Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 217 (2006) 395–423
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A drift-kinetic Semi-Lagrangian 4D code for ion turbulence simulation

V. Grandgirard [a,*], M. Brunetti [b], P. Bertrand [c], N. Besse [d], X. Garbet [a], P. Ghendrih [a], G. Manfredi [c], Y. Sarazin [a], O. Sauter [b], E. Sonnendrücker [d], J. Vaclavik [b], L. Villard [b]

[a] *CEA/DSM/Département de Recherche sur la Fusion Contrôlée, Association Euratom-CEA, Cadarache, 13108 St Paul-lez-Durance, France*
[b] *CRPP, Association Euratom-Confédération Suisse, EPFL, 1015 Lausanne, Switzerland*
[c] *LPMIA, Université Henri Poincaré-Nancy 1, P 239, 54506 Vandoeuvre-les-Nancy, France*
[d] *IRMA, Université Louis Pasteur, 7, rue René Descartes, 67084 Strasbourg Cedex, France*

Received 4 October 2004; received in revised form 4 January 2006; accepted 5 January 2006
Available online 20 March 2006

Abstract

A new code is presented here, named Gyrokinetic SEmi-LAgragian (GYSELA) code, which solves 4D drift-kinetic equations for ion temperature gradient driven turbulence in a cylinder (r, θ, z). The code validation is performed with the slab ITG mode that only depends on the parallel velocity. This code uses a semi-Lagrangian numerical scheme, which exhibits good properties of energy conservation in non-linear regime as well as an accurate description of fine spatial scales. The code has been validated in the linear and non-linear regimes. The GYSELA code is found to be stable over long simulation times (more than 20 times the linear growth rate of the most unstable mode), including for cases with a high resolution mesh ($\delta r \sim 0.1$ Larmor radius, $\delta z \sim 10$ Larmor radius).

Keywords: Semi-Lagrangian; Leap-frog; Time-splitting; Drift-kinetic; Ion-temperature-gradient; Conservation laws

1. Introduction

Turbulent transport is a key issue for controlled thermonuclear fusion based on magnetic confinement. The thermal confinement of a magnetized fusion plasma is essentially determined by turbulent heat conduction across the equilibrium magnetic field. In practice, the study of plasma turbulence requires to solve the Maxwell equations coupled to the calculation of the plasma response to the perturbed electromagnetic field. This response can be computed by using either a fluid or a kinetic description of the plasma. Solving 3D fluid equations is certainly the most convenient and fastest way to solve the problem given the set of well established

* Corresponding author.
E-mail address: virginie.grandgirard@cea.fr (V. Grandgirard).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.01.023

numerical techniques and the wealth of results obtained in the domain of fluid turbulence. However, it is known that the stability threshold given by fluid equations is lower than the actual (kinetic) value [1]. Also a fluid description usually overestimates turbulent fluxes [1]. This discrepancy comes partly from the resonant interactions between waves and particles (Landau resonances), which cannot be fully described with fluid equations. Also, the behavior of zonal flows, which play an important role in regulating turbulence, is not properly described by fluid equations in a weakly collisional regime. A first solution to overcome this difficulty is to introduce improved closure schemes in the set of fluid equations in order to recover the actual stability threshold and turbulent flux [2–4]. Comparing fluid and kinetic simulations provides a test of this closure scheme. In fact this task has proved to be much more difficult than expected [1]. The second solution is to solve the kinetic problem in order to compute accurately the turbulence in nearly collisionless plasmas. This is a formidable challenge. In principle, one has to solve a 6D kinetic equation (3D in space and 3D in velocity) to determine a distribution function, which yields current and charge densities once integrated over the velocity space. For strongly magnetized plasmas, averaging the kinetic equation over the cyclotron motion, which is faster than turbulent motion, reduces the dimensionality. The new kinetic equation, called "gyrokinetic", describes the distribution function in the 5D phase space (3D in space and 2D in velocity, namely $v_\parallel$ and $v_\perp$) associated to the guiding center motion. In this case, the adiabatic invariant, $\mu = mv_\perp^2/2B$ the action variable associated to the gyrophase, acts as a parameter. This 5D gyrokinetic problem is still very demanding in terms of numerics.

Two methods have been used up to now to investigate turbulence in the gyrokinetic regime. The first method is based on a Lagrangian approach. Particles in cell (PIC) codes, which are the most widely used in this category [5–11], consists in describing the plasma with a finite number of macro-particles. The trajectories of these particles are the characteristics of the Vlasov equation, whereas self-consistent fields are computed by gathering the charge and current densities of the particles on a mesh of the physical space [12]. Although this method allows one to obtain satisfying results with a small number of particles, it is well known that the reduction of the numerical noise inherent to the particle method requires a large number of particles. In particular the slow convergence with increasing number of particles is inherent to the PIC method, which is based on a statistical sampling of phase space. Improvements to the method have been brought by reinitializing the distribution of marker particles so as to concentrate them in regions of phase space where the perturbed part of the distribution function becomes large in absolute value [13–15]. Despite this significant improvement, known under the concept of importance sampling, there appears an upper bound in the simulation time, both due to the fact that $\|\delta f\|$ becomes of equal or even larger size than $\|f\|$ and due to the filamentation in velocity space which is a general property of the solution of the collisionless Vlasov equation hence effecting all numerical methods. The second method is Eulerian [16–22]. It consists in discretizing the Vlasov equation on a mesh of the phase space that remains fixed in time. The flux balance method (FBM) [23] uses a finite volume method for computing the average of the Vlasov equation on each cell on a fixed grid. More recently, the positive flux conservative (PFC) method [24] have been improved by introducing a slope limiter for the reconstruction of the distribution function to preserve the positivity and the mass. However, the trade-off for these improved conservation properties is a significant increase in the numerical dissipation.

The aim of this work is to use an intermediate method based on a semi-Lagrangian (SL) method [25]. This method has already been applied to calculate a turbulence driven by passing ions in 2D (1D in space, 1D in velocity) [26] and trapped ions in 3D (2D in space, 1D in velocity) [27]. In this paper a 4D model (3D in space and $v_\parallel$ (with $\mu = 0$)) for slab-ITG turbulence is used as a test bed. The purpose of the SL method is to take advantage of both the Lagrangian and Eulerian approaches, to have a good description of the phase space, in particular in regions where the density is low, as well as an enhanced numerical stability. In this approach, the mesh grid is kept fixed in time in the phase space (Eulerian method) and the Vlasov equation is integrated along the trajectories (Lagrangian method) using the invariance of the distribution function along the trajectories. Cubic spline interpolations are performed to evaluate the new value of the distribution function on the grid points. The integration along the trajectories is performed with a time-splitting algorithm, that allows to split the 4D advection equation into a sequence of 2D and 1D advections. The global numerical scheme is second order accurate in time by using a symmetrical time-splitting scheme and a leap-frog algorithm. Here, the full distribution function f is calculated in contrast with PIC codes that only calculate the perturbed

distribution function $\delta f = f - f_{eq}$ (where f_{eq} corresponds to a reference distribution function, usually Maxwellian). The PIC-δf codes have been recently revisited [14] so as to directly use the property of conservation of f along characteristics and avoiding completely the time integration of the δf equation but the information for f is known on a randomly chosen points that move in time. Therefore, in the semi-Lagrangian method, there is no constraint on the computation time related to the condition $\delta f \leqslant f$. However, a difficulty is faced in the numerics, namely the occurrence of negative values of the distribution function. Negative values may appear in regions of the phase space where the equilibrium distribution function is small when strong resonant interactions occur between waves and particles. This is due to the limitations in the interpolation procedures. This difficulty can be overcome by increasing the number of grid points and/or by changing the interpolation procedure [28]. Another numerical issue is energy conservation. An exact law of energy conservation can be built from the set of gyrokinetic equations. However this property of energy conservation is not always fulfilled during the simulations (unless implemented in the numerics). It will be shown here that SL method ensures good conservation properties if small scales are filtered.

The remainder of this paper is organized as follows. The physical model is described in Section 2. The numerical method is addressed in Section 3. The parallelization of the code is developed in Section 4. The numerical results are presented in Section 5 and improved schemes are analyzed in Section 6. Finally, a summary is given in Section 7.

2. Physical model

2.1. A drift-kinetic system in cylindrical geometry

The code presented in this paper is applied to a cylinder geometry with a reduction of the phase space to 4D. The goal of this work is to investigate turbulent transport in 5D in a realistic tokamak geometry together with the relevant physics of low frequency turbulent activity. A family of codes has been developed with increasing dimensionality 2D, 3D and 4D to assess the numerics that will be the backbone of the full 5D code. In the 4D version, a periodic cylindrical plasma of radius a and length $2\pi R$ is considered as a limit case of a stretched torus. The plasma is confined by a strong magnetic field which is uniform $\vec{B} = B\vec{e}_z$ where $\vec{e}_z$ stand for the unit vector in the toroidal direction z. In this collisionless plasma the electrons are assumed to respond adiabatically to the low frequency fluctuations. Concerning the ions, finite Larmor radius effects are neglected so that the trajectories are governed by the guiding-center (GC) trajectories

$$\frac{\mathrm{d}r}{\mathrm{d}t} = v_{\mathrm{GC}_r}; \quad r\frac{\mathrm{d}\theta}{\mathrm{d}t} = v_{\mathrm{GC}_\theta}; \quad \frac{\mathrm{d}z}{\mathrm{d}t} = v_\parallel; \quad \dot{v}_\parallel = \frac{q}{m_\mathrm{i}}E_z \tag{1}$$

where v_{GC_r} and v_{GC_θ} are the radial and poloidal components of the $E\times B$ drift velocity $\vec{v}_{\mathrm{GC}} = (\vec{E}\times\vec{B})/B^2$. $\vec{E}$ being the electric field, $q = Ze$ the ion charge and m_i the ion mass. $v_\parallel$ corresponds to the velocity along the magnetic field lines. Finally, it is assumed that fluctuations of the magnetic field are negligible. Thus the electrostatic approximation is used to compute the electric field, i.e., $\vec{E} = -\vec{\nabla}\Phi$, where the scalar Φ represents the electric potential. This simplified cylinder configuration does not take into account the toroidal effects but allows one to study slab ion temperature gradients driven modes (ITG). Given these assumptions, the distribution function f is a 4D phase space function that depends on the three cylindrical coordinates (r,θ,z) and on the parallel velocity $v_\parallel$. The evolution of this distribution function $f(r,\theta,z,v_\parallel,t)$ is described by the drift-kinetic Vlasov equation

$$\frac{\partial f}{\partial t} + \vec{v}_{\mathrm{GC}}\cdot\vec{\nabla}_\perp f + v_\parallel\frac{\partial f}{\partial z} + \dot{v}_\parallel\frac{\partial f}{\partial v_\parallel} = 0 \tag{2}$$

where $\vec{\nabla}_\perp = (\frac{\partial\cdot}{\partial r}, \frac{1}{r}\frac{\partial\cdot}{\partial\theta})$. This equation couples the $\vec{E}\times\vec{B}$ motion across the magnetic field to the motion parallel to the magnetic field. Self-consistency is ensured by the quasi-neutrality equation that relates the electric potential Φ to the first moment of the distribution function. Upon linearization, the quasi-neutrality reads

$$-\nabla_\perp\cdot\left[\frac{n_0(r)}{B\Omega_0}\nabla_\perp\Phi\right] + \frac{en_0(r)}{T_\mathrm{e}(r)}(\Phi - \langle\Phi\rangle) = n_\mathrm{i} - n_0 \tag{3}$$

where $\Omega_0 = q_i B_0/m_i$ is the ion cyclotron frequency, and T_e and n_0 are, respectively, the electron temperature and density profiles. The ion density profile is given by $n_i(r,\theta,z,t) = \int dv_\parallel f(r,\theta,z,v_\parallel,t)$ and $\langle\cdot\rangle$ represents the average on the magnetic field lines ($\langle\cdot\rangle = (1/L_z)\int\cdot\, dz$ with L_z the cylinder length). The first term on the left hand side corresponds to the linearized polarization term. The second term comes from the adiabatic response of the electrons. The expression $(\Phi - \langle\Phi\rangle)$ is due to the fact that the electron density fluctuations vanish for zonal modes [29].

2.2. Boundary and initial conditions

The distribution function is periodic in the θ and z directions, i.e.,

$$f(r,\theta,z,v_\parallel) = f(r,\theta+2\pi,z,v_\parallel)\ \forall\theta \quad \text{and} \quad f(r,\theta,z,v_\parallel) = f(r,\theta,z+L_z,v_\parallel)\ \forall z$$

Besides, we assume that there is no perturbation at the boundary in the non-periodic directions (r and $v_\parallel$). In the absence of buffer regions at the edge such boundary conditions prevent very long simulation times when the turbulence spreads till the center [30]. The plasma can be initialized by exciting a single ITG mode (m,n) (where m is a poloidal mode and n a toroidal mode) or by exciting a set of ITG modes with random amplitudes and phases. The distribution function is thus considered at the initial time as the sum of an equilibrium and a perturbated part: $f = f_{eq} + \delta f$. The equilibrium part f_{eq} is chosen as a local Maxwellian

$$f_{eq}(r,v_\parallel) = \frac{n_0(r)}{(2\pi T_i(r)/m_i)^{\frac{1}{2}}}\exp\left(-\frac{m_i v_\parallel^2}{2T_i(r)}\right) \tag{4}$$

while the perturbation δf is determined as

$$\delta f = f_{eq} g(r) h(v_\parallel)\delta p(z,\theta) \tag{5}$$

where $g(r)$ and $h(v_\parallel)$ are exponential functions such that $g(r = r_{min}) \sim g(r = r_{max}) \sim 0$ and $h(v_\parallel = v_{\parallel min}) \sim h(v_\parallel = v_{\parallel max}) \sim 0$. The perturbation δp can be initialized with a cosine function with a single poloidal mode m and a single toroidal mode n as

$$\delta p(z,\theta) = \epsilon\cos\left(\frac{2\pi n}{L_z}z + m\theta\right)$$

or with a bath of modes

$$\delta p(z,\theta) = \sum_{m,n}\epsilon_{mn}\cos\left(\frac{2\pi n}{L_z}z + m\theta + \phi_{mn}\right)$$

where ϵ_{mn} and ϕ_{mn} represent, respectively, a random amplitude and a random phase for the mode (m,n). The radial profiles of the ion and electron temperature (respectively, $T_i(r)$ and $T_e(r)$), as well as the radial density profile $n_0(r)$, are fixed in time and deduced by numerical integration of their gradient profiles given by the three parameters κ, Δr and r_p. For example,

$$\frac{1}{T_i(r)}\frac{dT_i(r)}{dr} = -\kappa_{T_i}\cosh^{-2}\left(\frac{r-r_p}{\Delta r_{T_i}}\right)$$

2.3. Energy conservation law

The kinetic energy, in fact the variation of the kinetic energy with respect to the equilibrium kinetic energy, is defined as

$$\delta\varepsilon_{kin} = \int m_i\frac{v_\parallel^2}{2}(f - f_{eq})\,dV\,dv_\parallel \quad \text{with } dV = r\,dr\,d\theta\,dz \tag{6}$$

Then according to Eqs. (2) and (3), the potential energy which satisfies $\frac{\partial\delta\varepsilon_{kin}}{\partial t} + \frac{\partial\delta\varepsilon_{pot}}{\partial t} = 0$ is given by (cf. Appendix A for more details)

$$\delta\varepsilon_{\mathrm{pot}} = \frac{q_{\mathrm{i}}}{2}\int (n_{\mathrm{i}} - n_0)\Phi \,\mathrm{d}V \tag{7}$$

A challenge for non-linear codes is the conservation of the total energy $\delta\varepsilon_{\mathrm{tot}} = \delta\varepsilon_{\mathrm{kin}} + \delta\varepsilon_{\mathrm{pot}} = \text{constant}$, a major property of the Vlasov equation. Errors in the energy conservation is used here as a measure of the code accuracy.

