

Improving conservation properties in a 5D gyrokinetic semi-Lagrangian code

Guillaume Latu, Virginie Grandgirard, Jérémie Abiteboul, Nicolas Crouseilles, Guilhem Dif-Pradalier, Xavier Garbet, Philippe Ghendrih, Michel Mehrenberger, Yanick Sarazin, Eric Sonnendrücker

▶ **To cite this version:**

Guillaume Latu, Virginie Grandgirard, Jérémie Abiteboul, Nicolas Crouseilles, Guilhem Dif-Pradalier, et al.. Improving conservation properties in a 5D gyrokinetic semi-Lagrangian code. The European Physical Journal D : Atomic, molecular, optical and plasma physics, 2014, 68 (11), pp.345. 10.1140/epjd/e2014-50209-1 . hal-00966162

HAL Id: hal-00966162
https://inria.hal.science/hal-00966162

Submitted on 26 Mar 2014

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

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

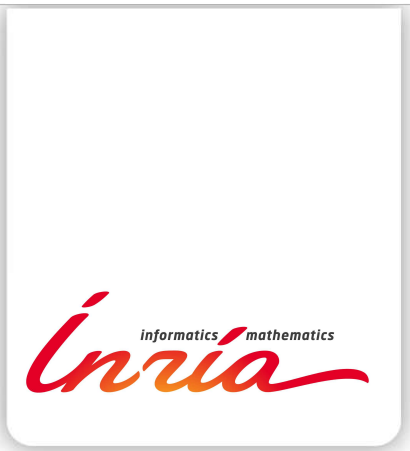


Improving conservation properties in a 5D gyrokinetic semi-Lagrangian code

G. Latu, V. Grandgirard, J. Abiteboul, N. Crouseilles,
G. Dif-Pradalier, X. Garbet, P. Ghendrih, M. Mehrenberger,
Y. Sarazin, E. Sonnendrücker

RESEARCH
REPORT
N° 8507
March 2014
Project-Teams IPSO

ISSN 0249-6399 ISRN INRIA/RR--8507--FR+ENG



Improving conservation properties in a 5D gyrokinetic semi-Lagrangian code

G. Latu*, V. Grandgirard*, J. Abiteboul†, N. Crouseilles‡,
G. Dif-Pradalier*, X. Garbet*, P. Ghendrih*,
M. Mehrenberger†, Y. Sarazin*, E. Sonnendrücker†

Project-Teams IPSO

Research Report n° 8507 — March 2014 — 20 pages

Abstract: In gyrokinetic turbulent simulations, the knowledge of some stationary states can help reducing numerical artifacts. Considering long-term simulations, the qualities of the Vlasov solver and of the radial boundary conditions have an impact on the conservation properties. In order to improve mass and energy conservation mainly, the following methods are investigated: fix the radial boundary conditions on a stationary state, use a 4D advection operator that avoids a directional splitting, interpolate with a delta-f approach. The combination of these techniques in the semi-Lagrangian code GYSELA leads to a net improvement of the conservation properties in 5D simulations.

Key-words: gyrokinetic model, conservation laws, semi-Lagrangian

* CEA Cadarache, IRFM bat. 513, 13108 Saint-Paul-les-Durance
† IPP, Garching, Germany
‡ INRIA IPSO & IRMAR, 263 av Général Leclerc, 35042 Rennes

RESEARCH CENTRE
RENNES – BRETAGNE ATLANTIQUE

Campus universitaire de Beaulieu
35042 Rennes Cedex

Résumé : La connaissance de certains états stationnaires peut permettre de réduire certains artéfacts numériques dans des simulations turbulentes gyrocinétiques. Lors de simulations en temps long, les qualités du solveur de Vlasov, et des conditions aux bords radiales ont un impact sur la conservation de quantités physiques. Pour autoriser de bonnes propriétés de conservation de la masse et de l'énergie notamment, les méthodes suivantes ont été envisagées : des conditions aux bords radiales figées sur un état stationnaire, une advection 4D pour éviter un splitting directionnel, un schéma d'interpolation de type delta-f. La combinaison de ces techniques dans le code semi-Lagrangien GYSELA a permis une amélioration nette des propriétés de conservation dans des simulations 5D.

Mots-clés : modèle gyrocinétique, lois de conservation, semi-Lagrangien

1 Introduction

The confinement properties of tokamak plasmas are governed by the gyrokinetic equation for each species coupled to Maxwell's equations for the electromagnetic fields. To better explain the physics inside tokamaks, turbulent transport dynamics and level can be investigated with the so-called gyrokinetic global codes. In order to provide reliable physical results, the used schemes should be adapted to lower the noise induced by the numerical methods.

GYSELA is a global nonlinear electrostatic code which solves the gyrokinetic equations in a five dimension phase space with a semi-Lagrangian method. With the version of GYSELA that we focus on in this paper, one can model the Ion Temperature Gradient instability for one ion species with adiabatic electrons. The work described here addresses improvements that suppress numerical artifacts appearing during 5D simulations.

Inaccurate description of the gyrokinetic equilibrium can yield unphysical excitation of zonal flow oscillations [ABH$^+$06]. Moreover, as stated in [ITK03, ABH$^+$06, DPGS$^+$08, IIK$^+$08], it is important to define the initial condition using a relevant gyrokinetic equilibrium, especially in the context of collisionless full-f simulations and long-term simulations. However, at long simulation times, irrespective of the choice for the initial state (local or canonical maxwellian), the turbulence robustly develops with identical statistical properties [DPGS$^+$08].

In the following, accuracy aspects are investigated for the Vlasov solver used in the GYSELA code. If proper care is not taken for both the Vlasov solver and the gyrokinetic initial equilibrium, one can observe that some conservation properties are not satisfied, for example total mass or energy.

The gyrokinetic framework for this study is introduced in Section 2, and also adaptations we have done on the radial boundary conditions to avoid anormal particle gains and losses. In Section 3, several new features are presented: an operator splitting (linear versus non-linear terms), a more accurate computation of particle displacement fields that are part of the semi-Lagrangian scheme, a 4D interpolation scheme, a delta-f interpolation technique. Then, Section 4 and 5 show numerical and experimental investigations to understand what kind of numerical instabilities we have met and how the solutions we propose fix these problems.

We acknowledge financial support from the Agence Nationale de la Recherche through ANR GYPSI contract and from the the G8-Exascale action NuFUSE. Some of the computations and simulations presented in this paper have been achieved using the GENCI resources (Turing machine), PRACE resources (Curie machine), IFERC resources (Helios machine).

2 Description of the context

2.1 Gyrokinetic Vlasov equation

The coordinate systems we consider is as follows. The spatial coordinates consists in (r, θ) the polar coordinates in the poloidal plane (the origin for the radius r is the magnetic axis and θ is the angle), φ the angle in the toroidal direction. The velocity parallel to the magnetic field is $v_\parallel$. The magnetic moment $\mu = mv_\perp^2/(2B)$ is an adiabatic invariant with $v_\perp$ the velocity in the plane orthogonal to the magnetic field. The computational domain is defined on $r \in [r_{\min}, r_{\max}], \theta \in [0, 2\pi], \varphi \in [0, 2\pi], v_\parallel \in [v_{\min}, v_{\max}], \mu \in [\mu_{\min}, \mu_{\max}]$. Let $\mathbf{z} = (r, \theta, \varphi, v_\parallel, \mu)$ be a variable describing the 5D phase space. The distribution function of the guiding-center is $f(\mathbf{z})$. The gyrokinetic Vlasov equation reads:

$$\partial_t f + \frac{1}{B_\parallel^*} \nabla_{\mathbf{z}} \cdot \left(\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} B_\parallel^* f \right) = 0$$

The time evolution of the gyro-center coordinates $(\vec{x}, v_\parallel, \mu)$ is given by the collision-less electrostatic gyrokinetic equations:

$$\frac{\mathrm{d}x^i}{\mathrm{d}t} = v_\parallel \vec{b}^* \cdot \vec{\nabla} x^i + \vec{v}_{Es_{GC}} \cdot \vec{\nabla} x^i + \vec{v}_{Ds} \cdot \vec{\nabla} x^i \tag{1}$$

$$m \frac{\mathrm{d}v_\parallel}{\mathrm{d}t} = -\mu \vec{b}^* \cdot \vec{\nabla} B - e \vec{b}^* \cdot \vec{\nabla} (J_0.\phi) + \frac{m v_\parallel}{B} \vec{v}_{Es_{GC}} \cdot \vec{\nabla} B \tag{2}$$

where x^i corresponds to the i-th covariant coordinate of $\vec{x}$, $\vec{B}$ is the magnetic field (notation B is the magnitude of $\vec{B}$), $\vec{J}$ stands for the plasma current density. Vacuum permittivity is

denoted μ_0. The $B^*_\parallel$ and $\vec{b}^*$ terms are defined as:

$$B^*_\parallel = B + \frac{mv_\parallel}{e\,B}\mu_0 \vec{b}\cdot\vec{J} \tag{3}$$

$$\vec{b}^* = \frac{\vec{B}}{B^*_\parallel} + \frac{mv_\parallel}{eB^*_\parallel}\,\frac{\mu_0\vec{J}}{B} \tag{4}$$

The advection terms are:

$$\vec{b}^*\cdot\vec{\nabla}x^i = b^{*i} = \frac{\vec{B}\cdot\vec{\nabla}x^i}{B^*_\parallel} + \frac{mv_\parallel}{eB^*_\parallel}\,\frac{\mu_0\vec{J}\cdot\vec{\nabla}x^i}{B}$$

$$\vec{v}_{Ds}\cdot\vec{\nabla}x^i = v^i_{Ds} = \left(\frac{mv^2_\parallel + \mu B}{eB^*_\parallel B}\right)[B, x^i]$$

$$\vec{v}_{Es_{GC}}\cdot\vec{\nabla}x^i = v^i_{Es_{GC}} = \frac{1}{B^*_\parallel}[J_0.\phi, x^i]$$

$$\vec{v}_{Es_{GC}}\cdot\vec{\nabla}B = -\frac{1}{B^*_\parallel}[B, J_0.\phi]\ .$$

The Poisson bracket is defined by $[F,G] = \vec{b}\cdot(\vec{\nabla}F\times\vec{\nabla}G)$. The term $v_{E_{GC}}$ represents the electric $E\times B$ drift velocity of the gyro-centers and v_D the curvature drift velocity. The Jacobian in phase space is $\mathcal{J}_{\mathbf{x}}.\mathcal{J}_{\mathbf{v}}$ with $\mathcal{J}_{\mathbf{x}}$ the jacobian in configuration space and $\mathcal{J}_{\mathbf{v}} = 2\pi\, B^*_{||}(r,\theta,v_{||})/m$ the jacobian in velocity space.

