Gyroaverage operator for a polar mesh

Christophe Steiner[1], Michel Mehrenberger[1,2], Nicolas Crouseilles[3], Virginie Grandgirard[4] and Guillaume Latu[4] [a]

[1] IRMA, Université de Strasbourg, France
[2] Max-Planck-Institut für Plasmaphysik, Garching, Germany
[3] Inria-Rennes Bretagne Atlantique, IPSO team et IRMAR, Université de Rennes 1, France
[4] CEA/DSM/IRFM, Association Euratom-CEA, Cadarache, France

Abstract. In this work, we are concerned with numerical approximation of the gyroaverage operators arising in plasma physics to take into account the effects of the finite Larmor radius corrections. The work initiated in [5] is extended here to polar geometries. A direct method is proposed in the space configuration which consists in integrating on the gyrocircles using interpolation operator (Hermite or cubic splines). Numerical comparisons with a standard method based on a Padé approximation are performed: *(i)* with analytical solutions, *(ii)* considering the $4D$ drift-kinetic model with one Larmor radius and *(iii)* on the classical linear DIII-D benchmark case [6]. In particular, we show that in the context of a drift-kinetic simulation, the proposed method has similar computational cost as the standard method and its precision is independent of the radius.

PACS. PACS-key discribing text of that key – PACS-key discribing text of that key

1 Introduction

In strongly magnetized plasma, when collision effects are negligible, one has to deal with kinetic models since fluid models, which assume that the distribution function is close to an equilibrium, are not suitable. However, the numerical solution of Vlasov type models is challenging since this model involves six dimensions in the phase space. Moreover, multi-scaled phenomena make the problem very difficult. Gyrokinetic theory enables to get rid of one of these constraints since the explicit dependence on the phase angle of the Vlasov equation is removed through gyrophase averaging while gyroradius effects are retained. The so-obtained five-dimensional function is coupled with the Poisson equation (or its asymptotic counterpart, the so-called quasi-neutrality equation) which is defined on the particle coordinates. Thus, solving the gyrokinetic Vlasov-Poisson system requires an operator that transforms the gyro-center phase space in the particles phase space. This operator is the so-called gyroaverage operator. We refer to an abundant literature around this subject (see [2,13,19] and references therein).

The present work is devoted to the numerical computation of the gyroaverage operator for a polar mesh, following a previous work which was designed for cartesian geometry [5].

We propose an alternative to the classical Padé approximation, that is especially employed in the gyrokinetic code GYSELA [8,9] and which is known to be only valid for small Larmor radius. The method is based on direct integration and interpolation. Close to the method used already in the GENE code (see [12,11]), it is applied in the context of $2D$ interpolation, as we consider here a fully global simulation.

The paper is organized as follows. In Section 2, we introduce the gyroaverage operator. The classical Padé approximation for the gyroaverage computation is described in Section 3. In Section 4, we present the method based on interpolation. A numerical comparison with analytical solutions is performed in Section 5. Finally, these gyroaverage operators will be applied to gyrokinetic simulations in Section 6.

2 Gyroaverage operator definition

Let $\boldsymbol{\rho}$ be the gyro-radius which is transverse to $\boldsymbol{b} = \boldsymbol{B}/B$ (with $\boldsymbol{B}$ the magnetic field) and which depends on the gyrophase $\alpha \in [0, 2\pi]$, i.e

$$\boldsymbol{\rho} = \rho(\cos(\alpha)\boldsymbol{e}_{\perp 1} + \sin(\alpha)\boldsymbol{e}_{\perp 2})$$

Here $\boldsymbol{e}_{\perp 1}$ and $\boldsymbol{e}_{\perp 2}$ are the unit vectors of a cartesian basis in the plane perpendicular to the magnetic field direction $\boldsymbol{b}$. Let $\boldsymbol{x}_G$ be the guiding-center radial coordinate and $\boldsymbol{x}$ the position of the particle in the real space.

These two quantities differ by a Larmor radius $\boldsymbol{\rho}$, i.e $\boldsymbol{x} = \boldsymbol{x}_G + \boldsymbol{\rho}$. The gyroaverage $\mathcal{J}_\rho(f)$ of any function $f : (r, \theta) \rightarrow g(r\cos(\theta), r\sin(\theta))$ depending on the spatial coordinates is defined as

$$\mathcal{J}_\rho(f)(r,\theta) = \frac{1}{2\pi} \int_0^{2\pi} g(\boldsymbol{x} + \boldsymbol{\rho}) d\alpha.$$

This gyro-average process consists in computing an average on the Larmor circle. It tends to damp any fluctuation which develops at sub-Larmor scales.

Introducing $\widehat{f}(\boldsymbol{k})$ the Fourier transform of f, with $\boldsymbol{k} = k(\cos(\theta), \sin(\theta))$ the wave vector, then the operation of gyro-average reads

$$\mathcal{J}_\rho(f)(\boldsymbol{x}_G) = \int_0^{2\pi} \frac{d\alpha}{2\pi} \int_{-\infty}^{+\infty} \frac{d^3\boldsymbol{k}}{(2\pi)^3} \widehat{f}(\boldsymbol{k}) \exp\{i\boldsymbol{k} \cdot (\boldsymbol{x}_G + \boldsymbol{\rho})\}$$
$$= \int_{-\infty}^{+\infty} \frac{d^3\boldsymbol{k}}{(2\pi)^3} \left[\int_0^{2\pi} \frac{d\alpha}{2\pi} \exp(ik_\perp \rho \cos\alpha) \right] \times$$
$$\widehat{f}(\boldsymbol{k}) \exp(i\boldsymbol{k} \cdot \boldsymbol{x}_G)$$

where $k_\perp$ is the norm of the transverse component of the wave vector $\boldsymbol{k}_\perp = \boldsymbol{k} - (\boldsymbol{b}.\boldsymbol{k})\boldsymbol{b}$. Let consider J_n the Bessel function of the first kind and of order $n \in \mathbb{N}$, i.e $\forall z \in \mathbb{C}$, $J_n(z) = \frac{i^{-n}}{\pi} \int_0^\pi \exp(iz\cos\theta) \cos(n\theta)\, d\theta$. Therefore, the previous gyro-average operation can be expressed as a function of the Bessel function of first order J_0 as

$$\mathcal{J}_\rho(f)(\boldsymbol{x}_G) = \int_{-\infty}^{+\infty} \frac{d^3\boldsymbol{k}}{(2\pi)^3} J_0(k_\perp \rho)\widehat{f}(\boldsymbol{k}) \mathrm{e}^{i\boldsymbol{k}\cdot\boldsymbol{x}_G} \qquad (1)$$

Let consider a uniform polar mesh $(r, \theta) \in [r_{\min}, r_{\max}] \times [0, 2\pi[$ with $N_r \times N_\theta$ cells, our goal is to approximate the operator

$$(f_{j,k}) \in \mathbb{R}^{(N_r+1)\times N_\theta} \mapsto (\mathcal{J}_\rho(f)_{j,k}) \in \mathbb{R}^{(N_r+1)\times N_\theta}.$$

Considering expression (1), in Fourier space the gyro-average reduces to the multiplication by the Bessel function of argument $k_\perp \rho$. Indeed, the Fourier transform of $\mathcal{J}_\rho(f)$ can be written as

$$\widehat{\mathcal{J}_\rho(f)}(\boldsymbol{k}) = \frac{1}{2\pi} \int_{\mathbb{R}^2} \int_0^{2\pi} f(\boldsymbol{x} + \boldsymbol{\rho}) d\alpha\ e^{-i\boldsymbol{x}\cdot\boldsymbol{k}} d\boldsymbol{x}$$
$$= \frac{1}{2\pi} \int_0^{2\pi} \int_{\mathbb{R}^2} f(\boldsymbol{x} + \boldsymbol{\rho}) e^{-i(\boldsymbol{x}+\boldsymbol{\rho})\cdot\boldsymbol{k}} d\boldsymbol{x}\ e^{i\boldsymbol{\rho}\cdot\boldsymbol{k}} d\alpha$$
$$= \left(\frac{1}{2\pi} \int_0^{2\pi} e^{ik\rho\cos(\alpha-\theta)} d\alpha \right) \widehat{f}(\boldsymbol{k})$$

which leads to

$$\widehat{\mathcal{J}_\rho(f)}(\boldsymbol{k}) = J_0(k\rho)\widehat{f}(\boldsymbol{k}) \qquad (2)$$

This operation is straightforward in simple geometry with periodic boundary conditions, such as in local codes. Conversely, in the case of global codes, the use of Fourier transform is not applicable for two main reasons: (i) radial boundary conditions are non periodic, and (ii) the

radial dependence of the Larmor radius has to be accounted for. Several approaches have been developed to overcome this difficulty. The most widespread method for this gyro-averaging process is to use a quadrature formula. In this context, the integral over the gyro-ring is usually approximated by a sum over four points on the gyro-ring [18]. This is rigorously equivalent to considering the Taylor expansion of the Bessel function at order two in the small argument limit, namely $J_0(k_\perp\rho) \simeq 1 - (k_\perp\rho)^2/4$, and equivalent to computing the transverse Laplacian at second order using finite differences. This method has been extended to achieve accuracy for large Larmor radius [14], i.e the number of points (starting with four) is linearly increased with the gyro-radius to guarantee the same number of points per arclength on the gyro-ring. In this approach –used *e.g.* in [15] and [16]– the points that are equidistantly distributed over the ring are rotated for each particle (or marker) by a random angle calculated every time step. This is performed on a finite element formalism and enables therefore high order accuracy by keeping the matricial formulation.

