Contributions à la simulation numérique et à la modélisation mathématique de problèmes issus de la physique des plasmas

Jacek Narski

Mémoire présenté par

Jacek NARSKI

en vue d'obtenir le diplôme

d'HABILITATION A DIRIGER RECHERCHES

Spécialité: Mathématiques Appliquées

CONTRIBUTIONS À LA SIMULATION NUMÉRIQUE ET À LA MODÉLISATION MATHÉMATIQUE DE PROBLÈMES ISSUS DE LA PHYSIQUE DES PLASMAS

Sutenue le 12 décembre 2017 devant le jury composé de :

Bruno Després (Université Paris VI)	Rapporteur
Francis Filbet (Université Paul Sabatier Toulouse III)	Coordinateur
Pauline Lafitte (Ecole Centrale Paris)	Présidente
Alexei Lozinski (Université de Franche-Comté)	Examinateur
Boniface Nkonga (Université de Nice Sophia-Antipolis)	Examinateur
Eric Sonnendrücker (Max Planck Institut für Plasmaphysik Garching)	Rapporteur

Contents

Chapter 1

Introduction

1 Research activities

My PhD project concerned mathematical studies of phase transition and alloy solidification. The goal was to incorporate to existing models the effect of shrinkage due to bigger density in the solid phase than in the liquid. The shrinkage is causing pressure drop and induces fluid flow in the domain and can result in physical defects in alloys. The approach considered belongs to a class of so called phase field models, where the phase field function varies smoothly from one in solid to zero in liquid through a slightly diffused interface. Alloy solidification itself is driven by temperature (assumed to be constant) and a local species concentration. The proposed model consisted of coupled set of PDEs: Navier-Stokes equations for fluid flow in the melt coupled with the phase field, non linear anisotropic phase field evolution equation coupled with the species concentration and species conservation equation coupled with both velocity field and phase field variable. In addition, the phase field functions varies smoothly from zero to one through an interface of thickness several orders of magnitude smaller than the size of the computational domain. This makes adaptive mesh refinement mandatory for numerical simulations. We introduced an adaptive algorithm based on *a posteriori* error estimation in order to build successive triangular meshes with high aspect ratio that ensure that the H^1-error of the solution is close to some prescribed tolerance. The results are presented in publications [84, 83, 85].

The goal of this report is to present my research activities posterior to my PhD thesis and strongly diverging from the above mentioned work. Two different axes can be distinguished. The first concerns numerical methods for strongly anisotropic elliptic and parabolic equations and is exposed in Chapter 2. It covers publications [25, 26, 30, 72, 31, 82] and collaboration with Pierre Degond, Fabrice Deluzet, Alexei Lozinski, Claudia Negulescu and Maurizio Ottaviani. The direct motivation for this work were numerical simulations related to plasma physics, were strong magnetic field is a source of the anisotropy in the system. The motion of magnetized particles is characterized by a fast gyration around magnetic field lines. The number of collisions in the plane perpendicular to the field is much more important than in the direction of the field. As a result, the

ratio between parallel and perpendicular diffusion coefficients can be very high achieving values as big as 10^{10}. Numerical resolution of such equations is a very delicate problem, as the discrete system can suffer from a very bad matrix conditioning impacting precision and convergence speed of iterative solvers. This is a direct result of the fact that the initial problem can become singular in the limit of infinite anisotropy, depending on boundary conditions. Above of that the locking phenomenon occurs when discrete mesh is not aligned with anisotropy direction. That is to say, the numerical solution tends to zero as the anisotropy in the system increases. This effect is caused by the polynomial approximation of the discrete solution: in the limit of infinite anisotropy the only polynomial that is constant in the privileged direction is constant on the mesh elements not aligned with the anisotropy. Taking into account the homogeneous Dirichlet boundary conditions, this polynomial is identically equal to zero. For those reasons any direct discretization of the anisotropic problem would fail to provide reliable results. Chapter 2 presents several methods conceived to overcome those difficulties. They all rely on a so called Asymptotic Preserving reformulation, where the initial problem is rewritten in an equivalent form with strong anisotropy removed from the equations. The resulting system does not suffer from bad conditioning nor from locking. The cost of this reformulation is addition of auxiliary variable(s). The methods are presented within the finite element framework, but are not limited to FEM and work equally well under finite difference or finite volume discretization.

The second axis of my research concentrates around fast numerical schemes for Boltzmann equation. It covers publications [34, 35, 81] and collaboration with Giacomo Dimarco, Raphaël Loubère and Thomas Rey. We are interested in kinetic description of gases, where the state of the system is described by so called distribution function. This function is defined in seven independent dimensions: three in physical space, three in velocity space and the time. This curse of dimensionality makes numerical simulations extremely resource consuming and not fitted for simple desktop stations due to memory limitations. Above of that the collisions between particle are modeled by multidimensional integrals that have to be evaluated at every time step and in every cell of space mesh. The existing methods are very time consuming, especially in seven dimensions. In order to perform realistic simulations one need to resort to high performance computing. Chapter 3 is devoted to development of efficient parallel methods for kinetic equations. Efficient parallelization strategies presented there in allow to perform complex full seven dimensional parallel simulations on wide range of systems: shared memory multi processor computational servers, GPU based systems and large distributed memory supercomputers, providing an efficient tool for rarefied gas modeling.

Finally, Chapter 4 briefly presents research not related to the main axes: a contribution to the multiscale finite element method for advection-diffusion and Stokes problems in perforated domain (in collaboration with P. Degond, A. Lozinski and Bagus Putra Muljadi) and a contribution to simulations of pedestrian flow (in collaboration with P. Degond, Jérôme Fehrenbach *et al.*).

2 Publication list

2.1 Publications in refereed journals

[JN1] [31] F. Deluzet and J. Narski. A two field iterated asymptotic-preserving method for highly anisotropic elliptic equations. *submitted*, 2017

[JN2] [81] J. Narski. Fast Kinetic Scheme : efficient MPI parallelization strategy for 3D Boltzmann equation. *to appear in CiCP*, 2018

[JN3] [35] G. Dimarco, R. Loubère, J. Narski, and T. Rey. An efficient numerical method for solving the Boltzmann equation in multidimensions. *Journal of Computational Physics*, 353(Supplement C):46–81, Jan. 2018

[JN4] [34] G. Dimarco, R. Loubère, and J. Narski. Towards an ultra efficient kinetic scheme. Part III: High-performance-computing. *Journal of Computational Physics*, 284:22–39, 2015

[JN5] [73] A. Lozinski, J. Narski, and C. Negulescu. Numerical analysis of an asymptotic-preserving scheme for anisotropic elliptic equations. *submitted, arXiv:1507.00879 [math]*, July 2015. arXiv: 1507.00879

[JN6] [79] B. P. Muljadi, J. Narski, A. Lozinski, and P. Degond. Nonconforming multiscale finite element method for stokes flows in heterogeneous media. part i: Methodologies and numerical experiments. *Multiscale Modeling & Simulation*, 13(4):1146–1172, 2015

[JN7] [29] P. Degond, A. Lozinski, B. P. Muljadi, and J. Narski. Crouzeix-raviart ms-fem with bubble functions for diffusion and advection-diffusion in perforated media. *Communications in Computational Physics*, 17(4):887–907, 2015

[JN8] [43] J. Fehrenbach, J. Narski, J. Hua, S. Lemercier, A. Jelić, C. Appert-Rolland, S. Donikian, J. Pettré, and P. Degond. Time-delayed follow-the-leader model for pedestrians walking in line. *Networks and Heterogeneous Media*, 10(3):579–608, 2015

[JN9] [82] J. Narski and M. Ottaviani. Asymptotic preserving scheme for strongly anisotropic parabolic equations for arbitrary anisotropy direction. *Computer Physics Communications*, 185(12):3189–3203, 2014

[JN10] [72] A. Lozinski, J. Narski, and C. Negulescu. Highly anisotropic nonlinear temperature balance equation and its numerical solution using asymptotic-preserving schemes of second order in time. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(06):1701–1724, 2014

[JN11] [30] P. Degond, A. Lozinski, J. Narski, and C. Negulescu. An asymptotic-preserving method for highly anisotropic elliptic equations based on a micro-macro decomposition. *Journal of Computational Physics*, 231(7):2724–2740, 2012

[JN12] [25] P. Degond, F. Deluzet, A. Lozinski, J. Narski, and C. Negulescu. Duality-based asymptotic-preserving method for highly anisotropic diffusion equations. *Commun. Math. Sci.*, 10(1):1–31, 2012

[JN13] [83] J. Narski and M. Picasso. Adaptive 3d finite elements with high aspect ratio for dendritic growth of a binary alloy including fluid flow induced by shrinkage. *FDMP: Fluid Dynamics & Materials Processing*, 3(1):49–64, 2007

[JN14] [84] J. Narski and M. Picasso. Adaptive finite elements with high aspect ratio for dendritic growth of a binary alloy including fluid flow induced by shrinkage. *Computer methods in applied mechanics and engineering*, 196(37):3562–3576, 2007

2.2 Proceedings

[JNP1] [36] G. Dimarco and J. Narski. Hybrid monte carlo schemes for plasma simulations. In *AIP Conference Proceedings*, volume 1389, pages 1130–1133. AIP, 2011

[JNP2] [26] P. Degond, F. Deluzet, D. Maldarella, J. Narski, and C. Negulescu. Hybrid model for the coupling of an asymptotic preserving scheme with the asymptotic limit model: The one dimensional case. In *ESAIM: Proceedings*, volume 32, pages 23–30. EDP Sciences, 2011

[JNP3] [85] J. Narski and M. Picasso. *Adaptive Finite Elements with High Aspect Ratio for Dendritic Growth of a Binary Alloy Including Fluid Flow Induced by Shrinkage*, pages 327–337. Birkhäuser Basel, Basel, 2007

Chapter 2

Asymptotic-Preserving methods for strongly anisotropic elliptic and parabolic equations

The first part is devoted to the numerical resolution of strongly anisotropic equations. Such problems are often encountered in many fields of application, for example flow in porous media [5, 60], semiconductor modeling [75], atmospheric or oceanic flows [103]. The initial motivation of this work was related to plasma physics: atmospheric plasma [64, 66], internal fusion plasma [9, 28] or plasma thrusters [1]. In this context, the anisotropy direction is defined by a direction of magnetic field, where the gyration of magnetized particles around field lines is causing a large number of collisions in a plane perpendicular to the field. On the other hand, the motion in the direction of the field is rather undisturbed. This explains the difference in the mobility of particles in parallel and perpendicular directions. The ratio of those two values, denoted as $1/\varepsilon$, can be as high as 10^{10}. The resulting problem becomes singular when ε approaches zero and the direct discretization of the equations leads to very badly conditioned systems. Resolution of those systems is unfeasible for small values of ε. Moreover, the anisotropy rate may change in the computational domain so that any method designed to work for strong anisotropy only would fail in isotropic regions. Therefore there is a need for a robust and accurate scheme that cover wide range of anisotropies.

Numerical methods for anisotropic equation have been extensively studied and many approaches have been proposed: for example finite volume methods with suitably chosen fluxes [42, 69, 105], finite difference schemes [57, 71, 94, 104], slope limiters for finite elements [68], domain decomposition [55, 67] and multigrid methods [54, 86]. Numerical integration along the anisotropy direction was considered in [21, 98]. Anisotropic problems in toroidal geometries were studied in [22, 93, 95]. The goal of this chapter is to present numerical methods to address this difficulty developed with Pierre Degond, Fabrice Deluzet, Alexei Lozinski, Claudia Negulescu and Maurizio Ottaviani in [25, 30, 31, 72, 73, 82]. The methods are based on the so called Asymptotic Preserving reformulation introduced initially in [63].

The chapter is divided into two parts: the first presents the prototype elliptic anisotropic

equation together with Asymptotic Preserving numerical methods. In the beginning, the Duality Based Asymptotic Preserving scheme (DB) [25] is introduced. It relies on an orthogonal decomposition of the solution to the initial problem into two parts: its mean part computed in the direction of the anisotropy and the L_2-orthogonal to the mean part complement of average zero along the anisotropy direction carrying information about oscillations. Those two functions belong to the spaces which are difficult to discretize for a general direction of anisotropy. This problem is overcome by a Lagrange multiplier technique and a system o five equations is finally obtained. Next the AP scheme based on the Micro-Macro decomposition (MM) is presented [30]. In the MM method the decomposition is no longer required to be orthogonal. The resulting scheme is a significant improvement over the DB method: the obtained system consists of only two equations. The field of application of both DB and MM schemes is however limited only to simple topologies of the anisotropy, where no closed field lines are allowed. This limitation has been first overcome for the parabolic problems in [82] by introduction of the stabilization term inspired by stabilization techniques for the Stokes problem. This Micro Macro scheme with Stabilization (MMS) was studied in more detail in [73] for elliptic equations. Both MM and MMS schemes have a matrix conditioning proportional to $1/h^4$ at best, with h being the mesh size. Moreover, the particular saddle point structure makes it difficult to find an efficient preconditioner, practically ruling out iterative solvers. This inconvenience has been eliminated in the last presented scheme for elliptic equations: an iterative regularised method [31], which relies on an iterative resolution of a mildly anisotropic equation. The main advantage of this method is that the problem solved in each iteration is a diffusion equation with conditioning proportional to $1/h^2$ and very well studied in literature. The second part of this chapter deals with parabolic problems: linear and non-linear heat equation, where the MM and MMS schemes are coupled with second order time discretization scheme. In particular [72] shows the existence, uniqueness and positivity of the solution to a non-linear heat equation supplied with Robin boundary conditions and encountered in the plasma physics. This equation is reformulated with the MM method. The proposed time discretization schemes include the first order implicit Euler, Crank-Nicolson and a diagonally implicit second order Runge Kutta methods. It is shown, that the Crank Nicolson scheme breaks the Asymptotic Preserving property of the method. It is argued, that the time discretization scheme that preserves the desired asymptotic should be L-stable. That is to say the numerical solution of a test problem $y' = ky$ should approach zero in just one iteration whenever $|k\Delta t|$ tends to infinity. This is not the case for the Crank Nicolson scheme. Finally, the [82] introduces first the stabilized MMS scheme in the context of linear heat transfer equation and presents some numerical results of the so called magnetic island (a region of closed field lines) moving in the computational domain, the test case inspired by magnetic filed lines reconnection encountered in plasma physics.

1 Elliptic equation

The mathematical problem we are interested in is given by

$$\begin{cases} -\nabla \cdot \mathbb{A}_\varepsilon \nabla u^\varepsilon = f & \text{in } \Omega, \\ n \cdot \mathbb{A}_\varepsilon \nabla u^\varepsilon = 0 & \text{on } \Gamma_N\,, \\ u^\varepsilon = 0 & \text{on } \Gamma_D\,, \end{cases} \tag{2.1}$$

where $\mathbb{A}_\varepsilon$ is an anisotropic matrix defined as

$$\mathbb{A}_\varepsilon = \frac{1}{\varepsilon} A_\parallel b \otimes b + (Id - b \otimes b) A_\perp (Id - b \otimes b)\,. \tag{2.2}$$

Let $b \in (C^\infty(\bar{\Omega}))^d$ be a smooth vector field in a domain $\Omega \subset \mathbb{R}^d$, with $d = 2, 3$ and $|b(x)| = 1$ for all $x \in \Omega$. Let us also decompose the boundary $\Gamma = \partial\Omega$ into two parts: Γ_D parallel to b and its complement Γ_N. That is to say:

$$\Gamma_D = \{x \in \Gamma \mid b(x) \cdot n = 0\}, \qquad \Gamma_N = \Gamma \setminus \Gamma_D, \tag{2.3}$$

where n is the outward normal to Ω.

Let us also decompose vectors $v \in \mathbb{R}^d$, gradients ∇u, with $u(x)$ a scalar function, and divergences $\nabla \cdot v$, with $v(x)$ a vector field, into a part parallel to the anisotropy direction and a part perpendicular to it:

$$\begin{aligned} &v_\parallel := (v \cdot b) b\,, && v_\perp := (Id - b \otimes b) v\,, && \text{such that} \quad v = v_\parallel + v_\perp\,, \\ &\nabla_\parallel u := (b \cdot \nabla u) b\,, && \nabla_\perp u := (Id - b \otimes b) \nabla u\,, && \text{such that} \quad \nabla u = \nabla_\parallel u + \nabla_\perp u\,, \\ &\nabla_\parallel \cdot v := \nabla \cdot v_\parallel\,, && \nabla_\perp \cdot v := \nabla \cdot v_\perp\,, && \text{such that} \quad \nabla \cdot v = \nabla_\parallel \cdot v + \nabla_\perp \cdot v\,, \end{aligned} \tag{2.4}$$

where we denoted by $\otimes$ the vector tensor product.

1.1 Singular Perturbation problem

Using the above introduced notation, the initial problem (2.1) can be restated as: find u^ε such that

$$\begin{cases} -\frac{1}{\varepsilon}\nabla_\parallel \cdot \left(A_\parallel \nabla_\parallel u^\varepsilon\right) - \nabla_\perp \cdot (A_\perp \nabla_\perp u^\varepsilon) = f & \text{in } \Omega, \\ \frac{1}{\varepsilon} n_\parallel \cdot \left(A_\parallel \nabla_\parallel u^\varepsilon\right) + n_\perp \cdot (A_\perp \nabla_\perp u^\varepsilon) = 0 & \text{on } \Gamma_N\,, \quad , \\ u^\varepsilon = 0 & \text{on } \Gamma_D\,, \end{cases} \tag{2.5}$$

with b such that $n \cdot b = 0$ on Γ_D and $n \cdot b \neq 0$ on Γ_N. We assume that the source term and diffusion coefficients fulfill the following hypothesis.

Hypothesis 1 *Let $f \in L^2(\Omega)$ and $\partial\overset{\circ}{\Omega}_D \neq \varnothing$. The diffusion coefficients $A_\parallel \in L^\infty(\Omega)$ and $A_\perp \in \mathbb{M}_{d\times d}(L^\infty(\Omega))$ are supposed to satisfy*

$$0 < A_0 \leq A_\parallel(x) \leq A_1\,, \quad f.a.a.\ x \in \Omega, \tag{2.6}$$

$$A_0||v||^2 \leq v^t A_\perp(x) v \leq A_1||v||^2\,, \quad \forall v \in \mathbb{R}^d \ \text{and f.a.a.}\ x \in \Omega. \tag{2.7}$$

Please note that instead of Neumann boundary condition one could consider the periodic boundary conditions. This problem is singular in the limit of $\varepsilon \to 0$. Indeed, when ε is set to 0, the problem reduces to

$$\begin{cases} -\nabla_\parallel \cdot A_\parallel \nabla_\parallel u^0 = 0 & \text{in } \Omega, \\ n_\parallel \cdot \left(A_\parallel \nabla u^0\right) = 0 & \text{on } \Gamma_N\,, \\ u^\varepsilon = 0 & \text{on } \Gamma_D\,, \end{cases} \tag{2.8}$$

which admits infinite number of solutions as any function that is constant in the direction of anisotropy solves this problem. The correct limit of the solution can be found by multiplying (2.8) by a test function in

$$\mathcal{G} = \{v \in H^1(\Omega) \mid \nabla_\parallel v = 0\ ,\ v|_{\Gamma_D} = 0\}, \tag{2.9}$$

the space of functions that are constant in the direction of anisotropy and has a zero trace on Γ_D. This leads to the following, well posed problem: find $u^0 \in \mathcal{G}$ such that

$$\int_\Omega (A_\perp \nabla_\perp) u^0 \cdot \nabla_\perp v = \int_\Omega f v \ \ , \ \ \forall v \in \mathcal{G}. \tag{2.10}$$

The philosophy of the Asymptotic Preserving schemes relies on a suitable reformulation of the initial equation (valid for all values of ε) in such a way, that for $\varepsilon \to 0$ the correct limit problem is obtained.

Another problem encountered when dealing with numerical resolution of anisotropic problems is the locking phenomenon [6, 7]. It is particularly visible when unstructured grid and/or variable anisotropy direction and low order numerical method is considered. Let us again consider the reduced problem (2.8). If discrete space does not contain functions that are constant in the direction of the anisotropy, than the solution of this problem is equal to 0. That is the case for example for unstructured triangular grid with $\mathbb{P}_1$ finite elements or for rectangular Cartesian grid with variable anisotropy direction. This phenomenon is not related to the fact that the reduced problem is ill posed on continuous level. For small non zero values of ε big enough that the matrix related to the discretized version of the (2.5) problem does not suffer from bad conditioning the locking phenomenon is manifested in the discrete solution converging to zero as ε gets smaller. Especially when first order approximation is used and the discrete grid is rather coarse.

1.2 Duality Based Asymptotic Preserving scheme

One of the remedies to the above mentioned problems is a decomposition of the solution into two parts: a part that is constant in the direction of the anisotropy and a part carrying information about fluctuations. The most evident decomposition is to express u, the solution to the original singular perturbation problem as sum of its average along the direction of anisotropy $p^\varepsilon_{\rm DB}$ and the fluctuating part $q^\varepsilon_{\rm DB}$ of average zero in the direction of the anisotropy:

$$u^\varepsilon = p^\varepsilon_{\rm DB} + q^\varepsilon_{\rm DB}. \tag{2.11}$$

The mean part $p^\varepsilon_{\rm DB}$ belongs to the previously defined space $\mathcal{G}$, subspace of $\mathcal{V}$ while $q^\varepsilon_{\rm DB}$ belongs to its L_2-orthogonal complement $\mathcal{A}$ defined by

$$\mathcal{A} := \{q \in \mathcal{V} \,|(q,p) = 0 \ , \ \forall p \in \mathcal{G}\}\,. \tag{2.12}$$

This kind of decomposition was already studied in the case of anisotropy aligned with coordinate system in [27] for diffusion equation and in [16] for the Euler-Lorentz equations. In [25] I have contributed to the generalization of this method to the case of variable anisotropy direction. The key idea of this method is to make use of three Lagrange multipliers to ensure the orthogonality of the decomposition and that the function $p^\varepsilon_{\rm DB}$ belongs to the space $\mathcal{G}$. The resulting system consists thus of five equations: two for $p^\varepsilon_{\rm DB}$ and $q^\varepsilon_{\rm DB}$ and three for Lagrange multipliers. The description of the method follows.

We suppose that the following hypothesis holds.

Hypothesis 2 *The Hilbert-space $\mathcal{V}$ admits the decomposition*

$$\mathcal{V} = \mathcal{G} \oplus^\perp \mathcal{A}, \tag{2.13}$$

with $\mathcal{G}$ given by (2.9) and $\mathcal{A}$ given by (2.12) and where the orthogonality of the direct sum is taken with respect to the L^2-norm. Denoting by P the orthogonal projection on $\mathcal{G}$ with respect to the L^2 inner product:

$$P : \mathcal{V} \to \mathcal{G} \ \ \textit{such that} \ \ (Pu,p) = (u,p) \ \ \forall u \in \mathcal{V},\, p \in \mathcal{G}\,, \tag{2.14}$$

we shall suppose that this mapping is continuous and that we have the Poincaré-Wirtinger inequality

$$||u - Pu||_{L^2(\Omega)} \leq C||\nabla_\parallel u||_{L^2(\Omega)}\,, \quad \forall u \in \mathcal{V}\,. \tag{2.15}$$

We note that this kind of decomposition is not self evident but us typically satisfied for domains of interest.

Let us also define the operator

$$Q : \mathcal{V} \to \mathcal{A}\,, \quad Q = I - P\,. \tag{2.16}$$

Each function $u \in \mathcal{V}$ can be decomposed uniquely as $u = p_{\rm DB} + q_{\rm DB}$, where $p_{\rm DB} = Pu \in \mathcal{G}$ and $q_{\rm DB} = Qu \in \mathcal{A}$. Let us now reformulate the original singular perturbation problem: find $(p_{\rm DB}, q_{\rm DB}) \in \mathcal{G} \times \mathcal{A}$ such that

$$\begin{cases} a_\perp(p^\varepsilon_{\rm DB}, v) + a_\perp(q^\varepsilon_{\rm DB}, v) = (f, v), & \forall v \in \mathcal{G}, \\ a_\parallel(q^\varepsilon_{\rm DB}, w) + \varepsilon a_\perp(q^\varepsilon_{\rm DB}, w) + \varepsilon a_\perp(p^\varepsilon_{\rm DB}, w) = \varepsilon(f, w), & \forall w \in \mathcal{A}. \end{cases} \tag{2.17}$$

This system, contrary to the original problem, does not degenerate when ε goes to zero. Indeed, putting explicitly $\varepsilon = 0$ yields the following limit problem: find $(p^0, q^0) \in \mathcal{G} \times \mathcal{A}$ such that

$$\begin{cases} a_\perp(p^0_{\rm DB}, v) + a_\perp(q^0_{\rm DB}, v) = (f, v), & \forall v \in \mathcal{G}, \\ a_\parallel(q^0_{\rm DB}, w) = 0, & \forall w \in \mathcal{A}, \end{cases} \tag{2.18}$$

which has a unique solution with $p^0_{\rm DB}$ being the unique solution of the limit problem and $q^0_{\rm DB} \equiv 0$. Moreover, for any positive ε, the functions u^ε, $p^\varepsilon_{\rm DB}$ and $q^\varepsilon_{\rm DB}$ are bounded independently of ε in the H^1-norm. Furthermore, they converge to the solution of the limit problem when $\varepsilon \to 0$. This is stated in the following theorem.

Theorem 1 *For every $\varepsilon > 0$ the Asymptotic Preserving formulation (2.17), under Hypotheses 1 and 2, admits a unique solution $(p^\varepsilon_{DB}, q^\varepsilon_{DB}) \in \mathcal{G} \times \mathcal{A}$, where $u^\varepsilon := p^\varepsilon_{DB} + q^\varepsilon_{DB}$ is the unique solution in $\mathcal{V}$ of the Singular Perturbation model (2.5).*
These solutions satisfy the bounds

$$||u^\varepsilon||_{H^1(\Omega)} \le C||f||_{L^2(\Omega)}\,, \quad ||q^\varepsilon_{DB}||_{H^1(\Omega)} \le C||f||_{L^2(\Omega)}\,, \quad ||p^\varepsilon_{DB}||_{H^1(\Omega)} \le C||f||_{L^2(\Omega)}\,, \tag{2.19}$$

with an ε-independent constant $C > 0$. Moreover, we have

$$u^\varepsilon \to u^0,\ p^\varepsilon_{DB} \to u^0 \ \text{ and } \ q^\varepsilon_{DB} \to 0 \quad \text{in} \quad H^1(\Omega) \ \text{as } \varepsilon \to 0\,, \tag{2.20}$$

where $u^0 \in \mathcal{G}$ is the unique solution of the Limit model (2.10).

We have proven this theorem in [25]. The existence and uniqueness of a solution to the Singular Perturbation problem (2.5) follows from the Lax-Milgram theorem. Uniqueness of the decomposition $u^\varepsilon = p^\varepsilon_{\rm DB} + q^\varepsilon_{\rm DB}$ yields the existence and uniqueness of a solution to the Asymptotic Preserving reformulation (2.17). The bound on u^ε is obtained by a standard elliptic argument and since $p^\varepsilon_{\rm DB}$ and $q^\varepsilon_{\rm DB}$ are L^2-orthogonal projections of u^ε on the spaces $\mathcal{G}$ and $\mathcal{A}$ respectively, they are also bounded in the H_1 norm. Finally, we observe that $p^\varepsilon_{\rm DB}$ and $q^\varepsilon_{\rm DB}$ weakly converge to $p^0_{\rm DB} = u^0$ and $q^0 \equiv 0$. The strong convergence is obtained after subtracting the first equation of the limit problem (2.18) from the first the Asymptotic Preserving reformulation (2.17), adding the second equation of (2.17) with suitably chosen test functions and application of Poincaré-Wirtinger and Young inequalities.

The spaces $\mathcal{A}$ and $\mathcal{G}$ are difficult to discretize in the general setting, *i.e.* when the anisotropy direction is not aligned with the coordinate system and varies in the computational domain. In order to overcome this difficulty one can resort to the Lagrange multiplier technique.

