Contributions to the study of MHD convection : theoretical framework, finite volume methods, and pre-exascale simulations

Rémi Bourgeois

▶ **To cite this version:**

Rémi Bourgeois. Contributions to the study of MHD convection : theoretical framework, finite volume methods, and pre-exascale simulations. Numerical Analysis [cs.NA]. Université Paris-Saclay, 2024. English. NNT : 2024UPASP044 . tel-04732720

HAL Id: tel-04732720
https://theses.hal.science/tel-04732720v1

Submitted on 11 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

université PARIS-SACLAY

Contributions to the study of MHD convection : theoretical framework, finite volume methods, and pre-exascale simulations

Contributions à l'étude de la convection MHD : cadre théorique, méthodes volumes finis et simulations à l'échelle pré-exascale

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°127 Astronomie & Astrophysique (AAIF)
Spécialité de doctorat : Astronomie & Astrophysique
Graduate School : Physique. Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **Maison de la Simulation (Université Paris-Saclay, UVSQ, Inria, CNRS, CEA)** sous la direction de **Pascal TREMBLIN**, Directeur de recherche, le co-encadrement de **Samuel KOKH**, Directeur de recherche

Thèse soutenue à Paris-Saclay, le 08 Juillet 2024, par

Rémi BOURGEOIS

Composition du jury

Membres du jury avec voix délibérative

Christian TENAUD Directeur de recherche, CNRS, Université Paris-Saclay	Président
Bruno DESPRÉS Professeur, CNRS, Sorbonne Université	Rapporteur & Examinateur
Romain TEYSSIER Professeur, Department of Astrophysical Sciences, Princeton University	Rapporteur & Examinateur
Isabelle BARAFFE Professeure, Physics and Astronomy, Exeter University	Examinatrice

THESE DE DOCTORAT
NNT : 2024UPASP044

université PARIS-SACLAY

ÉCOLE DOCTORALE
Astronomie
et Astrophysique
d'Île-de-France (AAIF)

Titre : Contributions à l'étude de la convection MHD : cadre théorique, méthodes volumes finis et simulations à l'échelle pré-exascale

Mots clés : convection, simulation numérique, analyse numérique, magnetohydrodynamique, calcul haute performance, régime bas Mach

Résumé : La convection est un phénomène omniprésent dans l'univers, jouant un rôle clé dans la structure des océans, des atmosphères planétaires et stellaires. Ce travail se concentre sur les aspects théoriques et numériques de l'étude de la convection. D'abord, les méthodes "tout-régime" et "équilibre" colocalisées, basées sur un splitting d'opérateur et particulièrement adaptées aux phénomènes convectifs, sont reformulées en splitting de flux, améliorant leur flexibilité. Ensuite, une extension de cette méthode à la magnétohydrodynamique est proposée. Sa stabilité est prouvée sans recours au contrôle de la valeur de $\nabla \cdot \boldsymbol{B}$, grâce au splitting et à l'emploi de termes de Powell. Par ailleurs, l'analyse de l'instabilité pour la convection diabatique est étendue à la magnétohydrodynamique et aux écoulements cisaillés. Une nouvelle instabilité "triple diffusive" et des estimations d'intensité de dynamo convective sont dérivées et validées par des simulations utilisant les méthodes développées. Enfin, une simulation à très grande échelle de dynamo convective sur le supercalculateur Adastra est présentée. L'intégration de PDI et Deisa, outils d'I/O modernes développés à la Maison de la Simulation, dans le code HPC "ARK" basé sur Kokkos+MPI est exposée. En complément, des travaux préliminaires sur l'utilisation de petits réseaux de neurones pour accélérer la méthode GP-MOOD sont présentés.

Title : Contributions to the study of MHD convection : theoretical framework, finite volume methods, and pre-exascale simulations

Keywords : convection, numerical simulation, numerical analysis, magnetohydrodynamics, high-performance computing, low Mach regime

Abstract : Convection is a ubiquitous phenomenon in the universe, playing a key role in the structure of oceans and planetary and stellar atmospheres. This work focuses on the theoretical and numerical aspects of studying convection. First, the "all-regime" and "equilibrium" collocated methods, based on operator splitting and particularly suited for convective phenomena, are reformulated in terms of flux splitting, enhancing their flexibility. Subsequently, an extension of this method to magnetohydrodynamics is proposed. Its stability is demonstrated without relying on controlling the value of $\nabla \cdot \boldsymbol{B}$, thanks to the splitting and the use of Powell terms. The analysis for diabatic convection is extended to magnetohydrodynamics and sheared flows. A new "triple diffusive" instability and estimates of convective dynamo intensity are derived and validated through simulations using the developed methods. Finally, a large-scale simulation of convective dynamo on the supercomputer Adastra is presented. The integration of PDI and Deisa, modern I/O tools developed at Maison de la Simulation, into the HPC code "ARK" based on Kokkos+MPI is discussed. Additionally, preliminary work on the use of small neural networks to accelerate the GP-MOOD method is presented.

Remerciements

Je tiens tout d'abord à remercier mes encadrants, Pascal et Samuel. Merci pour votre disponibilité malgré les responsabilités qui s'accumulent et votre expertise. Merci pour la bienveillance avec laquelle vous m'avez guidé tout au long de ma thèse, pour la résolution des difficultés scientifiques mais aussi personnelles, quand ma confiance en moi et ma motivation se faisaient plus rares. Je vous remercie de m'avoir traité d'égal à égal, comme un collègue, et d'avoir su quand me guider de près, et quand me laisser mon indépendance. Pascal, ton approche multidisciplinaire de la science, ta clairvoyance et ton efficacité sont une inspiration pour ma carrière de chercheur. Merci aussi de m'avoir enseigné le théorème d'Astérix :). Samuel, ton expertise, ta rigueur et ta connaissance profonde de notre domaine de travail sont impressionnantes et ont été indispensables au bon déroulement de ma thèse.

Je remercie également les membres de mon jury, Isabelle Baraffe, Christian Tenaud, Bruno Després et Romain Teyssier, d'avoir accepté de plonger dans mes travaux de thèse avec interêt.

Je remercie mes parents, Denise et Laurent, merci pour votre soutien inconditionnel et vos yeux admiratifs. Merci aussi de m'avoir accueilli systématiquement à bras ouverts à Canéjan, cela m'a évité beaucoup de déprime à Massy, et merci pour toute l'aide que vous me procurez dans chaque nouvelle étape de ma vie. Je tiens aussi à remercier mes frères. Vincent, merci pour ton soutien et pour tout ce que tu fais pour l'union de notre famille. Nicolas, merci aussi pour ton soutien et ta sagesse (et les bagarres). Merci Kareen de toujours m'accueillir chez vous et pour ta joie de vivre. Merci Jazz d'être là tout simplement, tu rends notre vie plus douce et j'ai hâte de te voir grandir. Merci Jin de toujours répondre présent à nos rassemblements.

Je remercie Thomas Padioleau pour toute son aide sur les aspects HPC/Kokkos de mes travaux, j'espère que tu es conscient du rôle crucial que tu as joué dans le déroulement de ma thèse, en particulier pendant le grand challenge. Beaucoup de choses n'auraient pas été possibles sans toi. Je remercie également Yushan pour son aide sur beaucoup d'aspects techniques de ce projet. I want to thank Felix for all the help you provided to me.

Je remercie la ZONE, (1 & 2), Lou et Aymeric, merci d'avoir égayé mon quotidien au labo, les nombreuses pauses et les afterworks. Bon courage pour la fin de votre thèse. Je remercie aussi mes collègues Jean-Marc, François-Xavier, Meriem, Hiba, Benjamin, Florent, Ester, Kévin, Paul et Paul, Sarah, Juan, Myriam, Mélissa. Je vous souhaite à tous le meilleur pour la suite. Mention spéciale pour Thierry qui reprend le flambeau en thèse, bonne chance à toi :). Merci à Amal pour ton aide avec DEISA, ma soutenance de thèse et ta bonne humeur pendant les temps moroses du COVID. Tu nous manques ! Merci à Julien Bigot pour sa participation à mon CSI et son aide sur quelques aspects techniques de PDI. Merci à Charles pour toutes les super sessions d'escalade, les nombreuses discussions et pour ton aide à la préparation de ma soutenance. Merci à Benoît, Édouard, Julien Thélot, Karim, Mathieu, Martial, Mathieu Simplice, Yuiichi et tous les autres. Merci à Valérie et Sarah pour votre sympathie et votre aide à travers toutes les tâches administratives. Finalement, merci à tout le labo pour le super cadeau de soutenance que vous m'avez offert.

Merci à Antoine Strugarek, Maxime Delorme et Sacha Brun pour votre collaboration pour le test du solveur MHD, nos interactions ont vraiment enrichi ce que l'on propose. Merci à Teddy Pichard

pour la discussion cruciale sur la combinaison convexe, qui nous a permis d'améliorer grandement la CFL de notre méthode numérique.

I want to express my deep gratitude to Dongwook Lee for believing in me back in 2018 and giving me a chance to come and study at UCSC. Your decision to accept me as your graduate student led to so much growth in all aspects of my life. Professionally, I learned so much about finite volume methods and this led to my first ever publication, which definitely helped in getting selected into my PhD program. Personally, it led to what I still consider to be the most fun year of my life and allowed me to make deeply valuable friendships that are still active to this day. Not only that, but you accepted to work with me during my PhD, allowing me to come again to my favorite town in the world and to conduct exciting work. Thanks for everything and I hope we get to work together soon again. I also want to thank all my colleagues at UCSC, Ian, Sean, Chris, Pascale, Nicholas, Youngjun, Martin, Peter, and all the others. I miss you all. I also want to thank Susie and Jon for being such amazing landlords and welcoming me in Betty. Thanks to Dori for your end-of-PhD wisdom and kind support. I also want to thank all the international papis, Yelle, Olivia, Morten, Mattias. I'm so thankful we got to hang out back in Europe. Alejandro, thanks for welcoming me in Colombia along with Lukiri, Elisa and titi. This trip was legendary. Oli, thanks for taking the time to come visit me in Bordeaux and for welcoming me in Utrecht. You're a truly special friend. Thanks to Matilda for the adventures during the weird times that were the beginning of my PhD while COVID was happening. Thanks to Wesley for welcoming me into your home in Colorado. I'm so glad we got to connect again.

Je tiens à remercier tout le chenil de la casse, A2, Duncan, Hugo, Julie, PJ, Simon, Émile, Tomy, Yann, Adrien et Laurie. Vous êtes ma seconde famille. Merci pour l'enthousiasme avec lequel vous m'accueillez à Bordeaux/Rennes/Rocambourt à chaque fois, et d'être aussi souvent venus me voir à Massy. Je devrais écrire une partie pour chacun.e d'entre vous, mais ce serait plus long que ma thèse, alors on se dira les choses en vrai. Je remercie aussi Anthony et Eva pour leur soutien, depuis le début. Merci à Titou, aka Saint Marmokak Kebab Artist, pour toutes les zbaberies depuis la prépa. Tu nous manques. Merci aussi à YoungBeuga, Alexia et Xou2 pour votre collaboration. Je remercie aussi la tribu : E, Jérémy, Manu, Guillaume et Yoann, pour les aventures de l'ENSEIRB. Merci à Jacques Roturier pour avoir suivi de loin et avec bienveillance ma carrière de chercheur depuis mes débuts en prépa. Merci à Léna pour les longues discussions sur le sens de la vie en tant que chercheur.e.

Un merci très spécial à Emma, tu es arrivée dans ma vie dans cette période très spéciale qu'est la fin de thèse, et tu as rendu cela tellement plus facile et paisible. Bon courage pour ton projet doctoral, tu peux compter sur mon aide.

Merci à mes nombreux colocs, Tim, Marylène et Julie, pour votre présence au quotidien. Merci à Charlène, tu m'as beaucoup manqué depuis ton départ. J'espère qu'on se fera ce week-end à Strasbourg :).

Merci à Luc Kerjouan et à toute la Marvelous BJJ pour l'ambiance familiale du club, dans lequel je me suis senti tout de suite accueilli.

Merci aux stagiaires que j'ai pu (co)encadrer, Valentin, Matthieu et Jona. Vous nous avez vraiment bluffés par le travail que vous avez fourni, c'était un plaisir de bosser avec vous. Je vous souhaite le meilleur pour la suite.

Je remercie Maria Giovanna et Erwan Adam de croire assez en moi pour m'offrir un poste à la suite de ma thèse, et pour votre patience et bienveillance pendant mes réflexions et le processus de

recrutement. Merci à mes nouveaux collègues du SGLS/LCAN, Adrien, Pierre, Élie, Guillaume, Luc et les autres pour votre accueil chaleureux au sein de l'équipe TRUST. J'ai hâte de travailler avec vous.

Je remercie Émilie Bourne, Paolo Ricci, Antoine Hoffman, Pierre-Henri Maire, Raphaël Loubère et Sébastien Guisset pour votre aide pendant ma prospection de poste. J'espère que nous serons amenés à collaborer à l'avenir.

Je remercie mes directeurs de stages, à commencer par Alain Albouy, pour m'avoir donné la chance de pouvoir faire mes premiers pas dans le monde de la recherche en fin de L3 à l'IMCCE. Cette expérience fut très enrichissante pour moi et confirma mon désir d'évoluer dans le monde de la recherche. Merci à Thanh-ha et Bruno pour tout ce que vous m'avez appris pendant mon stage de M1 et pour m'avoir mis en contact avec la Maison de la Simulation. Je pense aussi à mes professeur.es de Matmeca, M. Turpault, M. Mieussens, Mme Bonneton, je vous remercie pour votre encadrement et pour m'avoir guidé jusqu'à la thèse. Je souhaite aussi remercier mes professeurs de lycée, Mme Cassou et M. Poudens, pour m'avoir poussé très tôt à croire en une carrière scientifique.

Table des matières

Introduction générale version française

Contexte

La convection, bien que fréquemment perçue comme un phénomène simple et largement compris — l'air chaud s'élève tandis que l'air froid descend sous l'effet de la poussée d'Archimède — dépasse en réalité le cadre de la simple convection thermique. Par exemple, la convection humide dans l'atmosphère, est régulée non seulement par des gradients de température, mais également par les variations de la concentration en vapeur d'eau de l'air, elle-même influencée par le cycle de l'eau [Manabe et Strickler 1964]. L'air se charge en humidité à la surface des océans par l'évaporation et libère cette humidité sous forme de condensation dans les nuages une fois en altitude. De plus, ces nuages sont chauffés par les radiations venant du soleil. Cette interaction entre température, concentration en eau et sources externes rend la convection humide nettement plus subtile que le cas académique de la convection de Rayleigh-Bénard. De plus, sa compréhension est cruciale pour l'analyse du changement climatique étant donné que la convection humide impacte la distribution dans l'atmosphère de la vapeur d'eau, qui est un gaz à effet de serre puissant.

La convection joue aussi un rôle crucial en astrophysique, dans la structuration à grande échelle des océans, des atmosphères, des intérieurs terrestres et des étoiles, ainsi que des exo-planètes. L'omniprésence de la convection a conduit au développement de théories physiques visant à améliorer notre compréhension de ce phénomène. L'étude pionnière de [Schwarzschild 1906] a quantifié le gradient thermique nécessaire pour déclencher l'instabilité convective dans un fluide. En notant ∇_T ce gradient et ∇_{ad} le gradient adiabatique du fluide, le critère de Schwarzschild pour l'instabilité convective s'écrit $\nabla_T - \nabla_{ad} > 0$. Une perturbation initiale de l'équilibre hydrostatique dans le fluide s'amplifiera exponentiellement si ce critère est satisfait, entraînant l'établissement de mouvements convectifs macroscopiques. De prime abord, l'étude des critères d'instabilité peut paraître vaine du point de vue des applications au sens où il est peu probable que des atmosphères à l'équilibre instable existent, en attente de perturbation. En fait, l'utilité de cet exercice vient du fait que l'analyse de stabilité linéaire permet d'obtenir des estimations sur la structure des atmosphères dans lesquelles la convection est déjà active, dans le régime non-linéaire. Cette propriété surprenante vient du fait que la convection, même dans le régime non-linéaire, reste une perturbation d'amplitude faible de l'état d'équilibre hydrostatique.

Dans [Ledoux 1947], l'auteur introduit un gradient de poids moléculaire moyen, ∇_μ, dans l'analyse. Le critère devient alors $\nabla_T - \nabla_{ad} - \nabla_\mu > 0$. On peut imaginer un gradient thermique stabilisant et un gradient de poids moléculaire moyen déstabilisant; leur somme détermine la stabilité du fluide. Dans le contexte cette l'étude, la physique stellaire, ce poids moléculaire moyen est variable en raison de la présence d'éléments lourds tels que le fer.

[Stern 1960] étudie la convection thermohaline dans les océans, où le gradient de poids moléculaire moyen correspond à la salinité de l'eau. Il observe que la convection peut être déclenchée par la diffusion de la température et de la salinité, même dans des conditions initialement stables selon le critère de Ledoux. La figure 1 illustre les gradients initiaux analysés pour lesquels nous considérons le cas d'une diffusion thermique significativement plus rapide que la diffusion du sel. Une parcelle de fluide en haut du domaine qui commence à descendre se thermalisera rapidement tout en conservant sa salinité initiale, devenant plus dense que son environnement et accélérant sa descente, donnant lieu au mouvement convectif. La diffusion est la force motrice de cette instabilité.

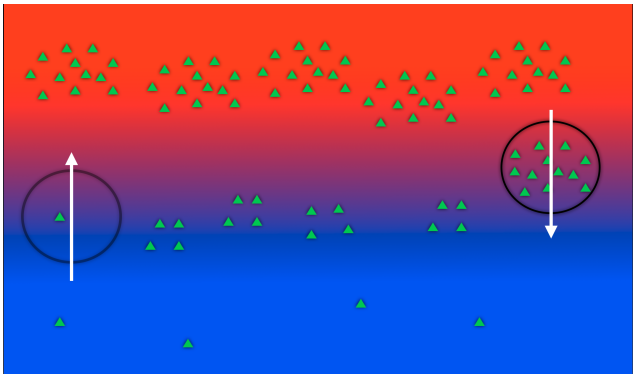


Figure 1 – Représentation de la convection double-diffusive. La température est représentée par la couleur, la salinité est représentée par la concentration en triangles verts.

Ce phénomène est également observé dans les intérieurs d'étoiles par la diffusion des éléments lourds, tels que le fer, donnant lieu à de la convection "à doigts", décrite dans [Ulrich 1972]. De même, la convection se produit dans l'atmosphère terrestre, où la composition est influencée par la concentration en vapeur d'eau ; on parle alors de convection humide [Von Bezold 1893]. Dans ce cas, la source de l'instabilité n'est pas la diffusion mais la condensation et l'évaporation de l'eau qui impacte la composition, ainsi que le pompage/relâche de chaleur latente qui impacte la température. La convection joue également un rôle fondamental dans les atmosphères des naines brunes et des exoplanètes géantes extrasolaires, comme montré dans [Tremblin et al. 2015 ; Tremblin et al. 2016 ; Tremblin et al. 2017]. Dans ces atmosphères, la convection est influencée par la composition chimique, notamment par la concentration de méthane (CH_4) et de monoxyde de carbone (CO). La convection n'y suit pas le régime de Ledoux, mais est pilotée par des termes sources. Le terme source compositionnel correspond aux réactions chimiques qui convertissent le CO en CH_4, tandis que le terme source thermique est le transfert radiatif affecté par l'opacité du CO et du CH_4. Finalement, la convection gaz/liquide peut se manifester dans les circuits de refroidissement des centrales nucléaires. L'eau en contact avec les parois du système de refroidissement peut se vaporiser si la température est trop élevée, et cette vapeur, plus légère que l'eau, est entraînée par convection. La prédiction de la génération de ces bulles et de leur transport dans les tuyères est centrale pour la sécurité des systèmes.

Depuis les travaux [Kato 1966 ; Baines et Gill 1969], la recherche théorique sur l'instabilité convective a marqué un ralentissement, la convection étant perçue comme un problème largement résolu théoriquement. Ce ralentissement s'est fait au profit de l'augmentation des études par simulation, dans un contexte d'explosion des moyens de calculs et de progrès significatifs en mathématiques appliquées. Par exemple, les travaux de cette thèse se basent sur les méthodes volume finis de [Chalons et al. 2016a ; Padioleau et al. 2019]. Ces méthodes permettent de capturer avec une grande précision le développement de l'instabilité convective, grâce à deux propriétés essentielles. La première est la précision à bas nombre de Mach, qui est caractéristique de la convection. Les méthodes volumes finis traditionnelles (à la Godunov) souffrent de forte diffusion numérique dans ce régime (sur les maillages de quadrangles et héxaédriques), ce qui les rend inefficaces pour étudier la convection. La seconde propriété est la capacité à préserver les équilibres hydrostatiques. Étant donné que la convection est un écart de faible amplitude d'un équilibre hydrostatique, il est crucial de pouvoir capturer cet équilibre avec précision afin d'analyser la croissance des perturbations sans introduire d'artéfacts numériques. Dans ce contexte, [Tremblin et al. 2019] a introduit un cadre unifié pour l'instabilité convective thermo-compositionnelle. Formel-

lement, l'analyse prend en compte des termes sources arbitraires affectant la température et la composition. Le cadre proposé permet de dériver de manière systématique les critères d'instabilité. Les bonnes propriétés des méthodes numériques utilisées ont joué un rôle crucial dans l'établissement et la vérification de l'analyse, donnant confiance aux simulations présentant des instabilités convectives nouvelles, déclenchées par la prise en compte des termes sources. Deux critères sont exposés : celui de Ledoux est retrouvé, et un nouveau critère dit diabatatique est dérivé et s'écrit $(\nabla_T - \nabla_{ad})\tau_T - \nabla_\mu \tau_X > 0$ où τ_T, τ_X sont les temps caractéristiques d'effet des termes sources sur la température et composition. Une comparaison avec le critère de Ledoux révèle que l'intensité relative des termes sources peut rendre instable un système autrement stable selon Ledoux, en modifiant l'importance relative des gradients. Ce cadre théorique englobe ainsi les différents types de convection mentionnés en table 1.

Contexte	τ_T	τ_X
Convection thermohaline	Diffusion thermique	Diffusion du sel
Atmosphères stellaires	Diffusion thermique	Diffusion des éléments lourds
Convection humide	Pompage / relâche de chaleur latente	Condensation/Évaporation
Circuits de refroidissement	Pompage / relâche de chaleur latente	Vaporisation
Naines brunes	Transfert radiatif	Réaction CO/CH_4

Table 1 – Nature des termes sources dans différents contextes de convection diabatique.

L'étude propose également une extension non-linéaire de la théorie, permettant des estimations sur la structure convective à grande échelle de ces systèmes ainsi que la calibration de flux moyens dans des codes de simulations 1D permettant la prédiction de structures atmosphériques.

Cependant, cette étude ainsi que celles que nous mentionnons plus haut sont restreintes à l'hydrodynamique. Pour l'analyse de la convection dans les plasmas stellaires, l'intégration d'un champ magnétique est essentielle. L'impact de ce champ magnétique sur le critère de Schwarzschild a fait l'objet d'études, notamment pour mieux comprendre la structure des atmosphères des étoiles magnétiques et des taches solaires en particulier, [Hughes et Proctor 1988; Gough et Tayler 1966; Newcomb 1961; Kovetz et Mestel 1967; Yu 1966]. La convection double-diffusive entre la température et le champ magnétique a également été mise en évidence dans [Yu et Cheng 1973], de manière analogue à la convection double-diffusive entre la température et la concentration d'un soluté. Un autre effet non étudié dans [Tremblin et al. 2019] est la présence de cisaillements i.e. des gradients verticaux de vitesses horizontales qui peuvent être causés par exemple par la force de Coriolis, et qui ont un effet fort sur la convection. À la lumières des découvertes permises par la prise en compte des termes sources dans la convection, cette thèse vise à explorer l'influence des champs magnétiques et du cisaillement, ainsi que de leurs termes sources associés, sur la convection diabatique. Cette thèse se concentre particulièrement sur la possibilité de découvrir de nouvelles instabilités en magnétohydrodynamique (MHD) qui pourraient être déclenchées par ces interactions, ainsi que d'enrichir notre compréhension des instabilités déjà connues. En cas de découverte de telles instabilités, l'objectif serait de les simuler et de les analyser théoriquement.

Les expériences numériques de [Tremblin et al. 2019] furent réalisées à de relativement faibles résolutions. Afin d'étudier le comportement à convergence du modèle, [Daley-Yates et al. 2021] a effectué une simulation de convection à grande échelle (4960^3) sur le supercalculateur Jean Zay de l'IDRIS. Cette simulation a été effectuée avec le code ARK [Padioleau et al. 2019] basée sur les librairies Kokkos+MPI pour la portabilité de performance et les exécutions massivement parallèles sur architectures multi-GPU/CPU. Les défis liés à la gestion de la taille des

données générées sont notables : avec une solution numérique pesant 5To, il devient impossible d'effectuer des sauvegardes fréquentes pour le post-traitement et l'interprétation physique. Des solutions telles que l'implémentation de réductions (moyennes, coupes) ont été adoptées. Cette simulation illustre le problème du "bottleneck" des entrées/sorties (E/S). En effet, les capacités de calcul augmentent beaucoup plus vite que les performances de stockage et de vitesse d'écriture sur disque [Gueroudji et al. 2021]. Face à cette problématique, le développement et l'utilisation de bibliothèques d'E/S adaptées au calcul haute performance deviennent indispensables. Une des solutions possibles au problème est le traitement in situ des données de simulations c'est-à-dire directement au cours de la simulation, au lieu de s'appuyer sur le traitement a posteriori des sauvegardes. À cet égard, des bibliothèques ont été développées au sein de la Maison de la Simulation. Notamment, la bibliothèque Deisa [Gueroudji et al. 2021] qui offre la possibilité d'employer des outils d'analyse fournis par la librairie python DASK en cours de simulation, via l'interface PDI [Roussel et al. 2017], sur des ressources de calcul différentes de celles utilisées pour la simulation.

Finalement, un axe important des travaux autour des méthodes volumes finis est le développement des méthodes d'ordre élevés qui capturent les chocs. Les chocs et discontinuités sont des phénomènes inhérents aux écoulements compressibles. Leur traitement est un défi du point de vue théorique et numérique. Les méthodes volumes finis doivent être assez précises pour capturer ces chocs et leur propagation, sans diffusion numérique excessive. Elles doivent aussi être assez stables pour ne pas générer d'oscillation parasite à partir de ces discontinuités. Trouver l'équilibre entre stabilité et précision est une tâche difficile qui fait l'objet d'une littérature importante [LeVeque et Leveque 1992 ; Toro 2001]. De nombreuses stratégies ont été développées et se basent toutes, directement ou indirectement, sur un pilotage de la diffusion numérique grâce à un choix de représentation de la solution sur la grille. Ces méthodes se distinguent en deux catégories : les limitations a priori et a posteriori. L'approche a priori consiste en l'évaluation des gradients via des fonctions non-linéaires qui garantissent la stabilité des simulations. Ces méthodes peuvent par exemple assurer le caractère TVD (Total Variation Diminishing) de la simulation. Elles diffèrent entre elles par la représentation de la solution sur la grille. On relève par exemple une représentation linéaire par morceaux (Piecewise Linear Method [Van Leer 1974]) ou les pentes sont limitées par des fonctions non-linéaires, parabolique par morceaux (Piecewise Parabolic Method [Colella et Woodward 1984 ; McCorquodale et Colella 2011]), ou d'ordre plus élevé avec les méthodes (W)-ENO (Weighted Essentially Non-Oscillatory [Liu et al. 1994 ; Gerolymos et al. 2009]) qui relaxent le caractère TVD au profit de représentations très précises, mais tout de même stables. On note aussi les méthodes basées sur des représentations en processus Gaussiens (GP-WENO [May et Lee 2024]). Plus récemment, la méthode MOOD (Multidimensional Optimal Order Detection), approche a posteriori, a été introduite [Clain et al. 2011 ; Diot et al. 2012]. Elle se base sur une boucle d'essai-correction où plusieurs représentations de stabilité croissante / précision décroissante sont appliquées les unes à la suite des autres jusqu'à ce que la solution respecte un ensemble de critères d'acceptabilité numérique. Cette méthode se démarque par sa précision au sens où elle choisit l'ordre optimal de représentation de la solution, pour un critère d'acceptabilité choisi. En revanche, elle souffre de quelques défauts, comme par exemple une parallélisation limitée. Celle-ci est due au fait que certaines zones de la simulation requièrent de multiples essais-erreurs, alors que d'autres ont un comportement valide dès la première méthode appliquée. Cela cause un déséquilibre de charge particulièrement problématique sur les architectures GPU. Des recherches actuelles explorent l'utilisation de réseaux de neurones pour prédire à priori l'ordre optimal que la méthode MOOD sélectionnerait a posteriori, afin de retrouver la parallélisabilité tout en gardant l'aspect optimal des méthodes MOOD [Bourriaud et al. 2020].

Objectifs de la thèse

Cette thèse se positionne à l'interface de la physique, de l'analyse numérique et du calcul haute performance. Elle vise principalement à étudier l'instabilité convective en MHD pour application au contexte des atmosphères stellaires. La simulation numérique étant une pierre angulaire du développement de la théorie de l'instabilité, les premiers objectifs de la thèse sont centrés sur l'amélioration des méthodes numériques et leur extension à la MHD. Le premier objectif est d'adapter le solveur de [Padioleau et al. 2019] aux méthodes d'ordre élevé afin d'en améliorer la précision. Cette adaptation est complexe car la méthode repose sur un splitting d'opérateur, difficilement combinable avec des algorithmes d'ordre élevé. Cela à tout de même été réalisé dans [Del Grosso et Chalons 2021] dans le cadre des équations de Saint Venant, mais nous cherchons ici une approche différente, plus simple pour les problèmes multi-dimensionels. Le deuxième objectif consiste à développer un solveur basé sur les mêmes outils de mathématiques appliquées mais adapté à la MHD, une tâche ardue en raison de la non-symétrie du système hyperbolique de la MHD idéale. Cette dernière entraîne l'apparition d'un terme source dans l'inégalité d'entropie, dont le signe n'est pas contrôlable et qui est proportionnel à la divergence du champ magnétique $\nabla \cdot \boldsymbol{B}$. Bien que nulle au niveau continu, cette divergence n'est pas nulle au niveau discret, causant des instabilités numériques. Notre but est de développer une solution innovante à ce problème via le cadre du splitting d'opérateur et de l'implémenter dans le code ARK. Le troisième objectif, central, vise à étendre l'analyse de stabilité linéaire et le cadre non-linéaire de [Tremblin et al. 2019] à des atmosphères possédant un champ magnétique et un profil de cisaillement de vitesse, pour identifier de nouveaux critères d'instabilité et examiner le régime non-linéaire de ces instabilités, à l'aide de simulations numériques effectuées avec le schéma que nous avons mis au point. Enfin, le quatrième objectif consiste à intégrer et utiliser des outils modernes de traitement des entrées/sorties, développés à la Maison de la Simulation (PDI, Deisa), pour surmonter le défi du bottleneck de l'E/S. Nous présenterons un Grand Challenge sur la machine Adastra, exécutant une simulation de très haute résolution de dynamo convective. Ce Grand Challenge teste les capacités de la machine ainsi que les outils d'E/S dans un contexte pré-exaflopique et permet une étude de convergence de nos estimations de dynamo convective. Le dernier objectif de cette thèse est le développement de petits réseaux de neurones pour la prédiction de l'ordre optimal au sens de la méthode GP-MOOD [Bourgeois et Lee 2022], dans la lignée de [Bourriaud et al. 2020] avec comme principale différence la nature 2D des problèmes étudiés ainsi que la considération de la méthode GP-MOOD plutôt que MOOD polynomial standard.

Description des travaux

Chapitre 1

Ce chapitre est consacré à la refonte de la méthode de séparation d'opérateurs, présentée dans [Padioleau et al. 2019] (OSLP), en une méthode de séparation de flux (FSLP). Cette dernière hérite des propriétés numériques de la méthode originale, comme la stabilité, la précision dans le régime bas Mach, et la préservation des équilibres hydrostatiques, tout en apportant des avantages supplémentaires significatifs. En premier lieu, la méthode FSLP peut s'implémenter comme une méthode volumes finis traditionnelle, basée sur une formule de flux aux interfaces, ce qui rend directe sa combinaison avec les méthodes d'ordre élevés. Elle se distingue de OSLP par un stencil plus compact et l'absence de nécessité de stocker un état intermédiaire, minimisant ainsi son empreinte mémoire. La stabilité de la méthode est démontrée sous une condition CFL, établie via un argument de combinaison convexe. Le nouveau schéma est interprété de multiples façons : comme une modification de l'étape de

transport dans OSLP, une combinaison convexe de mises à jour, et comme une méthode de relaxation. Nous étendons l'approche au second ordre et aux problèmes 2D, validant la méthode par une série de tests hydrodynamiques qui confirment sa stabilité et sa précision.

Chapitre 2

Le second chapitre aborde le développement d'un solveur volumes finis, centré et robuste pour la MHD idéale en multi-D. Le solveur est basé sur une séparation des flux similaire à celle du chapitre 1 et à des techniques de relaxation tirées de [Bouchut et al. 2007, 2010]. La séparation des flux de transport et magnéto-acoustique permet un traitement particulier (diffusif) de la composante normale du champ magnétique, résultant en une stabilité accrue par rapport aux méthodes volumes finis standard pour la MHD. Nous introduisons des formules spécifiques pour les vitesses de relaxation visant à garantir l'isotropie de la diffusion numérique. Une version entropique du solveur est proposée, elle inclut un terme source dit "de Powell" dans l'équation d'induction. Ce terme source vient restaurer l'inégalité d'entropie aussi bien au niveau discret que continu et rend la méthode stable dans les zones à bêta plasma arbitrairement faible, et améliore la précision dans celles à haut nombre d'Alfvén, au prix de la conservation du champ magnétique. Sans ce terme source, le solveur de relaxation conservatif présente une robustesse supérieure par rapport aux schémas de transport contraint et de divergence cleaning dans les tests à faible plasma bêta, mais ne permet pas d'effectuer des simulations stables à très bas plasma bêta. Nous proposons donc une stratégie hybride, où le terme source de Powell est employé uniquement dans les zones "difficiles" (bas plasma bêta, haut nombre d'Alfvén). Des expériences numériques confirment la stabilité de notre approche, y compris dans des conditions de très bas plasma bêta, sans que l'absence de traitement de la divergence du champ magnétique affecte la stabilité ou la validité des solutions.

Chapitre 3

Dans ce chapitre, nous élargissons l'analyse de stabilité linéaire de [Tremblin et al. 2019] aux plasmas et aux écoulements cisaillés en intégrant un champ magnétique horizontal et un gradient de vitesse verticale initiaux. Nous établissons trois critères d'instabilité, qui généralisent les critères existants, tels que ceux de Ledoux et de la convection double-diffusive, et un nouveau critère lié à un couplage de second ordre entre termes sources (convection triple diffusive). L'impact du cisaillement et des champs magnétiques sur l'instabilité convective est examiné, et nous proposons une extension non-linéaire de notre théorie, offrant des estimations pour les paramètres convectifs en régime saturé. Des expériences numériques, basées sur la méthode volumes finis du chapitre 2, dans les régimes linéaire et non-linéaire, valident notre analyse. Nous observons que la configuration géométrique du domaine influence significativement les résultats. En particulier, les écoulements dans des domaines cubiques semblent moins affectés par le cisaillement. Des simulations de dynamo convective sont également menées, liant l'intensité de la dynamo à nos prédictions théoriques non-linéaires. Enfin, nous utilisons les résultats de notre simulation de dynamo convective à très grande échelle, sur un maillage de 4096^3 cellules, effectuée sur le supercalculateur Adastra (CINES, Montpellier, France) pour étudier le comportement à convergence de notre système.

Chapitre 4

Ce chapitre détaille les aspects techniques liés à l'exécution de notre simulation de dynamo convective à grande échelle sur le supercalculateur Adastra. Ce système, classé 11e sur la liste Top500 et 3e sur la Green500, partage l'architecture du supercalculateur Frontier, avec moins de nœuds. En particulier, nous détaillons le couplage de ARK avec la librairie PDI pour le traitement des divers E/S que nous avons mis en place, motivés par

l'interprétation physique, à savoir des moyennes globales, des coupes et des profils verticaux. De plus, nous avons utilisé ce Grand Challenge comme opportunité pour montrer le bon fonctionnement de la librairie Deisa dans un contexte pré-exaflopique en exécutant une transformée de Fourier in situ, pendant la simulation, afin d'obtenir le spectre de puissance de notre expérience de convection. Finalement, nous présentons une étude de performance du code ARK sur différents types de GPUs, y compris ceux d'Adastra, ainsi qu'une étude de weak scaling.

Chapitre 5

Le dernier chapitre résulte d'un séjour de recherche de trois mois à l'Université de Californie à Santa Cruz, sous la supervision du Prof. Dongwook Lee. Il n'est pas dans la continuité directe des travaux sur la convection. Nous y explorons l'emploi de petits réseaux de neurones pour prédire à priori l'ordre de reconstruction optimal au sens de la méthode GP-MOOD [Bourgeois et Lee 2022] et d'en améliorer la parallélisabilité. Le cadre des méthodes volumes finis basées sur des processus Gaussiens est rappelé et les méthodes MOOD sont réintroduites. Nous décrivons la conception et l'intégration de réseaux de neurones dans la boucle GP-MOOD ainsi que notre procédure d'entraînement. Nous proposons une version naïve, très simplifiée d'apprentissage en ligne comme preuve de concept, plutôt qu'une approche de type "boîte noire" comme celle de [Bourriaud et al. 2020]. Nos résultats ne sont pas compétitifs avec les méthodes d'ordres élevés standard, mais offrent une voie de recherche intéressante.

Publications et communications

Les travaux effectués durant cette thèse ont fait l'objet de publications et ont été présentés à plusieurs conférences internationales, à savoir

- Mini-symposium, CANUM, Evian les Bains, France, Juin 2022, *An all-regime, well-balanced, positive and entropy satisfying one-step finite volume scheme for the Euler's equations of gas dynamics with gravity*.
- Séminaire, *Beyond Boussinesq* Workshop, Lyon, France, Octobre 2023, *Finite volume methods for compressible convection*.
- Séminaire, Swiss Plasma Center, June 2023, *A very large-scale convective dynamo Simulation powered by Kokkos and PDI*.
- [Bourgeois et al. 2024], *Recasting an operator splitting solver into a standard finite volume flux-based algorithm. The case of a Lagrange-projection-type method for gas dynamics.* Accepté dans Journal of computational physics, Janvier 2024.
- 2nd auteur de [Tremblin et al. 2024], *A multi-dimensional, robust, and cell-centered finite volume scheme for the ideal MHD equations.* Soumis à Journal of computational physics.
- *DynoStar : Simulation à très grand échelle de dynamo convective dans dans les atmosphères d'étoiles*, Grands challenges Adastra 2022

De plus les travaux du chapitre 3 font l'objet d'un article en cours de préparation.

General introduction english version

Context

Convection, although often perceived as a simple phenomenon and widely understood — hot air rises while cold air falls under the effect of buoyancy — actually extends beyond the scope of simple thermal convection. For example, moist convection in the atmosphere is governed by temperature gradients and variations in the water vapor concentration in the air, which is influenced by the water cycle [Manabe et Strickler 1964]. The air picks up moisture through evaporation at the ocean surface and releases it as condensation in clouds once it rises. Moreover, these clouds are heated by radiation from the sun. This interaction between temperature, water concentration, and external sources makes moist convection significantly more subtle than the academic case of Rayleigh-Bénard convection. Furthermore, its understanding is crucial for analyzing climate change since moist convection impacts the distribution of water vapor, a powerful greenhouse gas, in the atmosphere.

Convection also plays a crucial role in astrophysics, in the large-scale structuring of oceans, atmospheres, terrestrial interiors, stars, and exoplanets. This wide range of applications has led to the development of physical theories to improve our understanding of this phenomenon. The pioneering study by [Schwarzschild 1906] quantified the thermal gradient necessary to trigger convective instability in a fluid. Denoting ∇_T this gradient and ∇_{ad} the fluid's adiabatic gradient, the Schwarzschild criterion for convective instability is written $\nabla_T - \nabla_{ad} > 0$. An initial perturbation of the hydrostatic equilibrium in the fluid will amplify exponentially if this criterion is met, leading to macroscopic convective movements. At first glance, studying instability criteria might seem pointless from an application standpoint. Indeed, it is unlikely that real atmospheres are in unstable, unperturbed equilibrium. The interest of this exercise comes from the fact that linear stability analysis can help estimate the structure of atmospheres in which convection is already active in the non-linear regime. This surprising property comes from the fact that convection remains a weak amplitude perturbation of a hydrostatic equilibrium state, even in the non-linear regime.

In [Ledoux 1947], the author introduces a mean molecular weight gradient, ∇_μ, into the analysis. The criterion then becomes $\nabla_T - \nabla_{ad} - \nabla_\mu > 0$. One might consider a stabilizing thermal gradient and a destabilizing molecular weight gradient; by adding them together, we can predict the fluid's stability. In the stellar physics context of this study, this mean molecular weight is variable due to the presence of heavy elements such as iron.

[Stern 1960] studies thermohaline convection in the oceans, where the mean molecular weight gradient corresponds to the water's salinity. The study observes that convection can be triggered by the diffusion of temperature and salinity, even under initially stable conditions according to the Ledoux criterion. Figure 2 illustrates the initial gradients considered for analysis for which we consider the case of significantly faster thermal diffusion than salt diffusion. A fluid parcel at the top of the domain that starts descending will quickly thermalize while retaining its initial salinity, it will become denser than its surroundings and accelerating its descent, giving rise to convective movement. Diffusion is the driving force behind this instability.

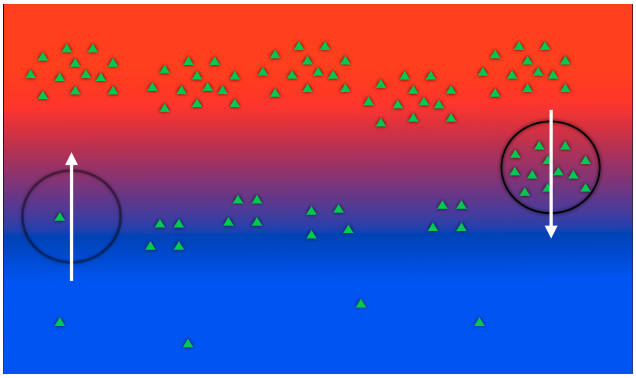


Figure 2 – Representation of double-diffusive convection. Temperature is represented by color, and the concentration of green triangles represents salinity.

This phenomenon, known as 'fingering convection,' also occurs in the interiors of stars due to the diffusion of heavy elements like iron, described in [Ulrich 1972]. Similarly, convection occurs in the Earth's atmosphere, where water vapor concentration influences the composition; we then speak of moist convection [Von Bezold 1893]. In this case, the source of the instability is not diffusion but condensation and evaporation of water that impacts the composition, as well as the pumping/release of latent heat that impacts the temperature. Convection also plays a fundamental role in the atmospheres of brown dwarfs and giant exoplanets, as shown in [Tremblin et al. 2015; Tremblin et al. 2016; Tremblin et al. 2017]. In these atmospheres, convection is influenced by chemical composition, notably by the concentration of methane (CH_4) and carbon monoxide (CO). Convection does not follow the Ledoux regime, but is driven by source terms. The compositional source term corresponds to chemical reactions that convert CO to CH_4, while the thermal source term is radiative transfer affected by the opacity of CO and CH_4. Finally, gas/liquid convection can occur in the cooling circuits of nuclear power plants. Water in contact with the walls of the cooling system can vaporize if the temperature is too high, and this vapor, being lighter than liquid water, is transported by convection. Predicting the generation of these bubbles and their transport in the nozzles is crucial to the safety of the systems. Since the works [Kato 1966; Baines et Gill 1969], theoretical research on convective instability has slowed down, as convection was seen as a problem entirely solved theoretically. This slowdown happened to the benefits of an increasing number of simulation studies in the context of an explosion of computing power and significant progress in applied mathematics. For instance, the volume methods of [Chalons et al. 2016a; Padioleau et al. 2019] are central to this thesis. These methods precisely capture the development of convective instability, thanks to two essential properties. The first is precision at low Mach numbers which characterize convection. Traditional Godunov-type finite volume methods suffer from strong numerical diffusion in this regime (on quadrangular and hexahedron meshes), which makes them ineffective for studying convection. The second property is the ability to preserve hydrostatic balances, or "well-balanced property". Since convection is a deviation of small amplitude from a hydrostatic equilibrium, it is crucial to precisely capture this equilibrium to analyze the growth of the perturbation without introducing numerical artifacts. In this context, [Tremblin et al. 2019] introduced a unified framework for thermo-compositional convective instability. Formally, the analysis considers arbitrary source terms affecting temperature and composition. The proposed framework allows a systematic derivation of the instability criteria. The good properties of these numerical methods have been crucial in establishing and verifying the analysis, building confidence in simulations that revealed new convective instabilities triggered by the source terms. Two criteria are exposed. The Ledoux criterion, and a

new so-called "diabatic criterion" that reads $(\nabla_T - \nabla_{ad})\tau_T - \nabla_\mu \tau_X > 0$ where τ_T, τ_X are the characteristic times of the effect of source terms on temperature and composition. A comparison with the Ledoux criterion reveals that the relative intensity of the source terms can make a system otherwise stable according to Ledoux, unstable, by altering the relative importance of the gradients. This theoretical framework thus encompasses the different types of convection mentioned in table 2.

Context	τ_T	τ_X
Thermohaline convection	Thermal diffusion	Salt diffusion
Stellar atmospheres	Thermal diffusion	Diffusion of heavy elements
Moist convection	Pumping / release of latent heat	Condensation/Evaporation
Cooling circuits	Pumping / release of latent heat	Vaporization
Brown dwarfs	Radiative transfer	CO/CH_4 reaction

Table 2 – Nature of the source terms in different contexts of diabatic convection.

The study also proposes a non-linear extension of the theory, allowing estimates on the large-scale convective structure of these systems and the calibration of mean fluxes in 1D simulation codes enabling the prediction of atmospheric structures.

The study we mentioned so far are limited to hydrodynamics. Integrating a magnetic field is essential for the analysis of convection in stellar plasmas. The impact of this magnetic field on the Schwarzschild criterion has been the subject of several studies, in order to better understand the structure of the atmospheres of magnetic stars and solar spots in particular [Hughes et Proctor 1988; Gough et Tayler 1966; Newcomb 1961; Kovetz et Mestel 1967; Yu 1966]. Double-diffusive convection between temperature and magnetic field has also been studied in [Yu et Cheng 1973] analogous to the double-diffusive convection between temperature and composition. Another effect not studied in [Tremblin et al. 2019] is the presence of shear, i.e., vertical gradients of horizontal velocities that can be caused, for example, by the Coriolis force, that have a strong effect on convection. In light of the discoveries enabled by considering source terms in convection, this thesis aims to explore the influence of magnetic fields and shear, as well as their associated source terms, on diabatic convection. This thesis focuses particularly on the possibility of discovering new magnetohydrodynamics (MHD) instabilities that could be triggered by these interactions, as well as enriching our understanding of already known instabilities. If new instabilities are discovered, the goal would be to simulate and theoretically analyze them.

The numerical experiments from [Tremblin et al. 2019] were conducted at relatively low resolutions. In order to study the model's convergence behavior, [Daley-Yates et al. 2021] performed a large-scale convection simulation on a 4960^3-cell mesh, on the Jean Zay supercomputer at IDRIS. This simulation was carried out with the ARK code [Padioleau et al. 2019] based on the Kokkos+MPI libraries for performance portability and massively parallel executions on multi-GPU/CPU architectures. The challenges of managing the large data volumes generated are significant : with a numerical solution of 5TB, frequent backups for post-processing and physical interpretation become impossible. Solutions such as implementing reductions (averages, slices) have been adopted. This simulation illustrates the "bottleneck" problem of input/output (I/O). Indeed, computing capabilities are increasing much faster than storage performance and disk write speed [Gueroudji et al. 2021]. In response to this issue, developing and utilizing I/O libraries tailored for high-performance computing is crucial. One possible solution to the problem is the in situ processing of simulation data, i.e., directly during the simulation, rather than relying on post-processing of backups. In this regard, libraries have been developed at Maison de la Simulation. Notably,

the Deisa library [Gueroudji et al. 2021] offers the possibility of using analysis tools provided by the DASK python library during the simulation, via the PDI interface [Roussel et al. 2017], on separate computational ressources than the ones used for the simulation.

Finally, an important aspect of finite volume methods is the development of shock-capturing high-order methods. Shocks and discontinuities are inherent phenomena in compressible flows. Handling them is a challenge from a theoretical and numerical point of view. Finite volume methods must be accurate enough to capture these shocks and their propagation, without excessive numerical diffusion. They must also be stable enough not to generate spurious oscillations from these discontinuities. Finding the balance between stability and accuracy is a difficult task that has driven significant literature [LeVeque et Leveque 1992; Toro 2001]. Many strategies have been developed and are all based on controlling numerical diffusion through a choice of solution representation on the grid. These methods fall into two categories : a priori and a posteriori limitations. The a priori approach involves evaluating gradients via non-linear functions that guarantee simulation stability. These methods may, for example, ensure the Total Variation Diminishing (TVD) character of the simulation. They differ by the representation of the solution on the grid. For example, a piecewise linear representation (Piecewise Linear Method [Van Leer 1974]) where slopes are limited by non-linear functions, piecewise parabolic (Piecewise Parabolic Method [Colella et Woodward 1984; McCorquodale et Colella 2011]), or higher order with methods such as (W)-ENO (Weighted Essentially Non-Oscillatory [Liu et al. 1994; Gerolymos et al. 2009]) which relax the TVD character for very precise, yet stable representations. There are also methods based on Gaussian process representations (GP-WENO [May et Lee 2024]). More recently, the MOOD (Multi-dimensional Optimal Order Detection) method, an a posteriori approach, was introduced [Clain et al. 2011; Diot et al. 2012]. It relies on a trial-and-error loop where several representations of increasing stability / decreasing accuracy are applied one after the other until the solution verifies a set of numerical acceptability criteria. This method stands out for its accuracy in that it chooses the optimal order of representation of the solution, according to the selected acceptability criterion. However, it suffers from a few drawbacks, such as limited parallelization. This is because certain simulation areas require multiple trial-and-errors, while others have valid behavior from the first method applied. This causes a particularly problematic load imbalance on GPU architectures. Current research is exploring the use of neural networks to predict a priori the optimal order that the MOOD method would select a posteriori, in order to restore parallelizability while maintaining the optimal aspect of the MOOD methods [Bourriaud et al. 2020].

Aim of the thesis

This thesis is at the interface of physics, numerical analysis, and high-performance computing. It primarily aims to study convective instability in MHD. As numerical simulation are central to guide theoretical development, the initial objectives of the thesis are focused on improving numerical methods and extending them to MHD. The first objective is to adapt the solver from [Padioleau et al. 2019] to high-order methods to enhance accuracy. This adaptation is complex as the method relies on an operator splitting, which is difficult to combine with high-order algorithms. This has been accomplished in [Del Grosso et Chalons 2021] for the Saint Venant equations, but we hope for a different, simpler approach for multi-dimensional problems. The second objective is to derive a solver for ideal MHD based on the same applied mathematics tools. This task is challenging due to the non-symmetry of the ideal MHD hyperbolic system. This results in a source term in the entropy inequality, whose sign is not controlable and proportional to the magnetic field divergence $\nabla \cdot \boldsymbol{B}$. Although it is 0 at the continuous level, this divergence is not zero at the discrete level, potentially causing numerical instabilities. We aim to develop an inno-

vative solution to this problem via the operator splitting framework and implement it in the ARK code. The third, central objective is to extend the linear stability analysis and non-linear framework of [Tremblin et al. 2019] to atmospheres presenting a magnetic field and a velocity shear profile, to identify new instability criteria and examine the non-linear regime of these instabilities, using the numerical methods we developed. Lastly, the fourth objective is to integrate and use modern input/output handling tools, developed at Maison de la Simulation (PDI, Deisa), to overcome the I/O bottleneck challenge. We will present a Grand Challenge on the Adastra machine, running a high-resolution convective dynamo simulation. This Grand Challenge tests the capabilities of the machine and the I/O tools in a pre-exascale context and allows a convergence study on our estimates of convective dynamo. The final objective of this thesis is the development of small neural networks for predicting the optimal order in the sense of the GP-MOOD method [Bourgeois et Lee 2022], following [Bourriaud et al. 2020] but with specific focus on GP-MOOD in a 2D context rather than standard polynomial MOOD.

Description of the work

Chapter 1

This chapter is dedicated to the recasting of the operator splitting method, presented in [Padioleau et al. 2019] (OSLP), into a flux splitting method (FSLP). This new method inherits the numerical properties of OSLP, such as stability, accuracy in low Mach regimes, and preservation of hydrostatic balances, along with new interesting properties. The FSLP method can be implemented as a traditional finite volume method, based on an interface flux formula, which can be combined with high-order methods. It differs from OSLP because it has a more compact stencil and does not need to store an intermediate state, thus minimizing its memory footprint. The method's stability is prooved under a CFL condition, established through a convex combination argument. The new scheme is interpreted in multiple ways : a modification of the transport step of OSLP, a convex combination of updates, and a relaxation method. We extend the approach to second-order and 2D problems, validating the method through hydrodynamic tests and confirming its stability and accuracy.

Chapter 2

The second chapter discusses the development of a centered and robust finite volume solver for ideal MHD in multi-D. The solver is based on a similar flux separation as Chapter 1 and relaxation techniques from [Bouchut et al. 2007, 2010]. The separation of transport and magneto-acoustic fluxes allows a particular (diffusive) treatment of the normal component of the magnetic field, resulting in increased stability compared to standard finite volume methods for MHD. We introduce specific relaxation speed formulas to ensure the isotropy of numerical diffusion. An entropy satisfying version of the solver is proposed, which includes a "Powell" source term in the induction equation. This source term restores the entropy inequality at both discrete and continuous levels. It makes the method stable in regions with arbitrarily low plasma beta, and improves accuracy in high Alfvén number regions, at the cost of magnetic field conservation. Without this source term, the conservative relaxation solver shows better robustness than constrained transport and divergence cleaning schemes in low plasma beta tests but does not allow stable simulations at very low plasma beta. We therefore propose a hybrid strategy, where the entropic correction is used only in "difficult" areas (low plasma beta, high Alfvén number). Numerical experiments confirm the stability of our approach, even in very low plasma beta regions, without the absence of magnetic field divergence treatment affecting the stability or validity of the solutions.

Chapter 3

This chapter extends the linear stability analysis from [Tremblin et al. 2019] to plasmas and sheared flows by integrating an initial horizontal magnetic field and vertical speed gradient. We establish three instability criteria, that generalize existing criteria such as those of Ledoux and double-diffusive convection, and a new criterion related to second-order coupling between source terms (triple diffusive convection). The impact of shear and magnetic fields on convective instability is examined, and we propose a non-linear extension of our theory, offering estimates for convective parameters in the saturated regime. Numerical experiments, based on the finite volume method of chapter 2, in the linear and non-linear regimes, validate our analysis. We observe that the geometry of the domain significantly influences the results. In particular, flows in cubic domains seem less affected by shear. Convective dynamo simulations are also conducted, linking dynamo intensity to our non-linear theoretical predictions. Finally, we use the results of our very high-resolution convective dynamo simulation on a 4096^3-cell grid mesh performed on the Adastra supercomputer (CINES, Montpellier, France) to study the convergence behavior of our system.

Chapter 4

This chapter details the technical aspects of running our large-scale convective dynamo simulation on the Adastra supercomputer. This system, ranked 11th in the Top500 list and 3rd in the Green500, shares the architecture of the Frontier supercomputer, with fewer nodes. In particular, we detail the coupling of ARK with the PDI library for handling the various I/Os, motivated by physical interpretation, namely global averages, slices, and vertical profiles. Additionally, we used this Grand Challenge as an opportunity to demonstrate the functionality of the Deisa library in a pre-exascale context by performing an in situ Fourier transform during the simulation to obtain the power spectrum of our convection experiment. Finally, we present a performance study of the ARK code on different types of GPUs, including those on Adastra, and a weak scaling study.

Chapter 5

The final chapter results from a three-month research visit at the University of California, Santa Cruz, under the supervision of Prof. Dongwook Lee. It is not directly related to the study of convection. We explore the use of small neural networks to predict the optimal reconstruction order a priori in the sense of the GP-MOOD method [Bourgeois et Lee 2022] in order to improve its parallelizability. The framework of Gaussian process-based finite volume methods is recalled, and the MOOD methods are reintroduced. We describe the design and integration of neural networks into the GP-MOOD loop and our training procedure. We propose a naive, greatly simplified online learning version as proof of concept, rather than a "black box" approach like that of [Bourriaud et al. 2020]. Our results are not competitive with standard high-order methods, but they offer an interesting research avenue.

Publications and communications

The work conducted during this thesis has led to several publications and presentations at various international conferences, as follows :

- Mini-symposium, CANUM, Evian les Bains, France, June 2022, *An all-regime, well-balanced, positive and entropy satisfying one-step finite volume scheme for the Euler's equations of gas dynamics with gravity*.
- Seminar, *Beyond Boussinesq* Workshop, Lyon, France, October 2023, *Finite volume methods for compressible convection*.

- Seminar, Swiss Plasma Center, June 2023, *A very large-scale convective dynamo Simulation powered by Kokkos and PDI*.
- [Bourgeois et al. 2024], *Recasting an operator splitting solver into a standard finite volume flux-based algorithm. The case of a Lagrange-projection-type method for gas dynamics.* Accepted in Journal of Computational Physics, January 2024.
- 2nd author of [Tremblin et al. 2024], *A multi-dimensional, robust, and cell-centered finite volume scheme for the ideal MHD equations.* Submitted to Journal of Computational Physics.
- *DynoStar : Very large-scale simulation of convective dynamo in stellar atmospheres*, Adastra Grand Challenges 2022.

Additionally, an article based on the material of Chapter 3 is in preparation.

1 - Recasting an operator splitting solver into a standard finite volume flux-based algorithm. The case of a Lagrange-projection-type method for gas dynamics

1.1 . Introduction

In this chapter, we consider the approximation of the compressible Euler equations in the presence of source terms derived from a smooth potential using a finite volume method. We aim to showcase the recasting of an Operator Splitting Lagrange-Projection (OSLP) finite volume algorithm into a corresponding flux-splitting method (FSLP). The original OSLP algorithm is well-suited for studying convection and was used to perform the numerical simulation of [Tremblin et al. 2019; Daley-Yates et al. 2021]. The flux-splitting method we consider here has several computational and implementation advantages compared to OSLP. It requires a smaller stencil, no intermediate state storage, and can be implemented as a fully explicit flux-based solver. The simplicity of the FSLP method allows us to combine our method effortlessly with standard means to derive higher-order methods such as MUSCL, ENO, WENO, and MOOD frameworks.

The OSLP algorithm we use as ground material for implementing an FSLP method is presented in [Padioleau et al. 2019]. It relies on a separate treatment of acoustic and transport effects, and it enjoys several interesting properties : it is stable under a CFL condition so that it ensures positivity for mass and internal energy and satisfies a discrete entropy inequality. The treatment of the source term in [Padioleau et al. 2019] allows to preserve stationary solution profiles at the discrete level so that the OSLP scheme satisfies a well-balanced property (see *e.g.* [Gosse et Le Roux 1996; Greenberg et Leroux 1996; LeVeque 1998; Gosse 2000; Gosse et Toscani 2004; Audusse et al. 2004; Lukáčová-Medvid'ová et al. 2007; Noelle et al. 2007; Castro Díaz et al. 2007; Pelanti et al. 2008; Gosse 2013; Käppeli et Mishra 2014; Chandrashekar et Klingenberg 2015; Desveaux et al. 2016; Chalons et al. 2016b; Michel-Dansac et al. 2016, 2016; Castro et al. 2017; Chertock et al. 2018; Padioleau et al. 2019; Castro et Parés 2020; Morales de Luna et al. 2020; Berberich et al. 2021; Del Grosso et Chalons 2021]). Moreover, when the Mach number that characterizes the ratio of the material velocity to the sound velocity is low, cell-centered finite volume methods may suffer an important loss of accuracy [Turkel 1987; Guillard et Viozat 1999; Guillard et Murrone 2004; Dellacherie 2010]. This question is connected to several delicate issues like the influence of the mesh geometry [Rieper et Bader 2009; Dellacherie 2010], the numerical diffusion (see for example [Dauvergne et al. 2008; Dellacherie 2010; Chalons et al. 2016a; Dellacherie et al. 2016; Zakerzadeh 2016; Barsukow 2021]) or the Asymptotic Preserving property with respect to incompressible models [P. Degond et M. Tang 2011; Cordier et al. 2012; Zakerzadeh 2016; Bispen et al. 2017; Berthon et al. 2020; Dimarco et al. 2017; Boscarino et al. 2018; Bouchut et al. 2020b] and has been extensively investigated in the literature for the past years through several approaches (see also [Paillere et al. 2000; Guillard et Murrone 2004; Beccantini et al. 2008; Dimarco et al. 2018; Boscheri et al. 2020; Bouchut et al. 2020a; Zeifang et al. 2020; Bruel et al. 2019]). Although it does not address the full spectrum of problems connected to the simulation of flows in the low Mach regime, a simple modification of the OSLP method ensures a uniform truncation error with respect to the Mach number [Chalons et al. 2016a; Padioleau et al. 2019]. These properties are very useful for performing numerical simulations of convection. Indeed, convection in both the linear and non linear regimes is characterized by a small Mach number and is a small perturbation around an hydrostatic equilibrium state. The resulting FSLP algorithm presented in this paper

performs equally concerning these aspects. Moreover, it profits from all the advantages of FSLP methods over OSLP mentioned above. It is also less computationally expensive in the low Mach regime, requiring fewer sweeps over the numerical solution to reach the same physical time. The derivation of the stability properties of the FSLP method requires novel mathematical developments that are shown thereafter.

The chapter is organized as follows : we first introduce the set of equations with the thermodynamical related hypotheses that support the stability properties of the model, and we present the stationary profiles and difficult regimes we will be interested in. Then, we will recall the OSLP method that we aim to recast into its FSLP version. We will modify the transport step in the original OSLP method so that both steps are revamped into one that can be viewed as a flux-splitting step. We will then provide a proof of stability for the FSLP method. We examine standard ways to extend the FSLP method to higher-order discretizations and multi-dimensional problems. Then we will see that the FSLP method can be connected to a new relaxation approximation of the Euler equations that proposes a single-step but separate treatment of the acoustic and transport effects. Finally, we will present one-dimensional and two-dimensional numerical experiments that demonstrate the good behavior of the scheme.

1.2 . Flow model

For the sake of clarity but without loss of generality, we focus on one-dimensional problems. We consider the Euler equations supplemented with a smooth potential source term $x \mapsto \phi(x)$,

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{S}(\boldsymbol{U}, \phi), \qquad \text{for } x \in \mathbb{R},\, t > 0, \tag{1.1}$$

with $\boldsymbol{U} = (\rho, \rho u, \rho E)^T$, $\boldsymbol{F}(\boldsymbol{U}) = (\rho u,\ u\rho u + p,\ u\rho E + pu)^T$ and $\boldsymbol{S}(\boldsymbol{U}, \phi) = -\rho \partial_x \phi (0, 1, u)^T$ where ϕ is smooth enough so that we can consider that $\partial_x \phi$ is also regular and bounded.

Although (1.1) is not strictly limited to flows accounting for gravitational forces, the stationary potential $x \mapsto \phi(x)$ will be referred to as the gravitational potential. The fields ρ, u, p, and E respectively denote the density, velocity, pressure, and specific total energy of the fluid. If $e = E - u^2/2$ is the specific internal energy, we define the set of admissible states

$$\Omega = \left\{ (\rho, \rho u, \rho E) \in \mathbb{R}^3 \,\middle|\, \rho > 0,\ e > 0 \right\}. \tag{1.2}$$

Let s be the specific entropy of the fluid. We consider an Equation of state (EOS) in the form of a mapping $(1/\rho, s) \mapsto e^{\text{EOS}}(1/\rho, s)$ that satisfies the classic Weyl assumptions [Weyl 1949 ; Chalons et al. 2016a] :

$$\frac{\partial e^{\text{EOS}}}{\partial (1/\rho)} < 0, \qquad \frac{\partial e^{\text{EOS}}}{\partial s} > 0, \qquad \frac{\partial^2 e^{\text{EOS}}}{\partial (1/\rho)^2} > 0, \tag{1.3a}$$

$$\frac{\partial^2 e^{\text{EOS}}}{\partial s^2} > 0, \qquad \left[\frac{\partial^2 e^{\text{EOS}}}{\partial (1/\rho)^2}\right]\left[\frac{\partial^2 e^{\text{EOS}}}{\partial s^2}\right] > \left[\frac{\partial^2 e^{\text{EOS}}}{\partial s \partial (1/\rho)}\right]^2, \qquad \frac{\partial^3 e^{\text{EOS}}}{\partial (1/\rho)^3} < 0. \tag{1.3b}$$

The temperature T and the pressure p of the fluids are related to the other parameters, respectively by $T = T^{\text{EOS}}(1/\rho, s) = \partial e^{\text{EOS}}/\partial s$ and $p = p^{\text{EOS}}(1/\rho, s) = -\partial e^{\text{EOS}}/\partial (1/\rho)$. It is possible to define a mapping $(1/\rho, e) \mapsto s^{\text{EOS}}(1/\rho, e)$ such that $e = e^{\text{EOS}}(1/\rho, s)$ if $s = s^{\text{EOS}}(1/\rho, e)$ so that we have the Gibbs relation

$$\mathrm{d}e + p\,\mathrm{d}(1/\rho) = T\mathrm{d}s. \tag{1.4}$$

Note that (1.3) imply that $-s^{\text{EOS}}(1/\rho, e)$ and $e^{\text{EOS}}(1/\rho, s)$ are strictly convex functions. Relations (1.3) also ensure that

$$\frac{\partial p^{\text{EOS}}}{\partial(1/\rho)}(1/\rho, s) < 0, \tag{1.5}$$

so that the sound velocity $c = \rho^{-1}\sqrt{-\partial p^{\text{EOS}}(1/\rho, s)/\partial(1/\rho)}$ is real valued. Let us recall now that the dimensionless quantity $\mathrm{Ma} = |u|/c$ is called the Mach number. We also make the classic assumption [Callen 1985] that

$$\mathscr{M}\, s(\mathscr{V}/\mathscr{M}, \mathscr{E}/\mathscr{M}) = S(\mathscr{M}, \mathscr{V}, \mathscr{E}), \tag{1.6}$$

where the (non-specific) entropy $(\mathscr{M}, \mathscr{V}, \mathscr{E}) \mapsto S(\mathscr{M}, \mathscr{V}, \mathscr{E})$ is a strictly concave homogeneous first-order function. Let us note that as

$$\frac{\partial S}{\partial \mathscr{E}}(\mathscr{M}, \mathscr{V}, \mathscr{E}) = \frac{\partial s}{\partial e}(\mathscr{V}/\mathscr{M}, \mathscr{E}/\mathscr{M}) = 1/T^{\text{EOS}}(\mathscr{V}/\mathscr{M}, \mathscr{E}/\mathscr{M}) > 0,$$

then $\mathscr{E} \mapsto S(\bar{\mathscr{M}}, \bar{\mathscr{V}}, \mathscr{E})$ is a strictly increasing function for a fixed $\mathscr{M}$ and $\mathscr{V}$.

Weak solutions of (1.1) also satisfy the entropy inequality

$$\partial_t(\rho s) + \partial_x(u\rho s) \geq 0, \tag{1.7}$$

where the inequality (1.7) is indeed an equality in the case of smooth solutions (see [Smoller 1983; R.J. LeVeque 2002; Godlewski et Raviart 1990; Serre 1999]).

We also are interested in the study of particular steady-state solutions of (1.1) called the hydrostatic equilibria that are classically defined by

$$\partial_x p = -\rho\partial_x\phi, \qquad\qquad u = 0. \tag{1.8}$$

For many years, significant efforts have been dedicated to developing so-called well-balanced numerical methods (see *e.g.* [Gosse et Le Roux 1996; Greenberg et Leroux 1996; LeVeque 1998; Gosse 2000; Gosse et Toscani 2004; Audusse et al. 2004; Lukáčová-Medvid'ová et al. 2007; Noelle et al. 2007; Castro Díaz et al. 2007; Pelanti et al. 2008; Gosse 2013; Käppeli et Mishra 2014; Chandrashekar et Klingenberg 2015; Desveaux et al. 2016; Chalons et al. 2016b; Michel-Dansac et al. 2016, 2016; Castro et al. 2017; Chertock et al. 2018; Padioleau et al. 2019; Castro et Parés 2020; Morales de Luna et al. 2020; Berberich et al. 2021; Del Grosso et Chalons 2021]) that allow preserving discrete equivalents of equilibrium solutions like (1.8). In the present work, we intend to investigate well-balanced finite volume approximations of (1.1) that are compatible with discrete equivalents of (1.7) and ensure that the fluid states $(\rho, \rho u, \rho E)$ remain in Ω.

Before going any further, let us introduce the notations for our space-time discretization : we consider a strictly increasing sequence $(x_{j+1/2})_{j\in\mathbb{Z}}$ and divide the real line into cells where the j^{th} cell is the interval $\big(x_{j-1/2}, x_{j+1/2}\big)$. The space step of j^{th} cell is $\Delta x_j = x_{j+1/2} - x_{j-1/2} > 0$ that we suppose constant and equal to Δx for the sake of simplicity. We note $\Delta t > 0$ the time step such that $t^{n+1} - t^n = \Delta t$ with $n \in \mathbb{N}$. For a given initial condition $x \mapsto \boldsymbol{U}^0(x)$, we consider a discrete initial data $\boldsymbol{U}_j^0$ defined by $\boldsymbol{U}_j^0 = \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} \boldsymbol{U}^0(x)\mathrm{d}x$, for $j \in \mathbb{Z}$. The algorithm proposed in this chapter aims at computing a first-order accurate (in both space and time) approximation of the cell-averaged values $\boldsymbol{U}_j^n$ of $\frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} \boldsymbol{U}\left(x, t^n\right)\mathrm{d}x$ where $x \mapsto \boldsymbol{U}\left(x, t^n\right)$ is the exact solution of (1.1) at time t^n by means of a conservative finite volume discretization of (1.1) of the form

$$\boldsymbol{U}_j^{n+1} - \boldsymbol{U}_j^n + \frac{\Delta t}{\Delta x}\big(\boldsymbol{F}_{j+1/2} - \boldsymbol{F}_{j-1/2}\big) = \Delta t\boldsymbol{S}_j. \tag{1.9}$$

1.3 . The original Operator Splitting Lagrange-Projection (OSLP) strategy

Operator splitting strategies allow simpler derivation of numerical methods by solving parts of the system separately and successively. However, this requires storing intermediate state values and may also necessitate specific treatments to implement higher order extension (see, for example [Del Pino et Jourdren 2006; Duboc et al. 2010; Morales de Luna et al. 2020; Del Grosso et Chalons 2021]).

In this section, we recall the properties of the OSLP method presented in [Padioleau et al. 2019]. It combines the all-regime method for gas dynamics proposed by [Chalons et al. 2016a] and the well-balanced treatment of source terms introduced in [Chalons et al. 2016b] in the context of the shallow water system. We chose to re-introduce all the discretization as the goal of the present chapter is to recast this particular OSLP algorithm into a flux-splitting Lagrange-Projection (FSLP) finite volume method, using very similar expressions. We emphasize that the algorithm presented in this section is not new and comes entirely from [Chalons et al. 2016a; Chalons et al. 2016b; Padioleau et al. 2019] and that the novelty of our work lies in a modification of this algorithm that will be detailed in section 1.4. The method is based on the splitting of (1.1) into an acoustic sub-system :

$$\begin{cases} \partial_t \rho + \rho \partial_x u = 0, & \text{(1.10a)} \\ \partial_t (\rho u) + \rho u \partial_x u + \partial_x p = -\rho \partial_x \phi, & \text{(1.10b)} \\ \partial_t (\rho E) + \rho E \partial_x u + \partial_x (p u) = -\rho u \partial_x \phi, & \text{(1.10c)} \end{cases}$$

and a transport sub-system :

$$\begin{cases} \partial_t \rho + u \partial_x \rho = 0, & \text{(1.11a)} \\ \partial_t (\rho u) + u \partial_x (\rho u) = 0, & \text{(1.11b)} \\ \partial_t (\rho E) + u \partial_x (\rho E) = 0. & \text{(1.11c)} \end{cases}$$

Given a fluid state U^n, this operator splitting algorithm can be decomposed as follows.

1. Update the fluid state U^n to the value U^{n+1-} by approximating the solution of (1.10) :

$$\begin{cases} L_j \rho_j^{n+1-} = \rho_j^n, & \text{(1.12a)} \\ L_j (\rho u)_j^{n+1-} = (\rho u)_j^n - \dfrac{\Delta t}{\Delta x} \left(\Pi_{j+1/2}^{*,\theta} - \Pi_{j-1/2}^{*,\theta} \right) - \Delta t \{ \rho \partial_x \phi \}_j^n, & \text{(1.12b)} \\ L_j (\rho E)_j^{n+1-} = (\rho E)_j^n - \dfrac{\Delta t}{\Delta x} \left(\Pi_{j+1/2}^{*,\theta} u_{j+1/2}^* - \Pi_{j-1/2}^{*,\theta} u_{j-1/2}^* \right) - \Delta t \{ \rho u \partial_x \phi \}_j^n, & \text{(1.12c)} \\ L_j = 1 + \dfrac{\Delta t}{\Delta x} \left(u_{j+1/2}^* - u_{j-1/2}^* \right). & \text{(1.12d)} \end{cases}$$

2. Update the fluid state U^{n+1-} to the value U^{n+1} by approximating the solution of (1.11) : for $\varphi \in \{\rho, \rho u, \rho E\}$

$$\varphi_j^{n+1} = \varphi_j^{n+1-} L_j - \frac{\Delta t}{\Delta x} \left(u_{j+1/2}^* \varphi_{j+1/2}^{n+1-} - u_{j-1/2}^* \varphi_{j-1/2}^{n+1-} \right) \tag{1.13}$$

with the upwind choice

$$\varphi_{j+1/2}^{n+1-} = \begin{cases} \varphi_j^{n+1-}, & \text{if } u_{j+1/2}^* \geq 0, \\ \varphi_{j+1}^{n+1-}, & \text{if } u_{j+1/2}^* < 0, \end{cases} \tag{1.14}$$

and the following formulas for the interface pressures and velocities

$$\begin{cases} u^*_{j+1/2} = \dfrac{\left(u^n_{j+1} + u^n_j\right)}{2} - \dfrac{1}{2a_{j+1/2}}\left(p^n_{j+1} - p^n_j + \dfrac{\rho^n_{j+1} + \rho^n_j}{2}(\phi^n_{j+1} - \phi^n_j)\right), & \text{(1.15a)} \\ \Pi^{*,\theta}_{j+1/2} = \dfrac{\left(p^n_{j+1} + p^n_j\right)}{2} - \theta_{j+1/2}\dfrac{a_{j+1/2}}{2}\left(u^n_{j+1} - u^n_j\right), & \text{(1.15b)} \end{cases}$$

as well as the source terms discretization :

$$\begin{cases} \{\rho\partial_x\phi\}^n_j = \dfrac{\{\rho\partial_x\phi\}_{j+1/2} + \{\rho\partial_x\phi\}_{j-1/2}}{2}, & \text{(1.16a)} \\ \{\rho u\partial_x\phi\}^n_j = \dfrac{u^*_{j+1/2}\{\rho\partial_x\phi\}_{j+1/2} + u^*_{j-1/2}\{\rho\partial_x\phi\}_{j-1/2}}{2}, & \text{(1.16b)} \\ \{\rho\partial_x\phi\}_{j+1/2} = \dfrac{\rho^n_{j+1} + \rho^n_j}{2}\dfrac{\phi_{j+1} - \phi_j}{\Delta x}. & \text{(1.16c)} \end{cases}$$

The constant parameter $a_{j+1/2}$ is a local choice of an approximate acoustic impedance a associated with each interface $j+1/2$. It should be chosen large enough so that (1.22) is satisfied, guaranteeing stability for the acoustic step. In practice, we choose

$$a_{j+1/2} = K\max\left(\rho^n_j c^n_j, \rho^n_{j+1}c^n_{j+1}\right) \qquad \text{with } K > 1. \tag{1.17}$$

In the tests of section 1.8 we will use $K = 1.1$.

The parameter θ enables the implementation of a low Mach flux correction that ensures a control of the numerical diffusion in the momentum equation. This simple strategy is modeled after [Dauvergne et al. 2008; Dellacherie 2010; Dellacherie et al. 2016]. Depending on the choice of θ, this correction takes effect whenever $\mathrm{Ma} < 1$. In our case, its sole purpose is to help preserving the accuracy in the low Mach regions of the computational domain by providing a uniform control of the truncation error with respect to Ma. We need to emphasize that this approach does not aim at addressing the full complexity of simulating flows in the low Mach regime that has been widely investigated in the literature and spans for example : from the study of the influence of the grid [Rieper et Bader 2009; Dellacherie 2010], the potential development of spurious modes [Dellacherie 2009; Jung et Perrier 2022], the development of asymptotic preserving methods [P. Degond et M. Tang 2011; Cordier et al. 2012; Zakerzadeh 2016; Bispen et al. 2017; Berthon et al. 2020; Dimarco et al. 2017; Boscarino et al. 2018; Bouchut et al. 2020b], implicit-explicit methods [Chalons et al. 2016a; Dimarco et al. 2018; Boscheri et al. 2020; Bouchut et al. 2020a; Zeifang et al. 2020] multi-dimensional control of the numerical diffusion [Barsukow 2021], use of preconditioning methods [Turkel 1987; Guillard et Viozat 1999; Paillere et al. 2000; Guillard et Murrone 2004; Beccantini et al. 2008] to the study of acoustics in low Mach regime [Bruel et al. 2019].

The discretization of the gravitational source term allows to exactly preserve the following discrete equivalent of the hydrostatic equilibrium (1.8) :

$$\Pi^n_{j+1} - \Pi^n_j = -\frac{\rho^n_{j+1} + \rho^n_j}{2}(\phi_{j+1} - \phi_j), \qquad u^n_j = 0, \qquad \forall j \in \mathbb{Z}, \forall n \in \mathbb{N}. \tag{1.18}$$

Note that the resolution of the acoustic system is performed via a Suliciu-type relaxation [Suliciu 1998; Bouchut 2004; Chalons et Coulombel 2008; Coquel et al. 2012a] following [Chalons et al. 2016a; Chalons et al. 2016b]. Both

steps can be rewritten as a fully conservative update formula :

$$\rho_j^{n+1} = \rho_j^n - \frac{\Delta t}{\Delta x}\left(u^*_{j+1/2}\rho^{n+1-}_{j+1/2} - u^*_{j-1/2}\rho^{n+1-}_{j-1/2}\right), \tag{1.19}$$

$$(\rho u)_j^{n+1} = (\rho u)_j^n - \frac{\Delta t}{\Delta x}\left(u^*_{j+1/2}(\rho u)^{n+1-}_{j+1/2} + \Pi^{\theta,*}_{j+1/2} - u^*_{j-1/2}(\rho u)^{n+1-}_{j-1/2} - \Pi^{\theta,*}_{j-1/2}\right) - \Delta t\{\rho\partial_x\phi\}_j^n, \tag{1.20}$$

$$(\rho E)_j^{n+1} = (\rho E)_j^n - \frac{\Delta t}{\Delta x}\left(u^*_{j+1/2}(\rho E)^{n+1-}_{j+1/2} + \Pi^{\theta,*}_{j+1/2}u^*_{j+1/2} - u^*_{j-1/2}(\rho E)^{n+1-}_{j-1/2} - \Pi^{\theta,*}_{j-1/2}u^*_{j-1/2}\right) - \Delta t\{\rho u\partial_x\phi\}_j^n. \tag{1.21}$$

The scheme (1.19)-(1.20)-(1.21) is proven to be positivity preserving for the density and the internal energy as well as entropy stable when Δt verifies both the acoustic CFL condition :

$$\frac{\Delta t}{\Delta x}\max_{j\in\mathbb{Z}}\left(\max\left(1/\rho_j^n, 1/\rho_{j+1}^n\right)a_{j+1/2}\right) \leq \frac{1}{2}, \tag{1.22}$$

and the transport CFL condition :

$$\Delta t \max_{j\in\mathbb{Z}}\left(\left(u^*_{j-\frac{1}{2}}\right)^+ - \left(u^*_{j+\frac{1}{2}}\right)^-\right) < \Delta x, \tag{1.23}$$

granted that the following inequality :

$$-\frac{1}{2a^2}\left(p^{\mathrm{EOS}}\left(\tau_k^{*,\theta}, s_k\right) - \Pi^*\right)^2 + \frac{(1-\theta)^2\left(u_{j+1}-u_j\right)^2}{8} \leq 0, \quad k = j, j+1, \tag{1.24}$$

where $\tau_j^{*,\theta} = 1/\rho_j^n + \frac{1}{a_{j+1/2}}\left(u^*_{j+1/2} - u_j^n\right)$ and $\tau_{j+1}^{*,\theta} = 1/\rho_{j+1}^n + \frac{1}{a_{j+1/2}}\left(u_{j+1}^n - u^*_{j+1/2}\right)$ is satisfied at each interface $j+1/2$. Just like in the original OSLP paper [Chalons et al. 2016a], the inequality (1.24) is not ensured by any mechanism in the numerical scheme. As a result, for small values of θ, we cannot guarantee that inequality (1.24) remains valid. This is a known issues of the low Mach correction proposed in [Chalons et al. 2016a] that is not adressed in the present study. Let us emphasize that entropy stability can be achieved through alternative criteria (see [Gallice 2003] and [Chan et al. 2021]), however the study of their performance in the low Mach regime is beyond the scope of this chapter. In section 1.4, we discuss how a simple modification of the transport step allows recasting this two-step OSLP algorithm into a one-step FSLP method while keeping the interesting properties of the original method : the well-balanced property, the accuracy in the low Mach regime, mass, and energy positivity and the discrete entropy inequality.

1.4 . Recasting the OSLP method into a Flux-Splitting Lagrange-Projection (FSLP) method ; a modification of the transport step

In this section, we discuss how a simple modification of the transport step (1.13) of the OSLP method (1.19)-(1.20)-(1.21) proposed by [Chalons et al. 2016b] leads to a much simpler FSLP algorithm. Flux-splitting methods have been used in many application contexts thanks to their ease of implementation that relies on building a discrete evaluation of the fluxes (see, for example, [Liu et al. 1998 ; Darracq et al. 1998 ; Evje et Fjelde 2002 ; Paillère

et al. 2003; García-Cascales et Paillère 2006]). These methods have been extensively developed for several decades (see, for example, [Steger et Warming 1981; Zha et Bilgen 1993; Darracq et al. 1998; Liou et Steffen 1993; Jameson 1995, 1995; Liou 1998, 2006, 1996; Bouchut 2003; Toro et Vázquez-Cendón 2012] and the references therein) yielding efficient simulation tools. Unfortunately, deriving theoretical results that ensure the good behavior of these methods is difficult, which contrasts with their good performance in practice. Before going any further, let us mention that the question of building Eulerian numerical fluxes relying on a Lagrangian approximation of the flow equations has been successfully investigated in the literature with different approaches [Dubroca 1999; Gallice 2000, 2003; Bouchut 2003; Chan et al. 2021].

A key contribution of the present chapter is the derivation of stability properties for the flux-splitting algorithm. These proofs are based on the following observation; let us consider a given hyperbolic problem with a source term for which the set of admissible states is convex (*e.g.* Euler's equations of gas dynamics or ideal Magneto-hydrodynamics);

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = S(\boldsymbol{U}). \tag{1.25}$$

We design a separation of the flux and source term into N parts $(F_p, S_p)_{1\leq p\leq N}$ so that :

$$\sum_{p=1}^{N} F_p(\boldsymbol{U}) = F(\boldsymbol{U}), \qquad \sum_{p=1}^{N} S_p(\boldsymbol{U}) = S(\boldsymbol{U}), \tag{1.26}$$

as well as a series of coefficients $\alpha_j^p \in (0,1)$ that sums up to 1; $\sum_{p=1}^{N} \alpha_j^p = 1$ for each cell j. Let us assume that we can build a discretization for each part where the sub-fluxes and sub-source terms are multiplied by the inverses of the coefficients. This allows to consider partially updated value or sub-updated value $U_j^{p,n+1}$ of the initial state U_j^n due to the influence of to the $p-th$ flux and source term, obtaining the $p-th$ sub-update :

$$\frac{\boldsymbol{U}_j^{p,n+1} - \boldsymbol{U}_j^n}{\Delta t} - \frac{1}{\alpha_j^p}[\partial_x F_p(\boldsymbol{U})]_j = \frac{1}{\alpha_j^p}[S_p(\boldsymbol{U})]_j \quad \forall p \in [1, N]. \tag{1.27}$$

Moreover, let us assume that each of these discretizations is stable under their respective local CFL condition :

$$\Delta t < \alpha_j^p \frac{\Delta x}{v_p^j} \tag{1.28}$$

where $v_p^j > 0$ is the local magnitude of the characteristic velocity associated with the discretization of the p-th flux/source term. By re-assembling the result of each part with the convex combination defined by the coefficients α^p,

$$\boldsymbol{U}_j^{n+1} := \sum_{p=0}^{N} \alpha_j^p U_j^{p,n+1} \tag{1.29}$$

we obtain a discretization consistent with (1.25), regardless of the value of the coefficients $\alpha_j^p \in (0,1)$. The full update is stable as a convex combination of the stable sub-updates (1.27). This means we can freely choose the coefficients α_j^p to optimize the CFL condition. Indeed, the update (1.29) is stable as long as each sub-update is stable i.e. :

$$\Delta t < \min\left(\alpha_j^1 \frac{\Delta x}{v_j^1}, \ldots, \alpha_j^N \frac{\Delta x}{v_j^N}\right). \tag{1.30}$$

For $p = 1, \dots, N$, let us now choose $\frac{\alpha_j^p}{v_p} = \frac{1}{v_j^1+v_j^2+\cdots+v_j^N}$, then $\min_p \left(\frac{\alpha_j^p}{v_j^p}\right) = \min_p \left(\frac{1}{v_j^1+v_j^2+\cdots+v_j^N}\right) = \frac{1}{v_j^1+v_j^2+\cdots+v_j^N}$. This provides the following local CFL condition :

$$\Delta t < \frac{\Delta x}{v_j^1 + v_j^2 + \cdots + v_j^N}. \tag{1.31}$$

In this work, we separate the system into $N = 2$ parts corresponding to the pressure and advection terms. This type of splitting is not new and can be found in [Darracq et al. 1998; Liou et Steffen 1993; Deshpande et al. 1994; Toro et Vázquez-Cendón 2012; Borah et al. 2016] without entropy stability theorems. Discretization techniques that also feature a separate treatment for the pressure and advection effects have been proposed for fractional step methods [Baraille et al. 1992; Buffard et Hérard 1997; Chalons et al. 2011; Coquel et al. 2012b; Chalons et al. 2016a; Chalons et al. 2016b; Chalons et al. 2017; Padioleau et al. 2019].

