Field-aligned interpolation for semi-Lagrangian gyrokinetic simulations

Guillaume Latu

Michel Mehrenberger

Yaman Güçlü

Maurizio Ottaviani

Eric Sonnendrücker

May 9, 2016

Abstract

This work is devoted to the study of field-aligned interpolation in semi-Lagrangian codes.[1] In the context of numerical simulations of magnetic fusion devices, this approach is motivated by the observation that gradients of the solution along the magnetic field lines are typically much smaller than along a perpendicular direction. In toroidal geometry, field-aligned interpolation consists of a 1D interpolation along the field line, combined with 2D interpolations on the poloidal planes (at the intersections with the field line). A theoretical justification of the method is provided in the simplified context of constant advection on a 2D periodic domain: unconditional stability is proven, and error estimates are given which highlight the advantages of field-aligned interpolation. The same methodology is successfully applied to the solution of the gyrokinetic Vlasov equation, for which we present the ion temperature gradient (ITG) instability as a classical test-case: first we solve this in cylindrical geometry (screw-pinch), and next in toroidal geometry (circular Tokamak). In the first case, the algorithm is implemented in Selalib (semi-Lagrangian library), and the numerical simulations provide linear growth rates that are in accordance with the linear dispersion analysis. In the second case, the algorithm is implemented in the Gysela code, and the numerical simulations are benchmarked with those employing the standard (not aligned) scheme. Numerical experiments show that field-aligned interpolation leads to considerable memory savings for the same level of accuracy; substantial savings are also expected in reactor-scale simulations.

Keywords: plasma physics; gyrokinetics; semi-Lagrangian scheme; Tokamak plasma

Contents

[1] This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

1 Introduction

In a Tokamak, due to the large confining magnetic field, a fast homogenisation of the different physical quantities occurs along the magnetic field lines; this leads to very smooth and small variations along the field lines, whereas the scale length of the variations is very small (comparable to the gyro-radius) in a perpendicular direction. This should be taken into account for more efficient simulations. It is typically done by using field aligned coordinates in many gyrokinetic codes. However this approach has the drawback of needing a non-conformal correction after one turn, either in the poloidal or the toroidal direction, which yields a break of symmetry on one section of the torus. More importantly, field-aligned coordinates become singular when approaching the separatrix in a divertor configuration, with potentially serious consequences on the robustness of the numerical algorithm that employs them.

A very promising alternative, which is very flexible in regard to the choice of coordinates, has been introduced by Hariri and Ottaviani [1], and an equivalent approach by Stegmeier et al. [2]. The main idea is to compute the derivatives locally along the field lines, getting the needed values for finite differences by interpolation to the intersection points of a field line with the poloidal planes. We are interested here in a thorough numerical investigation of this idea in the context of gyrokinetic simulations using semi-Lagrangian methods. Pioneering in this sense is the recent work by Kwon, Yi, Piao and Kim [3], where "field-aligned interpolation" is employed in a semi-Lagrangian gyrokinetic code for full-f turbulence simulations. Our work complements the above on the numerical analysis side, and it focuses on the following topics: convergence analysis (i.e. stability proof and error estimates), numerical verification against analytical solutions, and benchmarking with the classical (not aligned) algorithm. The reader interested in the physics context can consult the review article [4], and exhaustive information about the semi-Lagrangian method which was introduced in the context of gyrokinetic simulations in [5] and the GYSELA code are provided in [6].

In this work we use the so-called 'backward' semi-Lagrangian method, which consists of an advection phase, where the characteristic trajectories ending at the grid points are traced back in time from $t + \Delta t$ to t, and an interpolation phase, where the particle distribution function is interpolated at the origin of these trajectories using the known grid values at time t. By virtue of the method of characteristics, the solution on the grid is therefore known at time $t + \Delta t$. Moreover, in the GYSELA code the motion is split between the poloidal plane and the toroidal direction (and also the parallel velocity, but this will not play any role in this paper). In this context, the idea of taking derivatives along magnetic field lines can be naturally extended to semi-Lagrangian methods by replacing the advection and interpolation in the toroidal (φ coordinate in the torus geometry) direction by an advection and interpolation along magnetic field lines (combining a φ and θ motion).

1.1 Model equations

We are interested in solving the gyrokinetic Vlasov equation

$$\left(\frac{\partial}{\partial t} + \mathbf{u}(t,\mathbf{x},v_\parallel,\mu)\cdot\nabla + a_\parallel(t,\mathbf{x},v_\parallel,\mu)\frac{\partial}{\partial v_\parallel}\right) f(t,\mathbf{x},v_\parallel,\mu) = 0, \tag{1.1}$$

where $(\mathbf{x}, v_\parallel, \mu)$ are the gyro-center phase-space coordinates: $\mathbf{x}$ is the gyro-center position, $v_\parallel$ is the parallel velocity of the gyro-center, and $\mu \approx mv_\perp^2/(2B)$ is the modified magnetic moment (which is an exact invariant of motion). The equilibrium magnetic field $\mathbf{B}(\mathbf{x})$ is assumed to be static ($B = \|\mathbf{B}\|$ is its magnitude). In a semi-Lagrangian method we use the fact that the exact solution to (1.1) is constant along the phase-space characteristics $\big(\mathbf{X}(t), V_\parallel(t), M(t)\big)$, namely

$$\frac{d}{dt} f(t,\mathbf{X}(t),V_\parallel(t),M(t)) = 0 \qquad \text{with} \begin{cases} \dfrac{d\mathbf{X}}{dt} = \ \mathbf{u}(t,\mathbf{X},V_\parallel,M), \\ \dfrac{dV_\parallel}{dt} = a_\parallel(t,\mathbf{X},V_\parallel,M), \\ \dfrac{dM}{dt} = \ 0. \end{cases} \tag{1.2}$$

The fields $\mathbf{u}$ and $a_\parallel$, and therefore the characteristic trajectories in (1.2), are completely defined by the modified magnetic field $\mathbf{B}^*(\mathbf{x}, v_\parallel) = \mathbf{B}(\mathbf{x}) + (m/q)v_\parallel\nabla\times\mathbf{b}(\mathbf{x})$, where m and q are the mass and charge of a particle, and $\mathbf{b} = \mathbf{B}/B$, and by the gyro-center Hamiltonian $H(t, \mathbf{x}, v_\parallel, \mu)$. Then, defining $B_\parallel^* = \mathbf{b}\cdot\mathbf{B}^* = B + mv_\parallel/(qB)\mathbf{b}\cdot\nabla\times\mathbf{B}$, we have (see for example [7])

$$\mathbf{u}(t,\mathbf{x},v_\parallel,\mu) = \frac{1}{B_\parallel^*}\left(\frac{1}{m}\frac{\partial H}{\partial v_\parallel}\mathbf{B}^* + \frac{1}{q}\mathbf{b}\times\nabla H\right), \tag{1.3a}$$

$$a_\parallel(t,\mathbf{x},v_\parallel,\mu) = \frac{1}{B_\parallel^*}\left(-\frac{1}{m}\mathbf{B}^*\cdot\nabla H\right). \tag{1.3b}$$

We now neglect μ and focus on the reduced phase-space $(\mathbf{x}, v_\parallel)$, where we define the phase-space velocity $\xi = (\mathbf{u}, a_\parallel)$. The phase-space divergence of this field is

$$\operatorname{div}(\xi) = \frac{1}{B_\parallel^*}\left[\nabla\cdot\left(B_\parallel^*\mathbf{u}\right) + \frac{\partial}{\partial v_\parallel}\left(B_\parallel^* a_\parallel\right)\right],$$

where $B_\parallel^*(\mathbf{x}, v_\parallel)$ is present because it is the Jacobian determinant of the coordinate transformation that was used to obtain the gyrokinetic Vlasov equation. It is straightforward to see that

$$\begin{cases} \nabla\cdot\left(B_\parallel^*\mathbf{u}\right) = \ \dfrac{1}{m}\mathbf{B}^*\cdot\nabla\left(\dfrac{\partial H}{\partial v_\parallel}\right) + \dfrac{1}{q}\left(\nabla\times\mathbf{b}\right)\cdot\nabla H \\ \dfrac{\partial}{\partial v_\parallel}\left(B_\parallel^* a_\parallel\right) = -\dfrac{1}{m}\mathbf{B}^*\cdot\nabla\left(\dfrac{\partial H}{\partial v_\parallel}\right) - \dfrac{1}{q}\left(\nabla\times\mathbf{b}\right)\cdot\nabla H \end{cases} \qquad \text{and therefore:} \qquad \operatorname{div}(\xi) = 0.$$

Since the phase-space velocity field is incompressible (i.e. divergence-free), an equivalent formulation of (1.1) is the conservation equation

$$\frac{\partial}{\partial t}\left(B_\parallel^* f\right) + \nabla\cdot\left(\mathbf{u}B_\parallel^* f\right) + \frac{\partial}{\partial v_\parallel}\left(a_\parallel B_\parallel^* f\right) = 0.$$

In the electrostatic case the gyro-center Hamiltonian reads

$$H(t, \mathbf{x}, v_\parallel, \mu) = \frac{1}{2} m v_\parallel^2 + \mu B(\mathbf{x}) + q\langle\phi\rangle_\alpha(t, \mathbf{x}),$$

where ϕ is the electrostatic potential, and $\langle\cdot\rangle_\alpha$ is the gyro-average operator. (In the zero-Larmor-radius limit, we simply have that $\langle\phi\rangle_\alpha = \phi$.)

In general one should solve one gyrokinetic Vlasov equation for each particle species, and couple these to a Poisson equation for the self-consistent ϕ. For simplicity, in this paper we model only one ion species kinetically, we assume an adiabatic response of the electrons, and we make use of the quasi-neutrality approximation [6]. Under these hypotheses the electrostatic potential $\phi(t, \mathbf{x})$ satisfies the integro-differential equation

$$-\left(\nabla_\perp \cdot \frac{\rho_{th,i}^2}{\lambda_{D,i}^2} \nabla_\perp\right)\phi + \frac{1}{\lambda_{D,e}^2}\left(\phi - \langle\phi\rangle_f\right) = \frac{\sigma_i}{\epsilon_0},$$

where $\nabla_\perp = \nabla - \mathbf{b}(\mathbf{b}\cdot\nabla)$ is the perpendicular gradient operator, $\langle\cdot\rangle_f$ represents an averaging operator over the whole magnetic flux surface passing through $\mathbf{x}$, $\rho_{th,i}(t, \mathbf{x})$ is the thermal ion Larmor radius, and $\lambda_{D,i}(t, \mathbf{x})$ and $\lambda_{D,e}(t, \mathbf{x})$ are the Debye lengths for ions and electrons respectively. On the right-hand-side, the ion charge density $\sigma_i(t, \mathbf{x})$ is reconstructed from the gyro-center distribution function f as

$$\sigma(t, \mathbf{x}) = q_i \int f(t, \mathbf{x}', v_\parallel, \mu)\, \delta(\mathbf{x}' + \boldsymbol{\rho} - \mathbf{x})\, B_\parallel^*\, d\mathbf{x}' dv_\parallel d\mu\, d\alpha,$$

where α is the gyro-phase angle and $\boldsymbol{\rho}(\mathbf{x}', \mu, \alpha)$ is the gyro-radius vector.

We refrain from giving more details on the equations here; the interested reader may refer for example to [4] and references therein. In fact, since our focus is on assessing the field-aligned interpolation method, and not on performing a realistic turbulence simulation, the equations presented in this section will be further simplified for implementation in the Selalib and Gysela codes (sections 4 and 5).

1.2 Magnetic configurations

For our numerical simulations in Gysela (see Section 5), we will consider a circular magnetic equilibrium in a torus as defined in [6], with magnetic field

$$\mathbf{B} = \frac{B_0 R_0}{R}\left(\zeta(r)\hat{\boldsymbol{\theta}} + \hat{\boldsymbol{\varphi}}\right), \quad \zeta(r) = \frac{r}{q(r) R_0}, \tag{1.4}$$

where R_0 and B_0 are respectively the major radius and the magnetic field intensity at the magnetic axis, $R(r, \theta) = R_0 + r\cos\theta$, and $q(r)$ is the classical safety factor in the large aspect ratio limit ($r/R_0 \to 0$). The unit vectors $(\hat{\mathbf{r}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\varphi}})$ form an orthogonal basis of $\mathbb{R}^3$ as long as $R > 0$. We notice that the magnetic field (1.4) depends on r and θ through R, but not on φ. In order to verify that $\nabla\cdot\mathbf{B} = 0$, we recall that the divergence of a vector $\mathbf{a} = a_r\hat{\mathbf{r}} + a_\theta\hat{\boldsymbol{\theta}} + a_\varphi\hat{\boldsymbol{\varphi}}$ in toroidal components reads

$$\nabla\cdot\mathbf{a} = \frac{1}{rR}\left[\frac{\partial}{\partial r}(rR\, a_r) + \frac{\partial}{\partial\theta}(Ra_\theta) + r\frac{\partial a_\varphi}{\partial\varphi}\right],$$

and we observe that the magnetic field in (1.4) has $B_r = 0$, $\partial_\theta(RB_\theta) = 0$ and $\partial_\varphi B_\varphi = 0$. With regard to the unit vector $\mathbf{b}$, we have $b_r = 0$, $b_\theta = \zeta/\sqrt{1+\zeta^2}$ and $b_\varphi = 1/\sqrt{1+\zeta^2}$, so that $\nabla\cdot\mathbf{b} = -(b_\theta\sin\theta)/R \neq 0$. The magnetic field lines, parametrized by (r, θ, φ), are defined by the equations

$$\frac{\mathrm{d}r}{\mathrm{d}s} = B_r = 0, \quad r\frac{\mathrm{d}\theta}{\mathrm{d}s} = B_\theta = \frac{B_0 R_0}{R}\zeta(r), \quad R\frac{\mathrm{d}\varphi}{\mathrm{d}s} = B_\varphi = \frac{B_0 R_0}{R}. \tag{1.5}$$

From this it follows that

$$\frac{\mathrm{d}\theta}{\mathrm{d}s} = R\frac{\zeta(r)}{r}\frac{\mathrm{d}\varphi}{\mathrm{d}s} = \frac{1 + (r/R_0)\cos\theta}{q(r)}\frac{\mathrm{d}\varphi}{\mathrm{d}s},$$

and so in tokamaks with a large aspect ratio (i.e. with small r/R_0) the magnetic field lines are almost, but not exactly, straight lines in the (θ, φ) plane for a given r.

For our numerical simulations in Selalib (see Section 4) we shall consider the simplified case of a straight periodic cylinder, which amounts to taking $R = R_0$ in (1.4), and replacing the toroidal angular variable φ by a straight variable z. Then

$$\mathbf{B} = B_0 \left(\zeta(r)\hat{\boldsymbol{\theta}} + \hat{\mathbf{z}} \right), \quad \zeta(r) = \frac{\iota(r) r}{R_0}. \tag{1.6}$$

We see that the magnetic field is characterized by its central modulus B_0, the major radius R_0 and the rotational transform iota, which satisfies

$$\iota(r) = \frac{b_\theta / r}{b_z / R_0} = \frac{1}{q(r)}. \tag{1.7}$$

In order to verify that $\nabla \cdot \mathbf{B} = 0$, we recall that the divergence of a vector $\mathbf{a} = a_r \hat{\mathbf{r}} + a_\theta \hat{\boldsymbol{\theta}} + a_z \hat{\mathbf{z}}$ in cylindrical components reads

$$\nabla \cdot \mathbf{a} = \frac{1}{r} \frac{\partial}{\partial r}(r a_r) + \frac{1}{r} \frac{\partial a_\theta}{\partial \theta} + \frac{\partial a_z}{\partial z},$$

and we observe that the magnetic field in (1.6) has $B_r = 0$, $\partial_\theta B_\theta = 0$ and $\partial_z B_z = 0$. In a similar fashion we also notice that $\nabla \cdot \mathbf{b} = 0$ in this case. The magnetic field lines, parametrized by (r, θ, z), are defined by the equations

$$\frac{\mathrm{d}r}{\mathrm{d}s} = 0, \quad r \frac{\mathrm{d}\theta}{\mathrm{d}s} = B_\theta = B_0 \zeta(r), \quad \frac{\mathrm{d}z}{\mathrm{d}s} = B_z = B_0, \tag{1.8}$$

so that

$$\frac{\mathrm{d}\theta}{\mathrm{d}s} = \frac{\zeta(r)}{r} \frac{\mathrm{d}z}{\mathrm{d}s} = \frac{\iota(r)}{R_0} \frac{\mathrm{d}z}{\mathrm{d}s}.$$

Therefore the magnetic field lines are straight oblique lines in the (θ, z) plane for each given r.

1.3 Overview

The remainder of the paper is structured as follows. Section 2 describes the numerical algorithms that are employed for performing interpolation and differentiation in a 'field-aligned' fashion. Section 3 details the field-aligned semi-Lagrangian scheme in the simplified setting of the constant advection equation in 2D, and provides a rigorous proof of unconditional stability, together with an extensive error analysis. Section 4 presents a simplified gyrokinetic model in cylindrical geometry (screw pinch), which is implemented in Selalib (semi-Lagrangian library) and verified against an analytical solution. Section 5 presents a gyrokinetic model in toroidal geometry (circular Tokamak), which is implemented in the Gysela code and benchmarked against a standard (not aligned) version of the same code. Finally, Section 6 gives our conclusions and an outlook on possible future investigations.

2 Description of the Numerical Tools

2.1 Numerical scheme for a 2D aligned interpolation

To describe the 2D aligned interpolation method, we consider here a 2D plane along the dimensions θ and φ for example. Let us consider $\Delta\theta = 2\pi/N_\theta$, $\theta_i = i\,\Delta\theta$ and $\Delta\varphi = 2\pi/N_\varphi$, $\varphi_j = j\,\Delta\varphi$ with $(i,j) \in [0..N_\theta - 1] \times [0..N_\varphi - 1]$ to discretize the 2D plane. By periodicity we can extend this to $(i,j) \in \mathbb{Z}^2$. Let us consider a position $(\theta^\star, \varphi^\star)$ where we want to interpolate a function g, given that the values $g(\theta_i, \varphi_j)$ are already known. There exists a unique index $j^\star \in \mathbb{Z}$ and $0 \leq \beta < \Delta\varphi$ such that

$$\varphi^\star = \varphi_{j^\star} + \beta \ .$$

We then define

$$\varphi_{j^\star + k} = \varphi_{j^\star} + k\,\Delta\varphi, \quad k = r, .., s \ .$$

We will use information stored in the 1D slices $g(\theta = *, \varphi = \varphi_{j^\star + k})_{k=r,..,s}$ to perform the aligned interpolation at $(\theta^\star, \varphi^\star)$. Let us define a function *fieldline*$_\theta(\theta, \varphi, j)$ that gives a θ-value that corresponds to the intersection of the field line (or an approximation of the field line) that passes by the point (θ, φ) and the line $(\theta = *, \varphi_j)$. This function is the cornerstone of the method, it provides a way to interpolate using values that are close

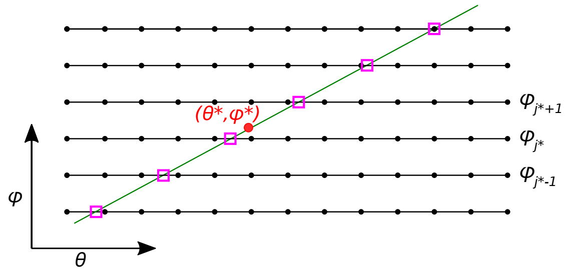

Figure 1: Illustration of the aligned interpolation scheme for a target point at position $(\theta^\star, \varphi^\star)$; the squares are located at $(\theta = \mathit{fieldline}_\theta(\theta^\star, \varphi^\star, j^\star + k), \varphi = \varphi_{j^\star+k})_{k=r,..,s}$; the values at square positions are interpolated using values known at black small points; the value at the red circle position (θ^*, φ^*) is interpolated using values known at the square positions.

each other, because locations of these values are aligned on the physical structures. The $\mathit{fieldline}_\theta$ function is chosen such as all interpolated points h_k are aligned on a single field line.

The first stage of the method is to compute $u_{\theta^\star,\varphi^\star}(k)_{k=r,..,s}$ by interpolating g at positions $(\mathit{fieldline}_\theta(\theta^\star, \varphi^\star, j^\star + k), \varphi_{j^\star+k})_{k=r,..,s}$. We currently employ cubic splines to interpolate along the θ direction on the 1D slices $g(\theta = *, \varphi = \varphi_{j^\star+k})_{k=r,..,s}$. The formula for $\mathit{fieldline}_\theta$ that we have been using so far is the linear approximation

$$\mathit{fieldline}_\theta(\theta^\star, \varphi^\star, j^\star + k) = \theta^\star + \iota(r)\left(\varphi_{j^\star+k} - \varphi^\star\right) ,$$

which is the equation of a straight line. This approximation is exact in the case of the screw-pinch described in Section 4, and it is very accurate in the case of the circular Tokamak in Section 5 (because of its large aspect-ratio). In any case, this function can be easily changed in the code: it is effectively a parameter of the method.

