THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE STRASBOURG

ÉCOLE DOCTORALE
MATHÉMATIQUES, SCIENCES DE L'INFORMATION ET DE L'INGÉNIEUR DE STRASBOURG

Spécialité : **MATHÉMATIQUES**

Présentée par
Marc WOLFF

Pour obtenir le grade de
DOCTEUR DE l'UNIVERSITÉ DE STRASBOURG

Intitulée

ANALYSE MATHÉMATIQUE ET NUMÉRIQUE DU SYSTÈME DE LA MAGNÉTOHYDRODYNAMIQUE RÉSISTIVE AVEC TERMES DE CHAMP MAGNÉTIQUE AUTO-GÉNÉRÉ

Dirigée par Éric SONNENDRUCKER

Laboratoire de rattachement
Institut de Recherche Mathématique Avancée
Université de Strasbourg

Organisme d'accueil
CEA, DAM, DIF
F-91297 Arpajon, France

Soutenue le 14 octobre 2011 devant le jury composé de :

M. JAOUEN Stéphane	Encadrant CEA
M. SONNENDRUCKER Éric	Directeur de thèse
M. COQUEL Frédéric	Rapporteur
M. MUNZ Claus-Dieter	Rapporteur
M. DESPRÉS Bruno	Examinateur
M. HELLUY Philippe	Examinateur

ANALYSE MATHÉMATIQUE ET NUMÉRIQUE DU SYSTÈME DE LA MAGNÉTOHYDRODYNAMIQUE RÉSISTIVE AVEC TERMES DE CHAMP MAGNÉTIQUE AUTO-GÉNÉRÉ

M. Wolff

ANALYSE MATHÉMATIQUE ET NUMÉRIQUE DU SYSTÈME DE LA MAGNÉTOHYDRODYNAMIQUE RÉSISTIVE AVEC TERMES DE CHAMP MAGNÉTIQUE AUTO-GÉNÉRÉ

Résumé

Ce travail est consacré à la construction de méthodes numériques permettant la simulation de processus d'implosion de coquilles en fusion par confinement inertiel (FCI) avec prise en compte des termes de champ magnétique auto-généré. Dans ce document, on commence par décrire le modèle de magnétohydrodynamique résistive à deux températures considéré ainsi que les relations de fermeture utilisées. Le système d'équations ainsi obtenu est alors divisé en sous-systèmes selon la nature de l'opérateur mathématique sous-jacent pour lesquels l'on propose ensuite des schémas numériques adaptés. On insiste notamment sur le développement de schémas volumes finis pour l'opérateur hyperbolique, ce dernier correspondant aux équations d'Euler ou de la magnétohydrodynamique idéale selon que l'on tienne compte ou non des termes de champ magnétique. Plus précisement, on propose une nouvelle classe de schémas d'ordre élevé à directions alternées construits dans le formalisme Lagrange + projection sur grille cartésienne qui présentent l'originalité d'être particulièrement bien adaptés aux calculateurs modernes grâce, entre autres, au traitement par directions alternées et à l'utilisation de techniques de viscosité artificielle. Cette propriété est illustrée par des mesures de performance séquentielle et d'efficacité parallèle. On combine ensuite les schémas hyperboliques développés avec des méthodes de type volumes finis permettant le traitement semi-implicite des termes de conduction thermique et résistive et une prise en compte explicite des termes de champ magnétique auto-générés. Afin d'étudier les caractéristiques et les effets des champs magnétiques auto-générés, on présente enfin un cas test de capsule FCI simulée à partir du début de la phase de décélération.

Mots-clés

Équations d'Euler, magnétohydrodynamique idéale, volumes finis, ordre élevé, formalisme Lagrange + projection, splitting directionnel, viscosité artificielle.

MATHEMATICAL AND NUMERICAL ANALYSIS OF THE RESISTIVE MAGNETOHYDRODYNAMICS SYSTEM OF EQUATIONS WITH SELF-GENERATED MAGNETIC FIELD TERMS

Abstract

This work is devoted to the construction of numerical methods that allow the accurate simulation of inertial confinement fusion (ICF) implosion processes by taking self-generated magnetic field terms into account. In the sequel, we first derive a two-temperature resistive magnetohydrodynamics model and describe the considered closure relations. The resulting system of equations is then split in several subsystems according to the nature of the underlying mathematical operator. Adequate numerical methods are then proposed for each of these subsystems. Particular attention is paid to the development of finite volume schemes for the hyperbolic operator which actually is the hydrodynamics or ideal magnetohydrodynamics system depending on whether magnetic fields are considered or not. More precisely, a new class of high-order accurate dimensionally split schemes for structured meshes is proposed using the Lagrange-remap formalism. One of these schemes' most innovative features is that they have been designed in order to take advantage of modern massively parallel computer architectures. This property can for example be illustrated by the dimensionally split approach or the use of artificial viscosity techniques and is practically highlighted by sequential performance and parallel efficiency figures. Hyperbolic schemes are then combined with finite volume methods for dealing with the thermal and resistive conduction operators and taking magnetic field generation into account. In order to study the characteristics and effects of self-generated magnetic field terms, simulation results are finally proposed with the complete two-temperature resistive magnetohydrodynamics model on a test problem that represents the state of an ICF capsule at the beginning of the deceleration phase.

Key words

Euler equations, ideal magnetohydrodynamics, finite volume methods, high-order accuracy, Lagrange-remap formalism, dimensional splitting, artificial viscosity.

Remerciements

Je tiens tout d'abord à remercier Stéphane Jaouen qui s'est occupé de l'encadrement de cette thèse au CEA et qui est en grande partie responsable de l'intérêt que je porte aujourd'hui à la discipline de la simulation numérique. Ces années de travail à tes côtés ont été très enrichissantes, aussi bien sur le plan personnel que sur le plan professionnel. Je garderai notamment un excellent souvenir - parmi tant d'autres - de notre fructueuse collaboration lors de l'édition 2010 du CEMRACS.

J'exprime toute ma gratitude à Éric Sonnendrucker, mon directeur de thèse, qui s'est toujours montré disponible en cas de besoin. Merci de ton aide et, de manière plus générale, merci de m'avoir transmis le goût des mathématiques appliquées au cours de l'ensemble de mes études à l'Université de Strasbourg.

Mes remerciements vont également à Hervé Jourdren, chef du laboratoire PNP au CEA, dont les conseils avisés se sont bien souvent révélés salvateurs. Je te suis très reconnaissant de m'avoir fait profiter de ta grande culture scientifique et de m'avoir poussé à m'initier aux joies du calcul hautes performances. Je remercie par ailleurs Bruno Scheurer, chef du service NEC, et Stéphanie Perrot, secrétaire de ce même service, qui ont contribué au bon déroulement de mon séjour au CEA sur les plans scientifique et administratif.

Frédéric Coquel et Claus-Dieter Munz m'ont fait l'honneur d'être les rapporteurs de cette thèse et je les en remercie. J'exprime également toute ma reconnaissance à Bruno Després ainsi qu'à Philippe Helluy pour leur participation en tant qu'examinateurs au jury de soutenance.

Ces remerciements ne seraient pas complets sans mentionner tous mes anciens collègues stagiaires et thésards de l'espace Nord qui ont contribué par leurs conseils à l'avancée de mes travaux et par leur bonne humeur à l'intérêt des pauses café : Gilles, Gaël, François, Stéphane (le mérites-tu vraiment ?), Manu, Jean-Yves, Sylvain, Jean-Baptiste, Jérôme, Cécile, Kévin et Gauthier.

Contents

Introduction

Contexte physique

La fusion par confinement inertiel (FCI) est un procédé consistant à porter une quantité de combustible - généralement un micro-ballon de deutérium-tritium (DT) gazeux enrobé dans une phase cryogénique - à des conditions de densité, de température et de pression permettant l'établissement de réactions de fusion nucléaire (on parle alors d'ignition). Pour cela, le micro-ballon est comprimé par un fort dépôt d'énergie sous l'effet duquel la coquille de DT se vaporise sous forme de plasma. On distingue deux approches permettant de réaliser ce dépôt d'énergie. Dans le cas de l'attaque dite directe, il est obtenu en dirigeant un grand nombre de lasers à haute énergie sur la cible de DT. Dans le cas de l'attaque dite indirecte, la cible est placée dans un cylindre métallique (appelé hohlraum) qui est lui-même soumis à une irradiation laser, entraînant l'émissions de rayons X qui sont absorbés par la surface du micro-ballon et créent les conditions de confinement désirées.

Cette étude se place dans le cadre du processus de fusion par confinement inertiel en attaque directe. On sait que, dans ce contexte, des difficultés peuvent survenir pendant la phase d'implosion [64] consécutive à l'irradiation laser en raison d'écarts à la sphéricité au niveau de la surface de la coquille. Ceux-ci proviennent généralement soit de défauts d'usinage du micro-ballon, soit de l'anisotropie de l'éclairement laser. Les perturbations que constituent ces écarts à la sphéricité sont amplifiées au cours du processus d'implosion et donnent naissance à des instabilités hydrodynamiques - généralement de type Richtmyer-Meshkov ou Rayleigh-Taylor - permettant l'apparition de champs magnétiques. En effet, comme nous le verrons au cours de cette étude, on observe des pertes de colinéarité des gradients de densité et de température au niveau des zones d'instabilités et celles-ci induisent l'apparition de termes de champ magnétique dits auto-générés (puisqu'ils ne proviennent pas d'un champ extérieur) dont les conséquences peuvent être tout à fait significatives. Ceux-ci ont notamment pour effet de réduire la conduction thermique électronique et sont susceptibles, suivant leur intensité, de modifier la distribution de température au voisinage du point chaud, modifiant ainsi les conditions d'ignition [42]. Dès lors, il semble important de suivre l'évolution des composantes de champ magnétique en plus des quantités thermodynamiques usuelles dans les codes de simulation.

Objectifs

L'objectif de cette étude est de développer des méthodes numériques permettant la prise en compte des termes de champ magnétique auto-généré dans les simulations d'implosion de micro-ballons. Pour cela, nous établirons tout d'abord un modèle de magnétohydrodynamique (MHD) résistive à deux températures représentant de manière suffisamment exhaustive la physique du problème considéré. Nous proposerons ensuite une méthode de résolution numérique du système d'équations établi en s'appuyant sur des schémas précis et efficaces. Une attention toute particulière sera portée à l'implémentation de ces schémas numériques

et plus globalement aux problématiques de performances et de parallélisme dans le but de tirer parti de la puissance des calculateurs modernes caractérisés par leurs architectures massivement parallèles. La dernière partie de cette thèse sera consacrée à l'exploitation du code de simulation développé, plus précisément à l'étude des termes de champ magnétique auto-généré et de leurs effets sur la base de résultats numériques préliminaires.

Contexte numérique

L'implosion de micro-ballons en fusion par confinement inertiel est un processus qui présente en théorie une symétrie sphérique, ce qui pourrait servir à réduire le nombre de dimensions dans le cadre de la simulation numérique. La prise en compte des défauts de sphéricité de la coquille implique cependant que l'on considère des géométries plus riches. L'approche la plus générale consisterait à résoudre le problème considéré en trois dimensions d'espace. Pour des raisons évidentes de coût de calcul, nous avons toutefois choisi de restreindre notre étude aux géométries axisymétriques en deux dimensions d'espace.

Nous avons par ailleurs choisi de discrétiser les équations en jeu par des schémas numériques de type volumes finis sur des maillages structurés, ce cadre étant le mieux adapté à la construction de méthodes de résolution à hautes performances.

Aspects hautes performances

La simulation numérique est un domaine en constante évolution dont les problématiques se rapprochent de plus en plus de celles du calcul hautes performances. D'une part, la nécessité de produire des résultats de simulation fiables fait que les systèmes physiques étudiés sont de plus en plus complexes. D'autre part, les architectures des calculateurs modernes évoluent vers des configurations massivement parallèles comportant un grand nombre de coeurs d'exécution et un espace mémoire par coeur qui diminue de facto. Ces réalités induisent des contraintes dans la construction et l'implémentation de méthodes numériques : celles-ci doivent à la fois être performantes (afin de permettre la résolution de problèmes multi-physiques complexes) et avoir de bonnes aptitudes de passage à l'échelle (compte tenu de l'architecture des calculateurs sur lesquels elles sont amenées à être exécutées). Le travail de thèse que nous présentons adopte une approche novatrice en ce sens que la majeure partie des choix effectués a été conditionnée par les contraintes exposées ci-dessus. Ceux-ci demandent bien entendu à être validés en pratique. Nous nous appuyerons pour cela sur des outils de mesure des performances que nous décrivons ci-dessous.

Outils de mesure des performances

Nous effectuons dans cette étude une analyse complète des performances de la plateforme de simulation mise en œuvre afin de déterminer si les choix effectués portent leurs fruits en pratique. Nous nous basons pour cela sur trois mesures - le *grind time*, le nombre d'opérations flottantes effectuées par seconde et l'efficacité parallèle - qui, pris ensemble, fournissent une estimation relativement fiable des capacités d'un code de calcul en termes de performances et de parallélisme.

Le *grind time*

Le *grind time* est défini comme le temps nécessaire à l'avancée en temps des inconnues par maille et par cycle de calcul :

$$\text{grind time} = \frac{\text{temps de restitution}}{\text{nombre de mailles} \times \text{nombre d'itérations}}.$$

Il s'agit donc de l'indicateur critique de performance puisque de lui dépendra le temps de restitution pour un problème et un maillage donnés. À titre indicatif, on considère qu'un code industriel se doit d'avoir un *grind time* inférieur à 10 microsecondes par maille et par cycle pour un coeur de calcul d'un processeur récent.

Le nombre d'opérations flottantes effectuées par seconde

Nous mesurerons également les performances de notre solveur en termes d'opérations flottantes effectuées par seconde exprimées en FLOPS (*floating operations per second*). En pratique, celles-ci sont comptées pendant l'exécution selon un protocole décrit en section 4.4.1. Le résultat obtenu est ensuite comparé à la performance crête théorique du processeur utilisé et donne une idée de son exploitation et du potentiel d'optimisation du code. À titre indicatif, on estime que les codes scientifiques exploitent en moyenne environ 10% de la puissance crête disponible.

L'efficacité parallèle

Nous évaluerons enfin l'efficacité parallèle de notre code de simulation qui mesure sa capacité à exploiter un grand nombre de processus lorsqu'il est exécuté en parallèle. Plus précisement, il s'agit du ratio du temps de restitution du solveur séquentiel par le temps de restitution du solveur parallèle sur un même problème. Cette mesure (appelée efficacité parallèle forte) peut cependant se révéler difficile à estimer. Pour un très grand nombre de processus, la taille du problème à considérer se doit alors d'être importante, ce qui implique de très importantes ressources mémoire et rend le temps de restitution du solveur séquentiel prohibitif. Nous adoptons par conséquent une approche différente consistant à calculer le rapport du temps d'exécution séquentiel sur un problème donné par le temps d'exécution sur N processus avec un maillage N fois plus grand (on parle alors d'efficacité parallèle faible).

Quelques principes permettant le développement de méthodes numériques performantes et passant à l'échelle

La mise en place d'une plateforme de simulation parallèle à hautes performances suppose que l'on respecte certains principes pour la conception et l'implémentation de méthodes numériques. Nous énonçons trois d'entre eux ci-dessous.

Principe n°1 : éviter les tests conditionnels

Nous éviterons autant que possible le recours à des tests conditionnels - en particulier lorsque le résultat du test ne peut être prédit de manière fiable [1] - dans le corps des fonctions de calcul. Ceux-ci peuvent en effet altérer notablement les performances d'un code de simulation.

- D'une part, la présence de tests conditionnels empêche en général le compilateur d'effectuer certaines optimisations. L'utilisation de tests dans une boucle peut notamment empêcher sa vectorisation et ainsi considérablement dégrader les performances, l'exécution de boucles vectorisées étant bien plus rapide que celle de boucles non vectorisées.
- D'autre part, les tests conditionnels sont très peu performants sur des architectures de type GPGPU, les cartes accélératrices ne disposant pas de mécanisme de prédiction de branchement.

[1] Les processeurs modernes s'appuyent sur un mécanisme appelé prédicteur de branchement qui vise à prédire le résultat des tests conditionnels et poursuit le lancement d'instructions dans la branche la plus probable. Par conséquent, un test sur un paramètre d'exécution constant peut être effectué très rapidement. Au contraire, les tests portant sur des variables dont la valeur ne peut être prédite sont effectués de manière peu efficace et sont donc à éviter autant que possible.

Il conviendra par conséquent d'éviter le recours à des tests conditionnels aussi bien dans la conception des schémas numériques que dans leur implémentation.

Principe n°2 : s'assurer de la proximité des données (algorithmes dits *cache-oblivious*)

Nous serons attentif aux problématiques de localité spatiale et temporelle des données dans le développement de méthodes numériques afin de maximiser la réutilisation des données présentes dans le cache du processeur, l'accès à une donnée située en cache étant environ 100 fois plus rapide que l'accès à une donnée stockée en mémoire centrale (RAM). En pratique, nous veillerons donc à ce qu'un maximum d'opérations flottantes soient effectuées sur des données proches les unes des autres, aussi bien en mémoire (contiguité en mémoire) qu'en temps.

Remarque 1. Un algorithme tirant parti des principes de localité spatiale et temporelle est dit *cache-oblivious*.

Principe n°3 : limiter le nombre de phases de communication en contexte parallélisé

Nous utilisons dans cette étude une méthode de parallélisation dite de décomposition de domaine. Dans ce contexte, les données sont distribuées sur plusieurs processus : chacun d'entre eux n'a donc la connaissance que d'une partie de l'information du maillage. Or, il est bien connu qu'un grand nombre de méthodes numériques s'appuie sur des supports de mailles (*stencils*) plus ou moins larges, notamment les méthodes d'interpolation. Ceci implique la présence de mailles dites fantômes autour du domaine physique mais également du sous-domaine affecté à chaque processus qu'il s'agit de remplir préalablement à toute opération sur une maille faisant appel à ses voisines. En parallèle, le remplissage des couches de mailles fantômes partagées par deux processus se traduit par des phases communications qui sont susceptibles de réduire l'efficacité parallèle. Il convient donc de limiter autant que possible le nombre de phases de communications en pratique. Cette contrainte pénalise d'office les méthodes d'intégration temporelle multi-pas (de type Runge-Kutta par exemple) : celles-ci nécessitent en effet la réalisation de plusieurs phases de communication par pas de temps, ce qui peut être considéré comme très pénalisant pour un solveur massivement parallèle.

Remarque 2. Les méthodes d'intégration temporelle multi-pas sont non seulement coûteuses en termes de phases de communication mais également en termes d'appels à l'équation d'état. Or, dans le cas d'équations d'état tabulées (*i.e.* pour laquelle il n'existe pas de relation de fermeture analytique), ce type d'appel est particulièrement onéreux.

Plan de l'étude

I. Établissement du modèle

L'évolution des fonctions de distribution des espèces électroniques et ioniques est régie par l'équation de Vlasov avec terme source de type colisionnel. Cette représentation cinétique du problème considéré constitue le point de départ de notre étude. Le chapitre 1 détaille en premier lieu l'obtention d'un jeu d'équations d'évolution pour le fluide moyen par le calcul des trois premiers moments de l'équation de Vlasov pour chaque espèce. Ces équations fluides sont ensuite combinées aux équations de Maxwell que satisfont les champs électrique et magnétique. Nous obtenons finalement une équation d'évolution pour le seul champ magnétique en négligeant le courant de déplacement (approximation MHD). Le système d'équations ainsi établi est complété par des relations de fermeture de type Braginskii [11] et un modèle physique de Decoster [28] définissant les tenseurs thermoélectrique, de conductivité, et de résistivité. Ce système comporte des opérateurs mathématiques de nature différente :

Sous-système	Opérateur mathématique	Schéma temporel
MHD idéale à deux températures	hyperbolique	explicite
conduction thermique	parabolique	semi-implicite
conduction résistive	parabolique	semi-implicite
termes sources	termes sources	explicite

TAB. 1 – Récapitulatif des différents sous-systèmes considérés dans cette étude.

