of appreciable amplitude. The second-order upwind scheme in Fig. 3.5c exhibits an artificial minimum with an amplitude 20% that of the true solution. If the step function represented a real quantity such as particle density in space, the dispersion error creates a region of negative density - clearly a nonphysical result.

Error can be mitigated by increasing the order of the scheme. Finite difference methods are beneficial in this regard: high-order schemes are relatively simple to implement, and doing so can decrease error by several orders of magnitude in some cases. Per table 3.5, if error must be limited to $O(10^{-5})$ for the 2D rotational problem, the third-order upwind scheme can be used with $256 \times 256 = 65536$ total grid points. However, simply choosing the fourth-order upwind scheme

with half the number of grid points in each dimension achieves a similar level of accuracy, and has the added benefit of reducing the computational cost by a factor of 4. Extrapolating to a three-dimensional, three-velocity simulation with the same number of grid points in each dimension, the computational (and memory) cost is reduced by a factor of 64. The benefit of utilizing a high-order method is thus both an increase in accuracy and a substantial decrease in problem size, with the only costs being the time needed to access an additional grid point, additional ghost cells at the boundaries to impose boundary conditions, and potentially a decrease in timestep size.

Reaching a Steady Solution

The results of the above numerical tests provide enough information to determine the conditions necessary to achieve steady state. Tables 3.3 and 3.4 show clearly that error can be controlled arbitrarily by increasing the spatial resolution (at the expense of significant computational cost). Indeed, Figure 3.6 provides the key result that regardless of how much error has accumulated in a solution, the true solution can be recovered by running the simulation with an increased resolution. Furthermore, the true solution is recovered more efficiently with higher-order schemes. Thus stability is the primary concern for reaching a steady solution; if it exists, steady state will be achieved eventually with any stable numerical scheme. Once the existence of such a solution has been found, obtaining a chosen level of accuracy is only a matter of spatial resolution.

Since the rate at which error is decreased with increasing resolution is faster by at least an order of magnitude for a high-order method (see Table 3.3), the fourth-order upwind scheme will be used for future simulations. The third-order method has a larger CFL factor which would allow for larger timesteps to be used, but the error control of the higher order scheme was considered more beneficial than including a larger timestep.

3.4 Numerical Test of a Vlasov-Poisson Plasma

With the numerical methods and their accuracies described, the finite difference method can be applied to the Vlasov-Poisson description of a plasma. The algorithm utilizes Strang splitting (Eq. 3.16) to convert the 1D1V Vlasov equation into sequential linear advection equations, with the electric field solved between half-spatial steps. The fourth-order upwind scheme is used to

discretize the phase space, while the RK4 method is used to discretize time. The Poisson equation is solved with the second-order central-difference method described in section 3.2, and the resulting $A\vec{\boldsymbol{x}} = \vec{\boldsymbol{b}}$ form is calculated with LU decomposition.

3.4.1 Landau Damping

Linear Landau Damping

If a collisionless electron plasma is initialized as a Maxwellian distribution with a sinusoidal perturbation of magnitude γ:

$$f(x, v_x) = (1.0 + \gamma \cos kx)\sqrt{\frac{m_e}{\pi k_B T_e}} \exp \frac{-m_e v_x^2}{2k_B T_e} \tag{3.26}$$

where n is the electron density, m is the electron mass, T_e is the electron temperature, and L is the domain size, the plasma will experience Landau damping. Chen[55] described the dispersion relation for a slightly perturbed Maxwellian electron plasma, $f(\vec{\boldsymbol{r}}, \vec{\boldsymbol{v}}, t) = f_0(\vec{\boldsymbol{v}}) + f_1(\vec{\boldsymbol{r}}, \vec{\boldsymbol{v}}, t)$, and derived an expression for the damping rate for the case when $k\lambda_D << 1$:

$$Im\Big(\frac{\omega}{\omega_{pe}}\Big) = -\sqrt{\pi}\Big(\frac{\omega_p}{k v_{th}}\Big)^3 \exp\Big(\frac{-1}{2k^2\lambda_D^2}\Big) \exp\Big(\frac{-3}{2}\Big) \tag{3.27}$$

where λ_D is the Debye length, ω_{pe} is the electron plasma frequency, and $k = 2\pi/L$ is the wavenumber.

To simulate Landau damping, the Maxwellian distribution was initialized with test parameters $m = 1$, $\epsilon_0 = 1$, $q_e = -1$, $T_e = 1$, and a small perturbation $\gamma = 0.01$, with boundary conditions set as periodic in space and open in velocity inside a domain from $X = [-\frac{\pi}{k}, \frac{\pi}{k}]$ and $V = [-V_{th}, V_{th}]$. Note that under these parameters, $\omega_{pe} = 1$ and $V_{th} = \sqrt{T_e} = 1$. A neutralizing background ion density was considered. To test the validity of the finite difference solution of the Vlasov-Poisson equation, the theoretical damping rate given by Eq. 3.27 was compared to the numerical result. The electrostatic energy was calculated at each timestep, and by applying a linear least-squares fit to the energy maxima over time, the exponential decay rate was found. The electrostatic energy is defined as $E_{norm} = \int \frac{1}{2} E^2 dx$.

In Figure 3.8, the damping rate of the initial linear decrease was calculated to be $Im(\frac{\omega}{\omega_{pe}}) =$

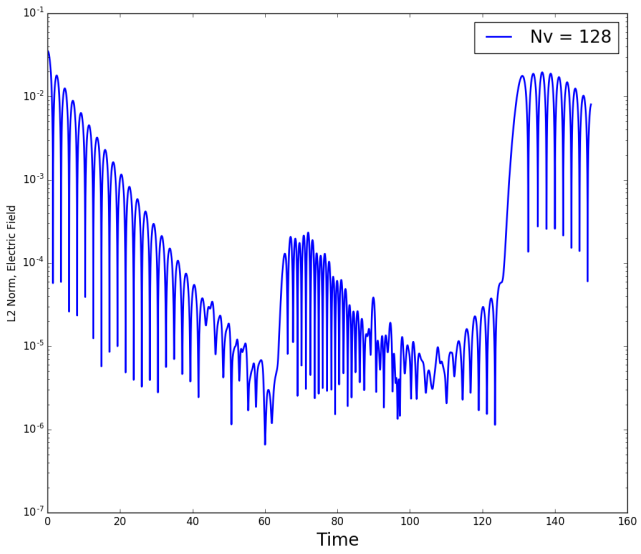


Figure 3.8: Field energy as a function of time. Note the characteristic recurrence effect at $t \approx 130$, caused by filamentation of the phase space.

0.1533. From Eq. 3.27 with $k = 0.5$, the value $Im(\frac{\omega}{\omega_{pe}}) = 0.1534$ is found, achieving good agreement. The steep jump in electrostatic energy is a well-known result of filamentation - finite phase space discretization with periodic boundaries will create increasingly fine structures which the velocity grid is eventually unable to resolve. Pezzi *et al.* described the effect as "recurrence" and reported an expression for calculating the recurrence time [56]:

$$T_R = \frac{2\pi}{kdV} \tag{3.28}$$

In order to compare the theoretical recurrence time expression, T_R was calculated for two values of dV. The results are plotted in Fig. 3.9, with the theoretical values plotted as vertical lines. The simulation shows agreement with the theoretical values.

Nonlinear Landau Damping

With large damping rates the system is quickly driven to a nonlinear state. The same initial conditions were used as in the linear case, with the perturbation factor increased to $\gamma = 0.5$. Plots of the phase space are shown in Fig. 3.11. With a strong perturbation factor, nonlinear effects begin to dominate the solution. This is evident in Fig. 3.11, in which phase space holes are forming

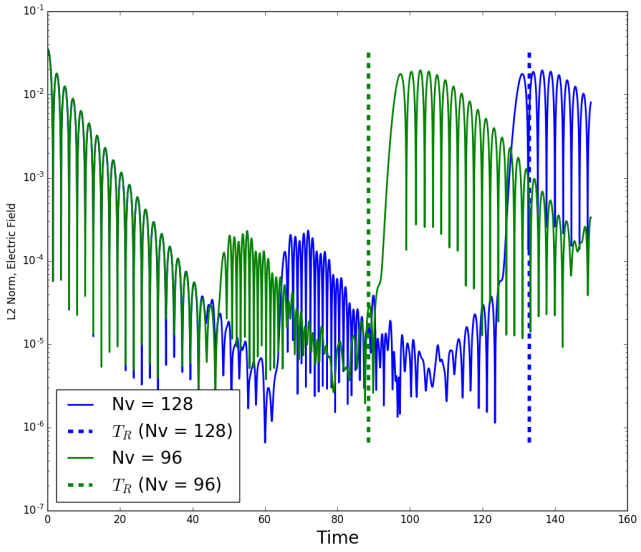


Figure 3.9: Field energy as a function of time. The recurrence times calculated from Eq. 3.28 are labeled with vertical lines.

in the solution. Note again the presence of filamentation in the phase space in Figure 3.11.

The electrostatic energy also evolves differently compared to the linear case - there is a short region of exponential decay before the energy begins to oscillate. This is presented in Fig. 3.10. The electrostatic energy of the nonlinear Landau damping case has been numerically tested extensively in the past, and the results from the current model shows agreement with previous simulations. Filbet and Sonnendrucker presented a comparison of different numerical solvers in the nonlinear damping regime [57], and Fig. 3.10 compares favorably to the high-resolution reference solution presented in their work.

3.4.2 Two-Stream Instability

A two-stream instability is caused by the interaction of two counterstreaming beams. The instability is characterized by the formation of phase space "holes" which trap charged particles. During the initial formation of a hole the energy of the electric field experiences a steep increase followed by a period of damped oscillation as the instability settles. A simulation of the two-stream instability was initialized with the same parameters as before: $m = 1$, $\epsilon_0 = 1$, $q_e = -1$, $T_e = 1$, and a small perturbation $\gamma = 0.01$. In order to ensure that error was mitigated long enough for damped

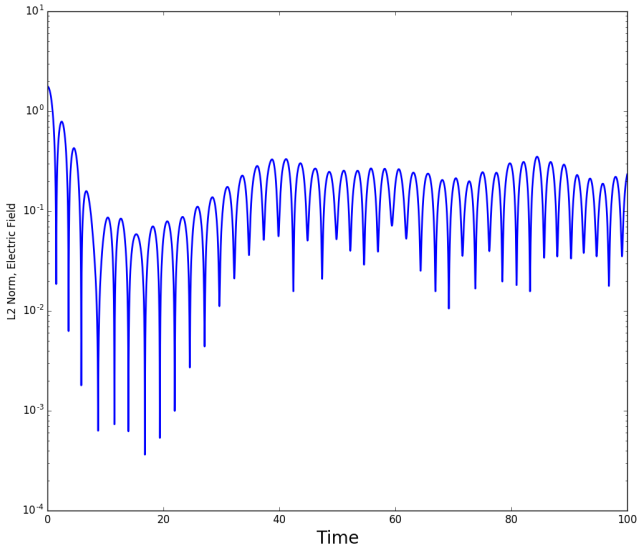


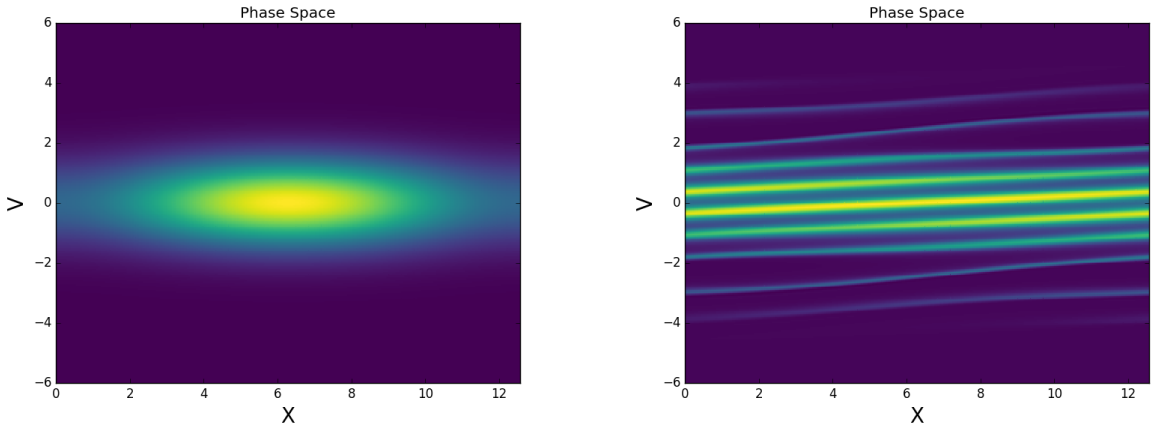
Figure 3.10: Field energy as a function of time for nonlinear Landau damping.

oscillation to be resolved, the grid was highly refined to $N_x = 2560$ grid points. The velocity grid was left at $N_v = 128$; the simulation is periodic in space and therefore limited heavily by the spatial resolution rather than the velocity resolution.

The initial distribution function was modified to include two peaks: one offset by a value $-V_0$ and the other by $+V_0$, where $V_0 = 1.5V_th$:

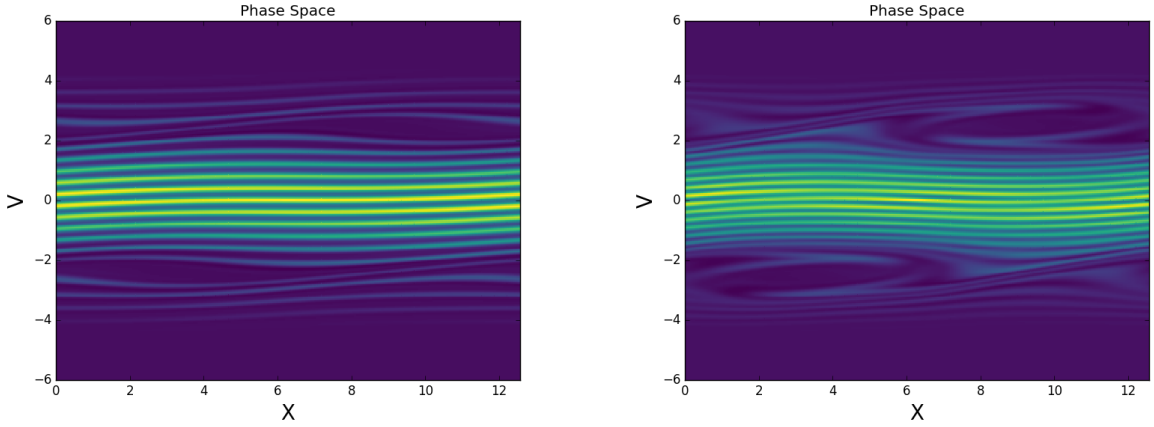
$$f(x, v_x) = (1.0 + \gamma \cos kx)\sqrt{\frac{m_e}{\pi k_B T_e}}\Big(\exp \frac{-m_e(v_x - v_0)^2}{2k_B T_e} + \exp \frac{-m_e(v_x + v_0)^2}{2k_B T_e}\Big) \tag{3.29}$$

As before, the simulation was allowed to evolve with a neutralizing ion background. The phase spaces in both the initial state and after $16\omega_{pe}$ are shown in Fig. 3.12. Filamentation of the phase space is once again visible as the phase space hole forms. The two-stream instability is characterized by a sharp initial increase in energy as the counter-flowing streams interact. Crouseilles *et al.* found that with the above parameters, the energy of the instability peaks at $t = 18\omega_p$, followed by oscillations as the particles are trapped in the resulting hole [58]. The electrostatic energy is shown in Fig. 3.13, showing good agreement with Crouseilles's results.



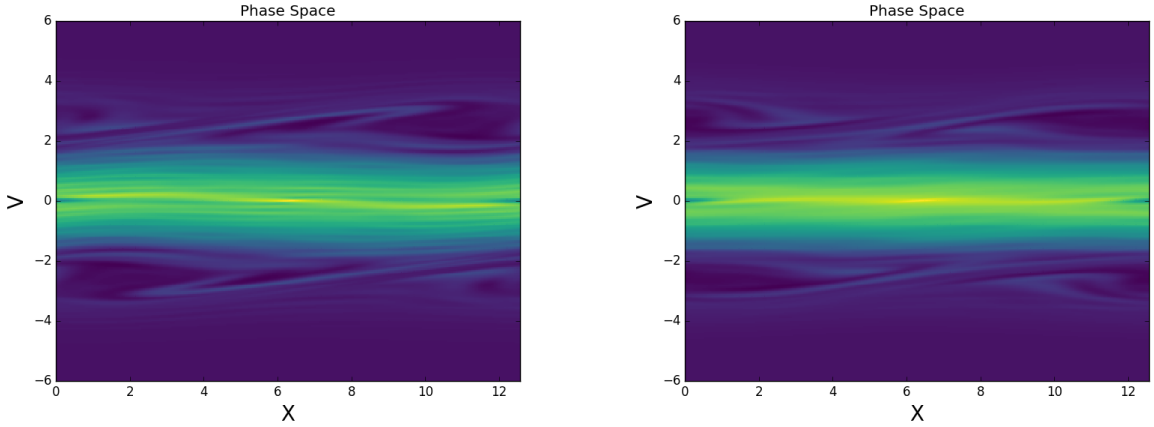
(a) $\Delta t = 0\omega_p$

(b) $\Delta t = 15\omega_p$



(c) $\Delta t = 30\omega_p$

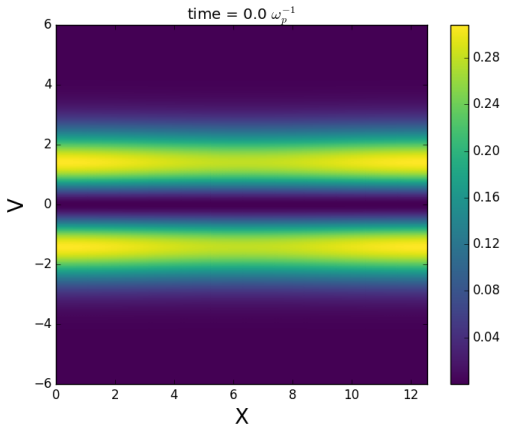
(d) $\Delta t = 45\omega_p$



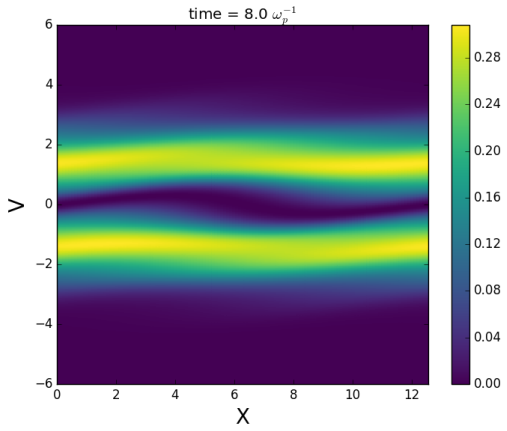
(e) $\Delta t = 60\omega_p$

(f) $\Delta t = 75\omega_p$

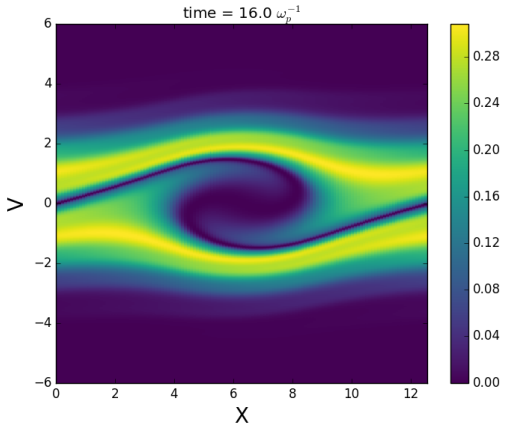
Figure 3.11: The evolution of the distribution function in phase space with a strong initial density perturbation, $\gamma = 0.5$. The perturbation quickly drives the system into a nonlinear regime, and the phasespace holes characteristic of nonlinear Landau damping begin forming between 15 and 30 ω_p. The distribution becomes "smeared" as the filamentation becomes too fine to resolve.



