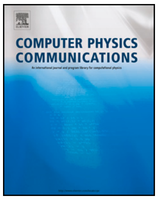
Computer Physics Communications 316 (2025) 109769

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc



Computational Physics

A field-aligned gyrokinetic solver based on discontinuous Galerkin in tokamak geometry

Gahyung Jo[a], Janghoon Seo[a], Jae-Min Kwon[a], Eisung Yoon[b],*

[a] Korea Institute of Fusion Energy, Daejeon, 34133, Republic of Korea
[b] Ulsan National Institute of Science and Technology, Ulsan, 44919, Republic of Korea

ARTICLE INFO

The review of this paper was arranged by Prof. Andrew Hazel

Keywords:
Gyrokinetic equation
Discontinuous Galerkin
Plasma simulation
Fusion plasma
Field-aligned mesh

ABSTRACT

This paper presents the development of a hyperbolic solver for the gyrokinetic equation in tokamak geometry. The discontinuous Galerkin method discretizes the gyrokinetic equation on the field-aligned mesh composed of twisted prism-shaped elements in the tokamak domain. The elements are generated by extending the vertices of unstructured triangular elements on a poloidal plane following the equilibrium magnetic field lines. A sub-triangulation is employed to transfer information between nonconforming meshes, which is inevitable when implementing the field-aligned mesh. The numerical integrations of elements in the field-aligned mesh are performed by transforming the numerical integrations of reference elements in a reference element. We investigate the impact of field-aligned mesh on the numerical interpolation of synthetic plasma fluctuation data generated by a ballooning function. The numerical tests show that the field-aligned mesh can significantly improve computational efficiencies. Additionally, we estimate a sufficient condition for a stable temporal discretization of the hyperbolic solver based on a Runge-Kutta method. The estimation indicates that the field-aligned mesh can allow a notable increase of the time step size for stable simulation. In the numerical experiments, the solver shows good conservations of physical quantities such as mass, kinetic energy, and toroidal canonical angular momentum.

1. Introduction

Micro-instabilities and resulting turbulent transports are important subjects of study because they are critical to understanding and predicting the physical properties of magnetically confined plasma in a fusion device. Due to extremely high temperatures and low collisionality of fusion plasmas, kinetic descriptions are often required.

Compared to the ion gyro-frequency ω_{ci}, plasma fluctuations comprising those instabilities and turbulences have relatively low frequencies $\omega \ll \omega_{ci}$. Regarding the spatial scales of the fluctuations, the wave numbers $k_\perp$ perpendicular to the direction of the confining magnetic field $\boldsymbol{B}_0$ are in a moderate range $k_\perp \rho_i \leq 1$, where ρ_i denotes the ion gyro-radius. The parallel wave numbers $k_\parallel$ satisfy $k_\parallel \ll k_\perp$. The five-dimensional (5D) gyrokinetic model, which is derived to describe such fluctuations efficiently, is used as a standard study tool.

In developing a numerical simulation code based on the gyrokinetic model, it is important to exploit those properties of the fluctuations, as the kinetic nature of the model often requires extremely huge computing resources. One key aspect is the relation $k_\parallel \ll k_\perp$. This implies that fluctuation eddies are highly elongated along ambient equilibrium magnetic fields confining those plasmas.

To exploit this, existing gyrokinetic codes employ numerical algorithms based on discretization along the equilibrium magnetic field. Particle-in-cell (PIC) codes employ field-aligned particle depositions and force interpolations [1–3]. Also, quasi-ballooning forms of finite elements are introduced in field solvers for PIC codes to represent fluctuating eddies [4–6] efficiently. For continuum codes, finite difference methods are implemented by selecting points along the equilibrium magnetic field [7–10]. Also, a general formulation of field-aligned discretization, which is independent of flux coordinate, was given in [11]. Further discussion and comparison of field-aligned and non-aligned approaches can be found in [12]. The hybridizable discontinuous Galerkin (HDG) methods [13] can be considered for complex tokamak geometries for fluid models in the plasma edge where the interaction between the plasma and the wall occurs. Additionally, high-order finite element methods (FEMs) are utilized in [13, 14].

---

* Corresponding author.
E-mail address: esyoon@unist.ac.kr (E.S. Yoon).

https://doi.org/10.1016/j.cpc.2025.109769
Received 14 February 2025; Received in revised form 1 July 2025; Accepted 15 July 2025

However, these previous works are primarily focused on PIC or point collocation methods. We note that there has been relatively little effort to develop related numerical techniques for the Galerkin finite element method (FEM). This is understandable as the Galerkin method requires an explicit partition of the simulation domain. Due to complex and irrational structures of equilibrium magnetic fields in fusion devices, field-aligned partitions will inevitably result in nonconforming elements. Developing a numerical simulation code based on those elements will be very challenging.

Considering the potential to develop an efficient and conservative code, however, developments of field-aligned discretization for FEM or the Galerkin method are highly desired as a necessary step. In this work, we report such a development that utilizes the discontinuous Galerkin method, which can provide advantages for kinetic and gyrokinetic simulations as discussed in [15,16]. Ultimately aiming for a whole device modeling of tokamak plasma in the future, the development in this work attempts 1) a formulation independent of flux coordinate, 2) unstructured mesh to cover the whole domain of tokamak device defined by plasma-facing components, and 3) field-aligned elements and high order basis functions upon them.

As numerical tests to examine the effectiveness of this field-aligned discretization, we perform numerical interpolations of synthetic plasma fluctuation data. The data are generated to have ballooning characters resembling the fluctuation eddies of typical micro-instabilities. By introducing different sets of basis functions on the field-aligned elements, we test the numerical properties of the field-aligned discretization algorithm.

The forthcoming part of the paper is organized as follows. In Section 2, we introduce the numerical method used in this study. In Section 3, we estimate the stability condition for the temporal integration. In Section 4, we present the results of numerical experiments. Finally, in Section 5, we summarize and discuss the implications of the present work for future development.

2. Simulation method and numerical implementation

We consider the gyrokinetic equation [17,18] in a 5D phase space $\Omega = \Omega_x \times \Omega_v$ as follows:

$$\frac{\partial B_\parallel^* f_s}{\partial t} + \frac{\partial \dot{\boldsymbol{X}} B_\parallel^* f_s}{\partial \boldsymbol{X}} + \frac{\partial \dot{v}_\parallel B_\parallel^* f_s}{\partial v_\parallel} = 0, \quad \boldsymbol{X} \in \Omega_x,\ (v_\parallel, u) \in \Omega_v,\ t \in [0, \infty) \tag{1}$$

where f_s denotes the gyrocenter distribution function of species s, $B_\parallel^*$ is the Jacobian of the gyrokinetic coordinates system, $\boldsymbol{X}$ and $v_\parallel$ represent gyrocenter coordinates and parallel velocity, respectively. The variable u is defined as $u := \sqrt{2\mu B_0 / m_s}$, where μ, B_0, and m_s are the magnetic moment, the equilibrium magnetic field at the magnetic axis, and the mass of species s, respectively [19].

The characteristic equations of $\dot{\boldsymbol{X}} (= d\boldsymbol{X}/dt)$ and $\dot{v}_\parallel (= dv_\parallel/dt)$ are given by

$$\frac{d\boldsymbol{X}}{dt} = v_\parallel \frac{\boldsymbol{B}^*}{B_\parallel^*} + \frac{\hat{b}}{B_\parallel^*} \times \frac{c}{q_s} \mu \nabla B \tag{2}$$

$$m_s \frac{dv_\parallel}{dt} = -\frac{\boldsymbol{B}^*}{B_\parallel^*} \cdot \mu \nabla B, \tag{3}$$

where $\boldsymbol{B}^*$ and $B_\parallel^*$ are defined as $\boldsymbol{B}^* = \boldsymbol{B} + (m_s c/q_s) v_\parallel \nabla \times \hat{b}$ and $B_\parallel^* = \hat{b} \cdot \boldsymbol{B}^*$ with equilibrium magnetic field $\boldsymbol{B}$. In this context, $\hat{b}$, c, and q_s represent the unit vectors along the equilibrium magnetic field, the speed of light, and the charge of species s, respectively. For the spatial domain $\Omega_x \subset \mathbb{R}^3$, a cylindrical coordinate system $\boldsymbol{X} = (R, z, \phi)$ is used as depicted in Fig. 1, and the poloidal angle θ is used for the coordinates convention. To prevent an eccentric discontinuity due to the multiplicity in the toroidal direction, we utilize an interval $[-\pi, \pi)$ instead of $[0, 2\pi)$ near the branch cut $\phi = 0$ (or $\phi = 2\pi$).

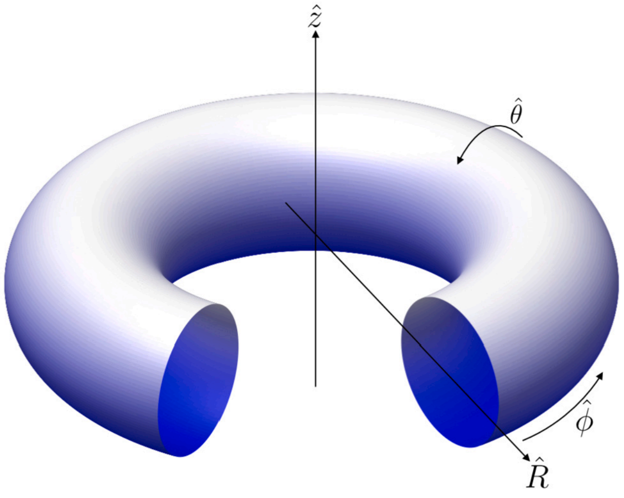


Fig. 1. Cylindrical coordinate system for tokamak geometries.

2.1. Discretization

A tokamak device domain $\Omega_x \subset \mathbb{R}^3$ is divided into a set of non-overlapping subdomains $\{\Omega_i\}_{i=0}^{N_\phi - 1}$. The subdomains are separated by poloidal planes equally spaced in the toroidal direction with step size $\Delta\phi = 2\pi / N_\phi$, where N_ϕ is the number of subdomains. The poloidal planes can be identified as interfaces that are common boundaries between two different subdomains. This partition can be written as

$$\Omega_x = \bigcup_{i=0}^{N_\phi - 1} \Omega_i = \bigcup_{i=0}^{N_\phi - 1} \left\{ \boldsymbol{X} \in \Omega_x : i\Delta\phi \le X_\phi < (i+1)\Delta\phi \right\}, \tag{4}$$

where X_ϕ is the toroidal component of $\boldsymbol{X}$ in the cylindrical coordinates. Fig. 2 shows an example of subdomains of the torus domain with $N_\phi = 4$.

To discretize Ω_x, we generate field-aligned meshes $\mathcal{T}_{h,i}^x$ $(i = 0, 1, \cdots, N_\phi - 1)$ for each subdomain by extending the triangles of unstructured meshes $\mathcal{T}_{h,i}^{\mathrm{pol}}$ along the magnetic field lines, where $\mathcal{T}_{h,i}^{\mathrm{pol}}$ are poloidal meshes that discretize the poloidal planes,

$$\Omega_i^{\mathrm{pol}} = \{ \boldsymbol{X} \in \Omega_i : X_\phi = (i + 1/2)\Delta\phi \}, \tag{5}$$

which are poloidal cross-sections of Ω_i at the center of each subdomain. There is no restriction on the shape of poloidal meshes, therefore they may have different shapes defined by the boundary of the plasma-facing components of the device. However, for simplicity, we consider isomorphic poloidal meshes in this study. The whole mesh for Ω_x can be written by

$$\mathcal{T}_h^x = \bigcup_i \mathcal{T}_{h,i}^x. \tag{6}$$

The field-aligned meshes consist of twisted prism-shaped elements generated by a parametric formula. The parametric elements are determined by the nodal points, which are obtained by tracking along the equilibrium magnetic field lines. We introduce a parametric transform τ_{K_x}, which maps the reference element $\hat{K}_x$ onto the parametric element $K_x \in \mathcal{T}_h^x$. The transform is defined as

$$\tau_{K_x}(\hat{\xi}) = \sum_{i=0}^{m_p - 1} \boldsymbol{a}_i \hat{\zeta}_i(\hat{\xi}), \tag{7}$$

where $\boldsymbol{a}_i$ are nodal points as shown in Fig. 3(a), and $\hat{\zeta}_i$ are the Lagrange polynomials on $\hat{K}_x$. The degree of $\hat{\zeta}_i$ depends on the accuracy required to approximate magnetic field lines in gyrokinetic simulation. This paper uses the parametric elements that need twelve nodal points ($m_p = 12$) as depicted in Fig. 3.

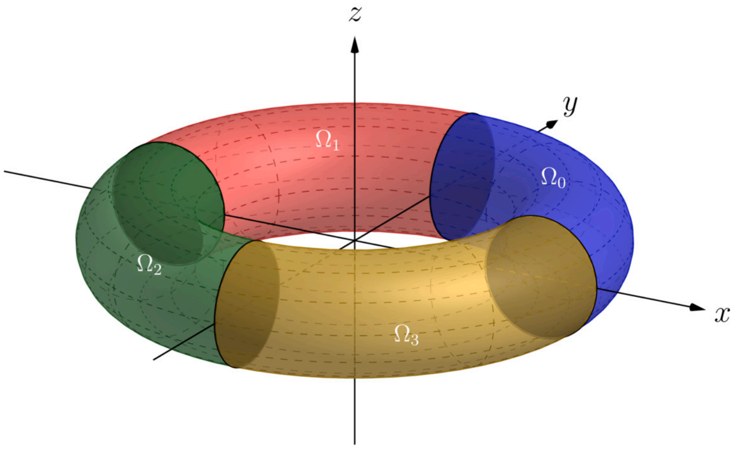
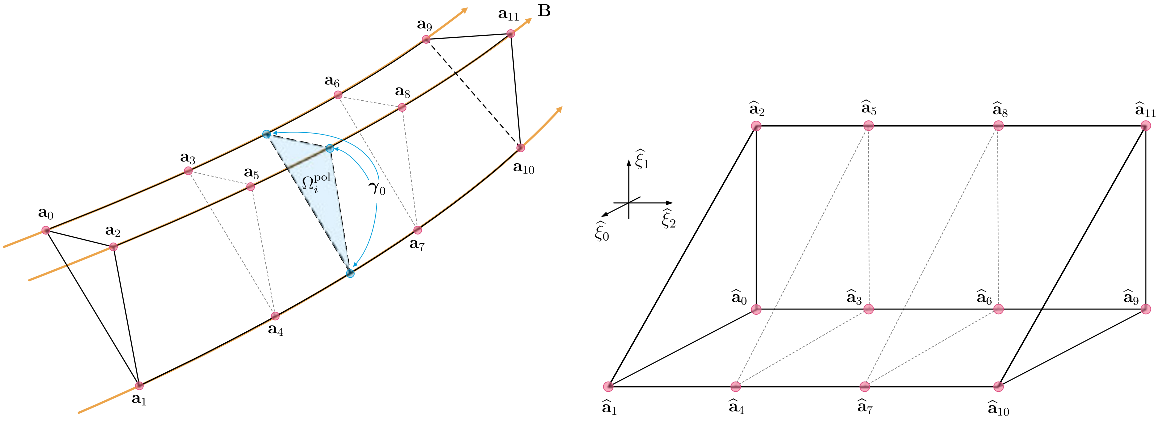


Fig. 2. Subdomains of a torus domain ($N_\phi = 4$).



(a) (b)

Fig. 3. Nodal points of (a) a physical element and (b) the reference element.

In Fig. 3(a), the nodal points can be classified into four groups: $\{\boldsymbol{a}_0,\boldsymbol{a}_1,\boldsymbol{a}_2\}$, $\{\boldsymbol{a}_3,\boldsymbol{a}_4,\boldsymbol{a}_5\}$, $\{\boldsymbol{a}_6,\boldsymbol{a}_7,\boldsymbol{a}_8\}$, and $\{\boldsymbol{a}_9,\boldsymbol{a}_{10},\boldsymbol{a}_{11}\}$. The points of each group belong to the same poloidal planes. The details of how to determine the nodal points will be shown in Sec. 2.2. Fig. 3(b) shows the reference element $\hat{K}_x = \hat{K}_x^{\text{pol}} \times \hat{K}_x^{\phi}$ where $\hat{K}_x^{\text{pol}} = \{(\hat{\xi}_0,\hat{\xi}_1) : 0 \le \hat{\xi}_0, \hat{\xi}_1 \le 1,\ \hat{\xi}_0+\hat{\xi}_1 \le 1\}$, $\hat{K}_x^{\phi} = [-1,1]$. Let $\mathbb{L}_k(\hat{K}_x^{\text{pol}})$ and $\mathbb{L}_k(\hat{K}_x^{\phi})$ be sets spanned by the 2-variate and univariate Lagrange polynomials defined on $\hat{K}_x^{\text{pol}}$ and $\hat{K}_x^{\phi}$ of degrees $\le k$, respectively. The basis function $\hat{\zeta}_i$ is given as follows:

$$\hat{\zeta}_i(\hat{\boldsymbol{\xi}}) = \hat{\zeta}_p^{\text{pol}}(\hat{\xi}_0,\hat{\xi}_1)\hat{\zeta}_q^{\phi}(\hat{\xi}_2), \tag{8}$$

where $i = 3q+p$, $\hat{\zeta}_p^{\text{pol}} \in \mathbb{L}_1(\hat{K}_x^{\text{pol}})$ and $\hat{\zeta}_q^{\phi} \in \mathbb{L}_3(\hat{K}_x^{\phi})$. Note that these basis functions are used only for constructing the parametric elements. The basis for representing the solution will be given in Sec. 2.3. Since the Lagrange polynomials in (7) and (8) satisfy the Kronecker delta property, i.e., $\hat{\zeta}_i(\hat{\boldsymbol{a}}_j) = \delta_{ij}$, τ_{K_x} maps $\hat{\boldsymbol{a}}_j \in \hat{K}_x$ to $\boldsymbol{a}_j \in K_x$ as follows:

$$\tau_{K_x}(\hat{\boldsymbol{a}}_j) = \sum_i \boldsymbol{a}_i \hat{\zeta}_i(\hat{\boldsymbol{a}}_j) = \sum_i \boldsymbol{a}_i \delta_{ij} = \boldsymbol{a}_j. \tag{9}$$

The physical elements (or parametric elements) are defined as the image of the parametric transforms, represented by $K_x = \tau_{K_x}(\hat{K}_x)$. Thus, the transforms define the physical elements rather than the physical elements induce the parametric transforms [20]. It is important to note that $\mathbb{L}_1(\hat{K}_x^{\text{pol}})$ is linear, so τ_{K_x} is an affine mapping for a given $\hat{\xi}_2$. This implies that the parametric transforms do not affect the accuracy and provide simplicity of numerical calculations on the interfaces of subdomain. Generally, there are no guarantees that a parametric transform has an inverse [20]. However, in this study, τ_{K_x} is monotonically increasing (or decreasing) in the toroidal direction because it is assumed that the toroidal component of the magnetic field has a positive lower bound (or a negative upper bound) on our domain; thus we can assume that τ_{K_x} is invertible.

2.2. *Tracking magnetic field*

We assume that the equilibrium magnetic fields have mathematically good properties such as smoothness, non-vanishing, and non-degeneracy. Moreover, the amplitude of toroidal components of magnetic fields has a positive lower bound, i.e., there is a constant $C > 0$ such that $|B_\phi| > C$. Thus, the toroidal angle ϕ can be used as a parameter for the magnetic field line $\boldsymbol{\gamma}(\phi) = \gamma_R(\phi)\hat{e}_R + \gamma_z(\phi)\hat{e}_z + \gamma_\phi(\phi)\hat{e}_\phi$. In order to find the nodal points for a parametric transform in (7), we determine the field line by solving the following initial value problem:

$$\frac{d\boldsymbol{\gamma}(\phi)}{d\phi} = \frac{\gamma_R(\phi)}{\hat{b}_\phi}\hat{b}(\boldsymbol{\gamma}(\phi)), \quad \boldsymbol{\gamma}(\phi_0) = \boldsymbol{\gamma}_0, \tag{10}$$

where $\hat{b}$ is the unit magnetic field, and $\hat{b}_\phi$ denotes the toroidal component of $\hat{b}$. The initial point $\boldsymbol{\gamma}_0$ is chosen as a vertex of the triangle on Ω_i^{pol} defined in (5). The 4th-order Runge-Kutta method gives the nodal points in (7) by solving (10) at $\phi = i\Delta\phi$, $(i+1/3)\Delta\phi$, $(i+2/3)\Delta\phi$, $(i+1)\Delta\phi$, which are equally distributed in the toroidal direction.

