

Utah State University

DigitalCommons@USU

All Graduate Theses and Dissertations, Fall
2023 to Present

Graduate Studies

8-2024

Implementing General Moment Equations for Parallel Closures in NIMROD

Hankyu Lee
Utah State University

Follow this and additional works at: <https://digitalcommons.usu.edu/etd2023>



Part of the [Physics Commons](#)

Recommended Citation

Lee, Hankyu, "Implementing General Moment Equations for Parallel Closures in NIMROD" (2024). *All Graduate Theses and Dissertations, Fall 2023 to Present*. 222.

<https://digitalcommons.usu.edu/etd2023/222>

This Dissertation is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations, Fall 2023 to Present by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.



IMPLEMENTING GENERAL MOMENT EQUATIONS FOR PARALLEL CLOSURES

IN NIMROD

by

Hankyu Lee

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Physics

Approved:

Jeong-Young Ji, Ph.D.
Major Professor

Eric D. Held, Ph.D.
Committee Member

J. Andrew Spencer, Ph.D.
Committee Member

James Wheeler, Ph.D.
Committee Member

Som Dutta, Ph.D.
Committee Member

D. Richard Cutler, Ph.D.
Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2024

Copyright © Hankyu Lee 2024

All Rights Reserved

ABSTRACT

Implementing general moment equations for parallel closures in NIMROD

by

Hankyu Lee, Doctor of Philosophy

Utah State University, 2024

Major Professor: Jeong-Young Ji, Ph.D.

Department: Physics

Implementing an advanced closure module significantly extends the capability of magnetohydrodynamics (MHD) fluid codes such as NIMROD. This module solves non-Maxwellian parallel moment equations to obtain parallel closures. These equations are derived by taking moments of the first-order drift kinetic equation with orthogonal velocity polynomials. We solve the system of parallel moment equations using two-dimensional finite elements in the poloidal plane, considering an axisymmetric magnetic field. The results indicate that high-resolution meshes and substantial memory capacity are essential. To address this, a hybrid method is implemented to solve the system in Fourier space over the poloidal angle of a tokamak and transforms results back to finite element space. The closures demonstrate convergence with an increasing number of moments and accurately solve the drift kinetic equation, making this approach suitable for studying kinetic effects in fluid simulations for nuclear fusion devices.

(77 pages)

PUBLIC ABSTRACT

Implementing general moment equations for parallel closures in NIMROD

Hankyu Lee

Understanding how magnetic fields impact plasma transport is essential for improving the efficiency of thermonuclear fusion power plants. To address the transport problem, both plasma fluid equations and Maxwell's equations must be solved. To solve these equations, it is necessary to derive closure relations that allow the system to be closed. Previous closure models are useful for describing the behavior of high-collisionality plasma but are not effective at low collisionality. To obtain closure relations valid for low collisionality, the first-order drift kinetic equation must be solved.

This study presents methods for numerically obtaining parallel closures for NIMROD code by deriving a system of parallel moment equations from the drift kinetic equation in an axisymmetric magnetic field. Two methods are introduced: one uses two-dimensional finite elements in the poloidal plane of a tokamak. The other is a hybrid method that reduces memory burden by using the Fourier method over the poloidal angle of the tokamak to obtain closures, and then converting them back to the finite element basis. The obtained closures show convergence with an increasing number of moments and accurately resolve the drift kinetic equation, making this approach effective for incorporating kinetic effects into fluid simulations for nuclear fusion devices.

ACKNOWLEDGMENTS

I am deeply grateful to my supervisor, Dr. Jeong-Young Ji, for his expert guidance, unwavering support, and invaluable insights throughout my research journey. His mentorship was crucial in the completion of this dissertation.

I would also like to express my sincere appreciation to my committee members, Dr. Eric D. Held, Dr. J. Andrew Spencer, Dr. James Wheeler, and Dr. Som Dutta, for their invaluable feedback and encouragement.

Special thanks go to my friend Chanju Lee, whose support and love have been a constant source of motivation and comfort during my academic pursuits.

Additionally, I extend my thanks to the NIMROD team for their productive discussions about code implementation, and I am grateful to the NERSC help desk for their in-depth consulting when I utilized the NERSC supercomputing system.

Finally, I acknowledge the generous support from the Howard L. Blood Scholarship, which significantly facilitated my research. This accomplishment would not have been possible without such backing.

Hankyu Lee

CONTENTS

	Page
ABSTRACT	iii
PUBLIC ABSTRACT	iv
ACKNOWLEDGMENTS	v
LIST OF FIGURES	viii
INTRODUCTION	1
KINETIC DESCRIPTION OF A PLASMA	3
KINETIC AND FLUID EQUATIONS	3
COLLISION OPERATOR	4
NEOCLASSICAL TRANSPORT AND PARALLEL MOMENT EQUATIONS	10
AXISYMMETRIC MAGNETIC FIELD	10
DIAMAGNETIC FLOW IN AN AXISYMMETRIC MAGNETIC FIELD	12
DRIFT KINETIC EQUATION	14
PARALLEL MOMENT EQUATIONS	16
OBTAINING ION CLOSURE RELATIONS	19
IMPLEMENTING PARALLEL MOMENT EQUATIONS USING FINITE ELEMENT METHOD	21
FINITE ELEMENT METHOD	21
IMPLEMENTING PARALLEL MOMENT EQUATIONS USING FINITE ELE- MENT METHOD	29
IMPLEMENTING MOMENT-FOURIER EQUATIONS	38
FOURIER METHOD OF SOLVING MOMENT EQUATIONS AND OBTAINING CLOSURES	38
OBTAINING FOURIER COEFFICIENTS OF FIELDS FOR IMPLEMENTING THE EQUATIONS	39
OBTAINING THE CLOSURE MATRIX	40
REPRESENTING CLOSURES WITH FINITE ELEMENT BASIS	42
FLUID EQUATIONS AND TIME ADVANCE IN NIMROD	49
LINEAR FLUID EQUATIONS AND CLOSURES	49
TIME ADVANCE OF FLUID EQUATIONS	50
CONCLUSION AND FUTURE WORK	59
REFERENCES	60

APPENDICES	62
A Algorithm for measuring magnetic field length between two nodes on a flux surface	63
B Data flow for the closure module in NIMROD	65
CURRICULUM VITAE	67

LIST OF FIGURES

	Page	
Figure 1	Diagram of differential cross section.	7
Figure 2	Differential cross section versus scattering angle for deuterium with kinetic energy 10 keV scattering off a tritium nucleus.	8
Figure 3	Tokamak model used in this study, featuring a major radius of 5 m and a minor radius of 0.5 m.	11
Figure 4	Quadratic basis functions for 1-D finite element with nodes at vertices and interior points.	24
Figure 5	Poloidal mesh (physical grid) used in this study shows the 2-D finite element cells in the R - Z plane.	26
Figure 6	Logical coordinates used in this study with ix (ξ), moving across constant- ψ surfaces, and iy (η), moving around in the periodic, poloidal angle.	27
Figure 7	<code>lagr_quad</code> type arrays over the red square on Fig. 6.	28
Figure 8	Ion \hat{h}_h closure.	31
Figure 9	Ion $\hat{\pi}_\pi$ closure.	32
Figure 10	Equilibrium ion temperature and number density across flux surfaces at $\theta = \frac{\pi}{2}$	33
Figure 11	Equilibrium ion temperature around a flux surface calculated by using linear and quadratic basis functions.	34
Figure 12	Ion parallel heat flux (left) and viscosity (right) closures driven by constant $\partial_{\parallel}T$ (top) and W_{\parallel} (bottom).	35
Figure 13	Ion parallel heat flux (left) and viscosity (right) closures driven by $\partial_{\parallel}T = \sin \theta$ (top) and $W_{\parallel} = \sin \theta$ (bottom).	36
Figure 14	Ion parallel heat flux (left) and viscosity (right) closures driven by $\partial_{\parallel}T = \cos \theta$ (top) and $W_{\parallel} = \cos \theta$ (bottom).	37
Figure 15	Ion parallel closures due to $dT_0/d\psi$ for $(\text{numL}, \text{numK}) = (4, 8)$, and $\text{nfo} = 1$ (red,dotted), 2 (green, solid), 3 (blue, solid), 4 (cyan, solid), 5 (black, dash-dotted).	45

Figure 16	Ion parallel closures due to $T_1 = \sin \theta$ for $(\text{numL}, \text{numK}) = (4, 8)$, and $\text{nfo} = 1$ (red, dotted), 2 (green, solid), 3 (blue, solid), 4 (cyan, solid), 5 (black, dash-dotted).	46
Figure 17	Ion parallel closures due to $T_1 = \cos \theta$ for $(\text{numL}, \text{numK}) = (4, 8)$, and $\text{nfo} = 1$ (red, dotted), 2 (green, solid), 3 (blue, solid), 4 (cyan, solid), 5 (black, dash-dotted).	47
Figure 18	Ion closures including all drives in (115) using $\text{nfo} = 4$, calculated by solving $(\text{numL}, \text{numK}) = (10, 20), (20, 40), (40, 80), (80, 160)$ system at $\epsilon = 0.03, 0.06, 0.09$.	48
Figure 19	Fluid quantities $\nabla \cdot \mathbf{V}_{\parallel}$, $\nabla \cdot \mathbf{V}_{\perp}$, $\nabla \cdot (\mathbf{V}_{\parallel} + \mathbf{V}_{\perp})$ (left), $p_0 \nabla \cdot \mathbf{V}_{\parallel}$, $p_0 \nabla \cdot \mathbf{V}_{\perp}$, $p_0 \nabla \cdot (\mathbf{V}_{\parallel} + \mathbf{V}_{\perp})$ (right) calculated by solving $(\text{numL}, \text{numK}) = (4, 8)$ system at $\epsilon = 0.03$ (blue), 0.06 (purple), 0.09 (red).	51
Figure 20	Fluid quantities $\nabla \cdot \mathbf{V}_{\parallel}$, $\nabla \cdot \mathbf{V}_{\perp}$, $\nabla \cdot (\mathbf{V}_{\parallel} + \mathbf{V}_{\perp})$ (left), $p_0 \nabla \cdot \mathbf{V}_{\parallel}$, $p_0 \nabla \cdot \mathbf{V}_{\perp}$, $p_0 \nabla \cdot (\mathbf{V}_{\parallel} + \mathbf{V}_{\perp})$ (right) calculated by solving $(\text{numL}, \text{numK}) = (40, 80)$ system at $\epsilon = 0.03$ (blue), 0.06 (purple), 0.09 (red).	52
Figure 21	Fluid quantities $\nabla \cdot \mathbf{h}_{\parallel}$ (top), $\nabla \cdot \mathbf{h}_{\perp}$ (middle), and $\nabla \cdot (\mathbf{h}_{\parallel} + \mathbf{h}_{\perp})$ (bottom) calculated by solving $(\text{numL}, \text{numK}) = (4, 8)$ (left) and $(\text{numL}, \text{numK}) = (40, 80)$ (right) systems at $\epsilon = 0.03$ (blue), 0.06 (purple), 0.09 (red).	53
Figure 22	Fluid quantities $T_0 \partial_{\parallel} n_1$ (top left), $n_0 \partial_{\parallel} T_1$ (bottom left), $\mathbf{b} \cdot \nabla \cdot \pi_{\parallel}$ (top right), and the sum of all terms (bottom right) calculated by solving the system $(\text{numL}, \text{numK}) = (4, 8)$ at $\epsilon = 0.03$ (blue), 0.06 (purple), 0.09 (red).	54
Figure 23	Fluid quantities $T_0 \partial_{\parallel} n_1$ (top left), $n_0 \partial_{\parallel} T_1$ (bottom left), $\mathbf{b} \cdot \nabla \cdot \pi_{\parallel}$ (top right), and the sum of all terms (bottom right) calculated by solving the system $(\text{numL}, \text{numK}) = (40, 80)$ at $\epsilon = 0.03$ (blue), 0.06 (purple), 0.09 (red).	55
Figure 24	Computation time in minutes for solving the systems with $(\text{numL}, \text{numK}) = (10, 20), (20, 40)$, and $(40, 80)$ using serial (blue) and parallel (orange) algorithms.	56
Figure 25	Time slices (starting from green to red) of perturbed number density (top), $\nabla \cdot \mathbf{V}_{\perp}$ (middle), and an enlarged image of $\nabla \cdot \mathbf{V}_{\perp}$ (bottom, inside the black circle) at $\epsilon = 0.03$.	58
Figure A.1	Zones for determining the status of a trial point.	64

INTRODUCTION

The distribution function of a plasma plays a pivotal role in the study of plasma physics, serving as a fundamental descriptor of the statistical properties of plasma particles in phase space. This function, typically denoted as $f(t, \mathbf{x}, \mathbf{v})$, represents the probability density of finding a particle at a position \mathbf{x} with velocity \mathbf{v} at time t . By integrating over this distribution, one can derive macroscopic plasma properties such as density, velocity, and temperature, thereby linking microscopic particle dynamics to observable plasma behavior. Understanding the distribution function and its evolution in phase space is essential for analyzing various plasma phenomena, from basic wave-particle interactions to complex kinetic behaviors in thermonuclear devices such as tokamaks.

In tokamaks, neoclassical transport theory provides a framework for understanding the transport and confinement of plasmas with inhomogeneous magnetic fields. This theory, which considers the effects of collisions and magnetic field configurations on particle motion, is essential for studying the physical properties of plasmas and improving the efficiency of thermonuclear fusion reactions. One objective of neoclassical transport theory is to develop advanced methods for obtaining parallel closures aligned with magnetic field directions. These closures, such as heat flux and viscosity, are essential for solving fluid equations related to number density, temperature, and flow velocity; thus named because they "close" these equations. Traditional models like Braginskii's ([Braginskii, 1965](#)) fail to capture kinetic effects in low-collisional plasmas. Solving the drift kinetic equation (DKE) is essential to obtain closures that accurately describe kinetic effects even in such plasmas.

Many previous studies have attempted to solve the DKE. Helander and Sigmar summarized a framework for analytically solving the DKE ([Helander & Sigmar, 2005](#)). Numerous computational codes have been developed for this purpose, and Held et al. solved it on velocity grids using the finite element method ([Held et al., 2015](#)). However, obtaining parallel closures using the DKE still requires considerable research.

Ji et al. proposed using the parallel moment approach ([Ji et al., 2023](#)) to solve the

DKE, aiming to obtain parallel closures. This method involves decomposing the gyro-averaged distribution function into moments and solving for these moments, allowing for a more nuanced understanding of plasma behavior under the influence of electromagnetic forces. The parallel moment approach promises a more robust and scalable solution to the complexities of the DKE, potentially leading to more accurate simulations that can inform both theoretical research and practical applications in plasma physics.

This work focuses on providing the advanced closure module based on the parallel moment approach into the NIMROD code (Sovinec et al., 2003), a well-regarded computational tool used in the study of magnetohydrodynamic (MHD) stability and the evolution of plasmas. The simulation tool enables detailed analysis by allowing adjustments and enhancements to meet specific research needs, such as investigating fluid dynamics under varying magnetic field conditions.

In the following chapters, we delve into the numerical implementation of parallel moment equations designed to apply the advanced closure model for plasma transport. In chapter 2, we introduce the kinetic theory of a plasma. In chapter 3, we overview a framework by solving the first-order drift kinetic equation in an axisymmetric magnetic field by using the moment approach. In chapter 4, we introduce the finite element method and describe the numerical framework for implementing this method in the NIMROD code. We present the closure results calculated by NIMROD and discuss some numerical challenges encountered. In chapter 5, we introduce the moment-Fourier equation, which is numerically beneficial for solving the parallel moment equations, and show calculated closure results. In chapter 6, we discuss time advance scheme of fluid equations and parallelism to enhance the time efficiency of the simulation. In chapter 7, we conclude and discuss future plans.

KINETIC DESCRIPTION OF A PLASMA

Kinetic and fluid equations

A distribution function $f(t, \mathbf{x}, \mathbf{v})$ contains the necessary information to describe transport phenomena of a plasma. In a given electric $\mathbf{E}(t, \mathbf{x})$ and magnetic field $\mathbf{B}(t, \mathbf{x})$, the distribution function for a plasma species a with mass m_a and charge q_a is governed by the kinetic equation:

$$\left[\frac{\partial}{\partial t} + \mathbf{v} \cdot \frac{\partial}{\partial \mathbf{x}} + \frac{q_a}{m_a} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial}{\partial \mathbf{v}} \right] f_a = C(f_a), \quad (1)$$

where $C(f_a)$ is the collision operator. Using this equation, we can derive equations for fluid quantities such as number density $n_a(t, \mathbf{x})$, temperature $T_a(t, \mathbf{x})$, and flow velocity $\mathbf{V}_a(t, \mathbf{x})$:

$$\frac{\partial n_a}{\partial t} + \nabla \cdot (n_a \mathbf{V}_a) = 0, \quad (2)$$

$$\frac{3}{2} n_a \left(\frac{\partial T_a}{\partial t} + \mathbf{V}_a \cdot \nabla T_a \right) + n_a T_a \nabla \cdot \mathbf{V}_a + \nabla \cdot \mathbf{h}_a + \nabla \mathbf{V}_a : \boldsymbol{\pi}_a = Q_a, \quad (3)$$

$$m_a n_a \left(\frac{\partial \mathbf{V}_a}{\partial t} + \mathbf{V}_a \cdot \nabla \mathbf{V}_a \right) - n_a q_a (\mathbf{E} + \mathbf{V}_a \times \mathbf{B}) + \nabla p_a + \nabla \cdot \boldsymbol{\pi}_a = \mathbf{R}_a. \quad (4)$$

These fluid quantities can be considered as moments of the distribution function. The number density, temperature, and flow velocity are defined as follows:

$$n_a(t, \mathbf{x}) \equiv \int d\mathbf{v} f_a(t, \mathbf{x}, \mathbf{v}), \quad (5)$$

$$n_a \mathbf{V}_a(t, \mathbf{x}) \equiv \int d\mathbf{v} \mathbf{v} f_a(t, \mathbf{x}, \mathbf{v}), \quad (6)$$

$$\frac{3}{2} n_a T_a(t, \mathbf{x}) \equiv \int d\mathbf{v} \frac{1}{2} m_a w_a^2 f(t, \mathbf{x}, \mathbf{v}), \quad (7)$$

where the random velocity $\mathbf{w}_a = \mathbf{v} - \mathbf{V}_a$. By multiplying $1, m_a \mathbf{w}_a, \frac{1}{2} m_a w_a^2$ to the kinetic equation and integrating it over the velocity space, we can obtain (2-4). Equations (2-4)

are considered open (incomplete for solving) due to closure quantities such as heat flux \mathbf{h}_a and viscosity $\boldsymbol{\pi}_a$, highlighted in red. To close the set of equations, these quantities must be expressed in terms of n, T , and \mathbf{V} .

When deriving moments of the kinetic equation, we need to calculate the moments of the collision operator. These collisional moments can be considered as rates of change due to collisions. For example, the integral

$$\int d\mathbf{v} C(f_a) = 0 \quad (8)$$

indicates that the collisional effects are assumed not to change the density (Gurnett & Bhattacharjee, 2017). Other collisional moments, such as

$$\int d\mathbf{v} m_a \mathbf{w}_a C(f_a) = \mathbf{R}_a, \quad (9)$$

$$\int d\mathbf{v} \frac{1}{2} m_a w_a^2 C(f_a) = Q_a, \quad (10)$$

are considered as the collisional momentum exchange rate and collisional heat exchange rate, respectively.

In local thermodynamic equilibrium, the lowest order terms of collisional moments vanish when all plasma species share the same flow velocity and temperature (Helander & Sigmar, 2005). Subsequently, the distribution function is described by the Maxwellian distribution function,

$$f_a^M(t, \mathbf{x}, \mathbf{v}) = \frac{n_{a0}}{\pi^{3/2} v_{a0}^3} e^{-((\mathbf{v} - \mathbf{V}_{a0})/v_{a0})^2}, \quad (11)$$

where subscript 0 denotes the equilibrium quantity and the thermal speed $v_{a0} = \sqrt{\frac{2T_{a0}}{m_a}}$. Here, n_a, T_a, \mathbf{V}_a , which are included in (11), are often referred to as Maxwellian moments.

Collision operator

The Boltzmann collision operator describes the binary collisions between plasma species a and b :

$$C(f_a, f_b) = \int d\mathbf{v}' \int d\Omega \sigma |\mathbf{v} - \mathbf{v}'| [f_a(\mathbf{v}_*) f_b(\mathbf{v}'_*) - f_a(\mathbf{v}) f_b(\mathbf{v}')], \quad (12)$$

where \mathbf{v}_* denotes the velocity after the collision, Ω is the solid angle, and σ is the differential cross section. However, a simpler form of the collision operator is required for analytic methods in neoclassical transport theory. Previous studies have developed simplified forms of operators based on observations from binary elastic collisions involving Coulomb interactions, known as Rutherford scattering. In these interactions, most collisions result in particles being deflected at small angles. The Rutherford scattering angle θ is determined by an impact parameter ρ as follows (Goldstein et al., 2017):

$$\sin \frac{\theta}{2} = \frac{1}{\sqrt{1 + \left(\frac{\rho}{\rho_0}\right)^2}}, \quad (13)$$

where

$$\rho_0 = \frac{|\alpha|}{m_{ab} u^2}, \quad (14)$$

$$\alpha = \frac{q_a q_b}{4\pi \epsilon_0}, \quad (15)$$

$$m_{ab} = \frac{m_a m_b}{m_a + m_b}, \quad (16)$$

and $\mathbf{u} = \mathbf{v} - \mathbf{v}'$. The differential cross section σ is

$$\sigma \equiv \frac{d\Sigma}{d\Omega}, \quad (17)$$

$$= \left| \frac{\rho d\rho d\varphi}{\sin \theta d\theta d\varphi} \right|, \quad (18)$$

$$= \frac{\rho_0^2}{4 \sin^4 \left(\frac{\theta}{2}\right)}, \quad (19)$$

where $d\Sigma = \rho d\rho d\varphi$ and $d\Omega = \sin \theta d\theta d\varphi$. Fig. 1 illustrates a diagram for understanding the differential cross section. A small impact parameter ρ results in the solid angle Ω encompassing nearly all directions, leading to a minimal differential cross section σ . On the other hand, a large ρ means Ω encompasses only a narrow range of directions, causing σ to

increase. Fig. 2 illustrates that in the majority of the cross section, most collisions between deuterium and tritium deflect particles at small angles, with the moving deuterium having a kinetic energy of 10 keV.