The second stage of the method consists in interpolating $g(\theta^\star, \varphi^\star)$ using the values aligned on the parallel direction we just get: $u_{\theta^\star,\varphi^\star}(k)_{k=r,..,s}$. To achieve this, we use Lagrange polynomials of degree $2d + 1$ LAG(2d+1) and take $r = -d, s = d + 1$. The pseudo-code implementation of the scheme is presented in Algorithm 1, and an illustration is given in Figure 1.

```
Input  : $g$, $theta^\star$, $phi^\star$
Output : $g^\dagger$

for $j = 0, N_\varphi$ do
    $\eta(i = *, j) \leftarrow$ spline coefficients for
    $g(i = *, j)$

for $j = 0, N_\varphi$ do
    for $i = 0, N_\theta$ do
        $\varphi^\star \leftarrow phi^\star(i, j)$; $\theta^\star \leftarrow theta^\star(i, j)$;
        $j^\star \leftarrow$ index of the left grid
              point close to $\varphi^\star$ ;
        for $k = -d, d{+}1$ do
            $\theta_k \leftarrow fieldline_\theta(\theta^\star, \varphi^\star, j^\star + k)$;
            $u_k \leftarrow$ 1D spline interpolation along
            $\theta$
                  at $\theta_k$ using $\eta(i = *, j^\star + k)$;
        $g^\dagger(i, j) \leftarrow$ 1D Lagrange interpolation
              using values $(u_k)_{k=-d,d+1}$
```

Algorithm 1: Aligned interpolation in 2D

```
Input  : $g$, $\epsilon$
Output : $dg/d\varphi$

for $j = 0, N_\varphi$ do
    $\eta(i = *, j) \leftarrow$ spline coefficients for $g(i = *, j)$

for $j = 0, N_\varphi$ do
    for $i = 0, N_\theta$ do
        for $k = -d, d{+}1$ do
            $\theta_k^+ \leftarrow fieldline_\theta(\theta_i, \varphi_j + \epsilon, j + k)$;
            $\theta_k^- \leftarrow fieldline_\theta(\theta_i, \varphi_j - \epsilon, j + k)$;
            $u_k^+ \leftarrow$ 1D spline interpolation along $\theta$
                  at $\theta_k^+$ using $\eta(i = *, j + k)$;
            $u_k^- \leftarrow$ 1D spline interpolation along $\theta$
                  at $\theta_k^-$ using $\eta(i = *, j + k)$;
        $u^+ \leftarrow$ 1D Lagrange interpolation
              using values $(u_k^+)_{k=-d,d+1}$;
        $u^- \leftarrow$ 1D Lagrange interpolation
              using values $(u_k^-)_{k=-d,d+1}$;
        $\frac{dg}{d\varphi}(i, j) \leftarrow \frac{u^+ + u^-}{2\epsilon}$
```

Algorithm 2: Derivatives along φ with aligned scheme

2.2 Algorithm for the aligned computation of derivatives

In the Gysela code (in Section 5), we need to evaluate φ derivatives of the electric potential to compute the non-linear terms appearing in the advection equations, but also in the diagnostics that compute a set of macroscopic physics variables. In order to do so with a reduced number of points in the φ direction (authorized by the aligned interpolation approach), a scheme should be designed to get an accurate approximation of these derivatives. We have evaluated two alternatives to estimate the $d\Phi/d\varphi$ derivative: the first one relies on estimating the derivative along the parallel direction $\mathbf{b}$ and then projecting over the φ direction, the second one uses aligned interpolation to compute two values of Φ at $\varphi \pm \epsilon$ and then computes the derivative by a finite difference formula. Algorithm 2 describes the second solution, which is effectively used in the Gysela code. The main idea is to compute $\Phi(\theta_i, \varphi_j)$ with the values of $\Phi(\theta_i, \varphi_j \pm \epsilon)$ that are accurately estimated with an aligned interpolation similar to Algorithm 1.
In the Selalib code (in Section 4), we do not need to evaluate φ derivatives of the electric potential, as in the equations everything is expressed in terms of derivatives in the poloidal plane or along the parallel direction $\mathbf{b}$. To evaluate the electric field along $\mathbf{b}$, we use for example a finite difference formula of order 6, which reads

$$\nabla\Phi \cdot \mathbf{b}(r_i, \theta_j, \varphi_k) \simeq \frac{1}{\Delta\varphi} \sum_{\ell=-3}^{3} w_\ell \tilde{\Phi}(r_i, \mathit{fieldline}_\theta(\theta_j, \varphi_k, k+\ell), \varphi_{k+\ell}),$$

with coefficients

$$w_0 = 0,\ w_1 = -w_{-1} = \frac{3}{4},\ w_2 = -w_{-2} = -\frac{3}{20},\ w_3 = -w_{-3} = \frac{1}{60},$$

and $\tilde{\Phi}(r_i, \mathit{fieldline}_\theta(\theta_j, \varphi_k, k+\ell), \varphi_{k+\ell})$ is obtained by interpolation (for example cubic splines) from the values $\Phi(r_i, \theta_l, \varphi_{k+\ell}),\ l = 0, \ldots, N_\theta$.

3 Theoretical Justification of the Approach for 2D Advection

Our drift-kinetic simulations with the GYSELA code (presented in section 5) will emphasize the practical advantages of the field-aligned approach over traditional tensor-product 2D interpolation schemes. Nevertheless, in order to trust the output of such a code when no analytical solutions are at hand, it is very desirable to have proven convergence (that is, consistency and stability) of the numerical methods employed. As it often happens in computational physics, we can provide such a proof only for a drastically reduced mathematical model; but even so, we gain useful insight and a certain degree of confidence in the final numerical scheme. Therefore, in this section we assess the convergence of our field-aligned semi-Lagrangian method when applied to the 2D constant advection equation for $f\colon \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R}$,

$$(\partial_t + b_\theta \partial_\theta + b_\varphi \partial_\varphi) f(t, \theta, \varphi) = 0, \qquad f(t=0, \theta, \varphi) = f_0(\theta, \varphi),$$

where the initial function $f_0 : \mathbb{R}^2 \to \mathbb{R}$ is 2π-periodic in θ and φ, and $\mathbf{b} = (b_\theta, b_\varphi)$ is the unit vector of a constant magnetic field (therefore $b_\theta, b_\varphi \in \mathbb{R}$ such that $b_\theta^2 + b_\varphi^2 = 1$). We assume that $b_\varphi \neq 0$, because this hypothesis is required by the scheme. The exact solution reads

$$f(t, \theta, \varphi) = f_0(\theta - b_\theta t, \varphi - b_\varphi t),$$

while the numerical solution $f^n_{i,j} \approx f(t_n, \theta_i, \varphi_j)$ is computed on a uniform grid with indices $n, i, j \in \mathbb{Z}$ and discretization parameters $\Delta t \in \mathbb{R}$, $\Delta\theta = \frac{2\pi}{N_\theta}$, and $\Delta\varphi = \frac{2\pi}{N_\varphi}$, where $N_\theta, N_\varphi \in \mathbb{N}^*$. Specifically, we have $t_n = n\Delta t$ and $(\theta_i, \varphi_j) = (\theta_0 + i\Delta\theta, \varphi_0 + j\Delta\varphi)$ with $\theta_0, \varphi_0 \in \mathbb{R}$. In our 'backward' semi-Lagrangian scheme, the solution at time t_{n+1} is obtained from the solution at time t_n as

$$f^{n+1}_{i,j} = f^n(\theta_i - b_\theta \Delta t, \varphi_j - b_\varphi \Delta t),$$

where $f^n(\theta, \varphi)$ is reconstructed from $f^n_{i,j}$ through field-aligned interpolation. By virtue of the linearity of the interpolation operator (essentially a linear discrete convolution), the stability of the scheme will be assessed by means of a standard Von Neumann analysis: upon taking the semi-discrete Fourier transform on both sides of the previous equation, we will get

$$\hat{f}^{n+1}(\omega_\theta, \omega_\varphi) = \rho(\omega_\theta, \omega_\varphi) \hat{f}^n(\omega_\theta, \omega_\varphi),$$

where $\rho : [-\pi, \pi] \times [-\pi, \pi] \to \mathbb{C}$ is the 'Fourier symbol', or 'amplification factor', for a given choice of discretization parameters. We will then prove stability by showing that

$$|\rho(\omega_\theta, \omega_\varphi)| \leq 1 \qquad \forall\ \omega_\theta, \omega_\varphi.$$

In the present version of our field-aligned semi-Lagrangian scheme, we make use of centered Lagrange interpolation along both directions $\mathbf{b}$ and θ. In particular, we assume odd order $2d_b + 1$ along $\mathbf{b}$, and $2d_\theta + 1$ along θ, with $d_b, d_\theta \in \mathbb{N}$. For any choice of (d_b, d_θ), we prove that such a scheme is unconditionally stable, i.e. stable for all values of $(\Delta t, N_\theta, N_\varphi)$,

Finally, we analyze the truncation error for single-mode initial conditions, proving that the scheme converges to the exact solution with order $2d_\theta + 2$ in N_θ and $2d_b + 2$ in N_φ, as expected. Our error estimates correctly recover the asymptotic behavior for $b_\theta \to 0$, where the scheme reduces to 1D Lagrange interpolation. In comparison to classical tensor-product 2D interpolation, we clarify how field-aligned interpolation allows for a reduced N_φ in those situations where the gradients along $\mathbf{b}$ are smaller than along the φ direction, as typical in magnetic confinement devices. Because of the additional interpolations along θ, our estimates suggest a slight increase in the error constant along this direction, but we expect such an effect to be negligible in practice. In fact, the numerical experiments in sections 4 and 5 will confirm that this is more than compensated by the gain along φ.

The outline of this section is as follows: section 3.1 provides the explicit update formula of our field-aligned scheme, section 3.2 gives a rigorous proof of unconditional stability, and section 3.3 assesses the truncation error of our scheme and compares it with the standard (not field-aligned) algorithm.

3.1 Update formula for field-aligned semi-Lagrangian scheme

For any given grid point (θ_i, φ_j), we trace the magnetic field line backward in time to obtain the foot of the characteristic $(\theta_i^*, \varphi_j^*)$, where $\varphi_j^* \in [\varphi_{j^*}, \varphi_{j^*+1})$. Since $b_\varphi \neq 0$ by assumption, the same magnetic field line intersects the grid lines at constant φ at the locations $(\theta_{i,k}^*, \varphi_{j^*+k})$ with $k \in \mathbb{Z}$. The basic idea of the field-aligned semi-Lagrangian method is to use 1D interpolation along θ to obtain the intermediate values $f_{i,j,k}^{n+1} = f^n(\theta_{i,k}^*, \varphi_{j^*+k})$, and then 1D interpolation along $\mathbf{b}$ to obtain $f_{i,j}^{n+1} = f^n(\theta_i^*, \varphi_j^*)$. Thanks to the constant $\mathbf{b}$ and uniform discretization, the concepts above will be succinctly formalized in the following discussion, leading to a very compact algorithm.

First, we consider the normalized displacement $-b_\varphi \Delta t / \Delta\varphi$ along the φ direction and decompose it into its integer and fractional parts:

$$-b_\varphi \Delta t = (r_\varphi + \alpha_\varphi)\Delta\varphi, \qquad r_\varphi \in \mathbb{Z}, \quad 0 \leq \alpha_\varphi < 1.$$

For Lagrange interpolation along $\mathbf{b}$, the integer shift r_φ is used to correctly place the stencil on the grid, and α_φ is the interpolation variable. We now turn to finding the displacements in the θ coordinate, which correspond to the intersections between the magnetic field line and the various grid lines at constant φ. For this purpose, we first define the flight times Δt_k such that

$$-b_\varphi \Delta t_k = (r_\varphi + k)\Delta\varphi, \qquad k = -d_b, \ldots, d_b + 1.$$

This is possible, as $b_\varphi \neq 0$. At each φ_{j^*+k}–intersection we now have the normalized diplacements along the θ direction as $-b_\theta \Delta t_k / \Delta\theta$, which we also decompose into integer and fractional parts:

$$-b_\theta \Delta t_k = (r_{\theta,k} + \alpha_{\theta,k})\Delta\theta, \qquad r_{\theta,k} \in \mathbb{Z}, \quad 0 \leq \alpha_{\theta,k} < 1.$$

For Lagrange interpolation along θ, the integer shifts $r_{\theta,k}$ are used to correctly place each stencil k on the grid, and $\alpha_{\theta,k}$ are the interpolation variables. We are now ready to compute the intermediate values $f_{i,j,k}^{n+1}$ at each φ_{j^*+k}–intersection through Lagrange interpolation along θ, as

$$f_{i,j,k}^{n+1} = \sum_{\ell=-d_\theta}^{d_\theta+1} L_\ell^{d_\theta}(\alpha_{\theta,k}) f_{i+r_{\theta,k}+\ell,\, j+r_\varphi+k}^{n}. \qquad k = -d_b, \ldots, d_b + 1,$$

and from these we compute the new solution $f_{i,j}^{n+1}$, using Lagrange interpolation along $\mathbf{b}$:

$$f_{i,j}^{n+1} = \sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi) f_{i,j,k}^{n+1}.$$

Combining the last two equations leads to the compact update formula

$$f_{i,j}^{n+1} = \sum_{\ell=-d_\theta}^{d_\theta+1} \sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi) L_\ell^{d_\theta}(\alpha_{\theta,k}) f_{i+r_{\theta,k}+\ell,j+r_\varphi+k}^n. \tag{3.1}$$

Here we recall that $i, j \in \mathbb{Z}$, and that $f_{i,j}^0$ is N_θ-periodic in i and N_φ-periodic in j. As a result, $f_{i,j}^n$ is N_θ-periodic in i and N_φ-periodic in j for $n \in \mathbb{N}$. For completeness, we also recall that L_k^d are the elementary Lagrange basis functions defined by

$$L_k^d(\alpha) = \prod_{\ell=-d,\ \ell\neq k}^{d+1} \frac{\alpha-\ell}{k-\ell}, \quad k=-d,\dots,d+1,\ \alpha\in\mathbb{R},\ d\in\mathbb{N}.$$

3.2 Proof of stability

3.2.1 Fourier symbol

We now turn to studying the Fourier symbol of our numerical scheme (3.1) and, for simplicity, we redefine $i := \sqrt{-1}$ as the imaginary unit. Because of its periodicity, the Fourier spectrum of $f_{i,j}^n$ contains only $N_\theta \times N_\varphi$ modes. Therefore, we can proceed by taking the 2D discrete Fourier transform of both sides of (3.1). For $i_1, j_1 \in \mathbb{Z}$ we get

$$\begin{aligned}&\sum_{i_2=0}^{N_\theta-1} \sum_{j_2=0}^{N_\varphi-1} f_{i_2,j_2}^{n+1} \exp\left(2\pi i \frac{i_1 i_2}{N_\theta}\right) \exp\left(2\pi i \frac{j_1 j_2}{N_\varphi}\right) = \\ &= \rho\left(\frac{2\pi i_1}{N_\theta}, \frac{2\pi j_1}{N_\varphi}\right) \sum_{i_2=0}^{N_\theta-1} \sum_{j_2=0}^{N_\varphi-1} f_{i_2,j_2}^{n} \exp\left(2\pi i \frac{i_1 i_2}{N_\theta}\right) \exp\left(2\pi i \frac{j_1 j_2}{N_\varphi}\right),\end{aligned}$$

where the Fourier symbol $\rho\colon [-\pi,\pi]^2 \to \mathbb{C}$ is

$$\rho(\omega_\theta, \omega_\varphi) = \sum_{\ell=-d_\theta}^{d_\theta+1} \sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi) L_\ell^{d_\theta}(\alpha_{\theta,k}) \exp(i(r_{\theta,k}+\ell)\omega_\theta) \exp(i(r_\varphi+k)\omega_\varphi).$$

Thanks to the relation

$$r_{\theta,k} + \alpha_{\theta,k} = (r_\varphi + k)\lambda, \qquad \lambda = \frac{b_\theta N_\theta}{b_\varphi N_\varphi} \in \mathbb{R},$$

we can parametrize the symbol in the variables $(\lambda, r_\varphi, \alpha_\varphi)$ as

$$\rho_{\lambda,r_\varphi,\alpha_\varphi}(\omega_\theta,\omega_\varphi) = \sum_{\ell=-d_\theta}^{d_\theta+1} \sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi) L_\ell^{d_\theta}((r_\varphi+k)\lambda - \lfloor (r_\varphi+k)\lambda \rfloor) \exp(i(\lfloor (r_\varphi+k)\lambda \rfloor + \ell)\omega_\theta) \exp(i(r_\varphi+k)\omega_\varphi), \tag{3.2}$$

where $\lfloor\cdot\rfloor : \mathbb{R} \to \mathbb{Z}$ is the floor function. In the spirit of the Von Neumann stability analysis, we are now led to compute the maximum absolute value S of the symbol above,

$$S = \sup_{0\le\alpha_\varphi<1,\ r_\varphi\in\mathbb{Z},\ \lambda,\omega_\theta,\omega_\varphi\in\mathbb{R}} \left|\rho_{\lambda,r_\varphi,\alpha_\varphi}(\omega_\theta,\omega_\varphi)\right| = \sup_{0\le\alpha_\varphi,\frac{\omega_\theta}{2\pi},\frac{\omega_\varphi}{2\pi}<1,\ r_\varphi\in\mathbb{Z},\ \lambda\in\mathbb{R}} \left|\rho_{\lambda,r_\varphi,\alpha_\varphi}(\omega_\theta,\omega_\varphi)\right|,$$

and to prove that $S \le 1$.

3.2.2 Relation to discrete Fourier transform (DFT) for rational λ

We suppose for the moment that $\lambda \in \mathbb{Q}$, and we represent it as

$$\lambda = \frac{m}{q}, \qquad \text{with } m \in \mathbb{Z},\ q \in \mathbb{N}^*, \text{ and } m, q \text{ coprime}.$$

So, for any $r_\varphi \in \mathbb{Z}$, we observe that $\alpha_{\theta,k}$ can only assume at most q different values, all rational:

$$\alpha_{\theta,k} = (r_\varphi + k)\lambda \bmod 1 = \frac{(r_\varphi + k)m \bmod q}{q} = \frac{s_k}{q}, \qquad k = -d_b, \dots, d_b + 1,$$

where we have introduced the natural sequence

$$s_k = (r_\varphi + k)m \bmod q \ \in \ \{0, \dots, q-1\}.$$

Under this assumption for λ we can use the following identity for any complex sequence a_k:

$$\sum_{k=-d_b}^{d_b+1} a_k = \sum_{k=-d_b}^{d_b+1} \sum_{p=0}^{q-1} \delta_{p,s_k} a_k = \sum_{p=0}^{q-1} \sum_{k=-d_b}^{d_b+1} \delta_{p,s_k} a_k.$$

Here $\delta\colon \mathbb{Z}^2 \to \{0,1\}$ is Kronecker's delta: for any $u, v \in \mathbb{Z}$, $\delta_{u,v} = 1$ if $u = v$, and 0 otherwise. In particular, if we let $a_k = G(\alpha_{\theta,k})c_k$ with $G\colon [0,1) \to \mathbb{C}$ and $c_k \in \mathbb{C}$, we get the important relation

$$\sum_{k=-d_b}^{d_b+1} G(\alpha_{\theta,k})c_k = \sum_{p=0}^{q-1} \sum_{k=-d_b}^{d_b+1} \delta_{p,s_k}\ G\left(\frac{s_k}{q}\right) c_k = \sum_{p=0}^{q-1} G\left(\frac{p}{q}\right) \sum_{k=-d_b}^{d_b+1} \delta_{p,s_k} c_k.$$

We can then write

$$\begin{aligned}
&\rho_{\lambda,r_\varphi,\alpha_\varphi}(\omega_\theta,\omega_\varphi) = \\
&\quad = \sum_{\ell=-d_\theta}^{d_\theta+1} \sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi) L_\ell^{d_\theta}(\alpha_{\theta,k}) \exp(i((r_\varphi + k)\lambda - \alpha_{\theta,k} + \ell)\omega_\theta) \exp(i(r_\varphi + k)\omega_\varphi) \\
&\quad = \sum_{p=0}^{q-1} \left[\sum_{\ell=-d_\theta}^{d_\theta+1} L_\ell^{d_\theta}\left(\frac{p}{q}\right) \exp\left(i\left(\ell - \frac{p}{q}\right)\omega_\theta\right)\right] \sum_{k=-d_b}^{d_b+1} \delta_{p,s_k} L_k^{d_b}(\alpha_\varphi) \exp(i(r_\varphi + k)(\omega_\varphi + \lambda\omega_\theta)).
\end{aligned}$$

We now focus on the term between square brackets, a complex sequence $w_p \in \mathbb{C}$ with $p = 0, \dots, q-1$, and we represent it as a sum of q Fourier modes by means of a discrete Fourier transform (DFT) and its inverse:

$$\sum_{\ell=-d_\theta}^{d_\theta+1} L_\ell^{d_\theta}\left(\frac{p}{q}\right) \exp\left(i\left(\ell - \frac{p}{q}\right)\omega_\theta\right) = w_p = \sum_{p_1=0}^{q-1} t_{p_1} \exp\left(i2\pi\frac{p\,p_1}{q}\right), \tag{3.3a}$$

$$t_{p_1} = \frac{1}{q}\sum_{p_2=0}^{q-1} w_{p_2} \exp\left(-i2\pi\frac{p_1 p_2}{q}\right). \tag{3.3b}$$

Incidentally, we notice that the sum of the Fourier coefficients defined in (3.3b) is equal to 1, as can be proven by looking at the term w_0 in (3.3a):

$$\sum_{p_1=0}^{q-1} t_{p_1} = w_0 = \sum_{\ell=-d_\theta}^{d_\theta+1} L_\ell^{d_\theta}(0) \exp(i\ell\omega_\theta) = \sum_{\ell=-d_\theta}^{d_\theta+1} \delta_{\ell,0} \exp(i\ell\omega_\theta) = 1. \tag{3.4}$$

By substituting (3.3a) into the Fourier symbol $\rho_{\lambda,r_\varphi,\alpha_\varphi}(\omega_\theta,\omega_\varphi)$ we can get rid of one of the sums, as well as of the Kronecker delta:

$$\begin{aligned}
&\rho_{\lambda,r_\varphi,\alpha_\varphi}(\omega_\theta,\omega_\varphi) = \\
&\quad = \sum_{p_1=0}^{q-1} t_{p_1} \sum_{p=0}^{q-1} \exp\left(i2\pi\frac{p_1 p}{q}\right) \sum_{k=-d_b}^{d_b+1} \delta_{p,s_k} L_k^{d_b}(\alpha_\varphi) \exp(i(r_\varphi + k)(\omega_\varphi + \lambda\omega_\theta)) \\
&\quad = \sum_{p_1=0}^{q-1} t_{p_1} \sum_{p=0}^{q-1} \sum_{k=-d_b}^{d_b+1} \delta_{p,s_k} L_k^{d_b}(\alpha_\varphi) \exp(i(r_\varphi + k)(\omega_\varphi + \lambda\omega_\theta)) \exp\left(i2\pi\frac{p_1 s_k}{q}\right) \\
&\quad = \sum_{p_1=0}^{q-1} t_{p_1} \sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi) \exp(i(r_\varphi + k)(\omega_\varphi + \lambda\omega_\theta)) \exp(i2\pi p_1 \alpha_{\theta,k}).
\end{aligned}$$