- un opérateur de type hyperbolique qui se trouve être l'opérateur de la MHD idéale ;
- deux opérateurs de type paraboliques régissant respectivement les phénomènes de conduction thermique et résistive ;
- un opérateur de type terme source représentant les éventuels termes de champ magnétique auto-généré ainsi que les contributions des effets Hall et Nernst.

Nous proposons dans le chapitre 2 une stratégie de résolution consistant à séparer le système d'équations initial en plusieurs sous-systèmes (à l'aide d'une méthode de splitting d'opérateur) selon la nature de l'opérateur mathématique sous-jacent. Le principal avantage de cette approche est qu'elle permet la construction de méthodes numériques adaptées à chaque opérateur et que l'étape de validation s'en trouve simplifiée puisqu'elle peut alors être effectuée sous-système par sous-système. Elle permet également d'adapter la discrétisation temporelle à chaque opérateur. Nous verrons que les schémas numériques proposés pour les équations de la MHD idéale s'appuyent sur une discrétisation temporelle explicite impliquant une condition de stabilité du type $\Delta t \leq \lambda \Delta x$. Or, la résolution explicite des termes de conduction induirait une contrainte bien plus importante sur le pas de temps du type $\Delta t \leq \lambda \Delta x^2$. Ceux-ci seront donc traités de manière semi-implicite. Enfin, nous proposerons une approche explicite pour le traitement des termes sources, celle-ci s'appuyant sur l'hypothèse que la contribution des termes sources est suffisamment faible pour ne pas violer la condition de stabilité des schémas hyperboliques. Les différents sous-systèmes considérés ainsi que la discrétisation temporelle retenue pour chacun d'entre eux sont résumés dans le tableau 1.

II. Construction de méthodes numériques d'ordre élevé pour les équations de l'hydrodynamique et de la MHD idéale

La seconde partie de notre étude est consacrée à la construction d'une nouvelle classe de schémas numériques d'ordre élevé en espace et en temps en régime non-linéaire sur grille cartésienne régulière pour les équations de l'hydrodynamique et de la MHD idéale en deux dimensions d'espace en géométries plane et axisymétrique. Ces schémas numériques, dits GoHy (pour *Godunov hybride*) et construits dans le formalisme Lagrange + projection, présentent la particularité d'avoir été conçus en respectant les principes évoqués précédemment (cf. page 3), ceci dans le but de tirer parti de la puissance des calculateurs modernes.

Schémas GoHy pour l'hydrodynamique plane

Nous détaillons tout d'abord dans le chapitre 4 la construction des schémas GoHy pour les équations de l'hydrodynamique plane et nous nous concentrons en premier lieu sur le cas monodimensionnel en section 4.1. Les équations d'Euler monodimensionnelles sont tout d'abord réécrites en coordonnées lagrangiennes puis sont discrétisées par une méthode de type volumes finis. Nous proposons alors une méthodologie reposant

sur la procédure dite de Cauchy-Kovalevskaya permettant de calculer des valeurs approchées des flux lagrangiens à l'ordre élevé et ainsi de construire un schéma volumes finis d'ordre élevé. Le recours à la procédure de Cauchy-Kovalevskaya se révélant particulièrement complexe à l'ordre élevé, nous décrivons également une technique permettant de l'appliquer automatiquement et de générer le code source correspondant à l'aide d'un logiciel de calcul formel (Maple en l'occurrence). Cette première étape nous permet de déterminer les valeurs moyennes des variables conservatives lagrangiennes sur un maillage non-uniforme. Celles-ci sont ensuite projetées sur le maillage régulier initial à l'aide d'une méthode de reconstruction polynomiale d'ordre élevé. Les schémas numériques Lagrange + projection ainsi obtenus sont en théorie particulièrement efficaces. En effet, ils n'ont recours à aucun test conditionnel (en accord avec le principe n°1), reposent sur un schéma en temps direct ne nécessitant qu'un seul appel à l'équation d'état et aux conditions de bord par itération (en accord avec le principe n°3) et utilisent uniquement des méthodes d'interpolation centrées peu coûteuses.

Nous proposons ensuite en section 4.2 d'étendre les schémas GoHy aux géométries bidimensionnelles en combinant le schéma monodimensionnel décrit précédemment avec des méthodes de splitting directionnel d'ordre élevé dont la précision est supérieure au classique splitting de Strang [77] d'ordre 2. Les techniques de splitting directionnel consistent à résoudre le système d'équations bidimensionnel considéré par plusieurs itérations d'un schéma monodimensionnel selon les directions horizontale et verticale avec un pas de temps pondéré de manière adéquate. L'ordre de la méthode de splitting dépend du nombre d'itérations et du choix des coefficients de pondération. Nous proposons dans cette étude plusieurs jeux de coefficients de l'ordre 3 à l'ordre 6. Ceux-ci sont pour l'essentiel tirés de la littérature sur les intégrateurs symplectiques [36, 61, 13, 95]. Notre contribution dans ce domaine consiste à proposer des séquences de splitting dites optimales au sens où elles permettent de maximiser le pas de temps.

Nous montrons par ailleurs qu'une simple combinaison d'un schéma monodimensionnel et d'une méthode de splitting directionnel d'ordre élevé ne permet pas la construction d'un schéma multidimensionnel d'ordre élevé. Il est en effet nécessaire d'effectuer une étape supplémentaire dite de reconstruction transverse permettant le passage du contexte monodimensionnel au contexte multidimensionnel. Nous proposons en section 4.2.2 une telle méthode opérant sur les variables conservatives qui se révèle contraignante en termes de conservativité. Nous indiquons par conséquent comment cette méthode de reconstruction peut être appliquée aux flux, ce qui permet de s'affranchir de tout problème de préservation de la conservativité.

La stratégie à directions alternées que nous proposons présente un avantage significatif par rapport aux méthodes classiques en termes de performances. En effet, comme nous le verrons en introduction de la section 4.2, les techniques de splitting directionnel permettent la mise en œuvre d'une implémentation opérant majoritairement sur des données contigües en mémoire, respectant ainsi le principe n°2 énoncé page 4.

La section 4.3 est consacrée au traitement des deux principales difficultés auxquelles sont confrontés de tels schémas d'ordre élevé en pratique. D'une part, comme la majorité des méthodes d'ordre élevé, les schémas que nous avons développés sont sujets à des phénomènes d'oscillations de Gibbs au voisinage des discontinuités qui sont susceptibles d'altérer la qualité de la solution numérique, voire de produire des résultats erronés. D'autre part, les schémas GoHy provoquent en raison de leur approche par directions alternées des pertes de symétrie sur certains cas pratiques présentant en théorie une symétrie de révolution. Ces deux problèmes sont traités par des techniques de viscosité artificielle : nous utilisons un modèle d'hyperviscosité d'ordre élevé inspiré de la littérature sur la simulation grandes échelles [21, 20] afin réduire les phénomènes d'oscillations sans réduire l'ordre des schémas tandis que la préservation de la symétrie est assurée par un modèle de viscosité artificielle d'ordre 1 combiné à un filtre évitant l'introduction excessive de dissipation numérique dans les zones présentant des chocs. Ces techniques présentent l'avantage par rapport

aux méthodes classiques (limiteurs, techniques d'interpolation à support adaptatif de type ENO) d'éviter le recours à des tests conditionnels (elles respectent donc le principe n°1 énoncé page 3) et se révèlent donc peu coûteuses. Notons cependant que les méthodes de viscosité artificielle reposent sur un certain nombre de paramètres qui doivent être ajustés pour chaque problème, rendant parfois leur mise en œuvre fastidieuse.

Afin de mesurer la capacité effective des schémas GoHy à tirer parti des calculateurs modernes, nous proposons en section 4.4 des mesures de performance séquentielle et d'efficacité parallèle à l'aide des outils décrits page 2. Celles-ci se révèlent conformes aux attentes. Elles montrent en effet que les schémas GoHy pour les équations de l'hydrodynamique sont capables d'exploiter 20 à 30% de la puissance crête des processeurs actuels tandis que l'efficacité parallèle de notre implémentation atteint environ 95% sur 256 processus.

Nous concluons le chapitre 4 par des résultats numériques obtenus sur des cas tests tirés de la littérature. Nous considérons tout d'abord des solutions régulières [53, 94] pour lesquelles nous réalisons une analyse de convergence. Un même problème est résolu sur des maillages de tailles différentes et l'erreur entre les solutions exacte et approchée est mesurée pour chaque finesse de maillage à l'aide d'une norme L^1 en espace et en temps. Ces résultats permettent alors de calculer l'ordre expérimental de convergence qui se révèle conforme à l'ordre théorique dans tous les cas que nous avons considérés. Nous effectuons par ailleurs des tests sur des problèmes non réguliers présentant des chocs et des discontinuités. Ceux-ci permettent, d'une part, de mettre en valeur les effets positifs des méthodes de viscosité artificielle sur la qualité des solutions numériques et, d'autre part, d'évaluer la robustesse des schémas GoHy. Cette dernière se révèle tout à fait satisfaisante en pratique.

Les schémas GoHy pour l'hydrodynamique ont fait l'objet d'une publication aux Comptes-Rendus de l'Académie des Sciences de Paris [32].

Extension des schémas GoHy à la MHD idéale

Le chapitre 5 est consacré à l'application des schémas GoHy au système de la MHD idéale. Celle-ci repose globalement sur les mêmes algorithmes que dans le cas de l'hydrodynamique. Une difficulté survient toutefois dans le cas multidimensionnel. Dans ce contexte, le champ magnétique doit en effet satisfaire la contrainte de divergence suivante :

$$\nabla \cdot \mathbf{B} = 0,$$

sans quoi la résolution numérique est susceptible de mener à des résultats erronés [10]. Or, le respect de cette contrainte n'est pas automatiquement garanti au niveau discret : il est par conséquent nécessaire de l'imposer numériquement. Nous présentons en section 5.2 plusieurs techniques tirées de la littérature répondant à cette problématique [10, 35, 88, 1, 29] et étudions leur capacité à être couplée avec l'approche à directions alternées que nous avons adoptée. Il se trouve que la technique dite de nettoyage hyperbolique de la divergence proposée par Dedner et al. [29] semble être la mieux adaptée. Nous décrivons donc comment elle peut être combinée aux schémas GoHy sans perte d'ordre.

Comme dans le cas de l'hydrodynamique, nous mesurons ensuite la performance séquentielle et l'efficacité parallèle des schémas GoHy pour la MHD idéale à l'aide des outils décrits page 2. Les résultats sont très satisfaisants : ils montrent en effet que les schémas GoHy sont capables de tirer parti de près de 40% de la puissance crête des processeurs actuels à l'ordre 4 tandis que l'efficacité parallèle faible estimée est supérieure à 95% pour 256 processus. Nous concluons le chapitre 5 par des résultats numériques sur des cas tests réguliers ou non tirés de la littérature. Ceux-ci montrent d'une part que l'ordre de convergence

expérimental des schémas GoHy correspond à l'ordre théorique et font état, d'autre part, d'une robustesse tout à fait satisfaisante.

Cette partie a fait l'objet d'une présentation à la conférence *Numerical Models for Controlled Fusion* qui s'est déroulée au mois d'avril 2009 sur l'île de Porquerolles. Celle-ci a débouché sur une publication dans les actes de cette conférence [93].

Cas des géométries axisymmétriques

Le chapitre 6 détaille l'extension des schémas GoHy aux géométries axisymétriques dans le cas de la MHD idéale. Deux principales difficultés surviennent dans ce contexte. D'une part, le calcul automatique des flux lagrangiens avec le logiciel Maple se révèle délicat le long de l'axe $r = 0$ en raison de la présence de termes en $1/r$. Ce problème est traité en modifiant quelque peu la procédure proposée en chapitre 4 pour les géométries planes. D'autre part, le cas axisymétrique se caractérise par la présence d'un terme non-conservatif qu'il convient de discrétiser de manière adéquate afin de construire des schémas d'ordre élevé. Nous proposons un algorithme d'approximation analogue à celui utilisé pour la construction des flux lagrangiens d'ordre élevé et l'illustrons par un exemple à l'ordre 2. En outre, nous détaillons également la mise en œuvre des techniques de viscosité artificielle et de nettoyage hyperbolique de la divergence dans le cas des géométries axisymétriques.

Nous concluons le chapitre 6 par des résultats numériques sur des cas tests réguliers [53, 70] et non réguliers tirés de la littérature. Une fois de plus, ceux-ci montrent que l'ordre expérimental de convergence correspond à l'ordre théorique et fait état de propriétés de robustesse satisfaisantes.

III. Vers des simulations d'implosion de capsule FCI

Prise en compte des opérateurs de diffusion et des termes sources

La résolution du modèle complet établi dans le chapitre 1 suppose que les schémas GoHy pour la MHD idéale soient couplés à des schémas de diffusion - permettant la prise en compte des opérateurs de conduction thermique et résistive - et à une discrétisation des termes sources, en particulier des termes régissant l'apparition de champ magnétique auto-généré. Ceci fait l'objet du chapitre 7. Dans cette partie de notre étude, les problématiques d'ordre élevé sont écartées : l'objectif consiste ici simplement à permettre la résolution du modèle complet par la mise en œuvre de méthodes numériques de type volumes finis tout à fait classiques. Une attention particulière est toutefois portée aux problématiques de performances et de parallélisme, notamment dans le cadre de la discrétisation semi-implicite des termes de conduction. Celle-ci suppose en effet la résolution de systèmes linéaires creux dont l'implémentation et l'optimisation en contexte parallèle sont non triviales. En pratique, nous résolvons ces systèmes par des méthodes de gradient conjugué ou de bigradient conjugué stabilisé [74] préconditionnés dont l'implémentation repose sur la bibliothèque Intel Math Kernel Library [47]. Nous proposons enfin des résultats numériques sur des problèmes de diffusion tirés de la littérature.

Exploitation du solveur développé sur un cas test d'implosion de coquille FCI

Le chapitre 8, dernière partie de cette étude, est consacré à l'exploitation du solveur que nous avons développé sur un cas test tiré de [80] dont la condition initiale correspond à l'état d'une coquille FCI en début de phase de décélération. Celle-ci a été obtenue en interpolant des profils de densité, vitesse et pression fournis par un code de FCI monodimensionnel [12], celui-ci ayant été utilisé afin de simuler l'implosion d'une coquille FCI jusqu'au début de la phase de décélération. Cette approche nous permet

de réaliser des simulations réalistes sans avoir à prendre en compte les phénomènes physiques intervenant dans le processus d'implosion complet, en particulier l'absorption laser. En effet, au début de la phase de décélération, le dépôt d'énergie laser est achevé et la cible est déjà mise en vitesse. Dans ce contexte, nous simulons la présence d'un écart à la sphéricité en introduisant au niveau de l'interface gaz/coquille une perturbation numérique par des polynômes de Legendre. Nous proposons enfin des résultats numériques avec des perturbations de différentes amplitudes afin d'analyser les caractéristiques du champ magnétique auto-généré et ses effets sur le plasma.

Une partie de cette étude a été menée lors de l'école d'été du CEMRACS[2] 2010 et a fait l'objet d'une publication dans les actes du CEMRACS [92].

Quelques notations

Avant de débuter cette étude, nous définissons quelques notations. Nous considérons dans la suite des domaines bidimensionnels $\Omega = [a_x; b_x] \times [a_y; b_y]$ sur lesquels nous définissons un maillage structuré de $N_x \times N_y$ mailles :

$$\Omega = \bigcup_{1 \leq i \leq N_x} \bigcup_{1 \leq j \leq N_y} [x_{i-\frac{1}{2}}; x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}; y_{j+\frac{1}{2}}] \quad \text{où} \quad \begin{cases} x_{i+\frac{1}{2}} = a_x + i\Delta x, \\ y_{j+\frac{1}{2}} = a_y + j\Delta y, \end{cases}$$

Δx et Δy désignant le pas (uniforme) du maillage selon les directions x et y :

$$\Delta x = \frac{b_x - a_x}{N_x}, \ \Delta y = \frac{b_y - a_y}{N_y}.$$

Les faces verticales des mailles sont par conséquent repérés par $x_{i+\frac{1}{2}}$ tandis que les faces horizontales sont notées $y_{j+\frac{1}{2}}$. Enfin, le point $(x_i; y_j)$ désigne le centre géométrique de la maille (i, j) :

$$x_i = \frac{x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}}{2}, \ y_j = \frac{y_{j-\frac{1}{2}} + y_{j+\frac{1}{2}}}{2}.$$

[2]Centre d'Été Mathématique de Recherche Avancée en Calcul Scientifique.

Introduction

Physical context

Inertial confinement fusion (ICF) is a process where nuclear fusion reactions are initiated by heating and compressing a fuel target - generally a capsule made of an external layer of cryogenic deuterium-tritium (DT) which encloses a volume of DT gas. In order to compress and heat the ICF capsule, energy is delivered to the outer layer of the target and vaporizes it into a plasma envelope. There are two ways to perform this energy delivery. In the case of direct drive ICF, it is done by irradiating the target with several high-energy laser beams. In the case of indirect drive ICF, the target is placed into a metal cylinder (called the hohlraum) which is irradiated by the laser beams instead of the target itself, causing the emission of X-rays. These are then absorbed by the capsule's surface and create the desired confinement conditions.

This study concerns direct drive inertial confinement fusion. In this context, it is well-known that some difficulties may be encountered during the implosion phase [64] which follows the laser irradiation due to slight sphericity deviations (shell rugosity and/or non-uniformity of the laser irradiation) on the outer surface of the shell. These sphericity deviations generate initial perturbations which are then amplified during the implosion process. Richtmyer-Meshkov and/or Rayleigh-Taylor instabilities develop, leading to the apperance of self-generated magnetic field. Indeed, as we will see in this study, the density and pressure gradients which are theoretically aligned are not colinear anymore in the vicinity of instabilities and thus cause the development of so-called self-generated magnetic field (they are said to be self-generated because they are not the consequence of an exterior field) whose effects may be significant. In particular, these magnetic fields reduce thermal conductivity. Depending on their intensity, they may also modify the temperature distribution in the vicinity of the hot spot and may thus modify the ignition conditions [42]. It therefore seems important to follow the evolution of magnetic field components additionally to the evolution of hydrodynamical quantities in simulation codes.

Objectives

This study is devoted to the development of numerical methods that take self-generated magnetic field terms into account and allow realistic simulations of the ICF implosion processes. To that end, we first derive a two-temperature resistive MHD model that represents in a sufficiently exhaustive way the physics of the considered problem. Then we propose accurate and efficient numerical methods for solving the evolution equations that have been derived. Particular attention will be paid to the implementation of these numerical schemes and, in a more general way, to high performance computing (HPC) aspects (performance optimization, scalability) so that the solver that we aim to develop is able to take advantage of modern massively parallel supercomputers. The last part of this study is devoted to the exploitation of the implemented simulation code, more precisely to the study of self-generated magnetic field terms and their effects on the basis of preliminary numerical results.

Numerical context

The implosion of capsules in the context of direct drive ICF theoretically is a spherically symmetric process, which could help to reduce the number of space dimensions in numerical simulations. Considering sphericity deviations on the surface of the shell nevertheless supposes to deal with more complex geometries. The most general approach would consist in solving the considered problem in the three-dimensional case but it would obviously lead to extremely expensive simulations. We therefore restrict this study to the case of two-dimensional axisymmetric geometries.

Moreover, we have chosen to build numerical schemes on structured meshes using the finite volume method, this framework being well-suited to the above-mentioned high performance constraint.

High performance computing aspects

Frontiers between the domains of numerical simulation and high performance computing (HPC) are getting thinner and thinner. On the one hand, in order to provide reliable numerical results, physical models which are built for simulation purposes become more and more complex. On the other hand, supercomputer architectures evolve towards massively parallel configurations with an increasing number of cores and a consequently decreasing memory space per core. These trends impose several constraints for building and implementing numerical methods: these have to be both efficient (in order to allow the resolution of complex multi-physics problems) and scalable on massively parallel computers. We propose in this study an innovative approach in the sense that the decision of implementing a certain technique has been mostly motivated by the above-mentioned constraints. These choices of course have to be justified on the basis of practical evidences. To that end, we use three performance measurement tools that we describe in the sequel.