By modifying the transport step of the original operator splitting algorithm (1.19)-(1.20)-(1.21) by computing the fluxes on the initial states n instead of the acoustic state $n+1-$:

$$\varphi_j^{n+1} = \varphi_j^{n+1-} L_j - \frac{\Delta t}{\Delta x}\left(u_{j+1/2}^* \varphi_{j+1/2}^n - u_{j-1/2}^* \varphi_{j-1/2}^n\right) \tag{1.32}$$

we obtain the following fully conservative update that we refer to as our FSLP method :

$$\begin{cases} \rho_j^{n+1} = \rho_j^n - \dfrac{\Delta t}{\Delta x}\left(u_{j+1/2}^* \rho_{j+1/2}^n - u_{j-1/2}^* \rho_{j-1/2}^n\right) \\ (\rho u)_j^{n+1} = (\rho u)_j^n - \dfrac{\Delta t}{\Delta x}\left(u_{j+1/2}^* (\rho u)_{j+1/2}^n + \Pi_{j+1/2}^{\theta,*} - u_{j-1/2}^* (\rho u)_{j-1/2}^n - \Pi_{j-1/2}^{\theta,*}\right) \\ \qquad - \Delta t \{\rho \partial_x \phi\}_j^n, \\ (\rho E)_j^{n+1} = (\rho E)_j^n - \dfrac{\Delta t}{\Delta x}\left(u_{j+1/2}^* (\rho E)_{j+1/2}^n + \Pi_{j+1/2}^{\theta,*} u_{j+1/2}^* \right. \\ \qquad \left. - u_{j-1/2}^* (\rho E)_{j-1/2}^n - \Pi_{j-1/2}^{\theta,*} u_{j-1/2}^*\right) - \Delta t \{\rho u \partial_x \phi\}_j^n. \end{cases} \tag{1.33}$$

Note that we keep the upwind choice for the transport scheme :

$$\varphi_{j+1/2}^n = \begin{cases} \varphi_j^n, \text{ if } u_{j+1/2}^* \geq 0, \\ \varphi_{j+1}^n, \text{ if } u_{j+1/2}^* < 0, \end{cases} \tag{1.34}$$

where $(u, \Pi)^*$ are given by (1.15). We provide the CFL condition associated with the new method :

$$\frac{\Delta t}{\Delta x} \max_{j \in \mathbb{Z}} \left(2 \max\left(1/\rho_j^n, 1/\rho_{j+1}^n\right) a_{j+1/2} + \left(u_{j-\frac{1}{2}}^*\right)^+ - \left(u_{j+\frac{1}{2}}^*\right)^-\right) < 1 \tag{1.35}$$

This CFL condition is obtained by picking as suggested above : $\alpha_j = \frac{\tilde{c}_j}{\tilde{c}_j + \tilde{v}_j}$ with $\tilde{u}_j = \left(u_{j-\frac{1}{2}}^*\right)^+ - \left(u_{j+\frac{1}{2}}^*\right)$ and $2\tilde{c}_j = \max\left(1/\rho_j^n, 1/\rho_{j+1}^n\right) a_{j+1/2}$. It is indeed of the form (1.31) with $N = 2$. It has the same characteristic speeds as the acoustic condition in (1.22) and the transport condition in (1.23), except that they are summed rather than satisfied separately. As a result, (1.35) is generally more restrictive than conditions (1.22), (1.23). The new method has several advantages compared to the original numerical scheme (1.19)-(1.20)-(1.21) :

1. The implementation of the flux-splitting version is much simpler than the operator-splitting version. Indeed, it can be implemented as a standard, simple flux-based finite volume method thanks to the following numerical flux formula :

$$\mathbf{F}^{\text{FSLP}}(U_L, U_R) = \begin{cases} u^* \rho_{LR} \\ u^* (\rho u)_{LR} + \Pi^{*,\theta} \\ u^* (\rho E)_{LR} + \Pi^{*,\theta} u^* \end{cases} \tag{1.36}$$

with

$$\varphi_{LR} = \begin{cases} \varphi_L & \text{if } u^* > 0, \\ \varphi_R & \text{otherwise.} \end{cases} \tag{1.37}$$

We can see in (1.36) that the flux evaluation clearly separates the pressure-related terms from the advection terms so that it can be affiliated with a family of methods proposed in the literature like [Darracq et al. 1998; Liou et Steffen 1993; Deshpande et al. 1994; Toro et Vázquez-Cendón 2012; Borah et al. 2016].

2. As the method can be implemented as a simple flux-based solver, it can be seamlessly combined with any existing flux-based high-order algorithm such as MUSCL[Leer 1977a, 1977b, 1979; Toro 2009], (W)ENO [Liu et al. 1994; Jiang et Shu 1996] or MOOD methods [Diot et al. 2013; Clain et al. 2011]. We detail the procedure for the extension to second order in section 1.6.2 and give some numerical examples in section 1.8. Note, however, that the well-balanced treatment of gravity is not straightforward to extend to high order and requires a careful examination that is beyond the scope of this chapter. Also, using the low Mach correction θ combined with a highly accurate high-order method can amplify numerical instabilities that already exist at first-order (checkerboard modes, for example). We do not address this issue in this chapter, as our focus is on demonstrating the recasting of the OSLP method into the FSLP method.
3. The FSLP method is more computationally efficient than the original OSLP method. The OSLP method requires two update loops per time step to compute a time step of size $\sim \Delta x / \max(v, c)$, where v and c are the velocities associated with transport and acoustic effects, respectively, as they appear in the CFL conditions. In contrast, the FSLP method only requires one loop per time step of size $\sim \Delta x/(v + c)$. This means that the FSLP method requires fewer sweeps to reach the same physical time, especially in the low Mach regime where $v \ll c$ or in the hypersonic regime where $v \gg c$, where it is expected to be more efficient. If $v = c$, both methods should have a comparable efficiency. We provide a performance analysis and discussion in section 1.8.9.
4. The new update formula eliminates the need to store the intermediate state U^{n+1-}, as it can be computed in a single sweep. This reduces the algorithm's memory footprint by approximately $2/3$, and reduces the stencil radius from two to one cell. The decrease in memory storage requirements can improve performance by reducing the time spent accessing the data arrays.

Despite the update formula being very similar, the mathematical background required to derive the stability properties of (1.33) is new. It is the object of the next section 1.5.

1.5 . Derivation of the stability properties for our new method

In this section, we focus on deriving the stability properties of our new FSLP scheme (1.33). To this end, we will perform a Suliciu-type relaxation [Suliciu 1998; Bouchut 2004; Chalons et Coulombel 2008; Coquel et al. 2012a] of the pressure term and introduce a surrogate specific volume. We then isolate two new sub-systems, the advection

and pressure sub-systems, for which we derive numerical fluxes. We then re-obtain our new method and derive its stability properties by performing a convex combination of the two fluxes. Note that the proof of stability for the pressure subsystem is similar to the acoustic sub-system in [Chalons et al. 2016a]. For this reason, we only recall this proof in the appendix for completeness.

1.5.1 . Relaxation and flux-splitting

We first apply a relaxation of the original Euler system. Manipulations of smooth solutions of (1.1) gives $\partial_t(\rho p) + \partial_x(u\rho p) + \rho^2 c^2 \partial_x u = 0$. We choose to perform a Suliciu-type approximation of the system (1.1) for $t \in [t^n, t^{n+1})$ by introducing a surrogate pressure Π and considering the relaxed system :

$$\begin{cases} \partial_t \rho + \partial_x(\rho u) = 0, \\ \partial_t(\rho u) + \partial_x\left(u\rho u + \Pi\right) = -\rho \partial_x \phi, \\ \partial_t(\rho E) + \partial_x(u\rho E + \Pi u) = -\rho u \partial_x \phi, \\ \partial_t(\rho \Pi) + \partial_x(u\rho\Pi + a^2 u) = \rho\lambda(p - \Pi). \end{cases} \tag{1.38}$$

The parameter λ is a frequency that characterizes the strength of the source term that drives Π towards the equilibrium $\Pi = p$. In the regime $\lambda \to \infty$, we formally recover (1.1). In our numerical solver context, we classically mimic the $\lambda \to \infty$ regime by enforcing $\Pi_j^n = p^{\text{EOS}}(1/\rho_j^n, e_j^n)$ at each time step and then solving (1.38) with $\lambda = 0$, which will be the case in all computations below without any ambiguities. We now introduce another auxiliary variable $\mathcal{T}$ and impose that it verifies

$$\partial_t(\rho \mathcal{T}) = 0. \tag{1.39}$$

We suppose that $\mathcal{T}(t = 0) = 1/\rho(t = 0)$ at the initial instant so that $\mathcal{T}(x, t)$ is equal to the specific volume $1/\rho(x, t)$ for all x and $t > 0$. Let us now re-write the system (1.38)-(1.39) in order to highlight three different operators that compose the flux and the source term of (1.38)-(1.39) following similar lines as [Darracq et al. 1998; Liou et Steffen 1993; Deshpande et al. 1994; Borah et al. 2016]

$$\partial_t \begin{pmatrix} \rho \\ \rho u \\ \rho E \\ \rho \Pi \\ \rho \mathcal{T} \end{pmatrix} + \partial_x \begin{pmatrix} \rho u \\ \rho u^2 \\ \rho E u \\ \rho \Pi u \\ u \end{pmatrix} + \partial_x \begin{pmatrix} 0 \\ \Pi \\ \Pi u \\ a^2 u \\ -u \end{pmatrix} = - \begin{pmatrix} 0 \\ \rho \\ \rho u \\ 0 \\ 0 \end{pmatrix} \partial_x \phi. \tag{1.40}$$

Let us underline that both Π and $\rho\mathcal{T}$ are only mathematical intermediates used to derive the scheme's stability properties. Indeed, these variables do not appear in the update formula (1.33), so that there is no need to evaluate and store them while implementing the algorithm. Let us introduce the convex combination parameter $\alpha \in (0, 1)$ and two subsystems associated with different parts of the fluxes and source terms featured in (1.40). The first

system gathers the source term and the flux associated with pressure terms weighted by $1/\alpha$

$$\begin{cases} \partial_t \rho = 0, \\ \partial_t(\rho u) + \dfrac{1}{\alpha}\partial_x(\Pi) = -\dfrac{1}{\alpha}\rho\partial_x\phi, \\ \partial_t(\rho E) + \dfrac{1}{\alpha}\partial_x(\Pi u) = -\dfrac{1}{\alpha}\rho u\partial_x\phi, \\ \partial_t(\rho\Pi) + \dfrac{1}{\alpha}\partial_x(a^2 u) = 0, \\ \partial_t(\rho\mathcal{T}) - \dfrac{1}{\alpha}\partial_x u = 0. \end{cases} \tag{1.41}$$

We will refer to (1.41) as the pressure system. The second sub-system is composed of the remaining terms that pertain to transport effects weighted by $1/1-\alpha$, it reads

$$\begin{cases} \partial_t(\rho\varphi) + \dfrac{1}{1-\alpha}\partial_x(u\rho\varphi) = 0, \qquad \varphi \in \{1, u, E, \Pi\} \\ \partial_t(\rho\mathcal{T}) + \dfrac{1}{1-\alpha}\partial_x u = 0, \end{cases} \tag{1.42}$$

and will be called the advection system.

The pressure system (1.41) is hyperbolic and involves the characteristic velocities $\{\pm\frac{1}{\alpha}a/\rho, 0, 0, 0\}$ that are all associated with linearly degenerate fields. The advection system (1.42) is only weakly hyperbolic as its Jacobian matrix admits $(1-\alpha)u$ as multiple eigenvalues but is not diagonalizable. Nevertheless, let us underline that the algorithms we will consider for approximating the solutions of (1.42) will verify a local maximum principle under a CFL condition so that stability will be ensured for the advection step (see section 1.5.2).

Before continuing, let us comment on equations (1.41) and (1.42). The factors α and $1-\alpha$ that appear in the fluxes and source terms of these equations correspond to the case $N = 2$ of the flux splitting stability argument presented at the beginning of section 1.4.

Then, although the trivial stationary equation (1.39) is now split into two unstationary parts within (1.41) and (1.42), the overall scheme will indeed guarantee that $(\rho\mathcal{T})_j^n = 1$ for $j \in \mathbb{Z}$ and $n \in \mathbb{N}$.

1.5.2 . The convex combination

We propose the following discretization strategy :

1. Compute $\boldsymbol{U}_j^P$ as the update of the initial state $\boldsymbol{U}_j^n$ by approximating the solution of (1.41) :

$$\begin{cases} \rho_j^P = \rho_j^n, \\ (\rho u)_j^P = (\rho u)_j^n - \dfrac{1}{\alpha}\dfrac{\Delta t}{\Delta x}\left(\Pi_{j+1/2}^{*,\theta} - \Pi_{j-1/2}^{*,\theta}\right) - \dfrac{1}{\alpha}\Delta t\left\{\rho\partial_x\phi\right\}_j^n, \\ (\rho E)_j^P = (\rho E)_j^n - \dfrac{1}{\alpha}\dfrac{\Delta t}{\Delta x}\left(\Pi_{j+1/2}^{*,\theta}u_{j+1/2}^* - \Pi_{j-1/2}^{*,\theta}u_{j-1/2}^*\right) - \dfrac{1}{\alpha}\Delta t\left\{\rho u\partial_x\phi\right\}_j^n, \\ (\rho\Pi)_j^P = (\rho\Pi)_j^n - \dfrac{1}{\alpha}\dfrac{\Delta t}{\Delta x}\left(a_{j+1/2}^2 u_{j+1/2}^* - a_{j-1/2}^2 u_{j-1/2}^*\right), \\ (\rho\mathcal{T})_j^P = 1 + \dfrac{1}{\alpha}\dfrac{\Delta t}{\Delta x}\left(u_{j+1/2}^* - u_{j-1/2}^*\right). \end{cases} \tag{1.43}$$

2. Compute $\boldsymbol{U}_j^A$ as the update of the initial state $\boldsymbol{U}_j^n$ by approximating the solution of (1.42) : for $\varphi \in \{1, u, E, \Pi\}$

$$\begin{cases} (\rho\varphi)_j^A = (\rho\varphi)_j^n - \dfrac{1}{1-\alpha}\dfrac{\Delta t}{\Delta x}\left(u_{j+1/2}^*(\rho\varphi)_{j+1/2}^n - u_{j-1/2}^*(\rho\varphi)_{j-1/2}^n\right), \\ (\rho\mathcal{T})_j^A = (\rho\mathcal{T})_j^n - \dfrac{1}{1-\alpha}\dfrac{\Delta t}{\Delta x}\left(u_{j+1/2}^* - u_{j-1/2}^*\right). \end{cases} \tag{1.44}$$

3. Evaluate $\boldsymbol{U}_j^{n+1}$ as the convex combination of $\boldsymbol{U}_j^P$ and $\boldsymbol{U}_j^A$:

$$\boldsymbol{U}_j^{n+1} = \alpha\boldsymbol{U}_j^P + (1-\alpha)\boldsymbol{U}_j^A \tag{1.45}$$

It can be verified that the update in (1.45) is equivalent to the FSLP scheme in (1.33), for any value of $\alpha \in (0, 1)$. This means that the flux of the FSLP scheme can be expressed as an arbitrary convex combination of the fluxes involved in the update. As explained at the beginning of section 1.4, this interpretation allows us to choose α optimally in order to obtain the least restrictive CFL condition, given by (1.35).

Remark 1.5.1. *The original operator splitting method proposed by [Chalons et al. 2016a] coincides with a Lagrange-Projection scheme when used in a 1D context. As a result, the Lagrange-Projection designation is used to design finite volume, acoustic/transport operator splitting methods for various hyperbolic systems in the literature. However, for 2D problems, the OSLP method does not correspond to a Lagrange-Projection method despite sharing similarities with the 1D version. Lagrange-Projection methods are operator-splitting methods consisting of a Lagrange step and a projection step. Our method does not split operators but fluxes, so we doubt it can still be interpreted as a Lagrange-Projection method. However, we choose to keep the designation as FSLP inherits its formula from the line of work stemming from the Lagrange-Projection literature.*

1.5.3 . Stability of the pressure step

In this section, we prove the stability of the pressure step. We chose to move all the derivations in the appendix as the arguments we use are already present in [Chalons et al. 2016b] in the proof of the stability of the acoustic step (1.12) of the OSLP method (1.19)-(1.20)-(1.21). We introduce the CFL condition associated with the pressure step.

$$\frac{1}{\alpha}\frac{\Delta t}{\Delta x}\max_{j\in\mathbb{Z}}\left(\max\left(1/\rho_j^n, 1/\rho_{j+1}^n\right)a_{j+1/2}\right) \leq \frac{1}{2}, \tag{1.46}$$

It is identical to the acoustic CFL (1.22) but $1/\alpha$ times as restrictive.

Proposition 1.5.1. *Suppose that a is chosen large enough so that* (1.17) *is verified and that both $\mathcal{T}_L^* > 0$, $\mathcal{T}_R^* > 0$ from* (1.113e) *are positive. Suppose also that the low Mach correction θ is chosen large enough so that* (1.133) *is valid. Under the CFL condition* (1.46) *we have that :*

1. *the density and the internal energy verify $\rho_j^P > 0$ and $e_j^P > 0$, for all j,*
2. *the discretization* (1.43) *satisfies the entropy inequality*

$$\rho_j^P s^{EOS}(\mathcal{T}_j^P, e_j^P) - \rho_j^n s(1/\rho_j^n, e_j^n) + \frac{1}{\alpha}\frac{\Delta t}{\Delta x}(q_{j+1/2}^n - q_{j-1/2}^n) \geq 0, \tag{1.47}$$

with $q_{j+1/2}^n = q_\Delta(\boldsymbol{U}_j^n, \boldsymbol{U}_{j+1}^n)$, where q_Δ is a flux function consistent with 0 as $\Delta t, \Delta x \to 0$.

Démonstration. The positivity of the internal energy and the entropy inequality of (1.47) are direct consequences of the approximate Riemann solver properties of proposition 1.C.2 and the consistency in the integral sense [Bouchut 2004]. □

The condition (1.133) is identical to the OSLP low Mach stability condition (1.24) but with the surrogate density $\mathcal{T}$ instead of $1/\rho$. In practice, conditions for stability for the pressure update are strictly the same as for the acoustic step (1.12) from [Chalons et al. 2016a] apart from the factor $1/\alpha$ in the CFL condition.

1.5.4 . Stability of the advection step

We introduce the CFL condition associated with the advection step.

$$\frac{1}{1-\alpha}\frac{\Delta t}{\Delta x}\max_{j\in\mathbb{Z}}\left(\left(u^*_{j-\frac{1}{2}}\right)^+-\left(u^*_{j+\frac{1}{2}}\right)^-\right)<1. \tag{1.48}$$

It is identical to the transport CFL (1.23) but $1/(1-\alpha)$ times as restrictive.

Proposition 1.5.2. *Under the CFL condition* (1.48)*, the discretization* (1.44) *of the advection subsystem verifies the following properties.*

1. $\boldsymbol{U}_j^A$ *is a positive linear combination of* $\boldsymbol{U}_{j-1}^n$, $\boldsymbol{U}_j^n$ *and* $\boldsymbol{U}_{j+1}^n$,
2. b_j^A *is a convex combination of* b_{j-1}^n, b_j^n *and* b_{j+1}^n *for* $b\in\{u,E,\mathcal{T}\}$,
3. *if* $e_j^n>0$ *for all* $j\in\mathbb{Z}$ *then* $e_j^A>0$ *for all* $j\in\mathbb{Z}$,
4. *the discretization* (1.44) *satisfies the entropy inequality*

$$\rho_j^A s^{EOS}\left(\mathcal{T}_j^A,e_j^A\right)-\rho_j^n s_j^n+\frac{1}{1-\alpha}\frac{\Delta t}{\Delta x}\left(u^*_{j+1/2}\rho^n_{j+1/2}s^n_{j+1/2}-u^*_{j-1/2}\rho^n_{j-1/2}s^n_{j-1/2}\right)\geq 0. \tag{1.49}$$

Démonstration. The advection scheme (1.44) can be recast into

$$\boldsymbol{U}_j^A=-\frac{1}{1-\alpha}\frac{\Delta t}{\Delta x}u^{*,-}_{j+1/2}\,\boldsymbol{U}^n_{j+1}+\frac{1}{1-\alpha}\frac{\Delta t}{\Delta x}u^{*,+}_{j-1/2}\,\boldsymbol{U}^n_{j-1}+\left[1-\frac{1}{1-\alpha}\frac{\Delta t}{\Delta x}(u^{*,+}_{j+1/2}-u^{*,-}_{j-1/2})\right]\boldsymbol{U}^n_j, \tag{1.50}$$

which proves 1. One can also write

$$\left(\frac{\boldsymbol{U}}{\rho}\right)_j^A=\lambda_j^{(+1)}\left(\frac{\boldsymbol{U}}{\rho}\right)_{j+1}^n+\lambda_j^{(0)}\left(\frac{\boldsymbol{U}}{\rho}\right)_j^n+\lambda_j^{(-1)}\left(\frac{\boldsymbol{U}}{\rho}\right)_{j-1}^n, \tag{1.51}$$

with

$$\begin{aligned}
&\lambda_j^{(+1)}=-\frac{1}{1-\alpha}\frac{\Delta t}{\Delta x}u^{*,-}_{j+1/2}\left(\frac{\rho^n_{j+1}}{\rho^A_j}\right),\lambda_j^{(0)}=\left[1-\frac{1}{1-\alpha}\frac{\Delta t}{\Delta x}(u^{*,+}_{j+1/2}-u^{*,-}_{j-1/2})\right]\left(\frac{\rho^n_j}{\rho^A_j}\right),\\
&\lambda_j^{(-1)}=\frac{1}{1-\alpha}\frac{\Delta t}{\Delta x}u^{*,+}_{j-1/2}\left(\frac{\rho^n_{j-1}}{\rho^A_j}\right).
\end{aligned} \tag{1.52}$$

By (1.50) we have that

$$\rho_j^A=-\frac{1}{1-\alpha}\frac{\Delta t}{\Delta x}u^{*,-}_{j+1/2}\,\rho^n_{j+1}+\frac{1}{1-\alpha}\frac{\Delta t}{\Delta x}u^{*,+}_{j-1/2}\,\rho^n_{j-1}+\left[1-\frac{1}{1-\alpha}\frac{\Delta t}{\Delta x}(u^{*,+}_{j+1/2}-u^{*,-}_{j-1/2})\right]\rho^n_j, \tag{1.53}$$

so that $\lambda_j^{(+1)} + \lambda_j^{(0)} + \lambda_j^{(-1)} = 1$, which proves that b_j^A is a convex combination of b_{j-1}^n, b_j^n and b_{j+1}^n for $b \in \{u, E\}$. Let us now consider the case of $\mathcal{T}^A$. By (1.44), we have that

$$\begin{aligned}\mathcal{T}_j^A = \mathcal{T}_j^n \frac{\rho_j^n}{\rho_j^A} - \frac{1}{1-\alpha}\frac{\Delta t}{\Delta x} u_{j+1/2}^{*,+} \frac{1}{\rho_j^A} - \frac{1}{1-\alpha}\frac{\Delta t}{\Delta x} u_{j+1/2}^{*,-} \frac{1}{\rho_j^A} + \frac{1}{1-\alpha}\frac{\Delta t}{\Delta x} u_{j-1/2}^{*,+} \frac{1}{\rho_j^A} \\ + \frac{1}{1-\alpha}\frac{\Delta t}{\Delta x} u_{j-1/2}^{*,-} \frac{1}{\rho_j^A}.\end{aligned} \tag{1.54}$$

However, since we chose $\rho_j^n \mathcal{T}_j^n = 1$ for all $i \in \mathbb{Z}$, We can write that

$$\begin{aligned}\mathcal{T}_j^A &= \mathcal{T}_j^n \frac{\rho_j^n}{\rho_j^A} - \frac{1}{1-\alpha}\frac{\Delta t}{\Delta x} u_{j+1/2}^{*,+} \frac{\rho_j^n}{\rho_j^A}\mathcal{T}_j^n - \frac{1}{1-\alpha}\frac{\Delta t}{\Delta x} u_{j+1/2}^{*,-} \frac{\rho_{j+1}^n}{\rho_j^A}\mathcal{T}_{j+1}^n \\ &\quad + \frac{1}{1-\alpha}\frac{\Delta t}{\Delta x} u_{j-1/2}^{*,+} \frac{\rho_{j-1}^n}{\rho_j^A}\mathcal{T}_{j-1}^n + \frac{1}{1-\alpha}\frac{\Delta t}{\Delta x} u_{j-1/2}^{*,-} \frac{\rho_j^n}{\rho_j^A}\mathcal{T}_j^n \\ &= \lambda_j^{(+1)}\mathcal{T}_{j+1}^n + \lambda_j^{(-1)}\mathcal{T}_{j-1}^n + \lambda_j^{(0)}\mathcal{T}_j^n.\end{aligned} \tag{1.55}$$

Consequently $\mathcal{T}_j^A$ is also a convex combination of $\mathcal{T}_{j-1}^n$, $\mathcal{T}_j^n$ and $\mathcal{T}_{j+1}^n$, which proves 2. For statement 3, we consider the concave function K introduced in the proof of lemma 1.A.1, and we have that $e_j^A = K(u_j^A, E_j^A)$. Thanks to statement 2, we can thus write that.

$$e_j^A = K\left(\sum_{k=0,\pm1} \lambda_j^{(k)} u_{j+k}^n, \sum_{k=0,\pm1} \lambda_j^{(k)} E_{j+k}^n\right) \geq \sum_{k=0,\pm1} \lambda_j^{(k)} K\big(u_{j+k}^n, E_{j+k}^n\big) = \sum_{k=0,\pm1} \lambda_j^{(k)} e_{j+k}^n > 0, \tag{1.56}$$

which proves statement 3.

Now using the lemma 1.A.1, we have that

$$s(\mathcal{T}_j^A, e_j^A) = \mathcal{U}(\mathcal{T}_j^A, u_j^A, E_j^A) = \mathcal{U}\left(\sum_{k=0,\pm1} \lambda_j^{(k)} \mathcal{T}_{j+k}^n, \sum_{k=0,\pm1} \lambda_j^{(k)} u_{j+k}^n, \sum_{k=0,\pm1} \lambda_j^{(k)} E_{j+k}^n\right) \tag{1.57}$$

$$\geq \sum_{k=0,\pm1} \lambda_j^{(k)} \mathcal{U}\big(\mathcal{T}_{j+k}^n, u_{j+k}^n, E_{j+k}^n\big) = \sum_{k=0,\pm1} \lambda_j^{(k)} s(\mathcal{T}_{j+k}^n, e_{j+k}^n). \tag{1.58}$$

This inequality also reads

$$\begin{aligned}s(\mathcal{T}_j^A, e_j^A) \geq s_j^n \frac{\rho_j^n}{\rho_j^A} - \frac{1}{1-\alpha}\frac{\Delta t}{\Delta x} u_{j+1/2}^{*,+} \frac{\rho_j^n}{\rho_j^A} s_j^n - \frac{1}{1-\alpha}\frac{\Delta t}{\Delta x} u_{j+1/2}^{*,-} \frac{\rho_{j+1}^n}{\rho_j^A} s_{j+1}^n \\ + \frac{1}{1-\alpha}\frac{\Delta t}{\Delta x} u_{j-1/2}^{*,+} \frac{\rho_{j-1}^n}{\rho_j^A} s_{j-1}^n + \frac{1}{1-\alpha}\frac{\Delta t}{\Delta x} u_{j-1/2}^{*,-} \frac{\rho_j^n}{\rho_j^A} s_j^n.\end{aligned} \tag{1.59}$$

If the CFL condition (1.48) is met then $\rho_j^A \geq 0$ and by multiplying (1.59) by ρ_j^A we get (1.49). □

1.5.5 . Stability of the FSLP method

Proposition 1.5.3. *If the following conditions are met*

1. *the CFL condition* (1.35) *is verified,*

2. *the parameter a is large enough so that* (1.17) *is verified and $\mathcal{T}_L^* > 0$, $\mathcal{T}_R^* > 0$ in* (1.113e) *for all $j \in \mathbb{Z}$,*

3. *both density and internal energies are positive, i.e. $\rho_j^n > 0$ and $e_j^n > 0$ for all $j \in \mathbb{Z}$,*

4. *the parameter θ is large enough so that* (1.133) *is valid at each interface,*

then the flux-splitting update (1.33)

(a) preserves positivity for both density and internal energy i.e. $\rho_j^{n+1} > 0$ and $e_j^{n+1} > 0$ for all $j \in \mathbb{Z}$,

(b) is endowed with the following entropy inequality :

$$\rho_j^{n+1} s(1/\rho_j^{n+1}, e_j^{n+1}) - \rho_j^n s(1/\rho_j^n, e_j^n) + \frac{\Delta t}{\Delta x}\left(Q_{j+1/2} - Q_{j-1/2}\right) \geq 0, \tag{1.60}$$

with $Q_{j+1/2} = u_{j+1/2}^* \rho_{j+1/2}^n s_{j+1/2}^n + q_{j+1/2}^n$.

Démonstration. (a) Let us start by ensuring that the CFL conditions (1.46), (1.48) are satisfied so that the advection and pressure steps are stable. By choosing α so that $\alpha c_j = (1-\alpha)v_j = v_j + c_j$ where $v_j = \left(\left(u_{j-\frac{1}{2}}^*\right)^+ - \left(u_{j+\frac{1}{2}}^*\right)^-\right)$ and

$$c_j = 2\max\left[\max\left(1/\rho_{j-1}^n, 1/\rho_j^n\right) a_{j-1/2}, \max\left(1/\rho_j^n, 1/\rho_{j+1}^n\right) a_{j+1/2}\right],$$

it is straightforward that (1.46), (1.48) are equivalent and correspond to (1.35). This choice of α seems local as it depends on the characteristic speed of each cell considered. However, it can be chosen globally as the minimizer of $f(\alpha) = \max_j(\alpha c_j, (1-\alpha)v_j) = c_i + v_i$ where i is the index of the cell with the largest speed sum of the simulation domain.

Thanks to the propositions 1.5.2 and 1.5.1 we have $\rho^A > 0$, $\rho^P > 0$, $e^A > 0$ and $e^P > 0$ thus, the positivity is straightforward for the density as $\rho_j^{n+1} = (\rho_j^A + \rho_j^P)/2 > 0$. For the internal energy, we consider the function $\Lambda((\rho, \rho u, \rho E)) = (\rho E) - \frac{(\rho u)^2}{2\rho}$, that is proven to be concave in 1.A.1. We have that :

$$\begin{aligned}
(\rho e)_j^{n+1} &= (\rho E)_j^{n+1} - \frac{((\rho u)_j^{n+1})^2}{2\rho_j^{n+1}} = \Lambda(\rho_j^{n+1}, (\rho u)_j^{n+1}, (\rho E)_j^{n+1}) \\
&= \Lambda\big((1-\alpha)\rho_j^A + \alpha\rho_j^P, (1-\alpha)(\rho u)_j^A + \alpha(\rho u)_j^P, (1-\alpha)(\rho E)_j^A + \alpha(\rho E)_j^P\big) \\
&\geq (1-\alpha)\Lambda\big(\rho_j^A, (\rho u)_j^A, (\rho E)_j^A\big) + \alpha\Lambda\big(\rho_j^P, (\rho u)_j^P, (\rho E)_j^P\big) = (1-\alpha)(\rho e)_j^A + \alpha(\rho e)_j^P > 0
\end{aligned} \tag{1.61}$$

by concavity.

For (b) : propositions 1.5.2 and 1.5.1 ensure that both entropy inequalities (1.47) and (1.49) are satisfied. We then use the concavity of the function $\eta(\rho, \rho\mathcal{T}, \rho u, \rho E) = \rho s\left(\frac{\rho\mathcal{T}}{\rho}, \frac{(\rho E)}{\rho} - \frac{(\rho u)^2}{2\rho^2}\right)$ that is proven in 1.A.1 and the fact that $(\rho\mathcal{T})_j^{n+1} = \alpha(\rho\mathcal{T})_j^P + (1-\alpha)(\rho\mathcal{T})_j^A = 1$. Noting $\alpha^P = \alpha$ and $\alpha^A = 1-\alpha$, we have :

$$\rho_j^{n+1} s(1/\rho_j^{n+1}, e_j^{n+1}) = \rho_j^{n+1} s\left(\frac{1}{\rho_j^{n+1}}, \frac{(\rho E)_j^{n+1}}{\rho_j^{n+1}} - \frac{1}{2}\left(\frac{(\rho u)_j^{n+1}}{\rho_j^{n+1}}\right)^2\right) = \eta(\rho_j^{n+1}, 1, (\rho u)_j^{n+1}, (\rho E)_j^{n+1})$$

$$= \eta(\rho_j^{n+1}, (\rho\mathcal{T})_j^{n+1}, (\rho u)_j^{n+1}, (\rho E)_j^{n+1}) = \eta\left(\sum_{k=A,P}\alpha^k\rho_j^k, \sum_{k=A,P}\alpha^k(\rho\mathcal{T})_j^k, \sum_{k=A,P}\alpha^k(\rho u)_j^k, \sum_{k=A,P}\alpha^k(\rho E)_j^k\right). \tag{1.62}$$

Thanks to appendix 1.5.2 we know that η is concave and thus we have :

$$\rho_j^{n+1} s(1/\rho_j^{n+1}, e_j^{n+1}) \geq \sum_{k=A,P} \alpha^k \eta\Big(\rho_j^k, (\rho\mathcal{T})_j^k, (\rho u)_j^k, (\rho E)_j^k\Big) = \sum_{k=A,P} \alpha^k \rho_j^k s\Big(\mathcal{T}_j^k, e_j^k\Big). \tag{1.63}$$

by concavity. Using (1.47) and (1.49), we get :

$$\begin{aligned}\rho_j^{n+1} s\Big(1/\rho_j^{n+1}, e_j^{n+1}\Big) \geq \rho_j^n s\big(1/\rho_j^n, e_j^n\big) - \frac{\Delta t}{\Delta x}(q_{j+1/2}^n - q_{j-1/2}^n) \\ - \frac{\Delta t}{\Delta x}(u_{j+1/2}^* \rho_{j+1/2}^n s_{j+1/2}^n - u_{j-1/2}^* \rho_{j-1/2}^n s_{j-1/2}^n),\end{aligned} \tag{1.64}$$

which proves (b). □

1.6 . Low Mach behavior, extension to multiple dimensions and higher order of accuracy

In this section, we briefly address the behavior of the scheme in the low Mach regime and propose simple means to extend the FSLP method to multi-dimensional problems and improve its accuracy with higher-order techniques.

1.6.1 . Low Mach behavior

Many simulation cases involve flows in which the material velocity is relatively low compared to the sound velocity. A common way to characterize this situation is to consider the numbers L, t_0, ρ_0, u_0, p_0, $u_0 = p_0\rho_0$, $c_0 = \sqrt{p_0/\rho_0}$ and $(\partial_x\phi)_0$ that are the characteristic magnitudes for length, time, density, velocity, pressure, sound velocity, and $\partial_x\phi$, respectively. We then introduce the following non-dimensional variables : $\tilde{x} = x/L$, $\tilde{t} = t/t_0$, $\tilde{\rho} = \rho/\rho_0$, $\tilde{u} = u/u_0$, $\tilde{e} = e/e_0$, $\tilde{p} = p/p_0$, $\widetilde{(\partial_x\phi)} = \partial_x\phi/(\partial_x\phi)_0$, and we define the Mach number Ma and the Froude number Fr by $\mathrm{Ma} = u_0/c_0$ and $\mathrm{Fr} = u_0/\sqrt{L(\partial_x\phi)_0}$. Following [Bispen et al. 2017; Thomann et al. 2020], we consider a particular flow regime such that $\mathrm{Ma} = \mathrm{Fr}$ so that the system (1.1) takes the following non-dimensional form

$$\begin{aligned}&\partial_{\tilde{t}}\tilde{\rho} + \partial_{\tilde{x}}(\tilde{\rho}\tilde{u}) = 0,\\ &\partial_{\tilde{t}}(\tilde{\rho}\tilde{u}) + \partial_{\tilde{x}}(\tilde{\rho}\tilde{u}^2) + \frac{1}{\mathrm{Ma}^2}\Big(\partial_{\tilde{x}}\tilde{p} + \tilde{\rho}\widetilde{(\partial_x\phi)}\Big) = 0,\\ &\partial_{\tilde{t}}(\tilde{\rho}\tilde{E}) + \partial_{\tilde{x}}(\tilde{\rho}\tilde{E}\tilde{u} + \tilde{p}\tilde{u}) = -\tilde{\rho}\tilde{u}\widetilde{(\partial_x\phi)}.\end{aligned} \tag{1.65}$$

Thanks to system (1.65), one can see that in the limit $\mathrm{Ma} \to 0$, a singularity may appear in the momentum equation. Supposing now that $\mathrm{Ma} \ll 1$, this suggests to distinguish two cases similarly as in [Chalons et al. 2016a] : in the first case the term $\partial_{\tilde{x}}\tilde{p} + \tilde{\rho}\widetilde{(\partial_x\phi)}$ will always remain of magnitude $O(\mathrm{Ma}^2)$, so that $\tilde{\rho}$, $\tilde{u}$ and $\tilde{E}$ will also remain of order $O(\mathrm{Ma}^0)$. In this case, we will say that the system is in the low Mach regime. In the second case, the term $\partial_{\tilde{x}}\tilde{p} + \tilde{\rho}\widetilde{(\partial_x\phi)}$ will not remain of magnitude $O(\mathrm{Ma}^2)$ in such way that $\tilde{\rho}\tilde{u}$, may experience large variations from $O(\mathrm{Ma}^2)$ to $O(\mathrm{Ma}^0)$, yielding significant growth of Ma and thus a change in the Mach regime. These variations characterize all-regime flows with respect to the Mach number. Let us remark that the finer definition of well-prepared initial conditions used in [Bispen et al. 2017] verifies the looser notion of low Mach regime considered in this work.

As it was mentioned earlier, the behavior of the Euler equations in the low Mach regime and adapted simulation strategies raise issues that have been intensively investigated for many years and are still very actively studied

(see [Turkel 1987; Guillard et Viozat 1999; Paillere et al. 2000; Guillard et Murrone 2004; Beccantini et al. 2008; Rieper et Bader 2009; Dauvergne et al. 2008; Dellacherie 2010; P. Degond et M. Tang 2011; Cordier et al. 2012; Dellacherie 2010; Chalons et al. 2016a; Dellacherie et al. 2016; Zakerzadeh 2016; Barsukow 2021; Zakerzadeh 2016; Bispen et al. 2017; Berthon et al. 2020; Dimarco et al. 2017; Boscarino et al. 2018; Bouchut et al. 2020b; Dimarco et al. 2018; Bruel et al. 2019; Boscheri et al. 2020; Bouchut et al. 2020a; Zeifang et al. 2020] and the references therein). In this work, we propose transposing the low Mach error analysis of the OSLP method presented in [Chalons et al. 2016a] to the FSLP scheme. This task is straightforward, although it requires lengthy and tedious calculations. Therefore, for the sake of brevity, we only recall the main points of this approach. We consider a non-dimensional expression of the FSLP solver for a one-dimensional problem and evaluate the truncation error obtained with a smooth solution of (1.65) that satisfies the low Mach regime hypothesis $\partial_{\tilde{x}}\tilde{p} + \widetilde{\tilde{\rho}(\partial_x \phi)} = O(\mathrm{Ma}^2)$. Similarly to the OSLP scheme, the magnitudes of the resulting truncation error estimates are uniform with respect to Ma except for the momentum equation that features an error term of order $O(\theta \Delta x/\mathrm{Ma})$. Consequently , choosing $\theta = O(\mathrm{Ma})$ when $\mathrm{Ma} \ll 1$ will help the scheme preserve a uniform truncation error with respect to Ma. A well-known consequence of this choice is that in regions where $\mathrm{Ma} \ll 1$, the non-centered part of the pressure term $\Pi^{*,\theta}_{j+1/2}$ will be moderated.

The numerical tests proposed in sections 1.8.4, 1.8.7 and 1.8.8 show that this simple correction work similarly for both FSLP and OSLP methods : in the low Mach regime, both schemes provide accurate results. Nevertheless, we need to emphasize that the modification of the scheme induced by θ is not flawless and should be considered with care. Spurious oscillations may occur [Dellacherie 2009; Jung et Perrier 2022] and the inequality (1.133) that ensures the entropy property of the scheme may not be verified in the limit $\mathrm{Ma} \to 0$.

Let us finally highlight that as in [Chalons et al. 2016a; Padioleau et al. 2019] the present approach is rather pragmatic and does not provide reliable analysis and explanation for the low Mach issues. Indeed, we do not study the delicate question of the asymptotic regime $\mathrm{Ma} \to 0$ [P. Degond et M. Tang 2011; Cordier et al. 2012; Zakerzadeh 2016; Bispen et al. 2017; Berthon et al. 2020; Dimarco et al. 2017; Boscarino et al. 2018; Bouchut et al. 2020b], we neither address the strong time step limitation due to the CFL conditions (1.46) when $\mathrm{Ma} \ll 1$ that can be circumvented by using Implicit-Explicit strategies [Chalons et al. 2016a; Dimarco et al. 2018; Boscheri et al. 2020; Bouchut et al. 2020a; Zeifang et al. 2020]. It seems possible to adapt the OSLP Implicit-Explicit strategy of [Chalons et al. 2016a] to the FSLP method. However such task falls beyond the scope of the present and will be investigated in future works. Moreover, the present lines are derived within a one-dimensional setting that does not allow a genuine complete study of allocated issues related to low Mach flows.

1.6.2 . Extension to higher order

The FSLP algorithm can be implemented thanks to a simple single-step evaluation of numerical fluxes. This enables the use of classical high-order enhancements that are available in the literature for finite volume methods such as MUSCL-Hancock [Leer 1977a, 1977b, 1979; Toro 2009; Godlewski et Raviart 2021; R.J. LeVeque 2002], (W)ENO [Liu et al. 1994; Jiang et Shu 1996] or MOOD [Diot et al. 2013; Clain et al. 2011]. For the sake of simplicity, in this chapter, we will only show numerical results with the MUSCL method for which the positivity can be proven under a half CFL condition. Let us consider a linear reconstruction of the primitive variables $\boldsymbol{V} = (\rho, u, p)$ in each cells

$$\tilde{\boldsymbol{V}}_j^n(x) = \boldsymbol{V}_j^n + (x - x_j)\boldsymbol{p}_j^n$$

where the slopes $\boldsymbol{p}_j^n = \boldsymbol{p}^n(\boldsymbol{V}_{j-1}^n, \boldsymbol{V}_j^n, \boldsymbol{V}_{j+1}^n)$ are obtained using a standard slope limiter such as the minmod function [Yee 1989]. Let us introduce the function $H : \boldsymbol{V} \mapsto \boldsymbol{U}$ that converts a state's conservative representation

into its corresponding set of primitive variables. The reconstruction provides a second-order evaluation of the conserved quantities at each interface with

$$\boldsymbol{U}^{n,HO}_{j+1/2,-} = H(\tilde{\boldsymbol{V}_j}^n(x_{j+1/2})), \qquad \boldsymbol{U}^{n,HO}_{j-1/2,+} = H(\tilde{\boldsymbol{V}_{j-1}}^n(x_{j-1/2})), \tag{1.66}$$

that we use to evaluate the FSLP flux function (1.36) at each interface by setting :

$$\begin{aligned}\boldsymbol{U}^{n+1}_j - \boldsymbol{U}^n_j + \frac{\Delta t}{\Delta x}\Big(\mathbf{F}^{\text{FSLP}}(\boldsymbol{U}^{n,HO}_{j+1/2,-}, \boldsymbol{U}^{n,HO}_{j+1/2,+}) - \mathbf{F}^{\text{FSLP}}(\boldsymbol{U}^{n,HO}_{j-1/2,-}, \boldsymbol{U}^{n,HO}_{j-1/2,+})\Big) \\ = \Delta t \boldsymbol{S}_j(\boldsymbol{U}^{n,HO}_{j+1/2,-}, \boldsymbol{U}^{n,HO}_{j+1/2,+}, \boldsymbol{U}^{n,HO}_{j-1/2,-}, \boldsymbol{U}^{n,HO}_{j-1/2,+}).\end{aligned} \tag{1.67}$$

The gravity source term can also be computed with the same formula as in the first-order method by replacing cell-averaged values with the high-precision face-centered values :

$$\boldsymbol{S}_j(\boldsymbol{U}^{n,HO}_{j+1/2,-}, \boldsymbol{U}^{n,HO}_{j+1/2,+}, \boldsymbol{U}^{n,HO}_{j-1/2,-}, \boldsymbol{U}^{n,HO}_{j-1/2,+}) = \begin{pmatrix} 0 \\ \{\rho\partial_x\phi\}^{n,HO}_j \\ \{\rho u\partial_x\phi\}^{n,HO}_j \end{pmatrix},$$

$$\text{with} \begin{cases} \{\rho\partial_x\phi\}^{n,HO}_j = \frac{(\rho\partial_x\phi)^{HO}_{j+1/2} + (\rho\partial_x\phi)^{HO}_{j-1/2}}{2}, \\ \{\rho u\partial_x\phi\}^{n,HO}_j = \frac{u^{*,HO}_{j+1/2}(\rho\partial_x\phi)^{HO}_{j+1/2} + u^{*,HO}_{j-1/2}(\rho\partial_x\phi)^{HO}_{j-1/2}}{2}, \\ (\rho\partial_x\phi)^{HO}_{j+1/2} = \frac{\rho^{n,HO}_{j+1/2,-} + \rho^{n,HO}_{j+1/2,-}}{2}(\partial_x\phi)^{HO}_{j+1/2}, \end{cases} \tag{1.68}$$

where $(\partial_x\phi)^{HO}_{j+1/2}$ is a second-order accurate evaluation of the derivative of the gravitational potential at the interface $x_{j+1/2}$. Note that if the potential is known explicitly, it can be computed exactly at the interface's coordinates $(\partial_x\phi)^{HO}_{j+1/2} = \partial_x\phi(x_{j+\frac{1}{2}})$. In the numerical results presented in section 1.8, we restrict ourselves to a simple linear gravitational potential field $\partial_x\phi = 0, \partial_y\phi = g$. The extension of the well-balanced property is not straightforward and beyond the scope of this chapter. The difficulty lies in predicting the exact amount of diffusion required to be added/removed to precisely cancel out the pressure gradients, as the high-order reconstruction processes are non-linear. Second-order well-balanced methods can be found in [Thomann et al. 2020; Chalons et Del Grosso 2022; Caballero-Cárdenas et al. 2023; Morales de Luna et al. 2020; Del Grosso et Chalons 2021]. The second-order extension (1.67) of the FSLP scheme is positive for density and internal energy as long as it is ensured that :

$$\frac{\Delta t}{\Delta x}\max_{j\in\mathbb{Z}}\left(2\max\left(1/\rho^n_j, 1/\rho^n_{j+1}\right)a_{j+1/2} + \left(u^*_{j-\frac{1}{2}}\right)^+ - \left(u^*_{j+\frac{1}{2}}\right)^-\right) < \frac{1}{2} \tag{1.69}$$

The stability of the second-order method under the conditions above is a direct consequence of the stability of the first-order method. For the second-order extension in time, one can use either the SSP-RK2 method [Spiteri et Ruuth 2002; Gottlieb et Shu 1998] or a classical Hancock update [Toro 2009]. The latter option is tested numerically in section 1.8.3 where we check the 2nd order of accuracy of the FSLP-MUSCL-Hancock method on the isentropic vortex test case [Shu 1998].

1.6.3 . Multidimensional extension

Before going any further, let us introduce the notations for our 2D space discretization : we consider two strictly increasing sequences $(x_{i+1/2})_{i\in\mathbb{Z}}$ and $(y_{j+1/2})_{j\in\mathbb{Z}}$ and divide the real plane into cells where the ij^{th}

cell is the interval $\left(x_{i-1/2}, x_{i+1/2}\right) \times \left(y_{j-1/2}, x_{j+1/2}\right)$. The space steps of the ij^{th} cell are $\Delta x = x_{i+1/2} - x_{i-1/2} > 0$ and $\Delta y_j = y_{j+1/2} - y_{j-1/2} > 0$. We consider a discrete initial data $\boldsymbol{U}_{ij}^0$ defined by $\boldsymbol{U}_{ij}^0 = \frac{1}{\Delta x_i \Delta y_j} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \boldsymbol{U}^0(x,y) \mathrm{d}x\mathrm{d}y$, for $(i,j) \in \mathbb{Z}^2$. Let us introduce the Euler equations of gas dynamicss in two dimensions of space :

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) + \partial_y \boldsymbol{G}(\boldsymbol{U}) = \boldsymbol{S}(\boldsymbol{U}, \phi), \qquad \text{for } (x,y) \in \mathbb{R}^2, t > 0, \tag{1.70}$$

with $\boldsymbol{U} = (\rho, \rho u, \rho v, \rho E)^T$, $\boldsymbol{F}(\boldsymbol{U}) = (\rho u,\ u\rho u + p, u\rho w,\ u\rho E + pu)^T$, $\boldsymbol{G}(\boldsymbol{U}) = (\rho v,\ v\rho u, v\rho w + p,\ v\rho E + pv)^T$, and $\boldsymbol{S}(\boldsymbol{U}, \phi) = -\rho \partial_x \phi (0,1,0,u)^T - \rho \partial_y \phi (0,0,1,v)^T$ where v is the velocity in the y direction and ϕ is smooth enough so that we can consider that $\partial_x \phi$, $\partial_y \phi$ are also regular and bounded. We take advantage of the rotational invariance of the 2D Euler system and discretize the fluxes direction by direction :

$$\left\{ \begin{array}{l} \rho_{i,j}^{n+1} = \rho_{i,j}^n - \dfrac{\Delta t}{\Delta x}\left(u^*_{i+1/2,j}\rho^n_{i+1/2,j} - u^*_{i-1/2,j}\rho^n_{i-1/2,j}\right) + \dfrac{\Delta t}{\Delta y}\left(v^*_{i,j+1/2}\rho^n_{i,j+1/2} - v^*_{i-1/2,j}\rho^n_{i-1/2,j}\right), \\ (\rho u)_{i,j}^{n+1} = (\rho u)_{i,j}^n - \dfrac{\Delta t}{\Delta x}\left(u^*_{i+1/2,j}(\rho u)^n_{i+1/2,j} + \Pi^{\theta^x,*}_{i+1/2,j} - u^*_{i-1/2,j}(\rho u)^n_{i-1/2,j} - \Pi^{\theta^x,*}_{i-1/2,j}\right) \\ \qquad\qquad + \dfrac{\Delta t}{\Delta y}\left(v^*_{i,j+1/2}(\rho u)^n_{i,j+1/2} - v^*_{i,j-1/2}(\rho u)^n_{i,j-1/2}\right) - \{\rho \partial_x \phi\}^n_{ij}, \\ (\rho v)_{i,j}^{n+1} = (\rho v)_{i,j}^n - \dfrac{\Delta t}{\Delta x}\left(u^*_{i+1/2,j}(\rho v)^n_{i+1/2,j} - u^*_{i-1/2,j}(\rho v)^n_{i-1/2,j}\right) \\ \qquad\qquad + \dfrac{\Delta t}{\Delta y}\left(v^*_{i,j+1/2}(\rho v)^n_{i,j+1/2} + \Pi^{\theta^y,*}_{i,j+1/2} - v^*_{i,j-1/2}(\rho v)^n_{i,j-1/2} - \Pi^{\theta^y,*}_{i,j-1/2}\right) - \{\rho \partial_y \phi\}^n_{ij}, \\ (\rho E)_{i,j}^{n+1} = (\rho E)_{i,j}^n - \dfrac{\Delta t}{\Delta x}\left(u^*_{i+1/2,j}(\rho E)^n_{i+1/2,j} + \Pi^{\theta^x,*}_{i+1/2,j}u^*_{i+1/2,j} - u^*_{i-1/2,j}(\rho E)^n_{i-1/2,j} - \Pi^{\theta^x,*}_{i-1/2,j}u^*_{i-1/2,j}\right) \\ \qquad\qquad + \dfrac{\Delta t}{\Delta y}\left(v^*_{i,j+1/2}(\rho E)^n_{i,j+1/2} + \Pi^{\theta^y,*}_{i,j+1/2}v^*_{i,j+1/2} - v^*_{i,j-1/2}(\rho E)^n_{i,j-1/2} - \Pi^{\theta^y,*}_{i,j-1/2}v^*_{i,j-1/2}\right) \\ \qquad\qquad - \{(\rho u \partial_x + \rho v \partial_y)\phi\}^n_{ij}. \end{array} \right. \tag{1.71}$$

with

$$\left\{ \begin{array}{l} u^*_{i+1/2,j} = \frac{u^n_{i+1,j}+u^n_{i,j}}{2} - \frac{1}{2a_{i+1/2,j}}\left(p^n_{i+1,j} - p^n_{i,j} + \frac{\rho^n_{i+1,j}+\rho^n_{i,j}}{2}(\phi_{i+1,j} - \phi_{i,j})\right), \\ v^*_{i+1/2,j} = \frac{v^n_{i,j+1}+v^n_{i,j}}{2} - \frac{1}{2a_{i,j+1/2}}\left(p^n_{i,j+1} - p^n_{i,j} + \frac{\rho^n_{i,j+1}+\rho^n_{i,j}}{2}(\phi_{i,j+1} - \phi_{i,j})\right), \\ \Pi^{*,\theta^x}_{i+1/2,j} = \frac{p^n_{i+1,j}+p^n_{i,j}}{2} - \theta^x_{i+1/2,j}\frac{a_{i+1/2,j}}{2}\left(u^n_{i+1,j} - u^n_{i,j}\right), \\ \Pi^{*,\theta^y}_{i,j+1/2} = \frac{p^n_{i,j+1}+p^n_{i,j}}{2} - \theta^y_{i,j+1/2}\frac{a_{i,j+1/2,}}{2}\left(u^n_{i,j+1} - u^n_{i,j}\right), \end{array} \right. \tag{1.72}$$

as well as the source terms discretization :

$$\left\{ \begin{array}{l} \{\rho\partial_x\phi\}^n_{i,j} = \frac{(\rho\partial_x\phi)_{i+1/2,j}+(\rho\partial_x\phi)_{i-1/2,j}}{2}, \\ \{\rho\partial_y\phi\}^n_{i,j} = \frac{(\rho\partial_y\phi)_{i,j+1/2}+(\rho\partial_y\phi)_{i,j-1/2}}{2}, \\ \{\rho u\partial_x\phi\}^n_{i,j} = \frac{u^*_{i+1/2,j}(\rho\partial_x\phi)_{i+1/2,j}+u^*_{i-1/2,j}(\rho\partial_x\phi)_{i-1/2,j}}{2}, \\ \{\rho u\partial_y\phi\}^n_{i,j} = \frac{v^*_{i,j+1/2}(\rho\partial_y\phi)_{i,j+1/2}+v^*_{i,j-1/2}(\rho\partial_y\phi)_{i,j-1/2}}{2}, \\ (\rho\partial_x\phi)_{i+1/2,j} = \frac{\rho^n_{j+1}+\rho^n_{i,j}}{2}\frac{\phi_{i+i,j}-\phi_{i,j}}{\Delta x}, \\ (\rho\partial_y\phi)_{i,j+1/2} = \frac{\rho^n_{i,j+1}+\rho^n_{i,j}}{2}\frac{\phi_{i,j+i}-\phi_{i,j}}{\Delta y}. \end{array} \right. \tag{1.73}$$

1.7 . Flux-splitting as a relaxation approximation

The goal of this section is to highlight the connection between the FSLP flux-splitting approach and a relaxation approximation. In the previous sections, we concluded that the FSLP approach could be expressed as an averaging procedure (1.45) where $\boldsymbol{U}_j^P$ and $\boldsymbol{U}_j^A$ are defined as approximate solutions of two systems (1.41) and (1.42) that respectively only account for the pressure and the advection effects. We propose to translate that three-step process thanks to a relaxation approximation. Suppose that $\alpha \in (0,1)$ is a constant and let ν be a positive parameter, we consider the system

$$\partial_t \begin{bmatrix} \rho^P \\ \rho^P u^P \\ \rho^P E^P \\ \rho^P \Pi^P \\ \rho^P \mathcal{T}^P \\ \phi \end{bmatrix} + \frac{1}{\alpha}\partial_x \begin{bmatrix} 0 \\ \Pi^P \\ \Pi^P u^P \\ a^2 u^P \\ -u^P \\ 0 \end{bmatrix} + \frac{1}{\alpha} \begin{bmatrix} 0 \\ \rho^P \\ \rho^P u^P \\ 0 \\ 0 \\ 0 \end{bmatrix} \partial_x \phi = \nu \begin{bmatrix} \alpha\rho^P + (1-\alpha)\rho^A - \rho^P \\ \alpha\rho^P u^P + (1-\alpha)\rho^A u^A - \rho^P u^P \\ \alpha\rho^P E^P + (1-\alpha)\rho^A E^A - \rho^P E^P \\ p^{\text{EOS}}(1/\rho^P, e^P) - \Pi^P \\ 1 - \rho^P \mathcal{T}^P \\ 0 \end{bmatrix}, \tag{74a$_\nu$}$$

$$\partial_t \begin{bmatrix} \rho^A \\ \rho^A u^A \\ \rho^A E^A \\ \rho^A \Pi^A \\ \rho^A \mathcal{T}^A \end{bmatrix} + \left(\frac{1}{1-\alpha}\right)\partial_x \begin{bmatrix} \rho^A u^P \\ \rho^A u^A u^P \\ \rho^A E^A u^P \\ \rho^A \Pi^A u^P \\ u^P \end{bmatrix} = \nu \begin{bmatrix} \alpha\rho^P + (1-\alpha)\rho^A - \rho^A \\ \alpha\rho^P u^P + (1-\alpha)\rho^A u^A - \rho^A u^A \\ \alpha\rho^P E^P + (1-\alpha)\rho^A E^A - \rho^A E^A \\ p^{\text{EOS}}(1/\rho^A, e^A) - \Pi^A \\ 1 - \rho^A \mathcal{T}^A \end{bmatrix}. \tag{74b$_\nu$}$$

The system (1.74$_\nu$) features a pair of duplicate conservative variables $(\boldsymbol{U}^P, \boldsymbol{U}^A)$ and 4 other variables : Π^P, Π^A, $\mathcal{T}^A$ and $\mathcal{T}^P$. The variables Π^P and Π^A are surrogate for the thermodynamical pressure, while $\mathcal{T}^A$ and $\mathcal{T}^P$ play the role of a pseudo-specific volume. It is possible to view (1.74$_\nu$) as a Suliciu relaxation approximation with a separation of the acoustic and transport operators. Indeed, (1.74$_\nu$) implies that

$$\partial_t \begin{bmatrix} \alpha\rho^P + (1-\alpha)\rho^A \\ \alpha\rho^P u^P + (1-\alpha)\rho^A u^A \\ \alpha\rho^P E^P + (1-\alpha)\rho^A E^A \end{bmatrix} + \partial_x \begin{bmatrix} \rho^A u^P \\ \rho^A u^A u^P + \Pi^P \\ \rho^A E^A u^P + \Pi^P u^P \end{bmatrix} = \begin{bmatrix} 0 \\ -\rho^P \\ -\rho^P u^P \end{bmatrix} \partial_x \phi \tag{1.75a}$$

$$\partial_t \big[\alpha\rho^P \Pi^P + (1-\alpha)\rho^A \Pi^A\big] + \partial_x \big(\rho^A \Pi^A u^P + a^2 u^P\big) \tag{1.75b}$$

$$= \nu \left[\alpha p^{\text{EOS}}\left(\frac{1}{\rho^P}, e^P\right) + (1-\alpha) p^{\text{EOS}}\left(\frac{1}{\rho^A}, e^A\right) - \alpha\Pi^P - (1-\alpha)\Pi^A\right], \tag{1.75c}$$

$$\partial_t \big[\alpha\rho^P \mathcal{T}^P + (1-\alpha)\rho^A \mathcal{T}^A\big] = \nu\big[1 - \alpha\rho^P \mathcal{T}^P - (1-\alpha)\rho^A \mathcal{T}^A\big]. \tag{1.75d}$$

Taking the limit $\nu \to +\infty$ formally enforces that $\boldsymbol{U}^P = \boldsymbol{U}^A = \boldsymbol{U} = (\rho, \rho u, \rho E)^T$ and $\Pi^A = \Pi^P = p^{\text{EOS}}(1/\rho, e)$, so that (1.75a) enables to retrieve the Euler system (1.1). This suggests that we can use the relaxation system (1.74$_\nu$) as an approximation of (1.1) in the limit $\nu \to +\infty$. The equation (1.75c) plays here a similar role as the surrogate pressure equation in the classic Suliciu approximation [Suliciu 1998; Chalons et Coulombel 2008; Coquel et al. 2012a]. The sole purpose of equation (1.75d) is to ensure that $\alpha\rho^P \mathcal{T}^P + (1-\alpha)\rho^A \mathcal{T}^A = 1$ in the regime $\nu \to \infty$. In our discretization strategy, we classically mimic the $\nu \to \infty$ regime for $t \in [t^n, t^{n+1})$, by enforcing $(\boldsymbol{U}^P, \Pi^P, \mathcal{T}^A, \boldsymbol{U}^A, \Pi^A, \mathcal{T}^A)(t = t^n) = (\boldsymbol{U}, p^{\text{EOS}}(1/\rho, e), 1/\rho, \boldsymbol{U}, p^{\text{EOS}}(1/\rho, e), 1/\rho)(t = t^n)$ and by solving the relaxation off-equilibrium system (1.74$_{\nu=0}$). The properties of the off-equilibrium system (1.74$_{\nu=0}$) are briefly summarized in the following proposition whose proof is given in 1.D.

Proposition 1.7.1. *The system (1.74$_{\nu=0}$) is hyperbolic with a set of characteristic velocities given by :* $\dfrac{u^P}{1-\alpha}$ *(with an algebraic multiplicity 4),* 0 *(with an algebraic multiplicity 5) and* $\pm\dfrac{a}{\alpha\rho^P}$*. Moreover, (1.74$_{\nu=0}$) only involves linearly degenerate fields.*

The relaxation formulation (1.74$_\nu$) sheds some more light on the similarities between the flux-splitting we propose here and the acoustic/transport operator splitting strategy presented in [Padioleau et al. 2019]. Indeed, the source term and pressure effects can be treated separately from the advection terms. The difference is that although the operators are separated, they are re-distributed within a larger single system instead of two separate systems.

By discretizing the pressure and advection parts of (1.9) identically than in section 1.5.2, we re-obtain the same update formula (1.33), which yields the FSLP scheme (1.36). Finally, let us mention that it is possible to build an alternate flux-splitting method for the system (1.1) by seeking the solution of the Riemann problem for (1.74$_{\nu=0}$). This option is not studied in the present work.

1.8 . Numerical experiments

In this section, we consider that the fluid is a perfect gas with the EOS $p=(\gamma-1)\rho e$ and that the potential ϕ takes the form $\phi(x,y)=-gy$ for tests that involve the source term.

We will present numerical experiments with the FSLP method and the HLLC Riemann solver [Toro 2009] using first and second-order discretizations. The second-order accuracy is achieved using a MUSCL-Hancock strategy [Toro 2009] for both the HLLC and FSLP solvers. Let us mention that the slope reconstruction is performed on the primitive variables with a minmod slope limiter [Godlewski et Raviart 1990 ; R.J. LeVeque 2002 ; Toro 2009]. For the OSLP method, noting $(a/\rho)_{j+1/2}=\max\left(1/\rho_j^n,1/\rho_{j+1}^n\right)a_{j+1/2}$, the time steps Δt is computed as follows :

$$\Delta t = C^{\text{CFL}}\,\Delta x\,\frac{1}{\max_{j\in\mathbb{Z}}\left[\max\left\{2\max\left[(a/\rho)_{j\pm1/2}\right],\left(\left(u^*_{j-\frac{1}{2}}\right)^+-\left(u^*_{j+\frac{1}{2}}\right)^-\right)\right\}\right]}, \tag{1.76}$$

For the FSLP method, it is computed as follows :

$$\Delta t = C^{\text{CFL}}\,\Delta x\,\frac{1}{\max_{j\in\mathbb{Z}}\left[2\max\left[(a/\rho)_{j\pm1/2}\right]+\left(\left(u^*_{j-\frac{1}{2}}\right)^+-\left(u^*_{j+\frac{1}{2}}\right)^-\right)\right]}, \tag{1.77}$$

where the parameter C^{CFL} is given by the table 1.1 so that the CFL conditions (1.22) and(1.23) for the OSLP method, (1.35) for the first-order FSLP method and (1.69) for the second-order FSLP method are all satisfied. For the HLLC solver, the standard CFL from [Toro et al. 1994] is used.

The parameter θ related to the low Mach correction is defined at each interface $(i+1/2,j)$ and $(i,j+1/2)$ by

$$\theta^x_{i+1/2,j}=\max\left(\mid u_{i,j}\mid/c_i,\mid u_{i+1,j}\mid/c_{i+1,j}\right),\qquad \theta^y_{i,j+1/2}=\max\left(\mid v_{i,j}\mid/c_i,\mid v_{i,j+1}\mid/c_{i,j+1}\right). \tag{1.78}$$

Numerical scheme	first-order	second-order
OSLP	1.0	N.A.
FSLP	1.0	1/2
HLLC	1.0	1/2

Table 1.1 – Values for C^{CFL} used in the simulations.

Note that our choice for the computation of θ differs from [Chalons et al. 2016a] that uses the inerface velocity u^*, v^*. Both choices give satisfactory results and are valid estimations of the local Mach number Ma. Depending on the interface values of velocities and pressure, one choice can be more or less diffusive than the other. However, no significant differences have been observed in our experiments.

1.8.1 . Sod shock tube test case

We consider here the classical Sod shock tube test case [Sod 1978; Toro 2009] : we set $\gamma = 1.4$ and the initial conditions are :

$$(\rho, u, p)(x, t = 0) = \begin{cases} (1,\ 0,\ 1) & \text{if} \quad x < 0.5, \\ (0.125,\ 0,\ 0.1) & \text{if} \quad x > 0.5. \end{cases}$$

The goal of this test is to study the ability of our solver to handle different wave types. The initial discontinuity generates three waves : a leftward going rarefaction, a contact discontinuity, and a shock that both travel towards the right of the computational domain. Figure 1.1 shows the profile obtained at $t = 0.2s$ with five different solvers : OSLP, FSLP/HLLC for the first and second-order methods. At first order, the HLLC solver provides the sharpest resolution of the shock and contact discontinuity. The differences between the FSLP and OSLP methods are hardly visible. None of the schemes suffers from spurious oscillations and both the position and the amplitude of the waves match the exact solution. We also note that the OSLP method is slightly sharper than the FSLP method on the rarefaction and contact discontinuity. In section 1.8.3, we compare the accuracy of both method on the isentropic vortex test case.

1.8.2 . Two-rarefaction test case

We now consider the two-rarefaction test proposed by Einfeldt [Einfeldt et al. 1991; Toro 2009] for a perfect gas with $\gamma = 1.4$. The initial conditions are

$$(\rho, u, p)(x, t = 0) = \begin{cases} (1,\ -2,\ 0.4), & \text{if } x < 0.5, \\ (1,\ 2,\ 0.4), & \text{if } x > 0.5. \end{cases}$$

The resulting wave pattern features two rarefaction waves that split from the position $x = 0.5$, traveling towards each end of the computational domain. As a result, a near vacuum region presenting low densities and pressures appears in the middle of the domain.

Figure 1.2 shows that all methods are robust enough to preserve positivity for mass, pressure, and energy so that they are able to reach the end of the simulation. Moreover, none of the numerical schemes exhibit entropy-related issues like the apparition of nonphysical shocks within the wave pattern.

1.8.3 . Grid convergence – the isentropic vortex test

The accuracy of our FSLP scheme equipped with a MUSCL-Hanckock strategy is considered on a classical 2D test problem called the non-linear isentropic vortex advection presented by [Shu 1998]. As in [Reyes et al. 2019],

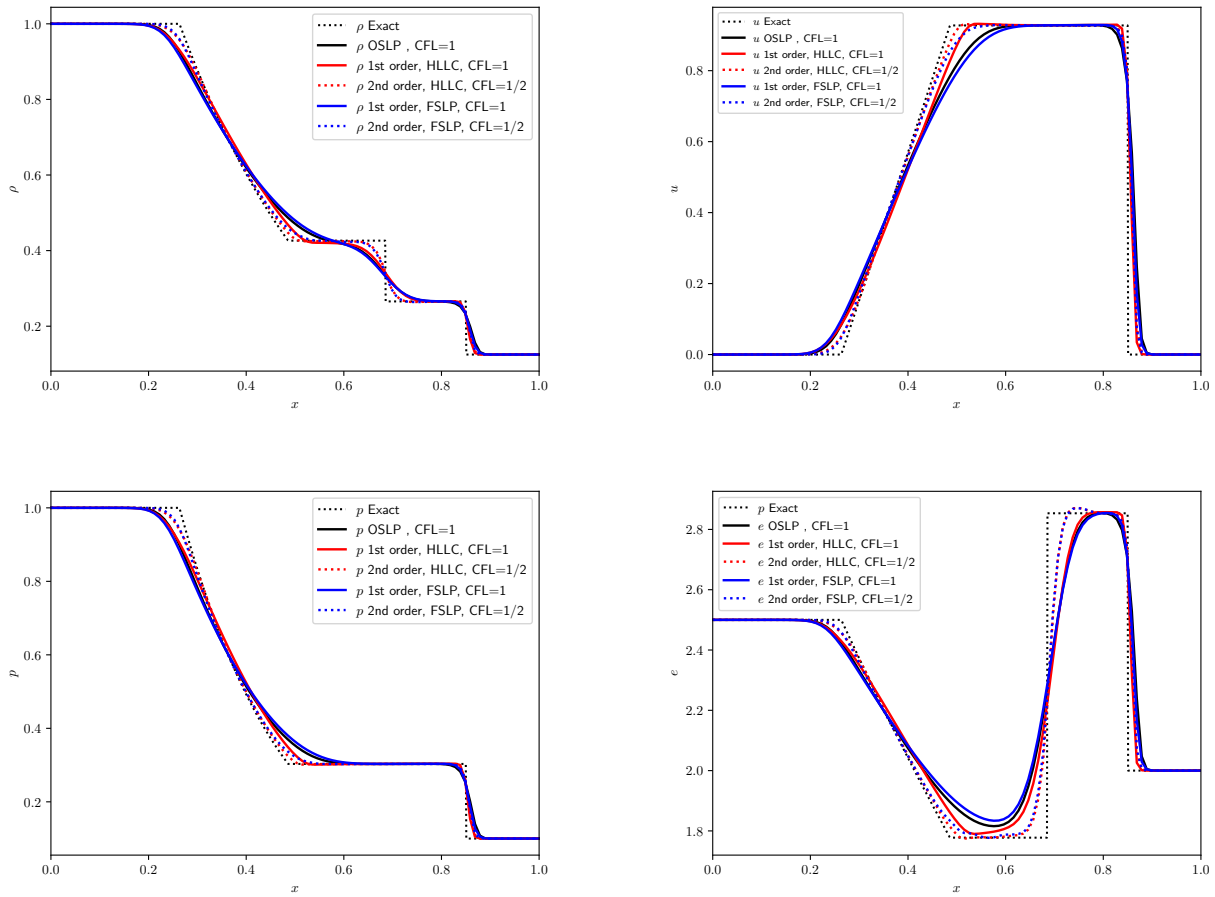


Figure 1.1 – Sod shock tube test case. Profile at $t = 0.2s$ of the density (top left), velocity (top right), pressure (bottom left), and specific internal energy (bottom right). The results are obtained with the OSLP method, first and second-order FSLP method, first and second-order HLLC scheme, and the exact solution on a 100-cell grid.

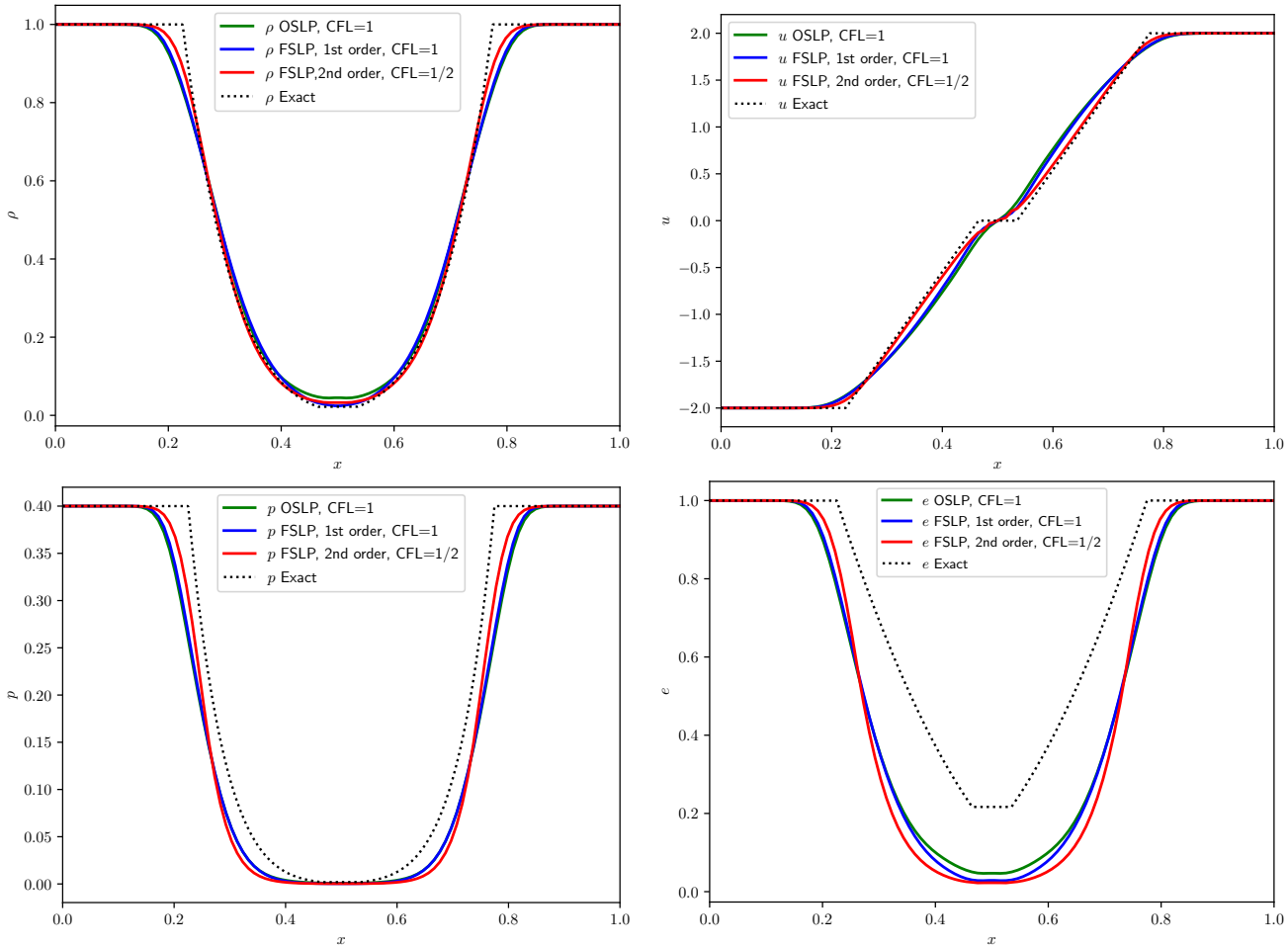


Figure 1.2 – Two-rarefaction test case. Profile at $t = 0.1s$ of the density (top left), velocity (top right), pressure (bottom left), and specific internal energy (bottom right). The results are obtained with the OSLP method, the first and second-order FSLP method, and the exact solution on a 100-cell grid.

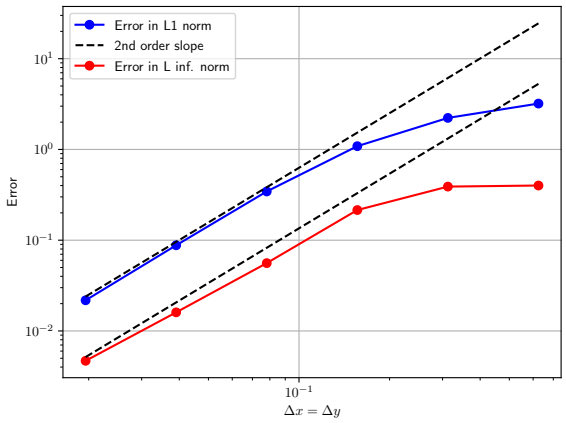


Figure 1.3 – Convergence study of the FSLP method extended to second order via a MUSCL-Hancock strategy

we double the original domain size to avoid self-interactions of the vortex across the periodic domain. The test involves a circular region centered at $(x_c, y_c) = (10, 10)$ on a periodic square domain, $[0, 20] \times [0, 20]$, where a Gaussian-shaped vortex with a rotating velocity field is initialized. The problem consists in advecting the vortex along the diagonal direction, therefore any departure from the initial condition (or the exact solution of the problem) will be considered numerical errors of the numerical method under consideration. The initial condition proposed in [Shu 1998] defines the values of the primitive variables at $t = 0$ as follows

$$\rho(x, y) = \left[1 - (\gamma - 1)\frac{\beta^2}{8\gamma\pi^2}e^{1-r^2}\right]^{\frac{1}{\gamma-1}}, \tag{1.79a}$$

$$u(x, y) = 1 - \frac{\beta}{2\pi}e^{\frac{1}{2}(1-r^2)}(y - y_c), \tag{1.79b}$$

$$v(x, y) = 1 + \frac{\beta}{2\pi}e^{\frac{1}{2}(1-r^2)}(x - x_c), \tag{1.79c}$$

$$p(x, y) = \rho(x, y)^{\gamma}, \tag{1.79d}$$

with $r = r(x, y) = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ and the vortex strength $\beta = 5$. Due to the velocity field, $(u, v) = (1, 1)$, the vortex is translated across the diagonal direction of the computational domain and returns to the initial position at $t = 20s$. The numerical error is then compared at this instant using the initial condition as the value of the exact solution. We run 6 simulations corresponding to the resolutions $[Nx, Ny] = [N, N]$, $N \in \{32, 64, 128, 256, 512, 1024\}$ and display the L^1 and L^∞ errors in figure 1.3. The L^1 and L^∞ errors are computed for the density as $\Delta x \Delta y \sum_{i,j} \mid \rho^n_{i,j} - \rho^{i,j}_0 \mid$ and $\max_{i,j} \mid \rho^n_{i,j} - \rho^{i,j}_0 \mid$ respectively. One can see that convergence rate of the numerical method follows a second-order slope, validating our high-order extension.

The isentropic vortex test cases also allows us to compare the accuracy of the OSLP and FSLP methods. We ran two simulations on 512^2 grids with both methods (at first order of accuracy). The L^1 error of the FSLP method is about 10% higher than the OSLP method. Note that this number may vary for different test cases and resolutions. In section 1.8.1, we also observed that the FSLP method is slightly less accurate on the Sod shock tube test case.

1.8.4 . The Gresho vortex

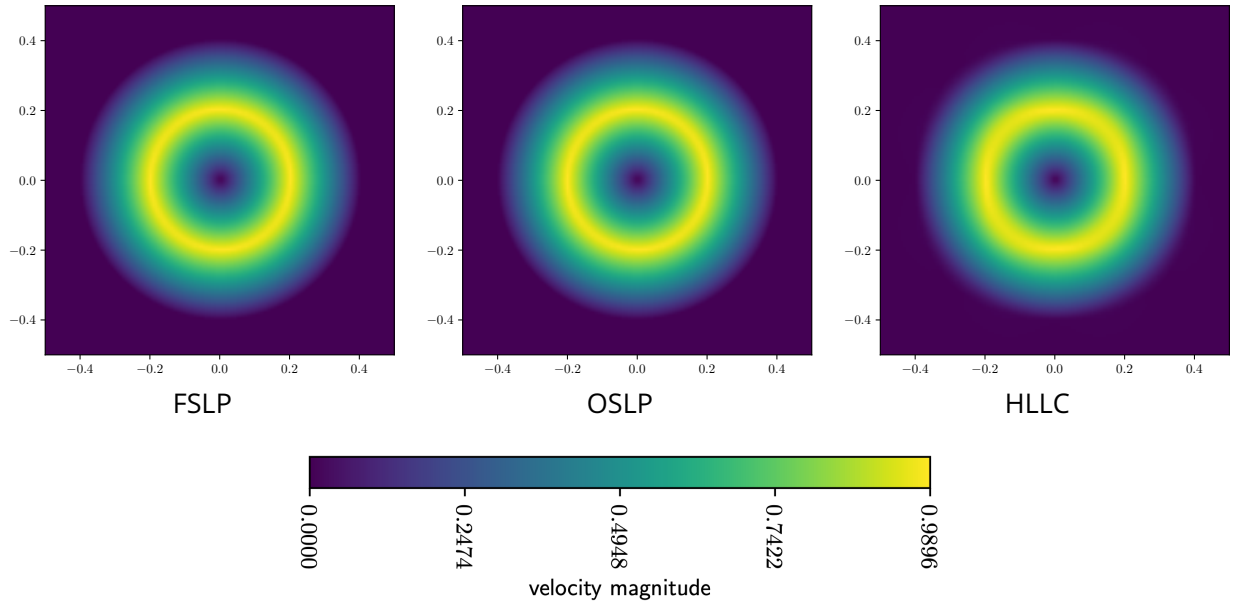


Figure 1.4 – Comparison of the final velocity magnitude map for the Gresho vortex test case with $Ma = 10^{-1}$ obtained with the FSLP, OSLP, and HLLC solvers on a 128×128-cell grid at $t = 0.1s$.

The Gresho vortex [Gresho et Chan 1990] involves a a stationary vortex that can be parameterized by the maximum value of the Mach number Ma across the computational domain. Therefore this test is very useful for studying the performance of numerical schemes in the low Mach regime. We consider a perfect gas with $\gamma = 1.4$. Using polar coordinates (r, θ), the initial conditions read :

$$\rho(r, \theta, t = 0) = 1, \tag{1.80a}$$

$$(u_r, u_\theta)\,(r, \theta, t = 0) = \begin{cases} (0, 5r) & \text{if } 0 \leq r < 0.2, \\ (0, 2 - 5r) & \text{if } 0.2 \leq r < 0.4, \\ (0, 0) & \text{if } 0.4 \leq r, \end{cases} \tag{1.80b}$$

$$p(r, \theta, t = 0) = \begin{cases} p_0 + 12.5r^2 & \text{if } 0 \leq r < 0.2, \\ p_0 + 12.5r^2 + 4 - 20r + 4\ln(5r) & \text{if } 0.2 \leq r < 0.4, \\ p_0 - 2 + 4\ln 2 & \text{if } 0.4 \leq r, \end{cases} \tag{1.80c}$$

where $p_0 = \frac{1}{\gamma Ma^2}$. For the simulations, we will use three different values for the reference Mach number : $Ma \in \{10^{-1}, 10^{-3}, 10^{-5}\}$. We will compare the distributions of the velocity magnitude obtained at $t = 10^{-2}s$ with the initial conditions.