3. Numerical method

3.1. Discretization of the quasi-neutrality equation

The discretization of the quasi-neutrality equation (3) is performed by projecting in Fourier space along the two periodic directions (θ and z) and by using finite differences in the radial direction. Indeed, let Φ and n_{i} be represented in terms of the Fourier expansion as:

$$\begin{cases} \Phi(r,\theta,z) = \sum\limits_m \sum\limits_n \Phi^{m,n}(r)\exp(\mathrm{i}m\theta)\exp(\mathrm{i}nz) \\ n_{\mathrm{i}}(r,\theta,z) = \sum\limits_m \sum\limits_n n_{\mathrm{i}}^{m,n}(r)\exp(\mathrm{i}m\theta)\exp(\mathrm{i}nz) \end{cases}$$

then Eq. (3) is rewritten in the wave number representation, for each poloidal and toroidal mode (m and n), as the following differential equation:

$$-\frac{\partial^2\Phi^{m,n}(r)}{\partial r^2} - \left[\frac{1}{r} + \frac{1}{n_0(r)}\frac{\mathrm{d}n_0(r)}{\mathrm{d}r}\right]\frac{\partial\Phi^{m,n}(r)}{\partial r} + \frac{m^2}{r^2}\Phi^{m,n}(r) + \frac{\Omega_0 e}{T_{\mathrm{e}}(r)}\left[\Phi^{m,n}(r) - \Phi^{m,0}(r)\right] = \frac{\Omega_0}{n_0(r)}\left[n_{\mathrm{i}}^{m,n}(r) - n_0(r)\right] \tag{8}$$

It should be noticed that the $(m,n) = (0,0)$ mode is included in the simulation, thus allowing for the generation of zonal flows. To avoid the difficulties raised by the divergence of $\frac{1}{r}$ for $r \to 0$, the problem is solved within a ring $r_{\min} \leqslant r \leqslant a$, with $r_{\min} = 10^{-5}$. The boundary conditions are Dirichlet conditions on the axis ($\Phi^{m,n}(r_{\min}) = 0$ for all m and n). Although such a condition looks somewhat artificial for the equilibrium mode $(m,n) = (0,0)$ (one should rather expect $\frac{\mathrm{d}}{\mathrm{d}r}\Phi(r_{\min}) = 0$, i.e., no poloidal rotation), it does not impact the numerical results we wish to emphasize in this paper. The plasma is considered like a conductor on the outer boundary, i.e., $\vec{E}_{\mathrm{tg}} = 0$, which means in Fourier space: $\mathrm{i}m\Phi^{m,n}(a) = 0$ and $\mathrm{i}n\Phi^{m,n}(a) = 0$. So if $m \neq 0$ or $n \neq 0$, $\Phi^{m,n}(a) = 0$ and $\Phi^{0,0}(a)$ is assumed equal to 0 too. Let N_r be the number of radial points. Given the boundary conditions afore mentioned and up to the second order in Δr, Eq. (8) leads to the tridiagonal $(N_r - 2)\times(N_r - 2)$ system

$$\begin{pmatrix} b_{r_2} & c_{r_2} & 0 & & \\ a_{r_3} & b_{r_3} & c_{r_3} & 0 & \\ & \ddots & \ddots & \ddots & \\ & 0 & a_{r_{N_r-2}} & b_{r_{N_r-2}} & c_{r_{N_r-2}} \\ & & 0 & a_{r_{N_r-1}} & b_{r_{N_r-1}} \end{pmatrix} \begin{pmatrix} \Phi_2^{m,n} \\ \Phi_3^{m,n} \\ \vdots \\ \Phi_{N_r-2}^{m,n} \\ \Phi_{N_r-1}^{m,n} \end{pmatrix} = \begin{pmatrix} \rho_2^{m,n} - a_{r_2}\Phi_1^{m,n} \\ \rho_3^{m,n} \\ \vdots \\ \rho_{N_r-2}^{m,n} \\ \rho_{N_r-1}^{m,n} - c_{r_{N_r-1}}\Phi_{N_r}^{m,n} \end{pmatrix}$$

with

$$\begin{cases} a_{r_i} = -\left(\frac{1}{\Delta r^2} - \frac{\alpha(r_i)}{2\Delta r}\right) \text{ where } \alpha(r_i) = \frac{1}{r} + \frac{1}{n_0(r_i)}\frac{\mathrm{d}n_0(r_i)}{\mathrm{d}r} \\ b_{r_i} = \frac{2}{\Delta r^2} + \frac{m^2}{r_i^2} + (1 - \delta_{n=0})\frac{\Omega_0 e}{T_{\mathrm{e}}(r_i)} \text{ with } \delta \text{ the Kronecker symbol} \\ c_{r_i} = -\left(\frac{1}{\Delta r^2} + \frac{\alpha(r_i)}{2\Delta r}\right) \\ \rho_i^{m,n} = \frac{1}{n_0(r_i)}\left(n_{\mathrm{i}}^{m,n}(r_i) - n_0(r_i)\right) \end{cases}$$

This tridiagonal system is solved by using a LU decomposition [31,32]. The projections in Fourier space are performed by FFT.

3.2. Solution of the Vlasov equation

3.2.1. Time-splitting

In this uniform field case, the Liouville theorem is verified, i.e., $\vec{\nabla}_\perp \cdot \vec{v}_{\mathrm{GC}} + \frac{\partial v_\parallel}{\partial z} + \frac{\partial \dot{v}_\parallel}{\partial v_\parallel} = 0$. This property characterizes the incompressibility of the gyro-center orbits. With this property the Vlasov equation (2) can be written in its conservative form

$$\frac{\partial f}{\partial t} + \vec{\nabla} \cdot (\vec{v}_{\mathrm{GC}} f) + \frac{\partial}{\partial z}(v_\parallel f) + \frac{\partial}{\partial v_\parallel}(\dot{v}_\parallel f) = 0$$

Therefore this equation can be solved (cf. proof in Ref. [33]) by splitting between space and velocity coordinates into three conservative equations:

$$\frac{\partial f}{\partial t} + \vec{\nabla}_\perp \cdot (\vec{v}_{\mathrm{GC}} f) = 0$$
$$\frac{\partial f}{\partial t} + \frac{\partial (v_\parallel f)}{\partial z} = 0$$
$$\frac{\partial f}{\partial t} + \frac{\partial (\dot{v}_\parallel f)}{\partial v_\parallel} = 0$$

All the Eulerian methods based on finite volume methods work on conservative form of equations while using a semi-Lagrangian method requires to work directly on advection equations. Since $\vec{\nabla} \cdot \vec{v}_{\mathrm{GC}} = 0$, $\frac{\partial v_\parallel}{\partial z} = 0$ and $\frac{\partial \dot{v}_\parallel}{\partial v_\parallel} = 0$ the previous system is equivalent to:

$$\frac{\partial f}{\partial t} + \vec{v}_{\mathrm{GC}} \cdot \vec{\nabla}_\perp f = 0 \tag{9}$$
$$\frac{\partial f}{\partial t} + v_\parallel \frac{\partial f}{\partial z} = 0 \tag{10}$$
$$\frac{\partial f}{\partial t} + \dot{v}_\parallel \frac{\partial f}{\partial v_\parallel} = 0 \tag{11}$$

So as to solve these three advection equations the following numerical scheme is adopted. Let $\widehat{r\theta}$ denotes the shift operator in (r,θ) direction over a time step Δt, associated to the advection term in Eq. (9). Similarly, $\hat{z}$ and $\hat{v}_\parallel$ denote the shift operators, respectively, in the z (Eq. (10)) and $v_\parallel$ directions (Eq. (11)). A splitting of Strang [34] is applied to keep a scheme of second order accuracy (cf. proof in Appendix B). Second order accuracy is obtained by imposing a symmetry in the application of the different shifts. In our case the most efficient sequence is $(\hat{v}_\parallel/2, \hat{z}/2, \widehat{r\theta}/2, \widehat{r\theta}/2, \hat{z}/2, \hat{v}_\parallel/2)$ (where factor 1/2 corresponds to a shift over a $\Delta t/2$) because with this sequence the two (r,θ) shifts in 2D can be connected. So that the algorithm time step can be summarized by $(\hat{v}_\parallel/2, \hat{z}/2, \widehat{r\theta}, \hat{z}/2, \hat{v}_\parallel/2, \hat{Q})$, where $\hat{Q}$ denotes symbolically that at this point the quasi-neutrality equation is solved to compute the electric potential and thereby the electric field. The shifts in the z and $v_\parallel$ directions are straightforward, but the one in the (r,θ) direction requires more attention. Indeed, if we consider the action of the $\widehat{r\theta}$ operator between times $t - \Delta t$ and $t + \Delta t$, the value of the electric field E at time t is required to keep a time scheme of second order. This value is calculated by using a leap-frog method, which involves the use of two distribution functions shifted in time by one time step.

3.2.2. Semi-Lagrangian concept

Let $\vec{\Gamma}$ be a position vector in the phase space such that $\vec{\Gamma} = (r, \theta, z, v_\parallel)$ and let $\vec{\Gamma}_i$ be a position vector which corresponds to a node of the mesh. The semi-Lagrangian method is based on the invariance of the distribution function f along its characteristics Eq. (1) because,

$$\frac{\mathrm{d}f}{\mathrm{d}t}(\vec{\Gamma}(t), t) = \frac{\partial f}{\partial t} + \frac{\mathrm{d}r}{\mathrm{d}t}\frac{\partial f}{\partial r} + r\frac{\mathrm{d}\theta}{\mathrm{d}t}\frac{1}{r}\frac{\partial f}{\partial \theta} + \frac{\mathrm{d}z}{\mathrm{d}t}\frac{\partial f}{\partial z} + \frac{\mathrm{d}v_\parallel}{\mathrm{d}t}\frac{\partial f}{\partial v_\parallel} = \frac{\partial f}{\partial t} + \vec{v}_{\mathrm{GC}} \cdot \vec{\nabla} f + v_\parallel \frac{\partial f}{\partial z} + \dot{v}_\parallel \frac{\partial f}{\partial v_\parallel} = 0$$

according to the Vlasov equation (2). Therefore, the distribution function can be computed at each time step on the same fixed grid, by using

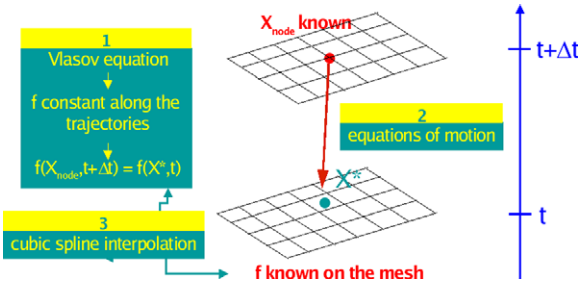


Fig. 1. Semi-Lagrangian basic concept.

$$f(\vec{\Gamma}_i(t_n+\Delta t),t_n+\Delta t)=f(\vec{\Gamma}(t_n,\vec{\Gamma}_i,t_n+\Delta t),t_n)$$

where $\vec{\Gamma}(t_n,\vec{\Gamma}_i,t_n+\Delta t)$ represents the solution of the characteristic at time step t_n is equal to $\vec{\Gamma}_i$ at time $t_n+\Delta t$. The method consists first in finding the foot of the characteristic at the time t_n: $\vec{\Gamma}(t_n,\vec{\Gamma}_i,t_n+\Delta t)$. The second step is to compute $f(\vec{\Gamma}(t_n,\vec{\Gamma}_i,t_n+\Delta t),t_n)$ by interpolation, because at this time, the distribution function is known over the whole fixed grid. This scheme is summarized in Fig. 1.

This sequence of operations can be applied separately on each advection equation appearing in the time-splitting algorithm. The computation of the foot of the characteristic for the 1D equations in the z and $v_\parallel$ directions are trivial unlike that for the 2D equation in (r,θ). This 2D equation cannot be divided into two 1D equations because $\frac{\partial v_{\mathrm{GC}r}}{\partial r}\neq 0$ and $\frac{\partial v_{\mathrm{GC}\theta}}{\partial \theta}\neq 0$.

3.2.3. Discretization of the (r,θ) motion equation

The 2D characteristic equation in the (r,θ) cross-section is performed in Cartesian coordinates to improve the numerical stability close to the axis. So computing the 2D trajectories is equivalent to solving the two following differential equations at first order:

$$\frac{\mathrm{d}x}{\mathrm{d}t}=v_{\mathrm{GC}_x}\quad\text{and}\quad\frac{\mathrm{d}y}{\mathrm{d}t}=v_{\mathrm{GC}_y}$$

where $v_{\mathrm{GC}_x}=E_x(x,y,z)/B_z$ and $v_{\mathrm{GC}_y}=E_y(x,y,z)/B_z$ represent the components of the $E\times B$ drift velocity in Cartesian coordinates. This system

$$\frac{\mathrm{d}\vec{X}}{\mathrm{d}t}=\vec{v}_{\mathrm{GC}X}(\vec{X},z,t) \tag{12}$$

is solved by using the parabolic assumption developed in [26]. Let $\vec{X}_{ij}$ be the position of $\vec{X}(t_n+\Delta t)$ at time $t_n+\Delta t$, then there exists a displacement $\vec{d}_{ij}$ tangent to the parabola such that (see Fig. 2)

$$\begin{cases}\vec{X}(t_n)=\vec{X}_{ij}-\vec{d}_{ij}\\ \vec{X}(t_n-\Delta t)=\vec{X}_{ij}-2\vec{d}_{ij}\end{cases}$$

Since the solution at second order of Eq. (12) can be written as

$$\frac{\vec{X}_{ij}-\vec{X}(t_n-\Delta t)}{2\Delta t}=\vec{v}_{\mathrm{GC}X}(\vec{X}(t_n),z_k,t_n)$$

where

$$\vec{X}(t_n)=\frac{\vec{X}(t_n+\Delta t)+\vec{X}(t_n-\Delta t)}{2}=\frac{\vec{X}_{ij}+\vec{X}(t_n-\Delta t)}{2}$$

the displacement $\vec{d}_{ij}$ can be calculated by solving the implicit equation

$$\vec{d}_{ij}=\Delta t\vec{v}_{\mathrm{GC}X}(\vec{X}_{ij}-\vec{d}_{ij},t_n) \tag{13}$$

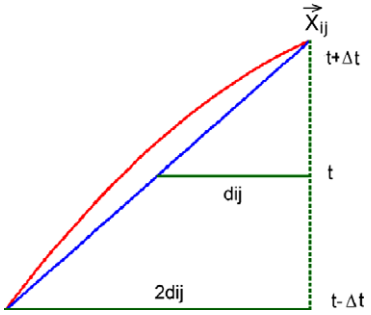


Fig. 2. Parabolic trajectory.