Some references concerning the framework we use to solve the gyrokinetic equations can be found in [GIVW10, GBB+06, GSG+08, BCG+11, SGA+11, AGG+11, ZGS+12]. Practically, we take $r_{\min} > 0$ to avoid the singular point $r_{\min} = 0$, and $r_{\max} \leq 1$ (because r is normalized to the minor radius of the torus). Let us remark that the set of equations we just described is a subset of what is accessible in the GYSELA code, especially no collision operator and no sources terms are taken it account in this paper.

2.2 Quasi-neutrality equation

2.2.1 Description of QN equation

In an electrostatic code, the field solver reduces to the numerical solving of a Poisson-like equation [Hah88]. In tokamak configurations, the plasma quasi-neutrality (denoted QN) approximation is currently made [GBB+06]. Electron inertia is ignored, which means that an adiabatic response of electrons is assumed. We define the operator $\nabla_\perp = (\partial_r, \frac{1}{r}\partial_\theta)$. We note n_0 the equilibrium density, B_0 the magnetic field at the magnetic axis and $T_e(r)$ the electronic temperature. We have also $B(r,\theta)$ the magnetic field, J_0 the Bessel function of first order and $k_\perp$ the transverse component of the wave vector. Hence, the QN equation can be written in dimensionless variables

$$-\frac{1}{n_0(r)}\nabla_\perp\cdot\left[\frac{n_0(r)}{B_0}\nabla_\perp\phi(r,\theta,\varphi)\right] + \frac{1}{T_e(r)}\left[\phi(r,\theta,\varphi) - \langle\,\phi\,\rangle(r)\right] = \rho_i(r,\theta,\varphi) \tag{5}$$

where ρ_i is defined by

$$\rho_i(r,\theta,\varphi) = \frac{1}{n_0(r)}\int\int \mathcal{J}_{\mathbf{v}}\, J_0(k_\perp\sqrt{2\mu})(f - f_{\rm ref})(r,\theta,\varphi,v_\parallel,\mu)\,dv_\parallel\,d\mu. \tag{6}$$

with $\langle\ .\ \rangle(r)$ the average on the flux surface labelled by r, and $f_{\rm ref}$ representing a reference distribution function. By assumption $\phi = 0$ for this distribution function $f = f_{\rm ref}$. Let us formaly define what are the flux surface operator (denoted $\langle\, g\,\rangle$) and the (θ,φ)-average operator (denoted $\bar{g}$) applied on a given function g

$$\bar{g}\,(r) = \frac{1}{4\pi^2}\int\int g\,\mathrm{d}\theta\,\mathrm{d}\varphi, \quad \langle\, g\,\rangle(r) = \tfrac{1}{4\pi^2\int \mathcal{J}_{\mathbf{x}}(r,\theta)\,\mathrm{d}\theta}\int\int \mathcal{J}_{\mathbf{x}}(r,\theta)\, g\,\mathrm{d}\theta\,\mathrm{d}\varphi\ . \tag{7}$$

Within the GYSELA setting we use in this work, we assume the following conservation property at any time step

$$\int\int\int \mathcal{J}_{\mathbf{x}}(r,\theta)\, n_0(r)\, \rho_i(r,\theta,\varphi)\, dr\,\mathrm{d}\theta\,\mathrm{d}\varphi = 0. \tag{8}$$

Let us remark that the variables ϕ, f, ρ_i depend also on time t. The function $f_{\rm ref}$ is fixed at startup very close to the initial distribution function $f^{t=0}$, and $f_{\rm ref}$ do not change over time.

2.2.2 Description of the QN solver

The equation (5) can be written as

$$\mathcal{P}\phi + \frac{1}{T_e}\left[\phi - \langle\,\phi\,\rangle\right] = \rho_i \tag{9}$$

where $\mathcal{P}$ is defined as

$$\mathcal{P} = -\frac{1}{n_0(r)}\nabla_\perp \cdot (n_0(r)\nabla_\perp) = -\left\{\frac{\partial^2}{\partial r^2} + \left[\frac{1}{r} + \frac{1}{n_0(r)}\frac{\mathrm{d}n_0(r)}{\mathrm{d}r}\right]\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial\theta^2}\right\}$$

By applying the (θ,φ)-average operator to the previous equation (9) and by using the fact that $\overline{\langle\,\phi\,\rangle} = \langle\,\phi\,\rangle$ then:

$$\mathcal{M}\bar{\phi} + \frac{1}{T_e}\left[\bar{\phi} - \langle\,\phi\,\rangle\right] = \bar{\rho}_i \tag{10}$$

where

$$\mathcal{M} = -\left\{\frac{\partial^2}{\partial r^2} + \left[\frac{1}{r} + \frac{1}{n_0(r)}\frac{\mathrm{d}n_0(r)}{\mathrm{d}r}\right]\frac{\partial}{\partial r}\right\}$$

One has the relation $\mathcal{M}\bar{\phi} = \mathcal{P}\bar{\phi}$. Let Υ be $\Upsilon = \phi - \bar{\phi}$ then, by subtracting (10) to (9), and by using appropriate boundary conditions (we will discuss in next subsection what kind of boundary conditions we use), we obtain an equation on $\Upsilon(r,\theta,\varphi)$ ($\forall\theta \in [0,2\pi]$ and $\forall\varphi \in [0,2\pi]$):

$$\begin{cases} \left(\mathcal{P} + \frac{1}{T_e}\right)\Upsilon(r,\theta,\varphi) = \varrho_i(r,\theta,\varphi) \quad \text{with} \quad \varrho_i = \rho_i - \bar{\rho}_i \qquad \forall r \in [r_{\min}, r_{\max}] \\ \Upsilon(r_{\min},\theta,\varphi), \Upsilon(r_{\max},\theta,\varphi) \text{ are given by boundary conditions.} \end{cases} \tag{11}$$

First, the unknown Υ can be solved in this equation without knowing $\langle\,\phi\,\rangle$ and $\bar{\phi}$. Second, to have access to the main unknown ϕ, we would like now to compute $\langle\,\phi\,\rangle$ and $\bar{\phi}$ (because $\phi = \Upsilon + \bar{\phi}$). The equation (10) can be rewritten as

$$\mathcal{P}\left(\bar{\phi} - \langle\,\phi\,\rangle\right) + \mathcal{P}\langle\,\phi\,\rangle + \frac{1}{T_e}\left[\bar{\phi} - \langle\,\phi\,\rangle\right] = \bar{\rho}_i \tag{12}$$

Then, using the fact that $\langle\,\Upsilon\,\rangle = \langle\,\phi\,\rangle - \langle\,\bar{\phi}\,\rangle = \langle\,\phi\,\rangle - \bar{\phi}$, the previous equation leads to the following system (omitting boundary condition issues):

$$\mathcal{P}\langle\,\phi\,\rangle = \bar{\rho}_i + \left(\mathcal{P} + \frac{1}{T_e}\right)\langle\,\Upsilon\,\rangle \tag{13}$$

We solve this equation (13) to get $\langle\phi\rangle$. Finally, we obtain the expression of the electric potential ϕ as[1]:

$$\phi(r,\theta,\varphi) = \Upsilon(r,\theta,\varphi) + \overbrace{\langle\,\phi\,\rangle(r) - \langle\,\Upsilon\,\rangle(r)}^{\bar{\phi}(r)} \tag{14}$$

To summarize, the solving of the equation (5) can be replaced by the solving of two simpler equations (11) and (13). First, we solve (11) in Υ. The variable φ plays the role of a parameter, we can solve a set of 2D Poisson-like equations (in the poloidal plane). Each 2D problem reduces to a projection in Fourier space in θ direction and then finite differencing in the radial direction. Second, equation (13) is treated as a differential equation only depending on the radial direction. Third, a sum is performed with Eq. (14) to get ϕ.

2.2.3 New boundary conditions

A first set of boundary conditions (denoted BC$_1$) is Dirichlet at $r_{\min}$ and $r_{\max}$. In (11), we impose $\Upsilon(r_{\min},\theta,\varphi) = 0$ and $\Upsilon(r_{\max},\theta,\varphi) = 0$ ($\forall\theta \in [0,2\pi]$, $\forall\varphi \in [0,2\pi]$). Concerning (13), we set $\langle\,\phi\,\rangle(r_{\min}) = \langle\,\phi\,\rangle(r_{\max}) = 0$ (we have then also $\bar{\phi}(r_{\min}) = \bar{\phi}(r_{\max}) = 0$). These conditions are easy to set up but there is one major drawback, the same potential (flux averaged) is forced at $r_{\min}$ and $r_{\max}$ which does not allow for the system to freely set a global radial gradient for $\langle\,\phi\,\rangle$. We will examplify soon why it is a problem.

A second set of boundary conditions (denoted BC$_2$) alleviates the constraint on the radial gradient of $\langle\,\phi\,\rangle$. As previously, we impose $\Upsilon(r_{\min},\theta,\varphi) = 0$ and $\Upsilon(r_{\max},\theta,\varphi) = 0$ ($\forall\theta \in [0,2\pi]$, $\forall\varphi \in [0,2\pi]$). Although this assumption simplifies the solver, the impact on the solution has

[1]Remark: In slab geometry (old versions of GYSELA), we used to suppose $\langle\,\phi\,\rangle = \bar{\phi}$, $\langle\,\Upsilon\,\rangle = 0$.

not been yet evaluated. Then, we fix a Neumann condition at $r_{\min}$ and Dirichlet at $r_{\max}$ for Eq. (13): $\frac{\partial}{\partial r}\langle\,\phi\,\rangle(r_{\min}) = 0$ and $\langle\,\phi\,\rangle(r_{\max}) = 0$. Let us assume that $r_{\min}$ is small enough (*i.e* near 0) and therefore that $\mathcal{J}_{\mathbf{x}}(r_{\min}, \theta)$ does not depend on θ. In this configuration (verified in practice) for r small enough, $\bar{\phi}(r) \approx \langle\phi\rangle(r)$ thanks to Eq. (7). We end up with some interesting properties:

$$\begin{cases} \partial_r\bar{\phi}(r_{\min}) \approx \partial_r\langle\,\phi\,\rangle(r_{\min}) = 0 \\ \bar{\phi}(r_{\max}) = \langle\,\phi\,\rangle(r_{\max}) = 0 \end{cases}$$

To show the impact of boundary conditions on a simulation, let us consider a simulation in which ITG turbulence has grown up. We consider the distribution function and associated ρ_i at one given time step. In Fig. (1), we plot the $\bar{\phi}(r)$ function using the two different boundary conditions for the same right-hand side ρ_i (close to $r_{\min}$ and $r_{\max}$, ρ_i function is zero, whereas in the center of the radial domain where the turbulence is located, ρ_i is non zero). For BC_1, the radial derivative of $\bar{\phi}(r)$ is non-zero at low r, that gives a net poloidal flow near $r_{\min}$. However, in BC_2 configuration $\bar{\phi}(r)$ is nearly constant near $r_{\min}$, then the spurious flow along θ direction disappears.