The field-following process does not lead to a proper discretization near the boundary as magnetic field lines can intersect the boundary

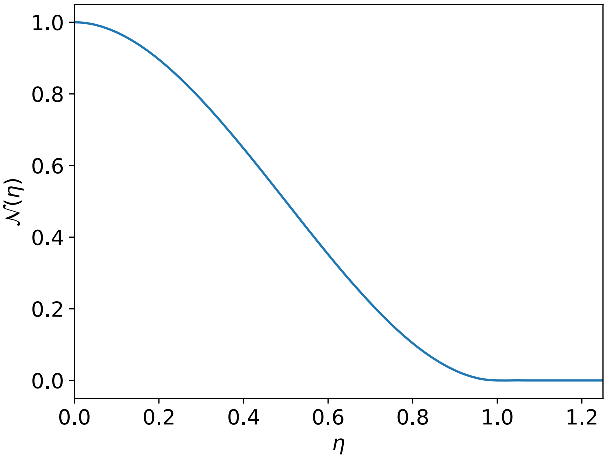


Fig. 4. Blending function.

and leave the simulation domain. In order to prevent the problem, the unit magnetic field $\hat{b}$ in (10) is replaced by a modified unit magnetic field $\tilde{b}$ defined as

$$\tilde{b}(\xi) = \mathcal{N}\left(\frac{d(\xi)}{d_0}\right)\hat{e}_\phi + \left(1 - \mathcal{N}\left(\frac{d(\xi)}{d_0}\right)\right)\hat{b}, \tag{11}$$

where d is the distance from the boundary of Ω_x given by

$$d(\xi) = \inf_{\xi' \in \partial\Omega_x} |\xi - \xi'|, \tag{12}$$

inf indicates the infimum that is the greatest lower bound, $\mathcal{N} : [0,\infty) \to \mathbb{R}$ is a blending function defined by (see Fig. 4)

$$\mathcal{N}(\eta) = \begin{cases} 2\eta^3 - 3\eta^2 + 1 & \text{if } 0 \le \eta < 1, \\ 0 & \text{if } \eta > 1, \end{cases} \tag{13}$$

and a small real number $d_0 > 0$ is empirically selected. In this paper, $d_0 \approx h/2$ where h denotes the poloidal mesh size. Since $0 \le \mathcal{N}(\eta) \le 1$, $\tilde{b}$ can be understood as an interpolation between $\hat{b}$ and $\hat{e}_\phi$. It is worth noting that $\tilde{b}(\xi) = \hat{e}_\phi$ if $d(\xi) \le d_0$ near the boundary.

Figs. 5(a) and 5(b) show the field-aligned meshes following the original and modified unit magnetic fields, respectively. The gray lines depict the original poloidal mesh at $\phi = 0$, while the red and blue lines represent the cross-sections of the field-aligned mesh at $\phi = -\Delta\phi$. In Fig. 5(a), it can be observed that some elements are unacceptable due to intersecting the boundary, while in Fig. 5(b) all elements are admissible.

The cylindrical coordinates system (R, z, ϕ) is widely used in gyrokinetic simulations. However, the cylindrical coordinate system has a complicated vector algebra. For instance, adding two vectors is not straightforward: $(R_1, z_1, \phi_1) + (R_2, z_2, \phi_2) \neq (R_1 + R_2, z_1 + z_2, \phi_1 + \phi_2)$. This complexity can harm the efficiency of calculating the parametric transforms τ_{K_x} in (7), so we combine a coordinates transform with the parametric transform. Our final parametric transform T_{K_x} is defined by

$$\begin{aligned} &T_{K_x} = \sigma \circ \tau_{K_x}, \quad \tau_{K_x} : \hat{\xi} \mapsto \tau_{K_x}(\hat{\xi}) \text{ (in Cartesian)}, \\ &\sigma : \boldsymbol{x} \mapsto \xi \text{ (in cylindrical)}, \end{aligned} \tag{14}$$

where σ transfers a Cartesian representation (x, y, z) to its cylindrical representation (R, z, ϕ) with the following relation:

$$x = R\cos\phi, \quad y = R\sin\phi, \quad z = z. \tag{15}$$

Notice that the Cartesian system (x, y, z) is right-handed ($\hat{e}_x \times \hat{e}_y \cdot \hat{e}_z > 0$), while (R, z, ϕ) represents a left-handed system. ($\hat{e}_R \times \hat{e}_z \cdot \hat{e}_\phi < 0$, see Fig. 1). The Jacobian determinant of T_{K_x} can be written as

$$\left|JT_{K_x}\right| = |J\sigma||J\tau_{K_x}| = -\frac{1}{R}\left|J\tau_{K_x}\right|, \tag{16}$$

where $J\tau_{K_x} \in \mathbb{R}^{3\times3}$ matrix is given by

$$J\tau_{K_x} = \left(\sum_{i=0}^{m-1} \boldsymbol{a}_i \frac{\partial\hat{\zeta}_i}{\partial\hat{\xi}_0} \quad \sum_{i=0}^{m-1} \boldsymbol{a}_i \frac{\partial\hat{\zeta}_i}{\partial\hat{\xi}_1} \quad \sum_{i=0}^{m-1} \boldsymbol{a}_i \frac{\partial\hat{\zeta}_i}{\partial\hat{\xi}_2}\right), \tag{17}$$

and $|J\sigma|$ is given by

$$|J\sigma| = \det\begin{pmatrix} \cos\phi & \sin\phi & 0 \\ 0 & 0 & 1 \\ -\frac{1}{R}\sin\phi & \frac{1}{R}\cos\phi & 0 \end{pmatrix} = -\frac{1}{R}. \tag{18}$$

Here, the negative sign in (18) comes from the inconsistency of the orients of Cartesian and cylindrical coordinates.

The velocity domain Ω_v is discretized by the rectangles such as

$$\Omega_v = \left\{(v_\parallel, u) : v_{\min} \le v_\parallel \le v_{\max},\ 0 \le u \le u_{\max}\right\} = \bigcup_{K_v \in \mathcal{Q}_h^v} K_v, \tag{19}$$

where $\mathcal{Q}_h^v$ denotes the regular mesh on velocity domain, and $K_v = [v_i, v_{i+1}] \times [u_j, u_{j+1}]$, which is the Cartesian product of subintervals in $[v_{\min}, v_{\max}]$ and $[0, u_{\max}]$, respectively.

2.3. FEM space

The basis functions defined on the sets $K = K_x \times K_v \in \mathcal{T}_h$ are utilized to approximate the solution of gyrokinetic equations in (1), where K_x and K_v are the spatial and velocity elements. It is essential to distinguish the solution basis functions and those for the parametric transforms in (9). The solution basis determines the accuracy of solver, while the basis for parametric transform affects the representation of field-following elements.

The construction of the solution basis functions follows a systematic process. It begins with the definition of the basis functions on the reference element $\hat{K}$ in Fig. 6 as the tensor products of Lagrange polynomials [20] on $\hat{K}_x$ and the serendipity basis functions [21,22] on $\hat{K}_v$, and the corresponding function space is given as

$$\hat{V}_h = \left\{\hat{\varphi} = \hat{\varphi}_x^{\text{pol}}\hat{\varphi}_x^{\phi}\hat{\varphi}_v : \hat{\varphi}_x^{\text{pol}} \in \mathbb{L}_k(\hat{K}_x^{\text{pol}}),\ \hat{\varphi}_x^{\phi} \in \mathbb{L}_k(\hat{K}_x^{\phi}),\ \hat{\varphi}_v \in \mathbb{S}_k(\hat{K}_v)\right\}, \tag{20}$$

where $\mathbb{S}_k(\hat{K}_v)$ is the serendipity polynomial space on $\hat{K}_v$. The solution basis functions are then acquired through the pullback operator as follows:

$$\varphi = \left(T_K^{-1}\right)^* \hat{\varphi} = \hat{\varphi} \circ T_K^{-1}, \tag{21}$$

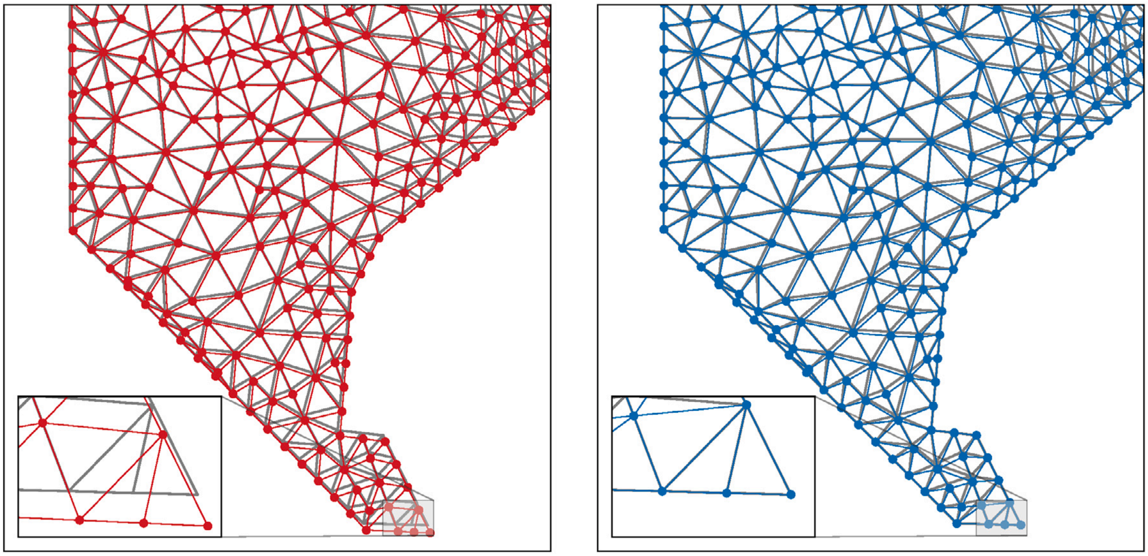
where $T_K : \hat{K} \to K$ is established as a tensor product $T_{K_x} \otimes T_{K_v}$, $T_{K_x} : \hat{K}_x \to K_x$ is the parametric transform introduced in (14), and $T_{K_v} : \hat{K}_v \to K_v$ is affine. The corresponding function space is given as

$$\mathbb{V}_h = \{\varphi = \hat{\varphi} \circ T_K^{-1} : \hat{\varphi} \in \hat{\mathbb{V}}_h \text{ for all } K \in \mathcal{T}_h\}. \tag{22}$$

The basis functions $\varphi \in \mathbb{V}_h$ may not be polynomials, even though the basis functions on the reference element are polynomials. However, this complication may be ignored in practical computations because all computations are performed in the reference element [20]. This paper conducts a thorough investigation using subparametric and isoparametric elements to understand the effect of basis order on accuracy; that is, linear, quadratic, and cubic basis functions are considered for the solution representation. The serendipity finite elements employ lower-order polynomials on quadrilateral meshes with reduced degrees of freedom. This approach preserves accuracy while diminishing the required quadrature order for the integrals [21].

2.4. Discontinuous Galerkin approach for gyrokinetic equations

The Runge-Kutta discontinuous Galerkin (RKDG) method [23,24] is employed to find an approximation solution f_h to (1). The method rep-



(a) (b)

Fig. 5. Field-aligned meshes following (a) the original unit magnetic field $\hat{b}$ and (b) the modified unit magnetic field $\tilde{b}$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

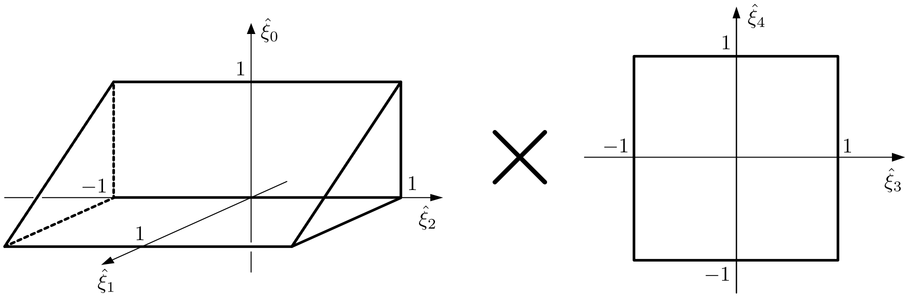


Fig. 6. Reference element consists of a Cartesian product of (left) the triangular prism $\hat{K}_x$ for spatial domain and (right) the square $\hat{K}_v$ for velocity space.

resents f_h as a linear combination of discontinuous polynomial basis functions $\varphi_i \in \mathbb{V}_h$ as follows:

$$f_h(\xi, t) = \sum_i f_i(t)\varphi_i(\xi), \tag{23}$$

where $\xi = (\boldsymbol{X}, v_\parallel, u) = (R, z, \phi, v_\parallel, u) \in \Omega$. For any test functions $\varphi_j \in \mathbb{V}_h$, the Galerkin approach gives the following,

$$\int_K \varphi_j \frac{\partial B_\parallel^* f_h}{\partial t} d\xi + \int_{\partial K} \varphi_j \boldsymbol{U}\cdot\boldsymbol{n}\widehat{f_h} B_\parallel^* d\sigma - \int_K \boldsymbol{U}\cdot\nabla\varphi_j B_\parallel^* f_h d\xi = 0, \tag{24}$$

where $\varphi_j(\xi) = 0$ if $\xi \notin K \in \mathcal{T}_h$, $\boldsymbol{n}$ denotes the outward normal vector on ∂K, and $\boldsymbol{U} = \dot{\xi} = (\dot{\boldsymbol{X}}, \dot{v}_\parallel, 0)$. We say that $\Gamma \subset K$ is a face of $K \in \mathcal{T}_h$, if it is a maximal connected open subset of either $\partial K \cap \partial K'$, for some $K' \in \mathcal{T}_h$ or $\partial K \cap \partial\Omega$, where $\partial\Omega$ is the boundary of domain [25]. Let $\mathcal{F}_h$ be the set of all faces. It is assumed that for each $\Gamma \in \mathcal{F}_h$ the normal vector $\boldsymbol{n}$ on Γ has an unique orientation. This paper follows the convention that $\boldsymbol{n}$ is the outward normal to ∂K^- and the inward normal to ∂K^+, where K^- and K^+ are neighboring elements such that $\Gamma = \partial K^- \cap \partial K^+$. The upwind numerical flux $\widehat{f_h}$ at $\xi \in \Gamma$ is given by

$$\widehat{f_h}(\xi) = \frac{1}{2}\left(f_{h,\Gamma}^+(\xi) + f_{h,\Gamma}^-(\xi)\right) + \frac{1}{2}\mathrm{sgn}(\boldsymbol{U}(\xi)\cdot\boldsymbol{n}(\xi))\left(f_{h,\Gamma}^+(\xi) - f_{h,\Gamma}^-(\xi)\right), \tag{25}$$

where $\mathrm{sgn} : \mathbb{R}\setminus\{0\} \to \mathbb{R}$ is the sign function defined by $\mathrm{sgn}(x) = x/|x|$, and $f_{h,\Gamma}^+(\xi)$ and $f_{h,\Gamma}^-(\xi)$ are defined as

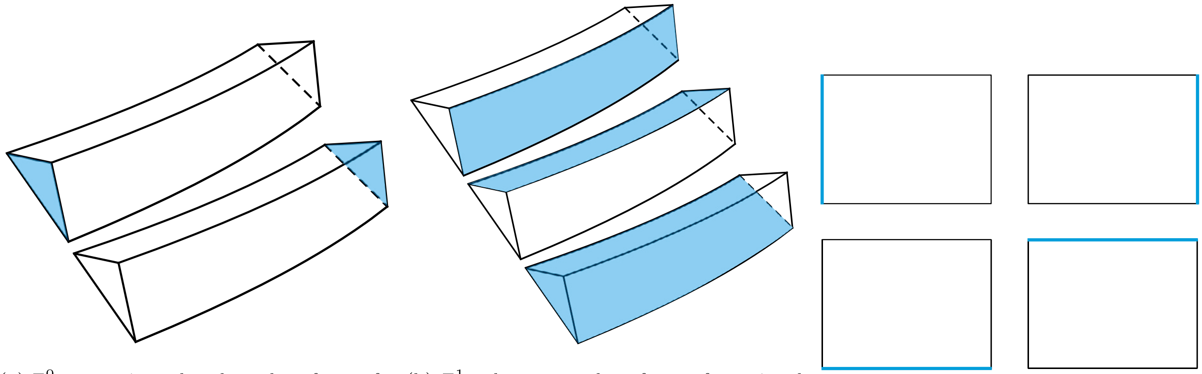
$$f_{h,\Gamma}^\pm(\xi) = \lim_{h\searrow 0} f_h(\xi \pm h\boldsymbol{n}(\xi)) \quad \text{for all } \xi \in \Gamma, \tag{26}$$

respectively. This paper discusses the gyrokinetic equations without time-varying fields, which results in time-invariant flux. The difference in the time-dependency fields case is only recalculating the surface integral (24) in the equations at each Runge-Kutta step.

Using the DG representation of f_h in (23) with the local coefficients $\{f_i^K\}$ according to $K \in \mathcal{T}_h$, we obtain

$$\begin{aligned}\sum_{i=1}^d \frac{\partial f_i^K(t)}{\partial t} \underbrace{\int_K \varphi_j\varphi_i R u B_\parallel^* dR dz d\phi dv_\parallel du}_{M_{ji}^K} &= \sum_{i=1}^d f_i \underbrace{\int_K \boldsymbol{U}\cdot\nabla\varphi_j\varphi_i R u B_\parallel^* dR dz d\phi dv_\parallel du}_{E_{ji}^K} \\ &\quad - \underbrace{\int_{\partial K} \varphi_j \boldsymbol{U}\cdot\boldsymbol{n}\widehat{f_h} R u B_\parallel^* dR dz d\phi dv_\parallel du}_{S_j^K},\end{aligned} \tag{27}$$

for $j = 1, \cdots, d$, where d is the degrees of freedom of basis, $M^K, E^K \in \mathbb{R}^{d\times d}$ are the element-wise local matrices, $S^K \in \mathbb{R}^d$ is a column



(a) Γ_x^0: two triangular-shaped surfaces of a twisted prism K_x.
(b) Γ_x^1: three curved surfaces of a twisted prism K_x.
(c) Γ_v: four edges of a rectangle K_v.

Fig. 7. Examples for the boundaries of elements.

vector assembled over ∂K. Here, the exact phase space volume is $(2\pi/B_0)B_\parallel^* Rud\,Rd\,zd\phi dv_\parallel du$, but the common constants $2\pi/B_0$ on both sides in (27) are omitted for simplicity. Finally, we have the following linear systems,

$$\sum_{i=1}^{d} M_{ji}^K \frac{\partial f_i^K}{\partial t} = -S_j^K + \sum_{i=1}^{d} E_{ji}^K f_i^K, \tag{28}$$

for $j = 1, \cdots, d$. An LDL^T-decomposition solver from the Eigen library [26] is used to invert M^K. The decomposition process occurs only once at the initial step. Each Rugen-Kutta step requires only a forward and backward substitution process. For the temporal discretization, a third-order SSP (Strong Stability Preserving) Runge-Kutta method [23] is used. The details of the procedure have been reported previously in [27].

2.5. *Evaluation of the integrals*

In this section, our focus lies on evaluating the matrices in (27). The integrals are computed numerically using the quadrature rules defined on the reference element shown in Fig. 6 and the change of variables via suitable transformations between the reference and physical elements.