Performance measurement tools

We perform in this study a complete performance analysis of the solver that we have developed in order to determine if the technical choices that have been made provide concrete benefits. This analysis relies on three measurements: the grind time, the number of floating point operations that are performed per second and the parallel efficiency. Altogether, these measurements give a reliable evaluation of performance and scalability.

The grind time

The grind time is defined as the time that is necessary to perform an iteration of a given numerical scheme in a single cell:

$$\text{grind time} = \frac{\text{restitution time}}{\text{number of cells} \times \text{number of iterations}}.$$

It is therefore the critical performance indicator since the restitution time entirely depends on it for a given problem and mesh size. For information, it is generally admitted that the grind time has to be under 10 microseconds per cell and per iteration for an industrial code running on a single core of a recent processor.

Number of floating point operations per second

The number of floating point operations performed per second (FLOPS) in another interesting performance indicator that we measure in this study. Floating point operations are counted during execution

following a protocol described in section 4.4.1. The result is then compared to the used processor's peak performance. It indicates whether the processor is well exploited and gives an evaluation of a code's performance optimization margin. For information, scientific codes generally exploit about 10% of the available peak power.

Parallel efficiency

The last measurement that we consider in this study is the parallel efficiency which evaluates the ability of a code to take advantage of parallel computers. More precisely, it is defined as the ratio of the sequential restitution time to the parallel one for a given problem. Nevertheless, this measurement, called *strong parallel efficiency*, may sometimes be difficult to obtain. Indeed, when dealing with a huge number of processes, problems that have to be considered are so large that they lead to prohibitive restitution times in the sequential context. Memory available for one core may also reveal unsufficient. As is classical in such a case, we therefore adopt a different approach which consists in evaluating the ratio of the sequential restitution time on a given problem to the execution time on N processes with a N times finer mesh. This measurement is called *weak parallel efficiency*.

A few principles for designing high performance and scalable numerical methods

In order to develop a parallel and high-performance simulation platform, one has to follow a few rules for building and implementing numerical methods. Three of these principles are given below.

Principle n°1 : avoiding conditional tests

Conditional tests have to be avoided as much as possible in computation loops, in particular when the test result cannot be predicted efficiently [3]. These may indeed have significantly degrade performances.

- On the one hand, the presence of conditional tests may prevent the compiler from performing some optimizations. For example, resorting to tests in a loop may prevent its vectorization and thus slow down execution.
- On the other hand, GPU architectures are unable to deal efficiently with conditional tests since they do not have branch predictor circuits.

Conditional tests should therefore be avoided as much as possible during both development and implementation phases of numerical schemes.

Principle n°2 : paying attention to data locality (cache-oblivious algorithms)

Particular attention has to be paid to the principles of spatial and temporal data locality for building and implementing numerical schemes in order to reuse cached data as much as possible. Accessing to a cached datum is indeed about 100 times faster than accessing to a datum stored in the central memory (RAM), which motivates the principle of reusing cached data. Numerical schemes should therefore mostly operate on data that are close the ones from the others in both memory (data contiguity) and time.

Remark 1. An algorithm that exploits the principles of spatial and temporal locality is said to be cache-oblivious.

[3] Modern processors use so-called branch predictor circuits which aims at guessing the result of if-then-else structures and automatically carry on the execution of instructions in the most likely branch. Conditional tests can therefore be performed very quickly on constant execution parameters whereas they noticeably decrease performance when they concern a variable whose value cannot be predicted.

Principle n°3 : limiting the number of communication phases between processes in parallel context

The solver that we have developed has been parallelized using a domain decomposition method. In this context, data are distributed among several processes : only a part of the computational domain is therefore stored in memory on each process. But several numerical methods - like interpolation methods - have to be performed on stencils of cells. This supposes the presence of ghost cells layers not only around the physical domain but also around the subdomain that is assigned to each process. These have to be filled prior to every function that operates on stencils of cells. In parallel context, filling ghost cells layers supposes to perform a communication phase in order to share data between processes but these are costly and may reduce the parallel efficiency. Communication phases therefore have to be avoided as much as possible. This constraint automatically discards multistep integration methods (like Runge-Kutta methods) since these require several communication phases per time step. They are therefore badly suited to massively parallel computer architectures.

Remark 2. Multistep temporal integration methods are expensive not only in terms of communication phases but also in terms of equation of state calls. Yet, in the case of tabulated equations of state (*i.e.* when closure relations are not analytical), equation of state calls are particularly costly.

Outline

I. Model derivation

The evolution of the ionic and electronic species distribution functions is governed by the Vlasov equation with a collisional source term. This kinetic representation of the problem is the starting point of our study. We firstly describe in chapter 1 the derivation of evolution equations for the mean fluid by calculating the three first moments of the Vlasov equation for each species. These are then combined with the Maxwell equations which are satisfied by the electric and magnetic fields. By neglecting the displacement current (MHD approximation), we get an evolution equation for the only magnetic field. The resulting system is then completed by Braginskii closure relations [11] and Decoster's collisions model [28] which provide expressions for the thermoelectric, conductivity and resistivity tensors. It involves mathematical operators of different natures:

- an hyperbolic operator which actually is the ideal MHD operator;
- two parabolic operators that respectively govern thermal and resistive conduction;
- a source term operator that takes the self-generated magnetic field terms into account as well as the contribution of Hall and Nernst effects.

We propose in chapter 2 a resolution strategy that consists in splitting the derived system of equations into several subsystems according to the nature of the underlying mathematical operator. The main advantage of this approach is that it allows to build numerical methods that are well-suited to each operator and it simplifies the validation step since it can be performed separately for each term. The time discretization method can also be appropriately chosen. We will see that numerical schemes that have been proposed for the ideal MHD equations rely on an explicit time discretization that implies a $\Delta t = \lambda \Delta x$ type stability condition. Yet, treating diffusion terms explicitly would lead to a far more constraining $\Delta t = \lambda \Delta x^2$ time step constraint. These therefore will be discretized semi-implicitly. We finally propose an explicit approach for dealing with source terms. We indeed assume that the source terms' contribution is weak enough not to

Subsystem	Mathematical operator	Time scheme
two-temperature ideal MHD	hyperbolic	explicit
thermal conduction	parabolic	semi-implicit
resistive conduction	parabolic	semi-implicit
source terms	source terms	explicit

Table 2: Subsystems resulting from the splitting of the complete model considered in this study.

fulfill the hyperbolic schemes' stability condition. The considered subsystems and the corresponding time discretization are summarized in Table 2.

II. Construction of high-order numerical methods for hydrodynamics and ideal MHD equations

The second part of this study is devoted to the construction of a new class of numerical schemes which are high-order accurate in both space and time in the non-linear regime on structured grids. These so-called GoHy (for Godunov hybrid) schemes have been built in the Lagrange-remap formalism for hydrodynamics and ideal MHD equations in both planar and axisymmetric geometries. Their most innovative feature is that they have been developed with the above-mentioned efficiency principles in mind so that they are able to take advantage of modern computer architectures.

GoHy schemes for planar hydrodynamics

We describe in chapter 4 the construction of GoHy schemes for planar hydrodynamics equations and being with the one-dimensional case (see section 4.1). The one-dimensional Euler equations are firstly written in Lagrangian coordinates and discretized using a finite volume method. We then propose a methodology for building high-order accurate approximate Lagrangian fluxes based on the so-called Cauchy-Kovalevskaya procedure that thus allows to build high-order accurate numerical schemes. Applying the Cauchy-Kovalevskaya (C-K) procedure reveals particularly complex when dealing with higher-orders. We therefore describe a technique based on the Maple software that automatically applies this C-K procedure and generates the corresponding source code. This first step allows us to determine updated values of the Lagrangian conservative variables on a non-uniform mesh. These are then remapped onto the initial regular grid using a high-order accurate polynomial reconstruction method. The resulting Lagrange-remap schemes are theoretically particularly efficient. Indeed, they do not involve conditional tests (in accord with principle n°1), rely on a one-step temporal scheme that requires only one equation of state and boundary condition call per iteration (in accord with principle n°3) and only resort to centered (and thus cheap) interpolation methods.

We extend GoHy schemes to two-dimensional geometries in section 4.2 by combining the one-dimensional scheme with high-order dimensional splitting methods whose accuracy is greater than the classical Strang splitting [77]. Dimensional splitting techniques consist in solving the considered two-dimensional system of equations by performing several iterations of the one-dimensional scheme along the horizontal and vertical directions with an appropriately weighted time step. The splitting method's accuracy depends on both the number of iterations and the time step weights. We propose in this study several sets of coefficients from

third to sixth-order accuracy. Most of these are taken from the literature on symplectic integrators [36, 61, 13, 95]. Our contribution here consists in determining optimal splitting sequences in the sense that the time step is maximized.

Moreover, we show that a trivial combination of a high-order one-dimensional scheme and a high-order dimensional splitting sequence does not allow to build a high-order multidimensional scheme. Indeed, an additional reconstruction step has to be performed in order to switch between the one-dimensional and multidimensional contexts. We propose in section 4.2.2 such a reconstruction method that operates on conservative variables but it reveals constraining in terms of conservativity preservation. We then show that this issue can be bypassed by moving the reconstruction step on fluxes.

The dimensionally split approach that we propose presents a major advantage over classical unsplit methods in terms of performance. Indeed, as we will see in section 4.2, dimensional splitting techniques allow cache-oblivious implementations that mostly operate on contiguous data and thus respect the second principle stated page 13.

Section 4.3 is devoted to the treatment of the two main issues that such high-order schemes may practically encounter. On the one hand, like most high-order methods, GoHy schemes are subject to the phenomenon of Gibbs oscillations in the vicinity of discontinuities which may degrade the numerical solutions quality or even produce unphysical results. On the other hand, due to their dimensionally split approach, GoHy schemes sometimes cause symmetry losses in practical cases that theoretically present a rotational symmetry. These two difficulties are treated thanks to artificial viscosity techniques. Oscillations are controlled using an hyperviscosity model inspired from the large eddy simulation (LES) literature [21, 20] whereas symmetry preservation is improved by a first-order artificial viscosity model which is combined with a filtering method in order not to degrade the shock-capturing features of GoHy schemes. These techniques present the advantage of avoiding conditional tests (in accord with the first principle stated page 13) unlike classical slop limiting or essentially non-oscillatory (ENO) methods, which makes them cheap in terms of computational cost. The use of artificial viscosity techniques nevertheless sometimes reveals tedious in practice since these rely on several parameters that have to be adjusted for each problem.

In order to assess the ability of GoHy schemes to take advantage of modern computer architectures, we provide in section 4.4 sequential performance and parallel efficiency figures that have been obtained using indicators described page 12. These figures match expectations. They indeed show that hydrodynamical GoHy schemes are able to exploit 20 to 30% of recent processors's peak performance whereas our implementation's parallel efficiency reaches 95% on 256 processes.

We conclude chapter 4 with numerical results obtained on test problems taken from the related literature. We first consider smooth solutions [53, 94] and carry out a convergence analysis: the same test problem is solved on different mesh sizes and the error between the exact and approximate solutions is measured for each mesh using a L^1 norm in space and time. These results then allow to compute the experimental order of convergence which actually matches expectations in all the considered situations. We also run computations on non-smooth test problems in order to illustrate the benefits of artificial viscosity techniques and to evaluate the robustness of GoHy schemes which reveals satisfying.

GoHy schemes for hydrodynamics have been published as part of the *Comptes-Rendus de l'Académie des Sciences de Paris* [32].

Extension of GoHy schemes to the ideal MHD equations

Chapter 5 is devoted to the description of GoHy schemes for the ideal MHD equations. These generally rely on the same methodology than in the case of hydrodynamics. An additional difficulty nevertheless arises in the multidimensional case. In this context, the magnetic field indeed has to satisfy the following divergence constraint:

$$\nabla \cdot \mathbf{B} = 0,$$

otherwise, the numerical resolution of the ideal MHD equations may lead to unphysical results [10]. This constraint is not automatically satisfied at the discrete level: it therefore has to be enforced numerically. We present in section 5.2 several well-known techniques taken from the literature [10, 35, 88, 1, 29] for imposing $\nabla \cdot \mathbf{B}$ in numerical simulations and study their ability to be coupled with the dimensionally split approach of GoHy schemes. It turns out that the so-called hyperbolic divergence cleaning technique proposed by Dedner *et al.* [29] seems to be the best suited. We therefore describe how it can be combined with GoHy schemes.

As in the case of hydrodynamics, we provide sequential performance and parallel efficiency figures for the ideal MHD GoHy schemes using the measurements tools described page 12. Results are very satisfying. They indeed show that GoHy schemes are able to exploit up to 40% of modern processors's peak performance at fourth-order accuracy while the parallel efficiency is over 95% on 256 processes. We then conclude chapter 5 with numerical results on both smooth and non-smooth test problems taken from the related literature. These show that the experimental order of convergence matches the theoretical order and exhibit satisfying robustness features.

GoHy schemes for the ideal MHD equations have been the subject of an oral presentation at the *Numerical Models for Controlled Fusion* (NMCF) conference which took place in April 2009 on the island of Porquerolles. They also have been published as part of the NMCF conference proceedings [93].

Extension to axisymmetric geometries

We detail in chapter 6 the extension of GoHy schemes to axisymmetric geometries in the case of ideal MHD equations. One encounters two main difficulties in this context. Firstly, the Maple-based automated computation of Lagrangian fluxes reveals tricky along the $r = 0$ axis due to the presence of $1/r$ terms. This issue is bypassed by slightly modifying the methodology proposed in chapter 4 for planar geometries. Secondly, the axisymmetric case is characterized by the presence of a non-conservative term in both hydrodynamics and ideal MHD equations which has to be appropriately discretized in order to achieve high-order accuracy. We propose an approximation procedure that is analog to the one used for building high-order accurate Lagrangian fluxes and illustrate its application in the second-order case. Moreover, we detail the extension of artificial viscosity and hyperbolic divergence cleaning techniques in the axisymmetric case.

We then conclude chapter 6 with numerical results on smooth [53, 70] and non-smooth test problems taken from the literature. Once again, these show that the experimental order of convergence matches the theoretical one and put satisfying robustness features forward.

III. Towards ICF shell implosion simulations

Discretization of the diffusion and source terms operators

The resolution of the complete model that has been derived in chapter 1 supposes to combine GoHy schemes with diffusion schemes - that take the thermal and resistive conduction terms into account - and a discretization of source terms, in particular the operator that governs magnetic field generation. Chapter 7 is devoted to these additional terms. Note that we do not aim to achieve high-order accuracy in this part of our study: our objective here is simply to allow the resolution of the complete two-temperature resistive MHD model using classical finite volume methods. Particular attention is however paid to performance and parallelism aspects, especially for the semi-implicit discretization of diffusion operators since it implies the resolution of sparse linear systems whose implementation and optimization are non-trivial topics. Practically, these systems are solved using preconditioned conjugate gradient or stabilized biconjugate gradient methods [74] that have been implemented with the help of the Intel Math Kernel Library [47]. We conclude chapter 7 with numerical results obtained on diffusion problems taken from the related literature.

ICF implosion test problem

The last part of this study is devoted to the carrying out of numerical experiments with the solver we have developed on a test problem taken from the literature [80] which corresponds to the state of an ICF capsule at the beginning of the deceleration phase. The initial condition has been obtained by interpolating density, velocity and pressure profiles generated by a one-dimensional ICF code [12] that has been used for simulating the implosion of an ICF capsule up to the beginning of the deceleration phase. This approach allows us to perform realistic ICF implosion simulations without taking into account all physical phenomena involved in such ICF implosion processes - in particular the laser absorption. Indeed, laser beams have already been shut down at the beginning of the deceleration phase and the target has been given an initial velocity. In this context, sphericity deviations are emulated by introducing a numerical perturbation on the gas/shell interface using Legendre polynomials. We finally present simulation results for the perturbed ICF test problem with different initial perturbation amplitudes in order to study self-generated magnetic fields and their effects on the plasma.

A part of this work has been done during the 2010 CEMRACS[4] summer school and has then been published in the CEMRACS proceedings [92].

A few notations

Before beginning this study, we introduce a few notations. We consider in the sequel two-dimensional $\Omega = [a_x; b_x] \times [a_y; b_y]$ domains on which we define structured meshes of $N_x \times N_y$ cells:

$$\Omega = \bigcup_{1\leq i\leq N_x} \bigcup_{1\leq j\leq N_y} [x_{i-\frac{1}{2}}; x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}; y_{j+\frac{1}{2}}] \quad \text{where} \quad \begin{cases} x_{i+\frac{1}{2}} = a_x + i\Delta x, \\ y_{j+\frac{1}{2}} = a_y + j\Delta y, \end{cases}$$

Δx et Δy denoting the (uniform) space step along the x and y directions:

$$\Delta x = \frac{b_x - a_x}{N_x}, \ \Delta y = \frac{b_y - a_y}{N_y}.$$

[4] *Centre d'Été Mathématique de Recherche Avancée en Calcul Scientifique.*

As a matter of fact, vertical and horizontal cell interfaces positions are respectively denoted by $x_{i+\frac{1}{2}}$ and $y_{j+\frac{1}{2}}$. The $(x_i; y_j)$ point denotes the middle of the (i, j) cell:

$$x_i = \frac{x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}}{2},\ y_j = \frac{y_{j-\frac{1}{2}} + y_{j+\frac{1}{2}}}{2}.$$

Part I

A two-temperature resistive MHD model for direct drive ICF

Chapter 1

Model equations

We derive in this chapter the partial differential equations that govern the evolution of both hydrodynamic quantities and magnetic field. To that end, we start from the Vlasov equation satisfied by the distribution functions of both ions and electrons and calculate its three first moments in order to derive fluid evolution equations for the mean fluid. These are then combined with the Maxwell equations so that the magnetic field's contribution is taken into account. The resulting system is finally completed by Braginskii's closure relations [11] and Decoster's collisions model [28].

This study has been done in CGS units. The physical constants that we will use in the sequel and their values in CGS units are summarized in Table 1.1.

Symbol and name		Value in CGS units
c	speed of light	$2.9979 \cdot 10^{10}$
e	elementary charge	$4.8 \cdot 10^{-10}$
k_B	Boltzmann constant	$1.3806 \cdot 10^{-16}$
m_e	mass of the electron	$9.1094 \cdot 10^{-28}$
m_p	mass of the proton	$1836\ m_e$
h	Planck constant	$6.6262 \cdot 10^{-27}$
$\bar{h}$	reduced Planck constant	$(2\pi)^{-1}h$
μ	permeability	4π

Table 1.1: A few physical constants and their values in CGS units.