Figures 1.4, 1.5, 1.6 give us the final velocity magnitude map for the Gresho vortex obtained with different solvers and Mach numbers. For $Ma = 10^{-1}$, we can see in figure 1.4 that on all three simulations, the initial velocity ring is preserved. Figure 1.5 displays the results for $Ma = 10^{-3}$: one can see that the FSLP and OSLP methods can both preserve the velocity ring thanks to the low Mach correction while the HLLC methods fail to do so. The same behavior is observed for $Ma = 10^{-5}$ (see figure 1.6). In order to measure the numerical diffusion effect of the solver, we evaluate the ratio $e_{\text{kin}}/e^0_{\text{kin}}$ of the kinetic energy obtained at the final instant and the initial

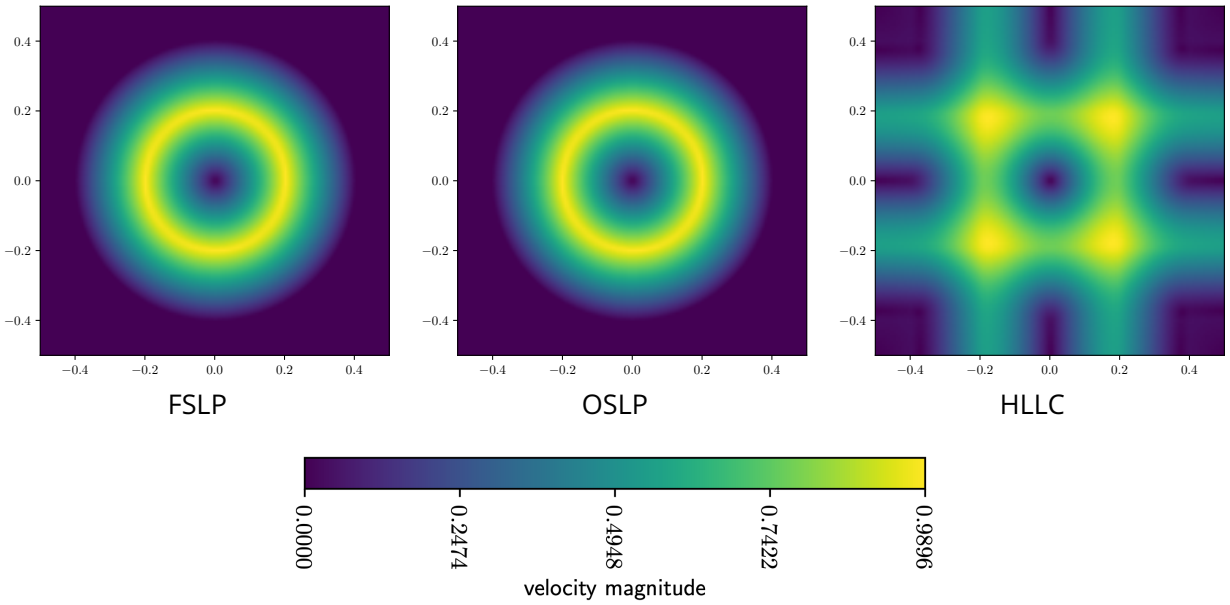


Figure 1.5 – Comparison of the final velocity magnitude map for the Gresho vortex test case with $Ma = 10^{-3}$ obtained with the FSLP, OSLP, and HLLC solvers on a 128×128-cell grid at $t = 0.1s$.

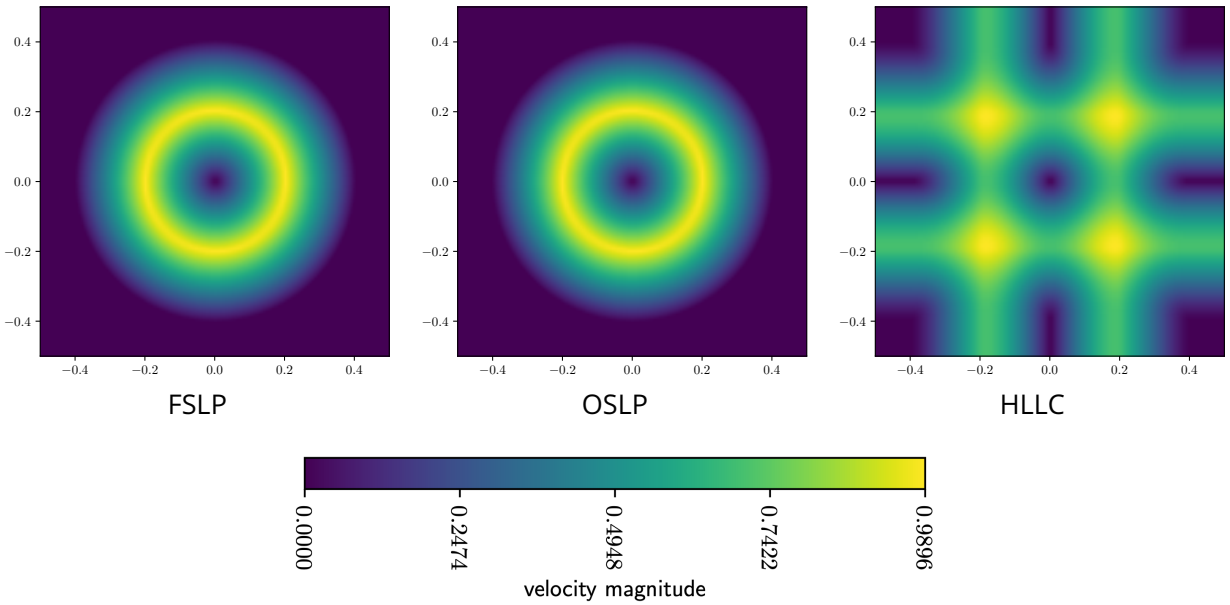


Figure 1.6 – Comparison of the final velocity magnitude map for the Gresho vortex test case with $Ma = 10^{-5}$ obtained with the FSLP, OSLP, and HLLC solvers on a 128×128-cell grid at $t = 0.1s$.

instant with

$$e_{\text{kin}} = \sum_j \frac{1}{2}\rho_j^n\big((u_j^n)^2 + (v_j^n)^2\big)\Delta x^2, \qquad e_{\text{kin}}^0 = \sum_j \frac{1}{2}\rho_j^0\big((u_j^0)^2 + (v_j^0)^2\big)\Delta x^2. \tag{1.81}$$

The results are displayed in table 1.2. They show that both FSLP and OSLP solvers better preserve the kinetic

Table 1.2 – Gresho vortex test case : evaluation of the kinetic energy in the computational domain for different values of the Mach number Ma.

	$Ma = 10^{-1}$	$Ma = 10^{-3}$	$Ma = 10^{-5}$
e_{kin}/e_{kin}^0 (at $t = 10^{-2}$) — OSLP scheme	0.9966	0.9966	0.9966
e_{kin}/e_{kin}^0 (at $t = 10^{-2}$) — FSLP scheme	0.9966	0.9966	0.9966
e_{kin}/e_{kin}^0 (at $t = 10^{-2}$) — HLLC scheme	0.9762	0.5262	0.5167

energy than the HLLC method in the low Mach regime.

1.8.5 . Two-dimensional Riemann problems

We now intend to study the ability of the FSLP method to capture more complex wave patterns in a two-dimensional setting, including shocks and rarefaction waves. To that end, we consider the popular 2D Riemann problem of the literature referred to as Configuration 3 in [Liska et Wendroff 2003]. The computational domain is the rectangle $[0, 1] \times [0, 1]$, with the initial conditions

$$(\rho, u, v, p)(x, y, t = 0) = \begin{cases} (0.138,\ 1.206,\ 1.206,\ 0.029) & \text{if } x < 0.8, y < 0.8 \quad \text{(bottom left)} \\ (0.5323,\ 0.0,\ 1.206,\ 0.3) & \text{if } x > 0.8, y < 0.8 \quad \text{(bottom right)} \\ (0.5323,\ 1.206,\ 0.0,\ 0.3) & \text{if } x < 0.8, y > 0.8 \quad \text{(top left)} \\ (1.5,\ 0.0,\ 0.0\ 1.5) & \text{if } x > 0.8, y > 0.8 \quad \text{(top right).} \end{cases} \tag{1.82}$$

We impose homogeneous Neumann conditions at the boundaries. We compute a reference solution thanks to a second-order HLLC method on a 384×384-grid. The waves at play produce a jet that propagates along the diagonal $x = y$ creating an important low Mach region in the center and the top right part of the domain (see figure 1.7).

Figure 1.8 shows a mapping of the density obtained with the OSLP (first-order), the FSLP (first and second-order), and the HLLC (first and second-order) schemes using a 128×128-cell mesh. One can see that the overall wave pattern is rendered successfully by all numerical schemes. The results of the FSLP, OSLP and HLLC schemes for first-order methods are similar. Second-order methods all better succeed in capturing the shape of the jet as depicted in figure 1.9. Although the HLLC scheme poorly performs in the low Mach regime on a coarse grid, this defect vanishes when one refines the grid [Dellacherie 2010]. Therefore, for the present test, we use a simulation performed with the HLLC solver on a 400^2 Cartesian grid as reference solution. The objective is here to attest that comparable accuracy can be obtained with the FSLP solver on a coarser grid. Nevertheless, we can note that spurious oscillations appear in the simulation performed with the second-order FSLP scheme with low Mach correction. These spurious waves propagate along the x and y axes in the top right part of the domain. We believe they are caused by the lack of numerical dissipation around the low Mach shocks due to the combination of the low Mach correction and the second-order reconstruction. A more careful choice of θ than (1.78) is required to

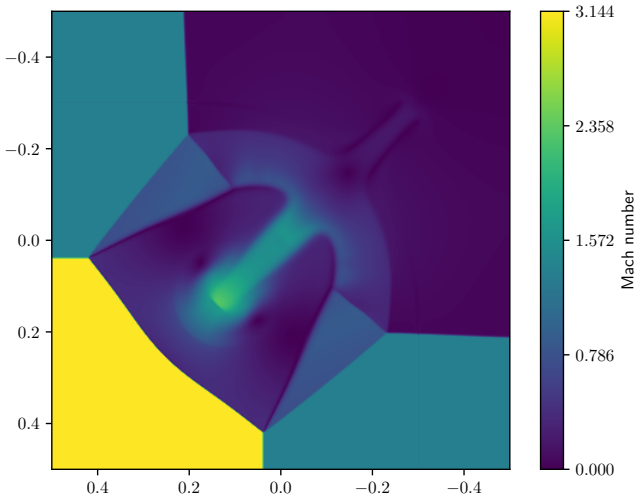


Figure 1.7 – 2D Riemann problem. Mapping of the Mach number as $t = 0.8s$. The reference simulation is obtained with a second-order HLLC method on a 384×384-cell mesh.

s

ensure the discrete entropy inequality (see (1.24)). Improving the second-order discretization for the FSLP scheme would, for example, require proposing a better choice for θ but such a task is beyond the scope of the present work.

1.8.6 . Hydrostatic equilibrium test

In order to challenge the well-balanced ability of the FSLP scheme, we consider the atmosphere at rest test (see, for example [Padioleau et al. 2019]). It involves a fluid column of a perfect gas in a rectangular $[0, 2] \times [0, 1]$ domain. For this test, the gravity acceleration is set to $g = -1$ so that $\phi(x, y) = -y$. For the EOS of the fluid, we set $\gamma = 5/3$ and $c_v = 1$, where c_v is the heat capacity at constant volume so that the temperature T of the gas is given by $e = c_v T$. We consider periodic boundary conditions for the left and right sides of the domain. At the top and bottom of the domain, wall boundaries are imposed for the normal velocity, while the temperature is linearly extrapolated. The initial condition is built by imposing a linear temperature profile as follows

$$T(x, y = 0, t = 0) = 3.78565, \qquad \boldsymbol{\nabla} T(x, y, t = 0) = (0, -1.2)^T, \tag{1.83a}$$

$$\rho(x, y = 0, t = 0) = 1, \qquad \boldsymbol{\nabla}(c_v(\gamma - 1)\rho T)(x, y, t = 0) = (0, \rho g)^T. \tag{1.83b}$$

The computational domain is discretized over a 100×50 on which we let the solver evolve the profile for $t \in [0, 100s]$. Table 1.3 displays the value of the $\max_{i,j} |v^n_{i,j}|$ at $t = 100s$ and shows that both the OSLP and the FSLP first-order methods preserve the velocity magnitude at zero-machine precision.

Table 1.3 – Hydrostatic equilibrium test. Measure of the velocity magnitude at $t = 100s$

Solver	OSLP	FSLP
Average speed	1.342×10^{-14}	2.056×10^{-14}

It is important to mention that a direct second-order extension of the well-balanced method, as presented in section 1.6.2 will fail to preserve the hydrostatic equilibrium. This question of designing a well-balanced high-

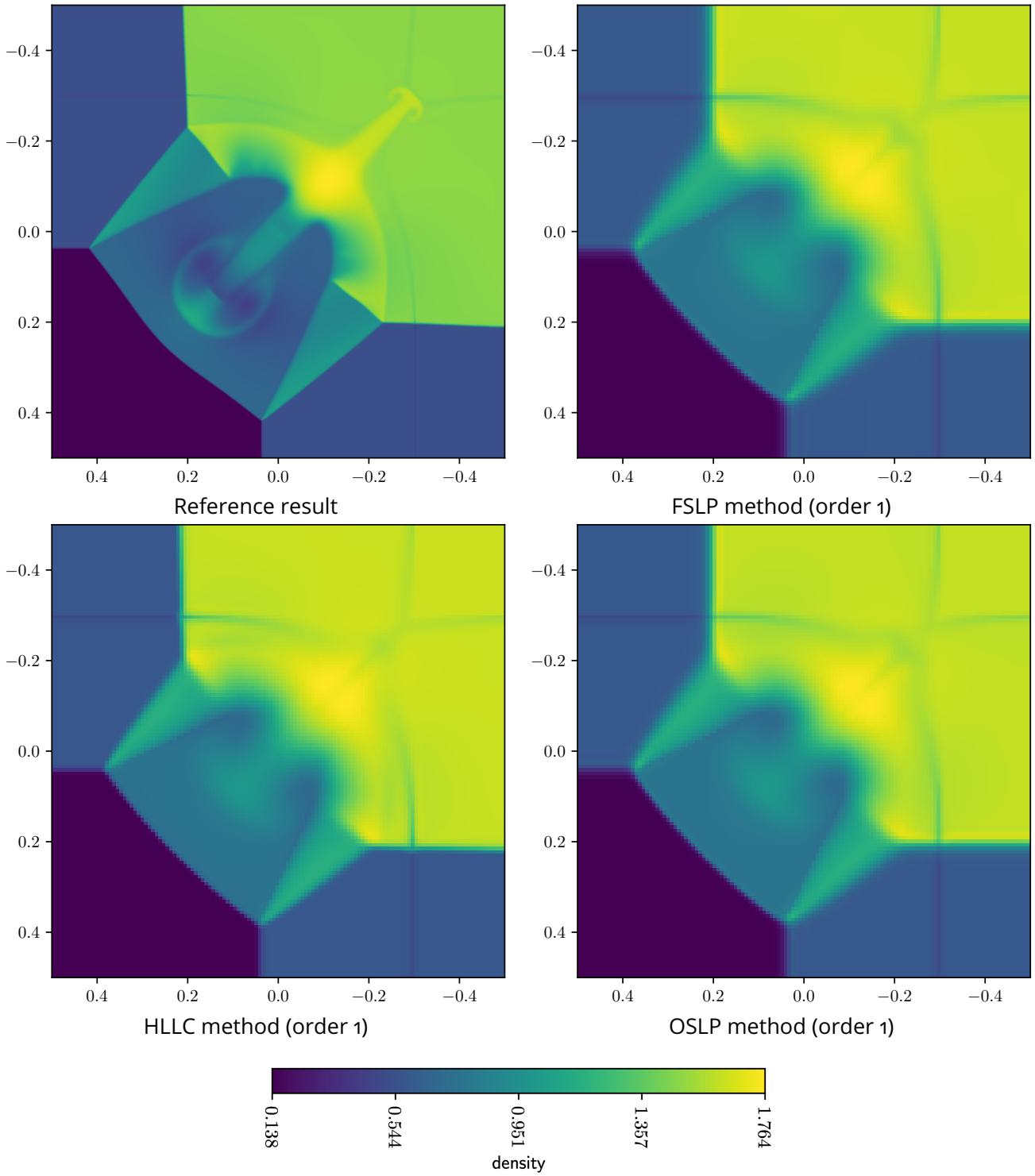


Figure 1.8 – 2D Riemann problem. Mapping of the density number as $t = 0.8s$. The reference simulation is obtained with a second-order HLLC method on a 384×384-cell mesh. The other simulations are performed on a 128×128-cell grid with the first-order FSLP method (top right), the HLLC first-order method (bottom left), and the first-order OSLP method (bottom right).

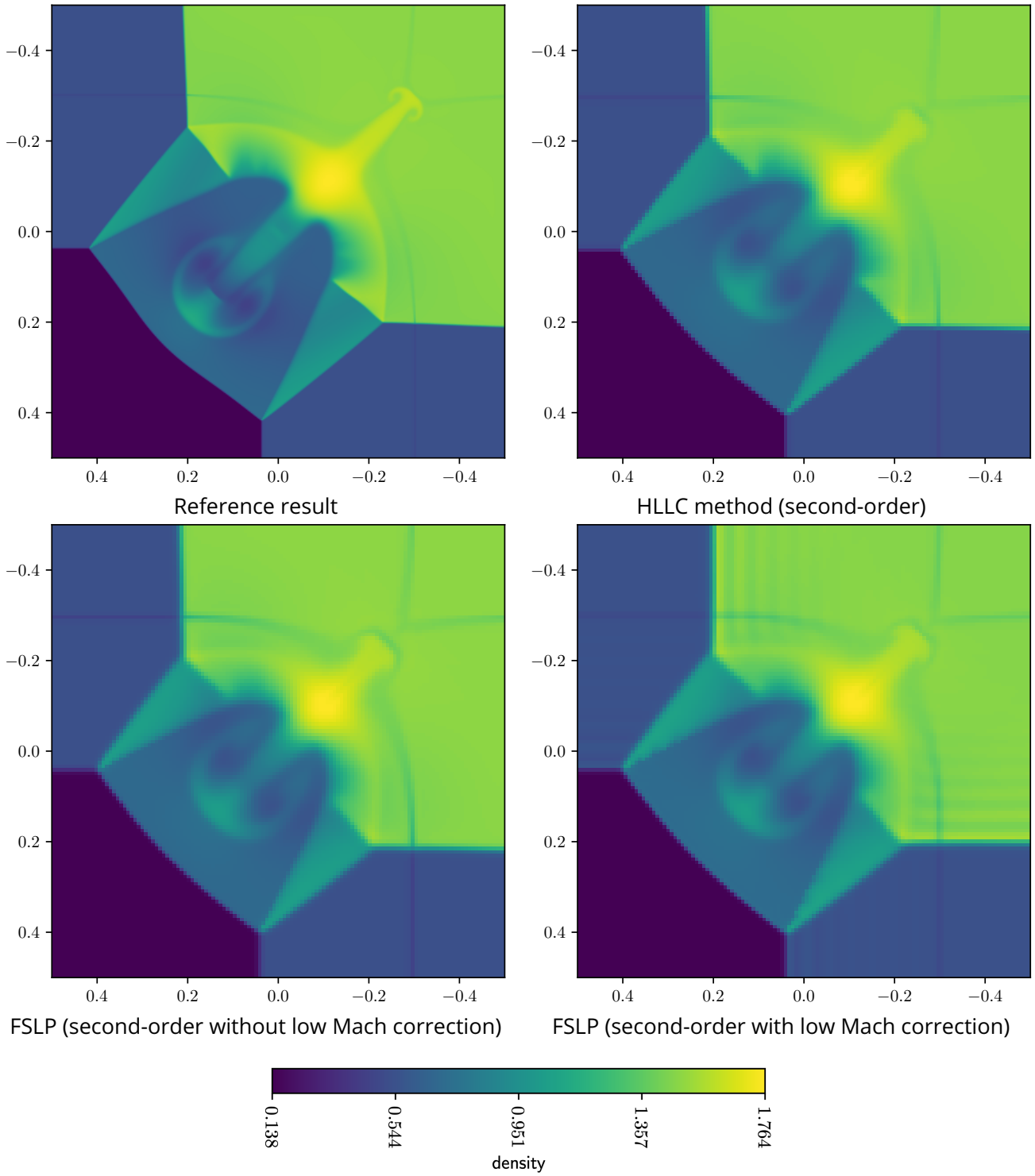


Figure 1.9 – 2D Riemann problem. Mapping of the density number as $t = 0.8s$. The reference simulation is obtained with a second-order HLLC method on a 384×384-cell mesh. The other simulations are performed on a 128×128-cell grid with the second-order HLLC method (top right), the FSLP second-order method without low Mach correction (bottom left), the second-order FSLP method with low Mach correction (bottom right).

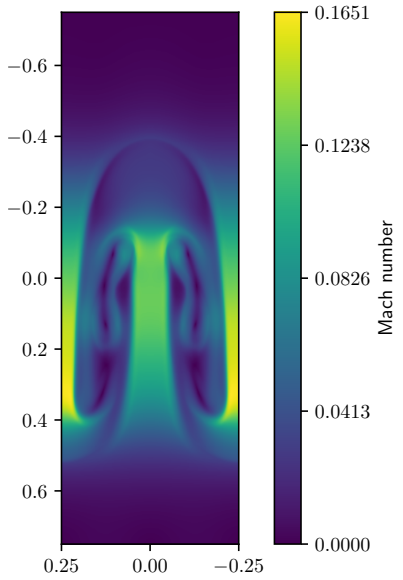


Figure 1.10 – Rayleigh-Taylor instability : mapping of the Mach number profile for reference solution obtained with a second-order HLLC method on a 200×600-cell mesh at $t = 12.4s$.

order method has been successfully investigated in the literature [Castro et al. 2017; Morales de Luna et al. 2020; Del Grosso et Chalons 2021]. Adapting these techniques to the FSLP scheme is beyond the scope of this chapter.

1.8.7 . Rayleigh-Taylor instability

We now consider the Rayleigh-Taylor test performed in [Padioleau et al. 2019] : the computational domain is $[-1/4, 1/4] \times [-3/4, 3/4]$ and the fluid is a perfect gas with $\gamma = 5/3$. At $t = 0$ a dense layer of fluid lies on top of a lighter layer so that the configuration is unstable. The gravity acceleration is $g = -0.1$ thus $\phi(x, y) = -0.1 \times y$. The initial conditions are given by

$$\rho(x, y, t = 0) = \begin{cases} 1 & \text{for } y < 0, \\ 2 & \text{for } y \geq 0, \end{cases} \tag{1.84a}$$

$$p(x, y, t = 0) = -\rho\phi, \tag{1.84b}$$

$$(u, v)(x, y, t = 0) = \left(0, \frac{C}{4}\left(1 + \cos\left(4\pi x\right)\right)\left(1 + \cos\left(3\pi y\right)\right)\right). \tag{1.84c}$$

The initial velocity (1.84c) imposes a single-mode perturbation of magnitude $C = 0.01$ that will break the hydrostatic equilibrium.

This test allows measuring and comparing the effect of the numerical diffusion of each method as it tends to limit the development of high-frequency modes in the instability. Figure 1.11 and 1.10 respectively show the density and Mach number of a reference second-order HLLC simulation obtained with a 200×600-cell mesh. We observe a sharp transition between both fluid layers, and the interface presents lateral arms with secondary rolls.

Figure 1.11 shows simulations ran with both the FSLP solver and the HLLC solver on a coarse 50×150-cell mesh obtained with first and second-order methods. The HLLC method presents an important amount of numerical diffusion : it only shows a single mode growth, and no lateral arm is created. On the other hand, The FSLP method

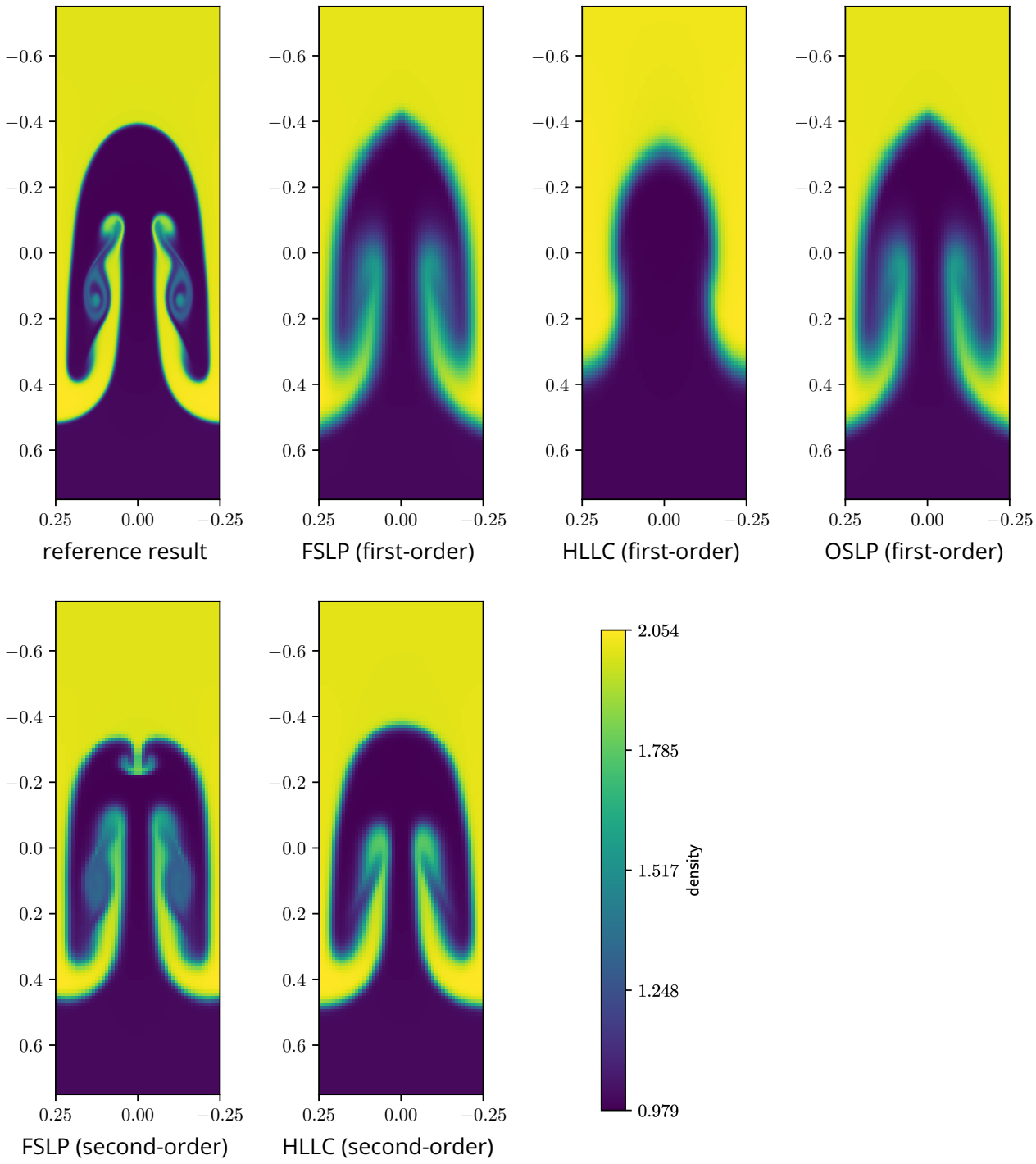


Figure 1.11 – Rayleigh-Taylor instability : mapping of the density profile for : the reference results obtained with the second-order HLLC method on a 200×600-cell grid, the FSLP (first and second-order), the HLLC (first and second-order), and the OSLP methods on a 50×150-cell mesh at $t = 12.4s$.

with low Mach correction can produce the arms that appear on the reference HLLC simulation. It shows that our new method can better capture high-frequency flow features with much lower resolution than the classic HLLC solver, similar to OSLP. This is due to the low Mach nature of this test : as displayed in figure 1.10 one can indeed see that $Ma \in [0, 0.165]$. Therefore the low Mach correction at play in the FSLP solver has an important effect on the result. Note, however how this correction does not fix the important amount of numerical diffusion that appears at the interface between both layers with the FSLP solver. At second-order, the HLLC solution shown in figure 1.11 does present lateral arms, similar to the first-order FSLP method. The second-order FSLP method presents many secondary rolls both on the front of the main mode and on the lateral arms. This agreement with the reference solution displayed in figure 1.11 shows the higher accuracy of the second-order FSLP method. Finally, let us mention that the results obtained with the OSLP in figure 1.11 resemble the first-order FSLP simulation of figure 1.11.

1.8.8 . The stationary vortex in a gravitational field

The stationary vortex in a gravity field test [Thomann et al. 2020] is a modified version of the Gresho vortex [Gresho et Chan 1990] where a gravitational field and a background hydrostatic equilibrium state are added. It allows testing the low Mach properties of numerical methods. We consider the sub-case of the setup proposed in [Thomann et al. 2020] with $\mathrm{Fr} = \mathrm{Ma}$, $RT = 1/\mathrm{Ma}$, and an adiabatic index $\gamma = 5/3$. We consider the domain $[0, 1]^2$ and define the radius from the center $r = (x - 0.5)^2 + (y - 0.5)^2$. The potential and the initial conditions are given by :

$$\Phi(r) = \begin{cases} 12.5r^2 & \text{if} \quad r \leq 0.2 \\ 0.5 - \ln(0.2) + \ln(r) & \text{if} \quad 0.2 < r \leq 0.4 \\ \ln(2) - 0.5\frac{r_c}{r_c - 0.4} + 2.5\frac{r_c}{r_c - 0.4}r - 1.25\frac{1}{r_c - 0.4}r^2 & \text{if} \quad 0.4 < r \leq r_c \\ \ln(2) - 0.5\frac{r_c}{r_c - 0.4} + 1.25\frac{r_c^2}{r_c - 0.4} & \text{if} \quad r > r_c, \end{cases}$$

with $r_c = 0.5$. The density is given by :

$$\rho = \exp\left(-Ma^2\Phi\right), \tag{1.85}$$

The radial velocity is null, and the tangential velocity is given by

$$u_\theta(r) = \frac{1}{u_r} \begin{cases} 5r & \text{if} \quad r \leq 0.2 \\ 2 - 5r & \text{if} \quad 0.2 < r \leq 0.4 \\ 0 & \text{if} \quad r > 0.4 \end{cases} \tag{1.86}$$

The pressure is $p = \rho/Ma^2 + p_2$ with :

$$p_2(r) = \frac{1}{u_r^2} \begin{cases} p_{21}(r) & \text{if} \quad r \leq 0.2 \\ p_{21}(0.2) + p_{22}(r) & \text{if} \quad 0.2 < r \leq 0.4 \\ p_{21}(0.2) + p_{22}(0.4) & \text{if} \quad r > 0.4 \end{cases} \tag{1.87}$$

where $u_r = 0.4\pi$ and

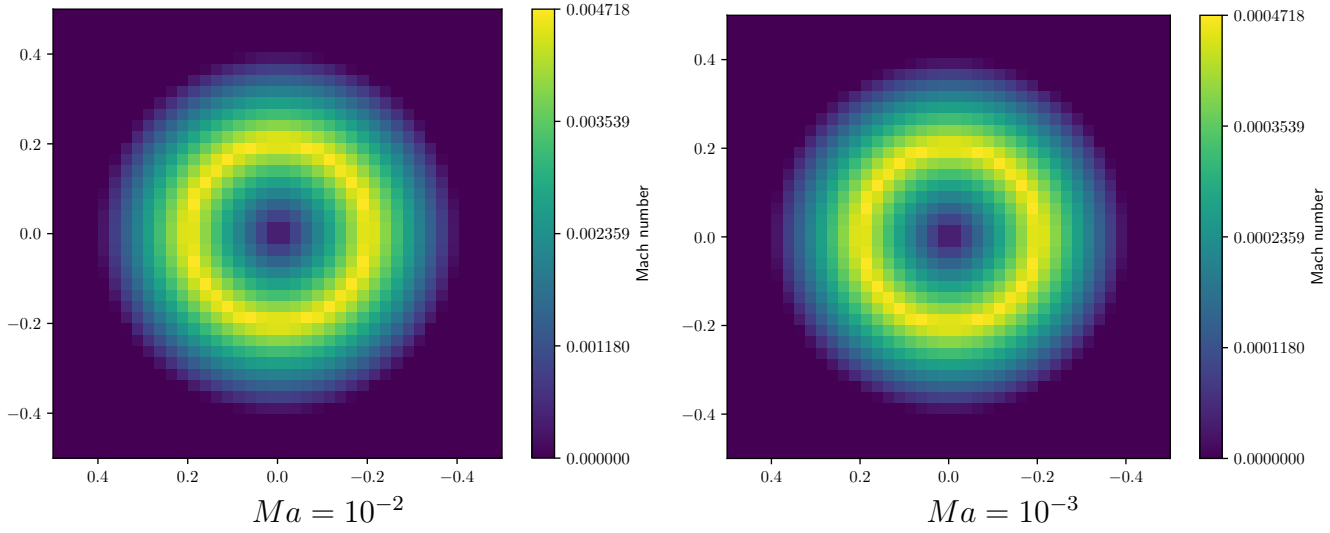


Figure 1.12 – Comparison of the initial Mach number distribution for the Gresho vortex test case with $Ma = 10^{-2}$ and $Ma = 10^{-3}$ obtained with the FSLP method with resolutions 40^2.

$$
\begin{aligned}
p_{21}(r) =& \left(1 - \exp\left(-12.5 Ma^2 r^2\right)\right) \\
p_{22}(r) =& \frac{1}{\left(1 - Ma^2\right)\left(1 - 0.5 Ma^2\right)} \exp\left((-0.5 + \ln(0.2)) Ma^2\right) \\
& \Big(r^{-Ma^2}\left(Ma^4(r(10 - 12.5r) - 2) - 4Ma^4(\gamma - 1)^2 + Ma^2(r(12.5r - 20) + 6)\right) \\
& + \exp\left(-\ln(0.2) Ma^2\right)\left(4 - 2.5 Ma^2 + 0.5 Ma^4\right)\Big) .
\end{aligned}
\tag{1.88}
$$

The initial Mach number distribution is shown for two configurations corresponding to $\mathrm{Ma} = 10^{-2}$ and $\mathrm{Ma} = 10^{-3}$ in figure 1.12. We let the vortex evolve until $t = 1s$ which corresponds to a full revolution and display the final Mach number distribution with different resolutions in figures 1.4, 1.14. We also give the final to initial kinetic energy ratio in table 1.4. It is clear from the figures and the table that the numerical diffusion is indeed roughly independent of the Mach regime.

Table 1.4 – Vortex in a gravitational potential test case : evaluation of the ratio kinetic energies e_{kin}/e^0_{kin} at $t = 1s$ in the computational domain for different values of the Mach number Ma.

	$Ma = 10^{-2}$	$Ma = 10^{-3}$
40^2	0.5723	0.5727
80^2	0.7261	0.7258
160^2	0.8386	0.8388

1.8.9 . Performance comparison : OSLP vs. FSLP

In this section, we compare the performances of both OSLP and FSLP methods. The tests were run on a single Nvidia K80 GPU on a 3D $(512 \times 384 \times 256)$ grid to load the chip's memory fully. As discussed in section 1.5.2,

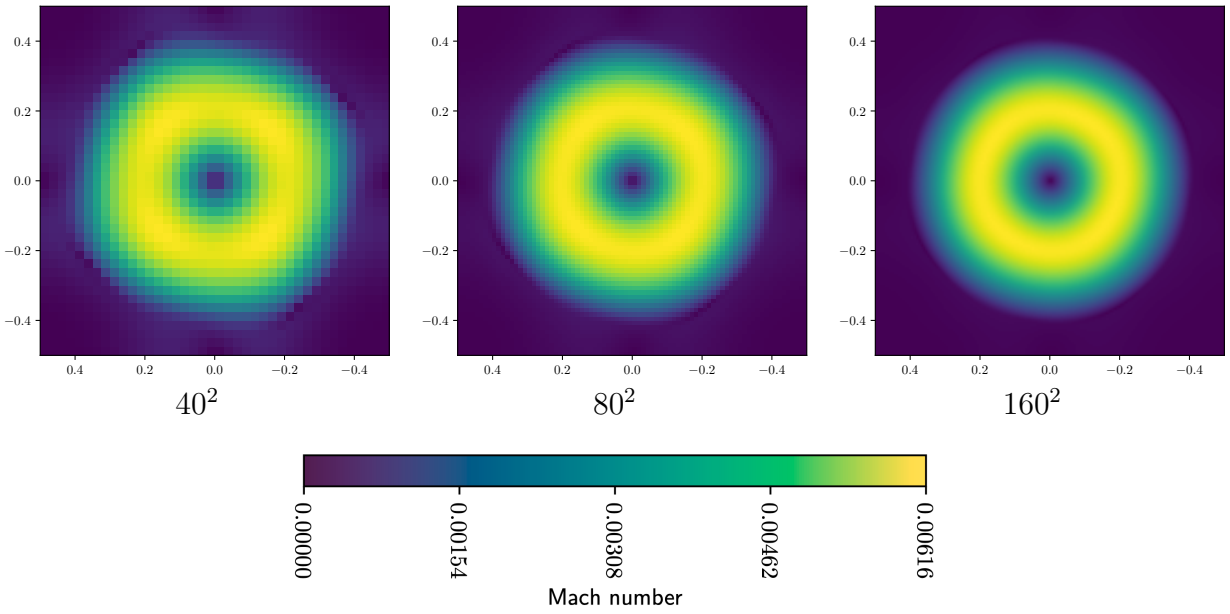


Figure 1.13 – Comparison of the Mach number distribution for the Gresho vortex test case with $Ma = 10^{-2}$ obtained with the FSLP method with resolutions $40^2, 80^2\ 160^2$ at $t = 1s$.

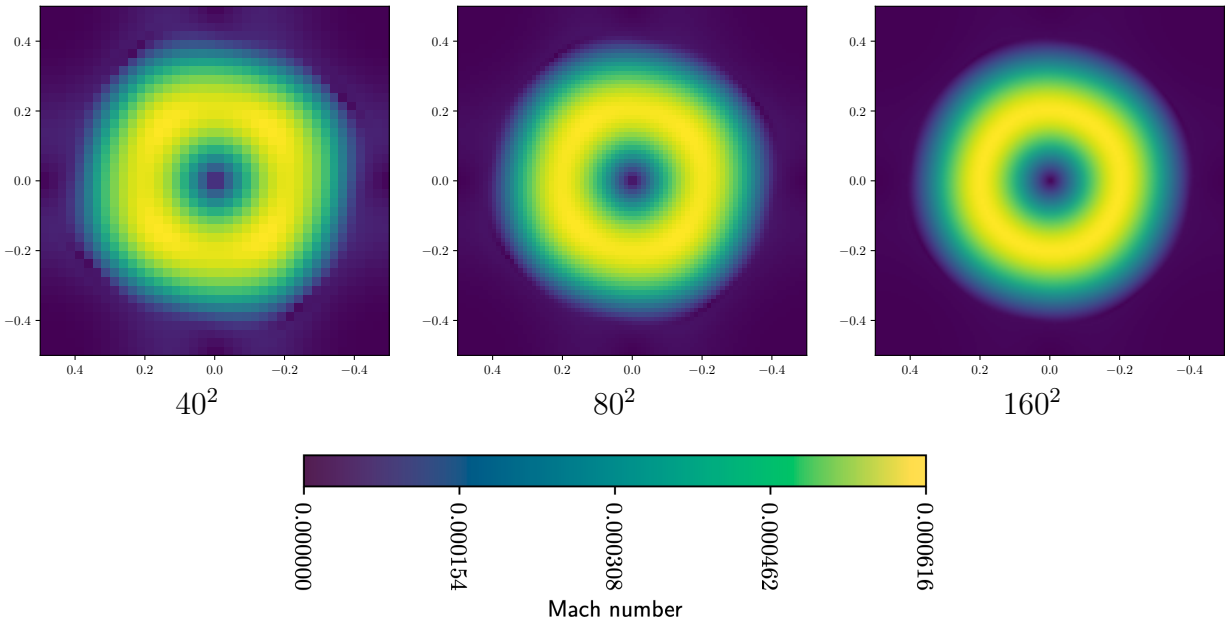


Figure 1.14 – Comparison of the Mach number distribution for the Gresho vortex test case with $Ma = 10^{-3}$ obtained with the FSLP method with resolutions $40^2, 80^2\ 160^2$ at $t = 1s$.

the relative performances of both methods may vary as a function of the Mach number. Indeed, the time step for the FSLP method follows $\Delta t \simeq \Delta x/(v+c)$ while the OSLP time step follows $\Delta t = \Delta x/\max(v,c)$. If $v \simeq c$, the OSLP time step is about two times larger than the FSLP method. However, if $c \gg v$ (low Mach regime), both time step coincides. On the other hand, we can expect that a single step of the FSLP method should be faster than a single step of the OSLP method, as it involves only one kernel instead of two. To illustrate this behavior, we document two test cases :

1. a 3D sod shock tube, to illustrate the $Ma \simeq 1$ behavior,
2. a 3D gresho vortex, to illustrate the $Ma \ll 1$ behavior.

Let us investigate the time required for both methods to reach a given physical time in both Mach regimes. Table 1.5 displays performance results. First, we note that the OSLP method requires 50% more memory than the FSLP method, as it needs to store the intermediate acoustic states on top of the two arrays storing the solution. This allows the FSLP to simulate on a finer grid than the OSLP method given a fixed amount of memory allocated for the computation. We also note that one step of the OSLP method requires about 30% more time than the FSLP method, as it requires two kernels to be applied successively. Since the Gresho test case is a low Mach test case, the time step sizes of the OSLP and FSLP methods coincide. As a result, the FSLP method is 30% faster than the FSLP method. On the other hand, for the Sod problem, the FSLP method requires 117 steps to reach the end time, while the OSLP method only needs 67 steps. As a result, reaching the final time with the FSLP method is 27% longer than with the OSLP method, despite the FSLP steps being faster. These results give a good idea of the relative efficiency of both methods, but they must be mitigated as they are heavily dependent on the implementation and the architecture used. Also, since the FSLP method opens the possiblity for a simple 2nd order extension, we believe it is still of interest even in the $v \simeq c$ regime. Finally, an implicit-explicit version of the FSLP method would likely be competitve with OSLP, as the CFL conditions would coincide. We plan to explore this option in our future work.

Table 1.5 – Performance comparison between OSLP and FSLP

Problem	Method	Steps	Step duration (s)	Total Time (s)	Memory req. (MiB)
Gresho	FSLP	267	0.27 (1.0)	78.2 (1.0)	$7.216\ 10^3$ (1.0)
Gresho	OSLP	267	0.35 (1.31)	102.2 (1.3)	$1.08\ 10^4$ (1.5)
Sod	FSLP	117(1.0)	0.27 (1.0)	34.9 (1.0)	$7.216\ 10^3$ (1.0)
Sod	OSLP	67 (0.57)	0.35 (1.3)	25.48 (0.73)	$1.08\ 10^4$ (1.5)

Reproducing the numerical experiments and figures

All the simulations shown in this chapter were performed with the open source code ARK^2-MHD, which can be found at `https://gitlab.erc-atmo.eu/remi.bourgeois/ark-2-mhd/-/tree/test_case_unsplit_paper_%232`. All parameter files and plotting scripts can be found in the folder `/test_case_unsplit_paper`.

1.9 . Conclusion

We have presented the recasting of an operator splitting Lagrange-Projection solver for gas dynamicss into a corresponding flux-splitting finite volume method. This FSLP method is obtained thanks to a simple modification :

it only differs from the OSLP method in the states used to compute the transport step. The method relies on a flux evaluation that separates pressure-related terms from the advection terms in the spirit of [Darracq et al. 1998; Liou et Steffen 1993; Deshpande et al. 1994; Toro et Vázquez-Cendón 2012; Borah et al. 2016]. Two different interpretations of this flux-splitting scheme were proposed to understand better and analyze the resulting method. First, we showed that the FSLP discretization could be written as a convex combination of two updated states resulting from approximating two subsystems that respectively account for pressure and advection effects. This approach allowed us to derive the stability properties of the proposed algorithm. Second, we discussed the interpretation of the FSLP method as the result of the discretization of a larger relaxation system that accounts separately for pressure and advection terms within a single step. We showed that the FSLP method is more computationally efficient than the OSLP method in the low Mach regime. As a flux-based solver, the resulting FSLP method was straightforwardly extended to multiple dimensions of space and to a high order of accuracy thanks to a standard MUSCL method.

The initial OSLP solver has several interesting numerical advantages : a well-balanced treatment of the source term and a low Mach fix that provides a uniform truncation error with respect to the Mach number. Both properties were preserved through the recasting process. The robustness and accuracy of our new flux-splitting method were tested against a set of benchmark problems, including one and two-dimensional problems, high and low Mach flows with first and second-order discretizations. The results further confirm the numerical stability of our approach.

In the next chapter, we present the extension of this approach to the multidimensional MHD system for which we perform a similar splitting between advection effects and pressure/magnetic effects, resulting in a remarquably stable numerical scheme. In chapter 3, we will used the resulting numerical scheme for MHD to perform MHD convection simulations. In the future, we plan to perform a similar recasting by considering an Implicit-Explicit OSLP solver to prevent the severe CFL limitations imposed by the sound velocity in the low Mach regime. The methods can also be extended to several other flow models like two-phase flow models, and the M1 model for radiative transfer.

Appendix

1.A . A few classic convexity properties

We recall hereafter a few classic convexity/concavity properties related to admissible states, entropy, and energy of our flow model that can be found in the literature (see for example [Godlewski et Raviart 2021]). We propose short self-contained proofs of these properties for the sake of completeness.

Lemma 1.A.1. *We have the following properties.*

(a) The function $\Lambda : \boldsymbol{U} = (\rho, \rho u, \rho E) \in [0, +\infty) \times \mathbb{R} \times [0, +\infty) \mapsto \Lambda(\boldsymbol{U}) = (\rho E) - \frac{(\rho u)^2}{2\rho}$ *is concave.*

(b) The set Ω *defined by* (1.2) *is convex.*

(c) The function $\mathcal{U} : (\mathcal{T}, u, E) \mapsto s\left(\mathcal{T}, E - \frac{u^2}{2}\right)$ *is strictly concave.*

(d) The function $\eta : (\rho, \rho\mathcal{T}, \rho u, \rho E) \mapsto \rho s\left(\frac{\rho\mathcal{T}}{\rho}, \frac{(\rho E)}{\rho} - \frac{(\rho u)^2}{2\rho^2}\right)$ *is strictly concave.*

Démonstration. Let $\theta_1 = 1 - \theta_2 \in [0, 1]$ and $\boldsymbol{U}_k \in [0, +\infty) \times \mathbb{R} \times [0, +\infty)$ for $k = 1, 2$, we have

$$\Lambda\left(\sum_{k=1,2} \theta_k \boldsymbol{U}_k\right) - \sum_{k=1,2} \theta_k \Lambda(\boldsymbol{U}_k) = \sum_{k=1,2} \theta_k (\rho E)_k + \frac{\theta_1\theta_2}{\sum_{k=1,2} \theta_k \rho_k} \left[(\rho u)_1 \sqrt{\frac{\rho_2}{\rho_1}} - (\rho u)_2 \sqrt{\frac{\rho_1}{\rho_2}}\right]^2 \geq 0, \quad (1.89)$$

which proves (a). For (b), consider again $\theta_1 = 1 - \theta_2 \in [0, 1]$ and $\boldsymbol{U}_k \in \Omega$, $k = 1, 2$. If we note $\boldsymbol{U} = \sum_{k=1,2} \theta_k \boldsymbol{U}_k$, then $\boldsymbol{U} \in \Omega$. Indeed, we have that $\sum_{k=1,2} \theta_k \rho_k \geq 0$, and as $\rho e = \Lambda(\boldsymbol{U}) \geq \sum_{k=1,2} \theta_k \Lambda(\boldsymbol{U}_k) = \sum_{k=1,2} \theta_k \rho_k e_k \geq 0$, where $e_k = E_k - (u_k^2)/2$. This implies that $e \geq 0$.

For (c) : The function $K : (u, E) \mapsto E - u^2/2$ is strictly concave and we have $\mathcal{U}(\mathcal{T}, u, E) = s(\mathcal{T}, K(u, E))$. Consider $\lambda \in [0, 1]$ and let us note $\lambda = \lambda_1$ and $\lambda_2 = 1 - \lambda$. We have that

$$\mathcal{U}\left(\sum_{k=1,2} \lambda_k \mathcal{T}_k, \sum_{k=1,2} \lambda_k u_k, \sum_{k=1,2} \lambda_k E_k\right) - \sum_{k=1,2} \lambda_k \mathcal{U}(\mathcal{T}_k, u_k, E_k) \quad (1.90)$$

$$= s\left(\sum_{k=1,2} \mathcal{T}_k, K\left(\sum_{k=1,2} \lambda_k u_k, \sum_{k=1,2} \lambda_k E_k\right)\right) - \sum_{k=1,2} \lambda_k s(\mathcal{T}_k, K(u_k, E_k)) \quad (1.91)$$

$$= s\left(\sum_{k=1,2} \mathcal{T}_k, K\left(\sum_{k=1,2} \lambda_k u_k, \sum_{k=1,2} \lambda_k E_k\right)\right) - s\left(\sum_{k=1,2} \mathcal{T}_k, \sum_{k=1,2} \lambda_k K(u_k, E_k)\right)$$

$$+ s\left(\sum_{k=1,2} \mathcal{T}_k, \sum_{k=1,2} \lambda_k K(u_k, E_k)\right) - \sum_{k=1,2} \lambda_k s(\mathcal{T}_k, K(u_k, E_k)). \quad (1.92)$$

As K is concave, we get that

$$K\left(\sum_{k=1,2}\lambda_k u_k, \sum_{k=1,2}\lambda_k E_k\right) \geq \sum_{k=1,2}\lambda_k K(u_k, E_k). \tag{1.93}$$

By (1.3) we know that $e' \mapsto s^{\text{EOS}}(\overline{\mathcal{T}}, e')$ is increasing so that

$$s\left(\sum_{k=1,2}\mathcal{T}_k, K\left(\sum_{k=1,2}\lambda_k u_k, \sum_{k=1,2}\lambda_k E_k\right)\right) - s\left(\sum_{k=1,2}\mathcal{T}_k, \sum_{k=1,2}\lambda_k K(u_k, E_k)\right) \geq 0. \tag{1.94}$$

We also know that s is concave therefore

$$s\left(\sum_{k=1,2}\mathcal{T}_k, \sum_{k=1,2}\lambda_k K(u_k, E_k)\right) - \sum_{k=1,2}\lambda_k s(\mathcal{T}_k, K(u_k, E_k)) \geq 0. \tag{1.95}$$

By replacing (1.94) and (1.95) into (1.92) we obtain that

$$\mathcal{U}\left(\sum_{k=1,2}\lambda_k\mathcal{T}_k, \sum_{k=1,2}\lambda_k u_k, \sum_{k=1,2}\lambda_k E_k\right) \geq \sum_{k=1,2}\lambda_k\mathcal{U}(\mathcal{T}_k, u_k, E_k). \tag{1.96}$$

for (d) : If we note again $\boldsymbol{U} = (\rho, \rho u, \rho E)$, by (1.6), we have

$$\eta(\rho, \rho\mathcal{T}, \rho u, \rho E) = \rho s\left(\frac{\rho\mathcal{T}}{\rho}, \frac{(\rho E)}{\rho} - \frac{(\rho u)^2}{2\rho^2}\right) = S\left(\rho, \rho\mathcal{T}, \rho E - \frac{(\rho u)^2}{2\rho}\right) = S(\rho, \rho\mathcal{T}, \Lambda(\boldsymbol{U})). \tag{1.97}$$

Now we consider $\boldsymbol{U}_k = (\rho_k, \rho_k u_k, \rho_k E_k) \in \Omega$ and $\mathcal{T}_k \geq 0$, $k = 1, 2$, we have

$$\begin{aligned}
&\eta\left(\sum_{k=1,2}\theta_k\rho_k, \sum_{k=1,2}\theta_k\rho_k\mathcal{T}_k, \sum_{k=1,2}\theta_k\rho_k u_k, \sum_{k=1,2}\theta_k\rho_k E_k\right) - \sum_{k=1,2}\theta_k\eta(\rho_k, \rho_k\mathcal{T}_k, \rho_k u_k, \rho_k E_k) \\
&\quad = S\left(\sum_{k=1,2}\theta_k\rho_k, \sum_{k=1,2}\theta_k\rho_k\mathcal{T}_k, \Lambda\left(\sum_{k=1,2}\theta_k\boldsymbol{U}_k\right)\right) - \sum_{k=1,2}\theta_k S(\rho_k, \rho_k\mathcal{T}_k, \Lambda(\boldsymbol{U}_k)) \\
&\quad = S\left(\sum_{k=1,2}\theta_k\rho_k, \sum_{k=1,2}\theta_k\rho_k\mathcal{T}_k, \Lambda\left(\sum_{k=1,2}\theta_k\boldsymbol{U}_k\right)\right) - S\left(\sum_{k=1,2}\theta_k\rho_k, \sum_{k=1,2}\theta_k\rho_k\mathcal{T}_k, \sum_{k=1,2}\theta_k\Lambda(\boldsymbol{U}_k)\right) \\
&\qquad + S\left(\sum_{k=1,2}\theta_k\rho_k, \sum_{k=1,2}\theta_k\rho_k\mathcal{T}_k, \sum_{k=1,2}\theta_k\Lambda(\boldsymbol{U}_k)\right) - \sum_{k=1,2}\theta_k S(\rho_k, \rho_k\mathcal{T}_k, \Lambda(\boldsymbol{U}_k))
\end{aligned} \tag{1.98}$$

As Λ is concave, we have $\Lambda\left(\sum_{k=1,2}\theta_k\boldsymbol{U}_k\right) \geq \sum_{k=1,2}\theta_k\Lambda(\boldsymbol{U}_k)$ and as $\mathscr{E}' \mapsto S(\overline{\rho}, \overline{\mathscr{V}}, \mathscr{E}')$ is increasing, we have

$$S\left(\sum_{k=1,2}\theta_k\rho_k, \sum_{k=1,2}\theta_k\rho_k\mathcal{T}_k, \Lambda\left(\sum_{k=1,2}\theta_k\boldsymbol{U}_k\right)\right) - S\left(\sum_{k=1,2}\theta_k\rho_k, \sum_{k=1,2}\theta_k\rho_k\mathcal{T}_k, \sum_{k=1,2}\theta_k\Lambda(\boldsymbol{U}_k)\right) \geq 0. \tag{1.99}$$

Using the fact that S is concave, we also get

$$S\left(\sum_{k=1,2}\theta_k\rho_k,\ \sum_{k=1,2}\theta_k\rho_k\mathcal{T}_k,\ \sum_{k=1,2}\theta_k\Lambda(\boldsymbol{U}_k)\right)-\sum_{k=1,2}\theta_k S(\rho_k,\rho_k\mathcal{T}_k,\Lambda(\boldsymbol{U}_k))\geq 0. \tag{1.100}$$

Injecting (1.99) and (1.100) into (1.98) provides

$$\eta\left(\sum_{k=1,2}\theta_k\rho_k,\ \sum_{k=1,2}\theta_k\rho_k\mathcal{T}_k,\ \sum_{k=1,2}\theta_k\rho_k u_k,\ \sum_{k=1,2}\theta_k\rho_k E_k\right)\geq\sum_{k=1,2}\theta_k\eta(\rho_k,\rho_k\mathcal{T}_k,\rho_k u_k,\rho_k E_k). \tag{1.101}$$

□

1.B . Approximate Riemann solver for the pressure subsystem

In this section, we present the derivation of an approximate Riemann solver for the pressure subsystem (1.41), following the lines of [Chalons et al. 2016a; Padioleau et al. 2019]. We express (1.41) in the following compact form :

$$\partial_t\boldsymbol{U}+2\partial_x\boldsymbol{P}(\boldsymbol{U})=\boldsymbol{S}(\boldsymbol{U}),\qquad \partial_t\Pi+\partial_x(2a^2u)=0,\qquad \partial_t(\rho\mathcal{T})-2\partial_x u^P=0,\qquad \partial_t\phi=0. \tag{1.102}$$

where $\boldsymbol{P}(\boldsymbol{U})^T=(0,\Pi,\Pi u)$. Let $\Delta x_L>0$, $\Delta x_R>0$, we consider $\bar{x}\in\mathbb{R}$ and the following piecewise initial data

$$\left(\boldsymbol{U},\Pi,\mathcal{T},\phi\right)(x,t=0)=\begin{cases}(\boldsymbol{U}_L,\Pi_L,\mathcal{T}_L,\phi_L) & \text{if } x\leq\bar{x},\\ (\boldsymbol{U}_R,\Pi_R,\mathcal{T}_R,\phi_R) & \text{if } x>\bar{x},\end{cases} \tag{1.103}$$

that verifies the equilibrium relations :

$$(\boldsymbol{U}_k,\Pi_k,\mathcal{T}_k,\phi_k)=\left[(\rho_k,\rho_k u_k,E_k)^T,p^{\text{EOS}}\left(\frac{1}{\rho_k},e_k\right),\frac{1}{\rho_k},\phi_k\right],\quad k=L,R, \tag{1.104}$$

with $\phi_L=\frac{1}{\Delta x_L}\int_{-\Delta x_L}^{0}\phi(\bar{x}+x)\,\mathrm{d}x$ and $\phi_R=\frac{1}{\Delta x_R}\int_{0}^{\Delta x_R}\phi(\bar{x}+x)\,\mathrm{d}x$. We seek a self-similar function $(\boldsymbol{U}_{\text{RP}},\Pi_{\text{RP}},\mathcal{T}_{\text{RP}},\phi_{\text{RP}})$ composed of four constant states separated by three discontinuities as follows :

$$\begin{aligned}&(\boldsymbol{U}_{\text{RP}},\Pi_{\text{RP}},\mathcal{T}_{\text{RP}},\phi_{\text{RP}})\left(\frac{x-\bar{x}}{t};\boldsymbol{U}_L,\Pi_L,\mathcal{T}_L,\phi_L,\boldsymbol{U}_R,\Pi_R,\mathcal{T}_R,\phi_R\right)\\ &=\begin{cases}(\boldsymbol{U}_L,\Pi_L,\mathcal{T}_L,\Phi_L), & \text{if } \frac{x-\bar{x}}{t}\leq-\frac{2a}{\rho_L},\\ (\boldsymbol{U}_L^*,\Pi_L^*,\mathcal{T}_L^*,\Phi_L), & \text{if } -\frac{2a}{\rho_L}<\frac{x-\bar{x}}{t}\leq 0,\\ (\boldsymbol{U}_R^*,\Pi_R^*,\mathcal{T}_R^*,\Phi_R), & \text{if } 0<\frac{x-\bar{x}}{t}\leq\frac{2a}{\rho_R},\\ (\boldsymbol{U}_R,\Pi_R,\mathcal{T}_R,\Phi_R), & \text{if } \frac{2a}{\rho_R}<\frac{x-\bar{x}}{t},\end{cases}\end{aligned} \tag{1.105}$$

where the intermediate states $\boldsymbol{U}_k^*$, Π_k^* and $\mathcal{T}_k^*$ are required to satisfy the four following properties.

1. The approximate Riemann solver should be consistent in the integral sense with the pressure subsystem (1.102) : for Δt such that $\frac{2a}{\min(\rho_L,\rho_R)}\Delta t<\frac{1}{2}\min(\Delta x_L,\Delta x_R)$, we have

$$\begin{bmatrix}2\boldsymbol{P}\left(\boldsymbol{U}_R\right)-2\boldsymbol{P}\left(\boldsymbol{U}_L\right)\\ 2a^2(u_R^*-u_L^*)\\ -(2u_R^*-2u_L^*)\end{bmatrix}=-\frac{2a}{\rho_L}\begin{bmatrix}\boldsymbol{U}_L^*-\boldsymbol{U}_L\\ (\rho\Pi)_L^*-(\rho\Pi)_L\\ (\rho\mathcal{T})_L^*-(\rho\mathcal{T})_L\end{bmatrix}+\frac{2a}{\rho_R}\begin{bmatrix}\boldsymbol{U}_R-\boldsymbol{U}_R^*\\ (\rho\Pi)_R-(\rho\Pi)_R^*\\ (\rho\mathcal{T})_R-(\rho\mathcal{T})_R^*\end{bmatrix}+(\Delta x_L+\Delta x_R)\left\{\boldsymbol{S}\right\}, \tag{1.106}$$

with $\{\boldsymbol{S}\}$ a function that is a consistent approximation of $\boldsymbol{S}$, that is to say :

$$\lim_{\substack{\Phi_L,\Phi_R\to\phi(\bar{x})\\ \Delta x_L,\Delta x_R\to 0\\ (\boldsymbol{U}_R,\Pi_R),(\boldsymbol{U}_L,\Pi_L)\to(\bar{\boldsymbol{U}},p^{\text{EOS}}(\bar{\rho},\bar{e}))}} \{\boldsymbol{S}\} = \boldsymbol{S}(\bar{\boldsymbol{U}},\phi)(x=\bar{x}). \tag{1.107}$$

2. In the case $\phi_L = \phi_R$, it should be degenerate to an approximate Riemann for the homogeneous problem obtained with (1.102) when $\boldsymbol{S} = \boldsymbol{0}$.
3. If $\boldsymbol{U}_L$ and $\boldsymbol{U}_R$ satisfy the following discrete version of the hydrostatic condition (1.8) :

$$\Pi_R - \Pi_L = -\frac{\rho_L+\rho_R}{2}(\phi_R-\phi_L), \qquad u_L = u_R = 0, \tag{1.108}$$

then $(\boldsymbol{U}_L^*,\Pi_L^*) = (\boldsymbol{U}_L,\Pi_L)$ and $(\boldsymbol{U}_R^*,\Pi_R^*) = (\boldsymbol{U}_R,\Pi_R)$.

Let us build the states $(\boldsymbol{U}_R^*,\Pi_R^*)$ and $(\boldsymbol{U}_L^*,\Pi_L^*)$ so that they verify the above properties. We note

$$\Pi_R^* - \Pi_L^* + \mathcal{M} = 0. \tag{1.109}$$

First, we impose that ρ_L^* and ρ_R^* are consistent with the exact solution of (1.102) by setting $\rho_L^* = \rho_L$ and $\rho_R^* = \rho_R$. Then we also require that the Rankine-Hugoniot jump conditions obtained in the case $\boldsymbol{S} = \boldsymbol{0}$ are valid across the waves of velocity $-2a/\rho_L$ and $+2a/\rho_R$

$$\frac{2a}{\rho_L}\begin{bmatrix}\boldsymbol{U}_L^*-\boldsymbol{U}_L\\ (\rho\Pi)_L^*-(\rho\Pi)_L\\ (\rho\mathcal{T})_L^*-(\rho\mathcal{T})_L\end{bmatrix} + \begin{bmatrix}2\boldsymbol{P}(\boldsymbol{U}_L^*)-2\boldsymbol{P}(\boldsymbol{U}_L)\\ 2a^2u_L^*-2a^2u_L\\ -2u_L^*+2u_L\end{bmatrix} = 0 \tag{1.110}$$

$$,-\frac{2a}{\rho_R}\begin{bmatrix}\boldsymbol{U}_R-\boldsymbol{U}_R^*\\ (\rho\Pi)_R-(\rho\Pi)_R^*\\ (\rho\mathcal{T})_R-(\rho\mathcal{T})_R^*\end{bmatrix} + \begin{bmatrix}2\boldsymbol{P}(\boldsymbol{U}_R)-2\boldsymbol{P}(\boldsymbol{U}_R^*)\\ 2a^2u_R-2a^2u_R^*\\ -2u_R+2u_R^*\end{bmatrix} = 0. \tag{1.111}$$

Finally, we postulate that the velocity is continuous across the stationary wave by setting

$$u_L^* = u_R^* = u^*, \tag{1.112}$$

and we also impose that $(\Pi u)_k^* = \Pi_k^* u_k^* = \Pi_k^* u^*$, $k = L, R$. Then, relations (1.106), (1.111), (1.109) yield

$$\rho_L^* = \rho_L, \qquad \rho_R^* = \rho_R, \tag{1.113a}$$

$$E_L^* = E_L - \frac{1}{a}\left((\Pi^*+\frac{\mathcal{M}}{2})u^* - \Pi_L u_L\right), \qquad E_R^* = E_R + \frac{1}{a}\left((\Pi^*-\frac{\mathcal{M}}{2})u^* - \Pi_R u_R\right), \tag{1.113b}$$

$$u^* = u_R^* = u_L^* = \frac{u_R+u_L}{2} - \frac{1}{2a}(\Pi_R-\Pi_L) - \frac{\mathcal{M}}{2a}, \quad \Pi^* = \frac{\Pi_R+\Pi_L}{2} - \frac{a}{2}(u_R-u_L), \tag{1.113c}$$

$$\Pi_L^* = \Pi^* + \frac{\mathcal{M}}{2}, \qquad \Pi_R^* = \Pi^* - \frac{\mathcal{M}}{2}, \tag{1.113d}$$

$$\mathcal{T}_L^* = \frac{1}{\rho_L} + \frac{1}{a}(u^*-u_L), \qquad \mathcal{T}_R^* = \frac{1}{\rho_R} - \frac{1}{a}(u^*-u_R), \tag{1.113e}$$

where the jump $\mathcal{M}$ can be identifed as

$$\mathcal{M} = \frac{\Delta x_L+\Delta x_R}{2}\{\rho\partial_x\phi\}, \qquad \mathcal{M}u^* = \frac{\Delta x_L+\Delta x_R}{2}\{\rho u\partial_x\phi\}. \tag{1.114}$$

At this point, the functions $\{\rho\partial_x\phi\}$ and $\{\rho u\partial_x\phi\}$ are still yet to be specified. Let us consider the constraint 3 : if it is satisfied then for a state that verifies (1.108) the jumps $\mathcal{M}$ and $\mathcal{M}u^*$ necessarily take the value $\mathcal{M} = -(\Pi_R - \Pi_L)$ and $\mathcal{M}u^* = 0$. A simple choice that fulfills this requirement is

$$\{\rho\partial_x\phi\} = (\rho_L + \rho_R)\frac{\phi_R - \phi_L}{\Delta x_L + \Delta x_R}, \qquad \{\rho\partial_x\phi\} = (\rho_L + \rho_R)u^*\frac{\phi_R - \phi_L}{\Delta x_L + \Delta x_R}. \tag{1.115a}$$

Relations (1.113) and (1.115a) give a complete definition of the approximate Riemann solver (1.105). This solver a definition for the conservative numerical flux $\boldsymbol{P}_\Delta(\boldsymbol{U}_L, \Pi_L, \phi_L, \boldsymbol{U}_R, \Pi_R, \phi_R)$ and a source term discretization (located at the interface) $\boldsymbol{S}_\Delta(\boldsymbol{U}_L, \Pi_L, \phi_L, \boldsymbol{U}_R, \Pi_R, \phi_R)$ thanks to the consistency in the integral sense. We get

$$\boldsymbol{P}_\Delta(\boldsymbol{U}_L, \Pi_L, \phi_L, \boldsymbol{U}_R, \Pi_R, \phi_R) = \frac{\boldsymbol{P}(\boldsymbol{U}_R, \Pi_R) + \boldsymbol{P}(\boldsymbol{U}_L, \Pi_L)}{2} - \frac{a}{2\rho_L}(\boldsymbol{U}_L^* - \boldsymbol{U}_L) - \frac{a}{2\rho_R}(\boldsymbol{U}_R - \boldsymbol{U}_R^*), \tag{1.116a}$$

$$\boldsymbol{S}_\Delta(\boldsymbol{U}_L, \Pi_L, \phi_L, \boldsymbol{U}_R, \Pi_R, \phi_R) = [0, -\{\rho\partial_x\phi\}, -\{\rho u\partial_x\phi\}]^T, \tag{1.116b}$$

so that for two neighbouring states $(\boldsymbol{U}_j^n, \Pi_j^n, \phi_j)$ and $(\boldsymbol{U}_{j+1}^n, \Pi_{j+1}^n, \phi_{j+1})$ across the cell interface $j + 1/2$ that separates the cell j and the cell $j + 1$, the numerical conservative flux $(0, \Pi_{j+1/2}^*, \Pi_{j+1/2}^* u_{j+1/2}^*)$ is defined by

$$(0, \Pi_{j+1/2}^*, \Pi_{j+1/2}^* u_{j+1/2}^*) = \boldsymbol{P}_\Delta(\boldsymbol{U}_j^n, \Pi_j^n, \phi_j, \boldsymbol{U}_{j+1}^n, \Pi_{j+1}^n, \phi_{j+1}), \tag{1.117}$$

and the discrete souce term $\boldsymbol{S}_j$ within the cell j is given by

$$\boldsymbol{S}_j = \frac{\Delta x_{j+1/2}}{2\Delta x_j}\boldsymbol{S}_{j+1/2} + \frac{\Delta x_{j-1/2}}{2\Delta x_j}\boldsymbol{S}_{j-1/2}, \qquad \boldsymbol{S}_{j+1/2} = \boldsymbol{S}_\Delta(\boldsymbol{U}_j^n, \Pi_j^n, \phi_j, \boldsymbol{U}_{j+1}^n, \Pi_{j+1}^n, \phi_{j+1}). \tag{1.118}$$

Let us now give some properties of the approximate Riemann solver. Let us note $e_k^* = E_k^* - (u_k^*)^2/2$, the following lemma is a direct consequence of (1.111) that exhibits a reminiscent property associated with the Riemann invariants associated of the system (1.102) when $\boldsymbol{S} = \boldsymbol{0}$.

Lemma 1.B.1.

$$e_k^* - \frac{(\Pi_k^*)^2}{2a^2} = e_k - \frac{(\Pi_k)^2}{2a^2}, \qquad \mathcal{T}_k^* + \frac{\Pi_k^*}{a} = \mathcal{T}_k + \frac{\Pi_k}{a}, \qquad k = L, R. \tag{1.119}$$

The following positivity result is a direct consequence of (1.113e).

Proposition 1.B.1.

1. *If a is chosen large enough then $\mathcal{T}_L^* > 0$ and $\mathcal{T}_R^* > 0$.*
2. *$\mathcal{T}_L^* > 0$ and $\mathcal{T}_R^* > 0$ is equivalent to $u_L - a\mathcal{T}_L = u_L - a/\rho_L < u^* < u_R + a\mathcal{T}_R = u_R + a/\rho_R$.*

Following the lines of [Chalons et al. 2016a], we first prove two preliminary stability-related results. The differences from Lemma 1 of [Chalons et al. 2016a] is that the Riemann states we are dealing with here depend on the $\mathcal{M}$ terms and that the specific volume we use is $\mathcal{T}$ instead of $1/\rho$ (that are different in the sub-system framework). However, the proof turns out to be almost identical.

Proposition 1.B.2. *Consider the intermediate states defined by* (1.113).
and noting $s_k = s^{EOS}(\mathcal{T}_k, s_k)$*, we have*

$$e_k^* - e^{EOS}\left(\mathcal{T}_k^*, s_k\right) - \frac{\left(p^{EOS}\left(\mathcal{T}_k^*, s_k\right) - \Pi_k^*\right)^2}{2a^2} \geq 0, \tag{1.120}$$

with $e_k^* = E_k^* - \frac{u_k^{*2}}{2}$.

Démonstration. We only describe the case $k = R$. Consider the function :

$$\begin{aligned}\chi(\mathcal{T}) = e^{\text{EOS}}\left(\mathcal{T}, s_R\right) - \frac{p^{\text{EOS}}\left(\mathcal{T}, s_R\right)^2}{2a^2} - e^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right) + \frac{p^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right)^2}{2a^2} \\ + p^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right)\left(\mathcal{T} + \frac{p^{\text{EOS}}\left(\mathcal{T}, s_R\right)}{a^2} - \mathcal{T}_R^* - \frac{p^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right)}{a^2}\right).\end{aligned} \tag{1.121}$$

One can check that $\chi'(\mathcal{T}) = \left(p^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right) - p^{\text{EOS}}\left(\mathcal{T}, s_R\right)\right)\left(1 - \rho^2 c^2\left(\mathcal{T}, s_R\right)/a^2\right)$. We have $\partial_{\mathcal{T}} p < 0$ from 1.3, we also assume that a is large enough. We have two different cases :

$$\begin{aligned}\mathcal{T}_R^* < \mathcal{T} < \mathcal{T}_R &\implies \chi'(\mathcal{T}) > 0 \quad \implies \chi(\mathcal{T}_R^*) < \chi(\mathcal{T}) < \chi(\mathcal{T}_R). \\ \mathcal{T}_R^* > \mathcal{T} > \mathcal{T}_R &\implies \chi'(\mathcal{T}) < 0 \quad \implies \chi(\mathcal{T}_R^*) < \chi(\mathcal{T}) < \chi(\mathcal{T}_R).\end{aligned} \tag{1.122}$$

As $\chi(\mathcal{T}_R^*) = 0$, we have $\chi(\mathcal{T}_R) > 0$, in both cases. Accounting for (1.119), we get

$$\begin{aligned}0 < \chi(\mathcal{T}_R) &= e^{\text{EOS}}\left(\mathcal{T}_R, s_R\right) - \frac{p^{\text{EOS}}\left(\mathcal{T}_R, s_R\right)^2}{2a^2} - e^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right) + \frac{p^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right)^2}{2a^2} \\ &\quad + p^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right)\left(\mathcal{T}_R + \frac{p^{\text{EOS}}\left(\mathcal{T}_R, s_R\right)}{a^2} - \mathcal{T}_R^* - \frac{p^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right)}{a^2}\right) \\ &= e_R^* - \frac{(\Pi_R^*)^2}{2a^2} - e^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right) + \frac{p^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right)^2}{2a^2} + p^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right)\left(\frac{\Pi_R^*}{a^2} - \frac{p^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right)}{a^2}\right) \\ &= e_R^* - e^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right) - \frac{\left(p^{\text{EOS}}\left(\mathcal{T}_R^*, s_R\right) - \Pi_R^*\right)^2}{2a^2}.\end{aligned} \tag{1.123}$$

Similar lines can be used for $k = L$. □

We present a result concerning the behavior of the numerical scheme in the low Mach regime defined in section 1.6.1 : we consider a one-dimensional smooth solution of the pressure subsystem (1.102) such that $\partial_{\tilde{x}}\tilde{p} + \widetilde{\tilde{\rho}(\partial_x \phi)} = O(\text{Ma}^2)$. Then, we proceed as in [Chalons et al. 2016a] by evaluating the truncation error (in the sense of the Finite Difference) obtained by substituting these low Mach flow parameters into the finite volume update formula derived from the fluxes (1.117). We obtain the following results.

Proposition 1.B.3. *In the low Mach regime, the rescaled discretization of the pressure system is consistent with*

$$\partial_{\tilde{t}}\tilde{\rho} = 0, \qquad \partial_{\tilde{t}}(\tilde{\rho}\tilde{u}) + \frac{1}{\text{Ma}^2}(\partial_{\tilde{x}}\tilde{p} + \widetilde{\tilde{\rho}(\partial_x \phi)}) = O(\Delta\tilde{t}) + O\left(\frac{\Delta\tilde{x}}{\text{Ma}}\right), \tag{1.124}$$

$$\partial_{\tilde{t}}(\tilde{\rho}\tilde{E}) + \partial_{\tilde{x}}(\tilde{p}\tilde{u}) = O(\Delta\tilde{t}) + O(\text{Ma}\,\Delta\tilde{x}). \tag{1.125}$$

If one performs a similar evaluation for the full FSLP scheme, one can see that the truncation error term $O\left(\frac{\Delta\tilde{x}}{\mathrm{Ma}}\right)$ that appears in the momentum equation of (1.125) will be the only error term whose magnitude is not uniform with respect to Ma. Similarly as in [Dellacherie 2010; Chalons et al. 2016a; Padioleau et al. 2019; Dellacherie et al. 2016], this truncation error term can be traced back to the non-centered part of $\Pi^*_{j+1/2}$. To tackle this issue, we adopt the modification used in [Chalons et al. 2016a; Padioleau et al. 2019] by replacing $\Pi^*_{j+1/2}$ with

$$\Pi^{*,\theta}_{j+1/2} = \frac{1}{2}\left(\Pi^n_j + \Pi^n_{j+1}\right) - \theta_{j+1/2}\frac{a_{j+1/2}}{2}\left(u^n_{j+1} - u^n_j\right), \tag{1.126}$$

where $\theta_{j+1/2} \in [0,1]$. This results in the update relation (1.12) that is a finite approximation of (1.102) with the flux definition (1.15). We will see in 1.C how this resulting modified flux can still be associated with an Approximate Riemann solver.

1.C . All-regime approximate Riemann solver for the pressure subsystem

Following similar lines as in [Chalons et al. 2016a] : although the modified pressure scheme (1.15) is defined as a flux scheme, it is possible to find an approximate Riemann solver $(\boldsymbol{U}^\theta_{\mathsf{RP}}, \Pi^\theta_{\mathsf{RP}}, \mathcal{T}^\theta_{\mathsf{RP}})$ that enables to retrieve the numerical flux $\boldsymbol{P}^\theta_{j+1/2} = (0, \Pi^{*,\theta}_{j+1/2}, \Pi^{*,\theta}_{j+1/2}u^*_{j+1/2})$. We suppose that $(\boldsymbol{U}^\theta_{\mathsf{RP}}, \Pi^\theta_{\mathsf{RP}}, \mathcal{T}^\theta_{\mathsf{RP}})$ has the same structure as $(\boldsymbol{U}_{\mathsf{RP}}, \Pi_{\mathsf{RP}}, \mathcal{T}_{\mathsf{RP}})$, we consider

$$(\boldsymbol{U}^\theta_{\mathsf{RP}}, \Pi^\theta_{\mathsf{RP}}, \mathcal{T}^\theta_{\mathsf{RP}}, \phi_{\mathsf{RP}})\left(\frac{x-\bar{x}}{t}; \boldsymbol{U}_L, \Pi_L, \mathcal{T}_L, \phi_L, \boldsymbol{U}_R, \Pi_R, \mathcal{T}_R, \phi_R\right) = \begin{cases} (\boldsymbol{U}_L, \Pi_L, \mathcal{T}_L, \Phi_L), & \text{if } \frac{x-\bar{x}}{t} \leq -\frac{a}{\rho_L}, \\ (\boldsymbol{U}^{*,\theta}_L, \Pi^{*,\theta}_L, \mathcal{T}^{*,\theta}_L, \Phi_L), & \text{if } -\frac{a}{\rho_L} < \frac{x-\bar{x}}{t} \leq 0, \\ (\boldsymbol{U}^{*,\theta}_R, \Pi^{*,\theta}_R, \mathcal{T}^{*,\theta}_R, \Phi_R), & \text{if } 0 < \frac{x-\bar{x}}{t} \leq \frac{a}{\rho_R}, \\ (\boldsymbol{U}_R, \Pi_R, \mathcal{T}_R, \Phi_R), & \text{if } \frac{a}{\rho_R} < \frac{x-\bar{x}}{t}, \end{cases} \tag{1.127}$$

where Π_k, $\mathcal{T}_k$ and Φ_k verify (1.104), $k = L, R$. The states $(\boldsymbol{U}^{*,\theta}_k, \Pi^{*,\theta}_k, \mathcal{T}^{*,\theta}_k)$, $k = L, R$ are yet to be defined. First, we impose that $(\boldsymbol{U}_{\mathsf{RP}}, \Pi_{\mathsf{RP}}, \mathcal{T}_{\mathsf{RP}})$ verifies the consistency in the integral sense

$$\begin{aligned} \begin{bmatrix} \frac{1}{\alpha}\boldsymbol{P}(\boldsymbol{U}_R) - \frac{1}{\alpha}\boldsymbol{P}(\boldsymbol{U}_L) \\ \frac{1}{\alpha}a^2(u_R - u_L) \\ -(\frac{1}{\alpha}u_R - \frac{1}{\alpha}u_L) \end{bmatrix} &= -\frac{a}{\alpha\rho_L}\begin{bmatrix} \boldsymbol{U}^{*,\theta}_L - \boldsymbol{U}_L \\ (\rho\Pi)^{*,\theta}_L - (\rho\Pi)_L \\ (\rho\mathcal{T})^{*,\theta}_L - (\rho\mathcal{T})_L \end{bmatrix} + \frac{a}{\alpha\rho_R}\begin{bmatrix} \boldsymbol{U}_R - \boldsymbol{U}^{*,\theta}_R \\ (\rho\Pi)_R - (\rho\Pi)^{*,\theta}_R \\ (\rho\mathcal{T})_R - (\rho\mathcal{T})^{*,\theta}_R \end{bmatrix} \\ &\quad + \frac{\Delta x_L + \Delta x_R}{2}\begin{bmatrix} \frac{1}{\alpha}\{\boldsymbol{S}\} \\ 0 \\ 0 \end{bmatrix}. \end{aligned} \tag{1.128}$$

We then enforce that the numerical flux resulting from (1.128) is $\boldsymbol{P}^\theta_\Delta$, which boils down to require that

$$\begin{aligned} &\begin{bmatrix} \frac{1}{\alpha}\boldsymbol{P}^\theta_\Delta \\ \frac{1}{\alpha}a^2u^\theta_\Delta \\ -\frac{1}{\alpha}u^\theta_\Delta \end{bmatrix}(\boldsymbol{U}_L, \Pi_L, \phi_L, \boldsymbol{U}_R, \Pi_R, \phi_R) \\ &= \begin{bmatrix} \boldsymbol{P}(\boldsymbol{U}_R, \Pi_R) + \boldsymbol{P}(\boldsymbol{U}_L, \Pi_L) \\ a^2u_R + a^2u_L \\ -(u_R + u_L) \end{bmatrix} - \frac{a}{\rho_L}\begin{bmatrix} \boldsymbol{U}^{,\theta}_L - \boldsymbol{U}_L \\ (\rho\Pi)^{,\theta}_L - (\rho\Pi)_L \\ (\rho\mathcal{T})^{,\theta}_L - (\rho\mathcal{T})_L \end{bmatrix} - \frac{a}{\rho_R}\begin{bmatrix} \boldsymbol{U}_R - \boldsymbol{U}^{,\theta}_R \\ (\rho\Pi)_R - (\rho\Pi)^{,\theta}_R \\ (\rho\mathcal{T})_R - (\rho\mathcal{T})^{,\theta}_R \end{bmatrix}. \end{aligned} \tag{1.129}$$

Choosing $\rho_k^{*,\theta} = \rho_k$, $k = L, R$, relation (1.128) and (1.129) provide a linear system with respect to $u_k^{*,\theta}$, $\Pi_k^{*,\theta}$, $\mathcal{T}_k^{*,\theta}$ and $E_k^{*,\theta}$, $k = 1, 2$ whose solution is

$$\rho_L^* = \rho_L, \qquad \rho_R^* = \rho_R, \tag{1.130a}$$

$$E_L^{*,\theta} = E_L^* - (1-\theta)\frac{u_R - u_L}{2}u^*, \qquad E_R^{*,\theta} = E_R^* + (1-\theta)\frac{u_R - u_L}{2}u^*, \tag{1.130b}$$

$$u_L^{*,\theta} = u^* - (1-\theta)\frac{u_R - u_L}{2}, \qquad u_R^{*,\theta} = u^* + (1-\theta)\frac{u_R - u_L}{2}, \tag{1.130c}$$

$$\Pi_L^{*,\theta} = \Pi_L^*, \qquad \Pi_R^{*,\theta} = \Pi_R^*, \tag{1.130d}$$

$$\mathcal{T}_L^{*,\theta} = \mathcal{T}_L^*, \qquad \mathcal{T}_R^{*,\theta} = \mathcal{T}_R^*. \tag{1.130e}$$

We now turn to positivity-preserving related properties. Let us note $e_k^{*,\theta} = E_k^{*,\theta} - u_k^{*,\theta 2}/2$, we have the following result.

Proposition 1.C.1. *Assuming again that a is large enough, we have*

$$e_k^{*,\theta} - e^{EOS}\left(\mathcal{T}_k^{*,\theta}, s_k\right) - \frac{\left(p^{EOS}\left(\mathcal{T}_k^{*,\theta}, s_k\right) - \Pi_k^{*,\theta}\right)^2}{2a^2} + \frac{(1-\theta)^2(u_R - u_L)^2}{8} \geq 0. \tag{1.131}$$

Démonstration. Let us consider the case $k = R$, by (1.130) we get

$$\begin{aligned} e_R^{*,\theta} - e_R^* &= E_R^{*,\theta} - E_R^* - \frac{1}{2}(u_R^{*,\theta 2} - u_R^{*\,2}) \\ &= (1-\theta)\frac{u_R - u_L}{2}u^* - \frac{1}{2}\left((u^*)^2 + u^*(1-\theta)(u_R - u_L) + (1-\theta)^2\frac{(u_R - u_L)^2}{4} - (u^*)^2\right) \\ &= -\frac{1}{8}(1-\theta)^2(u_R - u_L)^2. \end{aligned} \tag{1.132}$$

Using (1.120), we obtain

$$\begin{aligned} e_R^{*,\theta} - e^{\mathrm{EOS}}\left(\mathcal{T}_R^{*,\theta}, s_R\right) &= e_R^{*,\theta} - e_R^* + e_R^* - e^{\mathrm{EOS}}\left(\mathcal{T}_R^{*,\theta}, s_R\right) \\ &= -\frac{1}{8}(1-\theta)^2(u_R - u_L)^2 + e_R^* - e^{\mathrm{EOS}}\left(\mathcal{T}_R^{*,\theta}, s_R\right) \\ &\geq -\frac{1}{8}(1-\theta)^2(u_R - u_L)^2 + \frac{\left(p^{\mathrm{EOS}}\left(\mathcal{T}_k^{*,\theta}, s_k\right) - \Pi_k^{*,\theta}\right)^2}{2a^2}. \end{aligned}$$

Similar lines can be used for the case $k = L$. □

The relation (1.131) highlights the role of the inequality

$$\frac{1}{2a^2}\left(p^{\mathrm{EOS}}(\mathcal{T}_k^{*,\theta}, s_k) - \Pi_k^*\right)^2 - \frac{(1-\theta)^2\left(u_R - u_L\right)^2}{8} \geq 0, \quad k = L, R \tag{1.133}$$

in obtaining stability properties for the modified scheme. We have the following proposition.