Therefore, the distribution function after the collision can be approximated using a Taylor series

$$f(\mathbf{v}_*) \approx f(\mathbf{v}) + \Delta\mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{v}} + \frac{1}{2} \Delta\mathbf{v} \Delta\mathbf{v} : \frac{\partial^2 f}{\partial \mathbf{v}^2} \quad (20)$$

where $\Delta\mathbf{v} = \mathbf{v}_* - \mathbf{v}$. By using the center-of-mass frame, $\Delta\mathbf{v}$ can be changed to:

$$\Delta\mathbf{v} = \frac{m_{ab}}{m_a} \Delta\mathbf{u}, \quad (21)$$

where $\Delta\mathbf{u} = \mathbf{u}_* - \mathbf{u}$. Then, equation (12) can be expressed as (Ji, 2014a):

$$\begin{aligned} C(f_a, f_b) &= \int d\mathbf{v}' u \left[\frac{m_{ab}}{m_a} \langle \Delta\mathbf{u} \rangle_\Sigma \cdot \frac{\partial f_a}{\partial \mathbf{v}} f_b - \frac{m_{ab}}{m_b} f_a \langle \Delta\mathbf{u} \rangle_\Sigma \cdot \frac{\partial f_b}{\partial \mathbf{v}'} \right. \\ &\quad + \frac{1}{2} f_b \left(\frac{m_{ab}}{m_a} \right)^2 \langle \Delta\mathbf{u} \Delta\mathbf{u} \rangle_\Sigma : \frac{\partial}{\partial \mathbf{v}} \frac{\partial}{\partial \mathbf{v}} f_a \\ &\quad - \frac{m_{ab}}{m_a} \frac{m_{ab}}{m_b} \langle \Delta\mathbf{u} \Delta\mathbf{u} \rangle_\Sigma : \frac{\partial f_a}{\partial \mathbf{v}} \frac{\partial f_b}{\partial \mathbf{v}'} \\ &\quad \left. + \frac{1}{2} f_a \left(\frac{m_{ab}}{m_a} \right)^2 \langle \Delta\mathbf{u} \Delta\mathbf{u} \rangle_\Sigma : \frac{\partial}{\partial \mathbf{v}'} \frac{\partial}{\partial \mathbf{v}'} f_b, \right. \end{aligned} \quad (22)$$

where $\langle \Delta\mathbf{u} \rangle_\Sigma \equiv \int d\Sigma (\Delta\mathbf{u})$ and $\langle \Delta\mathbf{u} \Delta\mathbf{u} \rangle_\Sigma \equiv \int d\Sigma (\Delta\mathbf{u} \Delta\mathbf{u})$. In calculating $\langle \Delta\mathbf{u} \rangle_\Sigma$ and $\langle \Delta\mathbf{u} \Delta\mathbf{u} \rangle_\Sigma$, it is necessary to apply a cutoff at the lower bound of the integrals to avoid divergent values. This cutoff occurs at the angle θ_{\min} where ρ_{\max} equals the Debye length λ_D (Gurnett & Bhattacharjee, 2017). During this process, we also have defined a parameter $\Lambda_{ab} = \lambda_D / \rho_0$, where $\ln \Lambda_{ab}$ is called the Coulomb logarithm.

Consequently, by introducing the Landau tensor (Landau, 1936):

$$\mathbf{U} = \frac{u^2 \mathbf{I} - \mathbf{u}\mathbf{u}}{u^3}, \quad (23)$$

the collision operator (22) is transformed into the Landau collision operator:

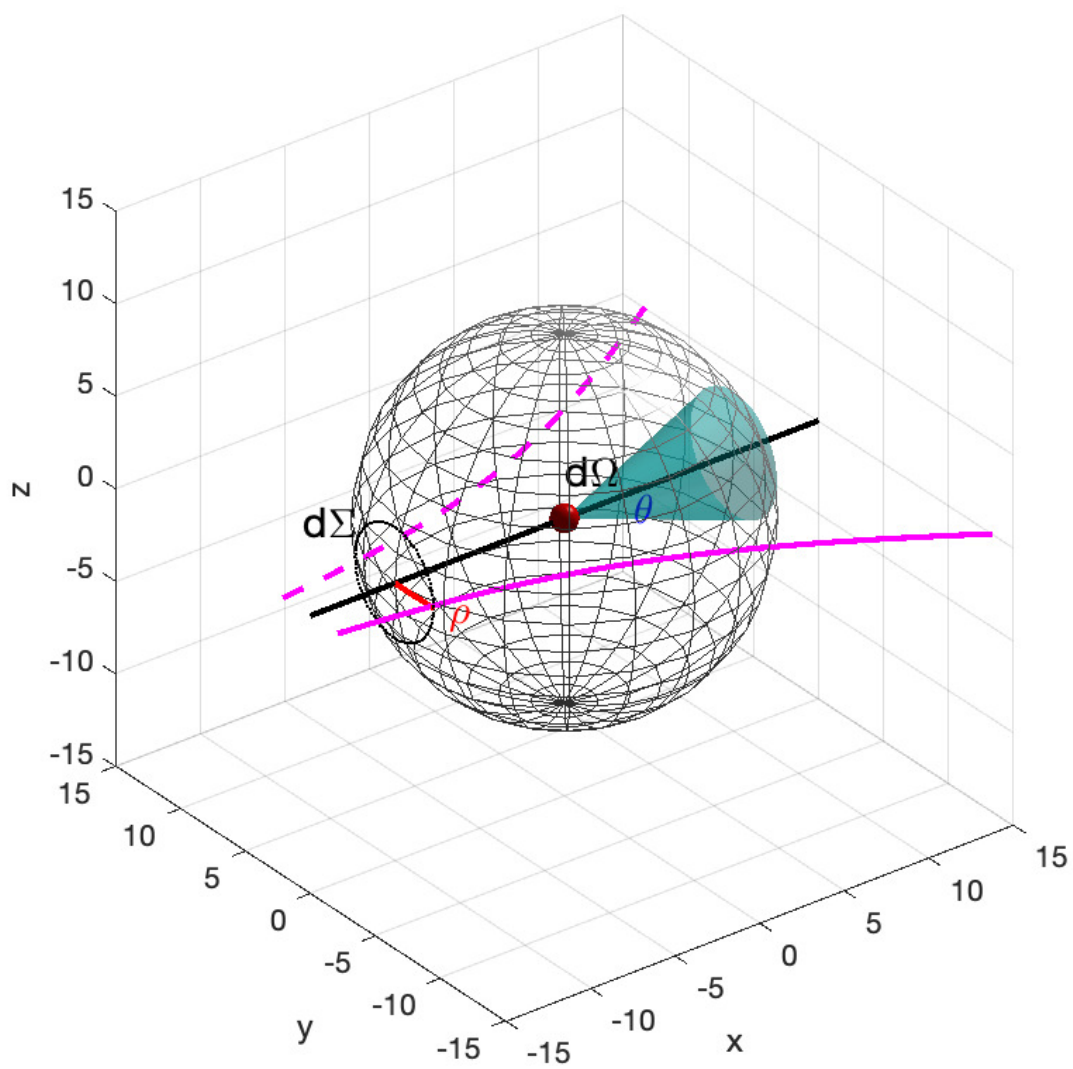


Figure 1: Diagram of differential cross section.

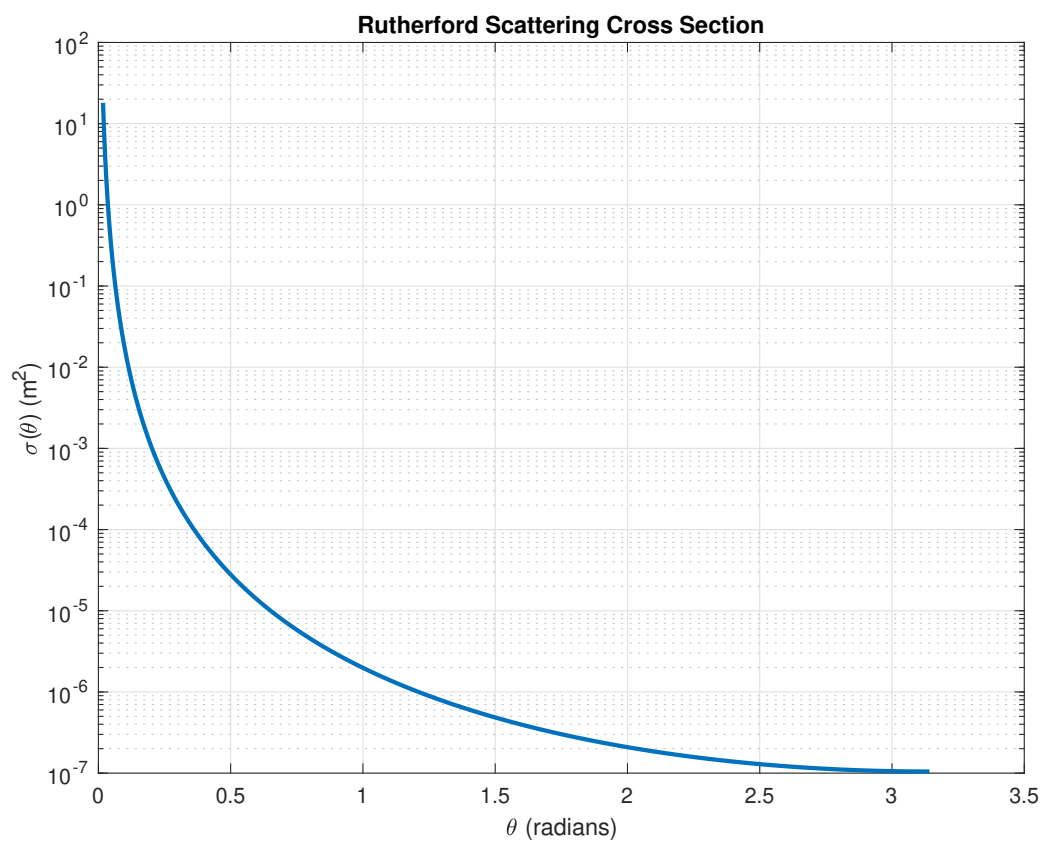


Figure 2: Differential cross section versus scattering angle for deuterium with kinetic energy 10 keV scattering off a tritium nucleus.

$$C(f_a, f_b) = \frac{\gamma_{ab}}{2} \frac{\partial}{\partial \mathbf{v}} \cdot \int d\mathbf{v}' \mathbf{U} \cdot \left(\frac{1}{m_a} \frac{\partial f_a}{\partial \mathbf{v}} f_b - f_a \frac{1}{m_b} \frac{\partial f_b}{\partial \mathbf{v}'} \right), \quad (24)$$

where

$$\gamma_{ab} = \frac{q_a^2 q_b^2 \ln \Lambda_{ab}}{4\pi \epsilon_0^2 m_a}. \quad (25)$$

Additionally, it is convenient to introduce the Rosenbluth potentials ([Rosenbluth et al., 1957](#)):

$$G_b(\mathbf{v}) = \int d\mathbf{v}' f_b(\mathbf{v}') |\mathbf{v} - \mathbf{v}'|, \quad (26)$$

$$H_b(\mathbf{v}) = \int d\mathbf{v}' f_b(\mathbf{v}') |\mathbf{v} - \mathbf{v}'|^{-1}, \quad (27)$$

to express (24) in a simpler form:

$$C(f_a, f_b) = \frac{\gamma_{ab}}{m_a} \left[\frac{1}{2} \frac{\partial}{\partial \mathbf{v}} \frac{\partial f_a}{\partial \mathbf{v}} : \frac{\partial}{\partial \mathbf{v}} \frac{\partial G_b}{\partial \mathbf{v}} + \left(1 - \frac{m_a}{m_b} \right) \frac{\partial f_a}{\partial \mathbf{v}} \cdot \frac{\partial H_b}{\partial \mathbf{v}} + 4\pi \frac{m_a}{m_b} f_a f_b \right]. \quad (28)$$

Many analytic studies employ a linearized collision operator. When the distribution function is separated into Maxwellian (f^M) and non-Maxwellian (f^N) components, the collision operator $C(f_a^N, f_b^N)$ can often be disregarded when f is very close to f^M . This form of the collision operator has been used to calculate the collisional moments in the moment approach ([Ji & Held, 2006](#)) which we discuss in the next chapter.

NEOCLASSICAL TRANSPORT AND PARALLEL MOMENT EQUATIONS

Axisymmetric magnetic field

Neoclassical transport theory was developed under the condition of an axisymmetric magnetic field. To describe a plasma in such a field, we introduce a cylindrical coordinate system (R, Z, φ) . The toroidal direction is defined as clockwise, with $\hat{\varphi} = \hat{\mathbf{R}} \times \hat{\mathbf{Z}}$. The magnetic field can be expressed in poloidal and toroidal directions as

$$\mathbf{B} = I\nabla\varphi + \nabla\varphi \times \nabla\psi, \quad (29)$$

where φ is the toroidal angle and ψ is referred to as the poloidal flux function (Helander & Sigmar, 2005). The function I can be understood by applying Ampere's law in the toroidal direction,

$$\oint \mathbf{B} \cdot d\ell_\varphi = \oint (I\nabla\varphi + \nabla\varphi \times \nabla\psi) \cdot d\ell_\varphi, \quad (30)$$

$$2\pi I = \mu_0 I_p, \quad (31)$$

where I_p is the poloidal current passing through the area enclosed by the loop of $d\ell_\varphi$ (Ji, 2014b).

A flux surface is a hypothetical surface on which the magnetic field vector lies. The poloidal flux function ψ serves as a label for these flux surfaces, where each value of ψ corresponds to a different flux surface. In an axisymmetric magnetic field, the nesting of these surfaces forms a toroidal structure, with each surface being toroidally symmetric. The innermost line of this structure is called the magnetic axis. Fig. 3 presents a diagram of the tokamak, which is the focus of this work. Within a poloidal plane, green concentric circles depict various flux surfaces, each can be labeled by a constant $\psi(R, Z)$. In this thesis, the configuration of the magnetic field is modeled as a circular field on a flux surface ψ , given

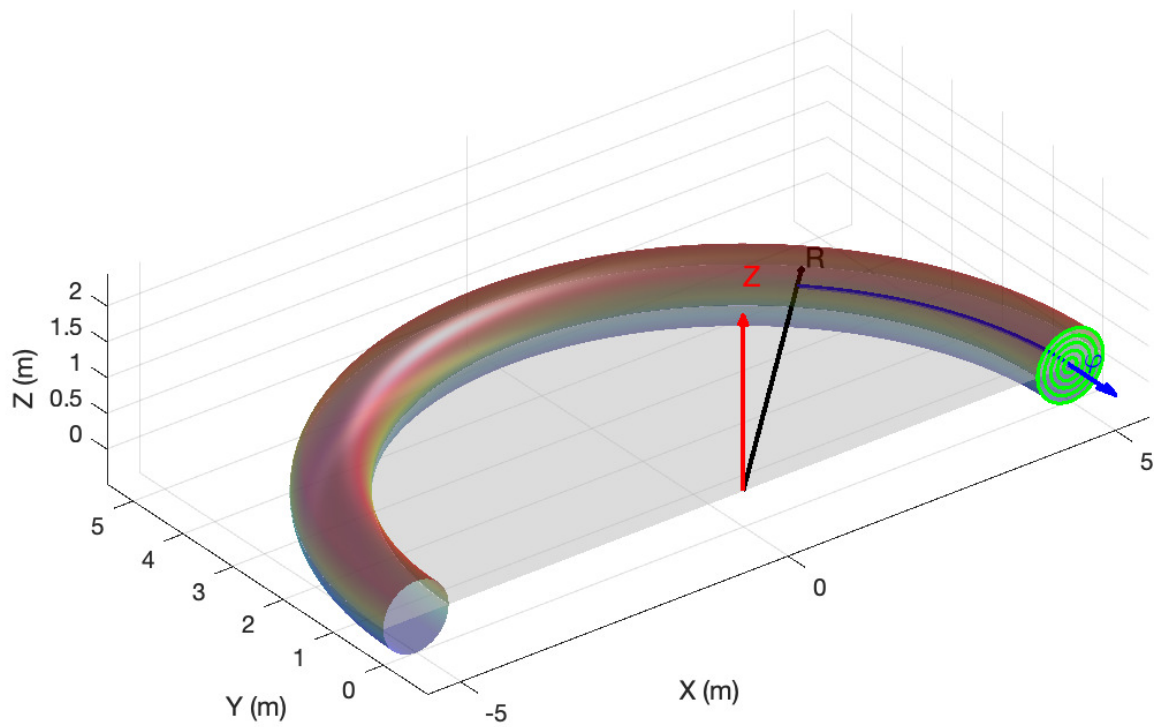


Figure 3: Tokamak model used in this study, featuring a major radius of 5 m and a minor radius of 0.5 m. Nested flux surfaces are indicated as the green circles in the poloidal cross section at the right. The magnetic axis is indicated by the blue line with the arrow.

by

$$B(\theta, \psi) = \frac{B_0(\psi)}{1 + \epsilon \cos \theta}, \quad (32)$$

where θ is the poloidal angle, B_0 represents the constant reference field, ϵ is the inverse aspect ratio, defined as the ratio of the minor radius r to the major radius R of the the flux surface ψ .

Diamagnetic flow in an axisymmetric magnetic field

In neoclassical transport theory, a plasma is assumed to be in a strong magnetic field. Particles in this magnetized plasma have very fast gyro-frequencies, as exemplified by electrons and ions with their respective frequencies $\Omega_e = q_e B / m_e$ for electrons and $\Omega_i = q_i B / m_i$ for ions. Consequently, $n_a q_a (\mathbf{V}_a \times \mathbf{B})$ in (4) can be considered the dominant term in the equation. Many analytic studies assume a negligible electric field, a distribution function that is very close to the Maxwellian, and $V \ll v_0$. They calculate the perpendicular flow velocity as follow:

$$\nabla p_a - n_a q_a (\mathbf{V}_a \times \mathbf{B}) = 0, \quad (33)$$

$$\mathbf{V}_{a\perp} = \frac{\mathbf{B} \times \nabla p_a}{n_a q_a B^2}, \quad (34)$$

where $\mathbf{V}_\perp = -\mathbf{b} \times (\mathbf{b} \times \mathbf{V})$ and $\mathbf{b} = \mathbf{B}/B$.

The perpendicular heat flux can be calculated in a similar manner. From the heat flux equation, which can be derived by taking the $\frac{1}{2} m_a \mathbf{w}_a w_a^2$ moment of (1), we have:

$$\frac{5p_a}{2m_a} \nabla T_a - \frac{q_a}{m_a} (\mathbf{h}_a \times \mathbf{B}) = 0, \quad (35)$$

resulting in:

$$\mathbf{h}_{a\perp} = \frac{5p_a}{2q_a B^2} \mathbf{B} \times \nabla T_a. \quad (36)$$

Taking the parallel component of (33,35), the parallel gradients of pressure and temperature are zero

$$\partial_{\parallel} p_a = 0, \quad (37)$$

$$\partial_{\parallel} T_a = 0, \quad (38)$$

where $\partial_{\parallel} = \mathbf{b} \cdot \nabla$. Since magnetic field lines lie on flux surfaces, pressure $p(\psi)$ and temperature $T(\psi)$ are constant on a flux surface. It is notable that the perpendicular heat flux, as described in equation (36) is not related to the collisionality of the plasma. In this work, we are interested in calculating the parallel component of closure quantities. The parallel closures are strongly related to the collisionality of the plasma.

Since $\mathbf{V} = \mathbf{V}_{\parallel} + \mathbf{V}_{\perp}$ and $\mathbf{h} = \mathbf{h}_{\parallel} + \mathbf{h}_{\perp}$, it is useful to calculate $\nabla \cdot \mathbf{V}_{\perp}$, and $\nabla \cdot \mathbf{h}_{\perp}$ for (2-3) in the axisymmetric magnetic field (29):

$$\nabla \cdot \mathbf{V}_{\perp} = \nabla \cdot \left(\frac{\mathbf{B} \times \nabla p}{nqB^2} \right), \quad (39)$$

$$= \nabla \cdot \left(\frac{\mathbf{B} \times \nabla \psi}{nqB^2} \frac{dp}{d\psi} \right). \quad (40)$$

Using (29), one can derive another form of the equation (40):

$$\mathbf{B} \times \nabla \psi = I \nabla \varphi \times \nabla \psi + (\nabla \varphi \times \nabla \psi) \times \nabla \psi, \quad (41)$$

$$= I \nabla \varphi \times \nabla \psi - \nabla \varphi |\nabla \psi|^2, \quad (42)$$

where $|\nabla \psi|^2$ can be expressed using the magnitude of \mathbf{B} :

$$\mathbf{B} \cdot \mathbf{B} = (I \nabla \varphi + \nabla \varphi \times \nabla \psi) \cdot (I \nabla \varphi + \nabla \varphi \times \nabla \psi), \quad (43)$$

$$= I^2 (\nabla \varphi)^2 + |\nabla \varphi|^2 |\nabla \psi|^2, \quad (44)$$

$$|\nabla \psi|^2 = R^2 B^2 - I^2. \quad (45)$$

Using (42) and (45), (40) becomes:

$$\frac{\mathbf{B} \times \nabla \psi}{B^2} = \frac{I}{B} \mathbf{b} - R \hat{\phi}, \quad (46)$$

$$\nabla \cdot \mathbf{b} = -\partial_{\parallel} \ln B, \quad (47)$$

$$\nabla \cdot \left(\frac{\mathbf{B} \times \nabla \psi}{nqB^2} \frac{dp}{d\psi} \right) = \nabla \cdot \left[\frac{1}{nq} \left(\frac{I \hat{\mathbf{b}}}{B} - R \hat{\phi} \right) \frac{dp}{d\psi} \right], \quad (48)$$

$$\nabla \cdot \mathbf{V}_{\perp} = -\frac{2I}{nqB} \partial_{\parallel} \ln B \frac{dp}{d\psi}. \quad (49)$$

In a similar manner, the divergence of the perpendicular heat flux becomes:

$$\nabla \cdot \mathbf{h}_{\perp} = -\frac{5pI}{qB} \partial_{\parallel} \ln B \frac{dT}{d\psi}. \quad (50)$$

Drift kinetic equation

In a strong magnetic field, the gyro-frequency of each charged species $\Omega_a = q_a B / m_a$ is significantly faster than other frequencies in the system. To derive the DKE, one can average the kinetic equation (1) over the rapidly changing gyroangle using new coordinates for the phase space $(x, y, z, U, \mu, \gamma)$. Here, $U = 1/2mv^2 + q\Phi$, where Φ is the electric potential, $\mu = mv_{\perp}^2/2B$ is the magnetic moment and γ is the gyroangle in the velocity space, which will be averaged out. By introducing a small parameter δ which represents the ratio of another frequency to Ω_a , we express the distribution function expansion as:

$$f = f_0 + \delta f_1 + \delta^2 f_2 + \dots, \quad (51)$$

to solve the gyro-averaged kinetic equation with the perturbation method. The gyro-averaging operator is defined as

$$\bar{A}(\mathbf{v}) \equiv \frac{1}{2\pi} \int_0^{2\pi} A(\mathbf{v}) d\gamma. \quad (52)$$

The zeroth-order solution of the distribution function \bar{f}_0 is the Maxwellian distribution function,

$$\bar{f}_0(\psi, s) = \frac{n_0}{\pi^{3/2} v_0^3} e^{-s^2}, \quad (53)$$

where $s = \frac{v}{v_0}$. Note that f_0 is independent to the gyro-angle and $f_0 = \bar{f}_0$.