1.1 A few macroscopic quantities

Let $f_\alpha(\mathbf{x}_\alpha, \mathbf{v}_\alpha, t)$ denote the distribution function of the α species (namely ions and electrons in practice) whose mass and electric charge are respectively given by m_α and Z_α. Its evolution is governed by the

following well-known Vlasov equation in CGS units [28]:

$$\partial_t f_\alpha + \mathbf{v}_\alpha \cdot \nabla f_\alpha + \frac{Z_\alpha e}{m_\alpha}\left(\mathbf{E} + \frac{\mathbf{v}_\alpha}{c} \wedge \mathbf{B}\right) \cdot \frac{\partial f_\alpha}{\partial \mathbf{v}_\alpha} = \sum_\beta C_{\alpha\beta}(f_\alpha, f_\beta), \tag{1.1}$$

where $\mathbf{E}$ is the electric field, $\mathbf{B}$ the magnetic field and $C_{\alpha\beta}$ a collision term between α and β species particles. Hydrodynamic quantities - particle density, momentum and total energy - are defined as the three first moments of the distribution function f_α:

$$\text{density} \qquad n_\alpha = \int f_\alpha \, d\mathbf{v}_\alpha, \tag{1.2a}$$

$$\text{momentum} \qquad n_\alpha \mathbf{u}_\alpha = \int f_\alpha \mathbf{v}_\alpha \, d\mathbf{v}_\alpha, \tag{1.2b}$$

$$\text{total energy} \qquad n_\alpha e_\alpha = \frac{1}{2}\int f_\alpha |\mathbf{v}_\alpha|^2 \, d\mathbf{v}_\alpha. \tag{1.2c}$$

The internal energy $n_\alpha \epsilon_\alpha$ is defined as the difference between total energy and kinetic energy:

$$\text{internal energy} \qquad n_\alpha \epsilon_\alpha = n_\alpha e_\alpha - \frac{1}{2} n_\alpha |\mathbf{u}_\alpha|^2. \tag{1.3}$$

We now define the (symmetric) constraint tensor and the heat flux:

$$\text{constraint tensor} \qquad \overline{\overline{P}}_\alpha = m_\alpha \int f_\alpha (\mathbf{v}_\alpha - \mathbf{u}_\alpha) \otimes (\mathbf{v}_\alpha - \mathbf{u}_\alpha) \, d\mathbf{v}_\alpha, \tag{1.4a}$$

$$\text{heat flux} \qquad \mathbf{q}_\alpha = \frac{m_\alpha}{2} \int f_\alpha |\mathbf{v}_\alpha - \mathbf{u}_\alpha|^2 (\mathbf{v}_\alpha - \mathbf{u}_\alpha) \, d\mathbf{v}_\alpha, \tag{1.4b}$$

The constraint tensor writes $\overline{\overline{P}}_\alpha = p_\alpha \overline{\overline{I}} + \overline{\overline{\Pi}}_\alpha$ where p_α is the gas pressure, $\overline{\overline{I}}$ the identity tensor and $\overline{\overline{\Pi}}_\alpha$ the viscous tensor. We finally introduce the $\mathbf{R}_{\alpha\beta}$ and $Q_{\alpha\beta}$ terms that respectively represent the mean change in momentum and energy as a consequence of collisions between particles of species α and β:

$$\text{mean change in momentum} \qquad \mathbf{R}_{\alpha\beta} = m_\alpha \int C_{\alpha\beta}(f_\alpha, f_\beta)(\mathbf{v}_\alpha - \mathbf{u}_\alpha) \, d\mathbf{v}, \tag{1.5a}$$

$$\text{mean change in energy} \qquad Q_{\alpha\beta} = \frac{m_\alpha}{2} \int C_{\alpha\beta}(f_\alpha, f_\beta) |\mathbf{v}_\alpha - \mathbf{u}_\alpha|^2 \, d\mathbf{v}. \tag{1.5b}$$

Since the momentum and energy are conserved in collisions [28], we have:

$$R_{\alpha\beta} + R_{\beta\alpha} = 0, \tag{1.6a}$$

$$Q_{\alpha\beta} + \mathbf{u}_\alpha \cdot R_{\alpha\beta} + Q_{\beta\alpha} + \mathbf{u}_\beta \cdot R_{\beta\alpha} = 0, \tag{1.6b}$$

and in particular:

$$R_{\alpha\alpha} = 0, \tag{1.7a}$$

$$Q_{\alpha\alpha} = 0. \tag{1.7b}$$

We are now able to derive the density, momentum and total energy equations for each species from the Vlasov equation (1.1).

1.2 From kinetic to fluid equations

Taking the three first moments of the Vlasov equation (1.1) leads to the fluid evolution equations satisfied by the density, momentum and total energy of the α species (details about these calculations can be found in Appendix A):

$$\partial_t n_\alpha + \nabla \cdot (n_\alpha \mathbf{u}_\alpha) = 0, \tag{1.8a}$$

$$\partial_t \left(m_\alpha n_\alpha \mathbf{u}_\alpha\right) + \nabla \cdot \left(m_\alpha n_\alpha \mathbf{u}_\alpha \otimes \mathbf{u}_\alpha + \overline{\overline{P}}_\alpha\right) - Z_\alpha n_\alpha e \left(\mathbf{E} + \frac{\mathbf{u}_\alpha}{c} \wedge \mathbf{B}\right) = \sum_\beta R_{\alpha\beta}, \tag{1.8b}$$

$$\partial_t \left(m_\alpha n_\alpha e_\alpha\right) + \nabla \cdot \left(m_\alpha n_\alpha e_\alpha \mathbf{u}_\alpha + \overline{\overline{P_\alpha}} \cdot \mathbf{u}_\alpha + \mathbf{q}_\alpha\right) - Z_\alpha n_\alpha e \mathbf{u}_\alpha \cdot \mathbf{E} = \sum_\beta \left(Q_{\alpha\beta} + \mathbf{u}_\alpha \cdot R_{\alpha\beta}\right). \tag{1.8c}$$

The equation that governs the evolution of internal energy can be obtained by differentiating equation (1.3) with respect to time and using the fluid equations (1.8a)-(1.8c). It finally writes:

$$m_\alpha n_\alpha \left(\partial_t \epsilon_\alpha + \mathbf{u}_\alpha \cdot \nabla \epsilon_\alpha\right) + \overline{\overline{P}}_\alpha : \nabla \mathbf{u}_\alpha + \nabla \cdot \mathbf{q}_\alpha = \sum_\beta Q_{\alpha\beta}. \tag{1.9}$$

1.3 Maxwell equations

In order to take the contribution of the magnetic field into account, fluid equations are completed by the Maxwell equations (written here in CGS units):

$$\begin{cases} c\nabla \times \mathbf{E} = -\partial_t \mathbf{B}, & \text{(Faraday's equation)} \\ c\nabla \times \mathbf{B} = \partial_t \mathbf{E} + \mu \mathbf{J}, & \text{(Ampère-Maxwell's equation)} \\ \nabla \cdot \mathbf{E} = \mu Q, & \text{(Poisson's equation)} \\ \nabla \cdot \mathbf{B} = 0, \end{cases}$$

where Q is the total electric charge and $\mathbf{J}$ the current density:

$$Q = e \sum_\alpha Z_\alpha n_\alpha,$$

$$\mathbf{J} = e \sum_\alpha Z_\alpha n_\alpha \mathbf{u}_\alpha.$$

1.4 Physical hypotheses

Before we derive the complete model equations, we state a few physical hypotheses that are commonly used in the area of astrophysics and inertial confinement fusion [27, 9].

Hypothesis 1. *The plasma is assumed to be locally neutral, i.e.* $Q = n_i Z_i - n_e = 0$.

Hypothesis 2. *Since the mass of the proton is about 1836 times higher than the mass of the electron (see Table 1.1), we assume that* $m_e \lll m_i$.

Hypothesis 3. *As suggested in the literature (see [27], p. 847), the viscous tensor is neglected, i.e.* $\Pi_i = \Pi_e = 0$.

Hypothesis 4. *(MHD approximation) The displacement current* $\partial_t \mathbf{E}$ *is neglected, i.e.* $\partial_t \mathbf{E} = 0$.

Remark 3. Hypothesis 4 can be justified by performing a scaling of the Maxwell equations. Introducing $\overline{L}$ the characteristic length, $\overline{t}$ the characteristic time, $\overline{J}$ the current density scaling factor, $\overline{E}$ and $\overline{B}$ the electric and magnetic fields scaling factors and setting:

$$\mathbf{x} = \overline{L}\mathbf{x}', \quad t = \overline{t}t', \quad \mathbf{J} = \overline{J}\mathbf{J}', \quad \mathbf{E} = \overline{E}\mathbf{E}', \quad \mathbf{B} = \overline{B}\mathbf{B}',$$

the Maxwell equations become (using the plasma local neutrality hypothesis):

$$\begin{cases} \dfrac{c\overline{E}}{\overline{L}}\nabla \times \mathbf{E}' + \dfrac{\overline{B}}{\overline{t}}\partial_t \mathbf{B}' = 0, \\ \dfrac{c\overline{B}}{\overline{L}}\nabla \times \mathbf{B}' - \dfrac{\overline{E}}{\overline{t}}\partial_t \mathbf{E}' = \mu\overline{J}\mathbf{J}', \\ \nabla \cdot \mathbf{E}' = 0, \\ \nabla \cdot \mathbf{B}' = 0. \end{cases} \tag{1.10}$$

We now scale the equations so that:

$$\frac{\overline{E}}{\overline{B}} = 1 \quad \text{and} \quad \frac{\mu\overline{LJ}}{c\overline{B}} = 1.$$

Dropping the primes, the dimensionless Maxwell equations then write:

$$\begin{cases} \nabla \times \mathbf{E} + \lambda\partial_t \mathbf{B} = 0, \\ \nabla \times \mathbf{B} - \lambda\partial_t \mathbf{E} = \mathbf{J}, \\ \nabla \cdot \mathbf{E} = 0, \\ \nabla \cdot \mathbf{B} = 0, \end{cases}$$

where $\lambda = \frac{\overline{L}}{c\overline{t}}$ is a very small factor in the non-relativistic case. The displacement current therefore can be neglected in the Ampère-Maxwell equation.

Note that according to Hypothesis 4 and to the Ampère-Maxwell equation, the current density writes:

$$\mathbf{J} = \frac{c}{\mu}\nabla \times \mathbf{B},$$

and it satisfies $\nabla \cdot \mathbf{J} = 0$.

1.5 Summarized equations for the mean fluid

We now wish to derive evolution equations for the mean fluid. Introducing $\rho_\alpha = m_\alpha n_\alpha$ the mass density of the α species, we first define the total mass density ρ and the mean velocity $\mathbf{u}$ from their ionic and electronic parts:

$$\rho = \rho_i + \rho_e, \tag{1.11a}$$

$$\mathbf{u} = (\rho_i \mathbf{u}_i + \rho_e \mathbf{u}_e)/\rho. \tag{1.11b}$$

The ionic and electronic mass densities and velocities can actually be determined from the total mass density ρ and mean velocity $\mathbf{u}$. Indeed, combining the definition of total mass density (1.11a) and Hypothesis 1 (plasma local neutrality) which rewrites:

$$m_e Z_i \rho_i - \rho_e = 0,$$

the ionic and electronic mass densities are given by:

$$\rho_i = \frac{\rho}{1 + Z_i \frac{m_e}{m_i}}, \tag{1.12a}$$

$$\rho_e = \frac{\rho}{1 + Z_i \frac{m_e}{m_i}} \frac{Z_i m_e}{m_i}. \tag{1.12b}$$

Moreover, injecting Hypothesis 1 into the current density definition $\mathbf{J} = e(n_i Z_i \mathbf{u}_i - n_e \mathbf{u}_e)$ leads to:

$$\mathbf{J} = e n_e (\mathbf{u}_i - \mathbf{u}_e). \tag{1.13}$$

Combining (1.13) with the definition of the mean velocity (1.11b), both ionic and electronic velocities can be determined:

$$\mathbf{u}_i = \mathbf{u} + \frac{m_e}{e\rho}\mathbf{J}, \tag{1.14a}$$

$$\mathbf{u}_e = \mathbf{u} - \frac{m_i}{e Z_i \rho}\mathbf{J}. \tag{1.14b}$$

The evolutions equations for the mean fluid can therefore be solved in terms of the ρ and $\mathbf{u}$ unknowns without considering additional closure relations.

1.5.1 Mass continuity equation

Adding equation (1.8a) for both ions and electrons leads to:

$$\partial_t \rho = \partial_t \rho_i + \partial_t \rho_e = -\nabla \cdot (\rho_i \mathbf{u}_i + \rho_e \mathbf{u}_e) = -\nabla \cdot (\rho \mathbf{u}),$$

i.e. the mass continuity equation is given by:

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

1.5.2 Momentum equation

The momentum equation can be obtained by adding equation (1.8b) for both ions and electrons:

$$\begin{aligned}
\partial_t (\rho \mathbf{u}) &= -\nabla \cdot \left(\overline{\overline{P}}_i + \overline{\overline{P}}_e \right) - \nabla \cdot (\rho_i \mathbf{u}_i \otimes \mathbf{u}_i + \rho_e \mathbf{u}_e \otimes \mathbf{u}_e) + e(Z_i n_i - n_e)\mathbf{E} + \frac{e}{c}(Z_i n_i \mathbf{u}_i - n_e \mathbf{u}_e) \wedge \mathbf{B}, \\
&= -\nabla \cdot \left(\overline{\overline{P}}_i + \overline{\overline{P}}_e \right) - \nabla \cdot (\rho_i \mathbf{u}_i \otimes \mathbf{u}_i + \rho_e \mathbf{u}_e \otimes \mathbf{u}_e) + \frac{1}{c}\mathbf{J} \wedge \mathbf{B}.
\end{aligned}$$

Using the definition of ionic and electronic velocities (1.14a) and (1.14b), it rewrites:

$$\partial_t (\rho \mathbf{u}) = -\nabla \cdot \left(\overline{\overline{P}}_i + \overline{\overline{P}}_e \right) - \nabla \cdot (\rho_i \mathbf{u}_i \otimes \mathbf{u} + \rho_e \mathbf{u}_e \otimes \mathbf{u}) - \nabla \cdot \left(\frac{m_e \rho_i}{\rho e} \mathbf{u}_i \otimes \mathbf{J} - \frac{m_i \rho_e}{\rho e Z_i} \mathbf{u}_e \otimes \mathbf{J} \right) + \frac{1}{c}\mathbf{J} \wedge \mathbf{B}.$$

We now use the fact that $\rho_i \mathbf{u}_i + \rho_e \mathbf{u}_e = \rho \mathbf{u}$ and $\rho_\alpha = m_\alpha n_\alpha$:

$$\begin{aligned}
\partial_t (\rho \mathbf{u}) &= -\nabla \cdot \left(\overline{\overline{P}}_i + \overline{\overline{P}}_e \right) - \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \frac{m_i m_e}{e Z_i} \nabla \cdot \left(\frac{n_i Z_i \mathbf{u}_i - n_e \mathbf{u}_e}{\rho} \otimes \mathbf{J} \right) + \frac{1}{c}\mathbf{J} \wedge \mathbf{B}, \\
&= -\nabla \cdot \left(\overline{\overline{P}}_i + \overline{\overline{P}}_e \right) - \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \frac{m_i m_e}{e^2 Z_i} \nabla \cdot \left(\frac{1}{\rho}\mathbf{J} \otimes \mathbf{J} \right) + \frac{1}{c}\mathbf{J} \wedge \mathbf{B}.
\end{aligned}$$

Since $\nabla \cdot \mathbf{J} = 0$, one can notice that:

$$\nabla \cdot \left(\frac{1}{\rho} \mathbf{J} \otimes \mathbf{J} \right) = \frac{\nabla \cdot \mathbf{J}}{\rho} \mathbf{J} + \left(\nabla \frac{\mathbf{J}}{\rho} \right) \mathbf{J} = \left(\nabla \frac{\mathbf{J}}{\rho} \right) \mathbf{J},$$

and the momentum equation is thus given by:

$$\partial_t (\rho \mathbf{u}) + \nabla \cdot \left(\overline{\overline{P}}_i + \overline{\overline{P}}_e \right) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \frac{1}{c} \mathbf{J} \wedge \mathbf{B} = - \frac{m_i m_e}{e^2 Z_i} \left(\nabla \frac{\mathbf{J}}{\rho} \right) \mathbf{J}. \tag{1.15}$$

Practically, the right hand side in equation (1.15) can be neglected (see [9], p. 697). Indeed, its ratio to the momentum advection term $\nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u})$ is of order:

$$\frac{m_i m_e}{e^2 Z_i} \frac{J^2}{\rho^2 u^2}. \tag{1.16}$$

Under Hypothesis 2, relation (1.13) implies that $\mathbf{J}$ is of the order of $e n_e \mathbf{u}$ and (1.12b) leads to:

$$\frac{m_i}{Z_i \rho} = \frac{m_i}{\rho_e} \frac{1}{Z_i + \frac{m_i}{m_e}} = \frac{m_e}{\rho_e} \frac{1}{Z_i \frac{m_e}{m_i} + 1} \approx \frac{m_e}{\rho_e} \approx \frac{1}{n_e}. \tag{1.17}$$

The quantity given in (1.16) is consequently of order $\frac{m_e}{m_i}$:

$$\frac{m_i m_e}{e^2 Z_i} \frac{J^2}{\rho^2 u^2} \approx \frac{m_i m_e}{e^2 Z_i} \frac{e^2 n_e^2}{\rho^2} \approx \frac{m_e n_e}{\rho} \approx \frac{\rho_e}{\rho} \approx \frac{m_e}{m_i}.$$

The right hand side in the momentum equation (1.15) can therefore be neglected according to Hypothesis 2. Finally, using the following relation:

$$\begin{aligned} \frac{1}{c} \mathbf{J} \wedge \mathbf{B} &= \frac{1}{\mu} (\mathbf{B} \cdot \nabla) \mathbf{B} - \frac{1}{2\mu} \nabla (\mathbf{B} \cdot \mathbf{B}), \\ &= \frac{1}{\mu} \nabla \cdot (\mathbf{B} \otimes \mathbf{B}) - \frac{1}{2\mu} \nabla (\mathbf{B} \cdot \mathbf{B}) \quad \text{since } \nabla \cdot \mathbf{B} = 0, \end{aligned}$$

the momentum equation rewrites:

$$\partial_t (\rho \mathbf{u}) + \nabla \cdot \left(\overline{\overline{P}}_i + \overline{\overline{P}}_e + \rho \mathbf{u} \otimes \mathbf{u} - \frac{1}{\mu} \mathbf{B} \otimes \mathbf{B} + \frac{\mathbf{B} \cdot \mathbf{B}}{2\mu} \right) = 0. \tag{1.18}$$

1.5.3 Generalized Ohm's law

Starting from the momentum equation for electrons and using Hypothesis 2, we now derive the generalized Ohm's law who links the electric field to other quantities. Divided by the ionic mass, (1.8b) writes:

$$\frac{m_e}{m_i} \left(\partial_t (n_e \mathbf{u}_e) + \nabla \cdot (n_e \mathbf{u}_e \otimes \mathbf{u}_e) \right) + \frac{1}{m_i} \nabla \cdot \overline{\overline{P}}_e + \frac{n_e e}{m_i} \left(\mathbf{E} + \frac{\mathbf{u_e}}{c} \wedge \mathbf{B} \right) = \frac{1}{m_i} \mathbf{R}_{ei}.$$

We then replace $\mathbf{u}_e$ using relation (1.14b) and neglect the m_e/m_i terms according to Hypothesis 2:

$$\frac{1}{m_i} \nabla \cdot \overline{\overline{P}}_e + \frac{n_e e}{m_i} \mathbf{E} + \frac{n_e e}{m_i} \frac{\mathbf{u}}{c} \wedge \mathbf{B} - \frac{n_e}{Z_i \rho} \frac{\mathbf{J}}{c} \wedge \mathbf{B} = \frac{1}{m_i} \mathbf{R}_{ei}.$$

Since $m_i Z_i / \rho \approx 1/n_e$ (see (1.17)), the previous relation finally writes:

$$\mathbf{E} + \frac{1}{e n_e} \nabla \cdot \overline{\overline{P}}_e + \frac{\mathbf{u}}{c} \wedge \mathbf{B} - \frac{1}{e c n_e} \mathbf{J} \wedge \mathbf{B} = \frac{1}{e n_e} \mathbf{R}_{ei}, \tag{1.19}$$

which is called the generalized Ohm's Law.

