SMAI-JCM

SMAI Journal of Computational Mathematics

Parallel kinetic scheme for transport equations in complex toroidal geometry

Matthieu Boileau, Bérenger Bramas, Emmanuel Franck, Romane Hélie, Philippe Helluy & Laurent Navoret

Volume 8 (2022), p. 249-271.

https://doi.org/10.5802/smai-jcm.86

The SMAI Journal of Computational Mathematics is a member of the Centre Mersenne for Open Scientific Publishing
http://www.centre-mersenne.org/
Submissions at https://smai-jcm.centre-mersenne.org/ojs/submission
e-ISSN: 2426-8399

SMAI Journal of Computational Mathematics
Vol. 8, 249-271 (2022)

Parallel kinetic scheme for transport equations in complex toroidal geometry

Matthieu Boileau [1]
Bérenger Bramas [2]
Emmanuel Franck [3]
Romane Hélie [4]
Philippe Helluy [5]
Laurent Navoret [6]

[1] Institut de Recherche Mathématique Avancée, UMR 7501, Université de Strasbourg et CNRS, 7 rue René Descartes, 67000 Strasbourg, France & INRIA Nancy-Grand Est, TONUS Project, Strasbourg, France
E-mail address: matthieu.boileau@math.unistra.fr
[2] INRIA Nancy-Grand Est, CAMUS Project, Strasbourg, France
E-mail address: berenger.bramas@inria.fr
[3] Institut de Recherche Mathématique Avancée, UMR 7501, Université de Strasbourg et CNRS, 7 rue René Descartes, 67000 Strasbourg, France & INRIA Nancy-Grand Est, TONUS Project, Strasbourg, France
E-mail address: emmanuel.franck@inria.fr
[4] Institut de Recherche Mathématique Avancée, UMR 7501, Université de Strasbourg et CNRS, 7 rue René Descartes, 67000 Strasbourg, France & INRIA Nancy-Grand Est, TONUS Project, Strasbourg, France
E-mail address: romane.helie@math.unistra.fr
[5] Institut de Recherche Mathématique Avancée, UMR 7501, Université de Strasbourg et CNRS, 7 rue René Descartes, 67000 Strasbourg, France & INRIA Nancy-Grand Est, TONUS Project, Strasbourg, France
E-mail address: philippe.helluy@math.unistra.fr
[6] Institut de Recherche Mathématique Avancée, UMR 7501, Université de Strasbourg et CNRS, 7 rue René Descartes, 67000 Strasbourg, France & INRIA Nancy-Grand Est, TONUS Project, Strasbourg, France
E-mail address: laurent.navoret@math.unistra.fr.

Abstract. We present an efficient solver for the conservative transport equation with variable coefficients in complex toroidal geometries. The solver is based on a kinetic formulation resembling the Lattice-Boltzmann approach. The chosen formalism allows obtaining an explicit and conservative scheme that requires no matrix inversion and whose CFL stability condition is independent from the poloidal dynamics. We present the method and its optimized parallel implementation on toroidal geometries. Two and three dimensional plasma physics test cases are carried out.

2020 Mathematics Subject Classification. 35L65, 65M12, 35Q83, 82D10.

Keywords. plasma physics, kinetic scheme, discontinuous Galerkin.

1. Introduction

Physics of tokamak raise a lot of computational challenges. Indeed, a tokamak is an experimental device designed for creating hot charged particle gas: in such a device, the charged particles are confined thanks to a large magnetic field. Their dynamics involve several multi-scale transport phenomena, which introduces stringent constraints on the discretization parameters and requires high-order

https://doi.org/10.5802/smai-jcm.86

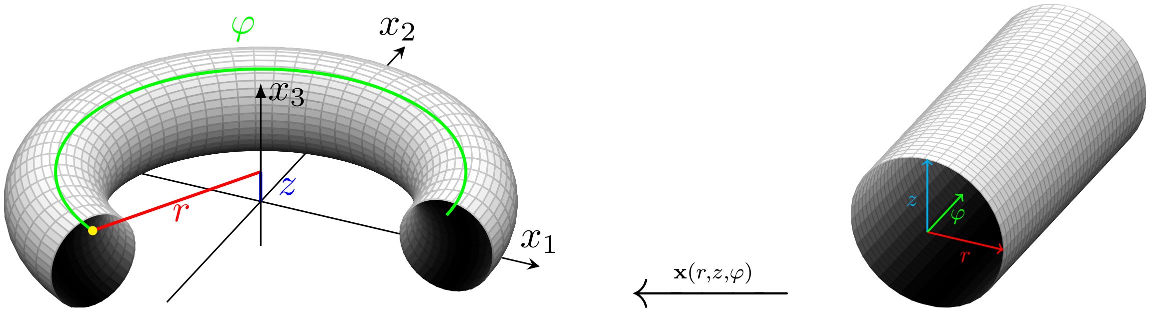

Figure 1.1. Cartesian coordinates (x_1, x_2, x_3) and cylindrical coordinates (r, z, φ).

schemes. Moreover, tokamak generally has a toroidal shape and presents a cylindrical symmetry around the vertical axis. The geometry of the poloidal plane, however, can be complicated and unstructured meshes are required. We thus aim at proposing an efficient numerical scheme for solving conservative transport equations in such three-dimensional toroidal geometries.

In tokamak, the large toroidal magnetic field results in a scale separation between toroidal and poloidal dynamics. Indeed, particles exhibit fast transport in the toroidal direction, while the poloidal dynamics follow incompressible dynamics. Explicit numerical scheme would impose the dependence of the time step to the mesh through the Courant–Friedrichs–Lévy (CFL) stability condition. Thus, to avoid too small time steps, the mesh should be perfectly adapted to the very complex three-dimensional dynamics. This is complicated in practice because the creation of the mesh is often an independent step of the simulation.

To dissociate the issues of numerical parameters from the mesh construction, CFL-less numerical methods have been proposed. For transport equations, one of the main such methods is the semi-Lagrangian one, introduced in [24]. Each iteration consists in computing the foot of the characteristics issued from the mesh nodes and then interpolating the solution at these points. Several variants have been proposed, see for instance [3, 21]. This method has been successfully used for full tokamak simulations in the Gysela code [18]. In these simulations, the computational domain is a torus with circular sections discretized with a polar mesh. Unfortunately, the standard semi-Lagrangian method does not easily handle full unstructured meshes, since stability and conservation issues arise. It is therefore difficult to extend the method to more general geometries.

Indeed, the shapes of the tokamak lead naturally to specific meshes. The domain being axisymmetric, the mesh can be structured in the toroidal direction. However, the poloidal sections of tokamak can be either circular like in the Tore Supra device [25] but can also have very complex geometry. The poloidal plane of the tokamak ITER has a so-called D-shape. This geometry is in fact chosen to optimize the confinement. For instance, the poloidal mesh can be constructed as a multi-patch Bézier mesh [19].

The numerical methods must naturally be adapted to such meshes both in their structure and parallelism. After rewriting the conservative transport equation in the cylindrical coordinates (r, z, φ) (see Figure 1.1), very simple numerical methods can be envisaged in the φ direction (e.g. Fourier or semi-Lagrangian method). But more sophisticated solvers have to be considered to deal with the transport in unstructured meshes in (r, z) planes.

In this work, we propose to adapt the CFL-less kinetic solver proposed in [7] to the toroidal geometry. The method reuses ideas coming from the Lattice-Boltzmann Method (LBM) [6, 16, 23, 26] or from the kinetic schemes [1, 4]. LBM have been introduced to solve the incompressible Navier–Stokes equation as well as advection-diffusion equations while kinetic schemes were introduced to solve hyperbolic systems. Note that generalized LBM have also been proposed for kinetic equations [11, 12]. The main point is to replace the conservative transport equation, where the velocity field is not constant, by a *few transport equations at constant velocities*, coupled by a stiff local source term. The coupled system is then solved with a splitting algorithm that separates the free transport steps and the stiff source terms. The free transport steps are easier to solve, because the transport step is done at constant velocity.

Here, we consider six constant velocities aligned to the Cartesian axes in the cylindrical coordinates, whose magnitudes may be different in the poloidal and toroidal directions: this is a so-called D3Q6 model. The magnitudes have to satisfy a so-called sub-characteristic condition to ensure the stability of the numerical scheme. We would like to emphasize that this condition does not involve the time and space discretization parameters. Taking different magnitudes enables us to handle a different scale between transport in poloidal and toroidal directions. Other sets of velocities have been proposed to better capture multi-scale effects [8, 15] but stability is then more delicate to study. We also refer to [14] for recent analysis of such schemes.

In order to make the time step independent from the possibly complicated poloidal mesh, the transport equations at constant velocities in poloidal planes are solved using a CFL-less implicit DG method [7]. The implicit method has actually an explicit cost because the transport velocities are constant: the transport scheme is just to invert a block-triangular linear system. For the transport in the toroidal directions, since the mesh is uniform, an exact transport solve is chosen like in Lattice-Boltzmann methods. These choices link the time step to the toroidal space discretization but still keep it independent from the poloidal mesh.

The whole method is conservative. Moreover, it is possible, using an adequate splitting algorithm with a so-called over-relaxation techniques [10] to achieve second-order accuracy in time. For Lattice Boltzmann schemes, this has been analyzed in [13]. High order spatial discretization is considered in the poloidal planes thanks to the DG method, while transport is exact in the toroidal direction. Thus, we finally obtain a second-order scheme well adapted to manage different dynamics in the poloidal and toroidal directions.

The resulting scheme has also nice parallelization possibilities. In a poloidal plane, the block-triangular linear systems resulting from the DG scheme that are well solved by an optimized task-based implementation [2, 5]. In the toroidal direction, the transport equations are solved by a simple shift operator. It is here implemented by simple MPI point-to-point communications.

In the following, we first present the whole mathematical, numerical and programming construction. We then verify its accuracy and efficiency with some pure transport test cases. Finally, in order to assess the usefulness of the method in more complex framework, we apply our transport solver to the numerical simulation of the diocotron instability in two-dimensional and three-dimensional configurations. This plasma physics test cases require to couple the transport solver with a Poisson solver and make the whole model non-linear.