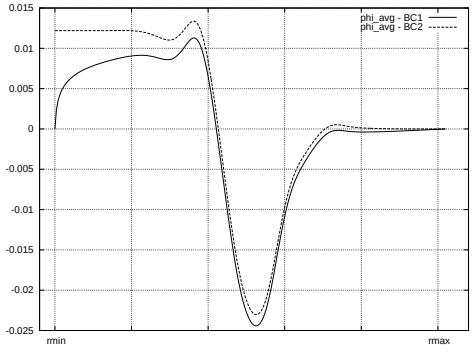


Figure 1: $\bar{\phi}(r)$ profiles for a simulation that models ITG at a given time step (tubulence has already grown up) using BC_1 and BC_2 boundary conditions

2.3 Defining proper radial boundaries

We have already described Vlasov and quasi-neutrality solvers that are two main components of the GYSELA code. Let us notice the computational domain is not formally closed in the radial and velocity directions. We will describe which changes have been done to prevent particles from escaping the computational domain at radial boundaries.

2.3.1 Description of the problem

The situation with eddies, turbulence located near the radial boundaries $r_{\min}$ or $r_{\max}$ is difficult to handle. It commonly leads to electric potential structures that generate fluxes of particles inside or outside the computational domain. Taking this into account in the mass and energy balance is tricky. Also, the derivatives of ϕ have to be evaluated at $r_{\min}$ and $r_{\max}$ for computing displacements of particles, it is complex to get them accurately. Furthermore, considering non-Dirichlet, non-Neumann boundary conditions in the QN and Vlasov equations is a difficult task.

Thus, we have retained the following practical solution: to impose the distribution function equal to a reference function in the vicinity of radial boundaries. The reference distribution function that we consider is invariant by Vlasov, and also it leads to null electric potential when solving the QN equation. It follows that simple radial boundary conditions for Vlasov and QN solvers are accessible, for example Dirichlet and Neumann are both possible. Also, as ρ_i (see Eq. (6)) will then be zero in the vicinity of $r_{\min}$ and $r_{\max}$, the ϕ potential is likely to be nearly constant in this area. Then, the influx/outflux of particles due to electric potential through radial boundaries of the computational domain should be null (not because of physics phenomena, but due to boundary conditions). This solution improves the setup of both Vlasov and QN solver and helps to close the computational domain in radial direction. Herafter, we describe a method that one can use to force the distribution function near the radial boundaries.

2.3.2 Adapting the Vlasov solver at radial boundaries

Let us decompose f as the sum of an equilibrium function f_{eq} and a perturbation δ_f:

$$f = f_{eq} + \delta_f$$

We will see afterwards how to build such a f_{eq} function. We define a radial mask function $\mathcal{H}(r)$

$$\begin{cases} \mathcal{H}(r_{\min}) = \mathcal{H}(r_{\max}) = 0 \\ \forall r \in [r_{\min}, r_{\max}], 0 \leq \mathcal{H}(r) \leq 1 \end{cases}$$

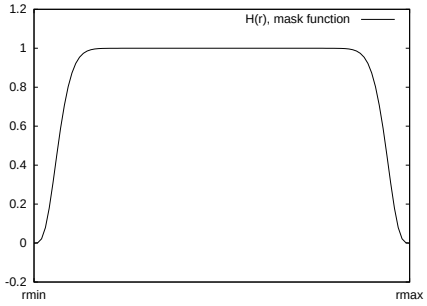


Figure 2: The $\mathcal{H}$ mask function

The mask function is smooth, continuous and equal to 1 almost everywhere except in the vicinity of radial boundaries. Near $r_{\min}$ and $r_{\max}$, the function $\mathcal{H}$ smoothly drops to zero. With the distribution funtion f, one can define:

$$\begin{cases} \delta_f = f - f_{eq} \\ f_\dagger = f_{eq} + \mathcal{H}\,\delta_f \\ \delta_{f_\dagger} = (1 - \mathcal{H})\,\delta_f \end{cases} \tag{15}$$

With this formulation, we have the property: $f = f_\dagger + \delta_{f_\dagger}$. The main benefit is that: function $f_\dagger$ is equal to f_{eq} at radial boundaries, and $f_\dagger$ is equal to f in the center of the radial domain. The improvement of radial boundary conditions consists in using $f_\dagger$ in the Vlasov solver instead of f (because $f_\dagger$ is equal to f_{eq} at $r_{\min}$ and $r_{\max}$).

Practically, the method consists in removing $\delta_{f_\dagger}$ from f to get $f_\dagger$ before the Vlasov step, and then solving Vlasov on $f_\dagger$ only. As $\delta_{f_\dagger}$ function contains a $(1-\mathcal{H})$ factor, $\delta_{f_\dagger}$ is zero almost everywhere except in the vicinity of $r_{\min}$ and $r_{\max}$. The $\delta_{f_\dagger}$ function is designed to contain a relative small set of particles that are likely to escape the computational domain. We discard these particles from the Vlasov solver on purpose. After the Vlasov step, we add back $\delta_{f_\dagger}$ to $f_\dagger$ to recover f. According to this procedure, $\delta_{f_\dagger}$ does not evolve in time through Vlasov solver, we can interpret this quantity as a reservoir of particles that has left the computational domain. Please note that the reference f_{eq} of Eq. (15) can be changed occasionally during a simulation in order to fit the macroscopic evolution of parameters such as temperature/density profiles. This approach closes the computational domain in radial direction, as the particles are trapped into $\delta_{f_\dagger}$ when they approach the radial boundary limits. Then, mass conservation is obtained, the main drawback concerns the physics model that is altered near $r_{\min}$ and $r_{\max}$.

2.3.3 Adapting the QN solver at radial boundaries

For the quasi-neutrality solver, we would like to avoid large derivatives of the potential ϕ in the vicinity of radial boundaries. To achieve this goal, we are considering the previous decomposition $f = f_\dagger + \delta_{f_\dagger}$. The $f_\dagger$ function is built in order to be equal to f_{eq} near $r_{\min}$ and $r_{\max}$. Then, we would like to use $f_\dagger$ and discard the $\delta_{f_\dagger}$ contribution in the RHS of the QN equation. However, we have also to take into account the $\delta_{f_\dagger}$ part in order that the total mass is unchanged. As we want to consider $\delta_{f_\dagger}$ as something that does not participate anymore in the dynamics of the system we transform this quantity in a passive mass in the RHS of QN equation. To do that, we just have to rewrite Eq. 6 in the following way:

$$\rho_i(r,\theta,\varphi) = c_i + \frac{1}{n_0(r)} \int \mathcal{J}_{\mathbf{v}} d\mu \int dv_\parallel J_0\,(f_\dagger - f_{eq}), \tag{16}$$

with

$$c_i = \frac{\int \mathcal{J}_{\mathbf{v}}\,\mathcal{J}_{\mathbf{x}}\,J_0(f_{eq} - f_\dagger)\,dv_\parallel\,d\mu\,dr\,d\theta\,d\varphi}{\int\,\mathcal{J}_{\mathbf{x}}\,n_0(r)\,dr\,d\theta\,d\varphi}$$

The term c_i is built to recover mass conservation defined in Eq. (8) (see also [MSM+13] for a focus on this problem in a reduced setting). These changes alter locally the electric

potential, near $r_{\min}$ and $r_{\max}$. Nevertheless, the energy balance is preserved, conservation of total energy remains true. In practice, the term c_i is expected to be relatively small[2], because it represents the small fraction of the particles that has left the computational domain through radial boundaries.

2.3.4 Numerical results

Conservation issues in a toroidal 4D simulation Let us consider a simulation with a single value $\mu \neq 0$. Due to magnetic curvature/gradient, the drift velocities at large $v_{\|}$ transport some turbulent eddies straight to $r_{\min}$ and $r_{\max}$. In this configuration, we will look at the impact of the solutions we have just proposed for boundary conditions, versus the original code setting. The test case and initial conditions will be given in the upcoming section 6.1.

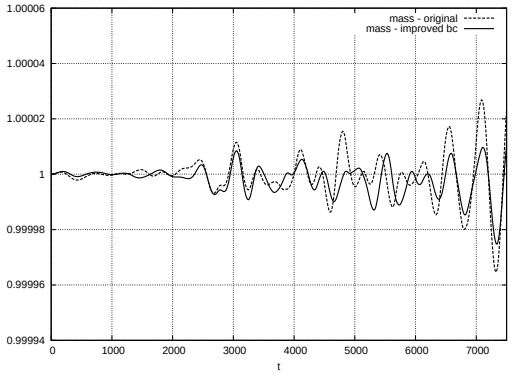


Figure 3: Evolution of the mass for a 4D toroidal test case, $\mu = 3$. The mass is normalized to 1 at $t = 0$.

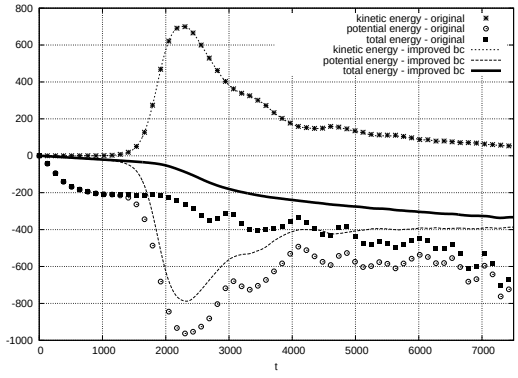


Figure 4: Evolution of the energies for a 4D toroidal test case, $\mu = 3$.