The first step is to avoid discretization of the $\mathcal{A}$ space by remarking that it is a L^2 orthogonal complement to the $\mathcal{G}$ space in $\mathcal{V}$. The Asymptotic Preserving reformulation is therefore reformulated as follows: find $(p^\varepsilon_{\rm DB}, q^\varepsilon_{\rm DB}, l^\varepsilon) \in \mathcal{G}\times\mathcal{V}\times\mathcal{G}$ such that

$$\left\{\begin{array}{ll} a_\perp(p^\varepsilon_{\rm DB}, v) + a_\perp(q^\varepsilon_{\rm DB}, v) = (f, v) & \forall v \in \mathcal{G}, \\ a_\parallel(q^\varepsilon_{\rm DB}, w) + \varepsilon a_\perp(q^\varepsilon_{\rm DB}, w) + \varepsilon a_\perp(p^\varepsilon_{\rm DB}, w) + (l^\varepsilon, w) = \varepsilon(f, w) & \forall w \in \mathcal{V}, \\ (q^\varepsilon_{\rm DB}, \chi) = 0 & \forall \chi \in \mathcal{G}. \end{array}\right. \tag{2.21}$$

The constraint $(q^\varepsilon_{\rm DB}, \chi) = 0$, $\forall \chi \in \mathcal{G}$ is forcing the solution $q^\varepsilon_{\rm DB}$ to belong to $\mathcal{A}$, and this property is carried over to the limit $\varepsilon \to 0$. We have thus circumvented the difficulty of discretizing $\mathcal{A}$ by introducing a new variable and enlarging the linear system.

In [25] we prove the following proposition.

Proposition 1 *Problems (2.17) and (2.21) are equivalent.*

We observe that if $(p^\varepsilon_{\rm DB}, q^\varepsilon_{\rm DB}) \in \mathcal{G}\times\mathcal{A}$ is the unique solution of (2.17) then $(p^\varepsilon_{\rm DB}, q^\varepsilon_{\rm DB}, 0)$ solves (2.21). Reciprocally, if $(p^\varepsilon_{\rm DB}, q^\varepsilon_{\rm DB}, l^\varepsilon) \in \mathcal{G}\times\mathcal{V}\times\mathcal{G}$ is a solution of (2.21) then the last equation of (2.21) implies that $q^\varepsilon_{\rm DB} \in \mathcal{A}$. Choosing a test function $w \in \mathcal{G}$ in the second equation and combining with the first equation leads to $(l^\varepsilon, w) = 0$ for all $w \in \mathcal{G}$ and hence $l^\varepsilon \equiv 0$.

In order to eliminate $\mathcal{G}$ from the equation we first note that

$$p \in \mathcal{G} \Leftrightarrow \left\{\begin{array}{l} \nabla_\parallel p = 0 \\ p \in \mathcal{V} \end{array}\right. \Leftrightarrow \left\{\begin{array}{l} \displaystyle\int_\Omega A_\parallel \nabla_\parallel p \cdot \nabla_\parallel \lambda \, dx = a_\parallel(p, \lambda) = 0, \ \ \forall \lambda \in \mathcal{L}_{in} \\ p \in \mathcal{V}\,, \end{array}\right. \tag{2.22}$$

where $\mathcal{L}_{in}$ is a functional space that should be chosen large enough so that one could find for any $p \in \mathcal{V}$ a $\lambda \in \mathcal{L}_{in}$ with $\nabla_\parallel \lambda = \nabla_\parallel p$. On the other hand, the space $\mathcal{L}_{in}$ should be not too large in order to ensure the uniqueness of the Lagrange multipliers in the unconstrained system. A space that satisfies these two requirements can be defined as

$$\mathcal{L}_{in} := \{\lambda \in L^2(\Omega) \mid \nabla_\parallel \lambda \in L^2(\Omega)\,, \ \ \lambda_{|\partial\Omega_{in}} = 0\}\,, \quad \text{with} \quad \partial\Omega_{in} := \{x \in \partial\Omega \mid b(x)\cdot n < 0\}\,. \tag{2.23}$$

The above, together with the characterization of $\mathcal{G}$, allows to reformulate the initial Singular Perturbation problem as: Find $(p^\varepsilon_{\rm DB},\ \lambda^\varepsilon,\ q^\varepsilon_{\rm DB},\ l^\varepsilon,\ \mu^\varepsilon) \in \mathcal{V}\times\mathcal{L}_{in}\times\mathcal{V}\times\mathcal{V}\times\mathcal{L}_{in}$ such that

$$(DB) \quad \left\{\begin{array}{l} a_\perp(p^\varepsilon_{\rm DB}, \eta) + a_\perp(q^\varepsilon_{\rm DB}, \eta) + a_\parallel(\eta, \lambda^\varepsilon) = (f, \eta)\ , \quad \forall \eta \in \mathcal{V}\,, \\ a_\parallel(p^\varepsilon_{\rm DB}, \kappa) = 0\,, \quad \forall \kappa \in \mathcal{L}_{in}\,, \\ a_\parallel(q^\varepsilon_{\rm DB}, \xi) + \varepsilon a_\perp(q^\varepsilon_{\rm DB}, \xi) + \varepsilon a_\perp(p^\varepsilon_{\rm DB}, \xi) + (l^\varepsilon, \xi) = \varepsilon\,(f, \xi)\ , \quad \forall \xi \in \mathcal{V}\,, \\ (q^\varepsilon_{\rm DB}, \chi) + a_\parallel(\chi, \mu^\varepsilon) = 0\,, \quad \forall \chi \in \mathcal{V}\,, \\ a_\parallel(l^\varepsilon, \tau) = 0\,, \quad \forall \tau \in \mathcal{L}_{in}\,. \end{array}\right. \tag{2.24}$$

In this formulation the original problem is replaced by a set of five differential equations. The advantage of this scheme is that it is Asymptotic Preserving as (2.21) but it does not require discretization of the spaces $\mathcal{A}$ and $\mathcal{G}$. Moreover, no change of coordinates is required to guarantee the property that $p^\varepsilon_{\rm DB}$ is constant along the b-field and $q^\varepsilon_{\rm DB}$ is of average zero. This is particularly important in the context of evolutive problems, where the direction of b can change in time. This formulation will be called the Duality Based Asymptotic Preserving method (DB).

1.3 Asymptotic Preserving method based on Micro-Macro decomposition

The above presented reformulation presents one inconvenience: four additional unknowns are introduced resulting in a system of five equations required to solve the original singular perturbation problem for solely one unknown. In order to reduce the number of equations and solve the problem without resorting to expensive Lagrange multiplier technique I have proposed a different kind of decomposition. Instead of taking $u^\varepsilon = p^\varepsilon_{\rm DB} + q^\varepsilon_{\rm DB}$, with $p^\varepsilon_{\rm DB}$ and $q^\varepsilon_{\rm DB}$ being L_2 orthogonal, let us write $u^\varepsilon = p^\varepsilon_{\rm MM} + \varepsilon q^\varepsilon_{\rm MM}$ with $p^\varepsilon_{\rm MM} \in \mathcal{G}$ constant in the direction of anisotropy (as in the DB scheme) and $q^\varepsilon_{\rm MM}$ this time belonging to the space of functions vanishing on the inflow part of the boundary ($b\cdot n<0$), namely $q^\varepsilon_{\rm MM} \in \mathcal{L}_{in}$. The novelty of this kind of Micro-Macro decomposition lies in the fact that the oscillating part $q^\varepsilon_{\rm MM}$ caries not only information about the fluctuations but contains also some information about the part constant in the direction of anisotropy. Contrary to the Duality Based scheme, the decomposition is not orthogonal and $p^\varepsilon_{\rm MM}$ is not the average of u^ε in the anisotropy direction. There is no need to discretize the space orthogonal to $\mathcal{G}$ in this new decomposition. Moreover, suitable reformulation allows to eliminate $p^\varepsilon_{\rm MM}$ from the equations and the discretization of $\mathcal{G}$ is no longer necessary. This idea is presented in [30] together with some theoretical investigations obtained in collaboration with Pierre Degond, Alexei Lozinski and Claudia Negulescu.

Plugging the decomposition $u^\varepsilon = p^\varepsilon_{\rm MM} + \varepsilon q^\varepsilon_{\rm MM}$ into the original problems yields the following equation:

$$a_\perp(p^\varepsilon_{\rm MM} + \varepsilon q^\varepsilon_{\rm MM}, v) + \frac{1}{\varepsilon} a_\parallel(p^\varepsilon_{\rm MM} + \varepsilon q^\varepsilon_{\rm MM}, v) = (f, v) \ \ \forall v \in \mathcal{V}, \tag{2.25}$$

which becomes

$$a_\perp(u^\varepsilon, v) + a_\parallel(q^\varepsilon_{\rm MM}, v) = (f, v) \ \ \forall v \in \mathcal{V}, \tag{2.26}$$

after switching back to u^ε in the first term and using the fact that $\nabla_\parallel p^\varepsilon_{\rm MM} = 0$ in the second. In order to close the system a relation coupling u^ε and $q^\varepsilon_{\rm MM}$ is added taking the parallel Laplacian of the ansatz: $-\nabla_\parallel \cdot A_\parallel \nabla_\parallel u^\varepsilon = -\nabla_\parallel \cdot A_\parallel \nabla_\parallel (p^\varepsilon_{\rm MM} + \varepsilon q^\varepsilon_{\rm MM}) = -\varepsilon \nabla_\parallel \cdot A_\parallel \nabla_\parallel q^\varepsilon_{\rm MM}$. This yields the following Asymptotic Preserving reformulation: find $(u^\varepsilon, q^\varepsilon_{\rm MM}) \in \mathcal{V} \times \mathcal{L}_{in}$, solution of

$$(MM) \quad \begin{cases} a_\perp(u^\varepsilon, v) + a_\parallel(q^\varepsilon_{\rm MM}, v) = (f, v), & \forall v \in \mathcal{V} \\ a_\parallel(u^\varepsilon, w) - \varepsilon a_\parallel(q^\varepsilon_{\rm MM}, w) = 0, & \forall w \in \mathcal{L}_{in}\,. \end{cases} \tag{2.27}$$

System (2.27) is an equivalent reformulation (for fixed $\varepsilon > 0$) of the original P-problem multiplied by ε:

$$(P) \qquad a_\parallel(u^\varepsilon, v) + \varepsilon a_\perp(u^\varepsilon, v) = \varepsilon(f, v)\,, \quad \forall v \in \mathcal{V}\,. \tag{2.28}$$

Indeed, if $u^\varepsilon \in \mathcal{V}$ solves (2.28), then we can construct a unique $q^\varepsilon_{\rm MM} \in \mathcal{L}_{in}$ such that $\nabla_\parallel q^\varepsilon_{\rm MM} = (1/\varepsilon)\nabla_\parallel u^\varepsilon$. This, in weak form, gives the second equation of (2.27). Replacing then $\nabla_\parallel u^\varepsilon$ by $\varepsilon\nabla_\parallel q^\varepsilon_{\rm MM}$ inside (2.28), we see that $(u^\varepsilon, q^\varepsilon_{\rm MM})$ solves also the first equation in (2.27). Conversely, if (2.27) has a solution $(u^\varepsilon, q^\varepsilon_{\rm MM}) \in \mathcal{V} \times \mathcal{L}_{in}$ then the second equation implies $\varepsilon\nabla_\parallel q^\varepsilon_{\rm MM} = \nabla_\parallel u^\varepsilon$, which inserted in the first one, leads to the weak formulation (2.28).

This leads to another derivation of the MM scheme. Instead of plugging the ansatz $u^\varepsilon = p^\varepsilon_{\rm MM} + \varepsilon q^\varepsilon_{\rm MM}$ with $q^\varepsilon_{\rm MM}$ being equal to zero on the inflow part of the boundary one could proceed in a following way. Let us first remark that the main source of numerical issues in the resolution of the initial Singular Perturbation problem is the dominant derivative in the direction of the anisotropy. In order to overcome this difficulty, a new variable $q^\varepsilon_{\rm MM}$ is introduced. This variable fulfils the following relation: $\varepsilon\nabla_\parallel \cdot A_\parallel \nabla_\parallel q^\varepsilon_{\rm MM} = \nabla_\parallel \cdot A_\parallel \nabla_\parallel u^\varepsilon$. This operation allows to eliminate the $\frac{1}{\varepsilon}$ terms from the equation. The additional variable is not unique, as it is defined up to any function constant in the direction of anisotropy. One method allowing to provide uniqueness is to fix the value of $q^\varepsilon_{\rm MM}$ on every field line, for example setting it to zero on the inflow part of the boundary.

The MM-formulation (2.27) has a unique solution $(u^\varepsilon, q^\varepsilon_{\rm MM}) \in \mathcal{V} \times \mathcal{L}_{in}$ for all $\varepsilon > 0$ and $f \in L^2(\Omega)$, where $u^\varepsilon \in \mathcal{V}$ is the unique solution of the P-problem (2.28). The advantage of (2.27) over (2.28) consists in the fact that taking formally the limit $\varepsilon \to 0$ in (2.27) leads to the correct limit problem (2.10). Indeed, setting $\varepsilon = 0$ in the MM-formulation (2.27), we obtain the following problem (referred hereafter as the L-model): Find $(u^0, q^0_{\rm MM}) \in \mathcal{V} \times \mathcal{L}_{in}$ such that

$$(L) \quad \begin{cases} a_\perp(u^0, v) + a_\parallel(q^0_{\rm MM}, v) = (f, v), & \forall v \in \mathcal{V}, \\ a_\parallel(u^0, w) = 0, & \forall w \in \mathcal{L}_{in}\,. \end{cases} \tag{2.29}$$

Remark that (2.29) is formally an equivalent formulation of the Limit problem (2.10). In particular, if $(u^0, q^0_{\rm MM}) \in \mathcal{V} \times \mathcal{L}_{in}$ is a solution of (2.29), then $u^0 \in \mathcal{G}$, where $\mathcal{G}$ is defined in (2.9) and u^0 solves (2.10). The additional unknown $q^0_{\rm MM}$ serves here as a Lagrange multiplier responsible for the constraint $u^0 \in \mathcal{G}$. The existence of this Lagrange multiplier $q^0_{\rm MM} \in \mathcal{L}_{in}$ is not completely straight-forward to prove, since we do not have an inf-sup property for the bilinear form $a_\parallel$ on the pair of spaces $\mathcal{V} \times \mathcal{L}_{in}$. Fortunately, we can prove the existence assuming $f \in L^2(\Omega)$, cf. Theorem 2, thus establishing rigorously the equivalence between (2.10) and (2.29), at least for $f \in L^2(\Omega)$. This shall be part of the aim of the next subsection. The uniqueness is given by

Lemma 1 *Suppose that Hypothesis A is satisfied, in particular that $f \in L^2(\Omega)$. Then the solution to (2.29), if it exists, is unique.*

Proof. It is sufficient, due to linearity, to consider $f = 0$. Let thus $(u^0, q^0_{\rm MM}) \in \mathcal{V} \times \mathcal{L}_{in}$ be the solution of (2.29) for $f = 0$. Taking then test functions $v \in \mathcal{G}$, we get immediately $u^0 = 0$, implying $a_\parallel(q^0_{\rm MM}, v) = 0$ for all $v \in \mathcal{V}$. By density arguments one gets then

$q^0_{\rm MM} = 0$. ■

Let us now study the asymptotic behaviour of the MM scheme as $\varepsilon \to 0$. We shall suppose that the computational domain Ω is a tube made of field lines of b

Hypothesis 3 *There exists a smooth coordinate system $(\xi_1, \ldots, \xi_n)$ on Ω with $(\xi_1, \ldots, \xi_{n-1}) \in D$, $\xi_n \in (0,1)$, D being a smooth domain in $\mathbb{R}^{n-1}$, such that the field lines of b are given by the coordinate lines $(\xi_1, \ldots, \xi_{n-1}) = const$. Moreover, Γ_{in} is represented by $\xi_n = 0$, $(\xi_1, \ldots, \xi_{n-1}) \in D$; Γ_{out} is represented by $\xi_n = 1$, $(\xi_1, \ldots, \xi_{n-1}) \in D$ and Γ_D is represented by $\xi_n \in (0,1)$, $(\xi_1, \ldots, \xi_{n-1}) \in \partial D$.*

Supposing that the Hypotheses 1 and 3 hold, we prove in [30] following theorem.

Theorem 2 *Let Hypothesis 1 and 3 be satisfied and moreover suppose that $A_\perp \in \mathbb{M}_{d\times d}(W^{2,\infty}(\Omega))$ and $A_\parallel \in W^{2,\infty}(\Omega)$. Then the MM-problem (2.27) admits a unique solution $(u^\varepsilon, q^\varepsilon_{MM}) \in \mathcal{V} \times \mathcal{L}_{in}$ for any $\varepsilon > 0$, where u^ε is the unique solution of problem (2.28). There exists also a unique solution $(u^0, q^0_{MM}) \in \mathcal{V} \times \mathcal{L}_{in}$ of the L-problem (2.29), where $u^0 \in \mathcal{G}$ solves problem (2.10). Moreover, we have the following convergences as $\varepsilon \to 0$*

$$u^\varepsilon \to u^0 \quad in \quad \mathcal{V}\,, \qquad q^\varepsilon_{MM} \rightharpoonup q^0_{MM} \quad in \quad \mathcal{L}_{in}\,,$$

and the following bounds hold

$$||\nabla_\perp u^\varepsilon - \nabla_\perp u^0||_{L^2} \le C\sqrt{\varepsilon}||f||_{L^2}\,, \quad ||\nabla_\parallel u^\varepsilon||_{L^2} \le C\varepsilon\,||f||_{L^2} \;\; and \;\; ||\nabla_\parallel q^\varepsilon_{MM}||_{L^2} \le C||f||_{L^2}\,.$$

with a constant $C > 0$ independent of ε and f.

The proof of this theorem makes use of following lemmas (again proven in [30]).

Lemma 2 *For any $u \in H^1(\Omega)$ and $\varepsilon > 0$, there exists a unique $q_{MM} \in \mathcal{L}_{in}$ satisfying $\varepsilon\nabla_\parallel q_{MM} = \nabla_\parallel u$ almost everywhere. Moreover, if $u \in H^2(\Omega)$ then $q_{MM} \in H^1(\Omega)$ and if $u \in \mathcal{V} \cap H^2(\Omega)$ then $q_{MM} \in \mathcal{V}$.*

Lemma 3 *Let Hypothesis 1 and 3 be satisfied and moreover suppose that $A_\perp \in \mathbb{M}_{d\times d}(W^{2,\infty}(\Omega))$ and $A_\parallel \in W^{2,\infty}(\Omega)$. Then the solution $u^0 \in \mathcal{G}$ of (2.10) belongs to $H^2(\Omega)$ and satisfies the estimates*

$$||u^0||_{H^2} \le C||f||_{L^2}\,, \tag{2.30}$$

with a constant C independent of f.

We prove the first lemma by observing that $q_{\rm MM}$ is obtained by subtracting from u its value on the inflow boundary on each field line and dividing the result by ε. The second is proven by noting that u^0 is constant along the field lines, rewriting the limit problem 2.10 the coordinate system of Hypothesis 3 and integrating over ξ_n. The resulting equation is a weak formulation of an elliptic problem for u^0 on D with homogeneous Dirichlet boundary conditions on ∂D. The result is obtained from regularity results for elliptic equations.

In order to prove bounds in Theorem 2 we first subtract the Limit problem (2.29) from the MM formulation (2.27). We observe that for any $v \in \mathcal{G}$:

$$a_\perp(u^\varepsilon - u^0, v) = 0. \tag{2.31}$$

Thanks to Lemma 2, $q^\varepsilon_{\rm MM} \in \mathcal{V}$ and we can choose $v = u^\varepsilon - \varepsilon q^\varepsilon_{\rm MM} - u^0 \in \mathcal{G}$ obtaining

$$a_\perp(u^\varepsilon - u^0, u^\varepsilon - \varepsilon q^\varepsilon_{\rm MM} - u^0) = 0 \tag{2.32}$$

and hence

$$a_\perp(u^\varepsilon - u^0, u^\varepsilon - u^0) - \varepsilon a_\perp(u^\varepsilon, q^\varepsilon_{\rm MM}) = -\varepsilon a_\perp(u^0, q^\varepsilon_{\rm MM}). \tag{2.33}$$

Choosing $v = \varepsilon q^\varepsilon_{\rm MM}$ in the first equation of the MM reformulation yields

$$a_\perp(u^\varepsilon - u^0, u^\varepsilon - u^0) + \varepsilon a_\perp(q^\varepsilon_{\rm MM}, q^\varepsilon_{\rm MM}) = -\varepsilon a_\perp(u^0, q^\varepsilon_{\rm MM}). \tag{2.34}$$

Thanks to Lemma 3, $u^0 \in H^2(\Omega)$ and $a_\perp(u^0, q^\varepsilon_{\rm MM})$ can be integrated by parts. Following estimate can be obtained

$$-\varepsilon a_\perp(u^0, q^\varepsilon_{\rm MM}) \le C||u^0||_{H^2(\Omega)} \left(||q^\varepsilon_{\rm MM}||^2_{L_2(\Gamma_N)} + ||q^\varepsilon_{\rm MM}||^2_{L_2(\Omega)} \right) \tag{2.35}$$

The Poincaré-Wirtinger and trace inequalities finally allow to prove first the bound on $||\nabla_\parallel q^\varepsilon_{\rm MM}||_{L_2(\Omega)}$ and then the estimates on $||\nabla_\parallel u^\varepsilon||_{L_2(\Omega)}$ and $||\nabla_\perp u^\varepsilon - \nabla_\perp u^0||_{L_2(\Omega)}$. The complete proof can be found in [30].

Some H^2-regularity results for the unique solution $u^\varepsilon \in \mathcal{V}$ of the P-problem (2.28) can be proven for a simplified geometry: $\Omega := (0, L_x) \times (0, L_y)$ and $b = (0, 1)$ assumed constant and aligned in the y direction. Let us thus study the system

$$\begin{cases} -\frac{1}{\varepsilon}\partial_y(A_y \partial_y u^\varepsilon) - \partial_x(A_x \partial_x u^\varepsilon) = f\,, & \text{in } \Omega \\ \partial_y u^\varepsilon = 0\,, & \text{for } y = 0, L_y \\ u^\varepsilon = 0\,, & \text{for } x = 0, L_x\,. \end{cases} \tag{2.36}$$

Theorem 3 *Take $\Omega := (0, L_x) \times (0, L_y)$, $b = (0, 1)$, suppose that Hypothesis A is satisfied and moreover that $A_x = (A_\perp)_{11} \in W^{2,\infty}(\Omega)$ and $A_y = A_\parallel \in W^{2,\infty}(\Omega)$. Then u^ε, the unique solution of (2.36), belongs to $H^2(\Omega)$ and we have the estimates*

$$||\partial_x u^\varepsilon||^2_{L^2} + \frac{1}{\varepsilon^2}||\partial_y u^\varepsilon||^2_{L^2} \le C||f||^2_{L^2}\,, \tag{2.37}$$

$$||\partial_{xx} u^\varepsilon||^2_{L^2} + \frac{1}{\varepsilon}||\partial_{xy} u^\varepsilon||^2_{L^2} + \frac{1}{\varepsilon^2}||\partial_{yy} u^\varepsilon||^2_{L^2} \le C||f||^2_{L^2}\,, \tag{2.38}$$

with $C > 0$ a constant independent of ε and f.

We prove this theorem in [30]. Moreover, we show in [73] the matrix conditioning of the MM scheme is proportional to $1/h^4$.

1.4 Stabilized Micro-Macro scheme

The MM scheme relies on the elimination of the dominant part in the original problem by a suitable substitution $\nabla_\parallel \cdot A_\parallel \nabla_\parallel u^\varepsilon = \varepsilon \nabla_\parallel \cdot A_\parallel \nabla_\parallel q^\varepsilon_{\mathrm{MM}}$. This kind of substitution is of course not unique, in the MM scheme $q^\varepsilon_{\mathrm{MM}}$ is chosen such that its trace on the inflow part of the boundary is equal to zero. This limits the application of the scheme to only simple topologies of the anisotropy field: no closed field lines can be present in the domain. In order to overcome this limitation a stabilized scheme has been developed, inspired by stabilization methods for Stokes or Navier Stokes equations.

It is well known that the naive finite element discretization of the Stokes equation leads to a discrete system that does not fulfill the *inf-sup* condition. In order to overcome this difficulty several methods based on a suitable choice of discrete spaces have been proposed, for example nonconforming finite elements (Crouzeix-Raviart) or $\mathbb{P}_2$-$\mathbb{P}_1$ finite elements. A different and much simpler from practical point of view approach is the stabilization technique. The core idea of this method is to modify the $\nabla \cdot u = 0$ constraint in suitable manner so the bilinear form associated with the Stokes problem becomes coercive and the convergence order of the scheme is conserved. This technique has been successfully applied during my PhD thesis to Navier-Stokes equations for simulations of flows and pressure variation during the solidification process [83, 84, 85]. The method proposed for strongly anisotropic elliptic equations is inspired by penalty stabilization for the Stokes problem [15] and was first introduced for anisotropic diffusion in the parabolic context in joint publication with Maurizio Ottaviani [82] in order to simulate anisotropic heat transfer in the domain containing so called "magnetic islands" — regions where anisotropy follows closed field lines.

In the stabilized version the auxiliary variable is no longer required to have a zero trace on the inflow part of the boundary. In order to guarantee uniqueness of the solution, the second equation is modified. A small penalty term, proportional to the mesh size is added to the second equation yielding it invertible. This penalty introduces some small additional error to the system, the convergence rate of the method is however conserved.

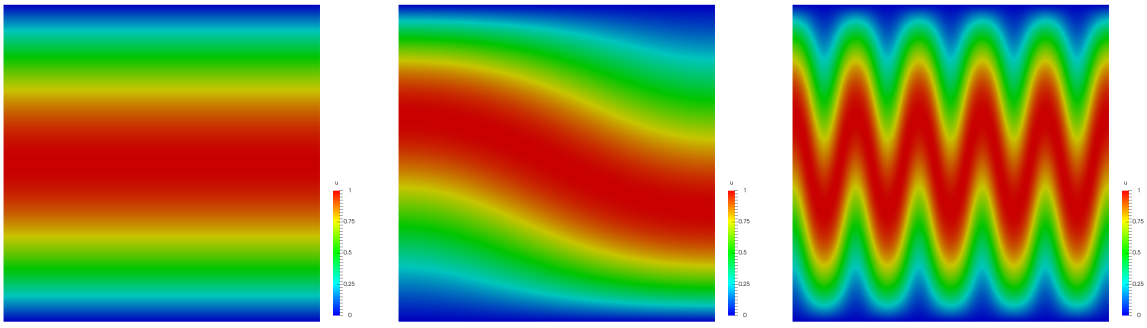
The stabilized method reads: find $(u^\varepsilon_{\mathrm{MMS}}, q^\varepsilon_{\mathrm{MMS}}) \in \mathcal{V} \times \mathcal{L}$ such that

$$(MMS) \quad \begin{cases} a_\perp(u^\varepsilon_{\mathrm{MMS}}, v) + a_\parallel(q^\varepsilon_{\mathrm{MMS}}, v) = (f, v), & \forall v \in \mathcal{V} \\ a_\parallel(u^\varepsilon_{\mathrm{MMS}}, w) - \varepsilon a_\parallel(q^\varepsilon_{\mathrm{MMS}}, w) - \sigma(q^\varepsilon_{\mathrm{MMS}}, w) = 0, & \forall w \in \mathcal{L}\,, \end{cases} \tag{2.39}$$

with the $\mathcal{L}$ being a Hilbert space of functions with no constraint on the inflow part of the boundary:

$$\mathcal{L} = \left\{\xi \in L^2(\Omega) \mid \nabla_\parallel \xi \in L^2(\Omega)\right\}, \quad (u, v)_{\mathcal{L}} := (u, w) + (\nabla_\parallel u, \nabla_\parallel w) \tag{2.40}$$

and σ is a small penalization parameter, independent of ε and mesh dependent. In order to preserve the convergence rate of the numerical method σ is set to h^k, where k is the order of the method.



$\alpha = 0$ $\alpha = 2,\ m = 1$ $\alpha = 2,\ m = 10$

Figure 2.1: The limit solution for different values of α and m.