1.5.4 Internal energy equations

Since $\rho_\alpha = m_\alpha n_\alpha$, the evolution equation for $(\rho_\alpha \epsilon_\alpha)$ can be derived using the internal energy equation (1.9):

$$\begin{aligned}\partial_t(\rho_\alpha\epsilon_\alpha) &= \rho_\alpha\partial_t\epsilon_\alpha + \epsilon_\alpha\partial_t\rho_\alpha,\\ &= -\mathbf{u}_\alpha\cdot\nabla\epsilon_\alpha - \overline{\overline{P}}_\alpha : \nabla\mathbf{u}_\alpha - \nabla\cdot\mathbf{q}_\alpha - \epsilon_\alpha\nabla\cdot(\rho_\alpha\mathbf{u}_\alpha) + \sum_\beta Q_{\alpha\beta},\\ &= -\nabla\cdot(\rho_\alpha\epsilon_\alpha\mathbf{u}_\alpha) - \overline{\overline{P}}_\alpha : \nabla\mathbf{u}_\alpha - \nabla\cdot\mathbf{q}_\alpha + \sum_\beta Q_{\alpha\beta}.\end{aligned}$$

Using the definition of ionic and electronic velocities (1.14a)-(1.14b), internal energy equations rewrite:

$$\partial_t\left(\rho_e\epsilon_e\right) + \nabla\cdot\left(\rho_e\epsilon_e\mathbf{u}\right) + \overline{\overline{P}}_e : \nabla\mathbf{u} + \nabla\cdot\mathbf{q}_e = Q_{ei} + \frac{m_i}{eZ_i}\left(\overline{\overline{P}}_e : \nabla\left(\frac{\mathbf{J}}{\rho}\right) + \nabla\cdot\left(\frac{\rho_e\epsilon_e}{\rho}\mathbf{J}\right)\right), \tag{1.20a}$$

$$\partial_t\left(\rho_i\epsilon_i\right) + \nabla\cdot\left(\rho_i\epsilon_i\mathbf{u}\right) + \overline{\overline{P}}_i : \nabla\mathbf{u} + \nabla\cdot\mathbf{q}_i = Q_{ie} - \frac{m_e}{e}\left(\overline{\overline{P}}_i : \nabla\left(\frac{\mathbf{J}}{\rho}\right) + \nabla\cdot\left(\frac{\rho_i\epsilon_i}{\rho}\mathbf{J}\right)\right). \tag{1.20b}$$

The last right hand side term in the ionic internal energy equation (1.20b) can be neglected according to Hypothesis 2 since we assume that $m_e \lll m_i$. We also recall that $m_i/(Z_i\rho) \approx 1/n_e$, see (1.17). These simplifications lead to the following evolution equations for internal energies:

$$\partial_t\left(\rho_e\epsilon_e\right) + \nabla\cdot\left(\rho_e\epsilon_e\mathbf{u}\right) + \overline{\overline{P}}_e : \nabla\mathbf{u} + \nabla\cdot\mathbf{q}_e = Q_{ei} + \overline{\overline{P}}_e : \nabla\left(\frac{\mathbf{J}}{en_e}\right) + \nabla\cdot\left(\frac{\rho_e\epsilon_e}{en_e}\mathbf{J}\right), \tag{1.21}$$

$$\partial_t\left(\rho_i\epsilon_i\right) + \nabla\cdot\left(\rho_i\epsilon_i\mathbf{u}\right) + \overline{\overline{P}}_i : \nabla\mathbf{u} + \nabla\cdot\mathbf{q}_i = Q_{ie}. \tag{1.22}$$

1.5.5 Magnetic field equation

Combining the generalized Ohm's law (1.19) and Faraday's equation, one gets the following evolution equation for the magnetic field:

$$\partial_t\mathbf{B} = c\nabla\times\left(\frac{\mathbf{u}}{c}\wedge\mathbf{B} + \frac{1}{en_e}\nabla\cdot\overline{\overline{P}}_e - \frac{1}{ecn_e}\mathbf{J}\wedge\mathbf{B} - \frac{1}{en_e}\mathbf{R}_{ei}\right).$$

Note that the $\nabla\times(\mathbf{u}\wedge\mathbf{B})$ term can be rewritten alternatively:

$$\nabla\times(\mathbf{u}\wedge\mathbf{B}) = -\nabla\cdot(\mathbf{u}\otimes\mathbf{B} - \mathbf{B}\otimes\mathbf{u}).$$

The magnetic field equation is thus finally given by:

$$\partial_t\mathbf{B} + \nabla\cdot(\mathbf{u}\otimes\mathbf{B} - \mathbf{B}\otimes\mathbf{u}) = \frac{c}{e}\nabla\times\left(\frac{1}{n_e}\nabla\cdot\overline{\overline{P}}_e\right) - \frac{1}{e}\nabla\times\left(\frac{1}{n_e}\mathbf{J}\wedge\mathbf{B}\right) - \frac{c}{e}\nabla\times\left(\frac{1}{n_e}\mathbf{R}_{ei}\right). \tag{1.23}$$

1.6 Closure relations

We now need some closure relations that complete the mass continuity, momentum, internal energy and magnetic field equations derived previously. More precisely, we have to provide expressions for the following quantities:

$$p_i,\ p_e,\ \mathbf{R}_{ei},\ Q_{ei},\ \mathbf{q}_e,\ \mathbf{q}_i.$$

1.6.1 Equations of state

In this study, we assume that both ions and electrons follow a perfect gas law. The pressure terms thus write:

$$p_\alpha = n_\alpha k_B T_\alpha,$$

where the temperature of the α species T_α is given by:

$$\epsilon_\alpha = \frac{3}{2}\frac{k_B T_\alpha}{m_\alpha}.$$

Introducing the heat capacity at constant volume $C_{v_\alpha} = \frac{3k_B}{2m_\alpha}$ and the ratio of specific heats $\gamma_\alpha = 5/3$, these relations rewrite:

$$\begin{aligned}\epsilon_\alpha &= C_{v_\alpha} T_\alpha,\\ p_\alpha &= (\gamma_\alpha - 1)\rho_\alpha \epsilon_\alpha.\end{aligned}$$

1.6.2 Transport terms

The transport terms $\mathbf{R}_{ei}$, Q_{ei}, $\mathbf{q}_e$ and $\mathbf{q}_i$ that respectively represent the friction term, the mean change in energy due to collisions and the heat fluxes still have to be determined. We consider in this study the following closure relations which have been initially proposed by Braginskii [11]:

$$\mathbf{R}_{ei} = e n_e \overline{\overline{\rho}} \cdot \mathbf{J} - n_e \overline{\overline{\beta}} \cdot \nabla\left(k_B T_e\right), \tag{1.24a}$$

$$Q_{ei} = \frac{1}{e n_e}\mathbf{R}_{ei}\cdot\mathbf{J} + 3\frac{m_e}{m_i} n_e \nu_{ei}\left(k_B T_i - k_B T_e\right), \tag{1.24b}$$

$$\mathbf{q}_e = -\frac{k_B T_e}{e}\overline{\overline{\beta}}\cdot\mathbf{J} - \overline{\overline{\kappa_e}}\cdot\nabla\left(k_B T_e\right), \tag{1.24c}$$

$$\mathbf{q}_i = -\overline{\overline{\kappa_i}}\cdot\nabla\left(k_B T_i\right). \tag{1.24d}$$

Here, $\overline{\overline{\rho}}$ denotes the resistivity tensor, $\overline{\overline{\beta}}$ the thermoelectric tensor, $\overline{\overline{\kappa_i}}$ and $\overline{\overline{\kappa_e}}$ the ionic and electronic conductivity tensors. These tensors are defined in the sequel using $(\parallel, \perp, \wedge)$ coordinates that refer to the direction of the magnetic field:

$$\mathbf{u}_\parallel = (\mathbf{h}\cdot\mathbf{u})\mathbf{h}, \quad \mathbf{u}_\perp = \mathbf{h}\wedge\mathbf{u}, \quad \mathbf{u}_\wedge = (\mathbf{h}\wedge\mathbf{u})\wedge\mathbf{h},$$

where $\mathbf{h}$ denotes a unit vector along the magnetic field direction. Using these notations, we write for any tensor $\overline{\overline{T}}$:

$$\overline{\overline{T}}\cdot\mathbf{u} = T^\parallel\mathbf{u}_\parallel + T^\perp\mathbf{u}_\perp + T^\wedge\mathbf{u}_\wedge. \tag{1.25}$$

Before giving tensors' expressions, we first introduce additional terms, namely Coulombian logarithms and collision frequencies.

Coulombian logarithms

According to previous works by Decoster [28], the Coulombian logarithm $\ln\Lambda_{\alpha\beta}$ is defined by:

$$\ln\Lambda_{\alpha\beta} = \ln\left(\frac{k_{\alpha\beta}^{\max}}{k_{\alpha\beta}^{\mathrm{D}}}\right),$$

where $k_{\alpha\beta}^{\max}$ is the de Broglie wavelength's inverse and $k_{\alpha\beta}^{\mathrm{D}}$ the Debye screening length's inverse. These write:

$$k_{\alpha\beta}^{\max} = \min\left(\frac{m_{\alpha\beta}v_{\alpha\beta}^2}{|Z_\alpha Z_\beta|e^2}, \frac{2m_{\alpha\beta}v_{\alpha\beta}}{\hbar}\right), \quad k_{\alpha\beta}^{\mathrm{D}} = \sqrt{\frac{4\pi e^2(m_\alpha T_\beta + m_\beta T_\alpha)}{k_B T_\alpha T_\beta}\left(\frac{n_i Z_i^2}{k_B m_i T_i} + \frac{n_e}{k_B T_e m_e}\right)},$$

where $m_{\alpha\beta}$ denotes the reduced mass and $v_{\alpha\beta}$ the mean velocity:

$$m_{\alpha\beta} = \frac{m_\alpha m_\beta}{m_\alpha + m_\beta}, \quad v_{\alpha\beta} = \sqrt{2\left(\frac{k_B T_\alpha}{m_\alpha} + \frac{k_B T_\beta}{m_\beta}\right)}.$$

Collision frequencies

The electron-electron, electron-ion and ion-ion collision frequencies (respectively denoted by ee, ei and ii subscripts) proposed by Decoster [28] write:

$$\nu_{ee} = \frac{4}{3}\sqrt{\pi}\frac{e^4 n_e \ln\Lambda_{ee}}{\sqrt{m_e}\left(k_B T_e\right)^{\frac{3}{2}}}, \tag{1.26a}$$

$$\nu_{ei} = \frac{4}{3}\sqrt{2\pi}\frac{Z_i^2 e^4 n_i \ln\Lambda_{ei}}{\sqrt{m_e}\left(k_B T_e\right)^{\frac{3}{2}}}, \tag{1.26b}$$

$$\nu_{ii} = \frac{4}{3}\sqrt{\pi}\frac{Z_i^4 e^4 n_i \ln\Lambda_{ii}}{\sqrt{m_i}\left(k_B T_i\right)^{\frac{3}{2}}}. \tag{1.26c}$$

We also define the electronic and ionic cyclotron frequencies:

$$\omega_e = \frac{e|\mathbf{B}|}{m_e c}, \quad \omega_i = \frac{eZ_i|\mathbf{B}|}{m_i c},$$

and the following χ_e and χ_i quantities (χ_e is the Hall parameter):

$$\chi_e = \frac{\omega_e}{\nu_{ei}}, \quad \chi_i = \frac{\omega_i}{\nu_{ii}}.$$

Resistivity tensor

According to Braginskii's closure relations [11], the resistivity tensor is given by:

$$\rho^{\parallel} = -\frac{m_e\nu_{ei}}{e^2 n_e}\alpha_0, \tag{1.27a}$$

$$\rho^{\perp} = \frac{m_e\nu_{ei}}{e^2 n_e}\left(1 - \frac{\alpha_1^{'}\chi_e^2 + \alpha_0^{'}}{\chi_e^4 + \delta_1\chi_e^2 + \delta_0}\right), \tag{1.27b}$$

$$\rho^{\wedge} = -\frac{m_e\nu_{ei}}{e^2 n_e}\chi_e\frac{\alpha_1^{''}\chi_e^2 + \alpha_0{''}}{\chi_e^4 + \delta_1\chi_e^2 + \delta_0}, \tag{1.27c}$$

with constants α_0, α_0', α_1', α_0'', α_1'', δ_0 and δ_1 given in Table 1.2.

Thermoelectric tensor

Braginskii's [11] thermoelectric tensor is given by:

$$\beta^{\parallel} = \beta_0, \tag{1.28a}$$

$$\beta^{\perp} = \frac{\beta_1^{'}\chi_e^2 + \beta_0^{'}}{\chi_e^4 + \delta_1\chi_e^2 + \delta_0}, \tag{1.28b}$$

$$\beta^{\wedge} = \chi_e \frac{\beta_1^{''}\chi_e^2 + \beta_0{''}}{\chi_e^4 + \delta_1\chi_e^2 + \delta_0}, \tag{1.28c}$$

with constants β_0, β_0', β_1', β_0'', β_1'', δ_0 and δ_1 given in Table 1.2.

Electronic and ionic conductivity tensors

Braginskii's [11] electronic and ionic conductivity tensors are respectively given by:

$$\kappa_e^{\parallel} = \frac{n_e k_B T_e}{m_e \nu_{ei}} \gamma_0, \tag{1.29a}$$

$$\kappa_e^{\perp} = \frac{n_e k_B T_e}{m_e \nu_{ei}} \frac{\gamma_1^{'}\chi_e^2 + \gamma_0^{'}}{\chi_e^4 + \delta_1\chi_e^2 + \delta_0}, \tag{1.29b}$$

$$\kappa_e^{\wedge} = \frac{n_e k_B T_e}{m_e \nu_{ei}} \chi_e \frac{\gamma_1^{''}\chi_e^2 + \gamma_0{''}}{\chi_e^4 + \delta_1\chi_e^2 + \delta_0}, \tag{1.29c}$$

and:

$$\kappa_i^{\parallel} = 3.906 \frac{n_i k_B T_i}{m_i \nu_{ii}}, \tag{1.30a}$$

$$\kappa_i^{\perp} = \frac{n_i k_B T_i}{m_i \nu_{ii}} \frac{2\chi_i^2 + 2.645}{\chi_i^4 + 2.70\chi_i^2 + 0.677}, \tag{1.30b}$$

$$\kappa_i^{\wedge} = \frac{n_i k_B T_i}{m_i \nu_{ii}} \chi_i \frac{\frac{5}{2}\chi_i^2 + 4.65}{\chi_i^4 + 2.70\chi_i^2 + 0.677}, \tag{1.30c}$$

with constants γ_0, γ_0', γ_1', γ_0'', γ_1'', δ_0 and δ_1 given in Table 1.2.

	$Z=1$	$Z=2$	$Z=3$	$Z=4$	$Z\to\infty$
δ_0	3.7703	1.0465	0.5214	0.4106	0.0961
δ_1	14.79	10.80	9.618	9.055	7.482
α_1'	6.416	5.523	5.226	5.077	4.63
α_0'	1.837	0.5956	0.3515	0.2566	0.0678
α_1''	1.704	1.704	1.704	1.704	1.704
α_0''	0.7796	0.3439	0.2400	0.1957	0.0940
β_1'	5.101	4.450	4.233	4.124	3.798
β_0'	2.681	0.9473	0.5905	0.4478	0.1461
β_1''	1.5	1.5	1.5	1.5	1.5
β_0''	3.053	1.784	1.442	1.285	0.877
γ_1'	4.664	3.957	3.721	3.604	3.25
γ_0'	11.92	5.118	3.525	2.841	1.20
γ_1''	2.5	2.5	2.5	2.5	2.5
γ_0''	21.67	15.37	13.53	12.65	10.23
$\alpha_0=1-\alpha_0'/\delta_0$	0.5129	0.4408	0.3965	0.3752	0.2949
$\beta_0=\beta_0'/\delta_0$	0.7110	0.9052	1.016	1.090	1.521
$\gamma_0=\gamma_0'/\delta_0$	3.1616	4.890	6.064	6.920	12.471

Table 1.2: Transport coefficients given by Braginkii [11].

Chapter 2

Splitting of the complete model

At this stage, the closure relations have been given and we are able to write the complete system of equations that governs the evolution of both hydrodynamic quantities and magnetic field. We first recall the physical hypotheses which have been stated in chapter 1.

- The plasma is assumed to be locally neutral, *i.e.* $Q = n_i Z_i - n_e = 0$.
- Equations have been derived in the $\frac{m_e}{m_i} \to 0$ limit.
- The viscous tensor is neglected in this study so that the constraint tensor writes $\overline{\overline{P}}_\alpha = p_\alpha \overline{\overline{I}}$.
- The MHD approximation is considered here, *i.e.* the displacement current $\partial_t \mathbf{E}$ is neglected and the Ampère-Maxwell equations thus writes:

$$\mathbf{J} = \frac{c}{\mu} \nabla \times \mathbf{B}.$$

Under these assumptions, the considered evolution equations are given by:

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot \left(\rho \mathbf{u} \otimes \mathbf{u} + \left(p_i + p_e + \frac{\mathbf{B}\cdot\mathbf{B}}{2\mu} \right) \underline{\mathbf{I}} - \frac{\mathbf{B}\otimes\mathbf{B}}{\mu} \right) = 0, \\ \partial_t (\rho_e \epsilon_e) + \nabla \cdot (\rho_e \epsilon_e \mathbf{u}) + p_e \nabla \cdot \mathbf{u} + \nabla \cdot \mathbf{q}_e - \left(\overline{\overline{\rho}} \cdot \mathbf{J} - \frac{1}{e} \overline{\overline{\beta}} \cdot \nabla (k_B T_e) \right) \cdot \mathbf{J} + p_e \nabla \left(\frac{\mathbf{J}}{e n_e} \right) + \nabla \cdot \left(\frac{\rho_e \epsilon_e}{e n_e} \mathbf{J} \right) = \Sigma_{ei}, \\ \partial_t (\rho_i \epsilon_i) + \nabla \cdot (\rho_i \epsilon_i \mathbf{u}) + p_i \nabla \cdot \mathbf{u} + \nabla \cdot \mathbf{q}_i = \Sigma_{ie}, \\ \partial_t \mathbf{B} + \nabla \cdot (\mathbf{u} \otimes \mathbf{B} - \mathbf{B} \otimes \mathbf{u}) - c \, \nabla \times \left(\frac{1}{e n_e} \nabla p_e - \frac{1}{e c n_e} \mathbf{J} \wedge \mathbf{B} + \frac{1}{e} \overline{\overline{\beta}} \cdot (k_B \nabla T_e) - \overline{\overline{\rho}} \cdot \mathbf{J} \right) = 0. \end{cases}$$

where Σ_{ei} denotes the following relaxation term:

$$\Sigma_{ei} = -\Sigma_{ie} = 3 \frac{m_e}{m_i} n_e \nu_{ei} k_B (T_i - T_e).$$

Introducing P the total pressure:

$$P = p_i + p_e + \frac{\mathbf{B} \cdot \mathbf{B}}{2\mu},$$

and the $\mathbf{S}_{\text{self}}$, $\mathbf{S}_{\text{Hall}}$ and $\mathbf{S}_{\text{Nernst}}$ source terms that respectively govern the magnetic field generation, the Hall effect and the Nernst effect:

$$\mathbf{S}_{\text{self}} = \frac{1}{e n_e} \nabla p_e, \quad \mathbf{S}_{\text{Hall}} = -\frac{1}{e c n_e} \mathbf{J} \wedge \mathbf{B}, \quad \mathbf{S}_{\text{Nernst}} = \frac{1}{e} \overline{\overline{\beta}} \cdot \nabla \left(k_B T_e \right),$$

the previous system rewrites:

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot \left(\rho \mathbf{u} \otimes \mathbf{u} + P \underline{\mathbf{I}} - \frac{\mathbf{B} \otimes \mathbf{B}}{\mu} \right) = 0, \\ \partial_t (\rho_e \epsilon_e) + \nabla \cdot (\rho_e \epsilon_e \mathbf{u}) + p_e \nabla \cdot \mathbf{u} + \nabla \cdot \mathbf{q}_e - \left(\overline{\overline{\rho}} \cdot \mathbf{J} - \mathbf{S}_{\text{Nernst}} \right) \cdot \mathbf{J} + p_e \nabla \left(\frac{\mathbf{J}}{e n_e} \right) + \nabla \cdot \left(\frac{\rho_e \epsilon_e}{e n_e} \mathbf{J} \right) = \Sigma_{ei}, \\ \partial_t (\rho_i \epsilon_i) + \nabla \cdot (\rho_i \epsilon_i \mathbf{u}) + p_i \nabla \cdot \mathbf{u} + \nabla \cdot \mathbf{q}_i = \Sigma_{ie}, \\ \partial_t \mathbf{B} + \nabla \cdot (\mathbf{u} \otimes \mathbf{B} - \mathbf{B} \otimes \mathbf{u}) - c \, \nabla \times \left(\mathbf{S}_{\text{self}} + \mathbf{S}_{\text{Hall}} + \mathbf{S}_{\text{Nernst}} - \overline{\overline{\rho}} \cdot \mathbf{J} \right) = 0. \end{cases} \tag{2.1}$$

System (2.1) relies on mathematical operators of different natures. We therefore have chosen to split it in several subsystems according to the nature of the underlying mathematical term. These are detailed in the sequel. This strategy allows us to develop numerical schemes that are well-suited to each operator. It also simplifies the validation step since it can be performed separately for each subsystem