The specific treatment of radial boundary conditions improves the time evolution of energy conservation (as shown in Fig. 4). The energy curves report the relative energy: the energy at time step t minus the energy at time step 0. Ideally, the total energy should remain 0 throughout the simulation. The curve denoted "`total energy - improved bc`" is closer to zero than the original one denoted "`total energy - original`". In the early time steps (from 0 up to $2000\,\Omega^{-1}$), the total energy conservation is improved significantly: the total energy in the original version (black squares) rapidly departs from 0, whereas with the new boundary conditions the total energy (black thick curve) remains closer to zero. We will see afterwards that other reasons explain why total energy is not well conserved after $t = 2000\,\Omega^{-1}$.

The impact on mass conservation is however negligible (see Fig. 3), meaning that mainly particles with large velocity modulus are concerned by the new boundary conditions (they represent a small percentage of the total mass, but quite a significant part of kinetic energy).

[2]One can also consider to set c_i to zero, then the conservation of Eq. (8) is no more assured but the possible impact of c_i on the radial profile of ϕ is decreased (Eq. (10)).

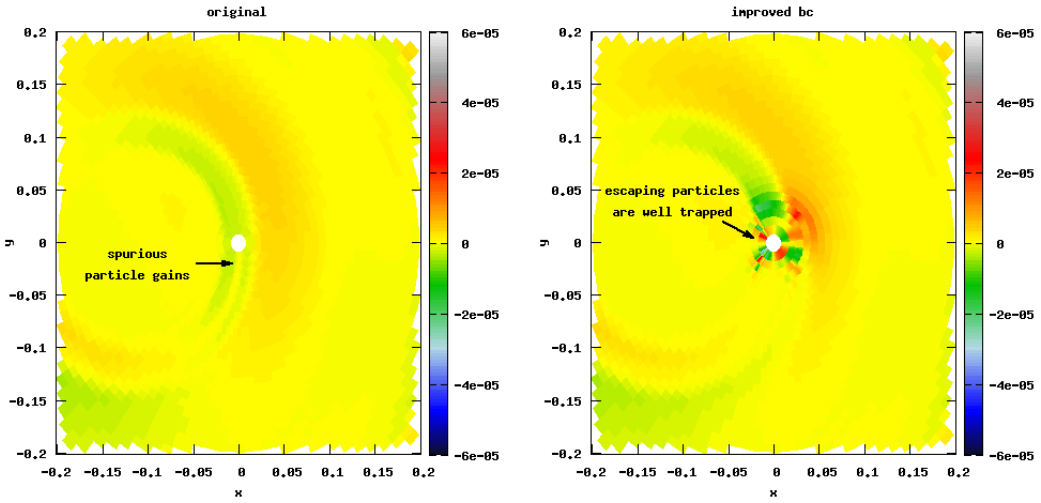


Figure 5: Poloidal cross section (in x, y coordinates) of δ_f with a zoom on the center of the domain, location in phase space : $v_\parallel = -5\, v_{th0},\ \varphi = 0,\ \mu = 3,\ t = 64\,\Omega^{-1}$. Left plot: original code, right plot: improved boundary conditions.

Distribution function cut in a toroidal 4D simulation We can now have a closer look at the distribution function itself. In Fig. 5, the poloidal cross section of $\delta_f = f - f_{eq}$ is shown at a specific location in phace space ($v_\parallel = -5\, v_{th0},\ \varphi = 0,\ t = 64\,\Omega^{-1}$, zoom on the center of poloidal cross section). A structure characterized by negative values of δ_f goes through the center of the poloidal cross section (top-down flow). On the left hand side, the original code has an external boundary in the center of the grid (white disk) that absorbs some of these negative values and a kind of tail is generated (this is a wrong behaviour that leads to particle gains). On the right hand side, the improved boundary conditions imply that many particles are trapped before they reach the central disk. These trapped particles create fine holes and bumps (located all around the center). They are not taken into account in the Vlasov solver (because we use $\delta_{f_\dagger}$ for Vlasov instead of δ_f), but they are kept in the distribution function to compute macroscopic values (such as kinetic energy, mass, ...). Hence, these particles do not flow out the computation domain which is a good property. This process of collecting particles is not due to the physics model, but an artificial mechanism that we add to avoid numerical issues (it alters locally the physics).

3 New numerical schemes

A set of new features are exposed in this Section: an operator splitting (linear versus non-linear terms), the precomputation of some displacements that are part of the semi-Lagrangian scheme, the 4D interpolation scheme, the *delta-f interpolation* technique.

We have designed a 4D advection strategy. The objectives of this approach were twofold:

- to evaluate pros and cons compared to the directional Strang splitting that is used in the current version of the GYSELA code,
- to combine this 4D advection with a *delta-f* approach, described in Section 3.5.

3.1 Global separation of linear/nonlinear terms

The equations (1) and (2) can be split into two parts, using the same kind of procedure as described in [IIK+08]. The first part includes the nonlinear terms that depend on the electric potential. The second part comprises all other terms. One can solve these two parts separately. On the first hand, the nonlinear operator is described by Eqs (19), (20) , on the second hand, the linear operator is presented in Eqs (17), (18).

Linear operator $\mathcal{L}$

$$\frac{\mathrm{d}x^i}{\mathrm{d}t} = v_\parallel \vec{b}^* \cdot \vec{\nabla} x^i + \vec{v}_{Ds} \cdot \vec{\nabla} x^i \qquad (17)$$

$$m\frac{\mathrm{d}v_\parallel}{\mathrm{d}t} = -\mu\, \vec{b}^* \cdot \vec{\nabla} B \qquad (18)$$

Nonlinear operator $\mathcal{N}$

$$\frac{\mathrm{d}x^i}{\mathrm{d}t} = \vec{v}_{Es_{GC}} \cdot \vec{\nabla} x^i \qquad (19)$$

$$m\frac{\mathrm{d}v_\parallel}{\mathrm{d}t} = -e\, \vec{b}^* \cdot \vec{\nabla}(J_0.\phi) + \frac{mv_\parallel}{B}\, \vec{v}_{Es_{GC}} \cdot \vec{\nabla} B \qquad (20)$$

The linear operator exhibits large displacements at large modulus of parallel velocity, and also induces shear flows. These features can interfere with the nonlinear dynamics that possibly involves small displacements that are not of the same order of magnitude. Moreover, as the dynamics generated by the two operators are different, the accuracy problems have possibly not the same characteristics for the two operators; and the limitations (CFL-like conditions) on the time step are also not the same. Ideally, one should be able to fix the numerical scheme and time step of linear and nonlinear operators independently in order to achieve a given accuracy.

The current semi-Lagrangian scheme implemented in GYSELA code uses Strang splitting in the Vlasov solver. This is a directional splitting with the following sequence $(\hat{v_\parallel}/2, \hat{\varphi}/2, \hat{r\theta}, \hat{\varphi}/2, \hat{v_\parallel}/2)$. The notation $\hat{v_\parallel}/2$ corresponds to the shift in $v_\parallel$ direction over $\Delta t/2$, $\hat{\varphi}/2$ the shift φ direction and $\hat{r\theta}$ the 2D shift in (r, θ) direction over Δt (detailed information can be found in [GBB+06]). But without a linear/nonlinear splitting, it may lead to some troubles at high parallel velocities. Indeed, during one single directional substep, some large shifts in φ, r and θ appear in the linear terms at high $|v_\parallel|$ (typically several spatial cells). Then, for not so large Δt, the evaluation of electric fields E that depends on spatial location is not done at the right spatial position at each substep of the directional splitting (except for the first substep). To correct this behaviour, a possibility is to take very small time step to recover small displacements in the linear operator and then a reasonable accuracy in the evaluation of E.

The splitting between the linear and nonlinear parts corrects this problem. The nonlinear operator is applied alone, thus the linear operator and its large shifts at high $v_\parallel$ modulus does not interact badly with the nonlinear solver. This approach is a little bit more expensive than the previous approach in term of computational cost. But it is counter balanced by the fact that one can take a larger time step Δt.

3.2 Precomputation of particle trajectories

The foot of a characteristic ending at a grid position $(r_i, \theta_j, \varphi_k, v_{\parallel\, l})$ is needed for the advection in the semi-Lagrangian method. Since the fields acting on particles for the linear part $\mathcal{L}$ do not depend on time t, one can approximate the foot of a particle trajectory - denoted $(r_i, \theta_j, \varphi_k, v_{\parallel\, l})^\star$ - once for all, for a given time step Δt. We have used Runge-Kutta time integration scheme RK2 with a small time step to precompute these particle trajectories. This approach is possible for the linear terms, but not for non-linear terms $\mathcal{N}$ that depend on E_t and ϕ_t. Let us choose a δt, such as $M\,\delta t = \Delta t$ with $M \in \mathbb{N}$ and M large enough. One can build a series as follows (using a α field deduced from Eqs. 17 and 18 at page 10):

$$\begin{pmatrix} r^{n+\frac{1}{2}} \\ \theta^{n+\frac{1}{2}} \\ \varphi^{n+\frac{1}{2}} \\ v_\parallel^{n+\frac{1}{2}} \end{pmatrix} = \begin{pmatrix} r^n \\ \theta^n \\ \varphi^n \\ v_\parallel^n \end{pmatrix} + \frac{\delta t}{2}\,\alpha \begin{pmatrix} r^n \\ \theta^n \\ \varphi^n \\ v_\parallel^n \end{pmatrix}, \text{ and } \begin{pmatrix} r^{n+1} \\ \theta^{n+1} \\ \varphi^{n+1} \\ v_\parallel^{n+1} \end{pmatrix} = \begin{pmatrix} r^n \\ \theta^n \\ \varphi^n \\ v_\parallel^n \end{pmatrix} + \delta t\,\alpha \begin{pmatrix} r^{n+\frac{1}{2}} \\ \theta^{n+\frac{1}{2}} \\ \varphi^{n+\frac{1}{2}} \\ v_\parallel^{n+\frac{1}{2}} \end{pmatrix}$$

The initial condition is set to $r^0 = r_i$, $\theta^0 = \theta_j$, $\varphi^0 = \varphi_k$, $v_\parallel^0 = v_{\parallel\, l}$. After M steps of Runge-Kutta iterations, it gives

$$\begin{pmatrix} r_i \\ \theta_j \\ \varphi_k \\ v_{\parallel\, l} \end{pmatrix}^\star = \begin{pmatrix} r^M \\ \theta^M \\ \varphi^M \\ v_\parallel^M \end{pmatrix}$$

Because these trajectories can be computed only once when the simulation starts, M can be taken quite large (we will assume $M = 64$ in the following). These precomputations do not impact significantly the global simulation time. We just need to store the results of these precomputations in memory and on the parallel file system (storing this information into files helps for the checkpoint/restart strategy). Other time integration schemes have been tried: RK3, RK4, and also larger values of M, no impact was seen in term of accuracy.