1.5 Numerical tests: comparison of the DB, MM and MMS schemes

Let us now compare in 2D numerically the above presented schemes. Let us start from constructing a numerical test case. First, we choose a limit solution

$$u^0 = \sin\left(\pi y + \alpha(y^2 - y)\cos(m\pi x)\right), \tag{2.41}$$

where α is a numerical constant aimed to control the variations of b and m is an integer describing the number of oscillation periods of the limit solution in the computational domain. For $\alpha = 0$, the limit solution is constant in the direction of the X axis. The limit solutions for different values of α and m are shown in Figure 2.1. Since u^0 is a limit solution, it is constant along the b field lines. Therefore we can determine the b field using the following implication

$$\nabla_\parallel u^0 = 0 \quad \Rightarrow \quad b_x \frac{\partial u^0}{\partial x} + b_y \frac{\partial u^0}{\partial y} = 0\,, \tag{2.42}$$

which yields for example

$$b = \frac{B}{|B|}\,, \quad B = \begin{pmatrix} \alpha(2y-1)\cos(m\pi x) + \pi \\ m\pi\alpha(y^2 - y)\sin(m\pi x) \end{pmatrix} \quad . \tag{2.43}$$

Note that the field B, constructed in this way, satisfies $\mathrm{div} B = 0$, which is an important property in the framework of plasma simulations. Furthermore, we have $B \neq 0$ in the computational domain. Now, we choose u^ε to be a function that converges, as $\varepsilon \to 0$, to the limit solution u^0, for example

$$u^\varepsilon = \sin\left(\pi y + \alpha(y^2 - y)\cos(m\pi x)\right) + \varepsilon \cos(2\pi x)\sin(\pi y)\,. \tag{2.44}$$

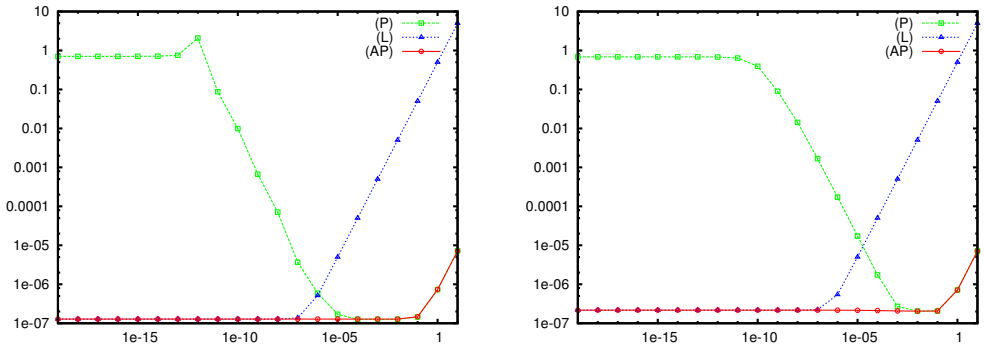
Finally, the force term is calculated, using the equation, i.e.

$$f = -\nabla_\perp \cdot (A_\perp \nabla_\perp u^\varepsilon) - \frac{1}{\varepsilon}\nabla_\parallel \cdot (A_\parallel \nabla_\parallel u^\varepsilon).$$

h	$\varepsilon = 1$, $\alpha = 0$		$\varepsilon = 10^{-10}$, $\alpha = 0$		$\varepsilon = 10^{-10}$, $\alpha = 2$	
	MM	MMS	MM	MMS	MM	MMS
0.1	5.39×10^{-3}	5.39×10^{-3}	1.19×10^{-3}	1.19×10^{-3}	2.81×10^{-3}	2.18×10^{-3}
0.05	6.97×10^{-4}	6.97×10^{-4}	1.49×10^{-4}	1.49×10^{-4}	3.16×10^{-4}	2.87×10^{-4}
0.025	8.79×10^{-5}	8.79×10^{-5}	1.86×10^{-5}	1.86×10^{-5}	3.77×10^{-5}	3.53×10^{-5}
0.0125	1.10×10^{-5}	1.10×10^{-5}	2.33×10^{-6}	2.33×10^{-6}	4.57×10^{-6}	4.31×10^{-6}
0.00625	1.38×10^{-6}	1.38×10^{-6}	2.91×10^{-7}	2.91×10^{-7}	5.60×10^{-7}	5.29×10^{-7}
0.003125	1.72×10^{-7}	1.72×10^{-7}	3.64×10^{-8}	3.64×10^{-8}	6.89×10^{-8}	6.52×10^{-8}
0.0015625	2.15×10^{-8}	2.15×10^{-8}	5.51×10^{-9}	4.78×10^{-9}	1.07×10^{-9}	8.05×10^{-9}

Table 2.1: Comparison of the L^2 relative precision $||u^\varepsilon - u_h^\varepsilon||_{L^2}/||u_h^\varepsilon||_{L^2}$ of MM and MMS schemes in both isotropic and anisotropic regimes for different mesh sizes and stabilization constant set to $\sigma = h^3$.

The numerical tests were performed with $m = 1$ and $\alpha = 0$ (constant direction of the anisotropy) or $\alpha = 2$ (mildly oscillating anisotropy direction). All methods give comparable precision regardless of the anisotropy strength ε. The relative L_2 error as a function of mesh size for $\varepsilon = 10^{-10}$ for MM and MMS schemes is shown on the Table 2.1. All three methods enjoy the theoretical order of convergence, which is three for the $\mathbb{Q}_2$-FEM. The error as a function of ε is plotted on Figure 2.2.



(a) constant anisotropy direction (b) variable anisotropy direction

Figure 2.2: Absolute L^2 errors for the DB scheme, the Limit problem and the singular perturbation method with anisotropy aligned with coordinate system ($\alpha = 0$) on the left and for variable anisotropy direction ($\alpha = 2$, $m = 1$) on the right. The error is plotted as a function of the parameter ε for mesh size $h = 0.005$

1.6 Regularized iterative scheme

The MM scheme in both variants, inflow and stabilized, has a serious drawback related to the matrix conditioning. Indeed, the matrix condition number scales as $1/h^4$ in the former case and as $1/h^{2+k}$ (k being the order of the method) in the latter. Also, no efficient

preconditioner for this kind of problem has been found so far. This makes iterative iterative methods (GMRES, CG) converge extremely slow and in practice the only efficient method to find a numerical solution is application of a direct LU decomposition. This method works very well when 2D problems are considered and allows to obtain a relative L_2 precision of the order of 10^{-10} on a laptop computer. However, if three dimensional simulations are required, the LU decomposition is no longer an option even for moderate size meshes (100^3). This is due to the memory footprint of the inverted matrix, much larger than in two dimensions for the matrices of the same size.

In order to overcome this inconvenience, I have proposed in [31] a regularized and iterative scheme, where the initial strongly anisotropic matrix $\mathbb{A}_\varepsilon$ is replaced by $\mathbb{A}_{\varepsilon_0}$, a matrix with the same direction of anisotropy, but with a much smaller ratio between coefficients. This only slightly anisotropic problem will be solved in a iterative manner with a suitable right hand side providing the convergence to the solution of the original problem. Typically ε_0 is taken as small, as possible. That is to say, ε_0 is chosen such that the initial problem is not yet singular and the numerical resolution does not suffer from locking. This value is of course b dependent as well as method dependent. Usually higher order methods are more locking-prone and smaller value of ε_0 can be employed. The joint work [31] with Fabrice Deluzet presents the derivation of this scheme together with theoretical justification on continuous level.

Let us now describe a two step iterative method to solve the singular perturbation problem (2.5). Let us consider $\tilde{\varepsilon}_0$ smaller than one but big enough so that the singular perturbation problem for $\varepsilon = \tilde{\varepsilon}_0$ is not yet singular nor the discretized system suffers from locking. Let us define $\varepsilon_0 = \max\{\tilde{\varepsilon}_0, \varepsilon\}$ so that ε_0 is never smaller than ε.

Let us now present a short derivation of the regularized iterative schemes. The idea behind the herein proposed scheme relies (as in the MM method) on the introduction of an additional variable that fulfils the following relation: $\varepsilon\nabla_\parallel \cdot A_\parallel \nabla_\parallel q = \nabla_\parallel \cdot A_\parallel \nabla_\parallel u$. This operation allows to eliminate the $\frac{1}{\varepsilon}$ terms from the equation. The system becomes:

$$\begin{cases} -\nabla_\parallel \cdot A_\parallel \nabla_\parallel q - \nabla_\perp \cdot A_\perp \nabla_\perp u = f, \\ -\nabla_\parallel \cdot A_\parallel \nabla_\parallel u = -\varepsilon \nabla_\parallel \cdot A_\parallel \nabla_\parallel q, \end{cases} \tag{2.45}$$

supplied with corresponding boundary conditions, which will be specified later. This system does not have unique solution as q is defined up to a function constant in the direction of the anisotropy; the second equation is not invertible without any restriction imposed on q. In the iterative scheme the above system is reformulated. The goal is to obtain a slightly anisotropic problem for u and an invertible problem for q. The first equation of (2.45) is used to express the perpendicular diffusion of u as a function of f and q. It is than rescaled and combined with the second yielding an anisotropic problem allowing to compute u if q is known. The first equation is used again to express the parallel diffusion of q as a function of f and u. The scheme is then rewritten in an iterative manner: initial value of q allows to compute a first approximation of u. This approximation is then used to update q, with the uniqueness of q assured by addition of the perpendicular diffusion to the both sides of the equation. The operation is repeated until convergence.

Let us now present the derivation of the scheme in more detail. Let us multiply the

first equation of (2.45) by ε_0 and add it to the second:

$$-\nabla \cdot \varepsilon_0 \mathbb{A}_{\varepsilon_0} \nabla u = \varepsilon_0 f + (\varepsilon_0 - \varepsilon)\nabla_\parallel \cdot A_\parallel \nabla_\parallel q, \tag{2.46}$$

allowing to find u uniquely if q is known. The next step is to decouple the problem and solve the two resulting equations in the iterative manner, finding first an approximation to u using q computed in the previous step, then recompute q and repeat until convergence:

$$\begin{cases} -\nabla \cdot \varepsilon_0 \mathbb{A}_{\varepsilon_0} \nabla u^{n+1} = \varepsilon_0 f + (\varepsilon_0 - \varepsilon)\nabla_\parallel \cdot A_\parallel \nabla_\parallel q^n, \\ -\nabla_\parallel \cdot A_\parallel \nabla_\parallel q^{n+1} = f + \nabla_\perp \cdot A_\perp \nabla_\perp u^{n+1}. \end{cases} \tag{2.47}$$

The second equation of this iterative scheme is not yet invertible. Let us now add the term $-\varepsilon_0 \nabla_\perp \cdot A_\perp \nabla_\perp q^{n+1}$ to the left hand side and subtract its equivalent for q^n from the right hand side. The resulting problem for q^{n+1} has a unique solution for given q^n and u^{n+1}. Finally, the two step iterative method is defined in the following way:

$$\begin{cases} -\nabla \cdot \varepsilon_0 \mathbb{A}_{\varepsilon_0} \nabla u^{n+1} = \varepsilon_0 f + (\varepsilon_0 - \varepsilon)\nabla_\parallel \cdot A_\parallel \nabla_\parallel q^n & \text{in } \Omega, \\ n \cdot \varepsilon_0 \mathbb{A}_{\varepsilon_0} \nabla u^{n+1} = -(\varepsilon_0 - \varepsilon) n \cdot \left(A_\parallel \nabla_\parallel q^n \right) & \text{on } \Gamma_N, \\ u^{n+1} = 0 & \text{on } \Gamma_D, \end{cases} \tag{2.48}$$

$$\begin{cases} -\nabla \cdot \varepsilon_0 \mathbb{A}_{\varepsilon_0} \nabla q^{n+1} = f + \nabla_\perp \cdot (A_\perp \nabla_\perp)(u^{n+1} - \varepsilon_0 q^n) & \text{in } \Omega, \\ n \cdot \varepsilon_0 \mathbb{A}_{\varepsilon_0} = -n \cdot \left(A_\perp \nabla_\perp \left(u^{n+1} - \varepsilon_0 q^n \right) \right) & \text{on } \Gamma_N, \\ q^{n+1} = 0 & \text{on } \Gamma_D, \end{cases} \tag{2.49}$$

where q^{n+1} is the intermediate step of the method, u^{n+1} is the approximation to u^ε and we choose $u^0 = q^0 = 0$. The variational formulation of the iterative scheme reads: find $(q^{n+1}, u^{n+1}) \in \mathcal{V}^2$ such that

$$a_\parallel(u^{n+1}, v) + \varepsilon_0 a_\perp(u^{n+1}, v) = \varepsilon_0 (f, v) - (\varepsilon_0 - \varepsilon) a_\parallel(q^n, v), \qquad \forall v \in \mathcal{V}, \tag{2.50}$$

$$a_\parallel(q^{n+1}, w) + \varepsilon_0 a_\perp(q^{n+1}, w) = (f, w) - a_\perp(u^{n+1} - \varepsilon_0 q^n, w), \qquad \forall w \in \mathcal{V}. \tag{2.51}$$

In this method, the original strongly anisotropic elliptic problem associated with the matrix $\mathbb{A}_\varepsilon$ is replaced by set of two only mildly anisotropic equations. Moreover, the matrix to be inverted in the first step of the iterative method is the same as in the second step, the only difference is in the right hand side of the equation. That is to say, the matrix has to be inverted only once, the rest of the iterative scheme is a fast triangular solve if a direct solver (LU/LLT) is employed. This method does note require any discretization of the space $\mathcal{G}$ of the functions constant in the direction of the anisotropy, which can be complicated for generic field b.

The next theorem shows, that the iterative scheme converges and that this limit solution solves the original Singular Perturbation problem.

Theorem 4 *Let the initial values $q^0, u^0 \in \mathcal{V}$. Than the series (q^n, u^n) defined by the iterative method (2.48-2.49) converges to a solution $(\bar{q}, \bar{u})$. Moreover, the stationary point $\bar{u}$ of the iterative scheme (2.48-2.49) solves uniquely the initial singular perturbation problem (2.5). Finally, the iterative scheme solves the correct limit problem (2.10) when $\varepsilon = 0$.*

We prove this theorem in [31]. The proof is based on the spectrum of the iteration matrix for the series auxiliary variable q^n. Let us first define the following operators $\Delta_{\varepsilon_0} = \Delta_\parallel + \varepsilon_0 \Delta_\perp$ with $\Delta_\parallel = \nabla_\parallel \cdot A_\parallel \nabla_\parallel$ and $\Delta_\perp = \nabla_\perp \cdot A_\perp \nabla_\perp$. Note, that Δ_{ε_0} is invertible for the boundary conditions supplied to the iterative scheme. Than the series q^n is defined as follows

$$q^{n+1} = A_I q^n + \Delta_{\varepsilon_0}^{-1} \Delta_\parallel \Delta_{\varepsilon_0}^{-1} f \tag{2.52}$$

with the iteration matrix A_I given by

$$A_I = Id - \frac{\varepsilon}{\varepsilon_0} \Delta_{\varepsilon_0}^{-1} \Delta_\parallel - \frac{\varepsilon_0 - \varepsilon}{\varepsilon_0} \left(\Delta_{\varepsilon_0}^{-1} \Delta_\parallel \right)^2 . \tag{2.53}$$

It is then argued that the spectrum of A_I is contained between 0 and 1, but the modes corresponding to the biggest eigenvalues are not present in q^n.

Remark 1 *The stationary point $(\bar{q}, \bar{u})$ is one of the solutions of the following problem: find $(q, u) \in \mathcal{V}^2$ such that*

$$\begin{cases} a_\perp(u, v) + a_\parallel(q, v) = (f, v) & \forall v \in \mathcal{V} \\ a_\parallel(u, w) - \varepsilon a_\parallel(q, w) = 0 & \forall w \in \mathcal{V} \end{cases} . \tag{2.54}$$

The above system does not admit a unique solution. Indeed, if (q, u) solves this problem, than $(q + g, u)$ also does for any function $g \in \mathcal{G}$. However u is unique. The stationary point $\bar{q}$ of the iterative process depends on the value of q^0.

Remark 2 *For any $n \geq 1$ and $q^0 = 0$ the function q^n is orthogonal to the space of functions constant in the direction of anisotropy with respect to the H^1 seminorm. Let us take $w \in \mathcal{G}$ in (2.51):*

$$\varepsilon_0 a_\perp(q^{n+1}, w) = (f, w) - a_\perp(u^{n+1}, w) + \varepsilon_0 a_\perp(q^n, w) \ , \ \forall w \in \mathcal{G}. \tag{2.55}$$

But $a_\perp(u^{n+1}, w) = (f, w)$ thanks to (2.50) with $v = w \in \mathcal{G}$. So finally

$$a_\perp(q^{n+1} - q^n, w) = 0 \ , \ \forall w \in \mathcal{G}, \tag{2.56}$$

so the difference between q^{n+1} and q^n is orthogonal to $\mathcal{G}$ with respect to the H^1 seminorm. Finally

$$a_\parallel(q^n, w) + a_\perp(q^n, w) = 0 \tag{2.57}$$

for all n since $q^0 = 0$. If $q^0 \neq 0$ than the projection of q^n on the space of functions constant in the direction of anisotropy does not change in the iterative process.

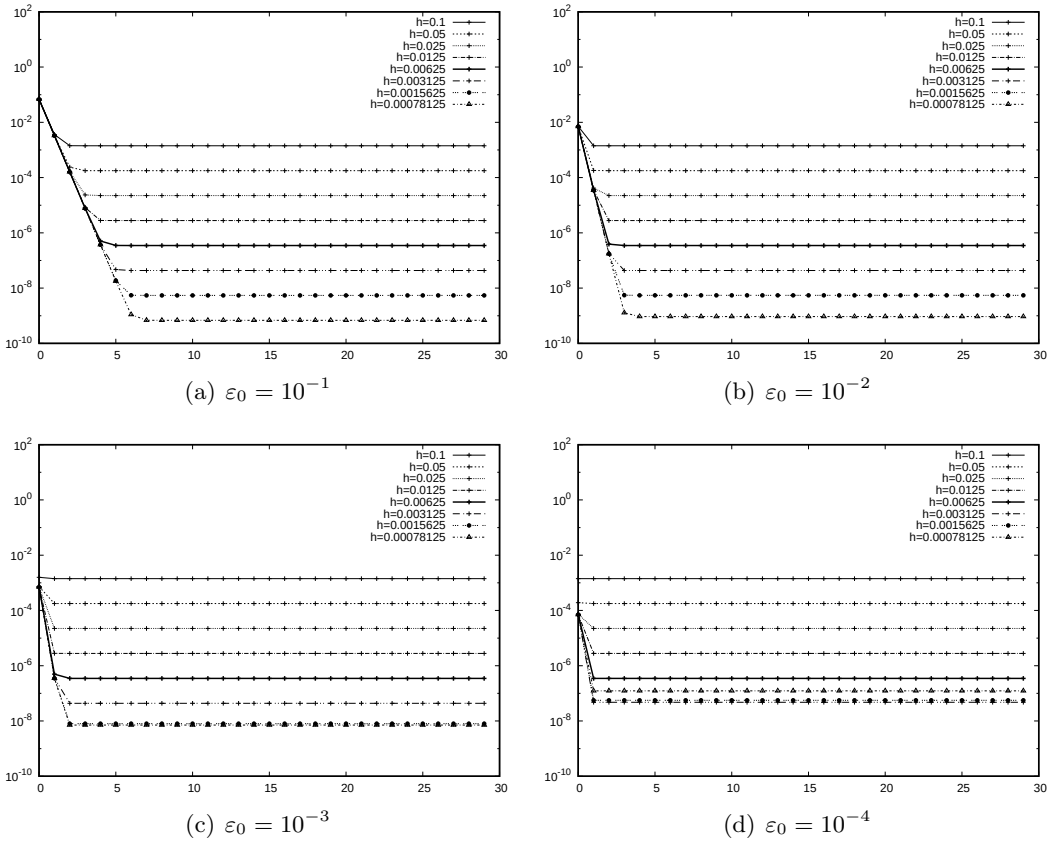
As matrix $-\nabla \cdot \varepsilon_0 \mathbb{A}_{\varepsilon_0} \nabla$ is invertible regardless on the anisotropy direction b, the method is well defined for all aniostropy topology. Moreover, the conditioning is not only ε independent, but it also scales as $1/\varepsilon_0 h^2$.

1.7 Numerical tests: regularized iterative scheme

Let us now study the numerical convergence of the iterative scheme. The analytical solution is again given by (2.44). The simulations were performed with $\mathbb{Q}_2$ FEM method and the aniostropy direction in the domain was constant ($\alpha = 0$), mildly variable ($\alpha = 2$, $m = 1$) and rapidly oscillating ($\alpha = 2$, $m = 10$). The tests were performed for mesh size varying from $h = 0.1$ (10 points in each direction) to $h = 0.00078125$ (1280 points in each direction) and for ε_0 equal 10^{-1}, 10^{-2}, 10^{-3} or 10^{-4}. The anisotropy strength was set to $\varepsilon = 10^{-20}$. The relative L_2 error is presented on Figures 2.3, 2.4 and 2.5. Let us first discuss the aligned anisotropy direction case. The iterative scheme converges relatively fast to the stationary point regardless of the value of ε_0. The slowest convergence is obtained for $\varepsilon_0 = 10^{-1}$ and the finest mesh (1280 points in each direction), where 10 iterations are required. The fastest convergence is obtained for the smallest value of ε_0, where merely 2 iterations are enough to reach the stationary point. The h of the stationary point convergence is optimal except for the finest meshes and $\varepsilon_0 = 10^{-4}$. In the latter case the matrix conditioning (proportional to $1/\varepsilon_0 h^2$) is to large and the numerical solution diverge when increasing the number of discretization points from 320 to 640 and than to 1280. Similarly the small divergence is present for 1280 points and $\varepsilon_0 = 10^{-3}$. Nevertheless, the iterative scheme is 10 to 100 more precise for the finest mesh than the MM scheme (except for $\varepsilon_0 = 10^{-4}$) — see Table 2.2. This lack of accuracy of the MM scheme can be explained by the large matrix conditioning of the second scheme, which is proportional to $1/h^4$. For the mildly variable anisotropy direction, the large value of ε_0 yield the slow convergence rate for fine meshes, but are rather fast and precise for coarse and intermediate meshes. For the fine mesh the best convergence rate is obtained for $\varepsilon_0 = 10^{-4}$, however for this choice the method diverges for the coarsest mesh due to the locking phenomenon. The precision of the converged solution is comparable with those of the MM scheme. Finally, for the rapidly oscillating anisotropy direction the method diverges for all values of ε_0 for the coarsest mesh of 10 points in each direction. This is not surprising since the number of points is far from sufficient to reflect the variations of the anisotropy direction. For fine meshes the best precision is obtained for $\varepsilon_0 = 10^{-3}$ and the best convergence rate for 10^{-4}. The stationary point exhibits a superconvergence in the tested range indicating that the locking affects the precision of the solution. The iterative scheme is less precise than the locking-prone MM scheme.

2 Parabolic equations

Let us now apply the ideas developed for elliptic problem to the parabolic equations: non linear temperature balance equation with open field lines and to the heat equation in the presence of closed field lines.



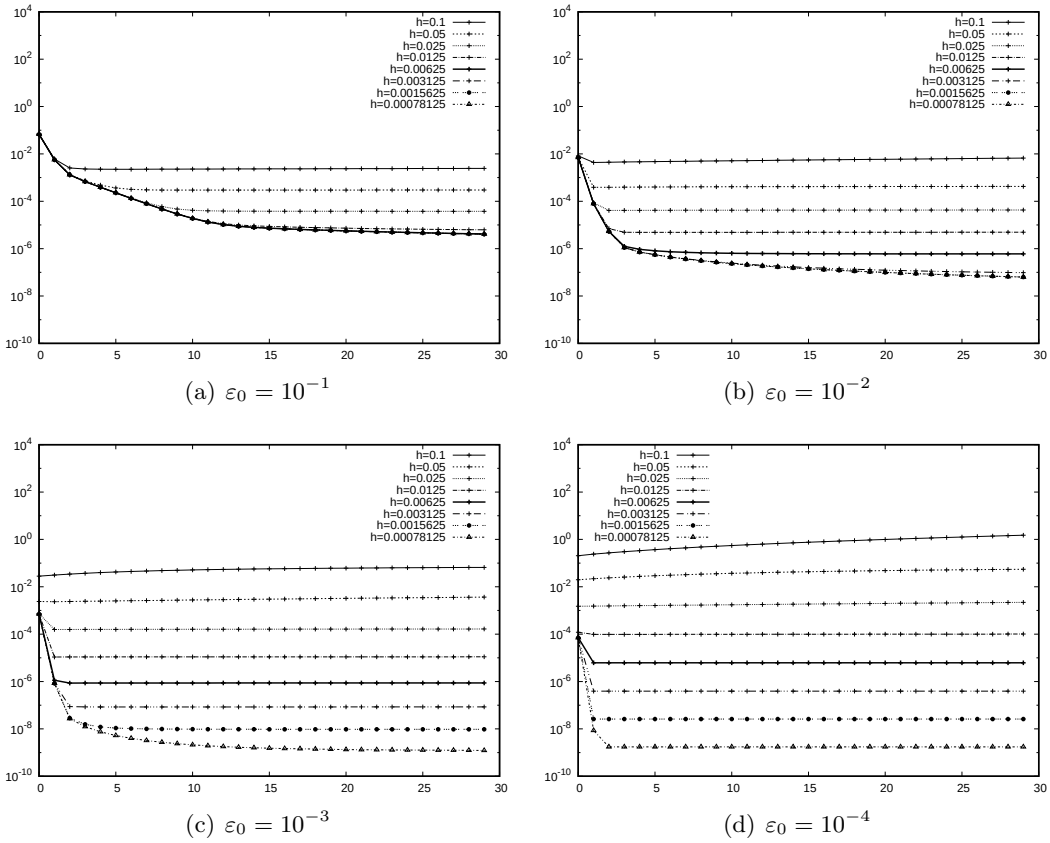
(a) $\varepsilon_0 = 10^{-1}$ (b) $\varepsilon_0 = 10^{-2}$

(c) $\varepsilon_0 = 10^{-3}$ (d) $\varepsilon_0 = 10^{-4}$

Figure 2.3: Relative L_2 error for anisotropy direction aligned with coordinate system ($\alpha = 0$) for different values of ε_0 and $\mathbb{Q}_2$-FEM.

	$\alpha = 0$				$\alpha = 2$, $m = 1$				$\alpha = 2$, $m = 10$			
method	L_2	#	H^1	#	L_2	#	H^1	#	L_2	#	H^1	#
MM	$9.68\ 10^{-8}$		$8.52\ 10^{-5}$		$1.47\ 10^{-9}$		$1.46\ 10^{-6}$		$4.31\ 10^{-7}$		$1.36\ 10^{-4}$	
$\varepsilon_0 = 10^{-1}$	$6.85\ 10^{-10}$	8	$8.98\ 10^{-7}$	6	$4.11\ 10^{-6}$	-	$2.86\ 10^{-5}$	-	$1.72\ 10^{-3}$	-	$4.84\ 10^{-3}$	-
$\varepsilon_0 = 10^{-2}$	$9.36\ 10^{-10}$	5	$8.98\ 10^{-7}$	3	$6.28\ 10^{-8}$	-	$1.74\ 10^{-6}$	-	$1.19\ 10^{-5}$	-	$1.52\ 10^{-4}$	-
$\varepsilon_0 = 10^{-3}$	$7.11\ 10^{-9}$	3	$8.98\ 10^{-7}$	3	$1.23\ 10^{-9}$	12	$1.42\ 10^{-6}$	4	$1.81\ 10^{-6}$	5	$1.36\ 10^{-4}$	3
$\varepsilon_0 = 10^{-4}$	$1.23\ 10^{-7}$	2	$9.07\ 10^{-7}$	2	$1.74\ 10^{-9}$	3	$1.43\ 10^{-6}$	2	$1.78\ 10^{-5}$	2	$1.38\ 10^{-4}$	2

Table 2.2: Comparison of the smallest relative numerical error for different values of ε_0 with the MMAP scheme for $\mathbb{Q}_2$-FEM on the 1280×1280 mesh together with number of iterations needed for convergence ("-" meaning that the method has not converged in 30 iterations).



(a) $\varepsilon_0 = 10^{-1}$ (b) $\varepsilon_0 = 10^{-2}$

(c) $\varepsilon_0 = 10^{-3}$ (d) $\varepsilon_0 = 10^{-4}$

Figure 2.4: Relative L_2 error for slowly varying anisotropy direction ($\alpha = 2$, $m = 1$) for different values of ε_0 and $\mathbb{Q}_2$-FEM.