2. Transport equation and reformulation

We consider the following conservative transport equation:

$$\partial_t \sigma + \nabla_{\mathbf{x}} \cdot (\sigma \mathbf{v}) = 0, \tag{2.1}$$

where the unknown $\sigma(\mathbf{x}, t)$ depends on a space variable $\mathbf{x} \in \mathbb{R}^3$ and of the time t. The velocity field $\mathbf{v}(\mathbf{x}, t)$ is given. As we are interested in solving this equation in a toroidal domain, we first rewrite an equivalent formulation in cylindrical coordinates. We then present its approximation by a system of transport equation at constant velocities using the Lattice-Boltzmann Method.

2.1. **In cylindrical coordinates**

Let us first consider an arbitrary change of coordinates $\mathbf{x} = \mathbf{x}(\mathbf{r})$. The Jacobian of this change of variables is denoted by $j(\mathbf{r}) = \det \mathbf{x}'(\mathbf{r})$. In this new set of coordinates, the transport equation (2.1) becomes:

$$\partial_t \rho + \nabla_{\mathbf{r}} \cdot (\rho \mathbf{u}) = 0. \tag{2.2}$$

where the new unknown $\rho(\mathbf{x}, t)$ and the new velocity field $\mathbf{u}(\mathbf{x}, t)$ are defined by:

$$\rho(\mathbf{r}, t) = j(\mathbf{r})\sigma(\mathbf{x}(\mathbf{r}), t), \tag{2.3}$$

$$\mathbf{u}(\mathbf{r}, t) = \mathbf{x}'(\mathbf{r})^{-1}\mathbf{v}(\mathbf{x}(\mathbf{r}), t). \tag{2.4}$$

For tokamak applications, we are particularly interested in the change from Cartesian to cylindrical coordinates. We denote by $\mathbf{x} = (x_1, x_2, x_3)^T$ the Cartesian coordinates and by $\mathbf{r} = (r, z, \varphi)^T$ the cylindrical coordinates. The change of variables is given by

$$\begin{aligned} x_1 &= r\cos\varphi, \\ x_2 &= r\sin\varphi, \\ x_3 &= z. \end{aligned}$$

We also define the cylindrical frame

$$\mathbf{e}_r = \begin{pmatrix} \cos\varphi \\ \sin\varphi \\ 0 \end{pmatrix}, \quad \boldsymbol{e}_z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \boldsymbol{e}_\varphi = r \begin{pmatrix} -\sin\varphi \\ \cos\varphi \\ 0 \end{pmatrix}.$$

which is the columns of $\mathbf{x}'(\mathbf{r})$. Note that the vector $\mathbf{e}_\varphi$ is not a unit vector: its Euclidean norm is equal to r. From (2.3), we define the new unknown:

$$\rho(\mathbf{r}, t) = r\sigma(\mathbf{x}, t).$$

Then we introduce the velocity fields $\mathbf{u} = (u_r, u_z, u_\varphi)^T$ as defined by (2.4). It is related to $\mathbf{v}$ by the following relation:

$$\mathbf{v}(\mathbf{x}, t) = \begin{pmatrix} v_1(\mathbf{x}, t) \\ v_2(\mathbf{x}, t) \\ v_3(\mathbf{x}, t) \end{pmatrix} = u_r(\mathbf{r}, t)\,\mathbf{e}_r + u_z(\mathbf{r}, t)\,\mathbf{e}_z + u_\varphi(\mathbf{r}, t)\,\mathbf{e}_\varphi.$$

Up to a change of unknowns, the initial conservative transport equation (2.1) has an equivalent conservative formulation (2.2) in cylindrical coordinates. Therefore, it is possible to solve the transport equation in cylindrical coordinates as we were in Cartesian coordinates. In particular equation (2.2) can be rewritten

$$\partial_t \rho + \boldsymbol{\nabla}_{(r,z)} \cdot \left(\rho \begin{pmatrix} u_r \\ u_z \end{pmatrix} \right) + \partial_\varphi(\rho u_\varphi) = 0, \tag{2.5}$$

and as proposed in this work, we can use a different discretization for the (r, z) and the φ coordinates. Indeed, the geometry will be supposed to be invariant in the φ direction, which will allow several optimizations.

2.2. **Kinetic relaxation method**

For solving the transport equation (2.2), the kinetic relaxation method consists in introducing n_v unknowns $f_k(\mathbf{r}, t)$ associated to n_v constant kinetic velocities in cylindrical coordinates $\boldsymbol{\lambda}_k$, $k = 0 \ldots n_v - 1$. The $n_v - 2$ first velocities are used for solving the transport problem in the (r, z) plane and the last two velocities are used for the φ direction. For instance, we can take $n_v = 6$, with 4 velocities in the poloidal plane

$$\boldsymbol{\lambda}_0 = \begin{pmatrix} \lambda_p \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{\lambda}_1 = \begin{pmatrix} -\lambda_p \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{\lambda}_2 = \begin{pmatrix} 0 \\ \lambda_p \\ 0 \end{pmatrix}, \quad \boldsymbol{\lambda}_3 = \begin{pmatrix} 0 \\ -\lambda_p \\ 0 \end{pmatrix},$$

and 2 velocities in the toroidal direction

$$\boldsymbol{\lambda}_4 = \begin{pmatrix} 0 \\ 0 \\ \lambda_t \end{pmatrix}, \quad \boldsymbol{\lambda}_5 = \begin{pmatrix} 0 \\ 0 \\ -\lambda_t \end{pmatrix},$$

where $\lambda_p > 0$ denotes the norm of the kinetic velocities of the poloidal plane and $\lambda_t > 0$ the norm of the velocities in the toroidal direction. This choice corresponds to the so-called D3Q6 method. We could also consider only 3 velocities in the poloidal planes which would lead to a D2Q3 method in the poloidal plane coupled to a D1Q2 method in the toroidal direction. Then the kinetic relaxation method consists in considering the following kinetic model

$$\partial_t f_k + \boldsymbol{\lambda}_k \cdot \boldsymbol{\nabla}_{\mathbf{r}} f_k = \frac{1}{\tau}(f_k^{eq} - f_k), \tag{2.6}$$

with $\tau > 0$ is the relaxation parameter and f_k^{eq} is the so-called equilibrium kinetic distribution defined by

$$f_k^{eq} = \frac{\rho}{n_v} + \frac{\rho \mathbf{u} \cdot \boldsymbol{\lambda}_k}{2\|\boldsymbol{\lambda}_k\|^2}, \tag{2.7}$$

with $\mathbf{u}$ is the velocity field defined in (2.4) but here ρ is defined by

$$\rho = \sum_{k=0}^{n_v-1} f_k. \tag{2.8}$$

The equilibrium kinetic distribution is chosen such that equation (2.6) is an approximation of equation (2.2). More precisely, in the limit where τ goes to 0, the density ρ defined by (2.8) tends to the solution to (2.2). This approximation is stable under a so-called sub-characteristic condition. Roughly speaking, it states that the ellipsoid, whose main axes are aligned to the kinetic speeds $\boldsymbol{\lambda}_k$, has to contain the velocity field $\mathbf{u}$. More precisely, the model is stable if all the velocities $\mathbf{u} = (u_r, u_z, u_\varphi)^T$ satisfy the following condition

$$\frac{u_r^2}{\lambda_p^2} + \frac{u_z^2}{\lambda_p^2} + \frac{u_\varphi^2}{\lambda_t^2} \leqslant \frac{1}{3}.$$

The computations are given in the appendix A. For more details, we refer to [1, 2, 4, 7, 13].

The main interest to transform (2.5) into (2.6) is that now the transport velocities are *constant.* This will dramatically simplify the resolution of the equations and allows designing explicit schemes that are unconditionally stable.

3. Transport solver

In this section we focus on the numerical schemes used for solving the transport equations involved in the kinetic model (2.6):

$$\partial_t f_k + \boldsymbol{\lambda}_k \cdot \boldsymbol{\nabla}_{\mathbf{r}} f_k = 0, \tag{3.1}$$

We want to solve them in domains with a cylindrical geometry. Indeed, we recall that, up to a change of variable (see section 2.1), it enables us to consider toroidal domains as required for tokamak simulations. The starting point is thus a two-dimensional unstructured mesh of one poloidal plane ($\varphi = \text{Cst}$). Here it is chosen made of second order curved quadrilaterals with eight nodes (“Q8” family in the finite element terminology). Then this mesh is extruded in the φ direction leading to a uniform structured mesh of space step $\Delta\varphi$ in that direction.

Two different transport solvers are used. For transport in the toroidal direction, we consider an exact solver to take advantage of the uniform mesh: this corresponds to the Lattice-Boltzmann method (LBM). For transport in the poloidal planes, we consider an implicit DG scheme. We first details the two numerical schemes and then briefly present their parallelization.

3.1. **Transport in the toroidal direction: LBM**

Only the last two kinetic components, f_{n_v-2} and f_{n_v-1}, are transported in the toroidal direction. We solve these transport equations with a simple shift, because of the structured mesh. This writes:

$$(f_{n_v-2})_j^{n+1} = (f_{n_v-2})_{j-1}^n, \quad (f_{n_v-1})_j^{n+1} = (f_{n_v-1})_{j+1}^n,$$

where j refers to the poloidal planes index. This requires to link the time step to the space step in the φ direction: $\Delta t = \Delta\varphi/\lambda_t$. This is the classical algorithm used in the Lattice-Boltzmann method.

3.2. **Transport in the poloidal direction: implicit DG**

Because the poloidal shape can be arbitrary, we have introduced an unstructured mesh and thus we consider a Discontinuous Galerkin (DG) approximation for transporting the $n_v - 2$ first kinetic components $f_0, \dots, f_{n_v-3}$ whose associated kinetic velocities are poloidal. In order to avoid stability constraints due to small mesh elements, we consider an implicit solver.

3.2.1. *Implicit DG formulation*

The objective is to solve a two-dimensional transport equation, with constant velocity

$$\partial_t f + \boldsymbol{\lambda} \cdot \boldsymbol{\nabla}_{(r,z)} f = 0.$$

where f and $\boldsymbol{\lambda}$ refer to one of the $n_v - 2$ first kinetic component and its associated kinetic velocity. In each cell L, we consider polynomial basis functions ψ_i^L of degree p. For efficiency reason, the basis functions are Lagrange polynomials based on Gauss–Lobatto quadrature points. The transported function f is approximated in cell L by a linear expansion on the basis functions

$$f(r,z,n\Delta t) \simeq f_L^n(r,z) = \sum_j f_{L,j}^n \psi_j^L(r,z), \quad (r,z) \in L.$$

The unknowns are the coefficients $f_{L,j}^n$ of the linear expansion.