The local integral for M^K in (27) can be evaluated as

$$M_{ji}^K = \int_K \varphi_i \varphi_j B_\parallel^* Rud\,Rd\,zd\phi dv_\parallel du \tag{29}$$

$$= |JT_{K_v}| \int_{\hat{K}} \hat{\varphi}_i(\hat{\xi})\hat{\varphi}_j(\hat{\xi})(B_\parallel^* \circ T_K)(\hat{\xi})|JT_{K_x}(\hat{\xi})|\hat{R}\hat{u}d\hat{\xi}, \tag{30}$$

where $d\hat{\xi} = d\hat{\xi}_0 d\hat{\xi}_1 d\hat{\xi}_2 d\hat{\xi}_3 d\hat{\xi}_4$, JT_{K_x} and JT_{K_v} denote Jacobian matrices of T_{K_x} and T_{K_v}, respectively. Here, $\hat{R} = \pi_R \circ T_K(\hat{\xi})$, $\hat{u} = \pi_u \circ T_K(\hat{\xi})$, where π_R and π_u are the projection operators on R and u directions, respectively. Similarly, the integral for E^K in (27) can be evaluated as

$$E_{ji}^K = \int_K \boldsymbol{U} \cdot \nabla \varphi_i \varphi_j B_\parallel^* Rud\,Rd\,zd\phi dv_\parallel du \tag{31}$$

$$\begin{aligned} &= |JT_{K_v}| \int_{\hat{K}} (\boldsymbol{U} \circ T_K)(\hat{\xi}) \cdot \left[\left(JT_K^{-T}(\hat{\xi})\right) \nabla\hat{\varphi}_i(\hat{\xi})\right] \\ &\quad \times \hat{\varphi}_j(\hat{\xi})(B_\parallel^* \circ T_K)(\hat{\xi})|JT_{K_x}(\hat{\xi})|\hat{R}\hat{u}d\hat{\xi}, \end{aligned} \tag{32}$$

where $JT_K^{-T} = ((JT_K)^T)^{-1}$ and $\boldsymbol{U}$ is the flux given in (2) and (3). Notice that the integrations involve the Jacobian determinant $|JT_{K_x}|$, which is a function of $\hat{\xi} \in \hat{K}$. Therefore, higher-order quadrature rules are necessary compared to affine cases with constant Jacobian determinants.

The surface integral over ∂K, unlike the volume integrals over K, requires additional configuration. The boundaries of elements can be expressed as the union of two sets: $\partial K = (\partial K_x \times K_v) \cup (K_x \times \partial K_v)$. This implies that the surface integrals over ∂K in (27) can be decomposed into a sum of integrals over the faces of three disjoint groups:

(a) $\mathcal{F}_h^0 = \{\Gamma_x^0 \times K_v : \Gamma_x^0$ is a triangular-shaped surface of $K_x \in \mathcal{T}_h^x$, $K_v \in \mathcal{Q}_h^v\}$ (See Fig. 7(a)).
(b) $\mathcal{F}_h^1 = \{\Gamma_x^1 \times K_v : \Gamma_x^1$ is a curved rectangular surface of $K_x \in \mathcal{T}_h^x$, $K_v \in \mathcal{Q}_h^v\}$ (See Fig. 7(b)).
(c) $\mathcal{F}_h^2 = \{K_x \times \Gamma_v : K_x \in \mathcal{T}_h^x$, Γ_v is an edge of $K_v \in \mathcal{Q}_h^v\}$ (See Fig. 7(c)),

where $\mathcal{F}_h = \mathcal{F}_h^0 \cup \mathcal{F}_h^1 \cup \mathcal{F}_h^2$ (disjoint). We will now proceed to discuss the details of surface integrals in the order of $\mathcal{F}_h^2$, $\mathcal{F}_h^1$, and $\mathcal{F}_h^0$.

For the surface integral over $\Gamma^2 = K_x \times \Gamma_v \in \mathcal{F}_h^2$, we regard a diffeomorphism q_v from $I = [-1, 1]$ to $\hat{\Gamma}_v \subset \partial\hat{K}_v$ such that

$$q_v(\lambda) = \frac{1}{2}(\boldsymbol{a} + \boldsymbol{b}) + \frac{\lambda}{2}(\boldsymbol{b} - \boldsymbol{a}), \tag{33}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are two endpoints of an edge of $\hat{K}_v$ depicted in Fig. 6. The transform $Q_{\Gamma^2} : \hat{K}_x \times I \to K_x \times \Gamma_v$ is defined as

$$Q_{\Gamma^2} = T_{K_x} \otimes Q_v, \tag{34}$$

where $Q_v = T_{K_v} \circ q_v$ as follows:

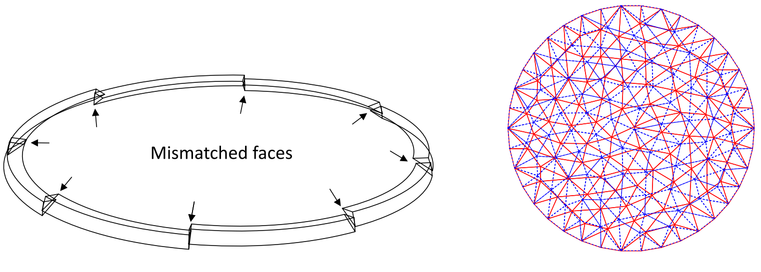
$$Q_v :\ I \xrightarrow{q_v} \hat{\Gamma}_v \xrightarrow{T_{K_v}} \Gamma_v \tag{35}$$

Here, T_{K_v} is the affine transform from $\hat{K}_v$ to $K_v \in \mathcal{Q}_h^v$. Then, using (21), the surface integral can be evaluated as follows:

$$\begin{aligned} S_j^2 &= \int_{K_x \times \Gamma_v} \varphi_j \boldsymbol{U} \cdot \boldsymbol{n}\widehat{f}_h B_\parallel^* Rd\,Rd\,zd\phi d\sigma_v \\ &= |JQ_v| \int_{\hat{K}_x \times I} \left(\hat{\varphi}_j \circ T_K^{-1} \circ Q_{\Gamma^2}\right)(\tilde{\xi}) \left((\boldsymbol{U} \cdot \boldsymbol{n}\widehat{f}_h) \circ Q_{\Gamma^2}\right) \\ &\quad \times (\tilde{\xi})\tilde{R}\tilde{u}|JT_{K_x}(\hat{\xi})|(B_\parallel^* \circ Q_{\Gamma^2})(\tilde{\xi})d\hat{\xi}_0 d\hat{\xi}_1 d\hat{\xi}_2 ds_3 \\ &= |JQ_v| \int_{\hat{K}_x \times I} \left(\hat{\varphi}_j \circ (id \otimes q_v)\right)(\tilde{\xi})(\boldsymbol{U} \cdot \boldsymbol{n}\widehat{f}_h) \\ &\quad \times (\xi)\tilde{R}\tilde{u}|JT_{K_x}(\hat{\xi})|B_\parallel^*(\xi)d\hat{\xi}_0 d\hat{\xi}_1 d\hat{\xi}_2 ds_3, \end{aligned} \tag{36}$$

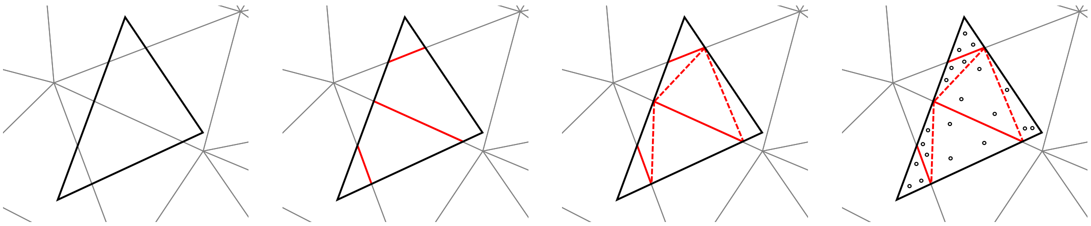
where id is the identity operator, $\tilde{\xi} = (\hat{\xi}_0, \hat{\xi}_1, \hat{\xi}_2, s_3) \in \hat{K}_x \times I$, and $\xi = Q_{\Gamma^2}(\tilde{\xi})$.

Similarly, we consider a diffeomorphism q_x^1 from $I^2 = [-1, 1]^2$ to $\hat{\Gamma}_x^1 \subset \partial\hat{K}_x$ for the surface integral over $\Gamma^1 = \Gamma_x^1 \times K_v \in \mathcal{F}_h^1$. The transform $Q_{\Gamma^1} : I^2 \times \hat{K}_v \to \Gamma_x^1 \times K_v$ is defined as



(a) (b)

Fig. 8. (a) Mismatched faces of twisted-prism shaped elements and (b) the red and blue lines represent the nonconforming meshes at the interface of different subdomains.



(a) Step 1 (b) Step 2 (c) Step 3 (d) Step 4

Fig. 9. Sub-triangulation steps.

$$Q_{\Gamma^1} = Q_x^1 \otimes T_{K_v}, \tag{37}$$

where $Q_x^1 = T_{K_x} \circ q_x^1$ as follows:

$$Q_x^1 : \; I^2 \xrightarrow{q_x^1} \hat{\Gamma}_x^1 \xrightarrow{T_{K_x}} \Gamma_x^1. \tag{38}$$

Let us define $\boldsymbol{\alpha} = \left(\partial Q_x^1/\partial s_1\right) \times \left(\partial Q_x^1/\partial s_2\right)$ (in cylindrical coordinates), where s_1 and s_2 are variables for Q_x^1. The integral over $\Gamma^1 = \Gamma_x^1 \times K_v$ can be evaluated as follows:

$$\begin{aligned} S_j^1 &= \int\limits_{\Gamma_x^1 \times K_v} \varphi_j(\boldsymbol{U} \cdot \boldsymbol{n}\widehat{f}_h)(\boldsymbol{\xi}) B_\parallel^* u d\sigma_x dv_\parallel du \\ &= |JT_v| \int\limits_{I^2 \times \hat{K}_v} (\hat{\varphi}_j \circ T_K^{-1} \circ Q_{\Gamma_1})(\tilde{\boldsymbol{\xi}}) \widehat{f}_h(\boldsymbol{\xi}) B_\parallel^* \circ Q_x^1(\tilde{\boldsymbol{\xi}}) \tilde{u} \boldsymbol{U} \cdot \boldsymbol{\alpha} ds_1 ds_2 d\hat{\xi}_3 d\hat{\xi}_4 \\ &= |JT_v| \int\limits_{I^2 \times \hat{K}_v} (\hat{\varphi}_j \circ (q_x^1 \otimes id))(\tilde{\boldsymbol{\xi}}) \widehat{f}_h(\boldsymbol{\xi}) B_\parallel^*(\boldsymbol{\xi}) \tilde{u} \boldsymbol{U}(\boldsymbol{\xi}) \cdot \boldsymbol{\alpha}(\tilde{\boldsymbol{\xi}}) ds_1 ds_2 d\hat{\xi}_3 d\hat{\xi}_4, \end{aligned} \tag{39}$$

where σ_x represents the infinitesimal surface area of Γ_x^1, $\tilde{\boldsymbol{\xi}} = (s_1, s_2, \hat{\xi}_3, \hat{\xi}_4) \in I^2 \times \hat{K}_v \subset \mathbb{R}^{2+2}$, $\boldsymbol{\xi} = Q_{\Gamma^1}(\tilde{\boldsymbol{\xi}})$, and $\tilde{u} = \pi_u \circ Q_{\Gamma^1}(\tilde{\boldsymbol{\xi}})$.

2.6. Subtriangulation for nonconforming mesh

The field-aligned meshes can improve the performance of code, but it can also lead to nonconforming meshes due to mismatched faces of the parametric elements generated from the different poloidal meshes, as depicted in Fig. 8(a). In the nonconforming meshes, adjacent elements may not share a complete face at the interfaces as shown in Fig. 8(b). For instance, for each nonempty common interface $F = \Omega_i \cap \Omega_j$ $(i \neq j)$, the trace on F of $\varphi_i|_F$ and $\varphi_j|_F$ can not coincide, where φ_i and φ_j are basis functions defined on $\mathcal{T}_{h,i}$ and $\mathcal{T}_{h,j}$, respectively. This mismatch can cause inconsistency in the FEM spaces. Consequently, simple DG formulations may not be able to completely transfer the flux between the nonconforming meshes for the surface integrals over $\Gamma^0 = \Gamma_x^0 \times K_v \in \mathcal{F}_h^0$. To address this issue, this study employs a sub-triangulation method involving the following steps:

(1) Consider a surface integral over the face of a curved element (black thick solid lines in Fig. 9(a)) with the integrand defined on the background mesh (gray lines in Fig. 9(a)).
(2) Slice the face into polygons using the edges of triangles in the background mesh (red lines in Fig. 9(b)).
(3) Divide the polygons of the face into the smaller triangles (red dot lines in Fig. 9(c)).
(4) Generate quadratures on the sub-triangles (empty circles in Fig. 9(d)).

To optimize the sub-triangulation process, we utilize the separating axis method [28] to efficiently determine if two triangles are overlapping. In the second step, the resulting polygons (triangles, rectangles, pentagons, and hexagons) from the overlapping triangles can be sorted into ten categories based on the number of intersection and interior points, as depicted in Fig. 10. Here, the intersection indicates a point where two edges of the front and background triangles cross, and the interior refers to a vertex of a triangle contained within the other triangle. This algorithm uses the background mesh to split the triangles on the front mesh. Since two sub-triangulations are needed for two nonconforming meshes at each subdomain interface, the sub-triangulation process is repeated twice after switching roles between the front and background meshes.

The sub-triangulation for a triangular face of twisted-prim gives a partition $\{\hat{\Gamma}_x^{0,\ell}\}_\ell$ along with the associated parametric transformation such that $\bigcup_\ell \hat{\Gamma}_x^{0,\ell} = \hat{\Gamma}_x^0 \subset \partial \hat{K}_x$, where $\hat{\Gamma}_x^{0,\ell} = T_{K_x}^{-1}(\Gamma_x^{0,\ell})$, as shown in Fig. 11. Let the reference triangle be denoted as D, which is equal to $\hat{K}_x$, but we use different notation for clarity. For the integral over $\Gamma^0 = \Gamma_x^0 \times K_v \in \mathcal{F}_h^0$, we consider the mappings $q_x^{0,\ell} : D \to \hat{\Gamma}_x^{0,\ell}$, which can be easily determined since they are affine transforms between two triangles. The transforms $Q_{\Gamma^{0,\ell}} : D \times \hat{K}_v \to \Gamma_x^{0,\ell} \times K_v$ are defined as

$$Q_{\Gamma^{0,\ell}} = Q_x^{0,\ell} \otimes T_{K_v}, \tag{40}$$

where $Q_x^{0,\ell} = T_{K_x} \circ q_x^{0,\ell}$ as follows:

$$\begin{tikzcd} & & \hat{\Gamma}_x^0 & \Gamma_x^0 \\ & & \cup & \cup \\ Q_x^{0,\ell} : & D \arrow[r, "q_x^{0,\ell}"] & \hat{\Gamma}_x^{0,\ell} \arrow[r, "T_{K_x}"] & \Gamma_x^{0,\ell} \end{tikzcd} \tag{41}$$

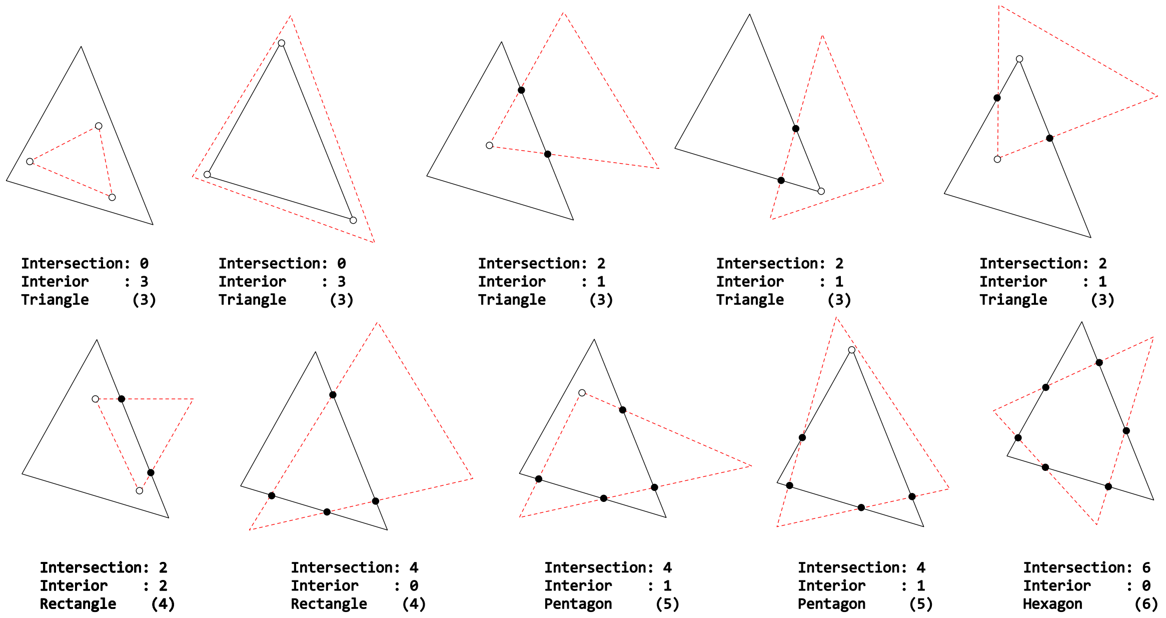


Fig. 10. Categories of polygons generated by overlapping triangles. The empty and filled circles represent intersection and interior points, respectively. The black solid and red dot lines imply the front and background triangles, respectively.

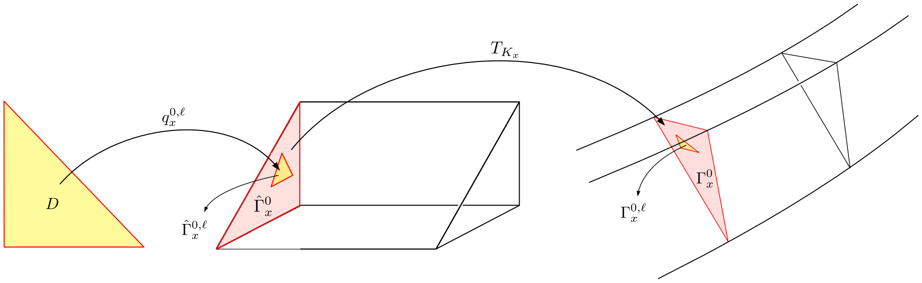


Fig. 11. Diagram of transform $Q_x^{0,\ell}$.

The integral over $\Gamma^0 \in \mathcal{F}_h^0$ can be evaluated as

$$\begin{aligned} S_j^0 &= \int_{\Gamma_x^0 \times K_v} \varphi_j \boldsymbol{U} \cdot \boldsymbol{n} \widehat{f}_h B_\parallel^* u d\sigma_x dv_\parallel du \\ &= \sum_\ell \int_{\Gamma_x^{0,\ell} \times K_v} \varphi_j \boldsymbol{U} \cdot \boldsymbol{n} \widehat{f}_h B_\parallel^* u d\sigma_x dv_\parallel du \\ &= |JT_{K_v}| \sum_\ell |JQ_x^{0,\ell}| \int_{D\times\hat{K}_v} \left(\hat{\varphi}_j \circ T_K^{-1} \circ Q_{\Gamma^{0,\ell}}\right) \\ &\quad \times (\tilde{\boldsymbol{\xi}}) \left((\boldsymbol{U} \cdot \boldsymbol{n} \hat{f}_h) \circ Q_{\Gamma^{0,\ell}}\right) (\tilde{\boldsymbol{\xi}}) \left(B_\parallel^* \circ Q_{\Gamma^{0,\ell}}\right) (\tilde{\boldsymbol{\xi}}) \tilde{R} \tilde{u} d\tilde{\boldsymbol{\xi}} \\ &= |JT_{K_v}| \sum_\ell |JQ_x^{0,\ell}| \int_{D\times\hat{K}_v} \left(\hat{\varphi}_j \circ (q_x^{0,\ell} \otimes id)\right) \\ &\quad \times (\tilde{\boldsymbol{\xi}}) (\boldsymbol{U} \cdot \boldsymbol{n} \hat{f}_h)(\boldsymbol{\xi}) B_\parallel^*(\boldsymbol{\xi}) \tilde{R} \tilde{u} ds_1 ds_2 d\hat{\xi}_3 d\hat{\xi}_4, \end{aligned} \tag{42}$$

where $\tilde{\boldsymbol{\xi}} = (s_1, s_2, \hat{\xi}_3, \hat{\xi}_4) \in D \times \hat{K}_v$, $JQ_x^{0,\ell} = \left(\partial Q_x^{0,\ell} / \partial s_1\right) \times \left(\partial Q_x^{0,\ell} / \partial s_2\right)$. Notice that $Q_x^{0,\ell}$ are affine mappings and their Jacobian determinants are constant. Hence, the lower-order quadrature is required compared to the integrals over $\Gamma^1 \in \mathcal{F}_h^1$ and $\Gamma^2 \in \mathcal{F}_h^2$.