Previous studies have employed the perturbation method to solve the δ^1 -order equation as follows:

$$\mathbf{v}_{\parallel} \cdot \nabla \bar{f}_1 + \mathbf{v}_D \cdot \nabla f_0 = C(\bar{f}_1), \quad (54)$$

where the drift velocity \mathbf{v}_D is defined by (Hazelton & Meiss, 2003):

$$\mathbf{v}_D = \frac{1}{\Omega} \mathbf{b} \times \left[\left(v_{\parallel}^2 + \frac{1}{2} v_{\perp}^2 \right) \frac{\nabla B}{B} + \frac{q}{m} \nabla \Phi \right] + \frac{v_{\parallel}^2}{\Omega} \frac{\mu_0 \mathbf{J}_{\perp}}{B}, \quad (55)$$

with Φ representing the electric potential and \mathbf{J} indicating the current density. The $\mathbf{v}_D \cdot \nabla f_0$ term in (54) can be expressed by using the relation as below:

$$U = \frac{1}{2} m v_{\parallel}^2 + \mu B + q \Phi, \quad (56)$$

$$m v_{\parallel} \nabla (v_{\parallel})_{U, \mu} = -\mu (\nabla B)_{U, \mu} - q (\nabla \Phi)_{U, \mu}, \quad (57)$$

where the subscript U, μ denotes the partial differentiation at fixed U and μ . With equation (57), $\mathbf{v}_D \cdot \nabla f_0$ is written as:

$$\mathbf{v}_D \cdot \nabla f_0 = (\mathbf{v}_D \cdot \nabla \psi) \frac{df_0}{d\psi}, \quad (58)$$

$$\mathbf{v}_D = v_{\parallel} \nabla \left(\frac{v_{\parallel}}{\Omega} \right)_{U, \mu} \times \mathbf{b} + \frac{v_{\parallel}^2}{\Omega} \frac{\mu_0 \mathbf{J}_{\perp}}{B}, \quad (59)$$

$$\mathbf{v}_D \cdot \nabla \psi = v_{\parallel} \left[\nabla \left(\frac{v_{\parallel}}{\Omega} \right)_{U, \mu} \times \mathbf{b} \right] \cdot \nabla \psi + \frac{v_{\parallel}^2}{\Omega} \frac{\mu_0 (\mathbf{J}_{\perp} \cdot \nabla \psi)}{B}, \quad (60)$$

$$= v_{\parallel} (\mathbf{b} \times \nabla \psi) \cdot \nabla \left(\frac{v_{\parallel}}{\Omega} \right)_{U, \mu}. \quad (61)$$

Using the relation (46), $\mathbf{v}_D \cdot \nabla \psi$ becomes,

$$\mathbf{v}_D \cdot \nabla \psi = v_{\parallel} \partial_{\parallel} \left(\frac{I v_{\parallel}}{\Omega} \right). \quad (62)$$

The final form of the first-order ion drift kinetic equation is expressed as

$$v_{\parallel} \partial_{\parallel} (\bar{f}_1 - F)_{\mu, w} = C (\bar{f}_1), \quad (63)$$

where we introduce a new coordinate $(x, y, z, w, \mu, \gamma)$ and $w = U - q\Phi$. The neoclassical driving term F is given by

$$F = -\frac{I v_{\parallel}}{\Omega} \frac{df_0}{d\psi} = -\frac{I v_{\parallel}}{\Omega} \left[\frac{d \ln p_0}{d\psi} + \left(s^2 - \frac{5}{2} \right) \frac{d \ln T_0}{d\psi} \right] f_0, \quad (64)$$

with p_0, T_0 the equilibrium pressure and temperature. In this work, ion-electron collisions are ignored when calculating the ion parallel closures.

Parallel moment equations

This section presents an overview of the moment approach employed to solve Eq. (63), introduced in (Ji et al., 2023). In this approach, the first order distribution function \bar{f}_1 is expanded in terms of the orthogonal velocity polynomials $\hat{P}^{lk}(v_{\parallel}, v)$

$$\bar{f}_1(t, \mathbf{x}, \mathbf{v}) = f_0 \sum_{lk} \hat{P}^{lk}(v_{\parallel}, v) \hat{M}^{lk}(t, \mathbf{x}). \quad (65)$$

The polynomial $\hat{P}^{lk}(v_{\parallel}, v)$ is given by

$$\hat{P}^{lk}(v_{\parallel}, v) = \frac{1}{\sqrt{\bar{\sigma}_{lk}}} P^{lk}, \quad (66)$$

$$P^{lk}(v_{\parallel}, v) = s^l P_l(\xi) L_k^{(l+1/2)}(s^2), \quad (67)$$

where $P_l(\xi)$ is a Legendre polynomial of the variable $\xi = v_{\parallel}/v$, and $L_k^{(l+1/2)}(s^2)$ is an associated Laguerre polynomial of the variable $s = v/v_0$. The normalization constants $\bar{\sigma}_{lk}$ are

(j, p)	\hat{M}^{jp}	Fluid quantity
(0, 0)	$\frac{n_1}{n_0}$	Number density
(0, 1)	$-\sqrt{\frac{3}{2}} \frac{T_1}{T_0}$	Temperature
(1, 0)	$\frac{\sqrt{2}u}{v_0}$	Parallel flow velocity ($u = V_{1\parallel}$)
(1, 1)	$-\sqrt{\frac{4}{5}} \frac{h_{\parallel}}{p_0 v_0}$	Parallel heat flux
(2, 0)	$\sqrt{\frac{3}{4}} \frac{\pi_{\parallel}}{p_0}$	Parallel viscosity

Table 1: Several lowest order moments

introduced through the orthogonality relations of the polynomials

$$\bar{\sigma}_{lk} = \bar{\sigma}_l \lambda_{lk}, \quad (68)$$

where $\bar{\sigma}_l = \frac{1}{2l+1}$ and $\lambda_{lk} = \frac{(l+k+1/2)!}{k!(1/2)!}$. Here, $\bar{\sigma}_l$ and λ_{lk} denote the normalization constants of P_l and $L_k^{(l+1/2)}$ polynomials as

$$\int_{-1}^1 d\xi P_j(\xi) P_l(\xi) = 2\bar{\sigma}_j \delta_{jl}, \quad (69)$$

$$\frac{1}{\pi^{3/2} v_0^3} \int d\mathbf{v} e^{-s^2} s^{2j} L_p^{j+1/2} L_k^{j+1/2} = \delta_{pk} \lambda_p^j. \quad (70)$$

The coefficient of the expansion \hat{M}^{lk} is called the parallel moment. Several lowest order moments are listed in the Table 1. The (0, 0), (0, 1), and (1, 0) moments are called the fluid moments.

The neoclassical drive in (63) can also be expanded in the polynomials as

$$v_{\parallel} \partial_{\parallel} \left(\frac{I v_{\parallel}}{\Omega} \right)_{\mu, w} \frac{df_0}{d\psi} = \frac{v_T \partial_{\parallel} \ln B}{B/B_0} f_0 \left[\left(2\hat{P}^{00} - 2\sqrt{\frac{2}{3}} \hat{P}^{01} + \frac{1}{\sqrt{3}} \hat{P}^{20} \right) \hat{p}_{0,\psi} \right. \\ \left. + \left(-5\sqrt{\frac{2}{3}} \hat{P}^{01} + 2\sqrt{\frac{10}{3}} \hat{P}^{02} + \frac{1}{\sqrt{3}} \hat{P}^{20} - \sqrt{\frac{7}{6}} \hat{P}^{21} \right) \hat{T}_{0,\psi} \right] \quad (71)$$

where

$$\hat{p}_{0,\psi} = \frac{I}{qv_0 B_0 n_0} \frac{dp_0}{d\psi}, \quad (72)$$

$$\hat{T}_{0,\psi} = \frac{I}{qv_0 B_0} \frac{dT_0}{d\psi}. \quad (73)$$

By taking the \hat{P}^{jp} moment of (63), we can obtain a parallel moment equation for the (j, p) moment

$$\int d\mathbf{v} \hat{P}^{jp} (v_{\parallel} \partial_{\parallel} \bar{f}_1 - v_{\parallel} \partial_{\parallel} F) = \int d\mathbf{v} \hat{P}^{jp} C(\bar{f}_1), \quad (74)$$

$$\sum_{lk} \psi^{jp, lk} \partial_{\parallel} \hat{M}^{lk} + \psi_B^{jp, lk} (\partial_{\parallel} \ln B) \hat{M}^{lk} = \frac{1}{\lambda_C} c^{jp, lk} \hat{M}^{lk} + \frac{\partial_{\parallel} \ln B}{B/B_0} (g_p^{jp} \hat{p}_{0,\psi} + g_T^{jp} \hat{T}_{0,\psi}), \quad (75)$$

where $\lambda_C = v_0 \tau_{ii}$ denotes the ion mean free path and τ_{ii} is the ion collision time. The constant coefficients $\psi^{jp, lk}$, $\psi_B^{jp, lk}$, $c^{jp, lk}$ (Ji & Held, 2014) are defined by

$$\int d\mathbf{v} v_{\parallel} \hat{P}^{jp} \hat{P}^{lk} f_0 = n_0 v_0 \psi^{jp, lk}, \quad (76)$$

$$\int d\mathbf{v} v_{\parallel} \hat{P}^{jp} (\partial_{\parallel} \hat{P}^{lk}) f_0 = n_0 v_0 (\partial_{\parallel} \ln B) \psi_B^{jp, lk}, \quad (77)$$

$$\int d\mathbf{v} v_{\parallel} \hat{P}^{jp} C(f_0 \hat{P}^{lk}) = \frac{n_0}{\tau_{ii}} \delta_{jl} c_{pk}^j. \quad (78)$$

Note that the collision coefficients c_{pk}^j are applied to \hat{M}^{jk} moments which share the same j index. The coefficients $\psi^{jp, lk}$ and $\psi_B^{jp, lk}$ are applied to \hat{M}^{lk} moments which have $l = j + 1$ or $l = j - 1$. Nonvanishing coefficients g^{jp} for the neoclassical drive are

$$g_p^{00} = 2, g_p^{01} = -2\sqrt{\frac{2}{3}}, g_p^{20} = \frac{1}{\sqrt{3}} \quad (79)$$

and

$$g_T^{01} = -5\sqrt{\frac{2}{3}}, g_T^{02} = 2\sqrt{\frac{10}{3}}, g_T^{20} = \frac{1}{\sqrt{3}}, g_T^{21} = -\sqrt{\frac{7}{6}}. \quad (80)$$

A system of equations for $j = 0, 1, \dots, L - 1$ and $p = 0, 1, \dots, K - 1$ as given by (75) can be expressed in matrix form as:

$$[\psi] \partial_{\parallel} [\hat{M}] + [\psi_B] (\partial_{\parallel} \ln B) [\hat{M}] = \frac{1}{\lambda_C} [c] [\hat{M}] + (\partial_{\parallel} \ln B / B / B_0) [g_p \hat{p}_{0,\psi} + g_T \hat{T}_{0,\psi}]. \quad (81)$$

Equation (81) has a solution that depends on the dimensions of the system (L, K) . The validity of this solution can be verified by checking the convergence of solutions with an increasing size of the system. The converged solution can be considered as the solution of (63).

Obtaining ion closure relations

In this work, we implement a set of moment equations to obtain ion parallel closures for fluid equations in the NIMROD code. For this purpose, the rows and columns corresponding to fluid moments in (81) must be removed. When some rows and columns are removed from a system of equations, the corresponding columns become drives in the system. For example, when we remove the first row and column of (82), the equivalent system should be (83):

$$\begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0N} \\ \mathbf{a_{10}} & \mathbf{a_{11}} & & \\ \vdots & & \ddots & \\ \mathbf{a_{N0}} & & & \mathbf{a_{NN}} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_N \end{bmatrix}, \quad (82)$$

$$\begin{bmatrix} \mathbf{a_{11}} & & \\ & \ddots & \\ & & \mathbf{a_{NN}} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix} - x_0 \begin{bmatrix} \mathbf{a_{10}} \\ \vdots \\ \mathbf{a_{N0}} \end{bmatrix}. \quad (83)$$

Therefore, the system of equations for obtaining closure relations can be expressed as follows:

$$\left[\psi' \partial_{\parallel} + \psi'_B (\partial_{\parallel} \ln B) - \frac{1}{\lambda_C} c' \right] [\hat{M}'] = [g_{\parallel}] + \left[(g'_p \hat{p}_0 + g'_T \hat{T}_0) (B_0 / B) (\partial_{\parallel} \ln B) \right] \quad (84)$$

where $'$ denotes the removal of rows and columns of fluid moments from their original form

in (111). Only rows correspond to (1, 1) and (2, 0) moments in the driving term g_{\parallel} are nonvanishing as

$$g_{\parallel}^{1,1} = \frac{\sqrt{5}}{2} \partial_{\parallel} \hat{T}, \quad (85)$$

$$g_{\parallel}^{2,0} = -\frac{\sqrt{3}}{2} W_{\parallel}, W_{\parallel} = \frac{4}{3} \left(\partial_{\parallel} + \frac{3}{2} \partial_{\parallel} \ln B \right) \hat{u} \quad (86)$$

where $\hat{T} = T_1/T_0$ and $\hat{u} = u/v_0$.

IMPLEMENTING PARALLEL MOMENT EQUATIONS USING FINITE ELEMENT METHOD

Finite Element Method

The finite element method (FEM) is a powerful numerical technique widely used in engineering and mathematical modeling to approximate solutions to complex problems. It is closely linked to the method of weighted residuals, a fundamental concept that minimizes the residuals of differential equations across the entire domain to enhance the accuracy of approximations. A popular variant of this is the Galerkin method, which specifically chooses weight functions that are the same as the trial functions in the finite element formulation. This selection simplifies the integration process and often enhances the stability and convergence of the solution, making the Galerkin method a preferred approach in many FEM applications (Karniadakis & Sherwin, 2005). In this chapter, we introduce the Galerkin method as used in NIMROD and discuss our results from implementing the parallel moment equations using this method.

Basis functions

In this study, we use the NIMROD code to solve (75) and the plasma fluid equations on a grid using the finite element method. A scalar physical field on the grid is represented by using polynomial basis functions $\alpha_i(R, Z)$,

$$A(R, Z) = \sum_i A_i \alpha_i(R, Z). \quad (87)$$

In (87), the i index is summed over the nodes of the grid, and the α_i are referred to as trial basis functions. Multiplying a differential equation by test basis functions and integrating over the domain converts the differential equation to a matrix equation for the coefficients of the trial basis functions. If the trial and test basis functions are chosen from the same set,

as they are in NIMROD, the finite element method is referred to as the Galerkin approach. For example, to solve a simple equation for $A(R, Z)$

$$A = \frac{\partial f(R, Z)}{\partial R}, \quad (88)$$

we need to multiply test basis functions α_j to the equation and integrate it over the entire domain

$$\int dRdZ \alpha_j \sum_i A_i \alpha_i = \int dRdZ \alpha_j \frac{\partial}{\partial R} \sum_i (f_i \alpha_i). \quad (89)$$

Here the index j runs over the entire set of basis functions. Equation (88) converts to an matrix equation for coefficients A_i such that

$$\sum_i \left(\int dRdZ \alpha_j \alpha_i \right) A_i = \int dRdZ \alpha_j \frac{\partial}{\partial R} \sum_i (f_i \alpha_i), \quad (90)$$

$$M_{ji} A_i = F_j. \quad (91)$$

A two-dimensional basis function $\alpha_i(R, Z)$ consists of two one-dimensional basis functions. For example, a linear one-dimensional basis function $\alpha_i(x)$ is shown in (92)

$$\alpha_i(x) = \begin{cases} 0 & x < x_{i-1} \\ \frac{x-x_{i-1}}{x_i-x_{i-1}} & x_{i-1} \leq x \leq x_i \\ \frac{x_{i+1}-x}{x_{i+1}-x_i} & x_i \leq x \leq x_{i+1} \\ 0 & x_{i+1} < x \end{cases} \quad (92)$$

where x_i denotes the i^{th} node of the one-dimension grid. Higher order basis functions are defined not only on a vertex x_i , but also at interior points of a finite element cell. A quadratic

basis function, as shown in Fig. 4, can have two types $\alpha_i(x)$ and $\beta_i(x)$ where

$$\alpha_i(x) = \begin{cases} 0 & x < x_{i-1} \\ \frac{(x-x_{i-1})(x-x_{i-\frac{1}{2}})}{(x_i-x_{i-1})(x_i-x_{i-\frac{1}{2}})} & x_{i-1} \leq x \leq x_i \\ \frac{(x-x_{i+1})(x-x_{i+\frac{1}{2}})}{(x_i-x_{i+1})(x_i-x_{i+\frac{1}{2}})} & x_i \leq x \leq x_{i+1} \\ 0 & x_{i+1} < x, \end{cases} \quad (93)$$

$$\beta_i(x) = \begin{cases} 0 & x \leq x_{i-1} \\ \frac{(x-x_{i-1})(x-x_i)}{(x_{i-\frac{1}{2}}-x_{i-1})(x_{i-\frac{1}{2}}-x_i)} & x_{i-1} \leq x \leq x_i \\ 0 & x_i \leq x. \end{cases} \quad (94)$$

Note that $\beta_i(x)$ has a maximum at the interior point $x = x_{i-\frac{1}{2}}$. Therefore, two-dimensional basis functions can be defined at interior points which are located in grid cells.

In the NIMROD code, the coefficients of basis functions are saved in four different `lagr_quad` (Lagrange quadrilateral) type arrays called `fs`, `fsh`, `fsv`, `fsi`. Fig. 6 and 7 illustrate examples of elements stored in the arrays for the case of quadratic basis functions. The `fs(iv,ix,iy)` array stores coefficients for grid-vertex nodes, where the index `iv` denotes the index of the vector component. The `fsh(iv,ib,ix,iy)` array stores coefficients for horizontal-side nodes. In `fsh`, a basis index `ib` is introduced for pointing basis functions on the horizontal side nodes from left to right. The `fsv(iv,ib,ix,iy)` array stores coefficients for vertical-side nodes from bottom to top. The `fsi(iv,ib,ix,iy)` array stores coefficients for internal nodes from bottom-left to top-right.

Numerical differentiation and integration on the logical coordinates

The physical coordinates (R, Z) which set nodes on a poloidal cross section can be transformed to the logical coordinates (ξ, η) for various conveniences in numerical implementation. Figs. 5-7 depict the physical and logical grids used in this study. The colored

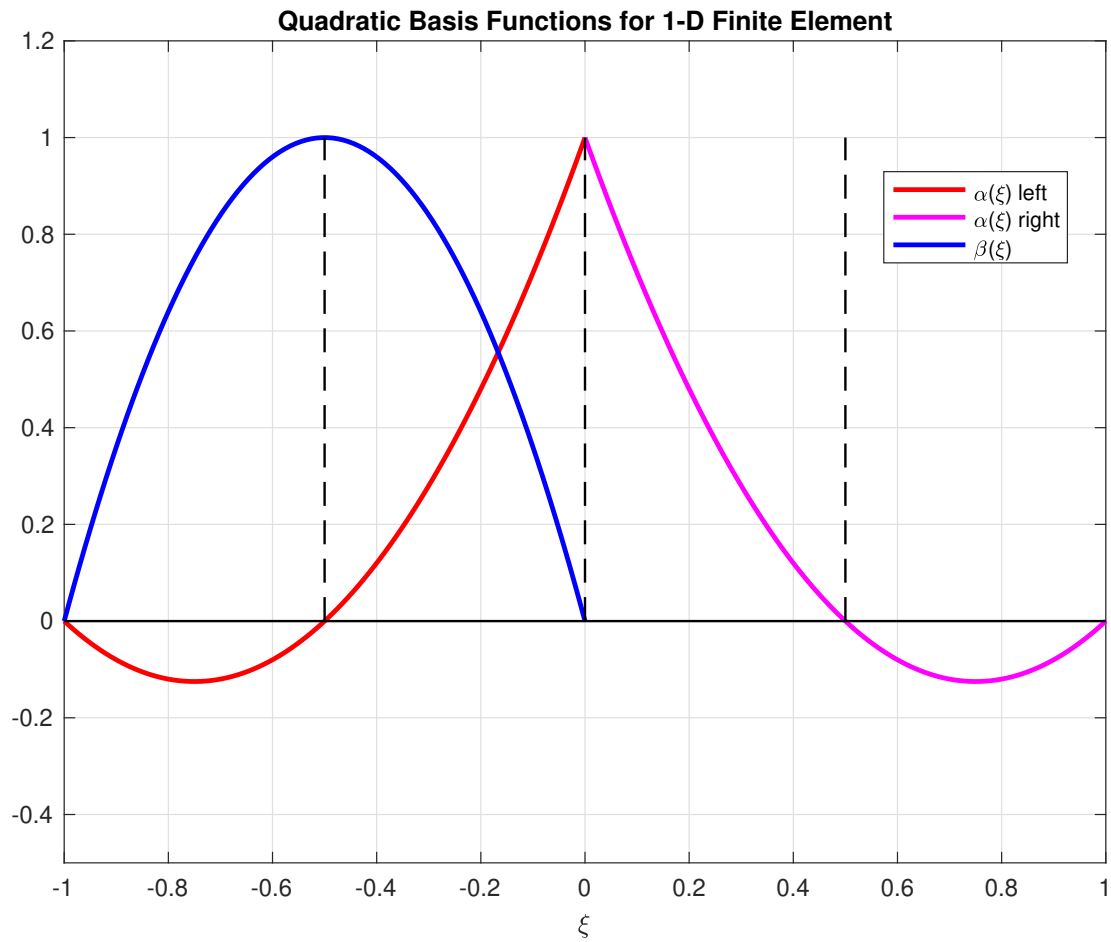


Figure 4: Quadratic basis functions for 1-D finite element with nodes at vertices and interior points.

blocks indicate that the block numbering progresses first in the η (poloidal angle)-direction then ξ (perpendicular to the flux surface)-direction within the logical coordinate system. These sub-grids in the logical coordinates are called the `rblocks` (Sovinec et al., 2003), which are rectangular blocks in the logical coordinates. The Jacobian of this transformation must be calculated to express results in the physical coordinates.

Differentiating a function is performed over the logical coordinates. NIMROD uses the chain rule to calculate differentiation over a physical coordinate. This involves using the derivative of the basis function with respect to the logical coordinates. The equations are given by:

$$\frac{d\alpha}{dR} = \frac{d\xi}{dR} \frac{d\alpha}{d\xi} + \frac{d\eta}{dR} \frac{d\alpha}{d\eta}, \quad (95)$$

$$\frac{d\alpha}{dZ} = \frac{d\xi}{dZ} \frac{d\alpha}{d\xi} + \frac{d\eta}{dZ} \frac{d\alpha}{d\eta}. \quad (96)$$

The metric related derivatives $d\xi/dR$, $d\eta/dR$, $d\xi/dZ$, $d\eta/dZ$ can be calculated from the mapping functions $R(\xi, \eta)$ and $Z(\xi, \eta)$ where R and Z are expanded using the same set of basis functions as the fields. This process is referred to as an iso-parametric mapping of the coordinates.