In order to avoid constraining CFL conditions, we only envisage implicit solvers. We first consider the simplest first order implicit DG approximation scheme. It reads: $\forall L, \forall i$,

$$\int_L \frac{f_L^n - f_L^{n-1}}{\Delta t} \psi_i^L - \int_L \boldsymbol{\lambda} \cdot \boldsymbol{\nabla} \psi_i^L f_L^n + \int_{\partial L} \Big((\boldsymbol{\lambda} \cdot \boldsymbol{n}_{LR})^+ f_L^n + (\boldsymbol{\lambda} \cdot \boldsymbol{n}_{LR})^- f_R^n \Big) \psi_i^L = 0. \tag{3.2}$$

where the basis functions ψ_i^L play the role of test functions, R denotes the neighbor cells along ∂L, $\mathbf{n}_{LR}$ is the unit normal vector on ∂L oriented from L to R (see Figure 3.1) and $(\boldsymbol{\lambda}\cdot\boldsymbol{n})^+ = \max(\boldsymbol{\lambda}\cdot\boldsymbol{n}, 0)$, $(\boldsymbol{\lambda}\cdot\boldsymbol{n})^- = \min(\boldsymbol{\lambda}\cdot\boldsymbol{n}, 0)$. We thus use an upwind numerical flux. For more details on the DG method, we refer for instance to [2, 20]. We can rewrite (3.2) in the matrix form

$$M_L f_L^n = M_L f_L^{n-1} + \Delta t \left(A_L f_L^n - \sum_{\substack{LR \in \partial L, \\ \boldsymbol{\lambda}\cdot\boldsymbol{n}_{LR} > 0}} B_{LR} f_L^n - \sum_{\substack{LR \in \partial L, \\ \boldsymbol{\lambda}\cdot\boldsymbol{n}_{LR} < 0}} C_{LR} f_R^n \right), \tag{3.3}$$

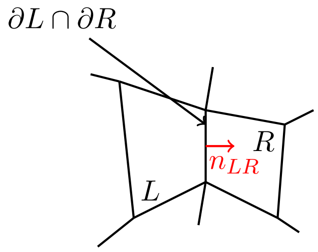

Figure 3.1. Normal vector convention.

where M_L denotes the mass matrix, A_L the volume advection matrix, B_{LR} and C_{LR} the flux advection matrices between cell L and downwind or upwind cells R: cell R is said to be upwind with respect to cell L if $\boldsymbol{\lambda} \cdot \boldsymbol{n}_{LR} < 0$ and downwind otherwise. Their coefficients are given by

$$(M_L)_{i,j} = \int_L \psi_i^L \psi_j^L, \quad (A_L)_{i,j} = \int_L (\boldsymbol{\lambda} \cdot \nabla \psi_i^L)\, \psi_j^L,$$

$$(B_{LR})_{i,j} = \int_{\partial L} (\boldsymbol{\lambda} \cdot \boldsymbol{n}_{LR}) \psi_i^L \psi_j^L \quad \text{and} \quad (C_{LR})_{i,j} = \int_{LR} (\boldsymbol{\lambda} \cdot \boldsymbol{n}_{LR}) \psi_i^L \psi_j^R.$$

In the above formalism, we only describe a first order time scheme. In practice, we actually use a second order implicit Crank–Nicolson time stepping. But it is very similar to the above description.

3.2.2. *Downwind algorithm*

One time-step of the implicit DG scheme consists in computing the distribution function at time t^n, f^n from the distribution function f^{n-1} of the previous time step. From (3.2) it is clear that one has to solve a linear system, as in any implicit scheme. However, because the kinetic velocity $\boldsymbol{\lambda}$ is constant and because we use the upwind numerical flux, the linear system is triangular. It can thus be solved cell by cell, simply sweeping the mesh in the direction of the velocity vector.

More precisely, expression (3.3) shows that the solution f^n depends only on the values of f^n in cell L and in the upwind cells. For a given velocity v we can build a dependency graph. The vertices of the graph are associated to cells and the edges to the cells interfaces or boundaries. We consider two fictitious additional vertices: the "upwind" vertex and the "downwind" vertex. The dependency graph for a simple unstructured mesh and a given constant velocity is represented in Figure 3.2. The construction can be generalized to any unstructured mesh with flat faces. This flatness condition ensures that the kinetic velocity crosses the faces in only one direction. This ensures that the graph does not contain loops. For more details, we refer to [2].

For solving one transport equation for a given constant velocity $\boldsymbol{\lambda}$, the algorithm is then the following:

(1) First we perform a topological ordering of the dependency graph.

(2) First time-step: Assembly, LU decomposition and storage of the local cell matrices. These computations can also be redone during each time-step for saving memory (but it is more CPU demanding).

(3) For each cell (in topological order):
   (a) Compute volume terms.
   (b) Compute upwind fluxes.
   (c) Solve the local linear system.
   (d) Extract the results to the downwind cells.

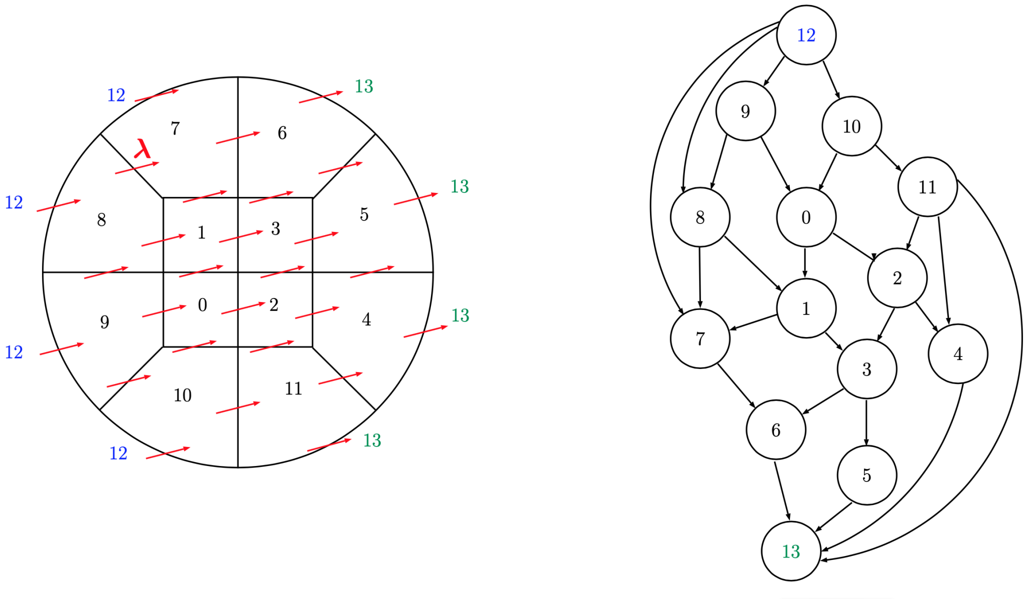

Figure 3.2. Left: Unstructured mesh of the disk, with a constant velocity field $\boldsymbol{\lambda}$ (in red). The velocity field is chosen to have a slight angle with respect to the horizontal axis to better highlight the dependency between cells. Label 12 represent where upwind boundary conditions should be provided. Right: Associated dependency graph. Transport is performed from top to bottom nodes.

For more details on our implementation of the implicit Discontinuous Galerkin method, we refer to [2].

3.3. **Parallelization**

The whole transport solver is parallelized. We adopt different strategies for the parallelism in the poloidal directions and for the toroidal direction.

3.3.1. *Poloidal parallelism: task graph.*

Because of the dependency graph we cannot perform all the computations in parallel. For instance, for the mesh of Figure 3.2. it is necessary to compute the solution in Cell 9 first. But then Cells 8 and 10 can be computed independently in parallel, *etc.*

The parallelization is done by a task graph approach. We have tested several implementations. For more details on the implementation, we refer for instance to [2], where the algorithm is parallelized with the StarPU runtime, or to [5] where we use a specialized DAG (Direct Acyclic Graph) clustering algorithm. The DAG algorithm relies on an OpenMP implementation.

3.3.2. *Toroidal parallelism: message passing*

For the toroidal direction, we associate to each poloidal plane one MPI process. Thus, the shift simply consists in an MPI send/receive operation with the neighbor poloidal planes. This approach imposes a constraint on the time step. An alternative would be to replace the shift by a semi-Lagrangian solver. This would imply exchanges with more neighbors and thus more MPI communications.

4. **Kinetic solver**

In this section, we present the whole numerical scheme to solve the kinetic model (2.6) and we validate it on two and three-dimensional test cases.

4.1. **Splitting**

The kinetic model (2.6) is solved using a splitting method. Starting from f_k^n the kinetic fields at time step $t_n = n\Delta t$, we make the following two steps to update them:

(1) We first solve the n_v free transport equations

$$\partial_t f_k + \boldsymbol{\lambda}_k \cdot \boldsymbol{\nabla}_{\mathbf{r}} f_k = 0,$$

using the transport solvers introduced in Section 3. We denote by f_k^* the value of the fields after the free transport step (3.1).

(2) then the relaxation to the equilibrium distribution, corresponding to the right-hand side of (2.6), is solved using an over-relaxation process as used in [7, 10, 17]. For obtaining the new value of the field at time step t_{n+1}, the kinetic fields f_k^* are recombined according to

$$f_k^{n+1} = \omega f_k^{*,eq} + (1-\omega) f_k^*. \tag{4.1}$$

where $\omega > 0$ is the numerical relaxation parameter. This scheme can be formally derived using a θ-scheme for the relaxation part of Equation (2.6) and this leads to the following relation: $\omega = \Delta t/(\tau + \theta \Delta t)$. This relation is actually not taken into account: the actual numerical scheme considers a given ω, independent from the time step and with no reference to the parameter τ. We refer to [8] for more details.

For $\omega = 1$, we obtained the classical transport-projection scheme. The over-relaxation formula (4.1) enables us to obtain second order accuracy in time when the relaxation parameter $\omega = 2$. When the relaxation parameter satisfies $1 \leq \omega < 2$, then the scheme is only first-order accurate and the time integration introduces a numerical damping. This damping can sometimes be used for numerical stabilization purpose.