In [5] the influence of the interpolation operator (which is of great importance when the quadrature points do not coincide with the grid points) has been studied and has shown that the cubic splines are a good candidate. Some techniques used in [5] taking advantages of the cartesian coordinates properties are no more valid in (r, θ) polar geometry. In this paper we present a method based on a direct integration of the gyroaverage operator which is directly applicable for global gyrokinetic code in toroidal geometry as GYSELA code. This new approach is tested for two different interpolation methods, one based on cubic splines and the other one on Hermite polynomials. Both are compared to the Padé approximation.

3 Method based on Padé approximation

One possible solution to compute the gyroaverage operator is to approximate the Bessel function with a Padé expansion $J_{\mathrm{Pade}}(k\rho) = 1/\left[1 + (k\rho)^2/4\right]$ (*e.g.* see [22]). As described in the following, such a Padé representation then requires the inversion of the Laplacian operator $\nabla_\perp^2$ in real space. Indeed, using this approximation, the relation (2) reads

$$\left(1 + \frac{(k\rho)^2}{4}\right) \widehat{\mathcal{J}_\rho(f)}(\boldsymbol{k}) = \widehat{f}(\boldsymbol{k})$$

which corresponds in real space to

$$\left(1 - \frac{\rho^2}{4}\Delta\right) \mathcal{J}_\rho(f)(\boldsymbol{x}) = f(\boldsymbol{x}).$$

We project f and $J(f)$ in the Fourier basis :

$$f(r,\theta) \approx \sum_{n=0}^{N_\theta - 1} A_n(r)e^{in\theta}, \qquad \mathcal{J}_\rho(f)(r,\theta) \approx \sum_{n=0}^{N_\theta - 1} B_n(r)e^{in\theta}.$$

The Laplacian in polar coordinates is expressed as

$$\Delta = \frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2}$$

then we get for $n = 0...N_\theta - 1$

$$-\frac{\rho^2}{4}B_n''(r) - \frac{\rho^2}{4r}B_n'(r) + \left(1 + \frac{\rho^2 n^2}{4r^2}\right)B_n(r) = A_n(r)$$

that can be solved by finite differences. For the results presented in the following, we use finite differences of second order which leads to a tridiagonal system.

This Padé approximation gives the correct limit in the large wavelengths limit $k\rho \ll 1$, while keeping J_{Pade} finite in the opposite limit $k\rho \to \infty$. The drawback is an over-damping of small scales: in the limit of large arguments $x \to \infty$, $J_{\mathrm{Pade}}(x) \to 4/x^2$, whereas $J_0 \to (2/\pi x)^{1/2}\cos(x - \pi/4)$ (see figure 2). The method proposed in the next section which is no more based on an approximation of the Bessel function but on the direct calculation of the integral on a circle of radius ρ has been developed to overcome this drawback.
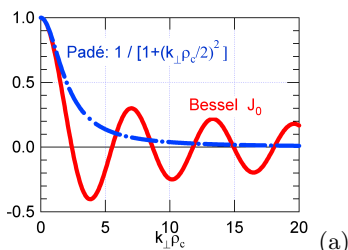
Fig. 1. The zero-th order Bessel function $J_0(k\rho)$ compare to its Padé approximation $1/\left[1 + (k\rho)^2/4\right]$.
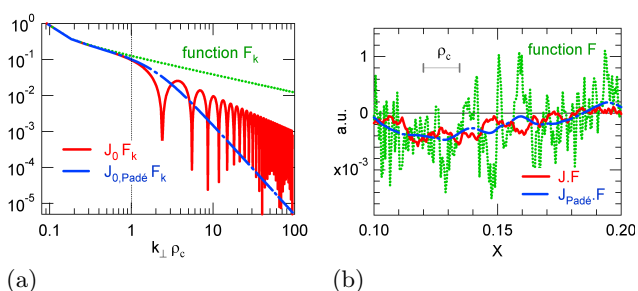
Fig. 2. Exact and approximated gyro-average operators applied on an arbitrary function F_k exhibiting a broad spectrum ranging from low to large wavelengths as compared with the Larmor radius ρ: (a) Representation in the Fourier space, (b) Representation in the real space (figures from [22]).

4 Method based on interpolation

We put N uniformly distributed points on the circle of integration and we approximate the function value at these points by interpolation. The gyroaverage is then obtained by the rectangle quadrature formula on these points. More precisely, for a given point (r_j, θ_k), the gyroaverage at this point is approximated by

$$J_\rho(f)_{j,k} \simeq$$
$$\frac{1}{2\pi}\sum_{\ell=0}^{N-1}\mathcal{P}(f)(r_j\cos\theta_k + \rho\cos\alpha_\ell, r_j\sin\theta_k + \rho\sin\alpha_\ell)\Delta\alpha,$$

where $\alpha_\ell = \ell\Delta\alpha$ and $\Delta\alpha = 2\pi/N$. Since the quadrature points do not coincide with grid points, we introduce an interpolation operator $\mathcal{P}$ which can be

– Hermite interpolation,
– Cubic splines interpolation.

As detailed in [20], the interpolation can be reformulated into a matrix-vector product

$$\mathcal{P}(f)(r_j\cos\theta_k + \rho\cos\alpha_\ell, r_j\sin\theta_k + \rho\sin\alpha_\ell) = (A_\ell c)_{j,k},$$

where c denotes the splines coefficient (cubic splines method) or the function values (Hermite method) so that the gyroaverage can be itself viewed as a matrix-vector product

$$J_\rho(f)_{j,k} = \frac{1}{2\pi}\sum_{\ell=0}^{N-1}(A_\ell f)_{j,k}\Delta\alpha = (A_\rho c)_{j,k}.$$

As a consequence, for a given Larmor radius ρ, the matrix A_ρ can be stored once for all.
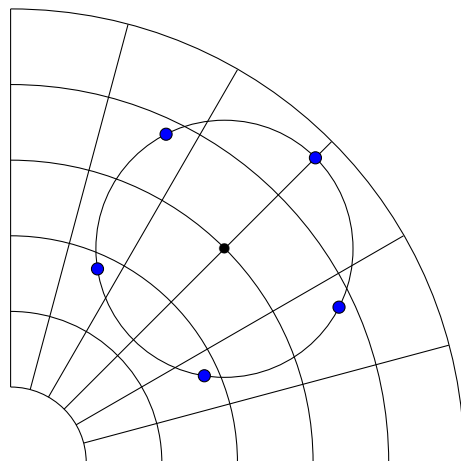
For each method, 2 versions are implemented :

– a basic version
– a version with precomputation where we first compute the matrix A_ρ such that

$$(\mathcal{J}_\rho(f)_{j,k}) = A_\rho c,$$

where c are spline coefficients (vector of size $(N_r + 1)N_\theta$) or function values and derivatives in the case of Hermite interpolation (size is $4(N_r + 1)N_\theta$).

ρ	Hermite (4)	(6)	(10)	(18)	splines	Padé
0	6	7	9	14	8	0.6
0.001	10	11	13	18	12	2
0.01	40	41	43	48	53	2
0.1	446	442	448	453	594	2

Table 1. Time (in sec) of Hermite precompute and Padé as a function of ρ with different orders for Hermite interpolation. Parameters : $r_{\min} = 0.1, r_{\max} = 0.9, N_r = N_\theta = 512, N = 1024$, 100 iterations of the gyroaverage.

ρ	Hermite basic	Hermite precompute
0	301	0.6
0.001	316	0.8
0.01	312	1
0.1	304	5

Table 2. Time (in sec) as a function of ρ for Hermite without and with precomputing. Parameters : $r_{\min} = 0.1, r_{\max} = 0.9, N_r = N_\theta = 128, N = 1024$, order of interpolation : 4, 100 iterations of the gyroaverage.

Remark 1 Note that for the Hermite interpolation, we need first to compute the derivatives at each cell interface. These derivatives are reconstructed by centered finite differences of arbitrary even order (we could also use odd order by having only a C^0 reconstruction, but then the size of c would increase to $9(N_r + 1)N_\theta$). In the numerical results, we will take the order 4.

Remark 2 Time comparison are given in Tables 1 and 2. The use of the precomputation version is here quite efficient, as the Larmor radius ρ is fixed and the matrix is the same for each value of θ, which implies that the storage is reduced; but the basic version permits to give a rough indication of the time that would be used for more general situations (where for example the storage would be an issue), that are not considered for the moment. Optimization strategies, in a parallel environment, may be the subject of further extensions.

Remark 3 We may wonder about the numerical cost of the Hermite method in comparison to the Padé approximation, especially for a large radius. Luckily, the picture will change in the $4D$ case, since the gyroaverage is applied only in 3D. As an example, we have obtained the following times in the case of a drift kinetic simulation (see subsection 6): $42s$ for PADE, $43s$ for Hermite interpolation with precomputation (using 1024 quadrature points). Without precomputation, the time for Hermite is $47s$ with 16 quadrature points and $270s$ with 1024 quadrature points. Computations are made on a local cluster of the University of Strasbourg using 16 processors (grid $32 \times 32 \times 32 \times 64$, 100 iterations). Further time measures will be detailed in subsection 6.

5 Numerical comparison with analytical solutions

5.1 Definition of a class of analytical solutions depending on boundary conditions

First, we give a family of functions whose gyroaverage is analytically known. For these functions, we obtain the gyroaverage just by multiplying them by the Bessel function. Let $m \geq 0$ be an integer and C_m be the Bessel function of the first kind (denoted by J_m) or the Bessel function of the second kind (denoted by Y_m). The following proposition gives the analytic expression of the gyroaverage of Fourier-Bessel type functions.