2.1 Hyperbolic operator: ideal MHD system of equations

The hyperbolic operator involved in system (2.1) is actually the ideal MHD system of equations. It writes:

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot \left(\rho \mathbf{u} \otimes \mathbf{u} + P \underline{\mathbf{I}} - \frac{\mathbf{B} \otimes \mathbf{B}}{\mu} \right) = 0, \\ \partial_t (\rho_e \epsilon_e) + \nabla \cdot (\rho_e \epsilon_e \mathbf{u}) + p_e \nabla \cdot \mathbf{u} = 0, \\ \partial_t (\rho_i \epsilon_i) + \nabla \cdot (\rho_i \epsilon_i \mathbf{u}) + p_i \nabla \cdot \mathbf{u} = 0, \\ \partial_t \mathbf{B} + \nabla \cdot (\mathbf{u} \otimes \mathbf{B} - \mathbf{B} \otimes \mathbf{u}) = 0. \end{cases} \tag{2.2}$$

In this study, we do not solve the internal energy formulation (2.2) of the ideal MHD equations. We indeed rewrite system (2.2) in a more convenient conservative form. To that end, we consider the total energy ρe whose definition is recalled below:

$$\rho e = \rho_i \epsilon_i + \rho_e \epsilon_e + \frac{\rho}{2} \mathbf{u} \cdot \mathbf{u} + \frac{\mathbf{B} \cdot \mathbf{B}}{2\mu}.$$

One can show that ρe satisfies the following evolution equation:

$$\partial_t (\rho e) + \nabla \cdot \left(\rho e \mathbf{u} + P \mathbf{u} - \frac{\mathbf{u} \cdot \mathbf{B}}{\mu} \mathbf{B} \right) = 0.$$

We also introduce another variable change which has been widely studied in the related literature [48, 22, 23, 31] and proved to be admissible in the case we are considering. It consists in considering the electronic entropy S_e defined by:

$$S_e = \ln \left(\delta(\rho_e \epsilon_e) \rho^{-\gamma_e} \right),$$

where δ is an arbitrary constant. The evolution equations of S_e is given by:

$$\begin{aligned}
\partial_t S_e &= \frac{1}{\rho_e \epsilon_e}\partial_t(\rho_e\epsilon_e) - \frac{\gamma_e}{\rho}\partial_t\rho, \\
&= -\frac{1}{\rho_e\epsilon_e}\left(\mathbf{u}\cdot\nabla(\rho_e\epsilon_e) + \rho_e\epsilon_e\nabla\cdot\mathbf{u} + p_e\nabla\cdot\mathbf{u}\right) + \frac{\gamma_e}{\rho}\left(\rho\nabla\cdot\mathbf{u} + \mathbf{u}\cdot\nabla\rho\right), \\
&= -\mathbf{u}\cdot\left(\nabla S_e + \frac{\gamma_e}{\rho}\nabla\rho\right) - \nabla\cdot\mathbf{u} - (\gamma_e - 1)\nabla\cdot\mathbf{u} + \gamma_e\nabla\cdot\mathbf{u} + \frac{\gamma_e}{\rho}\mathbf{u}\cdot\nabla\rho, \\
&= -\mathbf{u}\cdot\nabla S_e.
\end{aligned}$$

It leads to a conservative evolution equation for ρS_e:

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

We finally get the following conservative ideal MHD system of equations which has been obtained by replacing the internal energy equations by the total energy and electronic entropy ones:

$$\partial_t\begin{pmatrix}\rho\\ \rho\mathbf{u}\\ \rho e\\ \rho S_e\\ \mathbf{B}\end{pmatrix} + \nabla\cdot\begin{pmatrix}\rho\mathbf{u}\\ \rho\mathbf{u}\otimes\mathbf{u} + P\underline{\mathbf{I}} - \frac{\mathbf{B}\otimes\mathbf{B}}{\mu}\\ \rho e\mathbf{u} + P\mathbf{u} - \frac{\mathbf{u}\cdot\mathbf{B}}{\mu}\mathbf{B}\\ \rho S_e\mathbf{u}\\ \mathbf{u}\otimes\mathbf{B} - \mathbf{B}\otimes\mathbf{u}\end{pmatrix} = \mathbf{0}.$$

2.2 First parabolic operator: thermal conduction

A parabolic operator is involved for thermal conduction with an additional coupling term for the two-temperature case. The corresponding system of equations writes:

$$\begin{cases}
\partial_t\rho = 0, \\
\partial_t(\rho\mathbf{u}) = 0, \\
\partial_t(\rho_e\epsilon_e) + \nabla\cdot\mathbf{q}_e = 3\frac{m_e}{m_i}n_e\nu_{ei}k_B(T_i - T_e), \\
\partial_t(\rho_i\epsilon_i) + \nabla\cdot\mathbf{q}_i = 3\frac{m_e}{m_i}n_e\nu_{ei}k_B(T_e - T_i), \\
\partial_t\mathbf{B} = 0.
\end{cases} \tag{2.3}$$

Note that due to cancellation of coupling terms, the total energy equation is conservative:

$$\partial_t(\rho e) + \nabla\cdot(\mathbf{q}_i + \mathbf{q}_e) = 0.$$

2.3 Second parabolic operator: resistive conduction

Another parabolic operator governs resistive conduction. The corresponding system of equations writes:

$$\begin{cases} \partial_t \rho = 0, \\ \partial_t(\rho \mathbf{u}) = 0, \\ \partial_t(\rho_e \epsilon_e) - (\overline{\overline{\rho}} \cdot \mathbf{J}) \cdot \mathbf{J} = 0, \\ \partial_t(\rho_i \epsilon_i) = 0, \\ \partial_t \mathbf{B} + c \, \nabla \times (\overline{\overline{\rho}} \cdot \mathbf{J}) = 0. \end{cases} \tag{2.4}$$

It is again more convenient to solve the total energy equation since it can be written in a conservative form. Indeed, the magnetic pressure equation writes:

$$\begin{aligned} \partial_t \left(\frac{\mathbf{B} \cdot \mathbf{B}}{2\mu} \right) &= -\frac{c}{\mu} \mathbf{B} \cdot \nabla \times (\overline{\overline{\rho}} \cdot \mathbf{J}), \\ &= -\frac{c}{\mu} \nabla \cdot \left((\overline{\overline{\rho}} \cdot \mathbf{J}) \wedge \mathbf{B} \right) - \frac{c}{\mu} (\overline{\overline{\rho}} \cdot \mathbf{J}) \cdot \nabla \times \mathbf{B}, \\ &= -\frac{c}{\mu} \nabla \cdot \left((\overline{\overline{\rho}} \cdot \mathbf{J}) \wedge \mathbf{B} \right) - (\overline{\overline{\rho}} \cdot \mathbf{J}) \cdot \mathbf{J}, \end{aligned}$$

and it leads to the following evolution equation for total energy:

$$\begin{aligned} \partial_t(\rho e) &= \partial_t(\rho_e \epsilon_e) + \partial_t \left(\frac{\mathbf{B} \cdot \mathbf{B}}{2\mu} \right), \\ &= (\overline{\overline{\rho}} \cdot \mathbf{J}) \cdot \mathbf{J} - (\overline{\overline{\rho}} \cdot \mathbf{J}) \cdot \mathbf{J} - \frac{c}{\mu} \nabla \cdot \left((\overline{\overline{\rho}} \cdot \mathbf{J}) \wedge \mathbf{B} \right), \\ &= -\frac{c}{\mu} \nabla \cdot \left((\overline{\overline{\rho}} \cdot \mathbf{J}) \wedge \mathbf{B} \right). \end{aligned}$$

Finally, system (2.4) can be rewritten in a conservative form:

$$\begin{cases} \partial_t \rho = 0, \\ \partial_t(\rho \mathbf{u}) = 0, \\ \partial_t(\rho_i \epsilon_i) = 0, \\ \partial_t(\rho e) + \frac{c}{\mu} \nabla \cdot \left((\overline{\overline{\rho}} \cdot \mathbf{J}) \wedge \mathbf{B} \right) = 0, \\ \partial_t \mathbf{B} + c \, \nabla \times (\overline{\overline{\rho}} \cdot \mathbf{J}) = 0. \end{cases} \tag{2.5}$$

2.4 Source terms

The source terms effects are governed by the following set of equations:

$$\begin{cases} \partial_t \rho = 0, \\ \partial_t(\rho \mathbf{u}) = 0, \\ \partial_t(\rho_i \epsilon_i) = 0, \\ \partial_t(\rho_e \epsilon_e) + \mathbf{S}_{\text{Nernst}} \cdot \mathbf{J} = p_e \nabla \cdot \left(\frac{\mathbf{J}}{e n_e} \right) + \nabla \cdot \left(\frac{\rho_e \epsilon_e}{e n_e} \mathbf{J} \right), \\ \partial_t \mathbf{B} = c \, \nabla \times \left(\mathbf{S}_{\text{self}} + \mathbf{S}_{\text{Hall}} + \mathbf{S}_{\text{Nernst}} \right). \end{cases} \tag{2.6}$$

Once again, it is more convenient to solve the total energy equation since it can be written in a conservative form. To that end, we first consider the magnetic pressure equation:

$$\begin{aligned}\partial_t\left(\frac{\mathbf{B}\cdot\mathbf{B}}{2\mu}\right) &= \frac{1}{\mu}\mathbf{B}\cdot\partial_t\mathbf{B},\\ &= \frac{c}{\mu}\mathbf{B}\cdot\nabla\times\mathbf{S}_{\text{self}} + \frac{c}{\mu}\mathbf{B}\cdot\nabla\times\mathbf{S}_{\text{Hall}} + \frac{c}{\mu}\mathbf{B}\cdot\nabla\times\mathbf{S}_{\text{Nernst}},\end{aligned}$$

and since $\nabla\cdot(\mathbf{A}\wedge\mathbf{B}) = \mathbf{B}\cdot(\nabla\times\mathbf{A}) - \mathbf{A}\cdot(\nabla\times\mathbf{B})$:

$$\partial_t\left(\frac{\mathbf{B}\cdot\mathbf{B}}{2\mu}\right) = \frac{c}{\mu}\nabla\cdot(\mathbf{S}_{\text{self}}\wedge\mathbf{B} + \mathbf{S}_{\text{Hall}}\wedge\mathbf{B} + \mathbf{S}_{\text{Nernst}}\wedge\mathbf{B}) + \frac{c}{\mu}(\nabla\times\mathbf{B})\cdot(\mathbf{S}_{\text{self}} + \mathbf{S}_{\text{Hall}} + \mathbf{S}_{\text{Nernst}})\,.$$

We recall that $\mathbf{J} = \frac{c}{\mu}(\nabla\times\mathbf{B})$ and $\mathbf{S}_{\text{Hall}} = -\frac{1}{ecn_e}(\mathbf{J}\wedge\mathbf{B})$:

$$\partial_t\left(\frac{\mathbf{B}\cdot\mathbf{B}}{2\mu}\right) = \frac{c}{\mu}\nabla\cdot(\mathbf{S}_{\text{self}}\wedge\mathbf{B} + \mathbf{S}_{\text{Hall}}\wedge\mathbf{B} + \mathbf{S}_{\text{Nernst}}\wedge\mathbf{B}) + \mathbf{S}_{\text{self}}\cdot\mathbf{J} - \frac{1}{en_e}(\mathbf{J}\wedge\mathbf{B})\cdot\mathbf{J} + \mathbf{S}_{\text{Nernst}}\cdot\mathbf{J},$$

and finally, since $(\mathbf{J}\wedge\mathbf{B})\cdot\mathbf{J} = 0$:

$$\partial_t\left(\frac{\mathbf{B}\cdot\mathbf{B}}{2\mu}\right) = \frac{c}{\mu}\nabla\cdot(\mathbf{S}_{\text{self}}\wedge\mathbf{B} + \mathbf{S}_{\text{Hall}}\wedge\mathbf{B} + \mathbf{S}_{\text{Nernst}}\wedge\mathbf{B}) + \mathbf{S}_{\text{self}}\cdot\mathbf{J} + \mathbf{S}_{\text{Nernst}}\cdot\mathbf{J}.$$

We now recall that $\mathbf{S}_{\text{self}} = \frac{1}{en_e}\nabla p_e$ and we write the total energy equation:

$$\begin{aligned}\partial_t(\rho e) &= \partial_t(\rho_e\epsilon_e) + \partial_t\left(\frac{\mathbf{B}\cdot\mathbf{B}}{2\mu}\right),\\ &= -\mathbf{S}_{\text{Nernst}}\cdot\mathbf{J} + p_e\nabla\cdot\left(\frac{\mathbf{J}}{en_e}\right) + \nabla\cdot\left(\frac{\rho_e\epsilon_e}{en_e}\mathbf{J}\right) + \partial_t\left(\frac{\mathbf{B}\cdot\mathbf{B}}{2\mu}\right),\\ &= \frac{c}{\mu}\nabla\cdot(\mathbf{S}_{\text{self}}\wedge\mathbf{B} + \mathbf{S}_{\text{Hall}}\wedge\mathbf{B} + \mathbf{S}_{\text{Nernst}}\wedge\mathbf{B}) + \frac{1}{n_e}\nabla p_e\cdot\mathbf{J} + p_e\nabla\cdot\left(\frac{\mathbf{J}}{en_e}\right) + \nabla\cdot\left(\frac{\rho_e\epsilon_e}{en_e}\mathbf{J}\right),\\ &= \nabla\cdot\left(\frac{p_e+\rho_e\epsilon_e}{en_e}\mathbf{J} + \frac{c}{\mu}\mathbf{S}_{\text{self}}\wedge\mathbf{B} + \frac{c}{\mu}\mathbf{S}_{\text{Hall}}\wedge\mathbf{B} + \frac{c}{\mu}\mathbf{S}_{\text{Nernst}}\wedge\mathbf{B}\right).\end{aligned}$$

Finally, the conservative set of equations that governs source terms effects is given by:

$$\begin{cases}\partial_t\rho = 0,\\ \partial_t(\rho\mathbf{u}) = 0,\\ \partial_t(\rho_i\epsilon_i) = 0,\\ \partial_t(\rho e) = \nabla\cdot\left(\frac{p_e+\rho_e\epsilon_e}{en_e}\mathbf{J} + \frac{c}{\mu}\mathbf{S}_{\text{self}}\wedge\mathbf{B} + \frac{c}{\mu}\mathbf{S}_{\text{Hall}}\wedge\mathbf{B} + \frac{c}{\mu}\mathbf{S}_{\text{Nernst}}\wedge\mathbf{B}\right),\\ \partial_t\mathbf{B} = c\,\nabla\times(\mathbf{S}_{\text{self}} + \mathbf{S}_{\text{Hall}} + \mathbf{S}_{\text{Nernst}})\,.\end{cases} \tag{2.7}$$

Part II

High-order schemes for multidimensional compressible hydrodynamics and ideal magnetohydrodynamics

Chapter 3

State of the art high-order schemes for hydrodynamics and ideal MHD

The accurate simulation of hydrodynamical and magnetohydrodynamical flows is an important topic in several areas of science and engineering. While second order accurate simulations have been carried out for a while, recent advances have made it possible to go beyond second-order accuracy. We describe in this chapter a few well-known high-order numerical methods for solving the hydrodynamics and ideal MHD systems of conservation laws: the Discontinuous Galerkin (DG) methods, the weighted essentially non-oscillatory (WENO) interpolation techniques and the Arbitrary Derivative Riemann Problem (ADER) class of schemes. But contrarily to most high-order schemes reviews, we here propose a short analysis of these methods in terms of complexity and computational cost. In particular, we give a few hints about their predictable performances and parallelization abilities according to their appropriateness to the three principles that we stated in introduction (see page 13).

3.1 Discontinuous Galerkin (DG) methods

3.1.1 Principle of DG methods

Discontinuous Galerkin methods consists in seeking for each time t an approximate solution $\mathbf{U}_h$ to the hyperbolic system of conservation laws

$$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{0}, \tag{3.1}$$

in the finite element space of discontinuous functions

$$W_h = \{\mathbf{V}_h \in L^\infty(\Omega),\ \mathbf{V}_h|_K \in W(K)\ \forall K \in \mathcal{T}_h\}, \tag{3.2}$$

where $\mathcal{T}_h$ is a triangulation of the domain $\Omega \subset \mathbb{R}^d$ and $W(K)$ is the so-called local space. More precisely, one has to find a decomposition of $\mathbf{U}_h$ into functions of the local space for each element:

$$\mathbf{U}_h(t)|_K = \sum_{n=1}^{N} \Phi_n \mathbf{u}_n^K(t),\ \mathbf{u}_n^K(t) \in \mathbb{R}^d\ \forall n,$$

where N denotes the dimension of the local space $W(K)$ and Φ_n is a basis function of $W(K)$. The time-dependent $\mathbf{u}_n^K$ coefficients are called the degrees of freedom. In order to determine the approximate solution

$\mathbf{U}_h$, one needs the following weak formulation of (3.1):

$$\partial_t \int_K v\mathbf{U}\ d\Omega + \sum_{e\in\delta K} \underbrace{\int_e v(\mathbf{F}(\mathbf{U})\cdot\mathbf{n}_{e,K})\mathbf{n}_{e,K}\ d(\delta\Omega)}_{\mathcal{I}_e} - \underbrace{\int_K \mathbf{F}(\mathbf{U})\cdot\nabla v\ d\Omega}_{\mathcal{I}_K} = \mathbf{0}, \tag{3.3}$$

where v denotes a smooth test function and $\mathbf{n}_{e,K}$ the outward unit normal to the edge e. The weak formulation (3.3) now has to be discretized. To that end, the $\mathcal{I}_e$ and $\mathcal{I}_K$ integrals are approximated by quadrature rules:

$$\int_e v\left(\mathbf{F}(\mathbf{U})\cdot\mathbf{n}_{e,K}\right)\mathbf{n}_{e,K}\ d(\delta\Omega) \approx \sum_{l=1}^{L} \omega_l v(\mathbf{x}_l^e)\left(\mathbf{F}(\mathbf{U}(\mathbf{x}_l^e))\cdot\mathbf{n}_{e,K}(\mathbf{x}_l^e)\right)\mathbf{n}_{e,K}(\mathbf{x}_l^e), \tag{3.4}$$

$$\int_K \mathbf{F}(\mathbf{U})\cdot\nabla v\ d\Omega \approx \sum_{m=1}^{M} \omega_m \mathbf{F}(\mathbf{U}(\mathbf{x}_m^K))\cdot(\nabla v)(\mathbf{x}_m^K), \tag{3.5}$$

where $\mathbf{x}_l^e$ denotes the quadrature points along edge e and $\mathbf{x}_m^K$ the quadrature points over element K, both being respectively associated to the ω_l and ω_m weights. The quadrature rule used for discretizing $\mathcal{I}_e$ and $\mathcal{I}_K$ must be chosen in order to match the desired order: practically, it is often chosen so that each local space function is integrated exactly. For an exhaustive review of quadrature rules, we refer the reader to [68] and references therein. Then the $\mathbf{F}(\mathbf{U}(\mathbf{x}_l^e))$ flux term along edge e is replaced by the numerical flux $\mathbf{F}_l^{e,K}$ whose value at $\mathbf{x}_l^e$ is obtained from the interior and the exterior of the element K. More precisely, defining

$$\text{the interior value } \mathbf{U}_h(\mathbf{x}_{\text{int}}^K,t) = \lim_{\mathbf{y}\to\mathbf{x}_l^e,\ \mathbf{y}\in K} \mathbf{U}_h(\mathbf{y},t),$$

$$\text{and the exterior value } \mathbf{U}_h(\mathbf{x}_{\text{ext}}^K,t) = \begin{cases} \text{imposed by boundary conditions} & \text{if } \mathbf{x}_l^e \in \delta\Omega, \\ \lim\limits_{\mathbf{y}\to\mathbf{x}_l^e,\ \mathbf{y}\notin K} \mathbf{U}_h(\mathbf{y},t) & \text{otherwise.} \end{cases}$$

the numerical flux along the edge e of element K at $\mathbf{x}=\mathbf{x}_l^e$ is given by:

$$\mathbf{F}_l^{e,K} = \mathbf{H}\left(\mathbf{U}_h(\mathbf{x}_{\text{int}}^K,t),\ \mathbf{U}_h(\mathbf{x}_{\text{ext}}^K,t)\right),$$

where $\mathbf{H}$ is any exact or approximate Riemann solver. Once the $\mathcal{I}_e$ and $\mathcal{I}_K$ integrals have been computed, $\mathbf{U}$ has to be replaced by the decomposition of $\mathbf{U}_h$ into basis functions of the local space, leading to the following numerical scheme:

$$\sum_{n=1}^{N}\left(\partial_t \mathbf{u}_n^K(t)\right)\int_K v\Phi_n\ d\Omega = \sum_{m=1}^{M} \omega_m \mathbf{F}(\mathbf{U}(\mathbf{x}_m^K))\cdot(\nabla v)(\mathbf{x}_m^K) - \sum_{l=1}^{L} \omega_l v(\mathbf{x}_l^e)\left(\mathbf{F}_l^{e,K}\cdot\mathbf{n}_{e,K}(\mathbf{x}_l^e)\right)\mathbf{n}_{e,K}(\mathbf{x}_l^e), \tag{3.6}$$

with the initial condition

$$\mathbf{U}_h(t=0) = P_{W_h}\left(\mathbf{U}(t=0)\right), \tag{3.7}$$

P_{W_h} being a projection method onto the finite element space W_h. At this stage, one has to write scheme (3.6) with $v=\Phi_n$ for $1\le n\le N$. This strategy finally leads to a linear system for each element K whose unknowns are the degrees of freedom of $\mathbf{U}_h$ (contrarily to the finite element method that requires the resolution of a global linear system for the whole mesh).