4.2. **CFL condition**

As we are using an implicit transport solver in the poloidal plane, we want to emphasize that the time step is not constrained by the poloidal discretization or the poloidal speed λ_p. The only constraint comes from the choice of the shift transport solver in the toroidal direction which imposes the relation: $\lambda_t \Delta t = \Delta\varphi$. This implies a CFL bound for the toroidal dynamics:

$$n_{\mathrm{CFL}}^{\mathrm{tor}} = \frac{\lambda_t \Delta t}{\Delta\varphi} = 1. \tag{4.2}$$

However, the full CFL number is defined by

$$n_{\mathrm{CFL}}^{\mathrm{full}} = \frac{u_{\max}\Delta t}{\min(\delta_p, \Delta\varphi)} \tag{4.3}$$

where $\delta_p > 0$ denotes the minimal distance between two interpolation points in the poloidal plane and $u_{\max}$ the maximal advection speed over the simulation. Unlike explicit methods, this CFL number is not constrained by our scheme and as a consequence, δ_p can be set without stability considerations, i.e. independently of the possible advection speed in the toroidal direction.

4.3. **Boundary conditions**

In the φ direction, the boundary conditions are naturally periodic. At the poloidal planes boundaries, we consider Dirichlet boundary conditions. More precisely, in the DG scheme (3.2) if R corresponds to a boundary, then the unknown value of f_R is given by

$$f_R = f^{eq}(\rho, \mathbf{u}),$$

where ρ and $\mathbf{u}$ are imposed boundary data and f^{eq} is given by (2.7).

4.4. **Validation in two-dimensional geometry**

First we validate the kinetic solver described above in a single poloidal plane. We consider the two-dimensional rotation of a Gaussian pulse. The pulse is given by

$$g(r, z, \varphi, t) = \exp(-30\,((r' - 1)^2 + z'^2 + \varphi'^2)),$$

with

$$\begin{aligned} r' &= \cos(\beta t) r + \sin(\beta t) z, \\ z' &= \cos(\beta t) z - \sin(\beta t) r, \\ \varphi' &= 0, \end{aligned}$$

and we take $\beta = 1/4$. This function satisfies the advection equation

$$\partial_t g + \mathbf{u} \cdot \boldsymbol{\nabla} g = 0,$$

with

$$\mathbf{u} = \beta \begin{pmatrix} -z \\ r \\ 0 \end{pmatrix},$$

or equivalently the conservative transport equation:

$$\partial_t g + \boldsymbol{\nabla} \cdot (g\mathbf{u}) = 0,$$

since the velocity field is divergence free: $\boldsymbol{\nabla} \cdot \mathbf{u} = 0$. We solve this equation in the disk

$$\Omega = \Big\{ (r, z, \varphi), \quad (r - r_t)^2 + z^2 < 4, \quad \varphi = 0 \Big\}.$$

We numerically compute the above solution with two different meshes with refinement levels of 5 and 10 (see Figure 4.2) and with $n_t = 500$ and $n_t = 1000$ time steps. The time step is given by $\Delta t = 2\pi/n_t$. We take $\lambda_p = 1$, use polynomials of degree 2 in the DG method and $\omega = 2$. The numerical solution is plotted at time $t_{\max} = 2\pi$ on Figure 4.1. We observe that the Gaussian shape is well preserved by the LBM scheme. We also check the order of convergence in the L^2 norm. The measured numerical order is here 2.405, which is consistent with the theoretical order of convergence.

The time scheme is only second order accurate and is thus the limiting factor for the convergence. Anyway we also check the spatial convergence of the scheme for higher order DG approximation. For this, we use a sufficiently small time-step in such a way that the error due to the time approximation is negligible. We perform simulations with two meshes with two different refinements of 10 and 20 (see Figure 4.2). We then obtain the numerical orders of Table 4.1, which is still consistent with the theoretical order of convergence.

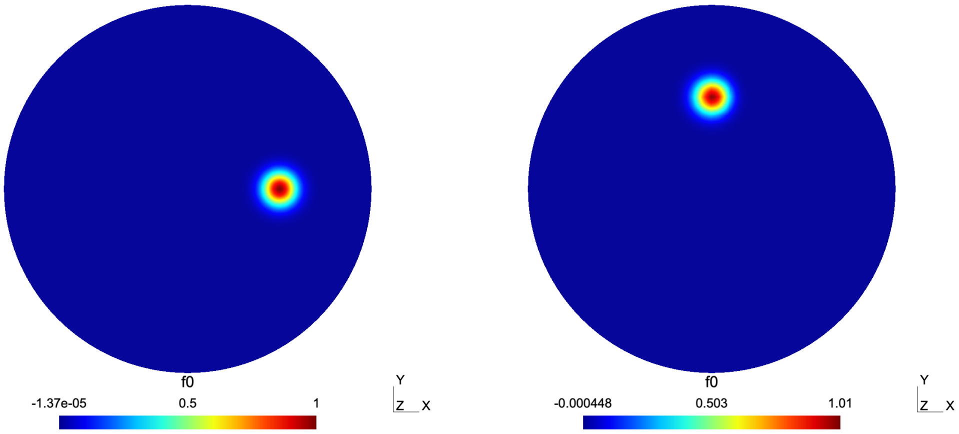

Figure 4.1. (Validation in two dimensions) Numerical solution at times $t = 0$ (left) and $t = 2\pi$ (right) after the Gaussian pulse has done a quarter turn. Parameters of the kinetic solver: $\Delta t = 2\pi/n_t$ with $n_t = 1000$, $\omega = 2$, $\lambda_p = 1$ and DG of order 2.

Table 4.1. (Validation in two dimensions) Numerical order of the DG scheme for several values of the polynomial order p. In this test, the time step is chosen in such a way that the time error is negligible. Parameters of the kinetic solver: $\lambda_p = 1$, $\omega = 2$.

p	num. order
1	0.997
2	2.68
3	3.772

4.5. **Validation in a 3D periodic cylinder**

Now we also activate transport in the third direction. We consider a three-dimensional helical shift of a Gaussian pulse. The pulse is given by

$$g(r,z,\varphi,t) = \exp(-30((r'-1)^2 + z'^2 + \varphi'^2)),$$

with

$$\begin{aligned} r' &= \cos(2\pi\gamma t)r + \sin(2\pi\gamma t)z, \\ z' &= \cos(2\pi\gamma t)z - \sin(2\pi\gamma t)r, \\ \varphi' &= \varphi + \beta t. \end{aligned}$$

We can also consider a periodic function in the z direction

$$g(r,z,\varphi,t) = \exp(-30((r'-1)^2 + z'^2)) \sin(\pi\varphi').$$

and we take $\gamma = 0.04$, $\beta = 0.25$. This function satisfies the transport equation

$$\partial_t g + \boldsymbol{\nabla}_{\mathbf{r}} \cdot (g\mathbf{u}) = 0,$$

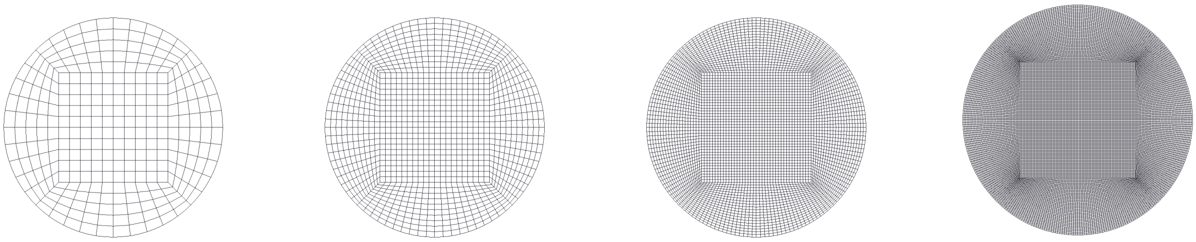

Figure 4.2. Poloidal meshes with refinements $n_{\text{raf}} = 5$, 10, 20 and 40. The disk is decomposed into 12 patches and each patch is meshed with n_{raf}^2 quadrilaterals, where n_{raf} is the refinement number.

with the divergence free velocity field

$$\mathbf{u} = \begin{pmatrix} -2\pi\gamma z \\ 2\pi\gamma r \\ -\beta \end{pmatrix}.$$

The computational domain is the cylinder

$$\Omega = \Big\{(r, z, \varphi), \quad (r - r_t)^2 + z^2 < 4, \quad -1 = \varphi_{\min} < \varphi < 1 = \varphi_{\max}\Big\}.$$

We consider a maximal time $t_{\max} = 1$. In this way, the Gaussian pulse will move a little bit, without touching the boundaries in the φ direction. As stated above, parallelism is managed by OpenMP and MPI. OpenMP is used for optimizing the transport solver in the poloidal planes. MPI communications are used for the parallelism in the toroidal direction. We recall that parallelism in the φ direction is directly linked to the number of poloidal planes, as one MPI process is associated to each plane. If, for instance, we choose $n_p = 64$ toroidal planes then the space step in the φ direction equals

$$\Delta\varphi = \frac{\varphi_{\max} - \varphi_{\min}}{n_p},$$

and there are n_p MPI processes. Moreover, with a kinetic speed $\lambda_t = 1$ and as we are considering LBM in the φ direction, the time step is given by

$$\Delta t = \frac{\Delta\varphi}{\lambda_t} = 0.03125.$$

We thus have to perform n_t time iterations

$$n_t = \frac{t_{\max}}{\Delta t} = 32.$$

In order to check the convergence order, we have considered three different meshes of the poloidal plane with a refinement of 5, 10, 20 and 40, as represented in Figure 4.2. While refining in the poloidal direction, we also refine in the toroidal direction by taking 32, 64, 128, 256 poloidal planes or equivalently MPI processes. We consider the second order DG scheme and we take $\lambda_p = \lambda_t = 1$. We compute the error in the L^2 norm and obtain the error curve of Figure 4.3. The order of convergence based on the 64 and 128 refinement levels is 2.226 (2.155 for 128-256 refinements). In this way, we have numerically validated the expected accuracy of the full kinetic scheme.