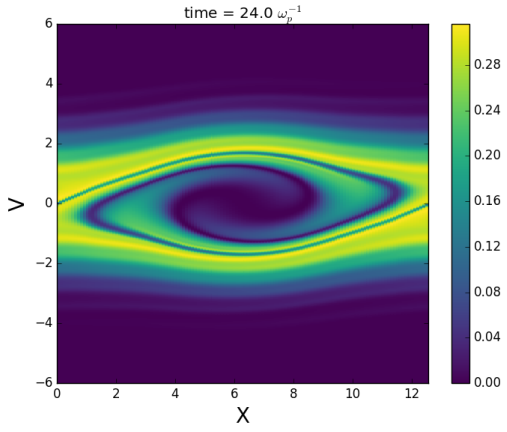
(a) $t = 0\omega_p$



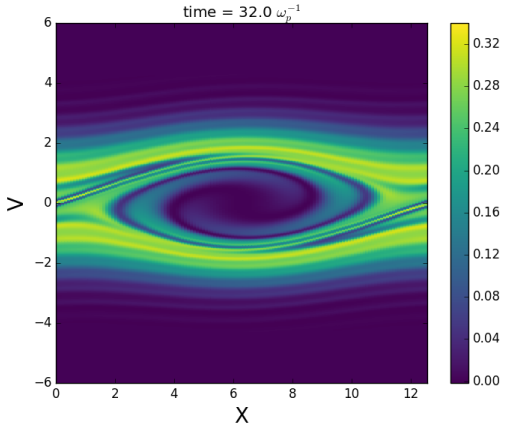
(b) $t = 8\omega_p$



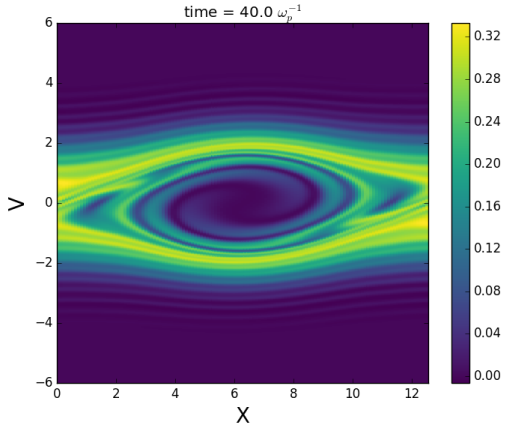
(c) $t = 16\omega_p$



(d) $t = 24\omega_p$

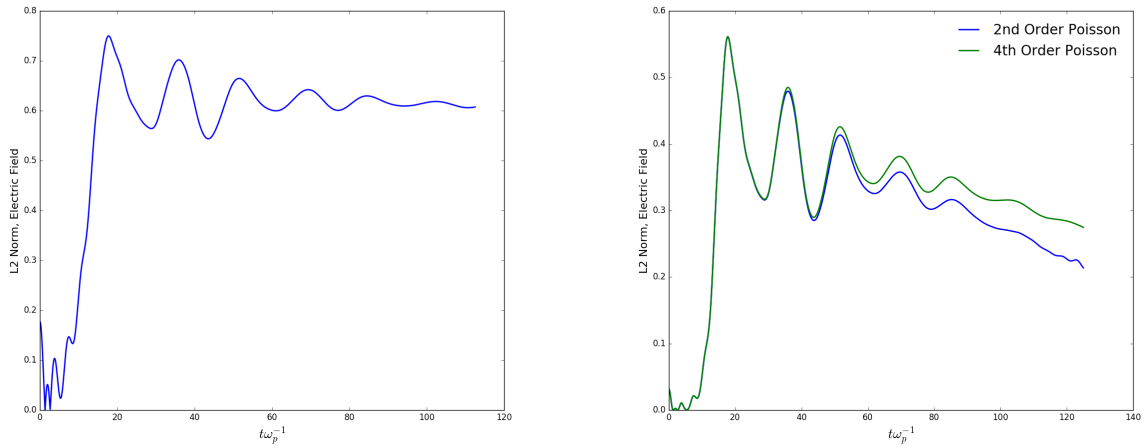


(e) $t = 32\omega_p$



(f) $t = 40\omega_p$

Figure 3.12: Snapshots of the phase space evolution of the two-stream instability. The phase space hole forms within the first $18\omega_p$, at which time the electrostatic energy peaks.



(a) L2 Norm, Refined Grid

(b) L2 Norm, 2nd and 4th Order Poisson Comparison

Figure 3.13: Field energy as a function of time for the two-stream instability for a highly refined grid, $N_x = 2560$ (a) and and a less refined grid, $N_x = 640$ (b). The steep increase in energy occurs at $t = 18\omega_p^{-1}$ the same time as in previous work [58]. The energy damps out to a constant value as the instability stabilizes after $t \approx 100\omega_p$. In (b), a less refined grid was used in order to illustrate the effect of increasing the Poisson solver from 2nd to 4th order.

3.5 Plasma Sheath Convergence Analysis

Unlike the simulations testing Landau damping and the two-stream instability, a plasma sheath is a bounded system. As explained in section 3.3.3, such a system will exhibit only a finite amount of errors. Additionally, stability analysis in section 3.2 concluded that the errors experienced by upwind schemes are relatively independent of the Courant number. Combined, these two facts ensure that a steady state solution - if it exists - will necessarily be found if the simulation is pushed through enough time steps if the solution remains stable; the accuracy of the solution then depends only on the spatial resolution.

A simulation of a near-wall plasma must be able to properly resolve the transition from bulk parameters to the surface. Electrons, having a larger thermal velocity than ions at the same temperature, will strike a material surface and thus leave the plasma faster than ions, causing a net charge differential in a region close to the wall. The negative charge will accelerate ions toward the wall until the net flux of charge is balanced. This charged region is the Debye sheath (DS), and is preceded by a "presheath" (PS) region acting as a gradual transition from the quasineutral

bulk plasma to the sheath entrance. A self-consistent near-wall plasma simulation must therefore satisfy several key conditions:

1. Spatial scales
    - Fine spatial resolution to capture steep gradients in DS, $O(\lambda_D)$
    - Simulation box large enough to contain PS, $O(100\lambda_D)$
2. Temporal scales
    - Timestep resolving characteristic times of species, $O(10^{-1}\omega_{ps}^{-1})$
    - Long simulation times to reach steady state or resolve transient behavior, $O(\mu s)$

In order to demonstrate the Vlasov-Poisson algorithm's ability to satisfy these constraints, the model was modified to include an absorbing wall at the boundary and physical units. The convergence of the steady state solution was analyzed by performing grid refinement.

3.5.1 Near-Wall Plasma Model

The plasma as described by Eqs. 3.14-3.16 is numerically treated with finite differences as shown previously. However, in contrast to the Landau damping and Two Stream Instability test simulations, the domain $x = [0, L]$ is bounded with a grounded wall at $x = 0$, while the plasma is held constant with bulk parameters at $x = L$. In principle all plasma species - ions, electrons, and neutrals - can be modeled together, but in order to avoid the significant computational cost of resolving electron motion, electrons are here assumed to be a background fluid following the Boltzmann relation. The Poisson equation under this assumption becomes nonlinear (Eq. 2.7) and is solved iteratively. Additionally, in order to further relax numerical complexity, neutrals are assumed to be a uniform background population, and the simulation is constrained to the electrostatic case in order to reduce the problem to 1D1V.

As an example calculation, the Vlasov-Poisson model is applied to a helium plasma at divertor-relevant conditions. The domain size, L, was chosen to be $2cm$ - large enough to resolve the transition region from stationary bulk conditions to the surface. The initial plasma parameters are summarized in Table 3.4. The fourth-order Runge Kutta and fourth-order upwind schemes are once again utilized for the finite difference discretization.

Table 3.6: Simulation parameters for the near-wall plasma sheath test.

	Ions (He)	Electrons
Mass (kg)	6.646×10^{-27}	9.11×10^{-31}
Temperature (eV)	10	10
Density (m^{-3})	1.0×10^{19}	1.0×10^{19}

3.5.2 Source Terms and the Collision Integral

The wall at $x = 0$ acts as a perfect particle sink. It follows that there must a source included to balance flux to the walls if particle density is to be conserved and steady state reached. A straightforward approach toward both solving the collision integral and including source terms is the BGK collision operator (see section 2.2), which takes the form:

$$\left(\frac{\partial f_s}{\partial t}\right)_{coll} = \nu_{bgk}(f_M - f(x, v, t)) \tag{3.30}$$

where $f(x, v, t)$ is the velocity distribution being modified, f_M is a background or "target" distribution function, and ν_{bgk} is the collision frequency. The exact forms of f_M and ν_{bgk} depend on the collisional process; for example, for electron-impact ionization ν_{bgk} is $n_n\langle\sigma v(T_e)\rangle$, with the density of background neutrals n_n. To a first approximation, sources and collisions can be combined into a single collision frequency with a uniform background distribution f_M. This approximation is based on the assumption that the net result of collisions in a plasma is to cause equilibration between the colliding species, thereby relaxing the plasma back toward Maxwellian conditions [59]. The collision operator under this assumption is often called a BGK "relaxation" term. While simplistic, the approximation has yielded results in good agreement with both experimental data [60, 61] and fluid models [11]. Because the goal of this section is to establish the ability of the Vlasov-Poisson algorithm to reach steady state and converge to a solution, the simplicity is considered adequate.

Since ν_{bgk} is being treated as a net collision frequency, it can only be understood as an approximation of plasma collisions. Following the method used by Coulette [60], an order of magnitude estimate can be obtained by using realistic reaction cross sections. The cross sections for electron-impact ionization [62] and charge exchange [63] were found and the reaction rates were calculated,

assuming a uniform background neutral density. The cross sections are shown in Fig. 3.14.

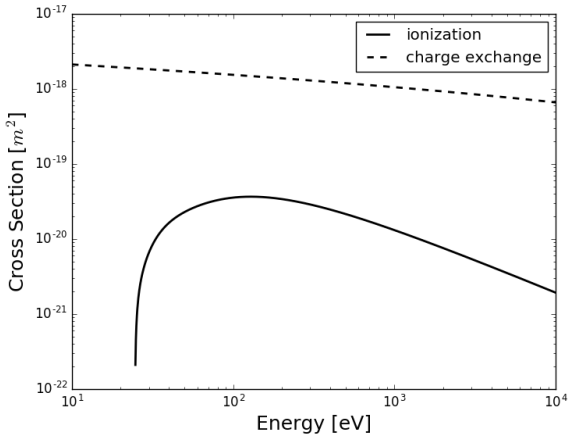


Figure 3.14: Ionization and charge exchange cross sections for He.

The charge exchange and ionization reaction rates with these cross sections and the parameters shown in Table 3.4 were calculated to be $\langle \sigma v \rangle_{ce} = 1.5 \times 10^{-12} m^3 s^{-1}$ and $\langle \sigma v \rangle_{ion} = 1.4 \times 10^{-15} m^3 s^{-1}$, respectively. Assuming a uniform neutral background at the same initial conditions, the collision frequency is thus $\nu_{bgk} = 1.5 \times 10^7 s^{-1}$, or $\nu_{bgk} \approx 7.67 \times 10^{-3} \omega_{pi}$. Based on these calculations, a relaxation frequency on the order of $10^{-3}\omega_{pi}$ was chosen for the sheath simulation. Note that this value was computed with bulk conditions; because ion temperatures and densities vary through the presheath and into the sheath, the value $7.67 \times 10^{-3}\omega_{pi}$ acts as an upper limit on the relaxation frequency for the considered processes.

3.5.3 Numerical Results and Convergence Analysis

In order to verify that the method described previously will converge to a steady condition, a test simulation of a plasma sheath was performed. The IVDF was initialized as a uniform Maxwellian in a domain of length $L = 2cm$ with a wall at $x = 0$ and bulk conditions at $x = L$, and initial parameters as shown in Table 3.4. As no plasma exists inside the wall, a Dirichlet boundary condition was imposed at $x = 0$ with $f(x < 0, \vec{v}) = 0$. The plasma was allowed to evolve through $1\mu s$ with timesteps $\Delta t = 10ps$, which was sufficiently small so that the stabilitiy condition was satisfied.

Steady State

As an initial test, the simulation was performed on a relatively coarse grid ($N_x = 1000$; $\Delta x \approx 2.7\lambda_D$) in order to verify that steady state is achieved. Note that the size of the unmagnetized Debye sheath is on the order of the Debye length. From the parameters in Table 3.4, $\lambda_D = 7.431 \times 10^{-6} m$; this suggests poor resolution of the plasma sheath. The initial and final phase spaces are shown in Fig. 3.15, which illustrates the ions being accelerated toward the wall near $x = 0$. In order to verify that steady state has been reached, the potential at the center of the domain as a function of time is presented in Fig. 3.16. The theoretical floating potential, given by:

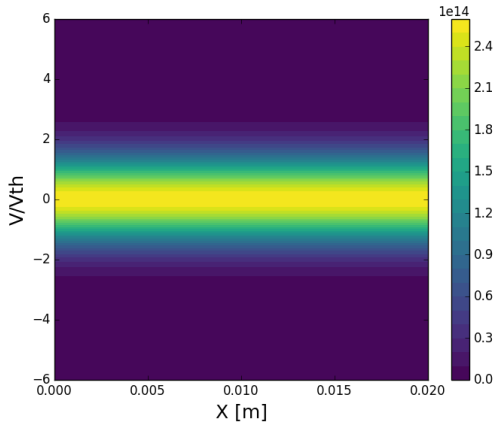
$$\phi_f = \frac{T_e}{2} ln\Big(\frac{m_i}{2\pi m_e(1 + T_i/T_e)}\Big) \tag{3.31}$$

yields a value of $\phi_f = 31.84$. Fig. 3.16 shows that the potential in the plasma is near this value - the drop over time is due to the slight decrease in density at $x = L/2$. The potential can be seen to level off after $t \approx 0.5\mu s$, suggesting that the system has reached a steady solution.

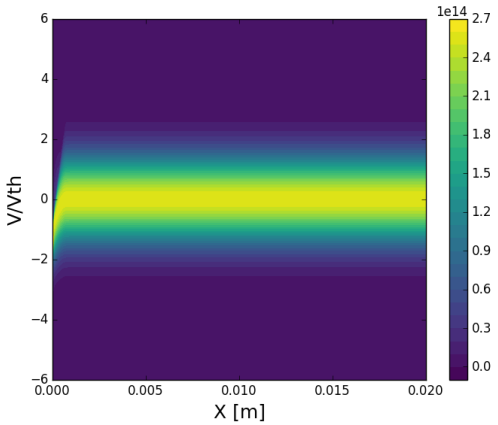
Convergence Test

While a steady state solution has been verified to exist, the reliability of the solution remains unknown. Section 3.3 demonstrated that the upwind schemes are indeed convergent - that is, as Δx approaches zero the solution becomes exact. It follows that since the plasma sheath has a steady state, the solution will converge to a true solution if the grid is refined sufficiently. In order to test for the convergence of the plasma sheath solution, the spatial resolution was increased and the electrostatic energy (see section 3.4.1) was calculated at the end of each simulation. The result is plotted in Fig. 3.17. Note that the number of velocity grid points is held constant at $Nv = 120$; refining the velocity grid has negligible effects on the convergence of the scheme because the solution is limited most strongly by the steep spatial gradients near $x = 0$. As the spatial grid is refined, the sheath becomes better resolved and the potential ϕ is calculated closer to the wall, increasing the energy in the system. The energy reaches an asymptotic limit around a grid size of $Nx = 4000$, corresponding to a grid size of $\Delta x = 0.67\lambda_D$.