Proposition. Let $z \in \mathbb{C}$. The gyroaverage of
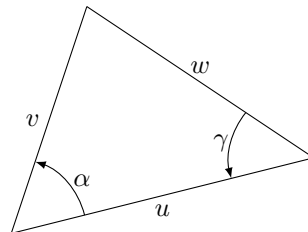
$$f(r,\theta) = C_m(zr)e^{im\theta}$$

reads

$$\mathcal{J}_\rho(f)(r_0,\theta_0) = J_0(z\rho)C_m(zr_0)e^{im\theta_0}.$$

Proof : By definition,

$$\mathcal{J}_\rho(f)(r,\theta)$$
$$= \frac{1}{2\pi} \int_0^{2\pi} \int_0^{+\infty} \int_0^{2\pi} C_m(zr_0)e^{im\theta_0}\delta_{\{\boldsymbol{x_0}=\boldsymbol{x}+\boldsymbol{\rho}\}}d\alpha dr_0 d\theta_0$$

where $\boldsymbol{x_0} = r_0(\cos(\theta_0),\sin(\theta_0))$, $\boldsymbol{x} = r_0(\cos(\theta),\sin(\theta))$ and $\boldsymbol{\rho} = \rho(\cos(\alpha),\sin(\alpha))$.

The additivity theorem of Graf for Bessel functions (see [1]) states that if u, v and w are the lengths of a triangle and α, γ the angles as shown in the following figure :

then for all integers m and all complex numbers z,

$$C_m(zw)e^{im\gamma} = \sum_{k=-\infty}^{\infty} C_{m+k}(zu)J_k(zv)e^{ik\alpha}.$$

Then, we obtain, with $v = \rho$, $w = r_0$, $u = r$, $\gamma = \theta_0 - \theta$

and $\alpha = \alpha$,

$$
\begin{aligned}
&\mathcal{J}_\rho(f)(r,\theta) \\
&= e^{im\theta} \frac{1}{2\pi} \int_0^{2\pi} \int_0^{+\infty} \int_0^{2\pi} C_m(zr_0) e^{im(\theta_0-\theta)} \times \\
&\quad \delta_{\{\boldsymbol{x_0}=\boldsymbol{x}+\boldsymbol{\rho}\}} d\alpha dr_0 d\theta_0 \\
&= e^{im\theta} \frac{1}{2\pi} \int_0^{2\pi} \int_0^{+\infty} \int_0^{2\pi} \left(\sum_{k=-\infty}^{\infty} C_{m+k}(zr) J_k(z\rho) e^{ik\alpha} \right) \times \\
&\quad \delta_{\{\boldsymbol{x_0}=\boldsymbol{x}+\boldsymbol{\rho}\}} d\alpha dr_0 d\theta_0 \\
&= e^{im\theta} \left(\sum_{k=-\infty}^{\infty} C_{m+k}(zr) J_k(z\rho) \right) \times \\
&\quad \frac{1}{2\pi} \int_0^{2\pi} \int_0^{+\infty} \int_0^{2\pi} e^{ik\alpha} \delta_{\{\boldsymbol{x_0}=\boldsymbol{x}+\boldsymbol{\rho}\}} d\alpha dr_0 d\theta_0 \\
&= e^{im\theta} \left(\sum_{k=-\infty}^{\infty} C_{m+k}(zr) J_k(z\rho) \right) \times \frac{1}{2\pi} \int_0^{2\pi} e^{ik\alpha} d\alpha.
\end{aligned}
$$

We use the fact that

$$
\frac{1}{2\pi} \int_0^{2\pi} e^{ik\alpha} d\alpha = \delta_{k,0}
$$

in order to conclude that

$$
\mathcal{J}_\rho(f)(r,\theta) = J_0(z\rho) C_m(zr) e^{im\theta}.
$$

□

In the following, we give some examples of these test functions depending on the boundary conditions we want to test.

Examples

1. $r_{\min} = 0$ and homogeneous Dirichlet condition on $r_{\max}$.

   Here we consider a disk and the function

   $$
   f_1(r,\theta) = J_m\left(r\frac{j_{m,\ell}}{r_{\max}} \right) e^{im\theta}
   $$

   where $j_{m,\ell}$ is the ℓth zero of J_m verifies

   $$
   f_1(r_{\max},\theta) = 0, \ 0 \le \theta < 2\pi,
   $$

   and its gyroaverage reads

   $$
   \mathcal{J}_\rho(f_1)(r_0,\theta_0) = J_0\left(\rho\frac{j_{m,\ell}}{r_{\max}} \right) f_1(r_0,\theta_0).
   $$

2. Homogeneous Dirichlet condition on $r_{\min} > 0$ and $r_{\max}$.

   The following function is defined on an annulus

   $$
   f_2(r,\theta) = \left(J_m(\gamma_{m,\ell}) Y_m\left(r\frac{\gamma_{m,\ell}}{r_{\max}} \right) - Y_m(\gamma_{m,\ell}) J_m\left(r\frac{\gamma_{m,\ell}}{r_{\max}} \right) \right) e^{im\theta}
   $$

where $\gamma_{m,\ell}$ is the ℓth zero of

$$
y \mapsto J_m(y) Y_m\left(y\frac{r_{\min}}{r_{\max}} \right) - Y_m(y) J_m\left(y\frac{r_{\min}}{r_{\max}} \right)
$$

verifies

$$
f_2(r_{\min},\theta) = 0, \quad f_2(r_{\max},\theta) = 0, \ 0 \le \theta < 2\pi,
$$

and its gyroaverage reads

$$
\mathcal{J}_\rho(f_2)(r_0,\theta_0) = J_0\left(\rho\frac{\gamma_{m,\ell}}{r_{\max}} \right) f_2(r_0,\theta_0).
$$

3. Homogeneous Neumann condition on $r_{\min} > 0$ and $r_{\max}$.

   The following function is defined on an annulus

   $$
   f_3(r,\theta) = \left(J'_m(\eta_{m,\ell}) Y_m\left(r\frac{\eta_{m,\ell}}{r_{\max}} \right) - Y'_m(\eta_{m,\ell}) J_m\left(r\frac{\eta_{m,\ell}}{r_{\max}} \right) \right) e^{im\theta}
   $$

   where $\eta_{m,\ell}$ is the ℓth zero of

   $$
   y \mapsto J'_m(y) Y'_m\left(y\frac{r_{\min}}{r_{\max}} \right) - Y'_m(y) J'_m\left(y\frac{r_{\min}}{r_{\max}} \right)
   $$

   verifies

   $$
   \partial_r f_3(r_{\min},\theta) = 0, \quad \partial_r f_3(r_{\max},\theta) = 0, \ 0 \le \theta < 2\pi,
   $$

   and its gyroaverage reads

   $$
   \mathcal{J}_\rho(f_3)(r_0,\theta_0) = J_0\left(\rho\frac{\eta_{m,\ell}}{r_{\max}} \right) f_3(r_0,\theta_0).
   $$

We have here used the fact that for $\mathcal{C}_n = J_n$ or Y_n, the derivative reads :

$$
\mathcal{C}'_n(r) = -\mathcal{C}_{n+1}(r) + \frac{n\mathcal{C}_n(r)}{r}.
$$

We show in Fig. 3 the real and imaginary parts of the function

$$
(r,\theta) \in [0,5] \times [0,2\pi] \mapsto J_1(r) e^{i\theta}.
$$

5.2 Numerical results

In this part, the different numerical methods are compared in the case of the second test case (Homogenous Dirichlet condition on $r_{\min}$ and $r_{\max}$) with $r_{\min} = 0.1$, $r_{\max} = 0.9$, $N_r = N_\theta = 512$, $\ell = 1$ and $m = 1,5,20$. In Tables 3 and 4, we give here the L^2-norm error of the gyroaverage function for $m = 1$ whereas Tables 5,6 refer to $m = 5$ and Tables 7,8 refer to $m = 20$. In Table 8, we use various orders of interpolation for Hermite. Note that for each method, the error is computed over the domain $[r_{\min} + \rho, r_{\max} - \rho]$.

ρ	Padé	$N=4$	$N=8$	$N=16$	$N=1024$
0	10^{-17}	10^{-17}	10^{-17}	10^{-17}	10^{-17}
0.001	4.10^{-11}	3.10^{-6}	3.10^{-6}	3.10^{-6}	3.10^{-6}
0.01	2.10^{-8}	3.10^{-6}	1.10^{-6}	4.10^{-7}	5.10^{-7}
0.1	2.10^{-4}	3.10^{-4}	1.10^{-5}	4.10^{-8}	5.10^{-8}

Table 3. Comparison Padé with Hermite ($m=1$).

ρ	Padé	$N=4$	$N=8$	$N=16$	$N=1024$
0	10^{-17}	10^{-17}	10^{-17}	10^{-17}	10^{-15}
0.001	4.10^{-11}	3.10^{-6}	3.10^{-6}	3.10^{-6}	3.10^{-6}
0.01	2.10^{-8}	3.10^{-6}	1.10^{-6}	5.10^{-7}	6.10^{-7}
0.1	2.10^{-4}	3.10^{-4}	1.10^{-5}	4.10^{-8}	9.10^{-8}

Table 4. Comparison Padé with splines ($m=1$).