2.7. Parallelization and flowchart

This paper extends the domain decomposition method (DDM) used in the previous study [27,10] to parallelize the new hyperbolic solver using MPI. The entire computational domain is divided into several non-overlapping subdomains, as discussed in Sec. 2.1. An equal number of MPI processes are assigned for computations on each subdomain Ω_i, which is discretized by its poloidal mesh $\mathcal{T}_{h,i}^{\text{pol}}$. The Zoltan Parallel Hypergraph Partitioner [29] is employed to partition the poloidal meshes, while the Parallel Unstructured Mesh Infrastructure [30] provides the ghosting layers for the poloidal partitions. The triangles within the poloidal meshes are extended in the toroidal direction by following the magnetic field lines. This process gives the partitioning for the field-aligned mesh within each subdomain. One MPI process is assigned for computations on each partition that consists of the twist-shaped prisms. Therefore, all elements managed by an MPI CPU belong to the same subdomain. The partitioning provided by the Zotan ensures load balancing, which significantly affects the overall parallelization performance of the hyperbolic solver.

To achieve parallelization in the toroidal direction, the ghost layer cannot be utilized due to the nonconforming mesh structure. Instead, the submesh generated by sub-triangulation enables effective MPI communication across the nonconforming interfaces of subdomains. During the initial phase of simulation, necessary information for the data exchange between MPI processes, i.e., communication pairs between interfaces of subdomains, is collected without relying on the partitioning and ghosting layers. As shown in the previous studies, this paper uses precomputation to accelerate simulation computations, but it necessitates a lot of system memory due to the enormous sub-triangles.

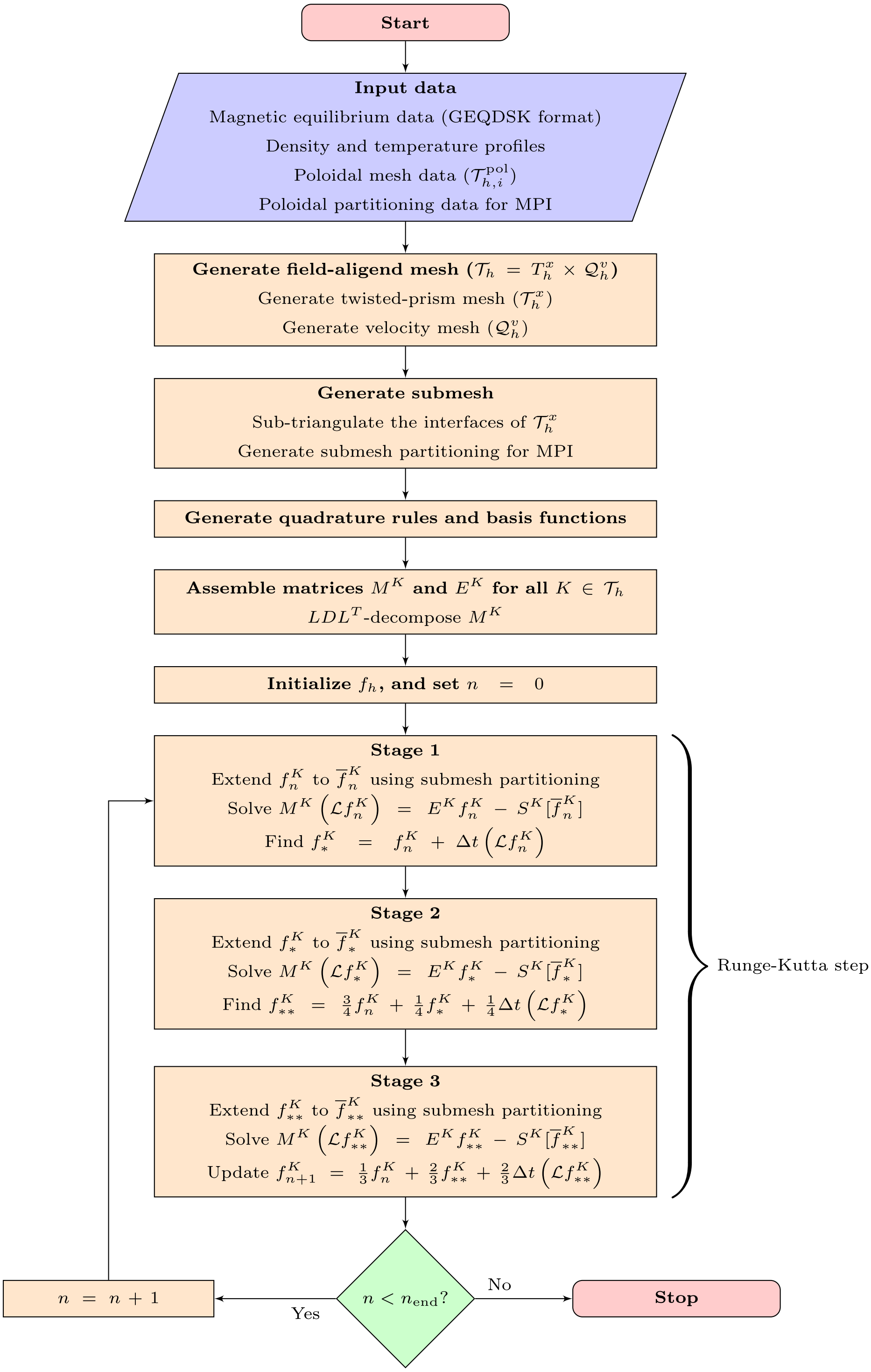


Fig. 12. Flowchart of simulation code, where f_n^K and $\overline{f}_n^K$ represent the local coefficient vector on $K \in \mathcal{T}_h$ at time step $t = t_n$ and its extended vector containing the coefficients on the neighborhood ghosting layers of K, respectively.

As a partial summary, Fig. 12 shows a flowchart of the simulation code. The code is maintained in the repository of the Korea Institute of Fusion Energy and can be provided upon request under an agreement.

3. Stability condition for temporal integration

The stability condition for the Runge-Kutta discontinuous methods is explained in [23]. For the linear hyperbolic equation $\partial_t f + \nabla \cdot (vf) = 0$, the stability condition is expressed as follows:

$$\Delta t \le C \min_j \frac{r_j}{v_j}, \tag{43}$$

where r_j represents the radius of the inscribed circle within the jth cell, v_j denotes the largest amplitude of v in the cell, and C is the CFL number, which varies depending on the order of the Runge-Kutta method. Let $\Delta \boldsymbol{X} \in \mathbb{R}^3$ denote a vector consists of the lengths of an element of $\mathcal{T}_h^x$. To establish a stability condition of the gyrokinetic equation, we consider a decomposition $\Delta \boldsymbol{X} = \Delta \boldsymbol{X}_{\mathrm{pol}} + \Delta \boldsymbol{X}_{\mathrm{tor}}$, where $\Delta \boldsymbol{X}_{\mathrm{pol}} = \Delta \boldsymbol{X} - \Delta \boldsymbol{X}_{\mathrm{tor}}$, and $\Delta \boldsymbol{X}_{\mathrm{tor}} = (\Delta \boldsymbol{X} \cdot \hat{\phi})\hat{\phi}$ for the primitive mesh and

$\Delta \boldsymbol{X}_{\text{tor}} = (\Delta \boldsymbol{X}\cdot\hat{b})\hat{b}$ for the field-aligned mesh. Since $\Delta \boldsymbol{X}_{\text{pol}}$ and $\Delta \boldsymbol{X}_{\text{tor}}$ are independent, we have a sufficient condition for stability such that if the time step Δt satisfies

$$\Delta t \lesssim C \min\left\{\Delta t_{v_\parallel}, \Delta t_{\text{pol}}, \Delta t_{\text{tor}}\right\}, \tag{44}$$

then Δt guarantees the stability condition (43), where $\Delta t_{v_\parallel}$, Δt_{pol}, and Δt_{tor} are constants satisfying $\Delta t_{v_\parallel} \le \min(\Delta v_\parallel/|\dot v_\parallel|)$, $\Delta t_{\text{pol}} \le \min(\Delta X_{\text{pol}}/|\dot{\boldsymbol{X}}_{\text{pol}}|)$, and $\Delta t_{\text{tor}} \le \min(\Delta X_{\text{tor}}/|\dot{\boldsymbol{X}}_{\text{tor}}|)$, respectively. For the sake of simplicity, C in (44) is set by 1, as our primary focus is on the relationships among $\Delta t_\parallel$, t_{pol}, and Δt_{tor}.

We begin by assuming the existence of positive constants $C_{v_\parallel}$, $C_{v_\perp}$, C_r, C_R, and ϵ that satisfying following conditions:

$$|v_\parallel/v_{\text{th,s}}| \le C_{v_\parallel}, \quad |v_\perp/v_{\text{th,s}}| \le C_{v_\perp}, \tag{45}$$

$$\frac{1}{C_r} \le R/R_0 \le C_R, \quad r/R_0 \le \epsilon, \tag{46}$$

where $v_{\text{th,s}}$ refers thermal velocity of species s, and r is the minor radius. Let $n_{v_\parallel}$ and $n_{v_\perp}$ denote the number of intervals corresponding to $v_\parallel$ and $v_\perp$ variables, respectively. This paper approximates B and $\nabla\times\hat{b}$ as follows:

$$B \approx B_0\frac{R_0}{R}, \quad \nabla B \approx -\frac{B_0 R_0}{R^2}\hat{R}, \quad \nabla\times\hat{b} \approx \frac{1}{R}\hat{z}. \tag{47}$$

Also, we assume that the main ion and species s are at the same temperature $T_i \approx T_s$. This assumption leads to the relation $m_i v_{\text{th,i}}^2 \approx m_s v_{\text{th,s}}^2$ resulting the expression:

$$\frac{v_{\text{th,i}}}{v_{\text{th,s}}} \approx \sqrt{\frac{m_s}{m_i}}, \tag{48}$$

where $v_{\text{th,i}}$ and $v_{\text{th,s}}$ are the thermal velocities of main ion and species s, respectively.

3.1. Estimation of $\Delta t_{v_\parallel}$

For the velocity domain, we have

$$\Delta v_\parallel = \frac{\max v_\parallel - \min v_\parallel}{n_{v_\parallel}} = \frac{2C_{v_\parallel} v_{\text{th,s}}}{n_{v_\parallel}}. \tag{49}$$

Assuming $(m_s c/q_s)\nabla\times\hat{b}\cdot\hat{R} \ll \boldsymbol{B}\cdot\hat{R}$ and considering $B_p = |\boldsymbol{B}_p| = \sqrt{(\boldsymbol{B}\cdot\hat{R})^2 + (\boldsymbol{B}\cdot\hat{z})^2} \ge |\boldsymbol{B}\cdot\hat{R}|$, we can see the following relation:

$$\boldsymbol{B}^*\cdot\hat{R} = \left(\boldsymbol{B} + \frac{m_s c}{q_s} v_\parallel \nabla\times\hat{b}\right)\cdot\hat{R} \approx \boldsymbol{B}\cdot\hat{R} \le B_p, \tag{50}$$

where $\boldsymbol{B}_p$ represents the poloidal part of $\boldsymbol{B}$. (3) implies that

$$\begin{aligned}
\dot v_\parallel &= -\frac{\boldsymbol{B}^*}{m_s B_\parallel^*}\cdot\mu_s\nabla B \\
&\approx \frac{v_\perp^2}{2B}\frac{\boldsymbol{B}^*\cdot\hat{R}}{B_\parallel^*}\frac{B_0 R_0}{R^2} && \left(\text{since } \mu_s = \frac{m_s v_\perp^2}{2B} \text{ and } (47)\right) \\
&\lesssim \frac{B_0}{B_\parallel^*}\frac{B_p}{B_t}\frac{v_\perp^2}{2}\frac{R_0}{R^2} && \left(\text{since } B = |\boldsymbol{B}_p + \boldsymbol{B}_t| \ge B_t \text{ and } (50)\right) \\
&\approx \frac{r v_\perp^2}{2qR_0 R} && \left(\text{since } \frac{B_0}{B_\parallel^*} \approx \frac{B_0}{B} \approx \frac{R}{R_0} \text{ and } q \approx \frac{r}{R_0}\frac{B_t}{B_p}\right) \\
&\le \epsilon C_r C_{v_\perp}^2 \frac{v_{\text{th,s}}^2}{2qR_0}, && (\text{since } (46) \text{ and } (45))
\end{aligned} \tag{51}$$

where m_s denotes mass of species s, q_s is charge of species s, and q is the safety factor. Normalization with $R_0/v_{\text{th,i}}$ gives that

$$\begin{aligned}
\frac{\Delta v_\parallel/|\dot v_\parallel|}{R_0/v_{\text{th,i}}} &\gtrsim \frac{v_{\text{th,i}}}{R_0}\frac{2C_{v_\parallel} v_{\text{th,s}} n_{v_\parallel}}{\epsilon C_r C_{v_\perp}^2 v_{\text{th,s}}^2/(2qR_0)} \\
&= \frac{4C_{v_\parallel}}{\epsilon C_r C_{v_\perp}^2}\frac{q}{n_{v_\parallel}}\frac{v_{\text{th,i}}}{v_{\text{th,s}}} \approx \frac{4C_{v_\parallel}}{\epsilon C_r C_{v_\perp}^2}\frac{q}{n_{v_\parallel}}\sqrt{\frac{m_s}{m_i}},
\end{aligned} \tag{52}$$

where m_s denotes the mass of species s, and (48) is used for the last term. Consequently, it can be concluded that $\Delta t_{v_\parallel}$ may be chosen such that

$$\frac{\Delta t_{v_\parallel}}{R_0/v_{\text{th},i}} \lesssim \frac{4C_{v_\parallel}}{\epsilon C_r C_{v_\perp}^2}\frac{q}{n_{v_\parallel}}\sqrt{\frac{m_s}{m_i}}. \tag{53}$$

3.2. Estimation of Δt_{pol}

Let us consider the poloidal meshes that are unstructured and quasi-uniform [25]. We assume that there is a positive constant C_K such that $|K| \le C_K\rho_i$ for all K in the poloidal meshes, where $|K|$ represents the diameter of K given as $|K| = \sup\{|\boldsymbol{x}' - \boldsymbol{y}'| : \boldsymbol{x}', \boldsymbol{y}' \in K\}$, and sup indicates the supremum that is the least upper bound. Thus, we have

$$\Delta X_{\text{pol}} = \min_{K\in\mathcal{T}_{h,x}^{\text{pol}}} |K| = C_K\rho_i \approx C_K\frac{m_i c v_{\text{th,i}}}{q_i B}. \tag{54}$$

The formulation for $\dot{\boldsymbol{X}}$ in (2) gives us that

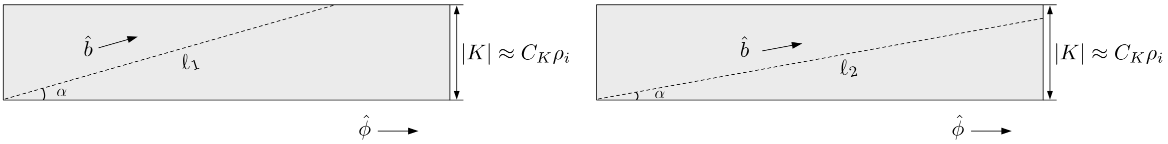
$$\begin{aligned}
\dot{\boldsymbol{X}} &= v_\parallel\frac{\boldsymbol{B}^*}{B_\parallel^*} + \frac{\hat{b}}{B_\parallel^*}\times\frac{c}{q_s}\mu_s\nabla B \\
&= \frac{v_\parallel}{B_\parallel^*}\left(\boldsymbol{B} + \frac{m_s c}{q_s}v_\parallel\nabla\times\hat{b}\right) + \frac{\hat{b}}{B_\parallel^*}\times\frac{c}{q_s}\mu_s\nabla B \\
&\approx \frac{v_\parallel}{B_\parallel^*}\left(\boldsymbol{B} + \frac{m_s c}{q_s}v_\parallel\frac{1}{R}\hat{z}\right) - \frac{\mu_s c}{q_s B_\parallel^*}\hat{b}\times\frac{B_0 R_0}{R^2}\hat{R} \qquad (\text{since } (47)) \\
&\approx \underbrace{v_\parallel\hat{b}}_{:=\dot{\boldsymbol{X}}_{\text{tor}}} + \underbrace{\frac{m_s c v_\parallel^2}{q_s R B_\parallel^*}\hat{z} - \frac{m_s c v_\perp^2}{2q_s R B_\parallel^*}\hat{b}\times\hat{R}}_{:=\dot{\boldsymbol{X}}_{\text{pol}}}. \\
&\qquad \left(\text{since } \mu_s = \frac{m_s v_\perp^2}{2B},\ B/B_\parallel^* \approx 1, \text{ and } (47)\right)
\end{aligned} \tag{55}$$

The poloidal velocity can be expressed by

$$\begin{aligned}
|\dot{\boldsymbol{X}}_{\text{pol}}| &\lesssim \frac{m_s c v_\parallel^2}{q_s R B_\parallel^*} + \frac{m_s c v_\perp^2}{2q_s R B_\parallel^*} \\
&\le \frac{C_r C_{v_\parallel}^2 m_s c v_{\text{th,s}}^2}{q_s R_0 B_\parallel^*} + \frac{C_r C_{v_\perp}^2 m_s c v_{\text{th,s}}^2}{2q_s R_0 B_\parallel^*} \qquad (\text{since } (45) \text{ and } (46)\ ) \\
&= C_r\left(C_{v_\parallel}^2 + \frac{1}{2}C_{v_\perp}^2\right)\frac{m_s c v_{\text{th,s}}^2}{q_s R_0 B_\parallel^*}.
\end{aligned} \tag{56}$$

Dividing $\Delta X_{\text{pol}}/|\dot{\boldsymbol{X}}_{\text{pol}}|$ by the normalization factor $R_0/v_{\text{th,i}}$ and using (54), (57), we arrive at

$$\begin{aligned}
\frac{\Delta X_{\text{pol}}/|\dot{\boldsymbol{X}}_{\text{pol}}|}{R_0/v_{\text{th,i}}} &\gtrsim \frac{C_K m_i c v_{\text{th,i}}/(q_i B)}{C_r\left(C_{v_\parallel}^2 + \frac{1}{2}C_{v_\perp}^2\right) m_s c v_{\text{th,s}}^2/(q_s R_0 B_\parallel^*)}\frac{v_{\text{th,i}}}{R_0} \\
&\approx \frac{C_K}{C_r\left(C_{v_\parallel}^2 + \frac{1}{2}C_{v_\perp}^2\right)}\frac{m_i v_{\text{th,i}}^2}{m_s v_{\text{th,s}}^2}\frac{q_s}{q_i} \\
&\qquad \left(\text{since } B/B_\parallel^* \approx 1\right) \\
&= \frac{C_K}{C_r\left(C_{v_\parallel}^2 + \frac{1}{2}C_{v_\perp}^2\right)}\frac{T_i}{T_s}\frac{q_s}{q_i}
\end{aligned} \tag{57}$$



(a)

(b)

Fig. 13. Examples of the time interval restriction in the primitive meshes by the (a) poloidal and (b) toroidal mesh sizes, where K indicates a poloidal element.

$$\approx \frac{C_K}{C_r\left(C_{v_\parallel}^2 + \frac{1}{2}C_{v_\perp}^2\right)} \frac{q_s}{q_i}. \quad \left(\text{since } T_i \approx T_s\right)$$

Consequently, we can get a sufficient condition for Δt_{pol} as follows:

$$\frac{\Delta t_{\mathrm{pol}}}{R_0/v_{\mathrm{th,i}}} \lesssim \frac{C_K}{C_r\left(C_{v_\parallel}^2 + \frac{1}{2}C_{v_\perp}^2\right)} \frac{q_s}{q_i}. \tag{58}$$

3.3. Estimation of $\Delta t_{\mathrm{tor}}^{\mathrm{pri}}$ for the primitive mesh

For the estimation of the primitive mesh, we consider two distinct cases, as depicted in Fig. 13. The figure illustrates a simplified element within the primitive mesh by representing the poloidal section as a line. The time step is influenced by the size of the mesh, specifically (a) poloidal mesh and (b) toroidal mesh, as presented in Fig. 13.