This implicit equation can be solved iteratively using a Newton–Raphson algorithm. Let the function g be defined by $g(\vec{d}_{ij}) = \vec{d}_{ij} - \Delta t \vec{v}_{GCX}(\vec{X}_{ij} - \vec{d}_{ij}, t_n)$, then the Newton iterate is given by

$$\vec{d}_{ij}^{m+1} = \vec{d}_{ij}^{m} - J_g^{-1}(\vec{d}_{ij}^{m}) g(\vec{d}_{ij}^{m}) \tag{14}$$

where J_g is the Jacobian matrix of g. So if we denote $(\alpha_{ij}, \beta_{ij})$ the coordinates of $\vec{d}_{ij}$ at the mesh knot $(r_i, \theta_j, z_k, v_{\|l})$ then assuming that $\vec{v}_{GCX}$ is linear in each grid cell, the Newton iterate yields:

$$\alpha_{ij}^{m+1} = \alpha_{ij}^{m} - \frac{1}{\Delta}[(\alpha_{ij}^{m} - \Delta t v_{GC_x})(1 + \Delta t \partial_y v_{GC_y}) - (\beta_{ij}^{m} - \Delta t v_{GC_y})(\Delta t \partial_y v_{GC_x})]$$

$$\beta_{ij}^{m+1} = \beta_{ij}^{m} + \frac{1}{\Delta}[(\alpha_{ij}^{m} - \Delta t v_{GC_x})(\Delta t \partial_x v_{GC_y}) - (\beta_{ij}^{m} - \Delta t v_{GC_y})(1 + \Delta t \partial_x v_{GC_x})]$$

where Δ corresponds to the determinant of J_g. This algorithm gives a good description of the trajectories. Fig. 3 shows the trajectories of 3 test-particles in a constant electric potential $\Phi(r, \theta)$, which follow the isopotential as predicted by the theory with a relative error of 0.1%.

The drawback of this method is that it requires the interpolation of $v_{GC_x}(x_i - \alpha_{ij}^{m}, y_j - \beta_{ij}^{m}, z_k)$ and $v_{GC_y}(x_i - \alpha_{ij}^{m}, y_j - \beta_{ij}^{m}, z_k)$. An another possibility to avoid this interpolation, performed with cubic splines, is to use a Taylor expansion. The first idea is to write Eq. (13) under the explicit form $\vec{d}_{ij}^{m+1} = \Delta t \vec{v}_{GCX}(\vec{X}_{ij} - \vec{d}_{ij}^{m}, t_n)$. So if $\vec{d}_{ij}^{0}$ is initialized at 0, then:

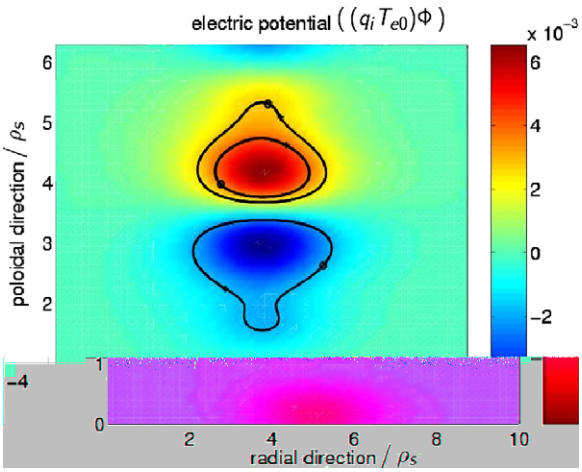


Fig. 3. Closed trajectories of 3 test-particles in a time-independent electric potential $\Phi(r, \theta)$ for $\Delta t = 0.5/\Omega_0$ and 50,000 iterations. The cross and the circle represent, respectively, the beginning and the end of the trajectories.

$$\vec{d}^{1}_{ij} = \Delta t \vec{v}_{\mathrm{GC}X}(\vec{X}_{ij} - \vec{d}^{0}_{ij}, t_n) = \Delta t \vec{v}_{\mathrm{GC}X}(\vec{X}_{ij}, t_n)$$
$$\vec{d}^{2}_{ij} = \Delta t \vec{v}_{\mathrm{GC}X}(\vec{X}_{ij} - \vec{d}^{1}_{ij}, t_n) = \Delta t \vec{v}_{\mathrm{GC}X}(\vec{X}_{ij} - \Delta t \vec{v}_{\mathrm{GC}X}(\vec{X}_{ij}, t_n), t_n)$$
$$= \Delta t \vec{v}_{\mathrm{GC}X}(\vec{X}_{ij}, t_n) - \Delta t J_v(\vec{X}_{ij}) \vec{d}^{1}_{ij} + \mathcal{O}((\Delta t)^3) \quad \text{(Taylor expansion at first order)}$$

then $\vec{d}^{2}_{ij} = \Delta t \vec{v}_{\mathrm{GC}X}(\vec{X}_{ij}, t_n) - \Delta t^2 J_v(\vec{X}_{ij}) \vec{v}_{\mathrm{GC}X}(\vec{X}_{ij}, t_n)$, where $J_v(\vec{X}_{ij})$ is the Jacobian matrix

$$\begin{pmatrix} \frac{\partial v_{\mathrm{GC}_x}(\vec{X}_{ij})}{\partial x} & \frac{\partial v_{\mathrm{GC}_x}(\vec{X}_{ij})}{\partial y} \\ \frac{\partial v_{\mathrm{GC}_y}(\vec{X}_{ij})}{\partial x} & \frac{\partial v_{\mathrm{GC}_y}(\vec{X}_{ij})}{\partial y} \end{pmatrix}$$

This simpler method is equivalent at second order to the Newton algorithm. Indeed, if $J_g^{-1}(\vec{d}_{ij}) = [I + \Delta t J_{\vec{v}_{\mathrm{GC}X}}(\vec{X}_{ij} - \vec{d}_{ij}, t_n)]^{-1}$ is expanded at second order as $J_g^{-1}(\vec{d}_{ij}) = I - \Delta t J_{\vec{v}_{\mathrm{GC}X}}(\vec{X}_{ij} - \vec{d}_{ij}, t_n) + \mathcal{O}((\Delta t)^2)$, then according to Eq. (14)

$$\vec{d}^{m+1}_{ij} = \Delta t \vec{v}_{\mathrm{GC}X}(\vec{X}_{ij} - \vec{d}^{m}_{ij}, t_n) + \Delta t J_{\vec{v}_{\mathrm{GC}X}}(\vec{X}_{ij} - \vec{d}^{m}_{ij}, t_n) \vec{d}^{m}_{ij} - \Delta t^2 J_{\vec{v}_{\mathrm{GC}X}}(\vec{X}_{ij} - \vec{d}^{m}_{ij}, t_n) \vec{v}_{\mathrm{GC}X}(\vec{X}_{ij} - \vec{d}^{m}_{ij}, t_n) + \mathcal{O}((\Delta t)^2)$$

so if $\vec{d}^{0}_{ij\,\mathrm{taylor}} = \vec{d}^{0}_{ij\,\mathrm{newton}} = 0$ then $\vec{d}^{2}_{ij\,\mathrm{taylor}} = \vec{d}^{1}_{ij\,\mathrm{newton}}$. The advantage of this Taylor method is that it requires the computation of the derivatives at first order of $\vec{v}_{\mathrm{GC}X}$ on nodes of the mesh and not on arbitrary points of space.

3.2.4. Cubic spline interpolation

When the characteristic foot is computed with the Taylor method, we need to compute $f(r^\star, \theta^\star, z_k, v_{\|l}, t_n - \Delta t)$ where $r^\star = \sqrt{(x_i - \alpha_{ij})^2 + (y_j - \beta_{ij})^2}$ and $\theta^\star = \arctan(\frac{y_j - \beta_{ij}}{x_i - \alpha_{ij}})$ are no longer grid points. Thus an interpolation is needed. In this case, a 2D interpolation (r, θ) is required, where z_k and $v_{\|l}$ are considered as parameters. For the resolution of Eqs. (10) and (11), 1D interpolations in the z direction (respectively, $v_\|$ direction) are required with r_i, θ_j and $v_\|$ (respectively, r_i, θ_j and z_k) fixed. So according to the advections, the 4D distribution function is interpolated on a 1D or 2D cubic spline basis. Let N_r, N_θ, N_z and $N_{v_\|}$ be the number of points, respectively, in r, θ, z and $v_\|$ directions. Then, for instance in the z advection, $f(r_i, \theta_j, z, v_{\|l})$ is approximated by

$$g_1(z) = f(r_i, \theta_j, z, v_{\|l}) = \sum_{v=-1}^{N_z+1} c_v \Lambda_v(z) \quad \forall r_i, \theta_j, v_{\|l}$$

where Λ are piecewise cubic polynomials (cf. [35]). In the case of an advection in (r, θ), f is defined as a 2D tensor product of cubic B-splines, as

$$g_2(r, \theta) = f(r, \theta, z_k, v_{\|l}) = \sum_{\alpha=-1}^{N_r+1} \sum_{\beta=-1}^{N_\theta+1} c_{\alpha,\beta} \Lambda_\alpha(r) \Lambda_\beta(\theta) \quad \forall z_k, v_{\|l}$$

The piecewise cubic polynomials Λ are twice continuously differentiable. For more details on the computation of the cubic spline coefficients see Appendix C.

3.3. Global algorithm

Taking into account all the previous steps, the global algorithm in time used to solve the 4D non-linear system (4D Vlasov equation + 3D quasi-neutrality equation) is summarized by the following sequence. Let the notations $n - 1$ and $n + 1$, respectively, correspond to the time $t_n - \Delta t$ and $t_n + \Delta t$. Given the distribution function at two times $t = t_{n-1}$ and $t = t_n$, then:

1. Computation of $E(t_n)$ with $f(t_n)$ by solving the quasi-neutrality equation.
2. Computation of $f^{n+1} = f(t_n + \Delta t)$ with $f^{n-1} = f(t_n - \Delta t)$ by using the centered electric field $E(t_n)$. This means using an algorithm of time-splitting on $2\Delta t$ according to the sequences $\hat{v}_\| \hat{z} 2(r, \hat{\theta}) \hat{z} \hat{v}_\|$, i.e.:
   (a) $f^\star(r, \theta, z, v_\|) = f^n(r, \theta, z, v_\| - \Delta t E_z(t_n))$,
   (b) $f^{\star\star}(r, \theta, z, v_\|) = f^\star(r, \theta, z - \Delta t v_\|, v_\|)$,

(c) $f^{\star\star\star}(r,\theta,z,v_\parallel) = f^{\star\star}(r-2\alpha_{ij},\theta-2\beta_{ij},z,v_\parallel)$ with $\vec{d}_{ij} = (\alpha_{ij},\beta_{ij})^t$ where the implicit equation $\vec{d}_{ij} = \Delta t \vec{v}_{GCX}(\vec{X}_{ij} - \vec{d}_{ij}, t_n)$ is solved by a Newton algorithm or by a Taylor method,
(d) $f^{4\star}(r,\theta,z,v_\parallel) = f^{\star\star\star}(r,\theta,z-\Delta t,v_\parallel)$,
(e) $f^{n+1} = f^{4\star}(r,\theta,z,v_\parallel - \Delta t E_z(t_n))$.

3. Leap-frog algorithm:
(a) $f(t_{n-1})$ becomes equal to $f(t_n)$,
(b) $f(t_n)$ becomes equal to $f(t_{n+1})$.

4. Parallel 4D code description

The 4D code is developed in Fortran 90 and parallelized with the MPI message passing library. It runs on SUN and ALPHA parallel computers as well as on PC cluster under Linux. At the moment only the 4D distribution function and Eq. (2) are parallelized. The discretization and the solution of the 3D quasi-neutrality equation is performed on each processor. As mentioned before the 4D Vlasov equation is solved by time-splitting. Hence the 4D discretization is replaced by a succession of discretizations of 2D advections in the (r,θ) direction and 1D advections in the z and $v_\parallel$ directions. To take advantage of this property, the 4D distribution function is saved in a 2D array where the first dimension corresponds to the directions (r,θ) and the second dimension corresponds to the 2 others directions $(z,v_\parallel)$. At each time, this 2D array is shared on processors according to the first or second dimension depending on the advection that is performed. Indeed, to resolve the 2D advection, each processor needs to know all the information on (r,θ). Therefore the 2D array is parallelized according to the second dimension. On the other hand, to solve the two 1D advections, each processor needs to know all the information on z or $v_\parallel$. So the 2D array must be transposed to be parallelized according to the first dimension. The advantage of this kind of parallelization is that the only communication between processors appears during the transposition of the 2D array and all the operations are fully local. The transposition is optimized for a number of processors which is a power of 2.

4.1. Speed-up

The performance of the parallel code is summarized in Table 1 for two different typical mesh sizes in $(r,\theta,z,v_\parallel)$: $(64\times128\times64\times64)$ and $(128\times64\times64\times128)$. All the tests have been performed on the parallel computer of the CEA (Commissariat à l'Energie Atomique) made of 180 quadri-processors. Each processor is an ALPHA EV68-1250 MHz with a power of 2.5 GFlops/s and a memory size of 1 GBytes.

As seen in Fig. 4, the speed-up (speed-up = monoprocessor time/CPU time) is poor for more than 64 processors. This is due to the fact that the computational time of the not parallelized 3D operations become non-negligible. The performance of the code will be improved in the future by parallelizing the resolution of the 3D quasi-neutrality equation.

5. Numerical results

The cylinder ITG instabilities correspond to small scale instabilities, which grow and saturate to a state of developed turbulence. In the following, the exponential increase of the amplitude of the initial perturbation will be called the linear phase. Due to the existence of energy invariants (like the number of particles for instance) and the self-consistent evolution, these perturbed modes cannot grow unbounded and a saturation

Table 1
CPU time in seconds for 1 global iteration for 2 different meshes: mesh1 = $(64\times128\times64\times64)$ and mesh2 = $(128\times64\times64\times128)$

Nb processors	1	2	4	8	16	32	64	128	256
Mesh1	309	155	85	50	30	21	16	15	14
Mesh2	×	313	167	94	52	31	21	17	16

The cross-symbol corresponds to a problem of insufficient virtual memory.

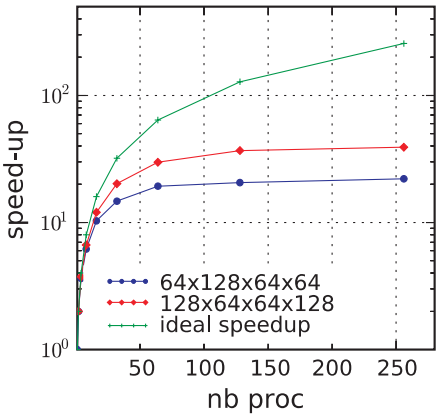


Fig. 4. Speed-up for two different meshes ($64 \times 128 \times 64 \times 64$) and ($128 \times 64 \times 64 \times 128$).

is reached. We will see that the GYSELA code is well adapted to compute this non-linear phase, which is obviously the most demanding and relevant part of the simulation.

5.1. Normalization

The numerical solution is performed using the normalized equations. In our case, the temperature is normalized to T_{e0}, where T_{e0} is defined such that $T_e(r_0)/T_{e0} = 1$ (where r_0 is a reference point). The time is normalized to Ω_0^{-1}, where $\Omega_0 = q_i B_0/m_i$ is the ion cyclotron frequency. The velocity is normalized to the sound speed $c_s = \sqrt{T_{e0}/m_i}$ and the normalization of the electric potential is defined by T_{e0}/q_i. Therefore, all the normalized quantities needed (represented with hat symbol) can be deduced and are summarized in Table 2.