Integrating a function over the domain is essential for constructing the matrix and the right-hand side as shown in (90). The integration over a specific interval $[x_1, x_2]$ can be approximated by a weighted sum of n function values at Gaussian quadrature points x_i

$$\int_{x_1}^{x_2} f(x) dx \approx \sum_{i=1}^n w_i f(x_i), \quad (97)$$

where the weights w_i are given by the Gauss-Legendre integration method (Press, 2007). Thus, the basis function information at Gaussian quadrature points are stored as an array $\alpha(w_g, x_g, y_g)$ where x_g, y_g denote the Gaussian quadrature points in the logical coordinates and w_g for the weights. The function value at the quadrature point $f(x_i)$ is saved in an `rb_qp_type` (rblock quadrature point type) array.

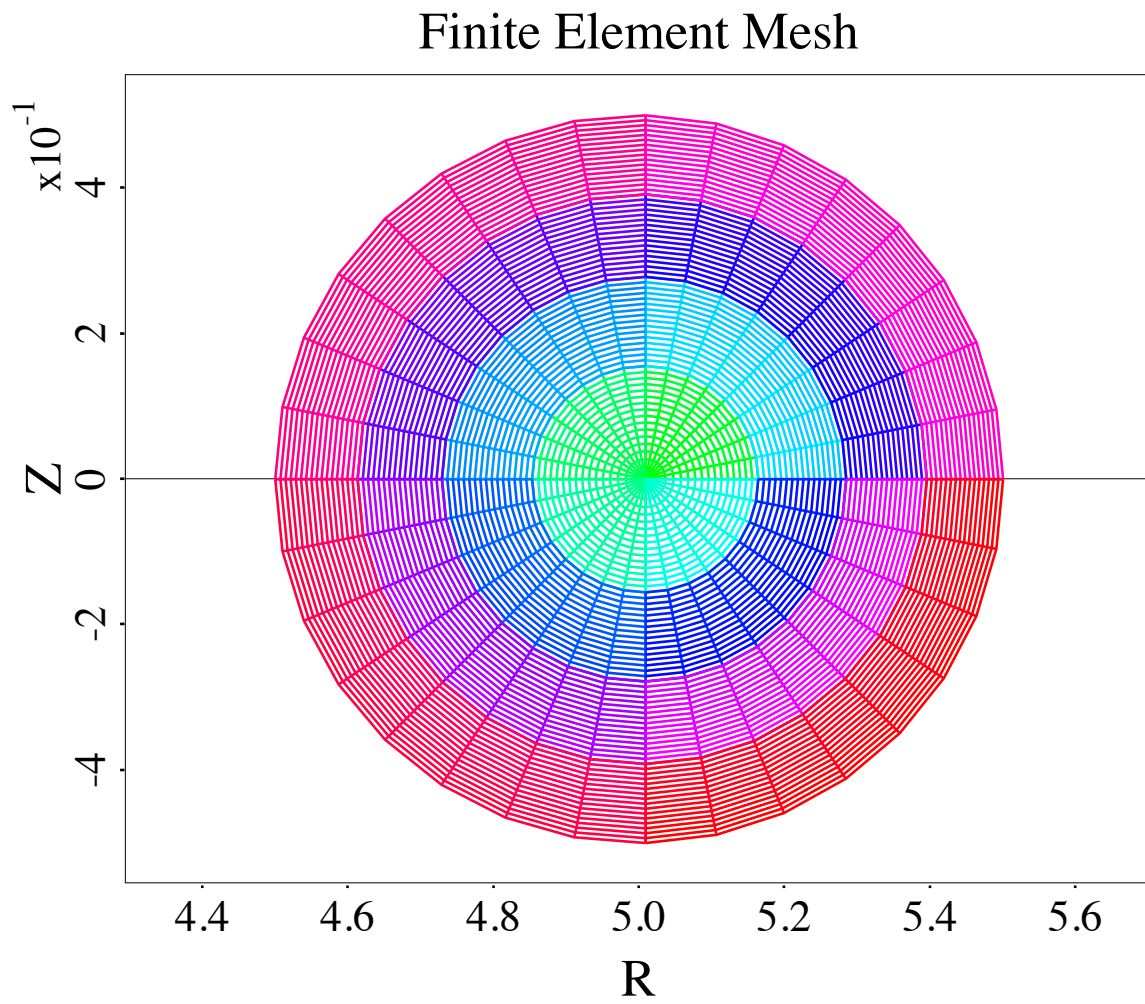


Figure 5: Poloidal mesh (physical grid) used in this study shows the 2-D finite element cells in the R - Z plane.

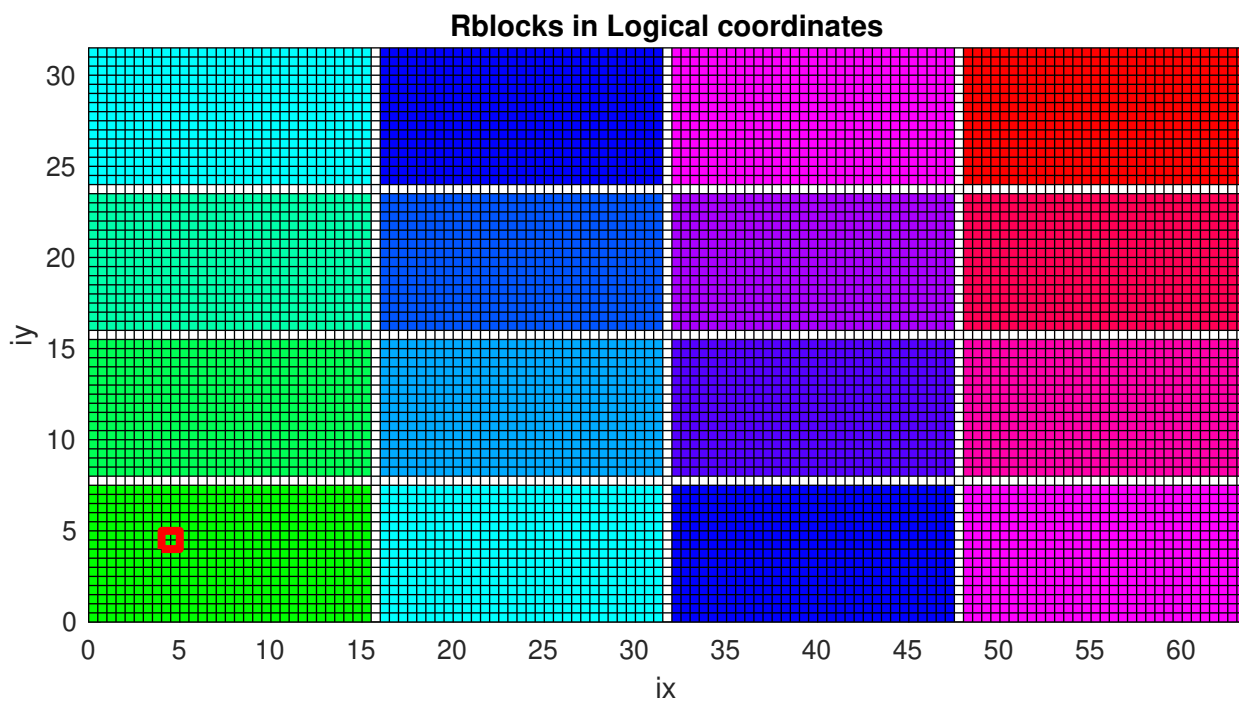


Figure 6: Logical coordinates used in this study with ix (ξ), moving across constant- ψ surfaces, and iy (η), moving around in the periodic, poloidal angle.

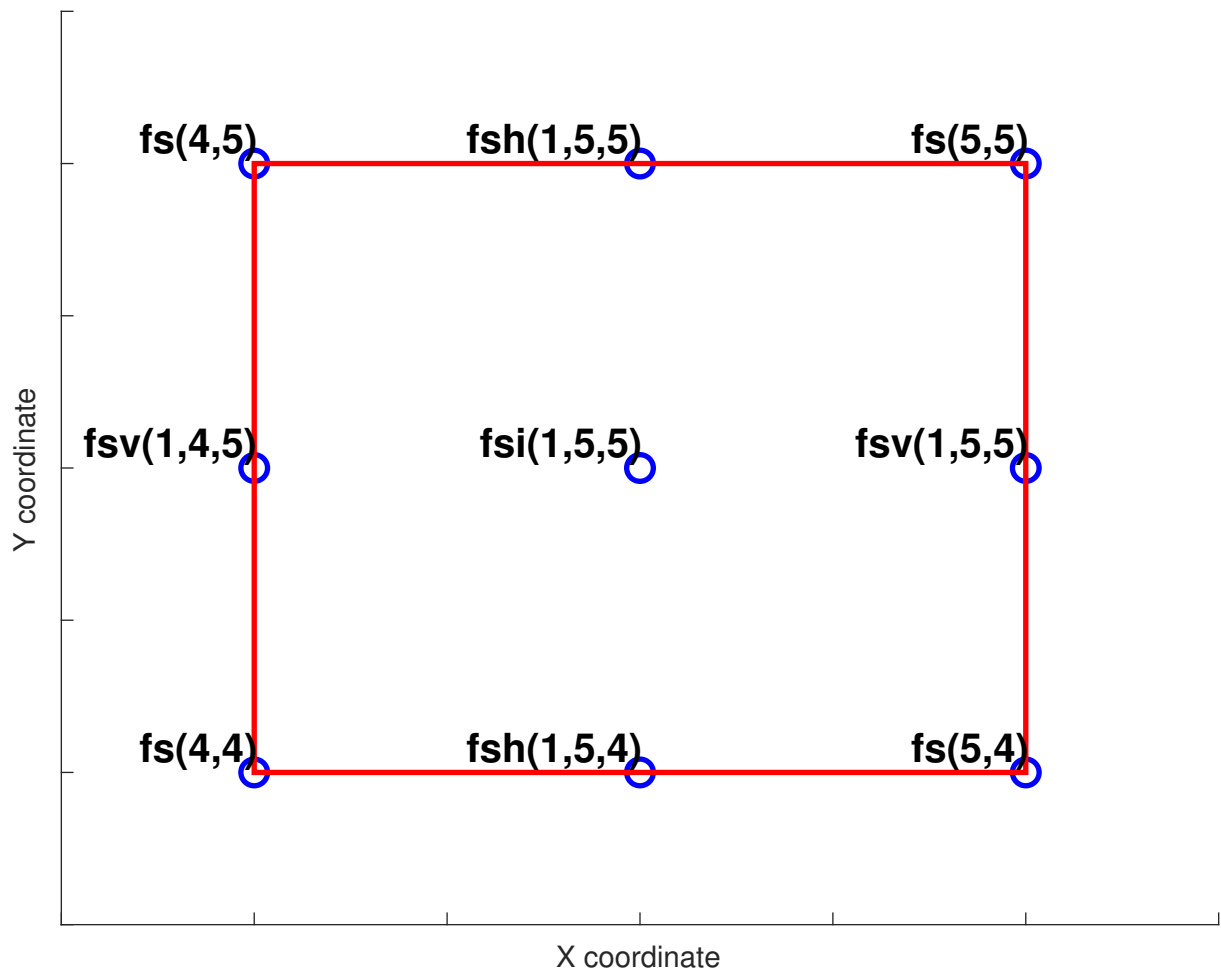


Figure 7: $lagr_quad$ type arrays over the red square on Fig. 6.

Implementing parallel moment equations using finite element method

In the system of equation (84), the moment variable \hat{M}'^{lk} is expanded using the basis functions

$$\hat{M}'^{lk}(R, Z) = \sum_i \hat{M}'_i{}^{lk} \alpha_i(R, Z). \quad (98)$$

The dimensions of the variable can be determined by the input parameters `numL`, `numK` where the index l runs from 0 to `numL` - 1 and the k index is defined as follows:

$$k = \begin{cases} 2, 3, \dots, \text{numK} + 1 & l = 0, \\ 1, 2, \dots, \text{numK} & l = 1, \\ 0, 1, \dots, \text{numK} - 1 & l > 1. \end{cases}$$

The operator terms on the left side are constructed by multiplying a test basis function α_j into the equation, resulting in:

$$\int dRdZ \alpha_j \left[\psi' \partial_{\parallel} + \psi'_B (\partial_{\parallel} \ln B) - \frac{1}{\lambda_C} c' \right] \sum_i \hat{M}'_i{}^{lk} \alpha_i(R, Z). \quad (99)$$

The dimension of the equation is number of total nodes in the poloidal plane times `numL` times `numK`. The right side of the equation becomes

$$\int dRdZ \alpha_j \sum_i \left[(g_{\parallel})_i + \left(g'_p (\hat{p}_{0,\psi})_i + g'_T (\hat{T}_{0,\psi})_i \right) (B_0/B)_i (\partial_{\parallel} \ln B)_i \right], \quad (100)$$

where A_i denotes the value of physical quantity $A(R, Z)$ at the i^{th} node.

Benchmarking integral closures in slab geometry

In unshered slab geometry, the magnetic field does not change in its direction. Under this condition, where $\partial_{\parallel} \ln B = 0$, the solution of (115) can be expressed as follow

$$\left[\psi' \partial_{\parallel} - \frac{1}{\lambda_C} c' \right] [\hat{M}'] = [g_{\parallel}]. \quad (101)$$

The solution of the equation (101) can be calculated analytically using the eigenvector method, and the closures obtained from the results are referred to as the integral closures (Ji et al., 2017).

Thermodynamic drives in (116,117) are applied in g_{\parallel} without the effect of $\partial_{\parallel} \ln B$. Assuming sinusoidal thermodynamic drives in this system,

$$T = T_0 + T_1 \sin \varphi, \quad (102)$$

$$V_{\parallel} = V_1 \sin \varphi, \quad (103)$$

where $\varphi = 2\pi\ell/\lambda$ and ℓ denotes the length measured along the magnetic field, ion closure quantities can be expressed as follows:

$$h_{\parallel}(\ell) = -\frac{1}{2}nT_1v_0\hat{h}_h \cos \varphi - nT_0V_1\hat{h}_{\pi} \sin \varphi, \quad (104)$$

$$\pi_{\parallel}(\ell) = -nT_1\hat{\pi}_h \sin \varphi - nT_0\frac{V_1}{v_0}\hat{\pi}_{\pi} \cos \varphi. \quad (105)$$

The dimensionless closures \hat{A}_B represent the effect of closure A from drive B , where the drive h denotes the temperature gradient and π denotes the velocity gradient drives. Figs. 8 and 9 show the results of \hat{h}_h and $\hat{\pi}_{\pi}$ calculated by solving (101) in NIMROD. In the graph, the green, blue, and black solid lines depict analytical results derived from solving moment equations of varying sizes. As the normalized mean free path $k = 2\pi\lambda_C/\lambda$ increases, the model requires a greater number of moments to reach a converged solution. At the collisionless limit, an infinite number of moments is required to obtain a convergent solution. This solution can be derived by solving a linearized kinetic equation (Ji et al., 2013). The known asymptotic behavior of these moment solutions is illustrated by a red solid line, incorporating both the moment solutions and their asymptotic tendencies (Ji et al., 2017).

Implementing parallel moment equations in the circular magnetic field

We tested the results of our implementation of (84) in the high-aspect-ratio tokamak grid depicted in Fig. 5 using the finite element method. Fig. 10 illustrates the equilibrium

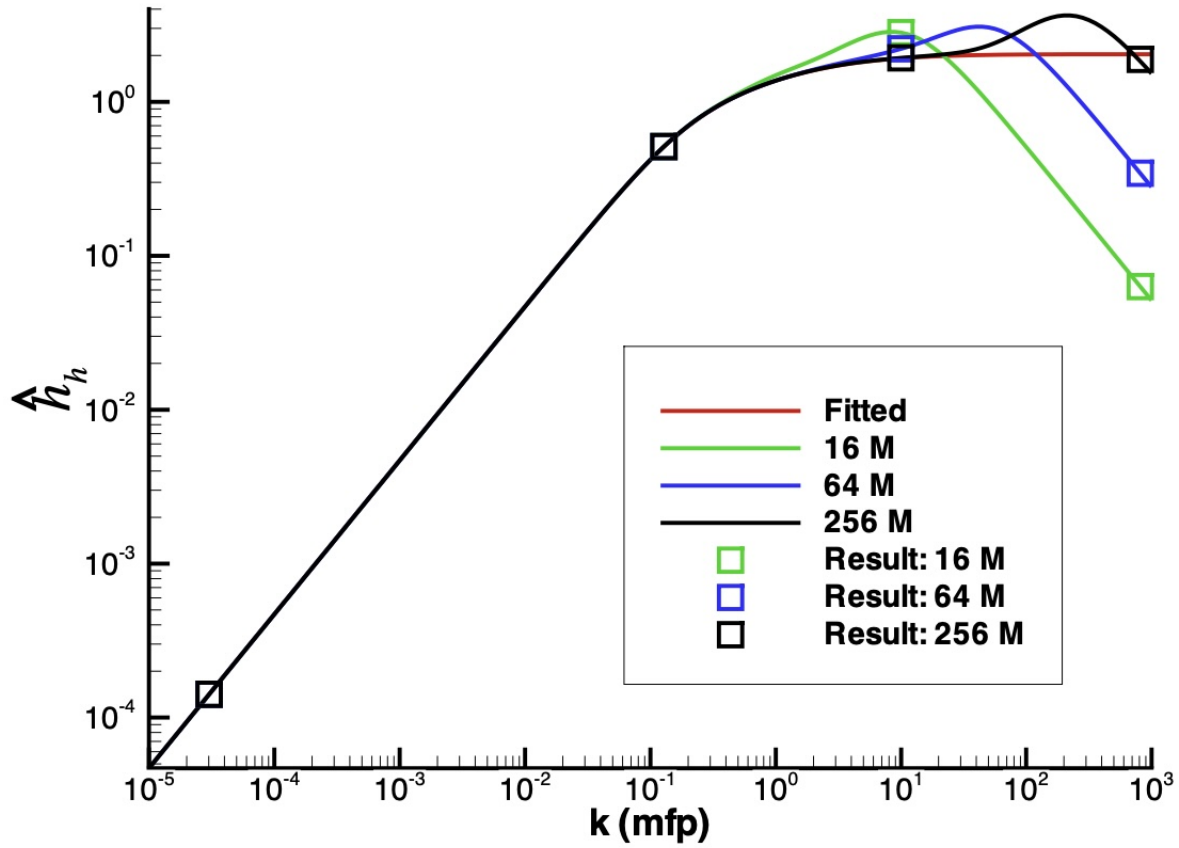


Figure 8: Ion \hat{h}_h closure. The green, blue, black squares represent results calculated in NIMROD by solving the systems with $(\text{numL}, \text{numK}) = (4, 4)$, $(8, 8)$, and $(16, 16)$ moment equations, respectively.

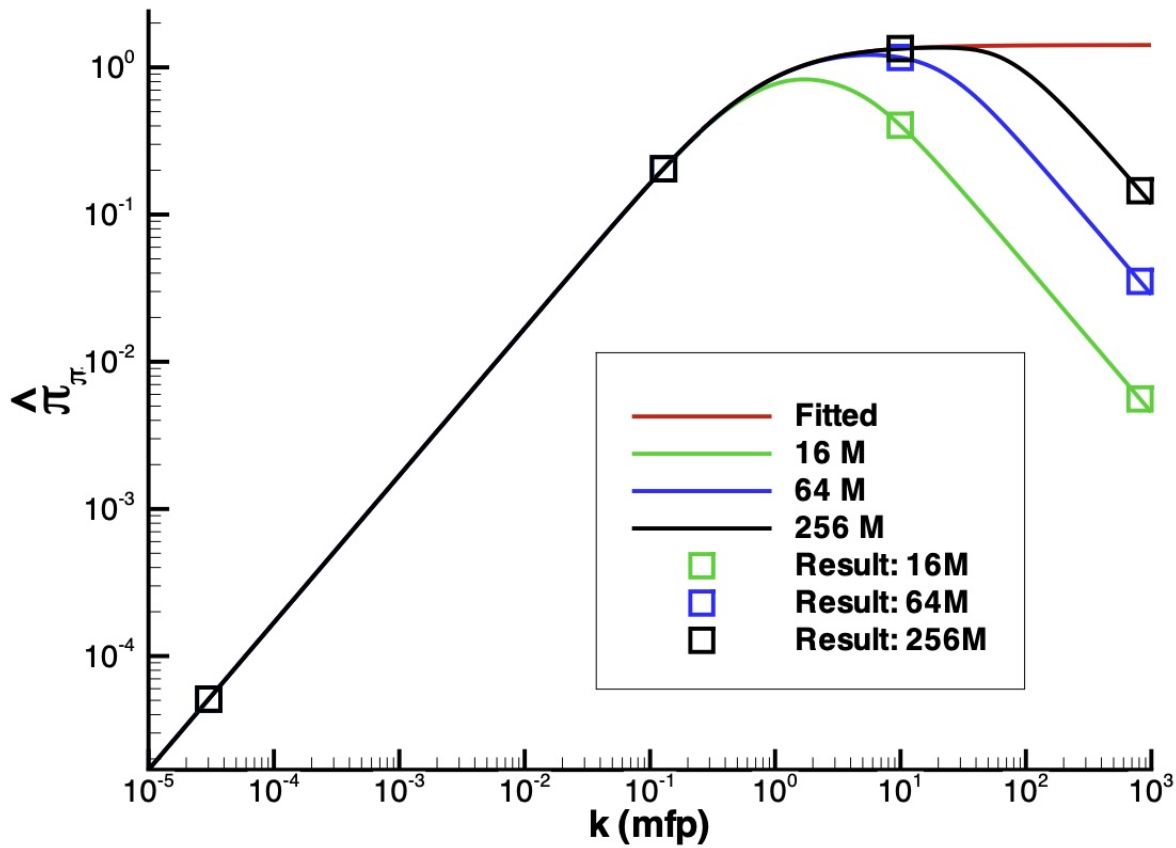


Figure 9: Ion $\hat{\pi}_{\pi}$ closure. The green, blue, black squares represent results calculated in NIMROD by solving the systems with $(\text{numL}, \text{numK}) = (4, 4)$, $(8, 8)$, and $(16, 16)$ moment equations, respectively.