We refer to (54), which indicates that $|K| \approx C_K \rho_i$ where $K \in \mathcal{T}_{h,x}^{\mathrm{pol}}$ is a poloidal element. In Fig. 13(a), α denotes the angle between the two vectors $\hat{b}$ and $\hat{\phi}$. Then, using $q \approx (rB_t)/(R_0 B_p)$, we have

$$\sin\alpha \approx \frac{B_p}{B} \approx \frac{C_K \rho_i}{\ell_1}. \tag{59}$$

This implies that

$$\ell_1 \approx C_K \rho_i \frac{B}{B_p} \geq C_K \rho_i \frac{B_t}{B_p} = C_K \rho_i \frac{qR_0}{r} \geq C_K \frac{q}{\epsilon}\rho_i. \tag{60}$$

Similarly, in Fig. 13(b), the length of the bottom edge can be approximated by $2\pi R/N_\phi$. Then we have

$$\cos\alpha \approx \frac{2\pi R/N_\phi}{\ell_2}. \tag{61}$$

This implies that

$$\ell_2 \approx \frac{2\pi R/N_\phi}{\cos\alpha} \geq \frac{2\pi R}{N_\phi} \geq \frac{2\pi R_0}{C_r N_\phi}. \tag{62}$$

(57) yields $|\dot{X}_{\mathrm{tor}}| \approx v_\parallel$. By normalizing $\Delta X_{\mathrm{tor}}/|\dot{X}_{\mathrm{tor}}|$ with respect to $R_0/v_{\mathrm{th,i}}$ and using (60) and (62), we obtain

$$\begin{aligned}
\frac{\Delta X_{\mathrm{tor}}/|\dot{X}_{\mathrm{tor}}|}{R_0/v_{\mathrm{th,i}}} &\approx \frac{v_{\mathrm{th,i}}}{R_0}\frac{\Delta X_{\mathrm{tor}}}{v_\parallel} \gtrsim \frac{\Delta X_{\mathrm{tor}}}{C_{v_\parallel} R_0}\frac{v_{\mathrm{th,i}}}{v_{\mathrm{th,s}}} \\
&\approx \frac{\Delta X_{\mathrm{tor}}}{C_{v_\parallel} R_0}\sqrt{\frac{m_s}{m_i}} = \frac{1}{C_{v_\parallel}}\sqrt{\frac{m_s}{m_i}}\frac{\min\{\ell_1,\ell_2\}}{R_0} \\
&\gtrsim \frac{1}{C_{v_\parallel}}\sqrt{\frac{m_s}{m_i}}\begin{cases} C_K \dfrac{\rho_i}{R_0}\dfrac{q}{\epsilon} & \text{if } \ell_1 < \ell_2, \\ \dfrac{1}{C_r}\dfrac{2\pi}{N_\phi} & \text{otherwise.}\end{cases}
\end{aligned} \tag{63}$$

Using $\cos\alpha \approx B_t/B$ in (61), the condition for $\ell_1 < \ell_2$ is approximated by

$$N_\phi < \frac{2\pi C_r}{C_K}\frac{\epsilon}{q}\frac{R_0}{\rho_i}. \tag{64}$$

Let N_ϕ^c denote the right term in (64). We can conclude that $\Delta t_{\mathrm{tor}}^{\mathrm{pri}}$ satisfying the following condition

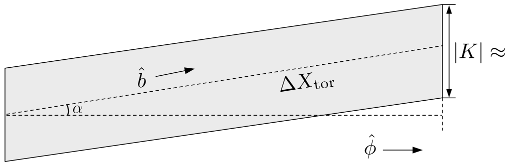


Fig. 14. Example of time interval restriction in the field-aligned mesh.

$$\frac{\Delta t_{\mathrm{tor}}^{\mathrm{pri}}}{R_0/v_{\mathrm{th,i}}} \lesssim \begin{cases} \dfrac{C_K}{C_{v_\parallel}}\dfrac{\rho_i}{R_0}\dfrac{q}{\epsilon}\sqrt{\dfrac{m_s}{m_i}} & \text{if } N_\phi < N_\phi^c, \\ \dfrac{1}{C_{v_\parallel} C_r}\dfrac{2\pi}{N_\phi}\sqrt{\dfrac{m_s}{m_i}} & \text{otherwise,}\end{cases} \tag{65}$$

where $\Delta t_{\mathrm{tor}}^{\mathrm{pri}}$ denotes the feasible time interval in the primitive mesh. This provides the temporal stability condition.

3.4. Estimation of $\Delta t_{\mathrm{tor}}^{\mathrm{fi}}$ with the field-aligned mesh

Fig. 14 shows a simplified twist-prism element in the field-aligned mesh. The angle α can be approximated as

$$\cos\alpha \approx \frac{2\pi R/N_\phi}{\Delta X_{\mathrm{tor}}}. \tag{66}$$

From this relation, we can obtain that

$$\Delta X_{\mathrm{tor}} \approx \frac{1}{\cos\alpha}\frac{2\pi R}{N_\phi} \geq \frac{2\pi R}{N_\phi}. \tag{67}$$

The last term relies on the approximation provided in (47). By normalizing $\Delta X_{\mathrm{tor}}/|\dot{X}_{\mathrm{tor}}|$ with $R_0/v_{\mathrm{th,i}}$ and using (56) and (67), we arrive at the following expression

$$\begin{aligned}
\frac{\Delta X_{\mathrm{tor}}/|\dot{X}_{\mathrm{tor}}|}{R_0/v_{\mathrm{th,i}}} &\gtrsim \frac{(2\pi R/N_\phi)/v_\parallel}{R_0/v_{\mathrm{th,i}}} = \frac{v_{\mathrm{th,i}}}{v_\parallel}\frac{R}{R_0}\frac{2\pi}{N_\phi} \\
&\geq \frac{1}{C_{v_\parallel} C_r}\frac{v_{\mathrm{th,i}}}{v_{\mathrm{th,s}}}\frac{2\pi}{N_\phi} \qquad (\text{since (45) and (46)}) \\
&\approx \frac{1}{C_{v_\parallel} C_r}\frac{2\pi}{N_\phi}\sqrt{\frac{m_s}{m_i}}. \qquad (\text{since (48)})
\end{aligned} \tag{68}$$

Therefore, it can be concluded that if $\Delta t_{\mathrm{tor}}^{\mathrm{fi}}$ satisfies

$$\Delta t_{\mathrm{tor}}^{\mathrm{fi}} \lesssim \frac{1}{C_{v_\parallel} C_r}\frac{2\pi}{N_\phi}\sqrt{\frac{m_s}{m_i}}, \tag{69}$$

then the stability condition is satisfied, where $\Delta t_{\mathrm{tor}}^{\mathrm{fi}}$ denotes the feasible time step in the field-aligned mesh.

3.5. Stability condition for electrons

We consider the sufficient condition for stability in electron simulation by analyzing $\Delta t_{v_\parallel}$, Δt_{pol}, and Δt_{tor}. Since the mass ratio m_s/m_i is significantly small for electrons, Δt_{pol} in (58) does not affect the time step size constraints, as it does not incorporate the mass ratio. Let us assume that $|K| \approx \rho_i$, leading to $C_K \approx 1$. For estimation of numerical

Table 1
Temporal stability test results using $N_\phi = 8$, $N_{\rm pol} = 22126$: ○ and ╳ denote stable and unstable computations, respectively.

Δt	Primitive mesh	Field-aligned mesh
1.0×10^{-3}	○	○
2.0×10^{-3}	╳	○
5.0×10^{-3}	╳	○
8.0×10^{-3}	╳	○
9.0×10^{-3}	╳	○
1.0×10^{-2}	╳	╳

Table 2
Number of elements for the convergence test.

$N(=N_{\rm pol}N_\phi)$	$N_{\rm pol}$	N_ϕ
10,864	1,358	8
93,536	5,846	16
708,032	22,126	32
6,154,368	96,162	64
51,105,536	399,262	128

values, we choose a set of parameters typical for gyrokinetic turbulence simulation $\rho_i \approx 0.003$, $a = \epsilon R_0$, $\epsilon = 0.3$, $q \approx 1.4$, $C_r^{-1} = 0.7$, $C_R = 1.3$, $N_\phi = 16$, $C_{v_\parallel} = C_{v_\perp} = 5$, and $n_{v_\parallel} = 10$. Under the parameter values, we find that $N_\phi^c \approx 534$, which is impractically large for feasible gyrokinetic simulations; thus, we consider only $N_\phi < N_\phi^c$ case in (65). Again, applying our simulation parameters with $N_\phi = 16$, we have

$$\frac{\Delta t_{v_\parallel}}{R_0/v_{\rm th,i}} \lesssim \frac{4C_{v_\parallel}}{\epsilon C_r C_{v_\perp}^2}\frac{q}{n_{v_\parallel}}\sqrt{\frac{m_s}{m_i}} \approx \frac{4\times5}{0.3\times0.7\times5^2}\frac{1.4}{10}\sqrt{\frac{m_s}{m_i}} \approx 5.33\times10^{-1}\sqrt{\frac{m_s}{m_i}}, \tag{70}$$

$$\frac{\Delta t_{\rm tor}^{\rm pri}}{R_0/v_{\rm th,i}} \lesssim \frac{C_K\rho_i}{C_{v_\parallel}}\frac{q}{\epsilon R_0}\sqrt{\frac{m_s}{m_i}} \approx \frac{0.003}{5}\frac{1.4}{0.3\times1.7}\sqrt{\frac{m_s}{m_i}} \approx 1.65\times10^{-3}\sqrt{\frac{m_s}{m_i}}, \tag{71}$$

$$\frac{\Delta t_{\rm tor}^{\rm fi}}{R_0/v_{\rm th,i}} \lesssim \frac{1}{C_{v_\parallel}C_r}\frac{2\pi}{N_\phi}\sqrt{\frac{m_s}{m_i}} \approx \frac{0.7}{5}\frac{2\pi}{16}\sqrt{\frac{m_s}{m_i}} \approx 5.50\times10^{-2}\sqrt{\frac{m_s}{m_i}}. \tag{72}$$

Thus, we can conclude that the $\Delta t_{\rm tor}$ mainly constrains the time step size. Additionally, field-aligned meshes can enhance the temporal stability condition compared to using the primitive mesh in electron simulations.

Table 1 shows the result of temporal stability test under the simulation parameters with $N_{\rm pol} = 22126$, which is lower than for $C_K \approx 1$. However, it should be sufficient to validate our estimations. The analysis shows that simulations utilizing a field-aligned mesh exhibit better stability condition than those using the primitive mesh. This finding supports the advantages of the field-aligned mesh in selecting time step size and improving overall computational performance. However, as discussed in Sec. 2.6, the field-aligned mesh necessitates the utilization of sub-triangulation. The process entails additional quadrature computations compared to the primitive mesh, which can reduce the overall computation performance. It is important to note that the field-aligned mesh may require less toroidal resolution than the primitive mesh to achieve similar accuracy. This advantage can compensate for the additional computational cost caused by the sub-triangulation. The details will be discussed in Sec. 4.2.

4. Numerical experiments

In this section, we present numerical experiment results to validate the new solver.

4.1. Convergence tests

We consider the ballooning function on the concentric circular domain as follows:

$$f_0(R,z,\phi) = C(R,z,\phi)e^{-\left(\frac{x_r-x_{r0}}{\Delta x_r}\right)^2} \tag{73}$$

where x_r denotes radial coordinate, x_{r0} is radial position of the surface rational, Δx_r is radial envelope width,

$$C(R,z,\phi) = \sum_{m=m_0-\Delta m}^{m_0+\Delta m} \cos\left(n\phi + s_B m\Lambda\right) e^{-\delta f_{ac}(m-nq)^2}. \tag{74}$$

Here, $s_B = \mathrm{sgn}(B_{\rm pol})$, $q = q(x_r)$ is safety factor, Λ is magnetic poloidal angle, δf_{ac} is radial mode width, n and m are the poloidal and toroidal mode numbers, respectively; $m_0 = nq(x_0)$ is resonant poloidal mode number, Δm is poloidal mode width. The concentric circular domain Ω_x is given by

$$\Omega_x = \left\{(R,z,\phi) : (R-R_0)^2 + z^2 \le a^2,\ 0 \le \phi < 2\pi\right\}, \tag{75}$$

where $R_0 = 1.7$ (m) is the major radius of the magnetic axis, and $a = 0.6$ (m) is the minor radius of the simulation boundary.

We perform the convergence tests with $n = 10$ using the q-profile in Fig. 16, where r/a represents the minor radius normalized by the maximal radius a. The ballooning function f_0 in (73) and (74) is projected into the DG spaces generated by linear, quadratic, and cubic polynomial bases through the cubic isoparametric transform introduced in Sec. 2. For the comparison, two kinds of meshes are considered. Fig. 15 shows (a) the primitive and (b) the field-aligned meshes. The primitive mesh consists of the elements aligned along the $\hat{e}_\phi$ (or rotating along z-axis), while the field-aligned mesh has elements aligned along the modified unit magnetic field $\tilde{b}$. The L^2-projection gives the projection of f_0 such that

$$\sum_{K_x\in\mathcal{T}_h^x}\int_{K_x}\left(\mathring{f}_h - f_0\right)\xi_j\, R\,dR\,dz\,d\phi = 0, \tag{76}$$

for all $\xi_j \in \mathbb{V}_h(\Omega_x)$, where $\mathring{f}_h = \sum_i c_i\xi_i$ is the projection of f_0. Table 2 shows the number of elements for the convergence tests, where $N_{\rm pol}$ is the number of elements on each poloidal plane, and N_ϕ is the number of toroidal intervals.

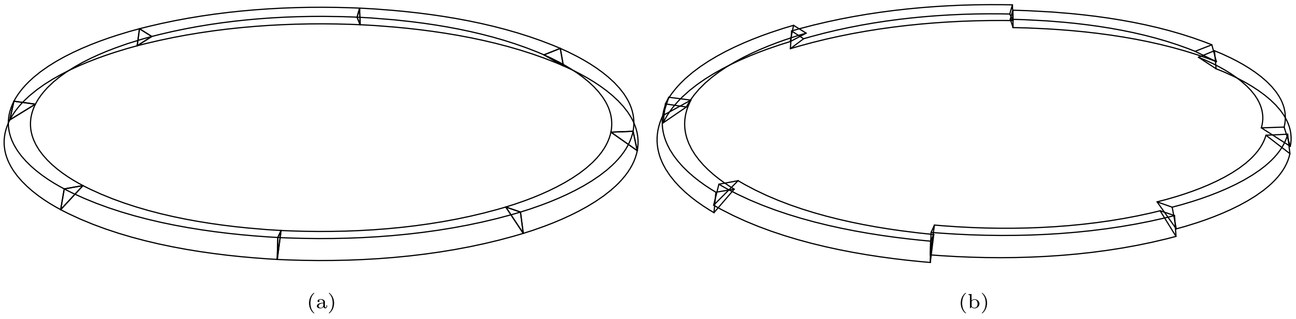
The L^2-norms of interpolation errors $e_h = f_h - f_0$ is calculated in the broken Sobolev space [25] as follows:

$$\|e_h\| = \left(\sum_{K\in\mathcal{T}_h}\int_K \left|f_h - f_0\right|^2 R\,dR\,dz\,d\phi\right)^{1/2}. \tag{77}$$

Fig. 17 shows the results of the convergence tests using primitive and field-aligned meshes, where the mesh size h is approximated as $\left(|\Omega_x|/N\right)^{1/3}$. Here, N is the number of elements, $|\Omega_x|$ is the volume of Ω_x. All cases give results that agree with the optimal accuracy orders [25], that is, $\|e_h\| = \mathcal{O}(h^{p+1})$ and $\|\nabla e_h\| = \mathcal{O}(h^p)$, where p is the basis order. From these results, we can conclude that the field-aligned mesh does not spoil the accuracy of L^2-projection scheme.

4.2. Accuracy test depending on the toroidal resolution

In this section, we perform a numerical experiment using linear, quadratic, and cubic basis to investigate the performance of field-aligned mesh corresponding to toroidal resolution with various mesh sizes listed in Table 3. The ballooning function in (73) is also considered with toroidal mode numbers $n = 1, 10, 20, 40$, while the poloidal mode numbers m are determined by satisfying the relation $q = m/n$ under given q-profile shown in Fig. 16.



(a)

(b)

Fig. 15. (a) Primitive and (b) field-aligned meshes ($N_\phi = 8$).

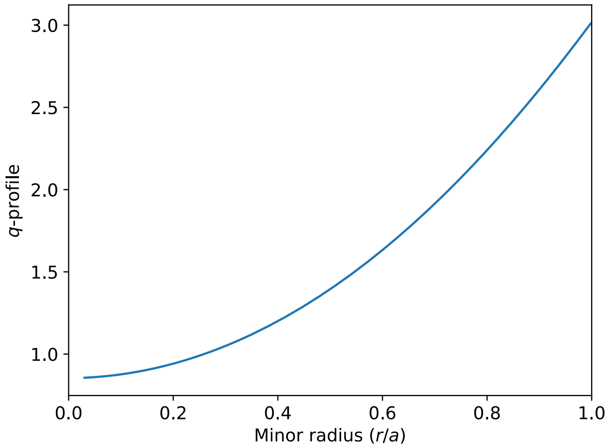


Fig. 16. q-profile plot.

Table 3
Number of elements in the interpolation test related to toroidal resolution.

$N(=N_{\text{pol}}N_\phi)$	N_{pol}	N_ϕ
6,388,192	399,262	16
12,776384	399,262	32
25,552,768	399,262	64
51,105,536	399,262	128
102,211,072	399,262	256
204,422,144	399,262	512

Fig. 18 shows the contours of the ballooning function related to the toroidal mode numbers: (a) poloidal contours at a plane $\phi = 0$ and (b) toroidal contours on a surface $r/R_0 = 0.2$. The higher toroidal mode numbers are associated with smaller and highly field-following structures, which require a dense mesh or numerous toroidal subintervals for accurate representation. In Fig. 19, the field-aligned cases demonstrate better results than the primitive cases, and this improvement tends to increase with higher order bases, except for the $n = 1$ cases. The numbers shown between the blue and red plots represent the error ratios between the primitive and field-aligned cases at each toroidal resolution. These ratios increase with the higher toroidal modes ($n = 10, 20, 40$). The ballooning function with $n = 1$ exhibits an almost isotropic structure, indicating that the field-aligned mesh does not provide an advantage in the cases. This trend also holds for both the gradients and parallel derivatives of the ballooning function, as shown in Figs. 20 and 21, respectively. Furthermore, the field-aligned cases saturate faster than primitive cases; they quickly reach the accuracy limit imposed by poloidal mesh resolution. In particular, the convergences of parallel derivatives for the primitive mesh fail to achieve sufficient convergence in these numerical tests, except for $n = 1$ cases. These results indicate that the field-aligned mesh is essential for improving the efficiency of simulations involving the field-following structure. It is interesting to notice that the field-aligned meshes show lower accuracies in representing ballooning structures with low toroidal mode numbers (e.g., $n = 1$ case in Fig. 20). The ballooning structures have different degrees of field-aligning. For high toroidal mode number cases, the structures closely follow ambient magnetic fields, and therefore, the interpolation accuracies are superior for the field-aligned meshes. On the other hand, ballooning structures with low toroidal model numbers do not closely follow ambient magnetic fields, which results in lower interpolation accuracies of the field-aligned meshes. However, these issues are relatively minor, as they arise from the almost isotropic nature of the target function, which can be disregarded in tokamak plasma environments.