ρ	Padé	$N=4$	$N=8$	$N=16$	$N=1024$
0	10^{-17}	10^{-17}	10^{-17}	10^{-17}	10^{-17}
0.001	1.10^{-8}	5.10^{-6}	5.10^{-6}	5.10^{-6}	5.10^{-6}
0.01	2.10^{-6}	2.10^{-6}	1.10^{-6}	8.10^{-7}	7.10^{-7}
0.1	1.10^{-3}	1.10^{-3}	3.10^{-5}	1.10^{-7}	6.10^{-8}

Table 5. Comparison Padé with Hermite ($m=5$).

ρ	Padé	$N=4$	$N=8$	$N=16$	$N=1024$
0	10^{-17}	10^{-17}	10^{-17}	10^{-17}	10^{-17}
0.001	1.10^{-8}	5.10^{-6}	5.10^{-6}	5.10^{-6}	5.10^{-6}
0.01	2.10^{-6}	2.10^{-6}	1.10^{-6}	8.10^{-7}	8.10^{-7}
0.1	1.10^{-3}	1.10^{-3}	3.10^{-5}	2.10^{-7}	1.10^{-7}

Table 6. Comparison Padé with splines ($m=5$).

ρ	Padé	$N=4$	$N=8$	$N=16$	$N=1024$
0	10^{-18}	10^{-17}	10^{-17}	10^{-17}	10^{-17}
0.001	1.10^{-8}	5.10^{-6}	4.10^{-6}	4.10^{-6}	4.10^{-6}
0.01	6.10^{-6}	1.10^{-6}	1.10^{-6}	7.10^{-7}	7.10^{-7}
0.1	9.10^{-3}	3.10^{-3}	1.10^{-5}	9.10^{-8}	6.10^{-8}

Table 7. Comparison Padé with Hermite ($m=20$).

ρ	Padé	$N=4$	$N=8$	$N=16$	$N=1024$
0	10^{-18}	10^{-17}	10^{-17}	10^{-17}	10^{-16}
0.001	1.10^{-8}	4.10^{-6}	4.10^{-6}	4.10^{-6}	4.10^{-6}
0.01	6.10^{-6}	1.10^{-6}	1.10^{-6}	8.10^{-7}	8.10^{-7}
0.1	9.10^{-3}	3.10^{-3}	1.10^{-5}	1.10^{-7}	1.10^{-7}

Table 8. Comparison Padé with splines ($m=20$).

ρ	Hermite(4)	(6)	(10)	(18)
0	10^{-17}	10^{-17}	10^{-17}	10^{-17}
0.001	3.10^{-6}	3.10^{-6}	3.10^{-6}	3.10^{-6}
0.01	5.10^{-7}	6.10^{-7}	6.10^{-7}	6.10^{-7}
0.1	5.10^{-8}	8.10^{-8}	1.10^{-7}	1.10^{-7}

Table 9. Comparison Hermite with different interpolation order. Parameters : $N = 1024$, $m = 1$.
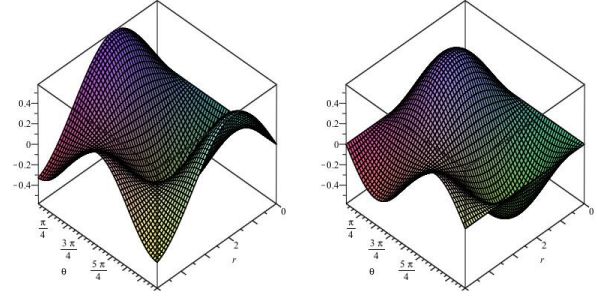
Fig. 3. Real and imaginary parts of the test function $(r,\theta) \mapsto J_1(r)\exp(i\theta)$.

Remark 4

1. For $\rho = 0$, all the methods are exact.
2. The method based on interpolation gives almost the same results with Hermite interpolation or cubic splines interpolation.
3. The method based on Padé approximation gives very good results for small values of ρ.
4. For large values of ρ, the method based on interpolation gives better results even with a relatively small number of points on the circle (for example with $\rho = 0.1$ and $N \geq 8$).

6 Application to gyrokinetic simulations

The computational effort to numerically solve the 6 dimensional Vlasov-Maxwell systems which describes plasma turbulence in tokamak plasmas still remains out of reach for present day supercomputers. All the numerical simulations performed until now in this domain take care of the gyrokinetic ordering to reduce this problem of one dimension. This ordering take into account the fact that (i) electromagnetic fluctuations occur on time scales much longer than charged particle gyration period ($\omega/\Omega_c \ll 1$ with ω the fluctuation frequency and Ω_c the cyclotron frequency), and (ii) the wavelength of these fluctuations is much smaller than the characteristic scale length of the gradients of magnetic field, density and temperature. See [7] for a detailed review on the gyrokinetic framework and simulations to compute turbulent transport in fusion plasmas. Within this gyro-ordering, the so-called gyrokinetic model can be derived (see [18]) by averaging on the fast gyration of charged particles around the magnetic field lines. The magnetic toroidal configuration considered in this paper is simplified. Indeed, magnetic flux surfaces are assumed concentric torii with circular cross-sections. The gyroaverage operator described in section 2 occurs in this reduction from 6 to 5 dimensions. The new 5D set of coordinates corresponds to: (i) 3D toroidal spatial coordinates (r,θ,φ) (with r the radial direction, θ and φ the poloidal (resp. toroidal) angle), and (ii) 2D in velocity space with $v_\parallel$ the velocity parallel to the magnetic field line and $\mu = mv_\perp^2/(2B)$ the magnetic moment where $v_\perp$ represents the velocity in the plane orthogonal to the magnetic field. It is important to note that in this ordering μ is

an adiabatic invariant, so it plays the role of a parameter in the 5D gyrokinetic Vlasov equation.

In the following, the 4D problem which is treated in section 6.1 corresponds to the case were we consider a unique value of μ, i.e the same Larmor radius is taken for all particles. The 5D problem, we deal with in section 6.2, several values of μ considered to take into account the dependence on the Larmor radius with $v_\perp$.

The time evolution of the gyrocenter distribution function $\bar{F}$ is given by the gyrokinetic conservative equation (see also Eqs (17)-(20) in [7]):

$$B_\parallel^* \frac{\partial \bar{F}}{\partial t} + \boldsymbol{\nabla} \cdot \left(B_\parallel^* \frac{d\mathbf{x}_G}{dt} \bar{F} \right) + \frac{\partial:}{\partial v_{G\parallel}} \left(B_\parallel^* \frac{dv_{G\parallel}}{dt} \bar{F} \right) = 0 \quad (3)$$

where $\mathbf{x}_G$ and $v_{G\parallel}$ are respectively the space coordinates and the parallel velocity of the guiding centers. In the electrostatic limit, for a particle of mass m and charge q the motion equations of the guiding centers are given by

$$\frac{d\mathbf{x}_G}{dt} = v_{G\parallel}\mathbf{b}^* + \mathbf{v}_{E\times B} + \mathbf{v}_D \quad (4)$$

$$m\frac{dv_{G\parallel}}{dt} = -\mu\boldsymbol{\nabla}_\parallel^* B - q\boldsymbol{\nabla}_\parallel^* \bar{\Phi} + mv_{G\parallel}\mathbf{v}_{E\times B} \cdot \frac{\boldsymbol{\nabla} B}{B} \quad (5)$$

where $\boldsymbol{\nabla}_\parallel^* \equiv \mathbf{b}^* \cdot \boldsymbol{\nabla}$, while $\mathbf{b}^*$ and $B_\parallel^*$ are defined by:

$$\mathbf{b}^* \equiv \frac{\mathbf{B}}{B_\parallel^*} + \frac{mv_{G\parallel}}{qB_\parallel^* B}\boldsymbol{\nabla} \times \mathbf{B} \quad (6)$$

$$B_\parallel^* \equiv B + \frac{mv_{G\parallel}}{qB}\mathbf{b} \cdot (\boldsymbol{\nabla} \times \mathbf{B}) \quad (7)$$

The '$\mathbf{E} \times \mathbf{B}$' drift is equal to $\mathbf{v}_{E\times B} = (1/B_\parallel^*)\mathbf{b} \times \boldsymbol{\nabla}\bar{\Phi}$ while curvature drift is defined as $\mathbf{v}_D = \left(\frac{mv_{G\parallel}^2 + \mu B}{qB_\parallel^*} \right) \mathbf{b} \times \frac{\boldsymbol{\nabla} B}{B}$.

We focus on the turbulent transport driven by collisionless ITG instability so electrons are assumed adiabatic. In this limit, the gyro-averaged electrostatic potential $\bar{\Phi}$ (equivalent to $\mathcal{J}_{\sqrt{2\mu}}\Phi$ notation) is solution of the self-consistently coupled 3D quasi-neutrality equation:

$$-\left(\partial_r^2 \Phi + \left(\frac{1}{r} + \frac{\partial_r n_0(r)}{n_0(r)} \right) \partial_r \Phi + \frac{1}{r^2}\partial_\theta^2 \Phi \right) +$$
$$\frac{1}{T_e(r)}(\Phi - \lambda\langle\Phi\rangle) = \frac{1}{n_0(r)}\mathcal{J}_{\sqrt{2\mu}}\left(\int f - f_{eq} dv \right), \quad (8)$$

$$\langle\Phi\rangle = \frac{1}{L}\int_0^L \Phi(r,\theta,z)dz, \quad (9)$$

where T_e and n_0 are electron temperature and density profiles which will be defined latter.

The new gyroaverage methods presented before have been tested with two codes: (i) the SELALIB plateform [21] for the 4D simplified case and (ii) the GYSELA code [9] for the benchmark with the classical Cyclone DIID 5D case. Both are based on a classical Backward semi-Lagrangian scheme (BSL) with cubic splines interpolation, predictor-corrector method.