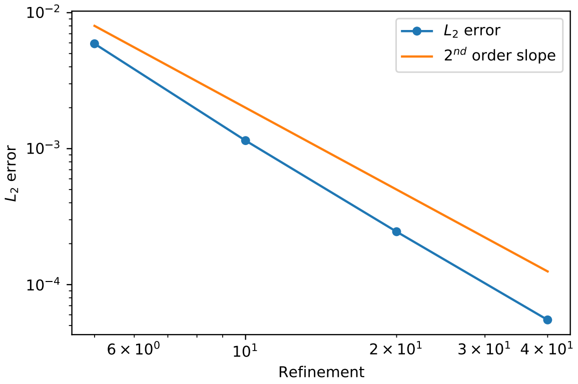

Figure 4.3. (Validation in three dimensions) L^2 error as a function of the refinement level. Parameters: $\lambda_t = \lambda_p = 1$, DG order $p = 2$.

5. Applications to plasma dynamics

In this section, we apply the transport solver to more complex test cases from plasma physics. We consider the transport of charged particles in a tokamak along the magnetic field lines. These charged particles create an electric field and the drift theory then implies that, in the mean, their motion is perturbed by this electric field.

For validation purpose, we neglect curvature effect and thus consider transport models in *straight toroidal domains*. Extension to curved domains will be studied in a future work. Moreover, next simulations consider *annular poloidal planes*, but there is no restriction on the geometry of the poloidal planes. This annular geometry is chosen in order to validate the model on classical test cases.

5.1. Drift-kinetic like model

Here we consider the following simplified model.. The charge density $\rho(\mathbf{r}, t)$ is transported according to equation (2.2) at the drift velocity

$$\mathbf{u} = \boldsymbol{E} \times \boldsymbol{e}_\varphi + \boldsymbol{B}, \tag{5.1}$$

where the electric field $\boldsymbol{E}$ is the gradient of the electric potential V,

$$\boldsymbol{E} = -\nabla_{\mathbf{r}} V. \tag{5.2}$$

and $\boldsymbol{B}$ denotes a magnetic field that is externally imposed, $\boldsymbol{B} = (b_r, b_z, b_\varphi)^T$. In other words, the drift velocity writes:

$$\mathbf{u} = \left(-\frac{\partial V}{\partial z} + b_r, \frac{\partial V}{\partial r} + b_z, b_\varphi\right)^T.$$

The electric potential, whose derivative in the φ direction is neglected, is a solution to a Poisson equation

$$-\Delta V = \rho, \tag{5.3}$$

where the Laplacian operator acts only in the poloidal plane:

$$\Delta = \frac{\partial^2}{\partial r^2} + \frac{\partial^2}{\partial z^2}.$$

We assume homogeneous Dirichlet boundary conditions for the potential. This implies that $\boldsymbol{E} \times \boldsymbol{e}_\varphi$ is tangent at the boundary. If $\boldsymbol{B}$ is also tangent to the boundary, then no mass escapes the domain. This model, made of the transport equation along $\boldsymbol{B}$ and the poloidal drift is a very simplified drift-kinetic model with a unique parallel velocity [18]. Thus we focus on the resolution of the transport dynamics in the spatial domain and do not consider physical kinetic effect.

From (5.1), (5.2) and (5.3) we see that the drift velocity is obtained from a linear operator applied to the density

$$\mathbf{u} = \mathbf{u}(\rho).$$

The drift-kinetic model can also be written

$$\partial_t \rho + \nabla_{\mathbf{r}} \cdot (\rho \mathbf{u}(\rho)) = 0, \tag{5.4}$$

and therefore is a non-linear model.

In addition to the transport solver described above, we thus have to implement a poloidal Poisson solver, which will be applied in each poloidal plane independently. We will test the accuracy of the coupling between the transport and the Poisson solver. In practice, we use a standard conforming finite element solver for the Poisson equation. The solver uses the same mesh and the same order of basis functions as the transport solver. Once the Poisson equation is solved, the electric potential is computed according to (5.2). The numerical electric field is thus generally time and space dependent and discontinuous at the interfaces between the DG cells.

Given a stationary solution to the drift-kinetic model, a slight perturbation is applied to this particular solution. In certain configurations one observes instabilities. The objective of the following two test cases is to compute numerically the instability rate. In simple geometries the instability rate can be evaluated by analytical or semi-analytical methods. It is then possible to test the accuracy of the Lattice-Boltzmann approach. We first consider a two-dimensional diocotron instability and then a fully three-dimensional instability.

5.2. **Two-dimensional diocotron test-case**

We suppose that $\boldsymbol{B} = (0,0,0)^T$. Then the solution does not depend on φ and we recover the so-called guiding-center model. The problem is thus indeed two-dimensional. We note $\alpha \in \mathbb{R}^+$ and $\theta \in [0, 2\pi)$, the polar coordinates in the poloidal plane:

$$\begin{aligned} r &= \alpha \cos\theta + r_t, \\ z &= \alpha \sin\theta, \end{aligned}$$

with $r_t > 0$ the radius of the torus. Here the computational domain is a circular annulus defined by

$$\Omega = \left\{(r, z, \varphi), \varphi = 0 \text{ and } \alpha_{\min} \leqslant \alpha \leqslant \alpha_{\max}\right\}.$$

We will test our solver on the Diocotron test case described in [9, 22]. The density is initialized with a continuous function

$$\rho(r, z, \varphi, t) = \tilde{\rho}(\alpha, \theta, t) = (1 + \varepsilon \cos(k\theta))\, e^{-\frac{(\alpha-\alpha_0)^2}{2\sigma^2}}, \tag{5.5}$$

with $\sigma > 0$ and $\alpha_0 \in (\alpha_{\min}, \alpha_{\max})$ and where $\varepsilon > 0$ and $k \in \mathbb{N}^*$ parametrize the perturbation. The electric potential V satisfies equation (5.3) with homogeneous Dirichlet boundary conditions at $\alpha = \alpha_{\min}$ and $\alpha = \alpha_{\max}$.

For $\varepsilon = 0$ this density is a solution to the drift kinetic model (5.4): indeed the density, and thus also the potential, has a cylindrical symmetry. Then, because of (5.1), the velocity $\mathbf{u}$ is parallel to the vector $(-z, r)$ and the transport equation leaves the initial density invariant.

When $\varepsilon > 0$, an instability may appear to depend on the parameters $\alpha_{\min}$, $\alpha_{\max}$ and α_0. In this special case, where the initial condition has cylindrical symmetry, we can estimate the instability growth rate with a simpler one-dimensional numerical method in order to compare it with the results of our solver. We explain now this simpler method.

5.2.1. *Instability rate*

In polar coordinates in the poloidal plane, the guiding-center model writes

$$\begin{aligned} \partial_t \tilde{\rho} + \frac{1}{\alpha}\left(\partial_\alpha \tilde{V} \partial_\theta \tilde{\rho} - \partial_\theta \tilde{V} \partial_\alpha \tilde{\rho}\right) = 0, \\ -\frac{1}{\alpha}\partial_\alpha\left(\alpha\,\partial_\alpha \tilde{V}\right) - \frac{1}{\alpha^2}\partial_{\theta^2}\tilde{V} = \tilde{\rho}. \end{aligned} \tag{5.6}$$

Let $(\tilde{\rho}_0(\alpha), \tilde{V}_0(\alpha))$ be a stationary radial solution to this system. We look for a solution which writes as the sum of the stationary solution and a small perturbation

$$\begin{aligned} \tilde{\rho}(\alpha,\theta,t) = \tilde{\rho}_0(\alpha) + \varepsilon\,\tilde{\rho}_k(\alpha)\,e^{ik\theta}\,e^{-i\eta t}, \\ \tilde{V}(\alpha,\theta,t) = \tilde{V}_0(\alpha) + \varepsilon\,\tilde{V}_k(\alpha)\,e^{ik\theta}\,e^{-i\eta t}. \end{aligned}$$

Inserting this ansatz into system (5.6), we obtain

$$-\eta\tilde{\rho}_k(\alpha) + \frac{k}{\alpha}\left(\tilde{V}_0'(\alpha)\tilde{\rho}_k(\alpha) - \tilde{V}_k(\alpha)\tilde{\rho}_0'(\alpha)\right) = 0, \tag{5.7}$$

$$-\tilde{V}_k''(\alpha) - \frac{1}{\alpha}\tilde{V}_k'(\alpha) + \frac{k^2}{\alpha^2}\tilde{V}_k(\alpha) = \tilde{\rho}_k(\alpha). \tag{5.8}$$

Plugging (5.8) into (5.7), this system can be rewritten as the following equation

$$\left(\frac{1}{\alpha}\tilde{V}_0'(\alpha) - c\right)\left(-\tilde{V}_k''(\alpha) - \frac{1}{\alpha}\tilde{V}_k'(\alpha) + \frac{k^2}{\alpha^2}\tilde{V}_k(\alpha)\right) = \frac{1}{\alpha}\tilde{V}_k(\alpha)\tilde{\rho}_0'(\alpha), \tag{5.9}$$

which is a generalized eigenvalue problem with eigenvalue η/k. We keep the eigenvalue with the largest imaginary part (if there exists) and the instability rate is then given by $\Im(\eta)$.

In the case (5.5), the radial solution is

$$\tilde{\rho}_0(\alpha) = e^{-\frac{(\alpha-\alpha 0)^2}{2\sigma^2}}.$$

Moreover, since $\tilde{V}_0'$ is solution to the following elliptic equation

$$-\tilde{V}_0''(\alpha) - \frac{1}{\alpha}\tilde{V}_0'(\alpha) = \tilde{\rho}_0(\alpha),$$

we obtain the following expression

$$\tilde{V}_0'(\alpha) = \frac{1}{\alpha}\left(c_1 - \sqrt{\frac{\pi}{2}}\alpha_0\sigma\,\mathrm{erf}\left(\frac{\alpha-\alpha_0}{\sqrt{2}\sigma}\right) + \sigma^2 e^{-\frac{(\alpha-\alpha 0)^2}{2\sigma^2}}\right),$$

and then

$$\tilde{V}_0(\alpha) = \int_{\alpha_{\min}}^{\alpha} \tilde{V}_0'(\alpha)d\alpha + c_2.$$

The Dirichlet boundary condition $\tilde{V}_0(\alpha_{\min}) = 0$ implies that $c_2 = 0$, and $\tilde{V}_0(\alpha_{\max}) = 0$ implies that $\int_{\alpha_{\min}}^{\alpha_{\max}} \tilde{V}_0'(\alpha)d\alpha = 0$. Therefore, we can approximate c_1 with a numerical integration.