Proposition 1.C.2. *Let us note : $s_k^{*,\theta} = s^{EOS}(\mathcal{T}_k^{*,\theta}, e_k^{*,\theta})$, if (1.133) is satisfied, then*

- *the modified approximate Riemann solver* (1.127) *preserves the positivity of the internal energy, that is to say :* $e_k^{*,\theta} > 0$, $k = R, L$,
- *the modified approximate Riemann solver* (1.127) *verifies* $s_k^{*,\theta} \geq s_k$, $k = R, L$,
- *the modified approximate Riemann solver* (1.127) *is entropy satisfying in the sense that*

$$-a(s_L^{*,\theta} - s_L) + a(s_R - s_R^{*,\theta}) \geq 0. \tag{1.134}$$

Démonstration. If (1.C.2) is satisfied, then $e_k^{*,\theta} \geq e^{\text{EOS}}\left(\mathcal{T}_k^{*,\theta}, s_k\right)$. By the assumption on the EOS, we have that $e_k^{*,\theta} > 0$. Now, considering a fixed $\overline{\mathcal{T}} > 0$, by (1.3) we know that $e' \mapsto s^{\text{EOS}}(\overline{\mathcal{T}}, e')$ is increasing, thus we deduce that $s^{\text{EOS}}(\mathcal{T}_k^{*,\theta}, e^{\text{EOS}}(\mathcal{T}_k^{*,\theta}, s_k^{*,\theta})) = s_k^{*,\theta} \geq s^{\text{EOS}}(\mathcal{T}_k^{*,\theta}, e^{\text{EOS}}(\mathcal{T}_k^{*,\theta}, s_k)) = s_k$, $k = L, R$. This implies (1.134). □

1.D . Eigenstructure of the off-equilibrium

We propose in this section to study the eigenstructure of the relaxation system ($1.74_{\nu=0}$). Let us first express the acoustic part of ($1.74_{\nu=0}$) using a change of variables : accounting for $e^P = E^P - (u^P)^2/2$, the evolution equations for E^P, for Π^P and $\mathcal{T}^P$ in ($74a_{\nu=0}$) yield

$$\partial_t(\rho^P e^P) + 2\Pi^P \partial_x u^P = 0, \qquad 2\partial_x u^P = \partial_t(\rho^P \Pi^P / a^2). \tag{1.135}$$

We thus obtain the stationary equations

$$\partial_t\left[e^P - \frac{(\Pi^P)^2}{2a^2}\right] = 0, \qquad \partial_t\left[\mathcal{T}^P + \frac{\Pi^P}{a^2}\right] = 0. \tag{1.136}$$

So now the acoustic subsystem ($74a_{\nu=0}$) takes the simple form

$$\partial_t \phi = 0, \qquad \partial_t \rho^P = 0, \qquad \partial_t\left[e^P - \frac{(\Pi^P)^2}{2a^2}\right] = 0, \tag{1.137a}$$

$$\partial_t(\rho^P u^P) + 2\partial_x \Pi^P + 2\rho^P \partial_x \phi^P = 0, \qquad \partial_t(\rho^P \Pi^P) + 2a^2 \partial_x u^P = 0, \qquad \partial_t\left[\mathcal{T}^P + \frac{\Pi^P}{a^2}\right] = 0. \tag{1.137b}$$

We now turn to the advection part of ($1.74_{\nu=0}$) : the subsystem ($74b_{\nu=0}$) takes the simple form

$$\partial_t \rho^A + \partial_x(2\rho^A u^P) = 0, \quad \partial_t\left[\rho^A \mathcal{T}^A - \frac{\rho^P \Pi^P}{a^2}\right] = 0, \quad \partial_t b^A + 2u^P \partial_x b^A = 0, \quad b^A \in \{u^A, E^A, \Pi^A\}. \tag{1.138}$$

Therefore if we set

$$\boldsymbol{W}^T = \left[u^P, \Pi^P, \rho^P, \phi, e^P - \frac{(\Pi^P)^2}{2a^2}, \mathcal{T}^P + \frac{\Pi^P}{a^2}, \rho^A \mathcal{T}^A - \frac{\rho^P \Pi^P}{a^2}, u^A, \Pi^A, E^A, \rho^A\right], \tag{1.139}$$

we can see that (1.74$_{\nu=0}$) can be recast into the following quasilinear system

$$\partial_t \boldsymbol{W} + \boldsymbol{M}(\boldsymbol{W})\partial_x \boldsymbol{W} = 0, \quad \boldsymbol{M}(\boldsymbol{W}) = \begin{bmatrix} 0 & \frac{2}{\rho^P} & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{2a^2}{\rho^P} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2u^P & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2u^P & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2u^P & 0 \\ 2\rho^A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2u^P \end{bmatrix}. \tag{1.140}$$

It is then straightforward to see that the eigenvalues of $\boldsymbol{M}(\boldsymbol{W})$ are $2u^P$ (with an algebraic multiplicity 4), 0 (with an algebraic multiplicity 5) and $\pm 2a/\rho^P$.

The eigenvectors $\left(\boldsymbol{r}_0^{(k)}\right)_{k=1,\dots,3}$, $\left(\boldsymbol{r}_{u^P}^{(k)}\right)_{k=1,\dots,4}$ and $\boldsymbol{r}_\pm$ that are respectively associated with 0, $2u^P$ and $\pm 2a/\rho^P$ are

$$\boldsymbol{r}_0^{(1)} = [0,0,1,0,0,0,0,0,0,0,0]^T, \qquad \boldsymbol{r}_0^{(2)} = \left[0,-\rho^P,0,1,0,0,0,0,0,0,0\right]^T, \tag{1.141a}$$

$$\boldsymbol{r}_0^{(3)} = [0,0,0,0,1,0,0,0,0,0,0]^T, \qquad \boldsymbol{r}_0^{(4)} = [0,0,0,0,0,1,0,0,0,0,0]^T \tag{1.141b}$$

$$\boldsymbol{r}_0^{(5)} = [0,0,0,0,0,0,1,0,0,0,0]^T, \tag{1.141c}$$

$$\boldsymbol{r}_{u^P}^{(1)} = [0,0,0,0,0,0,0,1,0,0,0]^T, \qquad \boldsymbol{r}_{u^P}^{(2)} = [0,0,0,0,0,0,0,0,1,0,0]^T, \tag{1.141d}$$

$$\boldsymbol{r}_{u^P}^{(3)} = [0,0,0,0,0,0,0,0,0,1,0]^T, \qquad \boldsymbol{r}_{u^P}^{(4)} = [0,0,0,0,0,0,0,0,0,0,1]^T, \tag{1.141e}$$

$$\boldsymbol{r}_+ = \left[1,a,0,0,0,0,0,0,0,0,-\frac{\rho^A\rho^P}{\rho^P u^P - a}\right]^T, \quad \boldsymbol{r}_- = \left[1,-a,0,0,0,0,0,0,0,0,-\frac{\rho^A\rho^P}{\rho^P u^P + a}\right]^T, \tag{1.141f}$$

so that (1.140) is hyperbolic and only involves linearly degenerate fields.

2 - A multi-dimensional, robust, and cell-centered finite-volume scheme for the ideal MHD equations

2.1 . Preamble

This chapter is mostly identical to [Tremblin et al. 2024]. In this preamble, I aim to clarify my contributions to this research. Specifically, my involvement did not include :

1. Developing the proof for the entropy inequality for the discretization of the magneto-acoustic system (see 2.7.1).
2. Writing sections 2.2 to 2.7.

However, my contributions were as follows :

1. Writing section 2.9, and conducting the associated numerical experiments.
2. Writing section 2.8 and the appendix.
3. Reviewing other sections, suggesting and implementing changes and improvements.
4. Conducting all the numerical experiments that allowed us to develop the form of the entropic correction we employ (2.53) as well as the strategy around it and the isotropic formula for the velocities c_a and c_b (2.37).
5. I supervised the internship of Valentin de Lia, during which we investigated the non-convergence of the magnetic Kelvin-Helmholtz instability in ideal MHD. Results are presented in section 2.8.

2.2 . Introduction

The ideal MHD equations are obtained by combining the Euler's equations of gas dynamic (1.1) for density ρ, momentum $\rho\mathbf{u}$, energy $\rho(e+\mathbf{u}^2/2)$, with the Faraday's law of induction describing the evolution of the magnetic field $\mathbf{B}$

$$\begin{aligned}
\partial_t \rho + \nabla\cdot(\rho\mathbf{u}) &= 0,\\
\partial_t(\rho\mathbf{u}) + \nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u}) &= -\nabla p + \mathbf{j}\times\mathbf{B},\\
\partial_t(\rho(e+\mathbf{u}^2/2)) + \nabla\cdot(\rho(e+\mathbf{u}^2/2)\mathbf{u}) &= -\nabla\cdot(p\mathbf{u}) + (\mathbf{j}\times\mathbf{B})\cdot\mathbf{u},\\
\partial_t\mathbf{B} + \nabla\times\mathbf{E} &= 0.
\end{aligned} \tag{2.1}$$

The term $\mathbf{j}\times\mathbf{B}$ is the Lorentz force. This system of equations is closed with the ideal Ohm's law $\mathbf{E} = -\mathbf{u}\times\mathbf{B}$, the low frequency Maxwell equation $\mathbf{j} = \nabla\times\mathbf{B}$ assuming a system of units in which the vacuum permeability is one, and an EOS connecting the pressure p to the density ρ (or specific volume $\tau = 1/\rho$) and internal energy e. The EOS also defines the specific physical entropy $s(\tau, e)$ assuming that $-s$ is a convex function of (τ, e). These thermodynamics quantities satisfy the Weyl's assumptions (1.3) and we have the Gibbs relation :

$$de + pd\tau = Tds. \tag{2.2}$$

This equivalently means that the internal energy is convex with respect to specific volume and entropy, hence the sound speed c_s defined by

$$c_s^2 = \left(\frac{\partial p}{\partial \rho}\right)_s \tag{2.3}$$

is real-valued to ensure that the eigenvalues of the system are real. Assuming smooth solutions of 2.1, one can show that they satisfy the following equation of conservation for the entropy (see appendix 2.C for the derivation)

$$\partial_t(\rho s) + \nabla \cdot (\rho s \mathbf{u}) = 0. \tag{2.4}$$

For the non-conservative form of the MHD equations, this holds for any value of the divergence of the magnetic field $\nabla \cdot \mathbf{B}$. Assuming that the divergence of the magnetic field is zero at an initial time $\nabla \cdot \mathbf{B} = 0$, it remains zero at all time following the divergence of the induction equation,

$$\partial_t(\nabla \cdot \mathbf{B}) = 0. \tag{2.5}$$

The free divergence constraint is therefore a consequence of the induction equation and not a dynamical constraint.

Equivalently, by adding terms proportional to $\nabla \cdot \mathbf{B}$ in the momentum and energy equations, one can obtain a conservative form for the MHD equations (see appendix 2.B for the derivation)

$$\begin{aligned}
\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) &= 0, \\
\partial_t(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + \sigma - \mathbf{B} \otimes \mathbf{B}) &= 0, \\
\partial_t(\rho E) + \nabla \cdot (\rho E \mathbf{u} + \sigma \cdot \mathbf{u} - (\mathbf{B} \cdot \mathbf{u})\mathbf{B}) &= 0, \\
\partial_t \mathbf{B} + \nabla \cdot (\mathbf{u} \otimes \mathbf{B} - \mathbf{B} \otimes \mathbf{u}) &= 0.
\end{aligned} \tag{2.6}$$

with $\sigma = (p + \mathbf{B}^2/2)\mathbf{I}$ and $E = e + \mathbf{u}^2/2 + \mathbf{B}^2/(2\rho)$. Assuming smooth solutions of 2.6, one can show that they satisfy the following equation for the evolution of the entropy by subtracting the evolution of the kinetic and magnetic energy from the evolution of the total energy (see appendix 2.C for the derivation)

$$\partial_t(\rho s) + \nabla \cdot (\rho s \mathbf{u}) = -\frac{\mathbf{u} \cdot \mathbf{B}}{T} \nabla \cdot \mathbf{B}, \tag{2.7}$$

which is compatible with entropy conservation only when $\nabla \cdot \mathbf{B} = 0$ in constrast to the non-conservative form presented above [Després 2011]. This shows that the entropy balance is closely related to the free divergence constraint for the conservative MHD equations.

In the case of discontinuities such as shocks and in order to ensure dissipation, the second law of thermodynamics must be enforced and implies the entropy inequality

$$\partial_t(\rho s) + \nabla \cdot (\rho s \mathbf{u}) \geq 0, \tag{2.8}$$

The positivity of density in addition to the entropy inequality imply then positivity of internal energy which must be ensured in order to obtain a stable numerical scheme. After discretization, truncation errors on the $\nabla \cdot \mathbf{B}$ source term in 2.7 therefore leads to some issues in order to obtain an entropy satisfying numerical scheme ensuring a discrete version of 2.8.

Several solutions exist to ensure a stable numerical scheme compatible with 2.8. The first one is to align the mesh with the magnetic configuration which allows an exact divergence free constraint at machine precision. This is for example the case of all 1D tests for which a constant B_x value is sufficient to ensure exactly $\nabla \cdot \mathbf{B} = 0$. A

second solution is to remove the divergence errors so that the source term in 2.7 is zero or as small as possible. Such a solution encompass the divergence-cleaning method (see [Brackbill et Barnes 1980; Ryu et al. 1998; Dai et Woodward 1998]) and the constrained transport method (see [Evans et Hawley 1988; Balsara et Spicer 1999; Tóth 2000; Fromang et al. 2006]), however, we point out that these methods are not entropy satisfying and may fail with negative internal energies. A third solution is to design an entropy satisfying numerical scheme ensuring 2.8 for any value of the divergence of the magnetic field. This solution has been explored using relaxation methods in [Bouchut et al. 2007, 2010]. The multi-dimensional solver needs the introduction of a non-conservative entropic correction in the induction equation in order to obtain a symmetric form of the MHD equations [Godunov 1972; Busto et Dumbser 2023; Gallice 2003]. Ensuring a stable numerical scheme compatible with 2.8 is important in regions of low plasma beta where the internal energy is small compared to the magnetic energy. It is also important in high Alfvén number regions, in order to avoid the generation of spurious perturbations. We follow in this chapter, a similar approach to [Bouchut et al. 2007, 2010], but take advantage of splitting techniques introduced in [Chalons et al. 2016a] and extended in chapter 1 as a flux splitting method to design a fully-conservative multi-dimensional MHD solver in regions of high plasma beta / low Alfvén number, and an entropy satisfying, non conservative version with an entropic correction in regions of low plasma beta / high Alfvén number.

2.3 . Magneto-acoustic/transport splitting

Similarly to [Chalons et al. 2016a], we propose the following splitting of the conservative MHD equations into a magneto-acoustic sub-system

$$\begin{aligned}
\partial_t \rho + \rho \nabla \cdot \mathbf{u} = 0, \\
\partial_t (\rho \mathbf{u}) + \rho \mathbf{u} \nabla \cdot \mathbf{u} + \nabla \cdot (\sigma - \mathbf{B} \otimes \mathbf{B}) = 0, \\
\partial_t (\rho E) + \rho E \nabla \cdot \mathbf{u} + \nabla \cdot (\sigma \cdot \mathbf{u} - (\mathbf{B} \cdot \mathbf{u}) \mathbf{B}) = 0, \\
\partial_t \mathbf{B} + \mathbf{B} \nabla \cdot \mathbf{u} - \nabla \cdot (\mathbf{B} \otimes \mathbf{u}) = 0,
\end{aligned} \tag{2.9}$$

and a transport sub-system

$$\begin{aligned}
\partial_t \rho + \mathbf{u} \cdot \nabla \rho = 0, \\
\partial_t (\rho \mathbf{u}) + \mathbf{u} \cdot \nabla (\rho \mathbf{u}) = 0, \\
\partial_t (\rho E) + \mathbf{u} \cdot \nabla (\rho E) = 0, \\
\partial_t \mathbf{B} + \mathbf{u} \cdot \nabla (\mathbf{B}) = 0.
\end{aligned} \tag{2.10}$$

We emphasize that all the components of the magnetic field are transported at velocity $\mathbf{u}$ in the transport sub-system. We then propose to approximate the solution of 2.6 by approximating the solutions of the two sub-systems 2.9 and 2.10, i.e. for a discrete state $\mathbf{U}_i^n = (\rho, \rho\mathbf{u}, \rho E, \mathbf{B})_i^n$ in a cell Ω_i at time t^n, the update to $\mathbf{U}_i^{n+1}$ is first an update from $\mathbf{U}_i^n$ to $\mathbf{U}_i^{n+1-}$ by approximating the solution of 2.9, then an update from $\mathbf{U}_i^{n+1-}$ to $\mathbf{U}_i^{n+1}$ by approximating the solution of 2.10. We present in Sect. 2.4 and in Sect. 2.5 the discretization and the entropy analysis for each sub-system respectively.

2.4 . Relaxation approximation of the magneto-acoustic sub-system

The relaxation approximation of the magneto-acoustic sub-system and the associated entropy analysis in Sect. 2.7 heavily relies on earlier works by [Bouchut et al. 2007, 2010]. We highlight two main differences in our

approach : we keep in the analysis gradients of the magnetic field perpendicular to the interface that appears in the multi-dimensional case and we propose a different choice of relaxation parameters in the 5-wave solver to ensure the strict hyperbolicity of the relaxed system.

The multi-dimensional scheme will be obtained by taking advantage of the rotational invariance of the magneto-acoustic sub-system, following the lines of [Godlewski et Raviart 1996]. We, therefore, rewrite sub-system 2.9 in 1D, and simplify it by using the density evolution equation

$$\begin{aligned} \rho\partial_t\tau - \partial_x u = 0, \\ \rho\partial_t \mathbf{u} + \partial_x(\sigma \mathbf{e}_x - B_x\mathbf{B}) = 0, \\ \rho\partial_t E + \partial_x(\sigma u_x - (\mathbf{B}\cdot\mathbf{u})B_x) = 0, \\ \rho\partial_t(\tau\mathbf{B}) - \partial_x(B_x\mathbf{u}) = 0, \end{aligned} \tag{2.11}$$

with $\mathbf{e}_x$, the unit vector normal to the interface, B_x, B_y, and B_z the components of the magnetic field and u_x, u_y, and u_z the components of the velocity field. The eigenvalues of this sub-system are given by

$$-u, 0, \pm c_{ms}, \pm c_{ma}, \pm c_{mf} \tag{2.12}$$

with c_{ma}, the magnetic Alfvén speed, c_{ms}, the slow magnetosonic speed, c_{mf}, the fast magnetosonic speed defined by

$$\begin{aligned} c_{ma} &= \frac{|B_x|}{\sqrt{\rho}}, \\ c_{ms}^2 &= \frac{1}{2}\left(c_s^2 + \frac{\mathbf{B}^2}{\rho} - \sqrt{\left(c_s^2 + \frac{\mathbf{B}^2}{\rho}\right)^2 - 4c_s^2 c_{ma}^2}\right), \\ c_{mf}^2 &= \frac{1}{2}\left(c_s^2 + \frac{\mathbf{B}^2}{\rho} + \sqrt{\left(c_s^2 + \frac{\mathbf{B}^2}{\rho}\right)^2 - 4c_s^2 c_{ma}^2}\right). \end{aligned} \tag{2.13}$$

We then introduce a relaxation procedure [Bouchut et al. 2007; Chalons et al. 2016a] with the relaxation pressures $\pi_{\mathbf{u}}$ playing the role of the fluxes in the impulsion equation and the relaxation variable r playing the role of the density in front of the time derivatives

$$\begin{aligned} r\partial_t\tau - \partial_x u = 0, \\ r\partial_t \mathbf{u} + \partial_x \pi_{\mathbf{u}} = 0, \\ r\partial_t E + \partial_x(\pi_{\mathbf{u}}\cdot\mathbf{u}) = 0, \\ r\partial_t(\tau\mathbf{B}) - \partial_x(B_x\mathbf{u}) = 0, \end{aligned} \tag{2.14}$$

with the following equations for the relaxation variables

$$\begin{aligned} \partial_t r = \frac{\rho - r}{\epsilon}, \\ r\partial_t\pi_u + (c_b^2 + b_y^2 + b_z^2)\partial_x u - c_a b_y \partial_x v - c_a b_z \partial_x w + d_x \partial_x B_x = \frac{\sigma - B_x^2 - \pi_u}{\epsilon}, \\ r\partial_t\pi_v - c_a b_y \partial_x u + c_a^2 \partial_x v + d_y \partial_x B_x = \frac{-B_x B_y - \pi_v}{\epsilon}, \\ r\partial_t\pi_w - c_a b_z \partial_x u + c_a^2 \partial_x w + d_z \partial_x B_x = \frac{-B_x B_z - \pi_w}{\epsilon}. \end{aligned} \tag{2.15}$$

The parameters c_a, c_b, b_y, b_z play the role of approximations of $\sqrt{\rho}|B_x|$, ρc_s, $\mathrm{sign}(B_x)\sqrt{\rho}B_y$, $\mathrm{sign}(B_x)\sqrt{\rho}B_z$, respectively, as in [Bouchut et al. 2007]. The extra parameters d_x, d_y, d_z are linked to the possibility of a non-constant B_x in the magneto-acoustic sub-system and play the role of approximations of $2B_x u/\tau + \mathbf{u}\cdot\mathbf{B}(\partial_e p - 1/\tau)$, $(B_x v + B_y u)/\tau$, and $(B_x w + B_z u)/\tau$, respectively. If these extra parameters are fixed to zero, the relaxation equations for $\pi_{\mathbf{u}}$ is the Lagrangian form of the relaxation equations used in [Bouchut et al. 2007]. By replacing all these parameters exactly by the quantities they approximate, Eq. 2.15 reduces to the evolution equation of $\sigma - B_x^2$, $-B_xB_y$, and $-B_xB_z$ in the limit $\epsilon \to \infty$. In order to obtain the same Riemann invariants as [Bouchut et al. 2007], we fix d_x, d_y, and d_z to zero and the other constants are evolved with

$$\partial_t c_a = \partial_t c_b = \partial_t b_y = \partial_t b_z = 0. \tag{2.16}$$

In the limit $\epsilon \to 0$, the relaxation equations in eq. 2.15 ensures that $r \to \rho$, $\pi_u \to \sigma - B_x^2$, $\pi_v \to -B_xB_y$, and $\pi_w \to -B_xB_z$. In this limit, Eq. 2.14 is then equivalent to Eq. 2.11. A classical approach to achieve the limit $\epsilon \to 0$ numerically is to first enforce the equilibrium relations $r = \rho$ and $\pi_{\mathbf{u}} = \sigma\mathbf{e}_x - B_x\mathbf{B}$ at time t^n and then solve 2.14 and 2.15 without the relaxation source terms. Using $L \equiv r/\rho$, the full system without the relaxation source term is

$$\begin{aligned}
\partial_t L - \partial_x u &= 0,\\
\partial_t(\rho L\mathbf{u}) + \partial_x \pi_{\mathbf{u}} &= 0,\\
\partial_t(\rho LE) + \partial_x(\pi_{\mathbf{u}}\cdot\mathbf{u}) &= 0,\\
\partial_t(L\mathbf{B}) - \partial_x(B_x\mathbf{u}) &= 0,\\
\partial_t(\rho L) &= 0,\\
\partial_t(\rho L\pi_u) + (c_b^2 + b_y^2 + b_z^2)\partial_x u - c_a b_y \partial_x v - c_a b_z \partial_x w &= 0,\\
\partial_t(\rho L\pi_v) - c_a b_y \partial_x u + c_a^2 \partial_x v &= 0,\\
\partial_t(\rho L\pi_w) - c_a b_z \partial_x u + c_a^2 \partial_x w &= 0.
\end{aligned} \tag{2.17}$$

After some tedious algebra, one can compute the eigenvalues of this system of 16 equations (including $\partial_t c_a = \partial_t c_b = \partial_t b_y = \partial_t b_z = 0$),

$$-u/L, 0, \pm c_{rs}/(\rho L), \pm c_{ra}/(\rho L), \pm c_{rf}/(\rho L) \tag{2.18}$$

with

$$\begin{aligned}
c_{ra} &= c_a,\\
c_{rs}^2 &= \frac{1}{2}\left(c_b^2 + c_a^2 + b_y^2 + b_z^2 - \sqrt{\left(c_b^2 + c_a^2 + b_y^2 + b_z^2\right)^2 - 4c_a^2 c_b^2}\right),\\
c_{rf}^2 &= \frac{1}{2}\left(c_b^2 + c_a^2 + b_y^2 + b_z^2 + \sqrt{\left(c_b^2 + c_a^2 + b_y^2 + b_z^2\right)^2 - 4c_a^2 c_b^2}\right).
\end{aligned} \tag{2.19}$$

The central wave at zero velocity has multiplicity 9. All the waves are linearly degenerate. Similarly to [Bouchut et al. 2007], $c_{rs} \le c_a \le c_{rf}$, $c_{rs} \le c_b \le c_{rf}$ and the eigenvalues of 2.17 match the eigenvalues of 2.11 for $c_a = \sqrt{\rho}|B_x|$, $c_b = \rho c_s$, $b_y = \mathrm{sign}(B_x)\sqrt{\rho}B_y$, $b_z = \mathrm{sign}(B_x)\sqrt{\rho}B_z$. Similarly to [Bouchut et al. 2007], a Chapman-Enskog analysis can be performed on the relaxation equations which leads to the following stability conditions

$$\frac{1}{\rho} - \frac{B_x^2}{c_a^2} \ge 0,$$

$$
\begin{aligned}
c_b^2 - \rho^2 c_s^2 &\geq 0, \\
(c_b^2 - \rho^2 c_s^2)\left(\frac{1}{\rho} - \frac{B_x^2}{c_a^2}\right) &\geq \left(B_y - \frac{B_x b_y}{c_a}\right)^2 + \left(B_z - \frac{B_x b_z}{c_a}\right)^2,
\end{aligned} \tag{2.20}
$$

in order to ensure positive eigenvalues of the entropy diffusion matrix.

The 3+1 and 5+1 wave solver The solution of the Riemann problem associated to 2.17 contain 7+1 waves in the general case, 7 waves that are identical to a Lagrangian version of the 1D relaxation solver presented in [Bouchut et al. 2007] to which we add a wave at $-u/L$ associated to B_x. Similarly to [Bouchut et al. 2010] we can design an approximate Riemann solver with 5+1 waves by choosing $b_y = b_z = 0$, or with 3+1 waves by choosing in addition $c_a = c_b = c$. The 5+1 wave solver is a good compromise between accuracy ad computational cost and we will use this approximation from now on.

We now look for strong Riemann invariants for the different waves by finding quantities transported at the corresponding wave speed [Godlewski et Raviart 1996]. B_x is a strong Riemann invariant associated to the wave at $-u/L$. Note that B_x is not constant but advected at velocity $-u/L$. B_x has to be understood as evaluated locally, upwind relative to the wave $-u/L$. c_a and c_b are strong Riemann invariants for the central wave with

$$
\frac{1}{\rho} + \frac{\pi_u}{c_b^2}, \frac{B_y}{\rho} + \frac{B_x}{c_a^2}\pi_v, \frac{B_z}{\rho} + \frac{B_x}{c_a^2}\pi_w, e + \frac{\mathbf{B}^2}{2\rho} - \frac{\pi_u^2}{2c_b^2} - \frac{\pi_v^2 + \pi_w^2}{2c_a^2} \tag{2.21}
$$

Similarly to [Bouchut et al. 2010], there are six strong Riemann invariants for the left and right waves $\pi_{\mathbf{u}} + c_{\mathbf{u}}\mathbf{u}$ and $\pi_{\mathbf{u}} - c_{\mathbf{u}}\mathbf{u}$, respectively, in which we have defined $c_{\mathbf{u}} = (c_b, c_a, c_a)$. Strong Riemann invariants for a given wave are weak Riemann invariants for the other waves. They are, therefore, weak Riemann invariants for the central wave, hence, $\mathbf{u}$ and $\pi_{\mathbf{u}}$ take the same value on the left and right of this wave that we shall define as $\mathbf{u}^*$ and $\pi_{\mathbf{u}}^*$ respectively. By using the weak Riemann invariants, we get

$$
\begin{aligned}
\mathbf{u}^* &= \frac{c_{\mathbf{u},l}\mathbf{u}_l + c_{\mathbf{u},r}\mathbf{u}_r + \pi_{\mathbf{u},l} - \pi_{\mathbf{u},r}}{c_{\mathbf{u},l} + c_{\mathbf{u},r}}, \\
\pi_{\mathbf{u}}^* &= \frac{c_{\mathbf{u},r}\pi_{\mathbf{u},l} + c_{\mathbf{u},l}\pi_{\mathbf{u},r} + c_{\mathbf{u},l}c_{\mathbf{u},r}(\mathbf{u}_l - \mathbf{u}_r)}{c_{\mathbf{u},l} + c_{\mathbf{u},r}}.
\end{aligned} \tag{2.22}
$$

Then one has

$$
B_x(x,t) = \begin{cases} B_{x,l} & \text{if } x/t < -u/L \\ B_{x,r} & \text{if } x/t > -u/L, \end{cases} \tag{2.23}
$$

hence, at the interface, we define $B_x^{-u^*} = B_x(0,t)$ with

$$
B_x^{-u^*} = \begin{cases} B_{x,l} & \text{if } u^* < 0 \\ B_{x,r} & \text{if } u^* > 0. \end{cases} \tag{2.24}
$$

The other intermediate states, e.g. $\tau_{l,r}^*$ and $e_{l,r}^*$ can be obtained by using 2.21, but are not needed for deriving the update of the numerical scheme. The discrete numerical scheme for the magneto-acoustic sub-system is then given by

$$
\begin{aligned}
L_i^{n+1-} &= 1 + \frac{\Delta t}{\Delta x}(u_{i+1/2}^* - u_{i-1/2}^*), \\
\rho_i^{n+1-} L_i^{n+1-} &= \rho_i^n,
\end{aligned}
$$

$$
\begin{aligned}
\rho_i^{n+1-}\mathbf{u}_i^{n+1-}L_i^{n+1-} &= \rho_i^n\mathbf{u}_i^n - \frac{\Delta t}{\Delta x}(\pi^*_{\mathbf{u},i+1/2} - \pi^*_{\mathbf{u},i-1/2}),\\
\rho_i^{n+1-}E_i^{n+1-}L_i^{n+1-} &= \rho_i^nE_i^n - \frac{\Delta t}{\Delta x}(\pi^*_{\mathbf{u},i+1/2}\cdot\mathbf{u}^*_{i+1/2} - \pi^*_{\mathbf{u},i-1/2}\cdot\mathbf{u}^*_{i-1/2}),\\
\mathbf{B}_i^{n+1-}L_i^{n+1-} &= \mathbf{B}_i^n + \frac{\Delta t}{\Delta x}(B^{-u^*}_{x,i+1/2}\mathbf{u}^*_{i+1/2} - B^{-u^*}_{x,i-1/2}\mathbf{u}^*_{i-1/2}),
\end{aligned}
\tag{2.25}
$$

with the CFL condition for this scheme

$$
\max_{i\in\mathbb{Z}}\left(\frac{c_{rf,i}}{\rho_i}\right)\Delta t \leq \frac{\Delta x}{2}
\tag{2.26}
$$

2.5 . Transport sub-system

The transport sub-system is a quasi-hyperbolic system that only involves the transport of conservative variables with the velocity u. We choose to approximate the solution of the 1D version of 2.10 thanks to a standard upwind finite volume approximation for $\mathbf{U} = (\rho, \rho\mathbf{u}, \rho E, \mathbf{B})$ by discretizing

$$
\frac{\partial \mathbf{U}}{\partial t} + u\frac{\partial \mathbf{U}}{\partial x} = \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial (u\mathbf{U})}{\partial x} - \mathbf{U}\frac{\partial u}{\partial x} = 0,
\tag{2.27}
$$

with

$$
\mathbf{U}_i^{n+1} = \mathbf{U}_i^{n+1-} - \frac{\Delta t}{\Delta x}(u^*_{i+1/2}\mathbf{U}_{i+1/2} - u^*_{i-1/2}\mathbf{U}_{i-1/2}) + \frac{\Delta t}{\Delta x}\mathbf{U}_i^{n+1-}(u^*_{i+1/2} - u^*_{i-1/2}),
\tag{2.28}
$$

with two possible choices of discretization for the interface states $\mathbf{U}_{i-1/2}$ and $\mathbf{U}_{i+1/2}$. The first choice

$$
\mathbf{U}_{i+1/2} = \begin{cases} \mathbf{U}_i^{n+1-} & \text{if } u^*_{i+1/2} \geq 0,\\ \mathbf{U}_{i+1}^{n+1-} & \text{if } u^*_{i+1/2} \leq 0, \end{cases}
\tag{2.29}
$$

leads to a magneto-acoustic+transport scheme of stencil 2 similar to [Chalons et al. 2016a]. The second choice

$$
\mathbf{U}_{i+1/2} = \begin{cases} \mathbf{U}_i^{n} & \text{if } u^*_{i+1/2} \geq 0,\\ \mathbf{U}_{i+1}^{n} & \text{if } u^*_{i+1/2} \leq 0, \end{cases}
\tag{2.30}
$$

leads to a magneto-acoustic+transport scheme of stencil 1 similar to [Bourgeois et al. 2024]/ chapter 1. We will refer to these choices of discretization as "stencil 1" and "stencil 2" in the rest of the paper. In both cases and using the notation $u^{\pm} = \frac{u\pm|u|}{2}$, the CFL condition of the transport sub-system is given by

$$
\max_{i\in\mathbb{Z}}((u^*_{i-1/2})^+ - (u^*_{i+1/2})^-)\Delta t \leq \Delta x.
\tag{2.31}
$$

The transport can also be written in the form

$$
\mathbf{U}_i^{n+1} = \mathbf{U}_i^{n+1-}L_i^{n+1-} - \frac{\Delta t}{\Delta x}(u^*_{i+1/2}\mathbf{U}_{i+1/2} - u^*_{i-1/2}\mathbf{U}_{i-1/2}).
\tag{2.32}
$$

2.6 . Magneto-acoustic+transport scheme

The global scheme is given by

$$
\rho_i^{n+1} = \rho_i^n \quad - \quad \frac{\Delta t}{\Delta x}(\rho_{i+1/2}u^*_{i+1/2} - \rho_{i-1/2}u^*_{i-1/2}),
$$

$$
\begin{aligned}
(\rho\mathbf{u})_i^{n+1} = (\rho\mathbf{u})_i^n &- \frac{\Delta t}{\Delta x}((\rho\mathbf{u})_{i+1/2}u^*_{i+1/2} + \pi^*_{\mathbf{u},i+1/2} \\
&\quad -(\rho\mathbf{u})_{i-1/2}u^*_{i-1/2} - \pi^*_{\mathbf{u},i-1/2}), \\
(\rho E)_i^{n+1} = (\rho E)_i^n &- \frac{\Delta t}{\Delta x}((\rho E)_{i+1/2}u^*_{i+1/2} + \pi^*_{\mathbf{u},i+1/2}\cdot\mathbf{u}^*_{i+1/2} \\
&\quad -(\rho E)_{i-1/2}u^*_{i-1/2} - \pi^*_{\mathbf{u},i-1/2}\cdot\mathbf{u}^*_{i-1/2}), \\
\mathbf{B}_i^{n+1} = \mathbf{B}_i^n &- \frac{\Delta t}{\Delta x}(\mathbf{B}_{i+1/2}u^*_{i+1/2} - B^{-u^*}_{x,i+1/2}\mathbf{u}^*_{i+1/2} \\
&\quad -\mathbf{B}_{i-1/2}u^*_{i-1/2} + B^{-u^*}_{x,i-1/2}\mathbf{u}^*_{i-1/2}).
\end{aligned} \tag{2.33}
$$

The global scheme of stencil 2 is stable under the most restrictive CFL condition between the magneto-acoustic and transport sub-systems. The scheme of stencil 1 is stable under a CFL condition involving the sum of the speeds of the magneto-acoustic and transport subsystem as demonstrated in [Bourgeois et al. 2024]/ chapter 1 and in Sect. 2.7.

2.7 . Entropy analysis

In this section, we first introduce under which conditions the 1D relaxation solver is entropy-satisfying. For a non-constant B_x in a multi-dimensional setup, it is clear that the fully-conservative solver is not entropy-satisfying : on the $-u/L$ wave, B_x is the only quantity that jumps, hence, induces a jump in internal energy because of the last Riemann invariant in 2.21. Similarly to [Bouchut et al. 2010], an entropy satisfying solver will require the introduction of an entropic correction to get a symmetric version of the MHD equations. We will present the multi-dimensional entropy-satisfying solver at the end of the section.

The choice of the relaxation parameter $c = c_a = c_b$ for the 3+1 wave approximate Riemann solver and c_a, c_b for the 5+1 wave solver is made to ensure that the solver is entropy satisfying for a constant B_x in 1D. If for all intermediate states $\mathbf{U}^*_{l,r}$, one has $\tau^*_{l,r} > 0$ and

$$
\begin{aligned}
(\rho^2 c_s^2)_{*,l,r} &\leq c_b^2, \\
\tau^*_{l,r} - \frac{B_x^2}{c_a^2} &\geq 0, \\
\left(B^2_{y,l,r} + B^2_{z,l,r}\right) &\leq (c_b^2 - (\rho^2 c_s^2)_{*,l,r})\left(\tau^*_{l,r} - \frac{B_x^2}{c_a^2}\right),
\end{aligned} \tag{2.34}
$$

with $(\rho^2 c_s^2)_{*,l,r} \equiv \sup_{\rho\in(\rho_*,\rho_l,\rho_r)}(\rho^2 c_s^2(\rho, s_{l,r}))$, there exists a numerical flux function $q^n_{i+1/2} = q(\mathbf{U}^n_i, \mathbf{U}^n_{i+1})$, consistent with zero (see [Chalons et al. 2016a]) such that

$$
\begin{aligned}
\rho_i^{n+1}s(\mathbf{U}_i^{n+1}) - \rho_i^n s(\mathbf{U}_i^n) &+ \frac{\Delta t}{\Delta x}(q^n_{i+1/2} + (\rho s)_{i+1/2}u^*_{i+1/2} \\
&\quad -q^n_{i-1/2} - (\rho s)_{i-1/2}u^*_{i-1/2}) \geq 0.
\end{aligned} \tag{2.35}
$$

Following [Bouchut et al. 2010], optimal choices of c_a and c_b for smooth solutions are given by

$$
\begin{aligned}
c_a^2 &= \rho(B_x^2 + |B_x|\sqrt{B_y^2 + B_z^2}) \\
c_b^2 &= \rho^2 c_s^2 + \rho(B_y^2 + B_z^2 + |B_x|\sqrt{B_y^2 + B_z^2})
\end{aligned} \tag{2.36}
$$

for the 5+1 wave solver and $c = \rho c_{mf}$ for the 3+1 wave solver. Optimal choices for discontinuous solutions are given in [Bouchut et al. 2010], however, in all the tests performed in Sect. 2.9 the smooth version has been sufficient

to ensure stability and is therefore preferred for its low computational cost. As noted in [Bouchut et al. 2010], the diffusion of the 5+1 solver is zero when $B_x = 0$ or $B_y^2 + B_z^2 = 0$ which means that the solver is exact in these conditions. We, however, point out that this is exactly where the MHD system is not strictly hyperbolic with $c_{ma} = 0$ for $B_x = 0$ and $c_{ma} = c_{ms}$ for $B_y^2 + B_z^2 = 0$. Therefore, in practice, we employ a more diffusing approximation for the choices of c_a and c_b by using the following inequality $|B_x|\sqrt{B_y^2 + B_z^2} \leq (B_x^2 + B_y^2 + B_z^2)/2$:

$$\begin{aligned} c_a^2 &= \rho(B_x^2 + (B_x^2 + B_y^2 + B_z^2)/2) \\ c_b^2 &= \rho^2 c_s^2 + \rho(B_y^2 + B_z^2 + (B_x^2 + B_y^2 + B_z^2)/2) \end{aligned} \tag{2.37}$$

to ensure the use of a stable strictly hyperbolic approximation even when B_x or $B_y^2 + B_z^2$ vanishes. It also helps with the isotropy of the numerical diffusion whenever there is a large difference between the normal and transverse magnetic intensity, avoiding the generation of spurious patterns. We decompose the proof of the entropy analysis of the global scheme into an entropy analysis of each sub-system, magneto-acoustic and transport, respectively.

2.7.1 . Entropy analysis of the magneto-acoustic sub-system in 1D

Proposition 1 : Let $s_{l,r} = s(\tau_{l,r}, e_{l,r})$. If the inequality

$$e_{l,r}^* \geq e(\tau_{l,r}^*, s_{l,r}) \tag{2.38}$$

is verified, there exists a numerical flux function $q_{i+1/2}^n = q(\mathbf{U}_i^n, \mathbf{U}_{i+1}^n)$, consistent with zero such that

$$L_i^{n+1-}\rho_i^{n+1-}s(\tau_i^{n+1-}, e_i^{n+1-}) - \rho_i^n s(\tau_i^n, e_i^n) + \frac{\Delta t}{\Delta x}(q_{i+1/2}^n - q_{i-1/2}^n) \geq 0 \tag{2.39}$$

Proof According to 2.2, at fixed τ, $e(\tau, s)$ is an increasing function of s, hence $e(\tau_{l,r}^*, s_{l,r}^*) \geq e(\tau_{l,r}^*, s_{l,r})$ implies $s_{l,r}^* \geq s_{l,r}$. This inequality then implies that for any $c > 0$

$$0 \geq -c(s_l^* - s_l) + c(s_r - s_r^*) \tag{2.40}$$

which is consistent with the integral form of the entropy inequality $\partial_t(s(\tau, e)) \geq 0$. As in [Chalons et al. 2016a], this implies the existence of $q_{i+1/2}^n = q(\mathbf{U}_i^n, \mathbf{U}_{i+1}^n)$ such that

$$s(\tau_i^{n+1-}, e_i^{n+1-}) - s(\tau_i^n, e_i^n) + \tau_i^n \frac{\Delta t}{\Delta x}(q_{i+1/2}^n - q_{i-1/2}^n) \geq 0 \tag{2.41}$$

The inequality 2.39 follows from $L_i^{n+1-}\rho_i^{n+1-} = \rho_i^n$.

Proposition 2 : The 5+1 wave approximate Riemann solver associated to the relaxation 2.17 of the magneto-acoustic sub-system is positive and satisfies all discrete entropy inequalities whenever for all intermediate states $\mathbf{U}_{l,r}^*$, one has $\tau_{l,r}^* > 0$ the inequalities 2.34 are verified.

Proof According to 2.21, the 5+1 wave relaxation Riemann problem has the same Riemann invariants as [Bouchut et al. 2010] apart from the addition of B_x as a strong Riemann invariant of the $-u/L$ wave. B_x has therefore to

be understood as evaluated locally according to 2.23. By introducing the decomposition into elementary dissipation terms similarly as in [Bouchut 2003], using the Riemann invariants 2.21 and defining $\sigma(\mathbf{U}) = p(\tau, s = s_{l,r}) + \mathbf{B}^2/2$, one can show that

$$e(\tau^*_{l,r}, s_{l,r}) - e^*_{l,r} = D_0(\mathbf{U}^*_{l,r}, \mathbf{U}_{l,r}) - \frac{1}{2}\left|\frac{\sigma(\mathbf{U}^*_{l,r})\mathbf{n} - B_x\mathbf{B}^* - \pi^*_{\mathbf{u}}}{c_{\mathbf{u}}}\right|^2, \tag{2.42}$$

with D_0 the dissipation associated to the central wave given by

$$\begin{aligned} D_0(\mathbf{U}^*_{l,r}, \mathbf{U}_{l,r}) &= e(\tau^*_{l,r}, s_{l,r}) - e(\tau_{l,r}, s_{l,r}) + p(\tau^*_{l,r}, s_{l,r})\left(\tau^*_{l,r} - \tau_{l,r}\right) \\ &+ \frac{1}{2c_b^2}\left(\sigma(\mathbf{U}^*_{l,r}) - \sigma(\mathbf{U}_{l,r})\right)^2 \\ &- \left(\tau_{l,r} - B_x^2/c_a^2\right)\frac{1}{2}\left|\mathbf{B}^* - \mathbf{B}_{l,r}\right|^2. \end{aligned} \tag{2.43}$$

The proof of proposition 2 then follows directly from the entropy analysis of [Bouchut et al. 2007] who showed that under 2.34 and by using 2.42, the inequality 2.38 is verified.

The final part of the analysis requires to give the conditions under which the relaxation approximation is positive for the intermediate states of the specific volume $\tau^*_{l,r} > 0$. These conditions for the relaxation parameters are provided in proposition 3.3 of [Bouchut et al. 2010], however we do not explicitly specify them here because we will use a less restrictive choice with Eq. (2.37) which seems sufficient in practice in all the numerical tests performed in Sect. 2.9.

2.7.2 . Entropy analysis of the transport sub-system in 1D

By using $u^{\pm} = \frac{u \pm |u|}{2}$, the transport step of the global scheme of stencil 2 can be written in the form

$$\mathbf{U}_i^{n+1} = \frac{\Delta t}{\Delta x}u^{*,+}_{i-1/2}\mathbf{U}_{i-1}^{n+1-} - \frac{\Delta t}{\Delta x}u^{*,-}_{i+1/2}\mathbf{U}_{i+1}^{n+1-} + \left(1 - \frac{\Delta t}{\Delta x}(u^{*,+}_{i-1/2} - u^{*,-}_{i+1/2})\right)\mathbf{U}_i^{n+1-}, \tag{2.44}$$

hence $\mathbf{U}_i^{n+1}$ is a convex combination of $\mathbf{U}_{i-1}^{n+1-}$, $\mathbf{U}_i^{n+1-}$ and $\mathbf{U}_{i+1}^{n+1-}$ as their pre-factors are positive and sum to 1. By convexity of the function $\mathbf{U} \to -\rho s(\mathbf{U})$

$$\rho_i^{n+1}s(\mathbf{U}_i^{n+1}) \geq \rho_i^{n+1-}L_i^{n+1-}s(\mathbf{U}_i^{n+1-}) - \frac{\Delta t}{\Delta x}((\rho s)_{i+1/2}u^*_{i+1/2} - (\rho s)_{i-1/2}u^*_{i-1/2}). \tag{2.45}$$

By combining, the inequalities 2.39 and 2.45 we obtain the inequality 2.35. Following [Bourgeois et al. 2024]/ chapter 1, for the global scheme of stencil 1, the transport step can be written in the form

$$\mathbf{U}_i^{n+1} = \alpha_i\mathbf{U}_i^A + \alpha_i(1-\alpha_i)\mathbf{U}_i^T \tag{2.46}$$

for any $\alpha_i \in]0,1[$ and

$$\begin{aligned} \mathbf{U}_i^A &= \mathbf{U}_i^n + \frac{1}{\alpha_i}\frac{\Delta t}{\Delta x}(\mathbf{U}_i^{n+1-}L_i^{n+1-} - \mathbf{U}_i^n), \\ \mathbf{U}_i^T &= \mathbf{U}_i^n - \frac{1}{1-\alpha_i}\frac{\Delta t}{\Delta x}(u^*_{i+1/2}\mathbf{U}_{i+1/2} - u^*_{i-1/2}\mathbf{U}_{i-1/2}), \end{aligned} \tag{2.47}$$

with $\mathbf{U}_i^A$ corresponding to a magneto-acoustic update with $\Delta t^A = \frac{1}{\alpha_i}\Delta t$ and $\mathbf{U}_i^T$ corresponding to a conservative transport update also with $\Delta t^T = \frac{1}{1-\alpha_i}\Delta t$. Following [Bourgeois et al. 2024]/ chapter 1, $\mathbf{U}_i^T/\rho_i^{n+1}$ can be

written as a convex combination of $\mathbf{U}_i^n/\rho_i^n$. Thus, we can also obtain 2.35 by using the convexity of 2.46 under the CFL conditions :

$$\max_{i\in\mathbb{Z}}((u^*_{i-1/2})^+ - (u^*_{i+1/2})^-)\frac{1}{\alpha_i}\Delta t \leq \Delta x. \tag{2.48}$$

$$\max_{i\in\mathbb{Z}}\left(\frac{c_{rf,i}}{\rho_i}\right)\frac{1}{1-\alpha_i}\Delta t \leq \frac{\Delta x}{2}. \tag{2.49}$$

As the local choice of α_i is free, we can pick it so that both conditions coincide, giving the following condition for the stencil 1 scheme :

$$\max_{i\in\mathbb{Z}}((u^*_{i-1/2})^+ - (u^*_{i+1/2})^- + 2\frac{c_{rf,i}}{\rho_i})\Delta t \leq \Delta x. \tag{2.50}$$

2.7.3 . Symmetric system for multi-dimensional MHD

Similarly to [Bouchut et al. 2010], we introduce an entropic correction on the induction equation proportional to $\nabla\cdot\mathbf{B}$,

$$\partial_t\mathbf{B} + \nabla\cdot(\mathbf{u}\otimes\mathbf{B} - \mathbf{B}\otimes\mathbf{u}) + \mathbf{u}\nabla\cdot\mathbf{B} = 0. \tag{2.51}$$

The rest of the MHD system is not changed and, of course, 2.51 is equivalent to the standard form when $\nabla\cdot\mathbf{B} = 0$. For smooth solutions follow the entropy evolution (see appendix 2.C for the derivation)

$$\partial_t(\rho s) + \nabla\cdot(\rho s\mathbf{u}) = 0. \tag{2.52}$$

We recall the discretization of this entropic correction as [Bouchut et al. 2010] which results in two different values of $B_x^{-u^*}$ at an interface, $B^{-u^*}_{x,i+1/2,l} = B^n_{x,i}$ and $B^{-u^*}_{x,i+1/2,r} = B^n_{x,i+1}$, hence giving a non-conservative discretization of the induction equation with

$$\begin{aligned}\mathbf{B}_i^{n+1} = \mathbf{B}_i^n \quad - \quad & \frac{\Delta t}{\Delta x}(\mathbf{B}_{i+1/2}u^*_{i+1/2} - B^n_{x,i}\mathbf{u}^*_{i+1/2} \\ & -\mathbf{B}_{i-1/2}u^*_{i-1/2} + B^n_{x,i}\mathbf{u}^*_{i-1/2}).\end{aligned} \tag{2.53}$$

With this non-conservative source term, the evolution equation of B_x is simply $\partial_t B_x = 0$ and the system becomes symmetric with an additional wave centered at 0 instead of the $-u/L$ wave [Godunov 1972]. The strong Riemann invariant B_x jumps at 0, similarly to the other Riemann invariant 2.21. As in [Bouchut et al. 2010], the 3+1 and 5+1 approximate Riemann solvers with the non-conservative entropic correction are entropy satisfying with the same proof presented above, B_x simply needs to be understood as evaluated locally with a jump on the central wave.

We emphasize that the normal component of the magnetic field for $B_x^{-u^*}$ in 2.53 is always the value at cell center $B^n_{x,i}$ both at first and second order. As noted by [Klingenberg et Waagan 2010], the entropic correction vanishes for smooth solutions at second order if one uses the reconstructed values at interfaces. The proposed discretization in 2.53 avoids this problem and can be employed for both 1st and 2nd order.

2.8 . The Kelvin Helmholtz instability in ideal MHD

In the initial stages of developing our numerical scheme, we did not consider incorporating the entropic correction (2.53). This was due to the fact that the standard MHD test cases we used did not present low plasma beta regions that would necessitate such terms. The need for the entropic correction became apparent when we

encountered numerical instabilities during simulations of the magnetic Kelvin-Helmholtz Instability (MKHI), but this was only evident at higher resolutions. To investigate this issue further, we hired Valentin de Lia as an intern. Through a convergence study on the instability, we found that the minimum value of the plasma beta number in the simulation decreased as the resolution increased (as illustrated in Figure 2.1), demonstrating a non-convergent behavior, consistent with the numerical instabilities we observed at high resolution. We believe that this is due to the abscence of magnetic reconnection in ideal MHD : The Kelvin-Helmholtz instability occurs when two fluids moving at different speeds come into contact, causing shear and rotational effects that lead to turbulent flow. According to Alfvén's theorem, the magnetic field lines must move with the fluid, resulting in the field lines being drawn significantly closer together in turbulent regions without the ability to reconnect. This effect intensifies as the numerical resistivity is reduced by the increased resolution. Without specific measures to ensure the solver's entropy satisfying property behavior, such as the implementation of the entropic correction, simulations with a highly refined mesh are, therefore, prone to the development of negative energies. Consequently, we concluded that the ideal MHD model is unsuitable for this test case. We now consider the same test but in the framework of resistive MHD with Ohmic heating, i.e. by adding $\eta\vec{\Delta}\vec{B}$ and $\eta\left((\vec{\nabla}\wedge\vec{B})^2+\vec{B}\cdot\vec{\Delta}\vec{B}\right)$ to the RHS of the induction and total energy equations respectively. We now obtain a convergent behavior where magnetic field lines correctly reconnect, and the plasma beta number remains controlled. Figure 2.2 also shows that there is a strong correlation between the Ohmic heating and variations in the plasma beta number. In particular, it is clear that Ohmic heating and magnetic resistivity are limiting the decrease of the plasma beta number. Note that with the addition of the entropic correction, we can simulate the ideal case at any resolution, even though the test itself does not converge.

One of the objectives of the present chapter is to showcase the behavior of our scheme for ideal MHD under low plasma beta conditions. The results we just showcased led us to reducing the plasma beta number in the classical blast test case, as detailed in Section 2.9, over treating the MKHI test case as we aim to focus on ideal MHD rather than resistive MHD.

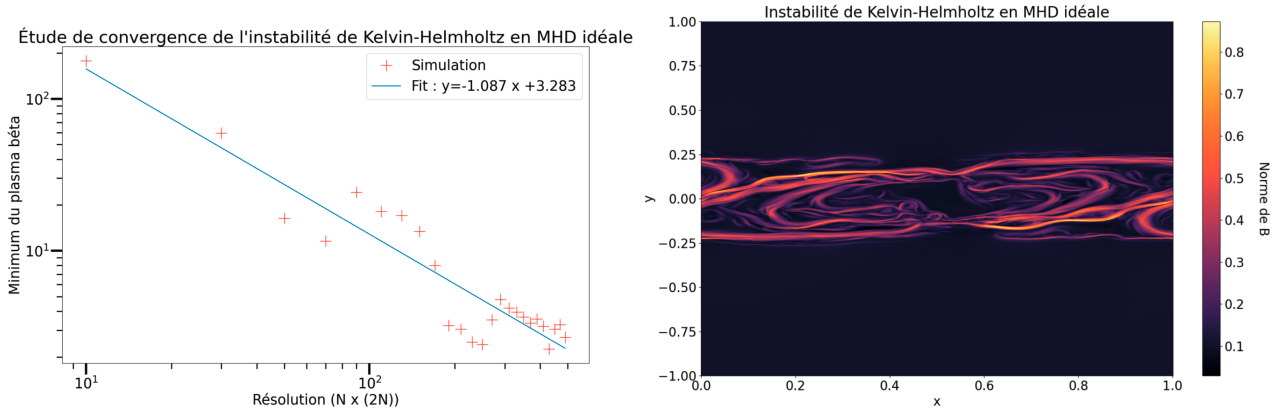


Figure 2.1 – Left : Minimum value of the plasma beta number reached in a MKHI simulation as a function of the resolution. Right : Norm of the magnetic field in a MKHI simulation. Courtesy of Valentin De Lia.

2.9 . Numerical results

All the simulations performed in this section are using a MUSCL-Hancock scheme [Van Leer 1974], delivering second order accuracy in space with states reconstructions and in time with a predictor-corrector step. Since our

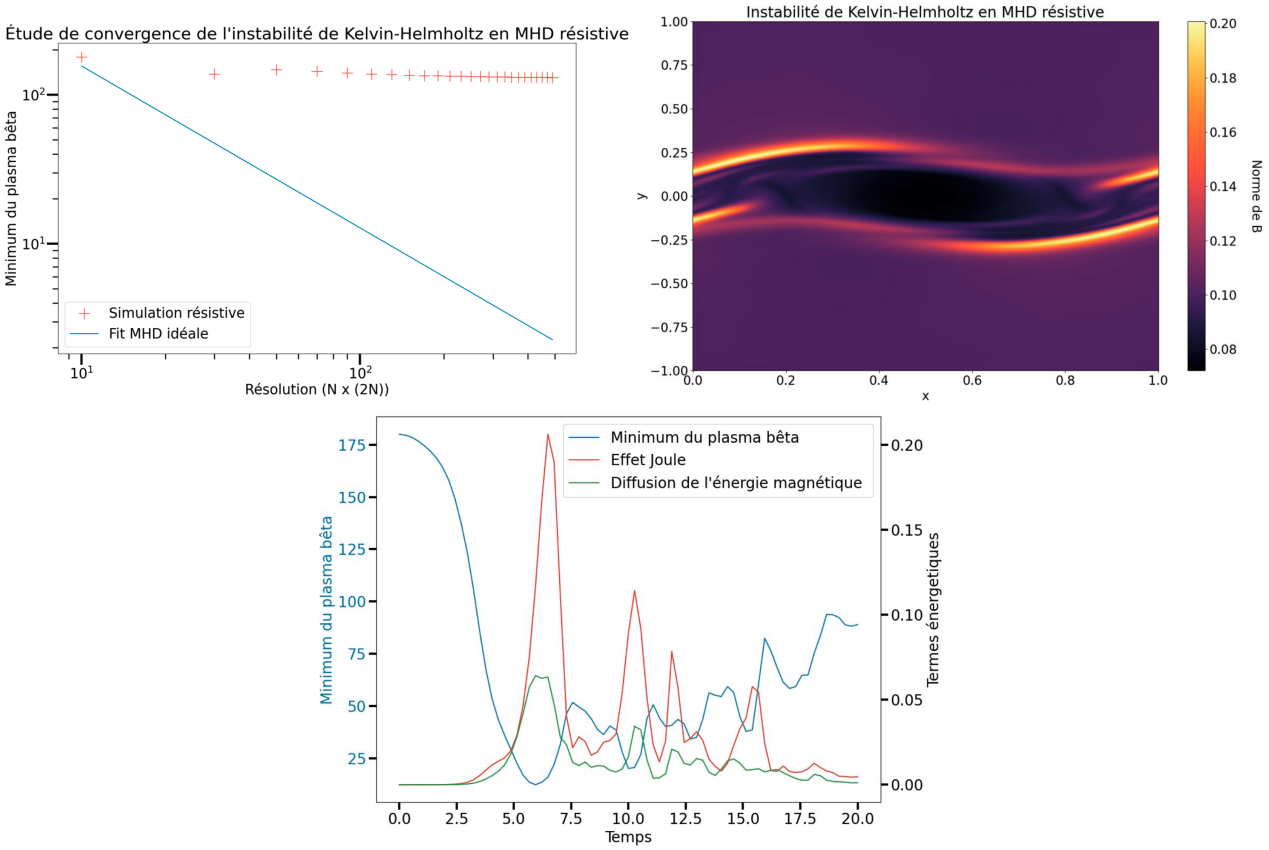


Figure 2.2 – Top left : Minimum value of the plasma beta number reached in a resistive MKHI simulation as a function of the resolution. Top right : Norm of the magnetic field in a resistive MKHI simulation. Bottom : Time series for the minimum value of the plasma beta number, the intensity of the Ohmic heating and magnetic field diffusion for a resistive MKHI simulation. Courtesy of Valentin De Lia.

scheme is cell-centered (unlike the constrained transport), the implementation is straightforward. We perform the extrapolation on the primitive variables $(\rho, p, \mathbf{u}, \mathbf{B})$ and use the classical minmod limiter in order to ensure the admissibility of the Riemann states. A fixed CFL number of 0.8 is used in all simulations with an ideal gas EOS. Note that this is higher than the 0.5 CFL number that is needed to prove the stability of the MUSCL method using a convex combination argument. We invite readers to take advantage of the practical usability of our method beyond it's provably stable conditions. All numerical experiments were conducted using the one step (stencil 1) $5+1$ waves solver with c_a and c_b given by (2.37) to avoid the loss of hyperbolicity of the relaxation whenever B_x or B_y and B_z vanish. On cells where the plasma beta number $\beta = p/\frac{\mathbf{B}^2}{2}$ is inferior to a tunable threshold β_{min}, we locally use the entropic correction 2.53 in order to ensure an entropy satisfying and stable solution, at the cost of the magnetic field conservation. In our experiments, we set $\beta_{min} = 10^{-3}$. We point out that an other choice could be to use a floor value for the internal energy (and density) at the cost of energy (and mass) conservation. This choice is quite often implemented for constrained transport or divergence cleaning schemes (see e.g. [Matsumoto et al. 2019; Dedner et al. 2002]). We also employ the entropic correction term whenever the local Alfvén number $Al = \sqrt{\rho}\frac{|u|}{|B|}$ is superior to another tunable treshold Al_{max} that we set to 10 in our experiments. While it is not crucial for stability, it improves the behavior of the solver in the high Alfvén regime,

see Sect. 2.9.2. Given our threshold choice, the entropic correction is only activated in the specifically designed low-plasma-beta blast problem (see Sect. 2.9.2) and the field loop advection test case (See Sect. 2.9.2).

2.9.1 . 1D tests cases

In this section, we reproduce several 1D Riemann problems that were used in [Bouchut et al. 2010]. The values of the left and right states, the final time, lenght of the domain and adiabatic indexes are given in table (2.9.1). The simulations were all performed with $\Delta x = 10^{-2}$. The reference solutions were all generated with the $5+1$ waves solver using $\Delta x = 5 \times 10^{-4}$.

Test case name, (γ, t_{end}, L)	ρ	(u, v, w)	p	(B_x, B_y, B_z)
Dai & Woodward, $(\frac{5}{3}, 0.2, 1.1)$				
L state	1.08	$(1.2, 0.01, 0.5)$	0.95	$(\frac{4}{\sqrt{4\pi}}, \frac{3.6}{\sqrt{4\pi}}, \frac{2}{\sqrt{4\pi}})$
R state	1.0	$(0.0, 0.0, 0.0)$	1.0	$(\frac{4}{\sqrt{4\pi}}, \frac{4}{\sqrt{4\pi}}, \frac{2}{\sqrt{4\pi}})$
Brio & Wu I, $(2.0, 0.2, 1.0)$				
L state	1.0	$(0.0, 0.0, 0.0)$	1.0	$(0.65, 1.0, 0.0)$
R state	0.125	$(0.0, 0.0, 0.0)$	0.1	$(0.65, -1.0, 0.0)$
Brio & Wu II, $(2.0, 0.012, 1.4)$				
L state	1.0	$(0.0, 0.0, 0.0)$	1000.0	$(0.0, 1.0, 0.0)$
R state	0.125	$(0.0, 0.0, 0.0)$	0.1	$(0.0, -1.0, 0.0)$
Slow rarefaction, $(\frac{5}{3}, 0.2, 1.0)$				
L state	1.0	$(0.0, 0.0, 0.0)$	2.0	$(1.0, 0.0, 0.0)$
R state	0.2	$(1.186, 2.967, 0.0)$	0.1368	$(1.0, 1.6405, 0.0)$
Expansion I, $(\frac{5}{3}, 0.15, 1.4)$				
L state	1.0	$(-3.1, 0.0, 0.0)$	0.45	$(0.0, 0.5, 0.0)$
R state	1.0	$(3.1, 0.0, 0.0)$	0.45	$(0.0, 0.5, 0.0)$
Expansion II, $(\frac{5}{3}, 0.15, 1.4)$				
L state	1.0	$(-3.1, 0.0, 0.0)$	0.45	$(1.0, 0.5, 0.0)$
R state	1.0	$(-3.1, 0.0, 0.0)$	0.45	$(1.0, 0.5, 0.0)$

Dai-Woodward shock tube

This shock tube configuration was introduced in [Woodward et Colella 1984]. During the computation, the solution displays the full eigen-structure of the MHD system as it generates shocks and discontinuities on all fields. We observe in figure 2.3 that our method captures the density and transverse magnetic field robustly, without spurious oscillations. We observe the effect of numerical diffusion smoothing the various waves. A density undershoot is observed at $x \simeq 0.7$ and is due to the choice of CFL number 0.8, higher than what the 0.5 allowed by the stability analysis of MUSCL methods. These results are very similar to what is obtained in [Bouchut et al. 2010].

Brio-Wu shock tube, configuration I

The Brio-Wu shock tube was first introduced in [Brio et Wu 1988]. The solution of this shock tube is composed of shocks, rarefactions, contact discontinuities and a compound wave, in this case a discontinuity attached to a slow rarefaction. In figure 2.4, we can see that our solver captures all features of the solution of this Riemann

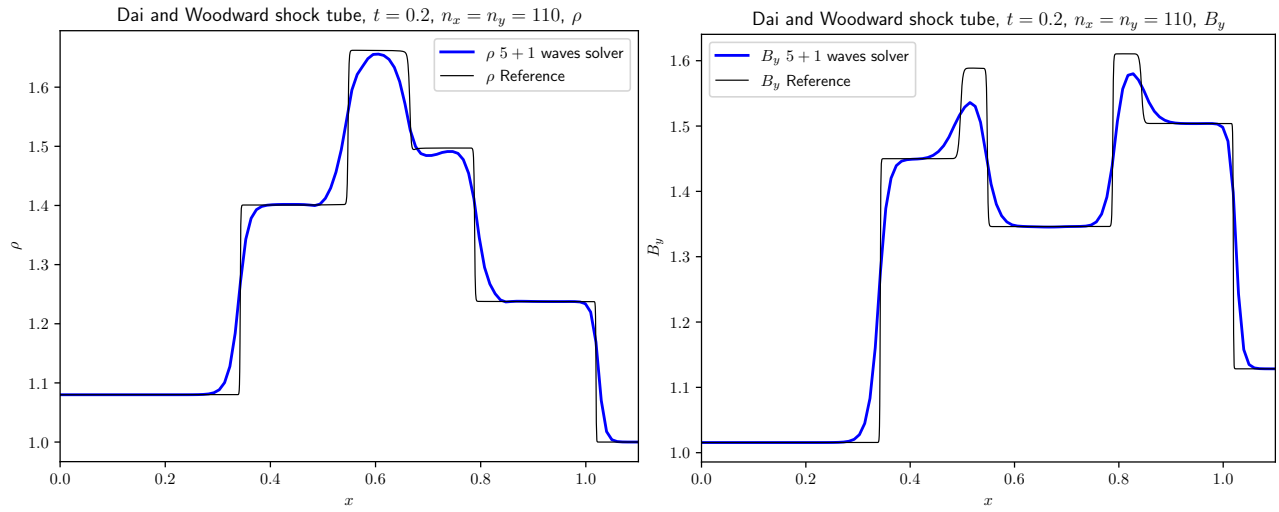


Figure 2.3 – ρ and B_y for the Dai and Woodward shock tube at $t = 0.2$, $5 + 1$ waves solver against a reference solution.

problem. The effect of diffusion is mainly observed on the $x \simeq 0.6$ shock and the density peak around $x \simeq 0.45$ as it is a very fine feature. At the same location, the low-resolution result does present a smoothed bump. These results are very similar to what is obtained in [Bouchut et al. 2010].

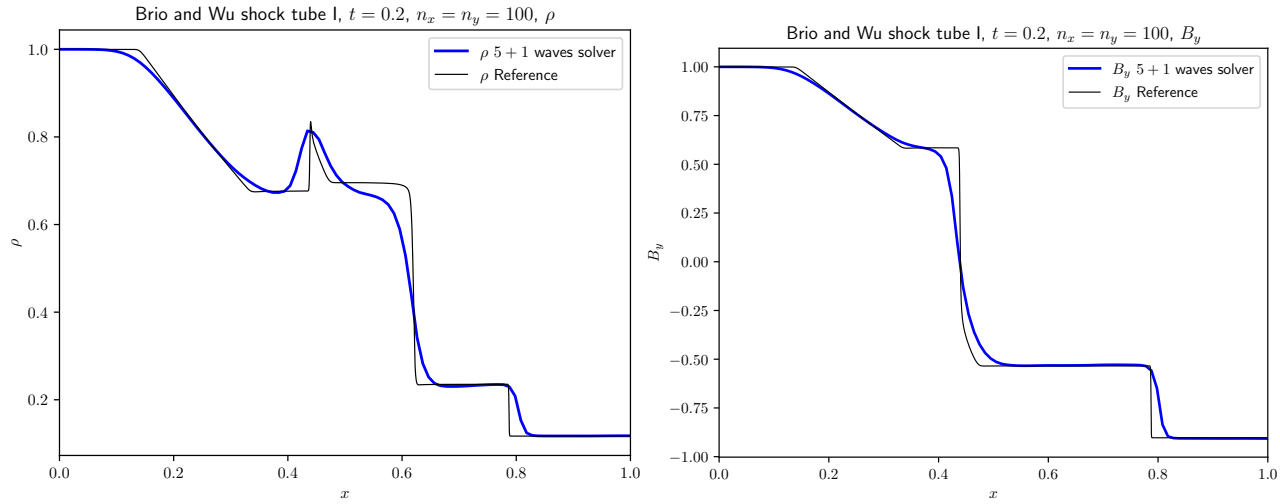


Figure 2.4 – ρ and B_y for the Brio and Wu -I- shock tube at $t = 0.2$, $5 + 1$ waves solver against a reference solution.

Brio-Wu shock tube, configuration II

The second Riemann problem from [Brio et Wu 1988] also involves a complex wave structure but with a high magneto-acoustic Mach number. In figure 2.5, we observe that our solver captures all features of the shock tube, similarly to the results of [Bouchut et al. 2010]. The effect of diffusion is mainly observed at $x \simeq 1.05$ where a

discontinuity and an undershoot are observed on the high resolution plot. This corresponds to the smoothed dip observed in the low-resolution solution.

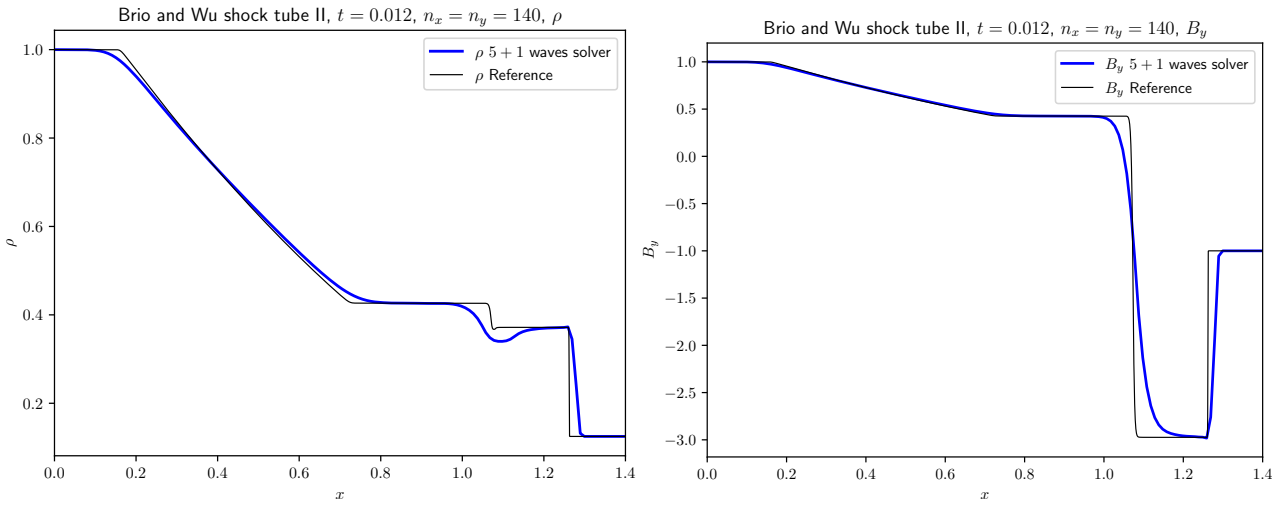


Figure 2.5 – ρ and B_y for the Brio and Wu -II- shock tube at $t = 0.012$, $5 + 1$ waves solver against a reference solution.

Slow rarefaction tube

This test has been first proposed in [Falle et al. 1998]. It involves a sonic point, where the slow magneto-acoustic speed equals the fluid velocity. This feature is problematic for linearized method like the Roe solver, but our scheme is stable as we can see in figure 2.6, just like the resolution shown in [Bouchut et al. 2010]. The $x \simeq 0.75$ dip and $x \simeq 0.85$ bump present on the high-resolution line are smoothed but still present on the low-resolution solution.

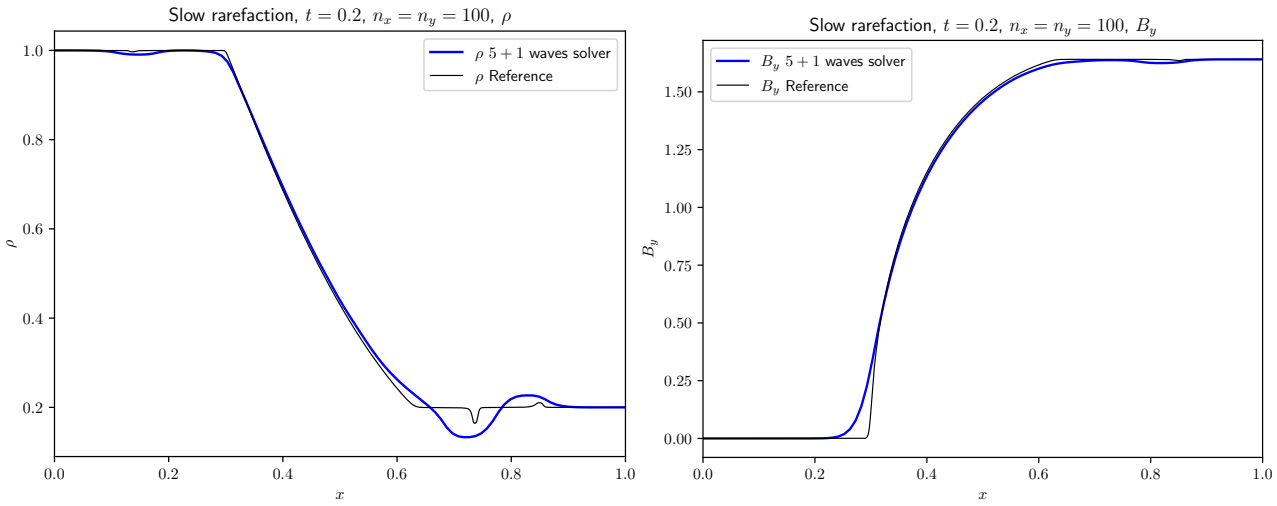


Figure 2.6 – ρ and B_y for the slow rarefaction tube at $t = 0.2$, $5 + 1$ waves solver against a reference solution.

Expansion problem, configuration I

This test is taken from [Miyoshi et Kusano 2005]. It consists of two out-going rarefaction separating a low density region that is difficult to tackle in a stable manner. Our solver is able to simulate this region as we can see in figure 2.7. The effect of numerical diffusion on the sharpness of the $x = 0.5$ density and magnetic field dip is visually enhanced by the use of the log scale. Similar results are found in [Bouchut et al. 2010].

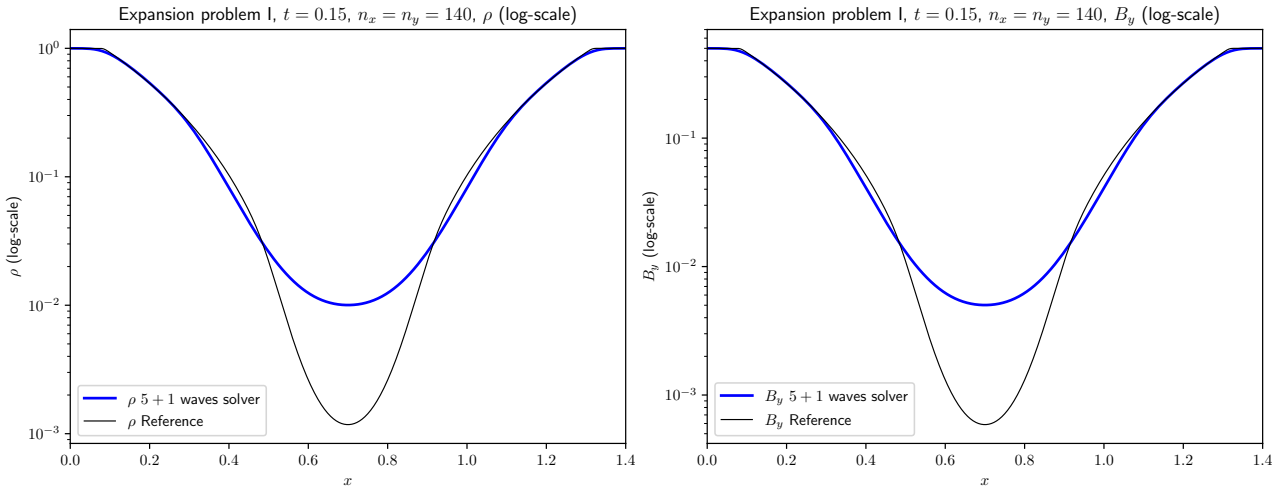


Figure 2.7 – ρ and B_y for the expansion -I- tube at $t = 0.15$, $5 + 1$ waves solver against a reference solution. logscale on the y-axis.

Expansion problem, configuration II

This test is a modification of 2.9.1 suggested by [Bouchut et al. 2010] where we simply set $B_x = 1.0$ instead of 0. Taking B_x nonzero causes the thermal pressure to be low in the central region which can be hard to tackle robustly. Nevertheless, we can see in figure 2.8 that our method is stable and provides results that are very similar to the ones presented in [Bouchut et al. 2010].

2.9.2 . 2D tests cases

All 2D test cases are using $\Delta x = \Delta y = \frac{1}{256}$. We also have tested all the resolutions between 64 and 2048 without any issue to report. In all 2D setups, the quantity r always refers to the distance from the center of the domain.

Orszag-Tang vortex

The Orszag-Tang vortex test case was first introduced by [Orszag et Tang 1979] and has become a standard multi-dimensional benchmark case for ideal MHD. The dynamic of this vortex involves the formation of shocks as well as interactions between them which are challenging to simulate robustly. For instance, 1D solvers like HLLD straightforwardly extended to 2D fail at this task. We recall that this problem takes place in the $[0 : 1]^2$ periodic

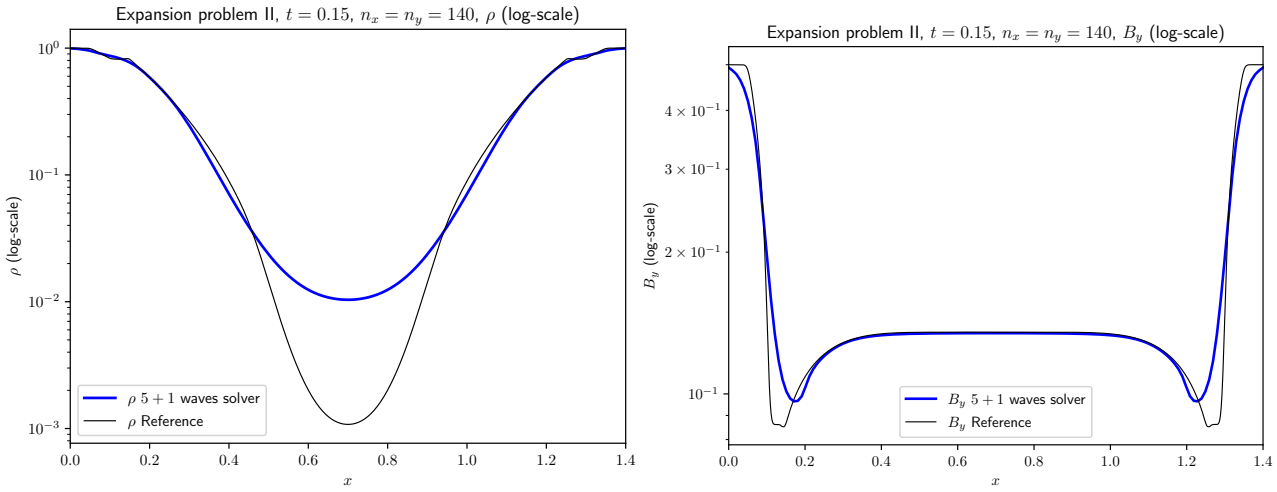


Figure 2.8 – ρ and B_y for the expansion -II- tube at $t = 0.15$, $5 + 1$ waves solver against a reference solution. logscale on the y-axis.

domain with initial data :

$$\begin{aligned}
\rho(x, y) &= \frac{25}{36\pi}, \\
p(x, y) &= \frac{5}{12\pi}, \\
\vec{u}(x, y) &= \begin{pmatrix} -\sin 2\pi y \\ \sin 2\pi x \end{pmatrix}, \\
\vec{B}(x, y) &= \frac{1}{\sqrt{4\pi}} \begin{pmatrix} -\sin 2\pi y \\ \sin 4\pi x \end{pmatrix}, \\
\gamma &= \frac{5}{3}.
\end{aligned}$$

We show the density map at $t = 0.5$ in Figure 2.9. We observe that the shocks and discontinuities are well captured without spurious numerical artifacts. We also notice the usual "eye-shape" high frequency feature at the center of the domain, demonstrating the accuracy of our solver. Note that this test does not show any low β zone. Thus, the solver is fully conservative with respect to $\mathbf{B}$ as the entropic correction terms are never activated.

Rotated shock tube

The rotated shock tube problem has been proposed in [Tóth 2000]. It consists of a 1D shock tube rotated by an angle θ in order to obtain a 2D shock propagation that is not aligned with the grid. The test takes place in the $[0 : 1]^2$ square with Neumann boundary conditions. The setup is given by :

$$\begin{aligned}
\theta &= \arctan(-2), \\
\mathbf{R}(\theta) &= \begin{pmatrix} \sin\theta & \cos\theta \\ \cos\theta & -\sin\theta \end{pmatrix},
\end{aligned}$$

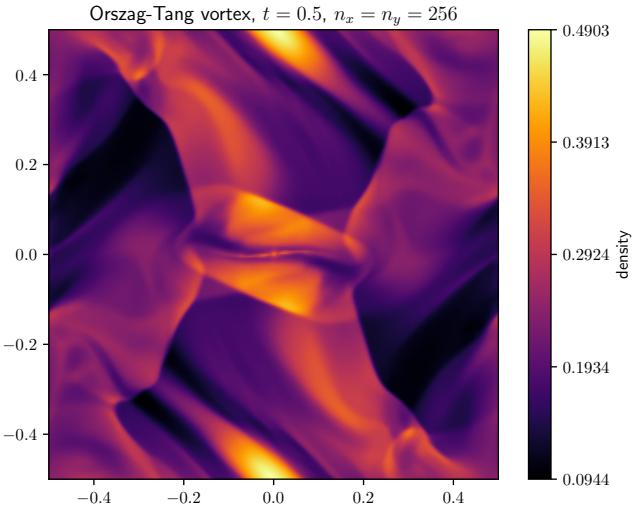


Figure 2.9 – Density map of the Orszag-Tang vortex at $t = 0.5$s

$$
\begin{aligned}
\mathbf{u}_0 &= \begin{pmatrix} 0 \\ 10 \end{pmatrix}, \\
\mathbf{B_0} &= \frac{5}{\sqrt{4\pi}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \\
(x_\theta, y_\theta) &= (\tan\theta(x - 0.5), y - 0.5),
\end{aligned}
$$

$$
\begin{aligned}
\rho(x, y) &= 1, \\
\mathbf{B}(x, y) &= \mathbf{R}(\theta)\mathbf{B_0}, \\
\mathbf{u}(x, y) &= \begin{cases} \mathbf{R}(\theta)\mathbf{u_0} & \text{for } x_\theta < y_\theta, \\ -\mathbf{R}(\theta)\mathbf{u_0} & \text{elsewhere.} \end{cases}, \\
p(x, y) &= \begin{cases} 20 & \text{for } x_\theta < y_\theta, \\ 1 & \text{elsewhere.} \end{cases}.
\end{aligned}
$$

Note that the magnetic field is initialized as a constant on the whole domain, hence the condition $\nabla \cdot \mathbf{B} = 0$ is verified at the beginning of the computation. Our solver is able to robustly and accurately simulate this rotated shock propagation. A quantity of interest in this problem is the component of the magnetic field that is parallel to the shock propagation. Without discretization error, this quantity should remain constant similarly to B_x in a purely 1D setup. In figure 2.10, we show the component of the magnetic field that is parallel to the shock propagation, with both $3 + 1$ and $5 + 1$ solvers. Both schemes produces discretization errors at the location of discontinuities, the errors with the $5 + 1$ waves solver are larger than the errors with the $3 + 1$ waves solver. These errors can be compared with [Tóth 2000] for constrained transport schemes and we point out that the $3 + 1$ and $5 + 1$ waves solvers produce less oscillations around the discontinuities.