5.2. Linear study

5.2.1. Computation of the growth rate and the instability threshold

The linearized Vlasov equation is obtained by separating the equilibrium distribution function from its perturbation in the Vlasov equation (2) and by keeping only the perturbations at first order. In this linear study, the equilibrium part f_{eq} is given by Eq. (4) and the perturbations are projected on a Fourier basis in θ and z directions as:

$$\delta f = \sum_{mn\omega} \delta f_{mn\omega}(r, v_\parallel) \exp[\mathrm{i}(m\theta + nz - \omega t)]$$

$$\Phi = \sum_{mn\omega} \Phi_{mn\omega}(r) \exp[\mathrm{i}(m\theta + nz - \omega t)]$$

According to these assumptions the linearized Vlasov equation is

$$\frac{\delta n_{i_{mn\omega}}}{n_0} = -\left(1 - \left\langle \frac{\omega - \omega_i^*}{\omega - k_\parallel v_\parallel} \right\rangle\right) \frac{q}{T_i(r)} \Phi_{mn\omega} \quad \text{with } \langle\cdot\rangle = \frac{1}{n_0} \int \cdot f_{eq}\, \mathrm{d}v_\parallel$$

where the diamagnetic frequency ω_i^* is given by

Table 2
Normalized quantities

$\hat{t} = \Omega_0 t$		$\hat{l} = (\Omega_0/c_s) l = l/\rho_s$
$\hat{v} = v/c_s$		$\hat{E} = (1/c_s B_0) E$
$\hat{T} = T/T_{e0}$	$\Rightarrow$	$\hat{n} = (\rho_s)^3 n$
$\hat{\Phi} = (q_i/T_{e0})\Phi$		$\hat{f} = (\rho_s)^3 c_s f$
$\hat{B} = B/B_0$		

$$\omega_{\mathrm{i}}^{*}=\left[\frac{m_{\mathrm{i}}v_{\parallel}^{2}}{2T_{\mathrm{i}}(r)}-\frac{1}{2}\right]\omega_{T_{\mathrm{i}}}^{*}+\omega_{n_{\mathrm{i}}}^{*}$$

with

$$\omega_{T_{\mathrm{i}}}^{*}=\frac{T_{\mathrm{i}}}{qB}\frac{\mathrm{d}\log T_{\mathrm{i}}(r)}{\mathrm{d}r}k_{\theta}\quad\text{and}\quad\omega_{n_{\mathrm{i}}}^{*}=\frac{1}{\eta}\omega_{T_{\mathrm{i}}}^{*}\quad\left(\eta=\frac{\mathrm{d}\ln T_{\mathrm{i}}}{\mathrm{d}\ln n_{0}}\right)$$

The Fourier wave numbers k_θ and $k_\parallel$ are, respectively, defined as $k_\theta = m/r$ and $k_\parallel = 2\pi n/L_z$. Let us remain here that we only consider particles with $\mu = 0$. In this case, the velocity space has one degree of freedom, hence leading to the coefficient 1/2 (instead of the usual 3/2 value) in the definition of ω_{i}^{*}.

The linearized quasi-neutrality equation is then

$$-\rho_{\mathrm{s}}^{2}\left[\frac{\partial^{2}\Phi_{mn\omega}(r)}{\partial r^{2}}+\left(\frac{1}{r}+\frac{1}{n_{0}(r)}\frac{\mathrm{d}n_{0}(r)}{\mathrm{d}r}\right)\frac{\partial\Phi_{mn\omega}(r)}{\partial r}\right]+\left(\rho_{\mathrm{s}}^{2}\frac{m^{2}}{r^{2}}+\frac{1}{Z_{\mathrm{i}}}\right)\Phi_{mn\omega}(r)=\frac{T_{\mathrm{e}}}{eZ_{\mathrm{i}}}\frac{\delta n_{\mathrm{i}_{mn\omega}}(r)}{n_{0}(r)}$$

Instead of solving the full differential equation, we use a test function of the form $\Phi_{mn\omega}(r)=\phi_{mn\omega}\exp[g(r)]$, then the previous equation can be written as

$$\left[\rho_{\mathrm{s}}^{2}\left(\kappa(r)+\frac{m^{2}}{r^{2}}\right)+\frac{1}{Z_{\mathrm{i}}}\right]\Phi_{mn\omega}(r)=\frac{T_{\mathrm{e}}}{eZ_{\mathrm{i}}}\frac{\delta n_{\mathrm{i}_{mn\omega}}(r)}{n_{0}(r)}$$

where $\kappa(r)$ is defined by

$$\kappa(r)=-\left[\frac{\partial^{2}g(r)}{\partial r^{2}}+\left(\frac{\partial g(r)}{\partial r}\right)^{2}+\left(\frac{1}{r}+\frac{1}{n_{0}(r)}\frac{\mathrm{d}n_{0}(r)}{\mathrm{d}r}\right)\frac{\partial g(r)}{\partial r}\right]$$

For the linear stability analysis, $\exp\{g(r)\}$ is chosen such that the profile of Φ is close to the numerical solution. Finally, the linearized dispersion relation can be deduced from the two previous relations

$$D(\omega)=\rho_{\mathrm{s}}^{2}\left(\kappa(r)+\frac{m^{2}}{r^{2}}\right)+\frac{1}{Z_{\mathrm{i}}}+\left(1-\left\langle\frac{\omega-\omega_{\mathrm{i}}^{*}}{\omega-k_{\parallel}v_{\parallel}}\right\rangle\right)\frac{T_{\mathrm{e}}}{T_{\mathrm{i}}}=0 \tag{15}$$

This local dispersion relation gives for each mode a relation between the real part of the frequency (temporal periodicity) and the wave number (spatial periodicity). This phase velocity characterizes the kind of waves that propagate in the plasma. The behavior of the linear growth rate γ (imaginary part of ω), with the Fourier wave number m and n, is given by the zeros of the linear dispersion relation $D(\omega) = 0$ where $\omega = \omega_{\mathrm{r}} + \mathrm{i}\gamma$ and ω_{r} is the real part of the frequency. The instability threshold corresponds to the case $\mathrm{Im}(\omega) = \gamma = 0$. $\lim_{\gamma\to 0^{+}} D(\omega) = 0$ is equivalent to the system of equations:

$$\varepsilon_{\mathrm{r}}=1+A(r)-\mathrm{PP}\left(\left\langle\frac{\omega_{\mathrm{r}}-\omega_{\mathrm{i}}^{*}}{\omega_{\mathrm{r}}-k_{\parallel}v_{\parallel}}\right\rangle\right)=0$$

$$\varepsilon_{\mathrm{i}}=\pi\left\langle\left(\omega_{\mathrm{r}}-\omega_{n_{\mathrm{i}}}^{*}-\omega_{T_{\mathrm{i}}}^{*}\left[\frac{v_{\parallel}^{2}}{v_{T_{\mathrm{i}}}^{2}}-\frac{1}{2}\right]\right)\delta(\omega_{\mathrm{r}}-k_{\parallel}v_{\parallel})\right\rangle=0$$

with $v_{T_{\mathrm{i}}}=\sqrt{T_{\mathrm{i}}/m_{\mathrm{i}}}$ the thermal velocity, where PP denotes the principal part and where

$$A(r)=\tau\rho_{\mathrm{s}}^{2}\left(\kappa(r)+\frac{m^{2}}{r^{2}}\right)+\frac{\tau}{Z_{\mathrm{i}}}$$

Thus using the relation

$$\lim_{\gamma\to 0^{+}}\frac{1}{x\pm\mathrm{i}\gamma}=\mathrm{PP}\left(\frac{1}{x}\right)\mp\mathrm{i}\pi\delta(x)$$

The relation between the ion temperature gradient and the density gradient is given by the following analytical expression:

$$\omega^*_{T_i} = \omega^*_{n_i} \pm \sqrt{\omega^{*2}_{n_i} + 2\omega^2_{\parallel}\tau(\tau + A(r))} \tag{16}$$

where $\omega_{\parallel} = k_{\parallel} v_{T_i}$. The dependence of $\omega^*_{T_i}$ on $\omega^*_{n_i}$ at the threshold is shown in Fig. 5. The distance to the threshold of the pair $(\omega^*_{T_i}, \omega^*_{n_i})$ for our numerical simulation is also shown in Fig. 5. In the limit $\omega^*_{n_i} \gg \omega_{\parallel}$, the threshold is $\eta = \omega^*_{T_i}/\omega^*_{n_i} = 2$.

According to Eq. (15) $D(\omega)$ is defined by

$$D(\omega) = \tau\left[\rho_s^2\left(\kappa(r) + \frac{m^2}{r^2}\right) + \frac{1}{Z_i}\right] + 1 - zZ(z)\left[-1 + \frac{\omega^*_{n_i}}{\omega} + \frac{\omega^*_{T_i}}{\omega}\left(-\frac{1}{2} + \frac{z}{Z(z)} + z^2\right)\right] \tag{17}$$

with $\tau = T_i/T_e$, $z = \frac{\omega}{k_{\parallel}}\sqrt{\frac{m_i}{2T_i}}$ and where $Z(z)$ represents the Fried and Conte function [36], i.e.,

$$Z(z) = \frac{1}{\sqrt{\pi}}\int_{-\infty}^{+\infty} \frac{\exp(-x^2)}{x - z}\,dx$$

The local dispersion relation (17) is solved for $r = r_{\text{peak}}$ and the zeros are found using a Davies method [37,38] coupled to a Newton algorithm. This spectral approach, which is clearly less time consuming than a global non-linear simulation, is currently used for microinstabilities analysis (code KINEZERO [39]). In our case this preliminary study is performed to check the validity of the physical input parameters. The results for a standard case are presented here. This numerical case corresponds to a 4D phase space $(r, \theta, z, v_{\parallel})$ defined by the following lengths: $L_r = 14.5\rho_s$, $L_\theta = 2\pi$, $L_z = 1508\rho_s$ and $v_{\parallel} \in [-6v_{T_i}, 6v_{T_i}]$. The electron temperature is assumed uniform. The density and the ion temperature profiles are, respectively, defined by the following parameters $\kappa_{n_0} = 0.8$, $\Delta r_{n_0} = 0.2$, $\kappa_{T_i} = 4$ and $\Delta r_{T_i} = 0.1$, which correspond to values of η larger than 2 in a sufficiently large region to study instabilities (cf. Fig. 6).

In this case the radial profile of the electric potential is approximated by

$$\Phi_{mn\omega}(r) \simeq \frac{\kappa_{T_i}}{\kappa_{n_0}} \exp\left[\frac{-(r - r_{\text{peak}})^2}{\frac{\Delta_{r_{n_0}}}{\Delta_{r_{T_i}}}}\right]$$

This analytical expression gives a good representation of the numerical solution. A comparison is shown in Fig. 7 at the time $t = 600/\Omega_0$ which corresponds to a time long enough for the development of the most unstable mode in the linear phase.

Analyzing the dependence of the linear growth rate γ on the poloidal mode number m for four different toroidal mode numbers ($n = 1, 2, 3, 4$) shows (cf. Fig. 8(a)) that the most unstable mode has an helicity $(m, n) = (11, 3)$ and that all the large wave numbers ($m > 10$) are unstable. The numerical problem caused

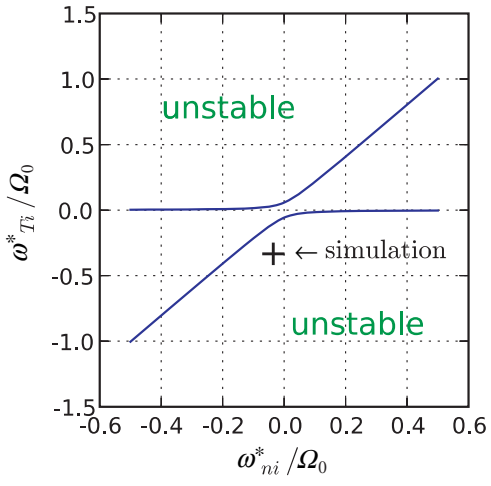
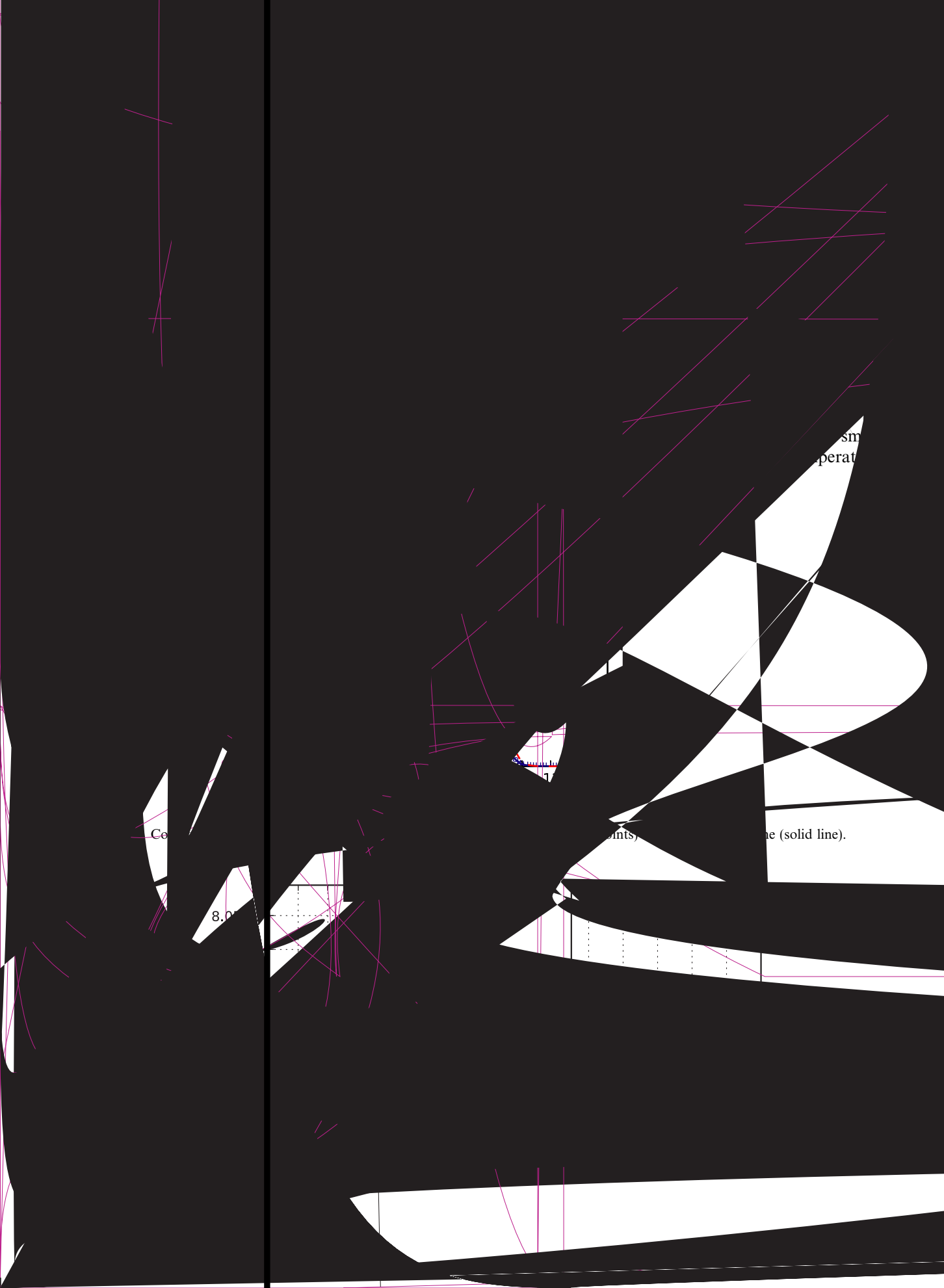


Fig. 5. Dependence of $\omega^*_{T_i}$ on $\omega^*_{n_i}$ at the threshold. The cross indicates the pair $(\omega^*_{T_i}, \omega^*_{n_i})$ in the numerical simulation.