The previous convergence results indicate that the field-aligned mesh can enhance overall computation efficiency compared to the primitive mesh. This improvement is achieved by reducing the required toroidal resolution while maintaining the same level of accuracy. However, the inevitable sub-triangulation for field-aligned mesh leads to additional computational costs. To evaluate the advantages and disadvantages of the field-aligned mesh, we perform a numerical test. Table 4 compares the computation times of simulations conducted using (a) the primitive mesh and (b) the field-aligned mesh. Based on the error convergence plots for the case $n = 10$ in Fig. 19(b), we assume that the primitive mesh requires approximately eight times more toroidal resolution than the field-aligned mesh to attain the same level of accuracy. Table 4 shows slightly reduced computation time for the field-aligned mesh compared to the primitive mesh under the same simulation parameters, excluding toroidal resolution. Consequently, the field-aligned mesh can offer considerable advantages in computational efficiency when considering the improvement of the stability conditions discussed in Sec. 3.

4.3. Conservation properties tests

We consider the concentric circular domain $\Omega = \Omega_x \times \Omega_v$, where Ω_x is defined in (75) and Ω_v is given by

$$\Omega_v = \left\{ (v_\parallel, u) : -v_{\max} \le v_\parallel \le v_{\max}, 0 \le u \le v_{\max} \right\}, \tag{78}$$

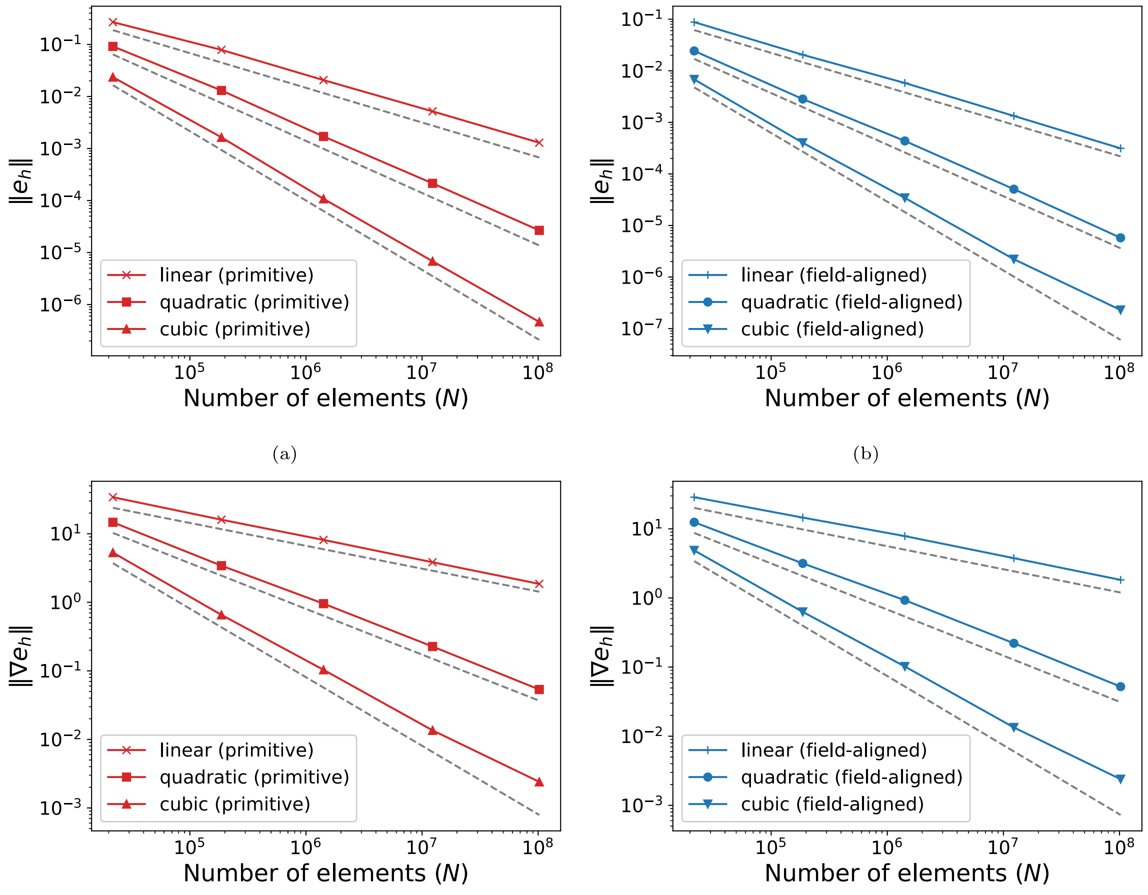
where ψ is the poloidal magnetic flux. Here, $v_{\max}/v_{\text{th,i}} = 5$ is used when $v_{\text{th,i}}$ is defined as $\sqrt{T_0/m_i}$, and $T_0 = 2$ (keV) is the ion temperature at the magnetic axis. Based on the initial density and temperature profiles shown in Fig. 22, the local Maxwellian function f_M defined as

$$f_M = \left(\frac{m_i}{2\pi}\right)^{\frac{3}{2}} \frac{n(\psi)}{T(\psi)^{3/2}} \exp\left(-\frac{m_i}{2T(\psi)}\left(v_\parallel^2 + \frac{u^2 B}{B_0}\right)\right), \tag{79}$$

where ψ is the poloidal magnetic flux. f_M is imposed at the initial time using the L^2-projection such that

$$\sum_{K \in \mathcal{T}_h} \int_K (f_M - f_h)\xi_j R u B_\parallel^* dR dz d\phi dv_\parallel du, \tag{80}$$

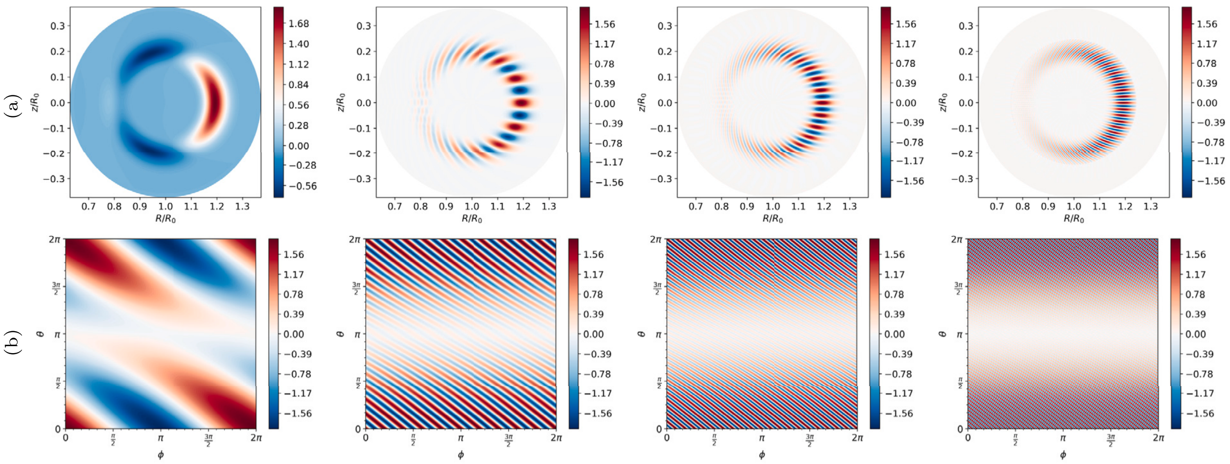
for all $\xi_j \in \mathbb{V}_h(\Omega)$, and the up-down symmetric boundary condition is applied, that is, $f(R, z, \phi, v_\parallel, u) = f(R, -z, \phi, v_\parallel, u)$ on the bound-



(a) (b)

(c) (d)

Fig. 17. Convergence test results using (left) the primitive and (right) the field-aligned meshes.



(n = 1) (n = 10) (n = 20) (n = 40)

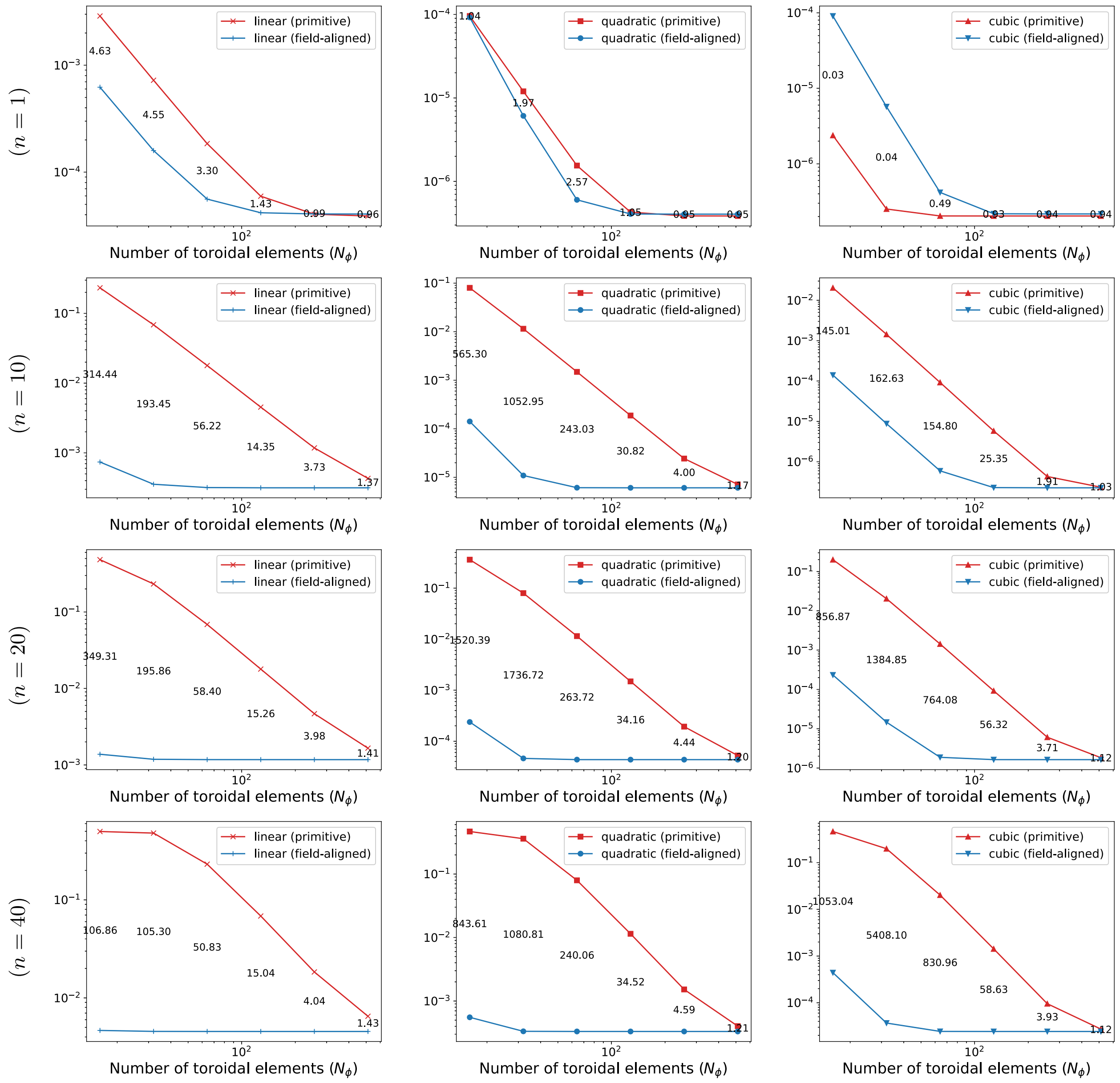
Fig. 18. Contour plots of ballooning functions of (a) poloidal section at $\phi = 0$ and (b) toroidal surface at $r/R_0 = 0.2$.

ary, and the zero-flux boundary condition is used in velocity space. In this simulation, the field-aligned meshes are constructed with 4, 8, 16, and 32 subdomains. Each poloidal mesh consists of 5846 unstructured triangles, and the velocity domain is discretized using 10×5 rectangles. This study uses the quadratic basis functions defined in (21). We investigate the conservation properties of the total mass ρ, kinetic energy E_K, and toroidal canonical momentum P_ϕ defined as

$$\rho = \int_{K\in\mathcal{T}_h} f_h B_\parallel^* d\xi, \tag{81}$$

$$E_K = \int_{K\in\mathcal{T}_h} \hat{E}_K f_h B_\parallel^* d\xi, \quad \hat{E}_K = v_\parallel^2 + u^2 B/B_0, \tag{82}$$

$$P_\phi = \int_{K\in\mathcal{T}_h} \hat{P}_\phi f_h B_\parallel^* d\xi, \quad \hat{P}_\phi = q_i\psi/c - m_i R v_\parallel B_\phi/B, \tag{83}$$



(a) Linear (b) Quadratic (c) Cubic

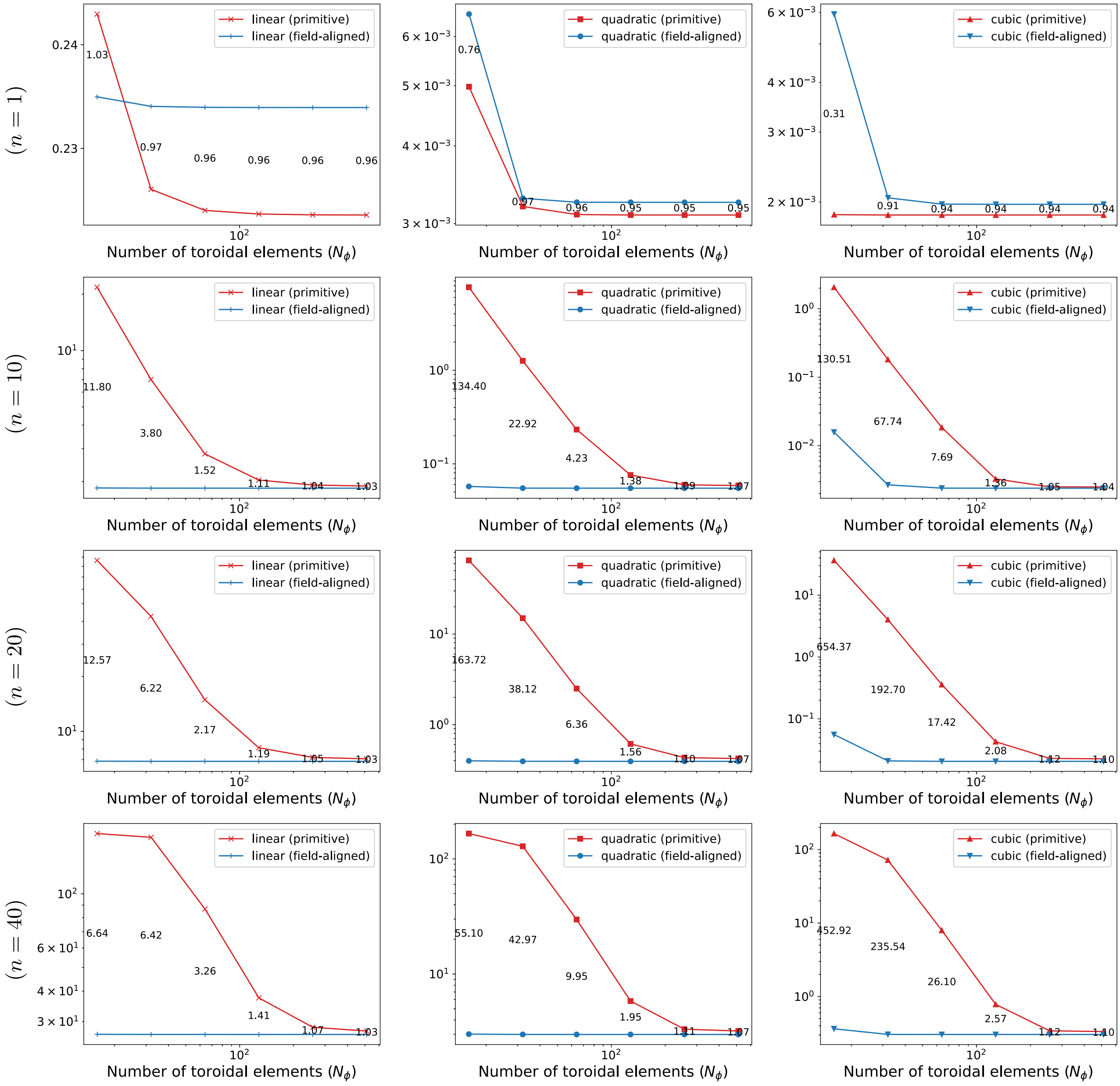
Fig. 19. Convergence slopes of $\|e_h\|$ depending on toroidal resolution (N_ϕ). The numbers between the slopes represent the error ratios of the primitive and field-aligned cases.

Table 4
Comparison between the computation times with the primitive and field-aligned meshes under the same simulation parameters, except for N_ϕ. *The number of sub-triangles in the primitive mesh is interpreted as a result of sub-triangulation that consists of one sub-triangle. Therefore, it is equivalent to the number of triangles in the poloidal section.

	Primitive mesh (a)	Field-aligned mesh (b)	Ratio (a)/(b)
Computation time (T)	~ 5.22 h	~ 3.65 h	~ 1.43
Number of sub-triangles (N_s)	5846*	68756	
Toroidal resolution (N_ϕ)	32	4	8

where $d\xi = Rud\,Rdzd\phi dv_\parallel du$, B_ϕ and q_i are the toroidal component of magnetic field and the ion charge, respectively. The initial f_h exhibits magnetic drift motions with time step $\Delta t = 10^{-3} R_0/v_{\rm th}$, resulting in the development of non-concentric patterns. Fig. 23 shows the initial and final ($t = 2R_0/v_{\rm th}$) contours of solutions at $(v_\parallel/v_{th}, u/v_{th}) = (1, 1/2)$.