In the following numerical solutions are computed using normalized equations. The temperature is normalized to T_{e0}, where T_{e0} is defined by the initial temperature profile such that $T_e(r_p)/T_{e0} = 1$. The time is normalized to the inverse of the ion cyclotron frequency $\omega_c = e_i B_0/m_i$. Velocities, including the parallel velocity, are expressed in units of the ion speed $v_{T0} = \sqrt{T_{e0}/m_i}$, the electric potential is normalized to T_{e0}/e_i and the magnetic field is normalized to B_0. Consequently, lengths are normalized to the Larmor radius $\rho = m_i v_{T0}/e_i B_0$ and the magnetic moment μ to T_{e0}/B_0.

6.1 Simplified 4D SLAB case

In this section, we consider a simplified model of the system of equations (3)-(9). A periodic cylindrical plasma of radius a and length $2\pi R$ (with R the major radius) is considered as a limit case of a stretched torus. The plasma is confined by a strong magnetic which is uniform $\mathbf{B} = B\mathbf{e}_z$ where $\mathbf{e}_z$ stand for the unit vector in the toroidal direction z. With theses assumptions the velocity drifts are reduced to the $\mathbf{E} \times \mathbf{B}$ drift. This SLAB 4D case is equivalent to the one treated in [17] or [8].

The equation satisfied by the distribution function of ions $f(t,r,\theta,z,v)$ following the guiding center movement reads :

$$\partial_t f - \left(\frac{\partial_\theta \mathcal{J}_{\sqrt{2\mu}}\Phi}{r} \right) \partial_r f + \left(\frac{\partial_r \mathcal{J}_{\sqrt{2\mu}}\Phi}{r} \right) \partial_\theta f +$$
$$v\partial_z f - \left(\partial_z \mathcal{J}_{\sqrt{2\mu}}\Phi \right) \partial_v f = 0. \quad (10)$$

for $(r,\theta,z,v) \in [r_{\min}, r_{\max}] \times [0, 2\pi] \times [0, L] \times [-v_{\max}, v_{\max}]$.

To deals with this equation system, we have used the SELALIB platform [21] with a classical semi-Lagrangian method with cubic splines interpolation, predictor corrector method and Verlet algorithm for the characteristics (see [4] for details). The platform has been improved by adding the one fixed μ capability and by implementing the three different gyroaverage operators described in sections 3 and 4. In our case the MPI parallelization is based on transpositions between (r,θ,v) domain decomposition and z domain decomposition. In this section the numerical instability growth rates are compared to the one deduced from the dispersion relation obtained by linearizing the self-consistent equation system (9)-(10).

6.1.1 Dispersion relation

In order to validate the linear part of the numerical results, we compute the dispersion relation with the gyroaverage. We make the following expansions:

$$f = f_0 + \varepsilon f_1 + \mathcal{O}(\varepsilon^2), \quad \phi = \phi_0 + \varepsilon\phi_1 + \mathcal{O}(\varepsilon^2)$$

with

$$f_0(r,v) = f_{eq}(r,v) = \frac{n_0(r)\exp\left(-\frac{v^2}{2T_i(r)}\right)}{(2\pi T_i(r))^{1/2}}, \quad \phi_0 = 0.$$

Then we obtain

$$\mathcal{J}_{\sqrt{2\mu}}(f) = \mathcal{J}_{\sqrt{2\mu}}(f_0 + \varepsilon f_1) + \mathcal{O}(\varepsilon^2),$$

and

$$\mathcal{J}_{\sqrt{2\mu}}(\phi) = \mathcal{J}_{\sqrt{2\mu}}(\bar{\phi}_0 + \varepsilon\bar{\phi}_1) + \mathcal{O}(\varepsilon^2).$$

Substituting the above relations into (10), we obtain

$$\partial_t f_1 - \frac{\partial_\theta \mathcal{J}_{\sqrt{2\mu}}(\phi_1)}{r}\partial_r f_0 + v\partial_z f_1 - \partial_z \mathcal{J}_{\sqrt{2\mu}}(\phi_1)\partial_v f_0 = \mathcal{O}(\varepsilon).$$

Similarly, the equation

$$-\left(\partial_r^2\phi + \left(\frac{1}{r} + \frac{\partial_r n_0(r)}{n_0(r)}\right)\partial_r\phi + \frac{1}{r^2}\partial_\theta^2\phi\right) +$$

$$\frac{1}{T_e(r)}(\phi - \lambda\langle\phi\rangle) = \frac{1}{n_0(r)}\mathcal{J}_{\sqrt{2\mu}}\left(\int f - f_{eq}dv\right)$$

becomes

$$-\left(\partial_r^2\phi_1 + \left(\frac{1}{r} + \frac{\partial_r n_0(r)}{n_0(r)}\right)\partial_r\phi_1 + \frac{1}{r^2}\partial_\theta^2\phi_1\right) +$$

$$\frac{1}{T_e(r)}(\phi_1 - \lambda\langle\phi_1\rangle) = \frac{1}{n_0(r)}\mathcal{J}_{\sqrt{2\mu}}\left(\int f_1 dv\right) + \mathcal{O}(\varepsilon). \quad (11)$$

We assume that the solutions have the form :

$$f_1 = f_{m,n,\omega}(r,v)e^{i(m\theta + kz - \omega t)}, \quad \phi_1 = \phi_{m,n,\omega}(r)e^{i(m\theta + kz - \omega t)}$$

$$\mathcal{J}_{\sqrt{2\mu}}(f_1) = \hat{f}_{m,n,\omega}(r,v)e^{i(m\theta + kz - \omega t)},$$
$$\mathcal{J}_{\sqrt{2\mu}}(\phi_1) = \hat{\phi}_{m,n,\omega}(r)e^{i(m\theta + kz - \omega t)}$$

with $k = \frac{2\pi n}{L}$. Then, we obtain

$$(-\omega + kv)f_{m,n,\omega} = \left(\frac{m}{r}\partial_r f_0 + k\partial_v f_0\right)\hat{\phi}_{m,n,\omega} \quad (12)$$

and the relation (11) becomes

$$-\left(\partial_r^2\phi_{m,n,\omega} + \left(\frac{1}{r} + \frac{\partial_r n_0(r)}{n_0(r)}\right)\partial_r\phi_{m,n,\omega} - \frac{m^2}{r^2}\phi_{m,n,\omega}\right) +$$

$$\frac{1}{T_e(r)}(\phi_{m,n,\omega} - \lambda\delta_n^0\phi_{m,0,\omega}) = \frac{1}{n_0(r)}\int \hat{f}_{m,n,\omega}dv.$$

If we assume that $m \neq 0$ and $n \neq 0$, the last relation and the equation (12) lead to :

$$-\left(\partial_r^2\phi_{m,n,\omega} + \left(\frac{1}{r} + \frac{\partial_r n_0(r)}{n_0(r)}\right)\partial_r\phi_{m,n,\omega} - \frac{m^2}{r^2}\phi_{m,n,\omega}\right) +$$

$$\frac{1}{T_e(r)}\phi_{m,n,\omega} = \frac{1}{n_0(r)}\hat{\phi}_{m,n,\omega}\int \frac{\hat{f}_{m,n,\omega}}{f_{m,n,\omega}}\frac{\frac{m}{r}\partial_r f_0 + k\partial_v f_0}{kv - \omega}dv$$

and then

$$-\left(\frac{\partial_r^2\phi_{m,n,\omega}}{\phi_{m,n,\omega}} + \left(\frac{1}{r} + \frac{\partial_r n_0(r)}{n_0(r)}\right)\frac{\partial_r\phi_{m,n,\omega}}{\phi_{m,n,\omega}} - \frac{m^2}{r^2}\right) +$$

$$\frac{1}{T_e(r)} = \frac{1}{n_0(r)}\frac{\hat{\phi}_{m,n,\omega}}{\phi_{m,n,\omega}}\int \frac{\hat{f}_{m,n,\omega}}{f_{m,n,\omega}}\frac{\frac{m}{r}\partial_r f_0 + k\partial_v f_0}{kv - \omega}dv.$$

We make the approximations :

$$\frac{\hat{\Phi}_{m,n,\omega}}{\Phi_{m,n,\omega}} \approx J_0(\kappa\sqrt{2\mu}), \quad \frac{\hat{f}_{m,n,\omega}}{f_{m,n,\omega}} \approx J_0(\kappa\sqrt{2\mu})$$

where $\kappa \in \mathbb{R}^+$. Rigorously the previous approximations are true when f and Φ are Fourier-Bessel functions in (r,θ), i.e. when

$$f_{m,n,\omega}(r,v) = J_m(\kappa r) \times g(v)$$

and $\Phi_{m,n,\omega}(r) = J_m(\kappa r)$ (see Proposition in Section 3). In general, we are not in this case and this will explain the differences we observe in Figure 7. Then, by considering the previous approximations, we obtain :

$$-\left(\frac{\partial_r^2\phi_{m,n,\omega}}{\phi_{m,n,\omega}} + \left(\frac{1}{r} + \frac{\partial_r n_0(r)}{n_0(r)}\right)\frac{\partial_r\phi_{m,n,\omega}}{\phi_{m,n,\omega}} - \frac{m^2}{r^2}\right) +$$

$$\frac{1}{T_e(r)} = J_0(\kappa\sqrt{2\mu})^2\frac{1}{n_0(r)}\int \frac{\frac{m}{r}\partial_r f_0 + k\partial_v f_0}{kv - \omega}dv.$$