5.2.2. Validation of the linear phase

The perturbation can be initialized with a single mode (m,n) as was done already in PIC-δf code [40]. This property is interesting for the study of specific modes. Besides, in this paper, this possibility is used to validate the linear phase of the non-linear code (see [41]). This validation is performed with the mode $(m,n) = (11,3)$ for which the analytical growth rate is equal to $\gamma = 6.259 \times 10^{-3}\Omega_0$. As seen before, the analytical approach is based on the approximation of the electric potential by a fixed profile close to the numerical one. The corresponding numerical simulation (for a mesh of 64 points in each directions and a time step of $\Delta t = 0.5/\Omega_0$) shows (cf. Fig. 9) that this assumption is valid for $t \leqslant 500/\Omega_0$. The numerical growth rate γ_{num} is computed with a linear fit between $t = 0$ and $t = 500/\Omega_0$ as $\gamma_{\text{num}} = [\log(\sqrt{\int \Phi^2(r_{\text{peak}},\theta,z)\,\mathrm{d}\theta\,\mathrm{d}z}) - \beta]/t$. In this case γ_{num} is equal to $6.15 \times 10^{-3}\Omega_0$, which gives a good agreement with the analytical value (relative error of 2%).

A second study has been performed by comparing the numerical growth rate obtained for superimposed ITG modes with different time steps and two different mesh sizes. This comparison, indicates that (cf. Fig. 10) the relative error is smaller than 1%, that validate the linear phase of the computation. Besides, benchmarks with two PIC codes (the linear code LORB5 [9] and the non-linear code ORB5 [8]) reveal good agreement (cf. [42]).

Tests have been performed for time steps larger than $20/\Omega_0$ for the mesh with 64 points in each directions. In this case the linear phase is no more properly described. Indeed, with this mesh $\Delta r = 0.23\rho_s$, $\Delta\theta = 0.0982\rho_s$, $\Delta z = 23.56\rho_s$ and $\Delta v_\parallel = 0.2326 c_s$ while $\max E_\theta = 3.3 \times 10^{-2} \times B_0 c_s$, $\max E_r = 3 \times 10^{-2} \times B_0 c_s$, $v_{\parallel\max}$=7.32c_s and

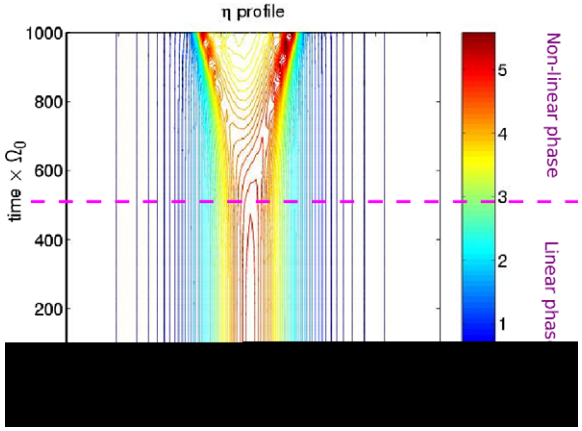


Fig. 9. Contour plot of the time evolution of the radial η profile.

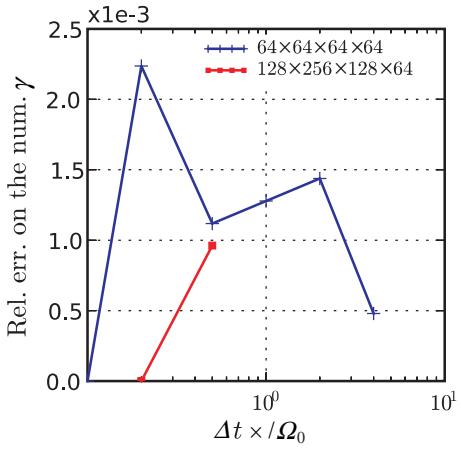
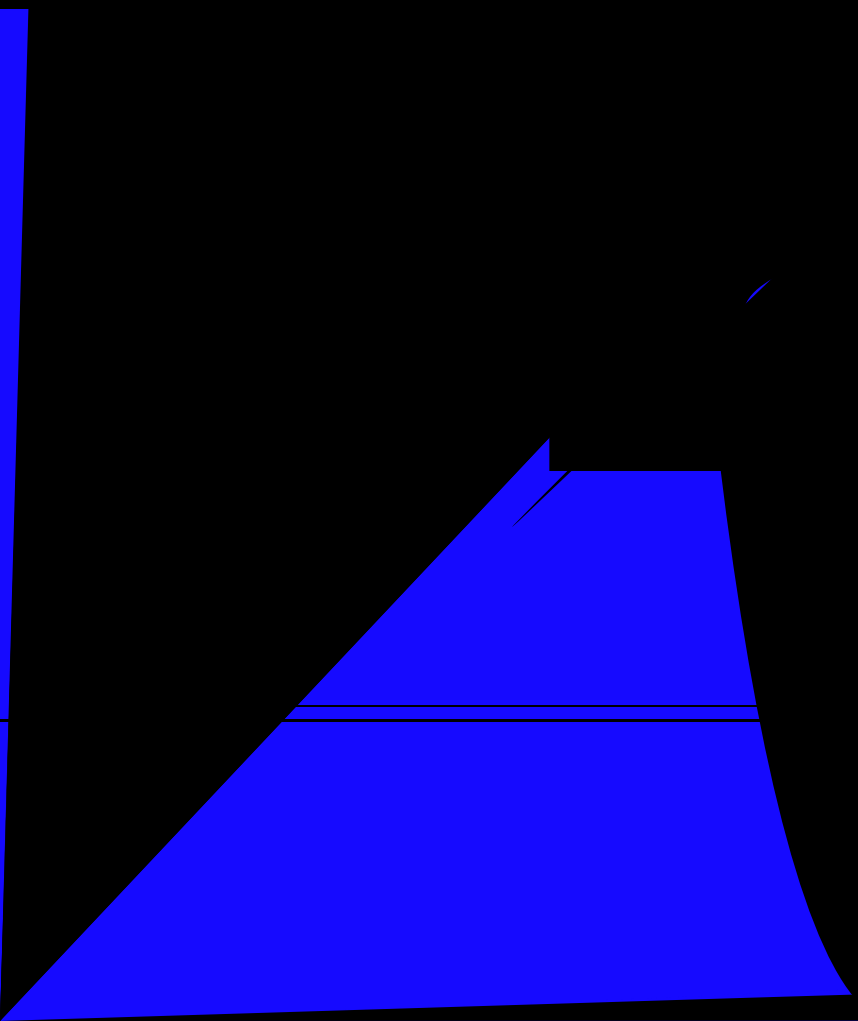


Fig. 10. Linear growth rate for two different mesh sizes ($64 \times 64 \times 64 \times 64$) and ($128 \times 256 \times 128 \times 64$) and for different time steps.



L^2-norm and the entropy are conserved with an error smaller than 0.3%. The relative error on the total energy remains below 2% all along the non-linear simulation. The maximum values (taken between $t = 0$ and $t = 3000/\Omega_0$) of the relative errors, reported in Table 3, prove that the tests on the conservation of the L^p-norm ($p = 1, 2$) and the entropy are necessary for validation of non-linear codes but not sufficient. Indeed, for instance, for a time step $\Delta t = 10/\Omega_0$ these three entities remain conserved with an accuracy better than 0.004% while the relative error on the total energy approaches the unacceptable value of 97%. The energy conservation test appears as the most sensitive validation test. The time evolution of the heat flux

$$Q(t) = \frac{1}{2}\int f v_\parallel^2 v_{\mathrm{GC}_r} \frac{\mathrm{d}\theta}{2\pi}\frac{\mathrm{d}z}{L_z}\mathrm{d}v_\parallel$$

at the radial position $r = r_{\mathrm{peak}}$ and the potential energy $\delta\varepsilon_{\mathrm{pot}}$ Eq. (7) for the four previous time steps ($\Delta t = 0.1/\Omega_0$, $\Delta t = 0.5/\Omega_0$, $\Delta t = 2/\Omega_0$ and $\Delta t = 8/\Omega_0$) is plotted in Fig. 12. Simulations at $\Delta t = 4.0/\Omega_0$ (not plotted) have also been performed and exhibit no significant difference with the case $\Delta t = 2.0/\Omega_0$. This figure shows that an error on the total energy of 93% generates a significant modification of the physical results. Conversely an error on the energy conservation of a couple of tens percent only lead to small differences in the computed heat flux and potential energy. So as to quantify the fine dynamics in such regimes, one can compute the largest spatial excursion of the distribution function in a single time step. This corresponds to $v_{Er}\Delta t$ (respectively, $v_{E\theta}\Delta t$) in the radial (respectively, poloidal) direction. Here, v_E corresponds to the $E \times B$ drift velocity. It turns out that, in both directions, the code starts failing in the non-linear regime when the excursion reaches about one grid cell, which is obtained for $\Delta t \approx 8/\Omega_0$.

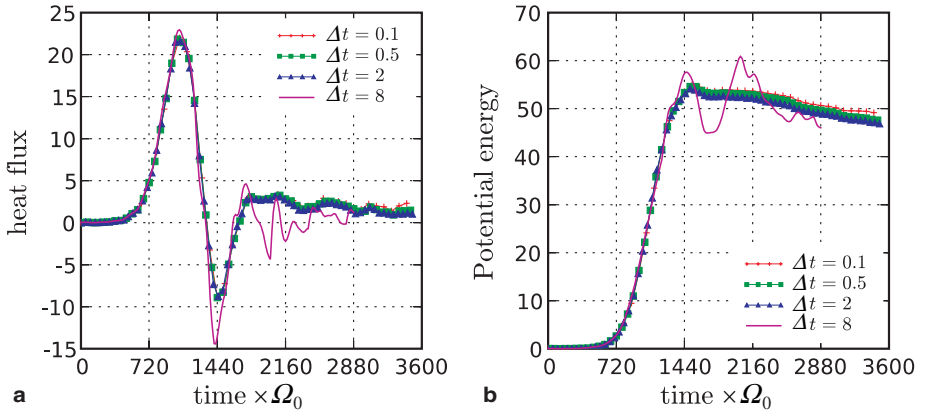


Fig. 12. Time evolution of (a) the heat flux Q at the radial position $r = r_{\mathrm{peak}}$ and (b) the potential energy, for four different time steps $\Delta t = 0.1/\Omega_0$, $\Delta t = 0.5/\Omega_0$, $\Delta t = 2/\Omega_0$ and $\Delta t = 8/\Omega_0$.

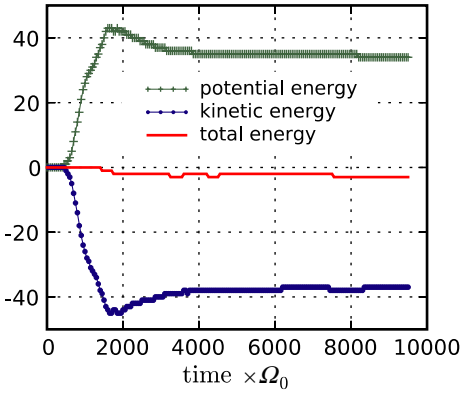


Fig. 13. Time evolution of the kinetic (dashed line), potential (dotted line) and total (solid line) energies for a mesh of $(128 \times 256 \times 128 \times 64)$ in $(r, \theta, z, v_\parallel)$ directions and a time step $\Delta t = 0.5/\Omega_0$.

5.3.3. A robust scheme

The GYSELA code is not only able to conserve the total energy with an error smaller than 2% (for $\Delta t = 0.1/\Omega_0$) but permits also to simulate turbulence well into the non-linear regime. In Fig. 13 is presented the evolution of the kinetic, potential and total energies up to $t = 9500/\Omega_0$ for the refined mesh of $(128 \times 256 \times 128 \times 64)$ in $(r, \theta, z, v_\parallel)$ directions and a time step $\Delta t = 0.5/\Omega_0$.

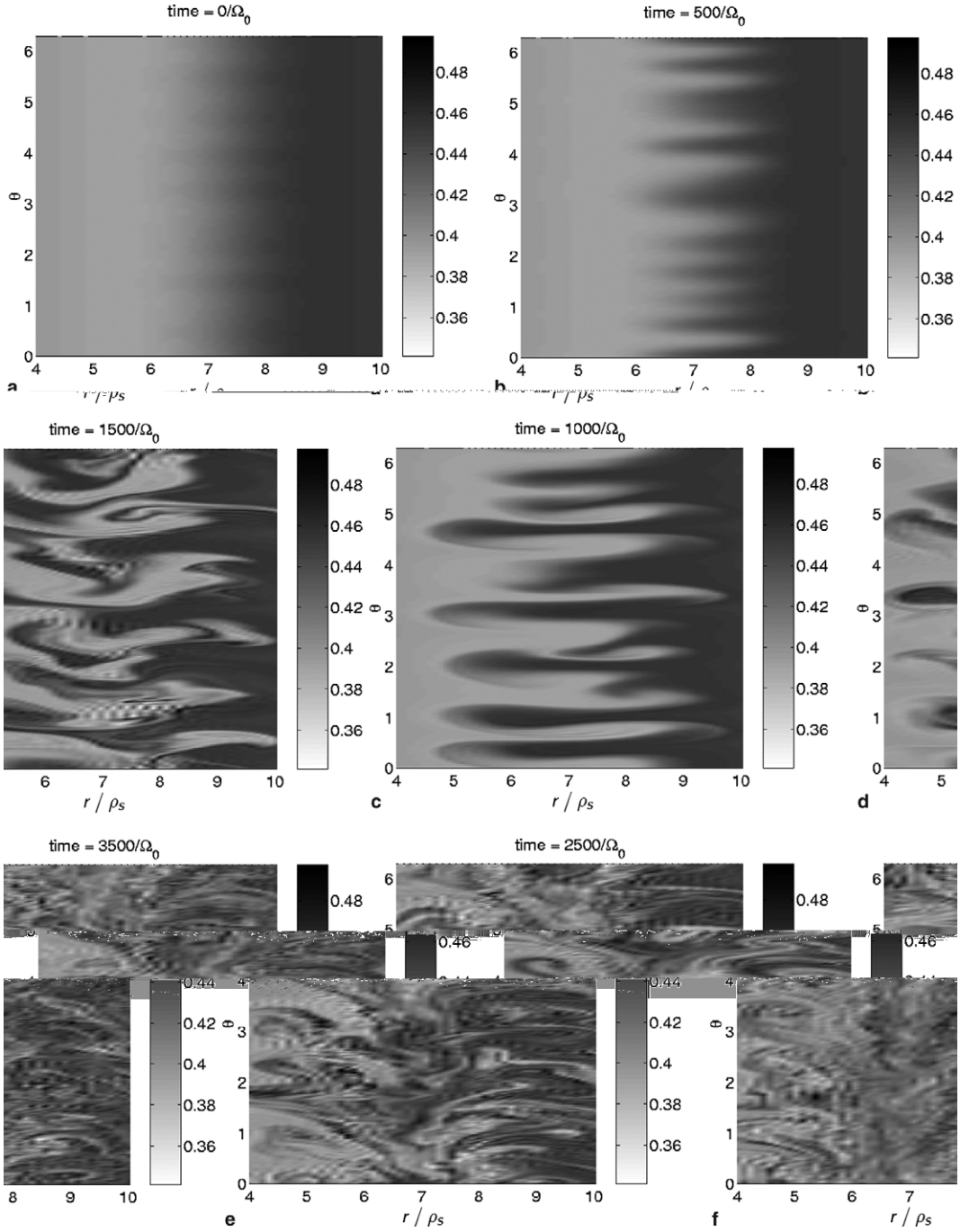


Fig. 14. (r, θ) cross-section of the distribution function at 6 different times: (a) initial time, (b) middle of the linear phase, (c) end of the linear phase, (d) beginning of the non-linear phase, (e, f) two times in the non-linear phase.