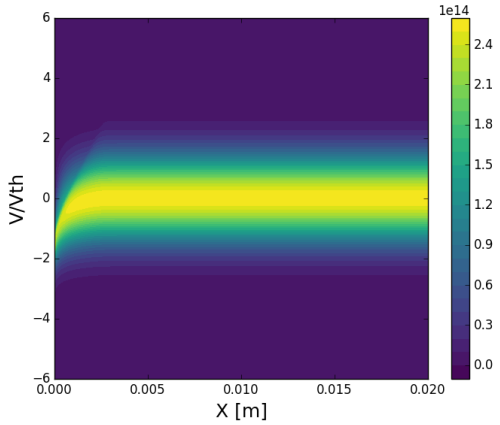
Using the refined grid, the features of the plasma sheath can be analyzed. Fig. 3.18 presents



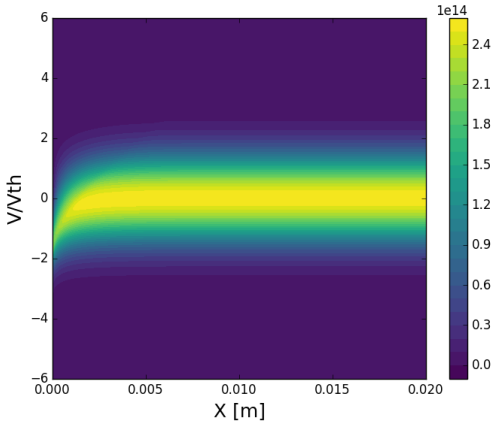
(a) $t = 0s$



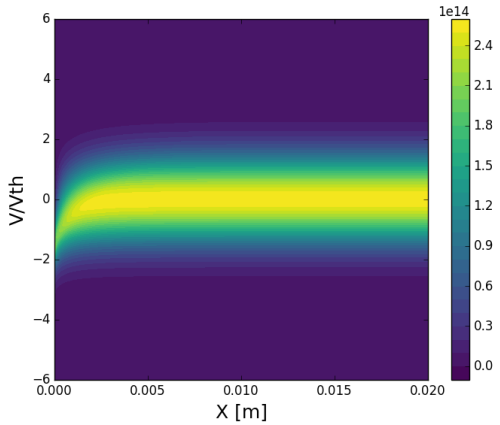
(b) $t = 0.04\mu s$



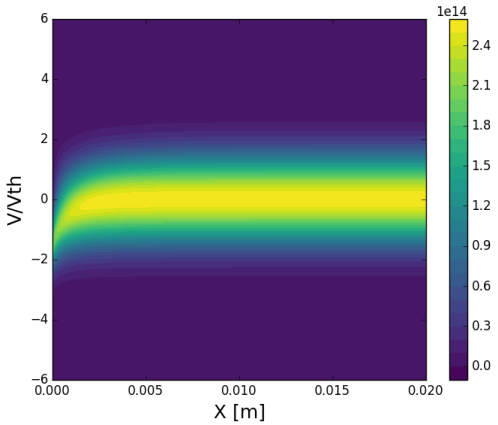
(c) $t = 0.16\mu s$



(d) $t = 0.36\mu s$

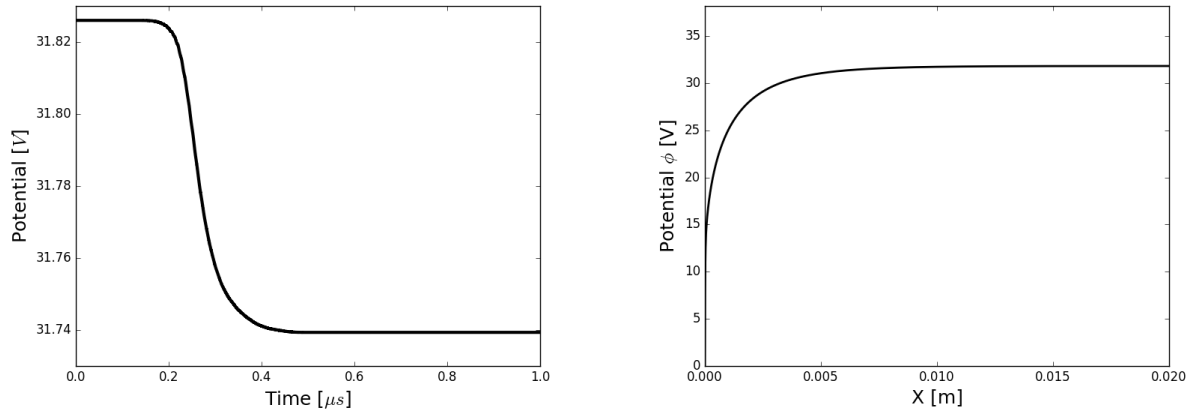


(e) $t = 0.52\mu s$



(f) $t = 1\mu s$

Figure 3.15: The 1D1V velocity distribution shown with a grid resolution of $N_x = 1000$. The grounded wall at $x = 0$ acts as a sink, while the BGK operator both acts as an ion source and drives the system toward thermal equilibrium.



(a) Potential at $x = L/2$ over time

(b) Potential across domain at $t = 1\mu s$

Figure 3.16: The electrostatic potential, ϕ, at the center of the domain ($x = L/2 = 1cm$) (a) and the plasma potential across the domain at $t = 1\mu s$ (b).

the evolution of the IVDF from the wall to the plasma bulk. The relaxation operator causes the IVDF to become strongly asymmetric in the presheath, but the asymmetry is decreased through the sheath; a similar result has been found in Coulette's simulations [60]. The ions are accelerated to $V \approx 2.2Vth$, showing agreement with the theoretical value of the the speed of ions impacting the wall with $T_e = T_i$ [64].

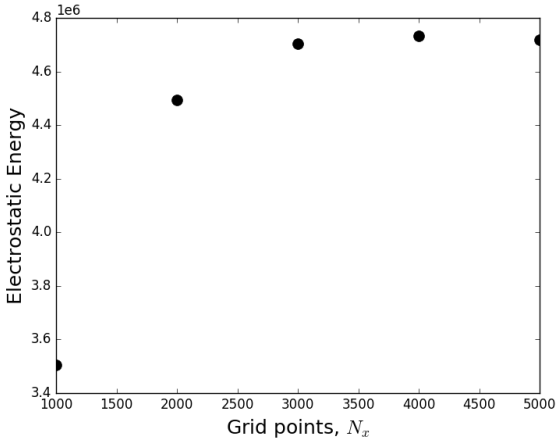


Figure 3.17: Convergence analysis of the plasma sheath equilibrium test. The plot shows the integral of the electrostatic energy in the simulation domain at $t = 1\mu s$. As the grid size is decreased, the sheath becomes better resolved. The energy reaching an asymptotic limit as the sheath becomes more accurate is evidence of the convergence of the numerical scheme.

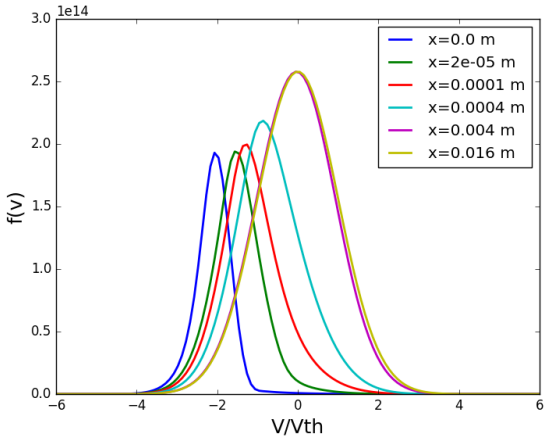


Figure 3.18: IVDF evolution from wall ($x = 0$) to plasma bulk. The BGK collision operator extends the low-energy tail of the accelerating distributions in the presheath.

Chapter 4

Modeling of Plasma-Material Interactions by Dynamically Coupling Boltzmann and BCA Codes

Three key physical components play a major role in plasma-surface interactions: (1) the plasma at the near surface, (2) the material response to the plasma, and (3) the effect of the wall's response on the plasma. The first component may be treated with the kinetic plasma model described in Chapter 3, which is capable of calculating velocity distributions of ions, neutrals, and electrons up to the material wall. The surface itself is often treated as a separate problem, but the plasma and the behavior of the material walls are coupled. Furthermore, because both impurities and the plasma itself may be far-from-equilibrium, a kinetic description of the material's effect on the near-wall plasma is necesary.

A method of treating plasma-material interactions through the dynamic coupling of a Eulerian plasma solver and a BCA surface model, fractal-TRIDYN, is presented in the following chapter. The model is capable of dynamically accounting for phenomena such as material sputtering, backscattering, and implantation with both mono-component and multi-component targets, along with their effects on the structure of the near-wall plasma. The plasma solver calculates the IEAD impacting the wall, while TRIDYN samples discrete particles from the provided distribution in order to calculate the IEADs (Ion Energy-Angle Distributions) of sputtered and reflected particles. Communication between the two models requires an accurate energy distribution at the material surface as input to TRIDYN and numerical density estimation techniques to reconstruct continuous distributions from the discrete data predicted by the BCA model.

This chapter first describes density estimation techniques, which are necessary for converting the discrete data provided by the BCA code to a continuous distribution that can be used in the Boltzmann plasma model. The coupling methodology and the statistical validity of the density estimation techniques are analyzed first with known and tractable distributions, followed by direct application to TRIDYN's output, with the goal of selecting the optimal density estimate by mini-

mizing error and computational cost. Example simulations of the coupled Boltzmann-BCA model applied to a He plasma incident on a Be wall are reported in section 4.5, intended as proof that the coupled codes work as intended.

4.1 Estimation of Density Functions

4.1.1 Density Estimation of a Random Sample

Given a random sample of n observations, $(X_1, X_2, ...X_n)$, a probability density function (PDF) f is the function that determines the underlying density of points, where the PDF f has the property: $\int f(x)dx = 1$. Density estimation is thus the process of calculating the PDF from a finite sample of observed data. The earliest form of a density estimate is the histogram: observations X_i are sorted into bins, where each bin defined as $[x_0 + mh, x_0 + (m+1)h]$ for $m \in \mathbb{Z}$ with a bin width h and bin origin x_0. The PDF $\hat{f}$ under the histogram estimation may then take the form:

$$\hat{f}(x) = \frac{1}{n} \times \frac{\text{number of X in bin x}}{\text{width of bin x}} \tag{4.1}$$

Histograms are straightforward to implement, but they are necessarily discontinuous and improper bin width selection can either completely obscure features of the underlying PDF or cause small features to appear more significant. Additionally, while histograms may be extended into higher dimensions, problems of the bin width and origin selection are only amplified as the number of dimensions increases. For visualization purposes histograms remain useful, but the problems associated with them have nevertheless driven research into alternative density estimation techniques.

Modern density estimation techniques may be split into two main approaches: *parametric* and *non-parametric*. Parametric density estimates assume the PDF may be fitted by some parametric family of functions, and the parameters required to match the data are found through maximization of likelihood functions [65]. As an example, assuming that the shape of the underlying density of observations X_i can be approximated by a finite sum of Gaussian distributions is the basis of the Gaussian Mixture model; the shape of f is predetermined, and the mean and variance of each Gaussian are the unknown quantities that must be iterated. Parametric density estimates were reviewed in [66], with mixture estimation techniques summarized in [67] and [68]. In contrast,

nonparametric estimates assume that an underlying distribution exists, but no strict form of the PDF is forced. The histogram can accurately model any generic distribution as the number of samples $n \to \infty$ without making any assumption about the data, and it is therefore a nonparametric estimate. While the histogram was devised much earlier, earnest research into nonparametric density estimation, and specifically kernel density estimation, was started in the 1950s [69]. A widely cited overview of density estimation for both parametric and nonparametric approaches is Silverman's book [70], which details many of the methods presented in this section. Further detail with applications to statistical problems is presented in [71]. A summary of divergence metrics for density estimates is included in [72]. For further information, the reader is referred to [73, 74].

Both parametric and nonparametric density estimation are heavily researched subjects in statistics, having applications in economics, archaeology, geography, and machine learning, among many other fields [75]. Parametric methods can be both computationally simple and straightforward to implement relative to nonparametric methods, but the latter can be applied to an arbitrary data set without having any prior knowledge of the shape or behavior of the underlying distribution function. Two common density estimation techniques are presented in the following sections: the parametric Gaussian Mixture Model (GMM) and the nonparametric Kernel Density Estimation (KDE).

4.1.2 Gaussian Mixture Model

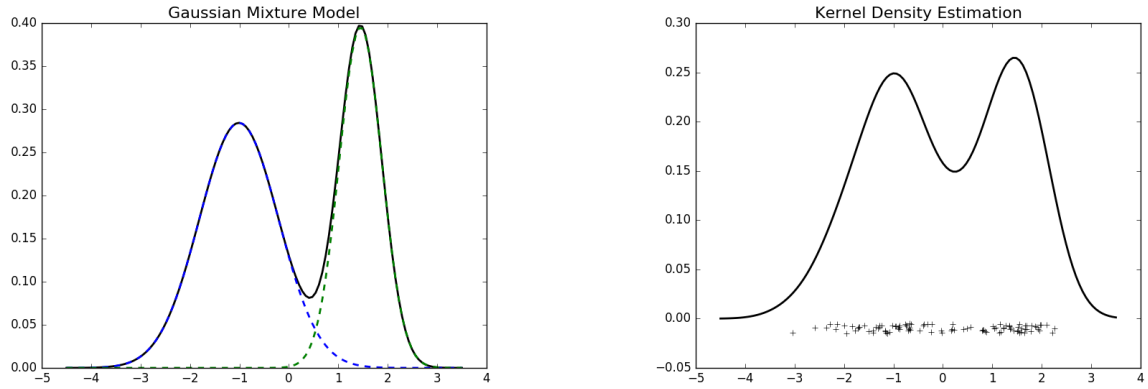
Mixture modeling is a form of unsupervised learning in which the underlying density of a discrete data set is approximated with fits to a series of prescribed distributions, and predictably the GMM utilizes a series of Gaussian distributions. A benefit of the GMM model is the ability to fit to a data set that has distinct groups; each group can be fit to a Gaussian with a separate mean and variance, allowing the multimodality of a distribution to be captured. While the model comes with the obvious constraint of assuming that a generic data sample will be a sum of Gaussian distributions, the assumption is widely applicable due to the central limit theorem.

In order to construct a PDF from a data sample, the GMM is parametrized by the component weights w, the component means μ, and the variance σ. For a sample with K components, the k^{th} component has an associated mean and variance of μ_k and σ_k, respectively, and the component

weights are constructed such that $\sum_{i=1}^{K} w_i = 1$. In the one-dimensional case, the estimated PDF ($\hat{f}$) may then be constructed with:

$$\hat{f}(x) = \sum_{i=1}^{K} w_i \mathcal{N}(x|\mu_i, \sigma_i)$$
$$\mathcal{N}(x|\mu_i, \sigma_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp \frac{-(x-\mu_i)^2}{2\sigma_i^2} \tag{4.2}$$

In order to estimate the values of μ_i and σ_i, GMM may be paired with an Expectation Maximimization (EM) algorithm, which is an iterative process that maximizes the probability of observed data [67, 76].



(a) Gaussian Mixture Model

(b) Kernel Density Estimation (Gaussian kernel)

Figure 4.1: Illustration of how density estimation is performed with the Gaussian Mixture Model (a) and Kernel Density estimation (b). Fitting was performed on 100 data samples drawn from a bimodal Gaussian distribution. In (a), the dotted lines are the two Gaussian profiles predicted by the GMM estimator. In (b), a Gaussian kernel is fit around each data point. In both cases the solid line is the estimated PDF, which is the summed contribution of each component.

4.1.3 Kernel Density Estimation

Kernel Density estimation (KDE), also known as the Parzen-Rosenblatt window method, is a fully nonparametric process - no assumptions are made about the data, and instead the shape of $\hat{f}$ is constructed by fitting a kernel to each data point in the sample. The shape of the kernel is variable, but the most widely used kernels are PDFs themselves and thus have the property that their integral is equal to 1, $\int_{-\infty}^{\infty} K(x)dx = 1$. With a given kernel, $\hat{f}$ can be constructed by setting

the bandwidth h (also called the window width),

$$\hat{f} = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - X_i}{h}) \tag{4.3}$$

where n is the sample size. Thus the KDE method is a sum of "bumps", with the shape of each bump determined by K and the width determined by h. The choice of kernel modifies the shape of the constructed PDF, but ultimately has a minor effect on accuracy (which depends much more strongly on the bandwidth); the choice should instead be made based on such traits as desired smoothness (differentiability) and computational efficiency [70]. Several kernel options are presented in table 4.1.

Table 4.1: Possible kernels for kernel density estimation. Expressions courtesy of [70].

Kernel	*K(x)*
Gaussian	$\frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)$
Epanechnikov	$\frac{3}{4}(1 - \frac{1}{5}x^2)$ for $\lvert x \rvert < \sqrt{5}$
Tophat	$\frac{1}{2}$ for $\lvert x \rvert < 1$

Bandwidth Selection

The bandwidth is analoguous to the bin width in a histogram, and as such the density estimate $\hat{f}$ depends strongly on the value chosen for h. Choosing a bandwidth can be done from a purely subjective standpoint - for example, one may guess and manually change the value until the distribution looks desirable. However, automatic methods that use properties of the data to calculate a bandwidth also exist. Indeed, for many applications including the present work, it is impractical to manually refine h each time a density estimate must be calculated.

A popular choice of automatic bandwidth calculation is Silverman's "rule of thumb" [73]. The rule was developed from the perspective of minimizing the mean integrated square error ($MISE$), which is a measure of the discrepancy between $\hat{f}$ and the true PDF, f:

$$MISE(\hat{f}) = \mathbb{E} \int (\hat{f}(x) - f(x))^2 dx \tag{4.4}$$

where $\mathbb{E}$ denotes the expectation. $MISE$ acts as a global measure of accuracy on $\hat{f}$. Through properties of the expectation, $MISE$ can be reduced to a sum of the integrated squared bias and integrated variance

$$MISE(\hat{f}) = \int [\mathbb{E}\,\hat{f}(x) - f(x)]^2 dx + \int \mathrm{var}\hat{f}(x) dx \tag{4.5}$$

Silverman minimizes this form to calculate an optimal bandwidth, h_{opt} [70],

$$h_{opt} = k_2^{-2/5} \Big[\int K(x)^2 dx \Big]^{1/5} \Big[\int f''(x)^2 dx \Big]^{-1/5} n^{-1/5} \tag{4.6}$$

where $k_2 = \int x^2 K(x) dx$. As the minimization procedure is most tractable in the case of the Gaussian distribution, the Gaussian kernel is used for Silverman's rule, revealing a succint expression for h_{opt}:

$$h_{opt} = 1.06 \sigma n^{-1/5} \tag{4.7}$$

where σ is the standard deviation of the data. Silverman's rule has been shown to be an acceptable choice for a wide range of density estimates, but it is not exact and can improperly resolve more complicated distributions - a bimodal distribution may be oversmoothed by the factor 1.06, for example. Silverman also performed the minimization procedure for other kernels and extended the expression into arbitrary dimensions:

$$h_{opt} = A(K) n^{-1/(d+4)} \sigma_i \tag{4.8}$$

where d is the number of dimensions being considered and $A(K)$ is a parameter dependent on the kernel being used. Suggested values of $A(K)$ are shown in table 4.2, as computed by Silverman. The standard deviation is computed along the i^{th} dimension. If the multidimensional bandwidth is treated as uniform, σ may be treated as an average deviation of the i dimensions. Some applications of kernel density estimation allow a separate bandwidth to be applied along each dimension, but such methods were not utilized in the present work.

Table 4.2: Kernel and bandwidth selection factor A(K), generalized for d dimensions. c_d is the volume of a sphere in d dimensions ($c_1 = 2$,$c_2 = \pi$, $c_3 = 4\pi/3$). [70]

Multivariate Kernel	*K(x)*	*A(K)*
Gaussian	$(2\pi)^{-d/2} \exp\left(-\frac{1}{2}x^T x\right)$	$[4/(2d+1)]^{1/(d+4)}$
Epanechnikov	$\frac{1}{2}c_d^{-1}(d+2)(1-x^T x)$	$[\frac{8d(d+2)(d+4)(2\sqrt{\pi})^d}{(2d+1)c_d}]^{1/(d+4)}$

4.1.4 Discrepancy Measurements: MISE and Relative Entropy

The value of $MISE$, also known as the L2 risk function, is the most widely used metric for comparing a density estimate $\hat{f}$ to the true distribution f. $MISE$ is simply the expected value of the integrated squared error (ISE), the latter of which can be directly calculated if the true underlying distribution is known,

$$ISE(\hat{f}) = \int (\hat{f}(x) - f(x))^2 dx. \tag{4.9}$$

As opposed to the squared error, which provides a point-wise discrepancy analysis, both $MISE$ and ISE are global error estimates. They are commonly used in statistical analysis due to their tractability. In future sections, "error" will refer to ISE unless otherwise specified.