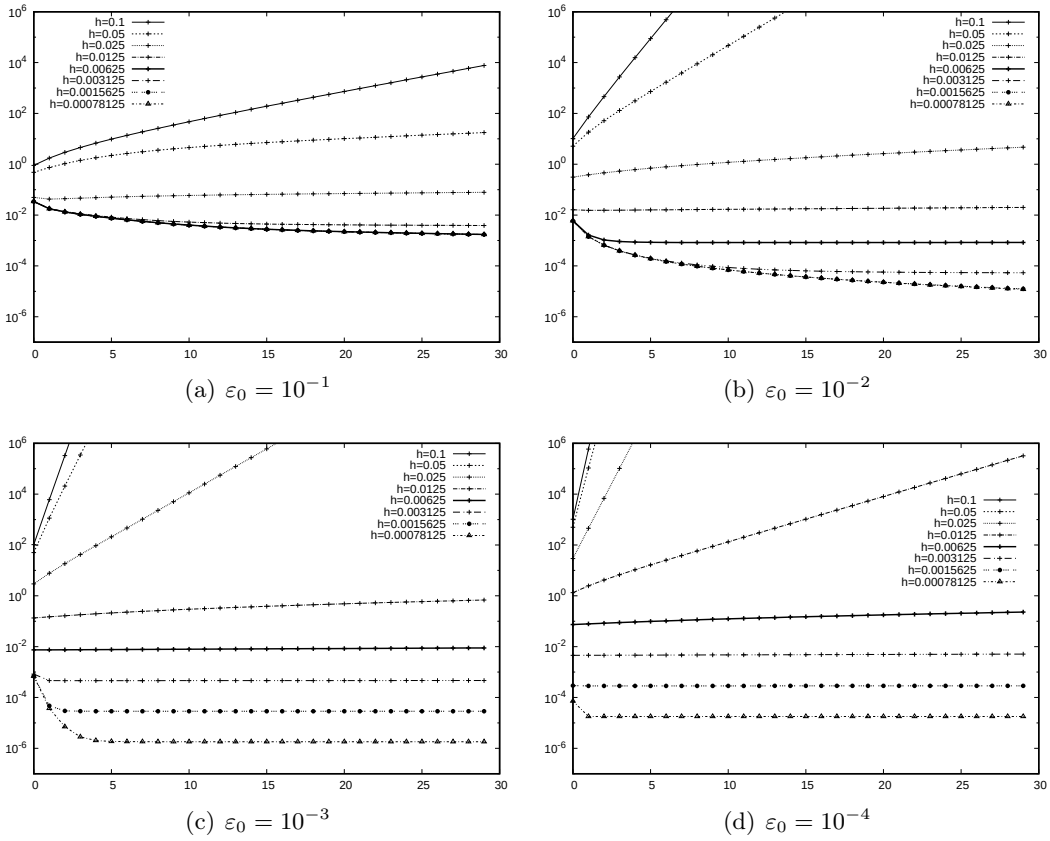
2.1 Non linear temperature balance equation

In this section we are interested in the heat diffusion in a magnetically confined plasma, expressed by a following nonlinear parabolic equation

$$\partial_t u - \nabla_\parallel \cdot (\kappa_\parallel(u)\nabla_\parallel u) - \nabla_\perp \cdot (\kappa_\perp \nabla_\perp u) = 0\,, \tag{2.58}$$

where the convection and turbulence effects are neglected. The diffusion in the direction perpendicular to the magnetic field lines is usually slow since the charged particles move mostly along the field lines. The corresponding diffusion coefficient $\kappa_\perp$ can be taken temperature independent. On the other hand, the coefficient describing the diffusion in the direction parallel to the magnetic field lines, $\kappa_\parallel$, is normally much larger and strongly temperature dependent and follows the Spitzer-Härm law $\kappa_\parallel(u) = \kappa_0 u^{5/2}$ [107]. Moreover, plasma temperatures are extremely high, so that this diffusion coefficient can become very big. Passing to non-dimensional variables, we shall write therefore the law for $\kappa_\parallel$ as

$$\kappa_\parallel(u) = \frac{1}{\varepsilon} u^{5/2}, \tag{2.59}$$



(a) $\varepsilon_0 = 10^{-1}$ (b) $\varepsilon_0 = 10^{-2}$

(c) $\varepsilon_0 = 10^{-3}$ (d) $\varepsilon_0 = 10^{-4}$

Figure 2.5: Relative L_2 error for rapidly varying anisotropy direction ($\alpha = 2$, $m = 10$) for different values of ε_0 and $\mathbb{Q}_2$-FEM.

where ε is a small parameter, $0 < \varepsilon \ll 1$. Due to the non linearity of parallel diffusion coefficient, the effective anisotropy in the system can vary strongly in the computational domain by several orders of magnitude. That is why an efficient and robust numerical scheme working for all values of ε is needed.

Let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ be a sufficiently smooth, bounded domain. Its boundary Γ can be decomposed as

$$\Gamma_{\parallel} := \{x \in \Gamma \ / \ b(x) \cdot n(x) = 0\},$$
$$\Gamma_{in} := \{x \in \Gamma \ / \ b(x) \cdot n(x) < 0\}, \quad \Gamma_{out} := \{x \in \Gamma \ / \ b(x) \cdot n(x) > 0\},$$

and $\Gamma_{\perp} = \Gamma_{in} \cup \Gamma_{out}$. The vector n is here the unit outward normal on Γ. We can now introduce the mathematical problem, we are interested to study. We are searching for the

particle (ions or electrons) temperature $u(t,x)$, solution of the evolution equation

$$(P)\quad \begin{cases} \partial_t u - \frac{1}{\varepsilon}\nabla_\parallel \cdot (A_\parallel u^{5/2}\nabla_\parallel u) - \nabla_\perp \cdot (A_\perp \nabla_\perp u) = 0\,, \quad \text{in} \quad [0,T]\times\Omega\,, \\ \frac{1}{\varepsilon} n_\parallel \cdot (A_\parallel u^{5/2}(t,\cdot)\nabla_\parallel u(t,\cdot)) + n_\perp \cdot (A_\perp \nabla_\perp u(t,\cdot)) = -\gamma\, u(t,\cdot)\,, \quad \text{on} \quad [0,T]\times\Gamma_\perp\,, \\ \nabla_\perp u(t,\cdot) = 0\,, \quad \text{on} \quad [0,T]\times\Gamma_\parallel\,, \\ u(0,\cdot) = u^0(\cdot)\,, \quad \text{in} \quad \Omega\,. \end{cases} \tag{2.60}$$

The coefficient γ is zero for electrons and $\gamma > 0$ for ions [97, 107]. The problem (2.60) describes the diffusion of an initial temperature u^0 within the time interval $[0,T]$ and its outflow through the boundary $\Gamma_\perp$. The parameter $0 < \varepsilon \ll 1$ can be very small and is responsible for the high anisotropy of the problem.

A numerical first order in time implicit Euler scheme based on a Micro-Macro decomposition for this problem was already proposed by Mentrelli and Negulescu in [76]. The nonlinearity was resolved by fixed point iterations. In [72] we have showed that if one needs a higher order discretization in time than a L-Stable method should be used, otherwise the Asymptotic Preserving property will be lost. The nonlinearity is treated by a linear extrapolation commonly used for Navier Stokes simulations similar to one already employed by myself in [83, 84, 85] for solidification problem.

Putting formally $\varepsilon = 0$ in (2.60) leads to the following ill-posed problem, admitting infinitely many solutions

$$\begin{cases} -\nabla_\parallel \cdot (A_\parallel u^{5/2}\nabla_\parallel u) = 0\,, \quad \text{in} \quad [0,T]\times\Omega\,, \\ n_\parallel \cdot (A_\parallel u^{5/2}(t,\cdot)\nabla_\parallel u(t,\cdot)) = 0\,, \quad \text{on} \quad [0,T]\times\Gamma_\perp\,, \\ \nabla_\perp u(t,\cdot) = 0\,, \quad \text{on} \quad [0,T]\times\Gamma_\parallel\,, \\ u(0,\cdot) = u^0(\cdot)\,, \quad \text{in} \quad \Omega\,. \end{cases} \tag{2.61}$$

Indeed, all functions which are constant along the field lines, meaning $\nabla_\parallel u \equiv 0$, and satisfying moreover the boundary condition on $\Gamma_\parallel$, are solutions of this problem. From a numerical point of view, this ill-posedness in the limit $\varepsilon \to 0$ can be detected by the fact, that trying to solve (2.60) with standard schemes leads to a linear system, which is very ill-conditioned for $0 < \varepsilon \ll 1$, in particular with a condition number of the order of $1/\varepsilon$.

The correct Limit problem can be found similarly as in the elliptic case. It is sufficient to multiply the initial formulation by a test function belonging to the space of functions constant in the direction of anisotropy and then integrate by parts. The resulting variational formulation is well posed in the limit $\varepsilon \to 0$ and reads: : find $u(t,\cdot) \in \mathcal{G}$ such that

$$(L)\ \ \langle \partial_t u(t,\cdot),\, v\rangle_{\mathcal{V}^*,\mathcal{V}} + \int_\Omega A_\perp \nabla_\perp u(t,\cdot)\cdot\nabla_\perp v\, dx + \gamma \int_{\Gamma_\perp} u(t,\cdot) v\, d\sigma = 0, \quad \forall v \in \mathcal{G} \tag{2.62}$$

for almost every $t \in (0,T)$.

Mathematical properties

The existence, uniqueness and positivity of the nonlinear heat equation can be shown for a more general equation

$$(P_m)\quad \begin{cases} \partial_t u - \nabla_\parallel \cdot (A_\parallel |u|^{m-1}\nabla_\parallel u) - \nabla_\perp \cdot (A_\perp \nabla_\perp u) = 0\,, \quad \text{in} \quad [0,T]\times\Omega\,, \\ A_\parallel |u|^{m-1} n_\parallel \cdot \nabla_\parallel u + A_\perp n_\perp \cdot \nabla_\perp u = -\gamma\, u\,, \quad \text{on} \quad [0,T]\times\Gamma_\perp\,, \\ \nabla_\perp u = 0\,, \quad \text{on} \quad [0,T]\times\Gamma_\parallel\,, \\ u(0,\cdot) = u^0(\cdot)\,, \quad \text{in} \quad \Omega\,, \end{cases} \tag{2.63}$$

for any $m \geq 1$. We obtain the particular case (2.60) by setting $m = 5/2+1$ and redefining $A_\parallel$ as $\frac{1}{\varepsilon}A_\parallel$ for any $\varepsilon > 0$. Equations of the type (2.63) are rather well studied in the literature. We refer to the classical works [40, 41, 70] as well as to the more modern literature on "The porous medium equation" as reviewed in [4, 106]. However, all these references normally treat only an isotropic version of the problem above, i.e. the nonlinearity of the type u^{m-1} is present in front of all the derivatives of u. An anisotropic equation of the form (2.63) is studied in [62], but only in the case $m < \frac{d+1}{d-1}$, so that the value of m pertinent to our application is not covered. Another feature of our setting, which is not sufficiently covered in the existing literature, is the prescription of Robin boundary conditions.

In order to show the existence, uniqueness and positivity the concept of weak solution of problem (2.63) is introduced.

Definition 1 ***(Weak solution)*** *Let $u^0 \in L^\infty(\Omega)$ and*

$$\begin{aligned} \mathcal{W} := \{ & u \in L^\infty(Q_\infty),\ \textit{such that } \forall\, T > 0 \\ & \nabla_\perp u \in L^2(Q_T)\,, \quad |u|^{m-1}\nabla_\parallel u \in L^2(Q_T)\,, \quad \partial_t u \in L^2(0,T;(H^1(\Omega))^*)\} \end{aligned}$$

where $Q_T := (0,T)\times\Omega$ denotes the time-space cylinder for any finite $T > 0$ or $T = \infty$. Then $u \in \mathcal{W}$ is called a weak solution of problem (2.63), if $u(0,\cdot) = u^0$ and if for all $T > 0$ one has

$$\begin{aligned} \int_0^T \langle \partial_t u(t,\cdot), \phi(t,\cdot)\rangle_{(H^1)^*,H^1}\, dt + \int_0^T \int_\Omega A_\parallel |u|^{m-1}\nabla_\parallel u \cdot \nabla_\parallel \phi\, dxdt \\ + \int_0^T \int_\Omega A_\perp \nabla_\perp u \cdot \nabla_\perp \phi\, dxdt + \gamma \int_0^T \int_{\Gamma_\perp} u\phi\, d\sigma\, dt = 0, \quad \forall \phi \in \mathcal{D} \end{aligned} \tag{2.64}$$

where $\mathcal{D} = L^2(0,T;H^1(\Omega))$.

In [72] we state and prove the following theorem.

Theorem 5 (Existence/Uniqueness/Positivity) *Let $m \geq 1$ and $u^0 \in L^\infty(\Omega)$ satisfy $0 < \beta \leq u^0 \leq M < \infty$ on Ω, for some $\beta > 0$. Under hypothesis 1, there exists a unique weak solution $u \in \mathcal{W}$ of (2.63), which satisfies $ce^{-Kt} \leq u \leq M$ a.e. on Q_∞, with some sufficiently small $c > 0$ and some sufficiently large $K > 0$.*

Semi-discretization in space

Let us write the variational formulation of the singular perturbation problem (2.60): find $u(t,\cdot) \in \mathcal{V} := H^1(\Omega)$ such that

$$(P) \qquad \langle \partial_t u(t,\cdot),\, v\rangle_{\mathcal{V}^*,\mathcal{V}} + \frac{1}{\varepsilon}\int_\Omega A_\parallel u^{5/2}\nabla_\parallel u(t,\cdot)\cdot\nabla_\parallel v\,dx \qquad (2.65)$$
$$+ \int_\Omega A_\perp \nabla_\perp u(t,\cdot)\cdot\nabla_\perp v\,dx - \int_{\Gamma_N} g_N(t,\cdot)v = 0, \quad \forall v \in \mathcal{V}$$

In order to solve numerically the initial singular perturbation problem, the method similar to the MM scheme is adopted. That is to say, the dominant part of the problem is replaced by an auxiliary variable $q \in \mathcal{L}_{in}$ such that $\nabla_\parallel\cdot\big(A_\parallel u^{5/2}(t,\cdot)\nabla_\parallel u(t,\cdot)\big) = \nabla_\parallel\cdot A_\parallel\nabla_\parallel q$. The reformulated problem, called in the sequel the Asymptotic-Preserving reformulation (AP-problem) reads: find $(u(t,\cdot), q(t,\cdot)) \in \mathcal{V}\times\mathcal{L}_{in}$, solution of

$$(AP) \quad \begin{cases} \langle \dfrac{\partial u}{\partial t},\, v\rangle_{\mathcal{V}^*,\mathcal{V}} + \displaystyle\int_\Omega (A_\perp\nabla_\perp u)\cdot\nabla_\perp v\,dx + \int_\Omega A_\parallel\nabla_\parallel q\cdot\nabla_\parallel v\,dx + \gamma\int_{\Gamma_\perp} uv\,ds = 0, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall v\in\mathcal{V} \\ \displaystyle\int_\Omega A_\parallel u^{5/2}\nabla_\parallel u\cdot\nabla_\parallel w\,dx - \varepsilon\int_\Omega A_\parallel\nabla_\parallel q\cdot\nabla_\parallel w\,dx = 0, \quad \forall w\in\mathcal{L}_{in}\,, \end{cases} \qquad (2.66)$$

System (2.66) is an equivalent reformulation (for fixed $\varepsilon > 0$) of the original P-problem (2.65). Putting now formally $\varepsilon = 0$ in (AP) leads to the well-posed limit problem

$$(L') \quad \begin{cases} \langle \dfrac{\partial u}{\partial t},\, v\rangle_{\mathcal{V}^*,\mathcal{V}} + \displaystyle\int_\Omega (A_\perp\nabla_\perp u)\cdot\nabla_\perp v\,dx + \int_\Omega A_\parallel\nabla_\parallel q\cdot\nabla_\parallel v\,dx + \gamma\int_{\Gamma_\perp} uv\,ds = 0, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall v\in\mathcal{V} \\ \displaystyle\int_\Omega A_\parallel u^{5/2}\nabla_\parallel u\cdot\nabla_\parallel w\,dx = 0, \quad \forall w\in\mathcal{L}_{in}\,, \end{cases} \qquad (2.67)$$

which is equivalent to problem (2.62). Note that q acts here as a Lagrange multiplier for the constraint $u \in \mathcal{G}$, which provides the uniqueness of the solution. Hence the AP-reformulation permits a continuous transition from the P-model to the L-model, which enables the uniform accuracy of the scheme with respect to ε.

Let us now choose a triangulation of the domain Ω with triangles or quadrangles of order h and introduce the finite element spaces $\mathcal{V}_h \subset \mathcal{V}$ and $\mathcal{L}_{in,h} \subset \mathcal{L}_{in}$ of type $\mathbb{P}_k$ or $\mathbb{Q}_k$ on this mesh. The finite element discretization of (2.66) writes then: find $(u_h, q_h) \in \mathcal{V}_h\times\mathcal{L}_{in,h}$ such that

$$(AP)_h \quad \begin{cases} \displaystyle\int_\Omega \frac{\partial u_h}{\partial t}v_h\,dx + \int_\Omega (A_\perp\nabla_\perp u_h)\cdot\nabla_\perp v_h\,dx + \int_\Omega A_\parallel\nabla_\parallel q_h\cdot\nabla_\parallel v_h\,dx + \gamma\int_{\Gamma_\perp} u_h v_h\,ds = 0, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall v_h\in\mathcal{V}_h \\ \displaystyle\int_\Omega A_\parallel u_h^{5/2}\nabla_\parallel u_h\cdot\nabla_\parallel w_h\,dx - \varepsilon\int_\Omega A_\parallel\nabla_\parallel q_h\cdot\nabla_\parallel w_h\,dx = 0, \quad \forall w\in\mathcal{L}_{in,h}\,. \end{cases} \qquad (2.68)$$

Remark that this system is continuous in time and also nonlinear, so that one has to develop now a procedure for the linearization and the discretization in time. This procedure has to be chosen carefully, such that the AP-property developed so far, is not destroyed. This is the aim of the next section.

Semi-discretization in time

Let us test three types of time discretization: the first implicit Euler scheme, the second order Crank-Nicolson method and the second order Diagonally Implicit L-stable Runge Kutta method.

Implicit Euler scheme Introducing the form

$$a_{\|nl}(\Psi,\Theta,\chi) := \int_\Omega A_\| \Psi^{5/2} \nabla_\| \Theta \cdot \nabla_\| \chi \, dx \,, \tag{2.69}$$

allows us to write the first order, implicit Euler method in the compact notation: Find $(u_h^{n+1}, q_h^{n+1}) \in \mathcal{V}_h \times \mathcal{L}_h$, solution of

$$(E_{AP}) \quad \left\{ \begin{array}{l} (u_h^{n+1}, v_h) + \tau \left(a_\perp(u_h^{n+1}, v_h) + a_\|(q_h^{n+1}, v_h) + \gamma \int_{\Gamma_\perp} u_h^{n+1} v_h \, ds \right) = (u_h^n, v_h) \\ a_{\|nl}(u_h^n, u_h^{n+1}, w_h) - \varepsilon a_\|(q_h^{n+1}, w_h) = 0 \,, \end{array} \right. , \tag{2.70}$$

where the non linear term $(u_h^{n+1})^{5/2}$ was replaced by a first order approximation in τ:

$$(u_h^{n+1})^{5/2} = (u_h^n + O(\tau))^{5/2} = (u_h^n)^{5/2} + O(\tau). \tag{2.71}$$

A slightly different first order AP-scheme was introduced in [76] for the resolution of the same temperature balance problem. There, the (P)-problem was firstly discretized in time (implicit Euler), then linearized by a fixed point mapping, and finally the AP reformulation applied. The numerical results obtained in [76] are similar to the present ones.

Crank-Nicolson scheme To construct a scheme, which is second order in time, one can come to the idea to employ the Crank-Nicolson scheme: Find $(u_h^{n+1}, q_h^{n+1}) \in \mathcal{V}_h \times \mathcal{L}_h$, solution of

$$\left\{ \begin{array}{l} (u_h^{n+1}, v_h) + \tau \left(a_\perp(u_h^{n+1/2}, v_h) + a_\|(q_h^{n+1}, v_h) + \gamma \int_{\Gamma_\perp} u_h^{n+1/2} v_h \, ds \right) = (u_h^n, v_h) \\ a_{\|nl}(u_h^{n+1/2}, u_h^{n+1/2}, w_h) - \varepsilon a_\|(q_h^{n+1}, w_h) = 0 \,. \end{array} \right. \tag{2.72}$$

As one can observe, we have to deal for each fixed n, with a nonlinear equation. In the linear terms, one can set $u_h^{n+1/2} = \frac{1}{2}\left(u_h^{n+1} + u_h^n\right)$. To linearize the term $a_{\|nl}(u_h^{n+1/2}, u_h^{n+1/2}, w_h)$

however, we shall use the standard linear extrapolation method. In other words, the non-linearity in this last formula, $(u_h^{n+1/2})^{5/2}$, will be replaced by a linearized second order approximation in τ:

$$(u_h^{n+1/2})^{5/2} = \left(u_h^n + \frac{1}{2}\left(u_h^n - u_h^{n-1}\right) + O(\tau^2)\right)^{5/2} = \left(u_h^n + \frac{1}{2}\left(u_h^n - u_h^{n-1}\right)\right)^{5/2} + O(\tau^2). \tag{2.73}$$

The resulting linear system reads finally: Find $(u_h^{n+1}, q_h^{n+1}) \in \mathcal{V}_h \times \mathcal{L}_h$, solution of

$$(CN_{AP}) \quad \begin{cases} (u_h^{n+1}, v_h) + \frac{\tau}{2}\left(a_\perp(u_h^{n+1}, v_h) + \gamma \int_{\Gamma_\perp} u_h^{n+1} v_h \, ds\right) + \tau a_\parallel(q_h^{n+1}, v_h) \\ \qquad = (u_h^n, v_h) - \frac{\tau}{2}\left(a_\perp(u_h^n, v_h) + \gamma \int_{\Gamma_\perp} u_h^n v_h \, ds\right), \\ \frac{1}{2} a_{\parallel nl}\left(\frac{1}{2}\left(3u_h^n - u_h^{n-1}\right), u_h^{n+1}, w_h\right) - \varepsilon a_\parallel(q_h^{n+1}, w_h) \\ \qquad = -\frac{1}{2} a_{\parallel nl}\left(\frac{1}{2}\left(3u_h^n - u_h^{n-1}\right), u_h^n, w_h\right). \end{cases} \tag{2.74}$$

Unfortunately this method is not Asymptotic-Preserving. For small values of ε one expects that the solution will immediately fall into the space of functions almost constant in the direction of the anisotropy, no matter what initial condition was imposed. In the case of the Crank-Nicolson scheme for large time steps compared to $\varepsilon/(u_h^n)^{5/2}$, the second equation in (2.74) will constrain the numerical solution to oscillate if the initial condition is not already in the suitable space. This requires the restrictive choice of a time step of the order of $\varepsilon/(u_h^n)^{5/2}$, which yields the method inapplicable in general cases. In other words, the Crank-Nicolson scheme is unable to model diffusion processes for large Δt, due to the inadequate approximation of the damping processes.

This is closely related to the notion of A-stability and L-stability for stiff equations (see for example [59]). Let us consider a test problem $y' = ky$. Runge-Kutta methods applied to this problem can be expressed as $y^{n+1} = \phi(k\tau)y^n$ with ϕ being a stability function. A method is A-stable when a numerical solution to the test problem approaches 0 as $t \to \infty$ for all $k > 0$. A method is L-stable if it is A-stable and if its stability function goes to zero as the time step goes to infinity. The AP property is strongly related to the L-stability. Indeed, if a method is A-stable and not L-stable, the numerical solution of the test problem still converges to zero for $k > 0$, but rapidly decaying components are damped very slowly since the stability function is not zero and therefore a scheme cannot be Asymptotic Preserving. The Crank-Nicolson scheme is A-stable but not L-stable and thus not suitable for small values of ε and large values of Δt.

L-stable Runge-Kutta method As we are interested in an AP-scheme, which is second order accurate in time, we propose now a two stage Diagonally Implicit Runge-Kutta (DIRK) second order scheme, which does not suffer from the limitations of the Crank-Nicolson discretization. The scheme is developed according to the following Butcher's

diagram:

$$\begin{array}{c|cc} \lambda & \lambda & 0 \\ 1 & 1-\lambda & \lambda \\ \hline & 1-\lambda & \lambda \end{array} \tag{2.75}$$

with $\lambda = 1 - \frac{1}{\sqrt{2}}$.

Remark 3 *(Butcher's diagram) The coefficients of the s-stage Runge-Kutta method are usually displayed in a Butcher's diagram:*

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array} . \tag{2.76}$$

Applying this method to approximate to following problem

$$\frac{\partial u}{\partial t} = Lu + f(t)\,, \tag{2.77}$$

reads: For given u^n, being an approximation of $u(t_n)$, the u^{n+1} is determined accordingly to:

$$u_i = u^n + \tau \sum_{j=1}^{s} a_{ij}(Lu_j + f(t + c_j\tau)), \tag{2.78}$$

$$u^{n+1} = u^n + \tau \sum_{j=1}^{s} b_j u_j. \tag{2.79}$$

If $b_j = a_{sj}$ for $j = 1, \ldots, s$ than $u^{n+1} = u_s$.

The scheme (2.75) is known to be L-stable, thus providing the Asymptotic Preserving property. The scheme writes: Find $(u_h^{n+1}, q_h^{n+1}) \in \mathcal{V}_h \times \mathcal{L}_h$, solution of

$$\left\{ \begin{array}{l} (u_{1,h}^{n+1}, v_h) + \tau\lambda \left(a_\perp(u_{1,h}^{n+1}, v_h) + \gamma \int_{\Gamma_\perp} u_{1,h}^{n+1} v_h \, ds + a_\parallel(q_{1,h}^{n+1}, v_h) \right) \\ \qquad\qquad = (u_h^n, v_h) \\ \\ a_{\parallel nl}\left(u_h^n + \lambda(u_h^n - u_h^{n-1}), u_{1,h}^{n+1}, w_h \right) - \varepsilon a_\parallel(q_{1,h}^{n+1}, w_h) = 0 \end{array} \right.$$

$$(RK_{AP}) \qquad \left\{ \begin{array}{l} (u_{2,h}^{n+1}, v_h) + \tau\lambda \left(a_\perp(u_{2,h}^{n+1}, v_h) + \gamma \int_{\Gamma_\perp} u_{2,h}^{n+1} v_h \, ds + a_\parallel(q_{2,h}^{n+1}, v_h) \right) \\ \qquad\qquad = (u_h^n, v_h) + \frac{1-\lambda}{\lambda} \left(u_{1,h}^{n+1} - u_h^n, v_h \right) \\ \\ a_{\parallel nl}\left(u_h^n + (u_h^n - u_h^{n-1}), u_{2,h}^{n+1}, w_h \right) - \varepsilon a_\parallel(q_{2,h}^{n+1}, w_h) = 0 \end{array} \right. \tag{2.80}$$

$$u_h^{n+1} = u_{2,h}^{n+1}\,, \qquad q_h^{n+1} = q_{2,h}^{n+1}\,,$$

h	L^2-error $\varepsilon = 1$			h	L^2-error $\varepsilon = 10^{-10}$		
	P	E_{AP}	RK_{AP}		P	E_{AP}	RK_{AP}
0.1	1.60×10^{-3}	1.60×10^{-3}	1.60×10^{-3}	0.1	7.3×10^{-1}	1.47×10^{-3}	1.47×10^{-3}
0.05	2.02×10^{-4}	2.02×10^{-4}	2.02×10^{-4}	0.05	7.3×10^{-1}	2.04×10^{-4}	2.04×10^{-4}
0.025	2.55×10^{-5}	2.55×10^{-5}	2.55×10^{-5}	0.025	7.3×10^{-1}	2.65×10^{-5}	2.65×10^{-5}
0.0125	3.2×10^{-6}	3.2×10^{-6}	3.2×10^{-6}	0.0125	4.9×10^{-1}	3.3×10^{-6}	3.3×10^{-6}
0.00625	4.0×10^{-7}	4.0×10^{-7}	4.0×10^{-7}	0.00625	1.04×10^{-1}	4.2×10^{-7}	4.2×10^{-7}

Table 2.3: The absolute error of u in the L^2-norm for different mesh sizes and $\varepsilon = 1$ or $\varepsilon = 10^{-10}$, using the singular perturbation scheme $(P)_{h\tau}$ and the two proposed AP-schemes for a time step of $\tau = 10^{-6}s$ and at instant $t = 10^{-4}$, with $T_m = 1$.

with $u_{1,h}^{n+1}$ (respectively $u_{2,h}^{n+1}$) being the solution of the first (respectively second) stage of the Runge-Kutta method. The terms $u_h^n + \lambda(u_h^n - u_h^{n-1})$ and $u_h^n + (u_h^n - u_h^{n-1})$ are respectively the second order time-approximations of $u_h(t + \lambda\tau)$ and $u_h(t + \tau)$, used to linearize the problem.

