



HAL
open science

Computing the double-gyroaverage term incorporating short-scale perturbation and steep equilibrium profile by the interpolation algorithm

Shuangxi Zhang, Michel Mehrenberger, Christophe Steiner

► **To cite this version:**

Shuangxi Zhang, Michel Mehrenberger, Christophe Steiner. Computing the double-gyroaverage term incorporating short-scale perturbation and steep equilibrium profile by the interpolation algorithm. 2019. hal-02009446v1

HAL Id: hal-02009446

<https://hal.science/hal-02009446v1>

Preprint submitted on 6 Feb 2019 (v1), last revised 1 Oct 2019 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Computing the double-gyroaverage term incorporating short-scale perturbation and steep equilibrium profile by the interpolation algorithm

Shuangxi Zhang

IRMA, Université de Strasbourg & Inria TONUS Team

Michel Mehrenberger

Aix-Marseille Université

Christophe Steiner

christophe.steiner1@gmail.com

February 6, 2019

Abstract

In gyrokinetic model and simulations, when the double-gyroaverage term incorporates the combining effect contributed by the finite Larmor radius, short-scales of the perturbation and steep gradient of the equilibrium profile, the low-order approximation of this term could generate unignorable error. This paper implements an interpolation algorithm to compute the double-gyroaverage term without low-order approximation to avoid this error. For a steep equilibrium density, the obvious difference between the density on the gyrocenter coordinate frame and the one on the particle coordinate frame should be accounted for in the quasi-neutrality equation. An Euler-Maclaurin-based quadrature integrating algorithm is developed to compute the quadrature integral for the distribution of the magnetic moment. The application of the interpolation algorithm to computing the double-gyroaverage term and to solving the quasi-neutrality equation is benchmarked by comparing the numerical results with the known analytical solutions. At last, to make the advantage of the interpolation solver clearer, the numerical comparison between the interpolation solver and a classical second order solver is carried out in a constant theta-pinch magnetic field configuration using SELALIB code. When the equilibrium profile is not steep and the perturbation only has the nonzero mode number along the parallel spatial dimension, the results computed by the two solvers match each other well. When the gradient of the equilibrium profile is steep, the interpolation solver provides a bigger driving effect for the ion-temperature-gradient modes which possess large polar mode numbers.

1 Introduction

The micro-scale turbulence plays a significant role on the confinement capability of the magnetized fusion plasma through its interaction with the low-frequency zonal flow [9, 26, 1, 26, 37, 12, 19, 38], the equilibrium profile at the pedestal region and the edge localized modes [39, 18, 3, 20, 28], etc. The importance of the pedestal region is reflected by the fact that the pressure of the plasma core is proportional to the pressure at the pedestal top [30, 35, 10]. The gradient of the equilibrium temperature and density at the pedestal region could be very sharp, alternatively, the truncation of $\exp(\boldsymbol{\rho} \cdot \nabla)$ acting over n_0 or T_0 at the first order, which is utilized by the standard gyrokinetic model [13, 25, 24], is not a good approximation, where $\boldsymbol{\rho}$ is the Larmor-radius vector defined as

$$\boldsymbol{\rho}(\mathbf{x}, \mu, \theta) = \frac{1}{q_s} \sqrt{\frac{2\mu m_s}{B}} (\mathbf{e}_1 \cos \theta + \mathbf{e}_2 \sin \theta).$$

Here, q_s, m_s, μ are the particles' charge, mass and the magnetic moment on the guiding-center coordinate frame; $\mathbf{e}_1, \mathbf{e}_2$ are the unit vectors perpendicular to unit vector \mathbf{b} of the equilibrium magnetic field, and $\mathbf{e}_1, \mathbf{e}_2, \mathbf{b}$ obey the right hand rule; B is the magnitude of the magnetic field; θ is the gyrophase angle; n_0, T_0 are the equilibrium density and temperature profile. In the gyrokinetic theory, the first gyroaverage term could incorporate the effect contributed by the finite Larmor radius and short scales of the perturbation at the core and edge tokamak plasma, while the double-gyroaverage term (DGT) incorporates the combining effect contributed by the finite Larmor radius, the short-scale perturbation and the steep-gradient equilibrium profile within the edge transport barrier of tokamak plasma [11, 24, 27, 36]. However, in the standard electrostatic gyrokinetic model and simulations, the approximation of DGT consists of up to the first order truncation of the Taylor expansion of the equilibrium profile and the second order truncation of the expansion of the perturbative potential [25, 13, 22, 29, 4, 5, 21, 6, 14]. The details of this method can be found in section (4). These kinds of low order approximation are not enough concerning the short-scale perturbations and the equilibrium profile possessing the steep gradient.

To overcome the mentioned drawbacks of the low-order approximation of DGT, this paper develops an interpolation algorithm to compute DGT for the purpose of the resolution of the short-scale perturbation and the steep equilibrium profile together. The interpolation algorithm can approach DGT with any accuracy by choosing enough interpolation points on the Larmor circle, with only the constraint from the length scale of the mesh. Meanwhile, contrary to the traditional way, the steep equilibrium density shouldn't keep the same before and after the gyroaverage operation, which is denoted by the symbol \mathcal{J} in this paper. And the obvious difference between the density on gyrocenter coordinate and the one on the particle coordinates should be accounted for in the quasi-neutrality equation (QNE). The obvious difference is revealed by Fig.(1), for which the normalisation scheme can be found in subsection (3.3) and the parameters to obtain this figure are provided in subsection (8.3.3). In Fig.(1), the purple line denotes the equilibrium density profile on the gyrocenter coordinates, which is given as the initial condition, while the density profiles on the particle coordinate derived by the gyroaverage operation computed with different number of μ are shown by other curves. The numerical integration of μ is done by the Euler-Maclaurin-based quadrature integration algorithm.

As a comparison to the interpolation solver, a classical 2nd order truncation of DGT is carried out in this paper to obtain QNE with the 2nd-order accuracy. It consists

of the truncation of the exponential operator $\exp(\boldsymbol{\rho} \cdot \nabla)$ over the potential up to the second order and the truncation of the operator $\exp(\boldsymbol{\rho} \cdot \nabla)$ acting on the density up to the first order. The numerical simulations are carried out to compare the two solvers. Since this paper is to compare the two solvers in general purpose, the deduction of flux surface average of the potential from the total potential is not carried out to obtain the electrons' adiabatic distribution. For the equilibrium profile without steep gradient, both algorithms have almost the same performance in terms of the perturbation only possessing the mode number along the parallel spatial dimension. For the steep equilibrium profile, the interpolation algorithm could provide a stronger driving effect for the perturbation of high polar mode numbers, and the saturation time of these modes computed by the interpolation algorithm is earlier.

The remaining parts of this paper are arranged as follows. Section (2) introduces the orders used in deriving the gyrokinetic model. The DGT and QNE incorporating the short-scale perturbation and steep equilibrium profile on the gyrocenter coordinate frame are derived in Section (3). Section (4) derives the QNE with the 2nd-order truncation. In Section (5), the interpolation solver is introduced. The benchmark of the interpolation algorithm is done in Sections (6) and (7) for the case of the single μ and μ obeying the distribution, respectively. The Euler-Maclaurin-based quadrature integrating algorithm is developed in Section (7) to compute the numerical quadrature integral of μ . The application of the interpolation algorithm and the Euler-Maclaurin-based quadrature integrating algorithm to the gyrokinetic simulations is provided by Section (8), where the parallel scheme of the whole simulations are presented.

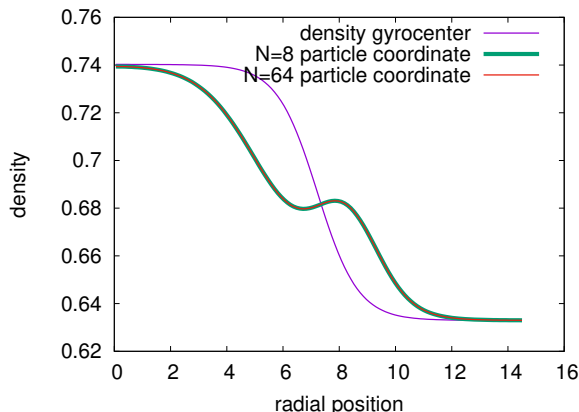


Figure 1: The equilibrium density profile on the gyrocenter coordinate before the gyroaverage operation is shown by the purple line, while the ones on the particle coordinate derived by the gyroaverage operation and with the quadrature integral of μ computed by the Euler-Maclaurin-based algorithm with the forward finite difference scheme for μ meshes of 8 and 64 nodes are presented by the other two curves. The radial distance is normalised by ρ_0 .

2 The basic orders

There are several basic orders or scales contained by the perturbation. The first one is the length scale $O(\varepsilon)$ of the nondimensionalized Larmor Radius being ε . The second one is the amplitude of the normalized electrostatic potential, whose ε -based order is

denoted as $O(\varepsilon^\sigma)$ with the power σ a positive real number. In the magnetically confined fusion plasmas, due to the fact that the charged particle can nearly migrate freely in the environment with collective interactions, the magnitude of the potential the particles feel must be much smaller than that of its kinetic energy.

The third one is the length scale of the gradient of the electrostatic potential. Define $\mathcal{K}_\perp \equiv |\frac{\nabla_\perp \phi}{\phi}|$ and $\mathcal{K}_\parallel \equiv |\frac{\nabla_\parallel \phi}{\phi}|$, where the subscript \perp and \parallel denotes the directions perpendicular and parallel the direction of equilibrium magnetic field, respectively. The gyrokinetic model of this paper adopts the scales for the perturbation

$$O\left(\varepsilon \mathcal{K}_\perp\right) = O(1), \quad O(\varepsilon \mathcal{K}_\parallel) = O(\varepsilon), \quad (1)$$

where the short scales of the perturbation are accounted for. For any equilibrium quantity $\mathcal{E} \in \{n_0, T_0\}$, this paper considers the scale

$$O\left(\varepsilon \left\| \frac{\nabla_\perp \mathcal{E}}{\mathcal{E}} \right\| \right) = O(1), \quad (2)$$

due to the steep gradient of the equilibrium profiles, as well as the following scales

$$O\left(\varepsilon \left\| \frac{\nabla_\parallel \mathcal{E}}{\mathcal{E}} \right\| \right) = O(\varepsilon), \quad O\left(\varepsilon \left\| \frac{\partial_U \mathcal{E}}{\mathcal{E}} \right\| \right) = O(\varepsilon),$$

where U is the parallel velocity and will be given later.

3 QNE incorporating the short-scale perturbation and steep equilibrium profile

3.1 The particle-coordinate density associated with the short-scale perturbation and steep equilibrium profile

The procedure to derive the gyrokinetic model is composited by two parts. The first one is to derive the coordinate transform by decoupling the gyroangle from the dynamics of other coordinates, while the second one is to obtain the gyrokinetic quasi-neutrality equation by inducing the transformation of the distribution through the derived coordinate transforms[11, 13, 2]. The first step is accomplished by implementing Lie transform perturbation method on the fundamental Lagrangian one-form. Generally, four kinds of coordinate frameworks are involved in the procedure. The first one is the full-orbit coordinate with the velocity part in Cartesian coordinates. It's denoted as $\bar{\mathbf{z}} \equiv (\mathbf{x}, \mathbf{v})$ here. The second one is obtained by transforming \mathbf{v} into the cylindrical coordinates, and it's written as $\mathbf{z} \equiv (\mathbf{x}, \mu_1, u_1, \theta_1)$ where $\mu_1 \equiv \frac{mv^2}{2B(\mathbf{x})}$, μ_1 is the velocity along the parallel direction, and θ_1 is the angle between $\boldsymbol{\rho}$ and \mathbf{e}_1 . The \mathbf{x} component in \mathbf{z} is still in full-orbit frame. The third one is the guiding-center coordinates $\bar{\mathbf{Z}} = (\bar{\mathbf{X}}, \bar{\mu}, \bar{U}, \bar{\theta})$, which is derived by decoupling $\bar{\theta}$ from the dynamics of the other coordinate components without the presence of the perturbation. The fourth one is the gyrocenter coordinate $\mathbf{Z} = (\mathbf{X}, \mu, U, \theta)$ which is derived by decoupling $\bar{\theta}$ from the dynamics of the other coordinate components with the presence of the perturbation. The coordinate transforms between $\bar{\mathbf{z}}, \mathbf{z}, \bar{\mathbf{Z}}$ and \mathbf{Z} are denoted as $\psi_f : \bar{\mathbf{z}} \rightarrow \mathbf{z}$, $\psi_{gc} : \mathbf{z} \rightarrow \bar{\mathbf{Z}}$ and $\psi_{gy} : \bar{\mathbf{Z}} \rightarrow \mathbf{Z}$, respectively, while the

distributions on the four kinds of coordinates are respectively written as $\bar{f}(\bar{\mathbf{z}})$, $f(\mathbf{z})$, $\bar{F}(\bar{\mathbf{Z}})$ and $F(\mathbf{Z})$.

The coordinate transform $\psi_{gc} : \mathbf{z} \rightarrow \bar{\mathbf{Z}}$ is given by

$$\bar{\mathbf{X}} = \mathbf{x} - \boldsymbol{\rho}_0(\mathbf{x}, \mu_1, \theta_1), \quad (3a)$$

$$\bar{\mu} = \mu_1, \quad (3b)$$

$$\bar{U} = u_1, \quad (3c)$$

$$\bar{\theta} = \theta_1. \quad (3d)$$

while the coordinate transform $\psi_{gy} : \bar{\mathbf{Z}} \rightarrow \mathbf{Z}$ is provided by the following equations

$$\mathbf{X} = \bar{\mathbf{X}}, \quad (4a)$$

$$\mu = \bar{\mu} + g_2^\mu(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta}), \quad (4b)$$

$$U = \bar{U}, \quad (4c)$$

$$\theta = \bar{\theta} + g_2^\theta(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta}). \quad (4d)$$

where

$$g_2^\mu(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta}) = \frac{q_s \Psi(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta})}{B(\bar{\mathbf{X}})}, \quad (5a)$$

$$g_2^\theta(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta}) = \frac{q_s \partial_{\bar{\mu}} \int_0^{\bar{\theta}} \Psi(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta}) d\bar{\theta}}{B(\bar{\mathbf{X}})}, \quad (5b)$$

and

$$\Psi(\bar{\mathbf{X}}, \bar{\mu}) \equiv \phi(\bar{\mathbf{X}} + \boldsymbol{\rho}_0) - \Phi(\bar{\mathbf{X}}, \bar{\mu}), \quad (6a)$$

$$\Phi(\bar{\mathbf{X}}, \bar{\mu}) \equiv \frac{1}{2\pi} \int_0^{2\pi} \phi(\bar{\mathbf{X}} + \boldsymbol{\rho}_0) d\theta, \quad (6b)$$

$$\boldsymbol{\rho}_0(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta}) = \frac{1}{q_s} \sqrt{\frac{2\bar{\mu}m_s}{B}} (\mathbf{e}_1 \cos \bar{\theta} + \mathbf{e}_2 \sin \bar{\theta}). \quad (6c)$$

And through the Lie transform perturbative method, we could derive the following Lagrangian on the gyrocenter coordinate frame

$$\mathcal{L} = (q_s \mathbf{A}(\mathbf{X}) + m_s U \mathbf{b}) \cdot \dot{\mathbf{X}} + \frac{m_s}{q_s} \mu \dot{\theta} - \left(\mu B(\mathbf{X}) + \frac{m_s U^2}{2} + \Phi(\mathbf{X}, \mu) \right), \quad (7)$$

where μ is a constant. The Euler-Lagrangian equations based on \mathcal{L} in Eq.(7) provides the equations of motion as follows:

$$\dot{\mathbf{X}} = \frac{q_s U \mathbf{B}^* - \mathbf{b} \times \nabla (\mu B + q_s \Phi(\mathbf{X}, \mu))}{q_s B_{\parallel}^*}, \quad (8a)$$

$$\dot{U} = \frac{\mathbf{B}^* \cdot \nabla (\mu B + q_s \Phi(\mathbf{X}, \mu))}{m_s B_{\parallel}^*}, \quad (8b)$$

where $\mathbf{B}^* = \mathbf{B} + \frac{m_s}{q_s} U \nabla \times \mathbf{b}$. If the denominator in Eq.(8a) is expanded by the order of the small factor $\frac{m_s}{q_s} U \nabla \times \mathbf{b}$, we could obtain the curvature drift part $\frac{m U^2 \kappa}{e B}$ where $\kappa = \frac{\nabla B}{B}$.