Because $\exp(i2\pi) = 1$, we now multiply the right-hand side by $\exp(i2\pi p_1 r_{\theta,k}) = 1$, use the fact that $r_{\theta,k} + \alpha_{\theta,k} = (r_\varphi + k)\lambda$, and then change p_1 with p to obtain

$$\rho_{\lambda,r_\varphi,\alpha_\varphi}(\omega_\theta,\omega_\varphi) = \sum_{p=0}^{q-1} t_p \sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi)\exp(i(r_\varphi+k)(\omega_\varphi+\lambda\omega_\theta))\exp(i2\pi p(r_\varphi+k)\lambda).$$

By properly rearranging the complex exponential factors according to their dependence on the indexes p and k, we also get

$$\rho_{\lambda,r_\varphi,\alpha_\varphi}(\omega_\theta,\omega_\varphi) = \sum_{p=0}^{q-1} t_p \exp(ir_\varphi\omega_p) \sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi)\exp(ik\omega_p), \tag{3.5a}$$

where we have introduced the frequencies $\omega_p \in \mathbb{R}$ for $p = 0,\ldots,q-1$ as

$$\omega_p = 2\pi p\lambda + \omega_\varphi + \lambda\omega_\theta. \tag{3.5b}$$

We now turn to studying the absolute value of the Fourier symbol in (3.5), which is the sum over p of q complex terms. We apply the triangular inequality to such a sum, and use the fact that the modulus of a complex exponential is equal to 1, to obtain the estimate

$$\left|\rho_{\lambda,r_\varphi,\alpha_\varphi}(\omega_\theta,\omega_\varphi)\right| \le \sum_{p=0}^{q-1} |t_p| \left| \sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi)\exp(ik\omega_p)\right|,$$

which can be factorized as

$$\left|\rho_{\lambda,r_\varphi,\alpha_\varphi}(\omega_\theta,\omega_\varphi)\right| \le \left(\sum_{p=0}^{q-1} |t_p|\right)\left(\sup_{0\le\omega\le2\pi}\left|\sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi)\exp(ik\omega)\right|\right). \tag{3.6}$$

The second factor on the right-hand side is typical of backward semi-Lagrangian schemes applied to the 1D advection equation. The stability analysis in [8], for example, has already shown that

$$\sup_{0\le\omega\le2\pi}\left|\sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi)\exp(ik\omega)\right| \le 1. \tag{3.7}$$

Therefore our attention will focus on the first factor, which must also be ≤ 1. If we can prove that $t_p \in \mathbb{R}^+$ for each $p = 0,\ldots,q-1$, then $|t_p| = t_p$ and we can use our previous result in (3.4) to obtain

$$\sum_{p=0}^{q-1} |t_p| = \sum_{p=0}^{q-1} t_p = 1. \tag{3.8}$$

For this purpose we first notice, because $\exp(i\ell\, 2\pi p) = 1$, that we can rewrite (3.3b) as

$$t_p = \frac{1}{q}\sum_{p_1=0}^{q-1}\sum_{\ell=-d_\theta}^{d_\theta+1} L_\ell^{d_\theta}\left(\frac{p_1}{q}\right)\exp\left(i\left(\ell-\frac{p_1}{q}\right)(\omega_\theta+2\pi p)\right), \qquad p = 0,\ldots,q-1. \tag{3.9}$$

Our stability analysis will now proceed in three stages. In section 3.2.3 we will prove that the Fourier coefficients t_p are all real, and in section 3.2.4 that they are non-negative. This implies (3.8) and therefore stability of our numerical scheme for any rational λ, according to (3.6) and (3.7). Finally, in section 3.2.5 we will extend this result to the general situation of real λ.

Remark 3.1. *This result of positivity of the DFT has direct connection with results of Ferretti [9, 10] stating equivalence between semi-Lagrangian and Lagrange-Galerkin methods under some assumptions, one of it being the positivity of the (continuous) Fourier transform. Such link may be further studied.*

3.2.3 Proving that the DFT is real

We now prove that $t_p \in \mathbb{R}$ for each $p = 0, \dots, q-1$. Given that the t_p coefficients are obtained through the DFT (3.3), this follows from the symmetry property $w_{q-p} = w_p^*$ for $p = 1, \dots, q-1$.
If we introduce the complex function of real variable

$$S_{q,d}(\omega) = \sum_{p_1=0}^{q-1} \sum_{\ell=-d}^{d+1} L_\ell^d \left(\frac{p_1}{q}\right) \exp\left(i\left(\ell - \frac{p_1}{q}\right)\omega\right), \qquad 0 \le \omega \le 2\pi q, \tag{3.10}$$

such that

$$t_p = \frac{1}{q} S_{q,d_\theta}(\omega_\theta + 2\pi p), \qquad p = 0, \dots, q-1, \tag{3.11}$$

it suffices to prove that the imaginary part of $S_{q,d}(\omega)$ is always zero in $[0, 2\pi q]$. To show this, we start from Euler's formula $\exp(ix) = \cos(x) + i\sin(x)$ and then make use of the symmetry of Lagrange basis functions on a uniform grid, namely

$$L_\ell^d(\alpha) = L_{-\ell+1}^d(1-\alpha), \qquad \ell = 1, \dots, d+1, \qquad 0 \le \alpha \le 1,$$

together with the identities

$$\sum_{\ell=-d}^{d+1} c_\ell = \sum_{\ell=1}^{d+1} (c_\ell + c_{-\ell+1}), \qquad \sum_{p=1}^{q-1} c_p = \sum_{p=1}^{q-1} c_{q-p},$$

to obtain

$$\begin{aligned}
\operatorname{Im}\{S_{q,d}(\omega)\} &= \sum_{p=0}^{q-1} \sum_{\ell=-d}^{d+1} L_\ell^d\left(\frac{p}{q}\right) \sin\left(\left(\ell - \frac{p}{q}\right)\omega\right) = \\
&= \sum_{\ell=-d}^{d+1} L_\ell^d(0)\sin(\ell\omega) + \sum_{p=1}^{q-1}\sum_{\ell=1}^{d+1}\left[L_\ell^d\left(\frac{p}{q}\right)\sin\left(\left(\ell - \frac{p}{q}\right)\omega\right) + L_{-\ell+1}^d\left(\frac{p}{q}\right)\sin\left(\left(-\ell + 1 - \frac{p}{q}\right)\omega\right)\right] \\
&= \sum_{\ell=-d}^{d+1} \delta_{\ell,0}\sin(\ell\omega) + \sum_{p=1}^{q-1}\sum_{\ell=1}^{d+1}\left[L_\ell^d\left(\frac{p}{q}\right)\sin\left(\left(\ell - \frac{p}{q}\right)\omega\right) + L_\ell^d\left(1 - \frac{p}{q}\right)\sin\left(\left(-\ell + 1 - \frac{p}{q}\right)\omega\right)\right] \\
&= \sin(0) + \sum_{p=1}^{q-1}\sum_{\ell=1}^{d+1} L_\ell^d\left(\frac{p}{q}\right)\left[\sin\left(\left(\ell - \frac{p}{q}\right)\omega\right) + \sin\left(\left(-\ell + \frac{p}{q}\right)\omega\right)\right] = 0.
\end{aligned}$$

Therefore we have proven that $t_p \in \mathbb{R}$ for $p = 0, \dots, q-1$.

3.2.4 Proving that the DFT is non-negative

Now, it remains to see if we can prove that

$$S_{q,d}(\omega) \ge 0 \quad \text{for all } 0 \le \omega \le 2\pi q. \tag{3.12}$$

If this inequality is true for $d = d_\theta$, from (3.11) we obtain that $t_p \ge 0$ for $p = 0, \dots, q-1$, and therefore (3.8) holds. From this follows the stability of our numerical scheme for any rational λ.
For $q = 1$, we have

$$S_{1,d}(\omega) = \sum_{\ell=-d}^{d+1} L_\ell^d(0)\cos(\ell\omega) = 1 \ge 0.$$

For $q > 1$, the situation is much more complicated and it requires a careful study of the function $S_{q,d}$ in the interval $[0, 2\pi q]$. We first notice that we can explicitly compute the values $S_{q,d}(2\pi n)$ for $n \in \mathbb{N}$, because (3.10) simplifies to

$$S_{q,d}(2\pi n) = \sum_{p=0}^{q-1}\left[\sum_{\ell=-d}^{d+1} L_\ell^d\left(\frac{p}{q}\right)\right]\exp\left(-i\frac{p}{q}2\pi n\right) = \sum_{p=0}^{q-1}\exp\left(-i\frac{p}{q}2\pi n\right),$$

where we have used the identity $(\exp(i2\pi))^{n\ell} = 1$ together with the partition of unity of the Lagrange interpolant, namely $\sum_{\ell=-d}^{d+1} L_\ell^d(\alpha) = 1$, for all $\alpha \in \mathbb{R}$. From the last equation we obtain that

$$S_{q,d}(0) = q, \qquad S_{q,d}(2\pi n) = 0 \quad \text{for } n = 1, \dots, q-1, \qquad S_{q,d}(2\pi q) = q, \tag{3.13}$$

and therefore $S_{q,d}$ has at least $q-1$ zeros in $(0, 2\pi q)$ and is strictly positive at the boundaries. In the following discussion we will show that there are no other zeros in the same interval, and that the function is convex at all zeros (and therefore positive in some open interval around each zero). By continuity, this proves that $S_{q,d}(\omega) \geq 0$ for all $0 \leq \omega \leq 2\pi q$. Our derivation is somewhat involved because most information will be extracted from $S'_{q,d}$, as in [11].

The derivative of (3.10) reads:

$$S'_{q,d}(\omega) = i \sum_{p=0}^{q-1} \sum_{\ell=-d}^{d+1} L_\ell^d\left(\frac{p}{q}\right)\left(\ell - \frac{p}{q}\right) \exp\left(i\left(\ell - \frac{p}{q}\right)\omega\right).$$

Now, for $\ell = -d, \dots, d+1$, we have $L_\ell^d(x) = \frac{\prod_{k=-d,\ k\neq\ell}^{d+1} x-k}{\prod_{k=-d,\ k\neq\ell}^{d+1} \ell-k}$

$$L_\ell^d\left(\frac{p}{q}\right)\left(\frac{p}{q} - \ell\right) = \frac{1}{\prod_{k=-d}^{\ell-1}(\ell-k)\prod_{k=\ell+1}^{d+1}(\ell-k)} \prod_{k=-d}^{d+1}\left(\frac{p}{q} - k\right) = \frac{(-1)^{d+1-\ell}}{(d+\ell)!(d+1-\ell)!} \prod_{k=-d}^{d+1}\left(\frac{p}{q} - k\right),$$

so that

$$S'_{q,d}(\omega) = -i\left[\sum_{\ell=-d}^{d+1} \frac{(-1)^{d+1-\ell}}{(d+\ell)!(d+1-\ell)!}\exp(i\ell\omega)\right] \sum_{p=0}^{q-1} \exp\left(-i\frac{p}{q}\omega\right) w_d\left(\frac{p}{q}\right),$$

with

$$w_d(x) = \prod_{k=-d}^{d+1}(x-k), \ x \in \mathbb{R}. \tag{3.14}$$

We point out that $w_d(x)$ is symmetric about the point $x = 1/2$, as

$$w_d(x) = \prod_{k=0}^{d}(x-k+1)(x+k) = \prod_{k=0}^{d}\left(\left(x - \frac{1}{2}\right)^2 - \left(k + \frac{1}{2}\right)^2\right).$$

Now, if we multiply by $\exp(id\omega)$ the term within square brackets in the expression for $S'_{q,d}(\omega)$, we can identify the sum therein as the polynomial expansion of a binomial power,

$$\begin{aligned}\sum_{\ell=-d}^{d+1} \frac{(-1)^{d+1-\ell}}{(d+\ell)!(d+1-\ell)!}\exp(i(\ell+d)\omega) &= \sum_{\ell=0}^{2d+1}\frac{(-1)^{2d+1-\ell}}{\ell!(2d+1-\ell)!}\exp(i\ell\omega) = \\ &= \frac{(-1)^{2d+1}}{(2d+1)!}\sum_{\ell=0}^{2d+1}\binom{2d+1}{\ell}(-\exp(i\omega))^\ell = \frac{(\exp(i\omega)-1)^{2d+1}}{(2d+1)!},\end{aligned}$$

and obtain therefore

$$S'_{q,d}(\omega) = \frac{-i\exp(-i\omega d)(\exp(i\omega)-1)^{2d+1}}{(2d+1)!}\sum_{p=0}^{q-1}\exp\left(-i\frac{p}{q}\omega\right) w_d\left(\frac{p}{q}\right).$$

The coefficient that appears in front of the summation can be reformulated as

$$\begin{aligned} -i\exp(-i\omega d)(\exp(i\omega)-1)^{2d+1} &= -i\exp\left(i\frac{\omega}{2}\right)\left(\exp\left(-i\frac{\omega}{2}\right)\right)^{2d+1}(\exp(i\omega)-1)^{2d+1} \\ &= -i\exp\left(i\frac{\omega}{2}\right)\left(\exp\left(i\frac{\omega}{2}\right) - \exp\left(-i\frac{\omega}{2}\right)\right)^{2d+1} = (-1)^d 2^{2d+1}\sin^{2d+1}\left(\frac{\omega}{2}\right)\exp\left(i\frac{\omega}{2}\right),\end{aligned}$$

which yields an expression for $S'_{q,d}(\omega)$ where all terms are real, apart from the complex exponential coefficients in the summation:

$$S'_{q,d}(\omega) = (-1)^d 2^{2d+1} \sin^{2d+1}\left(\frac{\omega}{2}\right) \sum_{p=1}^{q-1} \exp\left(i\left(\frac{1}{2} - \frac{p}{q}\right)\omega\right) w_d\left(\frac{p}{q}\right).$$

Because $S_{q,d}$ is real valued, so must be its derivatives. In fact, the imaginary part of the summation above is zero thanks to the fact that $w_d(0) = 0$ by definition, and $w_d(x) = w_d(1-x)$ by symmetry. Finally, we obtain

$$S'_{q,d}(\omega) = (-1)^d \frac{2^{2d+1}}{(2d+1)!} \sin^{2d+1}\left(\frac{\omega}{2}\right) \sigma_{q,d}(\omega), \tag{3.15}$$

with

$$\sigma_{q,d}(\omega) = \sum_{p=1}^{q-1} \cos\left(\left(\frac{1}{2} - \frac{p}{q}\right)\omega\right) w_d\left(\frac{p}{q}\right). \tag{3.16}$$

We will now study separately the two factors $\sin^{2d+1}(\omega/2)$ and $\sigma_{q,d}(\omega)$ that appear in (3.15). The former has zeros at $2n\pi$ for $n \in \mathbb{N}$, with $2d$ derivatives also zero at the same location. In fact, if we look at the asymptotic behavior near any of the points $\omega = 2n\pi$, we have

$$\sin^{2d+1}\left(\frac{\omega}{2}\right) = \left[(-1)^n \sin\left(\frac{\omega - 2n\pi}{2}\right)\right]^{2d+1} \underset{\omega=2n\pi}{\sim} (-1)^n \frac{(\omega - 2n\pi)^{2d+1}}{2^{2d+1}}, \tag{3.17}$$

where we used the angle sum identity for sines, together with $\cos(n\pi) = (-1)^n$ and $\sin(n\pi) = 0$. A comparison with the Taylor expansions around the same locations yields

$$\left[\sin^{2d+1}\left(\frac{\omega}{2}\right)\right]^{(j)}_{\omega=2n\pi} = 0 \qquad \text{for } j = 0, \dots, 2d,$$
$$\left[\sin^{2d+1}\left(\frac{\omega}{2}\right)\right]^{(2d+1)}_{\omega=2n\pi} = (-1)^n \frac{(2d+1)!}{2^{2d+1}}.$$

We now turn to study the term $\sigma_{q,d}(\omega)$ at the same locations. If we multiply (3.16) by $(-1)^{n+d}$ and then use the angle sum identity for cosines, again with $\cos(n\pi) = (-1)^n$ and $\sin(n\pi) = 0$, we obtain

$$(-1)^{n+d}\sigma_{q,d}(2n\pi) = (-1)^{n+d} \sum_{p=1}^{q-1} \cos\left(\left(\frac{1}{2} - \frac{p}{q}\right)2n\pi\right) w_d\left(\frac{p}{q}\right) = (-1)^d \sum_{p=1}^{q-1} \cos\left(2n\pi\frac{p}{q}\right) w_d\left(\frac{p}{q}\right).$$

We then will use the following discrete form of a lemma relating real convex functions to positive Fourier transforms (see [12, 13] and [14] for historical notes).

Lemma 3.1. *Let $q \geq 2$ be an integer and f_j be a sequence of $q+1$ real numbers with $j = 0, \dots, q$, such that $f_0 = f_q = 0$ and*

$$f_{j+1} - 2f_j + f_{j-1} \geq 0, \quad j = 1, \dots, q-1.$$

Then, we have

$$\sum_{p=1}^{q-1} \cos\left(2n\pi\frac{p}{q}\right) f_p \geq 0, \quad n = 1, \dots, q-1. \tag{3.18}$$

Moreover, if we have additionally $f_1 > f_2/2$, then (3.18) is strict for all $n = 1, \dots, q-1$.

Proof. Since $f_q = f_0 = 0$, and $0 < n < q$, we have

$$\sum_{p=1}^{q-1} \cos\left(2n\pi\frac{p}{q}\right) f_p = \sum_{p=0}^{q-1} \frac{\sin(2n\pi\frac{p+1/2}{q})}{2\sin(\frac{n\pi}{q})} f_p - \sum_{p=1}^{q} \frac{\sin(2n\pi\frac{p-1/2}{q})}{2\sin(\frac{n\pi}{q})} f_p = \sum_{p=0}^{q-1} \frac{\sin(2n\pi\frac{p+1/2}{q})}{2\sin(\frac{n\pi}{q})} (f_p - f_{p+1}).$$

Now, as $n \in \mathbb{N}$, we have

$$\sum_{p=1}^{q-1} \cos\left(2n\pi\frac{p}{q}\right) f_p = \sum_{p=0}^{q-2} \frac{\cos(2n\pi\frac{p+1}{q}) - 1}{4\sin^2(\frac{n\pi}{q})} (f_{p+1} - f_p) + \sum_{p=1}^{q-1} \frac{\cos(2n\pi\frac{p}{q}) - 1}{4\sin^2(\frac{n\pi}{q})} (f_p - f_{p+1}).$$

This leads to

$$\sum_{p=1}^{q-1} \cos\left(2n\pi\frac{p}{q}\right) f_p = \sum_{p=1}^{q-1} \frac{\cos(2n\pi\frac{p}{q}) - 1}{4\sin^2(\frac{n\pi}{q})}(2f_p - f_{p+1} - f_{p-1}) \geq 0.$$

Finally, if we also have $f_1 > \frac{f_2}{2}$, we get

$$\sum_{p=1}^{q-1} \cos\left(2n\pi\frac{p}{q}\right) f_p \geq \frac{\cos(2n\pi\frac{1}{q}) - 1}{4\sin^2(\frac{n\pi}{q})}(2f_1 - f_2) > 0.$$

□

Proposition 3.1. *Let $d \in \mathbb{N}$ and $w_d(x) = \prod_{k=-d}^{d+1}(x-k)$. Then $(-1)^d w_d$ is strictly convex on $[0,1]$.*

Proof. We know that w_d is a polynomial of degree $2d+2$ whose roots are k, $k = -d, \dots d+1$. By Rolle's theorem and since w_d' is a polynomial of degree $2d+1$, w_d' vanishes exactly one time in each interval $(k, k+1)$. We also have $w_d(1/2 + x) = w_d(1/2 - x)$ by symmetry, so the unique zero of w_d' in $(0,1)$ is $1/2$. By Rolle's theorem and since w_d'' is a polynomial of degree $2d$, looking at the variation table, we can see that $(-1)^d w_d'' < 0$ on $(t_{-1}, s_{-1}) \cup (s_1, t_1)$ and $(-1)^d w_d'' > 0$ on (s_{-1}, s_1), with t_{-1} the unique zero of w_d' in $(-1,0)$ and t_1 the unique zero of w_d' in $(1,2)$, and we have $s_{-1} \in (t_{-1}, 1/2)$, $s_1 \in (1/2, t_1)$. From the expression $w_d(x) = x(x-d-1)\prod_{k=1}^{d}(x^2 - k^2)$ we get $w_d''(0) = 2\prod_{k=1}^{d}(-k^2)$ (with the convention $\prod_{k=1}^{0} = 1$), and thus $(-1)^d w_d''(0) > 0$, which implies that $s_{-1} < 0$. By symmetry we have $(-1)^d w_d''(1) > 0$, and thus $s_1 > 1$. Finally, we have $(-1)^d w_d'' > 0$ on $[0,1]$. □

From Lemma 3.1, Proposition 3.1 and as $w_d(0) = w_d(1) = 0$, we deduce that

$$(-1)^{n+d}\sigma_{q,d}(2n\pi) > 0, \quad n = 1, \dots, q-1, \quad q > 2, \quad d \in \mathbb{N}. \tag{3.19}$$

We now turn to study the asymptotic behavior of $S_{q,d}(\omega)$ as $\omega \to 2n\pi$ for $n = 1, \dots, q-1$. Thanks to (3.19), we can substitute (3.17) into (3.15) to obtain an asymptotic expression for $S_{q,d}'$, which we integrate once using (3.13). Finally, we obtain the asymptotic equivalence

$$S_{q,d}(\omega) \underset{\omega=2n\pi}{\sim} \frac{(-1)^{n+d}\sigma_{q,d}(2n\pi)}{(2d+2)!}(\omega - 2n\pi)^{2d+2}, \quad n = 1, \dots, q-1, \quad q = 2, 3, \dots, \quad d \in \mathbb{N}, \tag{3.20}$$

which we compare with the Taylor expansion of $S_{q,d}$ about $\omega = 2n\pi$ for $n = 1, \dots, q-1$ to find

$$\begin{aligned} S_{q,d}^{(j)}(2n\pi) &= 0 \qquad \text{for } j = 0, \dots, 2d+1, \\ S_{q,d}^{(2d+2)}(2n\pi) &= (-1)^{n+d}\sigma_{q,d}(2n\pi) > 0. \end{aligned} \tag{3.21}$$

Because the first non-zero derivative is of even order and positive, we conclude that $S_{q,d}$ is convex at each of its zeros $2n\pi$ for $n = 1, \dots, q-1$. If we can prove that $S_{q,d}$ has no other zeros in $[0, 2q\pi]$, it follows that $S_{q,d} \geq 0$ in the whole interval.
We have now the following lemma.