For each time step we have therefore to assemble and solve two linearized problems. This method is two times slower than the Crank-Nicolson scheme, with the advantage however of maintaining the AP-property of the scheme, advantage which is crucial for $0 < \varepsilon \ll 1$.

Numerical results

The scheme was first tested on a case with known analytical solution, given by

$$u^0 = \left(\cos\left(\pi y + \alpha(y^2 - y)\cos(\pi x)\right) + 4\right) T_m e^{-t} \tag{2.81}$$

$$q = (u^0)^{-3/2} \sin(3\pi x)/3\pi \tag{2.82}$$

$$u^\varepsilon = u^0 + \varepsilon q. \tag{2.83}$$

In order to show numerically that the scheme has a correct order of convergence in space, we have fixed a time step to a small value such that the time discretization error is much smaller than the space discretization error. The simulations were than performed for 100 time steps and the numerical solution was compared with the analytical one for different mesh sizes. The results for $\varepsilon = 1$ and $\varepsilon = 10^{-10}$ are summarized in the Table 2.3 showing that the schemes enjoy the theoretical order of convergence.

Next, we have chosen small mesh size such that the space discretization error was smaller than the time discretization. Than we have performed numerical simulations until the fixed final time was reached. The Table 2.4 presents the L_2 error obtained at the end of the simulations for $\varepsilon = 1$ and $\varepsilon = 10^{-4}$. The correct time convergence rate (one for the implicit Euler, two for the DIRK scheme) is observed.

Finally the evolution of the initial Gaussian peak was investigated. The initial condition is given by

$$u(t = 0) = \frac{T_m}{2}\left(1 + e^{-50(x-0.5)^2 - 50(y-0.5)^2}\right), \tag{2.84}$$

τ	L^2-error $\varepsilon = 1$			τ	L^2-error $\varepsilon = 10^{-10}$		
	P	E_{AP}	RK_{AP}		P	E_{AP}	RK_{AP}
0.1	1.57×10^{-2}	1.57×10^{-2}	2.52×10^{-3}	0.1	6.14×10^{-1}	1.57×10^{-2}	2.90×10^{-4}
0.05	8.28×10^{-3}	8.28×10^{-3}	1.93×10^{-4}	0.05	6.30×10^{-1}	8.22×10^{-3}	7.21×10^{-5}
0.025	4.25×10^{-3}	4.25×10^{-3}	2.62×10^{-5}	0.025	6.92×10^{-1}	4.22×10^{-3}	1.80×10^{-5}
0.0125	2.37×10^{-3}	2.37×10^{-3}	6.54×10^{-6}	0.0125	7.08×10^{-1}	2.36×10^{-3}	4.91×10^{-6}
0.00625	1.08×10^{-3}	1.08×10^{-3}	1.50×10^{-6}	0.00625	7.26×10^{-1}	1.08×10^{-3}	1.15×10^{-6}
0.003125	5.44×10^{-4}	5.44×10^{-4}	4.08×10^{-7}	0.003125	7.42×10^{-1}	5.40×10^{-4}	3.43×10^{-7}
0.0015625	2.76×10^{-4}	2.76×10^{-4}	2.07×10^{-7}	0.0015625	6.42×10^{-1}	2.74×10^{-4}	2.05×10^{-7}

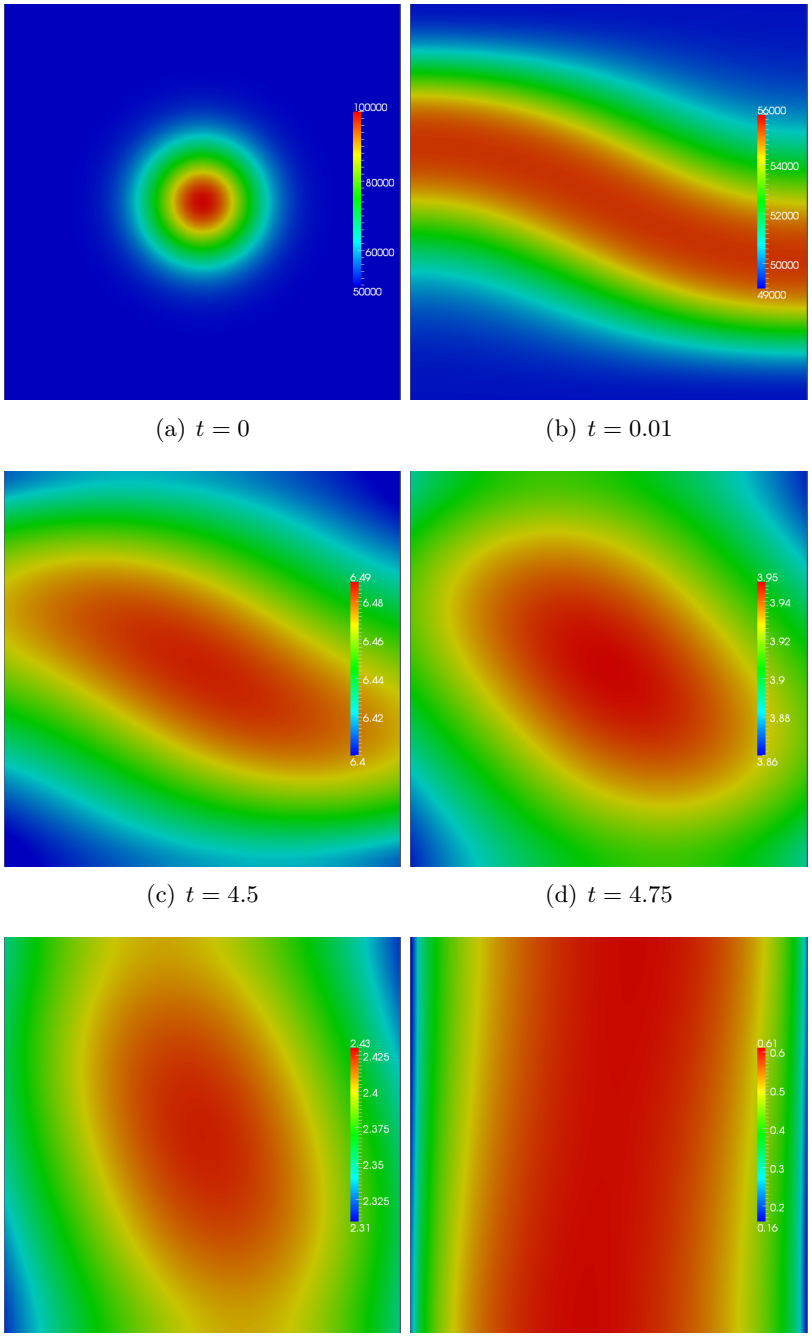
Table 2.4: The absolute error of u in the L^2-norm for different time step using the singular perturbation scheme $(P)_{h\tau}$ and two proposed AP-schemes for mesh size 200×200 at time $t = 0.1s$ with $T_m = 1$.

where $T_m = 10^5\, K$ is the maximal temperature in the domain and the anisotropy direction is given as in the previous tests. We perform numerical experiments with the choice of $\varepsilon = 1$. Note that a strong anisotropy is still present in the system due to large value of $u(t = 0)$. In fact one could rescale the problem and look for $\tilde{u}_h = u_h/T_m$. In this case the initial condition $\tilde{u}(t = 0)$ would be of the order 1 and the rescaled anisotropy strength would be $\tilde{\varepsilon} = T_m^{-5/2} = 10^{-12.5}$. We choose the time step $\tau = 0.01$ and perform numerical simulations on a fixed 50×50 grid with the final time set to $15s$. The results are presented on Figure 2.6. The time step is big compared to the time scale induced by the initial condition. After just one iteration of the algorithm the numerical solution is almost constant in the direction of the anisotropy. As temperature decays during the simulations, the anisotropy becomes weaker and finally, when temperature drops below 1, its direction inverses and the numerical solution aligns with the direction perpendicular to b.

2.2 Asymptotic Preserving scheme for arbitrary anisotropy direction

We are interested in a resolution of an anisotropic, two or three dimensional and this time linear heat problem defined on a domain Ω. Let the anisotropy direction be given by a smooth and normalized vector field b, $|b| = 1$ and let the computational domain Ω be a bounded and sufficiently smooth two or three dimensional subset of $\mathbb{R}^d$ with $d = 2, 3$. The domain Ω is equipped with a boundary Γ, which is decomposed accordingly to the boundary conditions into two parts: Γ_D and $\Gamma_N = \partial\Omega \backslash \Gamma_D$ with the Dirichlet and Neumann boundary condition imposed respectively. The boundaries Γ_N and Γ_D this time are not linked to the anisotropy.

The mathematical problem we are interested in reads: find the particle temperature



(a) $t = 0$ (b) $t = 0.01$

(c) $t = 4.5$ (d) $t = 4.75$

(e) $t = 5$ (f) $t = 6$

Figure 2.6: Numerical solution at different time steps for the Gaussian peak experiment, for $T_m = 10^5$ and $\varepsilon = 1$. Time step is $\tau = 0.01s$ and a mesh size of 50×50.

$u(t,x)$, solution of the evolution equation

$$(PH)\quad \begin{cases} \partial_t u - \frac{1}{\varepsilon}\nabla_\parallel \cdot (A_\parallel \nabla_\parallel u) - \nabla_\perp \cdot (A_\perp \nabla_\perp u) = 0\,, \quad \text{in} \quad [0,T]\times\Omega\,, \\ \frac{1}{\varepsilon} n_\parallel \cdot (A_\parallel \nabla_\parallel u(t,\cdot)) + n_\perp \cdot (A_\perp \nabla_\perp u(t,\cdot)) = g_N(t,\cdot)\,, \quad \text{on} \quad [0,T]\times\Gamma_N\,, \\ u(t,\cdot) = g_D(t,\cdot)\,, \quad \text{on} \quad [0,T]\times\Gamma_D\,, \\ u(0,\cdot) = u^0(\cdot)\,, \quad \text{in} \quad \Omega\,. \end{cases}$$

Again, putting formally $\varepsilon = 0$ in (PH) leads to the following reduced problem

$$\begin{cases} -\nabla_\parallel \cdot (A_\parallel \nabla_\parallel u) = 0\,, \quad \text{in} \quad [0,T]\times\Omega\,, \\ n_\parallel \cdot (A_\parallel \nabla_\parallel u(t,\cdot)) = 0\,, \quad \text{on} \quad [0,T]\times\Gamma_N\,, \\ u(t,\cdot) = g_D(t,\cdot)\,, \quad \text{on} \quad [0,T]\times\Gamma_D\,, \\ u(0,\cdot) = u^0(\cdot)\,, \quad \text{in} \quad \Omega \end{cases},$$

which may be ill-posed, depending on boundary conditions and the anisotropy field b. For example, when some field lines of b are closed in Ω the system would admit infinitely many solutions as any function constant along the closed lines of b (meaning $\nabla_\parallel u \equiv 0$) and satisfying the boundary conditions, solves the reduced problem. The same problem occurs when the field lines are open but do not pass through a boundary supplied with the Dirichlet conditions. This argument applies also to the case, where periodic (instead of Neumann) boundary conditions are imposed on Γ_N. This is the case in the numerical simulations related to the tokamak fusion plasma, where the computational domain is topologically equivalent to a torus. Numerical discretization of the original (PH) problem in the limit $\varepsilon \to 0$ can therefore lead to a very badly conditioned linear systems. In fact, the condition number is proportional to $1/\varepsilon$.

This problem was addressed in joint publication with Maurizio Ottaviani [82] where we were interested in developing a method capable of simulating more complicated aniostropy field topologies than open field lines. This is particularly important in numerical simulations related to plasma physics, where regions of closed field line may appear in a domain as a result of magnetic field reconnection. In this publication we introduce for a first time the stabilized AP scheme based on Micro-Macro decomposition presented before for the elliptic problem.

2.3 Semi-discretization in space

Let us consider for simplicity a homogeneous Dirichlet case $g_D = 0$, the non homogeneous case being a simple extension. Let us write the variational formulation of the singular perturbation problem (PH): find $u(t,\cdot) \in \mathcal{V} := H^1(\Omega)$ such that

$$\begin{aligned} \langle \partial_t u(t,\cdot),\, v\rangle_{\mathcal{V}^*,\mathcal{V}} + \frac{1}{\varepsilon}\int_\Omega A_\parallel \nabla_\parallel u(t,\cdot)\cdot\nabla_\parallel v\, dx \\ + \int_\Omega A_\perp \nabla_\perp u(t,\cdot)\cdot\nabla_\perp v\, dx - \int_{\Gamma_N} g_N(t,\cdot) v = 0, \quad \forall v \in \mathcal{V} \end{aligned} \tag{2.85}$$

for almost every $t \in (0,T)$. When dealing with non homogeneous Dirichlet boundary the variational formulation is written for a function $w = u - \tilde{g}$, where $\tilde{g}$ is a suitably chosen function such that $\tilde{g} \in \mathcal{V}$ and $\tilde{g}|_{\Gamma_D} = g_D$ and hence $w|_{\Gamma_D} = 0$. The weak formulation (2.85) is then obtained with additional terms including $\tilde{g}$ and test functions v added to the right hand side.

As already discussed in the previous section, taking the formal limit of $\varepsilon \to 0$ leads to an ill-posed problem:

$$\int_\Omega A_\parallel \nabla_\parallel u(t,\cdot) \cdot \nabla_\parallel v \, dx = 0$$

with any function belonging to the vector space of functions constant in the anisotropy direction:

$$\mathcal{G} := \{p \in \mathcal{V} \mid \nabla_\parallel p = 0 \text{ in } \Omega\}$$

being a solution.

A correct Limit problem can be established by seeking a solution in the subspace $\mathcal{G}$ instead of $\mathcal{V}$. In this case the leading order term (containing the parallel gradient) is eliminated from the equation and we are left with the following Limit problem: find $u(t,\cdot) \in \mathcal{G}$ such that

$$(L)\ \ \langle \partial_t u(t,\cdot),\, v\rangle_{\mathcal{V}^*,\mathcal{V}} + \int_\Omega A_\perp \nabla_\perp u(t,\cdot) \cdot \nabla_\perp v \, dx - \int_{\Gamma_N} g_N(t,\cdot) v = 0, \quad \forall v \in \mathcal{G} \tag{2.86}$$

for almost every $t \in (0,T)$.

Let us now again apply the MM strategy in order to eliminate the terms of order $O(1/\varepsilon)$ from the variational formulation. The uniqueness of q is provided by setting $q = 0$ on a part of the boundary Γ_{in} defined by

$$\Gamma_{in} := \{x \in \Gamma \ / \ b(x) \cdot n(x) < 0\},$$

i.e. the part of the boundary, where the field lines enter the domain. The following reformulated problem, called in the sequel the Asymptotic-Preserving reformulation (AP-problem) is proposed: find $(u(t,\cdot), q(t,\cdot)) \in \mathcal{V} \times \mathcal{L}_{in}$, solution of

$$(AP)\ \ \begin{cases} \langle \dfrac{\partial u}{\partial t},\, v\rangle_{\mathcal{V}^*,\mathcal{V}} + \displaystyle\int_\Omega (A_\perp \nabla_\perp u) \cdot \nabla_\perp v \, dx + \int_\Omega A_\parallel \nabla_\parallel q \cdot \nabla_\parallel v \, dx - \int_{\Gamma_N} g_N v \, ds = 0, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall v \in \mathcal{V} \\ \displaystyle\int_\Omega A_\parallel \nabla_\parallel u \cdot \nabla_\parallel w \, dx - \varepsilon \int_\Omega A_\parallel \nabla_\parallel q \cdot \nabla_\parallel w \, dx = 0, \quad \forall w \in \mathcal{L}_{in}\,, \end{cases} \tag{2.87}$$

where

$$\mathcal{L}_{in} := \{q \in L^2(\Omega) \ | \nabla_\parallel q \in L^2(\Omega) \text{ and } q|_{\Gamma_{in}} = 0\}.$$

The AP-problem is equivalent for fixed $\varepsilon > 0$ to the original P-problem (2.85). Moreover, putting formally $\varepsilon = 0$ in (AP) leads to a well-posed problem

$$(L') \quad \begin{cases} \langle \dfrac{\partial u}{\partial t}, v \rangle_{\mathcal{V}^*,\mathcal{V}} + \displaystyle\int_\Omega (A_\perp \nabla_\perp u) \cdot \nabla_\perp v \, dx + \int_\Omega A_\parallel \nabla_\parallel q \cdot \nabla_\parallel v \, dx - \int_{\Gamma_N} g_N v \, ds = 0, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall v \in \mathcal{V} \\ \displaystyle\int_\Omega A_\parallel \nabla_\parallel u \cdot \nabla_\parallel w \, dx = 0, \quad \forall w \in \mathcal{L}_{in}\,, \end{cases}$$

which is equivalent to the correct Limit problem (2.86). In this case, the auxiliary variable q acts as a Lagrange multiplier forcing u to be constant along b.

The drawback of this method is the choice of the space for the auxiliary variable. Imposing $q|_{\Gamma_{in}} = 0$ provides uniqueness of a solution but limits the application of the scheme to the case where all field lines are open. Indeed, fixing a value of q on the inflow boundary does not provide uniqueness of q on field lines which does not intersect with the inflow boundary (*i.e.* on closed field lines). In order to overcome this restriction we propose an approach based on requirement of the zero average of q along the field lines rather than on fixing the value of q on one of the boundaries.

Instead of demanding the auxiliary variable q to have a zero trace on the inflow part of boundary, let us require q to have average zero in the direction of the anisotropy. That is to say, let $q \in \mathcal{A}$. An asymptotic preserving method relying on this kind of decomposition reads: find $(u(t,\cdot), q(t,\cdot)) \in \mathcal{V} \times \mathcal{A}$, solution of

$$\begin{cases} \langle \dfrac{\partial u}{\partial t}, v \rangle_{\mathcal{V}^*,\mathcal{V}} + \displaystyle\int_\Omega (A_\perp \nabla_\perp u) \cdot \nabla_\perp v \, dx + \int_\Omega A_\parallel \nabla_\parallel q \cdot \nabla_\parallel v \, dx - \int_{\Gamma_N} g_N v \, ds = 0, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall v \in \mathcal{V} \\ \displaystyle\int_\Omega A_\parallel \nabla_\parallel u \cdot \nabla_\parallel w \, dx - \varepsilon \int_\Omega A_\parallel \nabla_\parallel q \cdot \nabla_\parallel w \, dx = 0, \quad \forall w \in \mathcal{A}\,, \end{cases} \tag{2.88}$$

This reformulation is equivalent to the original singular perturbation problem (2.2). Indeed, if (u, q) is a solution of (2.88) than choosing $v \in G \subset \mathcal{V}$ in the first equation yields

$$\langle \frac{\partial u}{\partial t}, v \rangle_{\mathcal{V}^*,\mathcal{V}} + \int_\Omega (A_\perp \nabla_\perp u) \cdot \nabla_\perp v \, dx - \int_{\Gamma_N} g_N v \, ds = 0, \quad \forall v \in \mathcal{G} \tag{2.89}$$

which is nothing else than the weak formulation of the original problem for test functions in $\mathcal{G}$. Similarly, if a test function of average zero is chosen in the first equation of (2.88), *i.e.* $v = w \in \mathcal{A} \subset \mathcal{V}$ then dividing the second equation by ε and adding it to the first leads to the variational formulation of the original problem for test functions in $\mathcal{A}$:

$$\langle \partial_t u(t,\cdot), v \rangle_{\mathcal{V}^*,\mathcal{V}} + \int_\Omega A_\perp \nabla_\perp u(t,\cdot) \cdot \nabla_\perp v \, dx + \frac{1}{\varepsilon} \int_\Omega A_\parallel \nabla_\parallel u(t,\cdot) \cdot \nabla_\parallel v \, dx \\ - \int_{\Gamma_N} g_N(t,\cdot) v = 0, \quad \forall v \in \mathcal{V}$$

The difficulty of discretizing the $\mathcal{A}$ space for arbitrary b can be overcome by a penalty stabilization and relaxation of the $\nabla_\parallel q = \frac{1}{\varepsilon} \nabla_\parallel u$ relation. This procedure allows to replace

the vector space $\mathcal{A}$ by easy to discretize one by the cost of some small additional error. This method shares some analogies with a penalty stabilization used for Stokes problem [15], where the $\nabla \cdot u = 0$ constraint is replaced by $\nabla \cdot u = h^2 \Delta p$ yielding the coercivity of the bilinear form associated with a finite element formulation of the problem and hence the uniqueness of the solution. The reason of stabilization is quite different in the case of the AP method since the uniqueness of the solution to (2.88) is already guaranteed. The goal is to develop a scheme which is easy to discretize in a general setting.

A new method is obtained by addition of a small penalization term to the second equation of (2.88). The APS-scheme reads: find $(u^\alpha(t,\cdot), q^\alpha(t,\cdot)) \in \mathcal{V} \times \mathcal{V}$, solution of

$$\left\{ \begin{array}{l} \langle \dfrac{\partial u^\alpha}{\partial t}, v \rangle_{\mathcal{V}^*,\mathcal{V}} + \displaystyle\int_\Omega (A_\perp \nabla_\perp u^\alpha) \cdot \nabla_\perp v \, dx + \int_\Omega A_\parallel \nabla_\parallel q^\alpha \cdot \nabla_\parallel v \, dx - \int_{\Gamma_N} g_N v \, ds = 0, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall v \in \mathcal{V} \\ \displaystyle\int_\Omega A_\parallel \nabla_\parallel u^\alpha \cdot \nabla_\parallel w \, dx - \varepsilon \int_\Omega A_\parallel \nabla_\parallel q^\alpha \cdot \nabla_\parallel w \, dx - \alpha \int_\Omega q^\alpha w = 0, \quad \forall w \in \mathcal{V}\,, \end{array} \right. \tag{2.90}$$

where α is a positive stabilization constant. Note that we are looking for q^α in $\mathcal{V}$, a vector space which is not linked to the anisotropy direction and is easy to discretize. Let us observe however that the stabilization term ensures that the auxiliary variable q^α belongs in fact to the space $\mathcal{A}$ for any non zero α. Indeed, choosing $w \in \mathcal{G}$ in the second equation of (2.90) yields

$$\int_\Omega q^\alpha w = 0 \quad \forall w \in \mathcal{G}. \tag{2.91}$$

That is to say, q^α is unique. In the stabilized scheme the relation $\nabla_\parallel u^\alpha = \varepsilon \nabla_\parallel q^\alpha$ is however no longer fulfilled as a perturbation proportional to α is added. If α is too big then the stabilization procedure introduces too much error. On the other hand, in the limit of $\alpha \to 0$ the solution of the (2.90) method converges to the one of (2.88). The formulation (2.88) is therefore the limit problem of (2.90) with respect to α. But if α is put to 0 than the uniqueness of q^α is lost. This suggests that in practice one should choose α to be of the order of truncation error so that the introduced error does not alter the convergence rate of the scheme and the solution remains unique.

Let us now choose a polygonalization of the domain Ω with polygons of the diameter approximately equal to h and introduce the finite element space $\mathcal{V}_h \subset \mathcal{V}$. The finite element discretization of (2.90) writes then: find $(u_h, q_h) \in \mathcal{V}_h \times \mathcal{V}_h$, the approximation of (u^α, q^α) such that

$$\left\{ \begin{array}{l} \displaystyle\int_\Omega \frac{\partial u_h}{\partial t} v_h \, dx + \int_\Omega (A_\perp \nabla_\perp u_h) \cdot \nabla_\perp v_h \, dx + \int_\Omega A_\parallel \nabla_\parallel q_h \cdot \nabla_\parallel v_h \, dx - \int_{\Gamma_N} g_N v_h \, ds = 0, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall v_h \in \mathcal{V}_h \\ \displaystyle\int_\Omega A_\parallel \nabla_\parallel u_h \cdot \nabla_\parallel w_h \, dx - \varepsilon \int_\Omega A_\parallel \nabla_\parallel q_h \cdot \nabla_\parallel w_h \, dx - h^{k+1} \int_\Omega q_h w_h = 0, \quad \forall w_h \in \mathcal{V}_h\,. \end{array} \right. \tag{2.92}$$

Remark that in order to ensure convergence rate in L^2-norm we have put $\alpha = h^{k+1}$, where k is the order of finite element method.

2.4 Semi-discretization in time

One should be extremely careful when discretizing in time the (2.92) scheme as not all numerical schemes conserve the AP property. In fact a chosen method should be L-stable. That is the reason why we have chosen the standard first order implicit Euler scheme and a second order, L-stable Runge-Kutta method.

Implicit Euler scheme

Let us write the first order, implicit Euler method in more compact notation: Find $(u_h^{n+1}, q_h^{n+1}) \in \mathcal{V}_h \times \mathcal{V}_h$, solution of

$$(E_{APS}) \quad \begin{cases} (u_h^{n+1}, v_h) + \tau \left(a_\perp(u_h^{n+1}, v_h) + a_\parallel(q_h^{n+1}, v_h) - \int_{\Gamma_N} g_N(t_{n+1}) v_h \, ds \right) = (u_h^n, v_h) \\ a_\parallel(u_h^{n+1}, w_h) - \varepsilon a_\parallel(q_h^{n+1}, w_h) - h^{k+1}(q_h^{n+1}, w_h) = 0 \,, \end{cases} .$$

L-stable Runge-Kutta method

In order to obtain a second order accurate in time scheme, we choose to implement a two stage Diagonally Implicit Runge-Kutta (DIRK) second order scheme already presented for the nonlinear problem. The second order AP-scheme writes: find $(u_h^{n+1}, q_h^{n+1}) \in \mathcal{V}_h \times \mathcal{V}_h$, solution of

$$(RK_{APS}) \quad \begin{cases} \left. \begin{array}{l} (u_{1,h}^{n+1}, v_h) + \tau\lambda \left(a_\perp(u_{1,h}^{n+1}, v_h) + a_\parallel(q_{1,h}^{n+1}, v_h) \right) - \int_{\Gamma_N} g_N(t_n + \lambda\tau) v_h \, ds \\ \qquad = (u_h^n, v_h) \end{array} \right. \\ a_\parallel \left(u_{1,h}^{n+1}, w_h \right) - \varepsilon a_\parallel(q_{1,h}^{n+1}, w_h) - h^{k+1}(q_{1,h}^{n+1}, w_h) = 0 \\ \left. \begin{array}{l} (u_{2,h}^{n+1}, v_h) + \tau\lambda \left(a_\perp(u_{2,h}^{n+1}, v_h) + a_\parallel(q_{2,h}^{n+1}, v_h) \right) - \int_{\Gamma_N} g_N(t_n + \tau) v_h \, ds \\ \qquad = (u_h^n, v_h) + \frac{1-\lambda}{\lambda} \left(u_{1,h}^{n+1} - u_h^n, v_h \right) \end{array} \right. \\ a_\parallel \left(u_{2,h}^{n+1}, w_h \right) - \varepsilon a_\parallel(q_{2,h}^{n+1}, w_h) - h^{k+1}(q_{2,h}^{n+1}, w_h) = 0 \end{cases}$$

$$u_h^{n+1} = u_{2,h}^{n+1} \,, \qquad q_h^{n+1} = q_{2,h}^{n+1} \,,$$

with $u_{1,h}^{n+1}$ and $u_{2,h}^{n+1}$ denote the solutions of the first and the second stage of the Runge-Kutta method.

Numerical tests show that the scheme is truly Asymptotic Preserving and that the optimal order of convergence is achieved in both time and space either for the formulation based on the MM method or for the stabilized version.