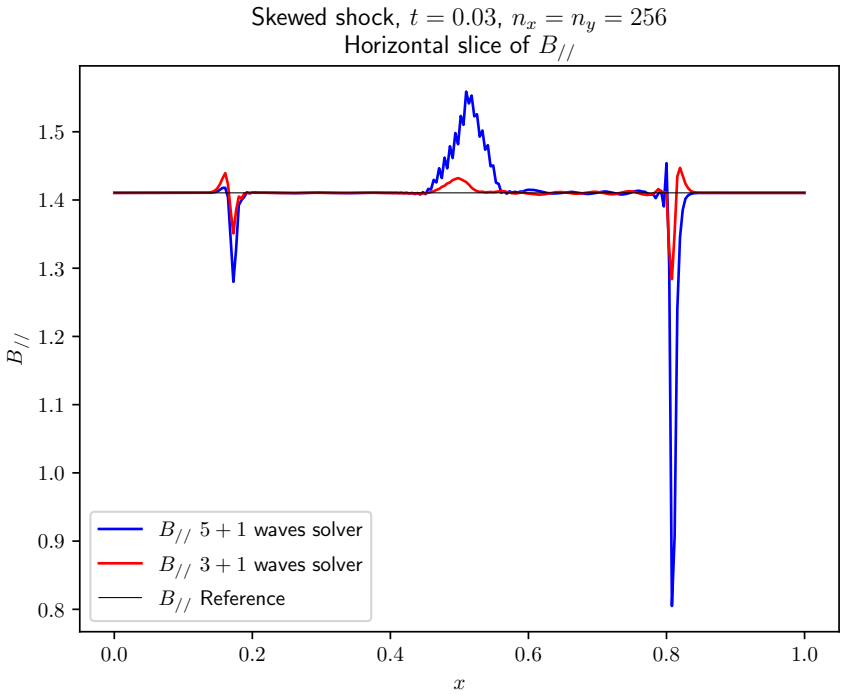


Figure 2.10 – Parallel component of the magnetic field along the rotated shock propagation at t=0.03.

MHD Blast - standard configuration

The Blast test case was introduced in [Stone et Gardiner 2009]. The setup takes place in the periodic $[0:1]^2$ square. A circular region of radius $r_c = 0.1$ is initialized with a greater pression than the rest of the domain. As the computation starts, the blast expands outwards in an elliptical shape due to the presence of a magnetic field. We recall the exact setup :

$$\begin{aligned} p(x,y) &= \begin{cases} 10 & \text{for } r < r_c, \\ 0.1 & \text{for } r \geq r_c, \end{cases} \\ \mathbf{B}(x,y) &= \begin{pmatrix} \sqrt{2\pi} \\ \sqrt{2\pi} \end{pmatrix}, \\ \gamma &= 5/3, \\ \rho(x,y) &= 1, \\ \mathbf{u}(x,y) &= 0. \end{aligned}$$

Our numerical method is able to simulate the expansion of this blast wave accurately and is stable as demonstrated in figure 2.11 where we show the density map at $t = 0.2$. We can see that the expanding wave is well captured. Note that this test does not show any low β zone. Thus, the solver is fully conservative with respect to $\mathbf{B}$.

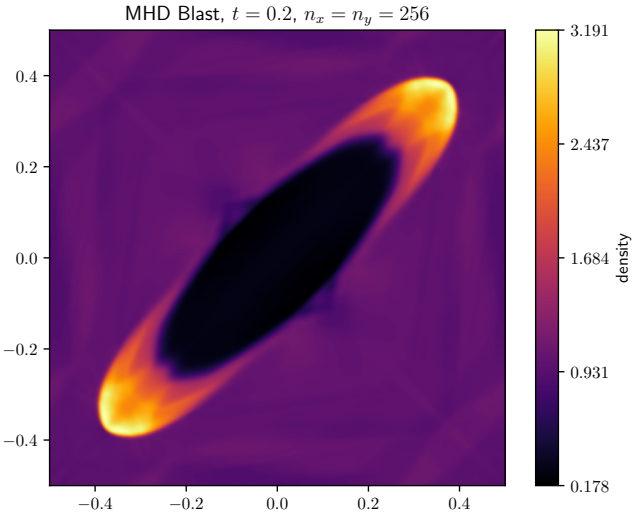


Figure 2.11 – Density map of the MHD Blast at $t = 0.2$s

.

MHD blast - Low β configuration

This test case is inspired from [Balsara 2012]. It consists of the same setup as section 2.9.2 with a lower $\beta \simeq 10^{-6}$:

$$\begin{aligned} p(x,y) &= \begin{cases} 1000 & \text{for } r < r_c, \\ 0.1 & \text{for } r \geq r_c, \end{cases} \\ \mathbf{B}(x,y) &= \begin{pmatrix} 250/\sqrt{2} \\ 250/\sqrt{2} \end{pmatrix}, \\ \gamma &= 1.4, \\ \rho(x,y) &= 1, \\ \mathbf{u}(x,y) &= 0. \end{aligned}$$

The dynamic of the low β blast wave is the same as in 2.9.2 but is harder to tackle as the simulation reaches the limit of the admissibility domain ($e \simeq 0$) and develops strong $\mathbf{B}$ gradients. Note that the $5+1$ wave solver and the constrained transport method [Vides, J. et al. 2013] fail to produce an admissible result as the computation presents negative internal energies (directly after few iterations). We point out that the $5+1$ solver seems, however, more robust than the constrained transport method on such problems : for lower values of the magnetic field $25/\sqrt{2}$, the relaxation solver is stable while the constrained transport method fails after few iterations. It is possible to still get an admissible result by artificially forcing the internal energy to stay above a small threshold (hence loosing energy conservation), a solution used here with the constrained transport method, or by using the entropic correction term (hence loosing the magnetic field conservation), a solution used here with the $5+1$ waves relaxation solver. In figures 2.12, we show the density map of this test case at $t = 0.02$ with our method and the energy-fixed constrained transport solver from the Heracles code [González et al. 2007]. Both methods are able to capture the low β Blast propagation, however, we point out that the $5+1$ waves solver is less diffusing as it reaches higher values for the magnetic field up (+18%).

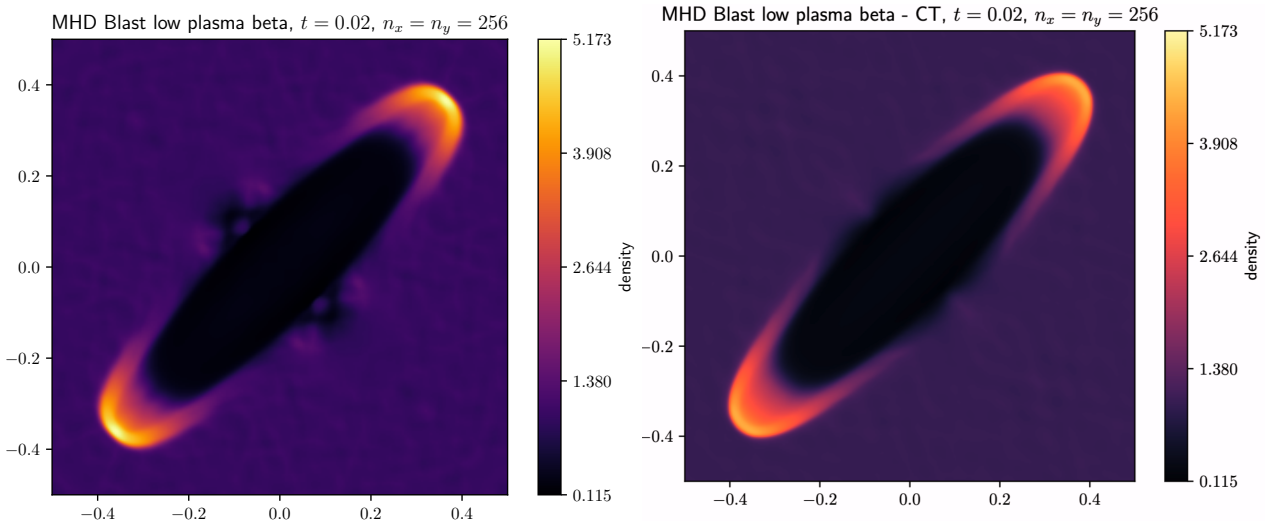


Figure 2.12 – Density map of the low β MHD blast at $t = 0.2$s with our solver and the Heracles code's constrained transport method [González et al. 2007], [Vides, J. et al. 2013].

MHD Rotor

The MHD Rotor test case was first introduced in [Balsara et Spicer 1999]. The setup consists of launching a rapidly spinning cylinder in a light ambient fluid. This rotation sends strong torsional Alfvén waves in the surrounding fluid. We initialize the solution in the $[0:1]^2$ periodic square as following :

$$
\begin{aligned}
p(x,y) &= 1.0, \\
\rho(x,y) &= \begin{cases} 10 & \text{for } r < r_0, \\ 1+9f & \text{for } r \geq r_1 \,\&\, r \leq r_0, \\ 1 & \text{elsewhere} \end{cases} \\
\mathbf{u}(x,y) &= \begin{cases} \frac{u_0}{r_0}\,(0.5-y, x-0.5) & \text{for } r < r_0, \\ \frac{f\,u_0}{r_0}\,(0.5-y, x-0.5) & \text{for } r \geq r_1 \,\&\, r \leq r_0, \\ (0,0) & \text{elsewhere} \end{cases} \\
\mathbf{B}(x,y) &= \begin{pmatrix} 5/\sqrt{4\pi} \\ 0 \end{pmatrix}, \\
\gamma &= 1.4, \\
(r_0, r_1) &= (0.1, 0.115), \\
f &= (r_1 - r)/(r_1 - r_0), \\
u_0 &= 2.
\end{aligned}
$$

We show the result of our simulation in figure 2.13. We observe that the central shear ring as well as the torsional waves are well captured by our solver. Note that this simulation does not require the use of entropic correction terms. Thus, the solver is fully conservative with respect to $\mathbf{B}$.

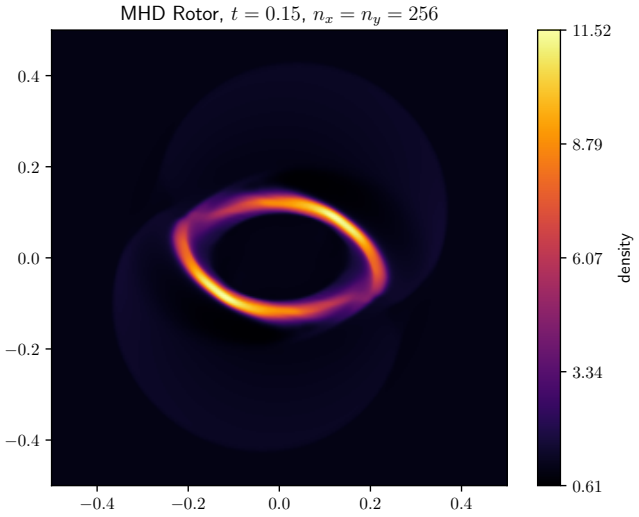


Figure 2.13 – Density map of the MHD Rotor at $t = 0.15$s

Field loop advection

This test was introduced in [Tóth et Odstrčil 1996] and involves advecting a field loop (a cylindrical current distribution) diagonally across the grid. One can choose any arbitrary angle. For the 2D results presented here, the problem domain is defined as $-1 < x < 1$ and $-0.5 < y < 0.5$. The flow has an inclination with $V_x = 2$ and $V_y = 1$. Both the density and pressure are set to 1.0, with the gas constant given by $\gamma = 5/3$. Periodic boundary conditions are applied across the domain. The magnetic field is initialized using an arbitrary vector potential. We set $A_z = \max([A_0(r_0 - r)], 0)$. This results in $(B_x, B_y)(r) = \frac{A_0}{r}(-x, y)$ if $r < r_0$, and $(0, 0)$ otherwise. We chose $A_0 = 0.001$ and set the radius for the loop as $r_0 = 0.3$. After a duration of $t = 2.0s$, the field loop is expected to have been advected and returned to its initial state. The quality of the solution can be assessed by comparing it to the initial solution shown in figure 2.14. The magnetic intensity, defined as $I = \sqrt{B_x^2 + B_y^2}$, obtained with our $5+1$ waves solver, is illustrated in figure 2.15. One can observe that the entropic correction helps with preserving the shape of the cylinder and suppresses the spurious patterns observed with the conservative method. The source terms are activated here as the Alfvén number is above $Al_{max} = 10$ in this test.

2.10 . Conclusion

In this chapter, we have developed a new multi-dimensional, robust, and cell-centered finite volume solver for ideal MHD. The solver is based on the flux splitting and relaxation techniques introduced in chapter 1, and can easily be extended to higher orders because of its reduced stencil. A symmetric version of the solver has been developed by introducing a entropic correction on the induction equation, in order to obtain an entropy-satisfying (but non-conservative for the magnetic field) scheme robust in low plasma beta regions and accurate in high Alfvén number regions. An other solution could be to use a floor value for the internal energy as classically done with constrained transport or divergence cleaning schemes that are not entropy satisfying. We, however, point out that the fully conservative relaxation solver is observed to be more robust than constrained transport schemes on low plasma beta test cases.

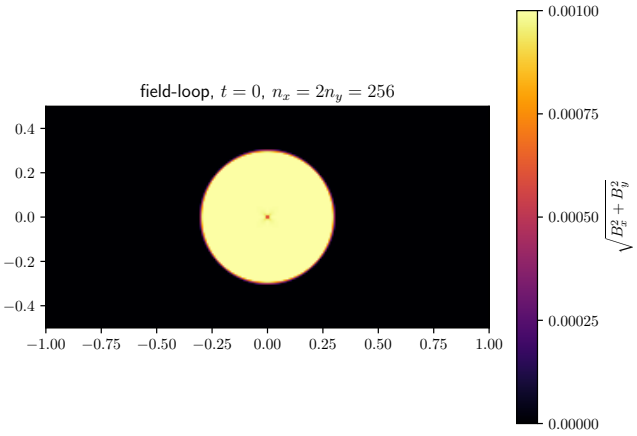


Figure 2.14 – Magnetic intensity of the field loop advection at time $t = 0$.

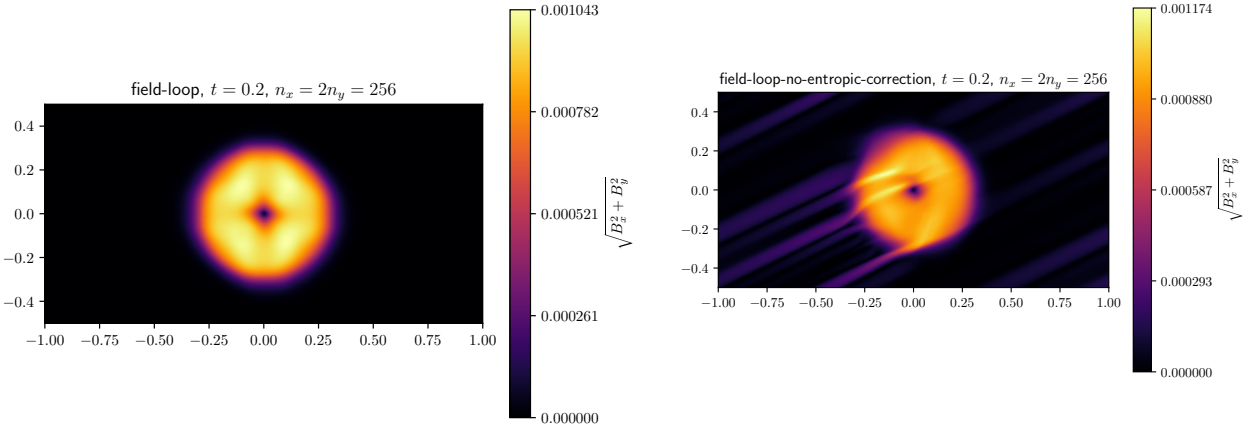


Figure 2.15 – Magnetic intensity of the field loop advection at time $t = 2.0$. Top : With the entropic correction. Bottom : Without the entropic correction.

This cell-centered scheme could be coupled to a divergence cleaning or constrained transport method. We, however, highlight that all the tests we have performed do not seem to require a specific treatment of the divergence of the magnetic field, and a divergence consistent with zero with errors proportional to Δx and Δt at the power of the order of the spatial and temporal reconstructions seem sufficient. It is a common belief that the stability of MHD numerical schemes is closely tied to errors in magnetic field divergence. However, our research, as presented in this chapter, suggests that this may not always be the case. To illustrate, we have successfully designed an entropy-satisfying MHD solver using the symmetric form of MHD equations (incorporating an entropic correction) without specifically addressing divergence issues. Furthermore, we have found that constrained transport schemes, while maintaining zero divergence at machine precision, do not necessarily satisfy entropy conditions and can fail to maintain positive internal energy in areas of low plasma beta.

Additionally, there is a prevalent view that errors in magnetic divergence significantly impact the physical accuracy of simulations, potentially leading to artificial magnetic monopoles. We offer several arguments to challenge this perspective. Even in constrained transport schemes, certain terms involving divergence in the conservative

forms of the Lorentz force and the energy evolution equation do not achieve zero at machine precision, despite a zero divergence. These residual terms in the entropy evolution equation are indeed the reason why constrained transport schemes are not entropy satisfying. Moreover, it can be demonstrated that constrained transport schemes are not immune to divergence errors. For example, the rotated shock tube case detailed in [Tóth 2000] shows that at the continuous level, a zero divergence equates to a constant magnetic field parallel to the shock tube. However, constrained transport schemes do not maintain this constant magnetic field at machine precision, thus resulting in "divergence errors" that are significant for the physics at play.

In conclusion, while ensuring zero magnetic divergence at machine precision in simulations is physically relevant, this is only feasible when aligning the grid to a specific magnetic field configuration. This issue is akin to preserving angular momentum in a rotating structure, achievable at machine precision only in a polar grid. Consequently, for simulations with highly dynamic magnetic fields, maintaining zero divergence at machine precision on a Cartesian grid may not be as critical with a solver that is entropy-satisfying.

The MHD relaxation solver presented in this chapter is a direct extension of the one developed for the Euler equations in chapter 1 and can be implemented in a one-step flux-update algorithm, that can easily be extended to higher orders and to non-ideal MHD. Because of its simplicity, this solver should also have improved performances compared to other multi-dimensional MHD solvers (constrained transport and divergence cleaning) and offers interesting possibilities for large-scale physical applicationes on the next generation of exascale supercomputers. The solver has been used for all numerical simulations in the next chapter which focuses on the convective instability in MHD, including a very high resolution simulation. It is worth noting that the convective regime we study is characterized by a high plasma beta number, so that the conservative version of the method can be employed safely.

Appendix

2.A . Useful vector identities

2.A.1 . Lorentz's force in conservative form

The first identity we derive is

$$\boldsymbol{j} \times \boldsymbol{B} = -(\nabla \cdot \boldsymbol{B})\boldsymbol{B} - \nabla \cdot \left(\frac{\boldsymbol{B}^2}{2} \boldsymbol{I} - \boldsymbol{B} \otimes \boldsymbol{B} \right). \tag{2.54}$$

We only verify this equality for the x component as the relationship for the two other components are checked by rotational invariance. We have $\boldsymbol{j} \times \boldsymbol{B} = (\nabla \times \boldsymbol{B}) \times \boldsymbol{B}$. Expanding the first component, we get $[(\nabla \times \boldsymbol{B}) \times \boldsymbol{B}]_x = B_z \left(\partial_z B_x - \partial_x B_z \right) - B_y \left(\partial_x B_y - \partial_y B_x \right)$. Moreover, $[\nabla \cdot \left(\frac{\boldsymbol{B}^2}{2} \boldsymbol{I} \right)]_x = B_x \partial_x B_x + B_y \partial_x B_y + B_z \partial_x B_z$. Lastly, $[\nabla \cdot (\boldsymbol{B} \otimes \boldsymbol{B})]_x = (\nabla \cdot \boldsymbol{B}) B_x + B_x \partial_x B_x + B_y \partial_y B_x + B_z \partial_z B_x$. Collecting the right hand side terms, we get $-(\nabla \cdot \boldsymbol{B}) B_x - B_x \partial_x B_x - B_y \partial_x B_y - B_z \partial_x B_z + (\nabla \cdot \boldsymbol{B}) B_x + B_x \partial_x B_x + B_y \partial_y B_x + B_z \partial_z B_x$ where both the terms proportional to the divergence of $\boldsymbol{B}$ and $B_x \partial_x B_x$ cancel out and provide the desired result.

2.A.2 . Fully developed Lorentz force

Using $\nabla \cdot (\boldsymbol{B} \otimes \boldsymbol{B}) = \boldsymbol{B}(\nabla \cdot \boldsymbol{B}) + (\boldsymbol{B} \cdot \nabla)\boldsymbol{B}$, we get :

$$\boldsymbol{j} \times \boldsymbol{B} = (\boldsymbol{B} \cdot \nabla)\boldsymbol{B} - \nabla \left(\frac{\boldsymbol{B}^2}{2} \right) \tag{2.55}$$

2.A.3 . Curl of a cross product

$$\nabla \times (\boldsymbol{u} \times \boldsymbol{B}) = \nabla \cdot (\boldsymbol{B} \otimes \boldsymbol{u} - \boldsymbol{u} \otimes \boldsymbol{B}). \tag{2.56}$$

$$\nabla \times (\boldsymbol{u} \times \boldsymbol{B}) = \boldsymbol{u}(\nabla \cdot \boldsymbol{B}) - \boldsymbol{B}(\nabla \cdot \boldsymbol{u}) + (\boldsymbol{B} \cdot \nabla)\boldsymbol{u} - (\boldsymbol{u} \cdot \nabla)\boldsymbol{B} \tag{2.57}$$

2.A.4 . Transport of a squared quantity

$$((\boldsymbol{u} \cdot \nabla)\boldsymbol{A}) \cdot \boldsymbol{A} = (\boldsymbol{u} \cdot \nabla)\frac{\boldsymbol{A}^2}{2} = \nabla \left(\frac{\boldsymbol{A}^2}{2} \right) \cdot \boldsymbol{u} \tag{2.58}$$

2.B . Deriving the conservative MHD equations

In this section our goal is to go from the non conservative MHD system :

$$\begin{aligned} \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) &= 0, \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) &= -\nabla p + \mathbf{j} \times \mathbf{B}, \\ \partial_t (\rho e) + \nabla \cdot (\rho e \mathbf{u}) &= -p \nabla \cdot \boldsymbol{u}, \\ \partial_t \mathbf{B} - \nabla \times (\boldsymbol{u} \times \boldsymbol{B}) &= 0. \end{aligned} \tag{2.59}$$

to the conservative MHD system.

$$\begin{aligned}\partial_t\rho + \nabla\cdot(\rho\mathbf{u}) &= 0,\\ \partial_t(\rho\mathbf{u}) + \nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u} + \sigma - \mathbf{B}\otimes\mathbf{B}) &= 0,\\ \partial_t(\rho E) + \nabla\cdot(\rho E\mathbf{u} + \sigma\mathbf{u} - (\mathbf{B}\cdot\mathbf{u})\mathbf{B}) &= 0,\\ \partial_t\mathbf{B} + \nabla\cdot(\mathbf{u}\otimes\mathbf{B} - \mathbf{B}\otimes\mathbf{u}) &= 0.\end{aligned} \tag{2.60}$$

Where $e_{mag} = \frac{\boldsymbol{B}^2}{2\rho}$ and $\sigma = p + \frac{\boldsymbol{B}^2}{2}$ Obtaining the conservative momentum equation is straightforward using (2.54), substituting for $\mathbf{j}\times\mathbf{B}$ and assuming $\nabla\cdot\boldsymbol{B} = 0$. Obtaining the conservative induction equation is also straightforward using (2.56) (note that using the $\nabla\cdot\boldsymbol{B} = 0$ hypothesis is not necessary to obtain the induction equation). This leaves us with deriving the total energy equation.

Kinetic energy evolution equation

From the non conservative momentum equation, we can deduce the evolution equation of the velocity $\partial_t\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \boldsymbol{\nabla p}/\rho = \boldsymbol{j}\times\boldsymbol{B}/\rho$. Dotting this equation against $\boldsymbol{u}$, we get $\partial_t\left(\frac{\boldsymbol{u}^2}{2}\right) + ((\boldsymbol{u}\cdot\nabla)\boldsymbol{u})\cdot\boldsymbol{u} + \boldsymbol{\nabla p}\cdot\boldsymbol{u}/\rho = (\boldsymbol{j}\times\boldsymbol{B})\cdot\boldsymbol{u}/\rho$. Using (2.58), we have that $((\boldsymbol{u}\cdot\nabla)\boldsymbol{u})\cdot\boldsymbol{u} = (\boldsymbol{u}\cdot\nabla)\left(\frac{\boldsymbol{u}^2}{2}\right)$. Substituting this transport term and multiplying by ρ, we get $\rho\partial_t\left(\frac{\boldsymbol{u}^2}{2}\right) + \rho(\boldsymbol{u}\cdot\nabla)\left(\frac{\boldsymbol{u}^2}{2}\right) + \boldsymbol{\nabla p}\cdot\boldsymbol{u} = (\boldsymbol{j}\times\boldsymbol{B})\cdot\boldsymbol{u}$. Adding $\frac{\boldsymbol{u}^2}{2}(\partial_t\rho + \nabla\cdot(\rho\boldsymbol{u})) = 0$, we get : $\partial_t(\rho\frac{\boldsymbol{u}^2}{2}) + \rho(\boldsymbol{u}\cdot\nabla)\left(\frac{\boldsymbol{u}^2}{2}\right) + \frac{\boldsymbol{u}^2}{2}\nabla\cdot(\rho\boldsymbol{u}) + \boldsymbol{\nabla p}\cdot\boldsymbol{u} = (\boldsymbol{j}\times\boldsymbol{B})\cdot\boldsymbol{u}$. Since $\rho(\boldsymbol{u}\cdot\nabla)\left(\frac{\boldsymbol{u}^2}{2}\right) + \frac{\boldsymbol{u}^2}{2}\nabla\cdot(\rho\boldsymbol{u}) = \nabla\cdot\left(\frac{\rho\boldsymbol{u}^2\boldsymbol{u}}{2}\right)$, noting $e_{kin} = \frac{\boldsymbol{u}^2}{2}$, we get :

$$\partial_t(\rho e_{kin}) + \nabla\cdot(\rho e_{kin}\boldsymbol{u}) + \boldsymbol{\nabla p}\cdot\boldsymbol{u} = (\boldsymbol{j}\times\boldsymbol{B})\cdot\boldsymbol{u}. \tag{2.61}$$

Summing this with the internal energy evolution equation, we get : $\partial_t(\rho(e+e_{kin})) + \nabla\cdot(\rho(e+e_{kin})\boldsymbol{u} + p\boldsymbol{u}) = (\boldsymbol{j}\times\boldsymbol{B})\cdot\boldsymbol{u}$. Replacing the right hand side using (2.55), we get :

$$\partial_t(\rho(e+e_{kin})) + \nabla\cdot(\rho(e+e_{kin})\boldsymbol{u} + p\boldsymbol{u}) = ((\boldsymbol{B}\cdot\nabla)\boldsymbol{B})\cdot\boldsymbol{u} - \nabla\left(\frac{\boldsymbol{B}^2}{2}\right)\cdot\boldsymbol{u}. \tag{2.62}$$

Magnetic energy evolution equation

Using the identity 2.57, we get $\partial_t\boldsymbol{B} - \boldsymbol{u}(\nabla\cdot\boldsymbol{B}) + \boldsymbol{B}(\nabla\cdot\boldsymbol{u}) - (\boldsymbol{B}\cdot\nabla)\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{B} = 0$. Dotting against $\boldsymbol{B}$, we get

$$\partial_t(\rho e_{mag}) - (\nabla\cdot\boldsymbol{B})(\boldsymbol{u}\cdot\boldsymbol{B}) + (\nabla\cdot\boldsymbol{u})\boldsymbol{B}^2 - ((\boldsymbol{B}\cdot\nabla)\boldsymbol{u})\cdot\boldsymbol{B} + ((\boldsymbol{u}\cdot\nabla)\boldsymbol{B})\cdot\boldsymbol{B} = 0. \tag{2.63}$$

Total energy evolution equation

Summing (2.62) and (2.63), we get $\partial_t(\rho E) + \nabla\cdot(\rho(e+e_{kin})\boldsymbol{u} + p\boldsymbol{u}) = ((\boldsymbol{B}\cdot\nabla)\boldsymbol{B})\cdot\boldsymbol{u} - \nabla\left(\frac{\boldsymbol{B}^2}{2}\right)\cdot\boldsymbol{u} + (\nabla\cdot\boldsymbol{B})(\boldsymbol{u}\cdot\boldsymbol{B}) - (\nabla\cdot\boldsymbol{u})\boldsymbol{B}^2 + ((\boldsymbol{B}\cdot\nabla)\boldsymbol{u})\cdot\boldsymbol{B} - ((\boldsymbol{u}\cdot\nabla)\boldsymbol{B})\cdot\boldsymbol{B}$. Using (2.58), we have $-((\boldsymbol{u}\cdot\nabla)\boldsymbol{B})\cdot\boldsymbol{B} - \nabla\left(\frac{\boldsymbol{B}^2}{2}\right)\cdot\boldsymbol{u} = \nabla(\boldsymbol{B}^2)\cdot\boldsymbol{u}$. Moreover, Since $\nabla(\boldsymbol{B}^2)\cdot\boldsymbol{u} + (\nabla\cdot\boldsymbol{u})\boldsymbol{B^2} = \nabla\cdot(\boldsymbol{B}^2\,\boldsymbol{u}) = \nabla\cdot(\rho e_{mag}\boldsymbol{u} + \boldsymbol{B}^2/2\,\boldsymbol{u})$, we can show that :

$$\partial_t(\rho E) + \nabla\cdot(\rho E\boldsymbol{u} + \sigma\boldsymbol{u}) = ((\boldsymbol{B}\cdot\nabla)\boldsymbol{B})\cdot\boldsymbol{u} + (\nabla\cdot\boldsymbol{B})(\boldsymbol{u}\cdot\boldsymbol{B}) + ((\boldsymbol{B}\cdot\nabla)\boldsymbol{u})\cdot\boldsymbol{B}. \tag{2.64}$$

As $((\boldsymbol{B}\cdot\nabla)\boldsymbol{B})\cdot\boldsymbol{u}+((\boldsymbol{B}\cdot\nabla)\boldsymbol{u})\cdot\boldsymbol{B}=(\boldsymbol{B}\cdot\nabla)(\boldsymbol{u}\cdot\boldsymbol{B})=\nabla(\boldsymbol{B}\cdot\boldsymbol{u})\cdot\boldsymbol{B}$ and $\nabla(\boldsymbol{B}\cdot\boldsymbol{u})\cdot\boldsymbol{B}+(\nabla\cdot\boldsymbol{B})(\boldsymbol{u}\cdot\boldsymbol{B})=\nabla\cdot((\nabla\cdot\boldsymbol{B})\cdot\boldsymbol{B})$, we get the desired result. Note that it is not required to assume $\nabla\cdot\boldsymbol{B}=0$ to obtain the conservative total energy equation.

2.C . Entropy inequality of the corrected system

2.C.1 . Entropy inequality of the non conservative MHD system

We start with the classical result of the entropy inequality of the MHD system (2.59), starting from the evolution equation of the internal energy. We note $D_t=\partial_t+u\nabla$. We have $D_te=-p(\nabla\cdot\boldsymbol{u})\tau$ where $\tau=1/\rho$. From the density evolution equation, we have that $D_t\tau=\tau(\nabla\cdot\boldsymbol{u})$. Therefore, $D_te+pD_t\tau=0$. Using the first principle of thermodynamics $de+pd\tau=Tds$, we get

$$D_ts=0. \tag{2.65}$$

2.C.2 . Entropy inequality of the conservative MHD system

To go from the non conservative system to the conservative system, we only had to cancel one term in the momentum equation, using the $\nabla\cdot\boldsymbol{B}=0$ hypothesis. This means that if we are discretizing the conservative momentum equation and that the numerical value of the divergence is not zero, were are in fact discretizing $\partial_t(\rho\mathbf{u})+\nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u})=-\nabla p+\mathbf{j}\times\mathbf{B}+(\nabla\cdot\boldsymbol{B})\boldsymbol{B}$. We want to derive the corresponding internal energy equation. We dot the momentum equation against $\boldsymbol{u}$ and subtract it to the conservative total energy equation. Doing this, we get $\partial_t(\rho e)+\nabla\cdot(\rho e\mathbf{u})=-p\nabla\cdot\boldsymbol{u}-(\nabla\cdot\boldsymbol{B})(\boldsymbol{B}\cdot\boldsymbol{u})$. Performing the same steps as above, we get $D_te+pD_t\tau=-\tau(\nabla\cdot\boldsymbol{B})(\boldsymbol{B}\cdot\boldsymbol{u})$ thus :

$$D_ts=-\frac{\tau}{T}(\nabla\cdot\boldsymbol{B})(\boldsymbol{B}\cdot\boldsymbol{u}) \tag{2.66}$$

2.C.3 . Entropy inequality of the conservative MHD system with the entropic correction

The strategy we propose in this chapter is to discretize

$$\partial_t\mathbf{B}+\nabla\cdot(\mathbf{u}\otimes\mathbf{B}-\mathbf{B}\otimes\mathbf{u})+\mathbf{u}\nabla\cdot\mathbf{B}=0. \tag{2.67}$$

instead of the conservative MHD system. To obtain the corresponding internal energy equation, we dot this equation against $\boldsymbol{B}$ and subtract it to the total energy equation, along with the kinetic energy equation. It is clear to see that the terms proportional to the divergence of $\boldsymbol{B}$ will cancel out and provide the standard internal energy equation, and a source-term-free entropy inequality. tion, and a source-term-free entropy inequality.

3 - Magneto-thermo-compositional sheared diabatic convection. Linear stability analysis, non-linear extension and numerical experiments

3.1 . Introduction

As the building block for Mixing Length Theory (MLT), linear stability analysis for convection plays an important role in the understanding of the structures of planets and stars across the universe. Instability criteria allow us to make both qualitative and quantitative arguments on the predominant physical processes that shape oceans [Timmermans et al. 2008; Gregg 1988], as well as the atmospheres of planets, stars, and the Earth [Ulrich 1972; Arakawa et Jung 2011], [Manabe et Strickler 1964; Tremblin et al. 2015; Tremblin et al. 2016; Tremblin et al. 2017; Denissenkov 2010; Wachlin et al. 2014; Stevens 2005; Zemskova et al. 2014]. The earliest stability analysis for convection was done in [Schwarzschild 1906] and quantifies how important the temperature gradient has to be compared to the adiabatic gradient of the fluid to allow thermal convection. In [Ledoux 1947], a mean molecular weight gradient was added to the analysis, extending Schwarzschild's result to thermo-compositional flows. Moreover, the influence of the magnetic field on the Schwarzschild criterion was intensively studied to understand better the structure of magnetic stars's atmospheres and sunspots in particular (see [Hughes et Proctor 1988],[Gough et Tayler 1966; Tayler 1973; Newcomb 1961; Kovetz et Mestel 1967; Yu 1966; Chandrasekhar 1961]). These studies conclude that the magnetic field has an inhibiting effect on convection but that it cannot completely stabilize a configuration that is Schwarzschild unstable. The role of source terms (such as thermal/chemical diffusion) was explored later. For instance, we refer to the early work of [Stern 1960] describing thermohaline convection with the difference in thermal and salinity diffusive time scales. Such processes are now referred to as double-diffusive convection. They are actively studied as they can occur in many physical circumstances when two gradients associated with density differences and diffusivities are at play (see, for instance, [Turner 1974; Brandt et Fernando 1995; Radko 2013; Huppert et Turner 1981; Garaud 2013, 2018; Radko 2014; Baines et Gill 1969; Stellmach et al. 2011]). Thermo-magnetic double-diffusive convection was also explored in [Yu et Cheng 1973]. In [Tremblin et al. 2019], an unifying framework for thermo-compositional convection was proposed. The analysis allows us to systematically derive instability criteria for hydrodynamical convection, encapsulating thermohaline convection in Earth's oceans, fingering convection in stellar atmospheres, moist convection in Earth's atmosphere, as well as radiative convection triggered by CO/CH_4 transition with radiative transfer in the atmospheres of brown dwarfs in one unique formalism. This chapter aims to extend this approach to plasmas and sheared flows by introducing a background horizontal magnetic field as well as a background horizontal velocity gradient to the analysis. As a result, we obtain three criteria for instability. The first one generalizes adiabatic instability criteria, such as Schwarzschild and Ledoux. The second one combines source terms and gradients, encapsulating most double-diffusive convection from the literature. The third criterion is new and involves products of pairs of source terms with the gradients. We discuss and observe similarities between the effect of shear and magnetic field on convective instability. We discuss how our criteria reduce to the ones from the literature when adequate terms are removed from our analysis to consider the corresponding subcases. A non-linear extension of the theory is proposed and provides us with rudimentary estimations for the various convection-related quantities in the saturated regime. We perform finite volume numerical experiments in the linear regime to validate our analysis.

The numerical method we use is the one of the last chapter along with the well-balanced treatment of gravity from chapter 1, see appendix 3.B. In particular, we conducted several parametric studies to check and control the existence of the new instability that we discovered and to verify the influence of shear and magnetic fields on convective instability behaves according to our theory. We also conduct several simulations beyond the linear regime of the instability to study the self-generation of shear and magnetic fields in convective flows. In particular, we highlight the importance of geometry in the intensity of the growth of shear modes in the non-linear regime in both 2D and 3D. Our findings imply that convection in cubic domains is not strongly affected by shear, while elongated domains of aspect ratios such as 2 :1 :1 present strongly sheared flows. Then, we perform several convective dynamo simulations in a cubic domain (to avoid the interaction of the magnetic field with shear) and link the intensity of the self-generation of magnetic energy to the non-linear theory we derived, cross-validating both approaches. We have not studied the interaction of the shear and magnetic fields in the present chapter, leaving this tedious task for future work. Checking the numerical influence of each variable on convection for all three criteria, in both the linear and non-linear regimes, would not be tractable. Instead, we focus on a handful of specific sub-cases, in both the linear and non-linear regime, to highlight our approach's validity and provide interesting estimations for classically difficult problems, e.g., convective dynamo. The potential applications of our analysis are broad, but we do not perform any mapping between the theory and applications here. The interest of the present work is its flexibility, and it should give the interested reader the tool to perform the mapping with his own convective application and potentially discover unexpected instabilities and/or refine their non-linear estimations.

3.2 . Linear regime

3.2.1 . Linearization of the equation system

In this section, we expand the linear stability analysis for (a)diabatic convection discussed in [Tremblin et al. 2019] to the context of sheared and magnetized fluids. We conduct a linear stability analysis under the Boussinesq approximation, including arbitrary source terms, to derive the criteria for convective instability. The analysis begins with the ideal MHD (MHD) equations, accounting for gravitational, compositional, thermal, and magnetic source terms :

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) &= 0, \\
\frac{\partial \rho u}{\partial t} + \nabla \cdot (\boldsymbol{u} \otimes \rho u - \boldsymbol{B} \otimes \boldsymbol{B}) + \nabla \left(P + \frac{1}{2}\boldsymbol{B}^2 \right) &= \rho \boldsymbol{g}, \\
\frac{\partial \rho \mathcal{E}}{\partial t} + \nabla \cdot \left(\left(\rho \mathcal{E} + P + \frac{1}{2}\boldsymbol{B}^2 \right) \boldsymbol{u} - (\boldsymbol{B} \cdot \boldsymbol{u}) \boldsymbol{B} \right) & \\
= \rho c_v \gamma \left(H - T \frac{\partial \log \mu}{\partial X} R \right) + \boldsymbol{B} \cdot \nabla \times \boldsymbol{Q}, & \\
\frac{\partial \boldsymbol{A}}{\partial t} + (\boldsymbol{u} \cdot \nabla) \boldsymbol{A} &= \boldsymbol{Q}, \\
\frac{\partial \rho X}{\partial t} + \nabla \cdot (\boldsymbol{u} \rho X) &= \rho R.
\end{aligned} \tag{3.1}$$

Where ρ is the fluid's density, $\boldsymbol{u}$ the velocity vector, $\mathcal{E} = e + \frac{\boldsymbol{u}^2}{2} + \frac{\boldsymbol{B}^2}{2\rho} + \phi$ the total energy, e the specific internal energy, ϕ the gravitational potential energy, $\boldsymbol{B}$ the magnetic field linked to the potential vector $\boldsymbol{A}$ by

$\boldsymbol{B} = \nabla \times \boldsymbol{A}$, $\boldsymbol{g} = -g\boldsymbol{e_z}$ the gravity vector with $g > 0$, $X \in [0, 1]$ the mass mixing ratio of an arbitrary chemical component transported by the fluid, P the pressure linked to the temperature T and mean molecular weight μ by the ideal EOS : $P = \rho k b T/\mu(X)$. The sources $H, \boldsymbol{Q}, R$ can model any non-hyperbolic terms on the temperature, magnetic potential vector, and composition. They all are functions of $T, X, P, \boldsymbol{A}$. H can model external heating, pumping and/or thermal diffusion, and Ohmic heating. $\boldsymbol{Q}$ can model magnetic resistivity and/or other non-ideal MHD effects. R can model chemical diffusion and/or reactions. The potential vector equation is derived from the induction equation in appendix 3.D and requires a gauge choice. However, the criteria we derive only feature the magnetic field, making them independent of the gauge choice. We emphasize that this change of variable is crucial to the feasability of the analysis. Following the lines of [Kato 1966], the energy and composition equations from (3.1) are re-written in terms of potential temperature $\theta = T(P_{\text{ref}}/P)^{\frac{\gamma-1}{\gamma}}$ and mass mixing ratio. The source term $\rho c_v \gamma \left(H - T\frac{\partial \log \mu}{\partial X} R\right)$ on the total energy is there to ensure that we obtain the following evolution equation of the potential temperature (see appendix 3.C for the derivation) :

$$\begin{aligned} &\frac{\partial \log \theta}{\partial t} + \boldsymbol{u} \cdot \nabla \log \theta = \frac{H}{T}, \\ &\frac{\partial X}{\partial t} + u \cdot \nabla X = R. \end{aligned} \tag{3.2}$$

The $\boldsymbol{B} \cdot \nabla \times \boldsymbol{Q}$ term ensures that the source term $\boldsymbol{Q}$ does not affect the internal energy; heating caused by the magnetic field should be included in a $\boldsymbol{B}$ dependancy of the heating source term H. The resulting system is linearized by rewriting all quantities as $q = q_0 + \delta q$ with δq representing the perturbation, assuming $\delta q^2 \simeq 0$ and q_0 being the background state The background state is a stationary solution of 3.1. It depends only on the altitude z and is defined by :

$$\begin{aligned} &\nabla \left(P_0 + \frac{1}{2}\boldsymbol{B}_0(z)^2\right) = \rho_0 \boldsymbol{g}, \\ &H(q_0) = R(q_0) = \boldsymbol{Q}(q_0) = 0, \\ &\boldsymbol{u}_0(z) = u_0(z)\boldsymbol{e_x}, \\ &\boldsymbol{A}_0(z) = A_0(z)\boldsymbol{e_y}, \\ &\boldsymbol{B}_0(z) = B_0(z)\boldsymbol{e_x} = -\frac{\partial A_0(z)}{\partial z}\boldsymbol{e_x}. \end{aligned} \tag{3.3}$$

Note that the orientation of $\boldsymbol{A}_0$ is chosen along y, imposing $\boldsymbol{B}_0$ along x, i.e., a horizontal magnetic field similar to [Tayler 1973; Newcomb 1961]. A vertical component of the magnetic field could be considered as in [Chandrasekhar 1961], but is left out of the present study as it significantly complicates the linear stability analysis by adding imaginary roots to the system's determinant. Moreover, a purely horizontal magnetic field suffices to exhibit the new double diabatic instability. Considering the y component of the magnetic field is unnecessary since it can be projected along the x axis with a simple domain rotation. However, considering the case where the initial shear profile is not aligned with the magnetic field would be useful. This will be addressed in future works. We mention that in the special case where $\boldsymbol{Q}$ models magnetic resistivity, the background potential vector must satisfy $\boldsymbol{Q}(\boldsymbol{A}_0) = \nu \nabla^2 \boldsymbol{A}_0$, implying $\boldsymbol{Q}(q_0) = \frac{\partial^2 A_0(z)}{\partial z^2}\boldsymbol{e_y} = -\frac{\partial B_0(z)}{\partial z}\boldsymbol{e_y} = 0$, i.e., the background magnetic field must be constant along z to be a compatible background quantity. The linearization of (3.2) around (3.3) :

$$
\begin{aligned}
\nabla \cdot (\delta \boldsymbol{u}) = 0, \\
\rho_0 \frac{\partial \delta \boldsymbol{u}}{\partial t} + u_0(z)\rho_0 \partial_x \delta \boldsymbol{u} + \delta w \rho_0 \frac{\partial u_0(z)}{\partial z} \boldsymbol{e_x} - B_0(z)\partial_x \delta \boldsymbol{B} \\
-\delta B_z \frac{\partial B_0(z)}{\partial z} \boldsymbol{e_x} + \nabla \left(\delta P + B_0(z)\delta B_x\right) - \delta \rho \boldsymbol{g} = 0, \\
\frac{\partial \delta T}{\partial t} - \frac{\gamma - 1}{\gamma} \frac{\delta P}{P_0} + u_0(z)\partial_x \delta T \\
+\delta w \cdot \left(\frac{\partial T_0}{\partial z} - \frac{\gamma - 1}{\gamma} \frac{T_0}{P_0} \frac{\partial P_0(z)}{\partial z} \right) = \Delta H, \\
\frac{\partial \delta \boldsymbol{A}}{\partial t} + u_0(z)\partial_x \delta \boldsymbol{A} + \delta w \frac{\partial A_0(z)}{\partial z} \boldsymbol{e_y} = \Delta \boldsymbol{Q}, \\
\frac{\partial \delta X}{\partial t} + u_0(z)\partial_x \delta X + \delta w \frac{\partial X_0}{\partial z} = \Delta R.
\end{aligned}
\tag{3.4}
$$

The terms $\Delta S = \sum_{K=T,X,P,A_x,A_y,A_z} S_K \delta K$, where $S_K = \frac{\partial S}{\partial K}$ for $S \in \{H, R, \boldsymbol{Q}\}$, are the partial derivatives of the source terms with respect to the temperature, mass mixing ratio, pressure, and magnetic potential vector. The system closes with the linearized ideal gas EOS :

$$
0 = \frac{\delta P}{P_0} = \frac{\delta \rho}{\rho_0} + \frac{\delta T}{T_0} - \frac{\partial \log \mu_0}{\partial X} \delta X. \tag{3.5}
$$

In the Boussinesq regime, the pressure perturbations are retained only in the momentum equation to balance the gravitational force. Following [Kato 1966], the pressure perturbations are therefore eliminated from the EOS and the potential temperature perturbation equation :

$$
\frac{\delta \theta}{\theta_0} = \frac{\delta T}{T_0} - \frac{\gamma - 1}{\gamma} \frac{\delta P}{P_0} \sim \frac{\delta T}{T} \tag{3.6}
$$

We also assume simpler dependencies for the source terms :

$$
\begin{aligned}
H(T, X, P, \boldsymbol{A}) = H(T, X), \\
R(T, X, P, \boldsymbol{A}) = R(T, X), \\
\boldsymbol{Q}_i(T, X, P, \boldsymbol{A}) = \boldsymbol{Q}_i(A_i),
\end{aligned}
$$

chemical production and heating depend only on the temperature and mass mixing ratio. The dependencies on P are neglected as we study the Boussinesq regime. Lastly, it is assumed that the source term for each magnetic potential vector field component depends only on that component. More general inter-dependencies between the source terms will be considered in future works. In particular, exploring a thermally-dependent magnetic source term could be useful to model temperature-dependent resistivity.

3.2.2 . The instability criteria

All additional details of the derivation of the instability criteria are included in Appendix 3.A. This includes transitioning to Fourier space, deriving dispersion relations, and identifying the criteria for instability by computing the determinant of the resulting linear system. These steps need $\frac{\partial B_0(z)}{\partial z}$ to be neglected to yield interpretable results (avoiding imaginary terms in the determinant of the system). Two arguments can support this approximation : -The vertical variations of the background magnetic field occur on much larger scales than those of the

other background quantities in this local analysis -The source term for the magnetic field modeling resistivity. The Ohmic heating source term $H_{\boldsymbol{A}}$ is also omitted as it is negligible in this linear stability analysis; it is associated with perturbed quantities raised at second order. Lastly, we rewrite the shear profile $\frac{\partial u_0}{\partial z} = \left| \frac{\partial u_0}{\partial z} \right|$ as its effect on convection can only be stabilizing, as justified in section 3.4. Several notations are introduced : $1/h_p = -\frac{\partial \log P_0}{\partial z}$, $\nabla_T = -h_p \frac{\partial \log T_0}{\partial z}, \nabla_{\text{ad}} = \frac{\gamma-1}{\gamma}, \omega'_X = R_X + T_0 R_T \frac{\partial \log \mu_0}{\partial X}, \omega'_T = H_T + \frac{1}{T_0} H_X \left(\frac{\partial \log \mu_0}{\partial X} \right)^{-1}, \nabla_\mu = -h_p \frac{\partial \log \mu_0}{\partial z}$, $\nabla_u = -h_p \frac{1}{u_0} \left| \frac{\partial u_0}{\partial z} \right| < 0$. The analysis of the dispersion relation shows that the flow becomes unstable if any one of three inequalities is met. The first one is :

$$\begin{aligned} \nabla_T - \nabla_{\text{ad}} - \nabla_\mu - \frac{k^2 k_x^2}{k_x^2 + k_y^2} \frac{h_p}{\rho_0 g} B_0^2 \\ - \frac{k_x k_z}{k_x^2 + k_y^2} \frac{u_0}{g} \nabla_u (H_T + Q_A + R_X) \\ - \frac{k^2}{k_x^2 + k_y^2} \frac{h_p}{g} \left(H_T R_X - H_X R_T + Q_A (H_T + R_X) \right) > 0. \end{aligned} \tag{3.7}$$

that is, the adiabatic thermo-magneto-compositional sheared criterion. It manifests as soon as one initial gradient is non-zero. It generalizes the Ledoux criterion $\nabla_T - \nabla_{\text{ad}} - \nabla_\mu > 0$ to magnetized flows and source terms. The magnetic field's intensity stabilizes and can neutralize a mode's growth if it is strong enough, similar to the background shear profile coupled to the centered source terms H_T, R_X, Q_A. Finally, the source terms can stabilize the configuration. The next criterion is given by :

$$\begin{aligned} (\nabla_T - \nabla_{\text{ad}})(\omega'_X + Q_A) - \nabla_\mu (\omega'_T + Q_A) - \frac{k^2 k_x^2}{k_x^2 + k_y^2} \frac{h_p}{\rho_0 g} B_0^2 (H_T + R_X) \\ - \frac{k_x k_z}{k_x^2 + k_y^2} \frac{u_0}{g} \nabla_u \left(-H_X R_T + Q_A R_X + H_T (Q_A + R_X) \right) \\ - \frac{k^2}{k_x^2 + k_y^2} \frac{h_p}{g} Q_A (H_T R_X - H_X R_T) < 0. \end{aligned} \tag{3.8}$$

That is the diabatic thermo-magneto-compositional sheared criterion. It is new and manifests only if two initial gradients and at least one corresponding source are non-zero. It generalizes the diabatic thermo-magneto criterion of [Yu et Cheng 1973] to compositional and sheared flows and all thermo-compositional double-diffusive processes. This instability only manifests in the presence of source terms. It can be unstable to the Schwarzschild criterion $\nabla_T - \nabla_{\text{ad}} > 0$, because of a favorable thermal gradient if source terms on either chemistry or magnetic field are present. It can be unstable because of the mean molecular weight gradient if source terms on either temperature or magnetic field are present. The magnetic field can stabilize the flow if source terms on temperature or chemistry are present. The background shear, coupled with the source terms stabilizes the configuration. The new and last criterion is given by :

$$\begin{aligned} \left((\nabla_T - \nabla_{\text{ad}}) \omega'_X - \nabla_\mu \omega'_T \right) Q_A - \frac{k^2 k_x^2}{k_x^2 + k_y^2} \frac{h_p}{\rho_0 g} B_0^2 \left(H_T R_X - H_X R_T \right) \\ - \frac{k_x k_z}{k_x^2 + k_y^2} Q_A \frac{u_0}{g} \nabla_u \left(H_T R_X - H_X R_T \right) > 0. \end{aligned}$$

(3.9)

that is, the double diabatic thermo-magneto-compositional sheared criterion. It involves pairs of source term's derivatives multiplied by the gradients. It can be unstable because of a favorable thermal gradient if source terms on chemistry and the magnetic field are present. It can be unstable because of the mean molecular weight gradient if source terms on temperature and magnetic field are present. The magnetic field can stabilize the growth of a mode if source terms on temperature and chemistry are present. Finally, the sheared velocity profile coupled to the magnetic source stabilizes the configuration. Two symmetries can be noted between the criteria : -The double diabatic criteria is essentially the diabatic criteria where sums of source terms are replaced by products -The source terms coupled with a given gradient (thermal, chemical, or magnetic) are the ones coupled to the other two gradients.

The present study includes the results from [Tremblin et al. 2019]. By removing the shear, the magnetic field and the magnetic source term, the double diabatic criterion does not manifest anymore. Moreover, the adiabatic and diabatic criteria reduce to the one found in [Tremblin et al. 2019], which include Schwarzschild convection, Ledoux convection, and double-diffusive convection.

3.2.3 . Double diabatic instability

To illustrate the new double diabatic instability branch, we perform several 2D fully compressible finite volume MHD instability simulations with initial thermal and chemical gradients, a horizontal magnetic field, their corresponding centered source terms $H_T,\ R_X,\ Q_A$ and no shear profile. We set $-H_T \to \infty$

In this limit, the criteria (3.8) and (3.9) reduce to :

$$-\nabla_\mu - k^2 \frac{h_p}{\rho_0 g} B_0^2 < 0, \tag{3.10}$$

$$-\nabla_\mu Q_A - k^2 \frac{h_p}{\rho_0 g} B_0^2 R_X > 0, \tag{3.11}$$

and the adiabatic criterion (3.7) remains unchanged. Introducing the notations $\phi = -\nabla_\mu / \left(k^2 \frac{h_p}{\rho_0 g} B_0^2\right)$ and $r = R_X/Q_A$, the criteria simplify to $\phi > 1$ and $\phi > r$ respectively. Figure 3.1 displays the different stability zones in the parameter space (ϕ, r). We can see an area in the parameter space (r, ϕ) that is exclusively unstable to the double diabatic criterion. To corroborate our findings with numerical evidence, we examine the (in)stability behavior of this parameter space with 20^2 fluid simulations spanning the range $[\phi, r] \in [0, 1]^2$. We illustrate their stability behavior with colored dots superimposed on the diagram. We notice a precise correlation between the theoretical instability zones and the simulation outcomes. It is important to note that points on the axis $r = \phi$ do not all exhibit the same behavior. These points lie precisely on the boundary between stable and unstable parameters. This occurs because our analysis is local, while the simulations occur in a finite space where the criteria may or may not be satisfied depending on the altitude. In the next sections sections, we will concentrate our experiments on the non-linear regime.

3.2.4 . Thermo-sheared instability

We now study the influence of an initial shear profile on the growth rate of the adiabatic instability. We consider a Schwarzschild unstable atmosphere $\nabla_T - \nabla_{ad} > 0$ with no magnetic field $B_0 = 0$, no chemistry along with a thermal source term $H_T < 0$ and an initial shear profile $\mid \frac{\partial u_0}{\partial z} \mid > 0$. Under these conditions, only the

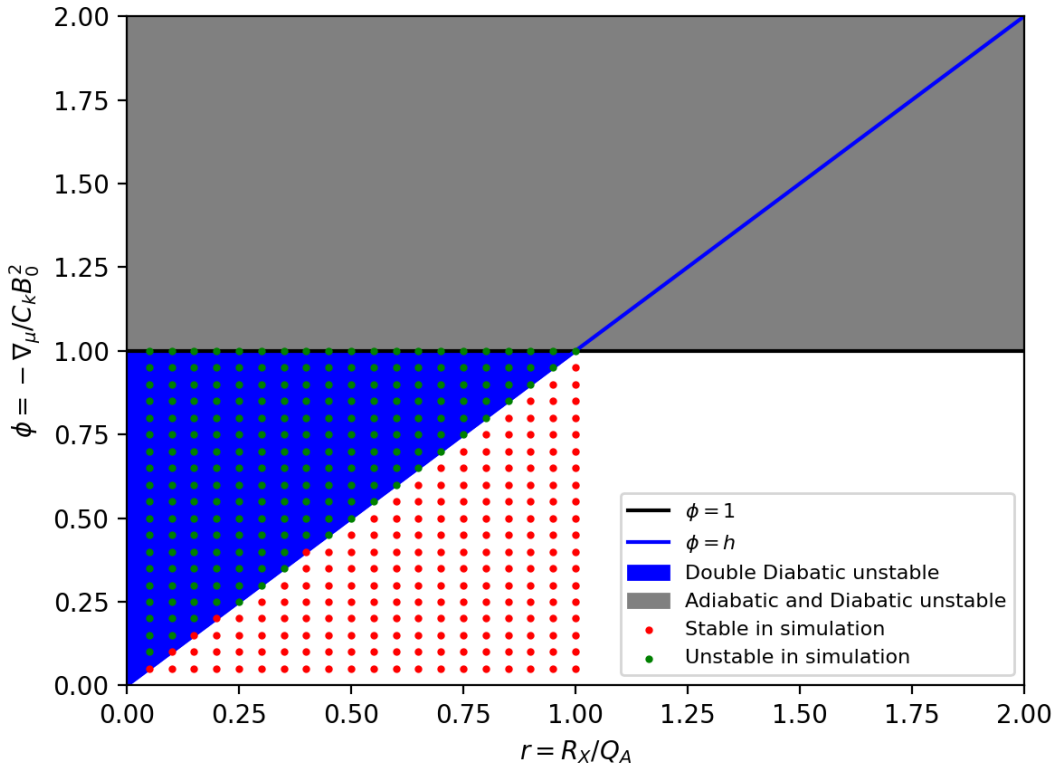


Figure 3.1 – Stability diagram in $\phi(r)$. The grey region denotes parameters that are adiabatic and diabatic unstable. The blue region denotes parameters that are double diabatic unstable. Points color corresponds to the simulation outcome : green for instability, red for stable behavior.

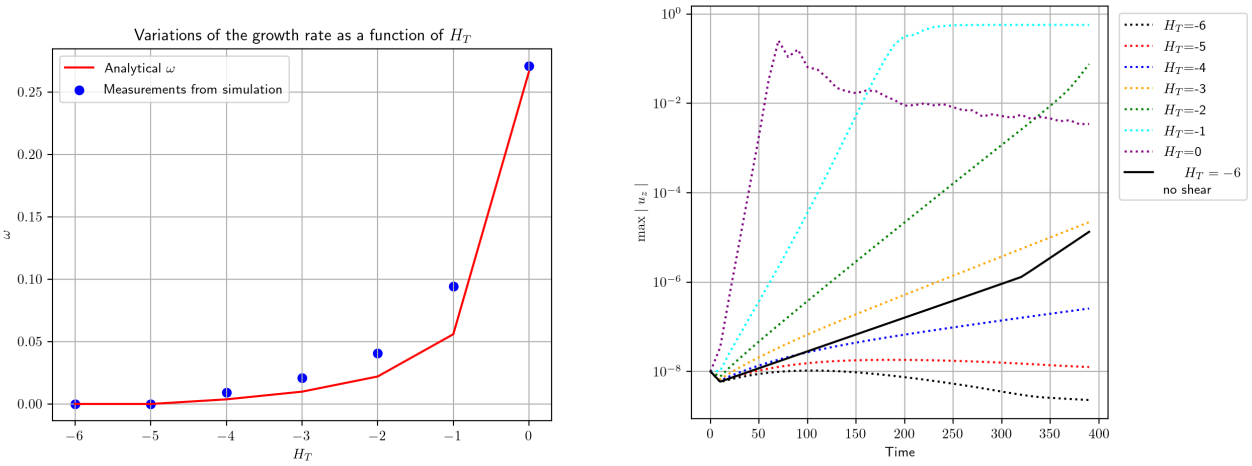


Figure 3.2 – Left : Evolution of the analytical growth rate as a function of H_T for the thermo-sheared adiabatic instability, theory, and simulation. Right : Time series of the vertical velocities in several adiabatic thermo-sheared instabilities with varying H_T.

adiabatic criterion (3.7) manifests and rewrites :

$$\nabla_T - \nabla_{ad} + \frac{k_z}{k_x}\frac{h_p}{g}\frac{\partial u_0}{\partial z}H_T > 0. \tag{3.12}$$

The coupled term involving shear and the source term is strictly positive and stabilizing. We will check several properties through simulation : By fixing $\mid \frac{\partial u_0}{\partial z} \mid > 0$, increasing $-H_T$ should allow us to progressively dampen the instability. Starting from a set of parameters that are stable according to the criterion due to the coupled shear and source term, we should retrieve an unstable behavior by removing the initial shear. Figure 3.2 shows the evolution of the growth rate as a function of H_T. The plot of the theoretical growth rate (red line) predicts that for $-H_T \in [0, 4]$, the stabilizing term is not strong enough to stabilize convection. For $-H_T \geq 5$, it stabilizes convection. We performed 7 instability simulations with varying H_T. The growth of the vertical velocities is shown in figure 3.2, and the associated growth rates are measured from our simulations and compared against the theory in figure 3.2, showing good correspondence. To illustrate the coupled nature of the shear stabilization, we perform the $H_T = -6$ simulation with no initial shear, and observed in figure 3.2 that the flow is indeed unstable.

3.3 . Non-linear regime

3.3.1 . Assumptions and resulting equation system

In this section, we aim to study the non-linear regime of the convective instability. In particular, we aim to provide simple estimations for the structure of active convective zones. We start with the observation that convection simulations go through three phases :

1. the linear phase where the perturbations grow exponentially with growth rate ω,
2. the transient phase where the perturbations are not negligible anymore and the vertical gradients of temperature/composition/shear/magnetic field are modified,

3. a statistically stationary phase where convection rolls are present, but the vertical gradients are not changing on a long time scale.

The goal of this section is to study the latter phase. To achieve this, we make the observation that the saturated regime is a small perturbation around a new hydrostatic equilibrium (determined by the mixing) so that studying the stationary linear system (i.e., with the constraint $\tilde{\omega} = \omega + ik_x u_0 = 0$) will give estimations that are representative of this regime. Let us derive our system, we start from (3.4), in which we inject the perturbation ansatz $\delta q = |\delta q| \exp\big(\tilde{\omega t} + i(k_x x + k_y y + k_z z)\big)$ (as in appendix 3.A) and assume $\tilde{\omega} = 0$. We look at the transport equations for the perturbations.

$$\begin{aligned}
\not{\omega}\delta \log\theta + \delta w \frac{\partial \log\theta_0}{\partial z} &= \frac{H_T}{T_0}\delta T + \frac{H_X}{T_0}\delta X, \\
\not{\omega}\delta X + \delta w \frac{\partial X_0}{\partial z} &= R_X \delta X + R_T \delta T, \\
\not{\omega}\delta u + \delta w \frac{\partial u_0}{\partial z} &= -i\frac{k_x \delta P}{\rho_0}, \\
\not{\omega}\delta A_y + \delta w \frac{\partial A_0}{\partial z} &= Q_A \delta A_y.
\end{aligned} \tag{3.13}$$

For the z and y velocity equations, we get :

$$\begin{aligned}
\not{\omega}\delta w + \delta A_y (k_x^2 + k_z^2) B_0(z) + ik_z \delta P \ + \delta\rho g = 0, \\
\not{\omega}\delta v + ik_y \delta P - k_y k_z \delta A_y = 0.
\end{aligned} \tag{3.14}$$

Injecting the second equation into the first one,

$$k_x^2 B_0 \delta A_y + \delta\rho g = 0. \tag{3.15}$$

Therefore, we can rewrite the EOS (3.5) in terms of perturbation of the potential vector or density, and replacing k_x by a characteristic length $1/L_x$ of the non-linear flow :

$$\frac{\delta T}{T_0} = -\frac{\delta\rho}{\rho_0} + \frac{\partial \log\mu_0}{\partial X}\delta X = \frac{B_0}{L_x^2 \rho_0 g}\delta A_y + \frac{\partial \log\mu_0}{\partial X}\delta X. \tag{3.16}$$

Also, by injecting (3.15) into (3.14), we get $\delta A_y k_z^2 B_0 + ik_z \delta P = 0$ which gives $ik_x \delta P = \delta\rho g \frac{k_x}{k_z} = \delta\rho g \frac{L_z}{L_x}$. This gives us the closed system :

$$\begin{aligned}
\delta w \frac{\partial \log\theta_0}{\partial z} &= \frac{H_T}{T_0}\delta T + \frac{H_X}{T_0}\delta X, \\
\delta w \frac{\partial X_0}{\partial z} &= R_X \delta X + R_T \delta T, \\
\delta w \frac{\partial u_0}{\partial z} &= -\frac{\delta\rho g}{\rho_0}\frac{L_z}{L_x}, \\
\delta w \frac{\partial A_0}{\partial z} &= Q_A \delta A_y.
\end{aligned} \tag{3.17}$$

Along with the closure (3.16) and the stationary condition

$$\omega = 0. \tag{3.18}$$

In order to study the non-linear regime of an instability, we can use (3.17)-(3.16) as well as the saturation (3.18). This last equation does not tell us explicitly on which instability criterion the flow has saturated. We argue that in the case where we are initially unstable to several criteria, the one associated with the strongest growth rate will be the one we saturate on. Let us look into an example with thermo-compostional diabatic convection. We consider a Ledoux unstable atmosphere where $\nabla_T - \nabla_{ad} > 0$ and $-\nabla_\mu < 0$, e.g., convection is driven by temperature and slowed down by composition. Let us also consider source terms such as $-H_T > -R_X$ so that we are also unstable to the diabatic criterion $(\nabla_T - \nabla_{ad})R_X - \nabla_\mu H_T < 0$. As mixing happens and gradients re-adjust towards stability, by reducing $(\nabla_T - \nabla_{ad})$ and increasing $-\nabla_\mu$. We can see that the diabatic criterion will saturate before the adiabatic criterion. At this point, we still have $\omega > 0$. More mixing will happen until $\nabla_T - \nabla_{ad} - \nabla_\mu = 0$, at which point $(\nabla_T - \nabla_{ad})R_X - \nabla_\mu H_T > 0$. In the general case, we assume that the initially strongest instability will be the one we saturate on.

In order to limit the number of variables and conduct fine parametric experiments in the non-linear regime, we resort to the use of strong relaxation towards equilibrium for the source terms, as per (3.44), to limit the number of degrees of freedom of the flow. For instance, if we wish to freeze the temperature gradient, we pick $-H_T \gg \omega$ with ω the initial linear growth rate of the instability, meaning that the relaxation time scale is much shorter than the mixing time scale.

This simplistic approach to the saturated regime allows us to study the non-linear coupling of the four physics involved, namely temperature, chemistry, magnetic field and shear. The correspondence of numerical simulations with our non-linear theory is less precise than in the linear regime, where the assumptions are more robust. The discrepancies between theoretical predictions and observed values in the non-linear regime can be attributed to several factors : the theory assumes background velocity and magnetic field profiles with no vertical components, an assumption that can be questioned by the presence of convection cells. This raises the difficult question of the convection cells being a perturbation versus a part of the saturated background. Additionally, we mention the impact of boundary conditions and the fully compressible nature of the simulation versus the Boussinesq approximation used in the theory.

We now focus on several subcases for which we conduct numerical experiments, including the self-generation of shear and magnetic fields in various geometries, and thermo-magneto convection. For each case, we link the behavior we observe in our simulations with our approach to the non-linear theory we just developed.

3.3.2 . 2D self-generation of shear

In this section, we start by making some observations about the generation of shear in 2D simulation and link these observations to our theory.

Observations

An intriguing phenomenon, often dismissed as a numerical artifact, occurs in 2D convective simulations where shear modes spontaneously emerge over time (see [Daley-Yates et al. 2021; Garaud et Brummell 2015]). We consider a 2D purely thermal flow that is Schwarzschild unstable $\nabla_T - \nabla_{ad} > 0$ along with a strong thermal source term H_T, no magnetic field, and a single-mode perturbation. In order to illustrate this phenomenon, we present three variations of this setup that we will refer to as scenarios A, B, and C and that corresponds to three different shear-related behaviors.

- **case A :** Our simulations begins in a square square domain. We apply a thermal source term $H_T = -1.0$,

and a perturbation of frequencies $(k_x, k_z) = (2\pi, \pi)$. Figure 3.3 illustrates that stationary convective rolls initially form in the simulation, only to be quickly destroyed, yielding a completely stable sheared flow. This transition from standard convection to stable shear is also very clear when looking at the time series for kinetic energies in the x and z directions. All perturbed variables, including these energies, exhibit exponential growth during the linear phase. In the phase of standard convection, the energies are roughly equal. The shift to stable shear is characterized by an exponential decrease in the z kinetic energy and an increase in x kinetic energy, indicating the destruction of the rolls and a transition to purely horizontal movement. This final state appears to be extremely stable.

- **Case B :** Expanding the box in the x-direction while keeping the perturbation unchanged along with periodic boundary conditions results in identical outcomes. This is because this modification alone effectively duplicates case A in a larger box. However, the longer box gives us the room to impose a longer perturbation $k_x = \pi$. This change, along with all other parameters ($\nabla_T - \nabla_{ad}$, H_T, g, h_p, k_z) kept unchanged leads to different results, as shown in figure 3.4. The convective rolls are generated and stably preserved and do not present any sheared behavior. In the non-linear regime, a small-scale oscillation of the x and z kinetic energies is observed, with no notable increasing/decreasing tendency.
- **Case C :** Finally, if we choose to increase the adiabatic instability strength by decreasing H_T to -0.25 we obtain yet another scenario depicted in Figure 3.5. Initially similar to case B, the rolls eventually exhibit apparent movement within the box (note how the ascending and descending fluid columns are translated horizontally). Moreover, we observe that the left convective cell is getting bigger while the other cell is getting smaller. Analyzing the kinetic energy time series reveals that this is linked to shear modes. During the transient standard convection phase, the x kinetic energy continues to rise, after the end of the linear phase, plateauing only later. It appears that the shear is sustained by feeding of z kinetic energy that is decreasing before the state reaches a steady state : stable sheared convection.

We emphasize that these phenomena manifest despite the simulation's complete conservation of transverse momentum up to machine precision. This property of the numerical scheme allows us to be certain that the modes are naturally generated by the flow rather than injected by the numerical method.

Proposed explanation

In order to study the non-linear regime of sheared flows, we look at our transport system (3.17), considering only the shear and temperature gradients, and the H_T source term.

$$\begin{aligned} \delta w \frac{\partial \log \theta_0}{\partial z} &= \frac{H_T}{T_0} \delta T, \\ \delta w \frac{\partial u_0}{\partial z} &= -\frac{\delta \rho g}{\rho_0} \frac{L_z}{L_x}. \end{aligned} \tag{3.19}$$

Since the flow is initially unstable to the Schwarzschild criterion, the $\omega = 0$ condition corresponds to the flow being marginally stable to the sheared adiabatic criterion,

$$\nabla_T - \nabla_{ad} - H_T \frac{L_x}{L_z} \frac{h_p}{g} \mid \frac{\partial u_0}{\partial z} \mid = 0. \tag{3.20}$$

As we picked a strong value of the thermal source term H_T, we can assume that the temperature profile is mostly driven by that source term. Formally, we introduce a small parameter ϵ such that $\delta w \frac{\partial \log \theta_0}{\partial z} = \frac{1}{\epsilon} \frac{H_T}{T_0} \delta T$. Taking

the limit $\epsilon \to 0$, we see that $T = T_0$, i.e., the temperature gradient in the saturated regime is the initial temperature gradient. For all three scenarios under consideration, the initial shear is zero, and the flow is Schwarzschild unstable. We observe a standard linear phase until saturation is reached, when (3.20) is satisfied, which rewrites

$$\left|\frac{\partial u_0}{\partial z}\right| = \frac{L_z}{L_x}(\nabla_T - \nabla_{ad})\frac{g}{H_T h_p}, \tag{3.21}$$

that is an estimation of the shear intensity in the non-linear regime. We emphasize that the temperature gradient, g, and h_p are unchanged across three cases. Moreover, they all exhibit convection rolls of vertical size $L_z = 1$ during their respective standard convection phases. The distinction between configurations A and B is the horizontal dimension of the convection cells, influenced by the box size and initial perturbation. Specifically, $L_x = 0.5$ for configuration A, whereas $L_x = 1$ for configurations B and C. This difference suggests that the shear required to saturate configuration A is approximately twice that of B, aligning with our observations where A transitions to stabilized shear and B to nearly shear-less convection. By the same reasoning, we anticipate that the shear generated by configuration C is approximately four times that of B, a prediction confirmed by the more pronounced shear manifestations in C compared to B. Moreover, the stability of the final flow state of Case A can be understood through the adiabatic criterion (3.7), where $k_x \approx 0$ (no horizontal variations) predicts an extremely stable behavior regardless of the strength of the thermal gradient. We believe stable shear is an extreme case of sheared convection where one convective cell grows enough to completely destroy the other. A difficult question remains : the prediction of the emergence of stable shear versus stabilized sheared convection. Interestingly, despite the expectation of higher shear in scenario C than A (by a factor of two), the reverse is observed where A tends to stable shear while C tends to stable sheared convection. We believe that this apparent paradox is explainable by the box size. First, convection cells are created by the instability. Then the shear increases until saturation; however, convection cells are destroyed if the saturation value for the shear exceeds the maximum value that the convective rolls can withstand. We think that the convective rolls in case C can stably admit more shear than the ones in case A because the left convective cell has more space to grow without spanning the whole domain. Lastly, it is easy to see that the geometry of the convective rolls during the standard convection phase is significantly influenced by the box's shape. For example, initiating a simulation in a square box with perturbation frequencies $(k_x, k_z) = (\pi, \pi)$ results in a final state identical to scenario A, where two convective cells of horizontal size $L_x = 0.5$ emerge and are then destroyed by shear. Indeed, as the initial total x momentum is 0, only an even amount of convective cells can emerge. Moreover, in all our experiments, the vertical size of the cells is $L_z = 1$. This could differ in the presence of a convective staircase, but that is beyond the scope of the present chapter.

3.3.3 . 3D self-generation of shear

In the adiabatic regime

In this section, we present our findings on the 3D self-generation of shear, particularly emphasizing the role of box geometry in the development of shear modes. We adopt physical parameters identical to those in Case C from the previous section but modify the domain's geometry and perturbation. Specifically, we examine a cubic domain with $(l_x, l_y, l_z) = (1, 1, 1)$ and an initial perturbation of $(k_x, k_y, k_z) = \pi(2, 2, 1)$, alongside a elongated box domain where $(l_x, l_y, l_z) = (2, 1, 1)$ and the perturbation is $(k_x, k_y, k_z) = \pi(1, 2, 1)$. Figure 3.6 reveals that convection quickly transitions to a turbulent state. At various time steps, no stable structures are observed; instead, convection cells constantly move and deform. This behavior suggests that shear in the x and y directions

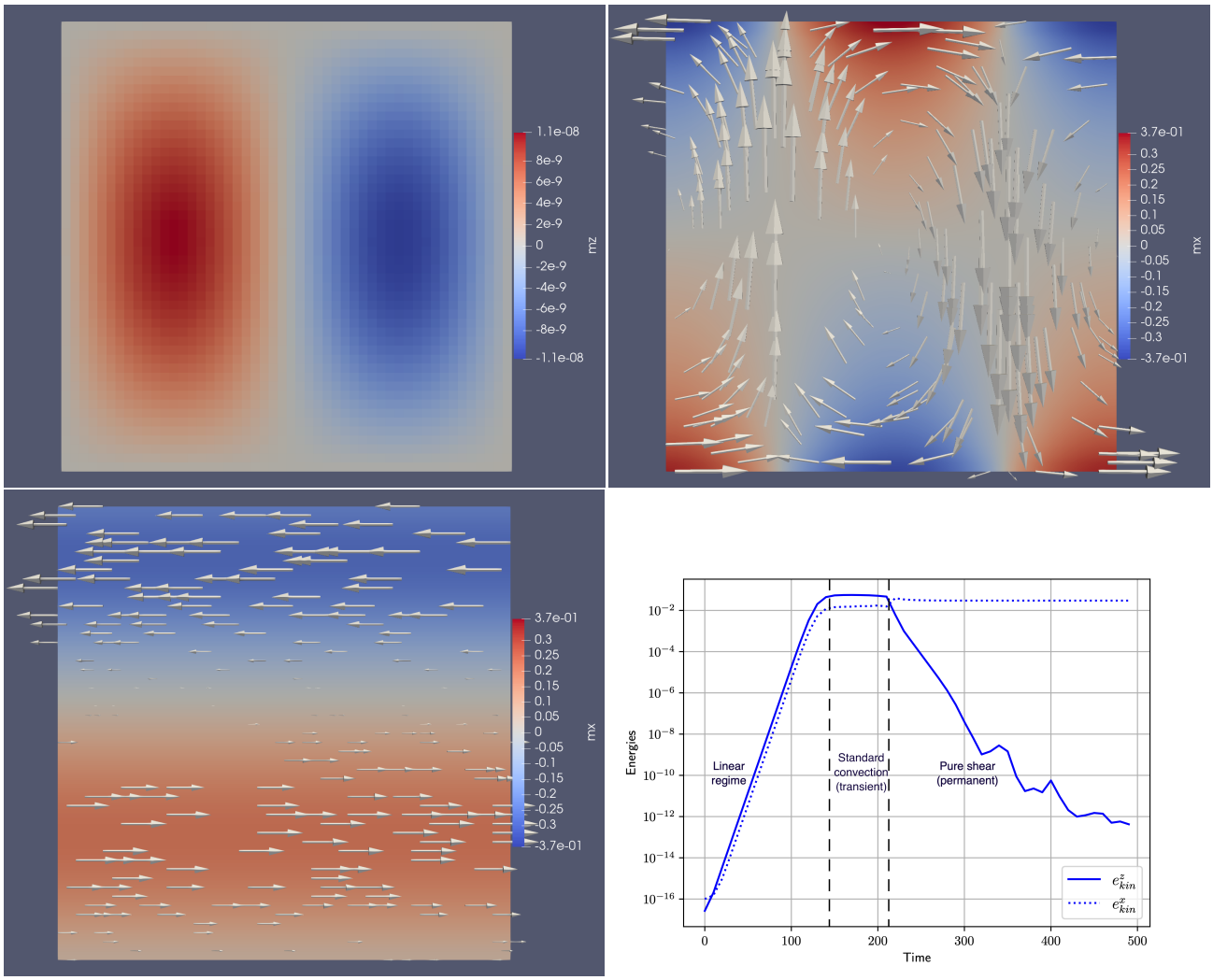


Figure 3.3 – Case A. Top left : initial vertical velocity perturbation. Top right : early stage with standard convection. Bottom left : late stage with stable shear. Bottom right : time series of the x and z kinetic energies.

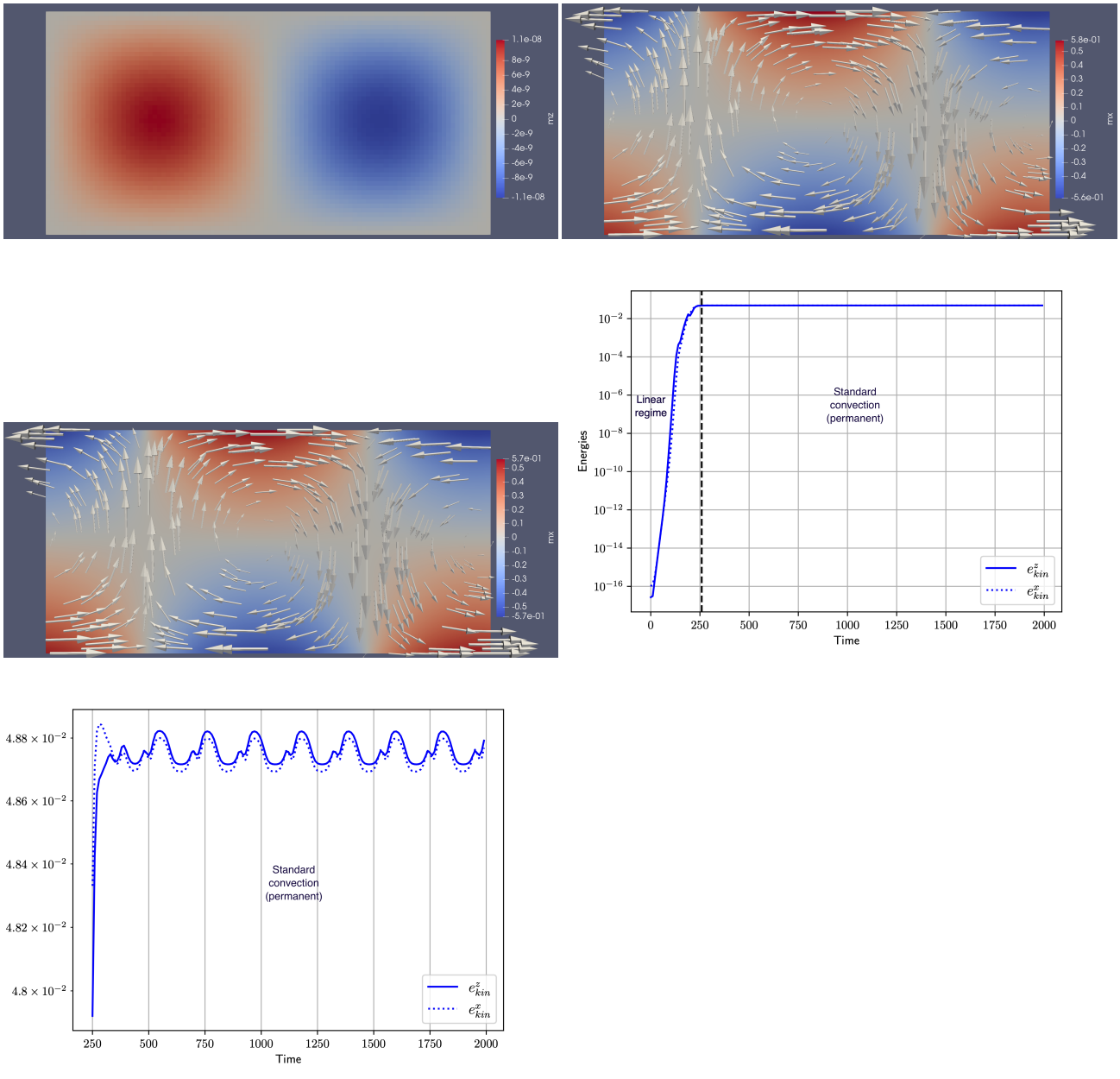


Figure 3.4 – Case B. Top left : initial vertical velocity perturbation. Top right : early stage with standard convection. Middle left : late stage with stable standard convection. Middle right : time series of the x and z kinetic energies. Bottom left : time series of the x and z kinetic energies from $t = 250$

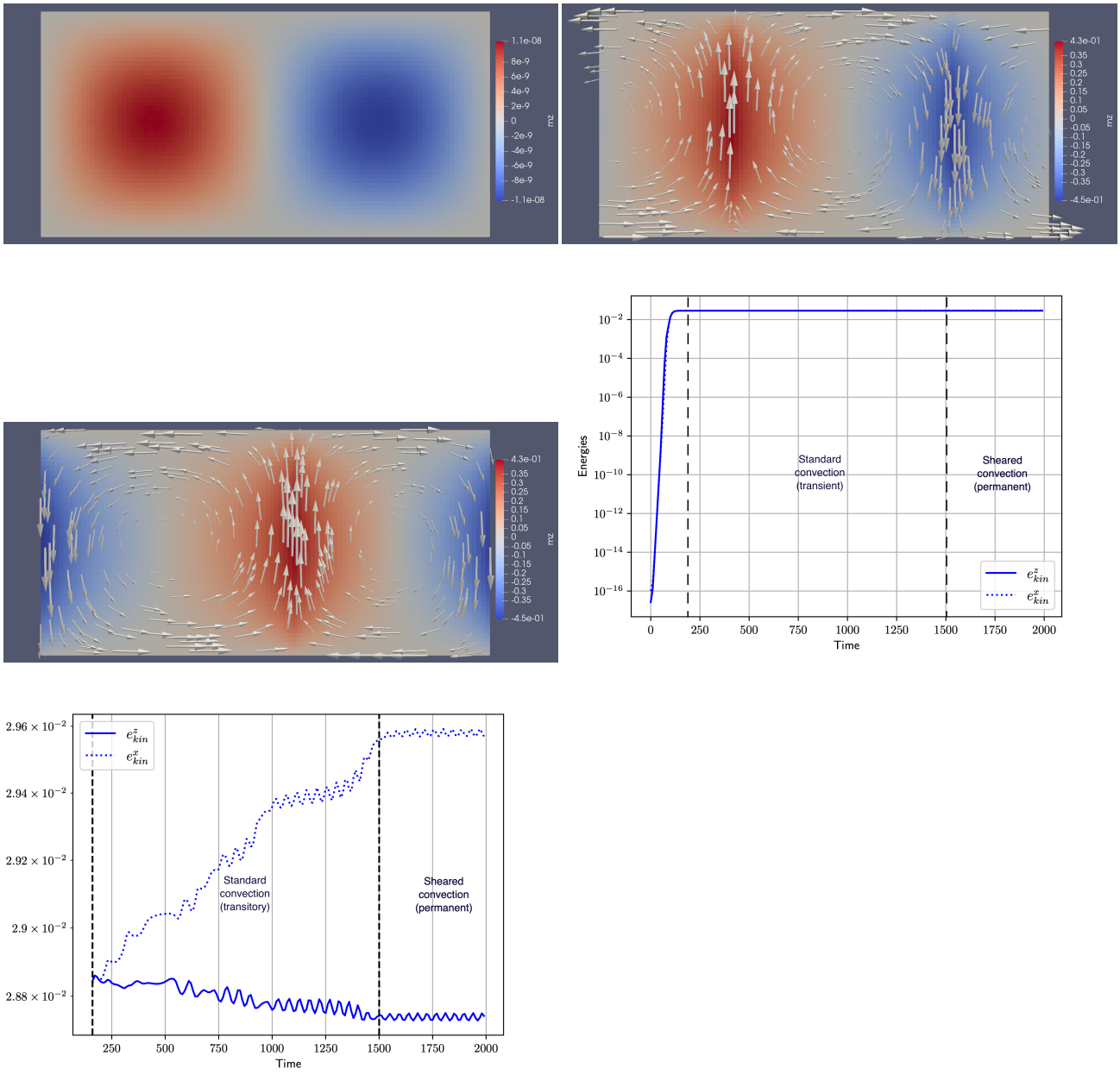


Figure 3.5 – Case C. Top left : initial vertical velocity perturbation. Top right : early stage with standard convection. Middle left : late stage with stable sheared convection. Middle right : time series of the x and z kinetic energies. Bottom left : time series of the x and z kinetic energies from $t = 200s$.

are competing, thus generating turbulence without stabilizing the convection. Notably, the kinetic energies in the x and y directions have similar orders of magnitude. The vertical kinetic energy remains unaffected by the growth of shear modes. Furthermore, the horizontal kinetic energies exhibit a phase opposition, where the peak of one corresponds to the lows of the other, highlighting the competition between them. In this configuration, it appears that shear does not significantly influence the mixing or marginal stability of the flow due to this competition. In contrast, Figure 3.7 shows the effect of elongating the box in the x direction. In the non-linear regime, a 2D-like flow structure becomes apparent and stable, characterized by two convection rolls aligned in the y direction and sheared along the x direction, resembling case C from the previous section. This observation indicates that x shear predominantly influences the convective regime and the flow's geometry; the y kinetic energy constitutes only 6% of the total kinetic energy, while the remainder is approximately equally divided between the x and z kinetic energies. These latter two energies are strongly correlated, and their time series are nearly identical. Additionally, there is a phase opposition between the y kinetic energy and the other two. Figure 3.8 compares the horizontal velocity profiles in the cubic box vs. the elongated box. We can see that there is no strong horizontal velocity gradient in the cubic box. However, we observe that the shear in the x direction is predominant in the elongated box, which is consistent with our other conclusions. We can link this change of behavior between the two configurations by looking at the geometrical prefactor in front of the shear term in the adiabatic criterion (3.7), which writes $\frac{k_x k_z}{k_x^2+k_y^2}$. The smaller the prefactor, the more shear we expect, as explored in the previous section. If we added shear in the y direction to the study, we would get another prefactor in front of the y shear, $\frac{k_y k_z}{k_x^2+k_y^2}$. The x shear and y shear prefactor ratio is $k_x/k_y = l_y/l_x$. As a result, for a cubic box, both shears should be of the same strength and competing. For an elongated box in the x direction, we expect the shear to be stronger in the x direction, which is consistent with our observations. The preferred direction in the elongated setup could be imposed in a realistic situation by, e.g., the Coriolis force in the case of stars and planets.