Lemma 3.2. *If $q \geq 2$, then $\sigma_{q,d}(\omega)$ as defined in (3.16) is a polynomial in $\cos(\frac{\omega}{2q})$ of degree $\leq q-2$.*

Proof. The result follows from the formula

$$\begin{aligned} \cos\left(\frac{\omega}{2q}(q-2p)\right) &= \mathrm{Re}\left\{\left[\exp\left(i\frac{\omega}{2q}\right)\right]^{q-2p}\right\} = \mathrm{Re}\left\{\left[i\sin\left(\frac{\omega}{2q}\right) + \cos\left(\frac{\omega}{2q}\right)\right]^{q-2p}\right\} = \\ &= \mathrm{Re}\left\{\sum_{k=0}^{q-2p}\binom{q-2p}{k} i^k \sin^k\left(\frac{\omega}{2q}\right)\cos^{q-2p-k}\left(\frac{\omega}{2q}\right)\right\} = \\ &= \sum_{\substack{k=0 \\ k \text{ even}}}^{q-2p}\binom{q-2p}{k}(-1)^{k/2}\left(1 - \cos^2\left(\frac{\omega}{2q}\right)\right)^{k/2}\cos^{q-2p-k}\left(\frac{\omega}{2q}\right). \end{aligned}$$

□

Because of this lemma and the fact that $\cos(\frac{\omega}{2q})$ is monotonically decreasing in $(0, 2q\pi)$, we have that $\sigma_{q,d}(\omega)$ has at most $q-2$ zeros in the same interval.
Since $S_{q,d}(2n\pi) = 0$ for $n = 1, \dots, q-1$, according to Rolle's theorem there exist $u_n \in (2n\pi, 2(n+1)\pi)$ such that $S'_{q,d}(u_n) = 0$, for $n = 1, \dots, q-2$. We then also have $\sigma_{q,d}(u_n) = 0$, because $\sin(\omega/2)$ has no zeros inside those intervals. So, we have found $q-2$ distinct roots for the polynomial of Lemma 3.2, which is non zero and of degree $\leq q-2$. We deduce that there is exactly one zero of $\sigma_{q,d}$ in the interval $(2n\pi, 2(n+1)\pi)$ for $n = 1, \dots, q-2$ and no zero of $\sigma_{q,d}$ in the interval $(0, 2\pi)$ and $(2(q-1)\pi, 2q\pi)$, and this is the same for $S'_{q,d}$. If we combine this information with the convexity of $S_{q,d}$ at $\omega = 2n\pi$, we conclude that:

- $S_{q,d}$ decreases monotonically from $S_{q,d}(0) = q$ to $S_{q,d}(2\pi) = 0$, and therefore $S_{q,d} \geq 0$ in $[0, 2\pi]$;
- In each interval $[2n\pi, 2(n+1)\pi]$ for $n = 1, \dots, q-2$, $S_{q,d}$ increases monotonically from $S_{q,d}(2n\pi) = 0$ to $S_{q,d}(u_n) > 0$ and then decreases monotonically to $S_{q,d}(2(n+1)\pi) = 0$, therefore $S_{q,d} \geq 0$ in $[2\pi, 2(q-1)\pi]$;
- $S_{q,d}$ increases monotonically from $S_{q,d}(2(q-1)\pi) = 0$ to $S_{q,d}(2q\pi) = q$, and therefore $S_{q,d} \geq 0$ in $[2(q-1)\pi, 2q\pi]$.

This proves that $S_{q,d}(\omega) \geq 0$ for all $0 \leq \omega \leq 2q\pi$, and therefore $t_p \in \mathbb{R}^+$ for $p = 1, \dots, q-1$. Accordingly, we obtain the identity (3.4) and hence the stability of our numerical scheme for any $\lambda \in \mathbb{Q}$.

3.2.5 Statement of unconditional stability

The stability of our numerical scheme for a general $\lambda \in \mathbb{R}$ follows from the stability proof already given, thanks to the density of the rational numbers in $\mathbb{R}$. Let $A : \mathbb{R} \to \mathbb{R}$ be the modulus of our Fourier symbol as a function of λ, for a given choice of r_φ, α_φ, ω_θ and ω_φ:

$$A(\lambda) = |\sigma_{\lambda, r_\varphi, \alpha_\varphi}(\omega_\theta, \omega_\varphi)|.$$

Because of the floor function that appears in the Fourier symbol (3.2), A presents discontinuities of the first kind at a set of isolated rational locations,

$$\left\{ \frac{n}{r_\varphi + k} \,\middle|\, n \in \mathbb{N};\ k \in \{-d_b, \dots, d_b + 1\} \setminus \{-r_\varphi\} \right\} \subset \mathbb{Q},$$

so that the minimum distance between two discontinuities is $1/(1+d_b+|r_\varphi|)$. Everywhere else A is continuous, and specifically so at all irrational values of λ. We now have two cases:

1. If $\lambda \in \mathbb{Q}$, we have already proven that $A(\lambda) \leq 1$;
2. If $\lambda \in \mathbb{R} \setminus \mathbb{Q}$, the function A is continuous in some open interval $(\lambda - \delta, \lambda + \delta)$ with $\delta > 0$. We now suppose that $A(\lambda) > 1$ and show that this leads to a contradiction. Because of continuity, there exists an open interval $(\lambda - \varepsilon, \lambda + \varepsilon)$ with $0 < \varepsilon \leq \delta$ where $A > 1$. Because of the density of $\mathbb{Q}$ in $\mathbb{R}$, there exists $\lambda^* \in \mathbb{Q} \cap (\lambda - \varepsilon, \lambda + \varepsilon)$ such that $A(\lambda^*) > 1$, but this contradicts case 1. Therefore we obtain again that $A(\lambda) \leq 1$.

With this we have proven that $|\sigma_{\lambda, r_\varphi, \alpha_\varphi}(\omega_\theta, \omega_\varphi)| \leq 1$ for all $\lambda \in \mathbb{R}$, $r_\varphi \in \mathbb{Z}$, $\alpha_\varphi \in [0, 1)$, and $(\omega_\theta, \omega_\varphi) \in [0, 2\pi]^2$. The numerical scheme so presented is unconditionally stable.

3.3 Truncation error and convergence

3.3.1 Approximation error for 1D centered Lagrange interpolation

We now focus on the truncation error due to 1D centered Lagrange interpolation on a uniform grid, of odd order $2d+1$ with $d \in \mathbb{N}$. To this end we repeat here part of the analysis done in [15], with a slight refinement on the final error estimates. The results of this section will then be used for assessing the error of our 2D field-aligned semi-Lagrangian scheme, and to compare it to the classical (non field-aligned) scheme.

Consider a function $g : \mathbb{R} \to \mathbb{C}$ smooth enough, which we sample on a uniform grid $z_j = j\Delta z$ with $j \in \mathbb{Z}$ and $\Delta z \in \mathbb{R}_+^*$. Without loss of generality, we focus on the location $z = \alpha\Delta z$ with $0 \leq \alpha \leq 1$. If we write the interpolation error at $\alpha\Delta z$ in terms of divided differences, which we reformulate in Peano form, we obtain

$$g(\alpha\Delta z) - \sum_{k=-d}^{d+1} L_k^d(\alpha) g(k\Delta z) = \Delta z^{2d+2} \frac{\prod_{\ell=-d}^{d+1}(\alpha-\ell)}{(2d+1)!} \int_{-d\Delta z}^{(d+1)\Delta z} Q_{\alpha,\Delta z}^{2d+2}(z)\partial_z^{2d+2} g(z)dz, \tag{3.22}$$

where $Q_{\alpha,\Delta z}^{2d+2}$ is the B-spline function over the points $\alpha\Delta z$ and $\ell\Delta z$, for $\ell = -d, \dots, d+1$, satisfying

$$\int_{-d\Delta z}^{(d+1)\Delta z} Q_{\alpha,\Delta z}^{2d+2}(z)dz = \frac{1}{2d+2}.$$

If we introduce the linear change of coordinates $\eta(z) = (z + d\Delta z)/((2d+1)\Delta z))$, we can write $Q_{\alpha,\Delta z}^{2d+2}(z)dz = B_{2d+2,\alpha}(\eta)d\eta$, where $B_{2d+2,\alpha}$ is the B-spline function over the $2d+3$ points

$$0 < \frac{1}{2d+1} < \cdots < \frac{d}{2d+1} \leq \frac{d+\alpha}{2d+1} \leq \frac{d+1}{2d+1} < \cdots < \frac{2d}{2d+1} < 1,$$

with the (same) normalization $\int_0^1 B_{2d+2,\alpha}(\eta)d\eta = (2d+2)^{-1}$. We note that $B_{2d+2,\alpha}(\eta) \geq 0$ for $0 \leq \eta \leq 1$. The identity (3.22) then rewrites

$$g(\alpha\Delta z) - \sum_{k=-d}^{d+1} L_k^d(\alpha) g(k\Delta z) = \Delta z^{2d+2} \frac{\prod_{\ell=-d}^{d+1}(\alpha-\ell)}{(2d+1)!} \int_0^1 B_{2d+2,\alpha}(\eta)\partial_z^{2d+2} g((-d+(2d+1)\eta)\Delta z)d\eta.$$

We now suppose that the function g is harmonic, i.e. $g(z) = \exp(i(\omega z + \phi))$ with $\omega \in \mathbb{R}$ and $\phi \in [0, 2\pi]$, and we proceed with estimating the maximum magnitude of the interpolation error over all possible values of α and ϕ. Since $\partial_z^{2d+2} g(z) = (i\omega)^{2d+2} g(z)$, the absolute value of the error is

$$\left| g(\alpha\Delta z) - \sum_{k=-d}^{d+1} L_k^d(\alpha) g(k\Delta z) \right| = (\omega\Delta z)^{2d+2} \frac{\prod_{\ell=-d}^{d+1}|\alpha-\ell|}{(2d+1)!} \left| \int_0^1 B_{2d+2,\alpha}(\eta)\exp(i(\omega z(\eta)+\phi))d\eta \right|.$$

The maximum magnitude of the integral factor is difficult to compute, nevertheless we can obtain an upper bound by using the triangular inequality for integrals, as

$$\left| \int_0^1 B_{2d+2,\alpha}(\eta)\exp(i(\omega z(\eta)+\phi))d\eta \right| \leq \int_0^1 B_{2d+2,\alpha}(\eta)\Big|\exp(i(\omega z(\eta)+\phi))\Big|d\eta = \int_0^1 B_{2d+2,\alpha}(\eta)d\eta = \frac{1}{2d+2},$$

with the understanding that such an estimate is sharp in the limit as $\Delta z \to 0$, which does not depend on α or ϕ:

$$\lim_{\Delta z \to 0} \left| \int_0^1 B_{2d+2,\alpha}(\eta)\exp(i(\omega z(\eta)+\phi))d\eta \right| = \left| \exp(i\phi) \int_0^1 B_{2d+2,\alpha}(\eta)d\eta \right| = \frac{1}{2d+2}.$$

The product in front of the integral can be written as

$$\prod_{\ell=-d}^{d+1}|\alpha-\ell| = \prod_{\ell=-d}^{0}(\alpha-\ell)\prod_{\ell=1}^{d+1}(\ell-\alpha) = \prod_{\ell=1}^{d+1}(\ell-(1-\alpha))(\ell-\alpha) = \alpha(1-\alpha)\prod_{\ell=2}^{d+1}\left[\left(\ell-\frac{1}{2}\right)^2 - \left(\alpha-\frac{1}{2}\right)^2\right],$$

where we have extracted the $\ell = 1$ factor because it goes to zero in the limits as $\alpha \to 0$ and as $\alpha \to 1$, and we would like to retain such an asymptotic behavior in our estimates. All factors in the final multiplication are strictly positive and achieve their maximum value for $\alpha = 1/2$, and the missing $\ell = 1$ term has value $(1 - 1/2)^2 = 1/4$. Therefore we can write the upper bound

$$\prod_{\ell=-d}^{d+1}|\alpha-\ell| \;\leq\; 4\alpha(1-\alpha)\prod_{\ell=1}^{d+1}\left(\ell-\frac{1}{2}\right)^2,$$

which reduces to an equality for $\alpha \in \{0, 1/2, 1\}$. We point out that, although this expression converges to the correct limit for $\alpha \to 0$, it overestimates the linear rate of convergence. In other words, this upper bound is not sharp. Such an expression can be explicitly evaluated as

$$\left[\prod_{\ell=1}^{d+1}\left(\ell - \frac{1}{2}\right)\right]^2 = \left[\frac{1}{2^{d+1}}\prod_{\ell=1}^{d+1}(2\ell+1)\right]^2 = \left[\frac{1}{2^{d+1}}\frac{\prod_{k=1}^{2d+2} k}{\prod_{\ell=1}^{d+1} 2\ell}\right]^2 = \left[\frac{(2d+2)!}{2^{2d+2}(d+1)!}\right]^2$$
$$= \frac{(2d+2)!}{(2^{2d+2})^2}\frac{(2d+2)!}{(d+1)!(d+1)!} = \frac{(2d+2)!}{4^{2d+2}}\binom{2d+2}{d+1},$$

where the central binomial coefficient can be approximated very accurately with the following upper bound, which is sharp in the limit as $d \to \infty$ and can be obtained in various ways (e.g. from Stirling's formula):

$$\binom{2d+2}{d+1} < \frac{2^{2d+2}}{\sqrt{\pi(d+1)}}.$$

Putting everything together we obtain the following upper bound, which is sharp in the limit as $\Delta z \to 0$ and $d \to \infty$:

$$\left| g(\alpha\Delta z) - \sum_{k=-d}^{d+1} L_k^d(\alpha) g(k\Delta z)\right| \le \left(\frac{\omega\Delta z}{2}\right)^{2d+2} \frac{4\alpha(1-\alpha)}{\sqrt{\pi(d+1)}}.$$

This formula will be used directly to construct an error bound for the 2D classical semi-Lagrangian scheme, which in turn will be the base of comparison for our 2D field-aligned method. Based on this estimate, we observe that:

1. The approximation error decreases with order $2d+2$ in the discretization parameter Δz, as expected;

2. In practical applications one seeks the largest value Δz that yields an error smaller than a certain threshold $\varepsilon \ll 1$; regardless of the order of the polynomial, this always implies that $\omega\Delta z < 2$, that is, at least π grid points must fit within the characteristic wavelength of $g(z)$;

3. When the interpolation procedure is part of a semi-Lagrangian scheme, the time step size Δt must be taken into consideration, because it directly effects the value of α; particularly important is the fact that, in the limit of $\Delta t/\Delta z \to 0$, we have $\alpha \to 0$ and therefore the error also goes to zero. For an extended discussion over the role of Δt in the convergence of semi-Lagrangian schemes we refer to [15].

3.3.2 Error estimate for field-aligned semi-Lagrangian scheme

We let $f(t_n)$ be the exact solution, and $f^{(n)}$ the numerical solution, at time t_n. We introduce some notation (see [15]): $\Pi : f \to (f_{i,j})$ is the discretization (sampling) operator on a uniform 2D grid, and $\mathcal{T}$ (resp. $\widetilde{\mathcal{T}}$) is the numerical (resp. exact) transport operator in direction $\mathbf{b}$, over one time step Δt. The (global) error then reads

$$e^{(n+1)} = \Pi f(t_{n+1}) - f^{(n+1)} = \Pi\widetilde{\mathcal{T}} f(t_n) - \mathcal{T}\left(\Pi f(t_n) - e^{(n)}\right) = \left(\Pi\widetilde{\mathcal{T}} - \mathcal{T}\Pi\right) f(t_n) + \mathcal{T} e^{(n)},$$

where we identify in $(\Pi\widetilde{\mathcal{T}} - \mathcal{T}\Pi) f(t_n)$ the "truncation error" introduced by the numerical scheme between time t_n and $t_n + \Delta t$. Since the scheme is proven to be unconditionally stable, the error cannot grow in the L^2-norm when transported by the numerical scheme, i.e. $\|\mathcal{T} e^{(n)}\|_2 \le \|e^{(n)}\|_2$. The triangular inequality then yields

$$\left\| e^{(n+1)} \right\|_2 \le \left\| (\Pi\widetilde{\mathcal{T}} - \mathcal{T}\Pi) f(t_n)\right\|_2 + \left\| e^{(n)}\right\|_2,$$

and if we proceed recursively up to time t_0, where $e^{(0)} = 0$ by construction, we obtain the upper bound

$$\left\| e^{(n)} \right\|_2 \ \le \ \sum_{k=0}^{n-1} \left\| (\Pi\widetilde{\mathcal{T}} - \mathcal{T}\Pi) f(t_k)\right\|_2, \tag{3.23}$$

that is, the norm of the (global) error at time t_n cannot be larger than the sum of the norms of the previous n truncation errors. Here we made use of the discrete L^2-norm, defined as

$$\|f\|_2 = \left[\frac{1}{N_\theta N_\varphi} \sum_{i_\theta=1}^{N_\theta} \sum_{i_\varphi=1}^{N_\varphi} \left(f_{i_\theta,i_\varphi} \right)^2 \right]^{\frac{1}{2}} . \tag{3.24}$$

The upper bound (3.23) provides us with an error estimate at time t_n, if an upper bound for the truncation error is available. Similarly to the analysis in the previous section, we now compute the truncation error for harmonic initial condition $f_0(\theta,\varphi) = \exp(i(n_\varphi \varphi + m_\theta \theta))$, for which the exact solution is simply

$$f(t,\theta,\varphi) = \exp(i(n_\varphi(\varphi - b_\varphi t) + m_\theta(\theta - b_\theta t))).$$

Under this assumption, the local truncation error for our field-aligned semi-Lagrangian scheme can be decomposed into two parts, as

$$\left(\Pi \widetilde{\mathcal{T}} - \mathcal{T} \Pi \right) f(t_n)_{i_\theta,i_\varphi} = A_1 + A_2,$$

where A_1 is the approximation error introduced by Lagrange interpolation in direction $\mathbf{b}$,

$$A_1 = f(t_n, \theta_{i_\theta} - b_\theta \Delta t, \varphi_{i_\varphi} - b_\varphi \Delta t) - \sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi) f(t_n, \theta_{i_\theta} - b_\theta \Delta t_k, \varphi_{i_\varphi + r_\varphi + k}),$$

and A_2 is the approximation error of Lagrange interpolation along θ, which is then interpolated along $\mathbf{b}$:

$$A_2 = \sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi) \left(f(t_n, \theta_{i_\theta} - b_\theta \Delta t_k, \varphi_{i_\varphi+r_\varphi+k}) - \sum_{\ell=-d_\theta}^{d_\theta+1} L_\ell^{d_\theta}(\alpha_{\theta,k}) f(t_n, \theta_{i_\theta + r_{\theta,k} + \ell}, \varphi_{i_\varphi + r_\varphi + k}) \right).$$

Here we recall the following definitions:

$$\begin{aligned} -b_\varphi \Delta t \;\; &= \Delta\varphi \left(r_\varphi + \alpha_\varphi \right) && \text{with } r_\varphi \in \mathbb{Z} \text{ and } \alpha_\varphi \in \mathbb{R}_{[0,1)}, \\ -b_\varphi \Delta t_k &= \Delta\varphi \left(r_\varphi + k \right) && \text{with } \Delta t_k \in \mathbb{R} \text{ and } k = -d_b, \dots, d_b+1, \\ -b_\theta \Delta t_k &= \Delta\theta \left(r_{\theta,k} + \alpha_{\theta,k} \right) && \text{with } r_{\theta,k} \in \mathbb{Z} \text{ and } \alpha_{\theta,k} \in \mathbb{R}_{[0,1)}. \end{aligned}$$

Furthermore, in the following calculation we will write $\theta_{i_\theta} = 2\pi i_\theta / N_\theta$ and $\varphi_{i_\varphi} = 2\pi i_\varphi / N_\varphi$. We first compute A_2: similarly to the previous section, we formulate the interpolation error in integral form and obtain

$$\begin{aligned} A_2 = \left(i \frac{m_\theta}{N_\theta} \right)^{2d_\theta+2} \frac{(2\pi)^{2d_\theta+2}}{(2d_\theta+1)!} f(t_n, \theta_{i_\theta}, \varphi_{i_\varphi+r_\varphi}) \sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi) \\ e^{2\pi i \left(\frac{m_\theta}{N_\theta} r_{\theta,k} + k \frac{n_\varphi}{N_\varphi} \right)} \prod_{\ell=-d_\theta}^{d_\theta+1} (\alpha_{\theta,k} - \ell) \int_0^1 B_{2d_\theta+2,\alpha_{\theta,k}}(\sigma) e^{2\pi i (-d_\theta + (2d_\theta+1)\sigma) \frac{m_\theta}{N_\theta}} \, d\sigma. \end{aligned}$$

We then have

$$|A_2| \le \left(\frac{|m_\theta|}{N_\theta} \right)^{2d_\theta+2} \frac{(2\pi)^{2d_\theta+2}}{(2d_\theta+1)!} \int_0^1 \left| \sum_{k=-d_b}^{d_b+1} L_k^{d_b}(\alpha_\varphi) e^{2\pi i \left(\frac{m_\theta}{N_\theta} r_{\theta,k} + k \frac{n_\varphi}{N_\varphi} \right)} \prod_{\ell=-d_\theta}^{d_\theta+1} (\alpha_{\theta,k} - \ell) \, B_{2d_\theta+2,\alpha_{\theta,k}}(\sigma) \right| d\sigma.$$

We then get

$$|A_2| \le \left(\frac{|m_\theta|}{N_\theta} \right)^{2d_\theta+2} \frac{(2\pi)^{2d_\theta+2}}{(2d_\theta+2)!} \sum_{k=-d_b}^{d_b+1} |L_k^{d_b}(\alpha_\varphi)| \prod_{\ell=-d_\theta}^{d_\theta+1} |\alpha_{\theta,k} - \ell| .$$