Fig. 24 presents the temporal evolution of ρ, E_K, and P_ϕ from their initial states. All cases exhibit perfect mass conservations up to a machine precision level, thanks to the successfully applying the sub-triangulation scheme introduced in Sec. 2.6. Since the kernels of physical quantities contain polynomial terms, including the polynomials in the trial function space significantly improves the conservations,



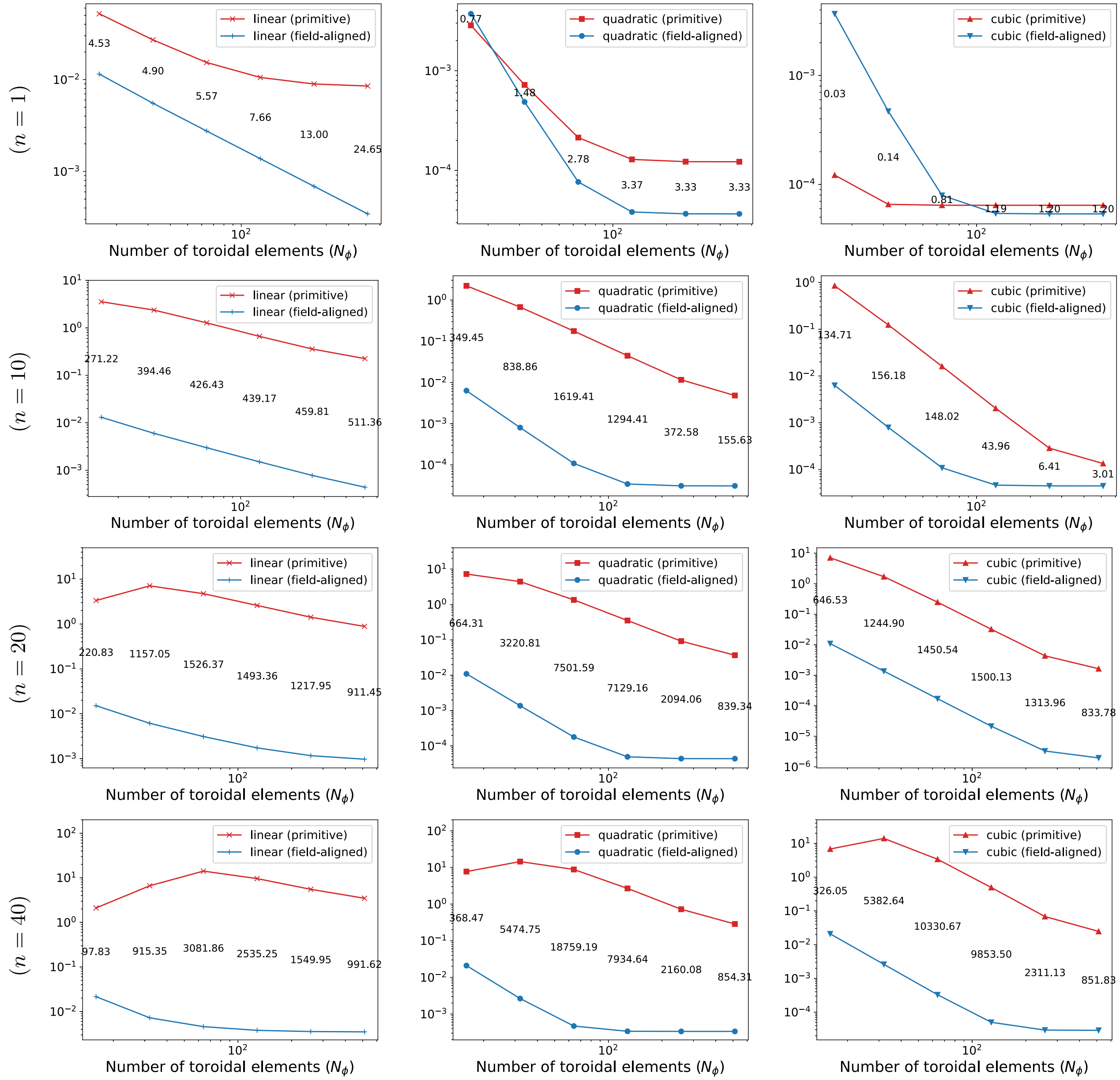
(a) Linear (b) Quadratic (c) Cubic

Fig. 20. Convergence slopes of $\|\nabla e_h\|$ depending on toroidal resolution (N_ϕ). The numbers between the slopes represent the error ratios of the primitive and field-aligned cases.

as highlighted in the previous research [27]. However, in this study, the basis functions within the physical elements are not polynomials due to parametric transformations, even though they are polynomials on the reference element. The second and third rows in Fig. 24 show improvements in the conservation of kinetic energy and angular momentum as the toroidal resolution increases. These results indicate that higher toroidal resolutions reduce the distortion of physical elements and enable the basis functions to better approximate the polynomial terms within the kernels. Therefore, this observation emphasizes that the toroidal resolution is an important factor for conservation. Fig. 25 shows the conservation properties on the primitive mesh. When comparing the results in Figs. 24 and 25, the conservation of the toroidal canonical angular momentum is mainly affected by computational accuracy, which is determined by the choice of basis and mesh size rather than mesh structures.

4.4. *KSTAR geometry*

We investigate the conservation properties in the KSTAR geometry using an absorbing boundary condition, which allows outward flux while preventing inflow. Figs. 26(a) and 26(b) show the contours of initial and final solutions. Our approach employs 99515 unstructured triangles in each poloidal mesh in Fig. 26(c) and 10×5 rectangular elements in the velocity domain. The entire domain is discretized into 796120 twisted prisms, with each nonconforming surface having 920182 subtriangles with $N_\phi = 8$. The time step size is $\Delta t = 5\times10^{-4}\ (R_0/v_{th})$, while the overall simulation duration is given by $T = 2\ (R_0/v_{th})$ Our analysis includes tracking variants of the physical properties in the equation in the KSTAR domain. By accounting for the fluxes passing through the boundary surface, we can verify the quality of the conservation properties. Fig. 27 shows the temporal evolution of physical quantities after adding their fluxes across the boundary surface. The results confirm that



(a) Linear (b) Quadratic (c) Cubic

Fig. 21. Convergence slopes of $\|\hat{b}\cdot\nabla e_h\|$ depending on toroidal resolution (N_ϕ). The numbers between the slopes represent the error ratios of the primitive and field-aligned cases.

these quantities satisfy conservations similar to those observed in concentric circular geometries.

4.5. Parallelization performance

To evaluate the parallelization performance of the solver, we conduct a strong scaling analysis by incrementally increasing the number of processors while maintaining a constant problem size. Fig. 28(a) shows the relation between the speedup (S) and the number of CPU cores (N_p) for two distinct problem sizes. Consider the unit computational time denoted as $T(1)$, which presents the computational time when a single process is used. Directly estimating the unit computational time is impossible because the solver requires more than one core to solve the target problem within a reasonable amount of time. Thus, this paper approximates $T(1)$ using an extrapolation such as $T(N_s)/N_s$, where N_s is the smallest number of CPU cores employed in the analysis. The speedup is defined as

$$S(N_p) = \frac{T(1)}{T(N_p)}, \tag{84}$$

where $T(N_p)$ denotes the computational time required to complete the simulation using N_p processors. The gray dashed lines in the figure indicate the ideal speedup defined as (84), while the red and blue solid lines correspond to the real speedups. The result reveals that the solver shows good scalability up to three thousand cores.

Fig. 28(b) presents the parallelization efficiencies (E) defined as the ratio of real speedup to ideal speedup, expressed as follows:

$$E(N_p) = \frac{S(N_p)}{N_p} = \frac{T(1)}{T(N_p) * N_p}. \tag{85}$$

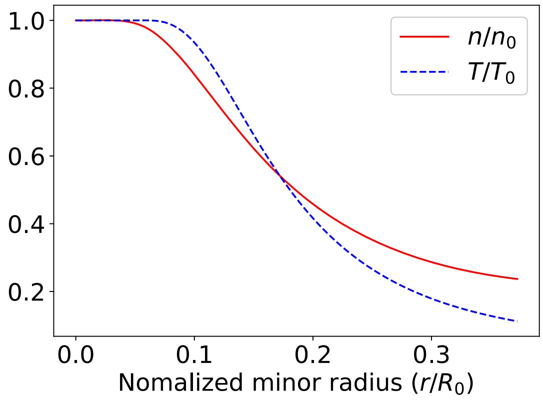


Fig. 22. Normalized density and temperature profiles where r is a minor radius.

The results indicate that the solver retains an acceptable level of efficiency, even with several thousand cores.

4.6. *Temporal stationary test*

We consider the canonical Maxwellian function for the stationary performance test as an initial value defined as

$$f_{CM} = \left(\frac{m_i}{2\pi}\right)^{\frac{3}{2}} \frac{n(\hat{P}_\phi)}{T(\hat{P}_\phi)^{3/2}} \exp\left(-\frac{m_i}{2T(\hat{P}_\phi)}\left(v_\parallel^2 + \frac{u^2 B}{B_0}\right)\right), \tag{86}$$

which is a stationary solution of the gyrokinetic equation that remains constant over time. Fig. 29 shows the cross-sections of the initial ($t = 0(R_0/v_{\text{th}})$) and final ($t = 10(R_0/v_{\text{th}})$) solution at $(R/R_0, z/R_0) = (1.2, 0.5)$ using 20×10 regular mesh in the velocity space and 5848×16 unstructured mesh in the spatial domain with the toroidal resolution $N_\phi = 16$. The plots indicate that f_{CM} remains nearly unchanged, which supports the fact that the solver can accurately represent the temporal behaviors of the canonical Maxwellian function. Fig. 30 presents the conservation properties of the temporal stationary test. The results show good conservations, which is consistent with the results in Sec. 4.3.

5. Summary and discussion

In this work, we developed novel numerical schemes to solve a partial differential equation describing plasma turbulence in toroidal geometry. Focusing on the structures of plasma turbulence, which are well-aligned with externally imposed equilibrium magnetic fields, our development aimed to provide a general mathematical framework 1) to partition the toroidal domain into curved sub-domains which are aligned with equilibrium magnetic fields and 2) to discretize the 5D gyrokinetic hyperbolic equation based on the discontinuous Galerkin method in the sub-domains. In the development, our main efforts were given to enable global simulation covering a whole toroidal domain bounded by the realistic geometry of magnetic fusion devices. The mathematical formulations were developed independent of flux coordinates, and the discretization was performed based on unstructured triangular meshes, which can represent arbitrarily complex domain shapes.

Though the schemes were developed for gyrokinetic simulation in toroidal geometry, it should be noted that the numerical constructions are made in a general way based upon differential geometric formulation of the equations. The developed schemes and techniques, e.g., the sub-division of the simulation domain, constructions of finite elements, and numerical quadratures in the sub-divided domains, are applicable for other numerical simulations of fusion plasma in toroidal geometry.

A dedicated effort was given to handle nonconforming meshes, which are inevitable due to the curved and non-periodic natures of equilibrium magnetic fields confining fusion plasma in toroidal geometry. The sorting and classification of the nonconforming overlapping patterns of triangular meshes, the sub-division of the overlapping regions, and the various numerical constructions therein were carried out in systematic and algorithmic ways, which can also be applied to other numerical problems.

For the newly developed field-aligned mesh, a new CFL condition was estimated to ensure stable temporal discretization, and it was compared with the condition for the conventional toroidally straight discretization. It was found that the new field-aligned mesh allows significantly increased time steps. However, near the boundary, the field-aligned mesh has a non-aligned structure similar to that of primitive mesh, as defined in (11). The non-aligned structure can restrict the temporal stability condition. To prevent this situation, it is necessary to implement an adaptive velocity grid [31] in real space. With the adaptive grid, the local maximum $v_\parallel$ is tied to the local temperature. Since the electron temperature near the boundary is typically much lower than the core electron temperature, the maximum $v_\parallel$ near the boundary is correspondingly smaller than the maximum $v_\parallel$ at the core. Due to the reduced $v_\parallel$ at the boundary, the temporal stability condition near the boundary becomes much less restrictive, even with the primitive mesh. Although the adaptive velocity grid has not yet been implemented in the current work, development is ongoing, and results will be reported in the near future.

Numerical tests were performed to study the interpolation accuracies of the newly proposed schemes by employing ballooning functions representing the characteristics of plasma turbulence structures in toroidal geometry. From the numerical tests conducted with three-dimensionally increasing mesh numbers, the interpolation accuracies of both field-aligned and conventional meshes showed expected convergence rates according to the order of the basis functions employed for the interpolations, though the field-aligned cases have lower proportionality constants, which result in much improved interpolation accuracies from the field-aligned meshes. More significantly, the numerical tests with varying toroidal resolutions showed that the field-aligned meshes provide great benefits in representing the ballooning structures of plasma turbulence. An order smaller grid resolutions were enough to achieve a similar level of interpolation accuracies.

Although the field-aligned mesh can enhance accuracy and temporal stability conditions, it requires a significant amount of computational resources due to the increased number of sub-triangles compared to the primitive mesh. One direct solution to address this issue is optimizing the sub-triangulation to minimize the number of sub-triangles, which will be investigated in our future work. Additionally, it is also worth considering using GPU acceleration.

It should be noted that the field-aligned meshes and related numerical schemes are designed in a flexible way to choose the degree of field-aligning, which is controlled by the blending function in (13). As the field-aligned meshes are mainly for turbulent eddies in the plasma core, while it is important to properly model the geometric aspects of equilibrium magnetic field lines crossing material surfaces, the blending function is designed to achieve the field-line following discretization of regions with certain distances from material boundaries. This numerical technique can provide flexibility in choosing proper discretization schemes for both core regions and near material boundaries.

It was also shown that the newly developed schemes can ensure good conservation properties. As expected from the physics of gyrokinetic plasma, the conservations of mass, canonical angular momentum, and kinetic energy were checked with varying simulation conditions. It was found that the mass is conserved up to the machine accuracy as it should be guaranteed by the discontinuous Galerkin scheme. The conservations of other quantities were found to be improved by increasing toroidal resolution.

Finally, the parallelization efficiency of the new schemes was tested on a CPU-based supercomputer. It was found that the new schemes allow a reasonable parallelization performance up to the number of CPU cores around 6000. However, it showed degraded results for the CPU number beyond 10000. Certainly, there should be more efforts to increase the

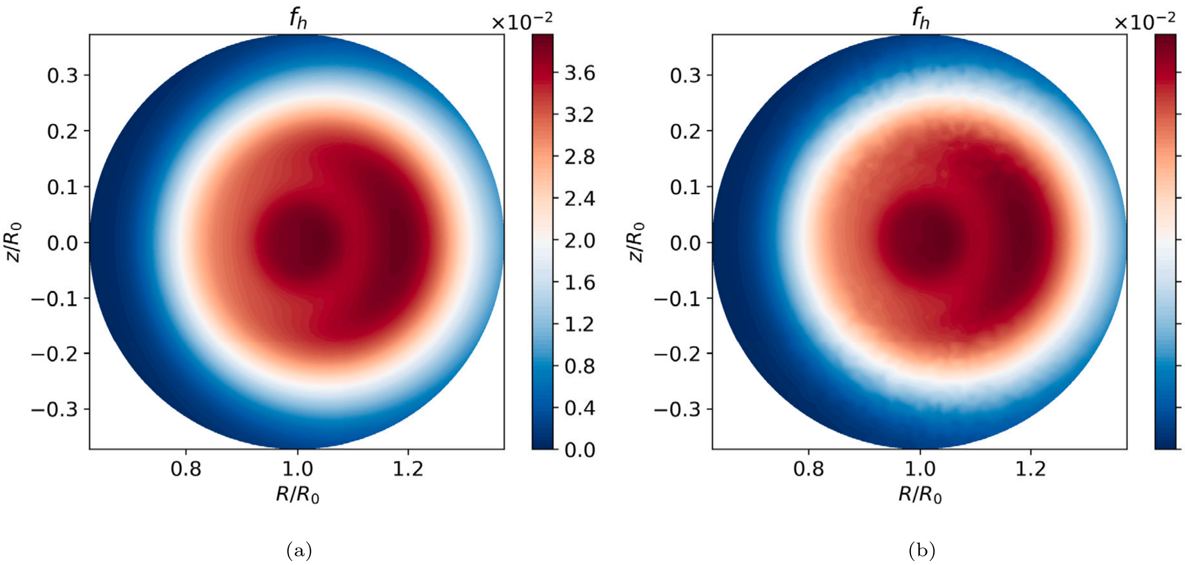


Fig. 23. Poloidal contour plots of (a) initial and (b) final ($t = 2R_0/v_{\text{th}}$) solutions at $(v_\parallel/v_{\text{th}}, u/v_{\text{th}}) = (1, 1/2)$.

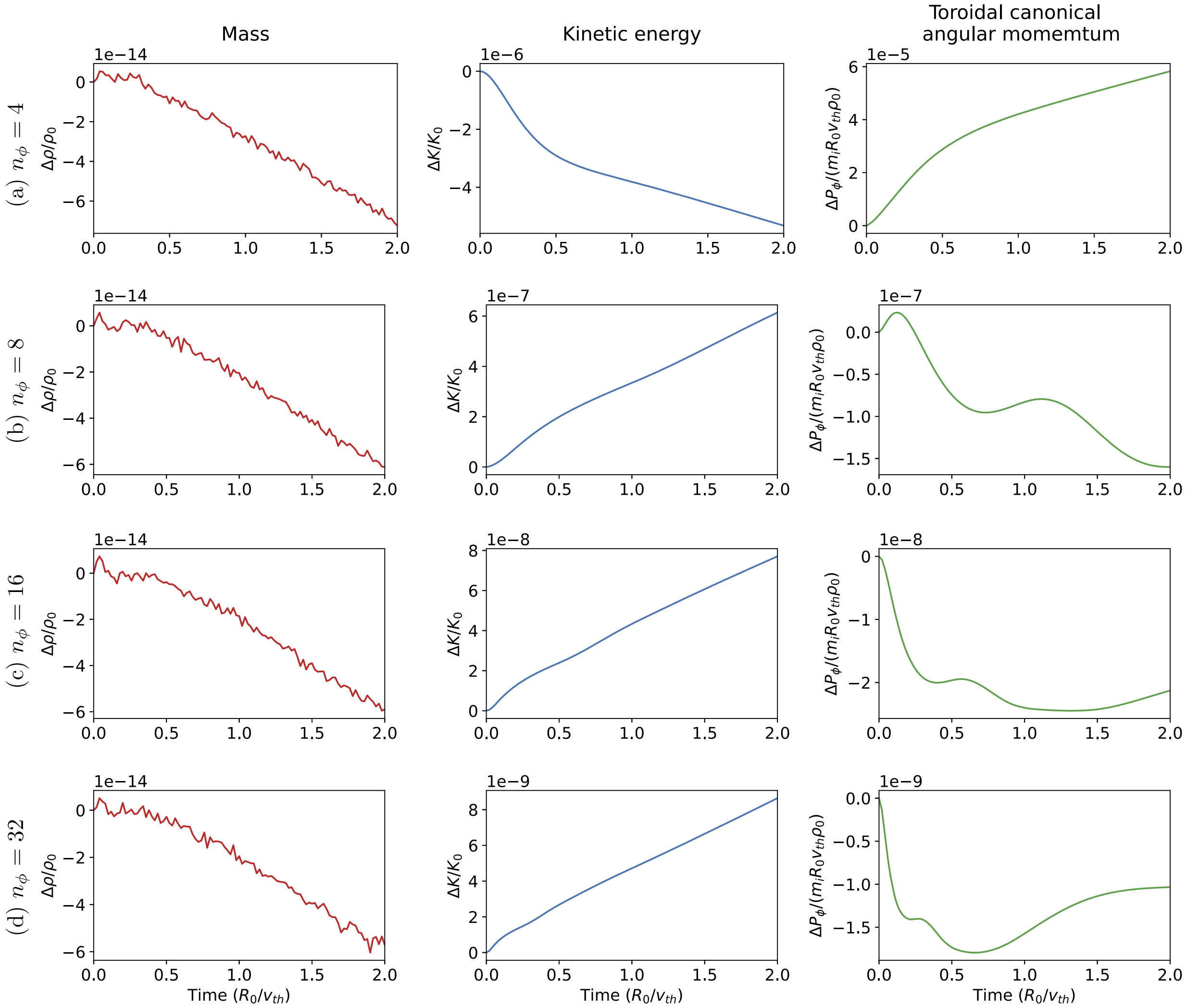


Fig. 24. Conservation properties on field-aligned meshes using quadratic basis ($N_\phi = 4, 8, 16, 32$).