By setting

$$I = \int \frac{\frac{m}{r}\partial_r f_0 + k\partial_v f_0}{kv - \omega}dv$$

and by using the expression of f_0, we have

$$I = \int \frac{-\frac{v}{T_i} + \frac{m}{kr}\left(\frac{\partial_r n_0}{n_0} - \frac{\partial_r T_i}{2T_i} + \frac{v^2\partial_r T_i}{2T_i^2}\right)}{v - \frac{\omega}{k}}f_0 dv.$$

Now, we introduce for $n \in \mathbb{N}$:

$$I_n = \frac{1}{n_0}\int v^n\frac{f_0}{v - \frac{\omega}{k}}f_0 dv$$

and we obtain the relations :

$$I_1 = 1 + \frac{\omega}{k}I_0, \qquad I_2 = \frac{\omega}{k}\left(1 + \frac{\omega}{k}I_0\right).$$

By using the change of variable $v = (2T_i(r))^{1/2}w$ and the expression of f_0, we have by setting $k^* = (2T_i)^{1/2}k$.

$$I_0 = \int \frac{\exp\left(-\frac{v^2}{2T_i}\right)}{(2\pi T_i)^{1/2}(v - \frac{\omega}{k})}dv$$

$$= \int \frac{\exp(-\omega)}{\pi^{1/2}((2T_i(r))^{1/2}w - \frac{\omega}{k})}dv$$

$$= \frac{1}{(2T_i)^{1/2}}Z\left(\frac{\omega}{k^*}\right)$$

with

$$Z(z) = \frac{1}{\sqrt{\pi}}\int \frac{\exp(-x^2)}{x - z}dx = i\sqrt{\pi}\exp(-z^2)(1 - \text{erf}(-iz)),$$

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-t^2)dt.$$

Finally, for $z = \omega/k^*$:

$$\frac{J_0(\kappa\sqrt{2\mu})^2}{n_0(r)}I = -\frac{1}{T_i}(1 + zZ(z)) +$$

$$\frac{m}{k^*r}\left(Z(z)\left(\frac{\partial_r n_0}{n_0} - \frac{\partial_r T_i}{2T_i}\right) + z(1 + zZ(z))\frac{\partial_r T_i}{T_i}\right). \quad (13)$$

6.1.2 Instability growth rate comparisons

We adapt a code available in SELALIB, that computes the zeros for the dispersion relation (13), as in [3], by adding the gyroaverage term. Figure 7 presents the instability rates as a function of μ. We obtain the two first curves by solving the dispersion relation with $J_0(\sqrt{2\mu})^2$ (curve in red) or by substituting $J_0(\sqrt{2\mu})$ by its Padé approximation (curve in green). We have chosen $\kappa = 1$, in (13). The two remaining curves are obtain numerically with the Padé method for the gyroaverage operator (curve in blue) or the method with Hermite interpolation (curve in magenta). It appears that the slope decreases faster with the Hermite interpolation method than with the Padé method. The slopes obtained with the numerical Padé are different from these obtained with the dispersion relation and the Padé approximation because the functions that we consider here are not Fourier-Bessel functions.

In the simulations, we will take $\lambda = 0$ (no zonal flow case). The initial distribution function reads :

$$f(0, r, \theta, z, v) = f_{eq}(r, v) \times$$
$$\left(1 + \varepsilon \exp\left(-\frac{(r - r_p)^2}{\delta r}\right) \cos\left(\frac{2\pi n}{L} z + m\theta\right)\right)$$

where the equilibrium function f_{eq} is

$$f_{eq}(r, v) = \frac{n_0(r) \exp\left(-\frac{v^2}{2T_i(r)}\right)}{(2\pi T_i(r))^{1/2}}.$$

The profiles T_i, T_e and n_0 are given by :

$$\mathcal{P}(r) = C_{\mathcal{P}} \exp\left(-\kappa_{\mathcal{P}} \delta r_{\mathcal{P}} \tanh\left(\frac{r - r_p}{\delta r_{\mathcal{P}}}\right)\right)$$

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

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

We consider the parameters of [3] [Medium case] :

$$r_{\min} = 0.1, r_{\max} = 14.5, v_{\max} = 7.32, \kappa_{n_0} = 0.055,$$
$$\kappa_{T_i} = \kappa_{T_e} = 0.27586, \delta r_{T_i} = \delta r_{T_i} = \frac{\delta r_{n_0}}{2} = 1.45,$$
$$\varepsilon = 10^{-6}, n = 1, m = 5,$$
$$L = 1506.759067, r_p = \frac{r_{\min} + r_{\max}}{2}, \delta_r = \frac{4\delta r_{n_0}}{\delta r_{T_i}}.$$

Numerical results are given in Fig. 4 – 8. We will consider here $N = 1024$ for Hermite, wiht precomputation. For small μ, the differences between Hermite and Padé are very small and they are comparable to the case without gyroaverage. When μ increases, the differences between the two methods increase and the structures become coarser. The gyroaverage tends to reduce the instability rate; the more μ is large, the more this rate is small

μ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Herm.	15	16	15	16	16	16	16	16	15
Padé	15	15	14	15	17	15	16	16	15

Table 10. Time (in min.) on HPC, mesocenter of the University of Strasbourg, with 32 processors (4 nodes). $N_r \times N_\theta \times N_z \times N_v = 64 \times 64 \times 32 \times 64$, $\Delta t = 5$, 1600 iterations. For Hermite, with precomputation : $N = 1024$.

μ	Hermite	Padé
0.5	12524	13326
1	12556	13454

Table 11. Final time (in sec.) reached for a 24 hours simulation with $N_r \times N_\theta \times N_z \times N_v = 128 \times 256 \times 128 \times 128$, $\Delta t = 2$. On Helios Computational Simulation Centre, International Fusion Energy Research Centre of the ITER Broader Approac. Supercomputer with 128 processors (8 nodes; each node having 16 threads). For Hermite, with precomputation : $N = 1024$.

(Fig. 7).
Time results (Tab. 10 and 11) show that the choice of the gyroaverage operator is not very influential in the total time. Indeed, the computation of the gyroaverage appears to be a 3D problem in a 4D environment.

6.2 Benchmark with the classical 5D Cyclone DIII-D case

For this part, the gyroaverage operator based on cubic spline and Hermite interpolation have been implemented in the GYSELA code [9] and compared to the existing Padé approximation. The 5D Vlasov-Poisson problem considered is the one described by equations (3)-(9). The numerical comparisons have been performed on a linear benchmark based on the classical cyclone DIII-D case [6]. This kind of typical benchmark had already been performed several years ago to validate the GYSELA code [10]. For the present tests, the same parameters than in section 4 of [10] have been used. The comparison between the Padé operator and the gyroaverage operator based on Hermite interpolation is presented in the following tabular for four unstable modes (m, n) with m the poloidal mode number and n the toroidal one. The results obtained with cubic spline interpolation are not detailed because very similar to the results obtained with the Hermite interpolation. For the simulations, we have used $N = 32$ quadrature points, with the Hermite method and without precomputations. We observe the same behaviour as in the previous testcase: the instability growth rate is smaller (except for the first point), with Hermite, and the values are in the same range.

7 Conclusion

We have validated the gyroaverage computation on polar geometry. Comparisons are made with classical Padé

(m, n)	$(4, 3)$	$(10, 7)$	$(14, 10)$	$(17, 12)$	$(21, 15)$
Padé	$9.6e^{-5}$	$5.9e^{-4}$	$8.34e^{-4}$	$8.63e^{-4}$	$6.29e^{-4}$
Herm.	$1.145e^{-4}$	$5.89e^{-4}$	$8.175e^{-4}$	$8.276e^{-4}$	$5.4e^{-4}$

**Table 12.** Linear mode growth rates for the Cyclone DIID-D base case.

approximation, considering on the one hand analytical cases, for which we know the exact solution, and on the other hand some basic gyrokinetic simulations: a $4D$ drift-kinetic model with one Larmor radius and the classical linear DIII-D benchmark case. We find that the introduction of the gyroaverage operation tends to diminish the growth rate of the instability and this is amplified, when considering the direct (right) gyroaverage operator, instead of the Padé approximation. Linear analysis predicts a similar behaviour, when we compare Padé approximation and the J_0 Bessel function. This is coherent as the Padé approximation is above the Bessel function for $k\rho$ relatively small (see on Figure 2). Note that the result remains at the qualitative level, as here, in polar geometry, the multiplication by the J_0 Bessel function is not the exact solution (expect for Fourier-Bessel functions), and this differs from the cartesian geometry.