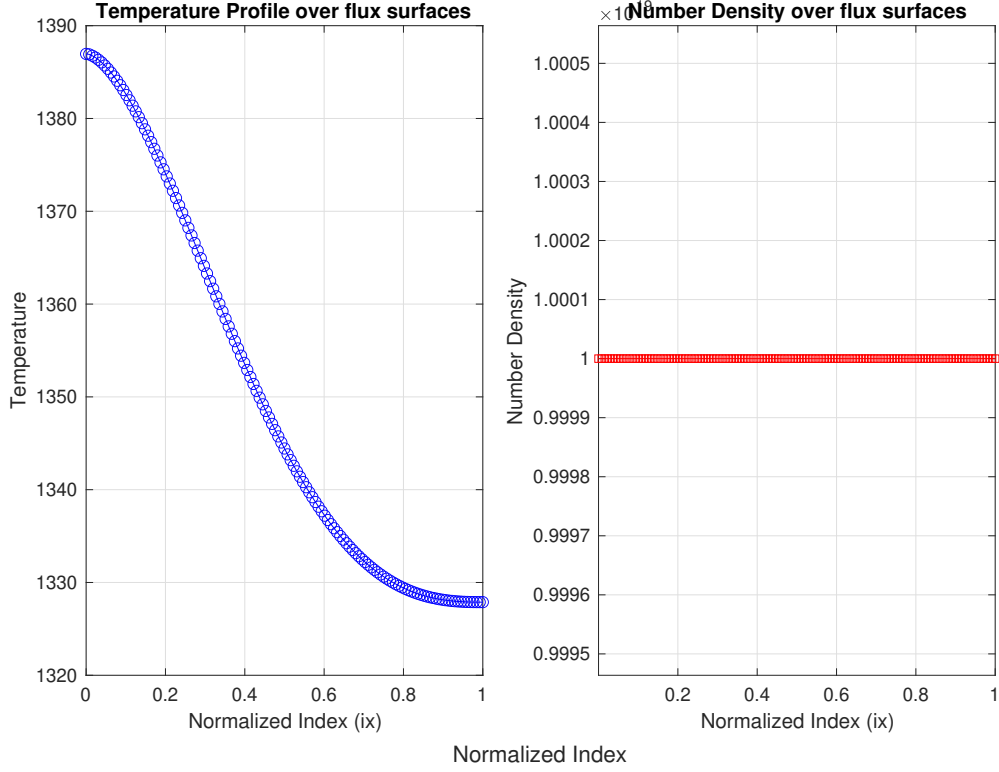


Figure 10: Equilibrium ion temperature and number density across flux surfaces at $\theta = \frac{\pi}{2}$.

profiles for n_0 and T_0 for this simulation. The equilibrium ion velocity is set to zero. Added on the equilibrium quantities, we structured the drives $\partial_{\parallel}T$ and W_{\parallel} as three types: constant, $\cos\theta$ and $\sin\theta$. The outcomes from these simulations were then rigorously compared with calculations performed purely in Fourier space, which will be introduced in the next section. This comparison was essential to validate the FEM simulations against a well-established method, ensuring that the FEM approach produced consistent and reliable results.

It is crucial that the implemented $\partial_{\parallel}T_1$ drive functioned as intended, and was not adversely affected by the equilibrium temperature T_0 . Fig. 11 illustrates the change of T_0 around a flux surface in poloidal angle, θ . To satisfy equation (38), the ion temperature should be constant on a flux surface. When we use the quadratic basis functions, the change of the ion temperature is negligible compared to the case when using linear basis functions.

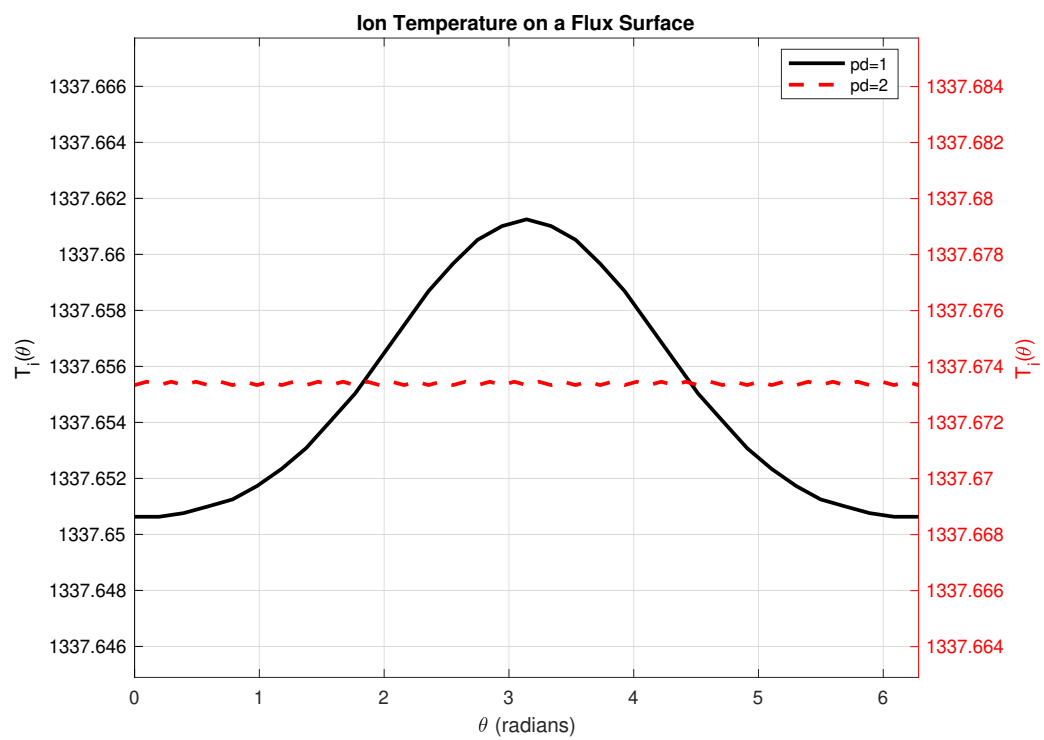


Figure 11: Equilibrium ion temperature around a flux surface calculated by using linear and quadratic basis functions.

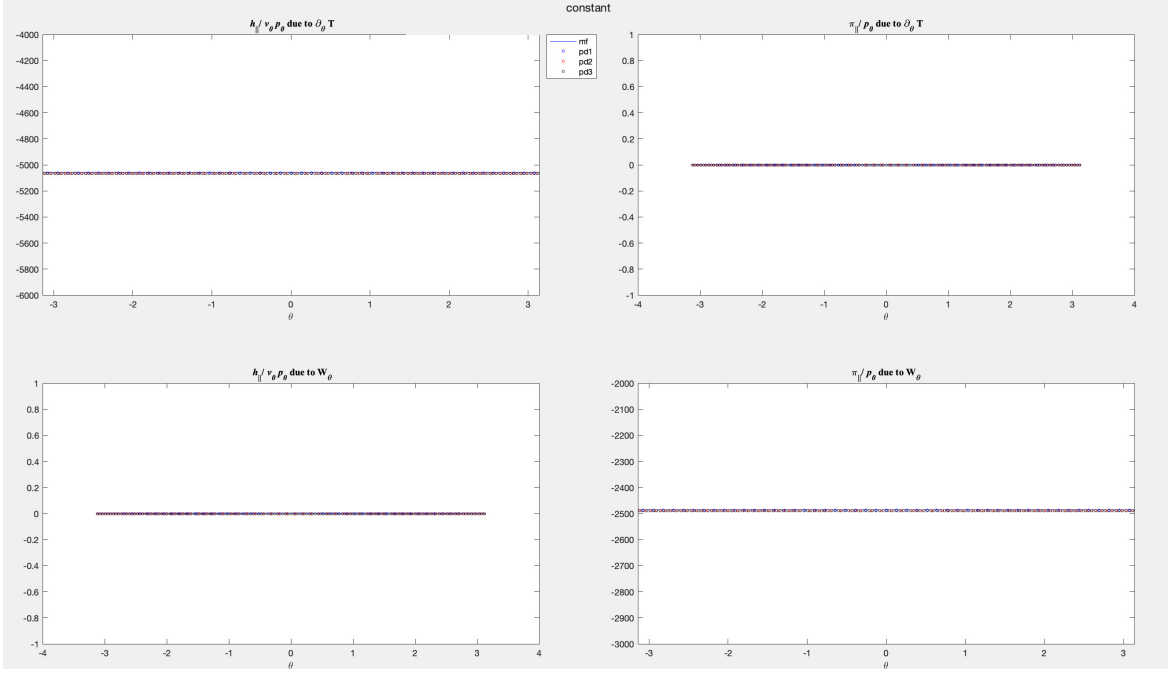


Figure 12: Ion parallel heat flux (left) and viscosity (right) closures driven by constant $\partial_{\parallel}T$ (top) and W_{\parallel} (bottom). Results are calculated by using linear (blue), quadratic (red), and cubic (black) basis functions.

Fig. 12 illustrates the results of parallel ion closures obtained by solving the $(\text{numL}, \text{numK}) = (4, 8)$ system, driven by constant drives. For testing purposes, only the collision operator is included in the moment equations as follows:

$$\left[-\frac{1}{\lambda_C} c' \right] \left[\hat{M}' \right] = [g_{\parallel}]. \quad (106)$$

Since the collision coefficient c'_{pk} is applied to \hat{M}^{jk} moments, the $\partial_{\parallel}T$ drive affects only the heat flux, and the W_{\parallel} drive affects only the parallel viscosity. Due to the long mean free path λ_C in the simulation, a constant effect of the drives can have a significant impact on parallel closures.

Fig. 13 and 14 illustrate the closure results tested by $\sin \theta$ and $\cos \theta$ drives. In this test, both the collision operator and the parallel gradient operators are included in the moment equation as follows:

$$\left[\psi' \partial_{\parallel} - \frac{1}{\lambda_C} c' \right] \left[\hat{M}' \right] = [g_{\parallel}]. \quad (107)$$

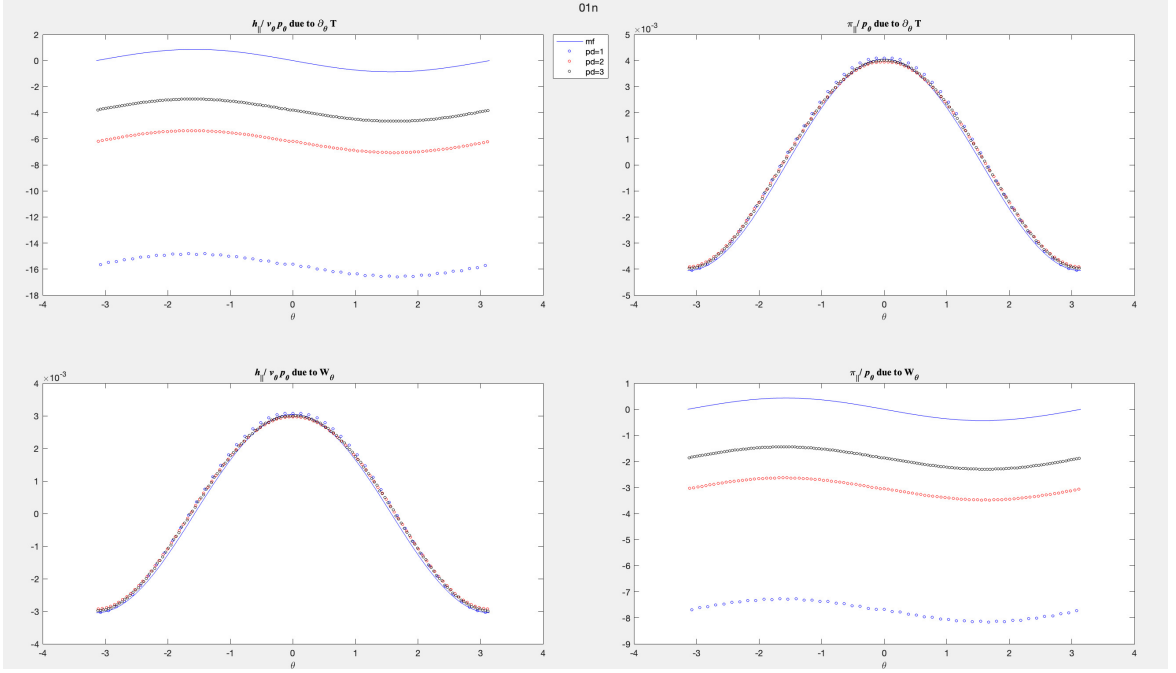


Figure 13: Ion parallel heat flux (left) and viscosity (right) closures driven by $\partial_{\parallel} T = \sin \theta$ (top) and $W_{\parallel} = \sin \theta$ (bottom). Results are calculated by using linear (blue), quadratic (red), and cubic (black) basis functions.

Since the coefficients $\psi^{jp,lk}$ applied to \hat{M}^{lk} moments which have $l = j + 1$ or $l = j - 1$, $\partial_{\parallel} T$ can affect the viscosity and W_{\parallel} can affect the heat flux. As the polynomial degree of the basis functions increases, the results from the finite element method for closures more closely approximate those from solving the moment-Fourier equations. However, using polynomial basis functions of cubic or higher degrees proves inefficient for time-advance simulations, particularly when numerous moments are required to achieve a convergent solution. The next chapter will introduce the moment-Fourier method, which is advantageous for lowering memory costs.

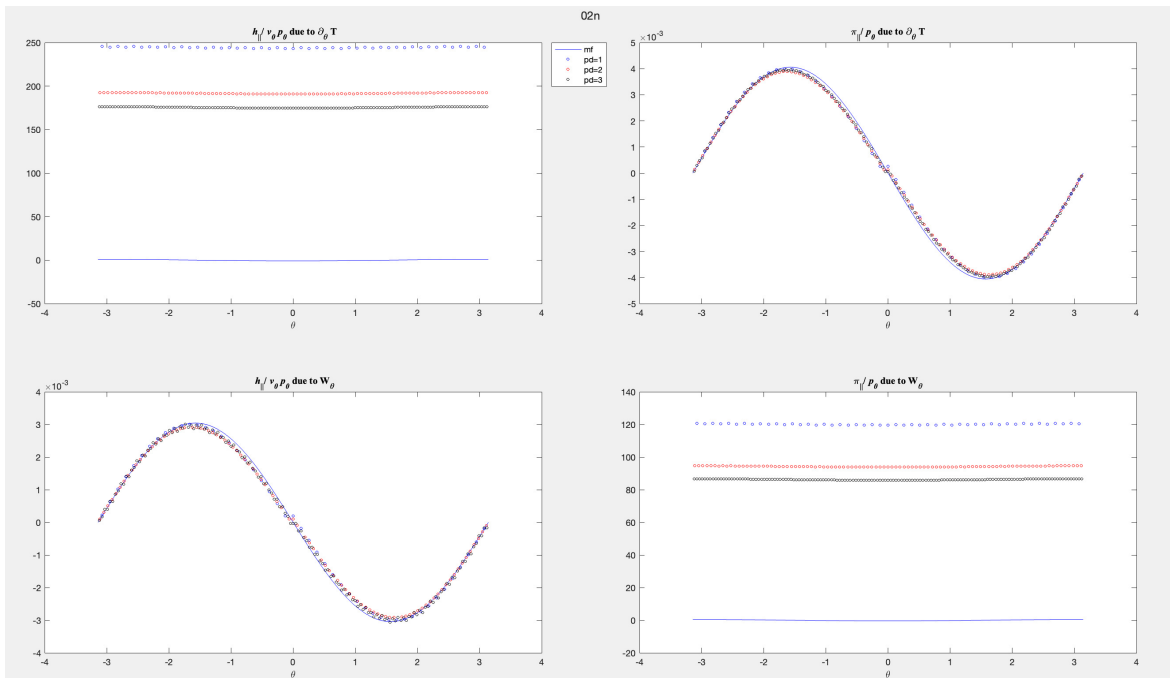


Figure 14: Ion parallel heat flux (left) and viscosity (right) closures driven by $\partial_{\parallel} T = \cos \theta$ (top) and $W_{\parallel} = \cos \theta$ (bottom). Results are calculated by using linear (blue), quadratic (red), and cubic (black) basis functions.

IMPLEMENTING MOMENT-FOURIER EQUATIONS

Fourier method of solving moment equations and obtaining closures

In the circular magnetic field (32), the parallel moment equation (75) can be expressed with respect to the poloidal angle θ . Using the relation $\partial_{\parallel} = \frac{\mathbf{B} \cdot \nabla \theta}{B} \frac{\partial}{\partial \theta}$, (75) can be converted to (108)

$$\sum_{lk} \psi^{jp, lk} \partial_{\theta} \hat{M}^{lk} + \psi_B^{jp, lk} (\partial_{\theta} \ln B) \hat{M}^{lk} = \frac{B}{B^{\theta} \lambda_C} c^{jp, lk} \hat{M}^{lk} + \frac{\partial_{\theta} \ln B}{B/B_0} \left(g_p^{jp} \hat{p}_{0, \psi} + g_T^{jp} \hat{T}_{0, \psi} \right), \quad (108)$$

where $B^{\theta} = \mathbf{B} \cdot \nabla \theta$. Every physical quantity in (108) is periodic with respect to the poloidal angle θ on a given flux surface. Using the Fourier method in the poloidal direction, we can convert the differential equations into an algebraic equation. In the Fourier method, a physical quantity $A(\theta)$ is expanded in Fourier series $A(\theta) = \sum_m A_{(m)} \varphi_{(m)}$, where the basis functions are defined as $\varphi_{(2n-1)} = \sin(n\theta)$ for odd terms, and $\varphi_{(2n)} = \cos(n\theta)$ for even terms. An operator is expressed as matrices in the Fourier representation by

$$O_{(i,j)} = \frac{1}{\sigma_{(i)}} \int d\theta \varphi_{(i)} O \varphi_{(j)}, \quad (109)$$

where $\sigma_{(0)} = 2\pi$ and $\sigma_{(i)} = \pi$ for $i > 0$. Then the parallel moment equation (108) can be expressed as a system of algebraic equations:

$$\psi^{jp, lk} (\partial_{\theta})_{(m,n)} \hat{M}_{(n)}^{lk} + \psi_B^{jp, lk} (\partial_{\theta} \ln B)_{(m,n)} \hat{M}_{(n)}^{lk} = c^{jp, lk} \left(\frac{B}{B^{\theta} \lambda_C} \right)_{(m,n)} \hat{M}_{(n)}^{lk} + \left(\frac{\partial_{\theta} \ln B}{B/B_0} \right)_{(m)} \left(g_p^{jp} \hat{p}_{0, \psi} + g_T^{jp} \hat{T}_{0, \psi} \right). \quad (110)$$

The system of algebraic equations which includes $j = 0, 1, \dots, L-1$ and $p = 0, 1, \dots, K-1$ as described by (110) can be expressed in matrix form,

$$\begin{aligned} \llbracket \psi \partial_\theta \rrbracket \llbracket \hat{M} \rrbracket + \llbracket \psi_B \partial_\theta \ln B \rrbracket \llbracket \hat{M} \rrbracket &= \llbracket c \left(B/B^\theta \lambda_C \right) \rrbracket \llbracket \hat{M} \rrbracket + \\ &\llbracket \left(g_p \hat{p}_0 + g_T \hat{T}_0 \right) (B_0/B) (\partial_\theta \ln B) \rrbracket. \end{aligned} \quad (111)$$

The dimension of the linear system is $\text{numL} \times \text{numK} \times \mathbf{n}_F$, where \mathbf{n}_F denotes the number of Fourier modes. The operators applied to $\llbracket \hat{M} \rrbracket$ are tensor products of the corresponding operators from (81) and (110) such as

$$\llbracket \psi \partial_\theta \rrbracket = [\psi] \otimes (\partial_\theta)_F, \quad (112)$$

$$\llbracket \psi_B \partial_\theta \ln B \rrbracket = [\psi_B] \otimes (\partial_\theta \ln B)_F, \quad (113)$$

$$\llbracket c \left(B/B^\theta \lambda_C \right) \rrbracket = [c] \otimes \left(B/B^\theta \lambda_C \right)_F, \quad (114)$$

where the Fourier matrix $(O)_F$ has a component in the i th row and j th column as described in (109). The structure of the system of equations (111) indicates that the matrix is non-invertible. The solution can be derived using singular value decomposition (Ji et al., 2023).

The system of equations for obtaining closure relations is given by

$$\begin{aligned} \llbracket \psi' \partial_\theta \rrbracket \llbracket \hat{M}' \rrbracket + \llbracket \psi'_B \partial_\theta \ln B \rrbracket \llbracket \hat{M}' \rrbracket &= \llbracket c' \left(B/B^\theta \lambda_C \right) \rrbracket \llbracket \hat{M}' \rrbracket + \llbracket g_\theta \rrbracket \\ &+ \llbracket \left(g'_p \hat{p}_0 + g'_T \hat{T}_0 \right) (B_0/B) (\partial_\theta \ln B) \rrbracket \end{aligned} \quad (115)$$

where $'$ denotes the removal of rows and columns of fluid moments from their original form in (111). The nonvanishing driving terms in g_θ are

$$g_\theta^{1,1} = \frac{\sqrt{5}}{2} \partial_\theta \hat{T}, \quad (116)$$

$$g_\theta^{2,0} = -\frac{\sqrt{3}}{2} W_\theta, \quad W_\theta = \frac{4}{3} \left(\partial_\theta + \frac{3}{2} \partial_\theta \ln B \right) \hat{u}. \quad (117)$$

Obtaining Fourier coefficients of fields for implementing the equations

To solve (115), it is necessary to obtain the Fourier coefficients of certain fields, such as $\partial_\theta \ln B$, $B/B^\theta \lambda_C$, $B_0/B \partial_\theta \ln B$. The magnetic field on a flux surface is used to represent the poloidal angle. To obtain the Fourier coefficients of these fields, the values of each field are sampled at equal magnetic field length intervals.

The `rblock` may be separated in the poloidal angle direction for a simulation. In this case, a single processor cannot obtain the Fourier coefficients of the fields for the entire poloidal angle region from 0 to 2π . Therefore, the fields needed to calculate the Fourier coefficients must be gathered from the separated `rblock` values into one block. The collected fields in this mapping process are stored in the global `rb_cel` block.

After obtaining the field values along the same magnetic field interval, they are subjected to a Fast Fourier Transform (FFT) to obtain the Fourier coefficients. Since the FFT library provides coefficients of $\exp[i\theta]$, it is converted to represent the coefficients for $\sin n\theta$ and $\cos n\theta$.