Building DG schemes finally consists in choosing three main parameters: the Riemann solver $\mathbf{H}$, the local space $W(K)$ (that governs the accuracy in space of the resulting scheme) and the time discretization method (that governs its accuracy in time).

	d = 1	**d = 2**	**d = 3**
r = 1	1	1	1
r = 2	2	3	4
r = 3	3	6	10
r = 4	4	10	20
r = 5	5	15	36

Table 3.1: Dimension of the local space $W(K)$ for different values of r (desired order of accuracy) and d (number of space dimensions).

3.1.2 Computational complexity analysis

For a given system of conservation laws, the computational complexity of DG methods mainly depends on the choice of the local space. DG schemes are actually known to be quite expensive since a dense linear system has to be solved for each element in order to determine the degrees of freedom of the approximate solution. The size of this linear system is given by:

$$\text{dimension of the local space} \times \text{number of unknowns in system (3.1)}.$$

For low-order schemes, the size of this system stays reasonable and its shape is usually convenient (it often involves a diagonal matrix, see [18]) but it increases noticeably when dealing with higher-order DG methods. Indeed, the usual choice for building an r-th order scheme consists in taking $W(K) = P_{r-1}(K)$ the space of $(r-1)$-th degree polynomials. Table 3.1 summarizes the dimension of $W(K)$ for different values of r and d (we recall that d denotes the dimension of Ω): one can notice that it rapidly increases in the multidimensional case and makes high-order DG methods very expensive, in particular in the case of complex systems like magnetohydrodynamics. Moreover, preserving robustness in the case of high-order accurate DG techniques usually imposes to resort to expensive slope limiters that involve many conditional tests and thus do not fulfill the first principle that we stated in introduction (see page 13). According to recent advances [39], DG schemes seem nevertheless to take advantage of GPU implementations and thus tend to become competitive in terms of computational performance.

3.1.3 Short review of DG schemes

Second and third-order accurate DG methods for the Euler equations have been introduced by Cockburn and Shu [18] who had previously designed such schemes in more simple frameworks [17, 16, 15, 14]. Their schemes rely on first and second-degree polynomial spaces combined with a Lax-Friedrichs numerical flux and Runge-Kutta time stepping which allows to reach high-order accuracy in time but at the cost of several boundary conditions calls. This approach thus does not fulfill the third principle that we stated in introduction, see page 14).

Dumbser and Munz propose in [34] a different approach that allows to design arbitrary high-order DG schemes with a one-step temporal discretization relying on the Cauchy-Kovalevskaya procedure described in section 3.3. Called the ADER-DG class of schemes, it resorts to classical polynomial local spaces and to the ADER methodology (which is described in section 3.3) for building high-order accurate numerical fluxes along edges. Dumbser *et al.* propose later in [33] an elegant unification of finite volume and DG

frameworks for building one-step schemes on unstructured meshes but this goes beyond the scope of DG techniques.

DG schemes have been firstly applied to magnetohydrodynamics by Warburton and Karniadakis [91] in association with Powell's eight wave formulation (see section 5.2.3) for preserving the divergence constraint $\nabla \cdot \mathbf{B} = 0$. These rely on classical polynomial local spaces combined with a multi-step Adams-Bashforth time discretization. Li and Shu [56] propose a different strategy with their locally divergence-free DG schemes for ideal MHD: they impose the divergence-free aspect of the magnetic field by choosing the local space for the magnetic field components so that the $\nabla \cdot \mathbf{B} = 0$ constraint is satisfied in each cell. High-order accuracy in time is achieved thanks to Runge-Kutta time stepping in this case. We can also mention works by Taube *et al.* [79] who extend the ADER-DG class of schemes to the case of ideal MHD but without any treatment for enforcing the divergence constraint.

3.2 Weighted essentially non-oscillatory (WENO) methods

High-order schemes are usually based on polynomial interpolations of discrete data which provide accuracy benefits on smooth solutions. Nevertheless, it is well-known that such interpolation techniques generate spurious oscillations in the vicinity of discontinuities (called Gibbs phenomena) which do not decay in magnitude when the mesh is refined. In the late 1980's, there were two main methods to deal with these unphysical oscillations.

- The first one consists in adding artificial viscosity that smoothens variable profiles and reduces oscillations phenomena, thus improving robustness. But these methods rely on parameters who are most of times test problem dependent and therefore need to be adjusted "by hand" to the considered problem.

- The second one is slope limiting. Applying slope limiters to high-order interpolants usually provides robustness improvements but at the cost of accuracy: such techniques indeed enforce first-order accuracy in the vicinity of discontinuities.

The ENO (for essentially non-oscillatory) idea introduced in [41] seems to be the first successful attempt to obtain a non-oscillatory interpolation technique for piecewise smooth functions. We present in this section the ENO algorithm and the WENO (weighted ENO) method that improves the basic ENO idea upon several points, both in the one-dimensional case. We then propose a computational complexity analysis of WENO techniques and finally recall a few of the numerous schemes that exploit WENO interpolations to achieve high-order accuracy.

3.2.1 Towards WENO interpolation

Preliminary to the ENO algorithm description, we first present in the sequel a standard conservative polynomial reconstruction method for the one-dimensional case that will also be used in the following chapters.

One-dimensional conservative polynomial reconstruction

Given the cell averages of a function $\phi(x)$:

$$\phi_i = \frac{1}{\Delta x_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \phi(\xi)\, d\xi, \tag{3.8}$$

we look for a polynomial $p_i(x)$ of degree at most $(k-1)$ for each cell $I_i = [x_{i-\frac{1}{2}}; x_{i+\frac{1}{2}}]$ such that:

$$\begin{cases} p_i(x) = \phi(x) + \mathcal{O}\left(\Delta x^k\right) \ \forall x \in I_i & \textit{i.e. } p_i \text{ is a } k\text{-th order accurate approximation of } \phi \text{ inside } I_i, \\ \displaystyle\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} p_i(\xi)\ d\xi = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \phi(\xi)\ d\xi & \textit{i.e. } \text{the cell average of } \phi \text{ over } I_i \text{ is preserved.} \end{cases} \tag{3.9}$$

To determine this polynomial, we first choose a stencil S_i around I_i based on r cells to the left and s cells to the right with $r + s + 1 = k$. S_i writes:

$$S_i = \{I_{i-r},\ \ldots,\ I_{i+s}\}. \tag{3.10}$$

We now consider the following primitive H_i of ϕ:

$$H_i(x) = \int_{x_{i-r-\frac{1}{2}}}^{x} \phi(\xi)\ d\xi.$$

The values of H_i on cell boundaries can easily be calculated, indeed:

$$\begin{aligned} H_i(x_{i-r-\frac{1}{2}}) &= 0, \\ H_i(x_{i-r+\frac{1}{2}}) &= \Delta x_{i-r}\phi_{i-r}, \\ H_i(x_{i-r+\frac{3}{2}}) &= \Delta x_{i-r}\phi_{i-r} + \Delta x_{i-r+1}\phi_{i-r+1}, \\ &\vdots \\ H_i(x_{i+s+\frac{1}{2}}) &= \sum_{j=-r}^{s} \Delta x_{i+j}\phi_{i+j}. \end{aligned}$$

Let P_i be the unique $(k+1)$-th degree Lagrange interpolating polynomial of H_i on the cell boundaries $\{x_{i-r-\frac{1}{2}}, \ldots, x_{i+s+\frac{1}{2}}\}$, *i.e.* the unique $(k+1)$-th degree polynomial that satisfies:

$$P_i\left(x_{i+j-\frac{1}{2}}\right) = H_i\left(x_{i+j-\frac{1}{2}}\right),\ -r \leq j \leq s+1. \tag{3.11}$$

It is well-known that P_i is a $(k+1)$-th order approximation of H_i if H_i is smooth enough over the S_i stencil:

$$P_i(x) = H_i(x) + \mathcal{O}\left(\Delta x^{k+1}\right)\ \forall x \in [x_{i-r-\frac{1}{2}}; x_{i+s+\frac{1}{2}}], \tag{3.12}$$

and we therefore have the following property.

Proposition 1. *The polynomial $p_i = P_i'$ satisfies both accuracy and conservatity conditions given by* (3.9).

Proof. According to (3.12), p_i clearly is a k-th order accurate approximation of ϕ inside I_i. Regarding the conservativity condition, one can notice that:

$$\begin{aligned} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} p_i(\xi)\ d\xi &= \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} P_i'(\xi)\ d\xi, \\ &= P_i\left(x_{i+\frac{1}{2}}\right) - P_i\left(x_{i-\frac{1}{2}}\right). \end{aligned}$$

Relation (3.11) shows that P_i and H_i coincide at $x_{i\pm\frac{1}{2}}$, implying that:

$$\begin{aligned}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} p_i(\xi)\ d\xi &= H_i\left(x_{i+\frac{1}{2}}\right) - H_i\left(x_{i-\frac{1}{2}}\right),\\ &= \Delta x_i \phi_i,\\ &= \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \phi(\xi)\ d\xi,\end{aligned}$$

which shows that the cell average of ϕ over I_i is preserved and thus ends the proof. □

We now describe the ENO algorithm but, for the sake of simplicity, we only consider the case of regular grids in the sequel.

Essentially non-oscillatory (ENO) interpolation

It is well-known that applying the standard reconstruction method on the S_i stencil leads to spurious oscillations when ϕ is not smooth inside S_i. This issue motivates the idea of selecting several candidate stencils (by taking different values for r and s in (3.10)) for performing the reconstruction step described in the previous section and then choosing the one where ϕ presents the fewest discontinuities (or even no discontinuity at all). This is the principle of the ENO procedure detailed hereafter introduced by Harten *et al.* [41]. To determine which stencil is the best suited for computing the polynomial interpolation, one requires a smoothness measurement criterion. In the case of the ENO algorithm on uniform grids, Harten *et al.* propose to resort to undivided differences defined by:

$$\begin{cases}\Phi\left\langle x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right\rangle = \phi_i,\\ \Phi\left\langle x_{i-\frac{1}{2}}, \ldots, x_{i+j+\frac{1}{2}}\right\rangle = \Phi\left\langle x_{i+\frac{1}{2}}, \ldots, x_{i+j+\frac{1}{2}}\right\rangle - \Phi\left\langle x_{i-\frac{1}{2}}, \ldots, x_{i+j-\frac{1}{2}}\right\rangle,\end{cases}$$

which present the following property:

$$\Phi\left\langle x_{i-\frac{1}{2}}, \ldots, x_{i+j+\frac{1}{2}}\right\rangle = \begin{cases}\mathcal{O}\left(\Delta x^{j-1}\right) & \text{if } \phi \text{ is smooth over } [x_{i-\frac{1}{2}}, x_{i+j+\frac{1}{2}}],\\ \mathcal{O}(1) & \text{otherwise.}\end{cases}$$

Note that undivided differences have to be replaced by divided differences when dealing with non-uniform grids, see [19]. We now summarize the ENO procedure.

1. Start with the two point stencil $S_{i,2} = \{x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\}$.

2. For l from 3 to k, assuming that $S_{i,l-1} = \{x_{r+\frac{1}{2}}, \ldots, x_{s-\frac{1}{2}}\}$, add one point to the left or the right of $S_{i,l-1}$ according to the values of undivided differences on both resulting stencils:

$$S_{i,l} = \begin{cases}\{x_{r-\frac{1}{2}}, \ldots, x_{s-\frac{1}{2}}\} & \text{if } \left|\Phi\left\langle x_{r-\frac{1}{2}}, \ldots, x_{s-\frac{1}{2}}\right\rangle\right| < \left|\Phi\left\langle x_{r+\frac{1}{2}}, \ldots, x_{s+\frac{1}{2}}\right\rangle\right|,\\ \{x_{r+\frac{1}{2}}, \ldots, x_{s+\frac{1}{2}}\} & \text{otherwise.}\end{cases}$$

3. Once the right stencil has been chosen, compute the polynomial reconstruction as explained in the previous section.

The resulting reconstruction of ϕ obviously satisfies the accuracy and conservativity conditions given by (3.9), at least theoretically. But despite being theoretically accurate and providing improvements upon classical polynomial reconstruction, ENO techniques practically show several weaknesses.

- ENO interpolation sometimes causes a loss of accuracy on smooth solutions in practical cases despite it theoretically guarantees to provide a high-order accurate polynomial reconstruction. It is indeed very sensitive to round-off errors in such cases and switches therefore frequently from one stencil to another while interpolations on fixed centered stencils would provide better results.
- In order to compute a k-th order accurate reconstruction, ENO techniques require a stencil of $2k-1$ cells but finally only use k values of the considered variable. One could imagine to make use of the information provided by the $k-1$ remaining cells, for example to improve accuracy in smooth regions.
- ENO techniques obviously require several conditional tests that do not fulfill our first principle. This point could be improved by considering another smoothness measurement operator.

Weighted essentially non-oscillatory (WENO) interpolation

The WENO method introduced by Liu *et al.* [58] is an attempt to improve ENO methods on the above-mentioned points. The basic idea is the following: instead of using only one of the candidate stencils to form the reconstruction, one uses a convex combination of all of them. More precisely, suppose that the k candidate stencils

$$S_i^j = \{I_{i-j}, \dots, I_i, \dots, I_{i-1+k-j}\},\ j \in \{0, \dots, k-1\}, \tag{3.13}$$

produce k different reconstructions ϕ_i^j of ϕ_i. WENO reconstruction consists in taking a linear combination of all of them:

$$\phi_i = \sum_{j=0}^{k-1} \omega_j \phi_i^j,$$

with ω_j weights satisfying:

$$\omega_j \geq 0 \quad \text{and} \quad \sum_j \omega_j = 1.$$

The key to success lies in the choice of these weights. The originally proposed coefficients write:

$$\omega_j = \frac{\alpha_j}{\sum_l \alpha_l} \quad \text{with} \quad \alpha_j = \frac{d_j}{(\epsilon + \beta_j)^2}, \tag{3.14}$$

with following notations.

- The d_j are real coefficients that are chosen so that:

$$\sum_{j=0}^{k-1} d_j \phi_i^j = \phi_i + \mathcal{O}\left(\Delta x^{2k-1}\right),$$

  if ϕ is smooth inside all of the candidate stencils in order to improve accuracy in smooth regions;

- β_j are positive smoothness indicators given by:

$$\beta_j = \sum_{l=0}^{k-1} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Delta x^{2l-1} \left(\phi^{(l)}\right)^2 dx \quad \text{where} \quad \phi^{(l)} = \frac{\partial^l \phi}{\partial x^l}.$$

If ϕ is smooth enough over I_i, the $\phi^{(l)}$ spatial derivatives can be estimated with at least first-order accuracy and β_j is of the order of the $l = 0$ term:

$$\beta_j = \mathcal{O}\left(\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Delta x^{-1} \cdot \Delta x^2 \, dx\right) = \mathcal{O}\left(\Delta x^2\right).$$

Otherwise, the $\phi^{(l)}$ spatial derivative is a $\mathcal{O}(1)$ and β_j is of the order of:

$$\beta_j = \mathcal{O}\left(\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Delta x^{-1} \, dx\right) = \mathcal{O}(1).$$

As a matter of fact, the smoothness indicator β_j takes small values in smooth regions and leads to high α_j weights. On the contrary, β_j takes high values in the vicinity of discontinuities, leading to weak weights.

- ϵ is a constant (usually set to 10^{-6}) that prevents from dividing by zero in the definition of α_j, for example when ϕ is constant over the S_j stencil.

All of these coefficients (the d_j coefficients, the β_j smoothness indicators and the polynomial interpolation coefficients in the case of uniform grids) have been summarized up to seventeenth-order in [38]. Note that the d_j coefficients are sometimes chosen not to improve accuracy but robustness, see [5] for example.

3.2.2 Computational complexity analysis of WENO interpolation

The major advantage of WENO methods upon the ENO approach in terms of computational cost is that they do not require any conditional test for selecting one of the candidate stencils. These have indeed been replaced by the estimation of smoothness indicators, making WENO techniques more able to exploit modern processors. They are nevertheless quite expensive since they require to compute several intermediate variables. This is all the more the case when dealing with multidimensional reconstructions since these generally operate on data that are not memory-contiguous and thus do not fulfill the second principle that we stated in introduction (see page 13). Several attempts to decrease the computational cost of WENO techniques can therefore be found in the related literature. Jiang *et al.* [51] and Shu [96] propose for example alternative smoothness measurements. Some progress has also been made in terms of performances by considering different basis of polynomials that allow some algorithmic simplifications (see [3] for example).

3.2.3 Short review of WENO-based high-order schemes

Finite difference and finite volume WENO schemes

In the case of hydrodynamics, WENO techniques have for example been used by Liu *et al.* [58], Jiang and Shu [51] and Balsara and Shu [6] in order to build high-order accurate finite difference schemes. They also have been exploited in finite volume context, see for example works by Balsara [3] and Balsara *et al.* [4] in which the case of structured meshes is discussed whereas Hu and Shu [46] treat the problem of unstructued meshes. In the case of ideal magnetohydrodynamics, we can for example mention works by Jiang and Wu [52] and Tang and Xu [78] who both propose finite difference WENO schemes that enforce the divergence constraint thanks to the projection scheme described in section 5.2.1. Most of these numerical methods resort to Runge-Kutta time-stepping in order to achieve temporal high-order accuracy. They therefore require several boundary conditions and equation of state calls and do not fulfill the third principle that we stated in introduction (see page 14).

Central WENO schemes

WENO techniques are also the main ingredient for building high-order accurate central schemes. These methods appeared more recently than the class of upwind schemes that mainly rely on the approach introduced by Godunov's upwind scheme [40] to deal with hyperbolic systems of conservation laws. They depart from upwind schemes in the sense that they rely neither on a Riemann solver (exact or approximate) nor on characteristics decomposition, which tends to reduce their complexity, especially in the case of multidimensional systems.