References

1. M. Abramowitz & I. A. Stegun, *Handbook of Mathematical Functions*, (Dover Publications, New York, 1965).
2. A. Brizard, J. Plasma Phys. 41, p. 541, (1989).
3. D. Coulette & N. Besse *Numerical comparisons of gyrokinetic multi-water-bag models.* JCP 248 (2013), 1–32.
4. N. Crouseilles, P. Glanc, S. A. Hirstoaga, E. Madaule, M. Mehrenberger & J. Pétri, *Semi-Lagrangian simulations on polar grids: from diocotron instability to ITG turbulence,* submitted.
5. N. Crouseilles, M. Mehrenberger & H. Sellama, *Numerical solution of the gyroaverage operator for the finite gyroradius guiding-center model*, CiCP 8 (2010), 484–510.
6. A. Dimits et al., *Comparisons and physics basis of tokamak transport models and turbulence simulations*, Physics of Plasmas, Vol. 7, No 3 (2000), 969–983.
7. X. Garbet, Y. Idomura, L. Villard & T.H. Watanabe, *Gyrokinetic simulations of turbulent transport*, Nuclear Fusion, Vol 50, No 4 (2010), 043002.
8. V. Grandgirard, M. Brunetti, P. Bertrand, N. Besse, X. Garbet, P. Ghendrih, G. Manfredi, Y. Sarazin, O. Sauter, E. Sonnendrücker, J. Vaclavik & L. Villard, *A drift-kinetic Semi-Lagrangian 4D code for ion turbulence simulation,* Journal of Computational Physics, 217(2):395–423, 2006.
9. V. Grandgirard, Y. Sarazin, X. Garbet, G. Dif-Pradalier, Ph. Ghendrih, N. Crouseilles, G. Latu, E. Sonnendrücker, N. Besse & P. Bertrand, *Computing ITG turbulence with a full-f semi-Lagrangian code,* Communications in Nonlinear Science and Numerical Simulation, 13(1): 81–87, 2008.
10. V. Grandgirard et al., *Computing ITG turbulence with a full-f semi-Lagrangian code*, Communications in Nonlinear Science and Numerical Simulation, Vol 13, No 1 (2008), 81–87.
11. T. Görler, *Multiscale effects in plasma microturbulence*, PhD, Ulm 2009.
12. T. Görler, X. Lapillonne, S. Brunner, T. Dannert, F. Jenko, F. Merz & D. Told, *The global version of the gyrokinetic turbulence code GENE.*, J. Comput. Physics 230(18): 7053-7071 (2011).
13. T. S. Hahm, *Nonlinear Gyrokinetic Equations for Tokamaks Microturbulence,* Phys. Fluids 31, p. 2670, (1988).
14. R. Hatzky, T.M. Tran, A. Konies, R. Kleiber & S.J. Allfrey, *Energy conservation in a nonlinear gyrokinetic particle-in-cell code for ion-temperature-gradient-driven modes in theta-pinch geometry*, Physics of Plasmas, Vol. 9, No 3 (2002), 898–912.
15. Y. Idomura, S. Tokuda, Y. Kishimoto & M. Wakatani, *Gyrokinetic theory of drift waves in negative shear tokamaks*, Nuclear Fusion, Vol. 41, No 4 (2001), 437.