For the Vlasov gyrokinetic simulation, we need to transform the distribution function from the gyrocenter coordinate to the full-orbit coordinate[13]. After obtaining the coordinate transform composited by Eqs.(3a-3d) and (4a-4d), given a distribution function on the gyrocenter coordinate $F_s(\mathbf{X}, \mu, U, t)$, the distribution function on the full orbit can be derived following the transform chain

$$F_s(\mathbf{X}, \mu, U) \xrightarrow{\psi_{gy}} \bar{F}_s(\bar{\mathbf{Z}}) \xrightarrow{\psi_{gc}} f_s(\mathbf{z}). \quad (9)$$

First, the total distribution function is separated into the sum of an equilibrium one plus a perturbative one as

$$F_s(\mathbf{X}, \mu, U) = F_{s0}(\mathbf{X}, \mu, U) + F_{s1}(\mathbf{X}, \mu, U). \quad (10)$$

Then, the approximation of the distribution on the guiding-center coordinate can be derived based on the coordinate transform given by Eqs.(4a-4d)

$$\begin{aligned} \bar{F}_s(\bar{\mathbf{Z}}) &= F_s(\bar{\mathbf{X}}, \bar{\mu} + g_2^\mu(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta}), \bar{U}) \\ &\approx F_s(\bar{\mathbf{X}}, \bar{\mu}, \bar{U}) + \frac{q_s \Psi(\bar{\mathbf{Z}})}{B(\bar{\mathbf{Z}})} \partial_{\bar{\mu}} F_{s0}(\bar{\mathbf{X}}, \bar{\mu}, \bar{U}) \\ &= F_s(\bar{\mathbf{X}}, \bar{\mu}, \bar{U}) - \frac{q_s \Psi(\bar{\mathbf{Z}})}{T_s(\bar{\mathbf{Z}})} F_{s0}(\bar{\mathbf{X}}, \bar{\mu}, \bar{U}), \end{aligned} \quad (11)$$

where the equilibrium distribution involving μ is assumed as

$$F_{s0\perp} = \frac{m_s}{2\pi T_s} \exp\left(-\frac{\mu B}{T_s}\right). \quad (12)$$

The approximation of $f_s(\mathbf{z})$ is derived with Eq.(11) as the base

$$\begin{aligned} f_s(\mathbf{z}) &= F_s(\mathbf{x} - \boldsymbol{\rho}_0(\mathbf{x}, \mu_1, \theta_1), \mu_1, u_1) - q_s \frac{\phi(\mathbf{x}) F_{s0}(\mathbf{x} - \boldsymbol{\rho}_0(\mathbf{x}, \mu_1, \theta_1), \mu_1, u_1)}{T_s(\mathbf{x})} \\ &\quad + q_s \left(\frac{\Phi F_{s0}}{T_s} \right) (\mathbf{x} - \boldsymbol{\rho}_0(\mathbf{x}, \mu_1, \theta_1), \mu_1, u_1). \end{aligned} \quad (13)$$

where $\left(\frac{\Phi F_{s0}}{T_s} \right) (\mathbf{x} - \boldsymbol{\rho}_0(\mathbf{x}, \mu_1, \theta_1), \mu_1, u_1)$ is the origin of DGT and comprises the effect combining together the finite Larmor radius, short-scale perturbation and steep equilibrium profile.

We assume that the equilibrium distribution F_{s0} can be decomposed as the product between the parallel part and the perpendicular part

$$F_{s0}(\mathbf{x}, \mu_1, u_1) = n_0(\mathbf{x}) F_{s0\parallel}(\mathbf{x}, u_1) F_{s0\perp}(\mathbf{x}, \mu_1), \quad (14)$$

with probability conservation satisfied by

$$\int F_{s0\parallel} du_1 = 1, \quad (15a)$$

$$\int F_{s0\perp} \frac{B(\mathbf{x})}{m_s} d\mu_1 d\theta_1 = 1, \quad (15b)$$

where under the equilibrium condition, the metric $B(\mathbf{x})/m_s$ is used.

Then, through the integral $n_s(\mathbf{x}, t) = \int f_s(\mathbf{z}) \frac{B(\mathbf{x})}{m_s} d\mu_1 du_1 d\theta_1$, the density can be assembled as

$$n_s(\mathbf{x}) = n_{sg0}(\mathbf{x}) - q_s \frac{n_{sg0}\phi}{T_s} + q_s \tilde{\phi}'(\mathbf{x}) + n_{sg1}(\mathbf{x}, t), \quad (16)$$

with

$$n_{sg0}(\mathbf{x}) = \int F_{s0}(\mathbf{x} - \boldsymbol{\rho}_0(\mathbf{z}), \mu_1, u_1) \frac{B(\mathbf{x})}{m_s} d\mu_1 du_1 d\theta_1, \quad (17a)$$

$$n_{sg1}(\mathbf{x}, t) = \int F_{s1}(\mathbf{x} - \boldsymbol{\rho}_0(\mathbf{z}), \mu_1, u_1) \frac{B(\mathbf{x})}{m_s} d\mu_1 du_1 d\theta_1, \quad (17b)$$

$$\tilde{\phi}'(\mathbf{x}) = \int \left(\frac{\Phi F_{s0}}{T_s} \right) (\mathbf{x} - \boldsymbol{\rho}_0, \mu_1, u_1) \frac{B(\mathbf{x})}{m_s} d\mu_1 du_1 d\theta_1. \quad (17c)$$

Here, the metric $B(\mathbf{x})/m_s$ of the phase space is used.

In this paper, we sometimes use the symbols $\mathcal{J}_1(\mu)$ and $\mathcal{J}_2(\mu)$ to denote the first and the second gyroaverage associated with the magnetic moment μ , specifically,

$$\mathcal{J}_1(\mu)\phi(\mathbf{x}) = \frac{1}{2\pi} \int \phi(\mathbf{x} + \boldsymbol{\rho}(\mu)) d\theta,$$

$$\mathcal{J}_2(\mu)\phi(\mathbf{x}) = \frac{1}{2\pi} \int \phi(\mathbf{x} - \boldsymbol{\rho}(\mu)) d\theta.$$

3.2 QNE with respect to the adiabatic distribution of electron and the steep equilibrium profile on gyrocenter-coordinate frame

As shown in Fig.(1), the equilibrium density profile possessing the steep gradient on the gyrocenter coordinate frame is obviously different from the one on the particle-coordinate frame after the gyroaverage operation. Therefore, contrary to the traditional way, the implementation of the equilibrium density on the particle-coordinate frame should be different from the one on the gyrocenter-coordinate frame if its profile is step in the radial dimension.

We consider a plasma only including protons and electrons. The electrons obey the adiabatic distribution on the particle-coordinate space

$$n_e(\mathbf{x}) = n_{g0}(\mathbf{x}) + \frac{en_{g0}(\mathbf{x})}{T_e} \phi(\mathbf{x}), \quad (18)$$

where the flux-surface average of the potential is not deducted from the total potential, because our purpose is to compare the two solvers. $n_{g0}(\mathbf{x})$ is the equilibrium density in the particle-coordinate space and needs to be solved. The initial equilibrium distribution of ion is given on the guiding-center coordinate system by

$$F_{i0}(\mathbf{x}, \mu, U) = \frac{n_0(\mathbf{x})}{(2\pi T_i(\mathbf{x}))^{3/2}} \exp\left(-\frac{U^2}{2T_i(\mathbf{x})} - \frac{\mu B(\mathbf{x})}{T_i(\mathbf{x})}\right),$$

from which the equilibrium density on the guiding-center coordinate is easily derived as $n_0(\mathbf{x})$. For the scales given by Eq.(2), the equilibrium density on the particle-coordinate space should be derive as

$$n'_0 = \int F_{i0}(\mathbf{x} - \boldsymbol{\rho}, \mu, U) \frac{B}{m_i} d\mu_1 dU d\theta \quad (19)$$

Taking into account of the quasi-neutrality equilibrium, it's derived that

$$n_{g0}(\mathbf{x}) = n'_0(\mathbf{x}). \quad (20)$$

Then, QNE of this plasma is

$$\frac{e\phi}{T_i} - \frac{e\tilde{\phi}'}{n_{g0}} + \frac{e\phi}{T_e} = \frac{n_{i1}}{n_{g0}}. \quad (21)$$

3.3 Normalization

In this paper, the quantities t, v, B, l, μ, T, ϕ are normalized by $t_0 \equiv \frac{m}{B_0 q_i}$, $v_0 \equiv \sqrt{\frac{T_{e0}}{m_i}}$, B_0 , $l_0 \equiv \frac{mv_0}{eB_0}$, $\mu_0 \equiv \frac{T_{e0}}{B_0}$, T_{e0} and $\phi_0 \equiv \frac{T_{e0}}{q_i}$, respectively, where $T_{e0} \equiv T_e(r_p)$ and $r_p \in [r_{min}, r_{max}]$ is the radial position of the peak of the initial distribution function. Then, by choosing the equilibrium perpendicular distribution given by Eq.(12), the normalized QNE is

$$\frac{\phi(\mathbf{x})}{T_i(\mathbf{x})} - \frac{\tilde{\Phi}(\mathbf{x})}{n_{g0}(\mathbf{x})} + \frac{\phi(\mathbf{x})}{T_e(\mathbf{x})} = \frac{n_{i1}(\mathbf{x})}{n_{g0}(\mathbf{x})}, \quad (22)$$

with the normalized DGT

$$\tilde{\Phi}(\mathbf{x}) = \int \left(\frac{\Phi F_{i0}}{T_i} \right) (\mathbf{x} - \boldsymbol{\rho}_0, \mu_1, u_1) B(\mathbf{x}) d\mu_1 du_1 d\theta_1, \quad (23)$$

$$\text{and } F_0 = \frac{n_0(r) \exp\left(-\frac{v^2}{2T_i(r)} - \frac{\mu B}{T_i(r)}\right)}{(2\pi T_i(r))^{3/2}}.$$

4 2nd-order approximation of QNE

We first need to take Taylor expansion of DGT. According to Appendix A, the expansion rule of a general function $\mathcal{A}(\mathbf{x} + \boldsymbol{\rho}_0)$ is

$$\mathcal{A}(\mathbf{x} + \boldsymbol{\rho}_0) = \sum_{n \geq 0} \frac{(\boldsymbol{\rho}_0 \cdot \nabla)^n}{n!} \mathcal{A}(\mathbf{x}).$$

The gyroaverage of the following terms are needed

$$\begin{aligned} \langle (\boldsymbol{\rho}_0 \cdot \nabla) \mathcal{A}(\mathbf{x}) \rangle &= 0, \\ \langle (\boldsymbol{\rho}_0 \cdot \nabla)^2 \mathcal{A}(\mathbf{x}) \rangle &= \frac{1}{2} \rho_0^2 \nabla_{\perp}^2 \mathcal{A}(\mathbf{x}), \end{aligned}$$

where

$$\langle \mathcal{A}(\mathbf{x} + \boldsymbol{\rho}_0) \rangle \equiv \frac{1}{2\pi} \int \mathcal{A}(\mathbf{x} + \boldsymbol{\rho}_0) d\theta.$$

The following identity to reduce the perpendicular distribution to the density is also necessary

$$\int \rho_0^2 \partial_{\mu} F_{i0\perp} B d\mu = \frac{-1}{B\pi},$$

The first gyroaverage expanded up to the second order is

$$\Phi(\mathbf{X}, \mu) \equiv \langle \phi(\mathbf{X} + \boldsymbol{\rho}_0) \rangle \approx \left(1 + \frac{\rho_0^2 \nabla_{\perp}^2}{4} \right) \phi(\mathbf{X}). \quad (26)$$

The second gyroaverage expanded up to the second order is

$$\begin{aligned}
\left\langle \left(\frac{\Phi F_{i0}}{T_i} \right) (\mathbf{x} - \boldsymbol{\rho}_0, \mu_1, u_1) \right\rangle &\approx \left(1 + \frac{\rho_0^2 \nabla_{\perp}^2}{4} \right) \left[\frac{F_{i0}}{T_i} \left(1 + \frac{\rho_0^2 \nabla_{\perp}^2}{4} \right) \phi \right] \\
&\approx \frac{F_{i0} \phi}{T_i} + \frac{\rho_0^2 \nabla_{\perp}^2}{4} \left(\frac{F_{i0} \phi}{T_i} \right) + \frac{F_{i0}}{T_i} \frac{\rho_0^2 \nabla_{\perp}^2}{4} \phi \\
&\approx \frac{F_{i0} \phi}{T_i} + \frac{\rho_0^2 \nabla_{\perp} \phi}{4} \cdot \nabla_{\perp} \left(\frac{F_{i0}}{T_i} \right) + \frac{F_{i0}}{T_i} \frac{\rho_0^2 \nabla_{\perp}^2}{2} \phi
\end{aligned} \tag{27}$$

The density derived from Eq.(27) is

$$2\pi \int \left\langle \left(\frac{\Phi F_{i0}}{T_i} \right) (\mathbf{x} - \boldsymbol{\rho}_0, \mu, U) \right\rangle B d\mu dU \approx \frac{n_0 \phi}{T_i} + \frac{\nabla_{\perp} \phi \cdot \nabla_{\perp} n_0}{2B^2} + \frac{n_0}{B^2} \nabla_{\perp}^2 \phi. \tag{28}$$

Then, QNE derived by the 2nd-order approximation is

$$\frac{\phi}{T_i} - \frac{n_0 \phi}{n_{g0} T_i} - \frac{\nabla_{\perp} \phi \cdot \nabla_{\perp} n_0}{2n_{g0} B^2} - \frac{n_0}{n_{g0} B^2} \nabla_{\perp}^2 \phi + \frac{\phi}{T_e} = \frac{n_{i1}}{n_{g0}}. \tag{29}$$

With the assumption of $B = 1$, Eq.(29) becomes

$$\frac{\phi}{T_i} - \frac{n_0 \phi}{n_{g0} T_i} - \frac{\nabla_{\perp} \phi \cdot \nabla_{\perp} n_0}{2n_{g0}} - \frac{n_0}{n_{g0}} \nabla_{\perp}^2 \phi + \frac{\phi}{T_e} = \frac{n_{i1}}{n_{g0}}. \tag{30}$$

In Eq.(30), the gradient term of n_0 contains a factor of 1/2, which doesn't exist in the standard gyrokinetic model. If we replace the last term on the right side of Eq.(13) by a term of the form $\Phi(\mathbf{x} - \boldsymbol{\rho}) \left(\frac{F_{s0}}{T_s} \right) (\mathbf{x} - \boldsymbol{\rho})$, the expansion of that term doesn't provide the 1/2 factor. However, as the derivation to obtain Eq.(13) illuminates, the last term in Eq.(13) is the correct one.

5 QNE solver comprising the interpolation algorithm to compute DGT

5.1 QNE solver

To obtain the QNE solver comprising of the interpolation algorithm, QNE in Eq.(22) is rewritten as

$$\frac{n_{g0}(\mathbf{x}) \phi(\mathbf{x})}{T_i(\mathbf{x})} - \tilde{\Phi}(\mathbf{x}) + \frac{n_{g0}(\mathbf{x}) \phi(\mathbf{x})}{T_e(\mathbf{x})} = n_{i1}(\mathbf{x}). \tag{31}$$

$\tilde{\Phi}(\mathbf{x})$ can be written as the discrete sum over μ and U as

$$\tilde{\Phi}(\mathbf{x}) = \sum_k \sum_j \tilde{\Phi}(\mathbf{x}, \mu_j, U_k) B \delta \mu_j \delta U_k,$$

where $\delta \mu_j$ and δU_k are the step length for μ and U . Here,

$$\tilde{\Phi}(\mathbf{x}, \mu_j, U_k) \equiv \frac{1}{2\pi} \int (\Phi_j \mathcal{F}_{sjk})(\mathbf{x} - \boldsymbol{\rho}_{0j}, \mu_j, U_k) d\theta \tag{32}$$

and in cylindrical coordinate frame, \mathcal{F}_{sjk} is defined

$$\mathcal{F}_{sjk} \equiv \frac{n_0(r) \exp\left(-\frac{U_k^2}{2T_s(r)} - \frac{\mu_j B}{T_s(r)}\right)}{(2\pi T_s(r))^{1/2} T_s^2(r)}.$$

$\Phi_j(r_h, \Theta_p)$ is used to denote the value at mesh node (r_h, Θ_p) of $\Phi(\mathbf{X}, \mu_j)$ as the first gyroaverage of ϕ with respect to the magnetic moment μ_j . It is computed on the grid discretized by the following right Riemann sum:

$$\Phi_j(r_h, \Theta_p) = \frac{1}{N_\theta} \sum_{l=0}^{N_\theta-1} \phi\left(r_h \cos \Theta_p + \rho_j \cos\left(\frac{2\pi l}{N_\theta}\right), r_h \sin \Theta_p + \rho_j \sin\left(\frac{2\pi l}{N_\theta}\right)\right), \quad (33)$$

where $\rho_j = \sqrt{2\mu_j}$. In the same way, $\tilde{\Phi}_j(r_h, \Theta_p)$ is used to denote the value at grid point (r_h, Θ_p) of $\tilde{\Phi}(\mathbf{X}, \mu_j)$ as the second gyroaverage of ϕ with respect to the magnetic moment μ_j . It is also approximated by the following right Riemann sum:

$$\tilde{\Phi}_{jk}(r_h, \Theta_p) = \frac{1}{N_\theta} \sum_{l=0}^{N_\theta-1} \bar{\Phi}_{jk}\left(r_h \cos \Theta_p - \rho_j \cos\left(\frac{2\pi l}{N_\theta}\right), r_h \sin \Theta_p - \rho_j \sin\left(\frac{2\pi l}{N_\theta}\right)\right), \quad (34)$$

where

$$\bar{\Phi}_{jk} = \Phi_j \mathcal{F}_{sjk}.$$

The respective symbols $+$ and $-$ in Eq.(42) and Eq.(34) should be paid attention. The cubic splines interpolation is used to calculate ϕ at the interpolated point on the Larmor circle. The radial projection on the boundaries for the points outside the domain is used and we consider 2π -periodic condition in Θ .