For finding out the unstable mode, we discretize the one-dimensional equation (5.9) using a finite difference method over the interval $[\alpha_{\min}, \alpha_{\max}]$. We approximate the first and second derivatives of $\tilde{V}_k(\alpha)$ by standard central differences. Finally, the solution of the equation (5.9) is numerically solved as a generalized eigenvalue problem. Of course, the number of discretization points is taken large enough, in order to achieve high accuracy.

5.2.2. *Validation*

We choose $\alpha_{\min} = 1$, $\alpha_{\max} = 10$, and the following parameters for the initial condition (5.5): $\alpha_0 = 4.5$, $\sigma = 0.5$, and $k = 2$. Using the method described in the above section, we obtain that this initial condition (5.5) leads to an instability of rate 0.15215.

To validate our code, we have done a dynamic numerical simulation with the full kinetic solver using the same parameters. We consider $\lambda_p = 7$, use polynomials of degree 2 in the DG method, a time step $\Delta t = 0.0125$ and a relaxation parameter $\omega = 1.999$. We consider a mesh of size $N_\alpha \times N_\theta = 100 \times 60$. Density plots are given in Figure 5.1. The execution takes approximately 1.5 hours. We define the poloidal CFL number by

$$n_{\mathrm{CFL}}^{\mathrm{pol}} = \frac{v_{\max}\Delta t}{\delta_p},$$

with v_{max} the maximal speed and δ_p the minimal distance between two interpolation points. In this case, the CFL number equals 2.35. With such a CFL number, the calculation would be unstable with a DG explicit method. In Figure 5.2, we plot the k-th Fourier mode in θ of the potential whose formula is given by:

$$h(t) = \frac{1}{(\alpha_{\max} - \alpha_{\min})} \int_{\alpha_{\min}}^{\alpha_{\max}} \int_0^{2\pi} e^{-ik\theta} \tilde{V}(\alpha, \theta, t) d\alpha d\theta \tag{5.10}$$

and estimate the slope in the linear growth regime. We observe an instability rate of 0.15123. This is in accordance with the theoretical instability rate.

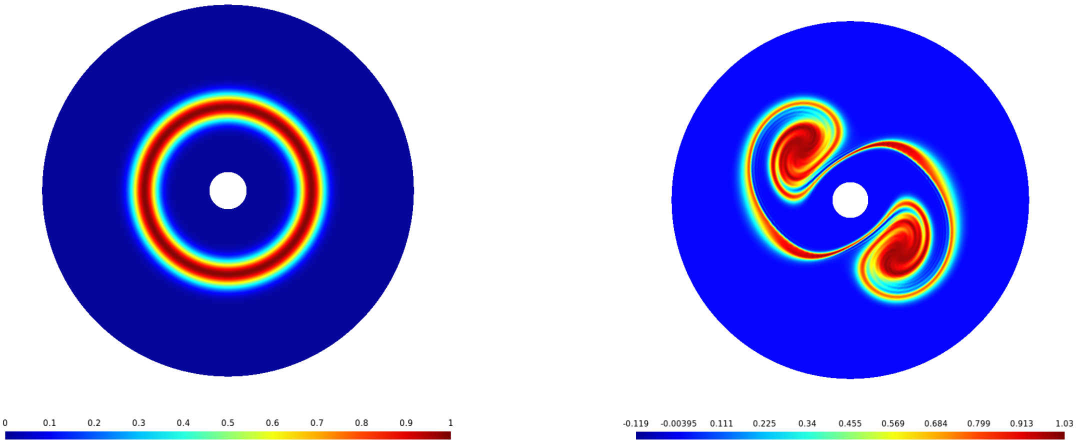

Figure 5.1. (Two-dimensional diocotron test case) Density at $t = 0$ and $t = 100$. Parameters: $\lambda_p = 7$, DG order $p = 2$, $\Delta t = 0.0125$, $\omega = 1.999$.

5.3. **Diocotron test-case in a periodic cylinder**

We consider now the three dimensional drift kinetic model (2.2)-(5.1)-(5.2)-(5.3) We choose a magnetic field $\boldsymbol{B}$ oriented in the θ and φ directions

$$\boldsymbol{B} = (-b_\theta \sin(\theta), b_\theta \cos(\theta), b_\varphi)^T,$$

with $b_\theta > 0$ and $b_\varphi > 0$. The computational domain is the cylinder

$$\Omega = \left\{ (r, z, \varphi) \;\middle|\; \alpha_{\min} \leqslant \sqrt{(r - r_t)^2 + z^2} \leqslant \alpha_{\max},\ 0 \leqslant \varphi \leqslant L \right\}.$$

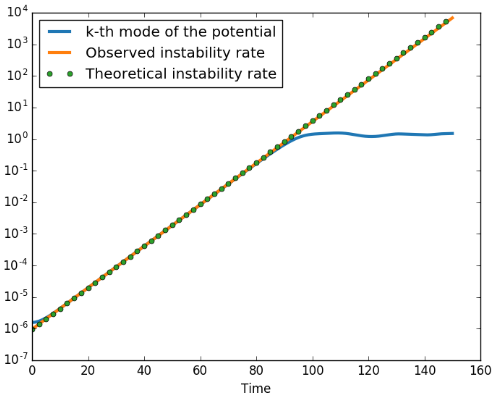

Figure 5.2. (Two-dimensional diocotron test case) k-th Fourier mode of the potential over time and the instability rate given by the slope of the Fourier mode between $t = 20$ and $t = 60$, compared to the theoretical instability rate. Parameters: $\lambda_p = 7$, DG order $p = 2$, $\Delta t = 0.0125$, $\omega = 1.999$.

We consider periodic boundary conditions in φ and homogeneous Dirichlet boundary conditions for the potential $\tilde{V}$. We consider the following initial density

$$\rho(r, z, \varphi, 0) = \tilde{\rho}(\alpha, \theta, \varphi, 0) = \left(1 + \varepsilon \cos\left(k\theta + \ell\varphi\frac{2\pi}{L}\right)\right) e^{-\frac{(\alpha-\alpha_0)^2}{2\sigma^2}}.$$

with $\sigma > 0$ and $\alpha_0 \in (\alpha_{\min}, \alpha_{\max})$ and where $\varepsilon > 0$ and $k, \ell \in \mathbb{N}^*$ parametrize the perturbation. Like in the two-dimensional case, we first derive the eigenvalue problem for calculating the instability rate. We show that this rate does not depend on the perturbation along φ.

5.3.1. *Instability rate*

With cylindrical coordinates, the system (2.1)-(5.2)-(5.3) becomes

$$\partial_t\tilde{\rho} - \frac{1}{\alpha}\partial_\theta\tilde{V}\ \partial_\alpha\tilde{\rho} + \left(\frac{1}{\alpha}\partial_\alpha\tilde{V} + b_\theta\right)\partial_\theta\tilde{\rho} + b_\varphi\partial_\varphi\tilde{\rho} = 0, \tag{5.11}$$

$$-\frac{1}{\alpha}\partial_\alpha(\alpha\partial_\alpha\tilde{V}) - \frac{1}{\alpha^2}\partial_{\theta^2}\tilde{V} = \tilde{\rho}. \tag{5.12}$$

Let $(\tilde{\rho}_0(\alpha), \tilde{V}_0(\alpha))$ be a stationary radial solution of this system, which only depends on α and is constant along the θ and z directions. We look for the equation satisfied by a small perturbation around this stationary solution

$$\tilde{\rho}(\alpha, \theta, \varphi, t) = \tilde{\rho}_0(\alpha) + \varepsilon\ \tilde{\rho}_{k,l}(\alpha)\ e^{ik\theta}e^{il\varphi}e^{-i\eta t},$$
$$\tilde{V}(\alpha, \theta, \varphi, t) = \tilde{V}_0(\alpha) + \varepsilon\ \tilde{V}_{k,l}(\alpha)\ e^{ik\theta}e^{il\varphi}e^{-i\eta t},$$

with $\varepsilon > 0$. We plug this expression into Equations (5.11)-(5.12) and keep only the $O(\epsilon)$ terms. We obtain

$$-\eta\tilde{\rho}_{k,\ell}(\alpha) - \frac{k}{\alpha}\tilde{\rho}_0'(\alpha)\tilde{V}_{k,\ell}(\alpha) + k\tilde{\rho}_{k,\ell}(\alpha)\left(\frac{1}{\alpha}\tilde{V}_0'(\alpha) + b_\theta\right) + \ell\tilde{\rho}_{k,\ell}(\alpha)b_\varphi = 0, \tag{5.13}$$

$$-\frac{1}{\alpha}\partial_\alpha(\alpha\tilde{V}_{k,\ell}'(\alpha)) + \frac{k^2}{\alpha^2}\tilde{V}_{k,\ell}(\alpha) = \tilde{\rho}_{k,\ell}(\alpha). \tag{5.14}$$

Plugging (5.14) into (5.13), we finally get

$$\left(\frac{1}{\alpha}\tilde{V}_0'(\alpha) + b_\theta + \frac{\ell}{k}b_\varphi - \frac{\eta}{k}\right)\left(-\tilde{V}_{k,\ell}''(\alpha) - \frac{1}{r}\tilde{V}_{k,\ell}'(\alpha) + \frac{k^2}{\alpha^2}\tilde{V}_{k,\ell}(\alpha)\right) = \frac{1}{\alpha}\tilde{\rho}_0'(\alpha)\tilde{V}_{k,\ell}(\alpha). \tag{5.15}$$

which is the same equation obtained in (5.9) for the two-dimensional case, except for the presence of the term $b_\theta + \frac{\ell}{k}b_\varphi$. Consequently, compared to the 2D case, the eigenvalues are only shifted by this real value and the instability rate does not change as it is equal to the largest imaginary part of the eigenvalues.