The Kullback-Leibler divergence, also known as relative entropy, is another discrepancy measurement that determines how much two distributions diverge from one another. Notably, the divergence is asymmetric and as such it is not a statistical metric, but it may be used to understand how much two distributions will deviate in behavior. A divergence value of 0 indicates the distributions behave identically. One possible interpretation of the divergence is the information loss incurred when using a function estimate $\hat{f}$ with respect to f,

$$D_{KL}(f|\hat{f}) = \int f \log\left(\frac{f}{\hat{f}}\right) dx \tag{4.10}$$

In the following sections we will use both types of measurements, MISE and Kullback-Leibler divergence (relative entropy), as quality indicators of the density estimators.

4.2 Multivariate Density Estimation and Error Analysis

4.2.1 Density Estimation: Python implementations

All simulations utilizing density estimation techniques are performed in Python 2.7 with the `scikit-learn` machine learning library [77], and plotting is performed with `matplotlib` [78]. In the following tests, kernel density estimation is performed with the package `sklearn.neighbors.KernelDensity`, which includes six possible kernels to choose from. The relative tolerance is defaulted to 1×10^{-8} and may be set with the `rtol` parameter, allowing for computational speed to be greatly increased at the expense of accuracy. While a straightforward implementation of KDE with N_e evaluations of N_p points has computational complexity of $O(N_e N_p)$, the `scikit-learn` algorithm utilizes KD Tree data structures in order to accelerate computation and reduce computational complexity.

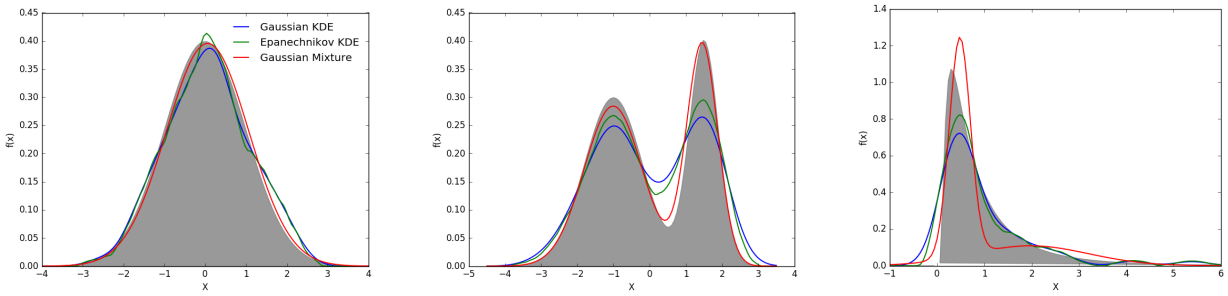
Gaussian mixture estimation is performed with the `sklearn.mixture.GMM` package. While `GMM` requires the number of fitting Gaussian distributions to be set as an input and therefore necessitates some information about the underlying data, the process has been automated by sweeping through several choices, with the best model being selected based on the minimization of the Akaike[1] information criterion [79]. Both `KernelDensity` and `GMM` are able to perform multivariate density estimation.

Univariate Density Estimation

Initial tests were performed in 1D by sampling from three different PDFs: a Gaussian distribution, a bimodal Gaussian, and an inverse Gaussian (Wald), shown in figure 4.2. 100 random samples were drawn from each distribution and density estimation was performed using the samples. Accuracy of the resulting density estimates is thus relative to the true distributions. For the kernel density cases, Gaussian and Epanechnikov kernels are used, with the bandwidths calculated with equation 4.6.

The Gaussian mixture estimates are predictably strong for the Gaussian and bimodal distributions, closely matching the original PDF with little variation. However, both kernel density

[1] The Akaike information criterion provides a method for model selection based on the relative quality of different statistical models for a given set of data.



(a) Gaussian (b) Bimodal Gaussian (c) Wald

Figure 4.2: Density estimation in one dimension using kernel density estimation (blue and green lines) and Gaussian mixtures (red lines). The gray filled plots are the exact probability density functions from which random samples were drawn. The accuracy of the resulting density estimates are determined by comparison to the true distributions.

estimates more closely resolve the tail of the Wald distribution in figure 4.2c. This is an expected result - since `GMM` approximates PDFs by summing together Gaussians, long tails and steep gradients (as for $x < 0$) are features that a standard Gaussian distribution lacks and will thus be poorly resolved. Comparing the kernel density estimates, the Epanechnikov kernel more closely approximates the ampitude of the distributions in all three cases, but the resulting estimates are less smooth than the Gaussian kernel estimates.

Multivariate Density Estimation

The `scikit-learn` library allows density estimation to be easily extended into higher dimensions. In figure 4.3, a two-dimensional Gaussian PDF was initialized and 100 random samples were drawn, which were used to construct densty estimates. The bandwidths for the kernel methods were selected with equation 4.8 and the values in table 4.2.

Results similar to those presented in figure 4.2 are seen in the two-dimensional case. Outliers in the data sample are noticeably weighted more heavily for the kernel density estimates than for the gaussian mixture, which causes local maxima and "bumps" to appear on the edges of the PDF. While reliable results were provided with only 100 samples in the one-dimensional case, such a sample size is less effective in two dimensions, as evidenced by the wide spread shown in all three density estimates in figure 4.3. The same result was found by both Silverman and Epanechnikov - the number of samples required to constrain the relative mean squared error to the same value

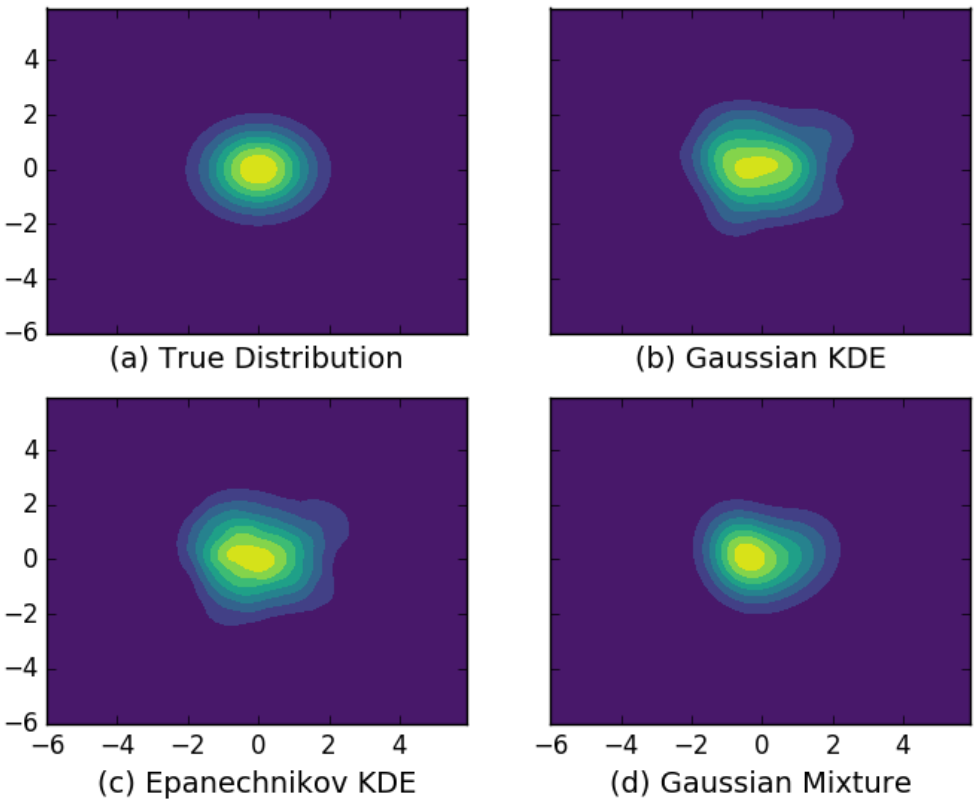


Figure 4.3: Two-dimensional density estimation example. 100 samples were drawn from a Gaussian distribution (a) and used to construct density estimates using a Gaussian kernel (b), an Epanechnikov kernel (b), and the Gaussian mixture model (c).

increases as the dimensionality of the problem increases.

Only minor differences are noticeable between the Epanechnikov and Gaussian kernels, specifically the spread of the peak. While the values of the bandwidths used for each kernel are different, the expressions in table 4.2 are derived using the same method of minimizing $MISE$ for the individual kernel. Indeed, the choice of kernel in the KDE approach will be shown to have a marginal effect on the accuracy of an estimate in the following section.

4.2.2 Error Analysis

Univariate Case

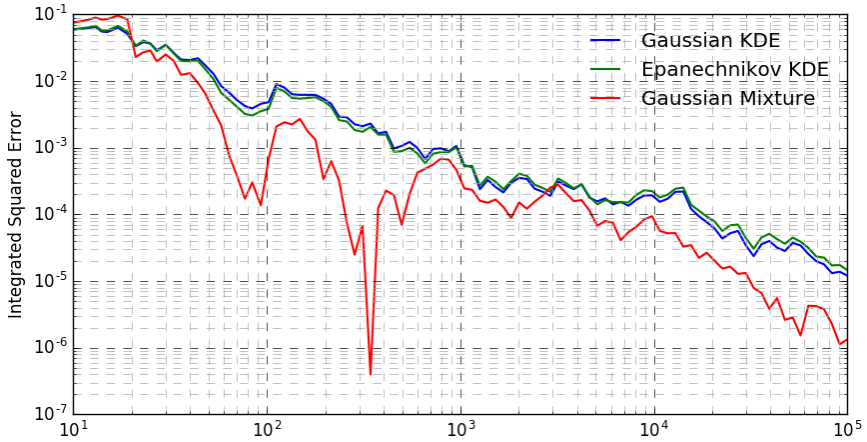
Equation 4.9 was used to calculate the integrated squared error for the density estimation techniques as a function of increasing sample size. Samples were drawn from Gaussian, bimodal Gaussian, and Wald distributions (figure 4.2). The calculated errors are shown in figure 4.4. The results correspond to the subjective appearance of the density estimates in figure 4.2: `GMM` performs better than both `KernelDensity` estimates for the Gaussian and bimodal Gaussian distributions, and the Wald distribution is poorly resolved by the Gaussian mixture technique. Nevertheless, figure 4.4c shows that all three methods struggle with the Wald distribution; the lowest ISE for the Wald test is an order of magnitude higher than the bimodal test and almost two orders of magnitude lower than the standard Gaussian example.

Multivariate Case

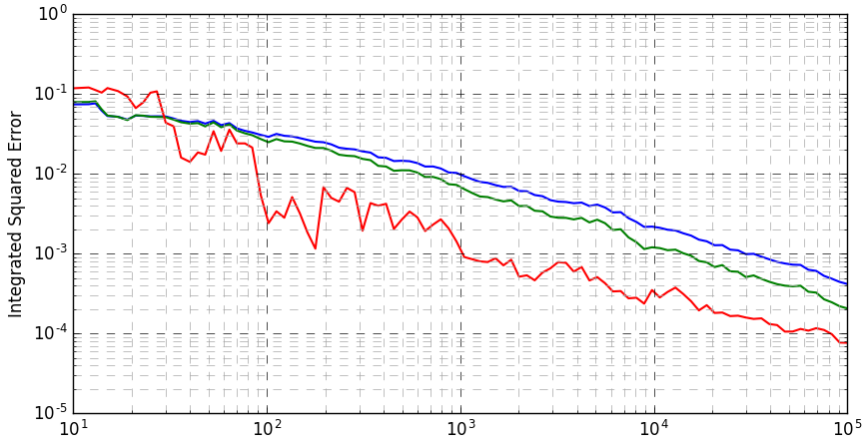
The effect of dimensionality on error was also examined by calculating the ISE and Kullback-Leibler divergence with increasing sample sizes one to three dimensions. Direct comparison between dimensions is complicated by the fact that ISE is not a dimension-free quantity, and it further depends strongly on the scale of the grid being used. Epanechnikov treated this issue by introducing an ISE scaling factor, $\int f^2 d\vec{\boldsymbol{x}}$, to construct a dimensionless quantity for comparison:

$$ISE(\hat{f}) = \frac{\int (\hat{f}(\vec{\boldsymbol{x}}) - f(\vec{\boldsymbol{x}}))^2 d\vec{\boldsymbol{x}}}{\int f(x)^2 d\vec{\boldsymbol{x}}} \tag{4.11}$$

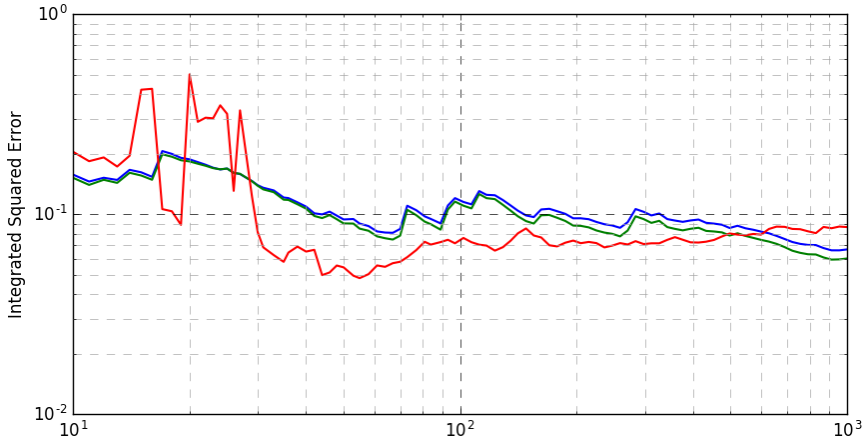
The error tests were performed with a standard Gaussian distribution centered at $x = 0$, as seen



(a) Gaussian



(b) Bimodal Gaussian



(c) Wald

Figure 4.4: Integrated squared error for the univariate density estimates as a function of sample size. The true distributions correspond to those shown in figure 4.2.

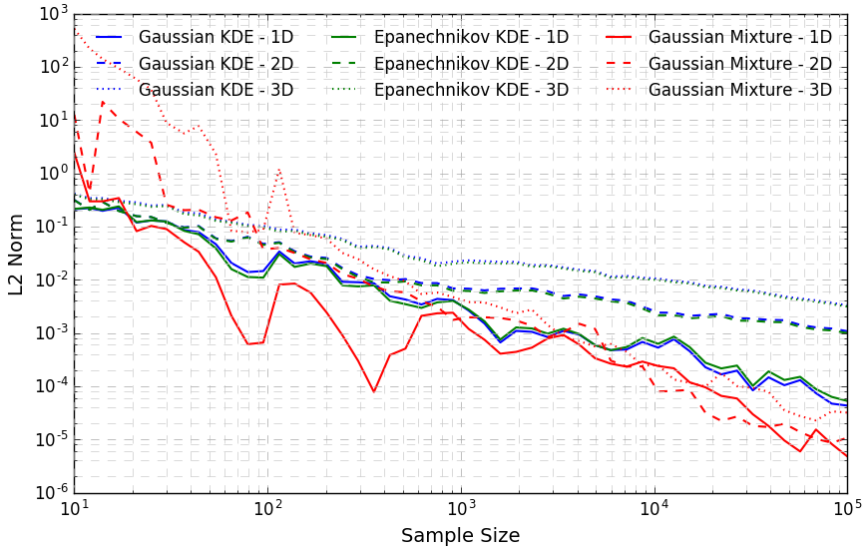
in figures 4.2a and 4.3a. The dimensionless ISE was used, while the Kullback-Leibler divergence was calculated from equation 4.10. The grid sizes were kept uniform across the three tests: $N_x = 60$ one dimension, $N_x, N_y = 60$ in two dimensions, and $N_x, N_y, N_z = 60$ in three dimensions. The results are reported in figure 4.5.

Increasing the dimensionality of the problem both increases amount of error in the system and decreases the rate of convergence. For example, increasing the sample size from 10 to 100,000 in the one-dimensional Gaussian KDE case decreases the error from 20% to 0.008% - a change of over three orders of magnitude. In three dimensions, the same density estimate ISE decreases from 30% to 0.3% only. The Gaussian Mixture model, which is well-suited to fitting a unimodal Gaussian, has a less drastic decrease in convergence rate. However, while noticably better with large sample sizes, the Gaussian mixture performs worse in all three dimensions at a low sample size than the KDE methods; it only performs better than the KDE estimates when the sample sizes is greater than 15, 30, and 100 in one, two, and three dimensions, respectively. The Kullback-Leibler divergence calculations show a similar trend to the ISE.

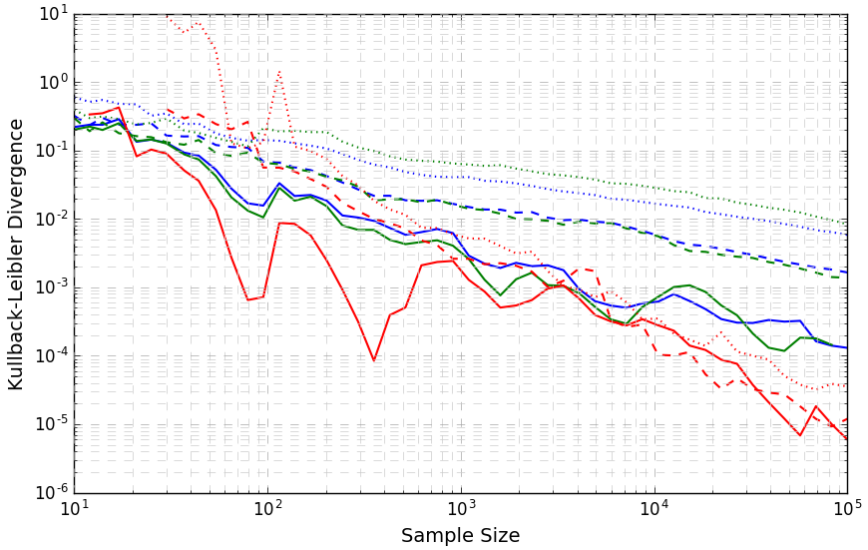
4.2.3 Computational Considerations

Density estimation must be performed frequently to construct sputtered and reflected velocity distributions in order to couple TRIDYN to the Boltzmann plasma model, and therefore the computational efficiency of each algorithm is of practical importance. The above error tests also recorded the computational speed of each density estimate as a function of sample size, shown in figure 4.6. The elapsed time for density estimates increases with both the number of samples and the dimensionality of the problem, with approximately an order of magnitude increase per dimension. Notably, the cost of the GMM model only marginally increases as the number of dimensions increase.