As in the analysis for 1D interpolation, we can provide an error bound for the product term and write therefore

$$|A_2| \le \left(\frac{\pi |m_\theta|}{N_\theta} \right)^{2d_\theta+2} \frac{4}{\sqrt{\pi(d_\theta+1)}} \sum_{k=-d_b}^{d_b+1} |L_k^{d_b}(\alpha_\varphi)| \alpha_{\theta,k}(1-\alpha_{\theta,k}). \tag{3.25}$$

For A_1 we have, writing $n_b = m_\theta \frac{b_\theta}{b_\varphi} + n_\varphi$

$$A_1 = \left(i \frac{n_b}{N_\varphi} \right)^{2d_b+2} \frac{(2\pi)^{2d_b+2}}{(2d_b+1)!} f(t_n, \theta_{i_\theta} + \frac{b_\theta}{b_\varphi} r_\varphi, \varphi_{i_\varphi + r_\varphi}) \prod_{\ell=-d_b}^{d_b+1} (\alpha_\varphi - \ell) \int_0^1 B_{2d_b+2,\alpha_\varphi}(\sigma) e^{2\pi i(-d_b+(2d_b+1)\sigma)\frac{n_b}{N_\varphi}} d\sigma,$$

which leads to

$$|A_1| \le \left(\frac{|n_b|}{N_\varphi} \right)^{2d_b+2} \frac{(2\pi)^{2d_b+2}}{(2d_b+2)!} \prod_{\ell=-d_b}^{d_b+1} |\alpha_\varphi - \ell| \,,$$

and therefore

$$|A_1| \le \left(\frac{\pi |n_b|}{N_\varphi} \right)^{2d_b+2} \frac{4\alpha_\varphi(1-\alpha_\varphi)}{\sqrt{\pi(d_b+1)}}. \tag{3.26}$$

Now we want an upper bound for the L^2-norm of the global error at time t_n. We have

$$\begin{aligned} \left\| e^{(n)} \right\|_2 &\le \sum_{k=0}^{n-1} \left[\frac{1}{N_\theta N_\varphi} \sum_{i_\theta=1}^{N_\theta} \sum_{i_\varphi=1}^{N_\varphi} \left([A_1 + A_2]_{i_\theta,i_\varphi}^{(k)} \right)^2 \right]^{\frac{1}{2}} \le \sum_{k=0}^{n-1} \max_{i_\theta,i_\varphi} \left| [A_1 + A_2]_{i_\theta,i_\varphi}^{(k)} \right| \\ &\le n \max_{i_\theta,i_\varphi,k} \left| [A_1]_{i_\theta,i_\varphi}^{(k)} \right| + n \max_{i_\theta,i_\varphi,k} \left| [A_2]_{i_\theta,i_\varphi}^{(k)} \right| . \end{aligned}$$

Our estimates for $|A_1|$ and $|A_2|$ are independent of the grid indices i_θ and i_φ, and therefore they also apply to the maximum over the domain. Moreover, we observe that such estimates apply to any time instant, because they are invariant to the rigid translation that the exact solution undergoes in time. Accordingly, our upper bound for the global error of the field-aligned semi-Lagrangian scheme is simply $\|e^{(n)}\|_2 \le n|A_1| + n|A_2|$, with $|A_1|$ bounded by (3.26) and $|A_2|$ bounded by (3.25):

$$\left\| e^{(n)} \right\|_2 \le n \left(\frac{\pi|m_\theta|}{N_\theta} \right)^{2d_\theta+2} \frac{4 \sum_{k=-d_b}^{d_b+1} |L_k^{d_b}(\alpha_\varphi)| \alpha_{\theta,k}(1-\alpha_{\theta,k})}{\sqrt{\pi(d_\theta+1)}} + n \left(\frac{\pi|n_b|}{N_\varphi} \right)^{2d_b+2} \frac{4\alpha_\varphi(1-\alpha_\varphi)}{\sqrt{\pi(d_b+1)}}. \tag{3.27}$$

We notice that for sufficiently small values of b_θ we have $r_{\theta,k} = 0$ and $\alpha_{\theta,k}(1-\alpha_{\theta,k}) \propto |b_\theta|$. Therefore in the limit as $b_\theta \to 0$ the first error term goes to zero and we recover the classical error bound for 1D semi-Lagrangian schemes with $n_b = n_\varphi$. In the following discussion we will assume that $b_\theta \neq 0$.
We now want to assess the consistency of the scheme, that is, whether at a fixed time $t_n = n\Delta t = T$ the global error goes to zero in the limit as $\Delta t, \Delta\theta, \Delta\varphi \to 0$ (or equivalently in the limit as $n, N_\theta, N_\varphi \to \infty$). If we assume that the three parameters converge to zero according to the algebraic relationships

$$\Delta\varphi = c\Delta\theta, \qquad \Delta t = \Delta\theta^\gamma, \qquad c, \gamma \in \mathbb{R}_+^*,$$

we can distinguish between two different cases:

1. If $0 < \gamma \le 1$, the Courant numbers along θ and φ either grow as $\Delta t \to 0$ (for $\gamma < 1$) or they are constant ($\gamma = 1$), therefore it is appropriate to use the upper bound $4\alpha(1-\alpha) \le 1$. Moreover, we use the identities

$$\begin{aligned} n &= \frac{T}{\Delta t} = \frac{T}{\Delta\theta^\gamma} = T \left(\frac{2\pi}{N_\theta} \right)^{-\gamma} = T \left(\frac{\pi|m_\theta|}{N_\theta} \right)^{-\gamma} \left(\frac{|m_\theta|}{2} \right)^\gamma, \\ n &= \frac{T}{\Delta t} = \frac{T}{(\Delta\varphi/c)^\gamma} = T \left(\frac{2\pi}{cN_\varphi} \right)^{-\gamma} = T \left(\frac{\pi|n_b|}{N_\varphi} \right)^{-\gamma} \left(\frac{c|n_b|}{2} \right)^\gamma, \end{aligned}$$

to obtain

$$\left\| e^{(n)} \right\|_2 \le T \left(\frac{\pi|m_\theta|}{N_\theta} \right)^{2d_\theta+2-\gamma} \frac{(|m_\theta|/2)^\gamma \, G_{d_b}}{\sqrt{\pi(d_\theta+1)}} + T \left(\frac{\pi|n_b|}{N_\varphi} \right)^{2d_b+2-\gamma} \frac{(c|n_b|/2)^\gamma}{\sqrt{\pi(d_b+1)}}, \tag{3.28}$$

where

$$G_d = \max_{\alpha\in[0,1]} \left(\sum_{k=-d}^{d+1} \left| L_k^d(\alpha) \right| \right), \qquad d \in \mathbb{N},$$

is the central local maximum of the Lebesgue function for Lagrange interpolation on $2d+2$ equispaced nodes [16,17]. Such a maximum is obtained for $\alpha = 1/2$, and corresponds to the Landau constant [18, 19]. The asymptotic behavior $G_d \sim \log(d)/\pi$ for $d \to \infty$ was predicted by Landau [20], and various bounds valid for all d have been given by many authors (e.g., see [21]). Here we report the computed values of practical interest:

$$\begin{array}{lllll} G_0 = 1, & G_1 = 1.25, & G_2 \approx 1.39, & G_3 \approx 1.49, & G_4 \approx 1.56, \\ G_5 \approx 1.62, & G_6 \approx 1.67, & G_7 \approx 1.72, & G_8 \approx 1.76, & G_9 \approx 1.79. \end{array}$$

2. If $\gamma > 1$, the Courant numbers along θ and φ go to zero, and therefore for Δt sufficiently small we have one of these two situations:

$$\text{a) if } b_\varphi < 0\text{:} \begin{cases} r_\varphi = 0 \\ \alpha_\varphi \to 0^+ \\ \alpha_{\theta,0} = 0 \end{cases} \qquad \text{b) if } b_\varphi > 0\text{:} \begin{cases} r_\varphi = -1 \\ \alpha_\varphi \to 1^- \\ \alpha_{\theta,1} = 0 \end{cases}$$

For the sake of brevity, we only consider case (a); since our Lagrange interpolant is constructed on an even number of equispaced nodes, it can be shown that the final result of this discussion is identical for case (b). As α_φ goes to zero, we now have

$$L_0^{d_b}(\alpha_\varphi) \underset{0}{\sim} 1, \qquad L_k^{d_b}(\alpha_\varphi) \underset{0}{\sim} \left(\frac{1}{k} \prod_{\substack{\ell=-d_b \\ \ell \notin \{0,k\}}}^{d_b+1} \frac{\ell}{\ell - k} \right) \alpha_\varphi = D_k^{d_b} \alpha_\varphi \quad \text{for } k \neq 0.$$

In general for $b_\theta \neq 0$ we have $\alpha_{\theta,k} \neq 0$ for $k \neq 0$. Therefore, if we use the upper bound $4\alpha_{\theta,k}(1-\alpha_{\theta,k}) \leq 1$ and the asymptotic equivalence

$$\sum_{k=-d_b}^{d_b+1} |L_k^{d_b}(\alpha_\varphi)| \underset{0}{\sim} \left(\sum_{\substack{k=-d_b \\ k\neq 0}}^{d_b+1} \left| D_k^{d_b} \right| \right) \alpha_\varphi = C_{d_b} \alpha_\varphi,$$

we obtain the estimate

$$4 \sum_{k=-d_b}^{d_b+1} |L_k^{d_b}(\alpha_\varphi)| \alpha_{\theta,k}(1-\alpha_{\theta,k}) \leq (C_{d_b} + \mathcal{O}(\Delta t))\, \alpha_\varphi,$$

where the $\mathcal{O}(\Delta t)\alpha_\varphi$ term represents the error that results from truncating the MacLaurin expansion of $\sum_{k=-d_b}^{d_b+1} |L_k^{d_b}(\alpha_\varphi)|$. Finally, we use this last estimate together with the identities

$$\begin{aligned} n\alpha_\varphi &= \frac{T}{\Delta t} \frac{|b_\varphi|\Delta t}{\Delta\varphi} = T \left(\frac{2\pi}{N_\varphi}\right)^{-1} |b_\varphi| = T \left(\frac{\pi|n_b|}{N_\varphi}\right)^{-1} \frac{|b_\varphi n_b|}{2}, \\ n\alpha_\varphi &= \frac{T}{\Delta t} \frac{|b_\theta|\Delta t}{\Delta\theta} \left|\frac{b_\varphi \Delta\theta}{b_\theta \Delta\varphi}\right| = T \left(\frac{2\pi}{N_\theta}\right)^{-1} \left|\frac{b_\theta}{\lambda}\right| = T \left(\frac{\pi|m_\theta|}{N_\theta}\right)^{-1} \frac{|b_\theta m_\theta|}{2|\lambda|}, \end{aligned}$$

to get the approximate upper bound (not valid in the limit as $b_\theta \to 0$)

$$\left\| e^{(n)} \right\|_2 \leq T \left(\frac{\pi|m_\theta|}{N_\theta}\right)^{2d_\theta+1} \frac{2|b_\theta m_\theta|}{\sqrt{\pi(d_\theta+1)}} \left[\frac{C_{d_b}}{|4\lambda|} + O(\Delta t) \right] + T \left(\frac{\pi|n_b|}{N_\varphi}\right)^{2d_b+1} \frac{2|b_\varphi n_b|}{\sqrt{\pi(d_b+1)}}. \tag{3.29}$$

The magnitude of $\lambda = b_\theta N_\theta/(b_\varphi N_\varphi)$ is discussed in the next section, where a comparison with the classical scheme is presented. Here we compute C_{d_b}, which grows logarithmically with d_b and has values

$$\begin{array}{lllll} C_0 = 1, & C_1 = 1.5, & C_2 \approx 1.83, & C_3 \approx 2.08, & C_4 \approx 2.28, \\ C_5 = 2.45, & C_6 \approx 2.59, & C_7 \approx 2.71, & C_8 \approx 2.83, & C_9 \approx 2.93. \end{array}$$

For all possible values of γ, we have shown that our field-aligned semi-Lagrangian scheme is consistent (i.e., the error goes to zero as $\Delta t, \Delta\theta, \Delta\varphi \to 0$). Given stability and consistency, we have proven convergence of our method.

Remark 3.2. *We point out that the limit as $\gamma \to 0$ corresponds to a constant Δt, i.e. no time refinement: the scheme correctly converges to the exact solution as N_θ and N_φ are increased, and our first estimate* (3.28) *applies. Conversely, the limit as $\gamma \to \infty$ corresponds to constant N_θ and N_φ, i.e. no spatial refinement: our second estimate* (3.29) *applies, because it is independent of γ, and the error goes to a constant value, without diverging, as Δt is reduced.*

3.3.3 Comparison with classical (not aligned) approach

If a standard tensor-product 2D interpolation is used, one could show that the two 1D interpolation operators exactly commute, and their corresponding approximation errors independently contribute to the local truncation error. As a result, an upper bound for the L^2-norm of the global error is simply

$$\left\| e^{(n)} \right\|_2 \leq n \left(\frac{\pi|m_\theta|}{N_\theta} \right)^{2d_\theta+2} \frac{4\alpha_\theta(1-\alpha_\theta)}{\sqrt{\pi(d_\theta+1)}} + n \left(\frac{\pi|n_\varphi|}{N_\varphi} \right)^{2d_\varphi+2} \frac{4\alpha_\varphi(1-\alpha_\varphi)}{\sqrt{\pi(d_\varphi+1)}}. \tag{3.30}$$

By comparing (3.27) with (3.30), we immediately notice that the second error term is much smaller in the field-aligned case if $|n_b| < |n_\varphi|$, that is, if the gradients along $\mathbf{b}$ are smaller than the gradients along φ. Specifically, if we assume that $d_\varphi = d_b = d$, the error is reduced by a factor $(n_b/n_\varphi)^{2d+2}$. Vice versa, if we seek to reduce the number of points along the φ direction for a given error level, then the field aligned scheme allows us to use only $N_\varphi|n_b/n_\varphi|$ points.

The first error term is more difficult to compare, because it has a more complicated form in the field aligned case. In general terms, we can say that the error constant is somewhat larger because of the additional interpolations that are required; in order to quantify this overhead, we now look at the rate of convergence of the classical scheme for the various values of γ:

1. For $0 \leq \gamma \leq 1$ we have the estimate

$$\left\| e^{(n)} \right\|_2 \leq T \left(\frac{\pi|m_\theta|}{N_\theta} \right)^{2d_\theta+2-\gamma} \frac{(|m_\theta|/2)^\gamma}{\sqrt{\pi(d_\theta+1)}} + T \left(\frac{\pi|n_\varphi|}{N_\varphi} \right)^{2d_\varphi+2-\gamma} \frac{(c|n_\varphi|/2)^\gamma}{\sqrt{\pi(d_\varphi+1)}},$$

   therefore the first error term of the field-aligned scheme in (3.28) is larger by a factor equal to the Landau constant G_{d_b}, which is smaller than 2 for the cases of practical interest;

2. For $\gamma > 1$ we have the estimate

$$\left\| e^{(n)} \right\|_2 \leq T \left(\frac{\pi|m_\theta|}{N_\theta} \right)^{2d_\theta+1} \frac{2|b_\theta m_\theta|}{\sqrt{\pi(d_\theta+1)}} + T \left(\frac{\pi|n_\varphi|}{N_\varphi} \right)^{2d_\varphi+1} \frac{2|b_\varphi n_\varphi|}{\sqrt{\pi(d_\varphi+1)}},$$

   therefore the first error term of the field-aligned scheme in (3.29) is multiplied by a factor $[C_{d_b}/|4\lambda| + O(\Delta t)]$. We have already shown that $C_d < 3$ for $d \leq 9$, therefore $C_{d_b}/4 < 1$ for the cases of practical interest. It now remains to see if $|\lambda| \geq 1$, which is not an obvious task given that both $|b_\theta/b_\varphi|$ and N_φ/N_θ are small numbers in practice. Here we assume that $|n_b/n_\varphi| \ll 1$, which is the condition that justifies the use of a field-aligned approach. An appropriate mesh for the classical scheme would require $c = N_\theta/N_\varphi \approx |m_\theta/n_\varphi|$, which yields the estimate

$$|\lambda| = \frac{b_\theta N_\theta}{b_\varphi N_\varphi} \approx \left| \frac{b_\theta m_\theta}{b_\varphi n_\varphi} \right| = \left| \frac{b_\theta}{b_\varphi} \frac{1}{n_\varphi} \frac{b_\varphi}{b_\theta} (n_b - n_\varphi) \right| = \left| \frac{n_b}{n_\varphi} - 1 \right| \approx 1,$$

   while an appropriate mesh for the field-aligned scheme would be coarser in φ, specifically $c = N_\theta/N_\varphi \approx |m_\theta/n_b|$, yielding

$$|\lambda| \approx \left| \frac{b_\theta m_\theta}{b_\varphi n_\varphi} \frac{n_\varphi}{n_b} \right| = \left| 1 - \frac{n_\varphi}{n_b} \right| \gg 1.$$

Therefore, we can say that $|\lambda| \geq 1$ in all situations where the mesh is not overly refined in φ. This leads to $C_{d_b}/|4\lambda| < 1$: for sufficiently small Δt the first error term is smaller for the field-aligned scheme than for the classical scheme.

Overall we can conclude that the field-aligned semi-Lagrangian scheme allows for important computational savings, of the order of $|n_\varphi/n_b|$, for those situations where the gradients are smaller along $\mathbf{b}$ than along φ. The price to pay is an increased error constant for convergence in N_θ. Such an increase is negligible for $0 \le \gamma \le 1$, which are the conditions where refinement usually occurs. In the unusual situation where the Δt refinement dominates ($\gamma > 1$), the increase may be substantial only if the mesh is overly refined in φ (which are not the conditions in which we intend to use the field-aligned scheme). In all cases, the order of convergence is unaffected.

4 A Screw Pinch Model in Cylindrical Geometry with Selalib

In order to validate the field aligned approach, we consider now a first simplified gyrokinetic simulation, developed in the framework of the Selalib library [22]. It consists in a $4D$ drift-kinetic equation in cylindrical geometry with an *oblique* magnetic field as defined in (1.6); the complete derivation is given in Appendix A. It is a generalization of the $4D$ drift-kinetic equation in cylindrical geometry with a *uniform* magnetic field in the z direction (corresponding to $\iota(r) = 0$ and thus $\zeta(r) = 0$ in (1.6)), which has been first developed in [5] and then reproduced in [23] for example. In the uniform case, we are able to check the linear phase behaviour, by solving numerically the dispersion relation and compare the simulation outputs with it (see also [24, 25]). Note that the dispersion relation depends on $k_\|$: this permits to compare simulations in the oblique and uniform case, in order to check the correctness of the simulations in the oblique case, as we will see. Another (more straightforward) way to validate the code will be to double the number of points along the φ direction (in practice, we will compare $N_\varphi = 32$ with $N_\varphi = 64$) and to observe that the results do not significantly change (convergence of numerical discretizations). We could also have compared the code with a standard (i.e. not using a field aligned interpolation) approach with a refined mesh, but this would have required to develop the corresponding code. Such an approach is not tackled in Section 4, but it will be employed in Section 5 in the framework of the GYSELA code.

4.1 Model equations

We look for $f = f(t, r, \theta, z, v_\|)$ satisfying

$$\partial_t f + [\phi, f] + v_\| \nabla_\| f - \nabla_\| \phi \, \partial_{v_\|} f = 0,$$

with

$$[\phi, f] = -\frac{\partial_\theta \phi}{r B_0} \partial_r f + \frac{\partial_r \phi}{r B_0} \partial_\theta f, \quad \nabla_\| = \mathbf{b} \cdot \nabla,$$

so that

$$\partial_t f - \frac{\partial_\theta \phi}{r B_0} \partial_r f + \left(\frac{\partial_r \phi}{r B_0} + v_\| \frac{b_\theta}{r} \right) \partial_\theta f + v_\| b_z \partial_z f - \left(b_\theta \frac{\partial_\theta \phi}{r} + b_z \partial_z \phi \right) \partial_{v_\|} f = 0, \tag{4.1}$$

for $t \in [0, t_{\text{end}}]$, $(r, \theta, z) \in [r_{\min}, r_{\max}] \times [0, 2\pi] \times [0, 2\pi R_0]$, and $v_\| \in [-v_{\max}, v_{\max}]$. Here, we have $z = R_0 \varphi$. The self-consistent potential $\phi = \phi(t, r, \theta, z)$ solves the quasi-neutral equation without zonal flow

$$-\left(\partial_r^2 \phi + \left(\frac{1}{r} + \frac{\partial_r n_0}{n_0} \right) \partial_r \phi + \frac{1}{r^2} \partial_\theta^2 \phi \right) + \frac{1}{T_e} \phi = \frac{1}{n_0} \left(\int_{-\infty}^{\infty} (f - f_{eq}) dv_\| \right). \tag{4.2}$$

When $\iota = \frac{b_\theta / r}{b_z / R_0} = 0$, we recover the classical drift kinetic model given in [5, 23] for example. A similar model has been simulated in [26], with $\iota = 0.8$ as an example, using a Particle in Cell method.

We note that all quantities appearing in these equations are non-dimensional. The equations themselves can be derived from (1.1), (1.3a), (1.3b) neglecting terms in power of $\iota r / R_0$ (see Appendix A).