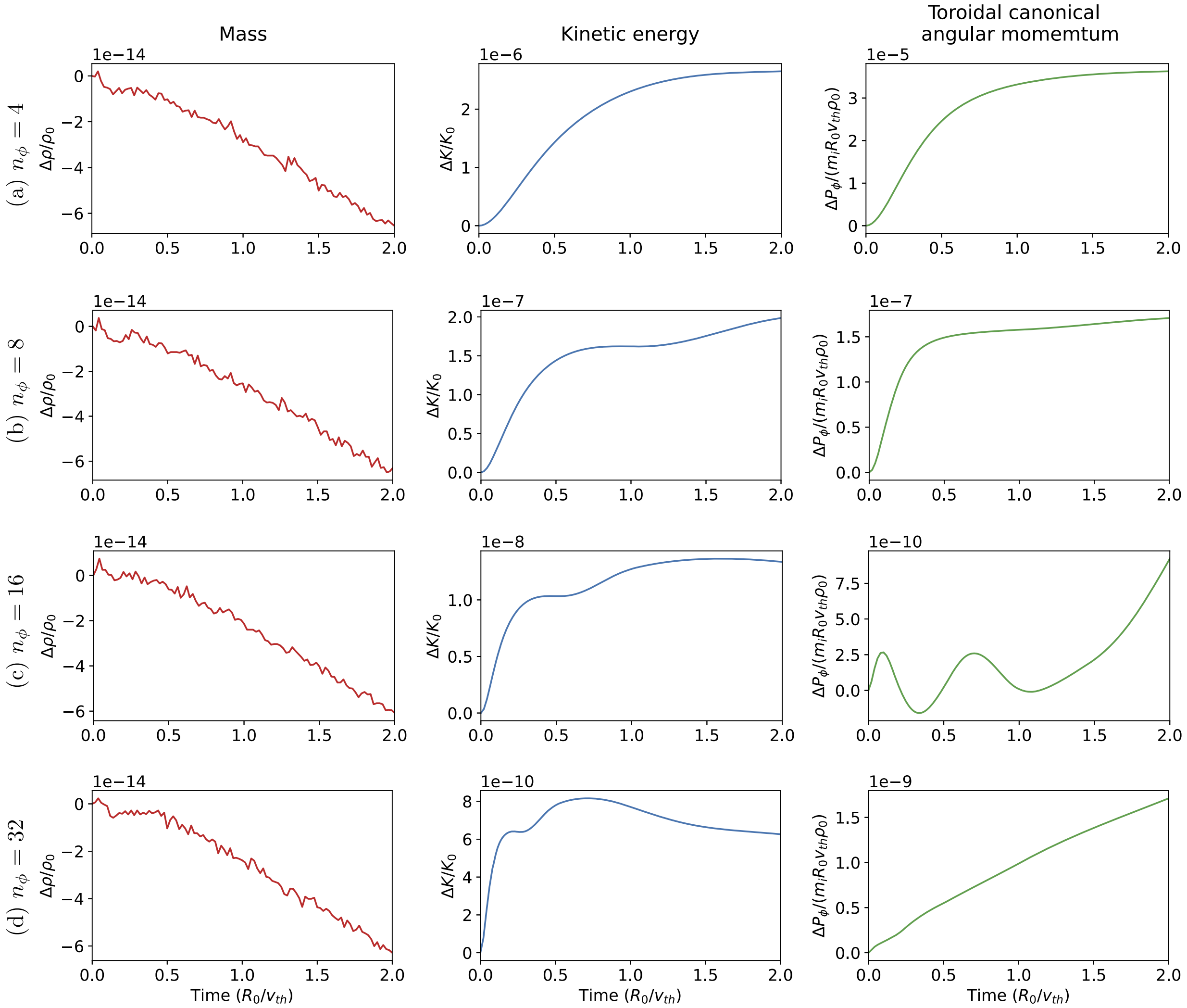
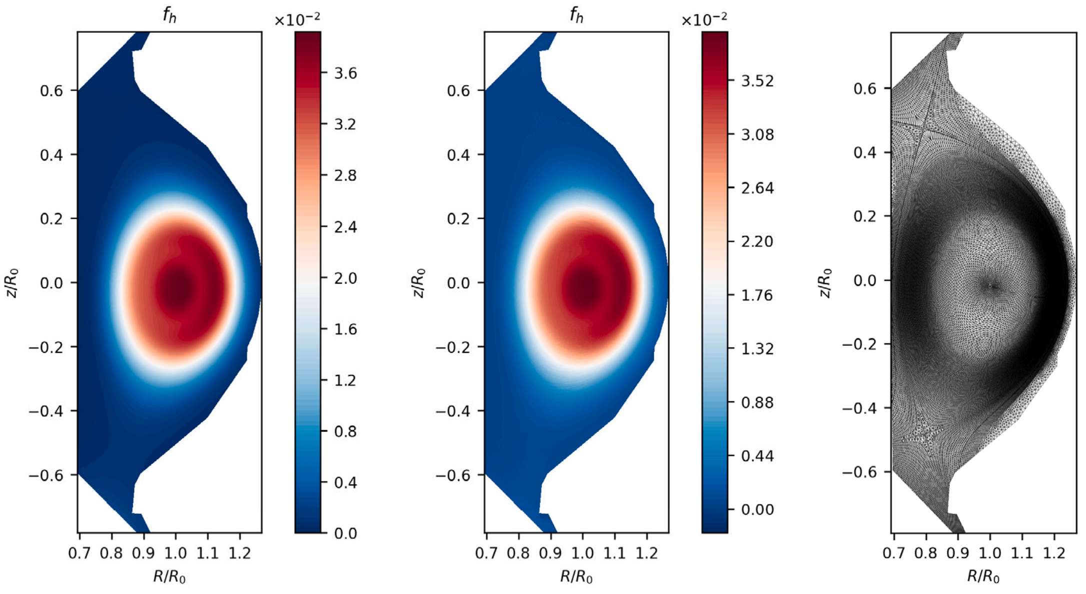


Fig. 25. Conservation properties on primitive meshes using quadratic basis ($N_\phi = 4, 8, 16, 32$).



(a)　(b)　(c)

Fig. 26. (a) Initial and (b) final solutions on the KSTAR domain using (c) the poloidal mesh.

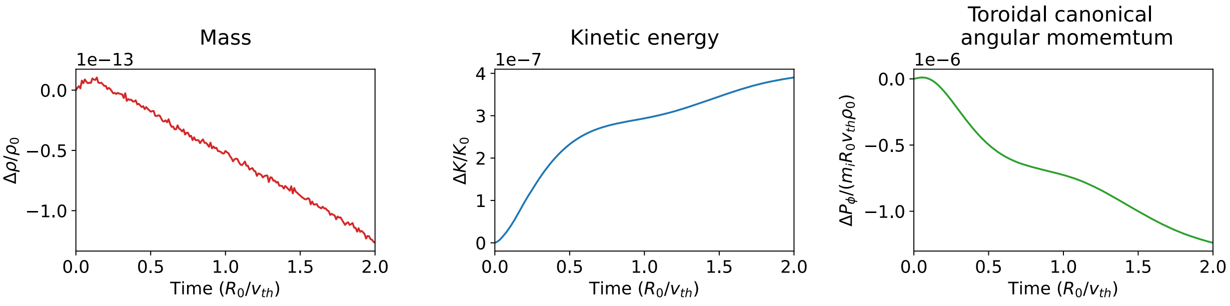
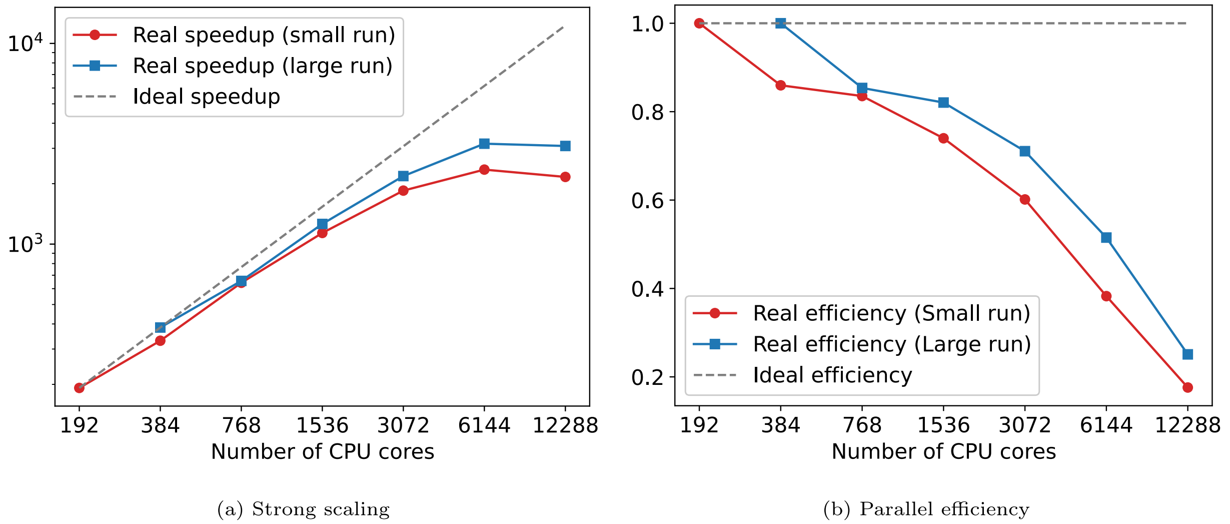


Fig. 27. Conservation properties of KSTAR domain.



(a) Strong scaling

(b) Parallel efficiency

Fig. 28. Parallelization performance test results.

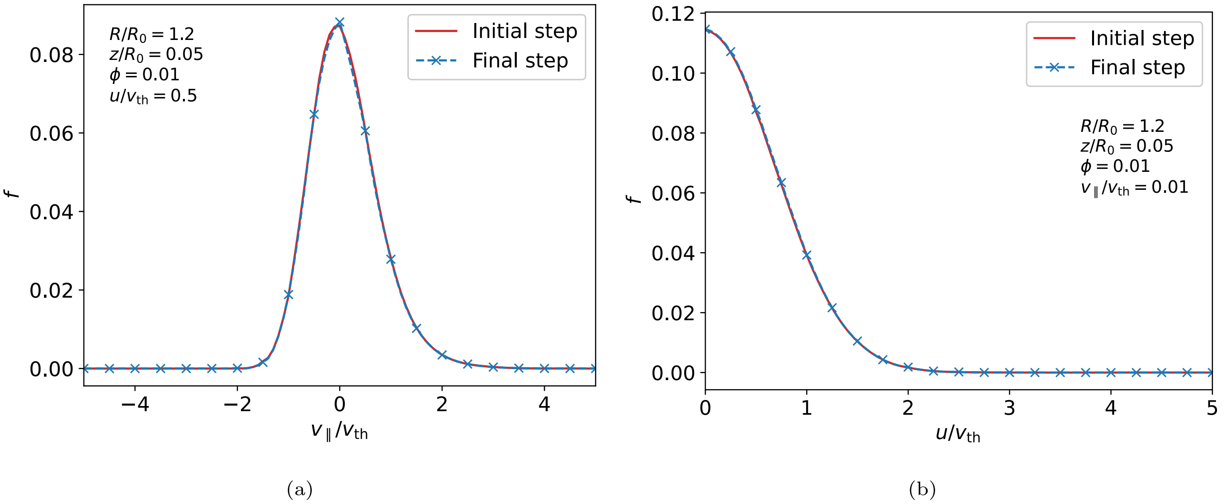


Fig. 29. Cross-sections of the canonical Maxwellian at the initial and final steps along the (a) $v_\parallel$-axis and (b) u-axis.

parallelization efficiency. However, it should be noted that the present results are from only CPU parallelization. Considering various numerical costs for nonconforming meshes and the high dimensional nature of the numerical fluxes, there is much room for improvement from additional GPU parallelization, which is left as future work. We want to emphasize that this work is a key step toward our final goal, i.e., the first principle whole device turbulence modeling of fusion devices based on the discontinuous Galerkin method. A key remaining challenge is the implementation of gyrokinetic field solvers, i.e., the Poisson and Ampere solvers, for time-evolving electromagnetic fluctuations. It is expected that the good numerical properties of the newly developed schemes will provide a firm basis to tackle the remaining numerical and computational challenges for the final goal.

CRediT authorship contribution statement

Gahyung Jo: Writing – original draft, Validation, Software, Methodology, Investigation. **Janghoon Seo:** Methodology, Conceptualization, Validation, Writing – original draft. **Jae-Min Kwon:** Conceptualization,

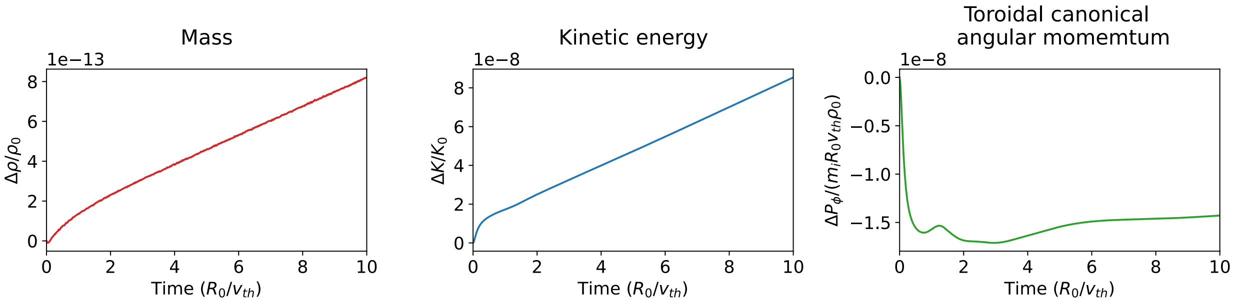


Fig. 30. Conservation properties of the temporal stationary test.

Methodology, Validation, Writing – original draft. **Eisung Yoon:** Writing – review & editing, Software, Investigation, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors wish to thank Juhyung Kim for the fruitful discussions. This research was supported by R&D Program of High Performance Fusion Simulation R&D (No. EN2541) through the Korea Institute of Fusion Energy (KFE) funded by the Government funds, Republic of Korea; and the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. RS-2023-00254695). Computing resources were provided on the KFE computer, KAIROS, funded by the Ministry of Science and ICT of the Republic of Korea (No. EN2541).

Data availability

Data will be made available on request.

References

[1] S. Jolliet, A. Bottino, P. Angelino, R. Hatzky, T. Tran, B. Mcmillan, O. Sauter, K. Appert, Y. Idomura, L. Villard, A global collisionless PIC code in magnetic coordinates, Comput. Phys. Commun. 177 (5) (2007) 409–425.

[2] S. Ku, C. Chang, P. Diamond, Full-f gyrokinetic particle simulation of centrally heated global ITG turbulence from magnetic axis to edge pedestal top in a realistic tokamak geometry, Nucl. Fusion 49 (11) (2009) 115021.

[3] S. De, T. Singh, A. Kuley, J. Bao, Z. Lin, G.Y. Sun, S. Sharma, A. Sen, Kinetic particle simulations in a global toroidal geometry, Phys. Plasmas 26 (8) (2019) 082507.

[4] M. Fivaz, S. Brunner, G. de Ridder, O. Sauter, T. Tran, J. Vaclavik, L. Villard, K. Appert, Finite element approach to global gyrokinetic particle-in-cell simulations using magnetic coordinates, Comput. Phys. Commun. 111 (1) (1998) 27–47.

[5] Y. Idomura, S. Tokuda, Y. Kishimoto, Global gyrokinetic simulation of ion temperature gradient driven turbulence in plasmas using a canonical Maxwellian distribution, Nucl. Fusion 43 (4) (2003) 234.

[6] J. Kwon, S. Yi, T. Rhee, P. Diamond, K. Miki, T. Hahm, J. Kim, O. Gürcan, C. McDevitt, Analysis of symmetry breaking mechanisms and the role of turbulence self-regulation in intrinsic rotation, Nucl. Fusion 52 (1) (2011) 013004.

[7] T. Dannert, Gyrokinetische simulation von plasmaturbulenz mit gefangenen teilchen und elektromagnetischen effekten, Ph.D. thesis, Technische Universität München, 2005.

[8] K. Obrejan, K. Imadera, J.-Q. Li, Y. Kishimoto, Development of a global toroidal gyrokinetic Vlasov code with new real space field solver, J. Plasma Fusion Res. 10 (special issue 2) (2015) 3403042.

[9] T.D. Rognlien, K. Bodi, R.H. Cohen, A.M. Dimits, M.A. Dorf, M.R. Dorr, S.I. Krasheninnikov, S.K. Nam, D.D. Ryutov, M.V. Umansky, X.Q. Xu, Advances in understanding tokamak edge/scrape-off layer transport, in: Report Number: LLNL-CONF-457153, 2010.

[10] D. Kim, J. Seo, G. Jo, J.-M. Kwon, E. Yoon, Nonlinear Fokker-Planck collision operator in Rosenbluth form for gyrokinetic simulations using discontinuous Galerkin method, Comput. Phys. Commun. 279 (2022) 108459.

[11] F. Hariri, M. Ottaviani, A flux-coordinate independent field-aligned approach to plasma turbulence simulations, Comput. Phys. Commun. 184 (11) (2013) 2419–2429.

[12] F. Schwander, E. Serre, H. Bufferand, G. Ciraolo, P. Ghendrih, Global fluid simulations of edge plasma turbulence in tokamaks: a review, Comput. Fluids 270 (2024) 106141.

[13] G. Giorgiani, S. Fernández-Méndez, A. Huerta, Hybridizable discontinuous Galerkin with degree adaptivity for the incompressible Navier–Stokes equations, Comput. Fluids 98 (2014) 196–208, 12th USNCCM mini-symposium of High-Order Methods for Computational Fluid Dynamics, a special issue dedicated to the 80th birthday of Professor Antony Jameson.

[14] M. Scotto d'Abusco, G. Giorgiani, J. Artaud, H. Bufferand, G. Ciraolo, P. Ghendrih, E. Serre, P. Tamain, Core-edge 2d fluid modeling of full tokamak discharge with varying magnetic equilibrium: from West start-up to ramp-down, Nucl. Fusion 62 (8) (2022) 086002.

[15] R. Heath, I. Gamba, P. Morrison, C. Michler, A discontinuous Galerkin method for the Vlasov–Poisson system, J. Comput. Phys. 231 (4) (2012) 1140–1174.

[16] N.R. Mandell, A. Hakim, G.W. Hammett, M. Francisquez, Electromagnetic full- gyrokinetics in the tokamak edge with discontinuous Galerkin methods, J. Plasma Phys. 86 (1) (2020) 905860109.

[17] A.J. Brizard, T.S. Hahm, Foundations of nonlinear gyrokinetic theory, Rev. Mod. Phys. 79 (2) (2007) 421–468.

[18] Y. Idomura, H. Urano, N. Aiba, S. Tokuda, Study of ion turbulent transport and profile formations using global gyrokinetic full-f Vlasov simulation, Nucl. Fusion 49 (6) (2009) 065029.

[19] J.-M. Kwon, D. Yi, X. Piao, P. Kim, Development of semi-Lagrangian gyrokinetic code for full-f turbulence simulation in general tokamak geometry, J. Comput. Phys. 283 (2015) 518–540.

[20] P.G. Ciarlet, The Finite Element Method for Elliptic Problems, Society for Industrial and Applied Mathematics, 2002.

[21] D.N. Arnold, G. Awanou, The serendipity family of finite elements, Found. Comput. Math. 11 (3) (2011) 337–344.

[22] J. Juno, A. Hakim, J. TenBarge, E. Shi, W. Dorland, Discontinuous Galerkin algorithms for fully kinetic plasmas, J. Comput. Phys. 353 (2018) 110–147.

[23] B. Cockburn, S. Hou, C.-W. Shu, The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV: the multidimensional case, Math. Comput. 54 (190) (1990) 545.

[24] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (3) (2001) 173–261.

[25] V. Dolejší, M. Feistauer, Discontinuous Galerkin Method, Analysis and Applications to Compressible Flow, 2015.

[26] G. Guennebaud, B. Jacob, et al., Eigen v3, http://eigen.tuxfamily.org, 2010.

[27] G. Jo, J.-M. Kwon, J. Seo, E. Yoon, Development of a gyrokinetic hyperbolic solver based on discontinuous Galerkin method in tokamak geometry, Comput. Phys. Commun. 273 (2022) 108265.

[28] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[29] E.G. Boman, U.V. Catalyurek, C. Chevalier, K.D. Devine, The Zoltan and Isorropia parallel toolkits for combinatorial scientific computing: partitioning, ordering, and coloring, Sci. Program. 20 (2) (2012) 129–150.

[30] D.A. Ibanez, E.S. Seol, C.W. Smith, M.S. Shephard, Pumi: parallel unstructured mesh infrastructure, ACM Trans. Math. Softw. 42 (3) (May 2016).

[31] W. Taitano, L. Chacón, A. Simakov, An adaptive, implicit, conservative, 1d-2v multi-species Vlasov–Fokker–Planck multi-scale solver in planar geometry, J. Comput. Phys. 365 (2018) 173–205.