Another characteristic of this code is that it directly computes the evolution of the total distribution function f. The evolution of the distribution function in a (r,θ)-cross-section is represented in Fig. 14.

6. Improved schemes

6.1. Positivity of the distribution function

The semi-Lagrangian method does not ensure that the distribution function will remain positive. Indeed, negative values of the distribution function can appear in the non-linear phase. In the present runs, the domain where the distribution function is found negative does not increase indefinitely. It is of the order of 5–6% of the whole phase space all along the simulation. The apparition of these negative values is due to the interpolation step, and therefore depends on the refinement of the mesh. The first negative values appear at time $t = 1190/\Omega_0$ for a mesh of $(64 \times 64 \times 64 \times 64)$ and at time $t = 1450/\Omega_0$ for a mesh of $(128 \times 256 \times 128 \times 64)$. Besides, the first negative values appear for a parallel velocity of 6 times the thermal velocity ($v_{\parallel} = 6v_{T_i}$) which corresponds to the region where the Maxwellian function is close to 0. Then, and whatever the mesh, the negative values do not propagate beyond $v_{\parallel} = 2.5v_{T_i}$ (cf. Fig. 15). Therefore, the existence of negative values does not seem to be a real problem in the cases we have run since the distribution function remains positive in the region of physical importance.

One can think of typically two classes of solutions to get rid of these negative values, in case they would become a serious limitation of the code. The first one would be to include collisions in the problem. This would

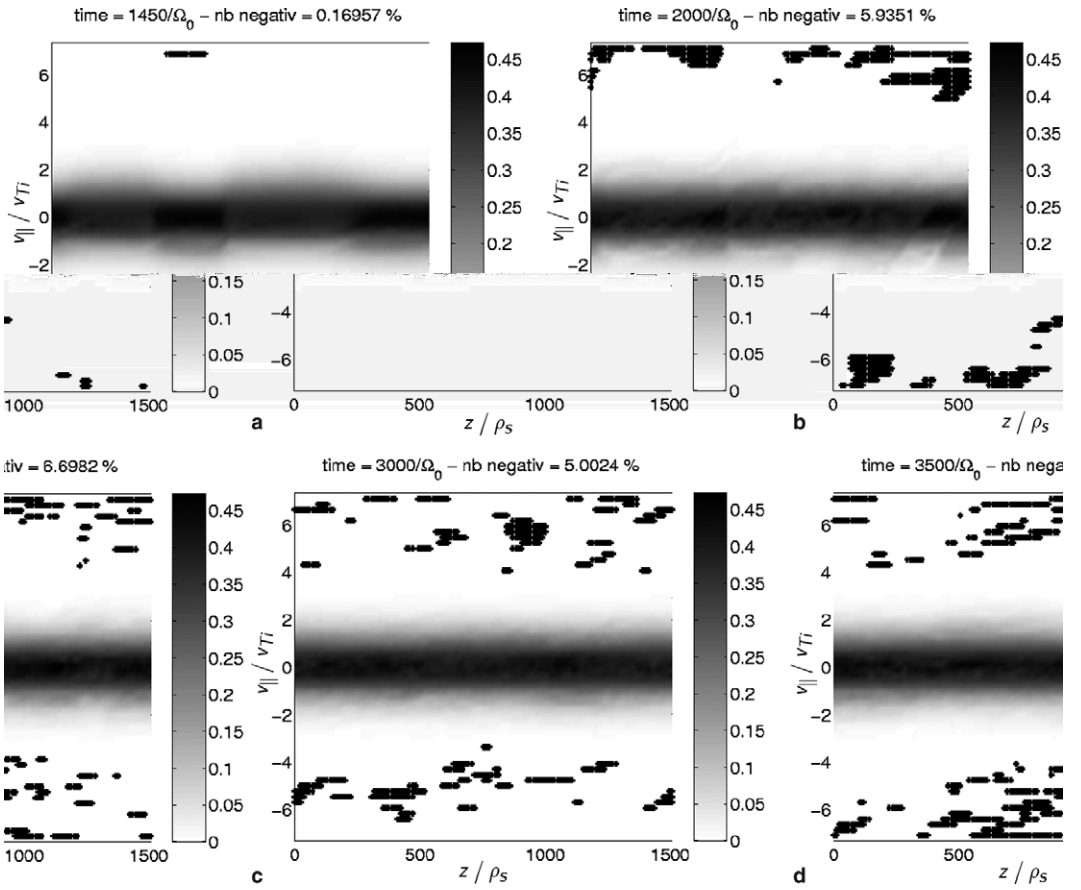


Fig. 15. $(z, v_{\parallel})$ cross-section of the distribution function at four different times: (a) $t = 1450/\Omega_0$, (b) $t = 2000/\Omega_0$, (c) $t = 3000/\Omega_0$ and (d) $t = 3500/\Omega_0$. Cross-points represent negative values of the distribution function.

dissipate small scales in the velocity space. In this case, one expects that the cubic spline interpolation should not generate spurious negative values of the distribution function. Such a collisional operator will be incorporated in the future. The second one is based on alternative numerical schemes. Two ideas have been tested on the present 4D model. The first possibility is to use a positive flux conservative (PFC) method [24] based on a finite volume method, that ensures by construction the conservation of the number of particles and preserves the positivity via a good choice of slope correctors. In this case, as since the equations are solved in their conservative form (finite volume principle), the 4D Vlasov equation (2) is replaced by the solution of the four following equations:

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial r}(v_{\mathrm{GC}_r} f) = 0 \tag{18}$$

$$\frac{\partial f}{\partial t} + \frac{1}{r}\frac{\partial}{\partial \theta}(v_{\mathrm{GC}_\theta} f) = 0 \tag{19}$$

$$\frac{\partial f}{\partial t} + \frac{\partial (v_\parallel f)}{\partial z} = 0 \tag{20}$$

$$\frac{\partial f}{\partial t} + \frac{\partial (\dot{v}_\parallel f)}{\partial v_\parallel} = 0 \tag{21}$$

The associated time-splitting scheme over two time steps is defined symmetrically by the sequence $(\bar{v}_\parallel, \bar{z}, \bar{r}, 2\bar{\theta}, \bar{r}, \bar{z}, \bar{v}_\parallel)$ where $\bar{r}$, $\bar{\theta}$, $\bar{z}$ and $\bar{v}_\parallel$ represent the operators, respectively, associated with Eqs. (18)–(21). The drawback of this PFC method is that it is dissipative and leads to a loss of conservation of the total energy. A less dissipative solution which simply consists in replacing the cubic spline interpolation of the distribution function by the cubic spline interpolation of the logarithm of this distribution function is developed and compared to the PFC method in [28] for the 2D standard Landau damping case. However it requires an increase of the mesh refinement to treat correctly the small scales [42]. Given that a simulation with a mesh of $(128 \times 256 \times 128 \times 64)$ needs more than 7 GBytes of memory and 25 s of CPU time for each iteration on 128 processors, more refined meshes are difficult to use given the present resources. The use of a non-equidistant mesh can reduce the number of points that are needed by a factor 2 [42].

6.2. Treatment of small scales

As discussed in the linear study (Fig. 8), all the poloidal modes ($m > 10$) are unstable. A refinement of the mesh in the poloidal direction will thus improve the treatment of the small scales but will not be sufficient. A way to limit this filamentation (cf. Fig. 14) is to cut off the small scales. This can be done by adding a numerical dissipation (white noise for instance). The drawback of this kind of dissipation is that its impact during the non-linear regime is difficult to control. For the moment a filter on the electric potential Φ is used in GYSELA, where all the poloidal Fourier modes larger than 16 are artificially set to 0. This numerical filter is temporary and will be replaced in the next version by a physical filter which consists in taking into account the finite Larmor radius effects. In this case, the numerical model will be governed by the following gyroaveraged equations:

$$\frac{\partial \bar{f}}{\partial t} + \frac{1}{rB}\frac{\partial (J_0\Phi)}{\partial r}\frac{\partial \bar{f}}{\partial \theta} - \frac{1}{rB}\frac{\partial (J_0\Phi)}{\partial \theta}\frac{\partial \bar{f}}{\partial r} + v_\parallel \frac{\partial \bar{f}}{\partial z} - \frac{q}{m_\mathrm{i}}\frac{\partial (J_0\Phi)}{\partial z}\frac{\partial \bar{f}}{\partial v_\parallel} = 0$$

$$-\nabla_\perp \cdot \left[\frac{n_0(r)}{B\Omega_0}\nabla_\perp \Phi\right] + \frac{e n_0(r)}{T_\mathrm{e}(r)}(\Phi - \langle \Phi \rangle_z) = 2\pi \int v_\perp \,\mathrm{d}v_\perp \int \mathrm{d}v_\parallel J_0 \cdot \bar{f} - J_0 n_0$$

where J_0 is an operator that takes the form of multiplication by the Bessel function of first order $J_0(\frac{k_\perp v_\perp}{\Omega})$ in the Fourier space. $\bar{f} = f_\mathrm{eq} + \delta\bar{f}$ is the gyroaveraged distribution function where $\delta\bar{f}$ is the perturbed part and where the equilibrium function f_eq is now defined by the following local Maxwellian:

$$f_\mathrm{eq} = \frac{n_0(r)}{(2\pi T_\mathrm{i}(r)/m_\mathrm{i})^{\frac{3}{2}}} \exp\left[-\frac{m_\mathrm{i}(v_\parallel^2 + v_\perp^2)}{2T_\mathrm{i}(r)}\right]$$

chosen such that $\int_0^\infty 2\pi v_\perp \,\mathrm{d}v_\perp \int_{-\infty}^{+\infty} \mathrm{d}v_\parallel f_\mathrm{eq} = n_0$.

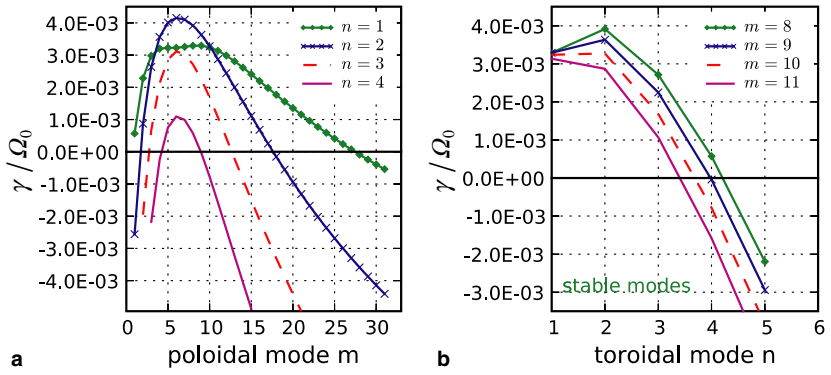


Fig. 16. Linear growth rate γ by taking into account the gyroaverage effects: (a) versus poloidal Fourier modes m for four different values of toroidal Fourier modes $n = 1, 2, 3, 4$; (b) versus toroidal Fourier modes n for fourth of the most unstable poloidal Fourier modes $m = 8, 9, 10, 11$.

In the limit $\rho_s \ll L_{n_i}, L_{T_i}$ the linearized dispersion relation associated to this 5D model is given by

$$\frac{\tau}{\Gamma_0(b)}\left[\rho_s^2\left(\kappa(r)+\frac{m^2}{r^2}\right)+\frac{1}{Z_i}\right]+1-z^2\frac{\omega^*_{T_i}}{\omega}+zZ(z)\left[1-\frac{\omega^*_{n_i}}{\omega}-\frac{\omega^*_{T_i}}{\omega}\left(z^2-\frac{1}{\eta_{i0}(b)}\right)\right]=0$$

with $b = k_\perp^2\rho_i^2/2$, $\eta_{i0}(b) = 2\Gamma_0(b)/[\Gamma_0(b) + 2b(\Gamma_0(b) - \Gamma_1(b))]$ and $\Gamma_j(b) = \exp(-b)I_j(b)$ where I_0 (respectively, I_1) represents the modified Bessel function of first (respectively, second) kind. Then the gyroaverage effects on the small scales can already be seen by comparing Fig. 8 to Fig. 16. In this case the high m modes are stable so that a fixed mesh size will be appropriate without the use of filtering.

As far as the velocity space is concerned, small scales are present in the bulk of the distribution function, although they are not highlighted in Fig. 15. However, the cubic spline interpolation generates some dissipation in the velocity space, which appears to be sufficient to damp the smallest scales. As a result, the distribution function does not exhibit strong filamentation features in velocity, in contrast with those reported in Ref. [43].

7. Conclusion

A new gyrokinetic 4D code, named GYSELA, has been developed to compute ion temperature gradient driven turbulence in a cylinder. This code uses a semi-Lagrangian numerical scheme, which is second order accurate in time. The Vlasov equation is solved with a time splitting of the advection, thus allowing an efficient parallelization. The GYSELA code has been validated in the linear phase by comparing the calculated growth rates and eigenmodes to the analytical values. Simulations in non-linear regime have been benchmarked against the ORB5 gyrokinetic code [42]. Also energy conservation is found to be respected provided small spatial scales are filtered. The massively parallel GYSELA code is found to be stable over long simulation times, even for high spatial resolution. This result is very promising. Three limitations have been encountered. First it is found that the speed-up performance saturates for a large number of processors. This saturation comes from the implementation of the quasi-neutrality condition, which is not parallelized. Second the error on energy conservation, although satisfactory, is consistent with an accuracy that ranges between expectations for first and second order accuracy. Third, the distribution function exhibits negative values in the domain of high velocities, i.e., in a domain where the equilibrium distribution function vanishes. The latter drawback originates from the development of small scales in the velocity space. It has been found to be quite benign in the simulations presented in this paper, since it only affects a very limited part of the phase space. In the future, a collisional operator should be added. In this case, small velocity scales should be dissipated and the specific problem of negative values should be overcome. Furthermore, the code will be subject to other improvements. Finite Larmor radius effects will be implemented, so that numerical filtering of small scales should no longer be necessary. The quasi-neutrality

condition will be parallelized, to improve the speed-up performance when the number of processors is large. Finally toroidal geometry and poloidal magnetic field will be implemented, in order to simulate a toroidal ion turbulence in a tokamak.

Acknowledgment

The authors thank C. Passeron for her constant help in the development of the code.

Appendix A. Expression of the energy conservation law

The kinetic energy is given by definition by

$$\delta\varepsilon_{\mathrm{kin}} = \int m_{\mathrm{i}} \frac{v_{\parallel}^2}{2} (f - f_M)\, \mathrm{d}V\, \mathrm{d}v_{\parallel} \quad \text{with } \mathrm{d}V = r\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}z$$

while the corresponding potential energy is deduced from the expression of the Vlasov equation (2) and the quasi-neutrality Eq. (3). The computation of the potential energy is presented in this appendix.