One of the main benefit of this method is: the feet of the characteristics are determined more

accurately than the Taylor expansion of α that is used in the current version of GYSELA, because the time step δt is a much smaller time step than Δt.

3.3 Interpolations by tensor product in 4D

In this Subsection, we will describe a strategy that performs 4D interpolations using tensor product of cubic B-splines. Let us consider a one-dimensional function g which is defined on a global domain $[x_{\min}, x_{\max}] \subset \mathbb{R}$. Suppose that we know the values $(g(x_i))_{\forall i=[0,N]}$ and we want to interpolate this discretized g function with cubic splines. The projection s of g onto the cubic splines basis reads

$$g(x) \simeq s(x) = \sum_{\nu=-1}^{N+1} \eta_\nu \mathcal{B}_\nu(x),$$

where $\mathcal{B}_\nu$ is the cubic B-spline. The interpolating spline s is uniquely determined by $(N+1)$ interpolating conditions

$$g(x_i) = s(x_i), \quad \forall i = [0, N],$$

and by the boundary conditions on the first derivative at both ends of the interval in order to obtain a $\mathcal{C}^1$ global approximation (periodic boundary conditions or others are also possible but not described here)

$$f'(x_0) \simeq s'(x_0), \quad f'(x_N) \simeq s'(x_N).$$

The η_ν coefficients are the solution of a linear system. A LU decomposition is used to find these unknowns $(\eta_\nu)_{\nu\in[-1,N+1]}$ depending on the inputs $(g(x_i))_{i\in[0,N]}$. Practically, as $\mathcal{B}_\nu$ are compact support, the interpolation of g at a single location x is computed with the formula:

$$g(x) \simeq s(x) = \sum_{\nu=m^0-1}^{m^0+2} \eta_\nu \mathcal{B}_\nu(x), \text{ with } m^0 = \lfloor N \frac{x - x_{\min}}{x_{\max} - x_{\min}} \rfloor$$

Let us consider now a 4D function g that will be projected on s in cubic spline basis. Using a tensor product of cubic B-spline in 4D, one can construct spline coefficients $\eta_{\nu^0,\nu^1,\nu^2,\nu^3}$, given that ν^d are the variables of the 4 dimensions. These new η coefficients can be found by using LU decompositions (four LU decompositions, one for each dimension). The interpolations in four dimensions are computed using the following expression (with m^d well chosen depending on x^d and on the discretization):

$$s(x^0,x^1,x^2,x^3) = \sum_{\nu^0=m^0-1}^{m^0+2} \sum_{\nu^1=m^1-1}^{m^1+2} \sum_{\nu^2=m^2-1}^{m^2+2} \sum_{\nu^3=m^3-1}^{m^3+2} \eta_{\nu^0,\nu^1,\nu^2,\nu^3} \mathcal{B}_{\nu^0}(x^0)\, \mathcal{B}_{\nu^1}(x^1)\, \mathcal{B}_{\nu^2}(x^2)\, \mathcal{B}_{\nu^3}(x^3)$$

3.4 4D numerical scheme

The usual way to perform a single time step in the GYSELA code consists of a series of directional advections: $(\hat{v_\parallel}/2, \hat{\varphi}/2, \hat{r\theta}, \hat{\varphi}/2, \hat{v_\parallel}/2)$. This Strang-splitting converges in $O(\Delta t^2)$, it decomposes in 4 steps of 1D advections, and 1 central step of a 2D advection.

Let us now consider an avoidance of this Strang splitting. Let us suppose that we advance in time *only the linear part* $\mathcal{L}$ and the feet of the characteristics are computed for all grid points in phase space (presented in Subsection 3.2). The Algorithm 1 sketches the corresponding 4D numerical scheme. The question of the parallelization of the computations will not be addressed in this paper, even if it is an important issue for a production code.

Algorithm 1: 4D numerical scheme (for a given μ value)

```
Compute the splines coefficients η^n of the 4D function f^n(r, θ, φ, v_∥ );
for All grid points (r_i, θ_j, φ_k, v_∥ l) do
    Get the precomputed foot of characteristic (r_i, θ_j, φ_k, v_∥ l)^⋆;
    Interpolate f^n in 4D at this location (r_i, θ_j, φ_k, v_∥ l)^⋆;
    f^{n+1}(r_i, θ_j, φ_k, v_∥ l) ← the interpolated value;
end
```

Let us remark that the $\mathcal{N}$ operator uses the directional splitting method and not 4D advections. It would imply too large computational costs to estimate 4D displacement fields for operator $\mathcal{N}$ that depends on time.

3.5 Delta-f interpolation strategy

The delta-f interpolation strategy is one of the goals that is targeted to achieve the conservation of invariant/equilibrium states. The Algorithm 2 describes how we have used delta-f techniques to improve the interpolation accuracy.

Algorithm 2: 4D delta-f numerical scheme (for a given μ value)

1 If needed, update f_{eq};
2 Compute the splines coefficients η^n of the 4D function ${\delta_f}^n = f^n - f_{eq}$;
3 **for** *All grid points* $(r_i, \theta_j, \varphi_k, v_{\parallel l})$ **do**
4 | Get precomputed foot $(r_i, \theta_j, \varphi_k, v_{\parallel l})^\star$;
5 | Interpolate ${\delta_f}^n$ in 4D at location $(r_i, \theta_j, \varphi_k, v_{\parallel l})^\star$;
6 | $f^{n+1}(r_i, \theta_j, \varphi_k, v_{\parallel l}) \leftarrow$ interpolated ${\delta_\mathrm{f}}^\mathrm{n}$value $+ f_{eq}(r_i, \theta_j, \varphi_k, v_{\parallel l})$;
7 **end**

This algorithm requires that f_{eq} is invariant for the Vlasov equation. If f_{eq} verifies this property, the value of f_{eq} at the foot of any characteristic is equal to the value of a given grid points, *i.e* $f_{eq}(r_i, \theta_j, \varphi_k, v_{\parallel l})^\star = f_{eq}(r_i, \theta_j, \varphi_k, v_{\parallel l})$.

It is trivial to show that if $f^n = f_{eq}$ at any spatial location, then δ_f^n is zero, then this method gives $f_{n+1} = f_{eq}$. So, we are able to conserve perfectly an invariant/equilibrium state. Furthermore, if we suppose that f^n is near f_{eq}, δ_f^n is small. The first Algo. 1 will generate interpolation errors that grow along with the spatial derivatives of f_{eq}. The new Algo. 2 has smaller interpolation errors that are only proportional to derivatives of δ_f. Please note that one may have update the f_{eq} function if this function is too far from f (first line of the algorithm). However, in the simulations we consider in this paper, we don't need to perform this step, as f remains close to f_{eq}.

Remark: The same delta-f scheme could also have been combined with the Strang splitting we have usually. Nevertheless, as f_{eq} is not preserved at each directional substep of the splitting, it may need some refined techniques to have it working.

4 Gyrokinetic simulations - reduced settings

In this Section, we evaluate the new methods presented previously on several simple test cases.

4.1 Unperturbed motion of particles

We are interested in numerical methods that are able to conserve equilibrium states of the Vlasov equation. First, it is a property that ensures a good level of accuracy of the Vlasov solver in full-scale simulations, but, second, such invariant property is also important by itself in order to establish reference scenarii to test/verify the code. Generally speaking, conservation properties are crucial to distinguish correct execution from bad one for a scientific simulation code. In addition, this is helpful for debugging purposes.

We focus in this Section on gyrokinetic models with one single μ value and with $\mu \neq 0$ (*i.e.* non drift-kinetic models, the case with $\mu = 0$ is simpler and is not considered here [LGA+12]). Let us take the following hypothesis: considering the global separation of linear/nonlinear terms, we discard $\mathcal{N}$ and keep only $\mathcal{L}$. The Quasi-neutrality solver is also switched off. Therefore, we use no field solver, (equivalent to impose ϕ=0). In this configuration, we estimate essentially the quality of the initial distribution function and the quality of the Vlasov solver for the linear terms ($\mathcal{L}$ operator), as we will see afterwards.

4.2 Test case 1

Description of the test case Let us consider an initial distribution function that should be steady for Vlasov equation. An equilibrium solution of the collisionless gyrokinetic equation must satisfy some conditions. In an axisymmetric toroidal configuration, a gyrokinetic Vlasov equilibrium is defined by three constants of motion: the magnetic moment μ, the energy $\mathcal{E}$=$m\, v_\parallel^2/2 + \mu\, B(r,\theta)$ (assuming ϕ is null), and the canonical toroidal angular momentum P_φ=$e\,\psi(r) + m\, I v_\parallel / B(r,\theta)$ (where I is a constant used in the definition of $\vec{B} = \frac{I}{B}(\vec{e_\varphi} + \frac{r}{q(r) R_0}\vec{e_\theta})$). The $\psi(r)$ function is defined thanks to the safety factor $q(r)$ by the relation: $d\psi/dr = -B_0\, r/q(r)$.

Let us assume that we have set up an initial equilibrium. One can estimate the accuracy of this equilibrium and of the Vlasov solver by:

- measuring the difference between the initial function and the distribution function at a given time step after several Vlasov solving steps.
- a convergence study in time and space discretizations for a simulation that includes several Vlasov solving steps.

For the sake of simplicity and to reduce the computational costs, the simulations presented in this Section are set up to $N_\mu = 1$. Let us initialize a simulation using a steady state. For an axisymmetric equilibrium, the distribution function is constant along the φ direction. It should remain constant using the $\mathcal{L}$ operator. Our goal is to quantify the numerical error induced by the numerical scheme, knowing that the difference to the initial distribution function should remain zero. The test case setting is characterized by (with $\rho^\star$ the relative gyroradius): μ=3., $\rho^\star$=.01, N_r=256, N_θ=256, N_φ=16, $N_{v_\parallel}$=128. The initial distribution function $f_{init} = f_{eq}$ is taken as :

$$\begin{cases} \forall P_\varphi \in [-\infty, P_{\varphi 1}], & f_{eq}(\mathcal{E},\mu,P_\varphi) = 0 \\ \forall P_\varphi \in [P_{\varphi 2}, \infty], & f_{eq}(\mathcal{E},\mu,P_\varphi) = e^{-\mathcal{E}} \\ \forall P_\varphi \in [P_{\varphi 1}, P_{\varphi 2}], & f_{eq}(\mathcal{E},\mu,P_\varphi) = e^{-\mathcal{E}} \frac{1}{2}(1+cos(\frac{\pi (P_{\varphi 2} - P_\varphi)}{P_{\varphi 2} - P_{\varphi 1}})) \end{cases}$$

where P_{φ_1} and P_{φ_2} are well chosen in order to localize the large gradient of the distribution function in the middle the radial profile. We will look at the norms $u_p(t) = \|f^t - f^{t=0}\|_p$ with $p = 1, \infty$. The timing unit is the ion cyclotronic time Ω^{-1}. The thermal velocity is denoted v_{th}.