Obtaining the closure matrix

To solve (115) on a given flux surface `ix`, we employ LU decomposition. Equation (118) illustrates an example of decomposing the system of equations when `numL` = 4:

$$\begin{bmatrix} -C^3 & \Psi^{3-} & 0 & 0 \\ \Psi^{2+} & -C^2 & \Psi^{2-} & 0 \\ 0 & \Psi^{1+} & -C^1 & \Psi^{1-} \\ 0 & 0 & \Psi^{0+} & -C^0 \end{bmatrix} = \begin{bmatrix} R_3 & & & \\ \Psi^{2+} & R_2 & & \\ & \Psi^{1+} & R_1 & \\ & & \Psi^{0+} & R_0 \end{bmatrix} \begin{bmatrix} 1 & A_3 & & \\ & 1 & A_2 & \\ & & 1 & A_1 \\ & & & 1 \end{bmatrix}, \quad (118)$$

$$= \begin{bmatrix} R_3 & A_3 R_3 & & \\ \Psi^{2+} & \Psi^{2+} A_3 + R_2 & A_2 R_2 & \\ & \Psi^{1+} & \Psi^{1+} A_2 + R_1 & A_1 R_1 \\ & & \Psi^{0+} & \Psi^{0+} A_1 + R_0 \end{bmatrix}. \quad (119)$$

By comparing this system (as shown in (118)) with (119), we can determine the values of `R` and `A`. Each problem in the process has a size of `numK` × `nF`, which reduces both the time and

memory required compared to solving the full system. The code then solves (120), where \mathbf{G} denotes the drive terms, and finally obtains the moment solution by solving (121)

$$\begin{bmatrix} \mathbf{R}_3 & & & & \\ \Psi^{2+} & \mathbf{R}_2 & & & \\ & \Psi^{1+} & \mathbf{R}_1 & & \\ & & \Psi^{0+} & \mathbf{R}_0 & \\ & & & & \end{bmatrix} \begin{bmatrix} y^3 \\ y^2 \\ y^1 \\ y^0 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{G}^2 \\ \mathbf{G}^1 \\ \mathbf{G}^0 \end{bmatrix}, \quad (120)$$

$$\begin{bmatrix} 1 & \mathbf{A}_3 & & & \\ & 1 & \mathbf{A}_2 & & \\ & & 1 & \mathbf{A}_1 & \\ & & & 1 & \end{bmatrix} \begin{bmatrix} \hat{\mathbf{M}}^3 \\ \hat{\mathbf{M}}^2 \\ \hat{\mathbf{M}}^1 \\ \hat{\mathbf{M}}^0 \end{bmatrix} = \begin{bmatrix} y^3 \\ y^2 \\ y^1 \\ y^0 \end{bmatrix}. \quad (121)$$

By using the relation between closure quantities and parallel moments in Table (1), we can obtain a closure matrix. The structure of the closure matrix for parallel heat flux is illustrated in (122):

$$\left[\begin{array}{c|c|c|c} h^{p_0} & h^{T_0} & K^{hh}D & K^{h\pi}D_W \end{array} \right]. \quad (122)$$

The number of rows in this matrix is \mathbf{fmax} and the number of column is $2\mathbf{fmax} + 2$. The number of column can be extended when considering ion-electron collisions. The first two columns represents the parallel heat flux driven by $dp_0/d\psi$ and $dT_0/d\psi$ drives, respectively. Note that one column is sufficient to describe the heat flux closure since these drives are assumed to be constant on a flux surface.

The next part of the matrix is $K^{hh}D$, which means the heat flux driven by $\partial_\theta T$ drive. The matrix already incorporates the effect of the parallel gradient operator D . Therefore,

multiplying the Fourier coefficient of the perturbed temperature by the matrix is sufficient to obtain the closure, as shown in (123):

$$\begin{bmatrix} h_{00}^h & \cdots & h_{0,\text{fmax}}^h \\ \vdots & \ddots & \\ h_{\text{fmax},0}^h & & h_{\text{fmax},\text{fmax}}^h \end{bmatrix} \begin{bmatrix} \hat{T}_0 \\ \hat{T}_1 \\ \vdots \\ \hat{T}_{\text{fmax}} \end{bmatrix}. \quad (123)$$

The last part of the matrix, $K^{h\pi} D_W$, indicates the heat flux driven by the W_θ drive. Like the previous part, this matrix already includes the effect of parallel gradient operator D_W (see Equation (117)). Thus, multiplying the Fourier coefficient of the perturbed parallel velocity by the matrix is sufficient to obtain the closure. The closure matrix for parallel viscosity has a similar structure.

Representing closures with finite element basis

The closure obtained through the process described in the above section is expressed in Fourier coefficients. To apply this to NIMROD fluid equations, it needs to be expressed in the coefficients of FEM basis functions. That is, the j th column of the closure matrix must be transformed as follows:

$$\begin{bmatrix} h_{0j} \\ \vdots \\ h_{\text{fmax},j} \end{bmatrix} \Rightarrow \begin{bmatrix} H_{0j} \\ \vdots \\ H_{\text{mpy},j} \end{bmatrix}, \quad (124)$$

where the `mpy` denotes the total nodes in the poloidal angle direction.

The transformation matrix Γ from vector h_j to H_j is shown in 125:

$$\begin{bmatrix} \Gamma_0 & \Gamma_1 & \cdots & \Gamma_{\text{fmax}} \end{bmatrix} \begin{bmatrix} h_{0j} \\ h_{1j} \\ \vdots \\ h_{\text{fmax},j} \end{bmatrix} = \begin{bmatrix} H_{0j} \\ H_{1j} \\ \vdots \\ H_{\text{mpy},j} \end{bmatrix}. \quad (125)$$

Each column in matrix Γ_j represents the finite element representation of j th Fourier mode. For example, Γ_0 represents a constant function over the poloidal angle, and Γ_1 represents $\sin \theta$ over the poloidal angle.

When Γ_1 represents $\sin \theta_{\text{iy}}$ for every iy node, we need to determine θ_{iy} . This can be achieved by calculating

$$d\theta = \frac{B^\theta}{B} d\ell. \quad (126)$$

In cases where B^θ/B is uniform over the flux surface, it simplifies to

$$\theta = 2\pi \frac{\ell_{\text{iy}}}{L}, \quad (127)$$

where ℓ_{iy} is the length of the magnetic field to a specific iy node, with $\ell_0 = 0$ and $\ell_{\text{mpy}} = L$. The detailed algorithm for calculating ℓ_{iy} is provided in Appendix A.

After converting the closure matrix to the finite element space, the Fourier coefficients of drives are multiplied by the matrix as shown in (128):

$$\begin{bmatrix} H_{00}^h & \cdots & H_{0,\text{fmax}}^h \\ \vdots & \ddots & \\ H_{\text{mpy},0}^h & & H_{\text{mpy},\text{fmax}}^h \end{bmatrix} \begin{bmatrix} \hat{T}_0 \\ \hat{T}_1 \\ \vdots \\ \hat{T}_{\text{fmax}} \end{bmatrix} = \hat{T}_0 \begin{bmatrix} H_{00}^h \\ \vdots \\ H_{\text{mpy},0}^h \end{bmatrix} + \cdots + \hat{T}_{\text{fmax}} \begin{bmatrix} H_{0,\text{fmax}}^h \\ \vdots \\ H_{\text{mpy},\text{fmax}}^h \end{bmatrix}. \quad (128)$$

The closures driven by $dp_0/d\psi$, $dT_0/d\psi$, $\partial_\theta T$, and W_θ are summed together to construct the final form of the closure. After obtaining the closure values at every node from 0 to `mpy` on every flux surfaces from 0 to `mpx`, the data are saved in the `lagr_quad` type arrays for representing the data as coefficients of basis functions in the fluid simulation.

Fig. 15 illustrates the ion parallel closures due to $dT_0/d\psi$ as the number of Fourier modes in equation (115) is increased. For each `nfo`, $n_F = 1 + 2 \times nfo$, which means the system of equations includes the constant mode and `nfo` modes of sine and cosine functions. To represent the Fourier coefficients of the closures using the finite element method, quadratic basis functions are used over `mpx` = 128, `mpy` = 64 nodes in the poloidal plane. Hereafter, all results are represented on the same finite element grid. The sufficient number of Fourier modes in a problem is typically determined by the behavior of $\partial_\theta \ln B$. Fig. 15 shows that including constant, $\sin \theta$ and $\cos \theta$ modes seems sufficient to solve the given system. Figs. 16 and 17 illustrates the ion closures due to $T_1 \sin \theta$ and $T_1 \cos \theta$ drives while increasing the number of Fourier modes. The more significant the effects of higher modes are, the more Fourier modes should be included in the system of equations. As the number of moments in the system increases, the closure approaches the convergent solution as shown in Fig. 18.

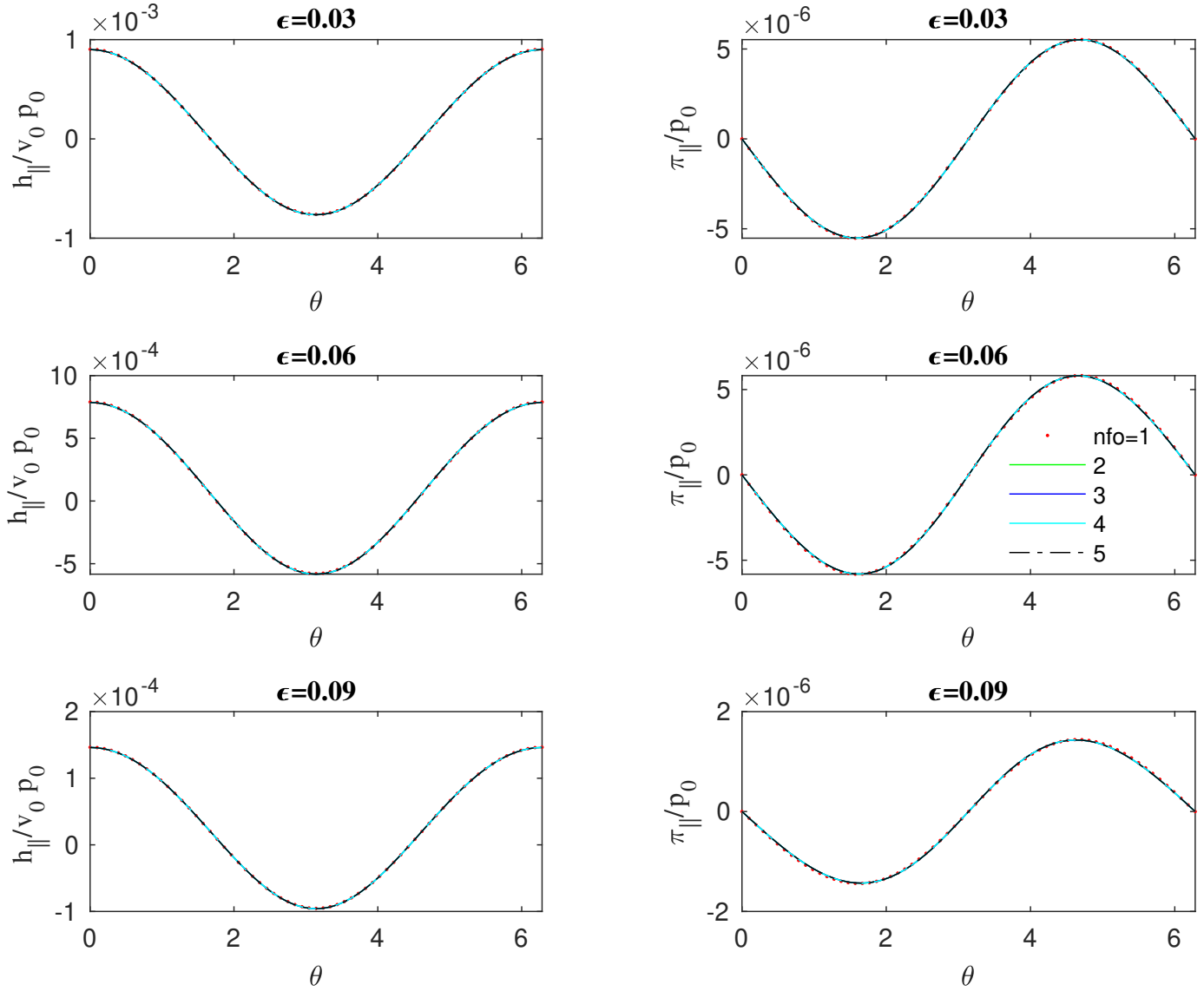


Figure 15: Ion parallel closures due to $dT_0/d\psi$ for $(\text{numL}, \text{numK}) = (4, 8)$, and $\text{nfo} = 1$ (red,dotted), 2 (green, solid), 3 (blue, solid), 4 (cyan, solid), 5 (black, dash-dotted).

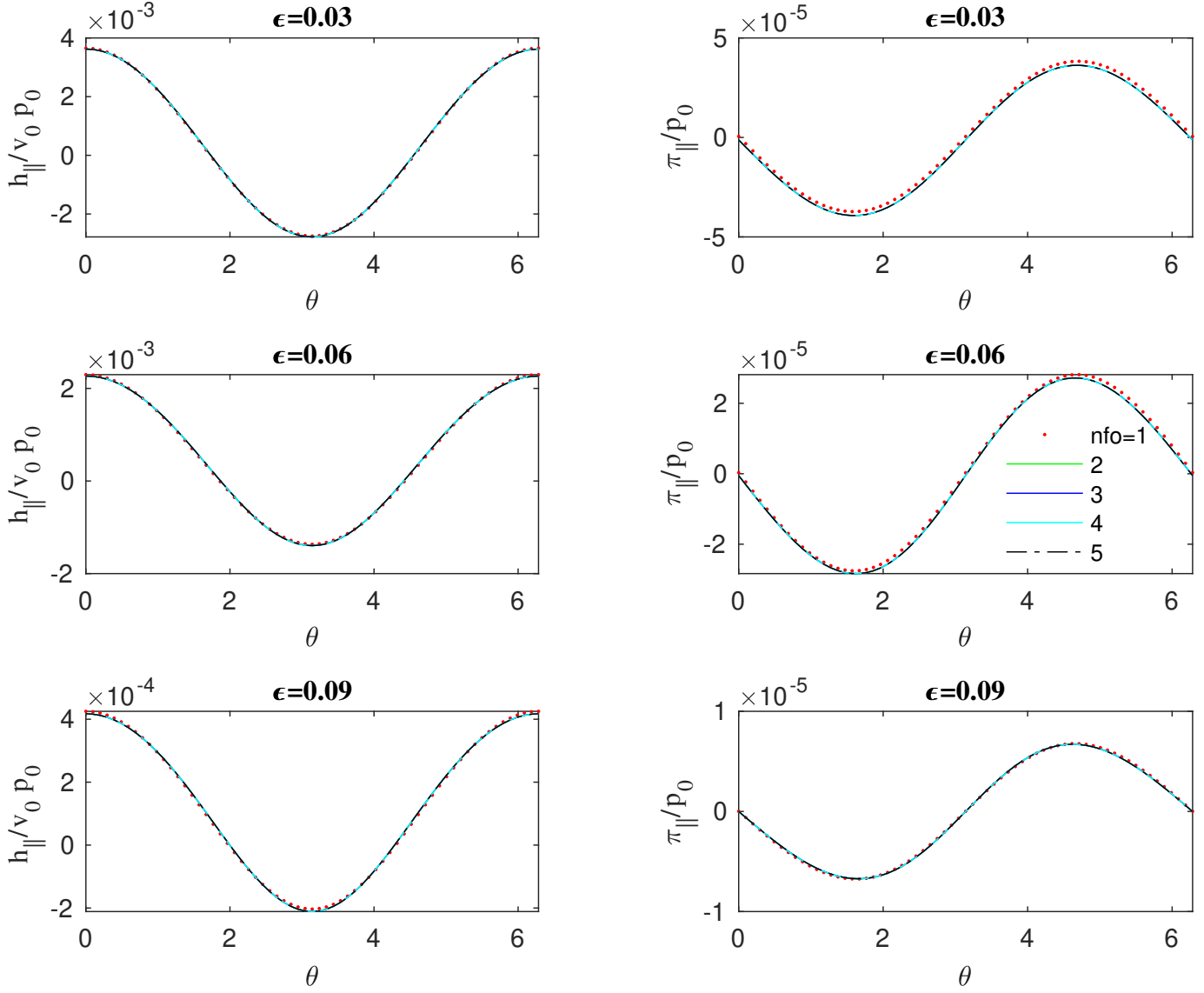


Figure 16: Ion parallel closures due to $T_1 = \sin \theta$ for $(\text{numL}, \text{numK}) = (4, 8)$, and $\text{nfo} = 1$ (red, dotted), 2 (green, solid), 3 (blue, solid), 4 (cyan, solid), 5 (black, dash-dotted).

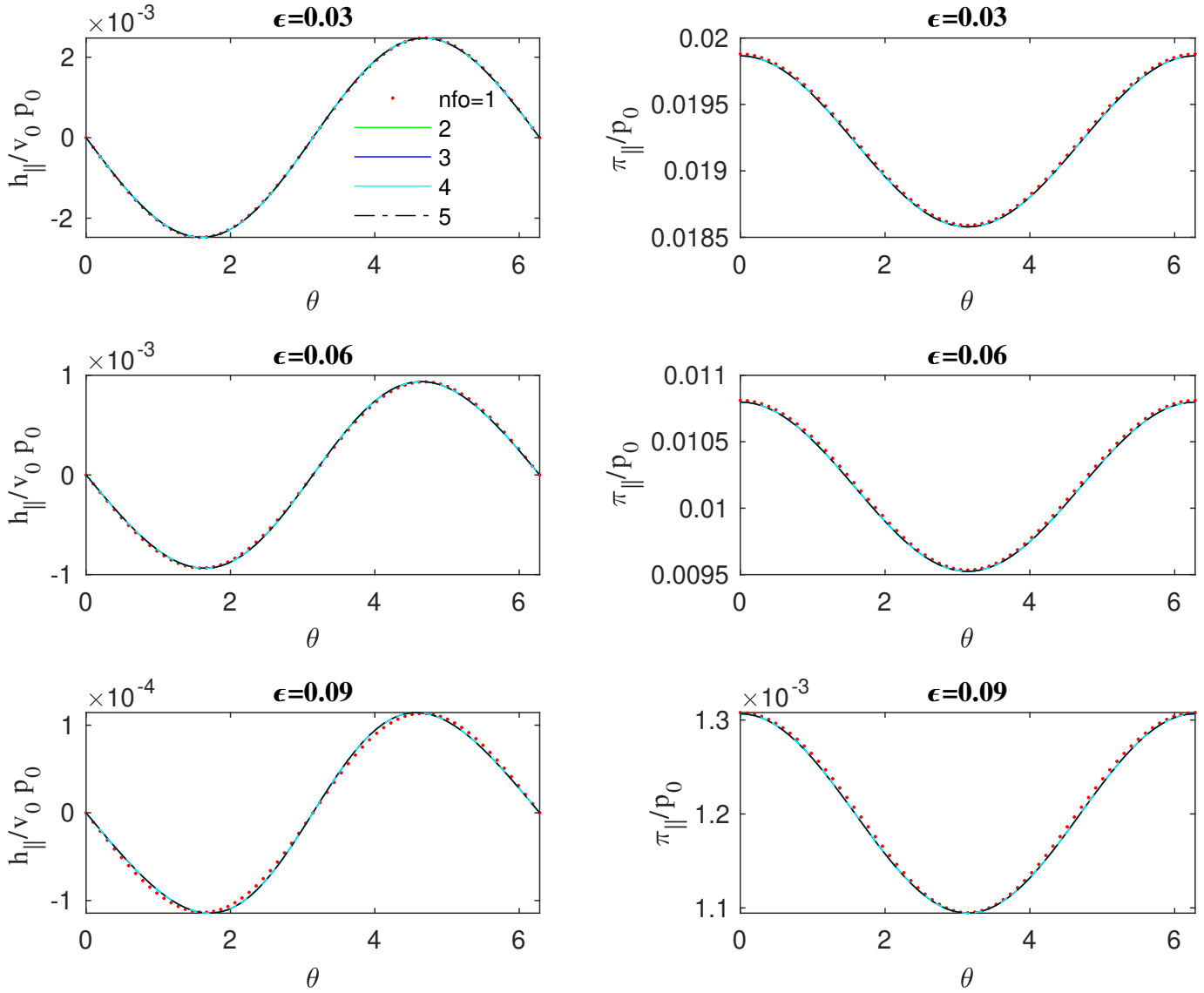


Figure 17: Ion parallel closures due to $T_1 = \cos \theta$ for $(\text{numL}, \text{numK}) = (4, 8)$, and $\text{nfo} = 1$ (red, dotted), 2 (green, solid), 3 (blue, solid), 4 (cyan, solid), 5 (black, dash-dotted).

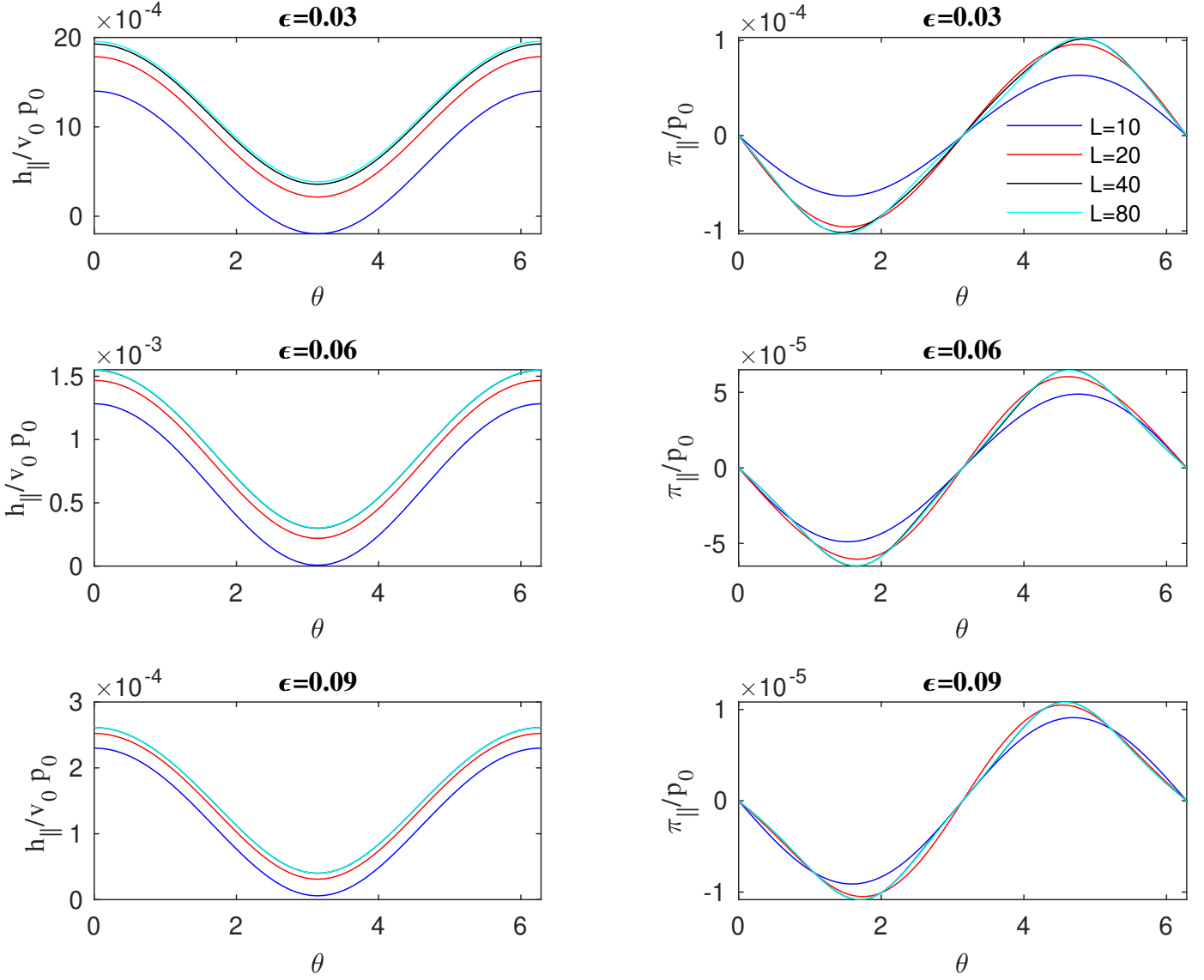


Figure 18: Ion closures including all drives in (115) using $\text{nfo} = 4$, calculated by solving $(\text{numL}, \text{numK}) = (10, 20), (20, 40), (40, 80), (80, 160)$ system at $\epsilon = 0.03, 0.06, 0.09$.