The Gaussian kernel is two orders of magnitude slower than both the Epanechnikov kernel and the GMM estimate at a sample size of 10^5. This is due in large part to the presence of an exponential factor (see table 4.2), causing each call to the Gaussian kernel to cost more CPU time than the Epanechnikov kernel.

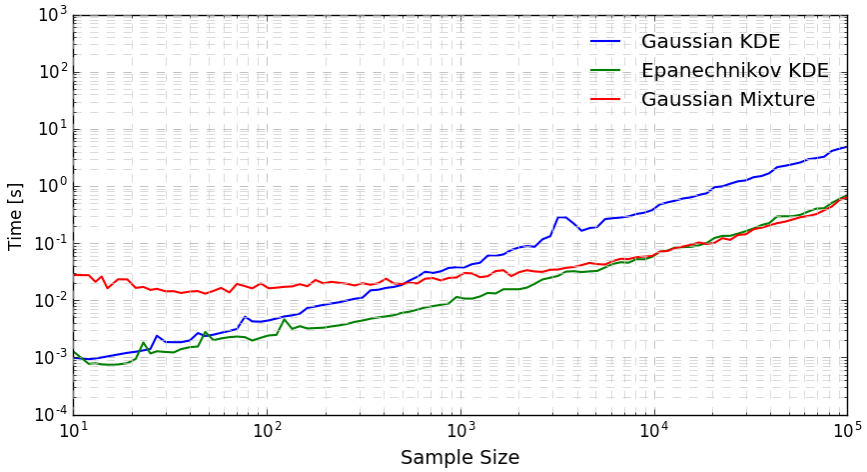


(a) L2 Norm (Integrated Squared Error)

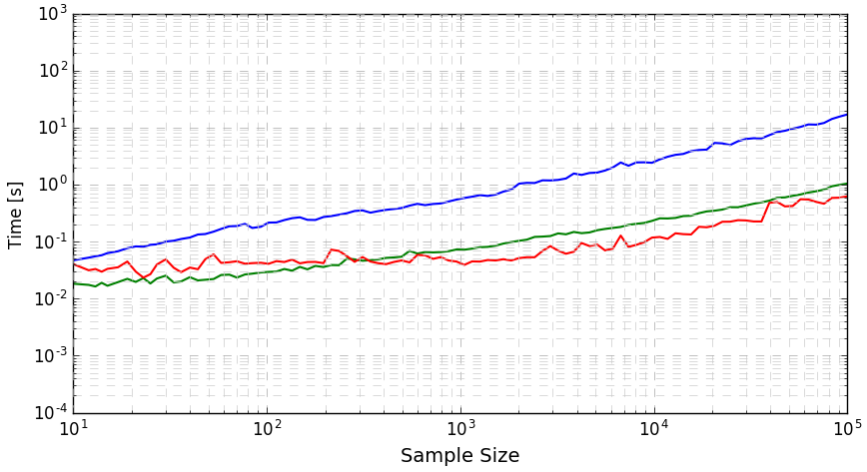


(b) Kullback-Leibler Divergence

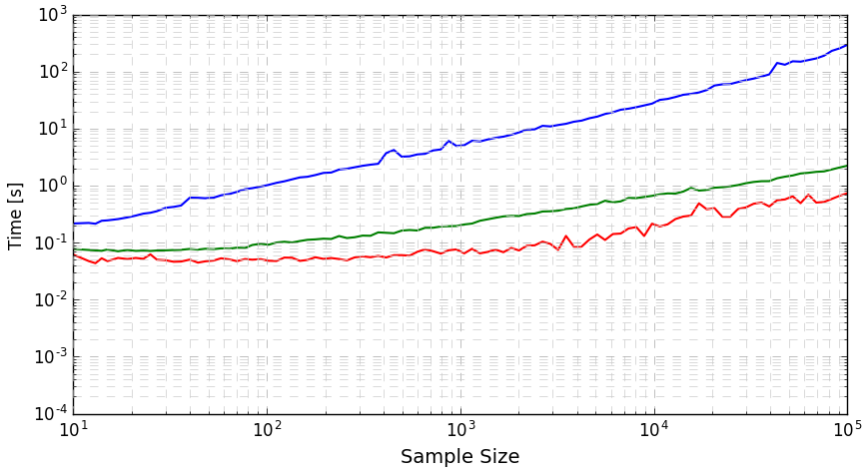
Figure 4.5: Integrated squared error (a) and Kullback-Leibler divergence (b) for the density estimates as a function of sample size. Samples were drawn and estimates were fitted in one through three dimensions. The solid lines are the one-dimensional calculations, the dashed lines are two-dimensional, and the dotted lines are three-dimensional.



(a) One Dimensional



(b) Two Dimensional



(c) Three dimensional

Figure 4.6: Computational time for the density estimates as a function of sample size. Samples were drawn from the standard Gaussian distribution in one (a), two (b), and three (b) dimensions.

4.3 Application to TRIDYN

4.3.1 TRIDYN Surface Model

The specific BCA model chosen to model the surface was Fractal-TRIDYN [80, 18], a modified version of TRIDYN that includes updated output files and a fractal description of the surface, allowing for possible surface morphology affects to be described. While the fractal functionality is possible, it was not used in the present work. TRIDYN is a frequently used tool for modeling sputtering yields and distributions [81, 82, 83]. The sputtering model follows the BCA method described in 2.3.2: ion bombardment is described by a sequence of binary collision events, with both the initial ion and recoil atoms being tracked. Ions that do not have enough energy to penetrate the surface are reflected. Recoil atoms have a starting energy of $T - E_b$, where T is the nuclear energy transfer and E_b is the bulk binding energy, with the condition that $T < E_d$, where E_d is the displacement threshold energy of the surface atoms. A particle is "stopped" in the bulk if its energy falls below the cutoff energy E_c. If a recoil atom gains enough energy through a collision to overcome the surface binding energy and move past the position that designates the surface, it is considered sputtered. This is true of both implanted ions and surface atoms, which allows recycling to be analyzed alongside sputtering.

As described in section 2.3, surface response to ion bombardment is crucially dependent on the interaction potential. TRIDYN uses the screened Coulomb potential:

$$V(R) = \frac{Z_a Z_b e^2}{4\pi\epsilon_0 R}\phi(\tfrac{R}{a}) \tag{4.12}$$

With the Kr-C screening function:

$$\phi(\xi) = 0.191e^{-0.279\xi} + 0.474e^{-0.637\xi} + 0.335e^{-1.919\xi} \tag{4.13}$$

where $\xi = R/a$. In this case, a is the screening distance defined by Firsov, equation 2.36. With this description the distance of closest approch for two interacting particles can be defined as R_c:

$$1 - \frac{V(R_c)}{E_c} - \left(\frac{p}{R_c}\right)^2 = 0 \tag{4.14}$$

E_c is the center of mass energy, $\frac{m_b}{m_a+m_b}E$.

Inelastic energy loss is treated as an energy loss term separate from the collisions, which aligns with the method of splitting stopping power into nuclear and electronic stopping outlined previously. Specifically, the electronic stopping cross section is modeled with Lindhard and Scharff's approach, Eq. 2.19. TRIDYN further splits electronic stopping into local and nonlocal energy loss. The former is considered the frictional force that slows down particles, with the energy loss defined as

$$\Delta E_{nl} = (\lambda - t) n S_{el} \tag{4.15}$$

where λ is the mean free path approximated as a function of density, $\lambda = n^{-1/3}$ and t is the asymptotic deflection point. Local energy loss in TRIDYN is calculated using the formula developed by Oen and Robinson [84].

As output, TRIDYN provides a discrete sample of sputtered and reflected particle energies and angles with respect to the (v_x, v_y, v_z) axes. The discrete data set must be converted into a continuous three-dimensional velocity distribution to be coupled to the Boltzmann plasma solver. As an example, density estimation is performed in the following section on sputtered particle data calculated by TRIDYN, which was run with He incident on W at an energy of $1keV$. For the purposes of the present work, the x-direction is oriented normal to the surface. Note that the PDF is therefore restricted to $v_x > 0$ since sputtered particles are necessarily directed away from the surface.

4.3.2 Method Selection Criteria

In section 4.2, the most suitable density estimate for a standard Gaussian was chosen based on the exact ISE, the Kullback-Leibler divergence, and computational time. However, calculating the exact error with equation 4.9 requires knowledge of the underlying distribution function. An exact distribution of sputtered and reflected particles is unknown, making the same criteria impossible to apply to the density estimation of TRIDYN's output. Instead, a purely data-based estimation of $MISE$ may be calculated with bootstrapping [85]. An optimal model of estimating $f(\vec{v})$ will thus be chosen based on two parameters: (1) bootstrapped $MISE$ and (2) the computational considerations shown previously.

Bootstrapped MISE

The bootstrapping method is a technique which estimates desired properties of a sample by replacing the true value with its estimate. The original random sampling process is imitated by drawing B bootstrapped samples from the original n samples $(X_1, X_2, ...X_n)$ with replacement and constructing bootstrapped density estimates $\hat{f}_B^*$. By the law of large numbers, as $B \to \infty$, $B^{-1}\sum_{j=1}^{B} \hat{f}_j^* \to \hat{f}$, and similarly $\hat{f} \to f$ as $n \to \infty$. For proof of the convergence of bootstrapping methods, the reader is referred to [86, 87].

The process for calculating bootstrapped $MISE$ as described by Faraway and Jhun [88] is as follows:

1. Generate n random samples, $X_1, X_2, ...X_n$ from the known distribution or physical process that must be estimated
2. Estimate the density function of the samples, $\hat{f}$
3. Draw B bootstrap samples $(X_1^*, X_2^*, ...X_n^*)$ from $(X_1, X_2, ...X_n)$ with replacement
4. Generate B bootstrap estimates $\hat{f}_B^*$ from $(X_1^*, X_2^*, ...X_n^*)$
5. Define the bootstrapped $MISE$, denoted $\widehat{MISE}$, as:

$$\widehat{MISE} = \frac{\sum_{j=1}^{B} \int (\hat{f}_j^*(\vec{\boldsymbol{x}}) - \hat{f}(\vec{\boldsymbol{x}}))^2 d\vec{\boldsymbol{x}}}{B \int \hat{f}(\vec{\boldsymbol{x}})^2 d\vec{\boldsymbol{x}}} \tag{4.16}$$

   where the integral in the denominator is the normalization factor (see section 4.3).

If the density estimates used to construct $\hat{f}$ and $\hat{f}_B^*$ are convergent, the estimate of $\widehat{MISE} \to MISE$, and an estimate of the error may therefore be calculated without any knowledge of the true distribution from which the original samples were drawn.

Figure 4.7 compares the ISE as calculated in section 4.3 to the bootstrapped $\widehat{MISE}$ with 20 bootstrap samples drawn from the original samples. Even with a small number of bootstraps, the $\widehat{MISE}$ calculation shows good agreement with the exact ISE. Note that the calculation was sufficient with a small number of bootstrap samples because the true distribution was a Gaussian,

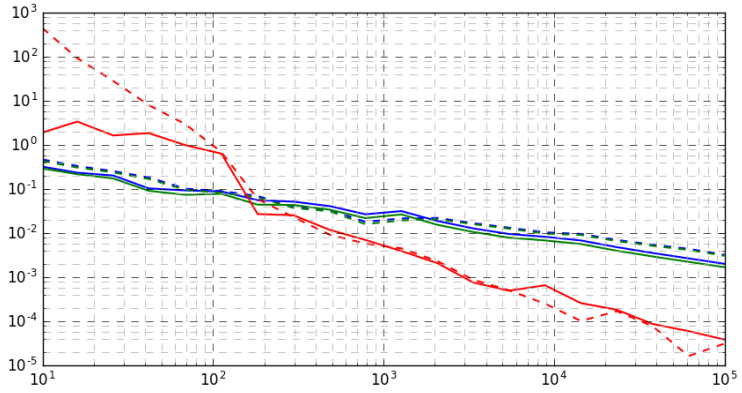


Figure 4.7: Integrated squared error compared to the bootstrapped mean integrated squared error in three dimensions. The solid line is the $\widehat{MISE}$ estimate, and the dotted lines are the ISE shown in figure 4.5.

and a larger bootstrap sample may be needed to ensure that the calculation converges for a more complicated or unknown distribution.

4.3.3 TRIDYN Density Estimate Error Calculation

In order to select an optimal method for velocity distribution estimation, density estimates were applied to TRIDYN velocity data and the $\widehat{MISE}$ was calculated as a function of sample size. Particle sputtering was simulated in TRIDYN utilizing He incident on W at a 60 degree angle of incidence. The output file contained approximately 40,000 sputtered particles. Sample sizes ranging from 10 to 10,000 were drawn from the sputtered data to calculate $\widehat{MISE}$, and density estimates were performed using kernel densities and the Gaussian mixture model. As the output of TRIDYN is smooth and unimodal, 50 bootstrapped samples were considered sufficient for the calculation. A velocity grid ranging from $(-8000, 8000)m/s$ was used in each dimension, with a grid size of $N_x, N_y, N_z = 60$. There is no restriction on the grid size for density estimation; however, as with the finite difference methods shown in Chapter 3, increasing the grid size incurs a drastic increase in computational cost.

The Gaussian mixture model is constrained to using a finite sum of Gaussian distributions, and the limitations of this approach are evident in figure 4.8. The large error associated with small sample sizes in the Gaussian mixture can be attributed to the presence of both a tail in the data and skewness in the V_x-direction, both of which deviate from Gaussian distributions. The

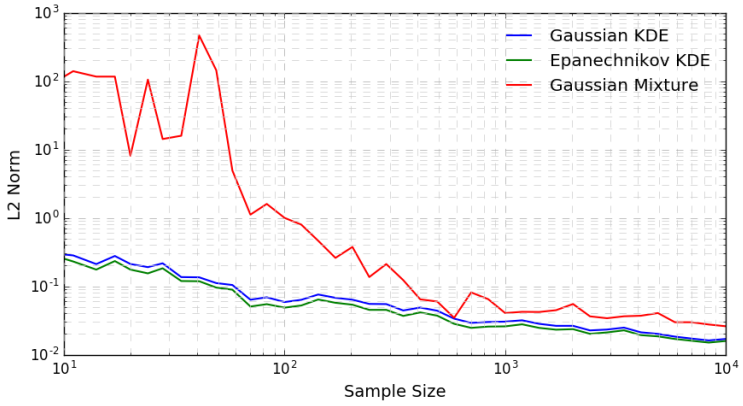


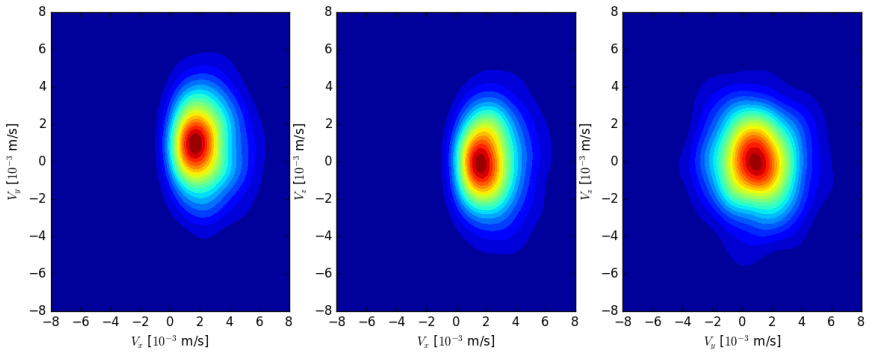
Figure 4.8: $\widehat{MISE}$ calculated with TRIDYN sputtering data of He incident on W with a 60 degree angle of incidence. Sample sizes range from 10 to 10,000.

KDE methods do not rely on a parametric fit and can resolve these features with a smaller sample relative to the mixture model.

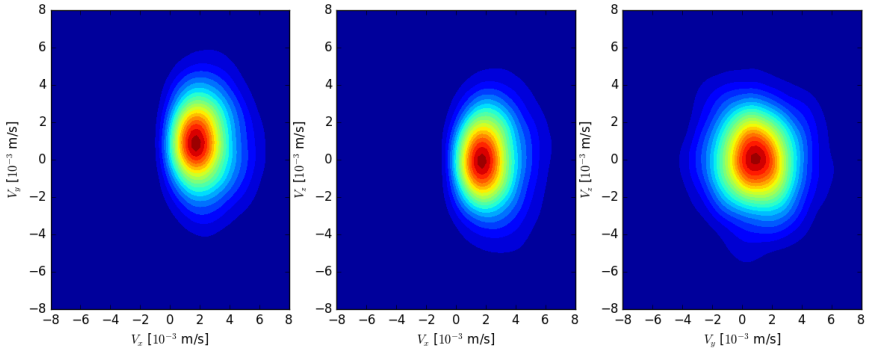
4.3.4 Method Selection

The results shown in figure 4.8 along with sections 4.4.3 and 4.4.4 provide constraints on the choice of density estimator. In order to constrain the error to a minimum of 10%, a sample size of 100 is needed for the KDE methods, while the Gaussian Mixture requires between 500 and 1000. Slices of the estimated velocity distributions using each method using 1000 samples are shown in figure 4.9. However, even at this level of error, the `GMM` approach diverges in the location of the peak and the spread of the data from the `KernelDensity` examples, which are almost indistinguishable from each other.