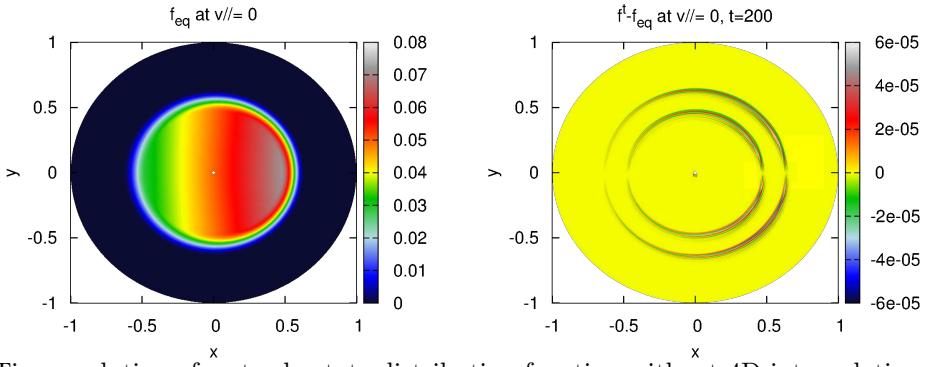


Figure 6: Time evolution of a steady state distribution function without 4D interpolation and without delta-f interpolation techniques, poloidal cut at $v_\parallel = 0, \varphi = 0$ (ideally $f^t - f_{eq}$ remains zero)

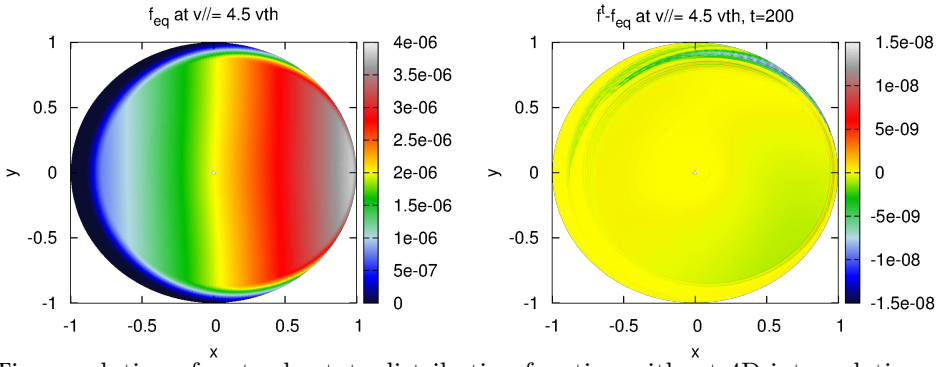


Figure 7: Time evolution of a steady state distribution function without 4D interpolation and without delta-f interpolation techniques, poloidal cut at $v_\parallel = 4.5\, v_{th}, \varphi = 0$ (ideally $f^t - f_{eq}$ remains zero)

Poloidal cuts are presented in Fig. 6, with ($v_\parallel$=0, φ=0 fixed) at t=0 and t=200. One can observe on this Figure that that numerical artifacts develop on f^t in a poloidal cut at $v_\parallel = 0$. These approximation errors come from the interpolation operator and from the computation of the feet of characteristics. These errors are small compared to the mean value of f^t, but errors can be located where f^t is close to zero (relative error is large).

On Fig. 7, a poloidal cut at $v_\parallel = 4.5\, v_{th}$ is presented. Let us consider energetic particles, some of them are able to encounter radial boundaries. This Figure shows that numerical

problems arise at external boundary (right plot, $t = 200$). The difficulty is the following: in the semi-Lagrangian scheme one looks for the position of one given particle back into the past and estimates the value of distribution function at this location. If this particle was outside the computational domain one time step ago, one has however to approximate the value of the distribution function with an ad-hoc procedure. As the equilibrium function f_{eq} is not defined outside the computational domain, our method is to stick escaping particles to the last radius $r = r_{\max}$. It is this procedure that induces the biggest numerical perturbation originating on the external boundary condition.

Using 4D advection technique combined with delta-f interpolation, the numerical artifacts shown in Fig. 6 and 7 do not appear at all. Indeed, the interpolation during the advection is performed on the distribution function $\delta_f^t = f^t - f_{eq}$, and δ_f^t is equal to zero at the beginning of the simulation. The δ_f^t function remains exactly zero over all time step. This approach solves the problem of preserving the invariant state f_{eq}.

Convergence in space The distribution function is not exactly conserved after some Vlasov solving steps using only $\mathcal{L}$ operator without delta-f interpolation scheme. Nevertheless, one can look at the convergence of the numerical scheme according to the spatial resolution. On Fig 8, we look at the temporal evolution of L^1 norm: $\|f - f_{eq}\|_1$.

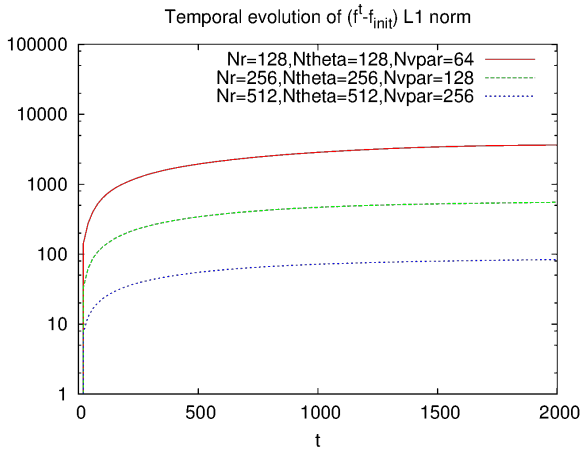


Figure 8: Time evolution of the L1 norm for a steady state distribution function (ideally $\|f - f_{eq}\|_1$ should remain zero)

Whenever the number of points in each dimension is multiplied by two, the error is divided by a factor *6.5*. This measurement corresponds roughly to the expected behaviour for the cubic B-spline we use for the interpolation.

4.3 Test case 2

Description of the test case The previous test case was focusing on preserving a steady state solution, but in practice the distribution function used in a realistic simulation can often be represented as the sum of a steady state solution plus a perturbation $f = f_{eq} + \delta_f$. There are multiple methods and several criteria to establish the best f_{eq} to use, we will not discuss this issue here. However, the following case will focus on the benefits brought by delta-f interpolation technique to represent the time evolution of a small perturbation over of a steady state solution of Vlasov. Let us define f_{init} as $f_{eq} + \delta_f^0$ with f_{eq} reads

$$\begin{cases} \forall P_\varphi \in [-\infty, P_{\varphi_1}], & f_{eq}(\mathcal{E}, \mu, P_\varphi) & = 0 \\ \forall P_\varphi \in [P_{\varphi_2}, \infty], & f_{eq}(\mathcal{E}, \mu, P_\varphi) & = e^{-\mathcal{E}} \\ \forall P_\varphi \in [P_{\varphi_1}, P_{\varphi_2}], & f_{eq}(\mathcal{E}, \mu, P_\varphi) & = e^{-\mathcal{E}} \frac{1}{2}(1 + cos(\frac{\pi (P_{\varphi_2} - P_\varphi)}{P_{\varphi_2} - P_{\varphi_1}})) \end{cases}$$

and $\delta_f^{t=0}$ is a smooth perturbation which is localized in the area of a single point $P^i = (\mathcal{E}^i, P_\varphi^i, \theta^i)$. The $\delta_f^{t=0}$ function does not depend on φ, this test case is axisymmetric and at any time step solution is identical for each φ value. On Fig. 9, the f_{eq} and $\delta_f^{t=0} = f^{t=0} - f_{eq}$ functions are plotted.

After a few time steps, the distribution function $\delta_f^{t=20}$ is shown in Fig. 10. In the left panel, the original version of the code is used, whereas in the right panel the 4D advections with delta-f

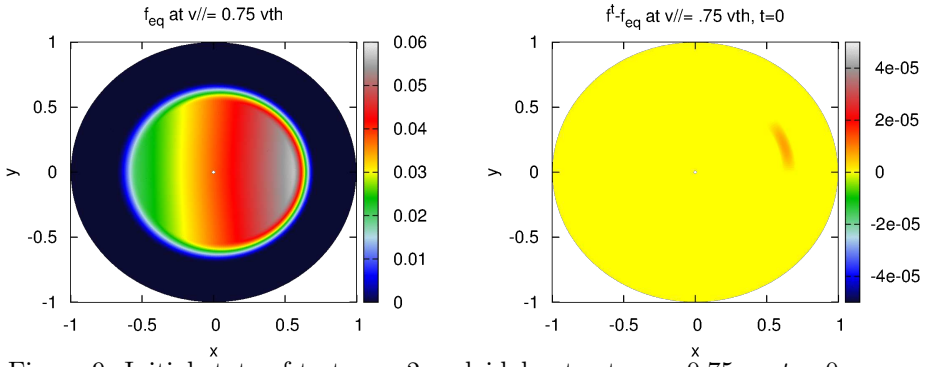


Figure 9: Initial state of test case 2, poloidal cuts at $v_{\|} = 0.75 v_{th}, t = 0$

interpolation schemes are switched on. Without the delta-f approach, artifacts spoil the signal: in the center at $r = r_{\min}$, and also in the regions where the cubic splines have difficulties fitting the f slope. With the delta-f interpolation, these problems disappear. There is not difficulty to handle feet of the characteristics outside the domain because it is assumed that one has $\delta_f = 0$ outside, which is a well known value. The artifacts due to the slope of f_{eq} do not arise because the interpolation operator acting on δ_f^t does not see this slope.