FLUID EQUATIONS AND TIME ADVANCE IN NIMROD

Linear fluid equations and closures

By solving (111), we can obtain the Fourier coefficients of perturbed number density $(\hat{n})_{\mathbf{F}}$, temperature $(\hat{T})_{\mathbf{F}}$, and parallel velocity $(\hat{u})_{\mathbf{F}}$. These solutions satisfy the fluid equations below:

$$\partial_{\theta}^{0+} \hat{u} = 2\hat{p}_{0,\psi} \frac{\partial_{\theta} \ln B}{B/B_0}, \quad (129)$$

$$\partial_{\theta}^{0+} \hat{u} + \partial_{\theta}^{0+} \hat{h} = \left(2\hat{p}_{0,\psi} + 5\hat{T}_{0,\psi} \right) \frac{\partial_{\theta} \ln B}{B/B_0}, \quad (130)$$

$$\partial_{\theta}^{1-} \hat{n} + \partial_{\theta}^{1-} \hat{T} + \partial_{\theta}^{1+} \hat{\pi} = 0, \quad (131)$$

where $\hat{h} = h_{\parallel}/v_0 p_0$ and $\hat{\pi} = \pi_{\parallel}/p_0$.

Using (40) and (50), equations (129-131) are equivalent to the steady-state, linear fluid equations:

$$\nabla \cdot (n_0 \mathbf{V}_1) = 0, \quad (132)$$

$$p_0 \nabla \cdot \mathbf{V}_1 + \nabla \cdot \mathbf{h}_1 = 0, \quad (133)$$

$$n_0 \partial_{\parallel} T_1 + T_0 \partial_{\parallel} n_1 + \mathbf{b} \cdot \nabla \cdot \pi_1 = 0, \quad (134)$$

where the subscript 0 stands for the equilibrium field and 1 stands for the perturbed field. The parallel component of the divergence of ion viscosity can be calculated as follows (Ji & Held, 2014):

$$\mathbf{b} \cdot \nabla \cdot \pi_1 = \left(\partial_{\parallel} - \frac{3}{2} \partial_{\parallel} \ln B \right) \pi_{1\parallel}. \quad (135)$$

Figs. 19-23 demonstrate that the implemented fluid quantities satisfy (132-134). In this implementation, quadratic basis functions are used over $\text{mpx} = 128$, $\text{mpy} = 64$ nodes in the poloidal plane. To verify the numerical convergence of the error in future studies, it will be

necessary to either increase the polynomial degree of the basis functions or the number of nodes in the poloidal plane.

Time advance of fluid equations

To test the time advance of the closure implementation so far, we use these Fourier coefficients $(\hat{n}, \hat{T}, \hat{u})_{\mathbf{F}}$ which satisfy (129-131). These coefficients are then converted into finite element space to set them as the initial condition. Subsequently, NIMROD calculates $\mathbf{V}_{1\perp}$ and $\mathbf{h}_{1\perp}$ using (34,36) and solves (115) to obtain ion parallel closures. Next, NIMROD advances fluid quantities starting with the parallel ion velocity, ion number density, and ion temperature as follows:

$$mn_0 \left(\frac{\Delta \mathbf{V}_{1\parallel}}{\Delta t} \right) + \partial_{\parallel} p_1 + \mathbf{b} \cdot \nabla \cdot \pi_1 = 0, \quad (136)$$

$$\left(\frac{\Delta n_1}{\Delta t} \right) + \nabla \cdot (n_0 \mathbf{V}_1) = 0, \quad (137)$$

$$\frac{3}{2} n_0 \left(\frac{\Delta T_1}{\Delta t} \right) + p_0 \nabla \cdot \mathbf{V}_1 + \nabla \cdot \mathbf{h}_1 = 0, \quad (138)$$

where $\mathbf{V}_1 = \mathbf{V}_{1\parallel} + \mathbf{V}_{1\perp}$ and $\mathbf{h}_1 = \mathbf{h}_{1\parallel} + \mathbf{h}_{1\perp}$. During the simulation, $\mathbf{V}_{1\perp}$ and $\mathbf{h}_{1\perp}$ remain unchanged since they are constructed from the equilibrium fields. The detailed flow of the entire closure module is shown in Appendix B.

To enhance time efficiency for the time loop solving (136-138), we have implemented parallelism over flux surface indices. Since solving (119) for all \mathbf{ix} indices is time-consuming, we have divided the range of \mathbf{ix} so that multiple processors can solve the divided problems in parallel. Next, during the process described in (128), we gather the divided results to enable all processors to construct the closure, allowing each processor to utilize the closure information within its assigned `rblock`. Fig. 24 illustrates the computation times for solving the systems with $(\text{numL}, \text{numK}) = (10, 20), (20, 40)$ and $(40, 80)$ using both serial and parallel algorithms. In the parallel implementation, we divide a $(\text{mpx}, \text{mpy}) = (128, 64)$ grid using quadratic basis functions into 16 sub-grids in the \mathbf{ix} direction. Utilizing the NERSC

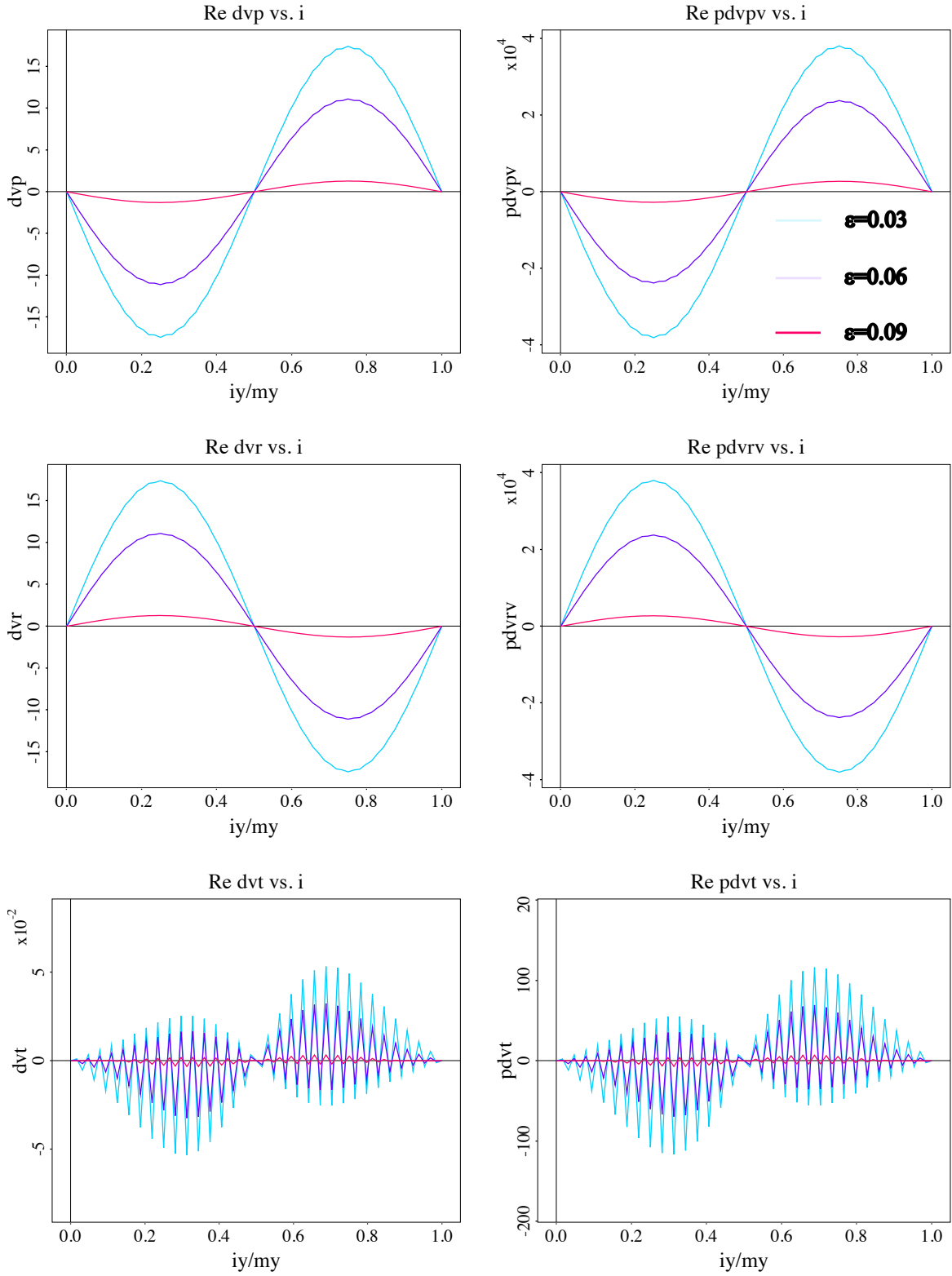


Figure 19: Fluid quantities $\nabla \cdot \mathbf{V}_{\parallel}$, $\nabla \cdot \mathbf{V}_{\perp}$, $\nabla \cdot (\mathbf{V}_{\parallel} + \mathbf{V}_{\perp})$ (left), $p_0 \nabla \cdot \mathbf{V}_{\parallel}$, $p_0 \nabla \cdot \mathbf{V}_{\perp}$, $p_0 \nabla \cdot (\mathbf{V}_{\parallel} + \mathbf{V}_{\perp})$ (right) calculated by solving $(\text{numL}, \text{numK}) = (4, 8)$ system at $\epsilon = 0.03$ (blue), 0.06 (purple), 0.09 (red).

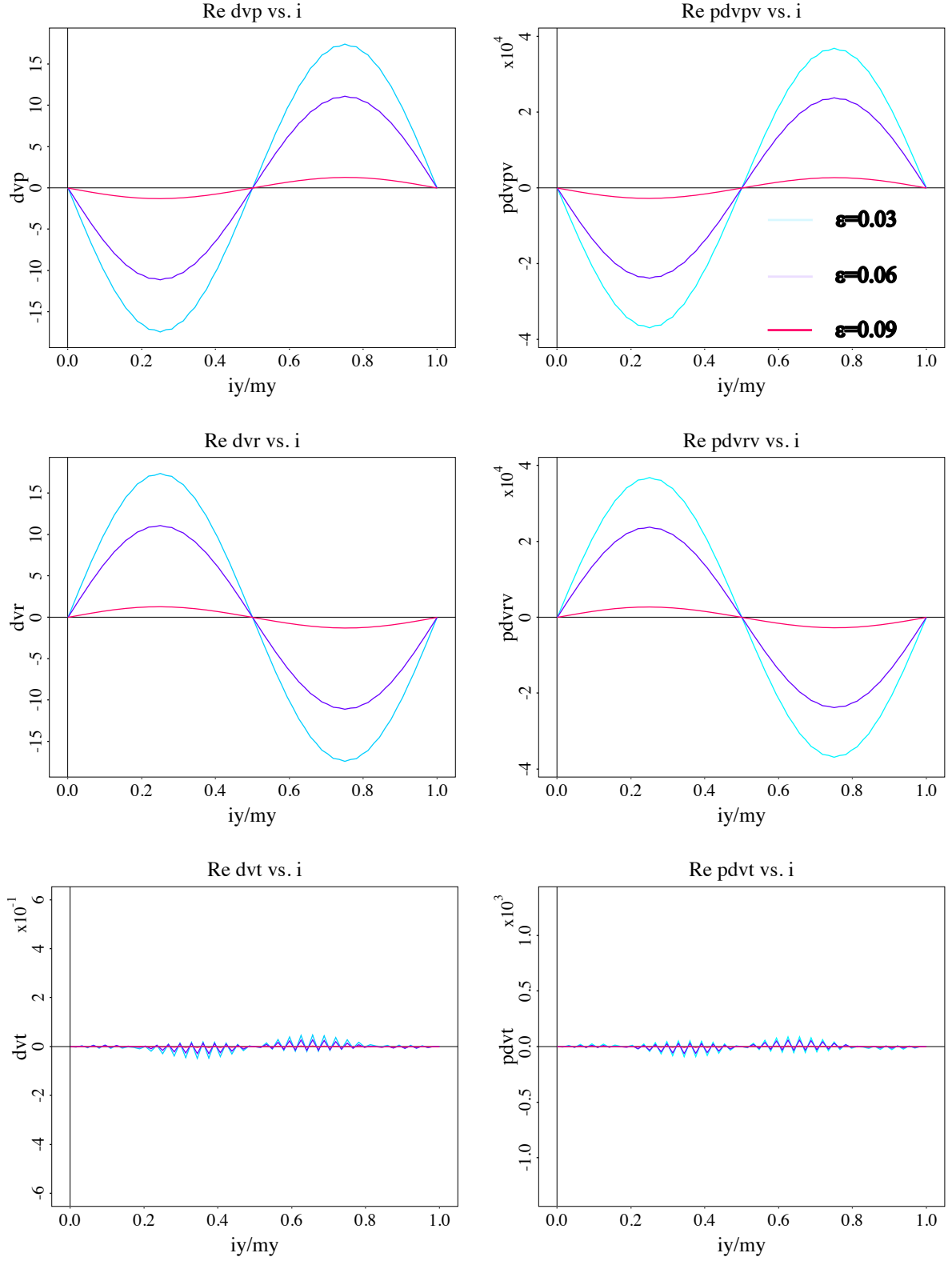


Figure 20: Fluid quantities $\nabla \cdot \mathbf{V}_{\parallel}$, $\nabla \cdot \mathbf{V}_{\perp}$, $\nabla \cdot (\mathbf{V}_{\parallel} + \mathbf{V}_{\perp})$ (left), $p_0 \nabla \cdot \mathbf{V}_{\parallel}$, $p_0 \nabla \cdot \mathbf{V}_{\perp}$, $p_0 \nabla \cdot (\mathbf{V}_{\parallel} + \mathbf{V}_{\perp})$ (right) calculated by solving $(\text{numL}, \text{numK}) = (40, 80)$ system at $\epsilon = 0.03$ (blue), 0.06 (purple), 0.09 (red).

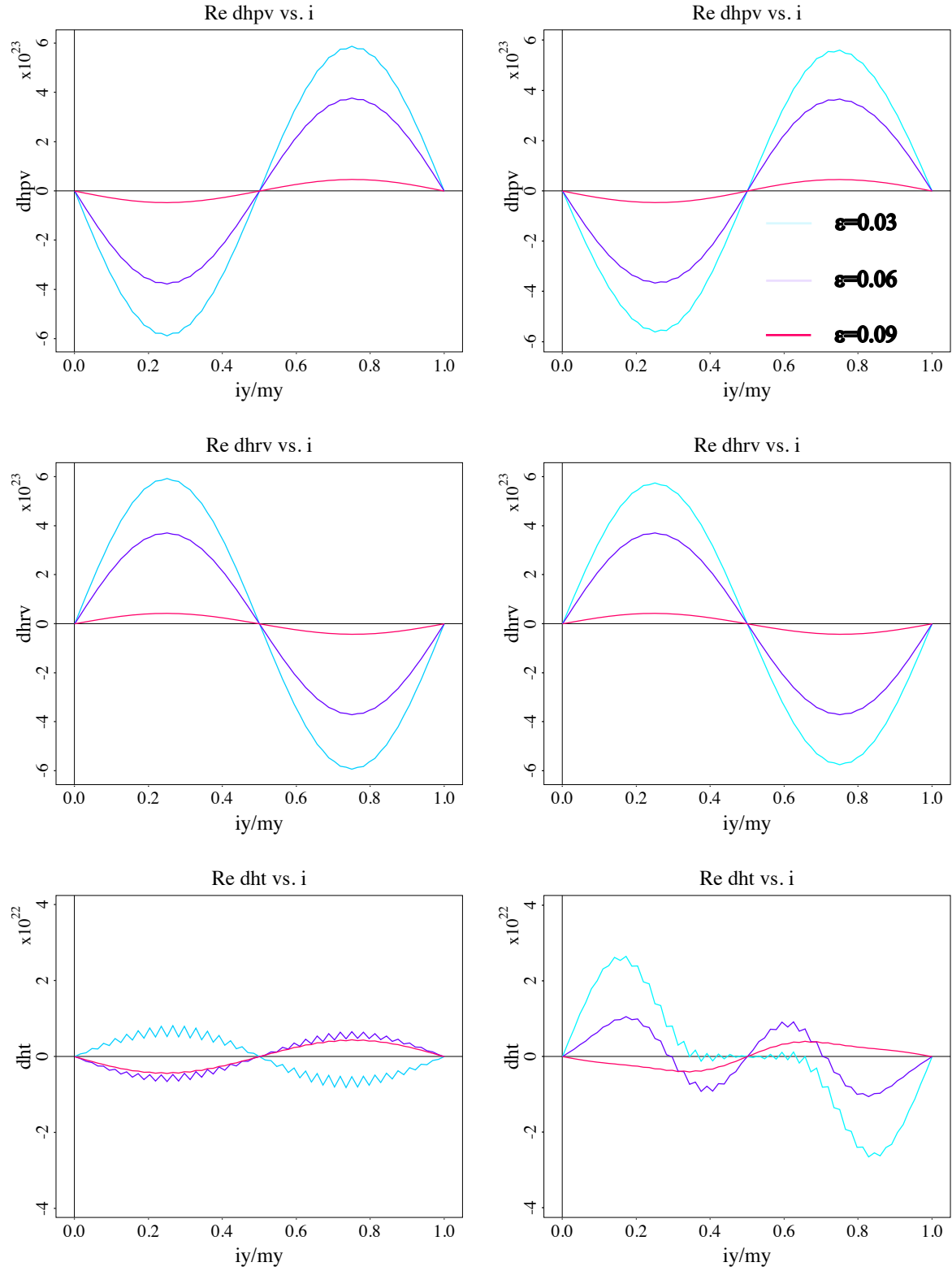


Figure 21: Fluid quantities $\nabla \cdot \mathbf{h}_{\parallel}$ (top), $\nabla \cdot \mathbf{h}_{\perp}$ (middle), and $\nabla \cdot (\mathbf{h}_{\parallel} + \mathbf{h}_{\perp})$ (bottom) calculated by solving $(\text{numL}, \text{numK}) = (4, 8)$ (left) and $(\text{numL}, \text{numK}) = (40, 80)$ (right) systems at $\epsilon = 0.03$ (blue), 0.06 (purple), 0.09 (red).

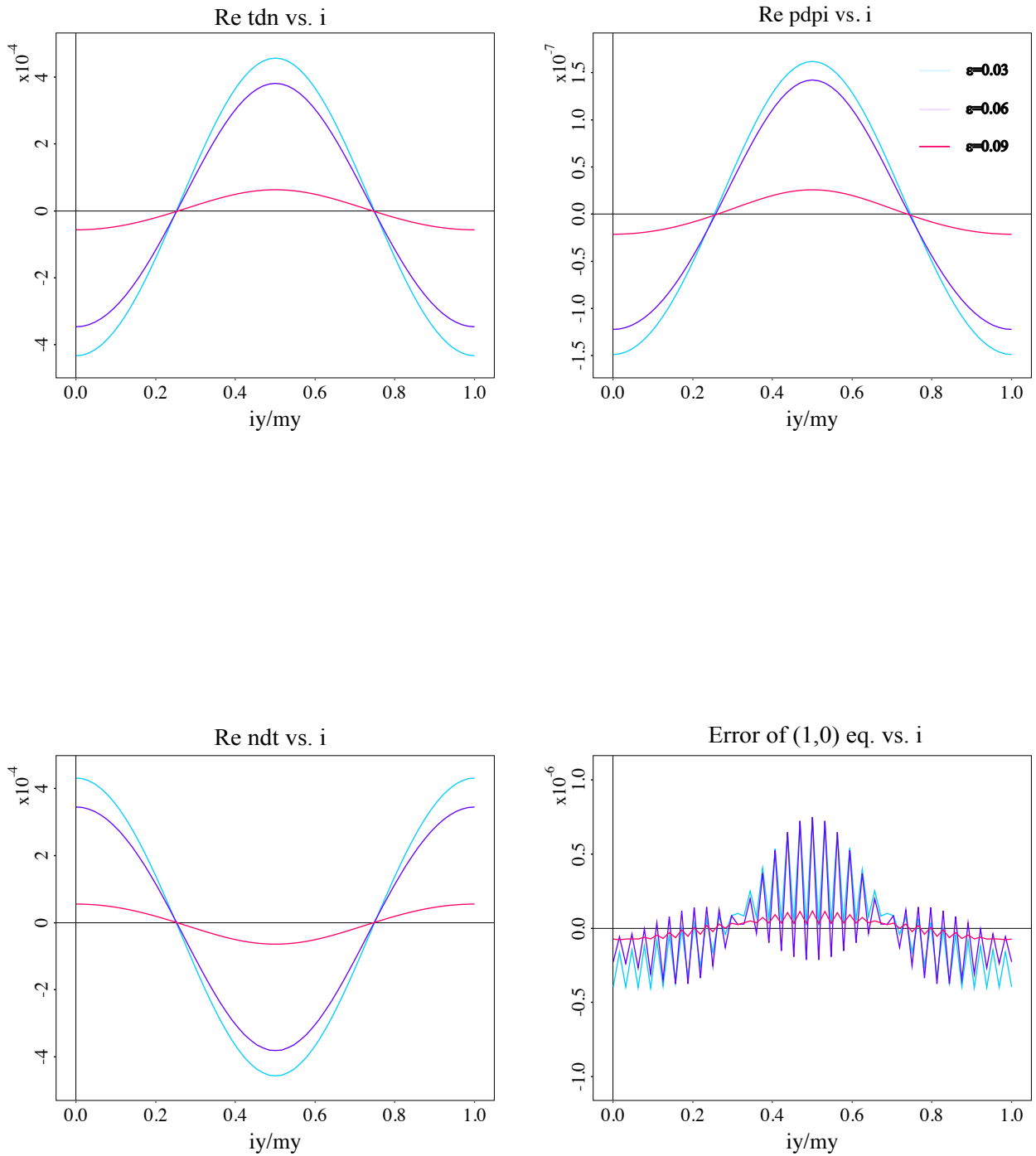


Figure 22: Fluid quantities $T_0 \partial_{\parallel} n_1$ (top left), $n_0 \partial_{\parallel} T_1$ (bottom left), $\mathbf{b} \cdot \nabla \cdot \pi_{\parallel}$ (top right), and the sum of all terms (bottom right) calculated by solving the system $(\text{numL}, \text{numK}) = (4, 8)$ at $\epsilon = 0.03$ (blue), 0.06 (purple), 0.09 (red).