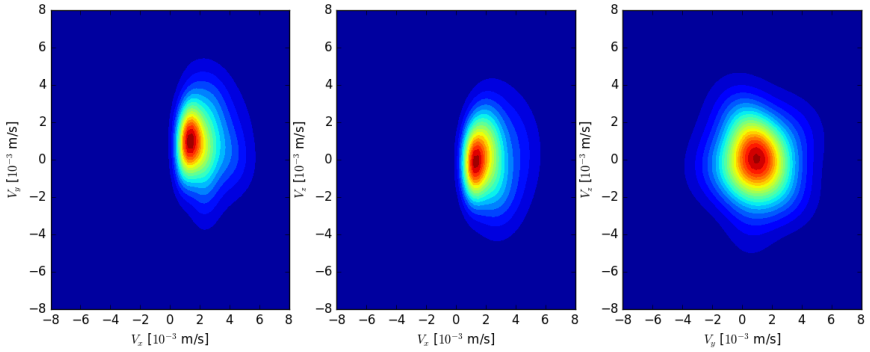
Computational considerations also play a role in selecting a density estimate. Figure 4.6 shows that the Gaussian kernel in `KernelDensity` is an order of magnitude slower than both the Epanechnikov kernel and the `GMM` model starting at a sample size of 100, and the discrepancy only grows larger as n increases. Since the error between the two kernels is negligible, the Gaussian kernel is considered a weaker choice. Computational cost of the Epanechnikov kernel and `GMM` vary by a maximum of a factor of four; however, note that the cost of `GMM` depends on how many mixtures were attempted. If the number of Gaussians needed to calculate the density estimate is known, as was the case in figure 4.6, only one call to `GMM` is necessary. In the case of estimating TRIDYN's output the number of Gaussians needed is unknown. The routine was optimized by test-



(a) Gaussian KDE



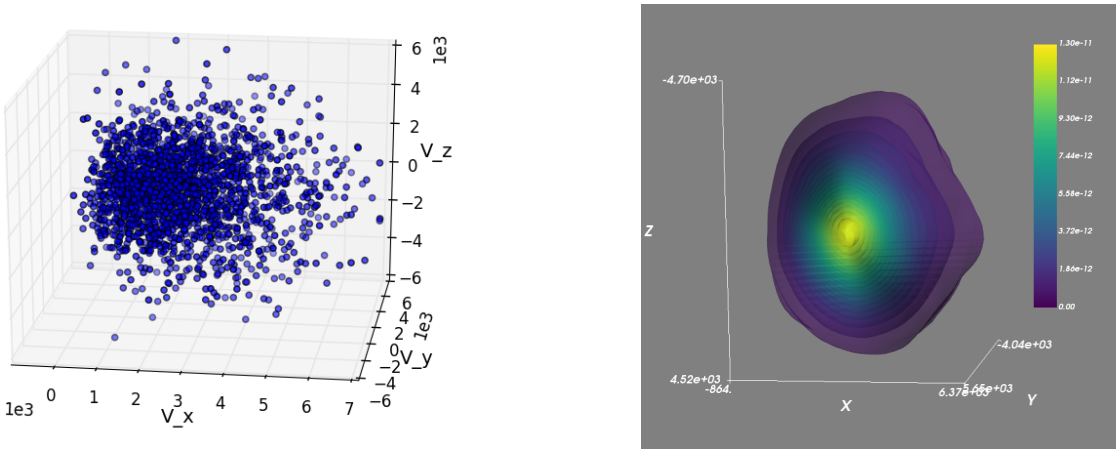
(b) Epanechnikov KDE



(c) Gaussian Mixture

Figure 4.9: Two dimensional slices of the density estimated velocity distributions with (a) a Gaussian KDE, (b) an Epanechnikov KDE, and (c) a Gaussian Mixture model. The left column shows V_x-V_y data (X-Y axes), the middle column is the V_x-V_z slice, and the rightmost column is the V_y-V_z slice. A sample size of 1000 was used.

ing eight separate mixtures, increasing the computational time proportionally. The Epanechnikov `KernelDensity` routine and `GMM` are therefore considered approximately equal in terms of computational cost. Since the Epanechnikov kernel converges faster with sample size than `GMM` and is computationally faster than the Gaussian kernel, an Epanechnikov KDE using the `KernelDensity` library is considered the optimal estimator for the purposes of this work. Figure 4.10b shows one example of three-dimensional output from TRIDYN using the Epanechnikov kernel, along with the discrete data in figure 4.10a as comparison.



(a) Raw TRIDYN Output

(b) Kernel Density Estimate

Figure 4.10: Example of three-dimensional density estimation. TRIDYN provides a finite set of particles (a), which is used to construct a continuous PDF estimate (b). The density estimation was performed with the `KernelDensity` model using an Epanechnikov kernel. Mayavi was used to visualize 3D data in (b) [89].

4.4 Boltzmann-BCA Coupling

Coupling of the Boltzmann plasma model with a model of the material surface must solve two issues: (1) the domain size matching, and (2) communication. The physical domain of the material module spans over the interatomic distance (angstrom) to a maximum size of the order of the particle range (less than a micrometer), which is still smaller than the size of one cell of the Boltzmann solver. Hence, the BCA code effectively acts as an independent model for the calculation of the boundary conditions of the plasma.

In the present model, charged species (electrons, plasma ions, ionized material returning toward

the wall) are always charge-neutralized upon contact with the wall. The neutralization process occurs at spatial scales depending on the surface binding energy, typically of the order of a few nanometers, which is much less than the characteristic size of the Debye sheath and magnetic presheath. As a consequence, a grounded metallic wall may be assumed to behave as an ideal charge-neutralizer for all charged species:

$$f_{\pm}(t, x, \boldsymbol{v}) = 0 \quad : \quad x < 0, x > L \tag{4.17}$$

where the subscript $\pm$ indicates the distribution of charged species. All the charged species (excepting electrons) coming in contact with the metallic wall and returning back to the plasma are therefore turned into the corresponding neutral species.

In addition to sputtering, a portion of incident particles will lack sufficient energy to penetrate the surface and will instead be reflected. The reflection process modifies both the particle number and the energy distribution of the population. Reflection is conventionally quantified by two constant coefficients, the particle reflection coefficient and the energy reflection coefficient. However, the reflection process depends on the energy of the impacting particles, and should be described with energy-dependent distribution rather than constant coefficients. Reflection is calculated on the fly by the BCA module, and the proposed model therefore retains the desired energy-dependent behavior.

4.4.1 TRIDYN Input from Boltzmann Model

TRIDYN is coupled to the Boltzmann model by using the velocity distribution of particles lost to the wall at each timestep as the distribution of particles incident on the surface. As velocity distributions are calculated in the present Boltzmann model, a conversion from $f(\vec{\boldsymbol{v}}) \rightarrow f(E, \theta)$ is performed.

At each timestep, the distribution of particles leaving the plasma volume and absorbed by the wall as calculated by the Boltzmann model is passed to the BCA code, Fractal-TRIDYN. The code runs a number of Monte Carlo particle histories (N_h) large enough to have convergence on the distributions of the backscattered species. TRIDYN includes a built-in error estimate based on the fluence of particles and N_h, with suggested values ranging from $10^4 - 10^6$. A low N_h causes multiple

problems in TRIDYN depending on the sputtering yield: if a material has a high SY, each incident particle can erode a large portion of the simulated surface and an error flag is triggered. For the purposes of coupling to the Boltzmann model, however, the number N_h of depends not only on the numerics but also the physics: a material with a low SY and low N_h may result in a simulated sputtering yield of zero, which is a misleading result: If the incident particle density is large, the number of sputtered particles even at low sputtering yields can be substantial. A simulation with consistently low sputtering yields and a small N_h will suggest that there are no impurities in the system.

The results of section 4.3 also place restrictions on the number of particle histories TRIDYN needs to run in order to achieve a valid sample size. In order for at least 1000 particle samples to be available, the number of histories N_h is defined as $N_h = 1000/SY$, where SY is the sputtering yield. Materials with low sputtering yields require more particles and thus a longer simulation in order to obtain a sample large enough for a valid density estimate.

4.4.2 Boltzmann Input from TRIDYN

With the discrete sample of particles provided by the BCA calculation, density estimates of the reflected and sputtered particles ($\hat{f}(\vec{\boldsymbol{v}})_s$ and $\hat{f}(\vec{\boldsymbol{v}})_r$, respectively) are constructed using the `KernelDensity` module of the `scikit-learn` library in Python. Per section 4.4, the Epanechnikov kernel was used in future calculations. Once the estimated $\hat{f}(\vec{\boldsymbol{v}})$ is constructed, it will be in the form of a PDF with a total density of 1. To ensure that the proper density of particles is being sputtered and reflected, the output density $n_{s,out}$ may be calculated from the definition of sputtering yield: $n_{s,out} = SY \times n_{in}$, where n_{in} is calculated from the distribution used as input to TRIDYN with equation 2.5.

$$\begin{aligned} n_{s,out} &= SY \times n_{in} \\ f(\vec{\boldsymbol{v}})_{s,out} &= \hat{f}(\vec{\boldsymbol{v}})_s \times n_{s,out} \end{aligned} \tag{4.18}$$

The final velocity distribution estimate $f(\vec{\boldsymbol{v}})_{s,out}$ is then passed as a boundary condition to the Boltzmann plasma model at the location of the wall. The reflected velocity distributions are treated identically, with SY being replaced with the reflected yield, RY.

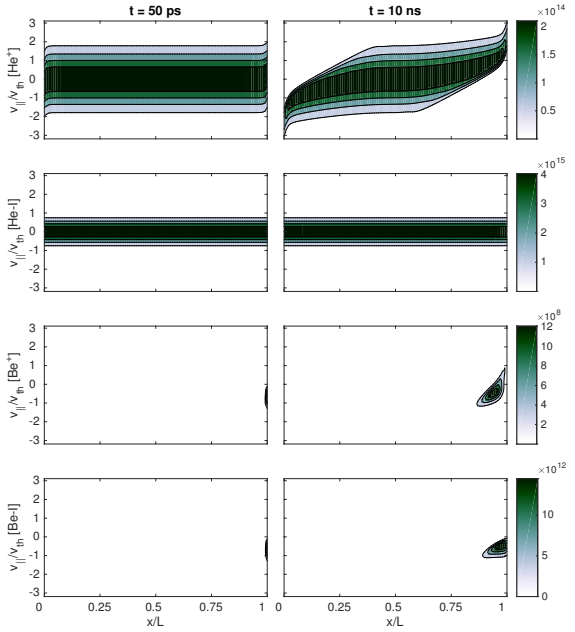
4.5 Coupled Boltzmann-BCA Simulations of Plasma-Material Interactions

Here the methodology presented in the previous sections is applied to two cases: section 4.5.1 details the study of microscopic erosion of a beryllium wall exposed to an ITER-grade helium plasma at typical divertor conditions in unmagnetized (1D1V) conditions, and section 4.5.2 extends the model into 1D3V to treat magnetic fields in DIII-D divertor conditions. The interest in simulating microscopic erosion of beryllium walls exposed to high-density, high-temperature plasmas is multi-faceted. On one side, beryllium is the candidate first-wall material of ITER. On the other side, due to the toxicity of its nano-dust, beryllium can be experimentally simulated at ITER-relevant conditions at only a few facilities in the world, which makes numerical studies of beryllium erosion highly relevant. In addition, this metal can retain a large fraction of the implanted gas (up to 35-40%), which poses significant challenges not only for fuel retention, but also in understanding the erosion mechanisms themselves since the composition of the surface layers dynamically changes over time during plasma exposure. The plasma ions implanting below the surface are trapped within the first surface layers, modifying the local composition and morphology and ultimately affecting the erosion dynamics.

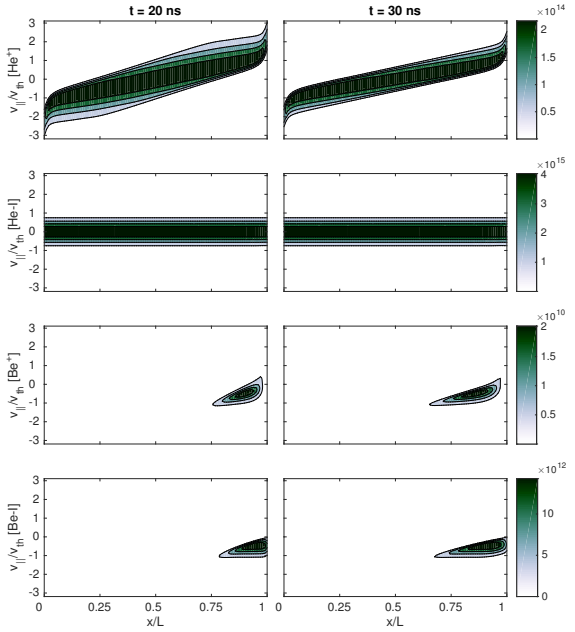
The two coupled codes, namely the Boltzmann solver and Fractal-TRIDYN, allow for a kinetic treatment of the interaction, including dynamic composition changes of the wall. In order to avoid the additional complexity of simulating the full SOL, the analysis is limited to the simulation of the near-wall region (∼1 mm) in both cases. Plasma sheath formation and depletion between two electrically grounded walls is simulated including four species: He^+ (He ions), He-I (He neutrals), Be^+ (Be ions), Be-I (Be neutrals). The plasma is allowed to propagate within a one-dimensional, L = 1 mm domain in physical space and within $\pm 5 v_{th}$ in the velocity space, where v_{th} is the thermal velocity of each species: $v_{th} = \left(\frac{k_b T_s}{M_s}\right)^{\frac{1}{2}}$. The values T_s and M_s are the temperature and mass, respectively, of the species being considered.

4.5.1 1D1V Unmagnetized Sheath

Helium ions begin with a Maxwellian distribution at $n_{He^+} = 10^{19}$ m^{-3} and $T_{He^+} = 10$ eV; helium neutrals are also starting with Maxwellian distribution at same density, $n_{He-I} = 10^{19}$ m^{-3}, and room temperature $T_{He-I} = 300$ K. The distributions of beryllium Be^+ and Be-I are initially set equal to zero: they evolve during the simulation as determined by the Boltzmann-BCA model. The simulation evolves for 40 nanoseconds with a timestep of 10 picoseconds. No perpendicular flux of helium ions from the core region is assumed, so the plasma is not replenished in this simulation. The only volumetric source generating new plasma ions is electron-impact ionization, which is however slower than the rate at which He ions leave the domain; as expected, the total particle density decreases over time.



(a) 50 ps (1st column) and 10 ns (2nd column).



(b) 20 ns (third column) and 30 ns (fourth column)

Figure 4.11: Phase space plots of each species. From the top: He ions, He neutrals, Be ions, and Be neutrals.

The beryllium wall is set only on the right boundary of the domain. The left wall is a classical perfectly-absorbing wall with no sputtering activity. During the simulation, the distribution functions of all the species impacting on the beryllium wall are passed to Fractal-TRIDYN.

The code solves for the distribution functions of the backscattered species, and it also updates

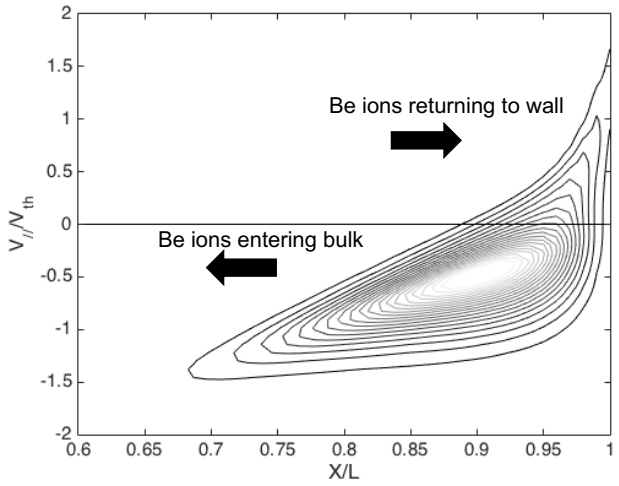


Figure 4.12: Detail of phase space near the material wall. A portion of the impurity distribution is ionized close enough to the wall to be pulled back by the sheath.

the material composition. The sputtered material –both beryllium and previously-implanted helium gas– enters the plasma domain in neutral state. The material is subsequently ionized and begins to interact with the sheath/presheath region of the plasma. A fraction of ionized particles at higher energy is able to overcome the sheath potential and enters into the plasma bulk. The fraction at lower energy is slowed down first, and then accelerated back to the wall by the sheath potential. The fraction of material ions returning to the wall contributes to self-sputtering.

Figures 4.11a and 4.11b show the phase spaces of each of the four species (He^+, He-I, Be^+, Be-I) at time t = 50ps, 10ns, 20ns, 30ns. In the presence of a grounded wall, the more mobile species (electrons) will reach the wall first and create a negative bias, causing the positively-charged ions to accelerate in the near-wall region. This transient of formation of the plasma sheath can be seen from the acceleration of helium ions at the left and right walls in the phase space plots (He^+, top row of each figure). Helium neutrals (He-I) are colder, so they do not move appreciably during the simulated ion time scale. Ionization acts as the only sink of He neutrals, while the sputtering of implanted He is a source. At time t = 50ps (early stage of the simulation), beryllium can be seen to start sputtering from the wall and a small amount is ionizing.

As time progresses, the Be ions have begun forming and drifting away from the wall while some of the slower ions that formed near the sheath are decelerated and pulled back to the wall, as illustrated by the detail of the phase space reported in Fig. 4.12. The majority of the sputtered ions

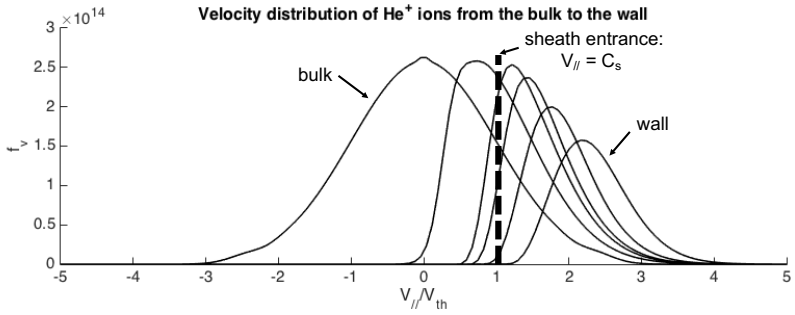
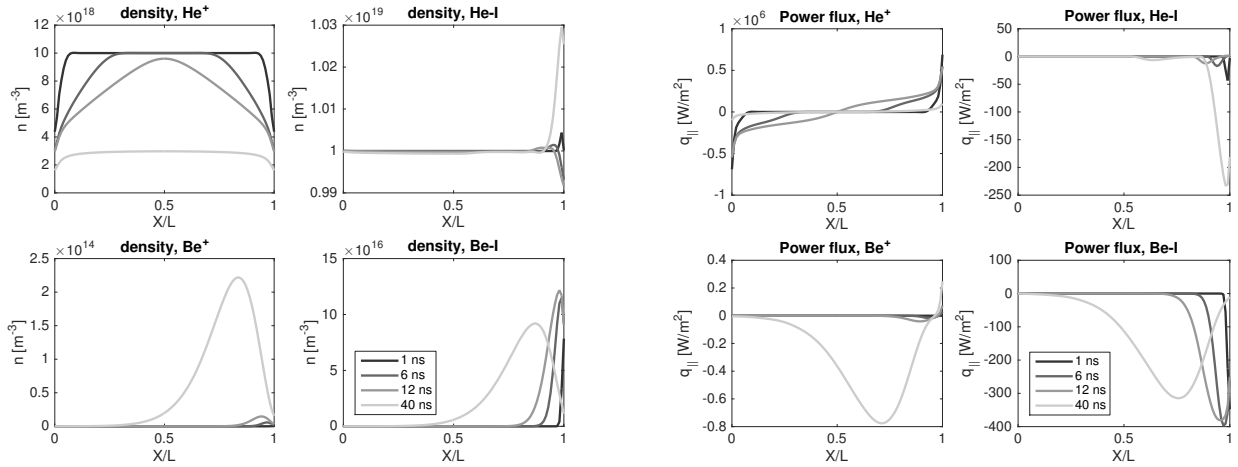


Figure 4.13: Velocity distribution evolution of He^+ from the plasma bulk to the material wall.

are created outside the sheath, allowing them to propagate into the plasma bulk almost undisturbed. The evolution of the velocity distribution of He^+ from the plasma bulk to the material wall is shown in Fig. 4.13. The small peak at the center of the bulk distribution occurs because of ionization: The He neutrals are at room temperature, so the He-I distribution is a narrow portion of the He^+ distribution.