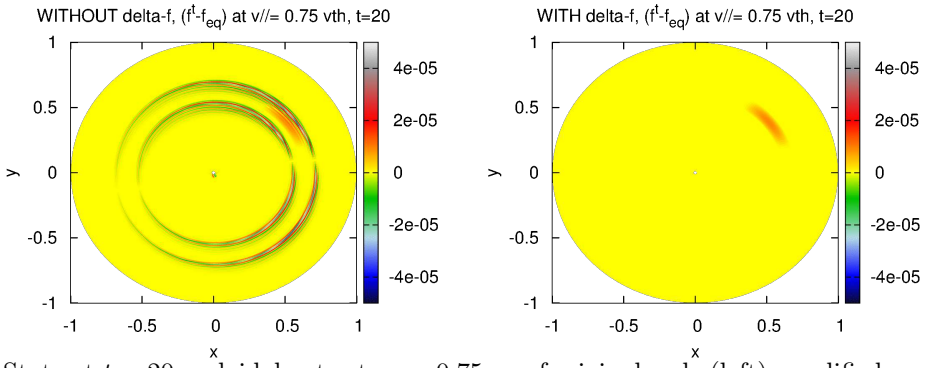


Figure 10: State at $t = 20$, poloidal cuts at $v_{\|} = 0.75 v_{th}$ of original code (left), modified version with delta-f interpolation (right)

To conclude this Section, the delta-f interpolation method tends to reduce the numerical noise. The main drawback is that we assume that: the distribution function is very near an equilibrium function at radial boundaries (δ_f almost zero). The efficiency of the method is increased if one can compute a steady state function f_{eq} close to the current distribution function.

5 Gyrokinetic simulations - Full-scale 4D settings

In the previous Section, the field solver of non-linear terms were switched off. In this Section, we consider a full-scale case including non-linear terms and quasi-neutrality solver, giving access to non-linear physics. The simulations presented earlier in Subsection 2.3.4 is the reference framework on top of which we will evaluate here a set of new methods. The improved boundary conditions is switched on in all the experiments presented in Sections 5 and 6.

5.1 Description of the setting

For the sake of simplicity, we will describe in this section only 4D experiments (with $\mu \neq 0$). The initial distribution function f_{init} has been chosen close to an equilibrium f_{eq} in order to maximize the benefit of the delta-f interpolation approach. The initial distribution function f_{init} is equal to f_{eq} plus a small perturbation (a bath of modes). The distribution f_{eq} is a function of the three motions invariants only, $P_\varphi, \mathcal{E}, \mu$. The f_{eq} is computed such that the radial profile of temperature $T_i(r)$ (averaged over θ, φ dimensions) and density $n_i(r)$ match the ones we prescribed in input. The radial gradients of these profiles are chosen in order that Ion Temperature Gradient (ITG) instability develops. The turbulence drive is ensured by thermal baths at the radial edges in $r_{\min}$ and$r_{\max}$ that are imposed during the whole simulation.

Some of the key parameters of the following simulation: μ=0.4, $\rho^\star$=.02, aspect_ratio=3, N_r= 128, N_θ=128, N_φ=64, $N_{v_\parallel}$=92 .

5.2 Impact of the Linear/Non-Linear splitting

Let us evaluate the improvements brought by all the techniques described in Sections 3.1, 3.2, 3.4. First, the new splitting introduces a clear separation between linear terms and non-linear terms, which improves quantitatively the time integration of particle trajectories. Second, the precomputation of particles displacements leads to an improved localization of the foot of characteristics in the semi-Lagrangian scheme, and then on the accuracy of the Vlasov solver. Third, the 4D advection avoids the possible discrepancy due to the directional splitting occuring at too large time steps.

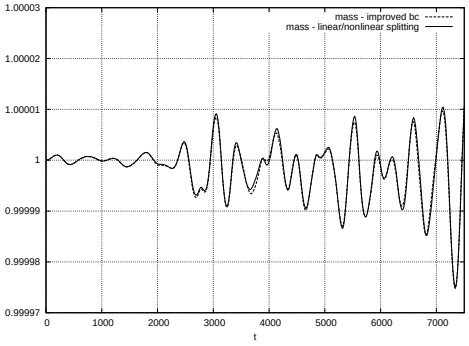


Figure 11: Evolution of the mass for a 4D toroidal test case, $\mu \neq 0$

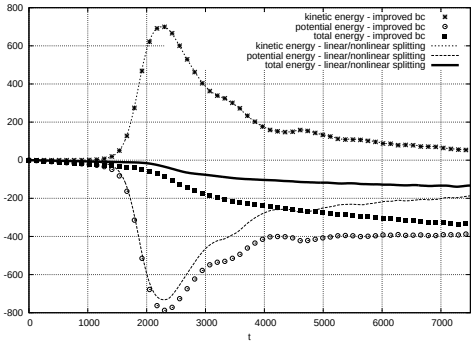


Figure 12: Evolution of the energies for a 4D toroidal test case, $\mu \neq 0$

The combination of these three techniques significantly improves the energy conservation property, as shown in Fig. 12. The total energy obtained with these techniques (black thick line) is closer to zero than the total energy previously observed (black plain squares). However, potential energy and mass curves (Fig. 11) are almost not modified. We deduce that these modifications essentially correct the particle trajectories of particles that have high $|v_\parallel|$.

5.3 Impact of the delta-f interpolation approach

We consider the delta-f interpolation in addition to the previous setting.

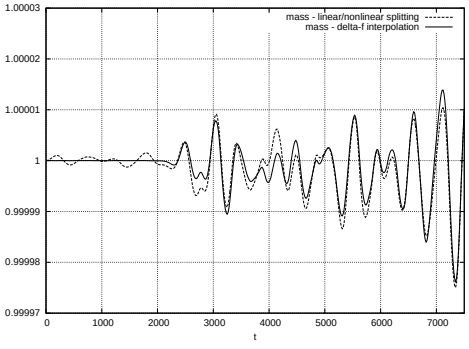


Figure 13: Evolution of the mass for a 4D toroidal test case, $\mu \neq 0$

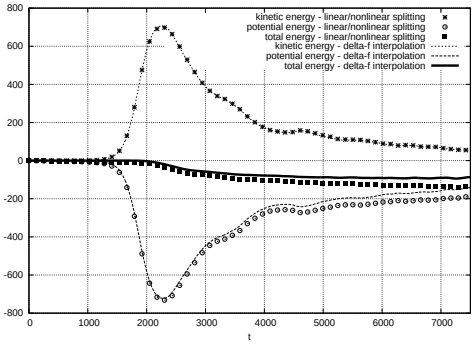


Figure 14: Evolution of the energies for a 4D toroidal test case, $\mu \neq 0$

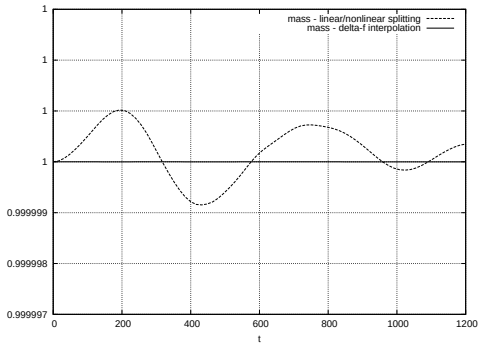


Figure 15: Zoom on the first time steps, mass for a 4D toroidal test case, $\mu \neq 0$

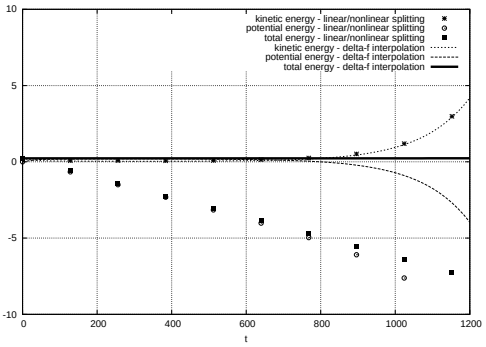


Figure 16: Zoom on the first time steps, energies for a 4D toroidal test case, $\mu \neq 0$

With delta-f version, the mass remains almost at the reference initial value on the period from $t = 0$ to $t = 2000$ (see Fig. 13 and a zoom on early time steps in Fig. 15). This is a quite desirable behaviour. However, once we reach the non-linear saturation phase ($t > 2000$), the mass conservation begins to degrade. Furthermore, the Fig. 16 shows a zoom on the energies during the beginning of the simulation. The total energy of delta-f interpolation version (black thick line) is a lot better than the previous simulation (black squares). Overall, in order to simulate fine phenomena that developp close to a steady state (for example $t < 1000$ here), the delta-f interpolation technique seems to be required.

Let us analyse more precisely the mass curves. Each of the operators $\mathcal{L}$ and $\mathcal{N}$ should theoretically conserve the mass independently. One can track during a simulation how much each operator degrades the mass. On specific simulations, we took a time step sufficiently low in order to have well converged simulations in time and space. In these simulations, we have observed:

- the operator $\mathcal{L}$ induces mass conservation errors (without delta-f strategy) in the beginning of the simulation before the saturation phase (meaning $t \leq 2000$ here). These errors are larger than those of operator $\mathcal{N}$. Using the delta-f approach, these errors caused by $\mathcal{L}$ are reduced by several orders of magnitude.

- the operator $\mathcal{N}$ is predominantly responsible for the lack of conservation of the mass in the saturation phase (meaning $t \geq 2000$ here).

6 Gyrokinetic simulations - Full-scale 5D settings

In this Section, we give some results about 5D gyrokinetic simulations which are more costly than 4D simulations.

6.1 Description of the setting

The initial distribution function f_{init} has been chosen as a function of the three motions invariants: $P_\varphi, \mathcal{E}, \mu$. The initial radial gradients are fixed in order to get Ion Temperature Gradient (ITG) instability. Here is a summary of the parameters of the simulation presented: $\rho^\star$=.013, aspect_ratio=3, N_r=256, N_θ=256, N_φ=64, $N_{v_\parallel}$=92, N_μ=4 .