5.3.2. *Validation*

The poloidal planes have the same dimension as in the two-dimensional test case ($\alpha_{\min} = 1$, $\alpha_{\max} = 10$) while the cylinder length is taken equal to $L = 200$. The magnetic field is supposed to have a large toroidal component compared with its poloidal ones: $b_\theta = 0.1$ and $b_\varphi = 200$. The distribution is initially concentrated around $\alpha_0 = 4.5$ with a Gaussian distribution of standard deviation $\sigma = 0.5$. Finally, the perturbation is taken of size $\varepsilon = 10^{-6}$ and involves modes $k = 2$ in the θ direction and $\ell = 1$ in the φ direction.

The numerical parameters of the simulations are as follows. Each poloidal mesh is composed of $4 \times N_\alpha \times N_\theta = 4 \times 80 \times 50$ elements. In the toroidal direction, we use $n_p = 128$ poloidal planes and so 128 MPI processes. Consistently with the involved advection velocities, we choose kinetic speeds of norm $\lambda_p = 7$ in the poloidal plane and $\lambda_t = 600$ in the toroidal direction. The time step is thus equal to $\Delta t = L/(n_p\lambda_t) = 0.002604$ and the relaxation parameter is chosen equal to $\omega = 1.99$.

Figure 5.3 shows the time evolution of the density and Figure 5.4 the density at final time $t_{\max} = 100$ for different poloidal planes. As expected, we can observe that the density in the poloidal planes are identical up to a rotation. As time goes on, two vortices develop and lead to fine structures in the poloidal planes as in the two-dimensional case. The computation takes 26 hours.

We would like to emphasize that the full CFL number, defined in (4.3), takes the value 33.17. This is due to the very fast speed in the toroidal direction which is not resolved by the poloidal plane mesh. Indeed, the minimal distance between interpolation points in the poloidal planes equals $\delta_p = 0.01570$ and the time step is not small enough to capture speed of order 200 for this fine resolution. An explicit scheme would require either decreasing the time step and thus increasing the toroidal discretization, or a coarser poloidal mesh at the cost of a loss of accuracy. Consequently, explicit schemes would not be able to cope with both the large velocity in the toroidal direction and a very fine grid in the poloidal plane.

In Figure 5.5, we plot the time evolution of quantity (5.10) integrated along φ. We compare the theoretical instability rate computed before (see Sec. (5.3.1)) to the one obtained by the three-dimensional kinetic solver. We expect an instability rate of 0.15215, and we obtain 0.15061, which again validates the kinetic solver.

When we decrease the number of MPI processes n_p, the number of planes in the poloidal direction decrease and the poloidal discretization step $\Delta\varphi$ is then smaller. Figure 5.6 shows the densities obtained at time $t_{max} = 100$ for the poloidal plane $\varphi = 0$ for $n_p = 32$, $n_p = 64$ and $n_p = 128$. As the number of MPI processes decreases, we observe that the solution is less and less precise and dispersion effect are more and more present. This is mainly due to the increase of the time step imposed by the Lattice-Boltzmann method. Other toroidal solver will be studied in future works.

5.4. **Parallel efficiency: weak scaling**

Now, we want to compare the time of execution for different numbers of MPI processes n_p. We keep the parameters used in the section 5.3, we only reduce the final time to $t_{max} = 1$, in order to get

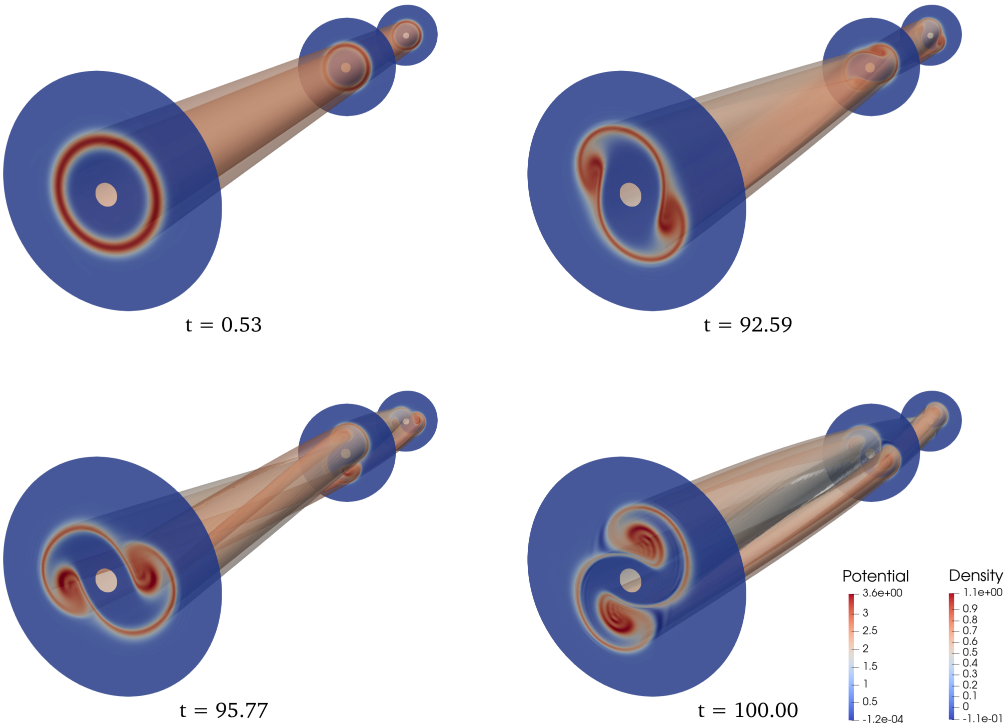

Figure 5.3. (Three-dimensional diocotron test case) Four successive instants of the 3D solution showing cross sections of density colormaps and density isosurface colored by the potential. Parameters: $\lambda_p = 7$, $\lambda_t = 600$, DG order $p = 2$, $\omega = 1.99$, $n_p = 65$.

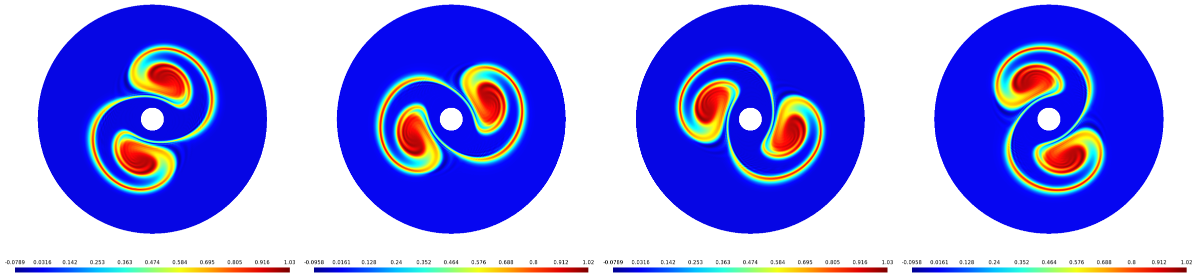

Figure 5.4. (Three-dimensional diocotron test case) Density at $t_{\max} = 100$ for the poloidal planes $\varphi = 0$, $\varphi = \frac{L}{4}$, $\varphi = \frac{L}{2}$ and $\varphi = \frac{3L}{4}$. Parameters: $\lambda_p = 7$, $\lambda_t = 600$, DG order $p = 2$, $\omega = 1.99$, $n_p = 128$.

shorter computations. As we have $\Delta t = \frac{L}{n_p \lambda_t}$, if the MPI communication time is neglected, we expect the time to double when the number of MPI processes is multiplied by 2. We perform numerical computations with $n_p = 6$, 12, 24, 48 and 96 MPI processes. We obtain the CPU time evolution of Figure 5.7. We observe a correct scaling of the parallel method when the number of MPI processes increases. We emphasize that for this test we have deactivated the OpenMP acceleration on each MPI node. For harnessing the OpenMP acceleration, we should run the code on a larger parallel computer.

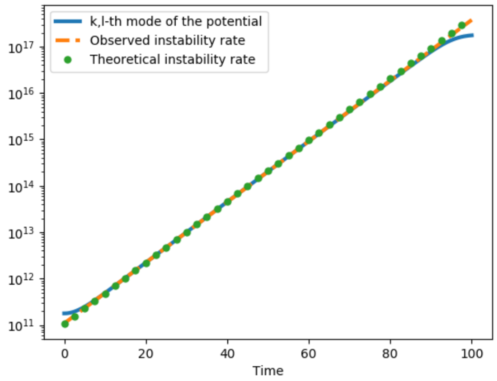

FIGURE 5.5. (Three-dimensional diocotron test case) Instability rate observed compared to the theoretical one. Parameters: $\lambda_p = 7$, $\lambda_t = 600$, DG order $p = 2$, $\omega = 1.99$, $n_p = 128$.

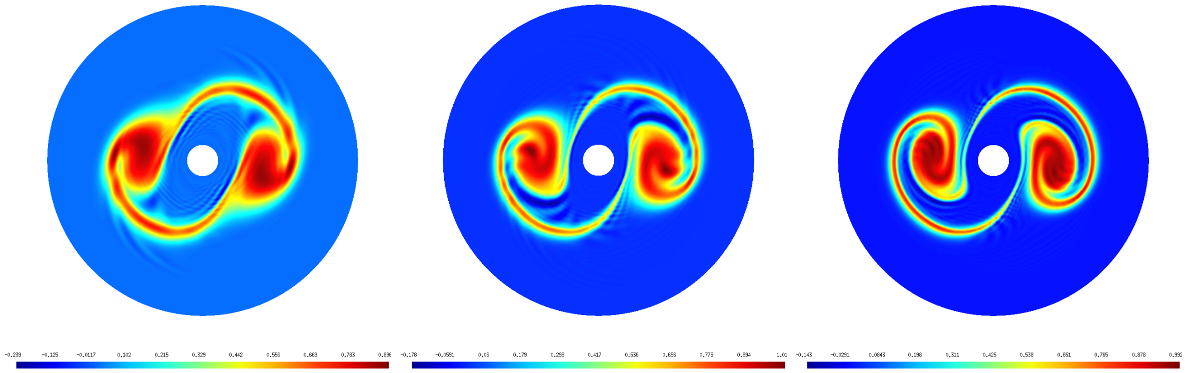

FIGURE 5.6. (Three-dimensional diocotron test case) Density at $t_{max} = 100$ in the poloidal plane $\varphi = 0$ obtained with $n_p = 32$, $n_p = 64$ and $n_p = 128$. Parameters: $L = 1$, $\lambda_p = 7$, $\lambda_t = 3$, DG order $p = 2$, $\omega = 1.99$.