In order to detail the steps of the solver, we note

$$\begin{aligned} \phi_{h,p} &:= \phi(r_h, \Theta_p), h \in \{0, \dots, N_r\}; p \in \{0, \dots, N_\Theta - 1\} \\ \Phi_{h,p,j} &:= \Phi_j(r_h, \Theta_p), j \in \{1, \dots, N_\mu\} \\ \tilde{\Phi}_{h,p,j} &:= \tilde{\Phi}_j(r_h, \Theta_p), \\ \phi &:= (\phi_{0,0}, \dots, \phi_{N_r,0}, \phi_{0,1}, \dots, \phi_{N_r,1}, \dots, \phi_{0,N_\Theta-1}, \dots, \phi_{N_r,N_\Theta-1})^t, \\ \Phi_j &:= (\Phi_{0,0,j}, \dots, \Phi_{N_r,0,j}, \Phi_{0,1,j}, \dots, \Phi_{N_r,1,j}, \dots, \Phi_{0,N_\Theta-1,j}, \dots, \Phi_{N_r,N_\Theta-1,j})^t, \\ \mathcal{F}_{sjk} &:= (\mathcal{F}_{0,0,jk}, \dots, \mathcal{F}_{N_r,0,jk}, \mathcal{F}_{0,1,jk}, \dots, \mathcal{F}_{N_r,1,jk}, \dots, \mathcal{F}_{0,N_\Theta-1,jk}, \dots, \mathcal{F}_{N_r,N_\Theta-1,jk})^t, \\ \bar{\Phi}_{jk} &:= (\bar{\Phi}_{0,0,jk}, \dots, \bar{\Phi}_{N_r,0,jk}, \bar{\Phi}_{0,1,jk}, \dots, \bar{\Phi}_{N_r,1,jk}, \dots, \bar{\Phi}_{0,N_\Theta-1,jk}, \dots, \bar{\Phi}_{N_r,N_\Theta-1,jk})^t, \\ \tilde{\Phi}_{jk} &:= (\tilde{\Phi}_{0,0,jk}, \dots, \tilde{\Phi}_{N_r,0,jk}, \tilde{\Phi}_{0,1,jk}, \dots, \tilde{\Phi}_{N_r,1,jk}, \dots, \tilde{\Phi}_{0,N_\Theta-1,jk}, \dots, \tilde{\Phi}_{N_r,N_\Theta-1,jk})^t. \end{aligned}$$

So, $\bar{\Phi}_{jk}$ can be written as the vector product between Φ_{jk} and \mathcal{F}_{sjk} as $\mathcal{F}_{sjk} \cdot \Phi_j$.

The specific procedure is given by:

1. Construction of the matrix $A^{spl} \in \mathcal{M}_{(N_r+1) \times (N_\Theta), (N_r+1) \times N_\Theta}$ such that $S = A^{spl} \phi$ is the vector of splines coefficients. Alternatively, ϕ can be written as $\phi = (A^{spl})^{-1} S$. The matrix A^{spl} is independent of the Larmor radius or the magnetic moment μ_j , while depending on the equilibrium quantities.

2. For a Larmor radius $\rho_j = \sqrt{2\mu_j}$ contributed by the magnetic moment μ_j , constructing the matrix $A_{l,\rho_j}^{contr} \in \mathcal{M}_{(N_r+1) \times N_\Theta, (N_r+1) \times N_\Theta}$ which gives the contribution of the gyroaverage of the radius ρ_j as the function of the splines coefficients. Here, $l = 1, 2$ denoting the first and the second gyroaverage, respectively.

3. The first gyroaverage is given by $\Phi_j = G_{1,\rho_j} \phi$, with $G_{1,\rho_j} \equiv A_{l,\rho_j}^{contr} A^{spl}$.

4. The second gyroaverage is written as the matrix form $\tilde{\Phi}_{jk} = G_{2,\rho_j} \bar{\Phi}_{jk}$, with $G_{2,\rho_j} \equiv A_{l,\rho_j}^{contr} A^{spl}$. To realise the matrix form between $\tilde{\Phi}_{jk}$ and ϕ , the vector product $\mathcal{F}_{sjk} \cdot \Phi_j$ can be written as the product between a diagonal matrix denoted as \mathbb{F}_{sjk} and Φ_j . Eventually, $\tilde{\Phi}_{jk}$ can be written as the matrix form

$$\tilde{\Phi}_{jk} = G_{2,\rho_j} \mathbb{F}_{sjk} G_{1,\rho_j} \phi. \quad (35)$$

Note that the elements of matrix $G_{2,\rho_j} \mathbb{F}_{sjk} G_{1,\rho_j}$ only depends on the equilibrium quantities. It can be computed once for all at the beginning of the simulation.

5. Then, the integral of μ contained by the double gyroaverage term can be formulated as the discrete sum

$$\tilde{\Phi} = \sum_{k=1}^{N_U} \sum_{j=1}^{N_\mu} \tilde{\Phi}_{jk} B \delta\mu_j \delta U_k$$

We construct the matrix of the quasi-neutrality operator in the Fourier basis. In fact, due to the periodic structure in the Θ dimension, A^{spl} and A_{l,ρ_j}^{contr} are recognized to have circulant block structure[17], which in Fourier basis can be transformed as block diagonal matrixes. More precisely, the matrixes A^{spl} , A_{l,ρ_j}^{contr} and \mathbb{F}_{s0jk} have the following structure

$$A^{spl} = \begin{bmatrix} A_0^{spl} & A_1^{spl} & \cdots & A_{N_{\Theta}-1}^{spl} \\ A_{N_{\Theta}-1}^{spl} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & A_1^{spl} \\ A_1^{spl} & \cdots & A_{N_{\Theta}-1}^{spl} & A_0^{spl} \end{bmatrix} \in \mathcal{M}_{(N_r+1) \times N_{\Theta}, (N_r+1) \times N_{\Theta}}(\mathbb{R}) \quad (36a)$$

$$A_{l,\rho_j}^{contr} = \begin{bmatrix} A_{l,\rho_j,0}^{contr} & A_{l,\rho_j,1}^{contr} & \cdots & A_{l,\rho_j,N_{\Theta}-1}^{contr} \\ A_{l,\rho_j,N_{\Theta}-1}^{contr} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & A_{l,\rho_j,1}^{contr} \\ A_{l,\rho_j,1}^{contr} & \cdots & A_{l,\rho_j,N_{\Theta}-1}^{contr} & A_{l,\rho_j,0}^{contr} \end{bmatrix} \in \mathcal{M}_{(N_r+1) \times N_{\Theta}, (N_r+1) \times N_{\Theta}}(\mathbb{R}) \quad (36b)$$

$$\mathbb{F}_{sjk} = \begin{bmatrix} \mathbb{F}_{s0jk} & 0 & \cdots & 0 \\ 0 & \mathbb{F}_{s0jk} & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbb{F}_{s0jk} \end{bmatrix} \in \mathcal{M}_{(N_r+1) \times N_{\Theta}, (N_r+1) \times N_{\Theta}}(\mathbb{R}) \quad (36c)$$

And \mathbb{F}_{s0jk} is

$$\mathbb{F}_{s0jk} = \begin{bmatrix} \mathcal{F}_{0jk} & 0 & \cdots & 0 \\ 0 & \mathcal{F}_{1jk} & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathcal{F}_{N_rjk} \end{bmatrix} \in \mathcal{M}_{N_r \times N_r}(\mathbb{R}) \quad (37)$$

Due to that \mathbb{F}_{sjk} is a diagonal matrix with only the main diagonal elements not equaling zero, the following arrangements of the product order in Eq.(35) equal

$$(G_{2,\rho_j} \mathbb{F}_{sjk}) G_{1,\rho_j} = G_{2,\rho_j} (\mathbb{F}_{sjk} G_{1,\rho_j}).$$

The product order on the left is chosen in this paper. Then, the product $A^{spl}\mathbb{F}_{sjk}$ is further denoted as \mathcal{A}_{jk}^{spl} in terms of the following expression

$$\mathcal{A}_{jk}^{spl} = \begin{bmatrix} \mathcal{A}_{0jk}^{spl} & \mathcal{A}_{1jk}^{spl} & \cdots & \mathcal{A}_{(N_\Theta-1)jk}^{spl} \\ \mathcal{A}_{(N_\Theta-1)jk}^{spl} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathcal{A}_{1jk}^{spl} \\ \mathcal{A}_{1jk}^{spl} & \cdots & \mathcal{A}_{(N_\Theta-1)jk}^{spl} & \mathcal{A}_{0jk}^{spl} \end{bmatrix}, \quad (38)$$

where

$$\mathcal{A}_{hjk}^{spl} = A_h^{spl}\mathbb{F}_{s0jk}, h \in \{0, 1, \dots, N_\Theta - 1\}.$$

The three matrixes are diagonalisable in the Fourier basis as

$$\begin{aligned} A^{spl} &= U_n D^{spl} U_n^*, \\ A_{l,\rho_j}^{contr} &= U_n D_{l,\rho_j}^{contr} U_n^*, \\ \mathcal{A}_{jk}^{spl} &= U_n \mathcal{D}_{jk}^{spl} U_n^*, \end{aligned}$$

where $*$ means the complex conjugate and the other matrixes are defined as follows

$$\begin{aligned} D^{spl} &= \begin{pmatrix} D_0^{spl} & & \\ & \ddots & \\ & & D_{N_\Theta-1}^{spl} \end{pmatrix}, \\ D_{l,\rho_j}^{contr} &= \begin{pmatrix} D_{l,\rho_j,0}^{contr} & & \\ & \ddots & \\ & & D_{l,\rho_j,N_\Theta-1}^{contr} \end{pmatrix}, \\ \mathcal{D}_{jk}^{spl} &= \begin{pmatrix} \mathcal{D}_{jk,0}^{spl} & & \\ & \ddots & \\ & & \mathcal{D}_{jk,N_\Theta-1}^{spl} \end{pmatrix}. \end{aligned}$$

$$\begin{aligned} D_m^{spl} &= \sum_{h=0}^{N_\Theta-1} A_h^{spl} e^{\frac{-2i\pi hm}{N_\Theta}}, \\ D_{l,\rho_j,m}^{contr} &= \sum_{h=0}^{N_\Theta-1} A_{l,\rho_j,h}^{contr} e^{\frac{-2i\pi hm}{N_\Theta}}, \\ \mathcal{D}_{jk,m}^{spl} &= \sum_{h=0}^{N_\Theta-1} \mathcal{A}_{hjk}^{spl} e^{\frac{-2i\pi hm}{N_\Theta}}, \end{aligned}$$

$$U_n = \begin{pmatrix} U_{n,0,0} & \cdots & U_{n,0,N_\Theta-1} \\ \vdots & \ddots & \vdots \\ U_{n,N_\Theta-1,0} & \cdots & U_{n,N_\Theta-1,N_\Theta-1} \end{pmatrix}, \quad U_{n,h,l} = \frac{1}{\sqrt{N_\Theta}} e^{\frac{2i\pi hl}{N_\Theta}} I_n, \quad (39)$$

where I_n is the identity matrix of size $n \times n$.

As explained before, the value of $\phi(\mathbf{x})$ and $n_1(\mathbf{x})$ on the grid points can be assembled to be the vectors ϕ and n_1 . Then, the first term and third term of Eq.(31) can be written

in a matrix form with only the main diagonal elements of the coefficient matrix unequal to zero.

Due to the circulant block structure of the matrix \mathcal{A}_{hjk}^{spl} , A^{spl} , A_{l,ρ_j}^{contr} and the coefficient matrixes of the first term and third term on the left of Eq.(31), the computation of the QNE solver comprising gyroaverage matrix $G_{2,\rho_j}\mathbb{F}_{sjk}G_{1,\rho_j}$ can be performed using a fast algorithm:

1. Change to Fourier basis by $\text{FFT}(\phi)$.
2. Compute the product $G_{2,\rho_j}\mathbb{F}_{sjk}G_{1,\rho_j}\phi$ to obtain the total coefficient matrix of the left side of QNE.
3. Use the subroutine of Lapack and Blas to compute the inverse matrix of the coefficient matrix.
4. Change the results to the real space by FFT^{-1} .

The use of polar mesh and FFT allows to make more quickly computations and provides a base for a future work in more complex geometry.

5.2 The algorithm to compute A^{spl} and A_{l,ρ_j}^{contr}

In our interpolation scheme, the following cubic spline function is used as the basis function

$$B_\alpha(x) = \frac{1}{6h^3} \begin{cases} (x - x_{\alpha-2})^3 & \text{if } x_{\alpha-2} \leq x \leq x_{\alpha-1} \\ h^3 + 3h^2(x - x_{\alpha-1}) + 3h(x - x_{\alpha-1})^2 - 3(x - x_{\alpha-1})^3 & \text{if } x_{\alpha-1} \leq x \leq x_\alpha \\ h^3 + 3h^2(x_{\alpha+1} - x) + 3h(x_{\alpha+1} - x)^2 - 3(x_{\alpha+1} - x)^3 & \text{if } x_\alpha \leq x \leq x_{\alpha+1} \\ (x_{\alpha+2} - x)^3 & \text{if } x_{\alpha+1} \leq x \leq x_{\alpha+2} \\ 0 & \text{otherwise} \end{cases}$$

The potential on the grid point (h, p) is

$$\phi_{h,p} = \sum_{a,b=-1}^2 C_{h+a,p+b} B_{h+a}(x_{1h}) B_{p+b}(x_{2p}), \quad (40)$$

where x_1 and x_2 can be the coordinates in the Cartesian frame and in the polar frame. $C_{h+a,p+b}$ is the weight of the two dimensional basis $B_{h+a}B_{p+b}$. $B_{h+a}(x_{1h})/B_{p+b}(x_{2p})$ is the value of the one dimensional basis B_{h+a}/B_{p+b} at the node x_{1h}/x_{2p} . By assembling $\phi_{h,p}$ into the vector ϕ as explained before, Eq.(40) can be rewritten as the tensor product of two matrixes multiplying the vector C comprising the weight coefficients of all the basises

$$\phi = \mathcal{B}_P(x_2) \otimes \mathcal{B}_b(x_1)C, \quad (41)$$

where

$$C \equiv (C_{0,0}, \dots, C_{N_r,0}, C_{0,1}, \dots, C_{N_r,1}, \dots, C_{0,N_\Theta-1}, \dots, C_{N_r,N_\Theta-1})^t,$$

and the subscript P denotes the periodic boundary condition used in the x_2 dimension, while b denotes the periodic or natural boundary condition used in x_1 dimension. $\mathcal{B}_b(x_1)$ and $\mathcal{B}_P(x_2)$ are the matrixes representing the value of $B_h(x_1)$, $h \in \{0, 1, 2, \dots, N_r\}$ and $B_p(x_2)$, $p \in \{0, 1, \dots, N_\Theta - 1\}$ on the nodes of the x_1 and x_2 domains, respectively. Here,

$\mathcal{B}_P(x_2)$ is of the structure proportional to

$$\begin{bmatrix} 4 & 1 & 0 & \cdots & 1 \\ 1 & 4 & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ 1 & \cdots & 0 & 1 & 4 \end{bmatrix}.$$

Because the rows of $\mathcal{B}_P(x_2)$ have the circulant property, the matrix $\mathcal{B}_P(x_2) \otimes \mathcal{B}_b(x_1)$ is of the circulant block structure. Its inverse can be written as

$$A^{spl} \equiv (\mathcal{B}_P(x_2) \otimes \mathcal{B}_b(x_1))^{-1} = \mathcal{B}_P^{-1}(x_2) \otimes \mathcal{B}_b^{-1}(x_1).$$

which is still a circulant block matrix. Then, we derive

$$A_p^{spl} = B_P^{-1}(1, p) \mathcal{B}_b^{-1},$$

where A_p^{spl} is defined in Eq.(36a) and $B_P^{-1}(1, p)$ is the $(1, p)$ th element of \mathcal{B}_P^{-1} .