The 4D Vlasov equation (2) can be written in its Hamiltonian form as

$$\frac{\partial f}{\partial t} - [H, f] = 0 \tag{22}$$

where H is the Hamiltonian of the guiding center trajectories

$$H = \frac{1}{2} m v_{\parallel}^2 + e\Phi(r, \theta, z) \tag{23}$$

and where the symbol [.,.] represents the generalized Poisson brackets with the generalized Poisson operator corresponding to

$$\begin{pmatrix} \frac{1}{eB_z} & 0 & 0 & 0 \\ 0 & -\frac{1}{eB_z} & 0 & 0 \\ 0 & 0 & -\frac{1}{m} & 0 \\ 0 & 0 & 0 & \frac{1}{m} \end{pmatrix}$$

So let Eq. (22) be multiplied by the Hamiltonian H and integrated on the phase space, then

$$\int \frac{\partial f}{\partial t} H\, \mathrm{d}V\, \mathrm{d}v_{\parallel} - \int H[H, f]\, \mathrm{d}V\, \mathrm{d}v_{\parallel} = 0$$

which, according to the Hamiltonian expression Eq. (23), yields

$$\int \frac{1}{2} m v_{\parallel}^2 \frac{\partial f}{\partial t} H\, \mathrm{d}V\, \mathrm{d}v_{\parallel} + \int e\Phi \frac{\partial f}{\partial t}\, \mathrm{d}V\, \mathrm{d}v_{\parallel} = \int H[H, f]\, \mathrm{d}V\, \mathrm{d}v_{\parallel} = \int [H, H] f\, \mathrm{d}V\, \mathrm{d}v_{\parallel} = 0$$

So using $\int f\, \mathrm{d}v_{\parallel} = n_{\mathrm{i}}$, for the density of guiding centers

$$\frac{\partial}{\partial t} \int \frac{1}{2} m v_{\parallel}^2 f\, \mathrm{d}V\, \mathrm{d}v_{\parallel} = -\int e\Phi \frac{\partial f}{\partial t}\, \mathrm{d}V\, \mathrm{d}v_{\parallel} = -e \int \Phi \frac{\partial n_{\mathrm{i}}}{\partial t}\, \mathrm{d}V$$

and replacing n_{i} in the previous equation by its expression deduced from the quasi-neutrality equation

$$n_{\mathrm{i}} = n_0 - \nabla_{\perp} \cdot \left[\frac{n_0(r)}{B\Omega_0} \nabla_{\perp} \Phi \right] + \frac{e n_0(r)}{T_{\mathrm{e}}(r)} (\Phi - \langle \Phi \rangle)$$

then

$$\frac{\partial \delta\varepsilon_{\mathrm{kin}}}{\partial t} = \frac{e}{B\Omega_0} \gamma_2 - e^2 \gamma_3$$

with

$$\gamma_2 = \int \Phi \frac{\partial}{\partial t}(\nabla_\perp \cdot [n_0(r)\nabla_\perp \Phi])\,\mathrm{d}V \quad \text{and} \quad \gamma_3 = \int \frac{n_0(r)}{T_e(r)}\Phi \frac{\partial}{\partial t}(\Phi - \langle\Phi\rangle)\,\mathrm{d}V$$

Integrating by part and taking into account the fact that the electric potential Φ is equal to 0 on the boundaries in the radial direction and periodic in the θ and z directions, then the expressions of γ_2 and γ_3 become

$$\gamma_2 = \frac{1}{2}\frac{\partial}{\partial t}\int \Phi\nabla_\perp \cdot [n_0(r)\nabla_\perp \Phi]\,\mathrm{d}V \quad \text{and} \quad \gamma_3 = \frac{1}{2}\frac{\partial}{\partial t}\int \frac{n_0(r)}{T_e(r)}\Phi(\Phi - \langle\Phi\rangle)\,\mathrm{d}V$$

so that the evolution of the kinetic energy is given by

$$\begin{aligned}\frac{\partial \delta\varepsilon_{\mathrm{kin}}}{\partial t} &= \frac{1}{2}\frac{\partial}{\partial t}\int e\Phi\left\{\frac{en_0(r)}{T_e(r)}(\Phi - \langle\Phi\rangle) - \nabla_\perp \cdot \left[\frac{n_0(r)}{B\Omega_0}\nabla_\perp \Phi\right]\right\}\mathrm{d}V \\ &= \frac{1}{2}\frac{\partial}{\partial t}\int e\Phi(n_i - n_0)\,\mathrm{d}V \quad \text{according to the quasi-neutrality equation.}\end{aligned}$$

And finally the previous equation can be expressed as $\frac{\partial \delta\varepsilon_{\mathrm{kin}}}{\partial t} + \frac{\partial \delta\varepsilon_{\mathrm{pot}}}{\partial t} = 0$ where the potential energy is defined as

$$\delta\varepsilon_{\mathrm{pot}} = \frac{1}{2}\int e\Phi(n_i - n_0)\,\mathrm{d}V$$

and satisfies the energy conservation law $\delta\varepsilon_{\mathrm{kin}} + \delta\varepsilon_{\mathrm{pot}} = \text{constant}$.

Appendix B. Time-splitting scheme of second order in time

As explained in the paper, the Vlasov equation (2) is solved with the following time-splitting scheme:

$$(\hat{v}_\parallel/2, \hat{z}/2, \widehat{r\theta}/2, \widehat{r\theta}/2, \hat{z}/2, \hat{v}_\parallel/2) \tag{24}$$

(where the coefficients 1/2 correspond to shifts on $\Delta t/2$). $\widehat{r\theta}$ denotes the shift operator in the (r,θ) plane in Δt, associated to the advection term in Eq. (9). Similarly, $\hat{z}$ and $\hat{v}_\parallel$ denote the shift operators, respectively, in the z (Eq. (10)) and $v_\parallel$ directions (Eq. (11)). In this appendix, a formal proof is presented, showing that such a choice in the sequence of the time-splitting ensures that the numerical scheme is of second order in time. This result is true regardless of the fact that the operators commute or not.

Formal proof: Formally, the Vlasov equation (2) can be written as follows:

$$\frac{\partial f}{\partial t} + (A + B + C)f = 0 \tag{25}$$

where A, B and C are the formal operators, respectively, defined by $A = \vec{v}_{\mathrm{GC}}\vec{\nabla}\cdot$, $B = v_\parallel \frac{\partial \cdot}{\partial z}$ and $C = \dot{v}_\parallel \frac{\partial \cdot}{\partial v_\parallel}$. The solutions of Eq. (25) are given by

$$f = f_0 \exp[(-A + B + C)t] \tag{26}$$

The formal expression associated to the time-splitting scheme Eq. (24) reads as follows:

$$f = f_0 \exp\left(-\frac{At}{2}\right)\exp\left(-\frac{Bt}{2}\right)\exp(-Ct)\exp\left(-\frac{Bt}{2}\right)\exp\left(-\frac{At}{2}\right) \tag{27}$$

In the following, we will prove that both expressions, Eqs. (26) and (27), are equivalent at second order.

Computation of $I = \exp[(-A + B + C)t]$:

The Taylor expansion at second order of $\exp[(-A + B + C)t]$ yields

$$\begin{aligned}I = \exp[(-A + B + C)t] &= 1 - (A + B + C)t + \frac{(A + B + C)^2}{2}t^2 + \mathcal{O}(t^3) \\ &= 1 - (A + B + C)t + \frac{t^2}{2}(A^2 + B^2 + C^2 + AB + BA + AC + CA + BC + CB) + \mathcal{O}(t^3)\end{aligned} \tag{28}$$

Computation of $J = \exp(-\frac{At}{2})\exp(-\frac{Bt}{2})\exp(-Ct)\exp(-\frac{Bt}{2})\exp(-\frac{At}{2})$:
Similarly,

$$\exp\left(-\frac{Bt}{2}\right)\exp\left(-\frac{At}{2}\right) = \left(1-\frac{Bt}{2}+\frac{B^2t^2}{8}+\mathcal{O}(t^3)\right)\left(1-\frac{At}{2}+\frac{A^2t^2}{8}+\mathcal{O}(t^3)\right)$$
$$= \left[1-\left(\frac{B}{2}+\frac{A}{2}\right)t+t^2\left(\frac{B^2}{8}+\frac{A^2}{8}+\frac{BA}{4}\right)\right]+\mathcal{O}(t^3)$$

where

$$J_1 = \exp(-Ct)\exp\left(-\frac{Bt}{2}\right)\exp\left(-\frac{At}{2}\right)$$
$$= \left(1-Ct+\frac{C^2t^2}{2}\right)\times\left[1-\left(\frac{B}{2}+\frac{A}{2}\right)t+t^2\left(\frac{B^2}{8}+\frac{A^2}{8}+\frac{BA}{4}\right)\right]+\mathcal{O}(t^3)$$
$$= \left[1-\left(\frac{A}{2}+B+C\right)t+t^2\left(\frac{C^2}{2}+\frac{B^2}{8}+\frac{A^2}{8}+\frac{BA}{4}+\frac{CA}{2}+\frac{CB}{2}\right)\right]+\mathcal{O}(t^3)$$

Hence

$$J_2 = \exp\left(-\frac{Bt}{2}\right)\times J_1$$
$$= \left(1-\frac{Bt}{2}+\frac{B^2t^2}{8}\right)\times\left[1-\left(\frac{A}{2}+B+C\right)t+t^2\left(\frac{C^2}{2}+\frac{B^2}{8}+\frac{A^2}{8}+\frac{BA}{4}+\frac{CA}{2}+\frac{CB}{2}\right)\right]$$
$$= \left[1-\left(\frac{A}{2}+B+C\right)t+t^2\left(\frac{C^2}{2}+\frac{B^2}{2}+\frac{A^2}{8}+\frac{BA}{2}+\frac{CA}{2}+\frac{CB}{2}+\frac{BC}{2}\right)\right]+\mathcal{O}(t^3)$$

and

$$J = \exp\left(-\frac{At}{2}\right)\times J_2$$
$$= \left(1-\frac{At}{2}+\frac{A^2t^2}{8}\right)\times\left[1-\left(\frac{A}{2}+B+C\right)t+t^2\left(\frac{C^2}{2}+\frac{B^2}{2}+\frac{A^2}{8}+\frac{BA}{2}+\frac{CA}{2}+\frac{CB}{2}+\frac{BC}{2}\right)\right]+\mathcal{O}(t^3)$$
$$= 1-(A+B+C)t+t^2\left(\frac{C^2}{2}+\frac{B^2}{2}+\frac{A^2}{4}+\frac{BA}{2}+\frac{CA}{2}+\frac{CB}{2}+\frac{BC}{2}+\frac{A^2}{4}+\frac{AB}{2}+\frac{AC}{2}\right)+\mathcal{O}(t^3)$$
$$= 1-(A+B+C)t+t^2\left(\frac{C^2}{2}+\frac{B^2}{2}+\frac{A^2}{2}+\frac{AB+BA}{2}+\frac{AC+CA}{2}+\frac{BC+CB}{2}\right)+\mathcal{O}(t^3) \quad (29)$$

This shows that Eq. (29) is equivalent to Eq. (28), which proves that Eqs. (26) and (27) are equivalent at second order. □

In conclusion, the time-splitting scheme Eq. (24) used in our global algorithm is of second order in time.

Appendix C. Cubic spline interpolation

The 4D Vlasov equation is solved by splitting in (r,θ), z and $v_\parallel$ directions. This requires interpolations of the distribution function $f(r,\theta,z,v_\parallel)$, 1D interpolations in the z and $v_\parallel$ directions and 2D interpolations in the (r,θ) plane. The description of these interpolations by using cubic splines and by taking into account the boundary conditions which are periodic in θ and z, and non-periodic in r and $v_\parallel$ directions is addressed here.

C.1. Cubic spline interpolation in 1D

Let $g(x)$ be a function defined in the x-direction with $x \in [x_0, x_{N_x}]$ where N_x represents the number of points in x (the step h being constant). Using a cubic spline for the interpolation of g consists in representing this function in terms of piecewise cubic polynomials Λ_α, twice continuously differentiable [35] as

$$g(x) = \sum_{\alpha=-1}^{N_x+1} c_\alpha \Lambda_\alpha(x)$$

where

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

with $h = |x_{N_x} - x_0|/N_x$.

Then the c_α coefficients are computed like the solution of the following system of equations:

$$g(x_i) = \sum_{\alpha=-1}^{N_x+1} c_\alpha \Lambda_\alpha(x_i), \quad i = 0, \ldots, N_x$$

This system contains $(N_x + 1)$ equations and $(N_x + 3)$ unknowns, so 2 other equations which depend on the boundary conditions are mandatory. According to Table 4, the $(N_x + 3, N_x + 3)$ matrix system to be solved becomes

$$\tilde{A}\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix} \quad \text{with} \begin{cases} u = (c_0, \ldots, c_{N_x})^t \\ v = (c_{N_x+1}, c_{-1})^t \\ b = (g(x_0), \ldots, g(x_{N_x}))^t \\ c = (\sigma_1, \sigma_2)^t \end{cases} \quad \text{and} \quad \tilde{A} = \begin{pmatrix} A & \gamma \\ \lambda & \delta \end{pmatrix}$$

where

$$\begin{cases} A \text{ is the } (N_r + 1) \times (N_r + 1) \text{ tridiagonal symmetric matrix: } \begin{pmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{pmatrix}, \\ \lambda \text{ is equal to the } 2 \times (N_r + 1) \text{ matrix: } \begin{pmatrix} 0 & \cdots & 0 & -3/h & 0 \\ 0 & 3/h & 0 & \cdots & 0 \end{pmatrix}, \\ \gamma \text{ is equal to the } (N_r + 1) \times 2 \text{ matrix: } \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}^t \text{ and} \\ \delta = \begin{pmatrix} \xi_1 & \xi_2 \\ \xi_3 & \xi_4 \end{pmatrix} = \begin{pmatrix} 3/h & 0 \\ 0 & -3/h \end{pmatrix} \end{cases}$$

The terms of the matrices c, λ, γ and δ are modified according to the boundary conditions but the resolution of the system is always the same and takes advantage of the fact that $\tilde{A}$ can be factorized in a LU form, like:

Table 4
Values of the cubic spline function $\Lambda_\alpha(x)$ and its first and second derivative

x	$x_{\alpha-2}$	$x_{\alpha-1}$	x_α	$x_{\alpha+1}$	$x_{\alpha+2}$
$\Lambda_\alpha(x)$	0	1	4	1	0
$\Lambda'_\alpha(x)$	0	$3/h$	0	$-3/h$	0
$\Lambda''_\alpha(x)$	0	$6/h^2$	$-12/h^2$	$6/h^2$	0

$$\tilde{A} = \begin{pmatrix} A & 0 \\ \lambda & \bar{\delta} \end{pmatrix} \times \begin{pmatrix} I & A^{-1}\gamma \\ 0 & I \end{pmatrix} \quad \text{with } \bar{\delta} = \delta - \lambda A^{-1}\gamma$$

With this LU factorization the matrix system is solved by forward and backward substitutions with the two successive sequences

$$\begin{pmatrix} A & 0 \\ \lambda & \bar{\delta} \end{pmatrix} \times \begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix} \text{ and then } \begin{pmatrix} I & A^{-1}\gamma \\ 0 & I \end{pmatrix} \times \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u' \\ v' \end{pmatrix}$$

So computation of the interpolation coefficients $\vec{c} = (\, c_{-1} \quad c_0 \quad \cdots \quad c_{N_x} \quad c_{N_x+1} \,)$ can be summarized in the following steps:

1. *Initialization*:
   (a) Factorize and store A in a LDL^t form.
   (b) Compute and store $A^{-1}\gamma$ using the previous factorization.
   (c) Assemble the (2×2) matrix $\bar{\delta} = \delta - \lambda A^{-1}\gamma$.