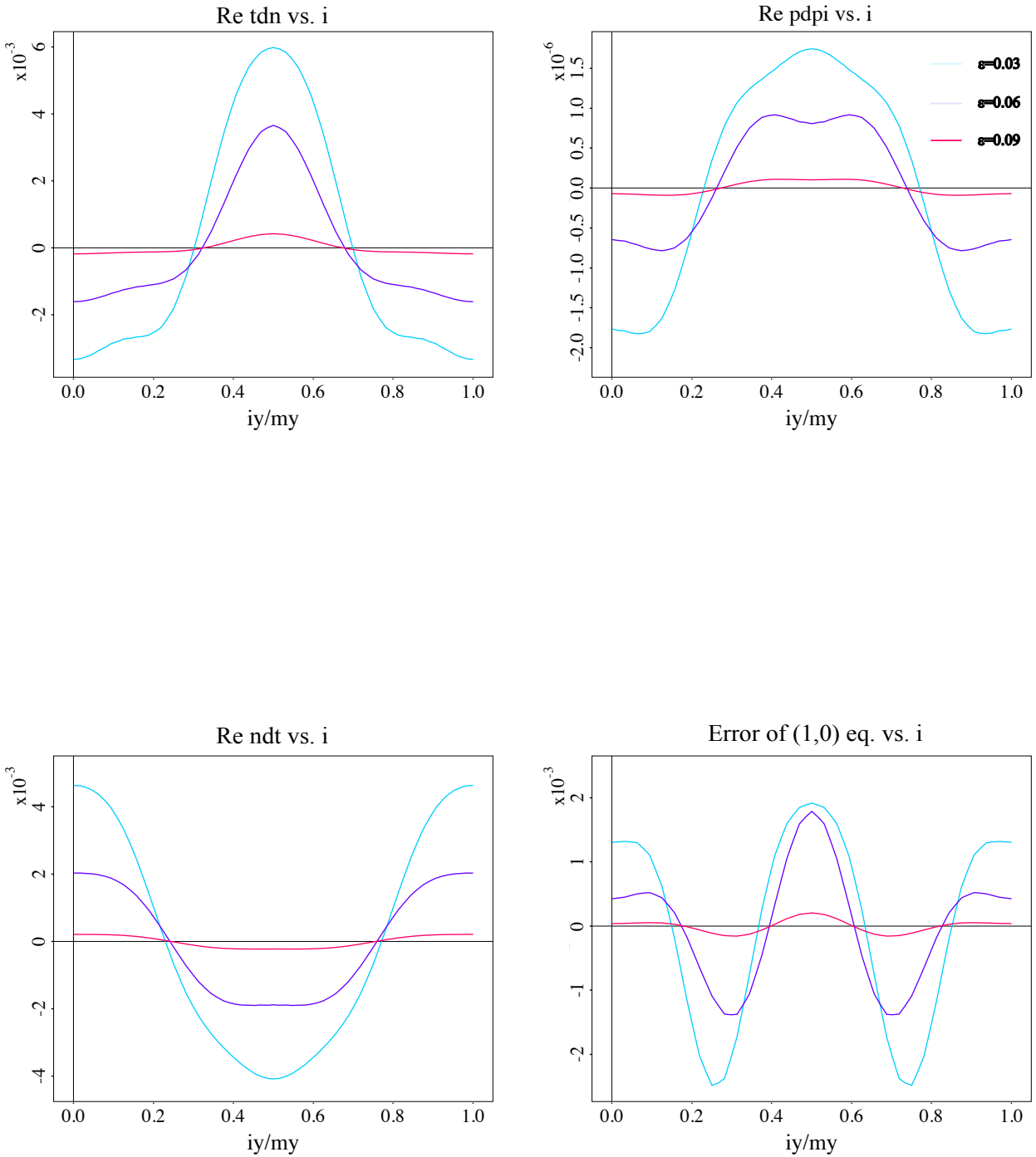


Figure 23: Fluid quantities $T_0 \partial_{\parallel} n_1$ (top left), $n_0 \partial_{\parallel} T_1$ (bottom left), $\mathbf{b} \cdot \nabla \cdot \pi_{\parallel}$ (top right), and the sum of all terms (bottom right) calculated by solving the system $(\text{numL}, \text{numK}) = (40, 80)$ at $\epsilon = 0.03$ (blue), 0.06 (purple), 0.09 (red).

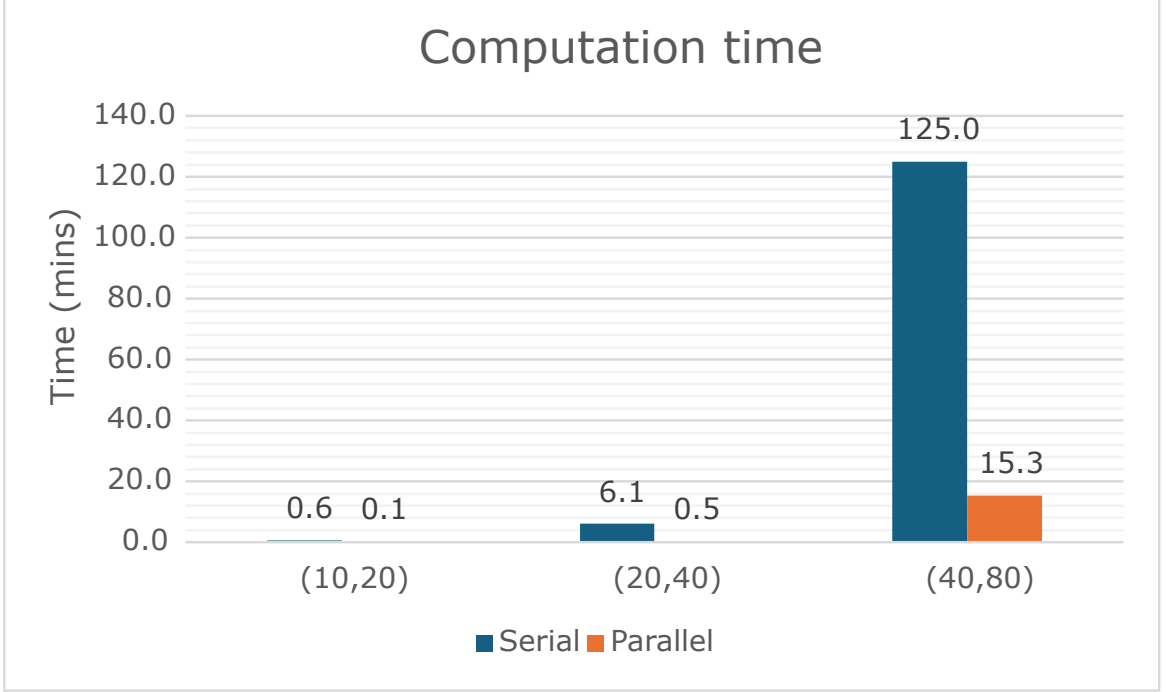


Figure 24: Computation time in minutes for solving the systems with $(\text{numL}, \text{numK}) = (10, 20), (20, 40),$ and $(40, 80)$ using serial (blue) and parallel (orange) algorithms.

Perlmutter supercomputing system with 16 processors, each processor solves a divided sub-problem.

As shown in the figures above, the initial conditions n_1, T_1, u_1 and parallel closures \mathbf{h}_1, π_1 satisfy (132-134) well. However, it is essential to address the errors that have arisen during the numerical implementation. Fig. 25 displays the time evolution of n_1 and $\nabla \cdot \mathbf{V}_1$ on a flux surface, where the numerical error in $\nabla \cdot \mathbf{V}_1$ potentially distort the n_1 profile as dictated by (137) unless mitigated in the code. To manage this, an error-diffusion approach (Sovinec, 2001) was adopted, incorporating an artificial diffusion term into the parallel velocity equation

$$mn_0 \left(\frac{\Delta \mathbf{V}_{1\parallel}}{\Delta t} \right) = -\partial_{\parallel} p_1 - \mathbf{b} \cdot \nabla \cdot \pi_1 + \kappa \nabla (\nabla \cdot \mathbf{V}_1). \quad (139)$$

This approach has demonstrated a reduction in the error of $\nabla \cdot \mathbf{V}_1$ and the stabilization of n_1 over time. Further investigation into the validity and impact of this diffusion term is

required for future studies.

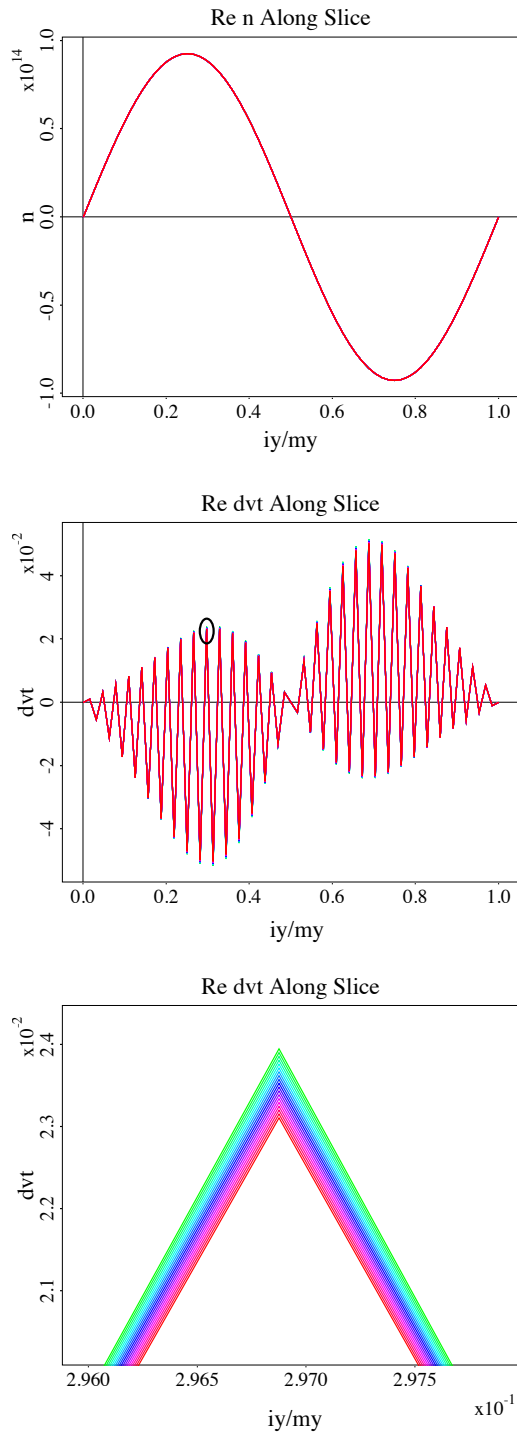


Figure 25: Time slices (starting from green to red) of perturbed number density (top), $\nabla \cdot \mathbf{V}_1$ (middle), and an enlarged image of $\nabla \cdot \mathbf{V}_1$ (bottom, inside the black circle) at $\epsilon = 0.03$. A diffusion term is added to stabilize the simulation.

CONCLUSION AND FUTURE WORK

We have implemented the non-Maxwellian moment-Fourier equations in NIMROD and obtained parallel closures that can be used with a finite element basis. The convergence of the closure solution indicates that we have successfully solved the DKE using the parallel moment method. In the near future, we expect to compare the results obtained from this code with those from other numerical DKE solvers.

Further investigation into the numerical stability of this code is required when the fluid equations are influenced by the parallel closures. Discussing the implementation of a semi-implicit operator for parallel closures within the fluid equations could be beneficial. Starting from arbitrary initial conditions, such as $n_1 = 0$, $T_1 = 0$, and $u = 0$, the simulation is expected to reach the steady state as obtained by solving (81). Incorporating the parallel closure module into the perpendicular component of fluid equations and non-linear terms such as $\mathbf{V} \cdot \nabla \mathbf{V}$ could enhance the simulation by capturing the non-linear effects that cannot be analyzed solely through the DKE.

By implementing the moment-Fourier equations for electron closures, more realistic plasma simulations can be achieved. In NIMROD's extended MHD equations, electron closures play a key role in the parallel component of Ohm's law:

$$E_{\parallel} \approx R_{e\parallel} - \frac{1}{ne} \mathbf{b} \cdot \nabla \cdot \pi_e \quad (140)$$

where the π_e is the electron viscosity tensor and \mathbf{R}_e is the electron collisional momentum exchange closure. The $R_{e\parallel}$ closure includes the effect of electron parallel moments as

$$R_{e\parallel} = \frac{m_e v_{Te}}{\tau_{ei}} \left[-n_e \frac{V_{e\parallel} - V_{i\parallel}}{v_{Te}} + \frac{1}{\sqrt{2}} \sum_{k=1} a_{ei}^{10k} N_{e\parallel}^{1k} \right] \quad (141)$$

where a_{ei}^{10k} is a collision coefficient (Ji et al., 2016). With these closures, the bootstrap current can be calculated and compared to the analytic results for large aspect ratio (Hirshman & Sigmar, 1976).

REFERENCES

- Braginskii, S. (1965). Transport processes in a plasma. *Reviews of plasma physics*, 1, 205.
- Goldstein, H., Poole, C. P., & Safko, J. L. (2017). *Classical mechanics*. Pearson.
- Gurnett, D. A., & Bhattacharjee, A. (2017). *Introduction to plasma physics: With space, laboratory and astrophysical applications*. Cambridge University Press.
- Hazeltine, R. D., & Meiss, J. D. (2003). Plasma confinement. In (chap. 4). Courier Corporation.
- Helander, P., & Sigmar, D. J. (2005). *Collisional transport in magnetized plasmas* (Vol. 4). Cambridge university press.
- Held, E., Kruger, S., Ji, J.-Y., Belli, E., & Lyons, B. (2015). Verification of continuum drift kinetic equation solvers in nimrod. *Physics of Plasmas*, 22(3), 032511.
- Hindmarsh, A. C. (1980). Lsode and lsodi, two new initial value ordinary differential equation solvers. *ACM Signum Newsletter*, 15(4), 10–11.
- Hirshman, S., & Sigmar, D. (1976). Approximate fokker–planck collision operator for transport theory applications. *The Physics of Fluids*, 19(10), 1532–1540.
- Ji, J.-Y. (2014a, May). *Complement 1 kinetic equation for plasmas*. Lecture notes.
- Ji, J.-Y. (2014b, May). *Lecture 5 derivation of the drift kinetic equation*. Lecture notes.
- Ji, J.-Y., & Held, E. D. (2006). Exact linearized coulomb collision operator in the moment expansion. *Physics of plasmas*, 13(10).
- Ji, J.-Y., & Held, E. D. (2014). A framework for moment equations for magnetized plasmas. *Physics of Plasmas*, 21(4).
- Ji, J.-Y., Held, E. D., & Jhang, H. (2013). Linearly exact parallel closures for slab geometry. *Physics of Plasmas*, 20(8).
- Ji, J.-Y., Held, E. D., Spencer, J. A., & Na, Y.-S. (2023). Moment-fourier approach to ion parallel fluid closures and transport for a toroidally confined plasma. *Plasma Physics*

and Controlled Fusion.

- Ji, J.-Y., Kim, S.-K., Held, E. D., & Na, Y.-S. (2016). Electron parallel closures for various ion charge numbers. *Physics of Plasmas*, *23*(3).
- Ji, J.-Y., Lee, H. Q., & Held, E. D. (2017). Ion parallel closures. *Physics of Plasmas*, *24*(2).
- Karniadakis, G., & Sherwin, S. J. (2005). *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, USA.
- Landau, L. (1936). Kinetic equation for the case of coulomb interaction. *Phys. Z. Sowjetunion*, *10*, 154-164.
- Press, W. H. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press.
- Rosenbluth, M. N., MacDonald, W. M., & Judd, D. L. (1957). Fokker-planck equation for an inverse-square force. *Physical Review*, *107*(1), 1.
- Sovinec, C. R. (2001, August). A tutorial on nimrod physics kernel code development [Computer software manual].
- Sovinec, C. R., Gianakon, T. A., Held, E. D., Kruger, S. E., Schnack, D. D., & Team, N. (2003). Nimrod: A computational laboratory for studying nonlinear fusion magneto-hydrodynamics. *Physics of Plasmas*, *10*(5), 1727–1732.

APPENDICES

APPENDIX A

Algorithm for measuring magnetic field length between two nodes on a flux surface

This algorithm is designed for calculating the magnetic field length $\ell_f = \ell_i + \Delta\ell$ at the final point (R_f, Z_f, φ_f) measured from an initial point on a flux surface (R_i, Z_i, φ_i) . Using the LSODE library ([Hindmarsh, 1980](#)), we can calculate the change in position along a certain magnetic field length:

$$\int_{R_i}^{R_f} dR = \int_{\ell_i}^{\ell_f} \frac{B_R}{B} d\ell, \quad (\text{A.1})$$

$$\int_{Z_i}^{Z_f} dZ = \int_{\ell_i}^{\ell_f} \frac{B_Z}{B} d\ell, \quad (\text{A.2})$$

This algorithm makes an initial guess for ℓ_f and employs a trial and error method to find the best ℓ_f within the range of tolerance.

Categorizing initial points is useful when assessing the status of a trial point, as illustrated in Fig. [A.1](#). The initial point (marked by a black square) is located in the first zone. In this zone, the code compares the Z -coordinate of the trial point (blue square) Z_* with the final point (red circle) Z_f . If $Z_* > Z_f$, the code determines that the trial point has surpassed the final point, leading to halving the guess for ℓ_f . Conversely, if $Z_* < Z_f$, it sets the trial point as the new initial point and halves the guess again.

In the second zone, the code assesses whether the trial point has passed the final point, based on its R -coordinate. A result of $R_* < R_f$ implies it has passed, while $R_* > R_f$ suggests it has not. The third and fourth zones employ similar methodologies to evaluate the position of the trial point relative to the final one.

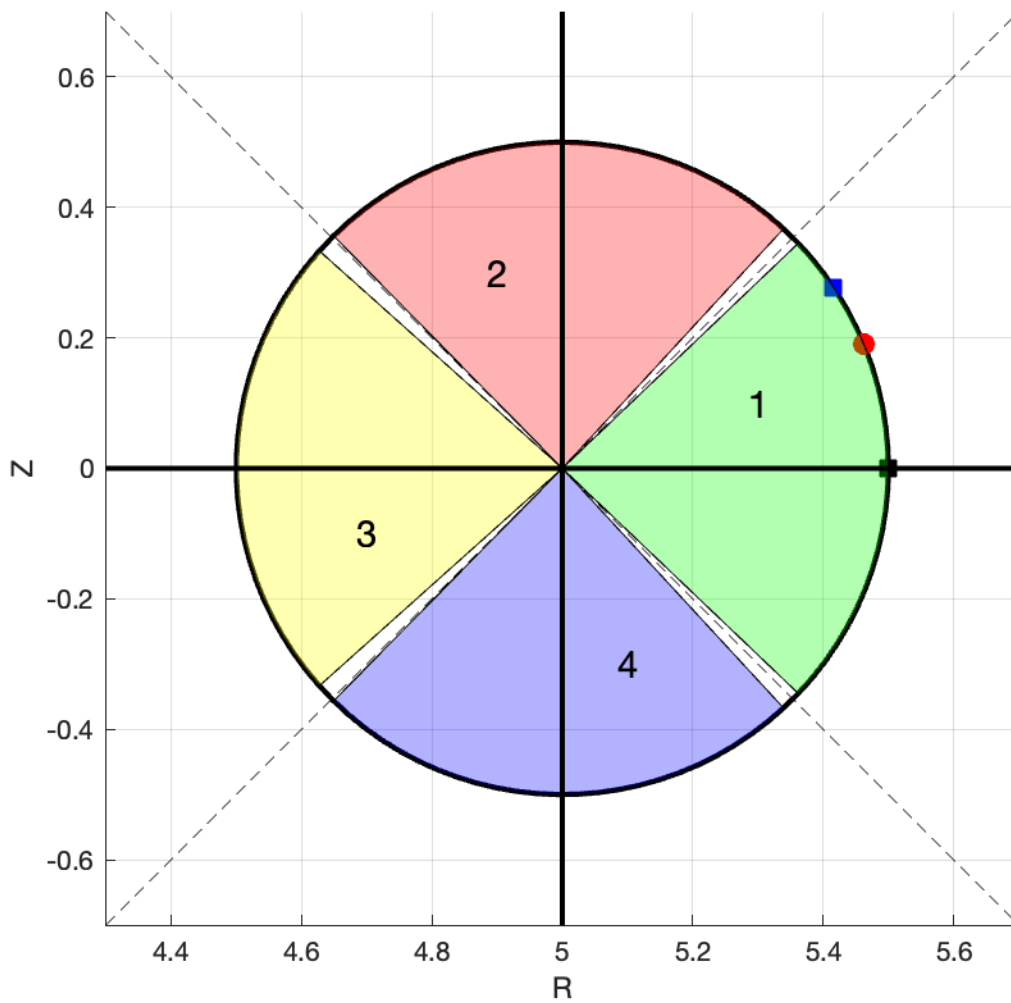


Figure A.1: Zones for determining the status of a trial point.

APPENDIX B

Data flow for the closure module in NIMROD

FGNIMEQ

1. Retrieve $\psi, \mathbf{B}_0, T_0, n_0$ from an input file
2. Identify the perturbed fields $n_1, T_1, \mathbf{V}_{\parallel}$
3. Generate an initial dump file

NIMROD

Closure Module

1. Compute fields for parallel moment equations from the initial dump file
2. Sample field values at the same \mathbf{B} interval
3. Obtain Fourier coefficients of the fields
4. Solve the system of parallel moment equations and generate the closure matrix (122)
5. Transform the matrix from Fourier space to real space
6. Obtain closures h_{\parallel} and π_{\parallel} by applying the Fourier coefficients of drives
7. Store the closure data in the `lagr_quad` type arrays
8. Compute \mathbf{V}_{\perp} and \mathbf{h}_{\perp}

Fluid Advance Module

1. Advance ion parallel velocity \mathbf{V}_{\parallel} with the π_{\parallel} closure
2. Advance ion number density n_1
3. Advance ion temperature T_1 with the \mathbf{h} closure. Return to step 1 of the Closure Module for the next loop.

CURRICULUM VITAE

HANKYU LEE

Logan, UT | hankyu.lee@usu.edu | (435) 265-2841

EDUCATION**Doctor of Philosophy**

Year of completion: 2024

Utah State University

Physics

Bachelor of Science

Year of completion: 2016

Pohang University of Science and Technology

Physics

RESEARCH EXPERIENCE

2016 - 2024

Utah State University

- Implementing general moment equations for parallel closures in NIMROD code
- Developing ion parallel closures for arbitrary atomic weights and charge numbers
- Studying the reduced drift kinetic equation by using finite difference method to obtain parallel closures in inhomogeneous magnetic field.

- Studying Coulomb-Milne problem by using general moment approach.

2015

Pohang University of Science and Technology

- MHD simulation for studying coronal heating mechanism.
- Studying upwind algorithms for discontinuous hyperbolic PDE problems (Outstanding Undergraduate Thesis Award).

2014

Australia's Nuclear Science and Technology Organization

- Developing embedded atom model for a high entropy refractory alloy (Mo-Nb-Ta-W)
- Collision cascade simulation by neutron radiation on the high entropy alloy

PUBLICATIONS

J.-Y. Ji, H. Q. Lee, and E. D. Held, Ion parallel closures, *Phys. Plasmas* 24, 022127 (2017)