For the gyroaverage operation, Eqs.(42) and (34) can be rewritten in the following form

$$\begin{aligned} \mathcal{J}_{lj} \phi(x_{1h}, x_{2p}) &= \frac{1}{N_\theta} \sum_{m=0}^{N_\theta-1} \phi(\mathcal{C}_{lj, hpm, 1}, \mathcal{C}_{lj, hpm, 2}) \\ &= \frac{1}{N_\theta} \sum_{m=0}^{N_\theta-1} \sum_{a, b=-1}^2 C_{lj, D_{hpma}, D_{hpm b}} val(a, b). \end{aligned} \quad (42)$$

The global coordinates of the m th interpolated points on the Larmor circle of the grid point (h, p) are computed as

$$\mathcal{C}_{lj, hpm, 1} = \mathcal{C}_{\min, 1} + (d_{hpm, 1} - 1) \delta_1 + c_1, \quad (43a)$$

$$\mathcal{C}_{lj, hpm, 2} = \mathcal{C}_{\min, 2} + (d_{hpm, 2} - 1) \delta_2 + c_2. \quad (43b)$$

In Eqs.(43a) and (43b), $\mathcal{C}_{\min, 1}$ and $\mathcal{C}_{\min, 2}$ are the respective minimal value of the x_1 and x_2 domain; δ_1 and δ_2 are the uniform step length in the respective dimension; $0 \leq c_1 < \delta_1$ and $0 \leq c_2 < \delta_2$ hold. $d_{hpm, 1}$ and $d_{hpm, 2}$ are the global nodes numbers of the interpolated point on the Larmor circle denoted by the indexes h, p, m . $\mathcal{C}_{lj, hpm, 1}$ and $\mathcal{C}_{lj, hpm, 2}$ can be the global coordinates in the Cartesian or polar coordinate frames. In the 2nd equality of Eq.(42), the potential at the interpolated point $(\mathcal{C}_{lj, hpm, 1}, \mathcal{C}_{lj, hpm, 2})$ is written by the sum of the contribution of the cubic spline basis. We use D_{hpma} and $D_{hpm b}$ to denote the global node number of the grid point associated with the indexes (a, b) . This grid point is where the basis function locates. Then, $val(a, b)$ as the spline function value at $(\mathcal{C}_{lj, hpm, 1}, \mathcal{C}_{lj, hpm, 2})$ is given by

$$val(a, b) \equiv B_{D_{hpma}}(c_1) B_{D_{hpm b}}(c_2).$$

$C_{lj, D_{hpma}, D_{hpm b}}$ in Eq.(42) is the weight coefficient of the cubic spline function locating at $(D_{hpma}, D_{hpm b})$. Eventually, by summing the repeated contribution nodes together, the gyroaverage of ϕ could be written into a matrix form

$$\mathcal{J}_{lj} \phi = A_{l, \rho_j}^{contr} C_{lj}.$$

Here, we want to mention that due to the finite value of μ_j , some interpolated points on the Larmor circle of the grid point close to the boundary could locate out of the chosen domain. At the out boundary, for a realistic plasma, the amplitude of the fluctuation of the potential is close to zero. So we approximate the contribution of such points by the contribution of their projection points on the boundary.

The elements of A_{l,ρ_j}^{contr} can be denoted as $A_{l,\rho_j}^{contr}((p_1 - 1) * (N_r + 1) + h_1, (p_2 - 1) * (N_r + 1) + h_2)$ with $p_1, p_2 \in \{0, \dots, N_r - 1\}$ and $h_1, h_2 \in \{0, \dots, N_\Theta - 1\}$. To compute A_{l,ρ_j}^{contr} , due to its circulant block structure, we only need to compute the block matrixes associated with $p_1 = 1$. The other rows of block matrixes associated with $p_1 \neq 1$ can be obtained by permuting the indexes p_2 s for $p_1 - 1$ times, as shown in Eq.(36b). The algorithm to compute $A_{l,\rho_j}^{contr}(h_1, (p_2 - 1) * (N_r + 1) + h_2)$ is given by Algorithm 1.

Algorithm 1 Compute $A_{l,\rho_j}^{contr}(h_1, (p_2 - 1) * N_r + h_2)$

Input: $\mu_j, l, \delta_1, \delta_2, N_r, N_\Theta, N_\theta$, **boundary condition**

```

1: for  $h_1 = 0, N_r$  do
2:   for  $m = 1, N_\theta$  do
3:     get  $d_{hpm,1}, d_{hpm,2}$ 
4:     get  $c_1, c_2$ 
5:     get  $val(4, 4)$ 
6:     for  $a = -1, 2$  do
7:       for  $b = -1, 2$  do
8:         get  $D_{hpm,a}, D_{hpm,b}$ 
9:          $A_{l,\rho_j}^{contr}(h_1, D_{hpm,a}, D_{hpm,b}) = A_{l,\rho_j}^{contr}(h_1, D_{hpm,a}, D_{hpm,b}) + val(a, b)$ 

```

6 Benchmark of the interpolation algorithm for the single μ case

To benchmark the interpolation algorithm, we integrate the absolute value of the error of the interpolation solver deviating from the analytical solution over the simulated domain, specifically,

$$\text{abs error} = \sum_{h,p} | (\text{interp})_{hp} - (\text{anal})_{hp} | \delta x_1 \delta x_2, \quad (44)$$

where $(\text{interp})_{hp}$ and $(\text{anal})_{hp}$ denote the value computed by the interpolation solver and by the analytical solution at the mesh node (h, p) , respectively. δx_1 and δx_2 are the uniform steps in x_1 and x_2 dimensions. For some cases, we also implement the same scheme to the 2nd-order solver.

6.1 1st example: double periodic boundary condition in Cartesian coordinate

In this example, the periodic boundary condition is used in both x_1 and x_2 dimensions. According to Eqs.(67) and (68), for the function $f(\mathbf{x}) = \cos(nx_1 + mx_2)$, the exact solution of $\mathcal{J}_1(\mu)f(\mathbf{x})$ and $\mathcal{J}_2(\mu)\mathcal{J}_1(\mu)f(\mathbf{x})$ are

$$\mathcal{J}_1(\mu) \cos(nx_1 + mx_2) = J_0(\rho(\mu)\sqrt{n^2 + m^2}) \cos(nx_1 + mx_2),$$

$$\mathcal{J}_2(\mu)\mathcal{J}_1(\mu)\cos(nx_1 + mx_2) = J_0^2(\rho(\mu)\sqrt{n^2 + m^2})\cos(nx_1 + mx_2).$$

Two cases are computed. For the first one, $n = 0, m = 2$ are used, and $\mu = 0.02, N_\theta = 100$, while for the second one, $n = 5, m = 5$ are chosen. For both cases, we compute the domain $(0, 2\pi) \times (0, 2\pi)$ which are divided into three meshes of $(16, 16), (32, 32), (64, 64)$ cells, respectively. The results are shown in Tables 1 and 2. It's found that compared with the 2nd-order solver, the interpolation algorithm has better accuracy and its results present the converging rate of the fourth-order accuracy, which is provided by the cubic spline interpolation.

Table 1: double per, Cartesian
 $m = 2, n = 0, \mu = 0.02, N_\theta = 100$

mesh	(16,16)	(32,32)	(64,64)
interpo,first	1.7084E-2	6.4524E-4	4.1522E-5
2nd,first	0.8738	0.9635	0.9872
interpo,double	3.2803E-2	1.2393E-3	7.9757E-5
2nd,double	1.7102	1.8882	1.9352

Table 2: double per, Cartesian
 $m = 5, n = 5, \mu = 0.02, N_\theta = 100$

mesh	(16,16)	(32,32)	(64,64)
interpo,first	1.3114	4.0744E-2	2.0800E-3
2nd,first	10.2513	10.8218	11.0119
interpo,double	1.3972	4.5496E-2	2.3258E-3
2nd,double	115.6807	16.7745	17.1429

6.2 2nd example: Natural + Periodic boundary condition in Cartesian coordinate

For this case, the periodic boundary condition is used in x_2 dimension, while the natural boundary condition is used in x_1 dimension. The function under gyroaverage is $f(\mathbf{x}) = x_2$. Its first gyroaverage and double gyroaverage are

$$\mathcal{J}_1(f(\mathbf{x})) = x_2, \mathcal{J}_2\mathcal{J}_1(f(\mathbf{x})) = x_2.$$

For this case, $\mu = 0.05$. The simulated domain $(0, 4) \times (0, 2\pi)$ are divided into two meshes including 32×32 and 64×64 cells, respectively. The integrals of the absolute value of the error over the simulated domain for the first and double gyroaverage are presented in Table 3. To obtain this, we ignored the boundary rows in the x_1 dimension, for which the natural boundary condition is used. The fact that the abs error of the 2nd-order solver is zero is due to that the first gyroaverage and the double one of $f(\mathbf{x})$ depends on the gradient of $f(\mathbf{x})$ only through $\nabla_\perp^2 f(\mathbf{x})$, as Eq.(26) shows. $\nabla_\perp^2 f(\mathbf{x})$ in Cartesian coordinates is written as $(\partial_{x_1}^2 + \partial_{x_2}^2)f(\mathbf{x})$. The second order derivative is approximated by the centered finite difference as $\partial_{x_1}^2 f(\mathbf{x}) = \frac{f_{j+1,k} - 2f_{j,k} + f_{j-1,k}}{\delta x_1^2}$ with the 2nd-order precision, where $f_{j+1,k} \equiv f(x_{1,j+1}, x_{2,k})$. It's obvious that the finite difference of $\partial_{x_1}^2$ over $f(\mathbf{x})$ being x_2 equals zero, so does $\partial_{x_2}^2$ over $f(\mathbf{x})$ except at the boundary. The abs error of the 2nd-order solver equaling zero is obtained when the boundary row are ignored.

Table 3: nat+per, Cartesian
 $f(\mathbf{x}) = x_2, \mu = 0.05, N_\theta = 100$

mesh	(32,32)	(64,64)
interpo,first	1.0297E-3	1.7507E-2
2nd,first	0.0	0.0
interpo,double	2.3478E-2	8.3417E-2
2nd,double	0.0	0.0

6.3 3rd example: Natural + Periodic boundary condition in the polar coordinate frame

For this case, the periodic boundary condition is used in Θ dimension, while the natural boundary condition is used in r dimension. We also use the test function $f(\mathbf{x}) = x_2$, and the domain is also chosen as $(0, 4) \times (0, 2\pi)$. The integral of the absolute value of the error is presented in Table 4. The boundary rows in r dimension are ignored to get these results. The exact profile of the first and double gyroaverage of the test function at the radial position $7/4$ along the polar dimension and the corresponding numerical solutions computed in the two meshes are plotted in the left figure of Fig.(2). The exact profile of the first and double gyroaverage of the test function profile at the polar angle $7\pi/8$ along the radial dimension and the corresponding numerical solutions computed in the two meshes are plotted in the right figure of Fig.(2). Both figures show that the results computed by the interpolation algorithm fit the exact value very well. But we can still notice a small error at the out boundary of x_1 dimension, where the natural boundary condition is used.

Table 4: nat+per, polar
 $f(\mathbf{x}) = x_2, \mu = 0.02, N_\theta = 100$

mesh	(32,32)	(64,64)
interpo,first	8.0966E-4	1.0698E-5
interpo,double	1.9712E-2	2.9669E-3

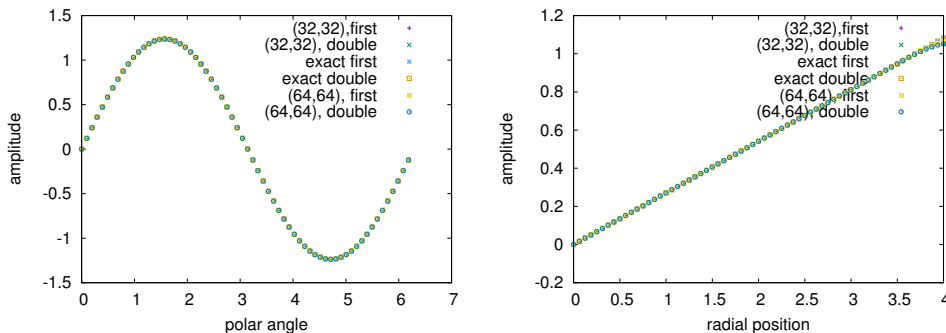


Figure 2: The left figure shows the exact value of the 1st and double gyroaverage of the test function and the value computed in the two meshes of the polar profile at radial position $7/4$. The right figure shows the radial profile at the polar angle $7\pi/8$.

6.4 Benchmark the QNE solver in the Cartesian coordinate frame and in the polar coordinate frame

The previous examples benchmark the interpolation algorithm to compute the gyroaverage operation. This subsection is dedicated to benchmark the QNE solver, which comprises of solving the inverse matrix of A^{spl} and A_{l,ρ_j}^{contri} by FFT based on the algorithm given in Subsec.(5.1), and the interpolation algorithm to obtain the gyroaverage.

For the convenience to find the exact solution of QNE, we simplify QNE in Eq.(22) by choosing $T_i = T_e = 0.5, n_0 = n_{g0} = 1.0$. We also solve the single μ case first, so that QNE is simplified as

$$4\phi(\mathbf{x}) - \mathcal{J}_2(\mu)\mathcal{J}_1(\mu)\phi(\mathbf{x}) = n_1(\mathbf{x}). \quad (45)$$

We first benchmark the QNE solver in Cartesian coordinate frame with double-periodic boundary condition. By choosing

$$n_1(\mathbf{x}) = \cos(nx_1 + mx_2),$$

the exact solution of Eq.(45) is

$$\phi(\mathbf{x}) = \frac{\cos(nx_1 + mx_2)}{4 - J_0^2(\rho(\mu)\sqrt{n^2 + m^2})}.$$

We choose $n = 3, m = 3, \mu = 1.0$ and $\mu = 0$. The simulated domain $(0, 4) \times (0, 2\pi)$ are divided into three meshes including $32 \times 32, 64 \times 64$ and 128×128 cells, respectively. The integrals of the absolute value of the error for the three kinds of mesh are presented in Table 5, which indicates that the QNE solver could provide a good solution. For the $\mu = 0.5$ case, we also plot in Fig.(3) the exact solution and the results computed by the QNE solver with node number in x_2 dimension equaling 20. The two solutions in Fig.(3) fit each other well.

Table 5: double per, Cartesian
 $n = 3, m = 3, N_\theta = 100$

mesh	(32,32)	(64,64)	(128,128)
$\mu = 1.0$	1.5204E-4	6.5066E-6	2.9072E-7
$\mu = 0.0$	1.2393E-13	1.4944E-13	1.3548E-13

To benchmark the QNE solver in the polar coordinate frame, the function n_1 in Eq.(45) is chosen as

$$n_1(\mathbf{x}) = x_2,$$

which leads to the exact solution of $\phi(\mathbf{x})$ through Eq.(45)

$$\phi(\mathbf{x}) = x_2.$$

The simulated domain $(0, 4) \times (0, 2\pi)$ is also divided into meshes of 32×32 and 64×64 cells, respectively. We computed two cases of $\mu = 0.02$ and $\mu = 0$. The integral of the absolute error over the simulated domain for the two μ cases are given in Table 6. The fourth-order converging rate doesn't appear anymore, due to the error at the boundary area where the natural boundary condition is used. The error has two sources. One is that the natural boundary condition is used. The other is that around boundary, due

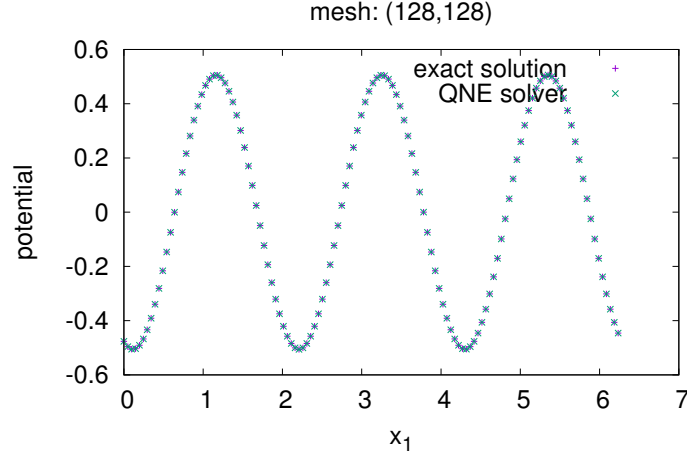


Figure 3: The exact solution and the numerical solution solved by the QNE solver at node 20 in the x_2 dimension.

to the finite value of the Larmor radius, there are interpolation points locating out of the simulated domain and we project these points to the corresponding points at the boundary.

We still plot the exact solution and the numerical solutions of the polar profile at the radial position $7/4$ and the radial profile at the polar angle $7\pi/8$ in Fig.(4), which presents good match between the two solutions.

Table 6: nat+per, polar

$$f(\mathbf{x}) = x_2, N_\theta = 100$$

mesh	(32,32)	(64,64)
$\mu = 0.02$	0.1186	0.1339
$\mu = 0.0$	2.6356E-13	1.4944E-13

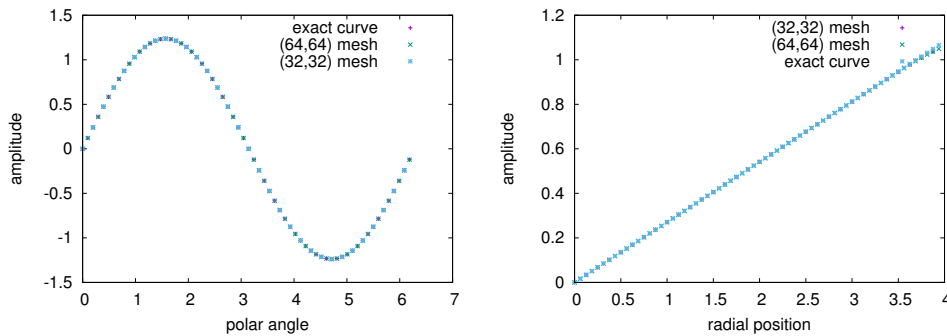


Figure 4: The left figure shows the exact solution and the numerical solutions of the polar profile at the radial position $7/4$. The right figure shows the radial profile at polar angle $7\pi/8$.

7 Benchmark of the interpolation algorithm for the multiple μ case

This benchmark is done in the polar-coordinate frame. The test function is chosen as

$$f(\mathbf{x}) = \cos(nx_1 + mx_2).$$

We consider the gyroaverage associated with μ obeying the distribution $e^{-\mu/a}$ where $a \in (0.5, 2.2)$ is chosen.