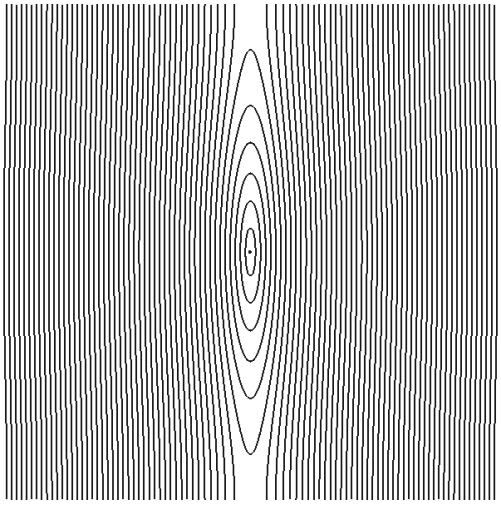


Figure 2.7: Magnetic island for $A = 0.01$

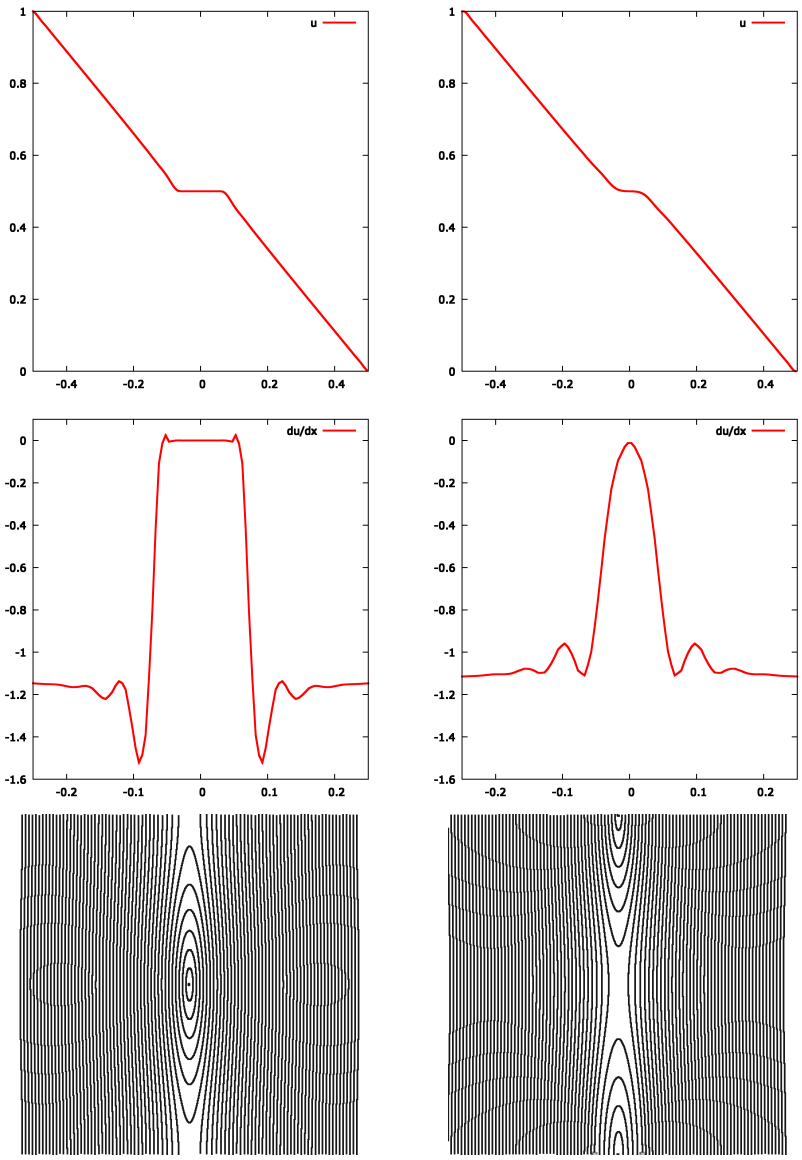
Magnetic island The stabilized scheme, contrary to the MM-based, can be successfully applied when the anisotropy direction field contains closed lines. This particular property is important in simulations related to the tokomak plasma, where anisotropy is driven by a magnetic field, which may form so-called "magnetic islands". Let us consider a square computational domain $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$ and a field b given by

$$b = \frac{B}{|B|}, \quad B = \begin{pmatrix} -A2\pi \sin(2\pi(y - \omega t)) \\ \pi \sin(\pi x) \end{pmatrix},$$

where A is some small constant. This field describe a topology with a magnetic island initially localized in the center of the domain and moving in time with a velocity ω. The island's width is given by $w = 4A^{1/2}/\pi$. This is the largest distance between the two branches of the separatrix, the line that divides the domain into regions of open and closed field lines. The two branches meet at the X-point, the saddle point of the vector potential. The island center, an extremum of the vector potential, is referred to as the O-point. If $A = 0$ the obtained field is aligned with the Y axis and points upwards (downwards) for $x > 0$ ($x < 0$). For $A > 0$ the magnetic island consisting of closed field lines appears in the region around $x = 0$. An example of magnetic island is given on Figure 2.7.

Simulations were performed with periodic boundary conditions applied on the top and bottom boundaries and with Dirichlet conditions on the left and right boundaries and for strong anisotropy ($\varepsilon = 10^{-10}$). The initial condition correspond to the stationary solution with no island present. That is to say, $u^0(x, y) = -x + \frac{1}{2}$. Temperature profiles and x component of its gradient through the axis $Y = 0$ at different time steps are shown on Figure 2.8. Numerical solution show the so called "profile flattening": the temperature is constant inside the magnetic island.

Finally, we have investigated the influence of the magnetic island rotation velocity for smaller anisotropy ($\varepsilon = 10^{-3}$). The rotation velocity changed from 10^3 to 10^6. The results are presented on Figure 2.9. For the 10^3 case the results does not deviate from a stationary

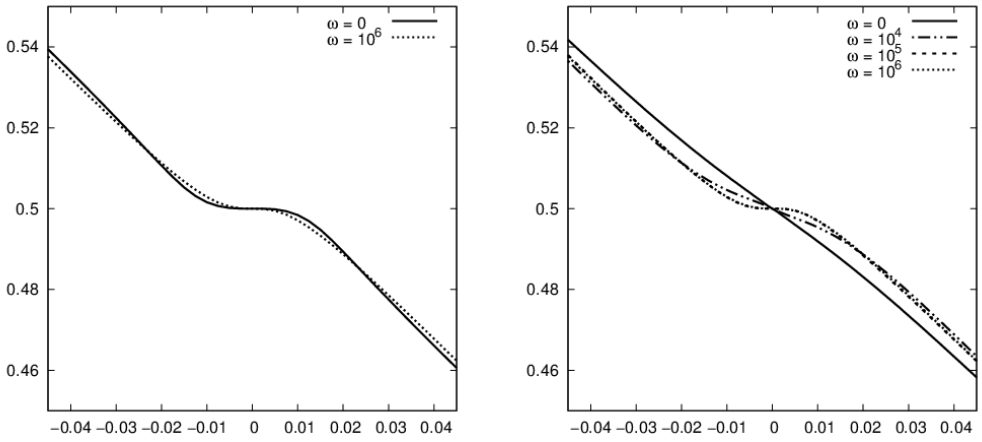


$t = 0.1$ $t = 0.15$

Figure 2.8: Temperature profiles along the X axis for a moving island ($A = 0.01, \omega = 10$) in the first row, x component of temperature gradient in the second row and a corresponding anisotropy field in the last row for different time steps for the Dirichlet BC.

case for a given time step. For large velocities the temperature profile on the line crossing the island's center is only slightly affected. However, when the profile is taken outside the island it becomes strongly rotation dependent. For a stationary island this profile is a straight line, but when the rotation speed increases, the profiles starts to flatten and for

the biggest value of ω this profile becomes the same as on the line crossing the island's center. The profile becomes homogeneous, *i.e.* independent of Y.



(a) (b)

Figure 2.9: Comparison between temperature profiles along the X axis for the stationary and rotating islands. The profile across the island center ($Y = Y_C$) on the left and away from the center ($Y = Y_C \pm 0.5$) on the right. Please note that temperature profiles for $\omega = 10^6$ and $\omega = 10^5$ are superposed.

Chapter 3

Efficient simulations of the Boltzmann equation

The goal of this chapter is to present the work done with Giacomo Dimarco, Raphaël Loubère and Thomas Rey concerning kinetic modeling and efficient numerical simulations of rarefied gases. It covers articles [34], [35] and [81].

In the kinetic theory of rarefied gases, the state of the system is described by a non negative distribution function $f(x, v, t)$ [20, 39]. This distribution function describes a density of particles moving with the velocity $v \in \mathbb{R}^3$ at the position $x \in \mathbb{R}^3$ at time t. The evolution of the system is governed by the six dimensional Boltzmann equation

$$\partial_t f + v \cdot \nabla_x f = \mathcal{Q}(f, f), \tag{3.1}$$

where the operator $\mathcal{Q}(f, f)$ is the collision operator and describes the effect of the particle collisions on the system. The macroscopic characteristics (density, momentum and energy) are obtained by integrating the distribution function multiplied by 1, v or $|v|^2$ over the velocity space:

$$U = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix} = \int_{\mathbb{R}^3} \phi(v) f dv,$$

where the vector $\phi(v)$ is given by $(1, v, \frac{1}{2}|v|^2)^T$. Typically, the collision operator conserves the macroscopic quantities of the system, *i.e.* the collisions preserve mass, momentum and energy. This is expressed as

$$\int_{\mathbb{R}^3} \phi(v) \mathcal{Q}(f, f) dv = 0$$

and hence the components of the vector $\phi(v)$ are referred to as *collision invariants*.

Multiplying the Boltzmann equation (3.1) by collision invariants and integrating over the velocity space yields a system of equations for evolution of macroscopic conservative variables

$$\frac{\partial}{\partial t} \int_{\mathbb{R}^3} f \phi(v) dv + \int_{\mathbb{R}^3} v \cdot \nabla_x f \phi(v) dv = 0. \tag{3.2}$$

This system is not closed as the second term involves higher order moments. However, when the system is at thermal equilibrium, the collision operator $\mathcal{Q}(f,f)=0$. The equilibrium is characterized by a local Maxwellian distribution

$$M[f]=\frac{\rho}{(2\pi T)^{3/2}}e^{-\frac{(u-v)^2}{2T}},$$

where u denotes a mean velocity, ρ is a density and the temperature T is related to the difference between the total and kinetic energy by the following relation:

$$\frac{3}{2}\rho T=E-\frac{1}{2}\rho|u|^2.$$

Replacing the distribution function f in (3.2) by the Maxwellian distribution $M[f]$ yields a closed system — a set of Euler equations

$$\begin{aligned}
&\frac{\partial}{\partial t}\rho+\nabla_x\cdot(\rho u)=0,\\
&\frac{\partial(\rho u)}{\partial t}+\nabla_x\cdot(\rho u\otimes u+pI)=0,\\
&\frac{\partial E}{\partial t}+\nabla_x\cdot((E+p)u)=0,
\end{aligned}$$

with the pressure following the ideal gas law $p=\rho T$.

The simplest collision operator providing the desired properties (conservation of collision invariants and vanishing at equilibrium) is the Bhatnagar-Gross-Krook (BGK) operator [10]

$$\mathcal{Q}_{BGK}(f,f)=\nu(M[f]-f),$$

where the inter-particle collisions are modelled as a relaxation process towards local equilibrium. The parameter $\nu=\nu(x,t)$ defines the collision frequency.

The classical Boltzmann collision operator is a multiple integral over the whole velocity space and all possible relative angles:

$$\mathcal{Q}_B(f,f)=\int_{\mathbb{R}^3}\int_{S^2}B(|v-v_\star|,\theta)\left(f(v')f(v'_\star)-f(v)f(v_\star)\right)d\omega dv_\star, \tag{3.3}$$

where v, $v_\star$ are velocities before collision, v', $v'_\star$ the velocities after collision and θ the angle between $v-v_\star$ and $v'-v'_\star$. The post collision velocities are given by

$$v'=\frac{1}{2}(v+v_\star+|v-v_\star|\omega)\ \ ,\ \ v'_\star=\frac{1}{2}(v+v_\star-|v-v_\star|\omega),$$

with ω being a vector on a unitary sphere S^2. The collision kernel B depends only on the relative velocity before collision and the deflection angle and has the form

$$B(|v-v_\star|,\theta)=|v-v_\star|\sigma(|v-v_\star|,\theta),$$

with σ being the scattering cross section. If the inverse k-th power forces between particles are considered, σ is given by

$$\sigma(|v - v_\star|, \theta) = b_\alpha(\theta)|v - v_\star|^{\alpha-1} \tag{3.4}$$

with $\alpha = (k-5)/(k-1)$. In the framework of the so called *variable hard spheres model* (VHS) [11] $b_\alpha(\theta)$ is constant: $b_\alpha(\theta) = C_\alpha$.

One of main challenges in numerical simulations of the Boltzmann equation is the dimensionality of the problem. The distribution function f depends at each instant of time on six independent variables: three space and three velocity coordinates. As a consequence, stochastic Monte Carlo methods are often applied [11, 17, 18, 61, 80]. They prove to be efficient in terms of computational cost, but are affected by stochastic fluctuations and slow convergence. On the other hand deterministic methods, such as finite volume schemes [77, 87, 88, 101, 111] or semi-Lagrangian approach to the transport part [23, 24, 46, 47, 56, 96], can reach higher order of accuracy but are much more resource demanding.

1 Fast Kinetic Schemes

Let us now present the Fast Kinetic Scheme (FKS) introduced by Dimarco and Loubère in [32, 33] for solving the Boltzmann equation (3.1). The FKS is a semi-Lagrangian scheme [23, 24, 48] which employs Discrete Velocity Model (DVM) [12, 77] approximation to the original problem.

Let us start by truncating a velocity space. Next we introduce a cubic grid of N_v equally spaced points in three dimensions and assume for simplicity that the grid step Δv is equal in every direction. Please note however that the FKS is not restricted to Cartesian grids in the velocity space. In fact the method rests unchanged even if unstructured and anisotropic velocity grids are taken into account. Let us now define an approximation of the continuous distribution function $f(x, v, t)$:

$$\tilde{f}_k(x, t) \approx f(x, v_k, t),$$

that is to say, continuous f is replaced by a vector $\tilde{f}$ and the following system of N_v equations is obtained:

$$\partial_t \tilde{f}_k + v_k \cdot \nabla_x \tilde{f}_k = \mathcal{Q}_k(\tilde{f}, \tilde{f}), \tag{3.5}$$

where $\mathcal{Q}_k(\tilde{f}, \tilde{f})$ is a suitable approximation of the collision operator for the discrete velocity point v_k. This set of equations is coupled only by the collision term.

Let us now discretize the physical space with a Cartesian grid consisting of N_s equally spaced points with a grid step Δx that is equal (for simplicity) in all three directions. Let us also introduce a time discretization with Δt being a time step and $t^n = t^0 + n\Delta t$ for any $n \geq 0$.

In the FKS framework, the equation (3.5) is solved with a first order splitting technique. First the transport step exactly solves the left hand side of the problem, then the collision

step introduces the interaction using the result of the transport step as a starting point:

$$\begin{aligned} \textit{Transport stage} &\longrightarrow \partial_t \tilde{f}_k + v_k \cdot \nabla_x \tilde{f}_k = 0, \\ \textit{Collisions stage} &\longrightarrow \partial_t \tilde{f}_k = \mathcal{Q}_k(\tilde{f}, \tilde{f}). \end{aligned} \tag{3.6}$$

Please note that higher order splitting techniques may also be considered.

Transport step

Let $f^n_{j,k}$ be a point-wise approximation of the distribution function at time t^n, position x_j and velocity v_k: $f^n_{j,k} = f(x_j, v_k, t^n)$. The main idea behind FKS is to solve the transport step (3.6) exactly. Let us define a piecewise constant in space approximation $\bar{f}^n_k(x)$ of the function $\tilde{f}_k(x, t^n)$ such that $\bar{f}^n_k(x) = f(x_j, v_k, t^n)$ if $x \in \Omega_j$ where $\Omega_j = [x_{j-1/2}, x_{j+1/2}]$ belongs to the space cell centered on x_j. The exact solution to the transport step at time t^n is therefore given by

$$\bar{f}^{\star,n+1}_k = \bar{f}^n_k(x - v_k \Delta t).$$

The function $\bar{f}^n_k$ is advected with a velocity v_k during a time step Δt. The discontinuities of $\bar{f}^{\star,n+1}_k$ do not coincide with space cell boundaries after the transport step.

Collision step

During the collision step the amplitude of the distribution function $\bar{f}$ is modified. The collision operator is solved locally on the space grid points and then extended to the whole domain Ω. The following equations (ordinary differential or integro-differential) are solved:

$$\partial_t f_{j,k} = \mathcal{Q}_k(f_{j,\cdot}, f_{j,\cdot}),$$

where $f_{j,k} = f(x_j, v_k, t)$ for all space and velocity grid points $j = 1, \ldots, N_s$ and $k = 1, \ldots, N_v$ and $f_{j,\cdot}$ is a N_v dimensional vector representing the distribution function at the space cell j composed of $f_{j,k}$. The initial data for this equation is provided by the transport step performed before. The time discretization chosen in this work is the first order explicit Euler scheme

$$f^{n+1}_{j,k} = f^{\star,n+1}_{j,k} + \Delta t \mathcal{Q}_k(f^{\star,n+1}_{j,\cdot}, f^{\star,n+1}_{j,\cdot}), \tag{3.7}$$

where $f^{\star,n+1}_{j,k} = \bar{f}^{\star,n+1}_k(x_j)$ is the value of transported distribution function at grid point x_j and $f^{\star,n+1}_{j,\cdot}$ is a vector composed of $f^{\star,n+1}_{j,k}$. Please note that other type of time integrators can be successfully implemented instead of this forward scheme. In particular the special care must be taken in the stiff limit, see Dimarco and Pareschi [37, 38] for further information.

Equation (3.7) furnishes a modified value of the distribution function at grid points x_j for velocity points v_k at time t^{n+1}. In order to obtain the value of f at every point of the domain a new piecewise constant function $\bar{\mathcal{Q}}_k$ is defined for every discrete velocity v_k:

$$\bar{\mathcal{Q}}^{n+1}_k(x) = \mathcal{Q}_k(f^{\star,n+1}_{j,\cdot}, f^{\star,n+1}_{j,\cdot}) \ , \ \forall x \ \text{ such that } \ \bar{f}^{\star,n+1}_k(x) = f^{\star,n+1}_{j,k},$$

that is to say, the collision operator at every point of Ω is approximated by a piecewise constant function with discontinuities located at the same points as the piecewise constant function that approximates the distribution function after the transport step. Thanks to this assumption, the spatially reconstructed distribution function after the collision step reads

$$\bar{f}_k^{n+1}(x) = \bar{f}_k^{\star,n+1}(x) + \Delta t \bar{\mathcal{Q}}_k^{n+1}(x).$$

This completes the description of the Fast Kinetic Scheme.

1.1 Collision operator

BGK approximation

If particle interaction is modeled by relaxation towards local equilibrium, the collision term $\mathcal{Q}_k(f_{j,\cdot}^{\star,n+1}, f_{j,\cdot}^{\star,n+1})$ becomes simply $\nu(\mathcal{E}_{j,k} - f_{j,k})$, where $\mathcal{E}_{j,k}$ is a suitable approximation of the Maxwell distribution for the velocity v_k at the grid point x_j. As the Maxwellian distribution depends on the macroscopic characteristics of the system that are unchanged during the relaxation step (since they are *collision invariants*), the relaxation step (3.7) becomes completely decoupled. In particular, $\mathcal{Q}_k$ depends only on one velocity point v_k and not on the others.

Boltzmann operator

If the Boltzmann operator is considered, the collision operator $\mathcal{Q}_k(f_{j,\cdot}^{\star,n+1}, f_{j,\cdot}^{\star,n+1})$ involves integration over whole velocity space for every point x_j on the space grid. The relaxation step is solved by means of Fast Spectral Scheme (FSS) and requires multiple Fourier transforms to be computed at every time step and at every space cell. The Fourier techniques for solving the Boltzmann integrals were first introduced by L. Pareschi and B. Perthame in [89] and by A. Bobylev and S. Rjasanow in [13]. Since then they have become a method of choice and are widely applied by many authors [2, 13, 14, 44, 45, 51, 52, 53, 90, 91, 92, 108, 109, 110]. We have chosen to implement the Fast Spectral Scheme (FSS) described in [78, 45] which has a complexity of the order of $\mathcal{O}(N_v \log(N_v))$. This method relies on the so-called Carleman representation [19] of the Boltzmann integral (3.3), introduction of the discrete number of reflection angles and identification of the convolution structures.

Let us present a particle interpretation of the FKS. Every point of the velocity grid represents a particle moving with velocity v_k. Every space cell Ω_j centered on the space grid point x_j contains exactly the same set of particles at exactly the same relative positions. Therefore one needs to store the particle position and velocity only in one generic cell and not in the whole domain. This reduces the memory requirements by a factor of seven: only mass of the particles is stored for every point of the $6D$ grid. While only three components of particle position and velocity vectors are only required for the generic reference cell.

The distribution function is related to particle masses by

$$f(x,v,t) = \sum_{j,k=1}^{N_s,N_v} \mathfrak{m}_{j,k}(t)\,\delta(x - x_{j,k}(t))\delta(v - v_{j,k}(t)), \quad v_{j,k}(t) = v_k,$$

where $x_{j,k}(t)$ is the particle position, $v_{j,k}(t)$ is its velocity and $\mathfrak{m}_{j,k}(t)$ particle mass. In the FKS the particle velocity is unchanged and the position is altered during the transport step:

$$x_{j,k}(t + \Delta t) = x_{j,k}(t) + v_{j,k}(t)\Delta t.$$

The transport step moves the particles in the reference cell. The motion of particles in the remaining cells is identical. If a given particle escapes the generic cell, another one with the same velocity (but different mass) enters the cell from the opposite side.

The collision step modifies the particle masses in every space cell:

$$\mathfrak{m}_{j,k}(t + \Delta t) = \mathfrak{m}_{j,k}(t) \; + \; \Delta t\, Q_k(v_{j,\cdot}),$$

where $Q_k(v_{j,\cdot})$ is the approximation of the collision (BGK or Boltzmann) operator in the center of the cell.

The macroscopic variables at time t^n are defined on the space grid only and are computed as a sum over particles in the given cell j:

$$U_j^n = \sum_{k=0}^{N_v} \phi(v_{j,k})\mathfrak{m}_{j,k}^n(\Delta v)^3.$$

As the collision step does not change the macroscopic conservative variables, they can be efficiently computed at time t^{n+1} after the transport step by adding the contribution from the particles leaving and entering the given cell j to the values at the previous time step. If a particle (j,k) is transported to the cell $j+\delta$ during the transport step, there is a sister particle entering the cell j from $j-\delta$. A suitable contribution has to be added to from the conservative variables in the cell j :

$$U_j^{n+1} = U_j^n + \sum_{k,\; x_{j,k}^{n+1} \in \Omega_{j+\delta},\; x_{j,k}^n \in \Omega_j} (\mathfrak{m}_{j-\delta,k}^n - \mathfrak{m}_{j,k}^n)\phi(v_{j,k})(\Delta v)^3 \tag{3.8}$$

Finally, the primitive variables (density, velocity and temperature) are computed from $U_j^{n+1} = \left(U_{j,1}^{n+1}, U_{j,2}^{n+1}, U_{j,3}^{n+1}\right)$:

$$\rho_j^{n+1} = U_1^{n+1}, \tag{3.9}$$

$$u_j^{n+1} = U_2^{n+1}/U_1^{n+1}, \tag{3.10}$$

$$T_j^{n+1} = \frac{2}{3}\left(U_3^{n+1} - \frac{\|U_2^{n+1}\|^2}{2U_1^{n+1}}\right)/U_1^{n+1}. \tag{3.11}$$

The most expensive part in the algorithm is the collision operator. Even in the case of the relatively simple BGK approximation the computation of the relaxation term takes 90% of the computational time on serial machines [34]. The cost of the Boltzmann integral is substantially greater, even if the Fast Spectral Scheme is employed. Indeed, a number of FFTs must be performed for every space cell and for every discrete angle in order to compute convolutions. If 16 discrete angles are considered, this number equals 96 and even if the size of those transforms is relatively small, this represents the main computational burden. Evaluation of the Boltzmann operator represents more than 99% of the computational time on serial machines. Fortunately the collision operator is in some sense decoupled from the FKS framework: it can be implemented independently of the FKS. This suggest a following strategy for the parallelization on distributed memory systems. On the upper level, the FKS is parallelized with MPI over available computational nodes. On the collision level, a suitable operator is implemented on the available node architecture: using the classical OpenMP type parallelism or the SIMD (Single Instruction, Multiple Data) programming model on GPUs or on the Intel Many Integrated Core (MIC) hardware. The implementation details of the collision operator does not influence the MPI scalability of the algorithm.

1.2 Sequential algorithm for FKS

We consider particle positions $\boldsymbol{X}_p^n$ and masses $m_{j,p}^n$ known at time t^n as well as conservative F_j^n and primitives variables $(\rho, \boldsymbol{U}, T)_j^n$. Then, one step of the algorithm reads:

1. *Transport of particles.* Displace N_v^3 particles, produce a list of N_{out} particles escaping the generic cell and store the δs determining the destination and provenance of associated sister particles.

2. *Update conservative variables.* If the particle has escaped the generic, add contribution to conservative variables. Reassign its mass and position with the ones of incoming sister particle.

3. *Update primitive variables.*

4. *Relaxation step.* Compute masses of N_v^3 particles after collisions, store them in an array of the size $N_v^3 \times N^3$.

The transport step is a simple particle position update computed for N_v^3 particles in the reference cell only. The second step updates the conservative variables in every space cell accordingly to (3.8). It has to be executed for every space cell and every particle leaving the reference cell. The third step computes the primitive variables accordingly to (3.9)-(3.11) and is executed in every space cell. The most demanding part of this algorithm is clearly the last step, where the expensive collision operator has to be evaluated for every particle in every space cell. It takes up to 99% of computational time, depending on the collision kernel employed.

The sequential algorithm is not suitable for full 3D/3D simulations due to two reason. The first is the size of the problem: required amount of memory largely exceeds capacity

of modern desktop stations, even for moderate size meshes. The second reason is related to computational complexity of collision kernel: runtime on single processor would be measured in months or years. That is why the need for efficient parallelization strategies arises and several approaches were proposed, including GPU computations [49, 50, 74] or OpenMP/MPI algorithms [3, 8, 58, 99, 100].

2 OpenMP and CUDA parallelization

Let us now present a parallelization scheme for shared memory systems, developed in collaboration with Giacomo Dimarco and Raphaël Loubère in [34].

Thanks to the particle interpretation of the scheme the parallelization becomes almost straightforward. The passage of information between the cells is only due to the transport part and for the particle-like interpretation it is then only due to the passage of particles between the different spacial cells. All the other steps of the kinetic scheme are local on the spacial mesh.

Let us now explore two parallelization for shared memory systems: a classical CPU parallelization using OpenMP (Open Multi-Processing) with shared memory, and Graphic Processing Unit (GPU) under CUDA (Compute Unified Device Architecture) parallel computing platform.

Classical parallel architecture: Open-MP

The modified algorithm reads:

1. *Relaxation step.* Divide the number of spatial cells by the number of processors. Compute in parallel the masses of N_v^3 particles after collision step, parallelization is performed on the loop over the number of mesh points in the physical space. This computation is local on the space mesh.

2. *Transport of particles.* Move in parallel N_v^3 particles. This step is done in only one space cell. The motion of particles in the other cells is the same.

3. *Update conservative variables.* Test in a parallel loop over the number of mesh points in the physical space if a particle has escaped from the generic cell. If so, add a contribution to F_j^{n+1} for every space cell. Update the particle position and exchange particle mass with the associated sister particle.

4. *Update primitive variables* in a parallel loop over the number of mesh points in the physical space.

Note that this algorithm is almost non-intrusive as only few omp pragma are needed to create such code.

Graphic Processing Unit (GPU) architecture: CUDA

The GPU parallelization demands more work. In fact one of the main drawbacks of the GPUs is the low level of memory capacity. Thus, it turns to be impossible to store on a single GPU the masses of all particles $m^n_{j,p}$ which is a vector of size $N^3 \times N^3_v$. Therefore we have chosen to group steps 1. to 3. in one sequential loop over all particles. Inside this loop we perform a parallel relaxation step (for one particle velocity and N^3 cells) using CUDA, then the transport step and (if particle has escaped from the generic cell), parallel update of the conservative variables. Finally, the update of primitive variables is performed on GPU.
This parallelization design can be summarized in the following algorithm.