In the diabatic regime

[Garaud et Brummell 2015] investigated the emergence of shear modes in 2D convection simulations, which was then thought to be a spurious effect from the numerical method used. The study was done in the context of double-diffusive convection (thermohaline) at a low Prandtl number. They note that these shear modes appear predominantly in regimes of low Prandtl numbers and high values of the density stratification. Additionally, they recommend resorting to 3D simulations in boxes with reduced vertical height, L_z, to avoid the development of these modes. This insight aligns with our understanding of shear dynamics. The framework of [Garaud et Brummell 2015] can be compared to ours by defining a stabilizing temperature gradient, $\nabla_T - \nabla_{ad} < 0$, and a destabilizing mean molecular weight gradient, $-\nabla_\mu > 0$, along with the derivatives of the source terms H_T and R_X so that the diabatic instability condition $(\nabla_T - \nabla_{ad})R_X - \nabla_\mu H_T < 0$ is verified. As the flow is initially unstable to the diabatic instability, (3.18) implies that it shall saturate according to :

$$\left|\frac{\partial u_0}{\partial z}\right| = \frac{L_z}{L_x}\frac{g}{h_p}\left((\nabla_T - \nabla_{ad})/H_T - \nabla_\mu/R_X\right). \tag{3.22}$$

and provide us with a non-linear estimation of the self-generated shear for diabatic convection. The Prandtl number is defined as the ratio of kinematic viscosity to thermal diffusivity. In our model, the kinematic viscosity is determined by numerical diffusion, implying that a low Prandtl number corresponds to a high value of H_T. In our framework, such a high value of H_T does increase the shear intensity as it reduces the contribution of the stabili-

zing thermal gradient in (3.22). Additionally, a smaller L_z corresponds with a lower shear magnitude, reinforcing the consistency of [Garaud et Brummell 2015]'s conclusions with our analysis.

3.3.4 . 2D thermo-magneto convection

In this section, we present simulations of the non-linear regime of a 2D diabatic thermo-magneto instability. We disregard chemistry effects and consider a Schwarzschild unstable atmosphere $\nabla_T - \nabla_{ad} > 0$ stabilized by a background uniform magnetic field and perturbed with a single-mode perturbation (k_x, k_y) such that it is stable to the adiabatic criterion $\nabla_T - \nabla_{ad} - (k_x^2 + k_z^2)\frac{h_p}{\rho_0 g} B_0^2 < 0$. We consider source terms derivatives $-H_T < -Q_A$ such that the diabatic instability criterion $(\nabla_T - \nabla_{ad})Q_A - H_T(k_x^2 + k_z^2)\frac{h_p}{\rho_0 g} B_0^2 < 0$ is met, leading to an unstable growth of the perturbation until diabatic saturation. The system (3.17)-(3.16) rewrites as :

$$\begin{aligned} \delta w \frac{\partial \log \theta_0}{\partial z} &= \frac{H_T}{T_0}\delta T, \\ \delta w \frac{\partial A_0}{\partial z} &= Q_A \delta A_y, \end{aligned} \tag{3.23}$$

and the saturation condition(3.18) becomes

$$(\nabla_T - \nabla_{ad})Q_A - (k_x^2 + k_z^2)\frac{h_p}{\rho_0 g} B_0^2 H_T = 0 \tag{3.24}$$

Given the initial magnetic field is strong, and the magnetic relaxation term $-Q_A > -H_T$ maintains it close to its initial value, we anticipate the less relaxed thermal gradient adjusts according to $\nabla_T - \nabla_{ad} \propto \frac{H_T}{Q_A} B_0^2$. To illustrate this, we introduce a small paramter ϵ such that $\delta w \frac{\partial A_0}{\partial z} = \frac{1}{\epsilon} Q_A \delta A_y$. In the limit $\epsilon \to 0$, we get $\delta A_y = 0$. By comparing the potential temperature gradient in several saturated simulations with different diffusion coefficients H_T, Q_A, we can test the non-linear regime theory. To this end, we define the ratio $r = H_T/Q_A$ and perform six simulations corresponding to $(r, H_T) \in (\{1/2, 1/3, 1/4\}, \{0.02, 0.04\})$ and plot the potential temperature as a function of altitude in figure 3.9. We make two observations that are consistent with our predictions,

1. at constant r, varying H_T has little effect on the temperature gradient,
2. the potential temperature profiles scale linearly with r as predicted by the non-linear theory.

For instance, we can calculate the following ratios from our simulation outputs :

$$\begin{aligned} \frac{\partial \log T}{\partial z}\Big|_{r=1/2}^{H_T=0.02} \Big/ \frac{\partial \log T}{\partial z}\Big|_{r=1/4}^{H_T=0.02} &= 2.09 \approx 2, \\ \frac{\partial \log T}{\partial z}\Big|_{r=1/2}^{H_T=0.04} \Big/ \frac{\partial \log T}{\partial z}\Big|_{r=1/4}^{H_T=0.04} &= 2.15 \approx 2, \\ \frac{\partial \log T}{\partial z}\Big|_{r=1/2}^{H_T=0.02} \Big/ \frac{\partial \log T}{\partial z}\Big|_{r=1/3}^{H_T=0.02} &= 1.54 \approx 1.5, \\ \frac{\partial \log T}{\partial z}\Big|_{r=1/2}^{H_T=0.04} \Big/ \frac{\partial \log T}{\partial z}\Big|_{r=1/3}^{H_T=0.04} &= 1.56 \approx 1.5. \end{aligned}$$

which validates our approach in this context.

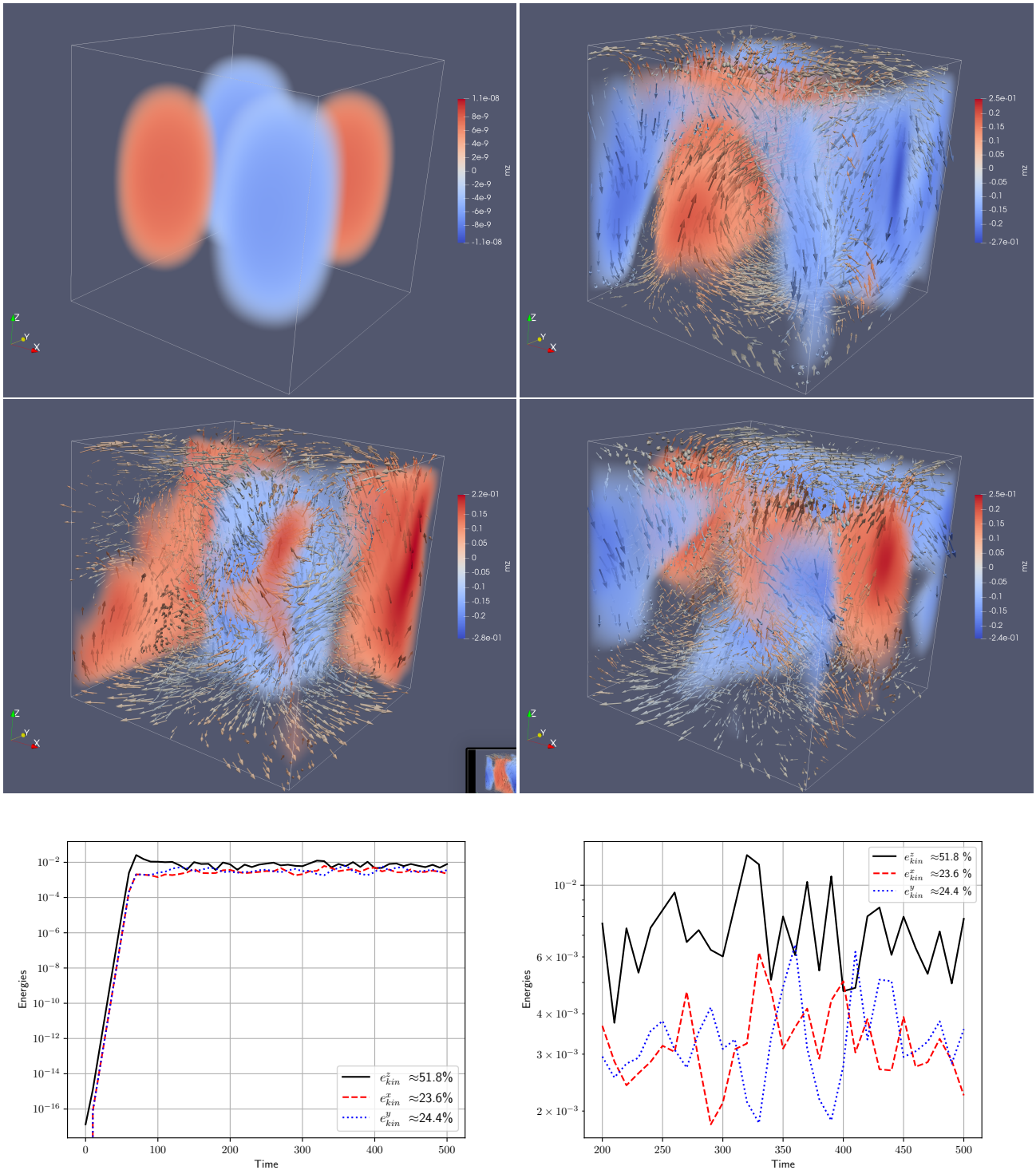


Figure 3.6 – Shear in a 3D cubic box. Top left : Initial perturbation. Top right : turbulent convection at $t = 100$. Middle left : turbulent convection at $t = 300$. Middle right : turbulent convection at $t = 500$. Bottom left : time series of the kinetic energies per direction. Bottom right : time series of the kinetic energies per direction between $t = 200$ and $t = 500$

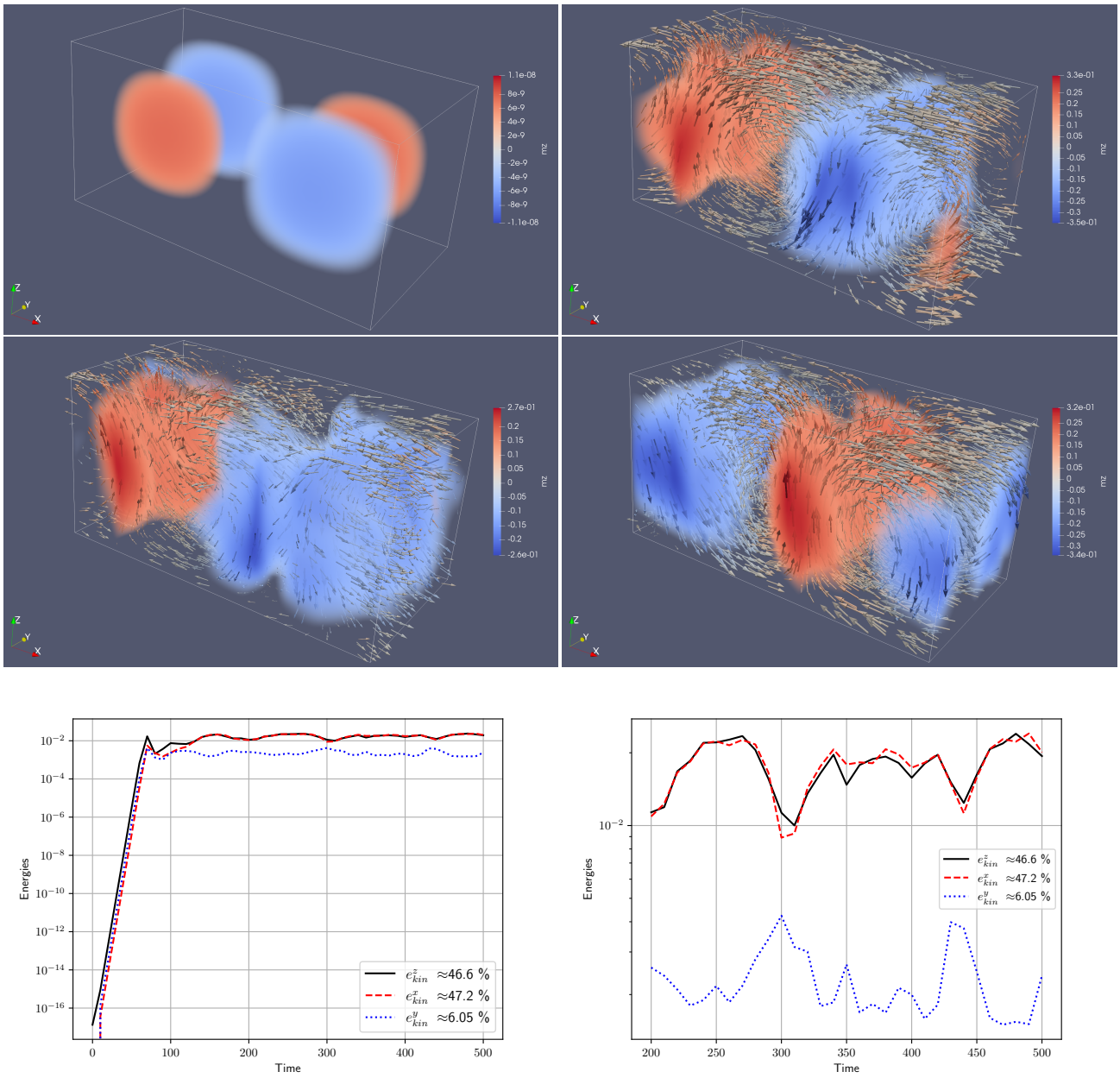


Figure 3.7 – Shear in a 3D elongated box. Top left : initial perturbation. Top right : turbulent convection at $t = 100$. Middle left : turbulent convection at $t = 300$. Middle right : turbulent convection at $t = 500$. Bottom left : time series of the kinetic energies per direction. Bottom right : time series of the kinetic energies per direction between $t = 200$ and $t = 500$

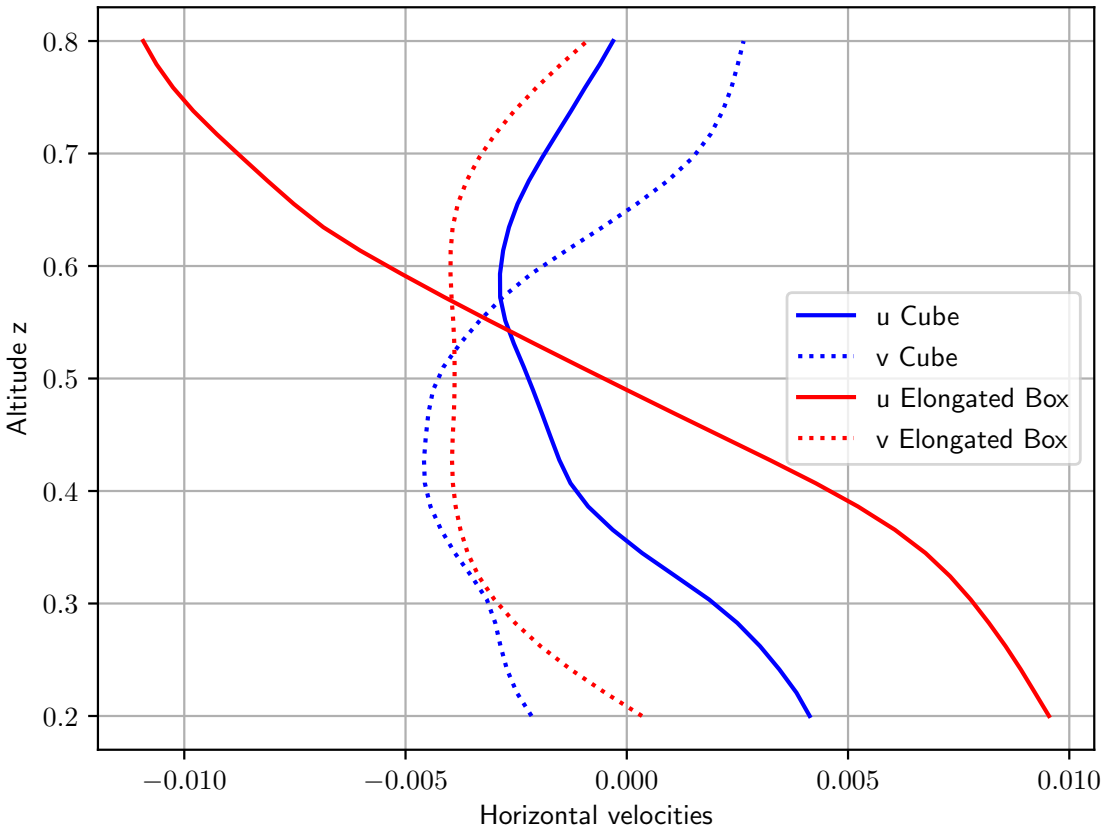


Figure 3.8 – Time average of the horizontal velocities in a cubic box vs. an elongated box. The time average is done from $t = 100$ until the end of the simulation.

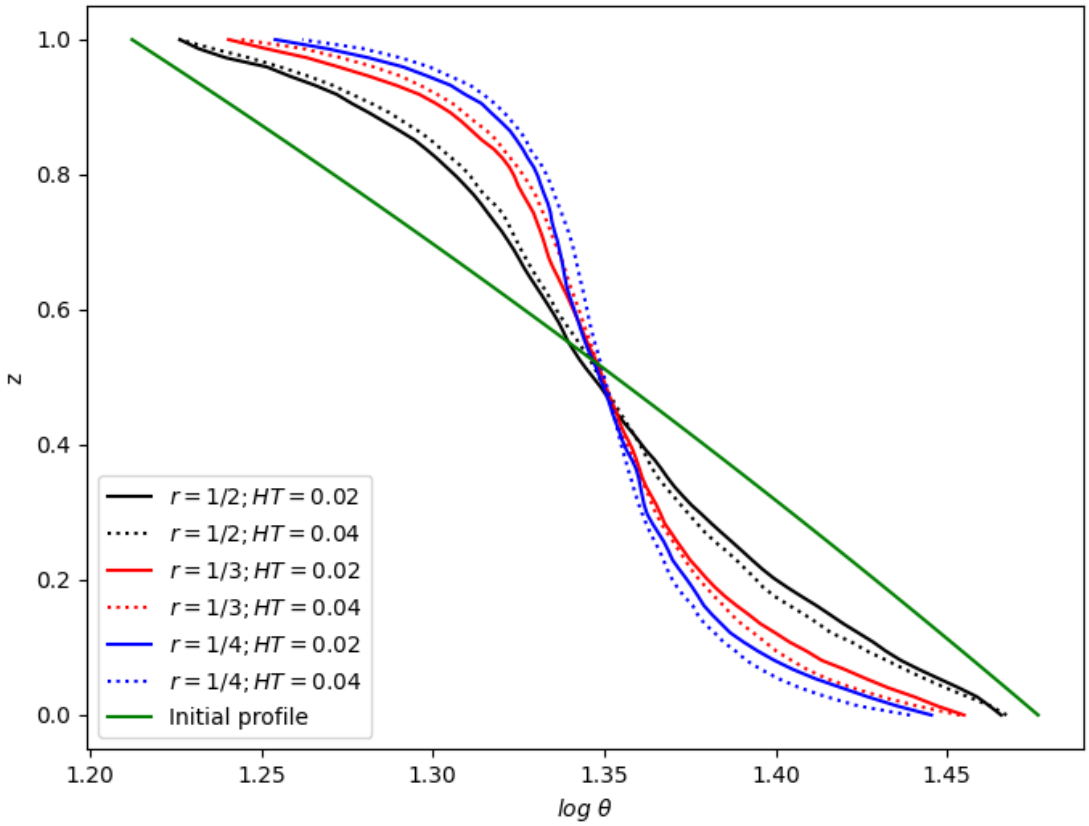


Figure 3.9 – Potential temperature profiles as a function of altitude in several non-linear 2D saturated simulations with varying $r = \frac{H_T}{Q_A}$ and H_T.

3.3.5 . 3D convective dynamo

In this section, we perform convective dynamo numerical experiments in the adiabatic and diabatic regimes. We aim to verify that the self-generated magnetic field observed in our simulation behaves according to the prediction of the non-linear theory. In all the following experiments, we consider a cubic geometry of the domain. This is because we want to minimize the generation of shear that could compete with the dynamo effect (see section 3.3.3). In order to map the non-linear theory to the simulation results, one needs to choose values for the geometrical prefactor $\frac{k^2 k_x^2}{k_x^2+k_y^2}$. This is because our estimations are coming from the condition (3.18), which includes this geometrical prefactor. A careful Fourier analysis of the flow would provide the primary modes, but we do not delve into such derivations. Instead, we select the values of the prefactor that provide the best match between theory and measurements from simulations. Then, we assume that $k_x = k_y = k_z = \frac{2\pi}{L}$ where L is a characteristic length of the simulation. This gives us $\frac{k^2 k_x^2}{k_x^2+k_y^2} = \frac{3}{2}\left(\frac{2\pi}{L}\right)^2$ and allows us to see if the value that we fit for the prefactor corresponds to a reasonable scale L. In the context of mean-field theories usually used in astrophysics (mixing length), this can be interpreted as a mixing length.

Adiabatic saturation via Dynamo Effect

Our initial setup is unstable to the Schwarzschild criterion due to a favorable initial temperature profile $\nabla_T - \nabla_{ad} > 0$. We consider a strong relaxation term for the temperature, $-H_T \gg \omega$. initially homogeneous, weak horizontal magnetic field, $B_0^2 \simeq 0$ that serves as a seed to the dynamo. We introduce a weak source term for the magnetic field, i.e., $Q_A \simeq 0$. From (3.7), and disregarding the shear's effect, our initial state is :

$$(\nabla_T - \nabla_{ad}) - \frac{k^2 k_x^2}{k_x^2 + k_y^2} \frac{h_p}{\rho_0 g} B_0^2 > 0 \tag{3.25}$$

For this setup, the system (3.17)-(3.16) rewrites as :

$$\begin{aligned} \delta w \frac{\partial \log \theta_0}{\partial z} &= \frac{H_T}{T_0} \delta T, \\ \delta w \frac{\partial A_0}{\partial z} &= Q_A \delta A_y \end{aligned} \tag{3.26}$$

And the saturation (3.18) will ensure

$$(\nabla_T - \nabla_{ad}) - \frac{k^2 k_x^2}{k_x^2 + k_y^2} \frac{h_p}{\rho_0 g} B_0^2 = 0. \tag{3.27}$$

As we picked a strong value of the source term H_T, we can assume that potential temperature is mainly driven by the source. We rewrite the potential temperature equation, introducing a small parameter ϵ such that $\delta w \frac{\partial \log \theta_0}{\partial z} = \frac{1}{\epsilon} \frac{H_T}{T_0} \delta T$. Taking the limit $\epsilon \to 0$, we see that $T = T_0$, i.e., the temperature gradient in the saturated regime is the initial temperature gradient. Hence, convection will increase the background magnetic field, B_0^2 until (3.27) is satisfied. We verify this estimation through numerical experiments by executing a series of convective dynamo simulations with varying initial temperature gradients. Figure 3.10 displays the temporal evolution of the mean kinetic (solid) and magnetic (dashed) energies. It is clear that the dynamo amplifies as the parameter $\nabla_T - \nabla_{ad}$ increases. In figure, 3.10, we represent the analytical estimation for the saturated magnetic energy along with the measurement from our simulations. We fit the parameter $\frac{k^2 k_x^2}{k_x^2 + k_y^2}$ that minimizes the difference between the estimation and the measures, giving us a value of $L = 0.34$, corresponding to a characteristic horizontal size of the convective cells the size of about a third of the domain's length.

Diabatic Saturation via Dynamo Effect

Our next experiment is very similar to the previous one, with the only difference being that we are now considering an initial diabatic instability. Our setup consists of an atmosphere stable according to the Ledoux criterion, with a stabilizing temperature profile, $\nabla_T - \nabla_{ad} < 0$, which is greater in magnitude than a destabilizing mean molecular gradient $\nabla_T - \nabla_{ad} - \nabla_\mu < 0$. We also consider an initially homogeneous, weak horizontal magnetic field, with $B_0^2 \simeq 0$. For source terms, we adopt a similar approach to previous experiments by selecting large values for the thermal and compositional sources $H_T = 2R_X$. We activate a very small magnetic source term $-Q_A \simeq 0$ as well. Disregarding the shear's effect, the diabatic criterion (3.8) simplifies to :

$$(\nabla_T - \nabla_{ad})(R_X + Q_A) - \nabla_\mu (H_T + Q_A) - \frac{k^2 k_x^2}{k_x^2 + k_y^2} \frac{h_p}{\rho_0 g} B_0^2 (H_T + R_X) < 0 \tag{3.28}$$

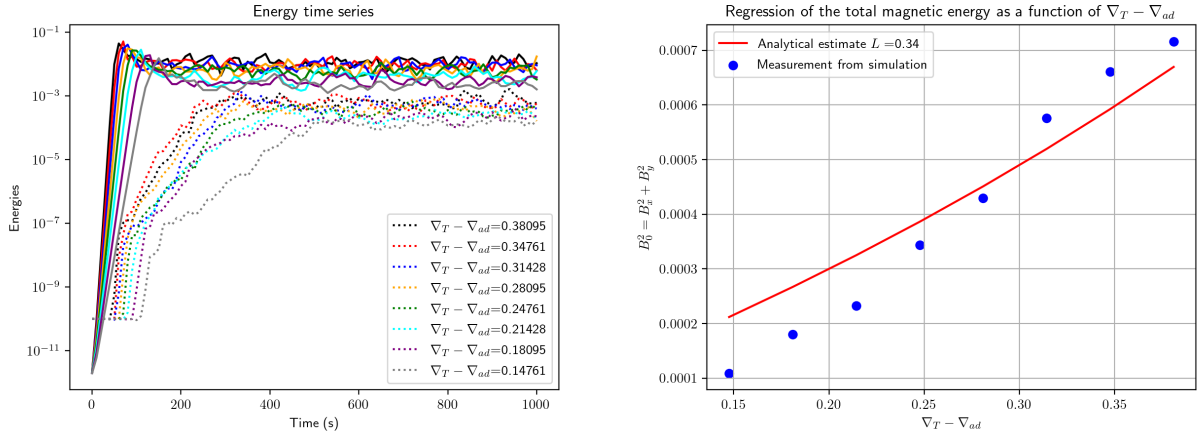


Figure 3.10 – Left : Time series of the mean kinetic (solid) and magnetic (dashed) energies in adiabatic convective dynamo simulations with varying $\nabla_T - \nabla_{ad}$. Right : Theoretical prediction and measurement of B_0^2 at saturation, as a function of $\nabla_T - \nabla_{ad}$. The represented value of $B_0^2 = B_x^2 + B_y^2$ corresponds to a time average starting from $t = 600s$.

With this configuration, the system (3.17) becomes :

$$\begin{aligned} \delta w \frac{\partial \log \theta_0}{\partial z} &= \frac{H_T}{T_0}\delta T, \\ \delta w \frac{\partial X_0}{\partial z} &= R_X \delta X, \\ \delta w \frac{\partial A_0}{\partial z} &= Q_A \delta A_y. \end{aligned} \tag{3.29}$$

The saturation writes

$$(\nabla_T - \nabla_{ad})(R_X + Q_A) - \nabla_\mu (H_T + Q_A) - \frac{k^2 k_x^2}{k_x^2 + k_y^2}\frac{h_p}{\rho_0 g} B_0^2 (H_T + R_X) = 0. \tag{3.30}$$

As we picked a strong value of the source terms R_X, H_T, we can assume that potential temperature and composition are mainly driven by the sources. We rewrite their equations, introducing a small parameter ϵ such that $\delta w \frac{\partial \log \theta_0}{\partial z} = \frac{1}{\epsilon}\frac{H_T}{T_0}\delta T$, $\delta w \frac{\partial X_0}{\partial z} = \frac{1}{\epsilon} R_X \delta X$. Taking the limit $\epsilon \to 0$, we see that $T = T_0$ and $X = X_0$, i.e., the temperature and composition gradients in the saturated regime are close to their initial values. Thus, the only way the flow can saturate is via the dynamo effect, increasing B_0^2 until it saturates (3.30) i.e

$$\frac{k^2 k_x^2}{k_x^2 + k_y^2}\frac{h_p}{\rho_0 g} B_0^2 = \frac{1}{H_T + R_X}\left(\overbrace{(\nabla_T - \nabla_{ad}) R_X - \nabla_\mu H_T}^{\text{Double-diffusive unstable}} + Q_A \underbrace{(\nabla_T - \nabla_{ad} - \nabla_\mu)}_{\text{Ledoux stable}} \right) \tag{3.31}$$

This estimation shows that the dynamo effect should be amplified with the unstable diabatic hydrodynamic gradient $(\nabla_T - \nabla_{ad}) R_X - \nabla_\mu H_T$ and dampened by the hydrodynamic adiabatic gradient, multiplied by Q_A : $Q_A(\nabla_T - \nabla_{ad} - \nabla_\mu)$. To test this hypothesis, we conduct a parametric study where we vary the values of ∇_μ and compare the expected and measured values of the magnetic energy in our simulation. Results are shown in

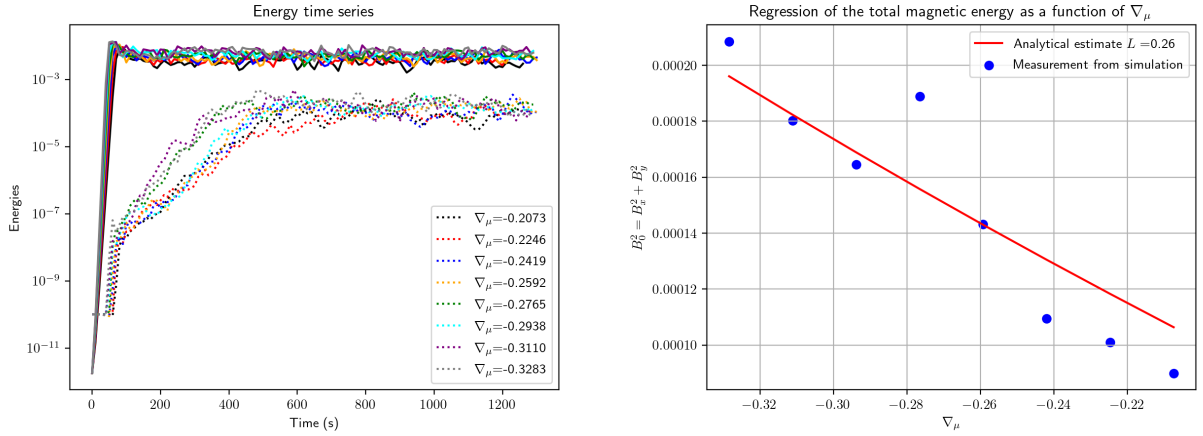


Figure 3.11 – Left : Time series of the mean kinetic (solid) and magnetic (dashed) energies in diabatic convective dynamo simulations with varying ∇_μ. Right : Theoretical prediction and measurement of B_0^2 at saturation, as a function of $\nabla_T - \nabla_{ad}$. The represented value of $B_0^2 = B_x^2 + B_y^2$ corresponds to a time average starting from $t = 1000s$.

figure 3.11 and provide a good correspondence. We perform a fit to obtain the value of the parameter $L = 0.26$, close and consistent to the adiabatic case. We observe one quite unsatisfying result for $\nabla_\mu \simeq -0.27$, and it is unclear why other points show a good correspondence. We suspect that the initial seed may have triggered the apparition of shear modes that polluted the simulation, but further analysis has to be done to clear the matter.

Convergence study for diabatic dynamo

We performed a very large-scale simulation of our convective setup as part of the Grand Challenges on the Adastra supercomputer (CINES, Montpellier, France). We coupled our code with the PDI and Deisa libraries [Roussel et al. 2017 ; Gueroudji et al. 2021] to tackle the I/O bottleneck of our simulation. Indeed, at the full 4096^3 resolution, a save of the solution weighs 5TB and cannot be stored at high frequency for later analysis. Therefore, we implemented several I/O routines to extract slices, vertical averages, and domain averages. The technicalities of the implementation of these high-performance I/O routines are detailed in the next chapter. Given that the simulation starts in a linear regime and considering that this instability requires an extended physical time to reach saturation, we use a checkpoint/restart and upscaling system. This strategy enables us to simulate the instability at a lower resolution, effectively bypassing the linear phase. Once the profiles have reached saturation, we incrementally double the resolution 4 times until the desired final resolution of 4096^3 is attained. The reachable time by a 4096^3 simulation starting from the initial conditions would not be enough to observe the convergence of the dynamo effect. The setup we consider is the one of the last section, except that we pick a lower value of $Q_A = -0.001$ and that we employ a second-order MUSCL Hancock scheme. Figure 3.12 shows the kinetic and magnetic energies time series through several upscaling. It also shows that the magnetic energy increases with the resolution. This is consistent with our estimation (3.31) as the effective value for Q_A is predominated by numerical diffusion on the magnetic field that reduces at each upcaling. Temperature and chemistry sources are also affected, but the effective quantities H_T and R_X are predominated by their large physical values. The kinematic viscosity picked is null in our simulation and the analysis. Therefore, it is also piloted by the numerical diffusion.

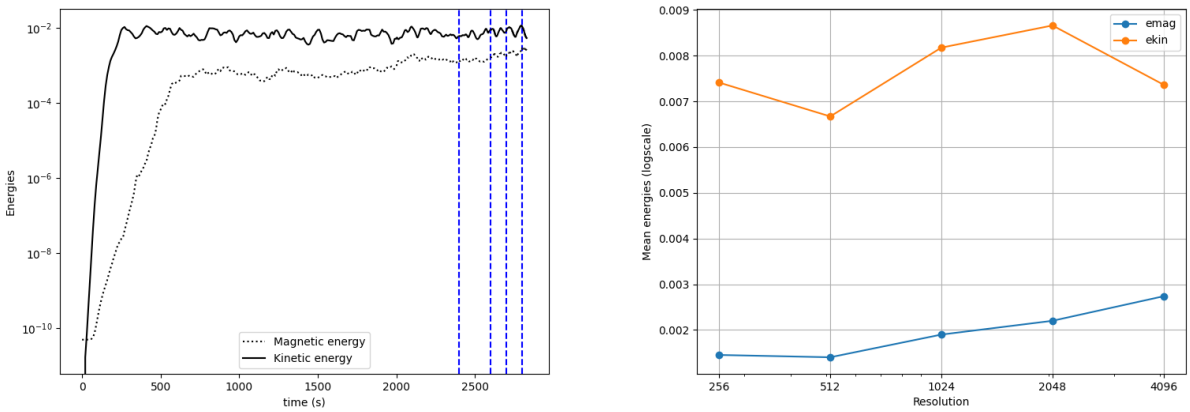


Figure 3.12 – Left : Evolution of the average kinetic and magnetic energies as a function of time. The vertical blue lines correspond to the resolution upscaling. Right : Average kinetic and magnetic energies at each resolution. The first point is computed starting at time $t = 2000s$ until the first upscaling.

It is unclear what is its impact on the dynamo effect we observe. This question will be answered in future work by taking it into account in our linear stability analysis. Figure 3.13 displays the kinetic and magnetic power spectra at various times and resolutions throughout the simulation. Distinct line colors differentiate each resolution. Dashed and solid lines represent the initial and final power spectra computed at the corresponding resolution for each color. It is observed that with increased resolution, the simulation excites more small-scale modes, and the turbulent slope decreases, indicating that kinetic energy is transferred to progressively smaller modes. The slope is consistent with Kolmogorov's cascade $E(k) \propto k^{-5/3}$. We observe that the different upscaling strongly boosts the low-frequency modes. The increase in the power of high-k modes in the magnetic energy suggests an inverse energy cascade. Finally, we discuss the magnetic equipartition. Figure 3.14 illustrates the vertical distribution of potential temperatures and magnetic intensity. The magnetic energy is roughly equipartitionned across all directions ($\pm 2\%$). Furthermore, as anticipated, due to the high values of chemical and thermal source terms, the profiles of potential temperatures deviate minimally from their initial conditions. The adiabatic temperature profile $\log\theta - \log\mu$ is maintained stable (negative gradient), while the diabatic profile $\log\theta - \frac{H_T}{R_X}\log\mu$ is maintained unstable (positive gradient). This allows the dynamo to occur, as it is the only degree of freedom left to saturate the instability. We provide vertical slices of the density perturbation, magnetic energy, and vertical velocities in figures 3.15, 3.16, 3.17. By density perturbation, we refer to the difference between the current and initial density. A descending, mushroom-shaped plume of heavy fluid is visible on the bottom left side of the density plot. This formation is also mirrored in the magnetic energy plot. Additionally, this column of descending fluid is discernible as a vertical stripe of negative vertical velocities in the velocity plot.

3.4 . Discussion

Our criteria, as given by equations (3.7), (3.8), and (3.9), show that shear and magnetic field share striking similarities. In the linear regime, both dampens the convective instability. Moving into the non-linear regime, both have the capability to saturate the criteria through self-generation. A horizontal magnetic field dampens

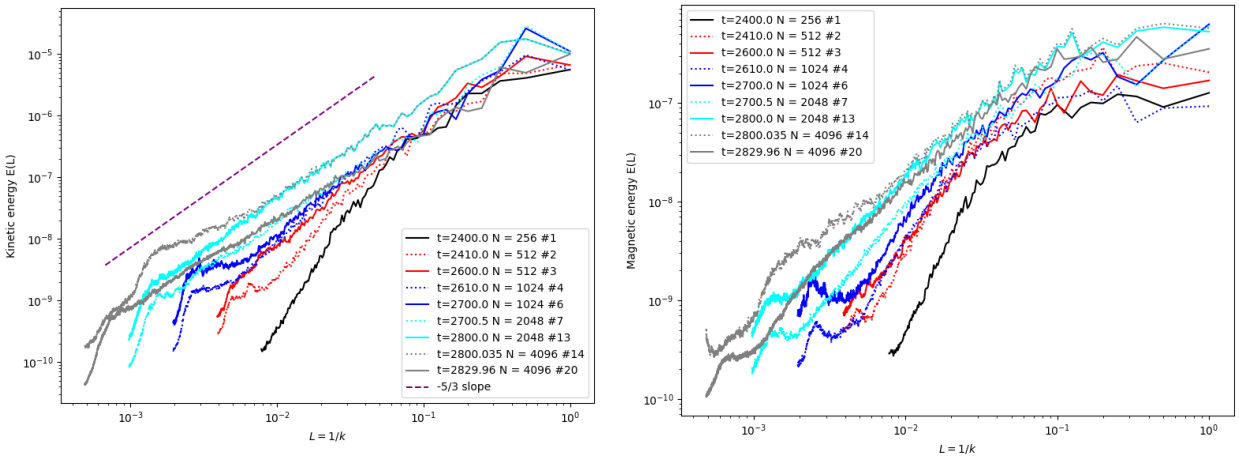


Figure 3.13 – Kinetic and magnetic power spectra at different times/resolutions

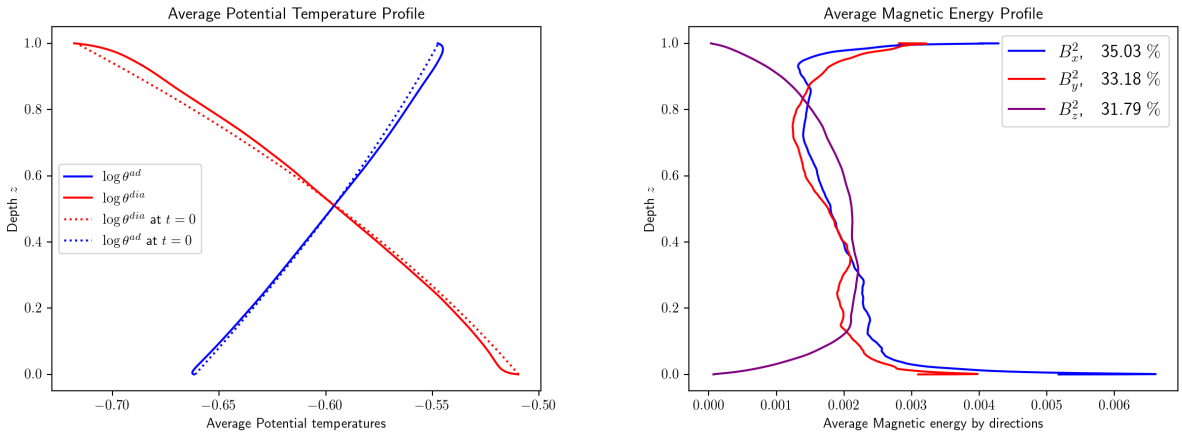


Figure 3.14 – Time average of the vertical profiles of the potential temperatures and magnetic intensities per direction. The potential temperature are purely hydrodynamical $\log\theta^{ad} = \log\theta - \log\mu$ and $\log\theta^{dia} = \log\theta - \frac{H_T}{R_X}\log\mu$. The time average is done on all outputs from the last 4096^3 resolution.

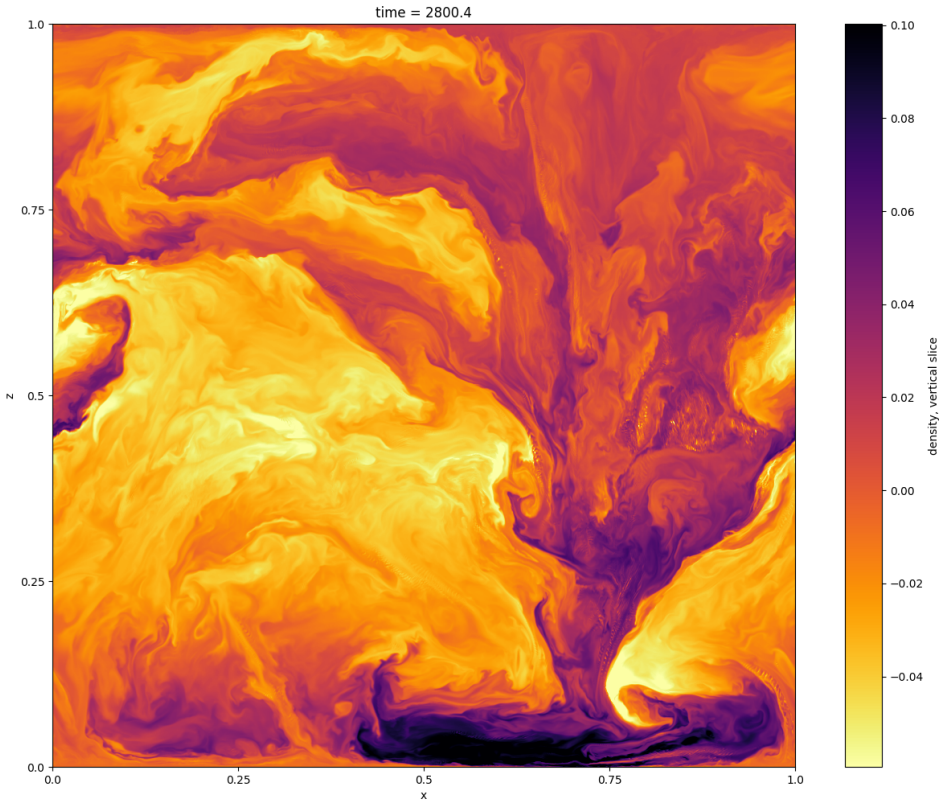


Figure 3.15 – Vertical slice of the density perturbation.

convective vertical speeds through magnetic tension as magnetic field ligns resist bending motions. On the other hand, shear converts vertical momentum to horizontal momentum through horizontal pressure gradients. Furthermore, the geometry of the flow and domain have a significant influence on both shear and magnetic field, as depicted by their geometrical prefactors involving k_x, k_y, k_z. However, a fundamental difference between shear and magnetic fields is the degree of their coupling with the source terms. Shear is coupled with products of $n+1$ source terms, while the magnetic field is coupled with products of n source terms, with $n = 0, 1, 2$ for the adiabatic, diabatic, and double diabatic instabilities, respectively.

In section 3.3.5, we presented dynamo experiments conducted in both adiabatic and diabatic settings. Two key factors enabled the generation of a significant amount of magnetic energy of approximately one-tenth to one-hundredth of the kinetic energy. The first factor is the usage of strong source terms, as shown in equation (3.44), which maintain the thermo-compositional gradients in a state of instability. The second factor is the cubic geometry, which ensures minimal growth of shear effects (as justified in section 3.3.3). This geometry results in competition between the x and y shear, preventing either from dominating the dynamo effect. For real physical cases, we propose that dynamo generation should occur in convective zones with suitable geometries and gra-

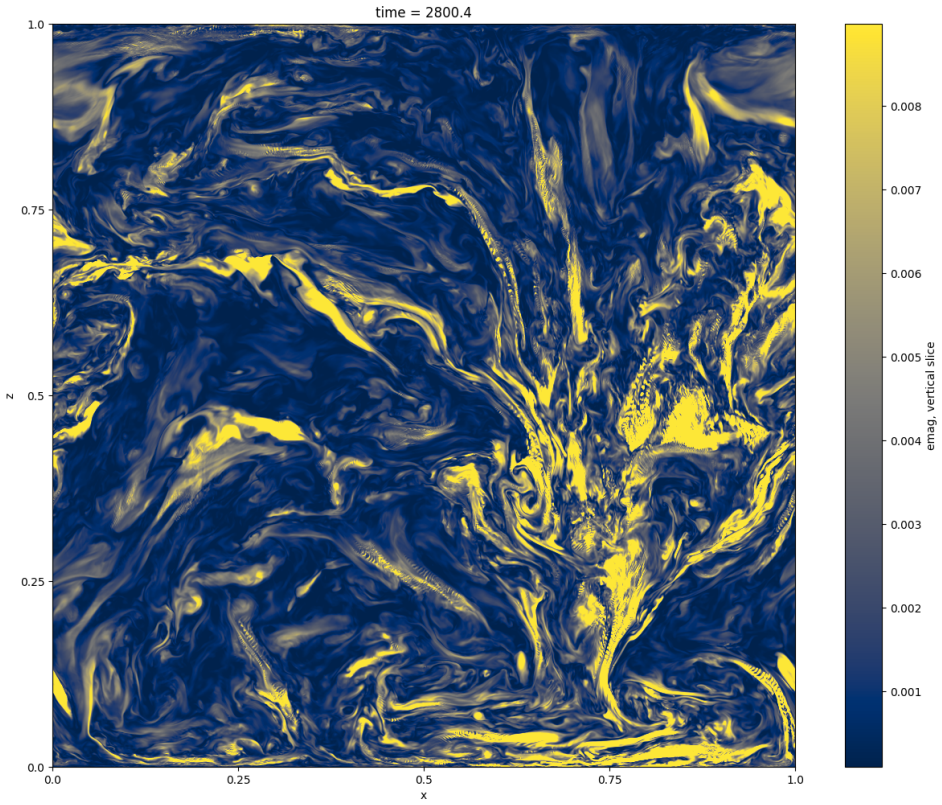


Figure 3.16 – Vertical slice of the magnetic energy.

dients that are maintained unstable by some process, e.g., fast diffusion coupled with strong constraints on the boundary values.

In the first section, we artificially set an absolute value around the shear (setting $\frac{\partial u_0}{\partial z} = \left| \frac{\partial u_0}{\partial z} \right|$) profile by arguing that shear can only have a stabilizing effect on convection. To understand why, one must consider several points and hypotheses :

- Convection is a 2D instability : a necessary condition to destabilize the flow is that the upward motion is unstable **and** the downward motion is also unstable. Conversely, a sufficient condition to stabilize the flow is that the upward motion is stabilized **or** the downward motion is stabilized, but not necessarily both.
- The energy source term is key in the process (similar to thermohaline convection). Hence, we hypothesize here, for the sake of simplicity, that it is infinitely fast, such that a perturbed bubble immediately adjusts its temperature to the environment.
- By Galilean invariance, we can always make the hypothesis that the environment at the top and bottom and the domain are moving in opposite directions. Thus, when moving up or down a perturbed bubble, it will always feel compression in the upwind direction of its horizontal movement because of the vertical

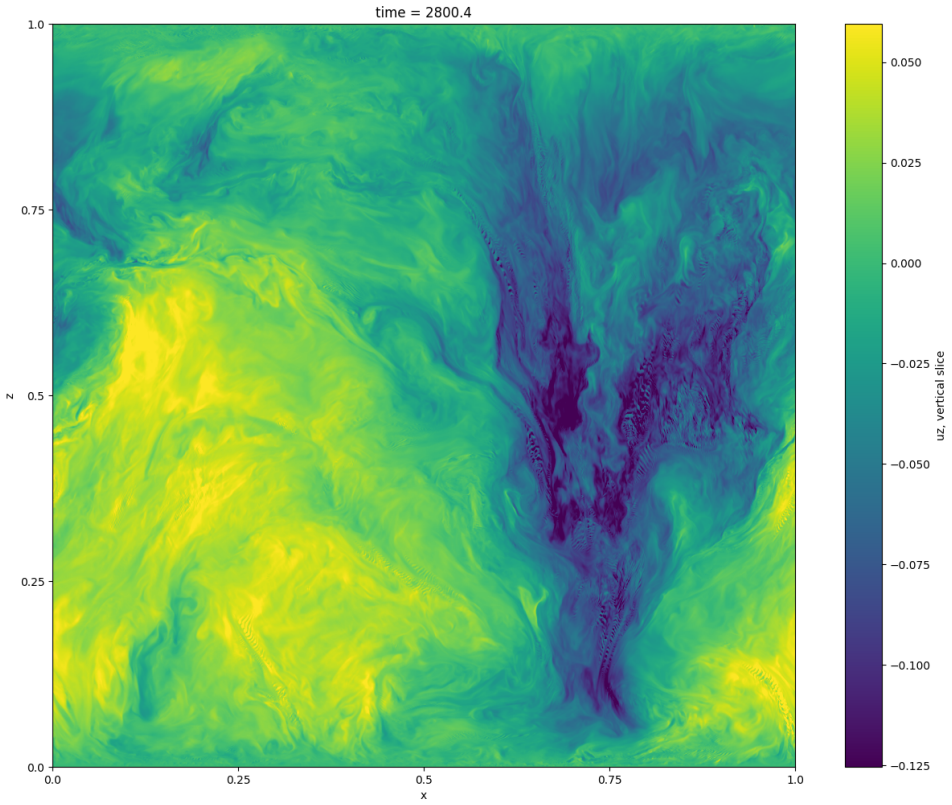


Figure 3.17 – Vertical slice of the vertical velocity.

shear.

Depending on the sign of the vertical shear, we have the two situations presented in Figure. 3.18. In both cases, the bubble perturbed in the upward direction will feel compression, implying an increase in pressure and density when the temperature adjustment is fast. Hence, the increase in density stabilizes the perturbation, and the upward motions are stabilized independently of the sign of the vertical shear : vertical shear can only stabilize convection.

Let us now look closer at the role of the background magnetic field. A common claim is that *a purely horizontal magnetic field cannot stabilize a fluid that is Schwarzschild-unstable* (see [Gough et Tayler 1966; Tayler 1973; Newcomb 1961; Kovetz et Mestel 1967; Yu 1966; Chandrasekhar 1961]). The criteria (3.7), (3.8), (3.9) tell us that the influence of the magnetic field on the growth of the instability is impacted by the geometric prefactor $\frac{k^2 k_x^2}{k_x^2+k_y^2}$. Therefore, given that the flow is Schwarzschild or Ledoux unstable, for any arbitrarily high value of B_0^2, one can consider a small enough value of k_x^2 for which the instability will grow. Moreover, a purely transverse 2D perturbation ($k_x = 0$) does not experience the effect of the magnetic field. This phenomenon is known as the interchange instability. In a 2D context, the lines can be purely transported by convection (interchanged) and, therefore, do

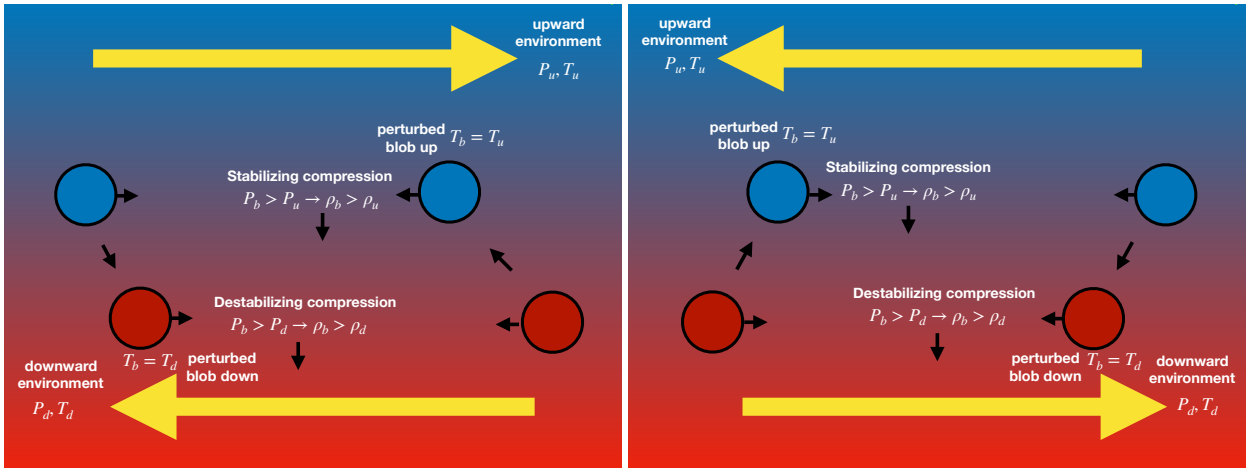


Figure 3.18 – Physical interpretation of the instability, depending on the sign of the vertical shear. Both cases are stabilizing convection.

not affect the process. Our criteria agree that the magnetic field cannot stabilize *the fluid* in the sense that the geometry of the perturbation we consider is not a property of the fluid. However, it can slow or cancel the growth of a given mode in a Schwarzschild unstable fluid. In our convective dynamo simulation, the magnetic field does saturate the criterion and stabilizes convection, as the 3D turbulent flow is not transverse with the also turbulent magnetic field.

We also comment on the finite resistivity case that is addressed in [Chandrasekhar 1961]. The work shows that as soon as resistivity is non 0, the magnetic field can not impact the instability criterion. Let us consider a Schwarzschild unstable flow and background magnetic field without a mean molecular weight gradient. The adiabatic criterion (3.7) becomes :

$$\nabla_T - \nabla_{ad} - \frac{k^2 k_x^2}{k_x^2 + k_y^2} \frac{h_p}{\rho_0 g} B_0^2 > \frac{k^2}{k_x^2 + k_y^2} \frac{h_p}{g} Q_A H_T, \tag{3.32}$$

and the diabatic criterion (3.8) becomes :

$$(\nabla_T - \nabla_{ad}) Q_A - \frac{k^2 k_x^2}{k_x^2 + k_y^2} \frac{h_p}{\rho_0 g} B_0^2 H_T < 0. \tag{3.33}$$

One can observe that as soon as the resistivity Q_A is nonzero and if thermal conduction and all other heating processes H_T are neglected, the diabatic criterion (3.33) reduces to the standard Schwarzschild criterion. In that sense, the horizontal magnetic field does not impact the Schwarzschild criterion in this context. However, the magnetic field can significantly impact the growth rate of instability, reducing it to a very large time scale, which can make the instability negligible in physical applications, as mentioned in the several works we cited. Moreover, when considering a fluid that is also thermally conductive, i.e., $H_T < 0$, it is clear that a strong enough magnetic field can stabilize the growth of a mode even if the fluid is Schwarzschild-unstable.

The double-diffusive thermo-magnetic case is explored in [Newcomb 1961], which references the early works of Rayleigh-Jeffreys [Rayleigh 1916; Jeffreys 1926] that suggested that the interaction of thermal and magnetic diffusion stabilizes thermo-magnetic convection. Our observations align with this in the generalized adiabatic

criterion (3.32), with the presence of the stabilizing product term $Q_A H_T$. However, the diabatic branch (3.33) introduces the possibility of triggering convection if the ratio H_T/Q_A is small enough, a scenario not considered by these early theories, but identified later by [Yu et Cheng 1973]. The initial studies by [Furth et al. 1963; Johnson et al. 1963] did observe the emergence of an additional mode in a fluid with finite resistivity ($Q_A < 0$), aligning with our diabatic branch.

A key takeaway of the present work is that taking into account additional physics and their corresponding source terms seems to systematically generate new instability criteria, which are closely linked to the previously known ones. The diabatic criterion, as discussed in [Tremblin et al. 2019], demonstrates that a Ledoux stable configuration can become unstable if the right source term is chosen sufficiently intense. Consider, for example, a Ledoux stable atmosphere where $\nabla_T - \nabla_{\text{ad}} < \nabla_\mu$, and assume there is no magnetic field or magnetic source term. When $H_T \to -\infty$, equation (3.8) simplifies to $-\nabla_\mu H_T < 0$, which is equivalent to $-\nabla_\mu > 0$. In cases where the limit of a source term is infinite (or much faster compared to others), the diabatic criterion (3.8) reduces to the adiabatic criterion (3.7), but ignoring the corresponding gradient. This relationship is similarly observed between the double diabatic and diabatic criteria, as shown in (3.9) and (3.8). For instance, consider a system stable under the diabatic criterion due to a significant background magnetic field. In the limit $Q_A \to -\infty$, the double diabatic criterion (3.9) reduces to $(\nabla_T - \nabla_{\text{ad}})R_X - \nabla_\mu H_T < 0$, i.e., the magnetic-field-less diabatic criterion. It is important to note that the roles of chemistry, temperature, and magnetic field can be straightforwardly interchanged.

Summarizing the approximations employed

In this section, we address the several approximations we adopt to derive the instability criteria. We utilize an ideal EOS for its simplicity. For specific physical systems, one must apply a more realistic EOS. The analysis takes place within the Boussinesq regime, removing pressure perturbations in the EOS but not from the momentum equation, and assuming that the perturbed velocity field is incompressible. We limit our scope to a horizontal and constant background magnetic field to enable the analysis of a linear system in three dimensions without imaginary terms. The shear profile and background magnetic field are set along the same direction, x. This is not a problem here as we do not focus on the interaction between the two effects. A thorough analysis would introduce an angle between them. We do not investigate the implications of employing magnetic field-dependent heating and reaction source terms (H_A and R_A), as well as a Q_T or Q_X. Lastly, we do not study the influence of kinematic viscosity on the instability. Such investigations are reserved for future studies and will likely provide very insightful criteria. For the non-linear regime, we assumed that the statistically stationnary state is a small perturbation around a hydrostatic equilibrium.

Reproducing the numerical experiments

The open-access code used to conduct the numerical experiments in the present chapter is available at `https://gitlab.erc-atmo.eu/remi.bourgeois/ark-2-mhd.git`. It implements the finite volume method described above using the Kokkos library, along with MPI and PDI, to manage multi-GPU parallelism for computations and I/O. All the simulation parameters files are located in the subdirectory `numerical_experiment_convection_paper`.

3.5 . Conclusion

In this chapter, we have performed a linear stability analysis and a non linear extension for magnetohydrodynamical convection, taking into account arbitrary source terms on the temperature, chemistry and magnetic field. As a result, we obtained stability criteria that encapsulate classical instability criteria from the literature but also a new, previously unknown criterion involving products of pairs of source terms partial derivatives (e.g., dissipation rates) with the background gradients. The non-linear theory is based on the assumption that stabilized convection is a small perturbation of a hydrostatic equilibrium. Numerical evidence of the manifestation of this new instability was provided. The role of the background shear and magnetic field were studied in both linear and non-linear regimes. We conducted convective dynamo numerical experiments and linked the results of our theory. We also studied the impact of the geometry of the box on the development of shear modes. Future work will include the development of a proper mixing length theory from our analysis. This will allow leveraging our framework by using it into 1D atmospherical codes, applying it to existing astrophysical bodies. taking into account an arbitrary orientation of the background magnetic field with respect to the shear profile, to study their interaction. Finally, we plan on adding kinematic viscosity (via an arbitrary source term on the momentum) to study its influence on convective dynamo. In the next chapter, we describe the work that went into conducting the large-scale convection simulation that we presented in section 3.3.5. In particular, we describe how we integrated modern I/O tools into our HPC code to deal with the large amount of data coming from the simulation. We also show the performance results of our code on the new Adastra supercomputer and compare them to other architectures.

Acknowledgements

We thank the Centre Informatique National de l'Enseignement Supérieur for the 210,000 GPU hours allocation used to run the large-scale convection simulation as part of the "Grand Challenges"

Appendix

3.A . Obtaining the matrix and deriving the criteria

3.A.1 . Obtaining the matrix

The linearization of the MHD system with gravity, compositional, energy, and magnetic source terms in the Boussinesq regime leads to the following system :

$$\begin{aligned}
\nabla \cdot (\delta \boldsymbol{u}) &= 0, \\
\rho_0 \partial_t \delta \boldsymbol{u} + u_0(z)\rho_0 \partial_x \delta \boldsymbol{u} + \delta w \rho_0 \frac{\partial u_0(z)}{\partial z} \boldsymbol{e_x} - B_0(z)\partial_x \delta \boldsymbol{B} - \delta B_z \frac{\partial B_0(z)}{\partial z} \boldsymbol{e_x} & \\
+\nabla\left(\delta P + B_0(z)\delta B_x\right) - \delta\rho \boldsymbol{g} &= 0, \\
\partial_t \delta T + u_0(z)\partial_x \delta T + \delta w \cdot \left(\frac{\partial T_0}{\partial z} - \frac{\gamma - 1}{\gamma}\frac{T_0}{P_0}\frac{\partial P_0(z)}{\partial z}\right) &= H_T \delta T + H_X \delta X + \boldsymbol{H_A} \cdot \delta \boldsymbol{A}, \\
\partial_t \delta \boldsymbol{A} + u_0(z)\partial_x \delta \boldsymbol{A} + \delta w \frac{\partial A_0(z)}{\partial z} \boldsymbol{e_y} &= \boldsymbol{Q_A} \delta \boldsymbol{A}, \\
\partial_t \delta X + u_0(z)\partial_x \delta X + \delta w \frac{\partial X_0}{\partial z} &= R_T \delta T + R_X \delta X, \\
\frac{\delta P}{P_0} &= \frac{\delta \rho}{\rho_0} + \frac{\delta T}{T_0} - \frac{\partial \log \mu_0}{\partial X}\delta X.
\end{aligned} \tag{3.34}$$

We then assume a single-mode ansatz for the perturbation : $\delta q = |\delta q| \exp(\omega t + i(k_x x + k_y y + k_z z))$ and obtain the following set of equations :

$$\begin{aligned}
\delta w \left(k_x \rho_0 \frac{\partial u_0(z)}{\partial z}\right) + \delta \rho k_z g + \delta P i k^2 + \delta A_y \left(-i(k_x^2 + k_z^2)\frac{\partial B_0(z)}{\partial z} + B_0(z) k^2 k_z\right) &= 0, \\
(\omega + i k_x u_0)\delta X + \delta w \frac{\partial X_0}{\partial z} &= R_X \delta X + R_T \delta T, \\
(\omega + i k_x u_0)\rho_0 \delta w + \delta A_y \left((k_x^2 + k_z^2)B_0(z) - i k_z \frac{\partial B_0(z)}{\partial z}\right) + (\omega + i k_x u_0)\delta T & \\
+\delta w \left(\frac{\partial T_0(z)}{\partial z} - \frac{\gamma - 1}{\gamma}\frac{T_0}{P_0}\frac{\partial P_0(z)}{\partial z}\right) &= H_X \delta X + H_T \delta T + H_{A_y} \delta A_y, \\
\delta A_x = \delta A_y &= 0, \\
\frac{\delta P}{P_0} &= \frac{\delta \rho}{\rho_0} + \frac{\delta T}{T_0} - \frac{\partial \log \mu_0}{\partial X}\delta X.
\end{aligned} \tag{3.35}$$

We define $1/h_p = -\frac{\partial \log P_0}{\partial z}$, $\nabla_T = -h_p \frac{\partial \log T_0}{\partial z}$, $\nabla_{ad} = \frac{\gamma-1}{\gamma}$, $Q_A = \frac{\partial Q_y}{\partial A_y}$, $H_A = \frac{\partial H}{\partial A_y}$ and perform the translation $(\omega + ik_x u_0) \rightarrow \omega$:

$$\begin{aligned}
\delta w \left(k_x \rho_0 \frac{\partial u_0(z)}{\partial z} \right) + \delta \rho k_z g + \delta P i k^2 + \delta A_y \left(-i(k_x^2 + k_z^2) \frac{\partial B_0(z)}{\partial z} + B_0(z) k^2 k_z \right) &= 0, \\
\omega \delta X + \delta w \frac{\partial X_0}{\partial z} - R_X \delta X - R_T \delta T &= 0, \\
\omega \rho_0 \delta w + \delta A_y \left((k_x^2 + k_z^2) B_0(z) - i k_z \frac{\partial B_0(z)}{\partial z} \right) + i k_z \delta P + \delta \rho g &= 0, \\
\omega \delta T - \delta w \frac{T_0}{h_p} (\nabla_T - \nabla_{ad}) - H_X \delta X - H_T \delta T - H_A \delta A_y &= 0, \\
\omega \delta A_y - \delta w B_0(z) - Q_A \delta A_y &= 0, \\
\frac{\delta \rho}{\rho_0} + \frac{\delta T}{T_0} - \frac{\partial \log \mu_0}{\partial X} \delta X &= 0.
\end{aligned} \tag{3.36}$$

We re-write this system of equations as a linear system $\boldsymbol{M}(\omega)\boldsymbol{\delta x} = \boldsymbol{0}$ with :

$$\boldsymbol{\delta x} = \begin{pmatrix} \delta\rho & \delta X & \delta w & \delta T & \delta P & \delta A_y \end{pmatrix}^T, \tag{3.37}$$

and the matrix

$$\boldsymbol{M}(\omega) = \begin{pmatrix}
k_z g & 0 & k_x \rho_0 \frac{\partial u_0}{\partial z} & 0 & ik^2 & -i(k_x^2 + k_z^2) \frac{\partial B_0(z)}{\partial z} + k_z k^2 B_0(z) \\
0 & \omega - R_X & \frac{\partial X_0}{\partial z} & -R_T & 0 & 0 \\
g & 0 & \rho_0 \, \omega & 0 & ik_z & (k_x^2 + k_z^2) B_0(z) - i k_z \frac{\partial B_0(z)}{\partial z} \\
0 & -H_X & -\frac{T_0}{h_p}(\nabla_T - \nabla_{ad}) & \omega - H_T & 0 & -H_A \\
\frac{-1}{\rho_0} & \frac{\partial log \mu_0}{\partial X_0} & 0 & -\frac{1}{T_0} & 0 & 0 \\
0 & 0 & -B_0(z) & 0 & 0 & \omega - Q_A
\end{pmatrix} \tag{3.38}$$

In this section, we look for conditions the roots of $P(\omega) = \text{Det } \boldsymbol{M}(\omega) = 0$ have at least one positive real solution (i.e., exponential growth of the perturbation; an instability). We limit ourselves to the study of the sign of the determinant's coefficients, using Hurwitz's criteria. We observe that the determinant is real only in a two-dimensional setup ($k^2 = k_x^2 + k_z^2$) or if $\frac{\partial B_0(z)}{\partial z} = 0$. Note that in the case where $\boldsymbol{Q}$ models magnetic resistivity, $Q_A = -k^2 \nu$ and $\frac{\partial B_0(z)}{\partial z} = 0$. To keep the study three-dimensional, we choose to neglect local background magnetic field gradients $\frac{\partial B_0(z)}{\partial z} = 0$. This can be justified by two of the following arguments : -The magnetic field variation being on a much larger scale than the other quantities -The source term on the magnetic field being magnetic resistivity. Note that in the case of Ohmic heating, $H(A_y) = \mu(\Delta A_y)^2 = \mu(\Delta \delta A_y)^2 = \nu k^4 \delta A_y^2 \sim 0$; In the linear regime, the contribution of Ohmic heating is negligible. We therefore set $H_A = 0$ as it has no influence on the instability.

3.A.2 . Thermo-magneto-compositional sheared convection criteria

The determinant of the matrix (3.38) is given by $P(\omega)/ik^2 = a_0 + a_1\omega + a_2\omega^2 + a_3\omega^3 + \omega^4$. Hurwitz's criterion ensures that if for any $i \in [0, 3]$, $a_i < 0$, then P has at least one root ω with a positive real part. We

examine the four coefficients :

$$
\begin{aligned}
a_3 &= -H_T - Q_A - R_X - \frac{k_x k_z}{k^2}\frac{\partial u_0}{\partial z},\\
a_2 &= k_x^2\frac{B_0^2}{\rho_0} - H_X R_T + Q_A R_X + H_T(Q_A + R_X) - \frac{k_x^2+k_y^2}{k^2}\frac{g}{h_p}(\nabla_T - \nabla_{ad} - \nabla_\mu)\\
&\quad + \frac{k_x k_z}{k^2}\frac{\partial u_0}{\partial z}(H_T + Q_A + R_X),\\
a_1 &= -k_x^2\frac{B_0^2}{\rho_0}(H_T + R_X) - Q_A(H_T R_X - H_X R_T) + \frac{k_x^2+k_y^2}{k^2}\frac{g}{h_p}((\nabla_T - \nabla_{ad})(\omega'_X + Q_A)\\
&\quad - \nabla_\mu(\omega'_T + Q_A)) - \frac{k_x k_z}{k^2}\frac{\partial u_0}{\partial z}(H_T Q_A - H_X R_T + R_X(H_T + Q_A)),\\
a_0 &= k_x^2\frac{B_0^2}{\rho_0}(H_T R_X - R_T H_X) - Q_A\frac{k_x^2+k_y^2}{k^2}\frac{g}{h_p}((\nabla_T - \nabla_{ad})\omega'_X - \nabla_\mu\omega'_T)\\
&\quad + Q_A\frac{k_x k_z}{k^2}\frac{\partial u_0}{\partial z}(H_T R_X - H_X R_T).
\end{aligned}
\tag{3.39}
$$

Where $\omega'_X = R_X + T_0 R_T \frac{\partial \log \mu_0}{\partial X}$, $\omega'_T = H_T + \frac{1}{T_0} H_X \left(\frac{\partial \log \mu_0}{\partial X}\right)^{-1}$ and $\nabla_\mu = -h_p \frac{\partial \log \mu_0}{\partial z}$. Since a shear-only profile cannot induce a convective instability, we have to assume $\frac{\partial u_0(z)}{\partial z} < 0$ and note $\frac{\partial u_0(z)}{\partial z} = -\left|\frac{\partial u_0(z)}{\partial z}\right|$. Realistic source terms satisfy $H_T, Q_A, R_X < 0$. The coefficients provide three instability criteria. $a_2 < 0$ gives :

$$
\nabla_T - \nabla_{ad} - \nabla_\mu - \frac{k^2 k_x^2}{k_x^2+k_y^2}\frac{h_p}{\rho_0 g}B_0^2 - \frac{k_x k_z}{k_x^2+k_y^2}\frac{u_0}{g}\nabla_u(H_T + Q_A + R_X) - \frac{k^2}{k_x^2+k_y^2}\frac{h_p}{g}S_{\text{ad}} > 0, \tag{3.40}
$$

with , $\nabla_u = -h_p \frac{1}{u_0}\left|\frac{\partial u_0}{\partial z}\right| < 0$ and $S_{\text{ad}} = -H_X R_T + Q_A R_X + H_T(Q_A + R_X) > 0$ (if cross-source terms derivatives are small). This is the adiabatic thermo-magneto-compositional criterion. $a_1 < 0$ gives :

$$
\begin{aligned}
&(\nabla_T - \nabla_{ad})(\omega'_X + Q_A) - \nabla_\mu(\omega'_T + Q_A) - \frac{k^2 k_x^2}{k_x^2+k_y^2}\frac{h_p}{\rho_0 g}B_0^2(H_T + R_X) - \frac{k_x k_z}{k_x^2+k_y^2}\frac{u_0}{g}\nabla_u S_{\text{ad}}\\
&- \frac{k^2}{k_x^2+k_y^2}\frac{h_p}{g}Q_A S_{\text{dia}} < 0,
\end{aligned}
\tag{3.41}
$$

with $S_{\text{dia}} = H_T R_X - H_X R_T > 0$. This is the diabatic thermo-magneto-compositional criterion. $a_0 < 0$ gives :

$$
\left((\nabla_T - \nabla_{ad})\omega'_X - \nabla_\mu\omega'_T\right)Q_A - \frac{k^2 k_x^2}{k_x^2+k_y^2}\frac{h_p}{\rho_0 g}B_0^2 S_{\text{dia}} - \frac{k_x k_z}{k_x^2+k_y^2}Q_A\frac{u_0}{g}\nabla_u S_{\text{dia}} > 0, \tag{3.42}
$$

this is the double diabatic thermo-magneto-compositional criterion.

3.B . Numerical setup

3.B.1 . Initial conditions

The atmosphere is originally at hydrostatic equilibrium $\nabla P_0(z) = -\rho_0(z)g$ with a linear temperature and mass mixing ratio profile $T_0(z) = T_{gnd} + z\nabla T$, $X_0(z) = X_{gnd} + z\nabla X$ and the ideal gas EOS $P_0(z) =$

$\rho_0(z)k_BT_0(z)/\mu(X_0(z))$. We use $1/\mu(X) = X/\mu_1 + (1-X)/\mu_2$ for the mean molecular weight. To initialize the density, we start with a bottom value ρ_{gnd} and integrate vertically using

$$\frac{P_{i+1}-P_i}{\Delta z} = -g\frac{\rho_{i+1}+\rho_i}{2} \leftrightarrow \rho_{i+1} = \rho_i \frac{\left(\frac{T_0(z_i)}{\mu(z_i)} - \frac{g}{2k_B}\right)}{\left(\frac{T_0(z_{i+1})}{\mu(z_{i+1})} + \frac{g}{2k_B}\right)}, \quad \boldsymbol{u}_i = 0 \tag{3.43}$$

which is a fully explicit recurrence expression for the ρ_i series as T_0 and μ_0 can be computed explicitly at any altitude z_i. One can perform the sanity check that for an isothermal atmosphere with $\nabla T = \nabla X = 0$, it holds that $\rho_{i+1} < \rho_i$. The pressure is then locally computed as $P_i = \rho_i k_B T_0(z_i)/\mu(X_0(z_i))$. The constant background magnetic field is simply initialized as $\boldsymbol{B} = (B_0, 0, 0)^T$. The velocity is set as $\boldsymbol{u}_0 = (0, 0, \delta w)^T$ where $\delta w(x, y, z)$ is the initial perturbation.

3.B.2 . Source terms employed

For our numerical experiments, we employ the following source terms

$$\partial_t \boldsymbol{B} = Q_A(\boldsymbol{B} - \boldsymbol{B}_0), \tag{3.44}$$
$$\partial_t T = H_T(T - T_0), \tag{3.45}$$
$$\partial_t X = R_X(X - X_0) \tag{3.46}$$

i.e., a linear relaxation towards the initial conditions. Consequently, for strong values of the partial derivatives of the source terms, the mixed quantities are forced to maintain their initial values, similarly to [Tremblin et al. 2019; Daley-Yates et al. 2021]. Our experiment does not involve crossed derivatives of the source terms H_X, R_T. As a result, our setup is entirely defined by the following set of parameters :

$$W = (g,\ \rho_{gnd},\ T_{gnd},\ \nabla T,\ X_{gnd},\ \nabla X,\ k_B,\ \gamma,\ B_0,\ \delta w, H_T, Q_A, R_X, \mu_1, \mu_2)\,. \tag{3.47}$$

3.B.3 . Numerical scheme - Ideal MHD

All our numerical simulations employ the finite volume method presented in [Tremblin et al. 2024]/ chapter 2. Specifically, we utilize the *1-cell stencil,* $3+1$ *waves* version of the solver. *1-cell stencil* refers to our choice of the flux-splitting version of the method, as opposed to the operator-splitting version. Both methods are presented in [Tremblin et al. 2024]/chapter 2, and the procedure to recast one into the other is detailed in [Bourgeois et al. 2024]/chapter 1 for the hydrodynamic case. The *1-cell stencil* approach offers several benefits, including better efficiency in low Mach regimes and ease of implementation. The *3+1 waves* choice indicates that we employ only $3+1$ waves in the derivation of the approximate Riemann solver for Lagrangian MHD, which corresponds to $\pm\rho c_{fm}$, the Lagrangian fast magneto-acoustic waves, the 0 wave and the $-u$ contact wave, where the normal component of the magnetic field jumps. Note : We employed the 3+1 waves solver instead of the 5+1 waves solver in our numerical experiments. This is due to the fact that we derived the final polished version of the 5+1 waves solver presented in chapter 2, only after performing our numerical experiments for convection. Indeed, it took us more time to derive a reliable 5+1 waves solver than a 3+1 waves one. The Powell-like source terms discussed in [Tremblin et al. 2024] are unnecessary for our purposes, given that the convective regime under consideration is characterized by a high plasma beta and low Alfvèn number. The method's principal advantage lies in its capacity to stably execute multi-dimensional MHD cell-centered simulations without the need for constrained transport

or divergence cleaning. We adapt the numerical method by incorporating a well-balanced treatment of gravity, analogous to the approaches in [Padioleau 2020; Padioleau et al. 2019; Bourgeois et al. 2024; Bouchut 2004; Chalons et al. 2016b; Del Grosso et Chalons 2021; Chalons et Del Grosso 2022], ensuring that when (3.43) holds, the interface velocities computed in the Riemann solver equal zero. Denoting n as the normal direction of the interface under consideration and $\boldsymbol{P} = (P + \frac{1}{2}(B_x^2 + B_y^2 + B_z^2))\boldsymbol{e_n} - B_n(B_x, B_y, B_z)$, we define the interface velocities :

$$\boldsymbol{u}^* = \frac{\boldsymbol{u_R} + \boldsymbol{u_L}}{2} - \frac{1}{c}\left(\frac{\boldsymbol{P_R} - \boldsymbol{P_L}}{2} + \boldsymbol{g}\frac{\rho_R - \rho_R}{2}\right), \tag{3.48}$$

(3.49)

which vanishes when (3.43) is met, enabling the accurate simulation of perturbations around this state. We also implement a low Mach correction following [Dauvergne et al. 2008; Chalons et al. 2016a, 2017; Dellacherie et al. 2016; Padioleau et al. 2019; Bourgeois et al. 2024], which consists in a reduction in numerical diffusion proportional to the local Mach number Ma :

$$\boldsymbol{P}^* = \frac{\boldsymbol{P_R} + \boldsymbol{P_L}}{2} - \frac{c\theta}{2}\frac{\boldsymbol{u_R} - \boldsymbol{u_L}}{2}, \tag{3.50}$$

by selecting $\theta \propto Ma$ locally. These modifications enhance the sharp capture of the convective instability, allowing for the precise discretization of perturbations around hydrostatic equilibrium. Moreover, employing a finite volume method guarantees the conservation of conservative quantities, thereby facilitating a fair simulation of the dynamo effect as no magnetic field is injected into the domain through the scheme.

3.B.4 . Numerical scheme - Sources

Following [Tremblin et al. 2019; Daley-Yates et al. 2021], we employ an implicit scheme following the hyperbolic finite volume update :

$$\frac{\boldsymbol{B}^{n+1} - \boldsymbol{B}^{\text{ad}}}{\Delta t} = Q_A(\boldsymbol{B}^{n+1} - \boldsymbol{B}_0), \tag{3.51}$$

$$\frac{T^{n+1} - T^{\text{ad}}}{\Delta t} = H_T(T^{n+1} - T_0), \tag{3.52}$$

$$\frac{X^{n+1} - X^{\text{ad}}}{\Delta t} = R_X(X^{n+1} - X_0), \tag{3.53}$$

which allows the time step to be selected based solely on the CFL condition of the finite volume scheme. Next, we recalculate the updated total energy with the new magnetic energy $e_{mag}^{n+1} = \frac{\boldsymbol{B}^{n+1\,2}}{2}$ and the new pressure $p^{n+1} = \frac{\rho^{\text{ad}} k_B T^{n+1}}{\mu(X^{n+1})}$. It is important to note that using this source term with a nearly zero background initial magnetic field $\boldsymbol{B_0}$ constitutes a magnetic field well.

3.B.5 . Boundary conditions

The x and y boundary conditions are periodic. The top and bottom conditions require a specific treatment. We focus on the top boundary here, but the parallel with the bottom case is straightforward. Let us denote N as the z index of the upper cells of the domain. First, we linearly extrapolate the temperature and mass mixing ratio to obtain the values in the ghost cell, i.e., $T_{\text{ghost}} - T_N = T_N - T_{N-1}$. Then, we compute the density and pressure using (3.43) with $L = N, R =$ ghost to enforce hydrostatic equilibrium. However, as

the simulation evolves from the initial conditions, the velocities may become non-zero. To ensure $u_z^* = 0$ at the boundary at all times, we invert the z component of the velocity $u_{z,\text{ghost}} = -u_{z,N}$ and replicate the z component of the magnetic field $B_{z,\text{ghost}} = B_{z,N}$ to maintain a zero magnetic pressure gradient at the boundary. An informed choice must be made for the vertical components of the magnetic field and velocities in the ghost cell. We can compute that the B_x flux through the z boundary is proportional to the x component of $\boldsymbol{u^*}$, $u_x^* = \frac{u_{x,\text{ghost}}+u_{x,N}}{2} - \frac{-B_z B_{x,\text{ghost}}+B_z B_{x,N}}{2c}$, while the ρu flux through the z boundary is proportional to the x component of $\boldsymbol{P^*}$, $P_x^* = -\frac{B_z B_{x,\text{ghost}}+B_z B_{x,N}}{2}$ (the diffusion part of the interface pressure at the boundary is always zero, thanks to the low Mach correction θ that equals zero as the local Mach number is computed with $u_z^* = 0$). As a result, to be fully conservative with respect to vertical momentum $\rho u, \rho v$, one must invert the vertical components of the magnetic field $B_{x,\text{ghost}} = -B_{x,N}, B_{y,\text{ghost}} = -B_{y,N}$, but this is at the loss of the magnetic field conservation. To preserve the vertical magnetic field B_x, B_y, one must invert the sign of the vertical components of the velocities $u_{\text{ghost}} = -u_N, v_{\text{ghost}} = -v_N$ and replicate the vertical components of the magnetic field $B_{x,\text{ghost}} = B_{x,N}, B_{y,\text{ghost}} = B_{y,N}$. Thus, it is not possible to simultaneously conserve both the magnetic field and momentum influx at the boundary. In our convective dynamo simulations, we choose to conserve the magnetic field up to machine precision as we aim to perform "fair" simulations of convective dynamo by generating magnetic energy with a constant total magnetic field. For purely hydrodynamic simulation aiming at studying shear, we choose to conserve horizontal momentum.