The integral of the distribution over μ is written as

$$\mathcal{I}_0(\mu_{\max}, n, m) = \int_0^{\mu_{\max}} e^{-\mu/a} d\mu$$

and the exact solution of the gyroaverage of the test function $f(\mathbf{x})$ are related to the following two functions

$$\mathcal{I}_1(\mu_{\max}, n, m) = \int_0^{\mu_{\max}} J_0(\rho(\mu)\sqrt{n^2 + m^2})e^{-\mu/a} d\mu$$

and

$$\mathcal{I}_2(\mu_{\max}, n, m) = \int_0^{\mu_{\max}} J_0^2(\rho(\mu)\sqrt{n^2 + m^2})e^{-\mu/a} d\mu.$$

The exact solution of the first gyroaverage is

$$\int_0^{\mu_{\max}} \mathcal{J}_1(\mu) f(\mathbf{x}) e^{-\mu/a} d\mu = \mathcal{I}_1(\mu_{\max}, n, m) f(\mathbf{x}),$$

and the exact solution of the double gyroaverage is

$$\int_0^{\mu_{\max}} \mathcal{J}_2(\mu) \mathcal{J}_1(\mu) f(\mathbf{x}) e^{-\mu/a} d\mu = \mathcal{I}_2(\mu_{\max}, n, m) f(\mathbf{x}),$$

which can be written into a discretized formula

$$\sum_{i=1}^{N_\mu} \mathcal{J}_2(\mu_i) \mathcal{J}_1(\mu_i) f(\mathbf{x}) e^{-\mu_i/a} \delta\mu_i.$$

7.1 The Euler-Maclaurin-based quadrature integration algorithm

$\tilde{\Phi}$ is given in Eq.(23) and F is proportional to $\exp(-\frac{\mu B}{T_i(r)})$. To obtain the integral over μ , the conventional way is to redefine a new magnetic moment by $\mu_{new} = \mu/T_i(r)$. Then, the exponential factor $\exp(-\frac{\mu B}{T_i(r)})$ becomes $\exp(-\mu_{new})$ and the Gauss-Laguerre quadrature is implemented to obtain the discretized sum to approximate the continuous integral. However, in the new term $\tilde{\Phi}$, the denominate of $T_i(r)$ in $\exp(-\frac{\mu B}{T_i(r)})$ together with $n_0(r)$ determines the radial density profile of the ions which possess the magnetic moment μ . Since $T_i(r)$ could possess the steep gradient in the radial dimension, its gyroaverage $\mathcal{J}(\mu)T_i(\mathbf{x})$ can't be ignored that it can not be eliminated from the denominate in the conventional way. Moreover, we can not use Gauss-Laguerre quadrature method to provide the same group of μ nodes and the associated weights along the radial dimension for different $T_i(r)$ along the radial dimension. Because the value of $T_i(r)$ is within a domain,

which is chosen as (0.5, 2.2) in this paper, and the gradient of $T_i(r)$ after the gyroaverage could also change, resulting in that the roots and weights provided by Gauss-Laguerre quadrature for the integral $\int_0^{\mu_{\max}} e^{-\mu} d\mu$ can not fit the integral $\int_0^{\mu_{\max}} e^{-\mu/a} d\mu$ for all a belonging to the domain (0.5, 2.2) which is chosen in this paper.

We therefore develop a quadrature integrating scheme based on the following Euler-Maclaurin quadrature formula

$$\int_b^c f(x)dx = \frac{h}{2}(f(b) + f(c)) + \sum_{k=1}^{n-1} hf(b+kh) - \sum_{r=1}^M \frac{h^{2r} B_{2r}}{(2r)!} (f^{2r-1}(c) - f^{2r-1}(b)) + R_p. \quad (46)$$

where $h = (c-b)/n$, $n-1$ is the number of the cells of the mesh, B_{2r} is the $2r$ -th Bernoulli number and R_p is the residual quantity. The order of the precision of the right side is beyond $O(h^{2M})$. For the distribution function proportional to $\exp(-\mu/a)$, R_p and the term depending on $f^{2r-1}(b)$ can be ignored for large enough b . B_{2r} can be solved by the recurrence relation

$$B_p = -\frac{1}{p+1} \sum_{k=0}^{p-1} \binom{p+1}{k} B_k.$$

The derivative $f^{2r-1}(b)$ is replaced by the finite difference scheme. For a function $F(x)$, the finite difference scheme of the d th order derivative with the p th order precision can be formally written as the following summation of the function value on the grid points

$$\frac{h^d}{d!} F^{(d)}(x) = \sum_{i=i_{\min}}^{i_{\max}} G_{dpi} F(x+ih) + O(h^{d+p}), \quad (47)$$

where G_{dpi} is the coefficient of the finite difference with the p th order precision for the d th order derivative, and the step length of the finite difference is the same with that used in Eq.(46). For the forward finite difference scheme of the d th order derivative with p th order precision, $i_{\min} = 0$ and $i_{\max} = d+p-1$ are derived, while for the centered finite difference scheme, we have $-i_{\min} = i_{\max} = (d+p-1)/2$. By truncating the Taylor expanding of $F(x+ih)$ at the order $O((ih)^{d+p-1})$, Eq.(47) can be rewritten as

$$\begin{aligned} F^{(d)}(x) &= \frac{d!}{h^d} \sum_{n=0}^{d+p-1} \left(\sum_{i=i_{\min}}^{i_{\max}} i^n G_{dpi} \right) \frac{h^n}{n!} F^n(x) + O(h^p) \\ &\approx \frac{d!}{h^d} \sum_{n=0}^{d+p-1} \left(\sum_{i=i_{\min}}^{i_{\max}} i^n G_{dpi} \right) \frac{h^n}{n!} F^n(x). \end{aligned} \quad (48)$$

The equaling of both sides of Eq.(48) results in the following identities

$$\sum_{i=i_{\min}}^{i_{\max}} i^n G_{dpi} = \begin{cases} 0, & 0 \leq n \leq d+p-1 \text{ and } n \neq d \\ 1, & n = d \end{cases},$$

through which the coefficient G_{dpi} s can be solved. By substituting Eq.(48) into the Euler-Maclaurin formula, we obtain its discretized version

$$\int_b^c f(x)dx = \frac{h}{2}(f(b) + f(c)) + \sum_{k=1}^{n-1} hf(b+kh) + h \sum_{r=1}^M \frac{B_{2r}}{2r} \sum_{i=i_{\min}}^{i_{\max}} G_{(2r-1)pi} f(b+ih). \quad (49)$$

For the forward finite difference scheme, Eq.(49) can be rearranged into

$$\int_b^c f(x)dx = \sum_{i=1}^{2M+p-2} \left[h + h \sum_{\substack{1 \leq r \leq M, \\ i \leq 2r+p-2}} \frac{B_{2r}}{2r} G_{(2r-1)pi} \right] f(b+ih) \quad (50)$$

$$+ \left[\frac{h}{2} + h \sum_{r=1}^M \frac{B_{2r}}{2r} G_{(2r-1)p0} \right] f(b) + \sum_{i=2M+p-1}^{n-1} hf(b+ih) + \frac{h}{2}f(c),$$

where we select the condition satisfying $2M + p - 1 \leq n - 1$. The weight coefficients of $f(b + ih)$ in Eq.(50) for all $i \in \{0, 1, 2, \dots, n - 1\}$ are computed and stored at the beginning of the simulation for the following quadrature integration.

For the central difference scheme, Eq.(49) can be rearranged in the same way into

$$\int_b^c f(x)dx = \sum_{i=1}^{(2M+p-2)/2} \left[hf(b+ih) + h \sum_{\substack{1 \leq r \leq M, \\ i \leq (2r+p-2)/2}} \frac{B_{2r}}{2r} (G_{(2r-1)pi}f(b+ih) + G_{(2r-1)p(-i)}f(b-ih)) \right] \quad (51)$$

$$+ \left[\frac{h}{2} + h \sum_{r=1}^M \frac{B_{2r}}{2r} G_{(2r-1)p0} \right] f(b) + \sum_{i=(2M+p)/2}^{n-1} hf(b+ih) + \frac{h}{2}f(c),$$

The value of $f(b - ih)$ is obtained by considering the even or odd property of $f(x - b)$ relative to the point $x = b$. The value of b equals 0 for the situation we consider, while the integrand is an even function relative to the point $b = 0$.

One more point to mention is that instead of μ , we implement v_{\perp} , which equals $\sqrt{2\mu}$, as the integration argument for the achievement of the high precision. So $d\mu = v_{\perp}dv_{\perp}$ and the minimal value of v_{\perp} equals 0. Based on this implementation, we developed the third one named derivative-reduction scheme. By replacing μ with v_{\perp} , the integral $\int f(2\mu)d\mu$ becomes $\int f(v_{\perp}^2)v_{\perp}dv_{\perp}$, with the new integrand being $g(v_{\perp}) \equiv f(v_{\perp}^2)v_{\perp}$. At the point $v_{\perp} = 0$, we use the following formula to reduce the order of the derivative

$$g^{2k-1}(0) = (2k - 1)f^{2k-2}(0).$$

Then, the third discretized expression of Euler-Maclaurin formula is derived as

$$\begin{aligned}
\int_0^c g(v_\perp) dv_\perp &\approx \frac{h}{2}(g(0) + g(c)) + \sum_{k=1}^{n-1} hg(kh) + \sum_{r=1}^M \frac{h^{2r} B_{2r}}{(2r)!} g^{2r-1}(0) \\
&= \frac{h}{2}(g(0) + g(c)) + \sum_{k=1}^{n-1} hg(kh) + \sum_{r=1}^M \frac{h^{2r} B_{2r}}{(2r)!} (2r-1) f^{2r-2}(0) \\
&= \frac{h}{2}(g(0) + g(c)) + \sum_{k=1}^{n-1} hg(kh) + \frac{h^2 B_2}{2} f(0) + h^2 \sum_{r=2}^M \frac{B_{2r}}{2r} \sum_{i=i_{\min}}^{i_{\max}} G_{(2r-2)pi} f(0) \\
&= \frac{h}{2} cf(c) + \sum_{k=1}^{n-1} h(kh) f(kh) + \frac{h^2 B_2}{2} f(0) + h^2 \sum_{r=1}^M \frac{B_{2r+2}}{2r+2} \sum_{i=i_{\min}}^{i_{\max}} G_{(2r)pi} f(0) \\
&= \sum_{i=1}^{(2M+p-1)/2} \left[h(ih) f(ih) + h^2 \sum_{\substack{1 \leq r \leq M, \\ i \leq (2r+p-1)/2}} \frac{B_{2r+2}}{2r+2} (G_{(2r)pi} + G_{(2r)p(-i)}) f(ih) \right] \\
&\quad + \frac{h^2 B_2}{2} f(0) + h^2 \sum_{r=1}^M \frac{B_{2r+2}}{2r+2} G_{(2r)p0} f(0) + \sum_{i=(2M+p-1)/2+1}^{n-1} h(ih) f(ih) + \frac{h}{2} cf(c),
\end{aligned} \tag{52}$$

where we have used the even property of the function $f(v_\perp^2)$ at $v_\perp = 0$, so that $f(-ih) = f(ih)$.

To test the accuracy and the convergent rate of the three schemes based on the Euler-Maclaurin formula, we apply them to the computation of the exponential integral $\int_0^{30} e^{-\mu/a} d\mu$ and the associated double gyroaverage integral $\int_0^{30} J_0^2(\sqrt{2\mu}) e^{-\mu/a} d\mu$. And we choose $M = 4, p = 6$ for the test purpose and for the following other numerical simulation examples. We also compare its results to those computed by the trapezoidal quadrature algorithm with uniform mesh. Through dividing the domain $(0, 30)$ into $1E + 7$ equal cells, the two integrals are approximated by the Riemann sum. By treating the Riemann sums as the exact value, the absolute value of the numerical error computed by the two algorithms are plotted in Fig.(5) for the three cases $a = 0.5, 1.0, 2.2$. Fig.(5) shows that compared with the trapezoidal quadrature algorithm, the three schemes of the Euler-Maclaurin-based algorithm provide a much higher precision order when the nodes number goes from 17 to 129.

7.2 Benchmark of DGT computed by the interpolation algorithm in the polar coordinate frame assisted by the forward scheme of the Euler-Maclaurin-based algorithm

This benchmark is done in the polar-coordinate frame and the forward scheme of the Euler-Maclaurin-based algorithm is implemented. In this benchmark, $n = 0, m = 1$ is chosen. The simulated domain of (r, Θ) is $(0, 16) \times (0, 2\pi)$, which is divided into 40×40 meshes. μ is in $(0, 20)$ which is divided into one mesh including 64 nodes. Only the case of $a = 1$ is computed here. The polar profile at $r = 6$ and the radial profile at $\Theta = \frac{7\pi}{8}$ are shown in Fig.(6).

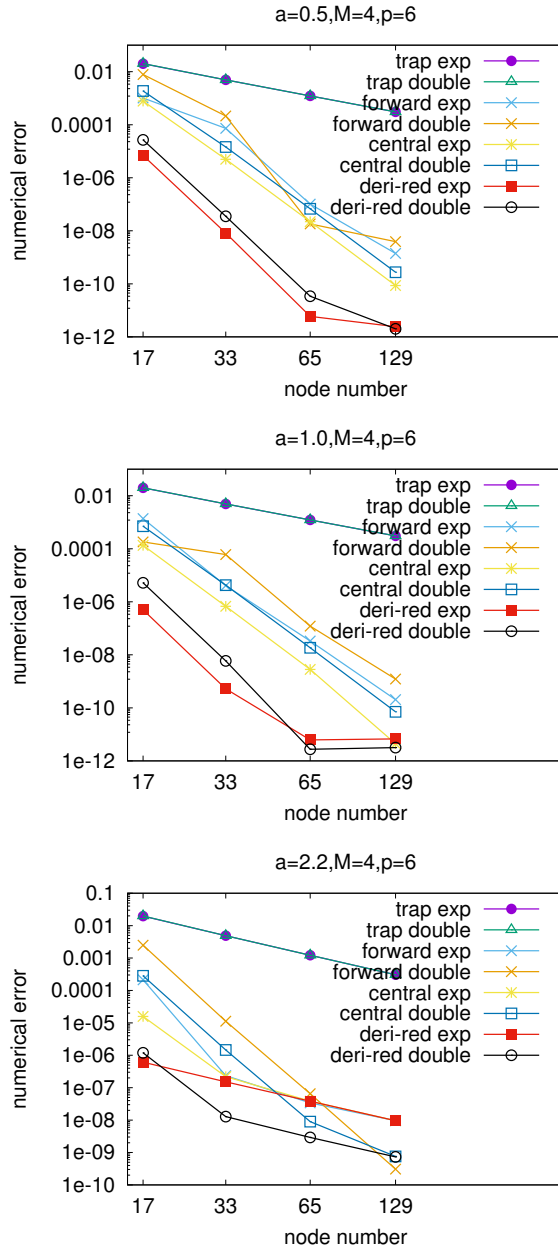


Figure 5: The three figures show the magnitude of the numerical error of the quadrature integral of $\int_0^{30} e^{-\mu/a} d\mu$ and $\int_0^{30} J_0^2(\sqrt{2\mu}) e^{-\mu/a} d\mu$ computed by the three schemes of Euler-Maclaurin-based quadrature integrating algorithm and the trapezoidal algorithm with uniform mesh. We choose $M = 4$ in the Euler-Maclaurin formula and $p = 6$ as the precision order of the finite difference. Here, "deri-red" denotes the derivative-reduction scheme, while "trap" denotes the trapezoidal algorithm.

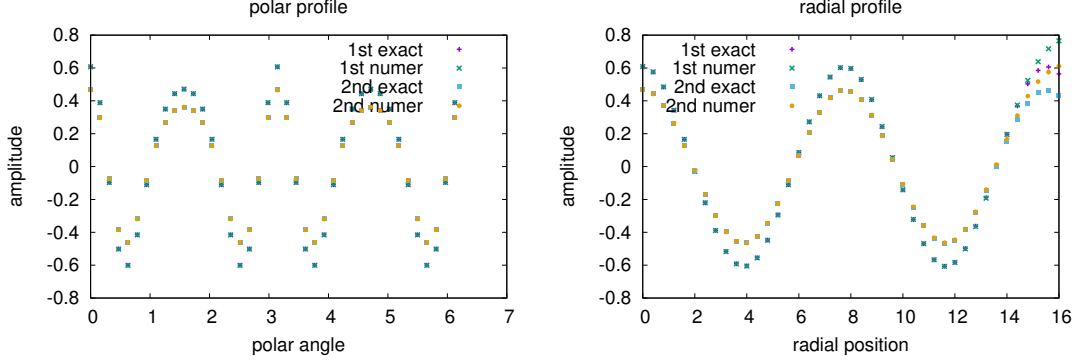


Figure 6: The left figure shows the exact and numerical polar profile at the radial position $r = 6$, while the right figure shows the exact and numerical radial profile at $\Theta = \frac{7\pi}{8}$.