1. *Copy from CPU to GPU.* Copy to the GPU memory all primitive and conservative variables.
2. *Loop over N^3_v particles*
   (a) *Copy from CPU to GPU.* Copy the mass array corresponding to a given velocity v_k to the GPU memory.
   (b) *Relaxation step* Compute relaxed masses of particles for every space cell using CUDA. Store the result on GPU.
   (c) *Transport step* Move every N^3_v particle and test if it has escaped the generic cell. If so, store the provenance cell of the sister particle.
   (d) *Update conservative variables.* If the particle has escaped the generic cell, add contribution to conservative variables. Reassign its mass and position with the ones of the incoming sister particle.
   (e) *Copy from GPU to CPU.* Copy the resulting mass array from the GPU memory to the CPU memory.
3. *Update primitive variables* in parallel on GPU.
4. *Copy from GPU to CPU.* Write to the CPU memory the updated conservative and primitive variables.

Observe that this relatively simple algorithm can be further improved to fit to multi GPU architectures. In this case one device is responsible for a subset of the velocity space. The above algorithm is well suited for the BGK collision kernel, which does not involve integrals over velocity space.

3 MPI parallelization

In this section we propose a parallelization strategy for distributed memory systems. The method presented herein is a subject of [81]. There are two possible approaches: the first one is similar to the algorithm for GPU systems. The velocity space decomposed and distributed over computational nodes keeping all spatial degrees of freedom at every node

This strategy was also chosen in [99] for the MPI implementation. Every computational node performs computations of a relaxation for a subset of velocity grid. Then the partial moments are evaluated. The total moments are obtained from gathering all contributions from all computational nodes. This approach is well suited for collision kernels that are local, *e.g.* for the BGK approximation, where the collision computed for a given position in physical and velocity space depends only on the distribution function at the same position and on total moments. In particular, the collision kernel should not require any additional information from neighbouring cell in the velocity mesh. As every node contains all spatial degrees of freedom, no particle escapes given computational node and no particle mass is exchanged with neighbouring nodes. The MPI communication is limited only to the partial moments. Another advantage is that even if a complicated domain is considered *i.e.* containing perforations, no complicated domain decomposition or load balancing techniques are required to ensure equal workloads across computational nodes. However, any collision kernel that is non-local would generate huge amount of communication between all processors. Boltzmann collision operator involve a double integral over velocity space meaning that at every iteration every node must have access to the whole velocity space. This approach is therefore not well suited for the Boltzmann operator.

The alternative possibility, adopted herein in view of Boltzmann collision kernel simulations, is to distribute spatial degrees of freedom over computational nodes, keeping on every mode a complete velocity space. Since the update of conservative variables (density, momentum and energy) requires an exchange of particle mass with neighbouring spatial cells (and does not involve any summation over whole physical space), the internodal memory transfer is limited to merely cells located on a boundary of a subdomain. Moreover, the information is exchanged with neighbouring nodes only and not with every node reserved for the computation. Comparison of the two approaches can be found in [100].

The spatial domain is decomposed into equally sized non-overlapping cuboids, pencils or slabs with ghost layers. Depending on the choice, every node has to communicate with 2 (for slabs), 8 (for pencils) or 26 (for cuboids) neighboring nodes. Cuboids usually minimize the size of ghost layers but have the biggest MPI overhead as they require more calls to MPI in order to communicate with all neighbors. Thus, the cuboid domain decomposition strategy is not always optimal and better results can be sometimes obtained when using pencils or slabs. The performance of the method can be improved by OpenMP or SIMD parallelization presented in the previous section and applied to the loops over velocity space at each node.

3.1 MPI version of FKS

The parallel algorithm is straightforward

1. *Initialization.* Divide the computational domain into $N_{MPI} = N_{MPI_x} \times N_{MPI_y} \times N_{MPI_z}$ equally sized cuboids. Allocate memory on each computational node: arrays of the size N_s/N_{MPI} for storing the conservative and primitive variables relative to a given subdomain and an array of the size $N_v \times (N_s/N_{MPI} + N_{\text{ghosts}})$ for storing masses relative to a given subdomain with additional ghost layers containing masses of particles in the adjacent space cells.

2. *Time iterations.* For every computational node:

   (a) *Relaxation step.* performed in parallel on GPU or with OpenMP for every particle in a given subdomain.

   (b) *Transport of particles.* Move in parallel N_v. This step is done in only one space cell in each subdomain. The motion of particles in the other cells is the same.

   (c) *Communication.* If a particle escapes given subdomain, broadcast its mass to suitable computational node.

   (d) *Update conservative variables.* If a particle has escaped the generic cell, add contribution to conservative variables. Reassign its mass and position with the ones of the incoming sister particle. For particles located on the boundary of the cuboid and escaping the subdomain use the values stored in the ghost cells in the previous step.

3. *Finalization.* Free memory and close MPI communication.

This algorithm clearly enjoys the weak scaling. Indeed, the communication is limited only to neighboring subdomains and the amount of data to be exchanged depends only on the local mesh size and chosen MPI topology (slabs / pencils / cuboids). In particular, it does not depend on number of computational nodes employed. The strong scaling is expected (especially for the resource consuming Boltzmann collision operator) and is indeed obtained. The results are presented in the following section.

4 Performance tests

4.1 OpenMP and GPU

The performance of the parallel algorithm was tested on the 3D Sod test case for the BGK collision kernel. The problem description is a 3D explosion problem [102], the initial state being given by the well known Sod shock tube problem. Let us consider a cubic domain of size $[0, 2]^3$. Left and right states of the 1D Sod problem are given by a density $\rho_L = 1$, mean velocity $\boldsymbol{U}_L = \mathbf{0}$ and temperature $\theta_L = 5$, while $\rho_R = 0.125$, $\boldsymbol{U}_R = \mathbf{0}$, $\theta_R = 4$. The gas is initially in thermodynamic equilibrium. The left state is set for any cell inside a ball centered in $(1, 1, 1)$ and of radius 0.2. The right state is set elsewhere. The computations are stopped at final time $t_{\text{final}} = 0.07$. The OpenMP simulations were performed on a computational server equipped with 4 Intel(R) Xeon(TM) E5-4650 processors running at 2.7 GHz (giving a total of 32 physical cores and 64 logical) with 512GB of RAM running under Debian Wheezy. GPU simulations are performed on a computational server equipped with dual Intel(R) Xeon(TM) E5-2650 processor running at 2.0GHz (16 physical and 32 virtual cores) with 128GB and 2 Nvidia GTX 780 units (3GB of memory, 2304 CUDA cores at 900MHz each) running under Debian Wheezy. The scalability tests for the OpenMP parallelization were performed on a $100^3 \times 15^3$ mesh. The results show the strong scaling close to perfect — see Figure 3.1. For the GPU version of the code the tests were performed on 100^3 space cells with velocity mesh ranging from 15^3 to 30^3 on one and

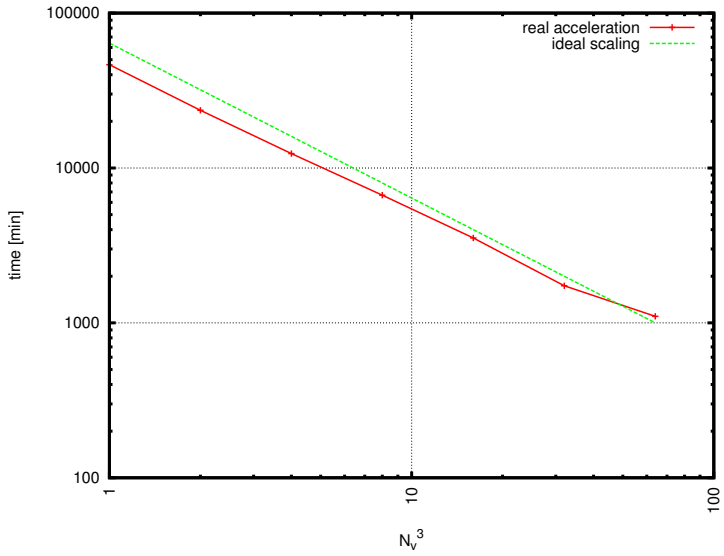


Figure 3.1: Scalability of the scheme using OpenMP on 32 core Intel Xeon computational server for 3D Sod problem for $N_c = 100^3$ for $\tau = 10^{-4}$. Log of time (in minutes) as a function of the log of the number of threads is presented.

two GPUs. The results presented on Figure 3.2 show that employing two graphical units yield approximately two times smaller runtime (compared to the runtime obtained on one graphical unit only). The OpenMP/GPU parallelization of the kinetic scheme allowed to perform complex numerical simulations for the BGK collision kernel in a reasonable time. The work [34] presents numerical results for truly six dimensional simulations of the re-entry test case (an object entering the atmosphere) and of the Kelvin Helmholtz instability [65] with the total runtime of the order of hours rather than months or years required by a sequential computer.

4.2 MPI scalability tests

Numerical tests for the MPI parallelization BGK collision kernel were performed on the EOS supercomputer at CALMIP, Toulouse. The supercomputer is equipped with 612 computational nodes, each of them containing two Intel® Ivy Bridge 2.8GHz 10 core CPUs and 64 GB of RAM. Each CPU was equipped with 25MB of cache memory. The code was compiled with gcc-5.3.0 and executed on 2 to 90 computational nodes. That is to say, on 40 to 1800 computational cores in parallel. The tests for Boltzmann collision kernel were performed on the EOS supercomputer ($N = 64^3$ meshes) and on the thin nodes of GENCI-TGCC supercomputer Curie for $N = 128^3$. The machine is equipped with 5040 B510 Bullx nodes (called thin nodes), each containing two Intel® Sandy Bridge 2.7GHz 8 core CPUs (20MB of cache memory) and 64GB of RAM. In the case of the Boltzmann

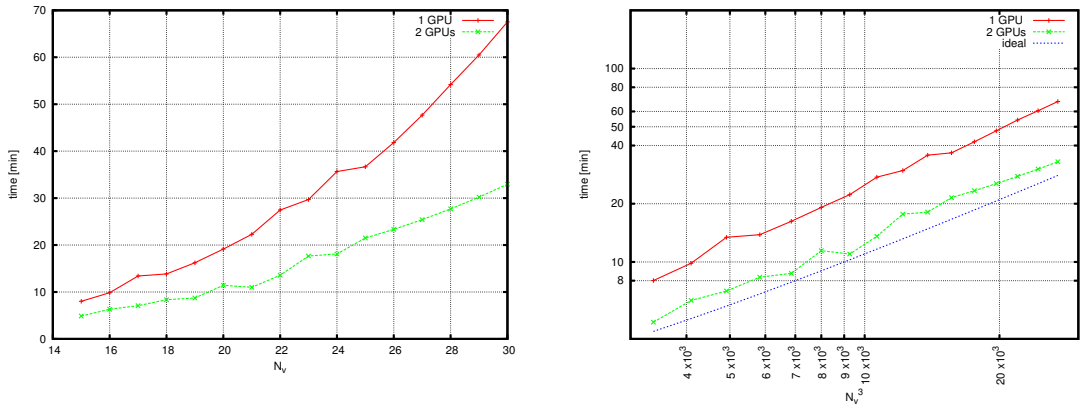


Figure 3.2: Scalability for FKS-GPU (1 GPU in red, 2 GPUs in green) for the 3D Sod problem for $N_c = 100^3$ for $\tau = 10^{-4}$ and varying velocity mesh $N_v = 15$ to 30. Left: CPU time (in minutes) as a function of N_v. Right: CPU time (in minutes) as a function of the number of velocity grid points in a spacial cell (N_v^3) compared to the ideal scaling (blue).

collision operator the Fast Fourier Transforms were computed by means of the fftw library, version 3.3.4. The code was executed with 20 OpenMP threads per node on EOS and 16 OpenMP threads per node on Curie (with hyperthreading disabled in both cases).

The performance of the parallel algorithm was tested on the 3D Sod test case presented above. The computations are stopped at final time $t_{\text{final}} = 0.07$ in all simulations except for the finest mesh for Boltzmann operator, where the computations were stopped after after 16 iterations. This number of times steps ensured that every discrete particle has changed the physical cell at least once. We consider the case in which $\tau = 10^{-1}$, *i.e.* far from the fluid limit. We are not interested in the convergence of the numerical solution but in the parallel efficiency only.

The scalability tests for the BGK operator were performed with the optimal domain decomposition strategy (see [81] for details) for four different meshes ($64^3 \times 32^3$, $128^3 \times 16^3$, $192^3 \times 16^3$ and $270^3 \times 16^3$). The run time as a function of computational cores is presented in Figure 3.3 and the speedup with parallel efficiency (relative to the smallest number of nodes employed in the test) in Figure 3.4. The proposed algorithm shows very good scaling property. The worst parallel efficiency, equal to ~ 0.77, was obtained for the $64^3 \times 32^3$ mesh. For remaining meshes the parallel efficiency was close to the ideal with some super linear behaviour due probably to the CPU cache and memory performance. Linear decrease in the efficiency is also observed. This is due to the relatively small computational burden of the BGK kernel, to small to hide the communication time.

Let us now turn our attention to the 3D Boltzmann collision kernel. This task is much more demanding in terms of computational time as the relaxation routine involves multiple expensive calls to the Fast Fourier Transform. The tests were run on following meshes:

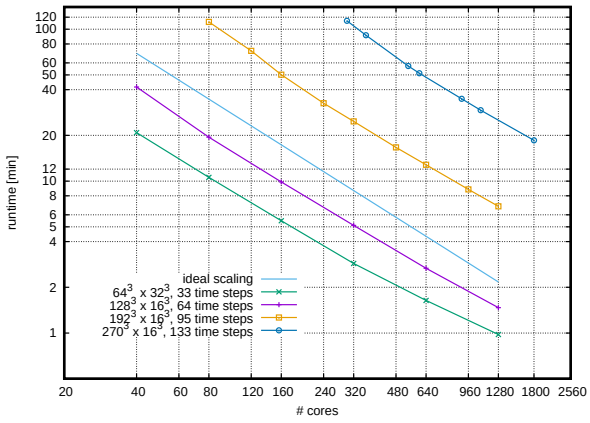


Figure 3.3: Computational time as a function of number of cores employed for BGK.

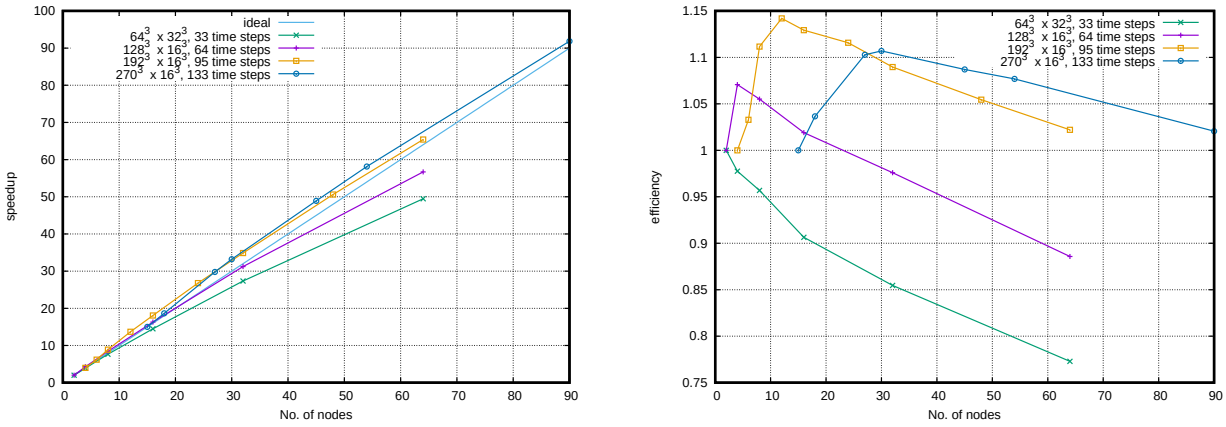


Figure 3.4: Speedup (left) and efficiency (right) as a function of number of computational nodes for BGK. A minimal number of nodes requisite to run the test is equal to 2 for $64^3 \times 32^3$ and $128^3 \times 16^3$ mesh and to 4 for the $192^3 \times 16^3$ mesh.

	case 1	case 2	case 3
N	64^3	64^3	128^3
N_v	16^3	32^3	32^3

The results are presented on Figures 3.5 and 3.6 and on Tables 3.1 and 3.2. For the last mesh computations were performed for until the final time $t_f = 0.07$ was reached (case 1 and 2) or were stopped after 16 iterations (case 3). The tests were run on N_{MPI} ranging from 16 to 1024. That is to say on 128 to 2048 processors and on 1024 to 16384 computational cores. The results are presented on Table 3.3 and on Figures 3.5 and 3.6.

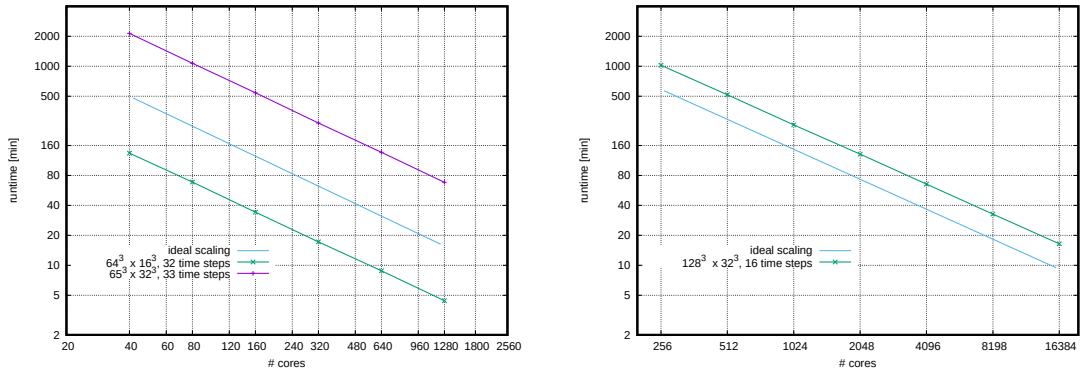


Figure 3.5: Computational time as a function of number of cores employed on EOS machine (left panel) and on TGCC-CURIE (right panel).

Due to the huge computational complexity of the collision operator the parallel efficiency is close to one (> 0.95) in all performed simulations (up to 1024 computational nodes).

MPI parallelization is indispensable if complex 6D involving the Boltzmann collision operator are required. In [81] a numerical simulations of five objects moving in high parts of the atmosphere are presented for the mesh consisting of $150^3 \times 32^3$ points. The total runtime on 1000 computational nodes (16000 computational cores) was approximately 18 hours, that is to say 286000 core hours, equivalent of 32 years of computations on a serial computer.

N_V	Vel.	Cell #	#nodes	N_{cycle}	Time(s)	T_{cycle}	T_{cell}	$T_{\text{cell/node}}$
16^3	$[-15,15]$	$64^3 \times 16^3$ $= 1.07 \times 10^9$	2	33	8043.07	244	$9.3\ 10^{-4}$	$1.86\ 10^{-3}$
			4		4119.41	125	$4.76\ 10^{-4}$	$1.90\ 10^{-3}$
			8		2053.01	62.2	$2.37\ 10^{-4}$	$1.90\ 10^{-3}$
			16		1031.67	31.3	$1.19\ 10^{-4}$	$1.91\ 10^{-3}$
			32		529.246	16	$6.12\ 10^{-5}$	$1.96\ 10^{-3}$
			64		264.725	8.02	$3.06\ 10^{-5}$	$1.96\ 10^{-3}$

Table 3.1: Performance tests on $64^3 \times 16^3$ mesh for Boltzmann collision kernel. Time per cycle is obtained by $T_{\text{cycle}} = T/N_{\text{cycle}}$, time per cycle per cell by $T_{\text{cell}} = T_{\text{cycle}}/N_c$ and time per cycle per node by $T_{\text{cycle/node}} = N_s T_{\text{cell}}$. Each node has 20 cores.

5 Comparison of Boltzmann and BGK collision operators

Finally, let us compare numerical results for the BGK and Boltzmann collision kernels in full $3D \times 3D$ simulations in order to show that both models can give noticeably different

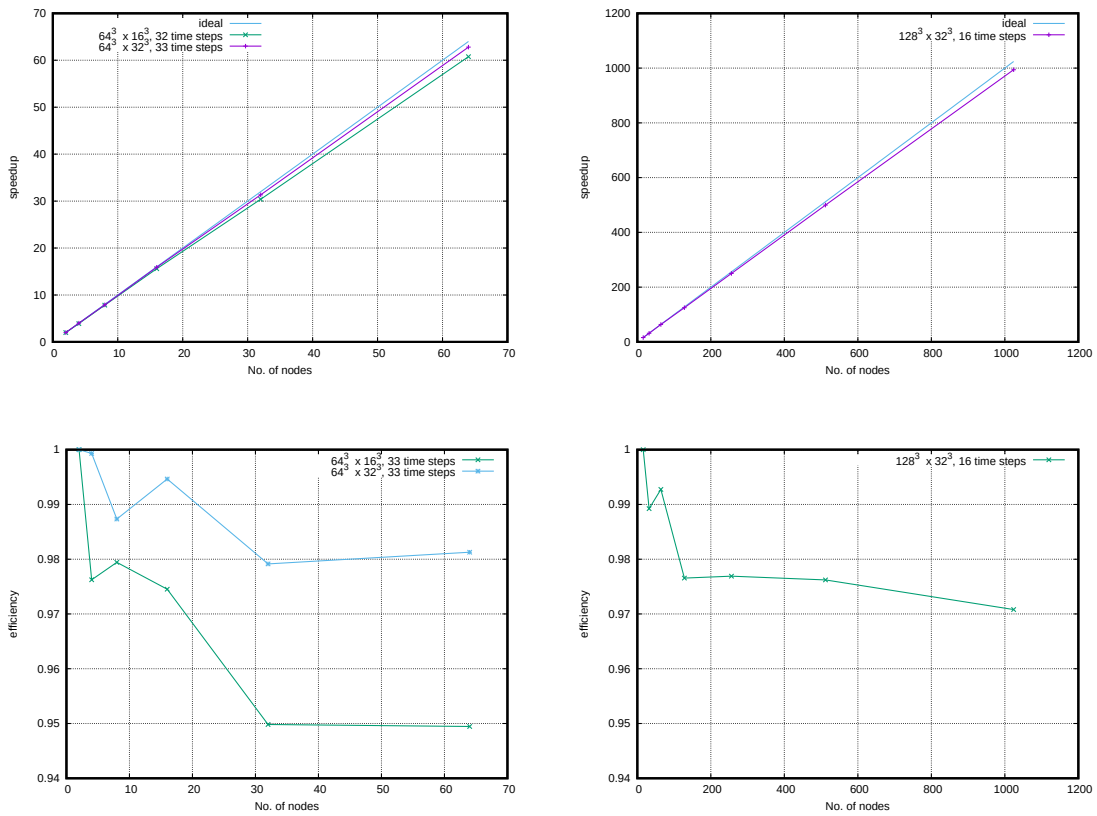


Figure 3.6: Speedup (top) and efficiency (bottom) as a function of number of computational nodes for 3D Boltzmann collision kernel on EOS machine (left panel) and on TGCC-CURIE (right panel).

results in some cases. More in-depth comparison including results for both two and three velocity dimensions and space dimensions ranging from zero (space homogeneous case) to three are reported in [35]. Here we consider a re-entry test case, where a static cuboid is placed inside a computational domain set to $\Omega = [0, 2]^3$ (see Fig.3.7). The velocity space is $[-10, 10]^3$ and discretized with 32^3 points. The relaxation parameter is set to $\tau = 0.3$. The initial density ρ is set to 1, the temperature $T = 1$ and the initial velocity is given by $(u_x, u_y, u_z) = (2, 0, 0)$. The final time is set to $t_{\text{final}} = 0.6$ leading to 379 time steps. The inflow boundary conditions are imposed on the left boundary ($x = 0$) while outflow boundary conditions on the remaining are imposed. Hard sphere molecules are considered for Boltzmann while for the BGK model the frequency ν is taken equal to $\mu = C_\alpha 4\pi(2\lambda\pi)^\alpha$. For both models the CFL condition considered is consequently given by

$$\Delta t \leq \min\left(\frac{\Delta x}{|v_{max}|}, \frac{\tau}{\mu}\right). \tag{3.12}$$

The results are shown for the temperature and the density in Figure 3.7 and the dis-

N_V	Vel.	Cell #	#nodes	N_{cycle}	Time(s)	T_{cycle}	T_{cell}	$T_{\text{cell/node}}$
32^3	$[-15, 15]$	$64^3 \times 32^3$ $= 8.6 \times 10^9$	2	33	128378	3890	$1.48\ 10^{-2}$	$2.97\ 10^{-2}$
			4		64233.2	1950	$7.43\ 10^{-3}$	$2.97\ 10^{-2}$
			8		32506.4	985	$3.76\ 10^{-3}$	$3.01\ 10^{-2}$
			16		16133.7	489	$1.87\ 10^{-3}$	$2.98\ 10^{-2}$
			32		8194.58	248	$9.47\ 10^{-4}$	$3.03\ 10^{-2}$
			64		4088.34	124	$4.73\ 10^{-4}$	$3.02\ 10^{-2}$

Table 3.2: Performance tests on $64^3 \times 32^3$ mesh for Boltzmann collision kernel. Time per cycle is obtained by $T_{\text{cycle}} = T/N_{\text{cycle}}$, time per cycle per cell by $T_{\text{cell}} = T_{\text{cycle}}/N_c$ and time per cycle per node by $T_{\text{cycle/node}} = N_s T_{\text{cell}}$. Each node has 20 cores.

N_V	Vel.	Cell #	#nodes	N_{cycle}	Time(s)	T_{cycle}	T_{cell}	$T_{\text{cell/node}}$
32^3	$[-15, 15]$	$128^3 \times 32^3$ $= 69 \times 10^9$	16	16	61346.4	3834.2	$1.83\ 10^{-3}$	$2.93\ 10^{-2}$
			32		31006.9	1937.9	$9.24\ 10^{-4}$	$2.96\ 10^{-2}$
			64		15448.8	965.6	$4.60\ 10^{-4}$	$2.95\ 10^{-2}$
			128		7852.4	490.8	$2.34\ 10^{-4}$	$3.00\ 10^{-2}$
			256		3924.8	245.3	$1.17\ 10^{-4}$	$2.99\ 10^{-2}$
			512		1963.7	122.7	$5.85\ 10^{-5}$	$3.00\ 10^{-2}$
			1024		987.37	61.71	$2.94\ 10^{-5}$	$3.01\ 10^{-2}$

Table 3.3: Performance tests on $128^3 \times 32^3$ mesh for Boltzmann collision kernel for 16 time steps. Time per cycle is obtained by $T_{\text{cycle}} = T/N_{\text{cycle}}$, time per cycle per cell by $T_{\text{cell}} = T_{\text{cycle}}/N_c$ and time per cycle per node by $T_{\text{cycle/node}} = N_s T_{\text{cell}}$. Each node has 16 cores.

crepancies between the two in Figure 3.8. The difference between the two models is clearly present in those results suggesting that the BGK model does not provide reliable numerical data in the rarefied gas simulations.