3.C . Deriving the total energy evolution equation

We start from the definition of the potential temperature $\theta = T\left(\frac{P_{ref}}{P}\right)^{\frac{\gamma-1}{\gamma}}$. Since $e = c_v T$ and $p = \rho e(\gamma - 1)$, we have $\theta = \frac{e^{1/\gamma}}{c_v}\left(\frac{P_{ref}}{\rho(\gamma-1)}\right)^{\frac{\gamma-1}{\gamma}}$. Taking the log and differentiating, we get $d\log\theta = \frac{de}{\gamma e} + \frac{\gamma-1}{\gamma}\frac{d\tau}{\tau} - d\log c_v = \frac{1}{c_v T\gamma}\left(de + pd\tau\right) - d\log c_v$. Using Gibb's relation, we get $T\log\theta = \frac{Tds}{c_v T\gamma} - d\log c_v$. We impose the definition of the source term H as $D_t \log\theta = H/T$. Moreover, $D_t \log c_v = -D_t \log\mu = \frac{\partial\log\mu}{\partial X} D_t X = \frac{\partial\log\mu}{\partial X} R$, giving us the following source term on the entropy $TD_t s = c_v\gamma\left(H - T\frac{\partial\log\mu}{\partial X}R\right)$. We want to derive the corresponding source term on the internal energy i.e. S in $D_t e = -p(\nabla\cdot\boldsymbol{u}) + S$. Since $TD_t s = D_t e + pD_t\tau = -p(\nabla\cdot\boldsymbol{u}) + S + p(\nabla\cdot\boldsymbol{u})$, this gives us $S = c_v\gamma\left(H - T\frac{\partial\log\mu}{\partial X}R\right)$. As a result, we get the following total energy equation $\frac{\partial\rho\mathcal{E}}{\partial t} + \nabla\cdot\left(\left(\rho\mathcal{E} + P + \frac{1}{2}\boldsymbol{B}^2\right)\boldsymbol{u} - (\boldsymbol{B}\cdot\boldsymbol{u})\boldsymbol{B}\right) = \rho c_v\gamma\left(H - T\frac{\partial\log\mu}{\partial X}R\right) + \boldsymbol{B}\cdot\nabla\times\boldsymbol{Q}$.

3.D . Obtaining the potential vector evolution equation

We start with the induction equation written in non-conservative form $\frac{\partial\boldsymbol{B}}{\partial t} - \nabla\times\boldsymbol{u}\times\boldsymbol{B} = 0$. We use the definition of the potential vector $\boldsymbol{B} = \nabla\times\boldsymbol{A}$ and get $\nabla\times\frac{\partial\boldsymbol{A}}{\partial t} - \nabla\times\boldsymbol{u}\times\nabla\times\boldsymbol{A} = 0$. Then, we recall that if $\boldsymbol{f}, \boldsymbol{g}$ are two vector functions, and if $\nabla\times\boldsymbol{f} = \nabla\times\boldsymbol{g}$ then there exist a scalar h which satisfies $\boldsymbol{f} = \boldsymbol{g} + \nabla h$. Therefore, we can "uncurl" our induction equation and introduce a scalar gauge ϕ such that $\frac{\partial\boldsymbol{A}}{\partial t} = \boldsymbol{u}\times\nabla\times\boldsymbol{A} + \nabla\phi$. Using vector identities, we get $\frac{\partial\boldsymbol{A}}{\partial t} = \nabla(\boldsymbol{u}\cdot\boldsymbol{A}) - (\boldsymbol{u}\cdot\nabla)\boldsymbol{A} + \nabla\phi$. By fixing the gauge $\phi = -\boldsymbol{u}\cdot\boldsymbol{A}$, we get the evolution equation used in this chapter.

4 - The Dynostar Grand Challenge on *Adastra*

4.1 . Introduction

In the previous chapter, we conducted analytical and numerical work on the convective instability in MHD. One notable result is the large-scale simulation we performed on the *Adastra* supercomputer as part of a series of "Grand Challenges". The *Adastra* system, ranking 11th on the Top500 and 3rd on the Green500 lists, possesses the same architecture as Frontier, the first ever exascale supercomputer, simply with fewer nodes. The goals of the Grand Challenge presented in this chapter are threefold :

- **evaluating the GPU partition of the Adastra supercomputer :** this partition includes 338 nodes, each equipped with an AMD Trento EPYC 7A53 64-core 2.0 GHz processor and four AMD Instinct MI250X accelerators. We evaluate the performance of our code on these GPUs and compare it to similar simulations performed on NVIDIA's A100, V100, P100, and K80 GPUs from the Ruche (Mésocentre Paris-Saclay) and Mdlslx83 (Local cluster at Maison de la Simulation) machines. Additionally, our code can scale to multiple GPUs. The Grand Challenge is an opportunity to test *Adastra* 's stability when running a full-scale application,
- **integrating and testing modern I/O tools : PDI and Deisa :** as we transition into the exascale era, computing capabilities evolve much faster than data storage capabilities, leading to the "I/O bottleneck". It is becoming impossible to store the full large-scale simulation outputs at regular intervals for later analysis. In our case, our maximum resolution is 4096^3 cells, for a total of 5TB of data per output. To tackle this challenge, we coupled our simulation code ARK^2-MHD with the Parallel Data Interface (PDI, Parallel Data Interface [Roussel et al. 2017]) and Dask-Enabled In-Situ Analysis (Deisa, Dask-enabled in situ analysis, [Gueroudji et al. 2021]) libraries. In this chapter, we detail the coupling process and showcase Deisa's capabilities by executing an in situ Dask-based Fast Fourier transform on running simulation data,
- **testing the physics of convective dynamo at a very high resolution :** this Grand Challenge presents an opportunity to execute a 3D convective dynamo finite volume simulation at an unprecedented scale. This enables us to evaluate our numerical model's convergence behavior, compare it to the theory developed in the previous chapter, and examine the turbulent power spectrum generated by our simulations. The physics of the results of this large-scale simulation were discussed in the previous chapter. Consequently, in this chapter, we focus on the first two points.

4.2 . Simulation description

4.2.1 . Physical description

Our physical setup consists of a diabatic (double-diffusive) instability leading to a dynamo effect, as described in section 3.3.5 from the previous chapter. We consider a cubic domain $[0, 1]^3$ filled with plasma initially at rest and at hydrostatic equilibrium. We set a stabilizing potential temperature gradient $\nabla_T - \nabla_{ad} < 0$ and a destabilizing mean molecular weight gradient $-\nabla_\mu > 0$ that is not strong enough to trigger Ledoux convection i.e. $\nabla_T - \nabla_{ad} - \nabla_\mu < 0$. We also add a weak initial background magnetic field that serves as a seed for the dynamo process $B_0^2 \simeq 10^{-10}$. The various gradients involved are illustrated in Figure 2. We then trigger diabatic convection by adding source terms on the temperature and chemistry $-H_T > -R_X$. As depicted in Figure 2, convection can

be triggered in a Ledoux stable atmosphere with fast thermal diffusion and slow chemical diffusion. A partition of fluid at the top of the box that begins to descend will quickly reach thermal equilibrium with its new environment while maintaining its original mean molecular weight, becoming denser and accelerating its descent. The driving force behind this instability lies in the source terms. In this case, the criterion for this instability is expressed as $(\nabla_T - \nabla_{ad})R_X - \nabla_\mu H_T < 0$ and is satisfied by the fluid parameters. The initially weak magnetic field is amplified as the hydrodynamical instability saturates and generates 3D convection rolls. This results in an increase in magnetic energy. This dynamo-induced magnetic field evolution is the primary focus of our experiment. The theory we developed in Chapter 4 provides an estimate for the intensity of the horizontal components of the magnetic field once the instability reaches saturation (3.31) :

$$\frac{k^2 k_x^2}{k_x^2 + k_y^2} \frac{h_p}{\rho_0 g} B_{0,\text{dia}}^2 = \frac{1}{H_T + R_X} \left(\overbrace{(\nabla_T - \nabla_{ad})R_X - \nabla_\mu H_T}^{\text{Double-diffusive unstable}} + Q_A \underbrace{(\nabla_T - \nabla_{ad} - \nabla_\mu)}_{\text{Ledoux stable}} \right). \tag{4.1}$$

This tells us that the saturated magnetic energy will be linearly increasing with the saturated diabatic gradients, and linearly decreasing with the saturated adiabatic gradient. Consequently, we choose strong thermal and chemical source terms to maintain the hydrodynamical profiles in their initial state of instability (as described by (3.44)). A low magnetic source term is also selected to minimize the "Ledoux stable" term in (4.1). Specifically, we choose $H_T = -1.2$, $R_X = -0.6$, and $Q_A = -0.001$. The small value of the latter means that the numerical diffusion pilots its effective value : $Q_A^{\text{eff}} = Q_A + C\Delta x^2$ for a 2nd order method. One main interest of the Grand Challenge is the impact of the spatial mesh resolution on this dynamo process. As we increase the resolution, the numerical diffusion should decrease and the dynamo intensity should increase.

4.2.2 . Strategy and subsequent I/O needs

The goal of this simulation is to study variations in the intensity of the dynamo effect. In particular, we will need to access the time series of the magnetic and kinetic energies. As our simulation scales up to a 4096^3 resolution and given that each cell contains 9 variables, namely density ρ, pressure P, the three components of velocity u_{xyz}, the three components of the magnetic field B_{xyz}, and the mixing mass ratio X. These variables are stored as double-precision numbers. The data size for a single full save amounts to approximately 5 TB. Consequently, it is impossible to save hundreds of outputs and compute the mean energies a posteriori. This leads to our first I/O need : computing and storing the mean kinetic and magnetic energies in the whole domain at high frequency.

Given that the simulation starts in a linear regime with the growth of the instability, and considering that this instability requires an extended physical time to reach saturation, we opt for a way to bypass the linear phase. The reachable time by a 4096^3 simulation starting from the initial conditions would not be enough to observe the convergence of the dynamo effect. Moreover, we expect and will observe many node failures, interrupting the simulation. This leads us to our second I/O need : writing the full solution on disk as checkpoints and reading these checkpoints to restart and upscale simulations. Once the flow has reached statistical saturation, we double the resolution. We repeat this process 4 times until the desired final resolution of 4096^3 is attained. It is imperative to verify the convergence at each resolution before performing upscalings by examining the turbulent energy cascade. In practice, this is done via a Fast Fourier transform (FFT) on the kinetic energy of a horizontal slice taken at the middle of the domain. This leads to our third I/O need : Storing slices of the solution at high frequencies. Our last I/O need is the extraction of vertical profiles of quantities of interest (such as potential temperature, mean molecular weight, shear and magnetic energy), that are central to the interpretation of convection simulations.

4.3 . The ARK^2-MHD code

The ARK^2-MHD code is open source and can be found at https://gitlab.erc-atmo.eu/remi.bourgeois/ark-2-mhd.git. A simple use case of the libraries used in ARK^2-MHD namely PDI, MPI (Message Passing Interface) and Kokkos for solving the linear heat equation can be found at https://github.com/rbourgeois33/heat-equation-hpc-tools

4.3.1 . Parallelism

The set of PDE we are using and the corresponding numerical scheme are described in the previous chapter; see equations (3.1) and appendix 3.B. Moreover, we employ a second order MUSCL-Hancock strategy, as suggested in Chapters 1 and 2. A fully explicit finite volume numerical scheme such as the one we employ consists of a stencil operation. Following the work of [Padioleau et al. 2019] we implement the finite volume update using Kokkos's abstractions for loops and reduction. The code is then organized with computation kernels, as follows :

1. the MHD flux kernel, wrapped in a `Kokkos::parallel_for`,
2. the source term operators, wrapped in a `Kokkos::parallel_for`,
3. the conservative to primitive conversions kernels, wrapped in a `Kokkos::parallel_for`,
4. the time-step computation kernel, wrapped in a `Kokkos::parallel_reduce` to find the min value of the time step over the domain.

Each computational kernel is programmed as a C++ functor. Note that in more recent codes, such as https://gitlab.maisondelasimulation.fr/lrousselhard/nova, the `KOKKOS_LAMBDA` abstraction (which are anonymous functors) has been preferred over functors. The computational domain is divided into multiple subdomains. Each subdomain is managed by a distinct MPI process. These MPI processes are responsible for evolving the solution in their respective subdomains and for communicating with adjacent processes through ghost cells. Within each process, parallelism is handled by Kokkos, enabling the use of OpenMP/Cuda/HIP. Additionally, all MPI processes write their subdomain concurrently in the output file, using the parallel HDF5 library through PDI.

4.3.2 . Performance analysis

In this section, we provide some performance analysis of our code, comparing its speed on several GPUs and a weak scaling analysis on Adastra. Note that all I/O are removed for these measurements.

Comparison with other GPUs

Comparing performances of our code on an AMD MI250x chip against a Nvidia GPU is a delicate task : each MI250X is a Multi-Chip Module (MCM) that contains 2 Graphics Compute Dies (GCDs), leading to 2 GCDs (2 MPI processes) per chip. We choose to compute and evaluate the performances (in terms of the number of cells updated per second) per chip by running simulations on full nodes and computing the corresponding performance of individual chips. We turn off all I/O during the performance tests, but we keep the 2nd-order accuracy, the boundary conditions and sources used in our convection setup. Table 4.1 details the performance results and MPI decomposition used to test each type of GPU. These results are also compiled in figure 4.1. We observe that our code's performance is roughly doubled between all succeeding generations of Nvidia GPUs. On the newest MI250x chip, we get a performance between the one obtained with the $V100$ and the $A100$ GPUs. Lastly, our performance on the Intel Max 1550 GPUs is slightly below the one we get on the $V100$.

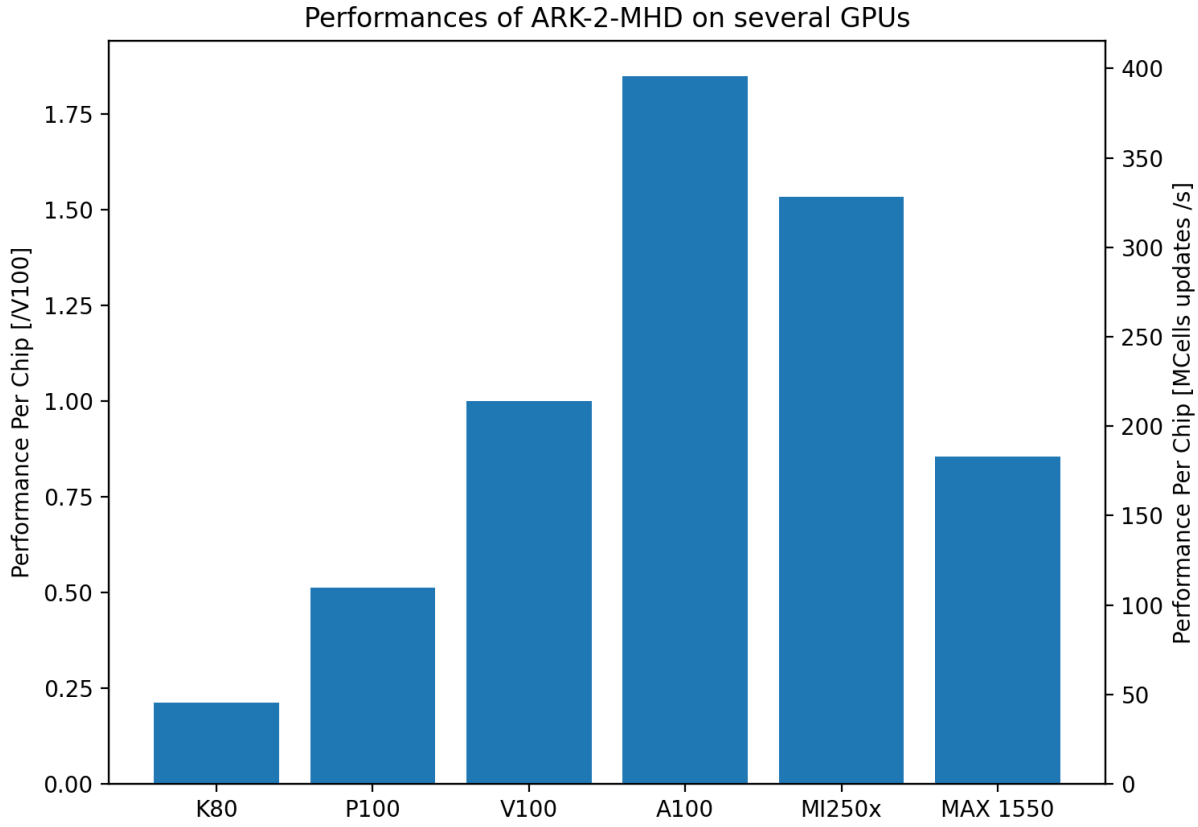


Figure 4.1 – Performance comparison with several GPUs : Nvidia's K80 (Mdlslx83), Nvidia's P100, V100 & A100 (Ruche), AMD's MI250x (Adastra) and Intel's MAX 1550.

Machine	Chip	Chips/Node	Node Perf.	Perf./chip	(m_x, m_y, m_z)	(n_x, n_y, n_z)
Mdlslx83	K80	8	363.491	45.4 (0.21)	(2,2,2)	(256,256,256)
Ruche	P100	2	219.433	109.7 (0.51)	(1,1,2)	(256,256,256)
Ruche	V100	4	856.21	214.0 (1.00)	(1,2,2)	(512,256,256)
Ruche	A100	4	1582.65	395.6 (1.84)	(1,2,2)	(512,256,256)
Adastra	MI250x	4	1312.8	328.2 (1.53)	(2,2,2)	(512,256,256)

Table 4.1 – Performance tests of ARK^2-MHD on single nodes of different machines. Performances are given in Mcell update/second. The tuple (m_x, m_y, m_z) describes the MPI cartesian domain decomposition decomposition. Each MPI subdomain has a resolution (n_x, n_y, n_z). All chips can run one process per chip, except the MI250x, which handles two.

Weak scaling

We conduct a weak scaling experiment, varying the number of nodes from 1 to 256, which corresponds to an increase from 8 to 2048 MPI processes or from 4 to 1024 GPUs. The results of this experiment are displayed in Figure 4.2. The data indicate near-ideal weak scaling performance.

4.3.3 . Scalable I/O through PDI

Our mesh resolution necessitates an I/O design where all the fully scalable reductions are executed directly on the GPUs during the simulation itself. The following section explains the various types of I/O produced by our simulation and elaborates on the specifics for each output or reduction. In particular, for each type of output, we will detail

- its description and why it is of interest,
- how it is computed (what is done in the simulation code versus what is done a-posteriori),
- the format of the output file and how it is exposed to the HDF5 library through a `PDI_MULTI_EXPOSE`.

Coupling ARK^2-MHD with PDI

Coupling PDI with a simulation code is straightforward and noninvasive. PDI needs to be initialized after MPI via a `PDI_init(PC_get(conf, ".pdi"))`. The `conf` variable corresponds to a paraconf tree that should be loaded right before the PDI initialization with `PC_tree_t conf = PC_parse_path("path_to_yaml_file")`. Note that PDI can be initialized before or after Kokkos. The Yaml file serves as a descriptive layer, specifying various attributes of the handled variables, such as their types and sizes. Data is transferred to the I/O library of choice through PDI thanks to instances of `PDI_MULTI_EXPOSE`. This setup allows for a streamlined interaction between the ARK^2-MHD simulation code and the I/O library (HDF5 in our case) through PDI.

Initialisation

The first instance of `PDI_MULTI_EXPOSE` performs the initialization. This involves setting up various discretization parameters and outlining the MPI cartesian decomposition to PDI. This step is crucial for ensuring that

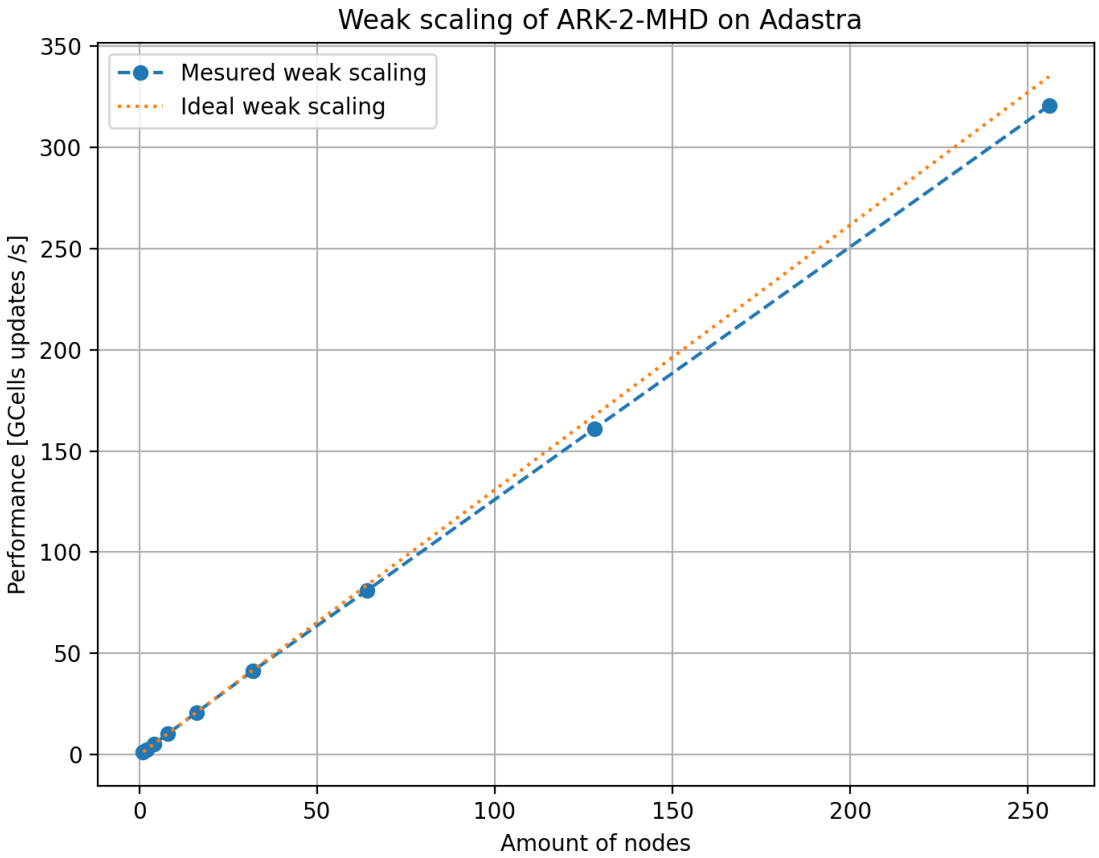


Figure 4.2 – Weak scaling on *Adastra*

PDI accurately locates each MPI subdomain in the whole domain, maintaining the consistency of different types of output. During the initialization, each process exposes :

- `pdi_init`, the event name,
- `mpi_coord`, (tuple) the coordinates of the current process in the MPI domain decomposition,
- `ncell`, (tuple) the number of cells in each direction of the whole simulation domain,
- `ncell_local`, (tuple) the number of cells in each direction of the current MPI subdomain,
- `ghost`, the width of the ghost cell zone,
- `nbvar`, the amount of variables (9 in our case),
- `prefix`, prefix for the output files,
- `restart_id`, name of the restart file to be read (only used in case of a restart).

Checkpoints

This `PDI_MULTI_EXPOSE` initiates the parallel writing of the entire solution into a single HDF5 file. Each MPI process exposes its local field, and because of the prior `mpi_coord` exposure during the initialization phase, PDI can locate these values in the output file. Each checkpoint file contains an array of dimension $N_x \times N_y \times N_z \times$`nbvar`. The exposed variables are :

- `checkpoint`, name of the event,
- `output_id`, output number,
- `time`, physical simulation time,
- `local_full_field`, solution contained in the current MPI subdomain,
- `filename`, name of the output file.

The next `PDI_MULTI_EXPOSE` initiates the writing of the XML file corresponding to the HDF5 checkpoint file. It is always called right after the checkpoint writing. The exposed variables are :

- `write_xml`, name of the event,
- `output_record`, xml file content,
- `output_id`, output number.

Global average of the magnetic & kinetic energies

As discussed in the introduction, two quantities of interest are the total kinetic and magnetic energies. They allow us to easily monitor the intensity of the dynamo effect by looking at their time series. First, we compute the energies locally in each cell of the domain. Then, we average the result over MPI subdomains. Note that this reduction is done directly on the simulation GPUs with `Kokkos::parallel_reduce`. Then, we perform a `MPI_AllReduce` to gather the results from each MPI subdomain into one domain average. The `PDI_MULTI_EXPOSE` exposes this result. The output file for the mean energy consists of a single file containing an array of size `mean_max_time_step`, `nranks`. During each expose, each MPI process writes the global mean value into the file at their corresponding MPI rank index and current time step. As a result, each MPI process writes the same

result separately. While this is sub-optimal from a storage standpoint, it avoids concurrent memory access. A possible means of improvement would be to have only one MPI process writing the result. The `PDI_MULTI_EXPOSE` exposes :

- `write_means`, name of the event,
- `rank`, MPI rank of the current process,
- `pdi_writer_mean_time_step`, output number for the mean values,
- `emag`, global average of the magnetic energy,
- `ekin`, global average of the kinetic energy.

Figure 3.12 illustrates the temporal evolution of the total kinetic and magnetic energies within the simulation domain.

Vertical profiles

Calculating vertical averages of convection-related quantities aids in interpreting the simulation. Each MPI process calculates the vertical profile of its subdomain and makes them available through PDI. Specifically, each process executes a vertical loop along the z-axis and a 2D horizontal `Kokkos::parallel_reduce` on the x and y axes for local averaging. These sub-profiles are gathered into whole domain profiles a posteriori. The `PDI_MULTI_EXPOSE` command exposes these vertical profiles. The output file contains a single array with dimensions `profile_max_time_step` $\times$ 7 $\times$ n_z $\times$ `nranks`. During each exposure, MPI processes write their average profiles to this array using their respective rank indexes and time steps. The number 7 in the dimensions corresponds to the amount of vertical quantities we are saving. The exposed variables are :

- `rank`, MPI rank of the current process,
- `pdi_writer_mean_time_step`, output number for mean,
- `vert_prof`, vertical profiles data,
- `output_id`, output number.

Figure 3.14 illustrates vertical profiles of potential temperature and magnetic energies.

Slices of the solution

We use slices of the solution to monitor the behavior of solutions at high mesh resolutions effectively. This allows us to observe unphysical oscillations and, through an FFT, to compute the kinetic energy power spectra, which helps verify the convergence of the flow before upscaling. These slices should be written during the simulation at a high frequencies, higher than the frequency of full I/O outputs. For this purpose, MPI processes associated with subdomains containing the domain's center expose their respective slices. The initial step in this process is to construct a multi-dimensional representation of the data on the GPU to facilitate slicing :

```
Kokkos::View<double***[9],Kokkos::LayoutLeft> qMD(m_q.data(),4+m_nx,4+
    m_ny,4+m_nz);
```

This operation creates `qMD`, a multi-dimensional representation of the contiguous GPU array `m_q` with dimensions `nbvar`$\times 4 + n_x \times 4 + n_y \times 4 + n_z$. The `LayoutLeft` specification ensures alignment with the layout of `m_q`. Then, a slice can be extracted using `Kokkos::subview` :

```
auto q_h_slice = Kokkos::subview(qMD, Kokkos::make_pair(2, m_nx+2),
    Kokkos::make_pair(2, m_ny+2), iz_middle+2, Kokkos::make_pair(0,9));
```

The `make_pair(a,b)` function serves a role analogous to the `a:b` syntax in Python, selecting a slice from the data. This action effectively extracts the middle slice from the multi-dimensional array. The $n_z + 2$ index corresponds to the middle slice. For compatibility with the PDI/HDF5 library, which requires a right-layout format, we declare a view with the `LayoutRight` configuration :

```
Kokkos::View<double**[9],Kokkos::LayoutRight> q_h_slice_gpu("
    q_h_slice_gpu", m_nx, m_ny);
```

The next steps involve copying the data from the original `LayoutLeft` format to the new `LayoutRight` structure, and then transferring this modified data to the host :

```
Kokkos::deep_copy(q_h_slice_gpu,q_h_slice);
Kokkos::deep_copy(m_q_h_slice_host,q_h_slice_gpu);
```

The first `deep_copy` transposes the data if necessary (if the default device layout is left). Finally, we expose the data to PDI with a `PDI_MULTI_EXPOSE` :

- `write_slice`, event name,
- `mpi_coord`, coordinate of the current MPI process in the cartesian domain decomposition,
- `pdi_writer_slice_time_step`, time step of slices exposition,
- `local_h_slice`, pointer to the slice exposed.

The slice files have dimensions `max_pdi_writer_time step`$\times N_x \times N_y \times 9$, corresponding to the slices of the 9 variables. Figures 3.15, 3.16, 3.17 were obtained using this process. Figure 3.13 presents the power spectra at several times during the Grand Challenge. These plots are obtained by performing a-posteriori FFTs on horizontal slices of the solution.

4.3.4 . *In-situ* analysis with Deisa

The Deisa library is not directly coupled with ARK, rather, it is executed separately on other computational ressources. The simulation code exposes the slice of interest to Deisa with a standard `PDI_MULTI_EXPOSE`. It is the same exposure as the previous one, with only the event name changing :

- `write_slice_Deisa`, as the event name,
- `mpi_coord`, the coordinate of the current MPI process in the cartesian domain decomposition,
- `pdi_writer_slice_time_step`, the time step of slices exposition,
- `local_h_slice`, a pointer to the exposed slice.

A separate, in situ process runs an instance of Deisa. Below is the Deisa Python script. First, we initialize Deisa by loading the Yaml file `Deisa_config.yml`, the mesh size in the z direction, the MPI decomposition, and retrieving the Deisa client and arrays :

```
Deisa = Deisa('scheduler.json', 'Deisa_config.yml')

with open('Deisa_config.yml') as file:
    data = YAML.load(file, Loader=YAML.FullLoader)
```

```
    nz = data["nz"]
    mz = data["mz"]
    prefix = data["prefix"]
    num_restart = data["num_restart"]

client = Deisa.get_client()
arrays = Deisa.get_Deisa_arrays()
arrays.check_contract()
```

Next, we select the slice, choosing the one corresponding to the third slice exposition (i.e., when `pdi_writer_slice_time_st` $= 3$). This choice is purely arbitrary.

```
slice = arrays["global_h_slice_Deisa"][3,:,:,:]
```

Next, we create the computation graph. The following lines correspond to the FFT calculations. They are descriptive and do not trigger any calculation at this point in the code. Deisa will wait for the data to be available through PDI to perform the computation.

```
ekin_Deisa=0.5*slice[:,:,id]*(slice[:,:,iu]*slice[:,:,iu]
+slice[:,:,iv]*slice[:,:,iv]+slice[:,:,iw]*slice[:,:,iw])/(mz*nz*mz*nz)
ekin_Deisa_rechunked = ekin_Deisa.rechunk({0: -1, 1: -1})
npix = ekin_Deisa_rechunked.shape[0]
fourier_image = da.fft.fftn(ekin_Deisa_rechunked)
fourier_amplitudes = da.absolute(fourier_image)**2
kfreq = da.fft.fftfreq(npix) * npix
kfreq2D = da.meshgrid(kfreq, kfreq)
knrm = da.sqrt(kfreq2D[0]**2 + kfreq2D[1]**2)
knrm = knrm.flatten()
fourier_amplitudes = fourier_amplitudes.flatten()
kbins = da.arange(0.5, npix//2+1, 1.)
kvals = 0.5 * (kbins[1:] + kbins[:-1])
```

Then, we submit the work to the scheduler and perform the computations :

```
s1,s2,s3,s4= client.persist([knrm, fourier_amplitudes, kbins,
   kvals])

arrays.validate_contract()

client.compute(s2).result()
client.compute(s1).result()
client.compute(s3).result()
client.compute(s4).result()
```

Lastly, we write the result to an HDF5 output file and terminate Deisa :

```
hf =
    h5py.File('fft_from_Deisa_'+prefix+"_"+str(num_restart)+'.h5',
    'w')
hf.create_dataset('knrm', data=knrm)
hf.create_dataset('fourier_amplitudes', data=fourier_amplitudes)
hf.create_dataset('kbins', data=kbins)
hf.create_dataset('kvals', data=kvals)
hf.close()
print("Done", flush=True)
client.shutdown()
```

Running this Deisa script during the simulation and computing the Fourier transform a posteriori allows us to validate the in situ pipeline we implemented. Figure 4.3 shows the kinetic power spectrum computed a-posteriori from a stored slice and in situ, using Deisa at the maximal 4096^3 resolution. We can see that they both coincide, showcasing the good functioning of Deisa in this extreme scale context.

4.4 . Grand Challenge proceedings

In this section, we describe the various strategies we employed to overcome the inherent technical difficulties that come with a Grand Challenge. As the machine was still in the testing phase, we expected and encountered many issues, such as node failures and competition for resources from other Grand Challenge projects. Moreover, as we are using Deisa and changing the resolution as well as the computational resources allocation, writing the bash scripts for job submission quickly becomes a tedious task. We chose to automate this process, as described below.

4.4.1 . Submission of simulation and in situ analysis jobs

We automated the creation of the bash job script using a Python script. This script requires several inputs : `num_restart` (the index of the restart), `mx, my, mz` (the MPI subdomain discretization in each direction), `Nx, Ny, Nz` (the cell count in each direction across the entire domain), and `tEnd` (the final time for this restart). From these inputs, the script calculates several values required by the slurm submission system : `nx, ny, nz` = `Nx/mx, Ny/my, Nz/mz` (the number of cells in each MPI subdomain), `num_proc` = `mx*my*mz` (the total number of simulation MPI processes), and `num_nodes` = `num_proc/8 + 3` (the total node count). Indeed, we share the simulation processes across the 8-process MI250x nodes and accounts for the three nodes required by the Deisa in situ analysis : one for the Dask scheduler, one for the Deisa client and one for the Deisa worker. Once all necessary information is known, the bash submission script and simulation input file can be generated. This script then allocates the precise number of nodes needed for both the simulation and in situ analysis and initiates the jobs.

4.4.2 . Dealing with nodes failures and time restrictions : Checkpoint programming

Given that the maximum allowable job duration on *Adastra* is 24 hours, and considering the unpredictability of the time required to write checkpoints due to the instability of the Lustre file system at the time of the Grand Challenge, we implemented an automated checkpoint system activated after 20 hours of computation. Constructing a real-time-based checkpoint system is delicate due to potential discrepancies in the internal real-time values

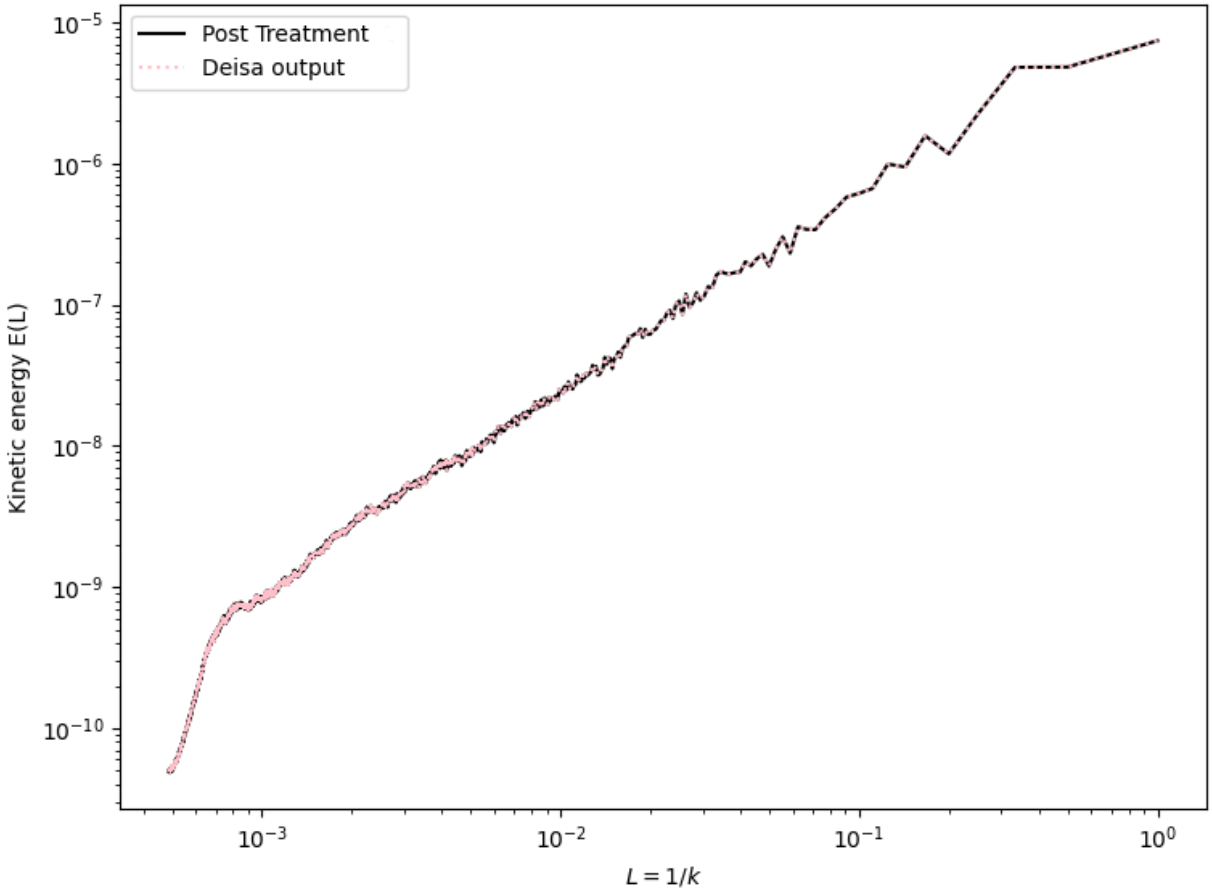


Figure 4.3 – Comparison of the kinetic power spectrum computed a-posteriori, and in situ, with Deisa at the maximal 4096^3 resolution

of the different MPI processes. Such discrepancies could lead to blocked states, where certain processes are in writing mode while others are in simulation mode. To avoid this, we employed the following strategy : Every 100 iterations, only one MPI process checks its internal clock to determine if the simulation has been running for more than 20 hours, subsequently communicating the result to the other processes using a `MPI_BCAST`. When the communicated value is true, all processes stop the simulation and initiate the writing of the checkpoint concurrently. Table 4.2 compiles the details of every restart that happened during the Grand Challenge.

Restart	Resolution	order	$\theta =$	t_0	t_{end}	Stop reason	Goal
0	256^3	1	Ma	0	2000	Finished	Linear phase
1	256^3	2	$max(Ma, 0.01)$	2000	2400	Finished	Order upscaling
2	512^3	"	"	2400	2440	Node Failure	Upscaling # 1
3	"	"	"	2440	2600	Finished	"
4	1024^3	"	"	2600	2620	Node Failure	Upscaling # 2
5	"	"	"	2620	2663	20h reached	"
6	"	"	"	2663	2700	Finished	"
7	2048^3	"	$max(Ma, 0.02)$	2700	2719	20h reached	Upscaling # 3
8	"	"	"	2719	2739	20h reached	"
9	"	"	"	2739	2740	Node Failure	"
10	"	"	"	2740	2759	20h reached	"
11	"	"	"	2759	2770	Node Failure	"
12	"	"	"	2770	2789	20h reached	"
13	"	"	"	2789	2800	Finished	"
14	4096^3	"	"	2800	2806.9	20h reached	Upscaling # 4
15	"	"	"	2806.9	2811	Node failure	"
16	"	"	"	2811	2818	20h reached	"
17	"	"	"	2818	2820	Node failure	"
18	"	"	"	2820	2821.5	Node failure	"
19	"	"	"	2821.5	2826	Node failure	"
20	"	"	"	2826	2830	Finished	"

Table 4.2 – Restart details

4.5 . Conclusion

In this chapter, we presented the physical setup and implementation details of our Grand Challenge simulation. Our discussion centered around the incorporation of the PDI library into the simulation code, emphasizing the scalable implementations of the various types of I/O. Furthermore, we highlighted the usage of the Deisa library, demonstrating its effective functioning through the computation of an in situ Dask-based Fast Fourier transform for power spectrum analysis. Future works include the development of a feedback mechanism from Deisa to the simulation code. This would involve implementing a convergence criterion for the power spectrum directly within Deisa. When the criterion is met, Deisa will trigger the simulation code to upscale the solution.

The present work paves the way towards such self-piloted large-scale convection simulation, and could also be applied to other types of numerical models.

Acknowledgements

We thank the Centre Informatique National de l'Enseignement Supérieur for the 210k GPU hours allocation used for this Grand Challenge project. I thank all the participants of this Grand Challenge, namely Pascal Tremblin, Thomas Padioleau, Samuel Kokh, Yushan Wang, Amal Gueroudji, Julien Bigot, Sainsbury Felix and Martial Mancip. We also thank Intel for letting us access their MAX 1550 GPUs.

5 - Mimicking the GP-MOOD method with neural networks in 2D. Early experiments.

5.1 . Introduction

In compressible flows, non-linear shock waves, contact discontinuities, and smooth regions may coexist. This variety creates the need for so-called *shock-capturing methods* (see [LeVeque et Leveque 1992]) that include a mechanism to identify and stably discretize jumps, minimizing numerical oscillations while maximizing accuracy on the smooth parts of the flow. The present work focuses on the Euler Equations on a 2D cartesian grid, however, the tools we develop could be extended to other hyperbolic systems and discretizations. Researchers in finite volume methods have developed various strategies to optimize accuracy in smooth flows while maintaining stability in strong gradients. These so-called "high-order shock-capturing methods" balance reducing numerical dissipation for accuracy and increasing it for stability. These techniques can be classified into two categories : a priori and a posteriori methods, based on their approach to managing smooth versus non-smooth flows. The a priori approach is older and the most widely adopted. It involves discretizing gradients using non-linear limiting procedures that provably ensure stability. Examples are : second-order piecewise linear TVD (Total Variation Diminishing) methods [Van Leer 1974 ; Tadmor 1988 ; Hubbard 1999 ; Harten 1997] higher-order polynomial techniques like the piecewise parabolic method (PPM,[Colella et Woodward 1984 ; McCorquodale et Colella 2011]), essentially non-oscillatory (ENO) methods (e.g., [Harten 1997 ; Shu 1998]), and weighted ENO (WENO) methods (e.g., [Liu et al. 1994 ; Jiang et Shu 1996 ; Balsara et Shu 2000 ; Gerolymos et al. 2009]). These shock-capturing methods use non-linear mechanisms or switches to detect local flow gradient magnitudes before updating the solution, ensuring stability on discontinuities and accuracy in smooth regions. A priori methods are computationally expensive due to the cell-by-cell calculations of non-linear limiters, and they introduce unavoidable numerical dissipation, reducing solution accuracy. Figure 5.1 displays the logical pipeline in conventional a priori shock-capturing FV methods.

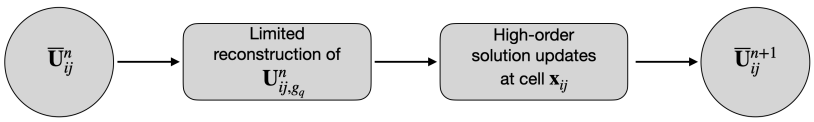


Figure 5.1 – The logical flow line of the solution updating procedure. Shown is the principle flow line in conventional a priori high-order methods where limited spatial data reconstructions are applied to *all* cells, regardless of local flow smoothness.

Introduced in [Clain et al. 2011], MOOD offers an alternative a posteriori detection principle to traditional a priori techniques. Initially, MOOD focused on high-order (up to third-order) two-dimensional polynomial approximations on unstructured grids, reducing polynomial order until each cell's solution meets the Discrete Maximum Principle (DMP). If the state within a cell does not meet the DMP criteria, it is recalculated, defining MOOD as an a posteriori scheme. The MOOD method's accuracy was enhanced to sixth-order in [Diot et al. 2012] and [Diot et al. 2013]. In [Bourgeois et Lee 2022], a set of new Gaussian Process (GP) reconstructions is introduced to the

MOOD paradigm, resulting in the GP-MOOD method. GP offers a cost-effective alternative to polynomial methods due to their smaller stencil and inherent accuracy variability. The paper also proposes a relaxation of the existing MOOD admissibility criteria with the so-called "Compressibility-Shock-Detection" (CSD) criterion. This modification improves the treatment of weakly compressible flows.

Despite their various numerical benefits, MOOD methods also need to be improved. For example, they suffer from a poor parallelization efficiency and incompatibility with implicit time discretizations. To address these challenges, the preliminary work of [Bourriaud et al. 2020] proposes to use a small Neural Network to learn the MOOD heuristic in a 1D context. This NN-based approach replaces the traditional a posteriori MOOD detection strategy with an a priori educated guess for selecting the appropriate polynomial accuracy order. The promising outcomes of this study inspire the present chapter. We aim to expand it to two-dimensional flows and combine it with the GP-MOOD method. Other works [Ray et Hesthaven 2018, 2019; Discacciati et al. 2020; Yu et Hesthaven 2022] have also proposed using NNs as a limiting procedure for high-order finite volume methods. However, their training approach differs as it is based on learning representations of canonical smooth and discontinuous functions rather than from simulation data.

To address the diversity of two-dimensional flows using cheap (small) NNs, we propose employing online learning instead of the pre-trained black box method referenced above. Our approach, though not an actual online learning framework alternating between simulation and training phases, is a simplified version that will be referred to as NN-GP-MOOD. This serves as a preliminary model for future research. The NN-GP-MOOD method is divided into three stages for each simulation : 1) dataset generation phase : simulate the first 10% of the problem and create the training dataset using GP-MOOD. 2) training Phase : train the neural network. 3) evaluation phase : complete the simulation using NN-GP-MOOD. This approach has shown success in several test cases. However, two limitations can be identified : in highly dynamic scenarios, such as the Mach 800 astrophysical jet test case, the NN is not able to provide stable results. Moreover, the training phase is still too long to be competitive with the base GP-MOOD method.

This chapter is organized as follows : first, we re-introduce the High-order GP finite volume formulation and the MOOD paradigm. Then, we detail how we integrated small NNs into the MOOD loop to make the method a posteriori again. We present our online learning procedure and provide numerical results that show the benefits and limits of our approach.

5.2 . High order Gaussian-processes finite volume formulation

5.2.1 . Governing equations and finite volume method

We are interested in solving a hyperbolic system of conservative laws in 2D,

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) + \partial_y \boldsymbol{G}(\boldsymbol{U}) = 0, \tag{5.1}$$

where $\boldsymbol{U}$ is the vector of conservative variables and $\boldsymbol{\mathcal{F}} = (\boldsymbol{F}, \boldsymbol{G})$ are the flux functions in x- and y-direction. For the Euler equations in 2D, the conservative variables and the flux functions are defined as,

$$\boldsymbol{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \quad \boldsymbol{F}(\boldsymbol{U}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u\left(\rho E + p\right) \end{bmatrix}, \quad \boldsymbol{G}(\boldsymbol{U}) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v\left(\rho E + p\right) \end{bmatrix}, \tag{5.2}$$

where ρ denotes the fluid density, u and v represent the x and y fluid velocity respectively, and ρE is the total energy. The system is closed with an ideal gas EOS, $p = (\gamma - 1)\left(\rho E - \frac{1}{2}\rho(u^2 + v^2)\right)$, where γ is the ratio of specific heat. The hyperbolic system in Eq. (5.2) is physically admissible if both $p > 0$ and $\rho > 0$, and a numerical method that maintains the positivity property is referred to as a positivity-preserving method. The basic form of the finite volume discretization of 5.1 is derived by integrating the equation over each cell $I_{ij} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$ of a uniform cartesian grid, and over a time interval $[t^n, t^{n+1}]$, yielding

$$\overline{\boldsymbol{U}}_{ij}^{n+1} = \overline{\boldsymbol{U}}_{ij}^{n} - \Delta t\, \mathbb{F}_{\nabla}, \tag{5.3}$$

where $\overline{\boldsymbol{U}}_{ij}^{n}$ is a vector of the volume-averaged conservative variables and $\mathbb{F}_{\nabla}$ is a collection of discretized spatial derivatives terms, including the face-averaged and temporally averaged flux in each spatial direction. Our choice for the discrete temporal update in 5.3 for high-order simulations is to use a multi-stage SSP-RK method [Gottlieb et Shu 1998; Spiteri et Ruuth 2002]. It leaves us to determine how to evaluate $\mathbb{F}_{\nabla}$ to meet the expected high-order accuracy. To this end, we recall a family of multidimensional FV reconstruction algorithms of GP introduced in [Lee et al. 2017; Diot et al. 2012; Bourgeois et Lee 2022].

5.2.2 . Achieving high-order discretization with GP

Gaussian quadrature rule

Following [Bourgeois et Lee 2022; May et Lee 2024], we use a q-point Gaussian quadrature rule to approximate the *face-averaged* fluxes at $2q$-th order accuracy using q many *pointwise* fluxes on each cell face. This gives us to write $\mathbb{F}_{\nabla}$ as

$$\mathbb{F}_{\nabla} = \frac{1}{\Delta x}\sum_{j_g=1}^{q} \omega_{j_g}\left(\boldsymbol{F}^*_{i+1/2,j_g} - \boldsymbol{F}^*_{i-1/2,j_g}\right) + \frac{1}{\Delta y}\sum_{i_g=1}^{q} \omega_{i_g}\left(\boldsymbol{G}^*_{i_g,j+1/2} - \boldsymbol{G}^*_{i_g,j-1/2}\right), \tag{5.4}$$

where i_g and j_g are the indices of the q-point Gaussian quadrature point locations on each x and y cell face; the corresponding ω_{i_g} and ω_{j_g} are the quadrature weights for the $2q$-th order numerical integration. The numerical fluxes $\boldsymbol{F}^*$ and $\boldsymbol{G}^*$ are *pointwise* fluxes at each respective cell face, obtained by solving the corresponding Riemann problems at the Gaussian quadrature points. A pair of high-order accurate *pointwise* Riemann states, $(\boldsymbol{U}_L, \boldsymbol{U}_R)$, are used as inputs to calculate the corresponding Riemann problems at each quadrature point. In each pair, the left $\boldsymbol{U}_L$ and the right $\boldsymbol{U}_R$ states are computed using a $(2R+1)$th-order GP reconstruction method described below.

GP reconstruction

[Bourgeois et Lee 2022] provides a family of $(2R+1)$th-order GP reconstruction methods. For our preliminary study of the NN approach, we limit ourselves to the 3rd order version of GP. The 3rd order GP reconstruction operates on :

(i) Input : a vector $\mathbf{q}$ consisting of the *volume-averaged* conservative variables (e.g., $\overline{\rho}_{ij}^{n}$) on a 2D local GP stencil of radius 1.

(ii) Output : an *unlimited* 3rd-order accurate conservative *pointwise* Riemann state of the same input variable (e.g., density) at each Gaussian quadrature point (e.g., $\rho_* = \rho(\mathbf{x}_*)$, where $\mathbf{x}_* = (x_{i\pm1/2}, y_{j_g})$ or $\mathbf{x}_* = (x_{i_g}, y_{j\pm1/2})$)

The input and output of the GP reconstruction process should not be confused with the input/output of the neural networks we will introduce later. The GP stencil of radius $R = 1$ defines a five-point cross-shape stencil. We consider a local labeling of the cell x_{ij} and its neighbor represented in figure 5.2. These states form a one-dimensional array of states, denoted by $\overline{\mathbf{q}}_{ij}$ given by

$$\overline{\mathbf{q}}_{ij} = (\bar{q}_1, \bar{q}_2, \bar{q}_3, \bar{q}_4, \bar{q}_5)^T . \tag{5.5}$$

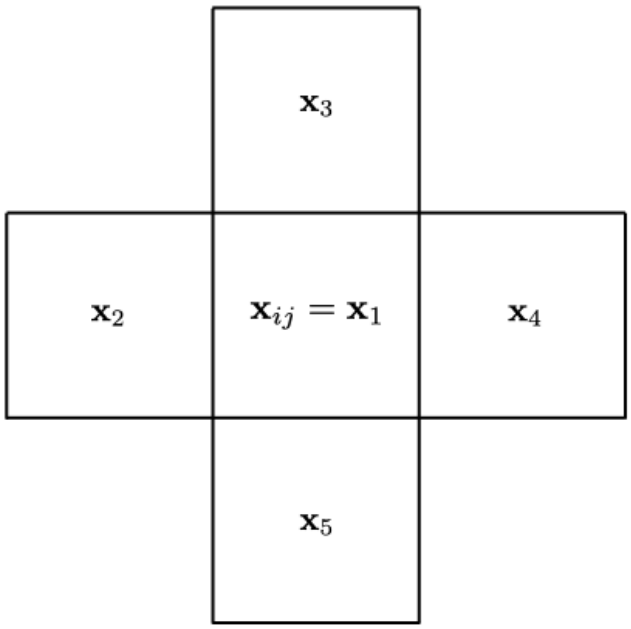


Figure 5.2 – The five-point GP stencil of radius $R = 1$ for the 3rd-order GP reconstruction method. The ordered labeling illustrates how the local volume-averaged conservative variables at $t = t^n$ are rearranged into a one-dimensional five-point array, $\overline{\mathbf{q}}_{ij}$.

Table 5.1 – Multi-point Gaussian quadrature rules (QR) are used in combination with GP. The quadrature points, g_m, are tabulated over the reference interval $[-0.5, 0.5]$ that maps to the unit length of each cell-face, e.g., $[y_{j-1/2}, y_{j+1/2}]$ at the x-normal cell-face. See also figure 5.3.

QR	g_1	ω_1	g_2	ω_2
2-point QR	$\frac{1}{2\sqrt{3}}$	$\frac{1}{2}$	$-\frac{1}{2\sqrt{3}}$	$\frac{1}{2}$

Extrapolate FV data with GP

An in-depth section about GP FV regression theory can be found in [Bourgeois et Lee 2022]. The GP reconstruction consists of a simple dot product between the volume-averaged data of the stencil and a prediction vector.

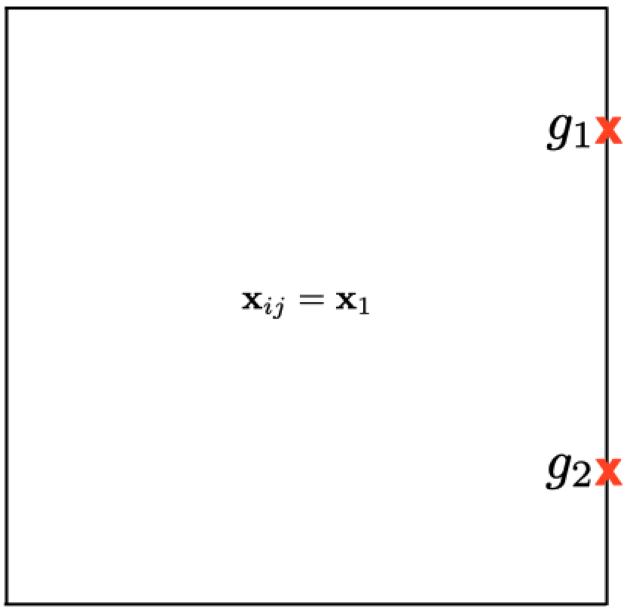


Figure 5.3 – 4th-order two-point Gaussian quadrature points, g_m, at the x-normal cell-face of the central cell $\mathbf{x}_{ij} = \mathbf{x}_1$ used with the 3rd-order GP-R1 method. The values of the quadrature points g_m and the corresponding weights ω_m are given in 5.1.

It provides a pointwise estimation $\tilde{m}$ at $\mathbf{x}_*$ that reads :

$$\tilde{m}_* = \tilde{m}(\mathbf{x}_*) = \mathbf{z}_*^T \overline{\mathbf{q}}_{ij}, \tag{5.6}$$

where $\mathbf{z}_*$ is called the prediction vector. The construction is not trivial and not covered here, we refer to [Bourgeois et Lee 2022] for details. As noted in [Reyes et al. 2018, 2019; Reeves et al. 2021], the prediction vector $\mathbf{z}_*$ is data-independent, only depending on the grid configuration. Therefore, in practice, $\mathbf{z}_*$ can (and should) be pre-computed before each simulation as soon as the grid geometry is defined. The value of $\mathbf{z}_*$ is saved and reused throughout the simulation, requiring only the $\mathcal{O}(N)$ dot-product calculation between $\mathbf{z}_*$ and the input vector, where N is the stencil size, during the simulation. In the case of a 2D regular Cartesian mesh using a 2-point Gaussian quadrature rule, we need 8 prediction vectors (2 per cell-face).

5.2.3 . Dependency domain

In this section, we want to clarify the difference between the *GP reconstruction stencil* and the *dependency domain of a cell*, for the GP-R1 method. The *GP reconstruction stencil* is the data the GP reconstructor needs to compute a pointwise value of the function as detailed in the section above. It is made of 5 cells, represented in Figure 5.2. On the other hand, the dependency domain of a cell when evolving it with the GP-R1 method is made of 13 cells and represented in figure 5.4. Indeed, to evolve the cell $\mathbf{x}_1$, the GP-R1 reconstructions have to be calculated on cells $\mathbf{x}_1, \underline{\mathbf{x}}_2, \mathbf{x}_3, \mathbf{x}_4$ and $\mathbf{x}_5$. Therefore, each of their GP-R1 reconstruction stencils is included in the dependency domain of the cell $\mathbf{x}_1$.

5.3 . MOOD method : A posteriori limiting strategy

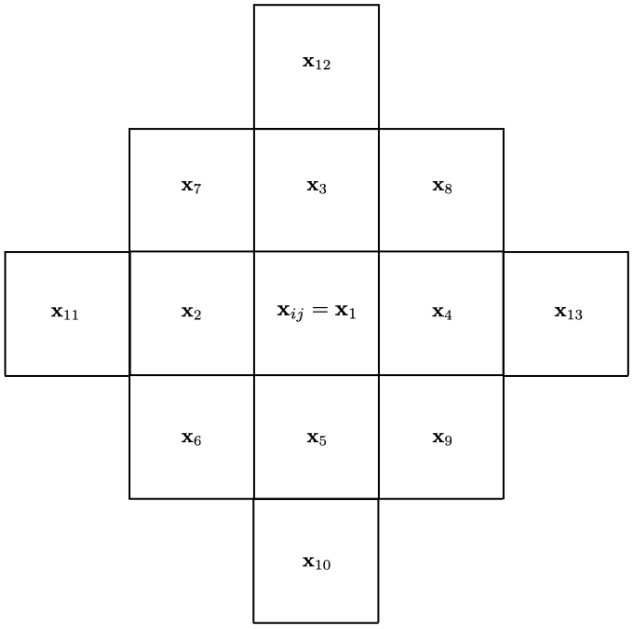


Figure 5.4 – The 13 points dependency domain of a cell when using the GP-R1 method.

5.3.1 . General idea

The GP reconstruction method can now be integrated into a a posteriori framework. The main idea in the MOOD method is the a posteriori limiting strategy [Clain et al. 2011; Diot et al. 2012; Diot et al. 2013; Diot 2012], which updates each cell with the most accurate solver available first, followed by the cell-by-cell inspection to see if a set of admissibility conditions are met locally. For example, let us suppose that the updated solution at $\mathbf{x}_{ij}$ after the first pass with the highest accurate solver fails to meet the admissibility constraints. In that case, the cell is reset to its original value and re-updated with a more stable, less accurate method. This is repeated until the constraints are verified everywhere. In the worst case, a local solution could end up with the most diffusive – but most stable – solver, e.g., the first-order method, in the regions where shocks and discontinuities are present. The MOOD method, by design, is endowed with the positivity-preserving property of the first-order method near sharp flow gradients while utilizing high-order solutions away from these troubled cells. This work aims to train a neural network to learn the posteriori behavior of the GP-MOOD and reproduce it a priori.

5.3.2 . The GP-MOOD method

In this section, we recall the three main building blocks of the MOOD algorithms.

(i) *The detection criteria :* the first component is a sequence of prescribed properties that the discrete numerical solution has to fulfill to be considered acceptable. These conditions are of two types : "Physical Admissibility Detection" (PAD) and "Numerical Admissible Detection" (NAD). In most fluid dynamics simulations, PAD ensures that the numerical solution represents an admissible flow state (e.g., positivity in pressure and density). More generally, they ensure that the numerical solution is acceptable with regard to the physical model. They only depend on the set of solved PDEs.

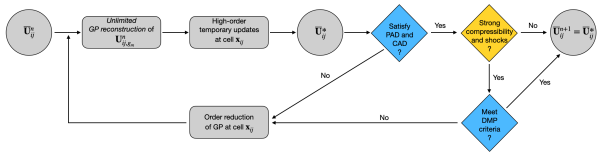
On the other hand, NAD ensures that the solution produced by the solver limits oscillations. It is based on a relaxed discrete maximum principle (DMP) [Diot et al. 2012; Bourgeois et Lee 2022]. It also includes

the detection of non-numeric values such as `NAN`'s and `Inf`'s, that is, the admissibility of the state from a computer science point of view (Computer Science Admissibility Detection or CAD). CAD is identical for all sets of PDEs. Suppose a candidate solution does not satisfy either of the PAD and NAD criteria in some cells. In that case, such cells are recorded as *troubled cells* and their discrete updates are repeated with a lower order method (see (iii) below).

(ii) *The safe scheme :* the second component is the choice of a numerical method used as the last resort when all the other high-order schemes have failed to produce an acceptable solution according to the detection criteria in (i). To this end, the first-order Godunov (FOG) scheme is the most popular, while the second-order MUSCL method could be used as well to improve the results on contact discontinuity (e.g., see [Padioleau 2020]). In this study, we use FOG as the safe scheme.

(iii) *The scheme cascade :* a family of reconstruction schemes is the third component that provides a sequential series of different reconstruction methods, from the most accurate available method to the safe scheme. The conventional MOOD method uses a set of unlimited *polynomial* reconstruction methods in different orders up to the 6th-order accuracy [Diot et al. 2012; Diot et al. 2013]. Alternatively, for the present study, we only use one GP reconstruction method of 3rd order, GP-R1 for which the sequence is GP-R1 $\longrightarrow$ FOG.

The logical loop pipeline of the GP-MOOD method is summarized in Figure 5.5. A concise description of the



(a)

Figure 5.5 – The flow chart of the GP-MOOD method.

MOOD algorithm is provided in [Bourgeois et Lee 2022].

5.3.3 . Limitation of a posteriori methods

In this section, we recall the drawbacks of the a posteriori limitation algorithm mentioned in [Bourriaud et al. 2020]. They include

1. the limited parallelization efficiency due to the required different treatment of troubled cells,
2. the incompatibility of the MOOD framework with an implicit time discretization due to the explicit nature of the a posteriori process,
3. the DMP CSD and PAD criteria being artificial heuristics, not based on solid mathematical foundations.

Replacing the detection loop with a NN would make the method a priori again, enabling good parallelization and compatibility with an implicit time discretization.

5.4 . A NN for optimal order detection in 2D

In this section, we briefly re-introduce the multi-layer perceptron and an example of its use for the design of shock-capturing methods. Then, we introduce the architecture of the MLP we use here for the local choice between the 1st and 3rd order method in our simulations.

5.4.1 . Multi Layer Perceptron (MLP)

A Multilayer Perceptron (MLP) is a feedforward NN composed of at least three layers of nodes, with an input layer, one or more hidden layers, and an output layer. Each node in a given layer is connected to every node of the next layer through a system of weighted connections. At each layer but the input, each node applies a non-linear activation function (ReLU [Nair et Hinton 2010] in our case) to its input to compute its output values (see [Goodfellow et al. 2016; Després 2022]). While MLPs can model complex relationships between inputs and outputs, they are prone to overfitting. Overfitting can be mitigated by several techniques, including early stopping, regularization, and dropout [Srivastava et al. 2014]. In our case, we are learning from data coming from a deterministic simulation algorithm. The absence of noise in the training data makes our case especially prone to overfitting. The backpropagation algorithm uses gradient descent (or a variant of it, in our case, the Adam optimizer [Kingma et Ba 2014]) to minimize the value of an error function [Rumelhart 1986].

5.4.2 . NN for shock-capturing methods

Using neural networks as limiting procedures for highly accurate fluid simulations has been the subject of recent research. Our chapter aims to extend the approach introduced in [Bourriaud et al. 2020] in two dimensions of space, for our GP-MOOD method. [Bourriaud et al. 2020] trains a local NN to learn the MOOD method behavior in 1D on the Euler equations and $M1$ model for radiative transfer. The model can learn from different Riemann problems and generalize its knowledge to new, unseen RPs. NNs have also been used as a troubled cell detector in the framework of Discontinuous Galerkin methods. In [Ray et Hesthaven 2018], the author trains a NN to detect troubled elements and adapt the numerical method locally. In [Ray et Hesthaven 2019], they extend their approach to two dimensions of space and use a greatly reduced amount of NN parameters, making the approach computationally competitive. In [Discacciati et al. 2020; Yu et Hesthaven 2022], they use a similar approach to pilot numerical viscosity in DG methods.

A fundamental difference between [Discacciati et al. 2020; Yu et Hesthaven 2022; Ray et Hesthaven 2018, 2019] versus [Bourriaud et al. 2020] and the present study is the nature of the training dataset. The latter group of papers uses training data from simulations. They attempt to learn the discretization of the specific problem considered (set of PDE + type of MOOD method). The latter trains their NNs on canonical functions, agnostically of the PDE. The goal is to learn to separate smooth functions from discontinuous functions on a given mesh and then use that knowledge to choose the best-adapted numerical methods locally.

5.4.3 . Architecture of the neural network

1. **Input :** As our NN is trying to learn the behavior generated by the $R = 1$, 3rd order GP-MOOD method, its input must be the considered cell's dependency domain during a time step. Since the dependency domain contains 13 cells with 4 variables each, this amount to 52 input neurons :

$$\mathbf{q}^{NN} = \begin{bmatrix} \boldsymbol{U}_1 \\ \boldsymbol{U}_2 \\ \dots \\ \boldsymbol{U}_{12} \\ \boldsymbol{U}_{13} \end{bmatrix} \in \mathbb{R}^{52} \tag{5.7}$$

2. **Output :** As our NN is supposed to choose between the 1st-order method and the 3rd-order GP method, the output is a vector of probabilities $r = (p_1, p_3)^T$ that satisfies :

$$p_1 + p_3 = 1, \quad p_1, p_3 > 0 \tag{5.8}$$

from which we can make local choices on discretization accuracy.

3. **Hidden layers :** We chose to use $L = 2$ hidden layers of 5 neurons each.
4. **Number of parameters :** Our NN therefore has $52 \times 5 + 5 \times 5 + 5 \times 2 = 295$ parameters.
5. **Activation functions :** We use the classical ReLu activation function [Nair et Hinton 2010] for all layers except the last one on which we use the softmax function to ensure (5.8).

5.4.4 . Integrating the NN in the simulation loop

This section describes how we practically choose the reconstruction accuracy in a given cell $\mathbf{x}_{ij} = \mathbf{x}_1$ once the NN has provided us the probability distribution (p_1, p_3). In particular, we describe how to compute the high-precision states at points g_1, g_2 in figure 5.3. Let us note $\tilde{\boldsymbol{m}}_{g_i}$ the 3rd order accurate GP reconstruction state at the gaussian point g_i, $i \in \{1, 2\}$, following the procedure 5.6. Let us consider the volume-averaged data $\overline{\boldsymbol{U}}^n$ contained in the considered cell. Finally, Let us note $\mathbf{q}_{g_i}^{NN}(\tilde{\boldsymbol{m}}_{g_i}, \overline{\boldsymbol{U}}^n, p_3 = 1 - p_1)$ the high precision reconstructed state we use in the NN-GP-MOOD method. A first simple choice would be :

$$\mathbf{q}_{g_i}^{NN}(\tilde{\boldsymbol{m}}_{g_i}, \overline{\boldsymbol{U}}^n, p_3) = \begin{cases} \tilde{\boldsymbol{m}}_{g_i}, & \text{if } p_3 \geq 0.5 \\ \overline{\boldsymbol{U}}^n, & \text{otherwise.} \end{cases} \tag{5.9}$$

Moreover, we want to encode uncertainty in the decision-making process; the NN might not provide a strong prediction but an output such as $(p_1 = 0.45, p_3 = 0.55)$. Therefore, we use a convex combination from the probability distribution when reconstructing the high-order GP pointwise values. It ensures the admissibility of the used high-precision state, given that both the central cell and the reconstructed GP states are admissible. We define $\mathbf{q}_{g_i}^{NN}$ by

$$\mathbf{q}_{g_i}^{NN}(\tilde{\boldsymbol{m}}_{g_i}, \overline{\boldsymbol{U}}^n, p_3) = \begin{cases} p_1 \overline{\boldsymbol{U}}^n + p_3 \tilde{\boldsymbol{m}}_{g_i}, & \text{if } p_3 \geq 0.5 \\ \overline{\boldsymbol{U}}^n, & \text{otherwise,} \end{cases} \tag{5.10}$$

instead of (5.9). Note that if $p_3 < 0.5$, we discard the high order GP state and use the first order method. We then proceed with the finite volume update once the high-order states are computed a priori. As mentioned in [Bourriaud et al. 2020], a a posteriori check is still required.

5.4.5 . Training procedure

We use Pytorch [Paszke et al. 2019] as our Machine learning framework and its implementation of the Adam Optimizer [Kingma et Ba 2014]. We optimize for the Cross entropy loss function [Good 1952]. Our choice differs from [Bourriaud et al. 2020] that uses the mean square error loss. It is motivated by the classification nature of the problem solved by the NN. We start with an initial learning rate of $lr = 0.01$ that is reduced whenever the training error stalls for 5 epochs. We stop the training after 300 epochs and use a batch size of 1024. We use the dropout method to avoid overfitting [Srivastava et al. 2014]. We chose a dropout rate of $p = 0.1$.

5.4.6 . Offline learning and online learning

Most of the literature on NNs as limiters for CFD uses a pre-trained NN as a black-box [Ray et Hesthaven 2018, 2019; Discacciati et al. 2020; Yu et Hesthaven 2022; Bourriaud et al. 2020]. This chapter proposes a proof of concept for an opposite approach where the NN is trained online as the simulation happens. To this end, we suggest a, very simplified version of online learning where the training is achieved on data from the first few time steps, and the NN is then used to complete the simulation. By construction, it cannot work well for cases where the nature of the flow changes dramatically during the simulation. For instance, time-dependent source terms or boundary conditions would require an actual online learning algorithm to be correctly simulated. A more sophisticated approach could be developed as we still use an a posteriori check by triggering training phases whenever that check fails. The advantage of such a method over a black box approach is that the employed NN is super-specialized. Therefore, it can be smaller than a black box model. However, training the model during the simulations is costly and requires complicated machinery to be implemented. Also, we must ensure that our training process and NN performances are as low in stochasticity as possible and as reliable as possible. We empathize that the approach we propose and test is not genuine online learning. We want to deliver a proof of concept for online learning by mimicking the MOOD method with NNs in 2D.

5.4.7 . Dataset constitution

During the dataset generation phase, at the end of each Runge-Kutta stage, we browse the solution array and the orders obtained with the GP-MOOD method. Since troubled cells are typically much rarer than non-troubled cells, we must be careful when generating the dataset to make sure it is not massively unbalanced. To this end, while browsing the array, we proceed as follows.

1. If the cell is troubled, we append the corresponding input-output training pair $(\boldsymbol{q}^{NN}, (p_1 = 1, p_3 = 0))$ to the training dataset.
2. If the cell is not troubled, we compute the current proportion of troubled cells currently in the dataset $p = N_1/(N_1 + N_3)$ where N_1 is the number of 1st order input-output datapoints, and N_3 is the number of 3rd order input-output datapoints. Then, we generate a random number $r \in [0, 1]$. If $r < p$, we add the pair $(\boldsymbol{q}^{NN}, (p_1 = 0, p_3 = 1))$ to the training dataset. Otherwise, we discard it.

We discard all constant inputs $q_1 = q_2 = ... = q_{13}$ (for which we automatically use the 3rd order method in the simulation code). We do not perform any data normalization as the NN is specialized for each simulation.

5.5 . Results

5.5.1 . Experimental protocol : a proof of concept for online learning

Unless specified otherwise, for each test case, we use the following procedure that corresponds to a very simplified version of online learning.

1. Dataset generation phase : simulate the first 10% steps of the problem and generate a training dataset, using GP-MOOD and the flux-splitting method from Chapter 1 with added transversal diffusion.
2. Training phase : train 10 different NNs.
3. Evaluation phase : complete the simulation with the ten different NN separately using NN-GP-MOOD.

Adding transverse diffusion means that we add a diffusion term on the non-normal component of the velocity. For example, assuming that we are computing fluxes in the x direction, we add $-\frac{a}{2}(v_R - v_L)$ to the ρv flux. This allows consistent results with a more isotropic diffusion, stabilizing the local instabilities stemming from the NN mis-predicting orders. We train and evaluate ten different NNs for each test case because training is stochastic, and different instances may give different results. We want to show when our results are consistent, and where they are not producing different outputs from one another. Indeed, our proposed online learning approach will use only one NN in practice. We want to ensure it is consistently performing well. For the relevant test cases, we will also perform an ablation analysis e.g., remove some key improvements of the training process one by one to assess their importance. The procedure we just described will be referred to as *"base"* while the ablated one will be referred to as :

1. **no dropout :** not using dropout, making the NN prone to over-fitting,
2. **no TD :** not using the transversal diffusion in the flux-splitting Riemann solver.

We also tried using the MSE (Mean Square Error) loss instead of the CEL (Cross Entropy Loss) loss without significant change in the quality of the solutions. For each problem, we will also compare the CPU time performance of GP-MOOD and NN-GP-MOOD (after training) by giving :

1. the total resolution time,
2. the time spent predicting the reconstruction order. It is 0 for the GP-MOOD method as it is a purely a posteriori method. For the NN-GP-MOOD method, it corresponds to the time spent evaluating the NN on the FV data to obtain the reconstruction orders,
3. time spent computing the first try. It is the time spent computing the solution with the original order maps (either 3 everywhere for the GP-MOOD method / 3 or 1 depending on the prediction of the NN for the NN-GP-MOOD method),
4. time spent correcting. It corresponds to the time spent recomputing the solution at a lower order. For the GP-MOOD method, it will never be 0 if the domain contains any shock/discontinuities. A well-trained NN should ensure that this time reduces to 0 with the NN-GP-MOOD method.

The NN-GP-MOOD approach aims to reduce the computation time of step as much as possible. Note that we do not provide the training times. They are typically much longer than the simulation times. Therefore, the approach is far from being competitive with standard methods in its current state.

5.5.2 . Reproducing the results

To reproduce the numerical results presented in this chapter, please refer to the Readme file of our GitHub repository https://github.com/slug-cfd/gp-mood.git

5.5.3 . 2D Riemann problem, configuration 3

The first test case is the 2D Riemann problem already introduced in section 1.8.5. We set the resolution at 256^2 and the CFL number at 0.8. Reference solutions obtained with the 1st order and the 3rd GP-MOOD method are shown in Figure 5.6. The GP-MOOD method allows a much sharper capture of the discontinuities and shocks than the first-order method. When assessing the quality of the solution obtained with the NN, we want to be mindful of

1. the symmetry of the output along the $y = x$ axis,
2. the amount of diffusion at the shocks and discontinuities,

3. the existence of the diagonal jet,
4. the minimal and maximal values reached by the density.

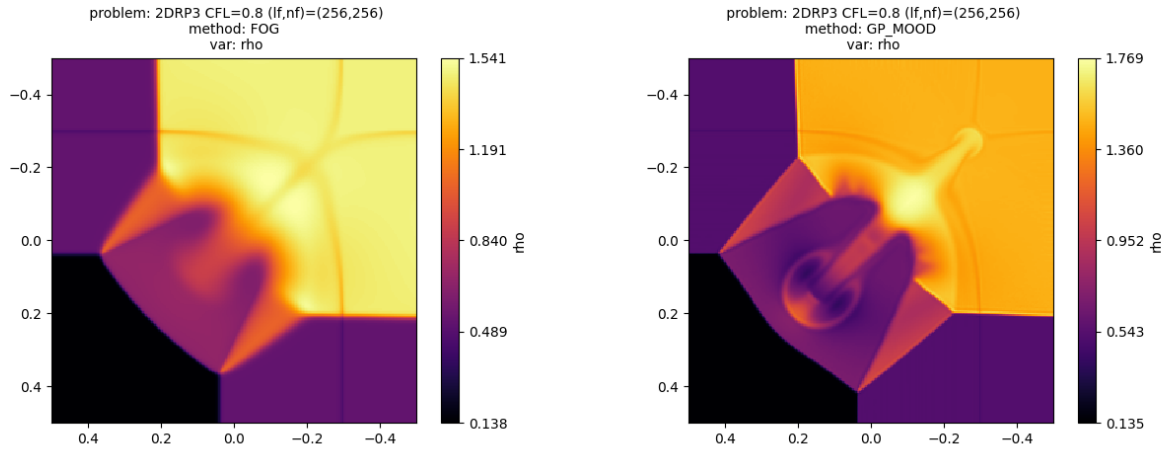


Figure 5.6 – Solutions of the 2DRP3 problem at $t = 0.8$s obtained with the 1st (Left) and 3rd order (Right) GP-MOOD methods.

Training dataset informations

We run the preliminary simulation phase for 90 steps ($t \simeq 0.075$). This generates a dataset of $424,559$ training units (input/output pairs). Removing all the doubled entries reduces that number to $122,635$. We train 10 different NNs and use them to complete the simulation, starting from the 91st step. Figure 5.7 shows the solution at the end of the dataset generation phase. Comparing it with Figure 5.6, we can see that the dataset includes the main shocks of the problem and the very beginning of the formation of the central flow structure.

Qualitative aspect of the results

Figure 5.8 shows ten different final density maps corresponding to ten simulations done with ten independently trained NNs. On all outputs, we can see that the symmetry is well preserved, the shocks and central features are captured sharply, the diagonal jet is present (but less sharply resolved than in the GP-MOOD solution in Figure 5.6), and the minimal and maximal values of the density are within the bound established by the GP-MOOD resolution 5.6. The NNs behave consistently and essentially all provide the same result, which is sharp, like the 3rd-order GP-MOOD method on the shocks and central feature but displays more numerical diffusion around the diagonal jet.

Amount of a posteriori corrections needed and performances

No non-admissible states were generated by the NN-GP-MOOD method on the 2DRP3 problems. Therefore, no a posteriori corrections were required. Figure 5.9 compares the number of detected cells by the GP-MOOD method and a priori by the NN-GP-MOOD method on the 2DRP3 problem, averaged over the ten different runs.

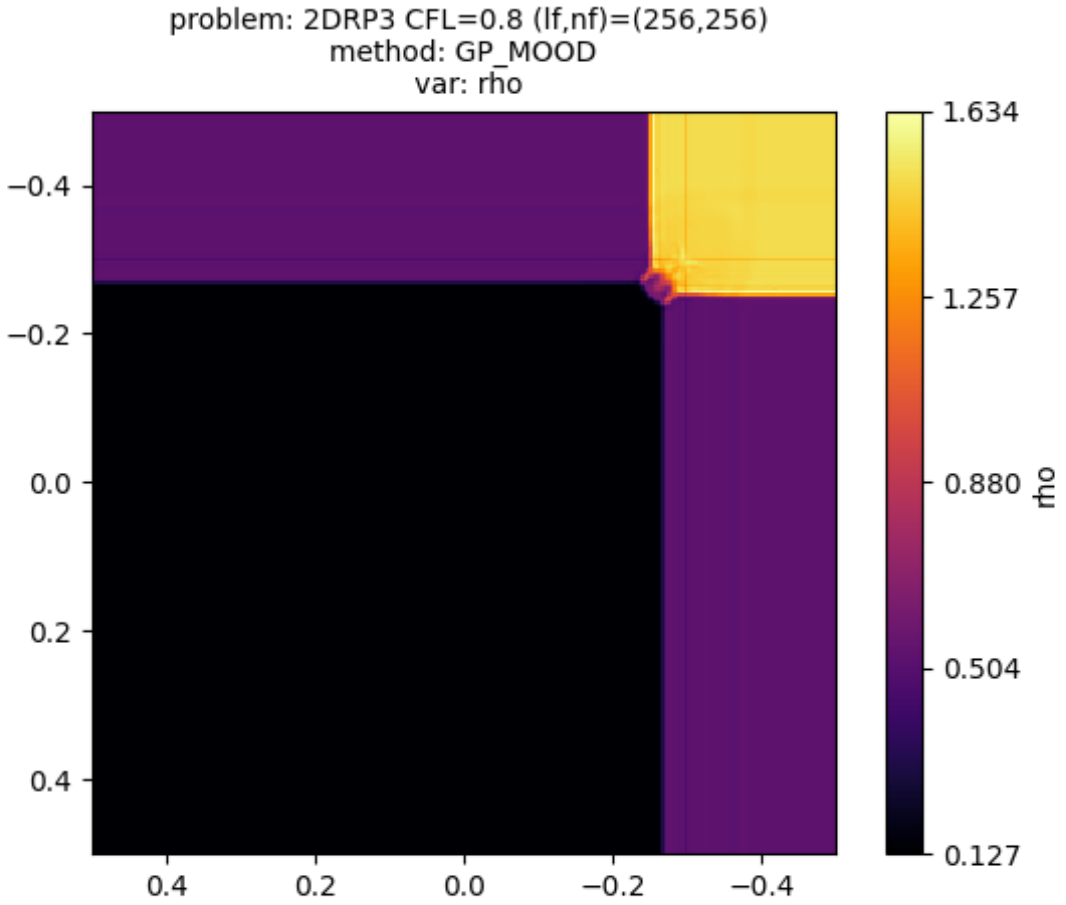


Figure 5.7 – Aspect of the solution to the 2DRP3 problem at the end of the dataset generation phase, at $t = 0.075s$

It is clear that the NN-GP-MOOD is more conservative and uses the first-order method more than the GP-MOOD method. Thankfully, as we saw in the previous section, it still provides highly accurate results. To better understand the difference in the number of detected cells, we can look at the order maps obtained with GP-MOOD versus NN-GP-MOOD in Figure 5.10. It shows clearly that the low-order region surrounding the shocks and discontinuities are thicker than with the GP-MOOD method. Table 5.2 gives information regarding the performance of GP-MOOD and NN-GP-MOOD on the 2DRP3 problem. The NN allows completely eliminating the a posteriori computations. However, evaluating the NN is still more expensive than the a posteriori computation, leading to an overall 30% better performance for the GP-MOOD method.

Ablation study

When not using dropout, the diagonal jet and the discontinuity between the central feature and the top right quadrant are less well resolved (see Figure 5.11). The max values in several resolutions are above the upper limit of 1.769. Moreover, 2 out of 10 simulations generate non-admissible states requiring the a posteriori correction loop. When not using transverse diffusion, 9 out of 10 simulations generate nonadmissible states requiring the a

	GP-MOOD	NN-GP-MOOD
Total time	68.9s (1.0)	90.5s (1.3)
Time spent predicting	NA.	36.4s
Time spent 1st try	57.8s	52.4s
Time spent correcting	9.4s	0s

Table 5.2 – Performance details on 2DRP3. Experiments are run on 1 CPU core.

posteriori correction loop. Unphysical oscillations are observed (see Figure 5.11).

5.5.4 . Sedov Blast wave

Next, we revisit the Sedov blast test [Sedov 1993] that consists in the propagation of a circular blast wave. We only explore a resolution of 256^2 and a CFL number of 0.8. The final time is $t_{end} = 0.05$. Reference solutions obtained with the 1st order and the 3rd GP-MOOD method are shown in Figure 5.12. The GP-MOOD method allows a sharper capture of the shock propagation than the first-order method. When assessing the quality of the solution obtained with the neural networks, we focus specifically on

1. the circular symmetry of the solution,
2. the amount of diffusion e.g. the thickness and magnitude of the shock (3.510 for the first order method, 4.672 for the third order method),
3. the minimal and maximal values reached by the density,
4. the absence of grid-aligned instabilities.

Training dataset informations

We run the preliminary simulation phase for 140 steps. This generates a dataset of $122,029$ training units (input/output pairs). We train 10 different NNs and use them to complete the simulation, starting from the 141st step. Figure 5.13 shows the solution at the end of the dataset generation phase. Comparing figure 5.13 with figure 5.12, we can see that the dataset includes the very beginning of the shock propagation.

Qualitative aspect of the results

Figure 5.14 shows ten different final density maps corresponding to ten simulations performed with ten independently trained NNs. On all outputs, we can see that the symmetry is consistently well preserved. The shock front is thin on all simulations, but the maximum density value varies in the $[4.226, 4.736]$ range, showing that the NN is prone to slightly underestimating/overestimating the reconstruction order. No grid-aligned instabilities are observed.