7.3 Benchmark of the QNE solver in the polar-coordinate frame assisted by the forward scheme of the Euler-Maclaurin-based algorithm

We also benchmark the QNE solver in the polar coordinate frame with μ obeying the distribution $e^{-\mu/a}$, with which the quasi-neutrality equation in this case is

$$4\phi(\mathbf{x}) - \int_0^{\mu_{\max}} \mathcal{I}_2(\mu) \mathcal{I}_1(\mu) \phi(\mathbf{x}) e^{-\mu/a} d\mu = n_1(\mathbf{x}), \quad (53)$$

where the same parameters $T_{i0}, T_{e0}, n_0, n_{g0}$ are used. We choose the test function of $n_1(\mathbf{x}) = \cos(nx_1 + mx_2)$. The exact solution of Eq.(53) is

$$\phi(\mathbf{x}) = \frac{\cos(nx_1 + mx_2)}{4 - \mathcal{I}_2(\mu_{\max}, n, m)}.$$

In this benchmark, the radial domain is chosen as $(0, 20)$. The domain of $r \times \Theta$ is divided into a mesh of 64×64 cells. The weights of the μ mesh for $a = 1, 0.5, 2.2$ are computed by the Euler-Maclaurin-based quadrature algorithm. The magnitude of the numerical error of $\mathcal{I}_0, \mathcal{I}_2$ and the abs error of the integration of the solution over the domain for the three a cases are given in Table 7. The computing formula of the abs error is presented by Eq.(44). In Fig.(7), the radial profile at $\Theta = \frac{7\pi}{8}$ and the polar profile at $r = \frac{35}{4}$ for $a = 0.5, a = 1, a = 2.2$ cases are plotted. The absolute value of the numerical error of the polar profile and the radial profile are plotted in the left and right figure of Fig.(8).

Table 7: $N_\mu = 64, n = 1, m = 1, \mu_{\max} = 20, N_\theta = 100$

	$a = 1.0$	$a = 0.5$	$a = 2.2$
$\mathcal{I}_0, \text{num error}$	2.4991E-5	2.4955E-5	2.662E-5
$\mathcal{I}_2, \text{num error}$	2.4908E-5	2.4925E-5	2.4915E-5
solution abs error	1.3817	0.653	3.6886

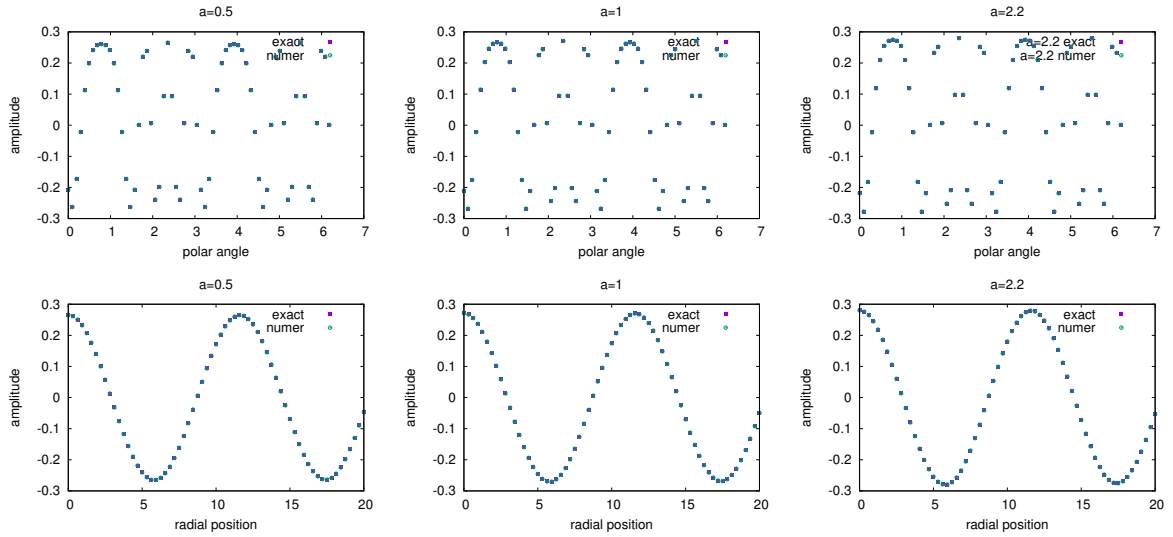


Figure 7: The radial profile at $\Theta = \frac{7\pi}{8}$ and the polar profile at $r = \frac{35}{4}$ for $a = 0.5, a = 1, a = 2.2$ cases are plotted.

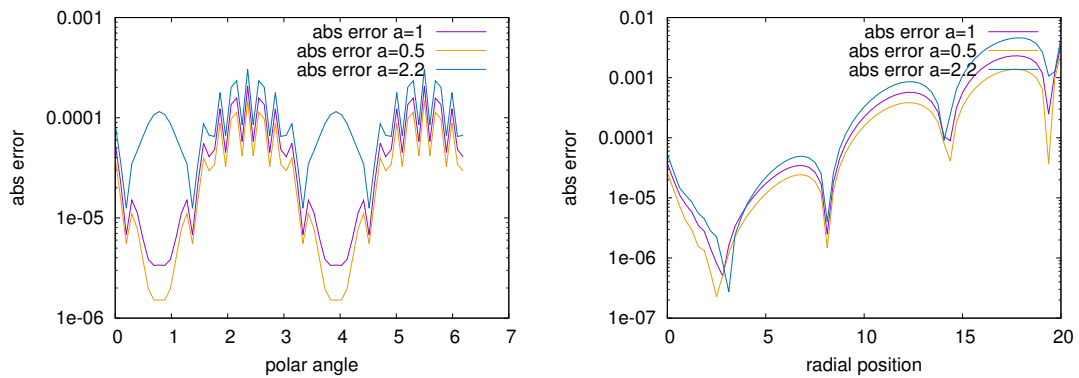


Figure 8: The left figure shows the abs error of the polar profile with $r = \frac{35}{4}$, while the right one presents the abs error of the radial profile with $\Theta = \frac{7\pi}{8}$.

8 Application of the QNE solver in gyrokinetic simulation concerning short-scale perturbation and steep equilibrium profile assisted by the forward scheme of the Euler-Maclaurin-based algorithm

8.1 The gyrokinetic model with constant cylindrical magnetic field configuration

The cylindrical coordinates frame with the constant theta-pinch magnetic field configuration is used in our simulations. The whole simulations are executed on the ATLAS HPC platform of IRMA by introducing our test modules into SeLaLiB code [31], which is a classic semi-Lagrangian library using cubic splines interpolation[8, 32, 15, 16]. The predictor-corrector method, the Verlet algorithm for computing the characteristics, and the Strang splitting of the advection of Vlasov equation are also implemented [23, 15, 33, 34, 7].

The gyrokinetic Vlasov equation in cylindrical coordinate frame is

$$\left[\partial_t + \left(\frac{1}{r}\partial_\Theta\Phi\partial_r - \frac{1}{r}\partial_r\Phi\partial_\Theta\right) + U\partial_{x_\parallel} + \partial_{x_\parallel}\Phi\partial_U\right]F = 0. \quad (54)$$

The characteristics are

$$\dot{r} = \frac{1}{r}\partial_\Theta\Phi, \quad (55a)$$

$$\dot{\Theta} = -\frac{1}{r}\partial_r\Phi, \quad (55b)$$

$$\dot{x}_\parallel = U, \quad (55c)$$

$$\dot{U} = \partial_{x_\parallel}\Phi \quad (55d)$$

The quasi-neutrality equation is given by Eq.(22).

8.2 The parallelization

8.2.1 The algorithm to reduce the distribution to the density

In the discrete version of μ , the distribution of ions associated with each μ_j with $j \in \{1, \dots, N_\mu\}$ is denoted as $F_j(\mathbf{x}, \mu_j, U)$. Due to the identity $\frac{d\mu_j}{dt} = 0$, $F_j(\mathbf{x}, \mu_j, U)$ satisfies the Vlasov equation in the cylindrical coordinate system

$$\left[\partial_t + \left(\frac{1}{r}\partial_\Theta\Phi(\mathbf{x}, \mu_j)\partial_r - \frac{1}{r}\partial_r\Phi(\mathbf{x}, \mu_j)\partial_\Theta\right) + U\partial_{x_\parallel} + \partial_{x_\parallel}\Phi(\mathbf{x}, \mu_j)\partial_U\right]F_j(\mathbf{x}, \mu_j, U) = 0, \quad (56)$$

where (r, Θ, x_\parallel) denotes the cylindrical coordinate frame. $F_j(\mathbf{x}, \mu_j, U)$ can be rewritten as the sum

$$F_j(\mathbf{x}, \mu_j, U) = F_{0j}(\mathbf{x}, \mu_j, U) + F_{1j}(\mathbf{x}, \mu_j, U),$$

with

$$F_{0j}(\mathbf{x}, \mu_j, U) = F_{0j\parallel}(\mathbf{x}, U)F_{0j\perp}(\mathbf{x}, \mu_j) \quad (57)$$

and $F_{0j\perp}(\mathbf{x}, \mu_j) = \frac{1}{2\pi T_i} \exp(-\frac{\mu_j B(\mathbf{x})}{T_i(\mathbf{x})})$. In the numerical simulation, $F_{0j}(\mathbf{x}, \mu_j, U)$ doesn't evolve. At each time step, $F_j(\mathbf{x}, \mu_j, U)$ is obtained by solving Eq.(56) and $F_{1j}(C, \mu_j, U)$ is derived by using $F_j(\mathbf{x}, \mu_j, U)$ minus $F_{0j}(\mathbf{x}, \mu_j, U)$.

The normalised version of Eqs.(17a) and (17b) to obtain n_{sg0} and n_{sg1} can be discretized in the μ dimension by obtaining a mesh of μ with N_μ nodes. The contribution of each μ_j component to the density n_{sg} and n_{sg0} can be written as

$$\begin{aligned} n_{sgj}(\mathbf{x}) &= B(\mathbf{x})\delta\mu_j \int F_s(\mathbf{x} - \boldsymbol{\rho}_0(\mathbf{x}, \mu_j), \mu_j, U) dU, \\ n_{sg0j}(\mathbf{x}, t) &= B(\mathbf{x})\delta\mu_j \int F_{s0}(\mathbf{x} - \boldsymbol{\rho}_0(\mathbf{x}, \mu_j), \mu_j, U) dU. \end{aligned}$$

Then, the n_{sg1j} can be derived as $n_{sgj} - n_{sg0j}$. And the summation of all μ_j s leads to the total density. The specific procedures to compute the density on the particle-coordinate frame are listed below:

1. The distribution $F_j(\mathbf{x}, \mu_j, U, t)$ for each μ_j at time moment t is computed by Eq.(56).
2. The integration of the distribution of each $F_j(\mathbf{x}, \mu_j, U, t)$ over U is first done by the sum $\mathcal{S}_j(\mathbf{x}, t) \equiv \sum_j F_j(\mathbf{x}, \mu_j, U_k, t)\delta U$.
3. The gyroaverage of $\mathcal{S}_j(\mathbf{x}, t)$ denoted as $\mathcal{J}_2(\mu_j)\mathcal{S}_j(\mathbf{x}, t)$ for each μ_j is done by utilising the coefficient matrix with the result being $A_{2,\rho_j}^{contr} A^{spl} \mathcal{S}_{jt}$, where \mathcal{S}_{jt} is a vector formed by $\mathcal{S}_j(\mathbf{x}, t)$ on the grids and is given as

$$\mathcal{S}_{jt} := (\mathcal{S}_{j,0,0}, \dots, \mathcal{S}_{j,N_r,0}, \mathcal{S}_{j,0,1}, \dots, \mathcal{S}_{j,N_r,1}, \dots, \mathcal{S}_{j,0,N_\Theta-1}, \dots, \mathcal{S}_{j,N_r,N_\Theta-1})^T$$

The total density on the particle-coordinate mesh contributed by the μ_j component is written as

$$n_{gj,hp} \equiv (\mathcal{J}_2(\mu_j)\mathcal{S}_j)_{hp}, \quad h \in \{0, \dots, N_r\}, p \in \{0, \dots, N_\Theta - 1\}.$$

4. The equilibrium density in the particle-coordinate frame obtained from the gyroaverage of the equilibrium distribution $F_{0j}(\mathbf{x}, \mu_j, U)$ for the μ_j component is done at the beginning following the procedure 2, 3 and is stored for the subsequent invoking. By inheriting the symbol used in Eq.(29), the j th equilibrium density on particle-coordinate spatial mesh is denoted as $n_{g0j,hp}$. Then, the perturbative density on the particle-coordinate spatial mesh contributed by the μ_j component is

$$n_{g1j,hp} = n_{gj,hp} - n_{g0j,hp}. \quad (58)$$

5. The total perturbative density on the particle-coordinate space computed by all $j \in \{1, \dots, N_\mu\}$ is obtained by "MPI_ALLREDUCE" the $n_{g1j,hp}$ as $\sum_j n_{g1j,hp} B_{hp} \delta\mu_j$, where B_{hp} is the magnetic field magnitude at the node denoted by (h, p) .

For an initial equilibrium density profile on the gyrocenter coordinate frame shown by the purple line of Fig.(1), assuming the equilibrium distribution of μ obeying Eq.(60), the density on the particle coordinate space after the gyroaverage operation is obtained and shown in Fig.(1).

8.2.2 The parallelization

MPI is used to the parallelization of the computing of μ . The processors are divided into N_μ sub-communicators. $F_j(\mathbf{x}, \mu_j, U)$ with $j \in \{1, \dots, N_\mu\}$ is exclusively computed by

the j th sub-communicator. And the respective precomputing matrixes of Φ_j are stored in the j th sub-communicator. $\tilde{\Phi}$ and $n_{i1}(\mathbf{x})$ in Eq.(23) and Eq.(17b) are computed by ‘‘MPI_ALLREDUCE’’ the respective quantity stored in the processors of the same ‘‘color’’ with respect to the respective sub-communicator.

To calculate the advection of distribution function in the 4D domain $(r, \Theta, x_{\parallel}, U)$, two parallelization schemes are involved: the one for parallelizing x_{\parallel} with r, Θ, U sequential is used to calculate the advection due to $\dot{r}, \dot{\Theta}, \dot{U}$; the other one for parallelizing r, Θ, U with x_{\parallel} sequential is used to compute the advection due to U . The 3D domain $(r, \Theta, x_{\parallel})$ with respect to the potential function and the density implements the scheme: the parallelization in x_{\parallel} with r, Θ sequential is used to compute the characteristics $\dot{r}, \dot{\Theta}, \dot{U}$ which are needed in computing the advection of the distribution in r, Θ, U dimensions, and to solve QNE in the poloidal cross section.

8.3 The simulation results

In the cylindrical coordinates system, the initial distribution is of the structure in Eq.(14). The distribution function is specifically given as

$$F(0, r, \Theta, x_{\parallel}) = F_{eq}(r, \mu, U) \times \left(1 + \eta \exp\left(-\frac{(r-r_p)^2}{\delta r}\right) \sum_{n,m} \cos\left(\frac{2\pi n}{L_{\parallel}} x_{\parallel} + m\Theta + \frac{2\pi p}{L_r} r\right) \right), \quad (59)$$

where the equilibrium function F_{eq} is

$$F_{eq}(r, \mu, U) = \frac{n_0(r) \exp\left(-\frac{U^2}{2T_i(r)} - \frac{\mu B}{T_i(r)}\right)}{(2\pi T_i(r))^{3/2}}, \quad (60)$$

and n, m, p are the respective mode numbers. The profile $T_i(r), T_e(r)$ and $n_0(r)$ are given by:

$$\mathcal{P}(r) = C_{\mathcal{P}} \exp\left(-k_{\mathcal{P}} \delta r_{\mathcal{P}} \tanh\left(\frac{r-r_{\mathcal{P}}}{\delta r_{\mathcal{P}}}\right)\right) \quad (61)$$

where $\mathcal{P} \in \{T_i, T_e, n_0\}$, $C_{T_i} = C_{T_e} = 1$ and

$$C_{n_0} = \frac{r_{\max} - r_{\min}}{\int_{r_{\min}}^{r_{\max}} \exp\left(-\kappa_{n_0} \delta r_{n_0} \tanh\left(\frac{r-r_{\mathcal{P}}}{\delta r_{n_0}}\right)\right) dr}. \quad (62)$$

For all the following simulation cases, the time step $\Delta t = 8$ is chosen and the data are stored every two time steps.

8.3.1 1st case: $n = 1, m = 0, p = 0$

This case implements the perturbation not including the polar and radial modes. The utilized equilibrium profiles of ions' density and temperature are not steep in the radial dimension and are given in Fig.(9). We consider the parameters: $\eta = 10^{-4}$, $k_{n_0} = 13.0$, $\kappa_{T_i} = \kappa_{T_e} = 20.0$, $\delta r_{n_0} = 2.0$, $\delta r_{T_i} = \delta r_{T_e} = 1.0$, $L_{\parallel} = 1506.759067$, $r_p = 0.45(r_{\min} + r_{\max})$, $\delta r = \frac{4\delta r_{n_0}}{\delta r_{T_i}}$.

The simulated domain of $r \times \Theta \times x_{\parallel} \times U \times \mu$ is chosen as $(0.5\Delta r, 14.5) \times [0, 2\pi) \times (0, 1506.759067) \times (-7.32, 7.32) \times (0, 20)$, where Δr is the length step in radial direction.

This domain is divided into a mesh of $64 \times 64 \times 32 \times 32 \times 63$ cells. The simulation is carried out by implementing 128 processes, which are executed by 4 nodes on ATLAS HPC of IRMA using hyperthreading. Each node has 24 CPUs and carries on 32 processes.

The evolution curves of the polar modes of the potential at the radial node 30 are shown in Fig.(10). The radial spectrum evolution is shown in Fig.(11). A good match of the results computed by the two solvers is revealed by these figures.