(a) Density (b) Energy flux

Figure 4.14: (a) Density and (b) Energy flux over time for each species.

Fig. 4.14a presents the densities of the species. The He neutrals are shown to increase in density over time, which is caused by the implantation and subsequent sputtering of He from the wall. The only sink of He neutrals is ionization, and given the fact that neutrals move much more slowly than the ions, an accumulation of He neutrals in proximity of the wall from sputtering was expected. Be neutrals, however, decrease in density over the course of the simulation due to the decrease in sputtering rate. As the He ion density decreases, there are fewer ions impacting the wall and thus

fewer Be atoms being sputtered. In Fig. 4.14b, it is shown that the power flux to the wall reaches a value of 0.7 MW/m^2 during the simulated transient.

4.5.2 1D3V Magnetized Sheath

The same setup was also simulated with a magnetic field in 1D3V. The plasma was started with Maxwellian distributions for He^+ and He-I and no Be species (Be^+ and Be-I) present, and the system was evolved for 30 ns. Figure 4.15 shows the phase spaces of each species after 30 ns in the magnetized case. The material module is again attached only to the wall at $x/L = 1.0$. The fact that particles will sputter with relatively evenly distributed v_y and v_z velocities, along with the fact that TRIDYN does yield sputtered particles with zero velocity, can explain the distinctive shapes of the V_{thy} and V_{thz} distributions.

The effect of the magnetic field is substantial, as seen by comparing Fig. 4.15 to Fig. 4.11b. Most of the $\boldsymbol{v} \times B$ acceleration occurs in the v_z dimension since B is almost parallel to the wall at 86^o. Additionally, the velocity of the particles incident on the walls decreases from near $3V_{thx}$ to $2V_{thx}$. The diminished velocity suggests that the sheath decreases in size as the magnetic angle increases, as predicted by Stangeby [90]. There is a larger density of impurity particles in the unmagnetized case than in the magnetized case. Due to the diminished perpendicular velocity of impacting particles, a smaller portion of the distribution will have sufficient energy to be implanted in the surface when compared to the unmagnetized case. The particles that cannot implant will instead be reflected, which accounts for the increased He neutral density in the magnetic simulation.

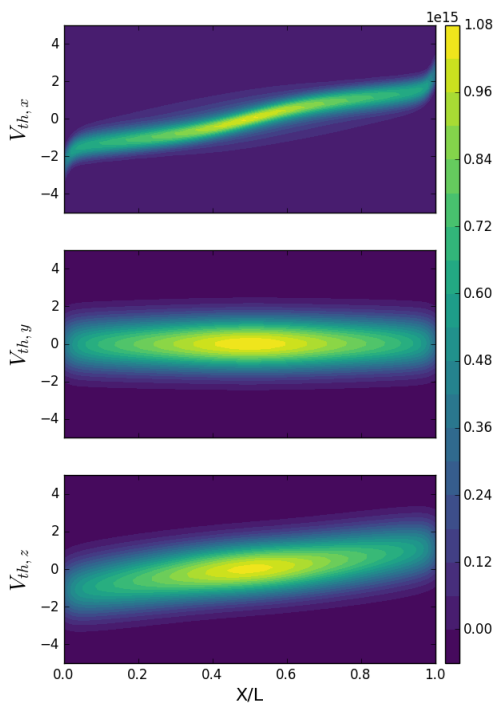
As in the unmagnetized case, Be neutrals begin to appear as the BCA module simulates sputtering, and they are ionzied more frequently as they pass from the sheath region to the plasma bulk. The magnetic field's effects are also visible in the Be ion distribution - since He ions are impacting the wall with lower energy, the sputtering yield decreases and fewer Be neutrals enter the plasma. The integral of the distribution of Be ions is thus smaller in the magnetized case. Both Be ions and neutrals are symmetric in the V_{thy} and V_{thz} dimensions, although a slight acceleration in v_z of the Be ions due to the magnetic field can be seen the magnetized case.

Wall Erosion

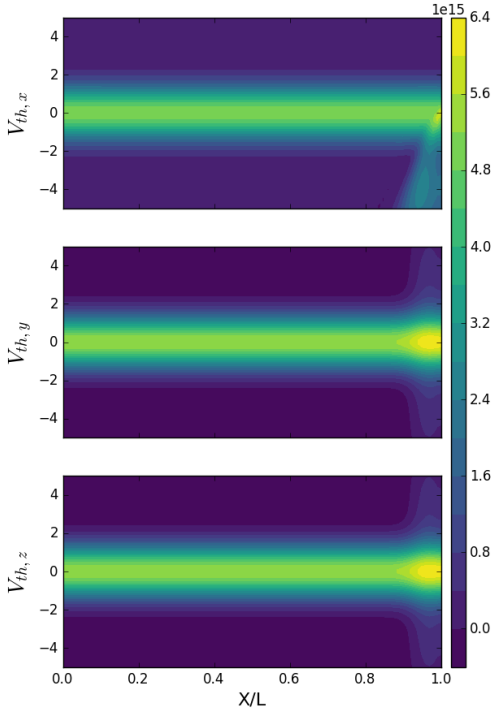
In addition to the phase spaces, a description of erosion was also calculated from the velocity distributions. The deceleration of ions in the v_x direction predictably causes the particles to impact the wall with less energy. This is quantified by the power flux, q_x, which are shown in Figs. 4.16a and 4.16b for He ions and Be ions, respectively. At the wall ($x/L = 1.0$), He ions have a power flux of approximately 2.5 MW/m^2 in the unmagnetized case, but is reduced to 1.25 MW/m^2 in the magnetized case. Similarly, the Be ion power flux at the wall decreases from 2 to almost 0 W/m^2; the negligible fluxes in this case are simply due to the fact that most Be ions are forming outside the sheath and are not decelerated sufficiently to be pulled back.

4.5.3 Computing Times

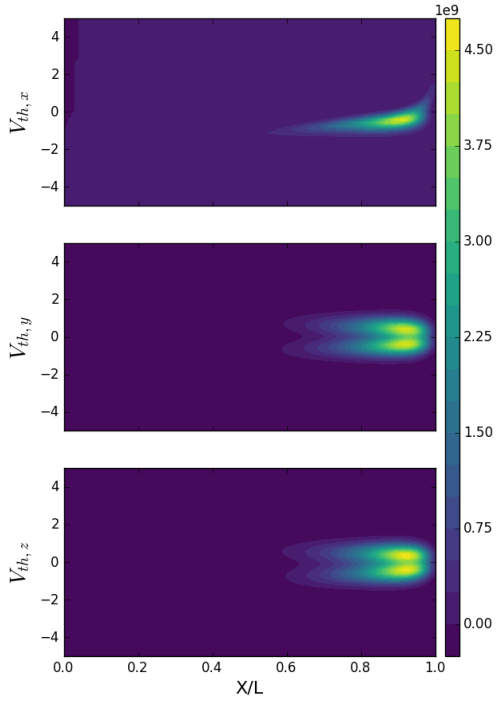
The plasma model for both 1D1V and 1D3V cases was implemented in Python, with advection steps fully vectorized using `numpy`. Fractal-TRIDYN is implemented in Fortran90. Both models were run on a single core of an Intel i5 2.7 GHz CPU in the present work. Each timestep of the plasma model required 0.02 seconds in 1D1V ($N_x = 128$, $N_{v,x} = 128$) and 6 seconds in 1D3V ($N_x = 100$, $N_{v,x}, N_{v,y}, N_{v,z} = 80$) on average. With 2.5×10^4 particle histories the TRIDYN simulation utilizes 3 seconds of computational time. Kernel density estimation using the Epanechnikov kernel averaged around 0.01 seconds in one dimension and 0.2 seconds in three dimensions, with small variations due to the available sample size. The total computational time of the simulations presented in sections 4.5.1 and 4.5.2 was 4 hours and 13 hours, respectively.



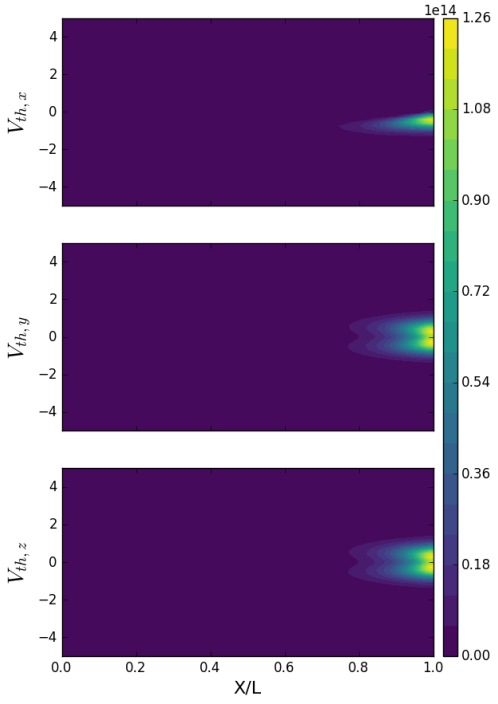
(a) He Ions



(b) He Neutrals

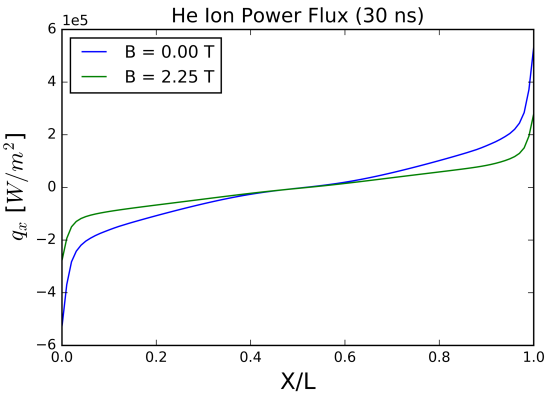


(c) Be Ions

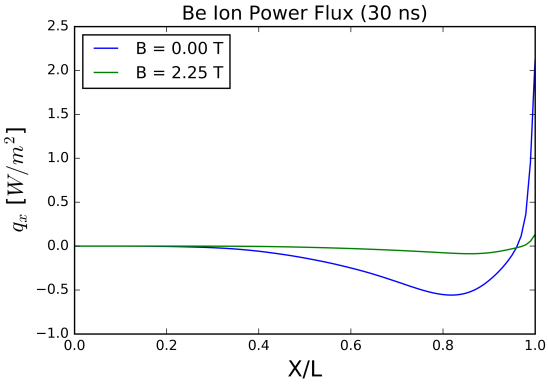


(d) Be Neutrals

Figure 4.15: Phase space of He ions (a), He neutrals (b), Be ions (c), and Be neutrals (d) in a 1D3V magnetized plasma after 30 ns. In each, the top figure is the $X - V_x$ plane, the middle is the $X - V_y$ plane, and the last is the $X - V_z$ plane.



(a) He Ions



(b) Be Ions

Figure 4.16: Power flux to the walls of He ions (left) and Be ions (right).

Chapter 5

Conclusions

Accurate modeling of plasma-material interactions, including the plasma sheath, material response, and impurity transport, is a necessary step toward the development of commercial magnetic fusion devices. This thesis focused on the developed a unified model of the plasma-material interface composed of a kinetic near-wall plasma and a Monte-Carlo treatment of the plasma-facing surface, with the goal of dynamically simulating the interaction of the plasma and impurities with the material surface.

A multi-species, time-dependent, continuum Boltzmann model for the plasma was developed in Chapter 3 based on upwind-biased finite difference schemes and the fourth-order Runge-Kutta algorithm. Stability tests show that error remains uniform in domains with variable flow speeds since upwind schemes are insensitive to different (stable) CFL values, and increasing the order of the numerical method was shown to decrease error by an order of magnitude at a given level of grid refinement. The fourth-order upwind model was used to accurately reproduce classical Landau damping and two-stream instability features found in literature. A simplified collisional plasma sheath model based on the Bhatnagar-Gross-Krook operator, including ionization and charge exchange, was used to verify the convergence of the finite difference methods.

Density estimation was utilized to construct continuous velocity distributions from the discrete data provided by TRIDYN in Chapter 4, facilitating communication between the two codes. Based on bootstraped calculations of the mean integrated squared error, a sample size of $\sim$1000 particles was determined to be sufficient to construct statistically valid velocity distributions, which in turn provides a constraint on the number of particle histories TRIDYN is required to run: $N_h \approx 1000/SY$, where SY is the sputtering yield.

As a proof of concept of the coupling method, an example simulation of the plasma-material interface involving a helium plasma facing a beryllium wall was presented. Both sputtered impuri-

ties and reflected plasma neutrals were shown to enter the plasma through the TRIDYN boundary, and particle densities and power fluxes of each species were calculated. The simulation provides a dynamic model of the the near-wall plasma and the first surface layers, which has the potential to be extended to large-scale problems for the SOL analysis including the material behavior.

5.1 Future Work

The presented method allows for the estimation of the gross and net erosion of a dynamically-evolving material wall exposed to a plasma, but validation is necessary to prove the predictive capabilities of the model. The role of individual collisional processes in plasma-surface interactions will also be analyzed. Charge exchange of both plasma and impurity species is expected to play a major role in surface erosion and redeposition within fusion devices. Additionally, improvements can be made to treatment of collisions; the BGK operator is appropriate for convergence analysis and order of magnitude estimates, but the collision frequency is held constant across velocity space and collisional effects due to velocity-dependent collisions are lost.

The coupled Boltzmann-BCA model itself should be improved in order to be competitive with other kinetic models. While high-order upwind schemes show strong amplitude conservation and low dispersion error, the method has some significant limitations. In particular, even at relatively coarse grids the stability conditions are expensive, requiring the timestep in a physically-relevant plasma sheath simulation to be as small as $O(ps)$. This causes simulation domains larger than $O(cm)$ to require a prohibitive amount of time to solve. In contrast, classical particle-in-cell simulations can reach timesteps of $O(ns)$ and do not suffer from high dimensionality as strongly as continuum methods. One possible improvement includes utilizing an adaptive mesh with a high level of refinement in the sheath and a coarse refinement in the plasma bulk, which would keep computational cost low even in larger simulation domains. High-order finite element and semi-Lagrangian methods have also been applied to Boltzmann plasmas with more competitive computational costs than presented here, and may be an appropriate alternative to finite difference methods to simulate the full scrape-off layer in a tokamak.

Appendix A

Fourth Order Runge-Kutta Amplification Factors

The amplification factors shown below were compiled from Zha and Lingamgunta [53].

A.1 Second Order Upwind Scheme

The full form of the amplification factor is found by following Eq. 3.13 and performing Von Neumann analysis. The real part of the amplification factor is:

$$\begin{aligned} Re(\lambda) = 1 &- \frac{\alpha}{2}(3 - 4\cos\beta + \cos 2\beta) + \frac{\alpha^2}{8}(9 - 24\cos\beta + 22\cos 2\beta - 8\cos 3\beta + \cos 4\beta) \\ &- \frac{\alpha^3}{48}(27 - 108\cos\beta + 171\cos 2\beta - 136\cos 3\beta + 57\cos 4\beta - 12\cos 5\beta + \cos 6\beta) \\ &+ \frac{\alpha^4}{384}(81 - 432\cos\beta + 972\cos 2\beta - 1200\cos 3\beta + 886\cos 4\beta \\ &\qquad - 400\cos 5\beta + 108\cos 6\beta - 16\cos 7\beta + \cos 8\beta) \end{aligned}$$

The imaginary part is:

$$\begin{aligned} Im(\lambda) = &-\frac{\alpha}{2}(4\sin\beta - \sin 2\beta) + \frac{\alpha^2}{8}(24\cos\beta - 22\sin 2\beta + 8\sin 3\beta - \sin 4\beta) \\ &- \frac{\alpha^3}{48}(108\sin\beta - 171\sin 2\beta + 136\sin 3\beta - 57\sin 4\beta - 12\sin 5\beta + \sin 6\beta) \\ &+ \frac{\alpha^4}{384}(432\sin\beta - 972\sin 2\beta + 1200\sin 3\beta - 886\sin 4\beta + 400\sin 5\beta \\ &\qquad - 108\sin 6\beta + 16\sin 7\beta - \sin 8\beta) \end{aligned}$$

A.2 Third Order Upwind Scheme

The amplification factor of the third order upwind scheme is similarly found, as performed by Zha *et al.*. The real part of the amplification factor is:

$$\begin{aligned} Re(\lambda) = 1 &- \frac{\alpha}{6}(3 - 4\cos\beta + \cos 2\beta) + \frac{\alpha^2}{72}(-15 - 20\cos\beta + 46\cos 2\beta - 12\cos 3\beta + \cos 4\beta) \\ &- \frac{\alpha^3}{1296}(-177 + 72\cos\beta + 315\cos 2\beta - 310\cos 3\beta + 117\cos 4\beta - 18\cos 5\beta + \cos 6\beta) \\ &+ \frac{\alpha^4}{31104}(-207 + 1256\cos\beta - 492\cos 2\beta - 2208\cos 3\beta + 2518\cos 4\beta \\ &\qquad - 1072\cos 5\beta + 228\cos 6\beta - 24\cos 7\beta + \cos 8\beta) \end{aligned}$$

The imaginary part is:

$$\begin{aligned} Im(\lambda) = -&\frac{\alpha}{6}(8\sin\beta - \sin 2\beta) + \frac{\alpha^2}{72}(44\cos\beta - 38\sin 2\beta + 12\sin 3\beta - \sin 4\beta) \\ &- \frac{\alpha^3}{1296}(-108\sin\beta - 243\sin 2\beta + 326\sin 3\beta - 117\sin 4\beta + 18\sin 5\beta - \sin 6\beta) \\ &+ \frac{\alpha^4}{31104}(-2488\sin\beta + 540\sin 2\beta + 2400\sin 3\beta - 2486\sin 4\beta \\ &\qquad + 1072\sin 5\beta - 228\sin 6\beta + 24\sin 7\beta - \sin 8\beta) \end{aligned}$$

A.3 Fourth Order Upwind Scheme

$$\begin{aligned} Re(\lambda) = 1 &- \frac{\alpha}{6}(3 - 4\cos\beta + \cos 2\beta) + \frac{\alpha^2}{72}(-15 - 20\cos\beta + 46\cos 2\beta - 12\cos 3\beta + \cos 4\beta) \\ &- \frac{\alpha^3}{1296}(-177 + 72\cos\beta + 315\cos 2\beta - 310\cos 3\beta + 117\cos 4\beta - 18\cos 5\beta + \cos 6\beta) \\ &+ \frac{\alpha^4}{31104}(-207 + 1256\cos\beta - 492\cos 2\beta - 2208\cos 3\beta + 2518\cos 4\beta \\ &\qquad - 1072\cos 5\beta + 228\cos 6\beta - 24\cos 7\beta + \cos 8\beta) \end{aligned}$$

The imaginary part is:

$$
\begin{aligned}
Im(\lambda) = -\frac{\alpha}{6}(8\sin\beta - \sin 2\beta) + \frac{\alpha^2}{72}(44\cos\beta - 38\sin 2\beta + 12\sin 3\beta - \sin 4\beta) \\
- \frac{\alpha^3}{1296}(-108\sin\beta - 243\sin 2\beta + 326\sin 3\beta - 117\sin 4\beta + 18\sin 5\beta - \sin 6\beta) \\
+ \frac{\alpha^4}{31104}(-2488\sin\beta + 540\sin 2\beta + 2400\sin 3\beta - 2486\sin 4\beta \\
+ 1072\sin 5\beta - 228\sin 6\beta + 24\sin 7\beta - \sin 8\beta)
\end{aligned}
$$

References

[1] R.M. Churchill, J.M. Canik, C.S. Chang, R. Hager, A.W. Leonard, R. Maingi, R. Nazikian, and D.P. Stotler. Kinetic simulations of scrape-off layer physics in the DIII-D tokamak. *Nuclear Materials and Energy*, December 2016.