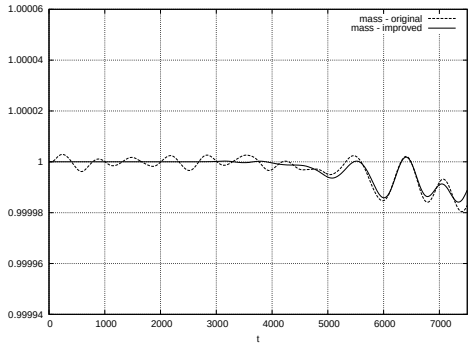


Figure 17: Evolution of the mass for a 5D toroidal test case

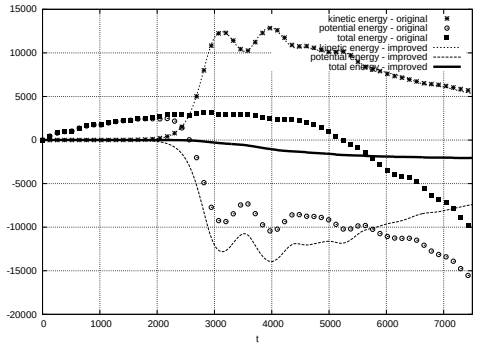


Figure 18: Evolution of the energies for a 5D toroidal test case

In Figures (17) and (18), a reference 5D case in the original version of the code is compared to the same case with the new version of the code that includes specific treatment for boundary conditions, operator splitting (separation of linear/non-linear operators), delta-f interpolation in 4D. It is noteworthy that the analysis of the different improvements that have been carried out for the 4D simulations in the previous Sections are also true in 5D setting. The mass conservation (Fig. 17) is enhanced in the early phase ($t < 2000$). The energy conservation is improved a lot by the new features during the whole simulation (*i.e.* in Fig. 18 the black thick line is closer to zero than the black squares).

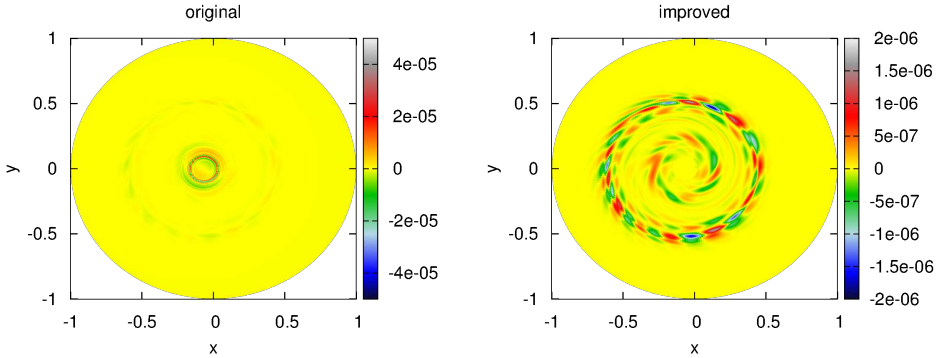


Figure 19: Poloidal cuts of the distribution function at $t = 1024$ for $v_{||} = 4.8 v_{th}$, $\mu = 0$ of original code (left), modified version with new boundary conditions and delta-f interpolation (right)

In Fig. 19, a comparison of the poloidal cuts of the distribution function during the 5D case (at $t = 1024$). The results from the original code are shown on the left-hand side of the figure, while outputs of the new version of the code including all the upgrades are shown on the right-hand side. For this time step, the turbulence is spoiled by errors coming from radial boundary conditions (left plot), but not in the improved version (right plot).

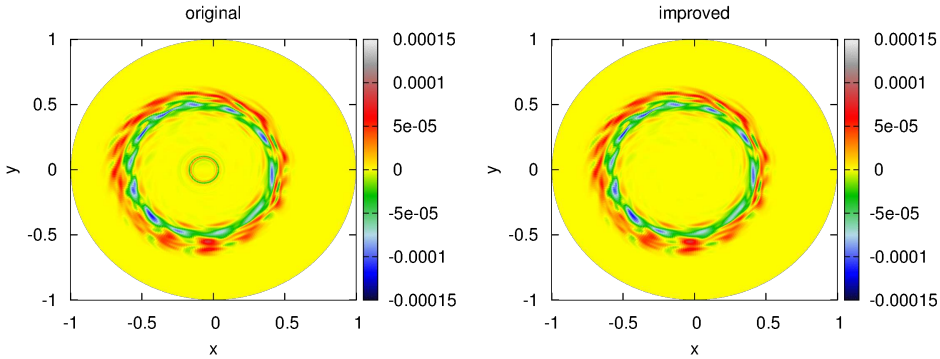


Figure 20: Poloidal cuts of the distribution function at $t = 4096$ for $v_{||} = 4.8 v_{th}$, $\mu = 0$ of original code (left), modified version with new boundary conditions and delta-f interpolation (right)

In Fig. 20, a similar comparison is performed at $t = 4096$, during the non-linear saturation

phase. The turbulence level has now increased, and the two figures in the left and right panels are quite similar. Some numerical artifacts are still there (original code), close to $r_{\min}$, but turbulent eddies are similar in both panels at macroscopic level.

Although the underlying set of gyrokinetic equations is energy-conserving, energy conservation is usually hard to observe in practice in gyrokinetic codes dedicated to Tokamak simulations (without enforcing energy conservation through a dedicated operator). Indeed, the global nonlinear 5D simulations are very demanding with respect to the numerical methods and the computational effort. Then, the results we obtained on the energy conservation are satisfying, they indicate that we are able to reach a reasonable level of accuracy.

7 Conclusion

Several numerical schemes have been investigated to improve the conservation properties expected in a gyrokinetic code. The semi-Lagrangian code GYSELA is enhanced by these schemes in terms of energy conservation, mass conservation and of preserving some stationary states. The first technique presented here targeted the radial boundary conditions. Radial boundaries are adapted in order to prevent particles from escaping the computational domain. Second, Vlasov solver has been split into two separate steps: the linear part and the non-linear part. This change improves energy conservation and the modeling of the quickest particles. Third, a 4D advection technique combined with a delta-f interpolation scheme and an accurate precomputation of the feet of characteristics allow: first, to better represent a distribution function close to a stationary state, second, to improve energy and mass conservation.

References

[ABH+06] P. Angelino, A. Bottino, R. Hatzky, S. Jolliet, O. Sauter, T. M. Tran, and L. Villard. On the definition of a kinetic equilibrium in global gyrokinetic simulations. *Physics of Plasmas*, 13(5):052304, 2006.

[AGG+11] J. Abiteboul, X. Garbet, V. Grandgirard, S. J. Allfrey, Ph. Ghendrih, G. Latu, Y. Sarazin, and A. Strugarek. Conservation equations and calculation of mean flows in gyrokinetics. *Physics of Plasmas*, 18(8):082503, 2011.

[BCG+11] J.-P. Braeunig, N. Crouseilles, V. Grandgirard, G. Latu, M. Mehrenberger, and E. Sonnendrücker. Some numerical aspects of the conservative PSM scheme in a 4D drift-kinetic code. Technical report, INRIA, December 2011. http://hal.archives-ouvertes.fr/hal-00650343.

[DPGS+08] G. Dif-Pradalier, V. Grandgirard, Y. Sarazin, X. Garbet, Ph. Ghendrih, and P. Angelino. On the influence of initial state on gyrokinetic simulations. *Physics of Plasmas*, 15(4):042315, 2008.

[GBB+06] V. Grandgirard, M. Brunetti, P. Bertrand, N. Besse, X. Garbet, P. Ghendrih, G. Manfredi, Y. Sarazin, O. Sauter, E. Sonnendrucker, J. Vaclavik, and L. Villard. A drift-kinetic Semi-Lagrangian 4D code for ion turbulence simulation. *Journal of Computational Physics*, 217(2):395 – 423, 2006.

[GIVW10] X. Garbet, Y. Idomura, L. Villard, and T.H. Watanabe. Gyrokinetic simulations of turbulent transport. *Nuclear Fusion*, 50(4):043002, 2010.

[GSG+08] V. Grandgirard, Y. Sarazin, X. Garbet, G. Dif-Pradalier, Ph. Ghendrih, N. Crouseilles, G. Latu, E. Sonnendrücker, N. Besse, and P. Bertrand. Computing ITG turbulence with a full-f semi-Lagrangian code. *Communications in Nonlinear Science and Numerical Simulation*, 13(1):81 – 87, 2008.

[Hah88] T. S. Hahm. Nonlinear gyrokinetic equations for tokamak microturbulence. *Physics of Fluids*, 31(9):2670–2673, 1988.

[IIK+08] Y. Idomura, M. Ida, T. Kano, N. Aiba, and S. Tokuda. Conservative global gyrokinetic toroidal full-f five-dimensional vlasov simulation. *Computer Physics Communications*, 179(6):391 – 403, 2008.

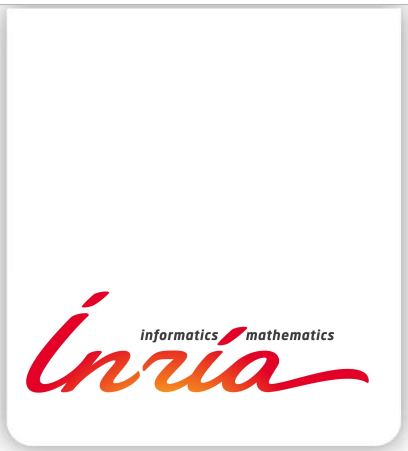
[ITK03] Y. Idomura, S. Tokuda, and Y. Kishimoto. Global gyrokinetic simulation of ion temperature gradient driven turbulence in plasmas with canonical maxwellian distribution. *Nuclear Fusion*, 43:234–243, 2003.

[LGA+12] G. Latu, V. Grandgirard, J. Abiteboul, M. Bergot, N. Crouseilles, X. Garbet, Ph. Ghendrih, M. Mehrenberger, Y. Sarazin, H. Sellama, E. Sonnendrücker, and D. Zarzoso. Accuracy of unperturbed motion of particles in a gyrokinetic semi-Lagrangian code. Rapport de recherche RR-8054, INRIA, September 2012.

[MSM+13] M. Mehrenberger, C. Steiner, L. Marradi, N. Crouseilles, E. Sonnendrücker, and B. Afeyan. Vlasov on GPU (VOG project). *ESAIM Proceedings*, 43:37–58, 2013.

[SGA+11] Y. Sarazin, V. Grandgirard, J. Abiteboul, S. Allfrey, X. Garbet, Ph. Ghendrih, G. Latu, A. Strugarek, G. Dif-Pradalier, P.H. Diamond, S. Ku, C.S. Chang, B.F. McMillan, T.M. Tran, L. Villard, S. Jolliet, A. Bottino, and P. Angelino. Predictions on heat transport and plasma rotation from global gyrokinetic simulations. *Nuclear Fusion*, 51(10):103023, 2011.

[ZGS+12] D. Zarzoso, X. Garbet, Y. Sarazin, R. Dumont, and V. Grandgirard. Fully kinetic description of the linear excitation and nonlinear saturation of fast-ion-driven geodesic acoustic mode instability. *Physics of Plasmas*, 19(2):022102, 2012.



RESEARCH CENTRE
RENNES – BRETAGNE ATLANTIQUE

Campus universitaire de Beaulieu
35042 Rennes Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399