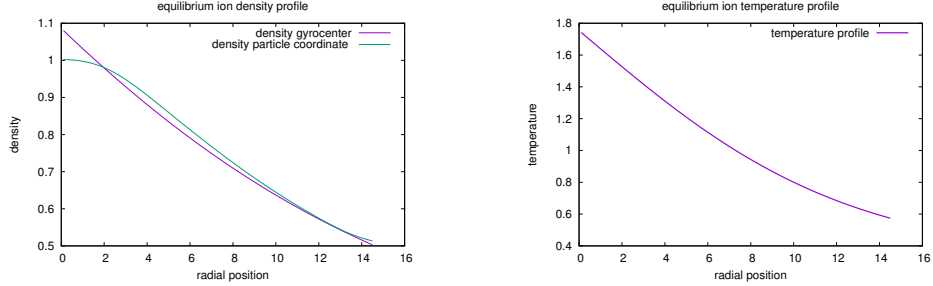


Figure 9: These figures are for the 1st case. The purple curve of the left figure is ion's equilibrium density profile on the gyrocenter coordinate frame, while the green curve is the ion's equilibrium density profile on the particle coordinate frame transformed from the purple curve by the gyroaverage operation. Ion's equilibrium temperature profile is given by the right figure.

8.3.2 2nd case: $n = 1, m = 15, p = 0$

For this case, we could obtain $k_{\Theta}\rho \approx 2$ at $r = 7$ with $\rho = 1$. We consider the parameters: $\eta = 10^{-4}$, $k_{n_0} = 13.0$, $\kappa_{T_i} = \kappa_{T_e} = 66.0$, $\delta r_{T_e} = \delta r_{T_i} = \delta r_{n_0} = 0.2$, $L_{\parallel} = 1506.759067$, $r_p = 0.45(r_{min} + r_{max})$, $\delta r = \frac{4\delta r_{n_0}}{\delta r_{T_i}}$. The equilibrium profile of ion's density and temperature are given by Fig.(12).

The simulated domain is the same with that of the 1st case and is divided into a mesh of $64 \times 128 \times 32 \times 32 \times 63$ cells. For this case, the implementation of the processes and the partition of these processes on the nodes of ATLAS are the same with that of the 1st case.

The potential profile on the polar cross section at time moment 16, 2096, 3680 computed by the two solvers are shown in Fig.(13). The ion temperature modes are driven by the equilibrium gradient. The evolution curves of the polar modes of the potential at the radial node 35 are presented in Fig.(14), which shows that the growth rate computed

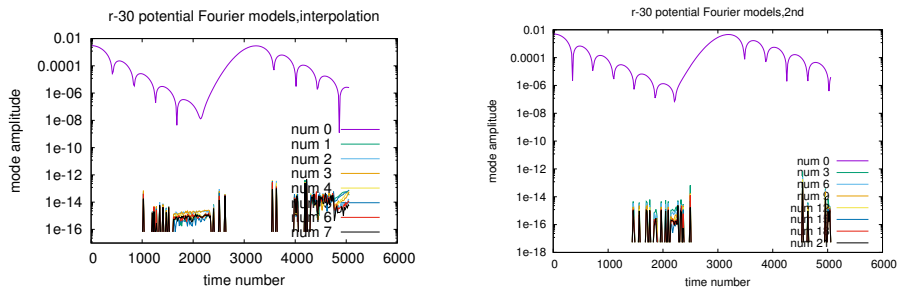


Figure 10: These figures are for the 1st case. The polar Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.

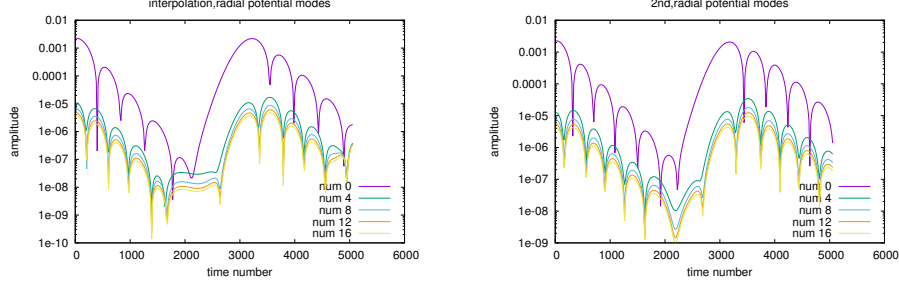


Figure 11: These figures are for the 1st case. The radial Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.

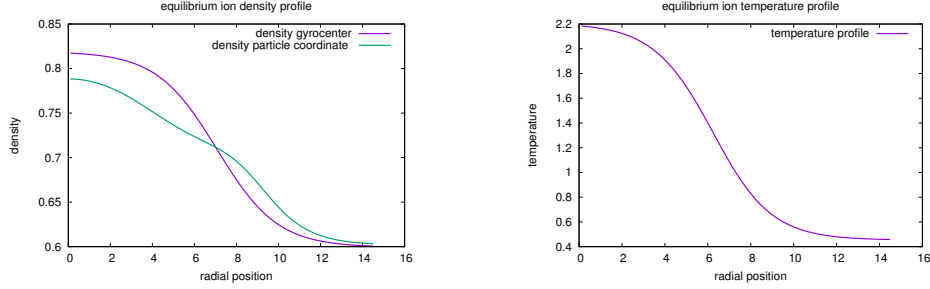


Figure 12: These figures are for the 2nd case. The purple curve of the left figure is ion's equilibrium density profile on the gyrocenter coordinate frame, while the green curve is on the particle coordinate frame transformed from the purple curve by the gyroaverage operation. Ion's equilibrium temperature profile is given by the right figure.

by the interpolation algorithm is larger and the the saturation time is earlier. The radial spectrum evolution is shown in Fig.(15).

8.3.3 3rd case: $n = 1, m = 5, p = 7$

For this case, we could obtain $k_r \rho \approx \pi$ with $\rho = 1$. We consider the parameters: $\eta = 10^{-4}$, $k_{n_0} = 13.0$, $\kappa_{T_i} = \kappa_{T_e} = 66.0$, $\delta r_{T_e} = \delta r_{T_i} = \delta r_{n_0} = 0.1$, $L_{\parallel} = 1506.759067$, $r_p = 0.45(r_{min} + r_{max})$, $\delta r = \frac{4\delta r_{n_0}}{\delta r_{T_i}}$. The equilibrium profile of ion's density and temperature are given by Fig.(16).

The simulated domain is the same and is divided into a mesh of $128 \times 64 \times 32 \times 32 \times 63$ cells. The implementation of the processes on the nodes of ATLAS doesn't change. The potential profile on the polar cross section at time moment 16,3200,5236 computed by the two solvers are shown in Fig.(17). The evolution curves of the polar modes of the potential at the radial node 35 are presented in Fig.(18). The radial spectrum evolution is given in Fig.(22).

8.3.4 4th case: $n = 1, m = 5, p = 0$

The parameters of this case are the same with those used in the 3rd case except that $p = 0$ is chosen. The results are plotted in Figs.(20) and (21). Compared with the simulation results of the 3rd case, it can be observed from both figures that the growth rate of the polar Fourier modes without the initial radial wavelength are much larger than that with the presence of the initial radial wavelength. Such a fact may contribute a new element to the turbulence induced transport theory, which states that the shear flow could reduce

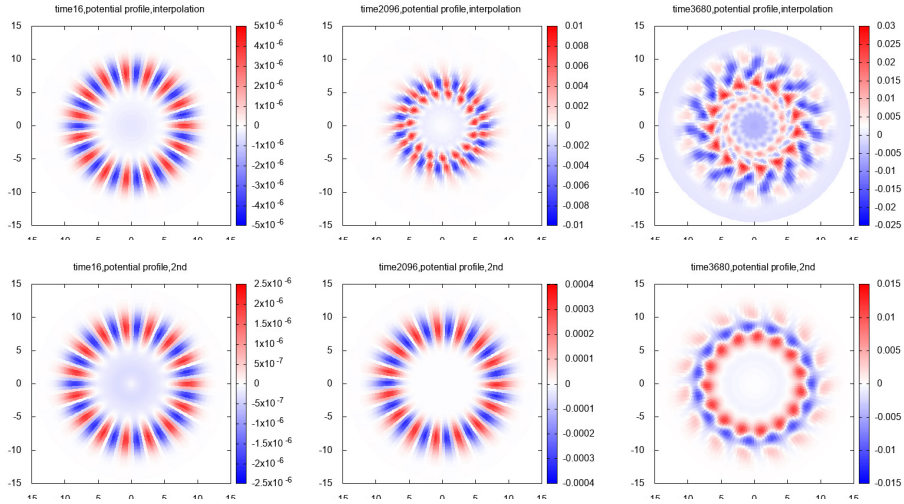


Figure 13: These figures are for the 2nd case. The upper row is the evolution of the potential profile over the polar cross section computed by the interpolation algorithm, while the lower row is computed by the 2nd-order solver.

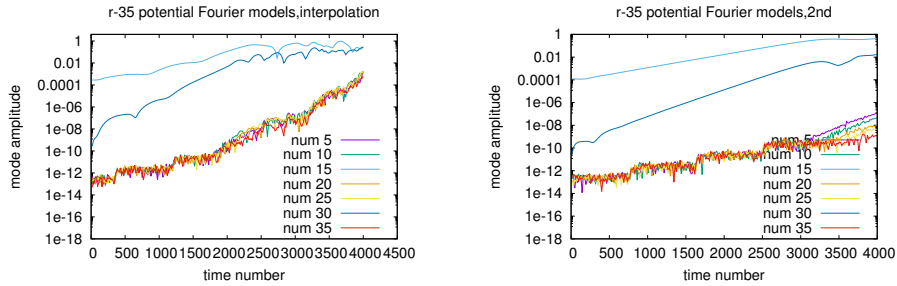


Figure 14: These figures are for the 2nd case. The polar Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.

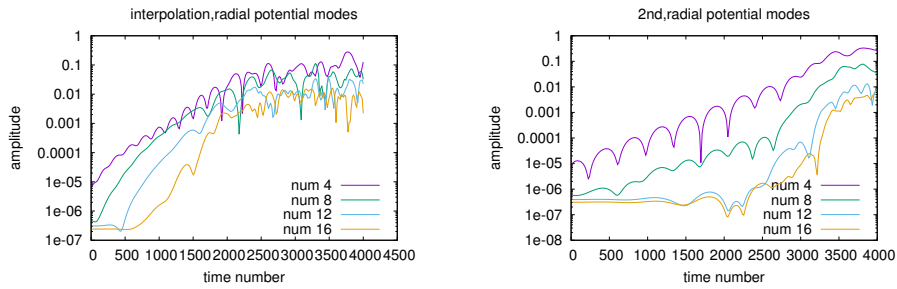


Figure 15: These figures are for the 2nd case. The radial Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.

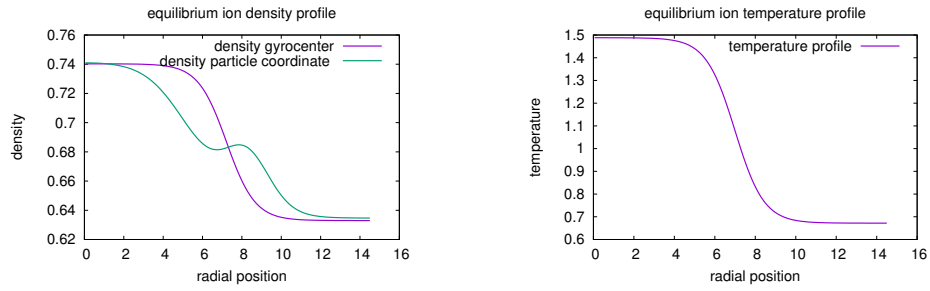


Figure 16: These figures are for the 3rd case. The purple curve of the left figure is ion's equilibrium density profile on the gyrocenter coordinate frame, while the green curve is on the particle coordinate frame transformed from the purple curve by the gyroaverage operation. Ion's equilibrium temperature profile is given by the right figure.

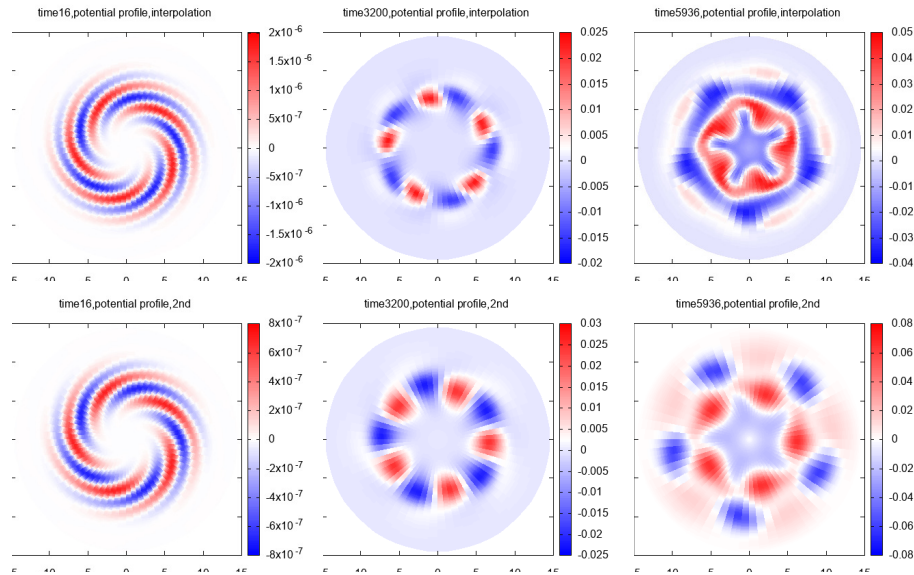


Figure 17: These figures are for the 3rd case. The cross-section profile of the potential computed by the interpolation solver and the 2nd-order solver.

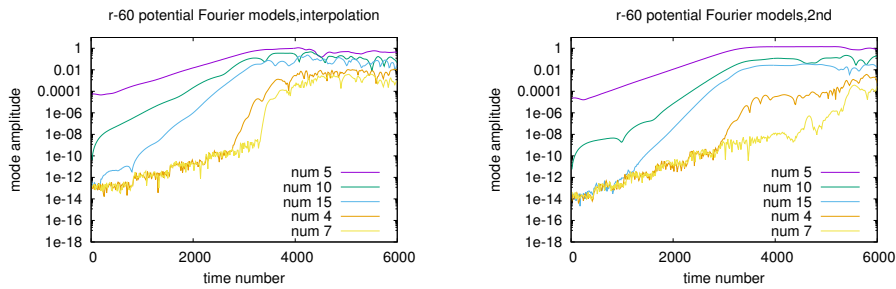


Figure 18: These figures are for the 3rd case. The polar Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.

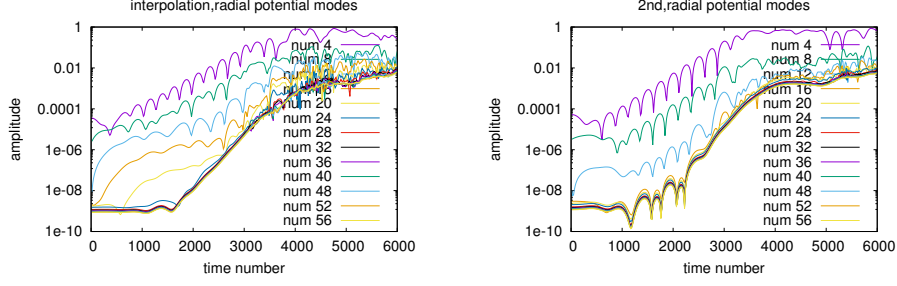


Figure 19: These figures are for the 3rd case. The radial Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.

the radial coherent length of the turbulent vortex, mitigating the transport rate as a consequence. Our simulations in 3rd and 4th cases suggest that with the presence of the low toroidal mode number, the short radial wavelength causes much smaller growth rate of the poloidal modes.

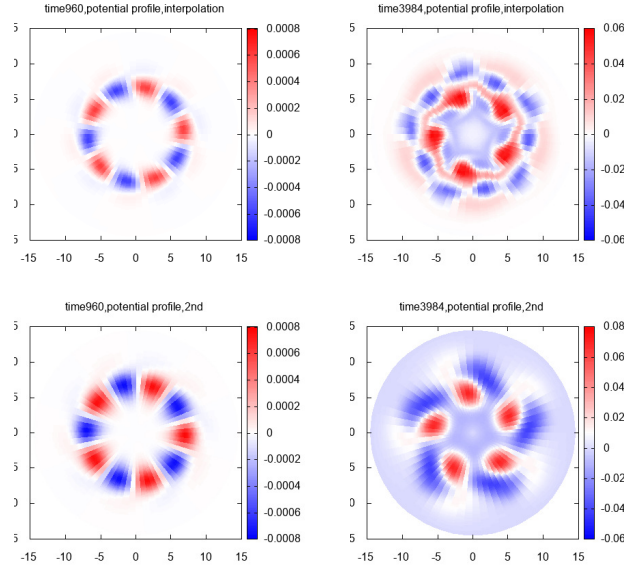


Figure 20: These figures are for the 4th case. The cross-section profile of the potential computed by the interpolation solver and the 2nd-order solver.