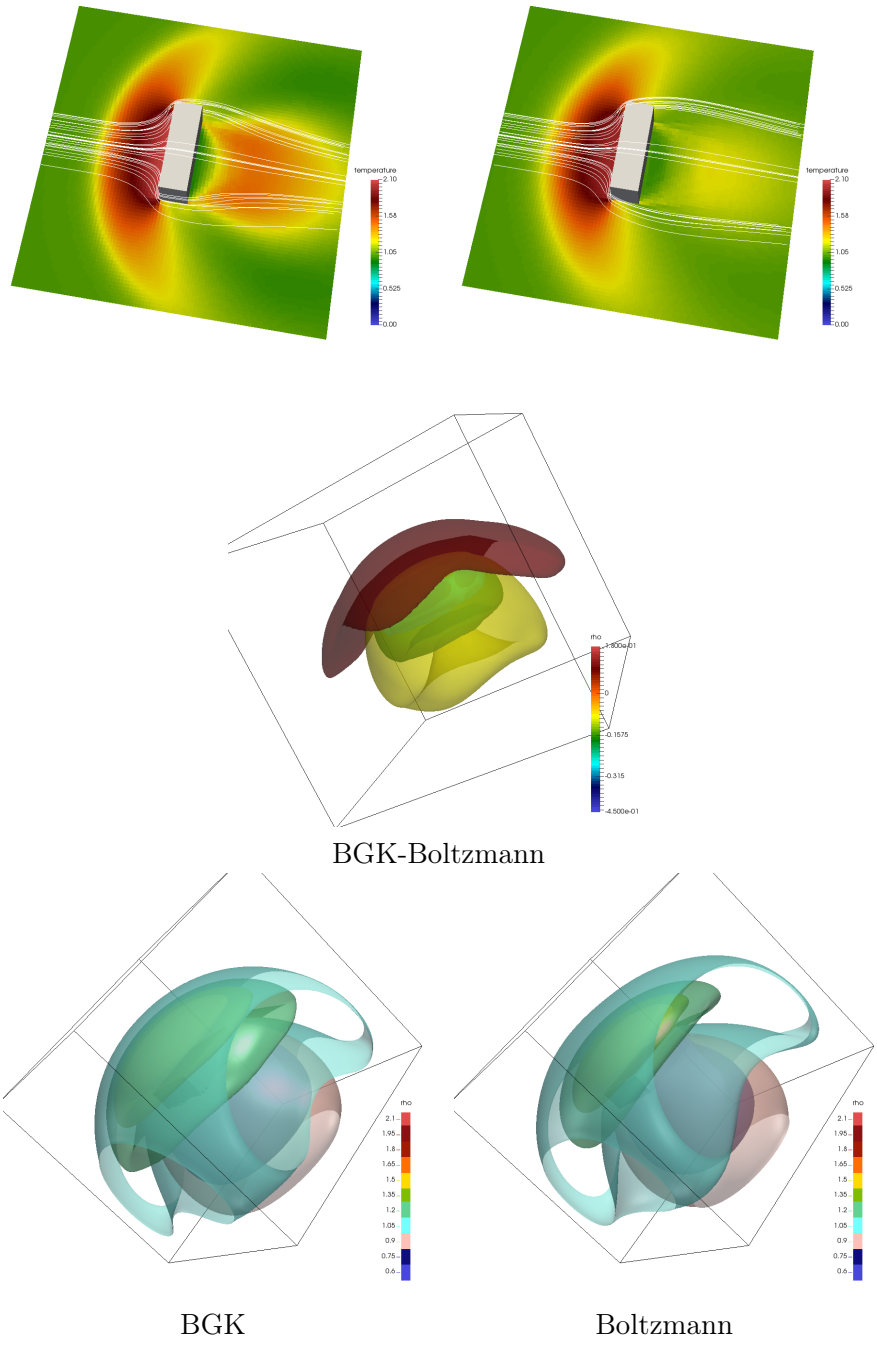
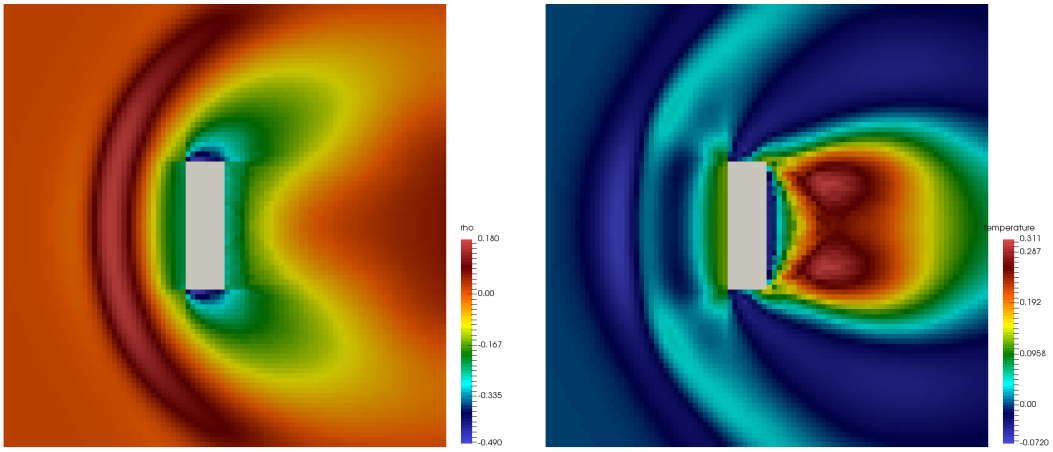


Figure 3.7: Test 4.1. Three dimensional re-entry test case for $\tau = 0.3$ with $M = 90\times90\times90$ spatial cells and $N = 32^3$ velocity cells. BGK model (left column), Boltzmann model (right column) at time $t_{\text{final}} = 0.6$. Top row: temperature field with velocity streamlines, bottom row: isosurfaces of the density. Middle row: isosurfaces of the density difference between BGK and Boltzmann models.



density temperature

Figure 3.8: Test 4.1. Three dimensional re-entry test case for $\tau = 0.3$ with $M = 90\times90\times90$ spatial cells and $N = 32^3$ velocity cells. Discrepancies between the BGK solution and the Boltzmann solution at time $t_{\text{final}} = 0.6$. Density on the left and temperature on the right.

Chapter 4

Other work

1 Multiscale Finite Element Method

Many physical and engineering problems contain multiple scales: for example transport in heterogeneous media, where matrix coefficients can oscillate rapidly in the computational domain or in perforated domain, where the domain itself exhibits small scale features. In order to obtain small scale properties of the solution the mesh employed in numerical simulations must be able to reflect those variations. The resulting discrete system is often intractable due to its size, even if high performance computing is considered. In order to overcome this problem one can resort to the so-called Multiscale Finite Element Method (MsFEM) [60]. In the MsFEM all numerical simulations are performed on a coarse mesh with a size that is in general bigger than small scale size of the problem. The basis functions however are not the standard finite element basis functions such as $\mathbb{P}_k$ or $\mathbb{Q}_k$, but pre-calculated ones. Those MsFEM basis functions are obtained by means of local fine scale simulations and reflect small scale behaviour of the solution. Together withe Pierre Degond, Alexei Lozinski and Bagus Muljadi we have proposed in [29] a variant of the MsFEM method which employs the Crouzeix-Raviart basis functions enriched by bubble functions for diffusion and advection-diffusion equations in perforated domains. This work was then extended in [79] to the Stokes problem.

2 Pedestrians and crowds

In [43] we have studied a time-delayed follow-the-leader model for pedestrians walking in line. In this model a pedestrians reaction to velocity and position change relative to its predecessor is not immediate. The numerical parameters are calibrated to fit experimental data. The resulting system of delay differential equations turn out to be unstable. As a remedy, some dissipation is added: the velocity is relaxed to a mean velocity of certain number of predecessors. Stability analysis confirms well-posedness of a relaxed system. Numerical simulations are in good agreement with experimental results for large pedestrian densities. Macroscopic features of the system, such us dynamics of traffic jams (number of jams, average velocity in jams, jam head velocity, number of pedestrians in jams) were

reproduced. On the other hand, low pedestrian density experimental results where not reproduced by the model. It suggests that some other mechanisms not taken into account by the model play important role for lower densities.

Bibliography

[1] J. Adam, J. Boeuf, N. Dubuit, M. Dudeck, L. Garrigues, D. Gresillon, A. Heron, G. Hagelaar, V. Kulaev, N. Lemoine, et al. Physics, simulation and diagnostics of Hall effect thrusters. *Plasma Physics and Controlled Fusion*, 50:124041, 2008.

[2] A. Alekssenko and E. Josyula. Deterministic solution of the Boltzmann equation using a discontinuous Galerkin velocity discretization. In A. C. P. A. I. of Physics, editor, *Proceedings of the 28th International Symposium on Rarefied Gas Dynamics*, volume 1501, pages 279–286, 2012.

[3] V. V. Aristov and S. A. Zabelok. A deterministic method for solving the Boltzmann equation with parallel computations. *Comput. Math. Math. Phys.*, 42(3):425–437, 2002.

[4] D. Aronson. The porous medium equation. *A. Fasano, M. Primicerio (Eds.), Nonlinear Diffusion Problems, Lecture Notes in Mathematics*, 1224:1–46, 1986.

[5] S. F. Ashby, W. J. Bosl, R. D. Falgout, S. G. Smith, A. F. Tompson, and T. J. Williams. A Numerical Simulation of Groundwater Flow and Contaminant Transport on the CRAY T3D and C90 Supercomputers. *International Journal of High Performance Computing Applications*, 13(1):80–93, 1999.

[6] I. Babuška and M. Suri. Locking effects in the finite element approximation of elasticity problems. *Numerische Mathematik*, 62(1):439–463, Dec. 1992.

[7] I. Babuška and M. Suri. On Locking and Robustness in the Finite Element Method. *SIAM Journal on Numerical Analysis*, 29(5):1261–1293, Oct. 1992.

[8] C. Baranger, J. Claudel, N. Hérouard, and L. Mieussens. Locally refined discrete velocity grids for stationary rarefied flow simulations. *Journal of Computational Physics*, 257, Part A:572 – 593, 2014.

[9] M. Beer, S. Cowley, and G. Hammett. Field-aligned coordinates for nonlinear simulations of tokamak turbulence. *Physics of Plasmas*, 2(7):2687, 1995.

[10] P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94(3):511–525, 1954.

[11] G. A. Bird. *Molecular gas dynamics and the direct simulation of gas flows*. Oxford University Press, 2nd edition, 1994.

[12] A. V. Bobylev, A. Palczewski, and J. Schneider. On approximation of the Boltzmann equation by discrete velocity models. *C. R. Acad. Sci. Paris Ser. I Math.*, 320(5):639–644, 1995.

[13] A. V. Bobylev and S. Rjasanow. Difference scheme for the Boltzmann equation based on the fast Fourier transform. *Eur. J. Mech. B Fluids*, 16(2):293–306, 1997.

[14] A. V. Bobylev and S. Rjasanow. Numerical solution of the Boltzmann equation using a fully conservative difference scheme based on the fast Fourier transform. In *Proceedings of the Fifth International Workshop on Mathematical Aspects of Fluid and Plasma Dynamics (Maui, HI, 1998)*, volume 29, pages 289–310, 2000.

[15] F. Brezzi and J. Douglas, Jr. Stabilized mixed methods for the Stokes problem. *Numer. Math.*, 53(1-2):225–235, 1988.

[16] S. Brull, P. Degond, and F. Deluzet. Numerical degenerate elliptic problems and their applications to magnetized plasma simulations. *Communications in Computational Physics*, 223(1):147–178, 2012.

[17] R. E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta numerica*, 7:1–49, 1998.

[18] R. E. Caflisch and L. Pareschi. Towards a hybrid Monte Carlo method for rarefied gas dynamics. In *Transport in Transition Regimes*, pages 57–73. Springer, 2004.

[19] T. Carleman. Sur la théorie de l'équation intégrodifférentielle de Boltzmann. *Acta Math.*, 60(1):91–146, 1933.

[20] C. Cercignani. *The Boltzmann equation and its applications*, volume 67 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1988.

[21] L. Chacón, D. del Castillo-Negrete, and C. D. Hauck. An asymptotic-preserving semi-lagrangian algorithm for the time-dependent anisotropic heat transport equation. *Journal of Computational Physics*, 272:719–746, Sept. 2014.

[22] N. Crouseilles, M. Kuhn, and G. Latu. Comparison of Numerical Solvers for Anisotropic Diffusion Equations Arising in Plasma Physics. *Journal of Scientific Computing*, pages 1–38, Feb. 2015.

[23] N. Crouseilles, M. Mehrenberger, and E. Sonnendrücker. Conservative semi-Lagrangian schemes for Vlasov equations. *Journal of Computational Physics*, 229(6):1927–1953, 2010.

[24] N. Crouseilles, T. Respaud, and E. Sonnendrücker. A forward semi-Lagrangian method for the numerical solution of the Vlasov equation. *Computer Physics Communications*, 180(10):1730–1745, 2009.

[25] P. Degond, F. Deluzet, A. Lozinski, J. Narski, and C. Negulescu. Duality-based asymptotic-preserving method for highly anisotropic diffusion equations. *Commun. Math. Sci.*, 10(1):1–31, 2012.

[26] P. Degond, F. Deluzet, D. Maldarella, J. Narski, and C. Negulescu. Hybrid model for the coupling of an asymptotic preserving scheme with the asymptotic limit model: The one dimensional case. In *ESAIM: Proceedings*, volume 32, pages 23–30. EDP Sciences, 2011.

[27] P. Degond, F. Deluzet, and C. Negulescu. An asymptotic preserving scheme for strongly anisotropic elliptic problems. *Multiscale Model. Simul.*, 8(2):645–666, 2009/10.

[28] P. Degond, F. Deluzet, A. Sangam, and M.-H. Vignal. An asymptotic preserving scheme for the Euler equations in a strong magnetic field. *J. Comput. Phys.*, 228(10):3540–3558, 2009.

[29] P. Degond, A. Lozinski, B. P. Muljadi, and J. Narski. Crouzeix-raviart msfem with bubble functions for diffusion and advection-diffusion in perforated media. *Communications in Computational Physics*, 17(4):887–907, 2015.

[30] P. Degond, A. Lozinski, J. Narski, and C. Negulescu. An asymptotic-preserving method for highly anisotropic elliptic equations based on a micro-macro decomposition. *Journal of Computational Physics*, 231(7):2724–2740, 2012.

[31] F. Deluzet and J. Narski. A two field iterated asymptotic-preserving method for highly anisotropic elliptic equations. *submitted*, 2017.

[32] G. Dimarco and R. Loubère. Towards an ultra efficient kinetic scheme. Part I: Basics on the BGK equation. *Journal of Computational Physics*, 255:680–698, 2013.

[33] G. Dimarco and R. Loubère. Towards an ultra efficient kinetic scheme. Part II: The high order case. *Journal of Computational Physics*, 255:699–719, 2013.

[34] G. Dimarco, R. Loubère, and J. Narski. Towards an ultra efficient kinetic scheme. Part III: High-performance-computing. *Journal of Computational Physics*, 284:22–39, 2015.

[35] G. Dimarco, R. Loubère, J. Narski, and T. Rey. An efficient numerical method for solving the Boltzmann equation in multidimensions. *Journal of Computational Physics*, 353(Supplement C):46–81, Jan. 2018.

[36] G. Dimarco and J. Narski. Hybrid monte carlo schemes for plasma simulations. In *AIP Conference Proceedings*, volume 1389, pages 1130–1133. AIP, 2011.

[37] G. Dimarco and L. Pareschi. High order asymptotic-preserving schemes for the Boltzmann equation. *C. R. Math. Acad. Sci. Paris*, 350(9-10):481–486, 2012.

[38] G. Dimarco and L. Pareschi. Asymptotic preserving implicit-explicit Runge-Kutta methods for nonlinear kinetic equations. *SIAM J. Numer. Anal.*, 51(2):1064–1087, 2013.

[39] G. Dimarco and L. Pareschi. Numerical methods for kinetic equations. *Acta Numer.*, 23:369–520, 2014.

[40] Y. Dubinskii. Some integral inequalities and the solvability of degenerate quasi-linear elliptic systems of differential equations. *Matematicheskii Sbornik*, 106(3):458–480, 1964.

[41] Y. Dubinskii. Weak convergence for nonlinear elliptic and parabolic equations. *Matematicheskii Sbornik*, 109(4):609–642, 1965.

[42] R. Eymard, T. Gallouët, and R. Herbin. A new finite volume scheme for anisotropic diffusion problems on general grids: convergence analysis. *C. R. Math. Acad. Sci. Paris*, 344(6):403–406, 2007.

[43] J. Fehrenbach, J. Narski, J. Hua, S. Lemercier, A. Jelić, C. Appert-Rolland, S. Donikian, J. Pettré, and P. Degond. Time-delayed follow-the-leader model for pedestrians walking in line. *Networks and Heterogeneous Media*, 10(3):579–608, 2015.

[44] F. Filbet and C. Mouhot. Analysis of spectral methods for the homogeneous Boltzmann Equation. *Trans. Amer. Math. Soc.*, 363:1947–1980, 2011.

[45] F. Filbet, C. Mouhot, and L. Pareschi. Solving the Boltzmann equation in N log2 N. *SIAM J. Sci. Comput.*, 28(3):1029–1053, 2007.

[46] F. Filbet and G. Russo. High order numerical methods for the space non-homogeneous Boltzmann equation. *J. Comput. Phys.*, 186(2):457–480, Apr. 2003.

[47] F. Filbet and G. Russo. Accurate numerical methods for the Boltzmann equation. In *Modeling and computational methods for kinetic equations*, pages 117–145. Springer, 2004.

[48] F. Filbet, E. Sonnendrücker, and P. Bertrand. Conservative numerical schemes for the Vlasov equation. *J. Comput. Phys.*, 172(1):166–187, Sept. 2001.

[49] A. Frezzotti, G. P. Ghiroldi, and L. Gibelli. Solving model kinetic equations on GPUs. *Comput. & Fluids*, 50:136–146, 2011.

[50] A. Frezzotti, G. P. Ghiroldi, and L. Gibelli. Solving the Boltzmann equation on GPUs. *Comput. Phys. Comm.*, 182(12):2445–2453, 2011.

[51] I. M. Gamba and J. R. Haack. A conservative spectral method for the Boltzmann equation with anisotropic scattering and the grazing collisions limit. *J. Comput. Phys.*, 270:40–57, 2014.

[52] I. M. Gamba and S. H. Tharkabhushanam. Spectral-Lagrangian methods for collisional models of non-equilibrium statistical states. *J. Comput. Phys.*, 228(6):2012–2036, Apr. 2009.

[53] I. M. Gamba and S. H. Tharkabhushanam. Shock and boundary structure formation by spectral-Lagrangian methods for the inhomogeneous Boltzmann transport equation. *J. Comput. Math.*, 28(4):430–460, 2010.

[54] M. W. Gee, J. J. Hu, and R. S. Tuminaro. A new smoothed aggregation multigrid method for anisotropic problems. *Numer. Linear Algebra Appl.*, 16(1):19–37, 2009.

[55] L. Giraud and R. S. Tuminaro. Schur complement preconditioners for anisotropic problems. *IMA J. Numer. Anal.*, 19(1):1–18, 1999.

[56] Y. Güçlü and W. N. G. Hitchon. A high order cell-centered semi-Lagrangian scheme for multi-dimensional kinetic simulations of neutral gas flows. *J. Comput. Phys.*, 231(8):3289–3316, 2012.

[57] S. Günter, Q. Yu, J. Krüger, and K. Lackner. Modelling of heat transport in magnetised plasmas using non-aligned coordinates. *Journal of Computational Physics*, 209(1):354 – 370, 2005.

[58] J. Haack and I. M. Gamba. High performance computing with a conservative spectral boltzmann solver. *28th International Symposium on Rarefied Gas Dynamics 2012*, 1501:334–341, 2012.

[59] E. Hairer and G. Wanner. *Solving ordinary differential equations. II*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2010. Stiff and differential-algebraic problems, Second revised edition, paperback.

[60] T. Y. Hou and X.-H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *J. Comput. Phys.*, 134(1):169–189, 1997.

[61] M. S. Ivanov and S. F. Gimelshein. Computational hypersonic rarefied flows. In *Annual review of fluid mechanics, Vol. 30*, volume 30 of *Annu. Rev. Fluid Mech.*, pages 469–505. Annual Reviews, Palo Alto, CA, 1998.

[62] H. Jian and B. Song. Solutions of the anisotropic porous medium equation in $\mathbb{R}^n$ under an l^1-initial value. *Nonlinear analysis*, 64(9):2098–2111, 2006.

[63] S. Jin. Efficient asymptotic-preserving (AP) schemes for some multiscale kinetic equations. *SIAM J. Sci. Comput.*, 21(2):441–454, 1999.

[64] M. Kelley, W. Swartz, and J. Makela. Mid-latitude ionospheric fluctuation spectra due to secondary EŨB instabilities. *Journal of Atmospheric and Solar-Terrestrial Physics*, 66(17):1559–1565, 2004.

[65] R. Keppens and G. Tóth. Nonlinear dynamics of kelvin-helmholtz unstable magnetized jets: Three-dimensional effects. *Physics of Plasmas*, 6(5):1461–1469, 1999.

[66] M. Keskinen, S. Ossakow, and B. Fejer. Three-dimensional nonlinear evolution of equatorial ionospheric spread-F bubbles. *Geophys. Res. Lett*, 30(16):4–1–4–4, 2003.

[67] B. N. Khoromskij and G. Wittum. Robust Schur complement method for strongly anisotropic elliptic equations. *Numer. Linear Algebra Appl.*, 6(8):621–653, 1999.

[68] D. Kuzmin, M. J. Shashkov, and D. Svyatskiy. A constrained finite element method satisfying the discrete maximum principle for anisotropic diffusion problems. *J. Comput. Phys.*, 228(9):3448–3463, 2009.

[69] C. Le Potier. Schéma volumes finis pour des opérateurs de diffusion fortement anisotropes sur des maillages non structurés. *C. R. Math. Acad. Sci. Paris*, 340(12):921–926, 2005.

[70] J. Lions. *Quelques méthodes de résolution des problèmes aux limites non linéaires.* Gauthier-Villars, 1969.

[71] K. Lipnikov, M. Shashkov, and D. Svyatskiy. The mimetic finite difference discretization of diffusion problem on unstructured polyhedral meshes. *J. Comput. Phys.*, 211(2):473–491, 2006.

[72] A. Lozinski, J. Narski, and C. Negulescu. Highly anisotropic nonlinear temperature balance equation and its numerical solution using asymptotic-preserving schemes of second order in time. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(06):1701–1724, 2014.

[73] A. Lozinski, J. Narski, and C. Negulescu. Numerical analysis of an asymptotic-preserving scheme for anisotropic elliptic equations. *submitted, arXiv:1507.00879 [math]*, July 2015. arXiv: 1507.00879.

[74] E. A. Malkov and M. S. Ivanov. Parallelization of algorithms for solving the boltzmann equation for gpu-based computations. *AIP Conference Proceedings*, 1333(1):946–951, 2011.

[75] T. Manku and A. Nathan. Electrical properties of silicon under nonuniform stress. *Journal of Applied Physics*, 74(3):1832–1837, 1993.

[76] A. Mentrelli and C. Negulescu. Asymptotic-preserving scheme for highly anisotropic non-linear diffusion equations. *Journal of Computational Physics*, 231(24):8229 – 8245, 2012.

[77] L. Mieussens. Discrete velocity model and implicit scheme for the BGK equation of rarefied gas dynamics. *Mathematical Models and Methods in Applied Sciences*, 10(08):1121–1149, 2000.

[78] C. Mouhot and L. Pareschi. Fast algorithms for computing the Boltzmann collision operator. *Math. Comp.*, 75(256):1833–1852 (electronic), 2006.

[79] B. P. Muljadi, J. Narski, A. Lozinski, and P. Degond. Nonconforming multiscale finite element method for stokes flows in heterogeneous media. part i: Methodologies and numerical experiments. *Multiscale Modeling & Simulation*, 13(4):1146–1172, 2015.

[80] K. Nanbu. Direct simulation scheme derived from the boltzmann equation. i. monocomponent gases. *Journal of the Physical Society of Japan*, 49(5):2042–2049, 1980.

[81] J. Narski. Fast Kinetic Scheme : efficient MPI parallelization strategy for 3D Boltzmann equation. *to appear in CiCP*, 2018.

[82] J. Narski and M. Ottaviani. Asymptotic preserving scheme for strongly anisotropic parabolic equations for arbitrary anisotropy direction. *Computer Physics Communications*, 185(12):3189–3203, 2014.

[83] J. Narski and M. Picasso. Adaptive 3d finite elements with high aspect ratio for dendritic growth of a binary alloy including fluid flow induced by shrinkage. *FDMP: Fluid Dynamics & Materials Processing*, 3(1):49–64, 2007.

[84] J. Narski and M. Picasso. Adaptive finite elements with high aspect ratio for dendritic growth of a binary alloy including fluid flow induced by shrinkage. *Computer methods in applied mechanics and engineering*, 196(37):3562–3576, 2007.

[85] J. Narski and M. Picasso. *Adaptive Finite Elements with High Aspect Ratio for Dendritic Growth of a Binary Alloy Including Fluid Flow Induced by Shrinkage*, pages 327–337. Birkhäuser Basel, Basel, 2007.

[86] Y. Notay. An aggregation-based algebraic multigrid method. Report GANMN 08-02, Université Libre de Bruxelles, Brussels, Belgium, 2008.

[87] A. Palczewski and J. Schneider. Existence, stability, and convergence of solutions of discrete velocity models to the boltzmann equation. *Journal of statistical physics*, 91(1-2):307–326, 1998.

[88] A. Palczewski, J. Schneider, and A. V. Bobylev. A consistency result for a discrete-velocity model of the Boltzmann equation. *SIAM journal on numerical analysis*, 34(5):1865–1883, 1997.

[89] L. Pareschi and B. Perthame. A fourier spectral method for homogeneous Boltzmann equations. *Transport Theory Statist. Phys.*, 25(3):369–382, 1996.

[90] L. Pareschi and G. Russo. Numerical solution of the Boltzmann equation I: Spectrally accurate approximation of the collision operator. *SIAM J. Numer. Anal.*, 37(4):1217–1245, 2000.

[91] L. Pareschi and G. Russo. On the stability of spectral methods for the homogeneous Boltzmann equation. *Trans. Theo. Stat. Phys.*, 29:431–447, 2000.

[92] L. Pareschi, G. Toscani, and C. Villani. Spectral methods for the non cut-off Boltzmann equation and numerical grazing collision limit. *Numer. Math.*, 93(3):527–548, 2003.

[93] A. Ratnani, E. Franck, B. Nkonga, A. Eksaeva, and M. Kazakova. Anisotropic Diffusion in Toroidal geometries. *ESAIM: Proceedings and Surveys*, 53:77–98, Mar. 2016.

[94] P. Sharma and G. W. Hammett. Preserving monotonicity in anisotropic diffusion. *J. Comput. Phys.*, 227:123–142, November 2007.

[95] P. Sharma and G. W. Hammett. A fast semi-implicit method for anisotropic diffusion. *Journal of Computational Physics*, 230(12):4899–4909, 2011.

[96] M. Shoucri and G. Knorr. Numerical integration of the Vlasov equation. *J. Computational Phys.*, 14(1):84–92, 1974.

[97] P. Tamain. *Etude des flux de matière dans le plasma de bord des tokamaks*. PhD thesis, Marseille 1: 2007., 2007.

[98] M. Tang and Y. Wang. An asymptotic preserving method for strongly anisotropic diffusion equations based on field line integration. *Journal of Computational Physics*, 330:735–748, 2017.

[99] V. Titarev. Efficient deterministic modelling of three-dimensional rarefied gas flows. *Communications in Computational Physics*, 12(01):162–192, 2012.

[100] V. Titarev, M. Dumbser, and S. Utyuzhnikov. Construction and comparison of parallel implicit kinetic solvers in three spatial dimensions. *Journal of Computational Physics*, 256:17–33, 2014.

[101] V. A. Titarev, S. V. Utyuzhnikov, and A. V. Chikitkin. OpenMP + MPI parallel implementation of a numerical method for solving a kinetic equation. *Comput. Math. Math. Phys.*, 56(11):1919–1928, 2016.

[102] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.

[103] A. M. Tréguier. Modélisation numérique pour l'océanographie physique. *Ann. Math. Blaise Pascal*, 9(2):345–361, 2002.

[104] B. van Es, B. Koren, and H. J. de Blank. Finite-difference schemes for anisotropic diffusion. *Journal of Computational Physics*, 272:526–549, Sept. 2014.

[105] B. van Es, B. Koren, and H. J. de Blank. Finite-volume scheme for anisotropic diffusion. *Journal of Computational Physics*, 306:422–442, 2016.

[106] J. Vázquez. *The porous medium equation: mathematical theory*. Oxford University Press, USA, 2007.

[107] J. Wesson. *Tokamaks*. Oxford University Press, New York, NY, 1987.

[108] L. Wu, H. Liu, Y. Zhang, and J. M. Reese. Influence of intermolecular potentials on rarefied gas flows: Fast spectral solutions of the Boltzmann equation. *Physics of Fluids (1994-present)*, 27(8):082002, 2015.

[109] L. Wu, C. White, T. J. Scanlon, J. M. Reese, and Y. Zhang. Deterministic numerical solutions of the Boltzmann equation using the fast spectral method. *Journal of Computational Physics*, 250:27–52, 2013.

[110] L. Wu, J. Zhang, J. M. Reese, and Y. Zhang. A fast spectral method for the Boltzmann equation for monatomic gas mixtures. *Journal of Computational Physics*, 298:602–621, 2015.

[111] S. Zabelok, R. Arslanbekov, and V. Kolobov. Adaptive kinetic-fluid solvers for heterogeneous computing architectures. *J. Comput. Phys.*, 303:455–469, 2015.