The boundary conditions on f are the following:

- Periodicity along θ, z and $v_\|$;
- Zeroth-order extrapolation along r, i.e. we give values to f outside the domain (for interpolation at the foot of the characteristic) according to the scheme

$$f(t, r, \theta, z, v_\|) = \begin{cases} f(t, r_{\min}, \theta, z, v_\|) & \text{if } r < r_{\min}, \\ f(t, r_{\max}, \theta, z, v_\|) & \text{if } r > r_{\max}. \end{cases}$$

The boundary conditions on ϕ are the following:

- Periodicity along θ and z;
- Neumann mode 0 (see [23]) at $r = r_{\min}$, that is, if we decompose ϕ into its Fourier modes $\widehat{\phi}_k$ along θ:
  - homogeneous Neumann boundary condition for the Fourier mode 0 ($\partial_r \widehat{\phi}_0(t, r_{\min}) = 0$), i.e. $\int_0^{2\pi} \partial_r \phi(t, r_{\min}, \theta) d\theta = 0$;
  - homogeneous Dirichlet boundary condition for all other Fourier modes ($\widehat{\phi}_k(t, r_{\min}) = 0 \ \forall k$), i.e. $\partial_\theta \phi(t, r_{\min}, \theta) = 0$.
- Homogeneous Dirichlet at $r = r_{\max}$, that is $\phi(t, r_{\max}, \theta) = 0$;

The initial function is given by

$$f(t=0, r, \theta, z, v_\parallel) = f_{\text{eq}}(r, v_\parallel) \left[1 + \epsilon \exp\left(-\frac{(r - r_p)^2}{\delta r}\right) \cos\left(m\theta + \frac{n}{R_0} z\right)\right],$$

where the equilibrium function is

$$f_{\text{eq}}(r, v_\parallel) = \frac{n_0(r)}{\sqrt{2\pi T_i(r)}} \exp\left(-\frac{v_\parallel^2}{2T_i(r)}\right).$$

The radial profiles $\{T_i, T_e, n_0\}$ have the analytical expressions

$$\mathcal{P}(r) = C_{\mathcal{P}} \exp\left(-\kappa_{\mathcal{P}}\, \delta r_{\mathcal{P}} \tanh\left(\frac{r - r_p}{\delta r_{\mathcal{P}}}\right)\right), \quad \mathcal{P} \in \{T_i, T_e, n_0\},$$

where the constants are

$$C_{T_i} = C_{T_e} = 1, \quad C_{n_0} = \frac{r_{\max} - r_{\min}}{\int_{r_{\min}}^{r_{\max}} \exp\left(-\kappa_{n_0} \delta r_{n_0} \tanh\left(\frac{r - r_p}{\delta r_{n_0}}\right)\right) dr}.$$

The dispersion relation reads (see Appendix B)

$$\begin{aligned} -\partial_r^2 \phi - \left(\frac{1}{r} + \frac{\partial_r n_0}{n_0}\right) \partial_r \phi + \frac{m^2}{r^2} \phi + \frac{1}{T_e} \phi = \\ = \left[-\frac{1}{T_i}(1 + \tilde{z} Z(\tilde{z})) + \frac{m}{k^* r B_0} \left(Z(\tilde{z}) \left(\frac{\partial_r n_0}{n_0} - \frac{\partial_r T_i}{2T_i}\right) + \tilde{z}(1 + \tilde{z} Z(\tilde{z})) \frac{\partial_r T_i}{T_i}\right)\right] \phi, \end{aligned}$$

with $\tilde{z} = \omega / k^*$, $k^* = k_\parallel \sqrt{2T_i}$, and $k_\parallel = (b_\theta m / r + b_z n / R_0)$. Here Z is the so-called 'plasma dispersion function' [27], defined as

$$Z(u) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{\exp(-x^2)}{x - u} dx = i\sqrt{\pi} \exp(-u^2)(1 + \text{erf}(iu)), \quad \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt.$$

Note that the dispersion relation depends on m and $k_\parallel$ and not directly on n. This means that taking different values of ι and n but with same m and $k_\parallel$ will lead to the same dispersion relation.

4.2 Numerical methods

For time-stepping of (4.1) we use a predictor-corrector scheme closely related to the explicit midpoint rule for ordinary differential equations: starting from the solution at time t, the fields at time $t + \Delta t/2$ are evaluated to first order accuracy in Δt (predictor), and are then used to update the solution at time $t + \Delta t$ to second order accuracy (corrector). Both the predictor and the corrector algorithms are splitting methods, where the 4D gyrokinetic Vlasov equation (4.1) is decomposed into three separate advection equations:

A. 2D advection on a magnetic flux surface (θ, z), with constant velocity $v_\parallel \mathbf{b}$;

$$\partial_t f + v_\parallel \nabla_\parallel f = 0;$$

B. 1D advection along $v_\parallel$, with constant velocity $\nabla_\parallel\tilde{\phi}$,

$$\partial_t f + \nabla_\parallel \tilde{\phi}\, \partial_{v_\parallel} f = 0;$$

C. 2D advection on the poloidal plane (r,θ),

$$\partial_t f + [\tilde{\phi}, f] = 0.$$

Here $\tilde{\phi}(r,\theta,z)$ is a constant-in-time approximation of the time varying field $\phi(t,r,\theta,z)$. The three equations above are all solved using backward semi-Lagrangian methods. Specifically, for equation A we use the field-aligned algorithm described in Section 3, with a slight modification: we use cubic spline interpolation in the θ direction, and 5th order Lagrange interpolation (field-aligned) in the z direction. For equation B we use 1D cubic spline interpolation, and the parallel gradient of $\tilde{\phi}$ is computed by 2nd order finite differences in the $\mathbf{b}$ direction. For equation C we use 2D tensor-product cubic spline interpolation, and since the flow field is not uniform, we calculate the feet of the 2D characteristic trajectories by means of the Verlet algorithm: let $\dot{X} = u_1(X,Y)$ and $\dot{Y} = u_2(X,Y)$ be the characteristic equations of C, and $(X^{n+1},Y^{n+1}) = (r_i,\theta_j)$ be the final position of one characteristic trajectory at time t_{n+1}; the foot (X^n,Y^n) of the characteristic is calculated as

$$\begin{aligned}
X^* &= X^{n+1} - \frac{\Delta t}{2}\, u_1(X^*, Y^{n+1}),\\
Y^n &= Y^{n+1} - \frac{\Delta t}{2}\left(u_2(X^*,Y^{n+1}) + u_2(X^*,Y^n)\right),\\
X^n &= X^* \quad - \frac{\Delta t}{2}\, u_1(X^*, Y^n).
\end{aligned}$$

We use Lie splitting (1st order) as predictor and Strang splitting (2nd order) as corrector; the complete time-stepping algorithm then reads:

1. Compute $\tilde{\phi}$ from f^n by solving the quasi-neutral equation (4.2);
2. Compute $\tilde{f}^{n+1/2}$ using Lie splitting: $\tilde{f}^{n+1/2} = C(\frac{\Delta t}{2})B(\frac{\Delta t}{2})A(\frac{\Delta t}{2})f^n$;
3. Compute $\tilde{\phi}$ from $\tilde{f}^{n+1/2}$ by solving (4.2) again;
4. Compute f^{n+1} using Strang splitting: $f^{n+1} = A(\frac{\Delta t}{2})B(\frac{\Delta t}{2})C(\Delta t)B(\frac{\Delta t}{2})A(\frac{\Delta t}{2})f^n$.

The quasi-neutral equation (4.2) is an elliptic partial differential equation in the variables (r,θ), therefore it can be solved independently on each poloidal plane $z = z^*$. Taking advantage of the linearity of the differential operator and of the periodicity of the domain, we apply the discrete Fourier transform in θ to both sides of (4.2). Since the factor in front of ∂_θ^2 does not depend on θ and the boundary conditions in r are homogeneous, each Fourier coefficient $\hat{\phi}_k(r)$ solves a separate 1D boundary value problem on $[r_{\min}, r_{\max}]$ and is independent of the other coefficients:

$$-\left[\partial_r^2 + \left(\frac{1}{r} + \frac{\partial_r n_0(r)}{n_0(r)}\right)\partial_r + \left(\frac{1}{T_e(r)} - \frac{k^2}{r^2}\right)\right]\hat{\phi}_k(r) = \hat{\rho}_k(r), \qquad k = 0,1,\ldots,N_\theta - 1.$$

For each mode k, this ordinary differential equation is collocated at the grid points $r = r_i$, and the derivatives are approximated by 2nd-order central finite differences. Once the proper boundary conditions are taken into account (see previous section), calculating $\{\hat{\phi}_k(r_i) : \forall i\}$ requires the solution of a tridiagonal linear system of size N_r. When all modes k are computed, the potential on the polar plane $z = z^*$ is reconstructed.

4.3 Numerical results

We consider the parameters of [24] (MEDIUM case)

$$r_{\min} = 0.1, \quad r_{\max} = 14.5, \quad \kappa_{n_0} = 0.055, \quad \kappa_{T_i} = \kappa_{T_e} = 0.27586, \quad \delta r_{T_i} = \delta r_{T_e} = \frac{\delta r_{n_0}}{2} = 1.45,$$

$$\epsilon = 10^{-6}, \quad R_0 = 239.8081535, \quad r_p = \frac{r_{\min} + r_{\max}}{2}, \quad \delta r = \frac{4\delta r_{n_0}}{\delta r_{T_i}}.$$

We take $B_0 = -1$. Given the magnetic field (1.6), we recall that $b_\theta = \zeta b_z$ and $\zeta = \iota r/R_0$, therefore $k_\| = (\iota m + n) b_z/R_0$. As our first test-case we consider a straight magnetic field with $\iota = 0$ and excite the mode $(m, n) = (15, 1)$, which leads to $k_\| = 1/R_0$. In our second test-case we consider a twisted magnetic field with $\iota = 0.8$ and choose $(m, n) = (15, -11)$, which leads to $k_\| = (0.8 \cdot 15 - 11)\, b_z/R_0 = b_z/R_0$. We note that we have for the second case $b_z = 1/\sqrt{1+\zeta^2}$ with $0 \le \zeta = \iota r/R_0 \le \iota r_{\max}/R_0 \le 0.05$, so that $|b_z - 1| \le 1.25 \cdot 10^{-3}$. The two cases have the same value of m and, thanks to the fact that $b_z \approx 1$, almost identical values of $k_\|$. Thus the dispersion relation, which only depends on m and $k_\|$, yields almost the same result in both cases, which means that the two simulations should give very similar results in the poloidal plane, at least in the linear phase (as it is also observed in [26]). The test-case parameters are summarized in Table 1, together with the resulting frequencies calculated from the analytical dispersion relation.

	ι	m	n	$k_\|$	$\Re(\omega)$	$\Im(\omega)$
Case 1	0	15	1	$1/R_0$	2.0480×10^{-3}	3.8174×10^{-3}
Case 2	0.8	15	-11	b_z/R_0	2.0471×10^{-3}	3.8166×10^{-3}

Table 1: Screw-pinch gyrokinetic model: input parameters and linear response. ι is the (constant) magnetic rotational transform, m and n are the polar and axial mode numbers of the initial conditions, $k_\|$ is the resulting parallel wave number, and ω is the complex frequency calculated from the dispersion relation. The first three significant digits of ω (both real and imaginary parts) are the same in both test-cases.

We take LAG5 for the interpolation along the parallel direction and cubic splines for the interpolation along θ. Finite differences of order 6 are used for the derivative computation along the parallel direction and cubic splines are used otherwise. When $\iota = 0$, we use the classical method with cubic splines for the interpolation along the z direction. In all simulations we take $N_r = 256$, $N_\theta = 512$ and $N_v = 128$; we use $N_\varphi = 32$ for $\iota = 0$, and $N_\varphi \in \{32, 64\}$ for $\iota = 0.8$.

On Figure 2 we see the evolution of the diagnostic quantity

$$D(t) = \sqrt{\int_{r_{\min}}^{r_{\max}} \int_0^{2\pi} \phi^2(t, r, \theta, z = 0)\, dr d\theta}, \tag{4.3}$$

which is closely related to the total potential energy on the plane $z = 0$. The linear phase (left plot) is in accordance with the dispersion relation, and differences between the three runs become significant only in the non linear phase (right plot).

On Figure 3, we see the poloidal cut $f(t, r, \theta, z = 0, v_\| = 0)$ at time $T = 4000$ (end of the linear phase) and time $T = 6000$ (non-linear phase). Again there is accordance between the the figures with more visible differences in the non linear phase. Note that the solution becomes very complex at $T = 6000$ with lot of small scales which are difficult to capture; we already observe some diffusion effect, due to finite size of the grid; this is not the case at time $T = 4000$, where convergence still seems to occur.

On Figure 4, we see cuts in the $\theta - z$ plane of the distribution function. We clearly see the structure of the mode $(m, n) = (15, 1)$ in the straight case and $(m, n) = (15, -11)$ in the oblique case for $T = 4000$. Note that in these figures we use raw data for the visualisation. Since the number of points in N_φ is purposely low, the corresponding plots in the $\theta - z$ plane are necessarily coarse. This is not an indication of numerical problems. Indeed a better visualisation in this plane can be achieved by reconstructing the distribution function on a finer mesh using the field aligned interpolation (it will be done in next section).

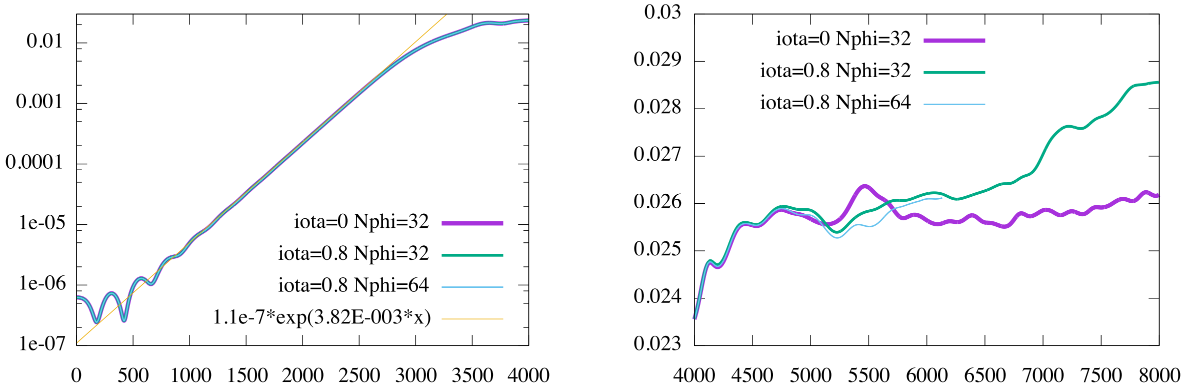

Figure 2: Screw-pinch gyrokinetic model: time evolution of the diagnostic quantity (4.3). On the left-hand side we plot the linear phase ($t \in [0, 4000]$) using a semi-logarithmic scale: all simulations follow the exponential growth rate computed from the dispersion relation. On the right-hand side we plot the non-linear phase ($t \in [4000, 8000]$): differences between the different simulations are visible on a linear scale.

5 Physical Cases in Gysela (Toroidal Geometry)

5.1 Gysela model

Let $\mathbf{z} = (r, \theta, \varphi, v_\parallel, \mu)$ be a variable describing the 5D phase space. The gyrokinetic Vlasov equation used by GYSELA is (1.1) in the electrostatic limit; further, all quantities are normalized. Temperatures are normalized to T_{e0}, i.e. the initial electron temperature at the mean radius $r_p = (r_{\min} + r_{\max})/2$, the electric potential is normalized to KT_{e0}/e_i, where K is the Boltzmann constant, and the magnetic field is normalized to B_0, i.e. the intensity at the magnetic axis. Time is normalized to the inverse of the ion cyclotron frequency $\omega_c = e_i\, B_0/m_i$ and velocities are given in units of the ion sound speed $v_{T_0} = \sqrt{KT_{e0}/m_i}$. Consequently, lengths are normalized to the Larmor radius $\rho_s = m_i\, v_{T_0}/(e_i B_0)$ and the magnetic moment μ to KT_{e0}/B_0. Finally the characteristic equations read

$$B_\parallel^* \frac{d\mathbf{X}}{dt} = v_\parallel \mathbf{B}^* + \mathbf{b} \times \nabla \left(\mu B + \langle \phi \rangle_\alpha\right),$$
$$B_\parallel^* \frac{dV_\parallel}{dt} = -\mathbf{B}^* \cdot \left(\mu B + \langle \phi \rangle_\alpha\right),$$

with $\mathbf{B}^* = \mathbf{B} + v_\parallel \nabla \times \mathbf{b}$ and $B_\parallel^* = \mathbf{b} \cdot \mathbf{B}^*$. In tokamak configurations, the plasma quasi-neutrality approximation is often made [5]. Electron inertia is ignored, which means that an adiabatic response of the electrons is assumed. We define the operator $\nabla_\perp = (\partial_r, \frac{1}{r}\partial_\theta)$, and we let $n_0(r)$ be the initial equilibrium density (integral over phase space - except r - of a reference equilibrium distribution function f_{ref}), and $T_e(r)$ be the electronic temperature. Further, J_0 the Bessel function of first order and $k_\perp$ the transverse component of the wave vector. Hence, the quasi-neutrality equation can be written in dimensionless variables as

$$-\frac{1}{n_0(r)} \nabla_\perp \cdot \left[\frac{n_0(r)}{B_0} \nabla_\perp \phi(r, \theta, \varphi)\right] + \frac{1}{T_e(r)} \left[\phi(r, \theta, \varphi) - \langle\, \phi \,\rangle\right] \;= \rho_i(r, \theta, \varphi), \tag{5.1}$$

where ρ_i is defined by

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

The potential ϕ couples back into the gyrokinetic Vlasov equation (1.1) through its derivatives, which play a major role in the term $\frac{d\mathbf{z}}{dt} B_\parallel^* f$. A detailed description of the model can be found in [6].

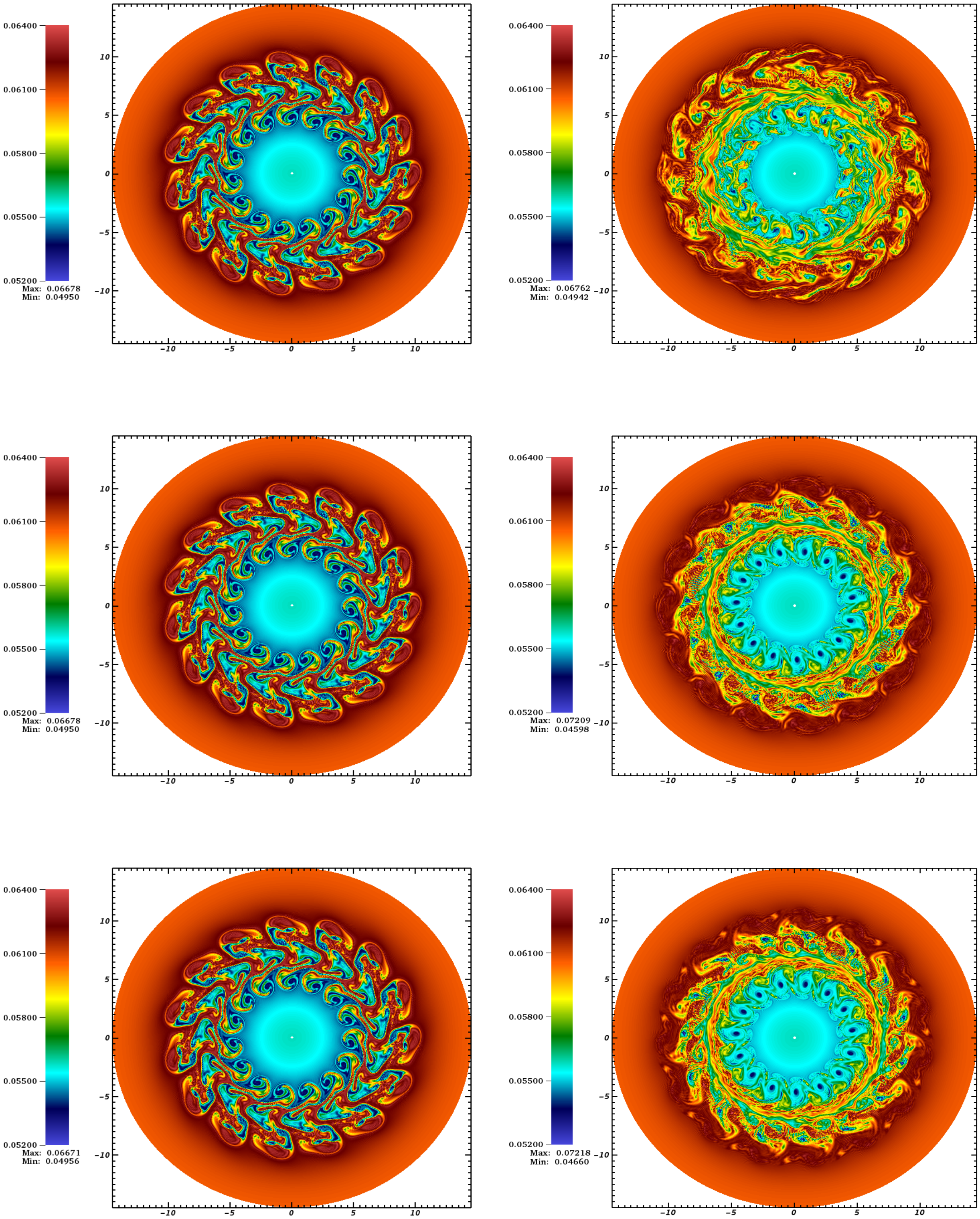

Figure 3: Screw-pinch gyrokinetic model: poloidal cut of the solution. We show $f(t = T, r, \theta, z = 0, v_\parallel = 0)$ at $T = 4000$ (left column) and $T = 6000$ (right column) for the three simulations: $\iota = 0, N_\varphi = 32$ (top row), $\iota = 0.8, N_\varphi = 32$ (middle row) and $\iota = 0.8, N_\varphi = 64$ (bottom row).