16. S. Jolliet, A. Bottino, P. Angelino, R. Hatzky, T.M. Tran, B.F. Mcmillan, O. Sauter, K. Appert, Y. Idomura & L. Villard, *A global collisionless PIC code in magnetic coordinates*, Comp. Phys. Comm, Vol 177, No 5, (2007) 409–425.
17. R. Klein E. Gravier, P. Morel, N. Besse & P. Bertrand, *Gyrokinetic water-bag modeling of a plasma column : Magnetic moment distribution and finite Larmor radius effects*, Physics of plasmas, vol. 16, 082106, (2009).
18. W. W. Lee, *Gyrokinetic approach in particle simulation*, Physics of Fluids, Vol. 26, No 2 (1983), 556–562.
19. Z. Lin & W. W. Lee, *Method for solving the gyrokinetic Poisson equation in general geometry*, Phys. Rev. E, Vol. 52, No. 5 (1995), 5646–5652.
20. M. Mehrenberger, C. Steiner, L. Marradi, N. Crouseilles, E. Sonnendrücker & B. Afeyan, *Vlasov on GPU*, ESAIM Proc., 2013.
21. SELALIB, http://selalib.gforge.inria.fr/
22. Y. Sarazin, V. Grandgirard, E. Fleurence, X. Garbet, Ph. Ghendrih, P. Bertrand & G. Depret., *Kinetic features of interchange turbulence*, Plasma Phys. Control. Fusion, Vol 47, No 10 (2005), 1817–1840.
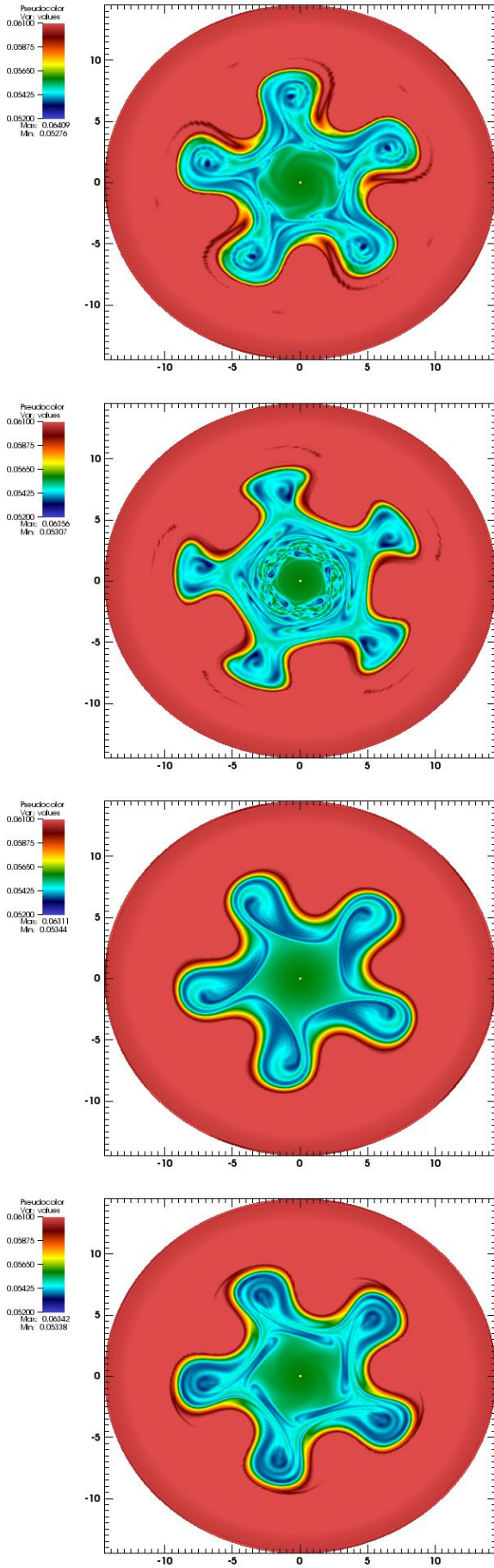
Fig. 4. Poloidal cut $f(r, \theta, 0, 0)$ at time $T = 7000$ for $128 \times 256 \times 128 \times 128$, $\Delta t = 2$, successively from top to bottom $\mu = 0.5$ with Hermite and then Padé; $\mu = 1$, with Hermite and then Padé.
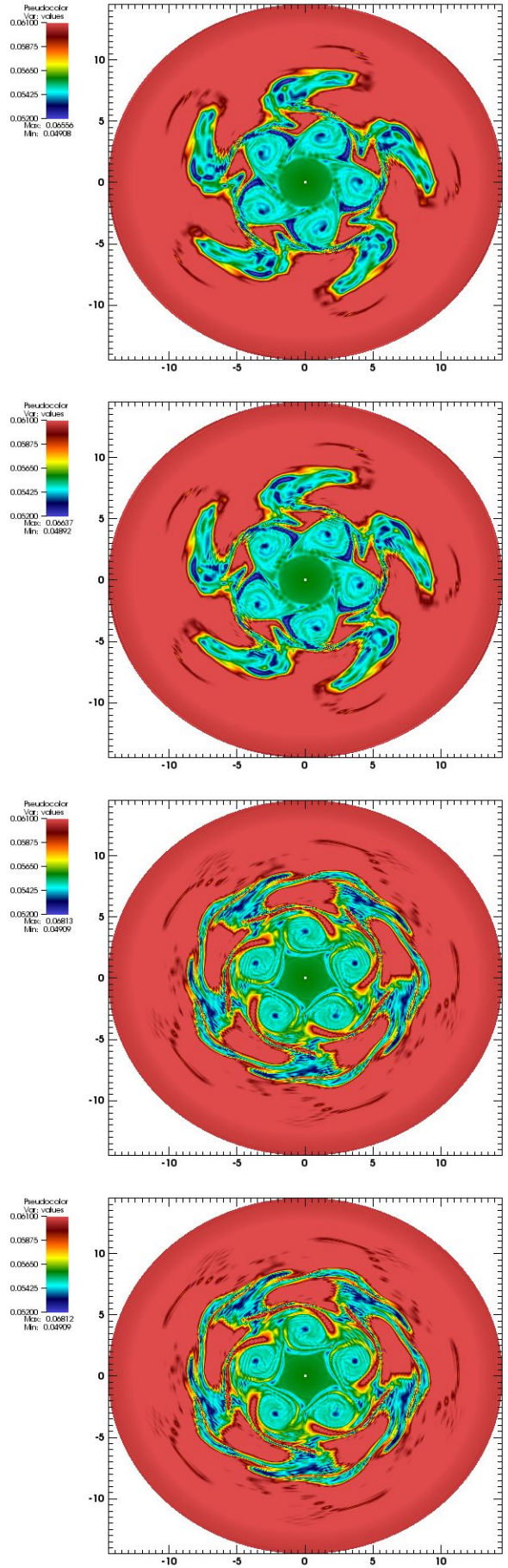
Fig. 5. Poloidal cut $f(r, \theta, 0, 0)$ at time $T = 5000$ for $128 \times 128 \times 128 \times 128$, $\Delta t = 1$, successively from top to bottom $\mu = 0.1$ with Hermite and then Padé; $\mu = 0$, with Hermite and then Padé.
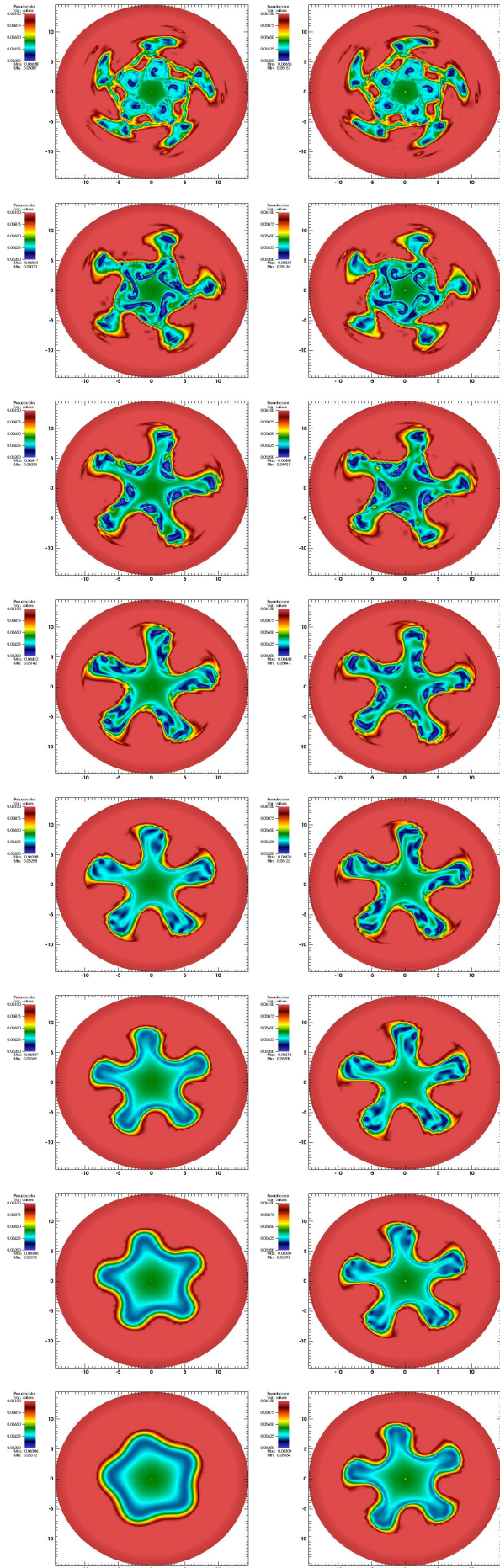
Fig. 6. Poloidal cut $f(r, \theta, 0, 0)$ at time $T = 5000$ for $64 \times 64 \times 32 \times 64$, $\Delta t = 5$ (left, Hermite; right, Padé) for $\mu = 0.1, \ldots, 0.8$ (from top to bottom).
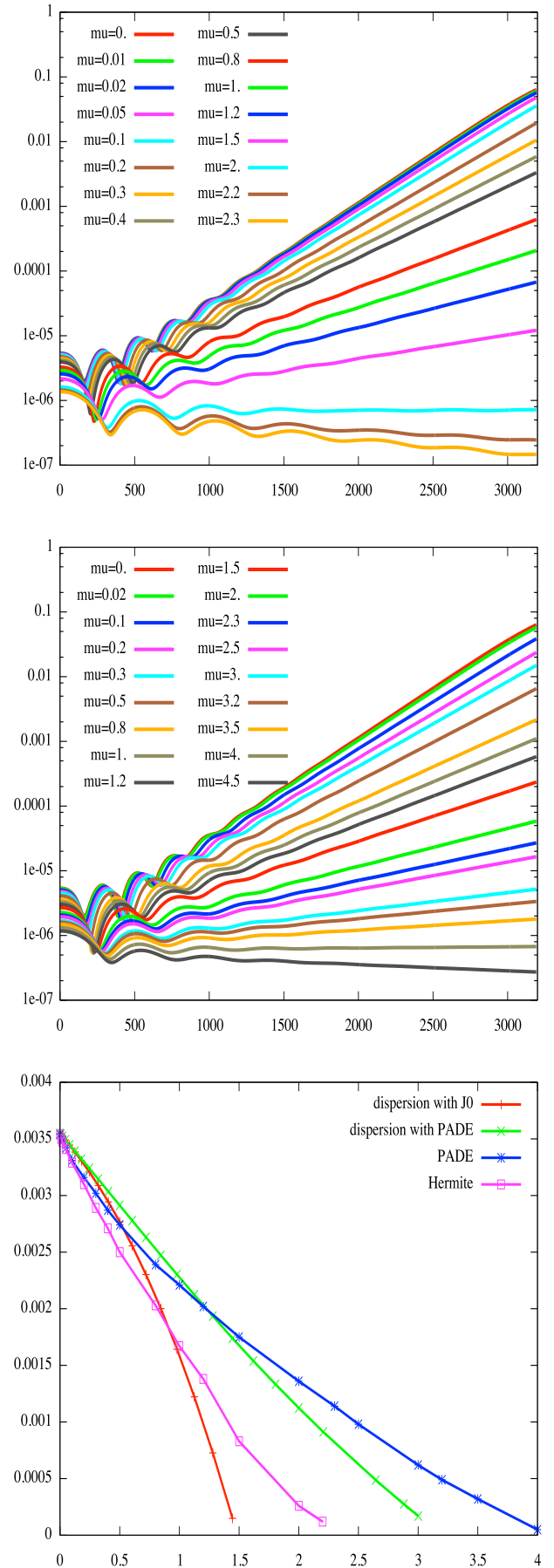
Fig. 7. Time evolution of $\int_{r_{\min}}^{r_{\max}} \int_0^{2\pi} \Phi(r, \theta, 0) r dr d\theta$: Hermite (top), Padé (middle). Bottom: instability rates as a function of μ; comparison between solution of dispersion relation (13) (using $J_0(\sqrt{2\mu})$ or its Padé approximation) and numerical results.
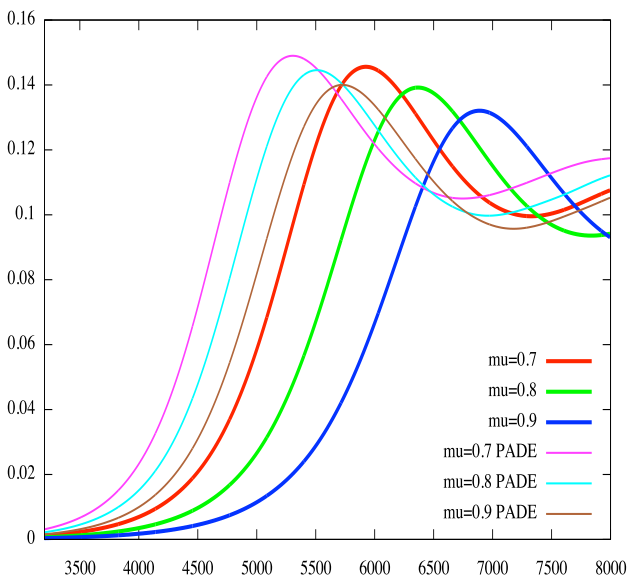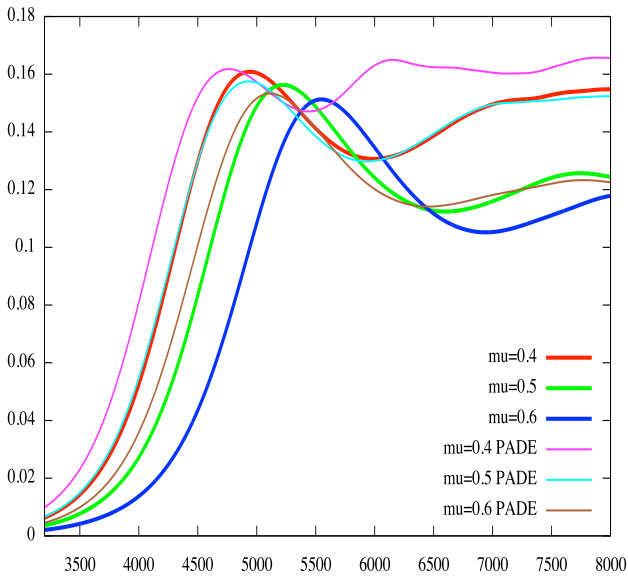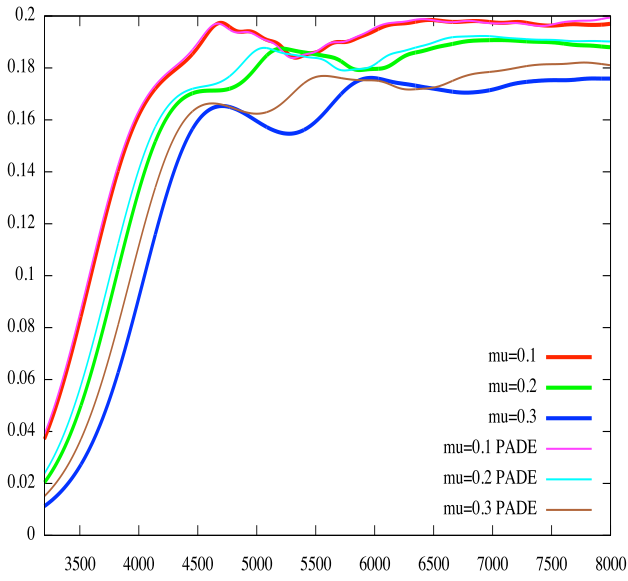
Fig. 8. Time evolution of $\int_{r_{\min}}^{r_{\max}} \int_0^{2\pi} \Phi(r, \theta, 0) r dr d\theta$. Comparison Hermite/Padé.