9 Conclusion and Discussion

This paper points out that the traditional low-order approximation of DGT is not suitable concerning the short-scale perturbation and the steep equilibrium profile. To obtain the more precise mesh-grids value of the function which is under the first gyroaverage or double gyroaverage in the presence of these extreme cases, the interpolation algorithm is developed by uniformly laying many points on the Larmor circle surrounding the grid point, the function value at which is obtained by utilising the cubic-spline piecewise basis function for the interpolation. Due to the periodic boundary condition of the cubic spline interpolation on the polar dimension, the interpolation coefficient matrix has the periodic block structure. Then, FFT can be implemented to speed up the process of obtaining the

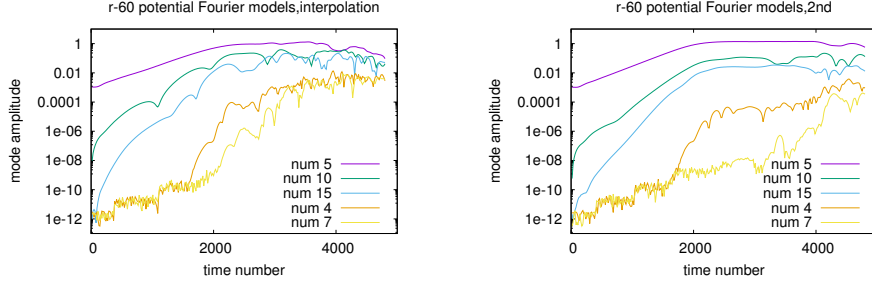


Figure 21: These figures are for the 4th case. The polar Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.

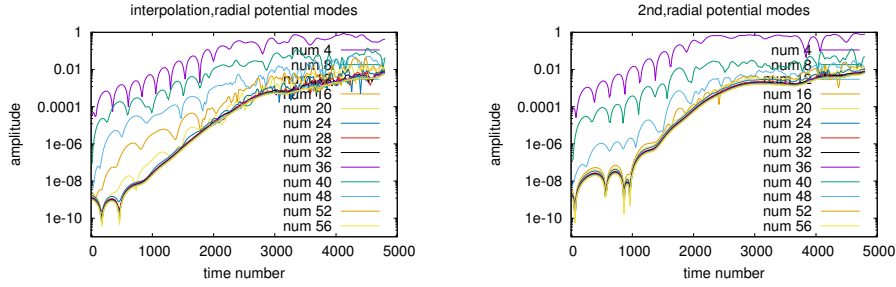


Figure 22: These figures are for the 4th case. The radial Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.

inverse of the discretized coefficient matrix of QNE. Since the density of each μ_j depends on a weight factor $\exp(\frac{-\mu B(\mathbf{x})}{T_e(\mathbf{x})})$ and all quantities as the functions of \mathbf{x} experience a transform $\mathbf{x} \rightarrow \mathbf{x} - \boldsymbol{\rho}(\mathbf{x}, \mu_j, \theta)$, the Gaussian-Laguerre quadrature integrating method can not be used and we developed an Euler-Maclaurin-based quadrature integrating method to obtain the quadrature integration for the distribution of μ .

The interpolation algorithm itself and the interpolation solver for the numerical computation of QNE are benchmarked under various boundary conditions based on various analytical solutions. It's found that numerical solutions computed by both cases match the analytical solutions very well. The interpolation solver is also applied to the gyrokinetic integrated simulation which comprises of the other Vlasov solver and the characteristic solver. The perturbations possessing various model numbers and the equilibrium profiles possessing various radial gradient are implemented in the integrated simulations. To observe the advantage of the interpolation algorithm, its numerical results are compared with those computed by the 2nd-order solver. When the equilibrium profile is not steep and the perturbation only has the nonzero mode number along the parallel spatial dimension, the results computed by the interpolation solver and the 2nd-order solver match each other well. When the gradient of the equilibrium profile is steep, the interpolation solver provides a bigger driving effect for the ion-temperature-gradient modes which possesses large polar mode numbers.

10 Acknowledgments

The first author appreciates the help of Dr. Sever Hirstoaga and Dr. Matthieu Boileau for the implementation of ATLAS HPC of IRMA.

Appendices

A The examples of the exact solution of the gyroaverage and double gyroaverage operation

We first consider a function of the form $f(x + \epsilon g(x))$ depending on one scalar argument and a small parameter. What we are interested in is its expansion over ϵ . The derivative of $f(x + \epsilon g(x))$ over ϵ at $\epsilon = 0$ is derived as follows

$$d_\epsilon f(z)|_{\epsilon=0} = d_\epsilon z \partial_z f(z)|_{\epsilon=0} = g(x) \partial_x f(x), \quad z \equiv x + \epsilon g(x), \quad (63)$$

where $d_\epsilon \equiv d/d\epsilon$. The second order derivative of $f(z)$ over ϵ at $\epsilon = 0$ is

$$\begin{aligned} d_\epsilon(d_\epsilon f(z))|_{\epsilon=0} &= d_\epsilon(d_\epsilon z \partial_z f(z))|_{\epsilon=0} = (d_\epsilon(d_\epsilon z)) \partial_z f(z)|_{\epsilon=0} + d_\epsilon z (d_\epsilon \partial_z f(z))|_{\epsilon=0} \\ &= d_\epsilon g(x)|_{\epsilon=0} + (d_\epsilon z)(d_\epsilon z) \partial_z \partial_z f(z)|_{\epsilon=0} = g^2(x) \partial_x^2 f(x). \end{aligned}$$

The second order derivative of $f(z)$ over ϵ at $\epsilon = 0$ is

$$d_\epsilon^2 f(x + \epsilon g(x))|_{\epsilon=0} = g^2(x) \partial_x^2 f(x).$$

It's easy to derive that the n -th derivative of $f(x + \epsilon g(x))$ over ϵ at $\epsilon = 0$ is

$$d_\epsilon^n f(x + \epsilon g(x))|_{\epsilon=0} = g^n(x) \partial_x^n f(x). \quad (64)$$

Then, the Taylor expansion of $f(x + \epsilon g(x))$ over ϵ is

$$f(x + \epsilon g(x)) = \sum_{n \geq 0} \frac{\epsilon^n}{n!} g^n(x) \partial_x^n f(x). \quad (65)$$

Now we change $g(x)$ to be the vector $\boldsymbol{\rho}$. In Cartesian coordinate frame, $\boldsymbol{\rho}(\mathbf{x}) \cdot \nabla$ can be written as

$$\boldsymbol{\rho}(\mathbf{x}) \cdot \nabla = \sum_i \rho_i(\mathbf{x}) \partial_{x_i},$$

Then, Eq.(65) is changed to be

$$f(\mathbf{x} + \boldsymbol{\rho}(\mathbf{x})) = \sum_{n \geq 0} \frac{1}{n!} \left(\sum_i \rho_i(\mathbf{x}) \partial'_{x_i} \right)^n f(\mathbf{x}) = \exp\left(\sum_i \rho_i(\mathbf{x}) \partial'_{x_i} \right) f(\mathbf{x}), \quad (66)$$

where $'$ means ∂'_{x_i} doesn't operate on any ρ_i s.

In Cartesian coordinate frame, we have

$$\boldsymbol{\rho} \cdot \nabla = \rho \cos \alpha_1 \partial'_{x_1} + \rho \cos(\alpha_1 - \pi/2) \partial'_{x_2},$$

where α_1 is the angle between the Larmor radius vector and \mathbf{e}_1 a unit vector. For

$$f(\mathbf{x}) = \exp(inx_1 + imx_2),$$

we have

$$\begin{aligned}
\mathcal{J}_1(\mu)f(\mathbf{x}) &= \langle \exp(in\rho \cos \alpha_1 + im\rho \cos(\alpha_1 - \pi/2)) \rangle \exp(inx_1 + imx_2) \\
&= \left\langle \exp(i\sqrt{n^2 + m^2} \cos(\alpha_1 + \beta + \pi/2)) \right\rangle \exp(inx_1 + imx_2) \\
&= \left\langle \sum_{q=0} i^q J_q(\rho\sqrt{n^2 + m^2}) \exp(iq(\alpha_1 + \beta + \pi/2)) \right\rangle \exp(inx_1 + imx_2) \\
&= J_0(\rho\sqrt{n^2 + m^2}) \exp(inx_1 + imx_2),
\end{aligned} \tag{67}$$

where $\cos \beta = n/\sqrt{n^2 + m^2}$. With the same reason, we can derive

$$\mathcal{J}_2(\mu)\mathcal{J}_1(\mu)f(\mathbf{x}) = J_0^2(\rho\sqrt{n^2 + m^2}) \exp(inx_1 + imx_2) \tag{68}$$

For $f_1(\mathbf{x}) = x_2$, it's easy to obtain the following results

$$\mathcal{J}_1(\mu)x_2 = x_2, \quad \mathcal{J}_2(\mu)\mathcal{J}_1(\mu)x_2 = x_2$$

References

- [1] H Biglari, P H Diamond, and P W Terry. Influence of sheared poloidal rotation on edge turbulence. *Phys. Fluids B Plasma Phys.*, 2(1):1, 1990.
- [2] A J Brizard. *Nonlinear Gyrokinetic Tokamak Physics*. PhD thesis, 1990.
- [3] K H Burrell, W P West, E J Doyle, M E Austin, T A Casper, P Gohil, C M Greenfield, R J Groebner, A W Hyatt, R J Jayakumar, D H Kaplan, L L Lao, A W Leonard, M A Makowski, G R McKee, T H Osborne, P B Snyder, W M Solomon, D M Thomas, T L Rhodes, E J Strait, M R Wade, G Wang, and L Zeng. Advances in understanding quiescent H-mode plasmas in DIII-D. *Phys. Plasmas*, 12(5):56121, 2005.
- [4] J Candy and R E Waltz. An Eulerian gyrokinetic-Maxwell solver. *J. Comput. Phys.*, 186(2):545–581, 2003.
- [5] C S Chang, S Ku, P Diamond, M Adams, R Barreto, Y Chen, J Cummings, E D’Azevedo, G Dif-Pradalier, S Ethier, L Greengard, T S Hahm, F Hinton, D Keyes, S Klasky, Z Lin, J Lofstead, G Park, S Parker, N Podhorszki, K Schwan, A Shoshani, D Silver, M Wolf, P Worley, H Weitzner, E Yoon, and D Zorin. Whole-volume integrated gyrokinetic simulation of plasma turbulence in realistic diverted-tokamak geometry. *J. Phys. Conf. Ser.*, 180(1):12057, 2009.
- [6] Yang Chen and Scott E Parker. A δf particle method for gyrokinetic simulations with kinetic electrons and electromagnetic perturbations. *J. Comput. Phys.*, 189(2):463–475, 2003.
- [7] D Coulette and N Besse. Numerical comparisons of gyrokinetic multi-water-bag models. *J. Comput. Phys.*, 2013.
- [8] Nicolas Crouseilles, Pierre Glanc, Sever A Hirstoaga, Eric Madaule, Michel Mehrenberger, and Jérôme Pétri. A new fully two-dimensional conservative semi-Lagrangian method: applications on polar grids, from diocotron instability to ITG turbulence. *Eur. Phys. J. D*, 68(9):252, sep 2014.

- [9] P H Diamond, S-I Itoh, K Itoh, and T S Hahm. Zonal flows in plasma—A review. *Plasma Phys. Control. Fusion*, 47(5):R35, 2005.
- [10] D Dickinson, S Saarelma, R Scannell, A Kirk, C M Roach, and H R Wilson. Towards the construction of a model to describe the inter-ELM evolution of the pedestal on MAST. *Plasma Phys. Control. Fusion*, 53(11):115010, 2011.
- [11] Daniel H E Dubin, John A Krommes, C Oberman, and W W Lee. Nonlinear gyrokinetic equations. *Phys. Fluids*, 26(12):3524–3535, 1983.
- [12] Francis F.Chen. *Introduction to plasma physics and controlled fusion*. Springer, 2010.
- [13] X Garbet, Y Idomura, L Villard, and T H Watanabe. Gyrokinetic simulations of turbulent transport. *Nucl. Fusion*, 50(4):43002, 2010.
- [14] T Gorler, X Lapillonne, S Brunner, T Dannert, F Jenko, F Merz, and D Told. The global version of the gyrokinetic turbulence code GENE. *J. Comput. Phys.*, 2011.
- [15] V Grandgirard, M Brunetti, P Bertrand, N Besse, X Garbet, P Ghendrih, G Manfredi, Y Sarazin, O Sauter, E Sonnendrücker, J Vaclavik, and L Villard. A drift-kinetic Semi-Lagrangian 4D code for ion turbulence simulation. *J. Comput. Phys.*, 217(2):395–423, 2006.
- [16] V Grandgirard, Y Sarazin, X Garbet, G Dif-Pradalier, Ph. Ghendrih, N Crouseilles, G Latu, E Sonnendrücker, N Besse, and P Bertrand. Computing ITG turbulence with a full-f semi-Lagrangian code. *Commun. Nonlinear Sci. Numer. Simul.*, 13(1):81–87, 2008.
- [17] Gray M Robert. *Toeplitz and Circulant Matrices: A Review*.
- [18] N Hayashi, T Takizuka, N Aiba, N Oyama, T Ozeki, S Wiesen, and V Parail. Integrated simulation of ELM energy loss and cycle in improved H-mode plasmas. *Nucl. Fusion*, 49(9):95015, 2009.
- [19] R D Hazeltine and J D Meiss. *Plasma Confinement*. Addison-Wesley, Advanced Book Program, 1992.
- [20] A E Hubbard. Physics and scaling of the H-mode pedestal. *Plasma Phys. Control. Fusion*, 42(5A):A15, 2000.
- [21] Yasuhiro Idomura, Masato Ida, Takuma Kano, Nobuyuki Aiba, and Shinji Tokuda. Conservative global gyrokinetic toroidal full-f five-dimensional Vlasov simulation. *Comput. Phys. Commun.*, 179(6):391–403, 2008.
- [22] F Jenko, W Dorland, M Kotschenreuther, and B N Rogers. Electron temperature gradient driven turbulence. *Phys. Plasmas*, 7(5):1904–1910, 2000.
- [23] Guillaume Latu, Michel Mehrenberger, Yaman Güçlü, Maurizio Ottaviani, and Eric Sonnendrücker. Field-Aligned Interpolation for Semi-Lagrangian Gyrokinetic Simulations. *J. Sci. Comput.*, 2017.

- [24] W W Lee. Gyrokinetic approach in particle simulation. *Phys. Fluids*, 26(2):556–562, 1983.
- [25] W W Lee. Gyrokinetic particle simulation model. *J. Comput. Phys.*, 72:243, 1987.
- [26] Z Lin, T S Hahm, W W Lee, W M Tang, and R B White. Turbulent Transport Reduction by Zonal Flows: Massively Parallel Simulations. *Science (80-.)*, 281(5384):1835–1837, 1998.
- [27] Z Lin, W M Tang, and W W Lee. Gyrokinetic particle simulation of neoclassical transport. *Phys. Plasmas*, 2(8):2975–2988, 1995.
- [28] J-S Lönnroth, V Parail, A Dnestrovskij, C Figarella, X Garbet, H Wilson, and JET-EFDA Contributors. Predictive transport modelling of type I ELMy H-mode dynamics using a theory-motivated combined ballooning–speeling model. *Plasma Phys. Control. Fusion*, 46(8):1197, 2004.
- [29] Scott E Parker, Charlson Kim, and Yang Chen. Large-scale gyrokinetic turbulence simulations: Effects of profile variation. *Phys. Plasmas*, 6(5):1709–1716, 1999.
- [30] F Ryter, C Angioni, M Beurskens, S Cirant, G T Hoang, G M D Hogeweij, F Imbeaux, A Jacchia, P Mantica, W Suttrop, and G Tardini. Experimental studies of electron transport. *Plasma Phys. Control. Fusion*, 43(12A):A323, 2001.
- [31] SELALIB. <http://selalib.gforge.inria.fr/>.
- [32] Eric Sonnendrücker, Jean Roche, Pierre Bertrand, and Alain Ghizzo. The Semi-Lagrangian Method for the Numerical Resolution of the Vlasov Equation. *J. Comput. Phys.*, 149(2):201–220, 1999.
- [33] Christophe Steiner, Michel Mehrenberger, Nicolas Crouseilles, Virginie Grandgirard, Guillaume Latu, and Fabien Rozar. Gyroaverage operator for a polar mesh. *Eur. Phys. J. D*, 69(1):18, jan 2015.
- [34] Christophe Steiner, Michel Mehrenberger, Nicolas Crouseilles, and Helluy Philippe. No Title. <https://hal.archives-ouvertes.fr/hal-01248179/document>, 2015.
- [35] W Suttrop. The physics of large and small edge localized modes. *Plasma Phys. Control. Fusion*, 42(5A):A1, 2000.
- [36] W X Wang, Z Lin, W M Tang, W W Lee, S Ethier, J L V Lewandowski, G Rewoldt, T S Hahm, and J Manickam. Gyro-kinetic simulation of global turbulent transport properties in tokamak experiments. *Phys. Plasmas*, 13(9):92505, 2006.
- [37] J Wesson. *Tokamaks*. Oxford Press, third edition, 2004.
- [38] R B White. *Theory of Tokamak Plasmas*. North Holland, Elsevier, 1st editio edition, 1989.
- [39] H Zohm. Edge localized modes (ELMs). *Plasma Phys. Control. Fusion*, 38(2):105, 1996.