Amount of a posteriori corrections needed and performances

Figure 5.15 compares the number of detected cells a posteriori by the GP-MOOD method and a priori by the NN-GP-MOOD method on the Sedov problem averaged over ten different runs. The NN-GP-MOOD is more conservative and uses the first-order method more than the GP-MOOD method, but the difference is less important than

for the 2DRP3 problem. Table 5.3 gives information regarding the performance of GP-MOOD and NN-GP-MOOD on the Sedov problem. The NN allows eliminating the a posteriori computations. However, evaluating the NN is still more expensive, leading to an overall 16% better performance for the GP-MOOD method.

	GP-MOOD	NN-GP-MOOD
Total time	54.0s (1.0)	62.8s (1.16)
Time spent predicting	NA.	12.2s
Time spent 1st try	49.0s	48.9s
Time spent correcting	3.5s	0s

Table 5.3 – Performance details on the Sedov test. Experiments are run on 1 CPU core.

Ablation study

1. **No dropout :** the shock front is over-diffused and non-symmetric in several simulations, see Figure 5.16.
2. **No TD :** the peak value of density is systematically much higher than with the GP-MOOD method, reaching values around 5. Some small grid-aligned instabilities are observed. See Figure 5.16.

5.5.5 . 2D Riemann problem, configuration 15

We now consider the configuration 15 of the 2D Riemann problems presented in [Liska et Wendroff 2003]. We only consider a resolution of 256^2 and a CFL number of 0.8. Reference solutions obtained with the 1st order and the 3rd GP-MOOD method are shown in Figure 5.17. The GP-MOOD method allows a sharper capture of the discontinuities and the central feature than the first-order method. When assessing the quality of the solution obtained with the neural networks, we focus here on

1. the amount of diffusion at the discontinuities,
2. the existence and sharpness of the central feature,
3. the minimal and maximal values reached by the density $\rho \in [0.4612, 1.046]$.

Training dataset informations

We run the preliminary simulation phase for 22 steps. This generates a dataset of $36,343$ training units (input/output pairs). Removing all the doubled entries reduces that number to $6,145$. We train 10 different NNs and use them to complete the simulation, starting from the 23rd step. Figure 5.18 shows the solution at the end of the dataset generation phase. Comparing it with 5.17, we can see that the dataset includes the main discontinuities of the problem and the very beginning of the formation of the central flow structure.

Qualitative aspect of the results

Figure 5.19 shows ten different final density maps corresponding to ten simulations done with ten independently trained NNs. On all outputs, we can see that the discontinuities and central features are captured sharply, and the minimal and maximal values of the density are within the bound established by the GP-MOOD resolution

from Figure5.17. The NNs behave consistently and essentially all provide the same result, which is sharp like the 3rd-order GP-MOOD method on the shocks and central feature but displays more numerical diffusion around the diagonal jet.

Amount of a posteriori corrections needed and performances

No non-admissible states were generated by the NN-GP-MOOD method on the 2DRP15 problems. Therefore, no a posteriori corrections were required. Figure 5.20 compares the number of detected cells by the GP-MOOD method and a priori by the NN-GP-MOOD method on the 2DRP15 problem, averaged over ten different runs. It is clear that the NN-GP-MOOD is more conservative and uses the first-order method more than the GP-MOOD method. Thankfully, as we saw in the previous section, it still provides highly accurate results. Table 5.4 gives information regarding the performance of GP-MOOD and NN-GP-MOOD on the 2DRP15 problem. For the 2DRP15

	GP-MOOD	NN-GP-MOOD
Total time	10.65s (1.0)	15.72s (1.5)
Time spent predicting	NA.	6.161s
Time spent 1st try	9.42s	9.23s
Time spent correcting	0.89s	0s

Table 5.4 – Performance details on 2DRP15. Experiments are run on 1 CPU core.

problem, The NN allows eliminating the a posteriori computations. However, evaluating the NN is still more expensive than the a posteriori computation, leading to an overall 50% better performance for the GP-MOOD method. This is especially true because of the small amount of troubled cells for this problem. All ablated experiments show similar good results.

5.5.6 . 2D Riemann problem, configuration 4

We now consider the configuration 15 of the 2D Riemann problems presented in [Liska et Wendroff 2003]. We only explore a resolution of 256^2 and a CFL number of 0.8. Reference solutions obtained with the 1st order and the 3rd GP-MOOD method are shown in Figure 5.21. The GP-MOOD method allows a sharper capture of the discontinuities and shock propagation than the first-order method. When assessing the quality of the solution obtained with the neural networks, we want to be mindful of the following :

1. the symmetry of the output along the $y = x$ axis,
2. the amount of diffusion at the shocks and discontinuities,
3. the minimal and maximal values reached by the density.

Training dataset informations

We run the preliminary simulation phase for 40 steps. This generates a dataset of $122,398$ training units (input/output pairs). Removing all the doubled entries reduces that number to $25,754$. We train 10 different NNs and use them to complete the simulation, starting from the 41st step. Figure 5.22 shows the solution at the end of the dataset generation phase. Comparing it with 5.21, we can see that the dataset includes the main shocks of the problem and the very beginning of the formation of the central flow structure.

Qualitative aspect of the results

Figure 5.23 shows ten different final density maps corresponding to ten simulations done with ten independently trained NNs. On all outputs, we can see that the symmetry is well preserved, and the central features are captured sharply. On 4 of the 10 outputs, the top-left/bottom-right shocks present excessive diffusion.

Amount of a posteriori corrections needed and performances

The NN-GP-MOOD method generated non-admissible states on the 2DRP4 problems (at most 0.005% of the cells), requiring a posteriori corrections. Figure 5.24 compares the number of detected cells a posteriori by the GP-MOOD method (black), a priori by the NN-GP-MOOD method (blue), and a priori + a posteriori by the NN-GP-MOOD (red) method, averaged over ten different runs. The difference between the red and blue lines represents the negative density/pressure cells generated by the NN-GP-MOOD method that had to be corrected a posteriori. Table 5.5 gives information regarding the performance of GP-MOOD and NN-GP-MOOD on the 2DRP4 problem. The NN allows to almost eliminates the need for a posteriori computations. However, evaluating the NN is still more expensive, leading to an overall 50% better performance for the GP-MOOD method.

	GP-MOOD	NN-GP-MOOD
Total time	30.4s (1.0)	45.44s (1.5)
Time spent predicting	NA.	17.73s
Time spent 1st try	26.69s	26.47s
Time spent correcting	2.94s	0.37s

Table 5.5 – Performance details on 2DRP4. Experiments are run on 1 CPU core.

Ablation study

Without transverse diffusion, 6 out of 10 simulations present excessive diffusion at the bottom-right/top-left shocks. The solutions look correct, nevertheless. Not using dropout does not impact the quality of the result.

5.5.7 . Mach 800 astrophysical jet

This last section focuses on the Mach 800 astrophysical jet problem. It is a variant of the Mach 100 jet presented in [Balsara 2012] and [Bourgeois et Lee 2022] with a faster jet injection. Our approach performs poorly on this test case and shows its limits. Indeed, as described in [Bourgeois et Lee 2022], the flow evolves quickly as many instabilities and wave/shock interactions occur during the resolution. These physical events are not included in the first 10% of the resolution that makes up the training dataset. Moreover, the velocity and density gradients are reaching extreme values. As a result, our approach has proved less reliable, as detailed below. Reference solutions obtained with the 1st order and the 3rd GP-MOOD method are shown in Figure 5.25. The GP-MOOD method allows a much sharper capture of shock propagation and secondary instabilities than the first-order method. When assessing the quality of the solution obtained with the NNs, we aim to consider the following carefully

1. the symmetry of the output along the $x = 0$ axis,
2. the amount of diffusion at the shocks front and in the central features,

3. the minimal and maximal values reached by the density.

Training dataset informations

We run the preliminary simulation phase for 150 steps. This generates a dataset of $97,115$ training units (input/output pairs). Removing all the doubled entries reduces that number to $96,934$. We train 10 different NNs and use them to complete the simulation, starting from the 151-th step. Figure 5.26 shows the solution at the end of the dataset generation phase. Comparing it with 5.25, we can see that the dataset includes the very beginning of the jet injection in the domain.

Qualitative aspect of the results

Six out of the ten NNs provide a very diffused, unsatisfactory result. We only provide one of them in Figure 5.27 as they are very similar to each other. In these cases, the Neural network classifies all cells in the shock envelope as troubled and provides a solution as diffused as the first-order resolution. The NN is completely missing the physics of the problem. We show the other 4 solutions in Figure 5.28. Each of them develops a different shape of the envelope. The axial symmetry is broken in different ways for each solution. They also all show central features that resemble the ones obtained with the GP-MOOD solution. Lastly, the maximal values of the density are higher than with the GP-MOOD method, suggesting that the NN does not limit the sharpness of the solution enough.

Amount of a posteriori corrections needed and performances

This analysis only considers the 4 NNs that provided an acceptable result. The NN-GP-MOOD method generates non-admissible states on the Mach 800 problems, requiring a posteriori corrections. Figure 5.29 compares the number of detected cells a posteriori by the GP-MOOD method (black), a priori by the NN-GP-MOOD method (blue), and a priori + a posteriori by the NN-GP-MOOD (red) method on the Mach 800 jet problem, averaged over 4 different runs. The difference between the red and blue lines represents the negative density/pressure cells generated by the NN-GP-MOOD method that had to be corrected a posteriori. Table 5.6 gives information regarding the performance of GP-MOOD and NN-GP-MOOD on the Mach 800 problem. The NN does not eliminate the need for a posteriori computations. However, evaluating the NN is still more expensive, leading to an overall 18% better performance for the GP-MOOD method. This is less than for previous problems as the amount of detected-corrected cells in the GP-MOOD method is important in this problem (typically 10%).

	GP-MOOD	NN-GP-MOOD
Total time	101.2s (1.0)	118.4s (1.18)
Time spent predicting	NA.	22.7s
Time spent 1st try	87.6s	88.7s
Time spent correcting	11.4s	4.85s

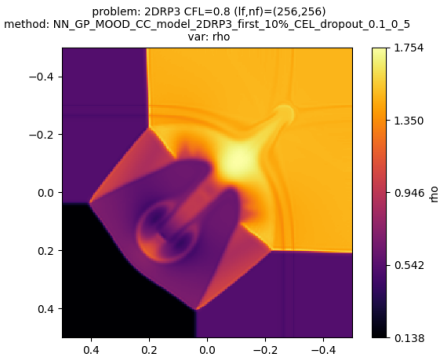
Table 5.6 – Performance details on the Mach800 jet. Experiments are run on 1 CPU core.

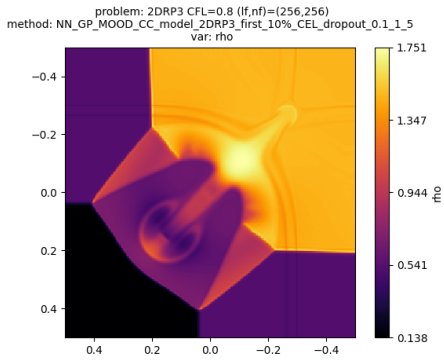
5.6 . Conclusion

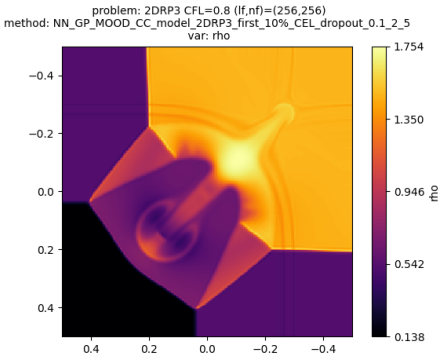
In this chapter, we presented a proof of concept for using online learning to mimic the a posteriori limitation of the GP-MOOD method. We re-introduced the GP-MOOD method and recalled the issues of a posteriori limiting that could be mitigated with an NN-based, a priori limitation. We presented the architecture of the small NNs we employ, how they are integrated into the simulation code, and the training procedure. We proposed a simplistic version of online learning where the NN is trained on data from the first 10% of the simulation considered and then used to complete it. We obtained encouraging results on several 2D Riemann problems and the Sedov Blastwave. Limitations of this oversimplified setup were highlighted on the Mach 800 jet test case. Moreover, the training duration of our NN is still too important to be competitive with standard high-order shock-capturing FV algorithm. This work will serve as a baseline from which we can improve to develop more competitive online and/or offline learning approaches. For example, we plan to use a black-box approach in our GP context. This would suppress the training time issue. On the other hand, we plan on developing a true, automated online learning algorithm to closely follow the simulation and switch between learning and simulation phases.

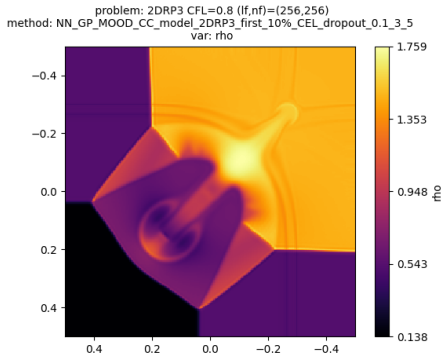
Acknowledgment

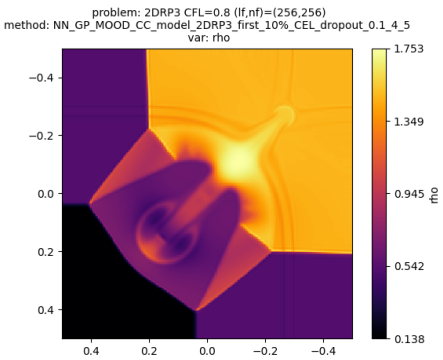
I am very thankful to Professor Dongwook Lee for funding my 3 months visit as a research scholar at the University of California, Santa Cruz, during which this work was conducted. I also thank Chris Degrendele for his precious help with this work.

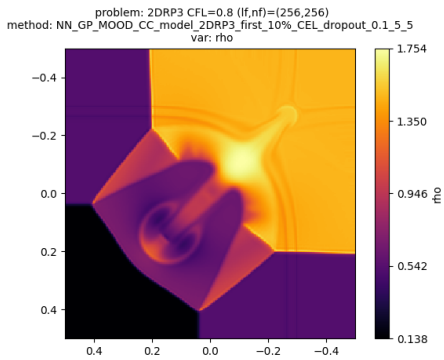

problem: 2DRP3 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP3_first_10%_CEL_dropout_0.1_0_5
var: rho
1.754
1.350
0.946
0.542
0.138
rho

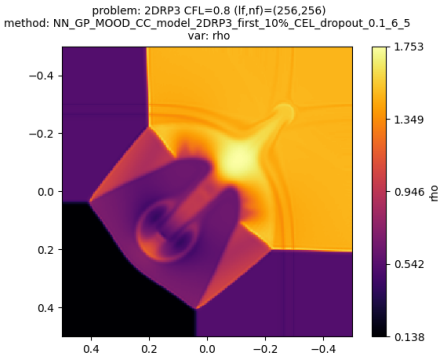

problem: 2DRP3 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP3_first_10%_CEL_dropout_0.1_1_5
var: rho
1.751
1.347
0.944
0.541
0.138
rho

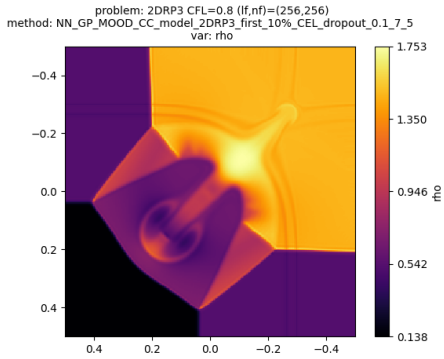

problem: 2DRP3 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP3_first_10%_CEL_dropout_0.1_2_5
var: rho
1.754
1.350
0.946
0.542
0.138
rho

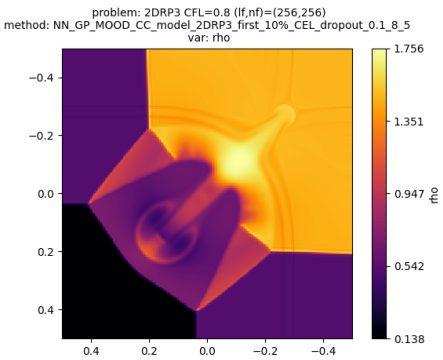

problem: 2DRP3 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP3_first_10%_CEL_dropout_0.1_3_5
var: rho
1.759
1.353
0.948
0.543
0.138
rho

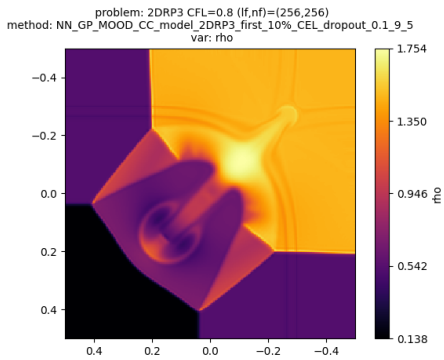

problem: 2DRP3 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP3_first_10%_CEL_dropout_0.1_4_5
var: rho
1.753
1.349
0.945
0.542
0.138
rho


problem: 2DRP3 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP3_first_10%_CEL_dropout_0.1_5_5
var: rho
1.754
1.350
0.946
0.542
0.138
rho


problem: 2DRP3 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP3_first_10%_CEL_dropout_0.1_6_5
var: rho
1.753
1.349
0.946
0.542
0.138
rho


problem: 2DRP3 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP3_first_10%_CEL_dropout_0.1_7_5
var: rho
1.753
1.350
0.946
0.542
0.138
rho


problem: 2DRP3 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP3_first_10%_CEL_dropout_0.1_8_5
var: rho
1.756
1.351
0.947
0.542
0.138
rho


problem: 2DRP3 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP3_first_10%_CEL_dropout_0.1_9_5
var: rho
1.754
1.350
0.946
0.542
0.138
rho

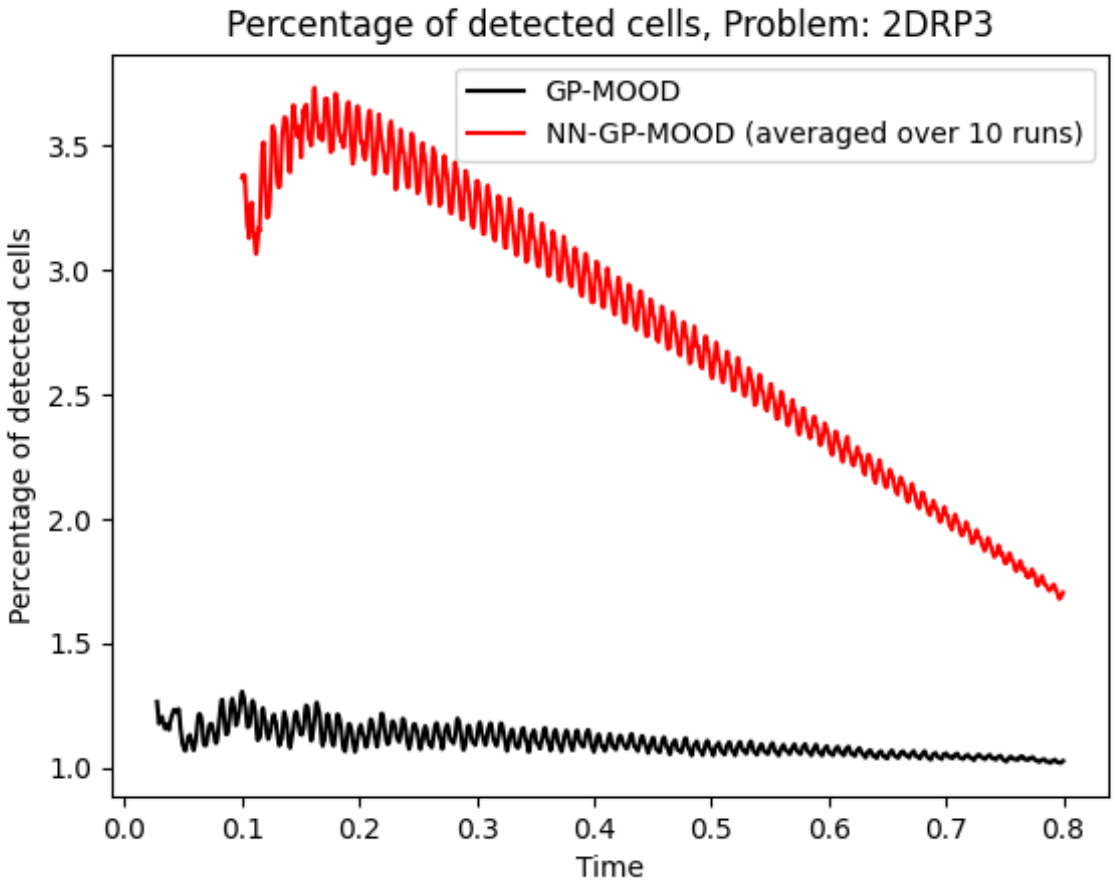


Figure 5.9 – Comparison of the number of detected cells a posteriori by the GP-MOOD method and a priori by the NN-GP-MOOD method on the 2DRP3 problem, averaged over 10 different runs.

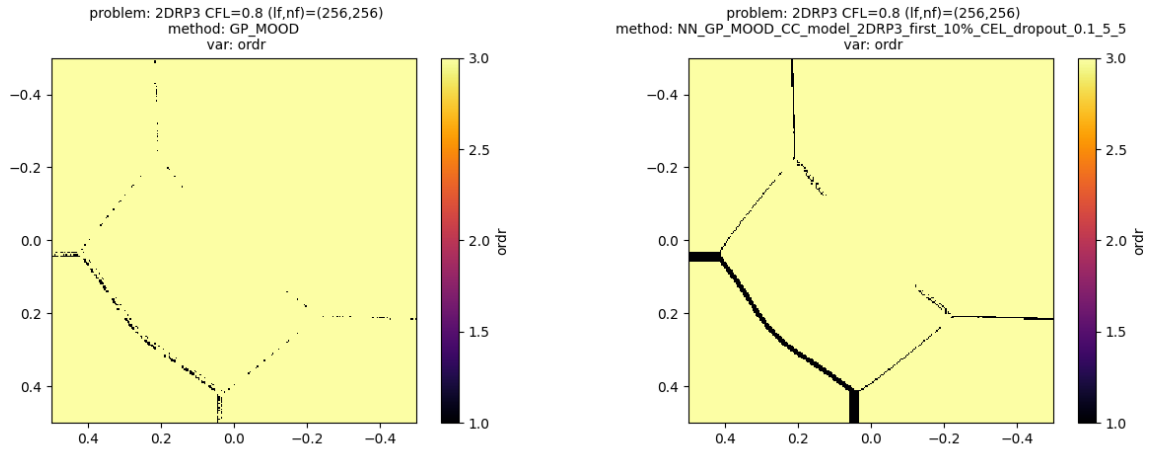


Figure 5.10 – Order maps obtained on the final step of the 2DRP3 problem with GP-MOOD and NN-GP-MOOD method

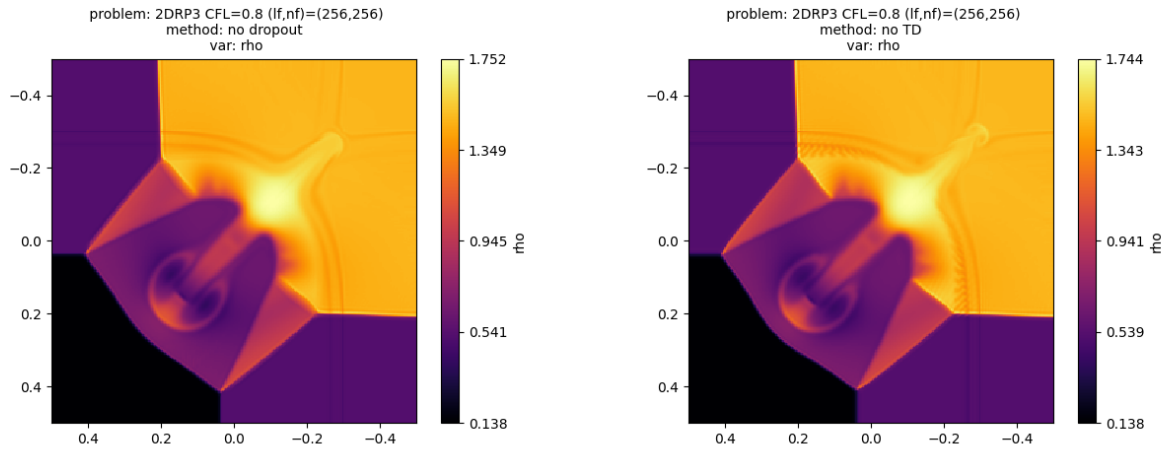


Figure 5.11 – 2DRP3 solution obtained with the NN-GP-MOOD method without dropout (Left) and without transverse diffusion (Right)

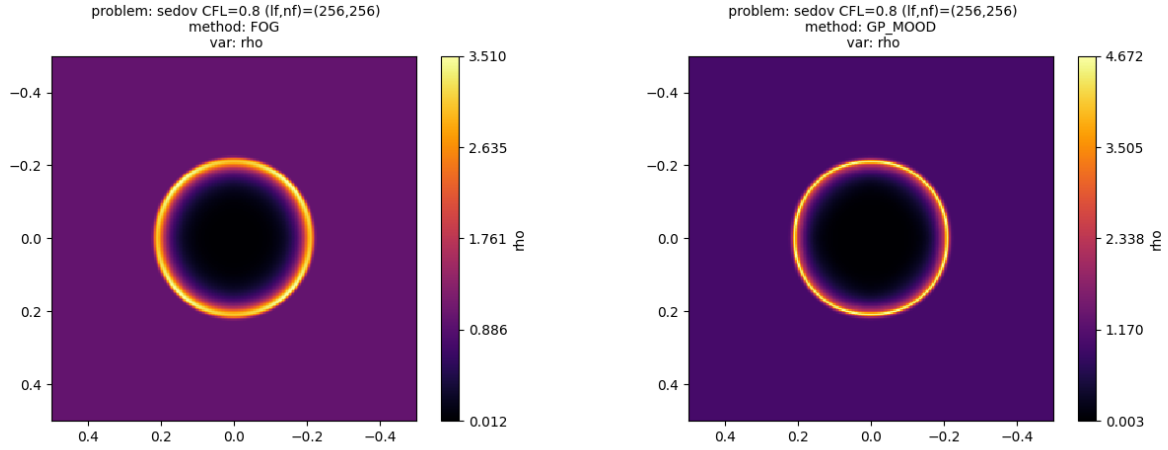


Figure 5.12 – Solutions of the Sedov problem at $t = 0.05$s obtained with the 1st (Left) and 3rd order (Right) GP-MOOD methods.

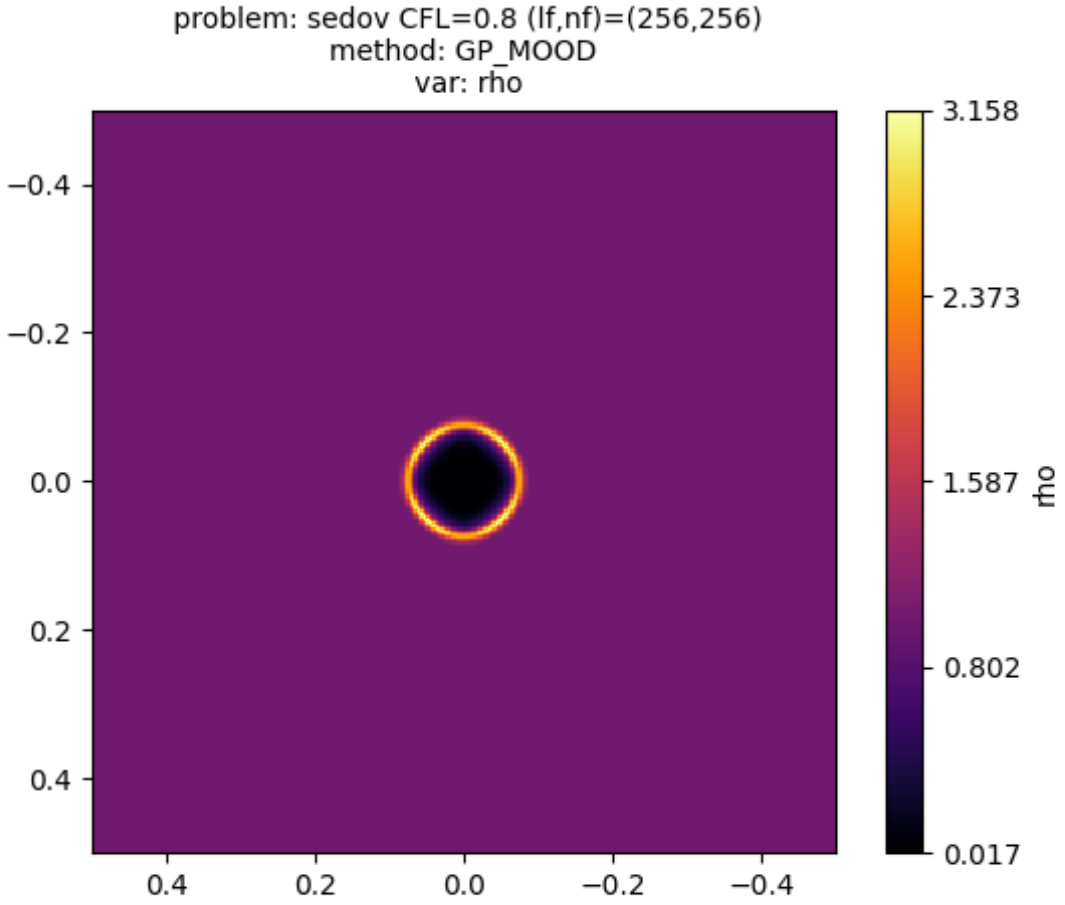
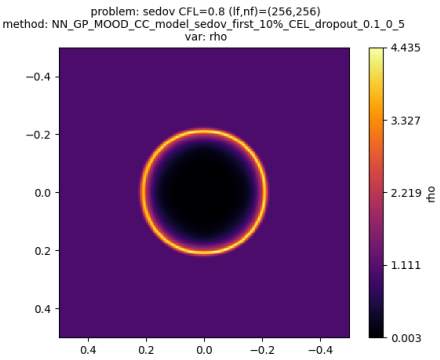
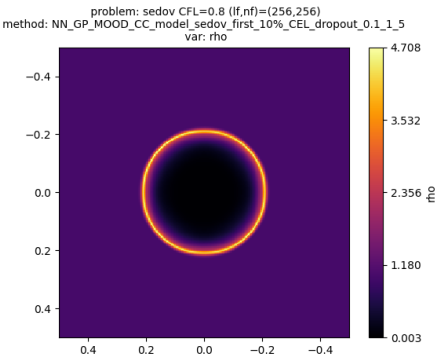
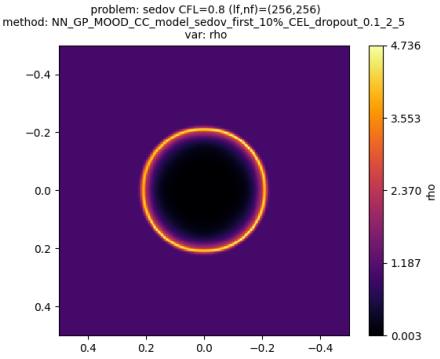
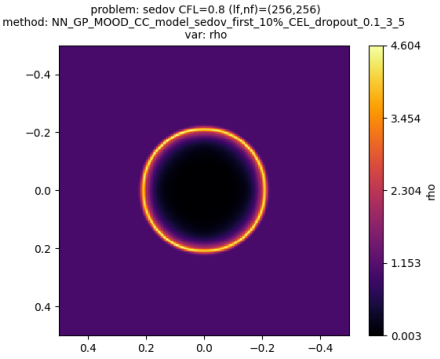


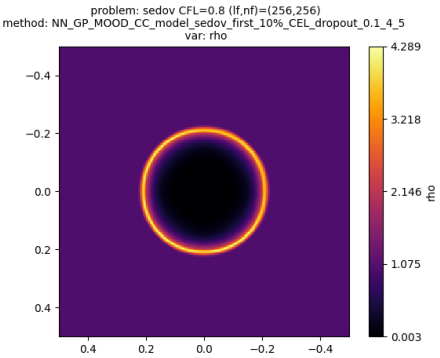
Figure 5.13 – Aspect of the solution to the Sedov problem at the end of the dataset generation phase

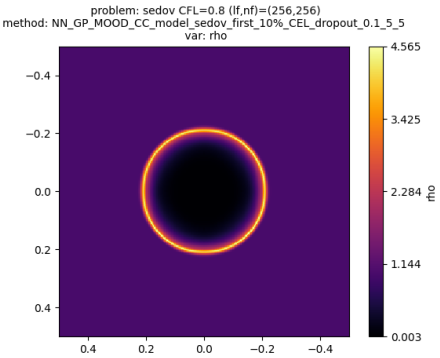

problem: sedov CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_sedov_first_10%_CEL_dropout_0.1_0_5
var: rho
4.435
3.327
2.219
1.111
0.003
rho

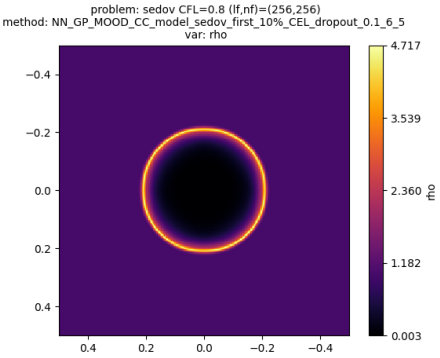

problem: sedov CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_sedov_first_10%_CEL_dropout_0.1_1_5
var: rho
4.708
3.532
2.356
1.180
0.003
rho

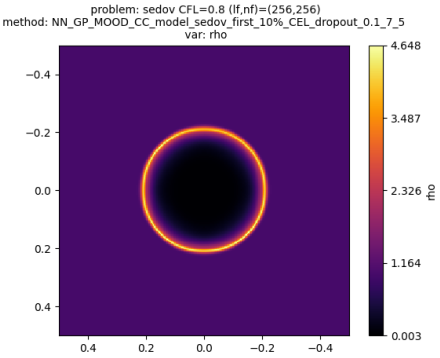

problem: sedov CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_sedov_first_10%_CEL_dropout_0.1_2_5
var: rho
4.736
3.553
2.370
1.187
0.003
rho

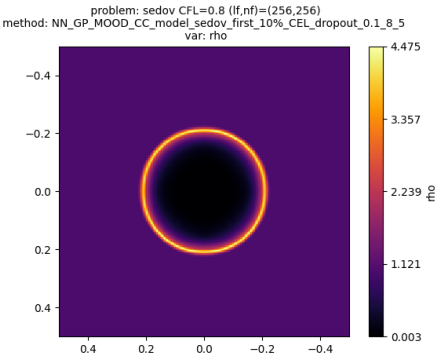

problem: sedov CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_sedov_first_10%_CEL_dropout_0.1_3_5
var: rho
4.604
3.454
2.304
1.153
0.003
rho

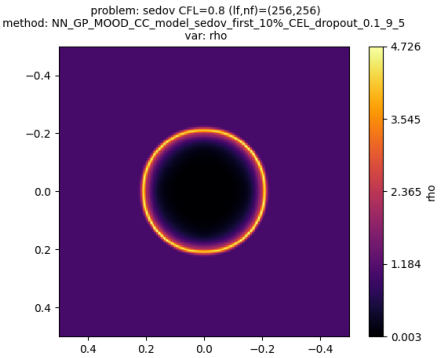

problem: sedov CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_sedov_first_10%_CEL_dropout_0.1_4_5
var: rho
4.289
3.218
2.146
1.075
0.003
rho


problem: sedov CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_sedov_first_10%_CEL_dropout_0.1_5_5
var: rho
4.565
3.425
2.284
1.144
0.003
rho


problem: sedov CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_sedov_first_10%_CEL_dropout_0.1_6_5
var: rho
4.717
3.539
2.360
1.182
0.003
rho


problem: sedov CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_sedov_first_10%_CEL_dropout_0.1_7_5
var: rho
4.648
3.487
2.326
1.164
0.003
rho


problem: sedov CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_sedov_first_10%_CEL_dropout_0.1_8_5
var: rho
4.475
3.357
2.239
1.121
0.003
rho


problem: sedov CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_sedov_first_10%_CEL_dropout_0.1_9_5
var: rho
4.726
3.545
2.365
1.184
0.003
rho

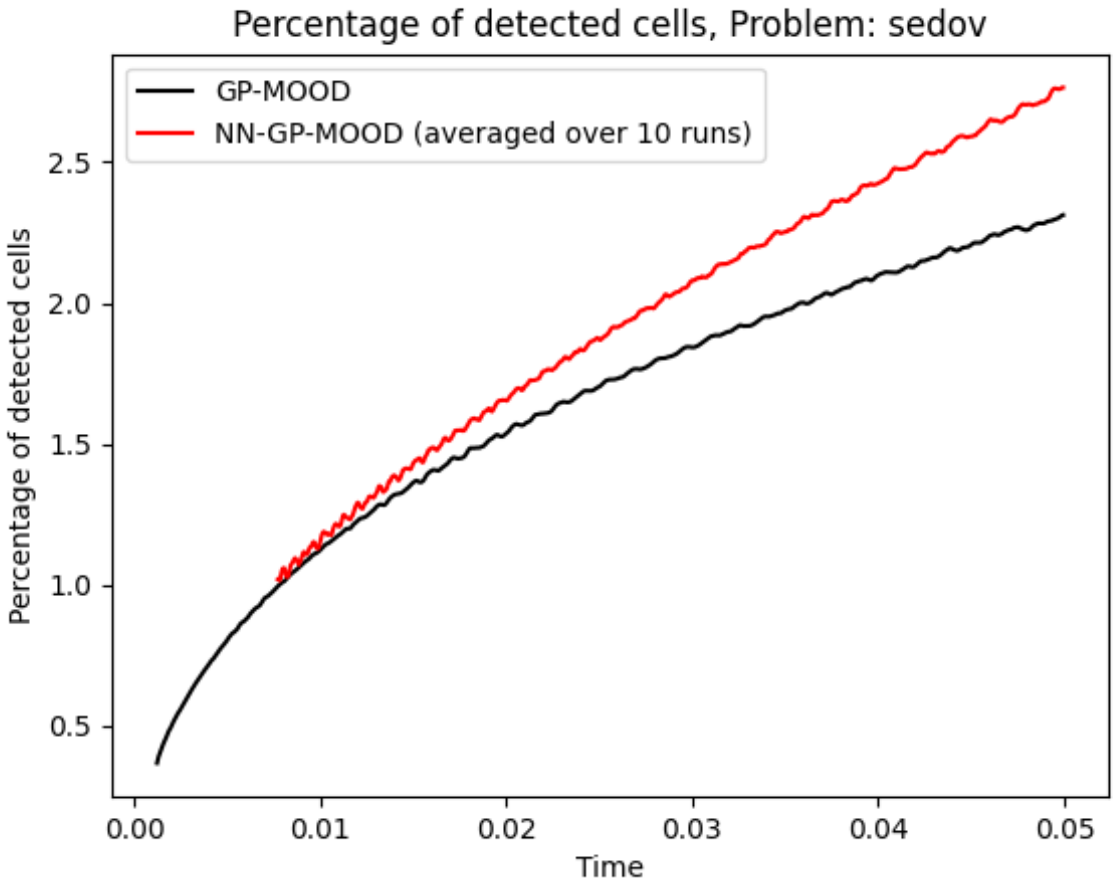


Figure 5.15 – Comparison of the number of detected cells a posteriori by the GP-MOOD method and a priori by the NN-GP-MOOD method on the Sedov problem, averaged over 10 different runs.

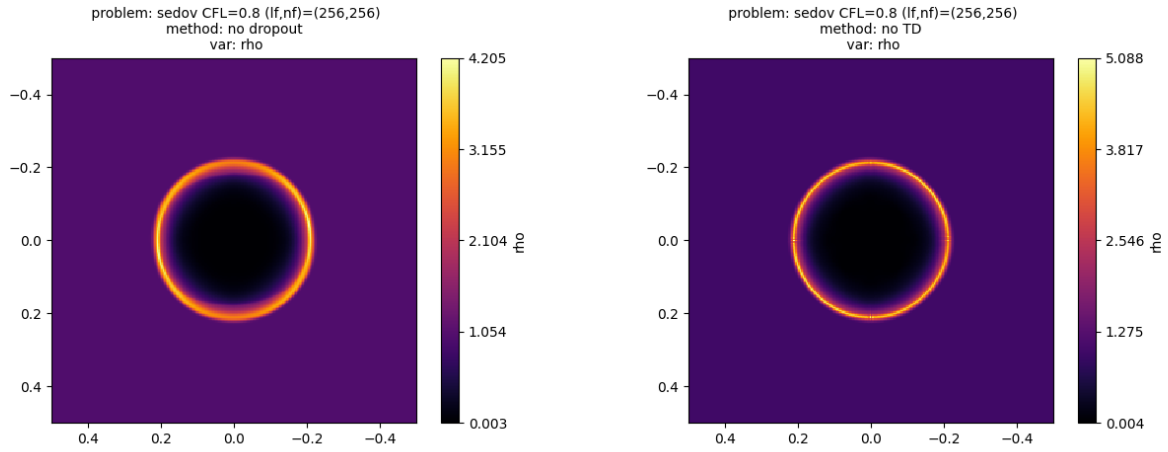


Figure 5.16 – Sedov solution obtained with the NN-GP-MOOD method without dropout (Left) and without transverse diffusion (Right)

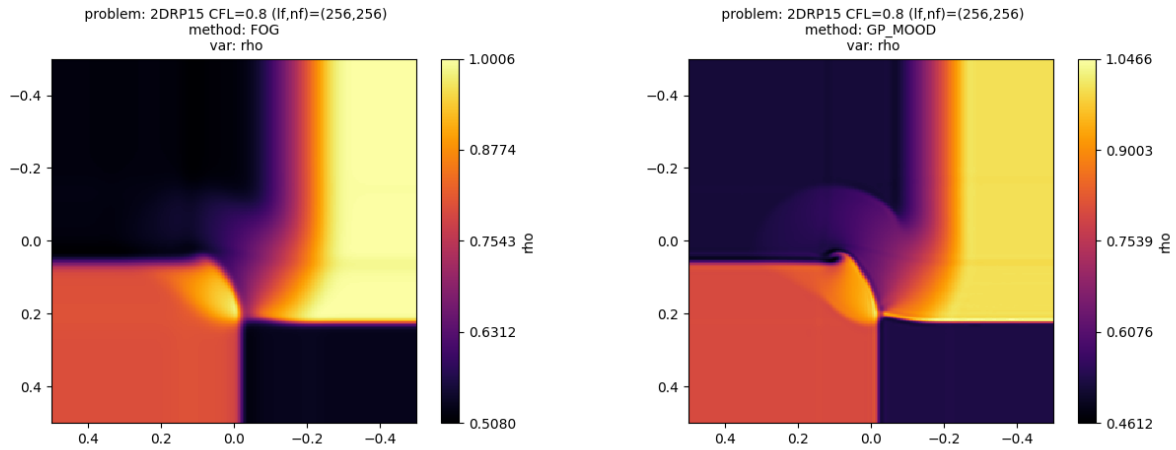


Figure 5.17 – Solutions of the 2DRP15 problem at $t = 0.2$s obtained with the 1st and 3rd order GP-MOOD methods.

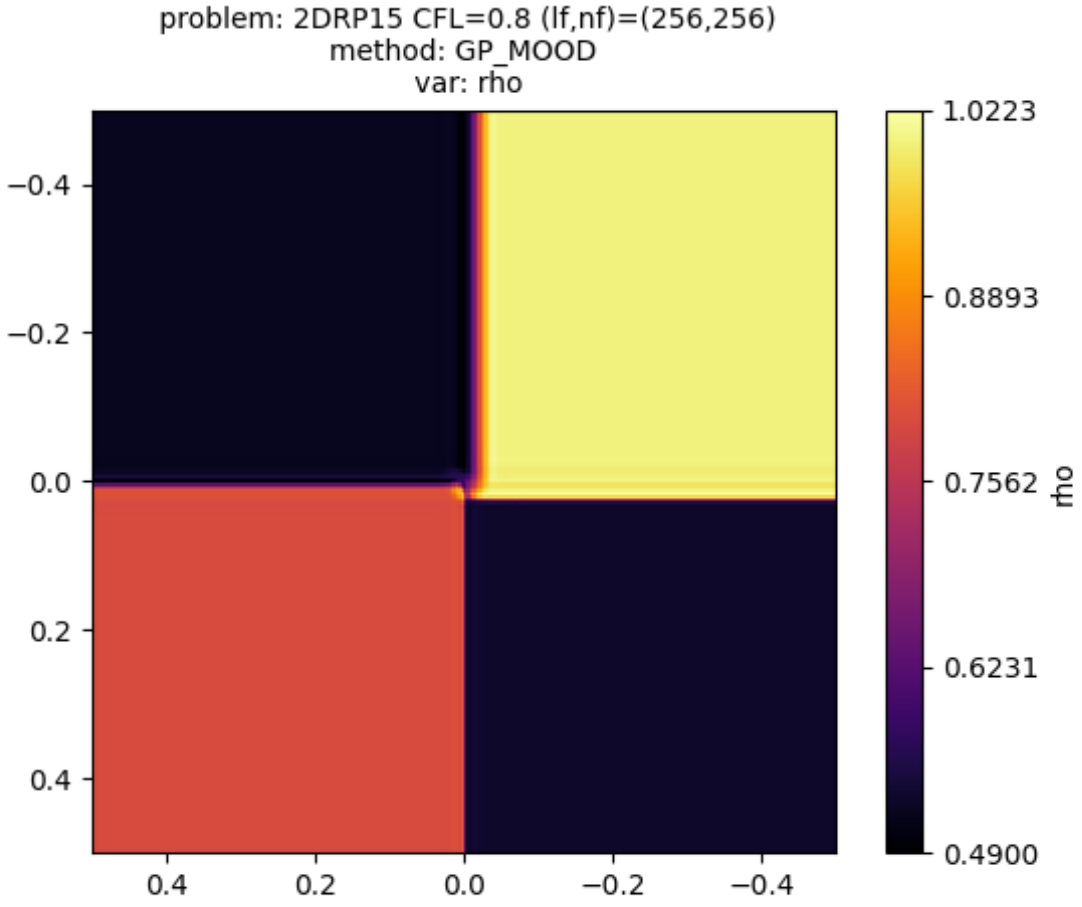
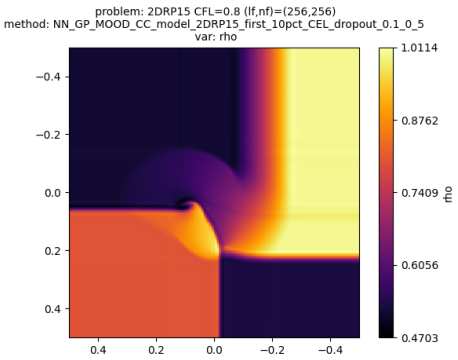
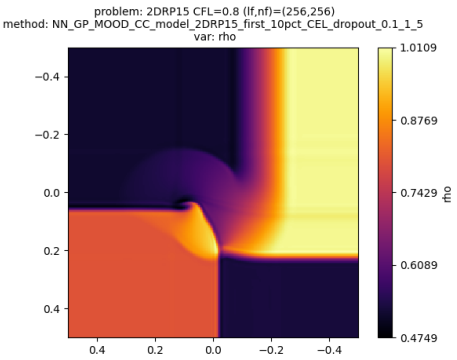
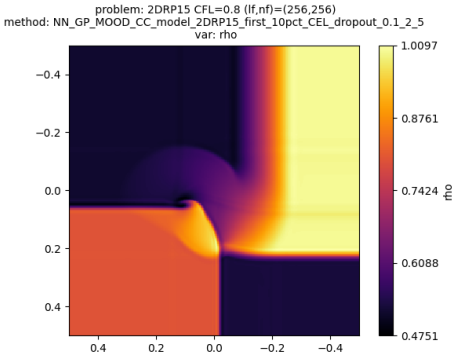
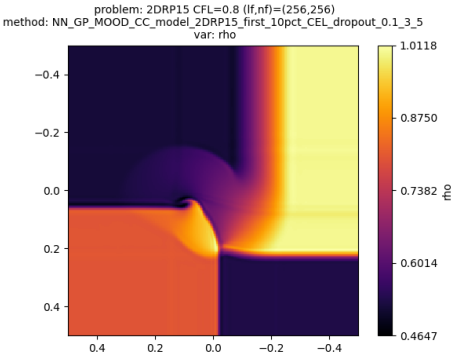


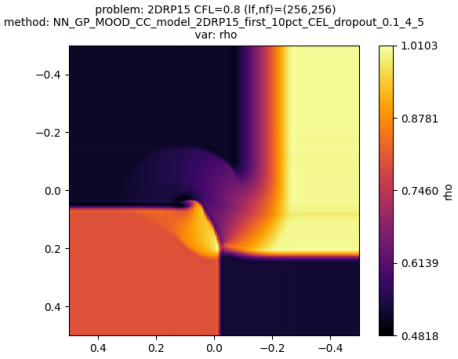
Figure 5.18 – Aspect of the solution to the 2DRP15 problem at the end of the dataset generation phase.

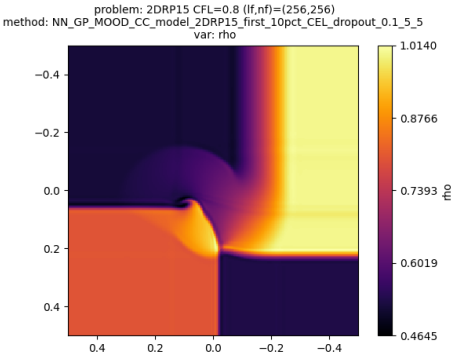

problem: 2DRP15 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP15_first_10pct_CEL_dropout_0.1_0_5
var: rho
1.0114
0.8762
0.7409
0.6056
0.4703
rho

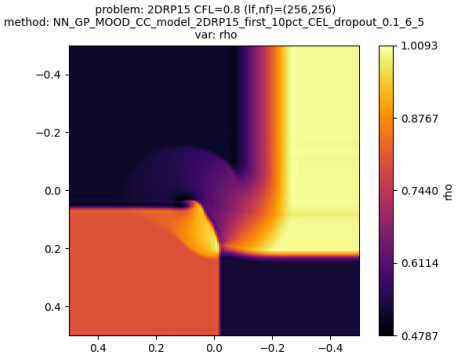

problem: 2DRP15 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP15_first_10pct_CEL_dropout_0.1_1_5
var: rho
1.0109
0.8769
0.7429
0.6089
0.4749
rho

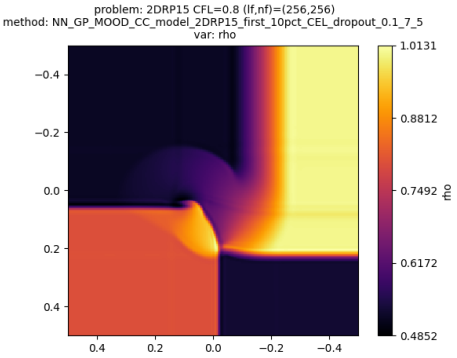

problem: 2DRP15 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP15_first_10pct_CEL_dropout_0.1_2_5
var: rho
1.0097
0.8761
0.7424
0.6088
0.4751
rho

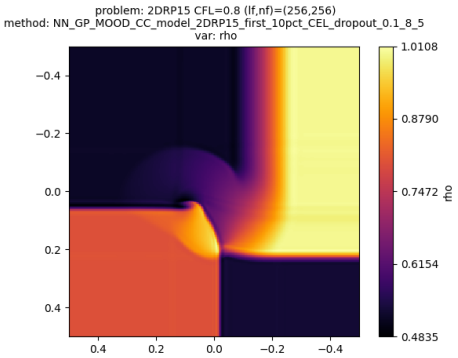

problem: 2DRP15 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP15_first_10pct_CEL_dropout_0.1_3_5
var: rho
1.0118
0.8750
0.7382
0.6014
0.4647
rho

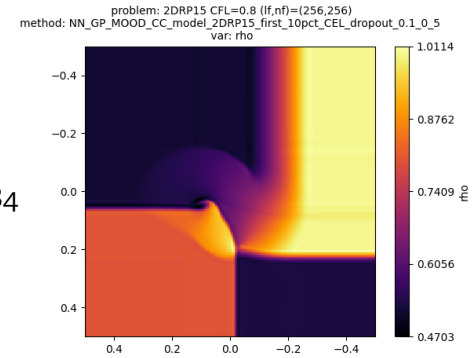

problem: 2DRP15 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP15_first_10pct_CEL_dropout_0.1_4_5
var: rho
1.0103
0.8781
0.7460
0.6139
0.4818
rho


problem: 2DRP15 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP15_first_10pct_CEL_dropout_0.1_5_5
var: rho
1.0140
0.8766
0.7393
0.6019
0.4645
rho


problem: 2DRP15 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP15_first_10pct_CEL_dropout_0.1_6_5
var: rho
1.0093
0.8767
0.7440
0.6114
0.4787
rho


problem: 2DRP15 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP15_first_10pct_CEL_dropout_0.1_7_5
var: rho
1.0131
0.8812
0.7492
0.6172
0.4852
rho


problem: 2DRP15 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP15_first_10pct_CEL_dropout_0.1_8_5
var: rho
1.0108
0.8790
0.7472
0.6154
0.4835
rho


problem: 2DRP15 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP15_first_10pct_CEL_dropout_0.1_0_5
var: rho
1.0114
0.8762
0.7409
0.6056
0.4703
rho

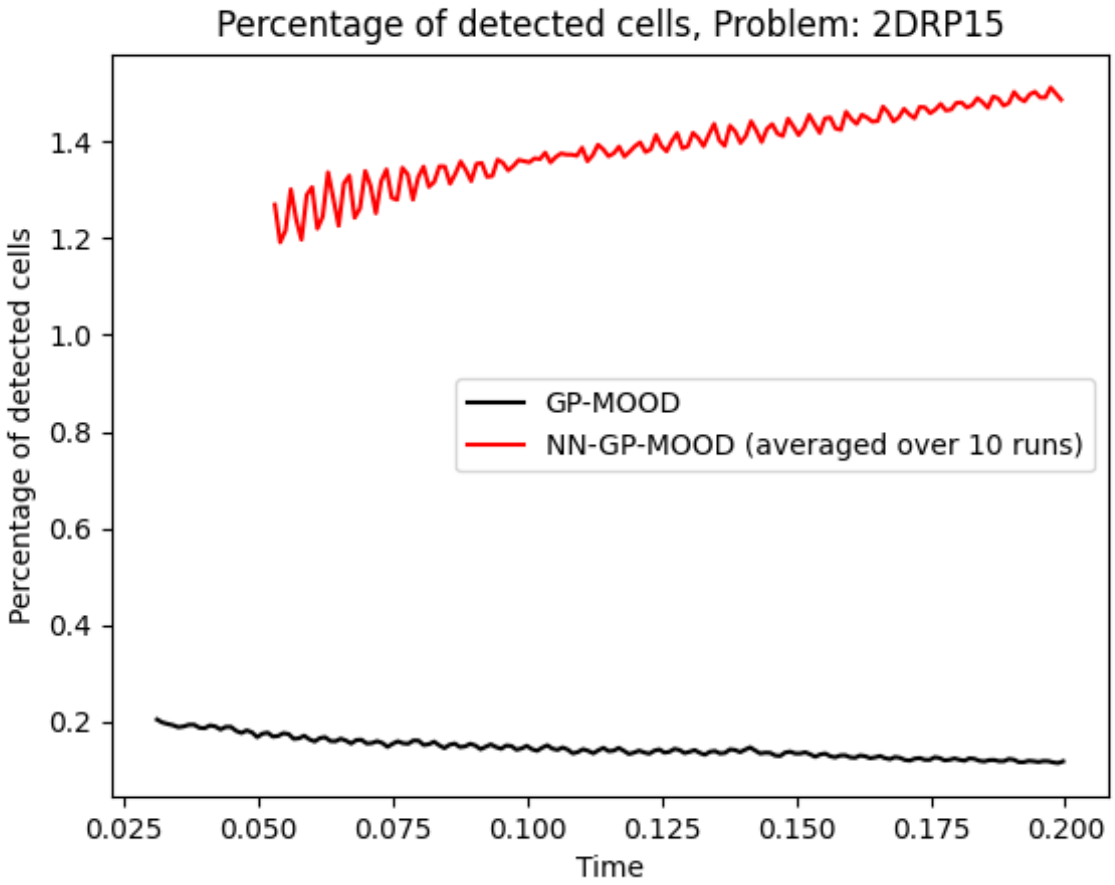


Figure 5.20 – Comparison of the number of detected cells a posteriori by the GP-MOOD method and a priori by the NN-GP-MOOD method on the 2DRP15 problem, averaged over 10 different runs.

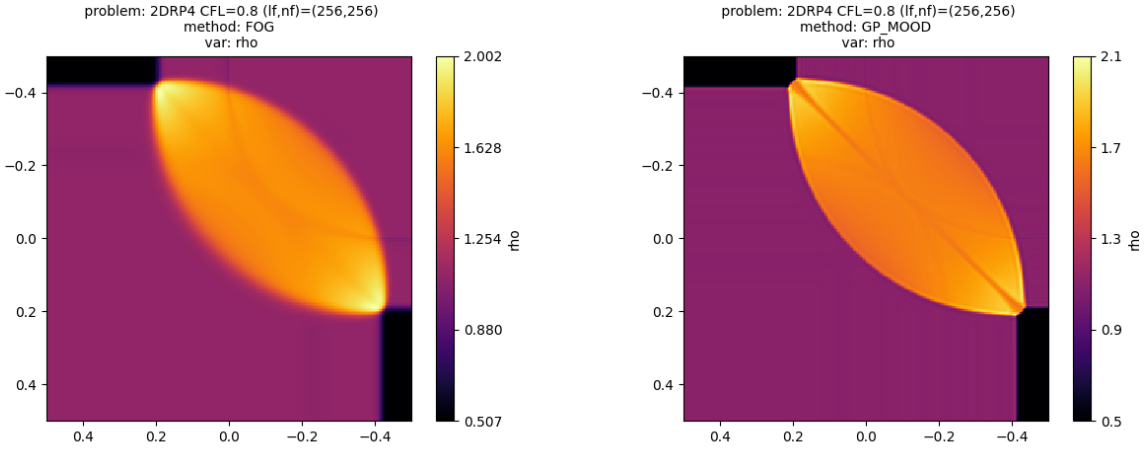


Figure 5.21 – Solutions of the 2DRP4 problem at $t = 0.25$s obtained with the 1st (Left) and 3rd order (Right) GP-MOOD method.

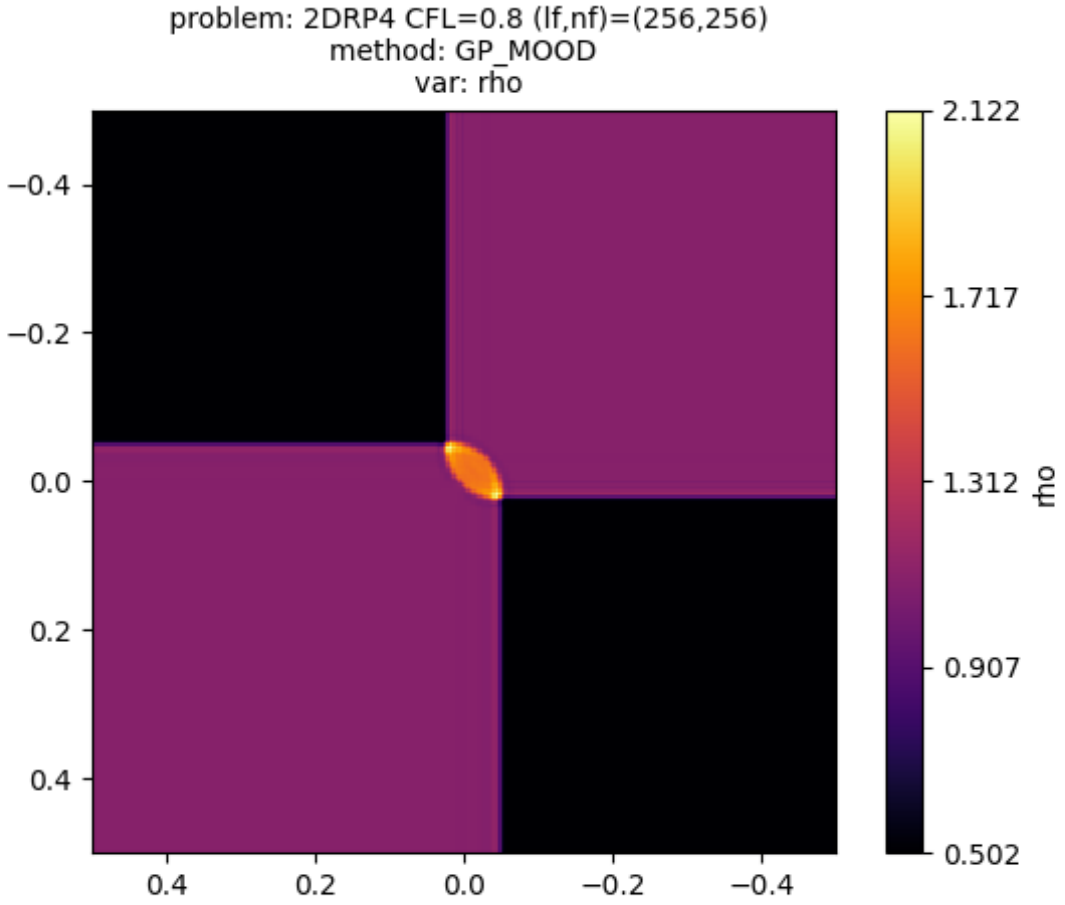
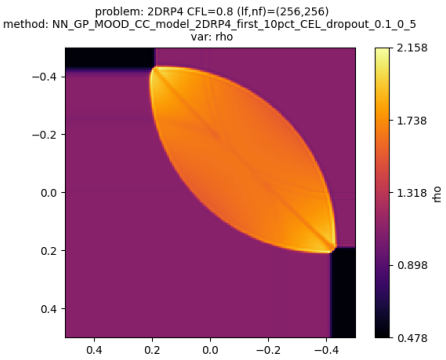
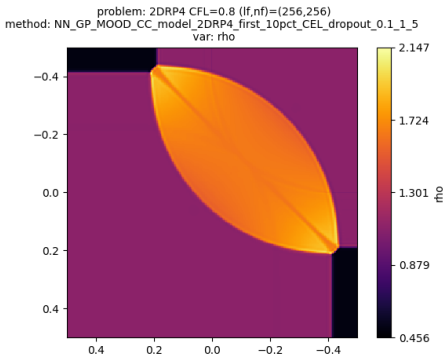
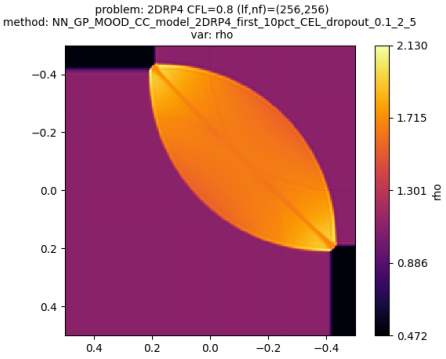
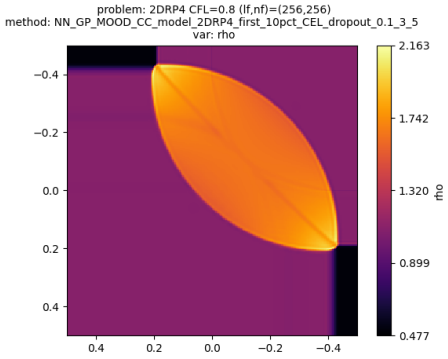


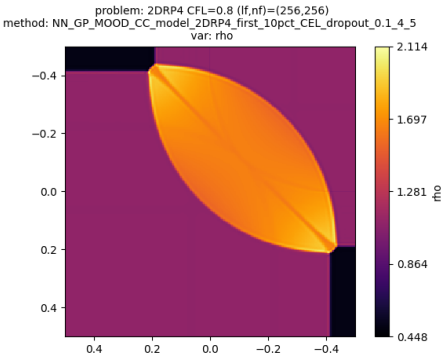
Figure 5.22 – Aspect of the solution to the 2DRP3 problem at the end of the dataset generation phase.

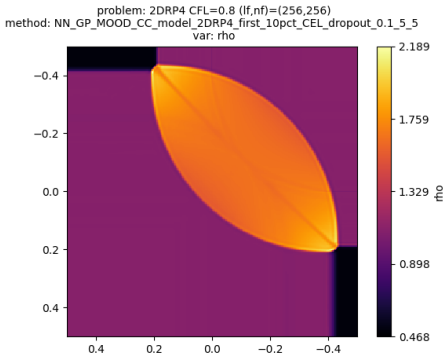

problem: 2DRP4 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP4_first_10pct_CEL_dropout_0.1_0_5
var: rho
2.158
1.738
1.318
0.898
0.478
rho

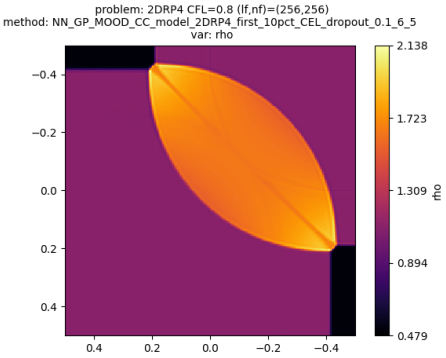

problem: 2DRP4 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP4_first_10pct_CEL_dropout_0.1_1_5
var: rho
2.147
1.724
1.301
0.879
0.456
rho

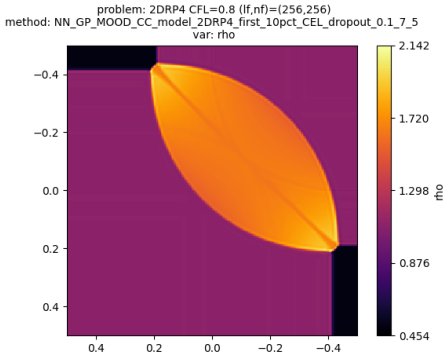

problem: 2DRP4 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP4_first_10pct_CEL_dropout_0.1_2_5
var: rho
2.130
1.715
1.301
0.886
0.472
rho

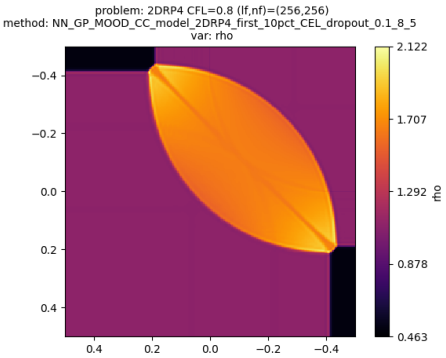

problem: 2DRP4 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP4_first_10pct_CEL_dropout_0.1_3_5
var: rho
2.163
1.742
1.320
0.899
0.477
rho

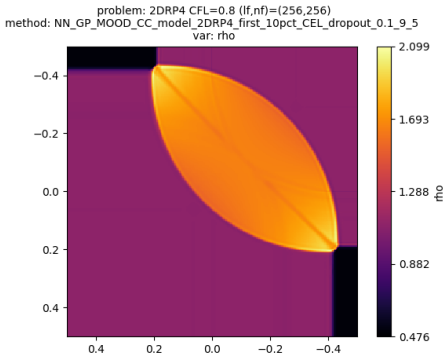

problem: 2DRP4 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP4_first_10pct_CEL_dropout_0.1_4_5
var: rho
2.114
1.697
1.281
0.864
0.448
rho


problem: 2DRP4 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP4_first_10pct_CEL_dropout_0.1_5_5
var: rho
2.189
1.759
1.329
0.898
0.468
rho


problem: 2DRP4 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP4_first_10pct_CEL_dropout_0.1_6_5
var: rho
2.138
1.723
1.309
0.894
0.479
rho


problem: 2DRP4 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP4_first_10pct_CEL_dropout_0.1_7_5
var: rho
2.142
1.720
1.298
0.876
0.454
rho


problem: 2DRP4 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP4_first_10pct_CEL_dropout_0.1_8_5
var: rho
2.122
1.707
1.292
0.878
0.463
rho


problem: 2DRP4 CFL=0.8 (lf,nf)=(256,256)
method: NN_GP_MOOD_CC_model_2DRP4_first_10pct_CEL_dropout_0.1_9_5
var: rho
2.099
1.693
1.288
0.882
0.476
rho

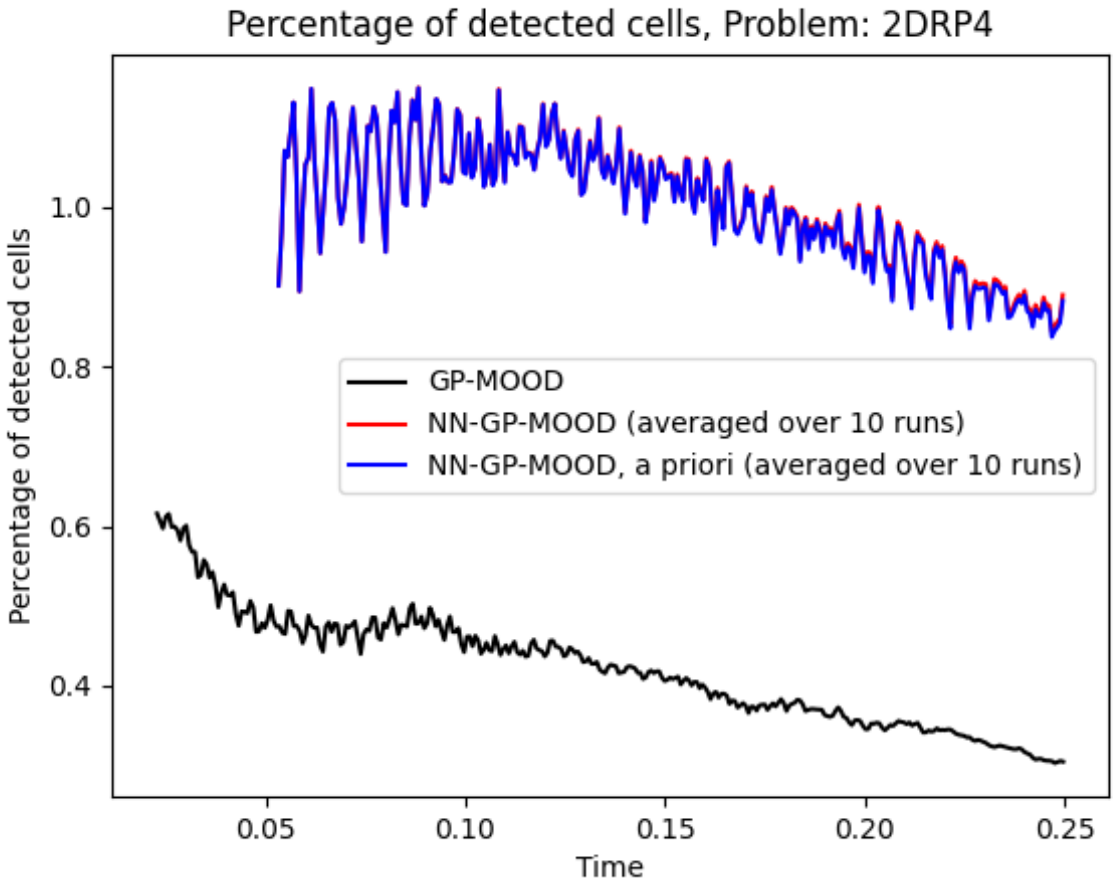


Figure 5.24 – Comparison of the number of detected cells a posteriori by the GP-MOOD method and a priori by the NN-GP-MOOD method on the 2DRP4 problem, averaged over 10 different runs.

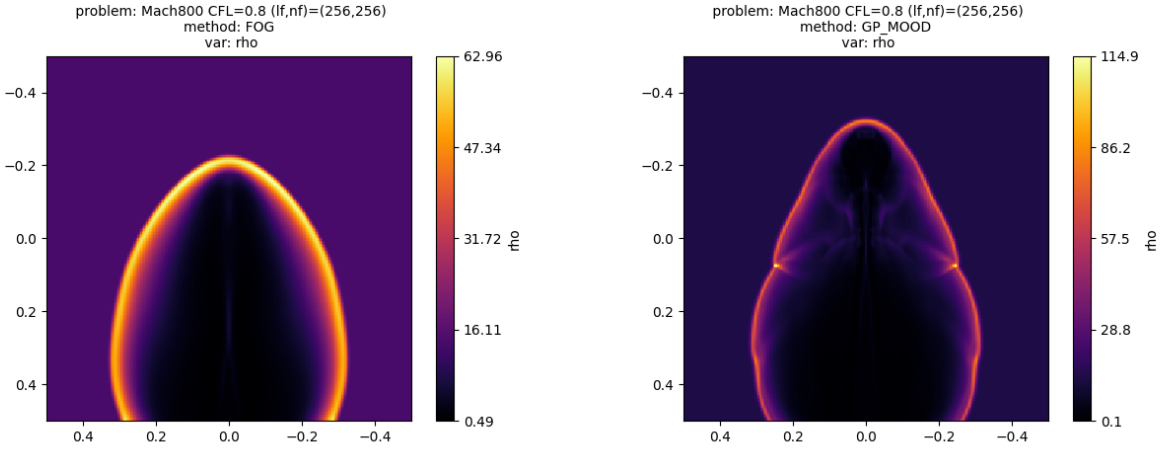


Figure 5.25 – Solutions of the Mach 800 problem obtained with the 1st (Left) and 3rd order (Right) GP-MOOD methods.

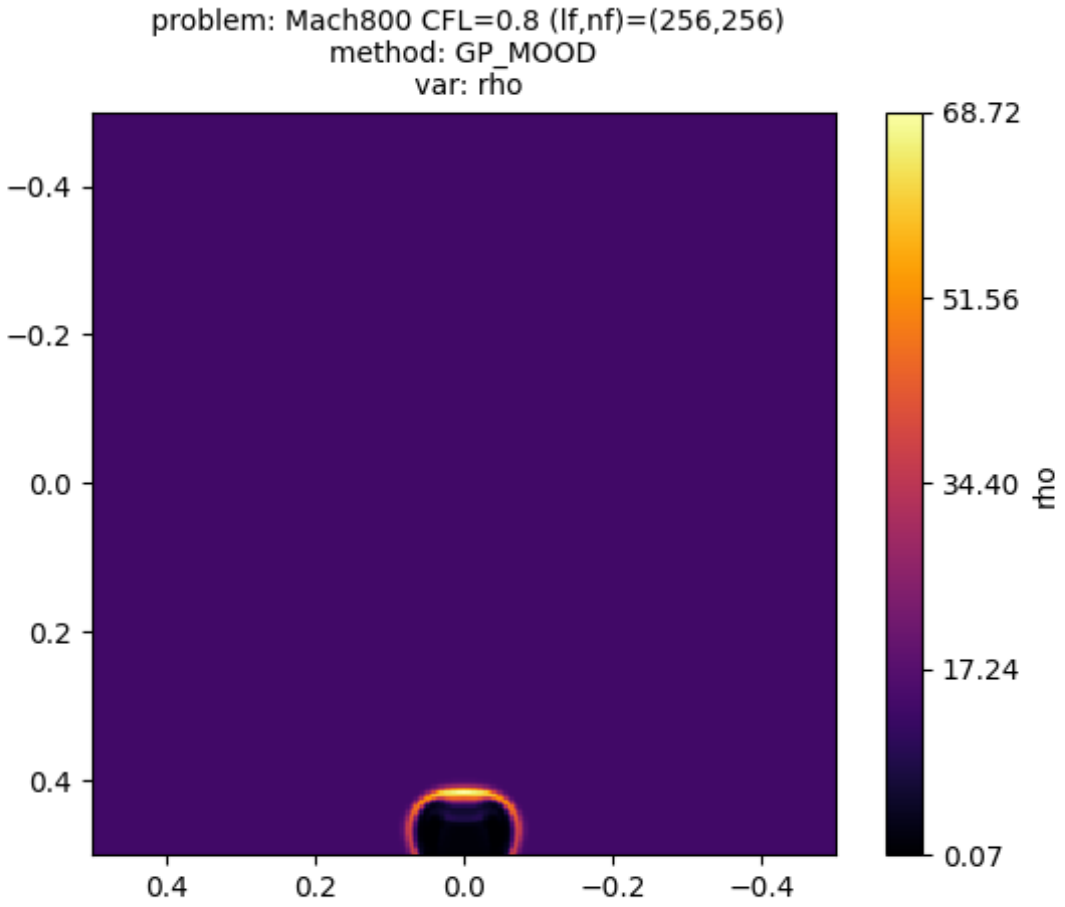


Figure 5.26 – Aspect of the solution to the Mach800 problem at the end of the dataset generation phase.

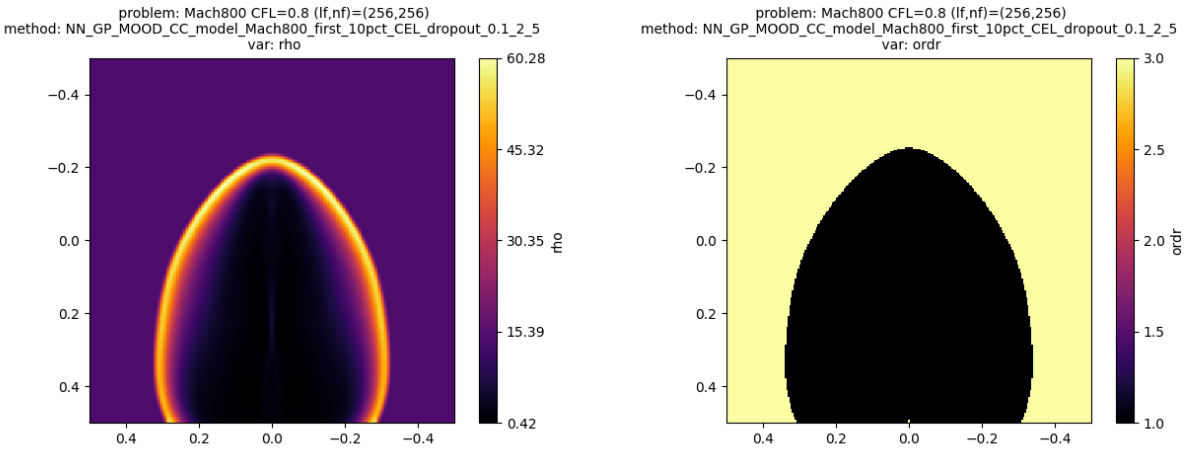


Figure 5.27 – Over-diffused solution of the Mach 800 problem obtained with the NN-GP-MOOD method. Left : density, Right : order map

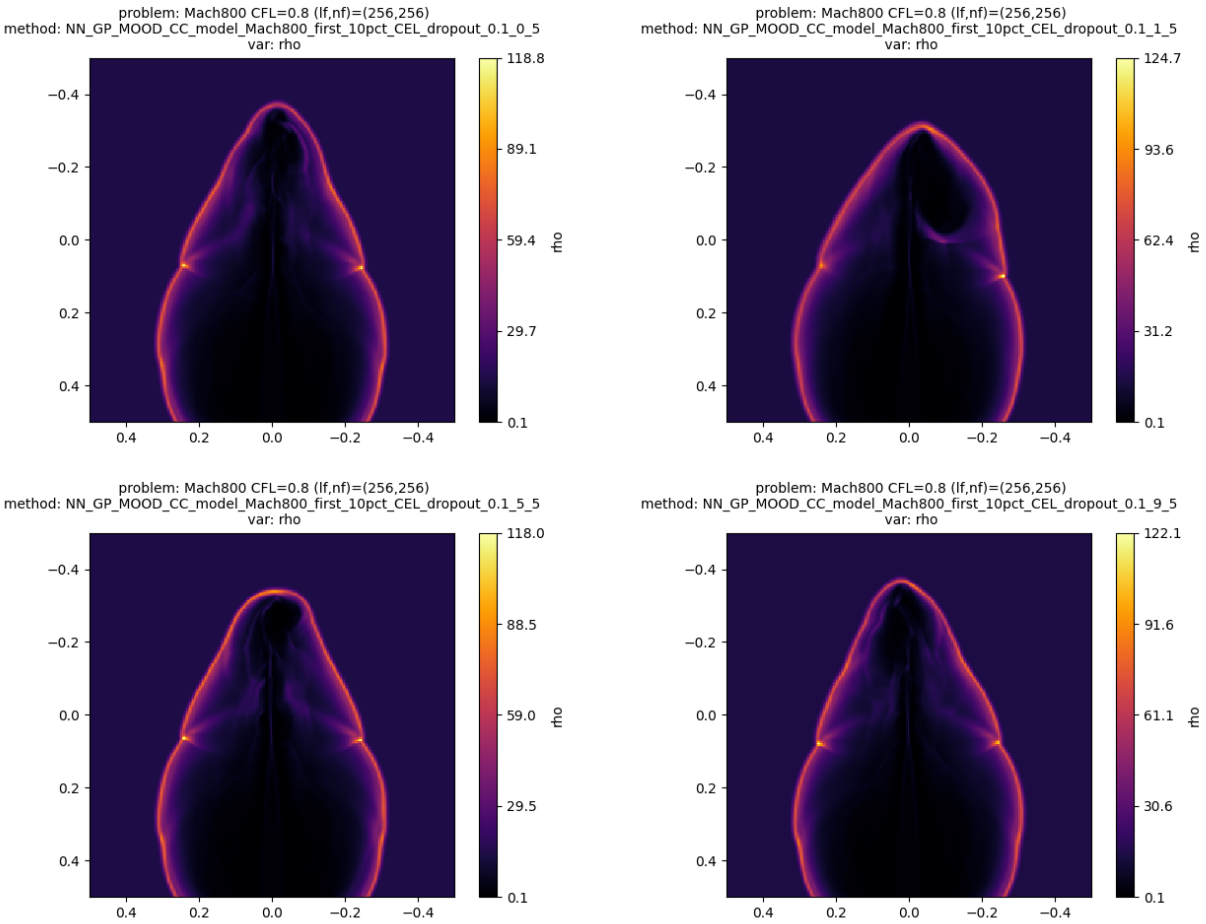


Figure 5.28 – Acceptable solutions of the Mach 800 problem obtained with the NN-GP-MOOD method.

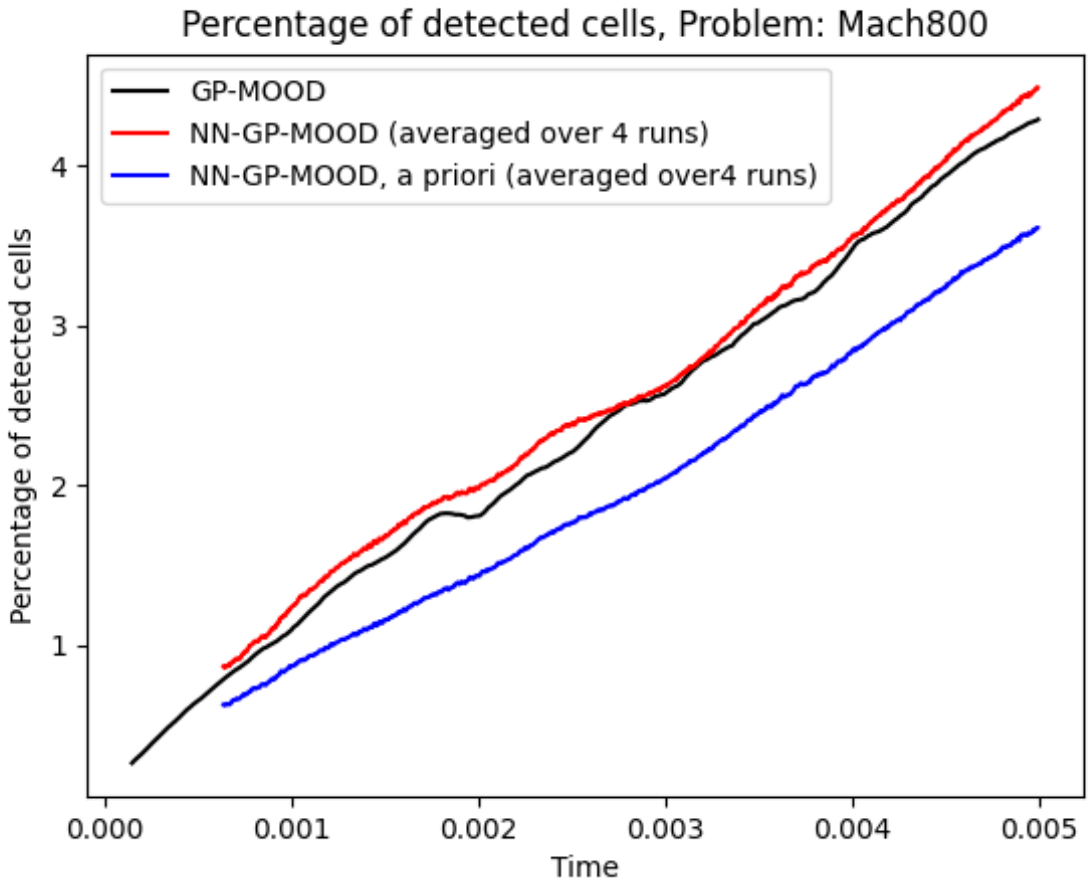


Figure 5.29 – Comparison of the number of detected cells a posteriori by the GP-MOOD method and a priori by the NN-GP-MOOD method on the Mach 800 problem, averaged over 4 different runs.


ÉCOLE DOCTORALE
Astronomie
et Astrophysique
d'Île-de-France (AAIF)
université
PARIS-SACLAY