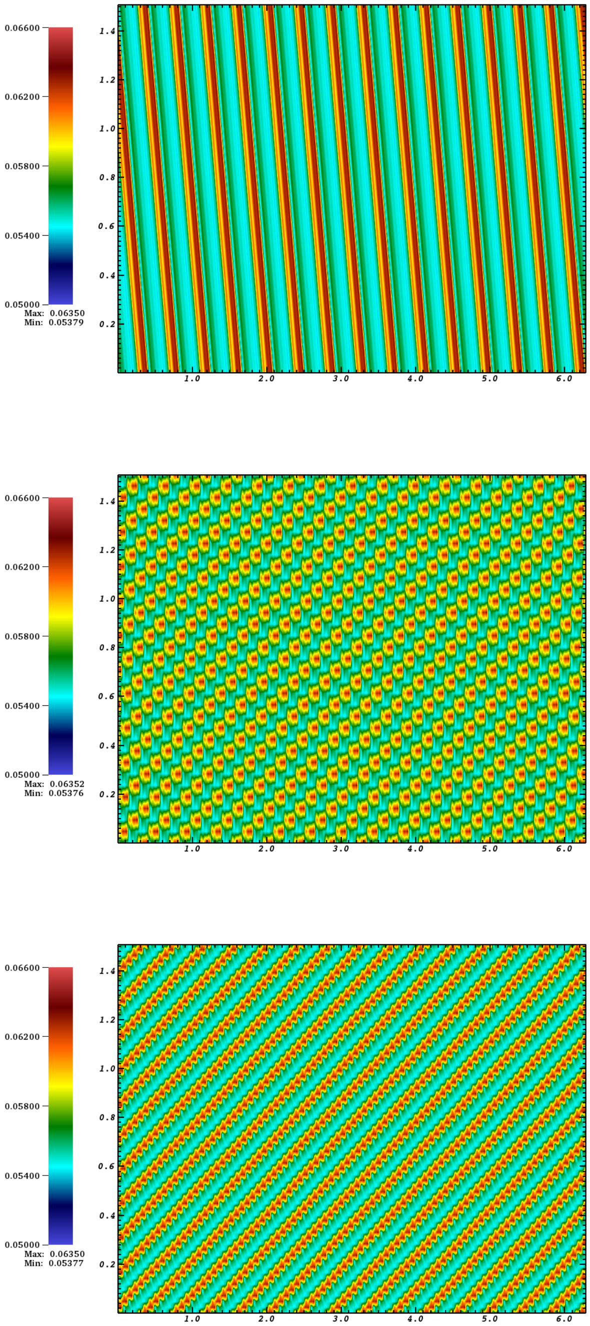

Figure 4: Screw-pinch gyrokinetic model: magnetic-surface cut of the solution. We show $f(t = T, r = (r_{\min} + r_{\max})/2, \theta, z, v_{\parallel} = 0)$ at $T = 4000$ for the three simulations: $\iota = 0, N_{\varphi} = 32$ (top), $\iota = 0.8, N_{\varphi} = 32$ (middle) and $\iota = 0.8, N_{\varphi} = 64$ (bottom).

5.2 Parallel algorithms

The algorithms and the parallelization strategies used in the Gysela code have been already described in previous works [28–30]. Algorithm 3 sketches the main features concerning the Vlasov solver that we are interested in here. The usual way to perform a single Vlasov solving in the Gysela code [5] consists of a series of directional advections: $(\hat{v_{\|}}/2, \hat{\varphi}/2, \hat{r\theta}, \hat{\varphi}/2, \hat{v_{\|}}/2)$. Each directional advection is performed with the semi-Lagrangian scheme. This procedure is named Strang-splitting and converges in $O(\Delta t^2)$. It decomposes the Vlasov solver into four 1D advections and one central 2D advection (in the poloidal plane (r, θ)). This solver uses two parallel domain decompositions for the distribution function f. The main rationale that justifies this approach is that advections along a given dimension need all points along this dimension in f. This constraint comes from the spline interpolants that we use actually. Therefore, the 1D advections along φ and $v_{\|}$ are performed with a domain decomposition that retains all points of f along these two dimensions $(\varphi, v_{\|})$ locally in the MPI process. Then, a transpose of the distributed data structure f is performed that involves large collective communications. Then, the 2D advection along both r and θ dimensions can be done, this step uses a local subdomain in φ, $v_{\|}$ and μ directions. After a second tranposition of f, two 1D advections are again performed.

```
1D advection in $v_{\|}$          $[\Delta t/2]$   $(\forall(\mu, r, \theta) = [local], \forall(\varphi, v_{\|}) = [*])$;
1D advection in $\varphi$          $[\Delta t/2]$   $(\forall(\mu, r, \theta) = [local], \forall(\varphi, v_{\|}) = [*])$;
**Transpose** $f$;
2D advection in $(r, \theta)$    $[\Delta t]$     $(\forall(\mu, \varphi, v_{\|}) = [local], \forall(r, \theta) = [*])$;
**Transpose** $f$;
1D advection in $\varphi$          $[\Delta t/2]$   $(\forall(\mu, r, \theta) = [local], \forall(\varphi, v_{\|}) = [*])$;
1D advection in $v_{\|}$          $[\Delta t/2]$   $(\forall(\mu, r, \theta) = [local], \forall(\varphi, v_{\|}) = [*])$;
```

Algorithm 3: Standard Gysela Vlasov solver

```
1:  1D advection in $v_{\|}$                        $(\forall(\mu, r, \theta) = [local], \forall(\varphi, v_{\|}) = [*])$;
2:  Get feet for 2D advection in $(\theta, \varphi)$   $(\forall(\mu, r, \theta) = [local], \forall(\varphi, v_{\|}) = [*])$;
3:  **Transpose** $f$, and redistribute feet;
4:  2D *aligned* advection in $(\theta, \varphi)$      $(\forall(\mu, v_{\|}) = [local], \forall(r, \theta, \varphi) = [*])$;
5:  **Transpose** $f$;
6:  2D advection in $(r, \theta)$                  $(\forall(\mu, \varphi, v_{\|}) = [local], \forall(r, \theta) = [*])$;
7:  Get feet for 2D advection in $(\theta, \varphi)$   $(\forall(\mu, \varphi, v_{\|}) = [local], \forall(r, \theta) = [*])$;
8:  **Transpose** $f$, and redistribute feet;
9:  2D *aligned* advection in $(\theta, \varphi)$      $(\forall(\mu, v_{\|}) = [local], \forall(r, \theta, \varphi) = [*])$;
10: **Transpose** $f$;
11: 1D advection in $v_{\|}$                        $(\forall(\mu, r, \theta) = [local], \forall(\varphi, v_{\|}) = [*])$;
```

Algorithm 4: New *aligned* Vlasov solver

In order to depart from the original algorithm to accommodate the aligned strategy, one can list the different constraints that must be taken into account. First, to use the aligned advection approach in (θ, φ) plane, it is of outmost importance to treat these two directions in a single step, it permits to apply easily the scheme introduced in section 2.1. Second, 2D advections in (r, θ) can not be suppressed or transformed into a simple advection along the r direction, because the non-linear terms in r and θ interact tightly. Third, to evaluate a new algorithm and a new Strang splitting, we should not undermine the existing parallelization strategy (to keep it simple in the Gysela code).

The proposed Algorithm 4 fulfills these constraints. The advections along $v_{\|}$ are unchanged. The aligned advections along (θ, φ) (lines 2, 4, 7, 9) replace the previous advections along the φ direction. All advective terms (except the non-linear ones) along the θ direction are treated in this aligned advection in (θ, φ). The 2D advections along (r, θ) are modified in order to keep only nonlinear terms in the θ direction. Advective terms along θ are split suitably between the 2D advection operator in (θ, φ) and 2D advection in (r, θ). Finally, this solution uses a new parallel decomposition at one single location only (distributing over MPI processes along $\mu, v_{\|}$) in the 2D aligned advection (lines 4, 9). Compared to the standard algorithm, the extra transpose and redistribute steps constitute a communication overhead. Another overhead comes from the computation of the feet of the characteristics (lines 2 and 7) that are performed in an already known parallel decomposition in order to have access to needed values that are stored with these parallel decompositions.

We then have a robust parallel solution that does not requires an entire overhaul of the Gysela code. Nevertheless some extra communications are created that we measure in the following. In a future work, we will be able to cut costs with a more sophisticate implementation. Indeed, several fixes can be foreseen. One among other solutions is described shortly hereafter. First, one can execute aligned advections (lines 4 and 7) using the usual parallel decomposition of line 6 $(\forall(\mu, \varphi, v_{\|}) = [local], \forall(r, \theta) = [*])$. It will require tricky (but not so costly) communication patterns to deal with the parallel decomposition along φ direction. Second, when this first change will be made we can mix the computations of feet (lines 2 and 7) with the corresponding 2D aligned advections and then eliminate the transposes of lines 5 and 8. Finally, with this new solution to come, we will reduce the communication volume: avoid transfer of the feet (lines 3 and 8) and remove the need of specific data distribution (lines 4 and 9) and associated data redistribution.

5.3 Numerical results with Gysela

In a gyrokinetic simulation with kinetic ions and adiabatic electrons it is to be expected that the smallest length scale is of the order of the ion Larmor radius ρ_s. This is due in part to the gyroaverage operator in configuration space, and in part to the averaging over μ that takes place when computing the charge density. Since ρ_s is also the quantity used for normalization of all lengths, we can say that a well-refined numerical simulation requires $\Delta r \ll 1$ and $r_{\max}\Delta\theta \ll 1$. A fundamental non-dimensional parameter in magnetic fusion devices is the ratio $\rho^* = \rho_s/A$, where A is the minor radius of the device. (In terms of non-dimensional quantities, the minor radius of the device is $a = A/\rho_s$ and therefore $\rho^* = 1/a$.) The number of degrees of freedom needed to represent a poloidal cut of the solution scale with $(\rho^*)^{-2}$, therefore smaller values of ρ^* lead to larger numerical simulations.

In order to have accurate and converged simulations, in this section we use a setup with a relatively large value of $\rho^* = 1/40$, and we consider a single μ-value of $\mu = 0$. Strictly speaking, in such a situation there is neither gyro-averaging nor μ-averaging, therefore there is no physical lower bound on the characteristic length scales; nevertheless, the solution is still well resolved at the end of our simulations. We investigate two physical cases with geometrical parameters

$$a = 40, \quad r_{\min} = 0.1\,a, \quad r_{\max} = 1.0\,a, \quad R_0 = 3\,a,$$

that differ in their safety factor profiles $q(r)$. Benchmarks have been realized with the 4D toroidal version of the Gysela code, on a fine computational domain of size

$$N_r = 256, \quad N_\theta = 256, \quad N_\varphi =< \mathit{not\ fixed} >, \quad N_{v_\parallel} = 48.$$

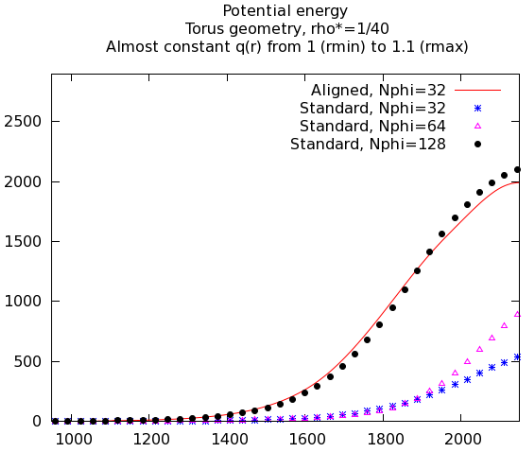

Figure 5: Potential energy plots for aligned or standard strategies. Toroidal configuration with almost constant safety factor along r direction.

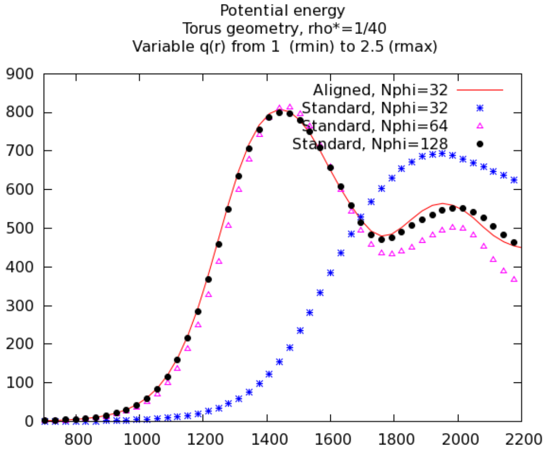

Figure 6: Potential energy plots for aligned or standard strategies. Toroidal configuration with safety factor depending on r coordinate.

A first case with an almost constant safety factor $q(r)$, slowly varying between $q(r_{\min}) = 1$ and $q(r_{\max}) = 1.1$, is illustrated by Figures 5, 7, and 8. A second case with a safety factor strongly depending on r, varying between $q(r_{\min}) = 1$ and $q(r_{\max}) = 2.5$, is illustrated by Figures 6, 9, and 10. The second case could be slightly more difficult to handle for the aligned approach, because the $\mathbf{b}$ direction depends on the r position through equation (1.7). Indeed, for each hyper-plane at a given r, the aligned advection algorithm uses possibly a different direction than for another r value. One can see on Fig. 5 that the standard approach with $N_\varphi = 128$ gives a similar result compared to aligned method with $N_\varphi = 32$. The two other curves with standard method and $N_\varphi = 32$ and $N_\varphi = 64$ are not converged along the φ direction and give substantially different potential energy evolutions. Figures 7 and 8 corroborate this fact by showing different cuts of the electric potential. In Figure 7, the two graphs at middle and bottom position show quite identical structures. It is important to notice that we have reconstructed finely the graph with $N_\varphi = 32$ in order to recover a fine resolution on the plots (through 4 aligned interpolations per original grid point, leading to a virtual $N_\varphi = 128$). In order to do that, we use Algorithm 1 with $(\theta^\star, \varphi^\star)$ being the grid points on the fine mesh.

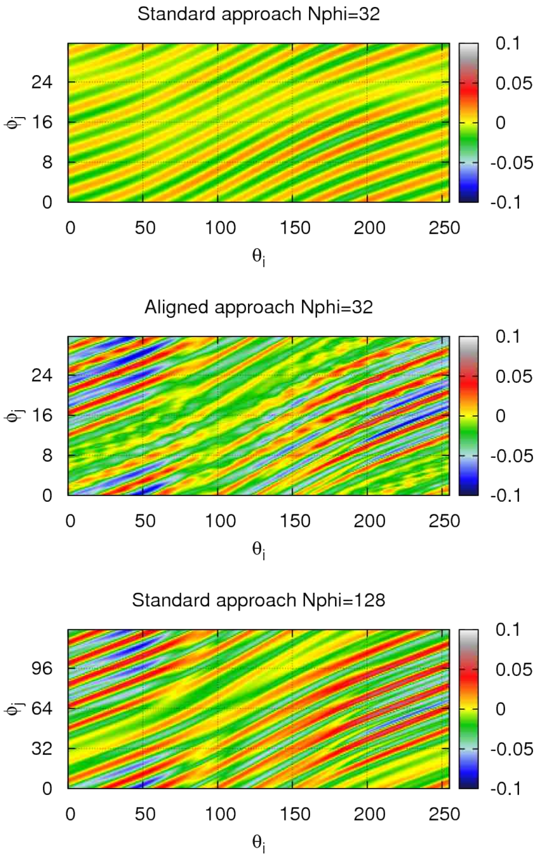

Figure 7: Cross-section of the electric potential at $r = 0.5$ and $t = 1672$. Standard simulation with $N_\varphi = 32$ (top), Aligned simulation with $N_\varphi = 32$ (middle), Standard simulation with $N_\varphi = 128$ (bottom).

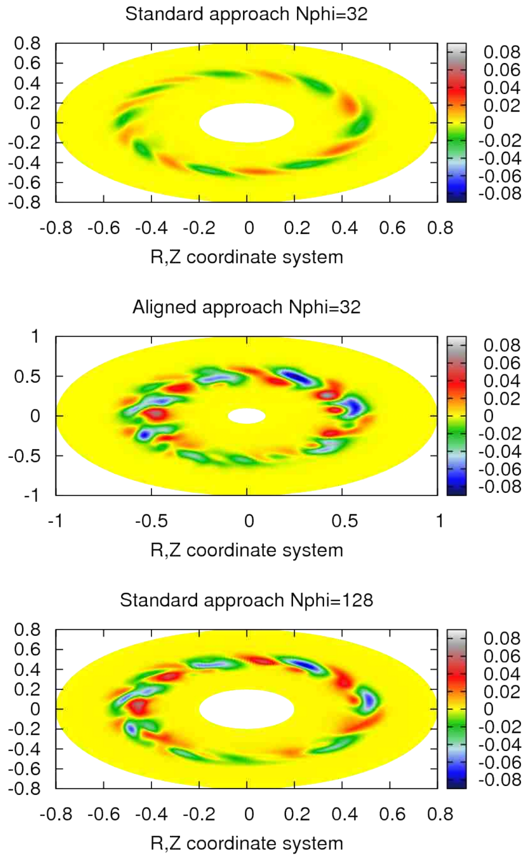

Figure 8: Poloidal cross-section of the electric potential at $\varphi = 0$ and $t = 1672$. Standard simulation with $N_\varphi = 32$ (top), Aligned simulation with $N_\varphi = 32$ (middle), Standard simulation with $N_\varphi = 128$ (bottom).

Figures 6, 9 and 10 show results for the second simulation with a strongly varying safety factor. Conclusions are quite analogous as the first simulation. On the left-hand side, one can see elongated structures along the parallel direction, which constitute the rationale that justifies why the aligned method reduces interpolation approximation errors. For these two simulations, we conclude that the aligned approach works well and permits to reduce by a factor of 4 the number of grid points in the φ direction for these cases at $\rho^\star = 1/40$. From the previous analysis, we also expect that, as ρ^* is further reduced to approach the ITER values of the order of 10^{-3} [31], it would not be necessary to increase the number of grid points in the φ direction in order to achieve comparable precision. Thus, our method could allow a saving of the order of 100 in grid points when employed in the context of realistic simulations of reactor scale devices.

5.4 Execution times comparison

As a matter of comparison between the standard and aligned methods, Table 2 gives typical execution times of Gysela for four short runs that employ the same configuration and grid size already described in Section 5.3 ($N_r = 256$, $N_\theta = 256$, $N_{v_\parallel} = 48$). For the aligned scheme we take $N_\varphi = 32$, while for the standard scheme we consider three different simulations with $N_\varphi \in \{32, 64, 128\}$. The time breakdown of specific regions of the code are shown in addition to the total run time.

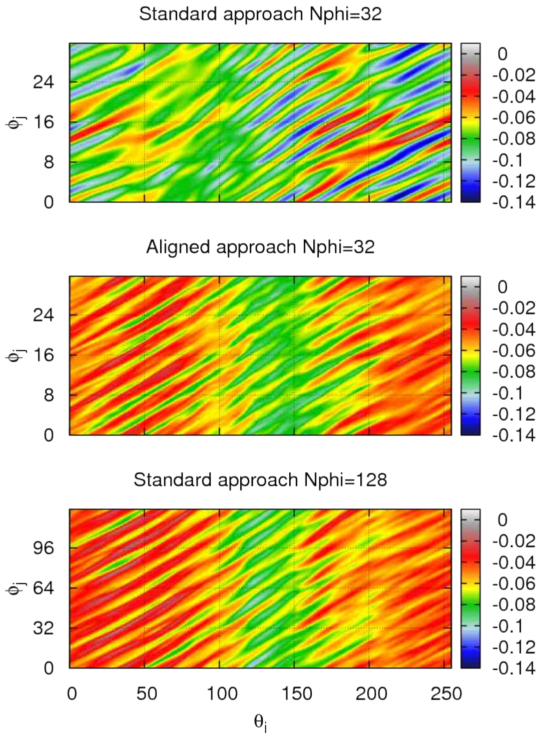

Figure 9: Cross-section of the electric potential at $r = 0.5$ and $t = 1984$. Standard simulation with $N_\varphi = 32$ (top), Aligned simulation with $N_\varphi = 32$ (middle), Standard simulation with $N_\varphi = 128$ (bottom).

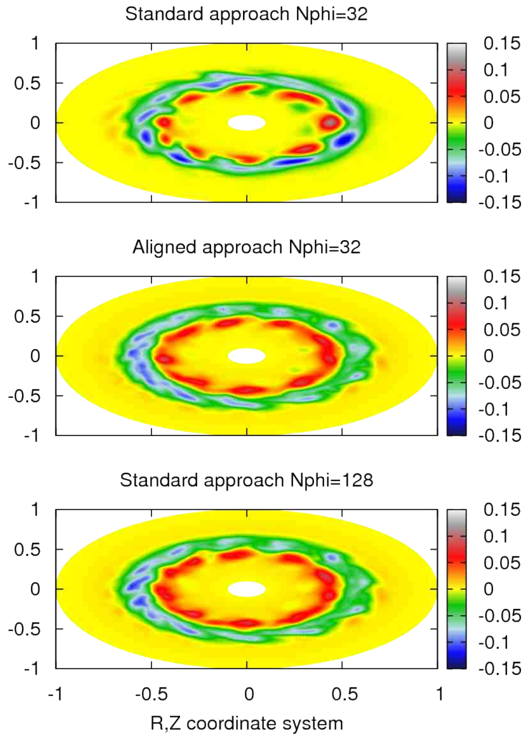

Figure 10: Poloidal cross-section of the electric potential at $\varphi = 0$ and $t = 1984$. Standard simulation with $N_\varphi = 32$ (top), Aligned simulation with $N_\varphi = 32$ (middle), Standard simulation with $N_\varphi = 128$ (bottom).

Execution Time	Aligned $N_\varphi = 32$	Standard $N_\varphi = 32$	Standard $N_\varphi = 64$	Standard $N_\varphi = 128$
transposes	40.0	9.3	28.0	68.6
advections	64.9	48.9	75.7	139.0
others	29.0	26.1	38.6	65.7
total run time	133.9	84.2	142.3	273.3

Table 2: Time (in s.) of a short Gysela run in the same configuration described in section 5.3.

Let us compare the timings for the aligned and standard methods at $N_\varphi = 32$. Firstly we observe that the execution times for the transposes is much higher with the aligned scheme, mainly because there are four transpose steps required (Algorithm 4) instead of two (Algorithm 3). The advection steps are also slightly more expensive with the aligned scheme, because the 1D advection along φ is replaced by the 2D advection aligned in (θ, φ). In fact, the 2D field-aligned interpolation of Algorithm 1 requires additional computations compared to simple 1D interpolations. The improvements and optimizations addressed at the end of Section 5.2 can contribute to decrease these overheads in the future.

Nevertheless, one can see that the aligned strategy with $N_\varphi = 32$ is already competitive against the standard approach with $N_\varphi = 64$ in terms of total run time, with the big benefit of requiring two times less memory to store the distribution function. Since Section 5.3 has shown that the aligned approach with $N_\varphi = 32$ is more accurate than the standard approach with $N_\varphi = 64$ (at least in the linear phase), we can conclude that there is a clear gain in using field-aligned interpolation in Gysela.

6 Conclusions

We have described a semi-Lagrangian method based on field-aligned interpolation, for the solution of the gyrokinetic Vlasov equation. The application of interest is the numerical simulation of magnetically confined plasmas in fusion devices. Thanks to the smooth variation of the solution in the direction of the magnetic field, field-alignment enhances the accuracy of the interpolation: for a given level of accuracy, this allows us to reduce the number of discretization points along the toroidal direction.