[2] E. Franconi and F. Brossa. Erosion-redeposition and impurity transport in the scrape-off layer of the FT tokamak. *Journal of Nuclear Materials*, 145:691–696, 1987.

[3] BD Wirth, KD Hammond, SI Krasheninnikov, and D Maroudas. Challenges and opportunities of modeling plasma-surface interactions in tungsten using high-performance computing. *Journal of Nuclear Materials*, 463:30–38, 2015.

[4] J. N. Brooks. Modeling of erosion and redeposition processes. Technical report, Argonne National Lab., 1992.

[5] M. A. Dorf, J. A. Hittinger, R. H. Cohen, and T. D. Rognlien. Continuum kinetic modeling of the tokamak plasma edge. *Physics of Plasmas*, 23(5):056102, May 2016.

[6] R. Dejarnac, M. Komm, J. Stockel, and R. Panek. Measurement of plasma flows into tile gaps. *Journal of Nuclear Materials*, 382(1):31–34, November 2008.

[7] Charles K. Birdsall and A. Bruce Langdon. *Plasma physics via computer simulation*. McGraw-Hill, New York, 1985.

[8] Rinat Khaziev and Davide Curreli. Ion energy-angle distribution functions at the plasma-material interface in oblique magnetic fields. *Physics of Plasmas*, 22(4):043503, April 2015.

[9] K J Bowers, B J Albright, L Yin, W Daughton, V Roytershteyn, B Bergen, and T J T Kwan. Advances in petascale kinetic plasma simulation with VPIC and Roadrunner. *Journal of Physics: Conference Series*, 180:012055, July 2009.

[10] C. S. Chang and S. Ku. Particle Simulation of Neoclassical Transport in the Plasma Edge. *Contrib. Plasma Phys.*, 46(7-9):496–503, September 2006.

[11] Petr Cagas, Ammar Hakim, James Juno, and Bhuvana Srinivasan. Continuum Kinetic and Multi-Fluid Simulations of Classical Sheaths. *Physics of Plasmas*, 24(2):022118, February 2017. arXiv: 1610.06529.

[12] Chio-Zong Cheng and Georg Knorr. The integration of the Vlasov equation in configuration space. *Journal of Computational Physics*, 22(3):330–351, 1976.

[13] Gregory J. Parker and Nicholas G. Hitchon. Convected Scheme Simulations of the Electron Distribution Function. *Jpn. J. Appl. Phys.*, 36:4799–4807, July 1997.

[14] Eric Sonnendrucker, Jean Roche, Pierre Bertrand, and Alain Ghizzo. The Semi-Lagrangian Method for the Numerical Resolution of the Vlasov Equation. *J. Comp. Phys.*, 149:201–220, 1999.

[15] Jing-Mei Qiu and Andrew Christlieb. A conservative high order semi-Lagrangian WENO method for the Vlasov equation. *Journal of Computational Physics*, 229(4):1130–1149, 2010.

[16] Francis Filbet and Chang Yang. Mixed semi-Lagrangian/finite difference methods for plasma simulations. *arXiv preprint arXiv:1409.8519*, 2014.

[17] Yasunori Yamamura and Yoshiyuki Mizuno. *Low-energy sputterings with Monte Carlo program ACAT*. Institute of Plasma Physics, Nayoga University, Nagoya, Japan, May 1985.

[18] W. Moller and W. Eckstein. Tridyn A TRIM simulation code including dynamic composition changes. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 2(1-3):814–818, March 1984.

[19] Olivier Coulaud, Patrice Bordat, Pierre Fayon, Vincent Lebris, Isabelle Baraille, and Ross Brown. Extensions of the siesta dft code for simulation of molecules. *arXiv:1302.4617 [cond-mat, physics:physics]*, February 2013. arXiv: 1302.4617.

[20] Jeffrey N. Brooks. Modeling of sputtering erosion/redepositionstatus and implications for fusion design. *Fusion Engineering and Design*, 60(4):515–526, 2002.

[21] T. D. Rognlien and M. E. Rensink. Edge-plasma models and characteristics for magnetic fusion energy devices. *Fusion engineering and design*, 60(4):497–514, 2002.

[22] Ahmed Hassanein. Prediction of material erosion and lifetime during major plasma instabilities in tokamak devices. *Fusion engineering and design*, 60(4):527–546, 2002.

[23] J. N. Brooks, A. Hassanein, A. Koniges, P. S. Krstic, T. D. Rognlien, T. Sizyuk, V. Sizyuk, and D. P. Stotler. Scientific and Computational Challenges in Coupled Plasma Edge/Plasma-Material Interactions for Fusion Tokamaks. *Contrib. Plasma Phys.*, 54(4-6):329–340, June 2014.

[24] A. Kirschner, D. Tskhakaya, G. Kawamura, D. Borodin, S. Brezinsek, R. Ding, Ch. Linsmeier, and J. Romazanov. Modelling of Impurity Transport and PlasmaWall Interaction in Fusion Devices with the ERO Code: Basics of the Code and Examples of Application. *Contrib. Plasma Phys.*, 56(6-8):622–627, August 2016.

[25] Ludwig Boltzmann. Further Studies on the Thermal Equilibrium of Gas Molecules (Reprint). In *The Kinetic Theory of Gases*, volume Volume 1 of *History of Modern Physical Sciences*, pages 262–349. Imperial College Press, July 2003. DOI: 10.1142/9781848161337_0015 DOI: 10.1142/9781848161337_0015.

[26] R.E. Johnson. Atomic and Molecular Collisions. In *Encyclopedia of Physical Science and Technology*, volume 2. Academic Press, Inc., 1987.

[27] A. A. Vlasov. The vibrational properties of an electron gas. *Physics-Uspekhi*, 10(6):721–733, 1968.

[28] Prabhu Lal Bhatnagar, Eugene P. Gross, and Max Krook. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Physical review*, 94(3):511, 1954.

[29] T. J. M. Boyd and J. J. Sanderson. *The Physics of Plasmas*. Cambridge University Press, January 2003. Google-Books-ID: bAmqvuGTUJ4C.

[30] Rainer Behrisch. Introduction and overview. In Dr Rainer Behrisch, editor, *Sputtering by Particle Bombardment I*, number 47 in Topics in Applied Physics, pages 1–8. Springer Berlin Heidelberg, 1981. DOI: 10.1007/3540105212_6.

[31] M. A. Lieberman and Allan J. Lichtenberg. *Principles of plasma discharges and materials processing*. Wiley-Interscience, Hoboken, N.J, 2nd ed edition, 2005. OCLC: ocm56752658.

[32] E Bringa and R.E. Johnson. Ion Interactions with Solids: Astrophysical Applications. In *Solid State Astrochemistry*, pages 357–393. Kluwer Academic Publishers, Netherlands, 2003.

[33] Lucille Giannuzzi, Brenda Prenitzer, and Brian Kempshall. Ion-Solid Interactions.

[34] Wolfgang Eckstein. *Computer Simulation of Ion-Solid Interactions*, volume 10 of *Springer Series in Materials Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1991. DOI: 10.1007/978-3-642-73513-4.

[35] L. A. Christel, J. F. Gibbons, and S. Mylroie. An application of the Boltzmann transport equation to ion range and damage distributions in multilayered targets. *Journal of Applied Physics*, 51(12):6176–6182, December 1980.

[36] J. Lindhard and M. Scharff. Energy dissipation by ions in the keV region. *Physical Review*, 124(1):128, 1961.

[37] Kunihiro Suzuki. Extended LindhardScharfSchiott Theory for Ion Implantation Profiles Expressed with Pearson Function. *Japanese Journal of Applied Physics*, 48(4):046510, April 2009.

[38] N. Bohr. On the constitution of atoms and molecules. *Philosophical Magazine*, 26(151):1–25, July 1913.

[39] James F. Ziegler, J. P. Biersack, and Matthias D. Ziegler. *SRIM, the Stopping and Range of Ions in Matter*. SRIM Company, January 2008. Google-Books-ID: JSN63qxPG5MC.

[40] Enrico Fermi and Edward Teller. The capture of negative mesotrons in matter. *Physical Review*, 72(5):399, 1947.

[41] D. Allan Bromley, editor. *Treatise on Heavy-Ion Science*. Springer US, Boston, MA, 1985. DOI: 10.1007/978-1-4615-8103-1.

[42] C. C. Rousseau, W. K. Chu, and D. Powers. Calculations of stopping cross sections for 0.8-to 2.0-MeV alpha particles. *Physical Review A*, 4(3):1066, 1971.

[43] J Lindhard. On the Properties of a Gas of Charged Particles. *Dan. Mat. Fys. Medd.*, 28(8), 1954.

[44] David B. Graves and Pascal Brault. Molecular dynamics for low temperature plasmasurface interaction studies. *Journal of Physics D: Applied Physics*, 42(19):194011, 2009.

[45] W. Eckstein, S. Hackel, D. Heinemann, and B. Fricke. Influence of the interaction potential on simulated sputtering and reflection data. *Zeitschrift fur Physik D Atoms, Molecules and Clusters*, 24(2):171–176, 1992.

[46] Alexander I. Melker. Potentials of interatomic interaction in molecular dynamics. *Rev. Adv. Mater. Sci*, 20:1–13, 2009.

[47] Eleuterio F. Toro. The Method of Godunov for Nonlinear Systems. In *Riemann Solvers and Numerical Methods for Fluid Dynamics*, pages 213–235. Springer Berlin Heidelberg, 2009. DOI: 10.1007/b79761_6.

[48] B. van Leer. Upwind and High-Resolution Methods for Compressible Flow: From Donor Cell to Residual-Distribution Schemes. *Commun. Comput. Phys.*, 1(2):192–206, April 2006.

[49] A. Jameson, W. Schmidt, and E. Turkel. Numerical Simulation of the Euler Equations by Finite Volume methods Using Runge-Kutta Time Stepping Schemes. Palo Alto, California, 1981.

[50] N. Crouseilles, M. Mehrenberger, and F. Vecil. Discontinuous Galerkin semi-Lagrangian method for Vlasov-Poisson. *ESAIM: Proceedings*, 32:211–230, October 2011.

[51] Akil Narayan and Andreas Klockner. Deterministic numerical schemes for the Boltzmann equation. *arXiv preprint arXiv:0911.3589*, 2009.

[52] G.I. Marchuck. Splitting and Alternating Direction Methods. In *Variational Methods for the Numerical Solution of Nonlinear Elliptic Problems*, CBMS-NSF Regional Conference Series in Applied Mathematics, pages 203–461. 2015.

[53] Ge-Cheng Zha and Chakradhar Lingamgunta. On the Accuracy of Runge-Kutta Methods for Unsteady Linear Wave Equation. *American Institute of Aeronautics and Astronautics*, 2003.

[54] Lawrence Takacs. A Two-Step Scheme for the Advection Equation with Minimized Dissipation and Dispersion Errors. *Monthly Weather Review*, 113:1050–1065, February 1985.

[55] Francis F. Chen. *Introduction to Plasma Physics and Controlled Fusion*. Springer US, Boston, MA, 1984. DOI: 10.1007/978-1-4757-5595-4.

[56] Oreste Pezzi, Enrico Camporeale, and Francesco Valentini. Collisional effects on the numerical recurrence in Vlasov-Poisson simulations. *Physics of Plasmas*, 23(2):022103, February 2016. arXiv: 1601.05240.

[57] F Filbet and E Sonnendrucker. Comparison of Eulerian Vlasov Solvers. *Computer Physics Communications*, (150):247–266, June 2002.

[58] Nicolas Crouseilles, Thomas Respaud, and Eric Sonnendrcker. A forward semi-Lagrangian method for the numerical solution of the Vlasov equation. *Computer Physics Communications*, 180(10):1730–1745, October 2009.

[59] Dan M. Goebel and Ira Katz. *Fundamentals of electric propulsion: ion and Hall thrusters*, volume 1. John Wiley & Sons, 2008.

[60] David Coulette and Giovanni Manfredi. An Eulerian Vlasov code for plasma-wall interactions. *Journal of Physics: Conference Series*, 561:012005, November 2014.

[61] S. Devaux and G. Manfredi. Vlasov simulations of plasma-wall interactions in a magnetized and weakly collisional plasma. *Physics of Plasmas*, 13(8):083504, August 2006.

[62] K. L. Bell, H. B. Gilbody, J. G. Hughes, A. E. Kingston, and F. J. Smith. Recommended Data on the Electron Impact Ionization of Light Atoms and Ions. *Journal of Physical and Chemical Reference Data*, 12(4):891–916, October 1983.

[63] S. A. Maiorov, O. F. Petrov, and V. E. Fortov. Calculation of resonant charge exchange cross sections of ions of rubidium, cesium, mercury, and noble gases. In *Proceedings of the 34th EPS Conference on Plasma Physics, Warsaw*, 2007.

[64] P. C. Stangeby. *The plasma boundary of magnetic fusion devices*. Plasma physics series. Institute of Physics Pub, Bristol ; Philadelphia, 2000.

[65] I. J. Good and R. A. Gaskins. Nonparametric Roughness Penalties for Probability Densities. *Biometrika*, 58(2):255–277, 1971.

[66] Jarek Duda. Rapid parametric density estimation. *arXiv:1702.02144 [cs]*, February 2017. arXiv: 1702.02144.

[67] Jonathan Q. Li and Andrew R. Barron. Mixture density estimation. In *Advances in neural information processing systems*, pages 279–285, 2000.

[68] Dilan Gorur and Carl Edward Rasmussen. Dirichlet process gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*, 25(4):653–664, 2010.

[69] Murray Rosenblatt. Remarks on Some Nonparametric Estimates of a Density Function. *Ann. Math. Statist.*, 27:832–837, 1956.

[70] Bernard W. Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[71] J. Thompson and R. Tapia. *Nonparametric Function Estimation, Modeling, and Simulation*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, January 1990. DOI: 10.1137/1.9781611971712.

[72] Marcin Budka, Bogdan Gabrys, and Katarzyna Musial. On Accuracy of PDF Divergence Estimators and Their Applicability to Representative Data Sampling. *Entropy*, 13(7):1229–1266, July 2011.

[73] David W. Scott. *Multivariate density estimation: theory, practice, and visualization*. Wiley series in probability and mathematical statistics. Wiley, New York, 1992.

[74] Marina Vannucci. *Nonparametric density estimation using wavelets*. Citeseer, 1995.

[75] Simon J. Sheather. Density Estimation. *Statistical Science*, 19(4):588–597, November 2004.

[76] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[77] Fabian Pedregosa, Gal Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and others. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[78] J. D. Hunter. Matplotlib: A 2d Graphics Environment. *Computing in Science Engineering*, 9(3):90–95, May 2007.

[79] Michael Snipes and D. Christopher Taylor. Model selection and Akaike Information Criteria: An example from wine ratings and prices. *Wine Economics and Policy*, 3(1):3–9, June 2014.

[80] David N. Ruzic. The effects of surface roughness characterized by fractal geometry on sputtering. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 47(2):118–125, 1990.

[81] H. Hofsss, K. Zhang, and A. Mutzke. Simulation of ion beam sputtering with SDTrimSP, TRIDYN and SRIM. *Applied Surface Science*, 310:134–141, August 2014.

[82] W. Moller and D. Guttler. Modeling of plasma-target interaction during reactive magnetron sputtering of TiN. *Journal of Applied Physics*, 102(9):094501, November 2007.

[83] T Kubart, T Nyberg, and S Berg. Modelling of low energy ion sputtering from oxide surfaces. *Journal of Physics D: Applied Physics*, 43(20):205204, May 2010.

[84] Ordean S. Oen and Mark T. Robinson. Computer studies of the reflection of light ions from solids. *Nuclear Instruments and Methods*, 132:647–653, January 1976.

[85] Bradley Efron. The Jackknife, the Bootstrap, and Other Resampling Plans. Technical Report 163, Stanford University, Stanford, California, December 1980.

[86] Jeffrey C. Miecznikowski, Dongliang Wang, and Alan Hutson. Bootstrap MISE Estimators to Obtain Bandwidth for Kernel Density Estimation. *Communications in Statistics - Simulation and Computation*, 39(7):1455–1469, July 2010.

[87] E. Mammen. *When does bootstrap work?: asymptotic results and simulations*. Number 77 in Lecture notes in statistics. Springer-Verlag, New York, 1992.

[88] Julian J. Faraway and Myoungshic Jhun. Bootstrap Choice of Bandwidth for Density Estimation. *Journal of the American Statistical Association*, 85(412), December 1990.

[89] Prabhu Ramachandran and Gael Varoquaux. Mayavi: a package for 3d visualization of scientific data. *Computing in Science and Engineering*, 13(2):40–51, 2011.

[90] P.C. Stangeby. The Chodura sheath for angles of a few degrees between the magnetic field and the surface of divertor targets and limiters. *Nuclear Fusion*, 52(8):083012, August 2012.