2. *Time loop*:
   (a) Compute and store $u' = A^{-1}b$ using the stored factorization of A.
   (b) Assemble $c - \lambda A^{-1}b$.
   (c) Solve the (2×2) system $\bar{\delta}v' = c - \lambda A^{-1}b$ using the Cramer formula for $\bar{\delta}$ inverse computation $\bar{\delta}^{-1} = \frac{1}{\det(\bar{\delta})}\begin{pmatrix} \bar{\xi}_4 & -\bar{\xi}_2 \\ -\bar{\xi}_3 & \bar{\xi}_1 \end{pmatrix}$.
   (d) Compute u using the previous storage of $A^{-1}\gamma$ by $u = u' - A^{-1}\gamma v$, where v is trivially equal to v'.

C.1.1. Periodic boundary conditions

The 1D cubic spline interpolation with periodic boundary conditions is used for instance for the interpolation of the distribution function needed in the z direction. Then,

$$f(r_i, \theta_j, z, v_{\|l}) = g(z) = \sum_{\mu=-1}^{N_z+1} c_\mu \Lambda_\mu(z) \quad \forall r_i, \theta_j, v_{\|l} \text{ nodes of the mesh}$$

The two necessary equations are obtained by using the first and second derivative continuity property of the cubic splines:

$$\begin{cases} g'(z_0) = g'(z_n) \\ g''(z_0) = g''(z_n) \end{cases}$$

which gives by using Table 4:

$$\begin{cases} -\frac{3}{h}c_{-1} + \frac{3}{h}c_1 + \frac{3}{h}c_{N_r-1} - \frac{3}{h}c_{N_r+1} = 0 \\ \frac{6}{h^2}c_{-1} - \frac{12}{h^2}c_0 + \frac{6}{h^2}c_1 - \frac{6}{h^2}c_{N_r-1} + \frac{12}{h^2}c_{N_r} - \frac{6}{h^2}c_{N_r+1} = 0 \end{cases}$$

This two equations determines the missing terms in the previous matrix system as follows:

$$\begin{cases} c = (\sigma_1, \sigma_2)^t = (0,0)^t, \\ \lambda = \begin{pmatrix} 0 & 3/h & 0 & \cdots & 0 & 3/h & 0 \\ -12/h^2 & 6/h^2 & 0 & \cdots & 0 & -6/h^2 & 12/h^2 \end{pmatrix} \text{ and} \\ \delta = \begin{pmatrix} \xi_1 & \xi_2 \\ \xi_3 & \xi_4 \end{pmatrix} = \begin{pmatrix} -3/h & -3/h \\ -6/h^2 & 6/h^2 \end{pmatrix} \end{cases}$$

C.1.2. Non-periodic boundary conditions

The interpolation of the distribution function in the $v_\|$ direction is performed by using the following formula:

$$f(r_i,\theta_j,z_k,v_\parallel)=g(v_\parallel)=\sum_{v=-1}^{N_v+1}c_v\Lambda_v(v_\parallel)\quad\forall r_i,\theta_j,z_k \text{ on the mesh}$$

The first derivatives of f are approximated at the boundaries $v_{\parallel 0}$ and $v_{\parallel\ N_v}$ ($v_\parallel\in[v_{\parallel 0},v_{\parallel N_v}]$) by a cubic Lagrange polynomials fit of f and defined as:

$$\lambda_{(3)}(v_\parallel)=\sum_{l=0}^{3}g(v_\parallel)L_l(v_\parallel)\quad\text{with } L_l(v_\parallel)=\prod_{j=0,j\neq l}^{3}\left(\frac{v_\parallel-v_{\parallel j}}{v_{\parallel l}-v_{\parallel j}}\right)$$

Then the two boundary equations are:

$$\begin{cases}\lambda'_{(3)}(v_{\parallel 0})=\sum_{v=-1}^{N_v+1}c_v\Lambda'_v(v_{\parallel 0})\\ \lambda'_{(3)}(v_{\parallel N_v})=\sum_{v=-1}^{N_v+1}c_v\Lambda'_v(v_{\parallel N_v})\end{cases}$$

which is equivalent to:

$$\begin{cases}-\frac{3}{h}c_{-1}+\frac{3}{h}c_1=-\frac{11}{6h}g(v_{\parallel 0})+\frac{3}{h}g(v_{\parallel 1})-\frac{3}{2h}g(v_{\parallel 2})+\frac{1}{3h}g(v_{\parallel 3})\\ -\frac{3}{h}c_{Nv-1}+\frac{3}{h}c_{Nv+1}=-\frac{1}{3h}g(v_{\parallel N_v-3})+\frac{3}{2h}g(v_{\parallel N_v-2})-\frac{3}{h}g(v_{\parallel N_v-1})+\frac{11}{6h}g(v_{\parallel N_v})\end{cases}$$

such that

$$\begin{cases}\sigma_1=-\frac{1}{9}g(v_{\parallel N_v-3})+\frac{1}{2}g(v_{\parallel N_v-2})-g(v_{\parallel N_v-1})+\frac{11}{18}g(v_{\parallel N_v}),\\ \sigma_2=-\frac{11}{18}g(v_{\parallel 0})+g(v_{\parallel 1})-\frac{1}{2}g(v_{\parallel 2})+\frac{1}{9}g(v_{\parallel 3}),\\ \lambda=\begin{pmatrix}0&\cdots&0&-1&0\\0&1&0&\cdots&0\end{pmatrix}\text{ and}\\ \delta=\begin{pmatrix}\xi_1&\xi_2\\\xi_3&\xi_4\end{pmatrix}=\begin{pmatrix}1&0\\0&-1\end{pmatrix}\end{cases}$$

C.2. Cubic spline interpolation in 2D

The 2D advection requires the interpolation of f in the (r,θ) plane. In this case, a cubic B-spline interpolation method is used for all $r\in[r_0,r_{N_r}]$ and $\theta\in[\theta_0,\theta_{N_\theta}]$

$$f(r,\theta,z_k,v_{\parallel l})=g(r,\theta)=\sum_{\alpha=-1}^{N_r+1}\sum_{\beta=-1}^{N_\theta+1}c(\alpha,\beta)\Lambda_\alpha(r)\Lambda_\beta(\theta)\quad\forall z_k,v_{\parallel l} \text{ on the mesh}$$

The interpolation coefficients can be computed by solving the $(N_r+3)\times(N_\theta+3)$ linear system:

$$g(r_i,\theta_j)=\sum_{\alpha=-1}^{N_r+1}\gamma(\alpha,j)\Lambda_\alpha(r_i)\text{ where }\gamma(\alpha,j)=\sum_{\beta=-1}^{N_\theta+1}c(\alpha,\beta)\Lambda_\beta(\theta_j)$$

Then for each j between 0 and N_θ, an unidimensional interpolation problem with non-periodic boundary conditions has to be solved (same resolution than in the $v_\parallel$ direction previously described). The second step consists in the resolution for each r_i ($i=-1,\ldots,N_r+1$) of an unidimensional periodic system given by

$$\sum_{\beta=-1}^{N_\theta+1}c(\alpha,\beta)\Lambda_\beta(\theta_j)=\gamma(\alpha,j)\quad\forall j\in[0,N_\theta]$$

In summary, the computation of the 2D interpolation coefficients is equivalent to:

- N_θ resolutions of the 1D non-periodic interpolation problem and
- N_r+3 resolutions of the 1D periodic interpolation problem.

References

[1] A.M. Dimits et al., Comparisons and physics basis of tokamak transport models and turbulence simulations, Phys. Plasmas 7 (3) (2000) 969–983.
[2] G.W. Hammett, F.W. Perkins, Fluid models for Landau damping with application to the ion-temperature-gradient instability, Phys. Rev. Lett. 64 (1990) 3019–3022.
[3] M.A. Beer, Gyrofluid Models of Turbulent Transport in Tokamaks, Ph.D. Thesis, Princeton University, 1995.
[4] R.E. Waltz, G.M. Staebler, W. Dorland, et al., A gyro-Landau-fluid transport model, Phys. Plasmas 4 (1997) 2482–2496.
[5] W.W. Lee, Gyrokinetic approach in particle simulation, Phys. Fluids 26 (2) (1983) 556.
[6] C.C. Kim, S.E. Parker, Massively Parallel three dimensional Toroidal gyrokinetic Flux-Tube turbulence Simulations, J. Comput. Phys. 161 (2) (2000) 589–604.
[7] Z. Lin, T.S. Hahm, W.W. Lee, W.M. Tang, R.B. White, Gyrokinetic simulations in general geometry and applications to collisional damping of zonal flows, Phys. Plasmas 7 (5) (2000) 1857–1862.
[8] T.M. Tran, K. Appert, M. Fivaz, G. Jost, J. Vaclavik, L. Villard, Global gyrokinetic simulation of Ion-Temperature-Gradient driven instabilities, Th. Fusion Plasmas, in: Proceedings of the International Workshop, Varenna, 1998, Ed. Compositori, Bologna, 1999, pp. 45–49.
[9] A. Bottino, T.M. Tran, O. Sauter, J. Vaclavik, L. Villard, Linear gyrokinetic simulations using particles for small perpendicular wavelength perturbations, Th. Fusion Plasmas, in: Proceedings of the International Workshop, Varenna, 2000, Ed. Compositori, Bologna, 2001, pp. 327–332.
[10] 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 (2003) 234–243.
[11] H. Sugama, T.H. Watanabe, W. Horton, Comparison between kinetic and fluid simulations of slab ion temperature gradient driven turbulence, Phys. Plasmas 10 (3) (2003) 726–736.
[12] C.K. Birdsall, A.B. Langdon, Plasma Physics via Computer Simulation, McGraw-Hill, New York, 1985.
[13] R. Hatzky, T.M. Tran, A. Könies, et al., Energy conservation in a nonlinear gyrokinetic Particle-In-Cell code for ion-temperature-gradient-driven modes in theta-pinch geometry, Phys. Plasmas 9 (3) (2002) 898–912.
[14] S.J. Allfrey, R. Hatzky, A revised delta f algorithm for nonlinear PIC simulations, Comp. Phys. Commun. 154 (2) (2003) 98–104.
[15] L. Villard, S.J. Allfrey, A. Bottino, M. Brunetti, G.L. Falchetto, V. Grandgirard, R. Hatzky, J. Nührenberg, A.G. Peeters, O. Sauter, S. Sorge, J. Vaclavik, Full radius linear and non-linear gyrokinetic simulations for Tokamaks and Stellerators: zonal flows, applied $E \times B$ flows, trapped electrons and finite beta, Nucl. Fusion 1 (2004) 172–180.
[16] E. Pohn, M. Shoucri, G. Kamelander, Eulerian Vlasov codes, Comp. Phys. Commun. 166 (2) (2005) 81–93.
[17] A. Ghizzo, P. Bertrand, E. Fijalkow, M.R. Feix, M. Shoucri, An Eulerian code for the study of drift-kinetic Vlasov equation, J. Comput. Phys. 108 (1) (1993) 105–121.
[18] G. Manfredi, M. Shoucri, R.O. Dendy, A. Ghizzo, P. Bertrand, Vlasov gyrokinetic simulations of ion-temperature-gradient driven instabilities, Phys. Plasmas 3 (1) (1996) 202–217.
[19] W. Dorland, F. Jenko, M. Kotschenreuther, B.N. Rogers, Electron temperature gradient turbulence, Phys. Rev. Lett. 85 (2000) 5579–5582.
[20] F. Jenko, W. Dorland, M. Kotschenreuther, B.N. Rogers, Massively parallel Vlasov simulation of electromagnetic drift-wave turbulence, Comp. Phys. Commun. 125 (2000) 196–209.
[21] M. Shoucri, Numerical simulation of plasma edge turbulence due to $E \times B$ flow velocity shear, Czech. J. Phys. 51 (10) (2001) 1139–1151.
[22] J. Candy, R.E. Waltz, An Eulerian gyrokinetic-Maxwell solver, J. Comput. Phys. 186 (2) (2003) 545–581.
[23] E. Fijalkow, A numerical solution of the Vlasov equation, Comp. Phys. Commun. 116 (1999) 319–328.
[24] F. Filbet, E. Sonnendrücker, P. Bertrand, Conservative numerical schemes for the Vlasov equation, J. Comput. Phys. 172 (2001) 166–187.
[25] C.Z. Cheng, G. Knorr, The integration of the Vlasov equation in configuration spaces, J. Comput. Phys. 22 (1976) 330–351.
[26] E. Sonnendrücker, J. Roche, The semi-Lagrangian method for the numerical resolution of Vlasov equation, J. Comput. Phys. 149 (1999) 201–220.
[27] G. Depret, X. Garbet, P. Bertrand, A. Ghizzo, Trapped-ion driven turbulence in tokamak plasmas, Plasma Phys. Control. Fusion 42 (2000) 949–971.
[28] M. Brunetti, V. Grandgirard, P. Bertrand, O. Sauter, J. Vaclavik, L. Villard, Fine-scale structures and negative density regions: comparison of numericals methods for solving advection equation, Transport Theory Stat. Phys. 34 (3–5) (2005) 261–274.
[29] F. Jenko, W. Dorland, B. Scott, Strinzi, in: Proceedings of the 20th International School of Plasma Physics on the Theory of Fusion Plasmas, Simulation and Theory of Electromagnetic ETG Turbulence, Varenna, 2002, pp. 147–152.
[30] Y. Sarazin, V. Grandgirard, E. Fleurence, X. Garbet, Ph. Ghendrih, P. Bertrand, G. Depret, Kinetic features of interchange turbulence, Plasma Phys. Control. Fusion 47 (2005) 1817–1839.
[31] W.H. Press, A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in Fortran, second ed., Cambridge University Press, Cambridge, 1992, pp. 43–50.
[32] R.F. Boisvert, Algorithms for special tridiagonal systems, SIAM J. Sci. Stat. Comput. 12 (2) (1991) 423–442.
[33] G.I. Marchuk, Method of Numerical Mathematics, Springer-Verlag, New York, 1982.
[34] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506–517.
[35] C. DeBoor, A practical guide to splinesApplied Mathematical Sciences, vol. 27, Springer-Verlag, New York, 2001.

[36] B.D. Fried, S.D. Conte, The Plasma Dispersion Function, Academic Press, New York, 1961.
[37] B. Davies, Locating the zeros of an analytic function, J. Comput. Phys. 66 (1986) 36–49.
[38] S. Brunner, J. Vaclavik, Global approach to the spectral problem of microinstabilities in a cylindrical plasma using a gyrokinetic model, Phys. Plasmas 5 (2) (1998) 365–375.
[39] C. Bourdelle, X. Garbet, G.T. Hoang, J. Ongena, R.V. Budny, Stability analysis of improved confinement discharges: internal barriers in Tore Supra and radiative improved mode in Textor, Nucl. Fusion 42 (2002) 892–902.
[40] M. Fivaz, S. Brunner, G. De Ridder, et al., Finite element approach to global gyrokinetic Particle-In-Cell simulations using magnetic coordinates, Comp. Phys. Commun. 111 (1998) 27–47.
[41] V. Grandgirard, P. Bertrand, G. Depret, X. Garbet, A. Ghizzo, G. Manfredi, O. Sauter, T.M. Tran, J. Vaclavick, L. Villard, A semi-Lagrangian method for the resolution of a drift-kinetic equation applied to ITG studies, in: 28th EPS Conference on Controlled Fusion and Plasma Physics – Funchal, Madeira, Portugal, 2001.
[42] M. Brunetti, V. Grandgirard, O. Sauter, J. Vaclavik, L. Villard, A semi-Lagrangian code for nonlinear global simulations of electrostatic drift-kinetic ITG modes, Comp. Phys. Commun. 163 (2004) 1–21.
[43] T.H. Watanabe, H. Sugama, Kinetic simulation of a quasisteady state in collisionless ion temperature gradient driven turbulence, Phys. Plasmas 9 (9) (2002) 3659–3662.