In the simplified setting of 2D constant advection, we have given a rigorous proof of convergence, as well as extensive error estimates which underline the advantages of field-aligned interpolation. We have implemented the scheme into two semi-Lagrangian codes, Selalib and GYSELA, for the solution of the 4D gyrokinetic Vlasov equation in the zero-Larmor-radius limit. We have used the ion temperature gradient (ITG) instability as a standard verification test-case in cylindrical (screw-pinch) and toroidal (circular Tokamak) geometries. In our benchmarks against the standard (not aligned) interpolation scheme, we have observed large reductions in memory footprint (up to a factor of 4), as well as moderate (but improvable) simulation speed-ups. Our estimates suggest that these gains will be even larger in reactor-scale simulations.

Field-aligned interpolation does not pose constraints on the 2D poloidal grids, and the use of magnetic flux coordinates is not necessary. Accordingly, the magnetic axis, as well as the X-point in a divertor configuration, do not pose theoretical problems. Therefore our semi-Lagrangian algorithms can be extended to more complex magnetic geometries, enabling the global simulation of diverted Tokamaks and Stellarators.

7 Appendix

Appendix A: Derivation of the model of Section 4

We consider the gyrokinetic equation (1.1) in the electrostatic case, and we set $m = q = 1$. The modified magnetic field then reads

$$\mathbf{B}^* = \mathbf{B} + v_\parallel \nabla \times \mathbf{b}, \qquad B^*_\parallel = \mathbf{B}^* \cdot \mathbf{b} = B + v_\parallel \nabla \times \mathbf{b} \cdot \mathbf{b}.$$

We further assume that we are in the zero-Larmor-radius limit and we choose $\mu = 0$ for simplicity. The gyro-center Hamiltonian then reads

$$H(t, \mathbf{x}, v_\parallel) = \frac{v_\parallel^2}{2} + \phi(t, \mathbf{x})$$

and the characteristic equations reduce to

$$\begin{aligned} B^*_\parallel \frac{d\mathbf{X}}{dt} &= v_\parallel \mathbf{B}^* + \mathbf{b} \times \nabla\phi, \\ B^*_\parallel \frac{dV_\parallel}{dt} &= -\mathbf{B}^* \cdot \nabla\phi. \end{aligned}$$

In the screw-pinch model of Section 4 we use the cylindrically symmetric magnetic equilibrium (1.6), where

$$\mathbf{B} = B\mathbf{b}, \quad \mathbf{b} = b_z \hat{\mathbf{z}} + b_\theta \hat{\boldsymbol{\theta}},$$

with

$$B = B_0\sqrt{1+\zeta^2}, \quad b_\theta = \frac{\zeta}{\sqrt{1+\zeta^2}}, \quad b_z = \frac{1}{\sqrt{1+\zeta^2}}, \quad \zeta = \frac{\iota r}{R_0}.$$

Here the rotational transform iota only depend on the radius, that is $\iota = \iota(r)$.
We now proceed with projecting the characteristic equations onto the non-orthogonal basis $(\hat{\mathbf{r}}, \hat{\boldsymbol{\theta}}, \mathbf{b})$; in this process we make the dependence of each component on ζ explicit. We recall that the curl of a vector $\mathbf{A} = A_\theta(r)\hat{\boldsymbol{\theta}} + A_z(r)\hat{\mathbf{z}}$ reads $\nabla \times \mathbf{A} = -A_z'(r)\hat{\boldsymbol{\theta}} + \frac{1}{r}(rA_\theta(r))'\hat{\mathbf{z}}$, therefore

$$\nabla \times \mathbf{b} = \frac{\zeta\zeta'}{(1+\zeta^2)^{3/2}}\hat{\boldsymbol{\theta}} + \frac{1}{r}\left(\frac{(\zeta r)'}{(1+\zeta^2)^{1/2}} - \frac{\zeta^2 r\zeta'}{(1+\zeta^2)^{3/2}}\right)\hat{\mathbf{z}},$$

which leads to

$$\nabla \times \mathbf{b} \cdot \mathbf{b} = \frac{\zeta^2\zeta'}{(1+\zeta^2)^2} + \frac{1}{r}\frac{(\zeta r)'}{(1+\zeta^2)} - \frac{\zeta^2\zeta'}{(1+\zeta^2)^2} = \frac{1}{r}\frac{(\zeta r)'}{(1+\zeta^2)} = \frac{\zeta' + \zeta/r}{1+\zeta^2}.$$

From this it follows that

$$B_\parallel^* = B_0\sqrt{1+\zeta^2} + \frac{\zeta' + \zeta/r}{1+\zeta^2} v_\parallel.$$

We then write

$$\begin{aligned}
\nabla\times\mathbf{b} = (\nabla\times\mathbf{b})_\theta\hat{\boldsymbol{\theta}} + (\nabla\times\mathbf{b})_z\hat{\mathbf{z}} &= \left((\nabla\times\mathbf{b})_\theta - \frac{b_\theta}{b_z}(\nabla\times\mathbf{b})_z\right)\hat{\boldsymbol{\theta}} + \frac{(\nabla\times\mathbf{b})_z}{b_z}\mathbf{b} \\
&= \left(\frac{\zeta\zeta'}{(1+\zeta^2)^{1/2}} - \frac{\zeta}{r}\frac{(\zeta r)'}{(1+\zeta^2)^{1/2}}\right)\hat{\boldsymbol{\theta}} + \frac{1}{r}\left((\zeta r)' - \frac{\zeta^2 r\zeta'}{1+\zeta^2}\right)\mathbf{b} \\
= \frac{-\zeta^2}{r(1+\zeta^2)^{1/2}}\hat{\boldsymbol{\theta}} + \left(\zeta'\frac{1-\zeta^2}{1+\zeta^2} + \frac{\zeta}{r}\right)\mathbf{b} &= \frac{-\zeta^2}{r(1+\zeta^2)^{1/2}}\hat{\boldsymbol{\theta}} + \left(\frac{\zeta'+\zeta/r}{1+\zeta^2} + \frac{\zeta^2}{1+\zeta^2}(\zeta/r - \zeta')\right)\mathbf{b},
\end{aligned}$$

so that

$$\mathbf{B}^* = B_\parallel^*\mathbf{b} - \frac{\zeta^2 v_\parallel}{r(1+\zeta^2)^{1/2}}\hat{\boldsymbol{\theta}} + \frac{\zeta^2 v_\parallel}{1+\zeta^2}(\zeta/r - \zeta')\mathbf{b}.$$

We have

$$\nabla\phi\cdot\mathbf{b} = \frac{1}{r}\partial_\theta\phi b_\theta + \partial_z\phi b_z,$$

so that

$$\partial_z\phi = \frac{1}{b_z}\nabla\phi\cdot\mathbf{b} - \frac{1}{r}\partial_\theta\phi\frac{b_\theta}{b_z}.$$

Now, we have

$$\begin{aligned}
\mathbf{b}\times\nabla\phi &= (b_\theta\partial_z\phi - b_z\frac{\partial_\theta\phi}{r})\hat{\mathbf{r}} + b_z\partial_r\phi\hat{\boldsymbol{\theta}} - b_\theta\partial_r\phi\hat{\mathbf{z}} \\
&= (\frac{b_\theta}{b_z}\nabla\phi\cdot\mathbf{b} - \frac{\partial_\theta\phi}{b_z r})\hat{\mathbf{r}} + \frac{\partial_r\phi}{b_z}\hat{\boldsymbol{\theta}} - \frac{b_\theta}{b_z}\partial_r\phi\mathbf{b}
\end{aligned}$$

and

$$\mathbf{B}^*\cdot\nabla\phi = \left(B_\parallel^* + \frac{\zeta^2 v_\parallel}{1+\zeta^2}(\zeta/r - \zeta')\right)\nabla\phi\cdot\mathbf{b} - \frac{\zeta^2 v_\parallel}{r^2(1+\zeta^2)^{1/2}}\partial_\theta\phi.$$

Finally we can write the characteristic equations in the $(\hat{\mathbf{r}}, \hat{\boldsymbol{\theta}}, \mathbf{b})$ basis, as

$$\begin{aligned}
B_\parallel^*\frac{d\mathbf{X}}{dt} &= \left(-(1+\zeta^2)^{1/2}\frac{\partial_\theta\phi}{r} + \zeta\mathbf{b}\cdot\nabla\phi\right)\hat{\mathbf{r}} + \left((1+\zeta^2)^{1/2}\partial_r\phi - \frac{\zeta^2 v_\parallel^2}{r(1+\zeta^2)^{1/2}}\right)\hat{\boldsymbol{\theta}} \\
&\quad + \left(B_\parallel^* v_\parallel + \frac{\zeta^2 v_\parallel^2}{1+\zeta^2}(\zeta/r - \zeta') - \zeta\partial_r\phi\right)\mathbf{b}, \\
B_\parallel^*\frac{dV_\parallel}{dt} &= -\left(B_\parallel^* + \frac{\zeta^2 v_\parallel}{1+\zeta^2}(\zeta/r - \zeta')\right)\nabla\phi\cdot\mathbf{b} + \frac{\zeta^2 v_\parallel}{r^2(1+\zeta^2)^{1/2}}\partial_\theta\phi,
\end{aligned}$$

where $B_\parallel^* = B_0\sqrt{1+\zeta^2} + v_\parallel(\zeta' + \zeta/r)/(1+\zeta^2)$. If we now let $\zeta \to 0$ and $\zeta' \to 0$ while keeping our basis unchanged, the equations above reduce to

$$\begin{aligned}
\frac{d\mathbf{X}}{dt} &= -\frac{\partial_\theta\phi}{rB_0}\hat{\mathbf{r}} + \frac{\partial_r\phi}{B_0}\hat{\boldsymbol{\theta}} + v_\parallel\mathbf{b}, \\
\frac{dV_\parallel}{dt} &= -\mathbf{b}\cdot\nabla\phi,
\end{aligned}$$

which correspond to (4.1). We notice that under this approximation we have let $B_\parallel^* \to B_0$. Thanks to the fact that the magnetic field (1.6) has the property $\nabla\cdot\mathbf{b} = 0$, the resulting phase-space flow is still divergence-free, as

$$\nabla\cdot\mathbf{u} = \frac{1}{r}\frac{\partial}{\partial r}\left(-\frac{\partial_\theta\phi}{B_0}\right) + \frac{1}{r}\frac{\partial}{\partial\theta}\left(\frac{\partial_r\phi}{B_0}\right) + v_\parallel\nabla\cdot\mathbf{b} = 0, \qquad \frac{\partial a_\parallel}{\partial v_\parallel} = \frac{\partial}{\partial v_\parallel}\left(-\mathbf{b}\cdot\nabla\phi\right) = 0.$$

Therefore the reduced model (4.1) conserves mass, defined as the phase-space integral of $B_0 f$.

Appendix B: Dispersion equation

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

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

We obtain

$$\partial_t f_1 - \frac{\partial_\theta \phi_1}{rB_0} \partial_r f_0 + v b_z \partial_z f_1 + v \frac{b_\theta}{r} \partial_\theta f_1 - \left(b_\theta \frac{\partial_\theta \phi_1}{r} + b_z \partial_z \phi_1 \right) \partial_v f_0 = \mathcal{O}(\varepsilon).$$

and

$$-\left(\partial_r^2 \phi_1 + \left(\frac{1}{r} + \frac{\partial_r n_0}{n_0} \right) \partial_r \phi_1 + \frac{1}{r^2} \partial_\theta^2 \phi_1 \right) + \frac{1}{T_e} \phi_1 = \frac{1}{n_0} \int f_1 dv + \mathcal{O}(\varepsilon).$$

We assume that the solutions have the form :

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

with $k = \frac{n}{R}$. Then, we obtain

$$(-\omega + kvb_z + v\frac{mb_\theta}{r}) f_{m,n,\omega} = \left(\frac{m}{rB_0} \partial_r f_0 + \left(b_\theta \frac{m}{r} + b_z k \right) \partial_v f_0 \right) \phi_{m,n,\omega}$$

and

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

We get, as $k_\| = \left(b_\theta \frac{m}{r} + b_z k \right)$

$$\begin{aligned} -\left(\partial_r^2 \phi_{m,n,\omega} + \left(\frac{1}{r} + \frac{\partial_r n_0}{n_0} \right) \partial_r \phi_{m,n,\omega} - \frac{m^2}{r^2} \phi_{m,n,\omega} \right) + \frac{1}{T_e} \phi_{m,n,\omega} \\ = \frac{1}{n_0} \phi_{m,n,\omega} \int \frac{\frac{m}{rB_0} \partial_r f_0 + k_\| \partial_v f_0}{v k_\| - \omega} dv \end{aligned}$$

By using the expression of f_0, we have

$$I = \int \frac{-\frac{v}{T_i} + \frac{m}{k_\| r B_0} \left(\frac{\partial_r n_0}{n_0} - \frac{\partial_r T_i}{2T_i} + \frac{v^2 \partial_r T_i}{2T_i^2} \right)}{v - \frac{\omega}{k_\|}} f_0 dv.$$

We introduce for $\ell \in \mathbb{N}$:

$$I_\ell(u) = \frac{1}{n_0} \int v^\ell \frac{f_0}{v - u} f_0 dv,$$

so that

$$\frac{I}{n_0} = -\frac{1}{T_i} I_1 \left(\frac{\omega}{k_\|} \right) + \frac{m}{k_\| r B_0} \left[\left(\frac{\partial_r n_0}{n_0} - \frac{\partial_r T_i}{2T_i} \right) I_0 \left(\frac{\omega}{k_\|} \right) + \frac{\partial_r T_i}{2T_i^2} I_2 \left(\frac{\omega}{k_\|} \right) \right].$$

We use the relations :

$$I_0 = \frac{1}{(2T_i)^{1/2}} Z \left(\frac{u}{(2T_i)^{1/2}} \right), \qquad I_1 = 1 + uI_0, \qquad I_2 = u\left(1 + uI_0\right),$$

with

$$Z(u) = \frac{1}{\sqrt{\pi}} \int \frac{\exp(-x^2)}{x-u} dx = i\sqrt{\pi} \exp(-u^2)(1 - \operatorname{erf}(-iu)),$$

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

The dispersion relation is, putting $\phi = \phi_{m,n,\omega}$ for convenience,

$$\begin{aligned} A &= -\partial_r^2 \phi - \left(\frac{1}{r} + \frac{\partial_r n_0}{n_0} \right) \partial_r \phi + \frac{m^2}{r^2} \phi + \frac{1}{T_e} \phi \\ &= \left[-\frac{1}{T_i}(1 + zZ(z)) + \frac{m}{k^* r B_0} \left(Z(z) \left(\frac{\partial_r n_0}{n_0} - \frac{\partial_r T_i}{2T_i} \right) + z(1 + zZ(z)) \frac{\partial_r T_i}{T_i} \right) \right] \phi, \end{aligned}$$

with $k^* = (2T_i)^{1/2} k_\|$, and $z = \frac{\omega}{k^*}$, and recalling that $k_\| = \left(b_\theta \frac{m}{r} + b_z k \right)$. Note that the dispersion relation depends on m and $k_\|$ and not directly on n. This means that taking different values of ι and n but with same m and $k_\|$ will lead to the same dispersion relation.

References

[1] F. Hariri and M. Ottaviani. A flux-coordinate independent field-aligned approach to plasma turbulence simulations. *Computer Physics Communications*, 184(11):2419–2429, 2013. http://dx.doi.org/10.1016/j.cpc.2013.06.005.

[2] Andreas Stegmeir, David Coster, Omar Maj, Klaus Hallatschek, and Karl Lackner. The field line map approach for simulations of magnetically confined plasmas. *Comput. Phys. Commun.*, 198:139–153, 2015.

[3] J.-M. Kwon, D. Yi, X. Piao, and P. Kim. Development of semi-Lagrangian gyrokinetic code for full-f turbulence simulation in general tokamak geometry. *Journal of Computational Physics*, 283:518–540, 2015. http://dx.doi.org/10.1016/j.jcp.2014.12.017.

[4] X. Garbet, Y. Idomura, L. Villard, and T.H. Watanabe. Gyrokinetic simulations of turbulent transport. *Nuclear Fusion*, 50(4):043002, 2010. http://stacks.iop.org/0029-5515/50/i=4/a=043002.

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

[6] Virginie Grandgirard, Jérémie Abiteboul, Julien Bigot, Thomas Cartier-Michaud, Nicolas Crouseilles, Charles Erhlacher, Damien Esteve, Guilhem Dif-Pradalier, Xavier Garbet, Philippe Ghendrih, Guillaume Latu, Michel Mehrenberger, Claudia Norscini, Chantal Passeron, Fabien Rozar, Yanick Sarazin, Antoine Strugarek, Eric Sonnendrücker, and David Zarzoso. A 5D gyrokinetic full-f global semi-Lagrangian code for flux-driven ion turbulence simulations. https://hal-cea.archives-ouvertes.fr/cea-01153011, July 2015.

[7] A. Bottino and E. Sonnendrücker. Monte Carlo Particle-In-Cell methods for the simulation of the Vlasov–Maxwell gyrokinetic equations. *Journal of Plasma Physics*, 81(05):435810501, 2015.

[8] Nicolas Besse and Michel Mehrenberger. Convergence of classes of high order semi-Lagrangian schemes for the Vlasov equation. *Mathematics of Computation*, 77:93–123, 2008.

[9] R. Ferretti. Equivalence of semi-Lagrangian and Lagrange-Galerkin schemes under constant advection speed. *J. Comput. Math.*, 28(4):461–473, 2010.

[10] R. Ferretti. On the relationship between semi-Lagrangian and Lagrange-Galerkin schemes. *Numer. Math.*, 124(1):31–56, 2013.

[11] F. Boyer. Lecture notes (in French): Aspects théoriques et numériques de l'équation de transport, Université de Aix-Marseille, June 2014. `http://www.math.univ-toulouse.fr/~fboyer/_media/enseignements/cours_transport_fboyer.pdf`.

[12] G. Pólya. Remarks on characteristic functions. In *Proc. [First] Berkeley Symp. on Math. Statist. and Prob.*, pages 115–123. University of California Press, 1949. `http://projecteuclid.org/euclid.bsmsp/1166219202`.

[13] E.O. Tuck. On positivity of Fourier transforms. *Bull. Austral. Math. Soc.*, 74:133–138, 2006.

[14] J. Steward. Positive definite functions and generalizations, an historical survey. *Rocky Mountain J. Math.*, 6(3):409–434, 1976.

[15] Frédérique Charles, Bruno Després, and Michel Mehrenberger. Enhanced convergence estimates for semi-Lagrangian schemes Application to the Vlasov-Poisson equation. *SIAM Journal of Numerical Analysis*, 51:840–863, 2013.

[16] A. Schönhage. Fehlerfortpflanzung bei interpolation. *Numerische Mathematik*, 3:62–71, 1961. `http:dx.doi.org/10.1007/BF01386001`.

[17] T. M. Mills and Simon J. Smith. On the lebesgue function for lagrange interpolation with equidistant nodes. *Journal of the Australian Mathematical Society (Series A)*, 52(1):111–118, 2 1992.

[18] G. N. Watson. The constants of landau and lebesgue. *The Quarterly Journal of Mathematics*, os-1(1):310–318, 1930.

[19] D. Cvijović and H.M. Srivastava. Asymptotics of the landau constants and their relationship with hypergeometric functions. *Taiwanese Journal of Mathematics*, 13(3):855–870, 2009.

[20] E. Landau. Abschätzung der koeffizientensumme einer potenzreihe. *Archiv der Math. und Phys.*, 3(21):42–50 & 250–255, 1913.

[21] Emil C. Popa and Nicolae-Adrian Secelean. Estimates for the constants of landau and lebesgue via some inequalities for the wallis ratio. *Journal of Computational and Applied Mathematics*, 269:68–74, 2014.

[22] SELALIB. `http://selalib.gforge.inria.fr`, 2014.

[23] Nicolas Crouseilles, Pierre Glanc, SeverA. Hirstoaga, Eric Madaule, Michel Mehrenberger, and Jérôme Pétri. A new fully two-dimensional conservative semi-Lagrangian method: applications on polar grids, from diocotron instability to ITG turbulence. *The European Physical Journal D*, 68(9), 2014. `http://dx.doi.org/10.1140/epjd/e2014-50180-9`.

[24] David Coulette and Nicolas Besse. Numerical comparisons of gyrokinetic multi-water-bag models. *J. Comput. Physics*, 248:1–32, 2013. `http://dx.doi.org/10.1016/j.jcp.2013.03.065`.

[25] Christophe Steiner, Michel Mehrenberger, Nicolas Crouseilles, Virginie Grandgirard, Guillaume Latu, and Fabien Rozar. Gyroaverage operator for a polar mesh. *The European Physical Journal D*, 69(18), 2015. `http://dx.doi.org/10.1140/epjd/e2014-50211-7`.

[26] E. Sanchez, R. Kleiber, R. Hatzky, Alejandro Soba, Xavier Sáez, F. Castejon, and José Ma. Cela. Linear and nonlinear simulations using the EUTERPE gyrokinetic code. *IEEE Transactions on Plasma Science*, 38(9):2119–2128, Sept 2010.

[27] Burton D. Fried and Samuel D. Conte. *The Plasma Dispersion Function: The Hilbert Transform of the Gaussian.* Academic Press, 1961.

[28] G. Latu, N. Crouseilles, V. Grandgirard, and E. Sonnendrücker. Gyrokinetic semi-Lagrangian parallel simulation using a hybrid OpenMP/MPI programming. In *Recent Advances in PVM and MPI*, volume 4757 of *Lecture Notes in Computer Science*, pages 356–364. Springer, 2007.

[29] G. Latu, V. Grandgirard, N. Crouseilles, and G. Dif-Pradalier. Scalable quasineutral solver for gyrokinetic simulation. In *PPAM (2), LNCS 7204*, pages 221–231, 2011. `http://dx.doi.org/10.1007/978-3-642-31500-8_23`.

[30] J. Bigot, V. Grandgirard, G. Latu, C. Passeron, F. Rozar, and O. Thomine. Scaling GYSELA code beyond 32K-cores on Blue Gene/Q. In *ESAIM: PROCEEDINGS*, volume CEMRACS 2012 of *43*, pages 117–135, Luminy, France, 2013.

[31] R. Aymar, P. Barabaschi, and Y. Shimomura. The ITER design. *Plasma Phys. Control. Fusion*, 44:519–565, 2002.