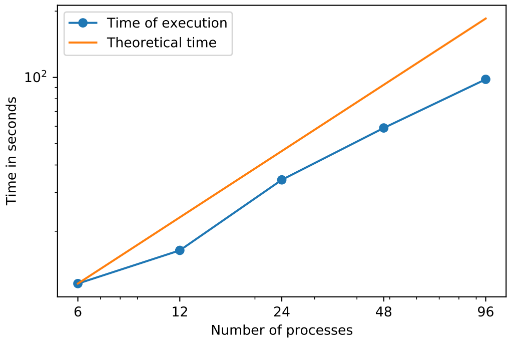

FIGURE 5.7. (Parallel weak scaling) Time of execution according to the number of MPI processes.

6. Conclusion

In this work, we have proposed a new optimized numerical method for solving non-homogeneous conservative transport equations in toroidal geometries. The method is conservative, high-order in space and time and has the complexity of a time-explicit scheme. It is also able to handle unstructured meshes of the poloidal plane, which is very useful for numerical simulations in tokamaks, like ITER. Finally, it presents many features that allow an efficient parallelization.

The method has been first validated on academic transport test cases. The method has then been applied to more physical configurations. The transport has been coupled with a toroidal Poisson solver for computing the electric potential generated by the charge density motion. In this framework, we have been able to validate the method on the estimation of diocotron instability rates.

We are currently working on the extension of the method to other physical models: with more realistic toroidal geometries, more complex transport models, several populations of particles, and richer gyrokinetic models. We also plan to run the solver on larger supercomputers.

Appendix A. Subcharacteristic stability condition

We consider the kinetic model (2.6). The sub-characteristic stability condition is a stability condition associated to the equivalent equation of (2.6) in the limit $\tau \to 0$. Indeed, in that asymptotic, the equivalent equation writes:

$$\partial_t \rho + \nabla_{\mathbf{r}} \cdot (\rho \mathbf{u}) = \tau \, \nabla_{\mathbf{r}} \cdot (\mathcal{D} \nabla_{\mathbf{r}} \rho) + O(\tau^2),$$

where the $O(\tau)$ term involves a diffusion matrix given by:

$$\mathcal{D} = \begin{pmatrix} \frac{\lambda_p^2}{3} - u_r^2 & -u_r u_z & -u_r u_\varphi \\ -u_r u_z & \frac{\lambda_p^2}{3} - u_z^2 & -u_z u_\varphi \\ -u_r u_\varphi & -u_z u_\varphi & \frac{\lambda_t^2}{3} - u_\varphi^2 \end{pmatrix}.$$

We refer to [4] for the derivation of this equivalent equation. For this equation to be L^2 stable, this diffusion matrix has to be positive definite: this is the sub-characteristic condition. Here, the eigenvalues of $\mathcal{D}$ are

$$d_1 = \frac{\lambda_p^2}{3}, \quad d_2 = \frac{1}{2}\left(-\alpha + \frac{\lambda_p^2 + \lambda_t^2}{3} - \|\mathbf{u}\|^2\right), \quad d_3 = \frac{1}{2}\left(\alpha + \frac{\lambda_p^2 + \lambda_t^2}{3} - \|\mathbf{u}\|^2\right),$$

with

$$\alpha = \sqrt{\left(\frac{\lambda_p^2 + \lambda_t^2}{3} - \|\mathbf{u}\|^2\right)^2 + 4\frac{\lambda_p^2}{3}\left(u_\varphi^2 - \frac{\lambda_t^2}{3}\right) + 4\frac{\lambda_t^2}{3}\,(u_r^2 + u_z^2)}.$$

As d_1 is always positive and $d_2 \leqslant d_3$, the model is stable if and only if $d_2 \geqslant 0$ or equivalently

$$\frac{\lambda_p^2 + \lambda_t^2}{3} - \|\mathbf{u}\|^2 \geqslant \alpha$$

After some easy computations, we find that this is also equivalent

$$\frac{u_r^2}{\lambda_p^2} + \frac{u_z^2}{\lambda_p^2} + \frac{u_\varphi^2}{\lambda_t^2} \leqslant \frac{1}{3},$$

which states that all the velocity field $\mathbf{u}$ have to belongs to the ellipsoid with parameters $\left(\frac{\lambda_p}{\sqrt{3}}, \frac{\lambda_p}{\sqrt{3}}, \frac{\lambda_t}{\sqrt{3}}\right)$.

Acknowledgements

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

References

[1] D. Aregba-Driollet and R. Natalini. Discrete kinetic schemes for multidimensional systems of conservation laws. *SIAM J. Numer. Anal.*, 37(6):1973–2004, 2000.

[2] J. Badwaik, M. Boileau, D. Coulette, E. Franck, P. Helluy, C. Klingenberg, L. Mendoza, and H. Oberlin. Task-based parallelization of an implicit kinetic scheme. *ESAIM, Proc. Surv.*, 63:60–77, 2018.

[3] N. Besse and M. Mehrenberger. Convergence of classes of high-order semi-Lagrangian schemes for the Vlasov–Poisson system. *Math. Comput.*, 77(261):93–123, 2008.

[4] F. Bouchut. Construction of BGK models with a family of kinetic entropies for a given system of conservation laws. *J. Stat. Phys.*, 95(1-2):113–170, 1999.

[5] B. Bramas and A. Ketterlin. Improving parallel executions by increasing task granularity in task-based runtime systems using acyclic DAG clustering. *PeerJ Computer Science*, 6, 2020.

[6] S. Chen and G. D. Doolen. Lattice Boltzmann method for fluid flows. *Annu. Rev. Fluid Mech.*, 30(1):329–364, 1998.

[7] D. Coulette, E. Franck, P. Helluy, M. Mehrenberger, and L. Navoret. High-order implicit palindromic Discontinuous Galerkin method for kinetic-relaxation approximation. *Comput. Fluids*, 2019.

[8] C. Courtès, D. Coulette, E. Franck, and L. Navoret. Vectorial kinetic relaxation model with central velocity. application to implicit relaxations schemes. *Commun. Comput. Phys.*, 27(4), 2020.

[9] N. Crouseilles, P. Glanc, S. A. Hirstoaga, E. Madaule, M. Mehrenberger, and J. Pétri. A new fully two-dimensional conservative semi-Lagrangian method: applications on polar grids, from diocotron instability to ITG turbulence. *Eur. Phys. J. D, Atomic Mol. Opt. Plasma Phys.*, 68(9), 2014.

[10] P. J. Dellar. An interpretation and derivation of the lattice Boltzmann method using Strang splitting. *Comput. Math. Appl.*, 65(2):129–141, 2013.

[11] G. Dimarco and R. Loubère. Towards an ultra efficient kinetic scheme. Part I: Basics on the BGK equation. *J. Comput. Phys.*, 255:680–698, 2013.

[12] G. Dimarco, R. Loubère, J. Narski, and T. Rey. An efficient numerical method for solving the Boltzmann equation in multidimensions. *J. Comput. Phys.*, 353:46–81, 2018.

[13] F. Drui, E. Franck, P. Helluy, and L. Navoret. An analysis of over-relaxation in a kinetic approximation of systems of conservation laws. *C. R. Méc. Acad. Sci. Paris*, 347(3):259–269, 2019.

[14] F. Dubois. Nonlinear fourth order Taylor expansion of lattice Boltzmann schemes. *Asymptotic Anal.*, 127(4):297–337, 2022.

[15] F. Dubois, T. Février, and B. Graille. Lattice Boltzmann schemes with relative velocities. *Commun. Comput. Phys.*, 17(4):1088–1112, 2015.

[16] F. Dubois and P. Lallemand. Towards higher order lattice Boltzmann schemes. *J. Stat. Mech. Theory Exp.*, 2009(6), 2009.

[17] B. Graille. Approximation of mono-dimensional hyperbolic systems: A lattice Boltzmann scheme as a relaxation method. *J. Comput. Phys.*, 266:74–88, 2014.

[18] V. Grandgirard, J. Abiteboul, J. Bigot, T. Cartier-Michaud, N. Crouseilles, G. Dif-Pradalier, C. Ehrlacher, D. Esteve, X. Garbet, P. Ghendrih, et al. A 5D gyrokinetic full-f global semi-Lagrangian code for flux-driven ion turbulence simulations. *Comput. Phys. Commun.*, 207:35–68, 2016.

[19] H. Guillard, J. Lakhlili, A. Loseille, A. Loyer, B. Nkonga, . Ratnani, and A. Elarif. Tokamesh: A software for mesh generation in Tokamaks. Research Report, RR-9230, CASTOR., 2018.

[20] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications.* Springer, 2007.

[21] G. Latu, M. Mehrenberger, Y. Güçlü, M. Ottaviani, and E. Sonnendrücker. Field-aligned interpolation for semi-Lagrangian gyrokinetic simulations. *J. Sci. Comput.*, 74(3):1601–1650, 2018.

[22] E. Madaule, S. A. Hirstoaga, M. Mehrenberger, and J. Pétri. Semi-Lagrangian simulations of the diocotron instability. Research report, 2013.

[23] Y. H. Qian, D. d'Humières, and P. Lallemand. Lattice BGK models for Navier-Stokes equation. *Eur. Phys. Lett.*, 17(6):479, 1992.

[24] E. Sonnendrücker, J. Roche, P. Bertrand, and A. Ghizzo. The semi-Lagrangian method for the numerical resolution of the Vlasov equation. *J. Comput. Phys.*, 149(2):201–220, 1999.

[25] A. Storelli, L. Vermare, P. Hennequin, Ö. D. Gürcan, G. Dif-Pradalier, Y. Sarazin, X. Garbet, T. Görler, R. Singh, P. Morel, V. Grandgirard, P. Ghendrih, and Tore Supra team. Comprehensive comparisons of geodesic acoustic mode characteristics and dynamics between Tore Supra experiments and gyrokinetic simulations. *Phys. Plasmas*, 22(6):062508, 2015.

[26] R. Zhang, H. Fan, and H. Chen. A lattice Boltzmann approach for solving scalar transport equations. *Philos. Trans. R. Soc. Lond., Ser. A*, 369(1944):2264–2273, 2011.