Central schemes generalize the idea of the Lax-Friedrichs scheme [37]. We describe them shortly in the sequel in the one-dimensional case. To that end, we consider the following system of conservation laws:

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = \mathbf{0}.$$

Introducing $\mathbf{U}_i^n$ the cell average of $\mathbf{U}$ over the $I_i = [x_{i-\frac{1}{2}}; x_{i+\frac{1}{2}}]$ cell at time t^n, the first step for building high-order accurate central schemes consists in computing a conservative, accurate and non-oscillatory polynomial reconstruction $\mathbf{P}_i(x, t^n)$ of $\mathbf{U}$ in I_i from its cell averages. Originally, Nessyahu and Tadmor [66] used a MUSCL-type interpolant combined with a minmod limiter in order to build $\mathbf{P}_i(x, t^n)$ and thus obtained a second-order accurate central scheme. WENO techniques allow to break the second-order accuracy barrier by providing higher-order interpolants. Once the polynomial reconstruction has been computed, a staggered representation of $\mathbf{U}$ can be evolved according to the following scheme:

$$\begin{aligned}
\mathbf{U}_{i+\frac{1}{2}}^{n+1} &= \frac{1}{\Delta x} \int_{x_i}^{x_{i+1}} \mathbf{P}_i(x, t^n)\, dx + \frac{\Delta t}{\Delta x} \left[\mathbf{F}\left(\mathbf{U}(x_i, t^n)\right) - \mathbf{F}\left(\mathbf{U}(x_{i+1}, t^n)\right) \right], \\
\mathbf{U}_{i}^{n+2} &= \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{P}_{i+\frac{1}{2}}(x, t^{n+1})\, dx + \frac{\Delta t}{\Delta x} \left[\mathbf{F}\left(\mathbf{U}(x_{i-\frac{1}{2}}, t^{n+1})\right) - \mathbf{F}\left(\mathbf{U}(x_{i+\frac{1}{2}}, t^{n+1})\right) \right], \\
\mathbf{U}_{i+\frac{1}{2}}^{n+3} &= \dots
\end{aligned}$$

and so on.

Remark 4. It is possible to reformulate central schemes without loss of accuracy in order to transform them into non-staggered schemes that do not switch between several cell averages representations (see Jiang *et al.* [50]).

Levy *et al.* propose in [54] a high-order accurate central WENO scheme with a Runge Kutta time integrator for one-dimensional systems and they apply it to hydrodynamics. They then extend these works to multidimensional geometries in [55]. The case of central WENO schemes for ideal magnetohydrodynamics is discussed among others by Ziegler [97] and Li [57] that both resort to Runge Kutta time schemes but nevertheless propose different strategies. On the one hand, Ziegler proposes a fully non-staggered scheme that enforces the divergence constraint thanks to the constrained transport method presented in section 5.2.2. On the other hand, Li builds a staggered high-order central scheme that relies on both constrained transport and divergence-free reconstruction (see section 5.2.2) methods to impose $\nabla \cdot \mathbf{B} = 0$. In this case, the constrained transport algorithm operates on the interface centered representation of the magnetic field whereas the divergence-free reconstructions are applied to cell-centered representations.

3.3 Arbitrary Derivative Riemann Problem (ADER) schemes

3.3.1 Detailed ADER methodology

The ADER approach has been originally introduced by Toro *et al.* [86, 87]. It denotes a class of finite volume schemes that reach high-order accuracy in both space and time by combining Godunov's

upwind approach [40] and the so-called Cauchy-Kovalevskaya procedure. ADER schemes have firstly been developed for one-dimensional non-linear systems of conservation laws [82] and have then been applied to multidimensional scalar equations [84]. The case of non-linear multidimensional systems is studied in [85]. In this section, we briefly describe principles of ADER schemes for non-linear one-dimensional systems of equations. In this context, the finite volume discretization of the following one-dimensional hyperbolic system of conservation laws:

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = \mathbf{0}, \tag{3.15}$$

is given by:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2}}\right). \tag{3.16}$$

Here, $\mathbf{U}_i^n$ denotes the cell average of the solution at time t^n and $\mathbf{F}_{i+\frac{1}{2}}$ the time average of the physical flux accross the cell interface located at $x = x_{i+\frac{1}{2}}$:

$$\mathbf{F}_{i+\frac{1}{2}} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \mathbf{F}(\mathbf{U}(x_{i+\frac{1}{2}}, t)\, dt. \tag{3.17}$$

The first step in ADER fluxes evaluation consists in computing point-wise values of the solution at time t^n via high-order polynomials, more precisely using WENO reconstructions that prevent the generation of spurious oscillations on discontinuities. After this reconstruction step, the solution is represented by vectors $\mathbf{p}_i(x)$ of polynomials in each cell. Then the following generalized Riemann problem has to be solved at each cell interface:

$$\begin{aligned} &\text{PDE:} \quad \partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = \mathbf{0}, \\ &\text{IC:} \quad \mathbf{U}(x,0) = \begin{cases} \mathbf{p}_i(x) & x < x_{i+\frac{1}{2}}, \\ \mathbf{p}_{i+1}(x) & x > x_{i+\frac{1}{2}}, \end{cases} \end{aligned} \tag{3.18}$$

in order to compute an r-th order accurate approximate solution for the interface state $\mathbf{U}(x_{i+\frac{1}{2}}, \tau)$ ($\tau = t - t^n$ is the local time) appearing in (3.17). To that end, a semi-analytical method has been developed [83]. It consists in solving r conventional Riemann problems (namely one non-linear and $(r-1)$ linear problems) instead of solving the generalized Riemann problem (3.18). It relies on the following truncated Taylor expansion in time:

$$\mathbf{U}(x_{i+\frac{1}{2}}, \tau) = \mathbf{U}(x_{i+\frac{1}{2}}, 0) + \sum_{k=1}^{r-1} \frac{\tau^k}{k!} \frac{\partial^k \mathbf{U}}{\partial t^k}(x_{i+\frac{1}{2}}, 0). \tag{3.19}$$

The leading term $\mathbf{U}(x_{i+\frac{1}{2}}, 0)$ is obtained by solving the following non-linear conventional Riemann problem:

$$\begin{aligned} &\text{PDE:} \quad \partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = \mathbf{0}, \\ &\text{IC:} \quad \mathbf{U}(x,0) = \begin{cases} \mathbf{p}_i(x_{i+\frac{1}{2}}) & x < x_{i+\frac{1}{2}}, \\ \mathbf{p}_{i+1}(x_{i+\frac{1}{2}}) & x > x_{i+\frac{1}{2}}. \end{cases} \end{aligned} \tag{3.20}$$

This can be achieved using any exact or approximate Riemann solver. The higher-order terms are evaluated in two steps. First, the time-derivatives are replaced by spatial derivatives using the Cauchy-Kovalevskaya procedure which is illustrated below. Starting from:

$$\partial_t \mathbf{U} = -\partial_x \mathbf{F}(\mathbf{U}) = -\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right) \partial_x \mathbf{U}, \tag{3.21a}$$

one can determine an expression of $\partial_{tx}\mathbf{U}$ using (3.21a):

$$\partial_{tx}\mathbf{U} = \partial_x\left(\partial_t\mathbf{U}\right) = -\left(\frac{\partial^2\mathbf{F}}{\partial\mathbf{U}^2}\right)\left(\partial_x\mathbf{U}\right)^2 - \left(\frac{\partial\mathbf{F}}{\partial\mathbf{U}}\right)\partial_{xx}\mathbf{U}. \tag{3.21b}$$

Combining relations (3.21a) and (3.21b) then allows to rewrite $\partial_{tt}\mathbf{U}$ without time-derivatives:

$$\partial_{tt}\mathbf{U} = \partial_t\left(\partial_t\mathbf{U}\right) = -\left(\frac{\partial^2\mathbf{F}}{\partial\mathbf{U}^2}\right)\left(\partial_t\mathbf{U}\right)\left(\partial_x\mathbf{U}\right) - \left(\frac{\partial\mathbf{F}}{\partial\mathbf{U}}\right)\partial_{xt}\mathbf{U}, \tag{3.21c}$$

and so on. Spatial derivatives that appear once the Cauchy-Kovalevskaya procedure has been applied have to be computed at $x = x_{i+\frac{1}{2}}$ in order to evaluate the Taylor expansion of $\mathbf{U}(x_{i+\frac{1}{2}},\tau)$ given in (3.19). To that end, one has to consider evolution equations of the spatial derivatives:

$$\mathbf{U}^{(k)} = \frac{\partial^k\mathbf{U}}{\partial x^k},\ k \in \{1,\dots,r-1\},$$

that can be derived from (3.15):

$$\partial_t\mathbf{U}^{(k)} + \mathbf{A}\partial_x\mathbf{U}^{(k)} = \mathbf{H}\left(\mathbf{U},\mathbf{U}^{(1)},\dots,\mathbf{U}^{(k-1)}\right), \tag{3.22}$$

where $\mathbf{H}$ is a non-linear source term that depends on lower-order derivatives. Since spatial derivatives are evaluated at the local time $\tau = 0$, the influence of this source is neglected. Equation (3.22) is then linearized around the leading term $\mathbf{U}(x_{i+\frac{1}{2}},0)$, the $\mathbf{U}^{(k)}$ derivatives can finally be computed by solving the following linear conventional Riemann problems:

$$\begin{aligned}
&\text{PDE:}\quad \partial_t\mathbf{U}^{(k)} + \mathbf{A}_{i+\frac{1}{2}}\partial_x\mathbf{U}^{(k)} = \mathbf{0},\ \mathbf{A}_{i+\frac{1}{2}} = \mathbf{A}\left(\mathbf{U}(x_{i+\frac{1}{2}},0)\right),\\
&\text{IC:}\quad \mathbf{U}^{(k)} = \begin{cases}\mathbf{p}_i^{(k)}(x_{i+\frac{1}{2}}) & x < x_{i+\frac{1}{2}},\\ \mathbf{p}_{i+1}^{(k)}(x_{i+\frac{1}{2}}) & x > x_{i+\frac{1}{2}}.\end{cases}
\end{aligned} \tag{3.23}$$

At this stage, all the terms of the Taylor expansion (3.19) have been estimated and the numerical flux given by (3.17) now has to be computed so that the conservative variables can be updated according to the finite volume scheme (3.16). To that end, two options are available. The first option called *state-expansion* ADER consists in discretizing the time integral in (3.17) using a r-th order accurate quadrature rule:

$$\mathbf{F}_{i+\frac{1}{2}} \approx \frac{1}{\Delta t}\sum_{l=1}^{L}\omega_l\mathbf{F}(\mathbf{U}(x_{i+\frac{1}{2}},\tau_l),$$

where τ_l denotes the quadrature points along $[t^n;t^{n+1}]$ and ω_l the associated weights. The implementation of state-expansion ADER fluxes is straightforward: one simply has to apply several times the above-described ADER methodology with $\tau = \tau_l$ for $1 \le l \le L$. The second strategy called *flux-expansion* ADER replaces the quadrature rule with a method that is similar to the one used for the computation of $\mathbf{U}(x_{i+\frac{1}{2}},\tau)$. A Taylor expansion in time is performed on the physical flux at $x = x_{i+\frac{1}{2}}$ leading to the following high-order accurate approximate flux term:

$$\mathbf{F}_{i+\frac{1}{2}} \approx \mathbf{F}(x_{i+\frac{1}{2}},0) + \sum_{k=1}^{r-1}\frac{\Delta t^k}{(k+1)!}\left(\frac{\partial^k\mathbf{F}}{\partial t^k}\right)(x_{i+\frac{1}{2}},0).$$

The leading term $\mathbf{F}(x_{i+\frac{1}{2}},0)$ is computed as a certain monotone flux of the conventional Riemann problem (3.20) whereas time-derivatives of the flux are expressed via time-derivatives of the intercell state $\mathbf{U}(x_{i+\frac{1}{2}},0)$ which are known at this stage from (3.23). No numerical quadrature is required in this case.

3.3.2 Computational complexity analysis

The most expensive part of ADER schemes in terms of computational cost is the high-order reconstruction of point-wise values of the solution that resorts to WENO techniques. As we have shown in section 3.2, this step requires numerous computations (polynomial interpolation, smoothness indicators and weights on several stencils). The complexity of WENO techniques also increases noticeably in the multidimensional case since these (even more) numerous computations are performed on data that are not memory-contiguous. In [85], the authors claim that this step takes about 60% of the total computational time. Switching time-derivatives into spatial ones using the Cauchy-Kovalevskaya procedure can also reveal very costly when dealing with high-order accurate simulations in multidimensional geometries. We will indeed see in section 4.1.2 that this step leads to heavy algebraic calculations that have to be properly performed so that the resulting solver is able to take advantage of modern processors. On the contrary, the resolution of Riemann problems is said to be relatively cheap (about 5% of the total computational time according to [85]). Regarding the two flux computation options, one should favour the flux-expansion approach since state-expansion ADER schemes resort to subcycling for the time discretization which require several equation of state calls and do not fulfill the third principle that we stated in introduction (see page 14).

3.3.3 Short review of ADER schemes

We have seen in section 3.1 that Dumbser and Munz exploit ADER numerical fluxes in [34] in a Discontinuous Galerkin framework. ADER techniques are used in this case to compute numerical fluxes at cell interfaces but the authors propose a more efficient and generic way of performing the Cauchy-Kovalevskaya procedure that relies on a generalized Leibniz rule. Further developments have been performed by Balsara *et al.* [5] who propose ADER schemes for ideal MHD with divergence-free reconstruction of the magnetic field. Contrarily to original ADER methods, this new class of schemes does not resort anymore to the Cauchy-Kovalevskaya procedure but to a continuous Galerkin formulation for both space and time discretizations. They also provide major improvements regarding the WENO reconstruction which is made less expensive thanks to an appropriate choice of polynomials and algorithmic simplifications, see [4]. The resulting schemes prove to be accurate for both smooth and non-smooth problems and competitive in terms of computational time.

3.4 Conclusions on the state of the art high-order schemes

We conclude this state of the art for high-order schemes for hydrodynamics and ideal MHD with a few comments about the different techniques that have been described in this chapter and their appropriateness to the three principles stated in introduction.

ENO and WENO techniques are widely used to achieve high-order accuracy but we will not follow this path in our study. Indeed, ENO methods do not fulfill the first principle that we stated in introduction (see page 13) which imposes to avoid conditional tests and WENO interpolations reveal quite costly as shown in [85]. We will therefore restrict to less expensive centered polynomial reconstructions in the sequel and try to control oscillations with the help of artificial viscosity techniques.

Multistep time integration techniques have to be avoided in the context of massively parallel computer architectures since they do not fulfill the third principle that we stated page 14 and require additional calls to the equation of state (see remark 2 page 14) compared to one-step methods. In this study, we will therefore

adopt the approach presented in the previous section that consists in applying the Cauchy-Kovalevskaya procedure to remove time-derivatives from approximate fluxes and thus build direct temporal schemes.

A noticeable drawback of the high-order methods we have studied in this chapter is that none of these can be easily implemented in a cache-oblivious way. Indeed, they all rely on multidimensional computations that do not operate naturally on contiguous data. We propose in the sequel an innovative approach based on dimensional splitting techniques that aims at improving this point. We will indeed see in section 4.2 that dimensionally split schemes can be implemented in such a way that they mostly operate on contiguous datasets and thus respect the second principle that we stated in introduction (see page 13).

Chapter 4

High-order dimensionally split Lagrange-remap schemes for compressible hydrodynamics in planar geometry

We propose in this chapter a new class of finite volume numerical schemes on Cartesian meshes for solving the compressible hydrodynamics system of equations which writes:

$$\partial_t \begin{pmatrix} \rho \\ \rho \mathbf{u} \\ \rho e \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + p \underline{\mathbf{I}} \\ (\rho e + p)\mathbf{u} \end{pmatrix} = \mathbf{0}, \tag{4.1}$$

where ρ denotes the density, $\mathbf{u}$ the velocity, ρe the total energy, p the pressure and $\underline{\mathbf{I}}$ the identity tensor. Built in the Lagrange-remap formalism, these so-called GoHy schemes (for *Godunov Hybrid*) - which have been firstly introduced in a purely Lagrangian framework [45] - are high-order accurate in space and time in the non-linear regime. To our knowledge, they are the first successful attempt to achieve high-order accuracy in the Lagrange-remap formalism. GoHy schemes have been designed to take advantage of modern computer architectures, which is an innovative feature compared to existing high-order accurate methods. It can be illustrated by several technical choices that we detail below.

First of all, like most highly accurate techniques, GoHy schemes resort to high-order polynomial interpolations and are thus subject to unphysical Gibbs oscillations that must be efficiently controlled in order to provide good robustness properties. Existing methods usually rely either on slope limiters or on WENO techniques (that we presented in section 3.2) but none of these strategies reveals satisfying, especially in terms of performance.

- Limiters reduce accuracy in the vicinity of discontinuities and their implementation resorts to many conditional tests. Such techniques therefore do not fulfill the first principle for designing efficient numerical schemes that we stated in introduction (see page 13).

- The WENO approach requires the computation of several intermediate variables (high-order interpolants on different stencils, smoothness indicators, weights) that makes them very expensive.

GoHy schemes depart from existing high-order accurate methods in the sense that oscillations are controlled thanks to hyperviscosity models inspired from previous works [20] in the area of large eddy simulations (LES). Hyperviscosity techniques basically consist in adding an artificial diffusion term that is proportional to a high exponent of the space step and thus quickly tends to zero when the mesh is refined. The accuracy of the underlying scheme is consequently preserved. This approach suffers a few drawbracks - it involves for example several parameters that are test problem dependent and must be adjusted manually - but it also presents major advantages in terms of performance. Indeed, hyperviscosity models are completely free of conditional tests and reveal quite cheap. They therefore represent a worthy alternative to limiting techniques and WENO-type methods.

Another original feature of GoHy schemes lies in their multidimensional extension which is achieved by combining an efficient one-dimensional scheme with an high-order accurate dimensional splitting technique. Dimensional splitting is a well-known method for extending one-dimensional schemes to multidimensional geometries but existing split schemes usually rely on the classical Strang splitting [77] and are thus at most second-order accurate. This study goes beyond by providing up to sixth-order accurate splitting sequences, some of those being taken from the literature on symplectic integrators [36, 61, 13, 95]. The dimensionally split approach presents several benefits over unsplit methods.

- Firstly, dimensional splitting allows to design schemes that operate on contiguous data. Indeed, instead of working on multidimensional datasets, dimensionally split methods operate on one-dimensional datasets and may therefore be implemented in a cache-oblivious way. In this sense, they are compatible with the second principle that we stated in introduction (see page 13).

- Secondly, dimensional splitting techniques are well-suited to parallelization. Indeed, dimensionally split schemes solve a given set of equations on several one-dimensional datasets of the computational domain along a particular direction. An important point here is that all of these resolution steps are independent: they can therefore be performed simultaneously, allowing a parallel implementation. This feature is nevertheless not exploited in our case. As we will see in section 4.4, the solver that we developed has been parallelized using a domain decomposition method that differs from the above-described parallel programming model but both approaches could be mixed together. Practically, this would lead to an hybrid programming model that combines a multiprocess strategy for the domain decomposition and a multithreaded treatment of one-dimensional datasets.

- Thirdly, dimensional splitting significantly simplifies the implementation of multidimensional numerical schemes. Practically, dimensionally split codes simply rely on a function that implements the one-dimensional scheme and on a procedure that applies it along each direction of the considered geometry.

The outline of this chapter is the following. We first present the one-dimensional numerical scheme that has been developed and describe its extension to the two-dimensional case using dimensional splitting techniques. We then detail an artificial viscosity method that aims at controlling Gibbs oscillations using high-order hyperviscosity (as indicated previously) and to preserve symmetry for relevant problems thanks to a first-order artificial viscosity model. In order to show that our schemes are well-suited for modern computers, we then provide a few performance and parallel efficiency measurements. This chapter will be concluded with numerical results on smooth and non-smooth test problems taken from the related literature in order to check the practical accuracy and robustness of GoHy schemes. The GoHy schemes presented in this chapter have been published in [32].

4.1 One-dimensional high-order Lagrange-remap schemes

We first focus on the one-dimensional Euler system of equations which writes:

$$\partial_t \begin{pmatrix} \rho \\ \rho u \\ \rho e \end{pmatrix} + \partial_x \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho e u + p u \end{pmatrix} = \mathbf{0}. \tag{4.2}$$

Note that we drop the velocity subscript for the one-dimensional case since there is no ambiguousness: u denotes in this section the velocity component along the x-direction. Building numerical schemes in the Lagrange-remap formalism consists in solving the considered system of equations in two steps.

1. **Lagrangian step**. System (4.2) is first solved in Lagrangian coordinates between t^n and t^{n+1}. To that end, we introduce in section 4.1.1 the $(x,t) \rightarrow (X,t)$ variable change from Eulerian coordinates x to Lagrangian coordinates X. We then rewrite the Eulerian system (4.2) in Lagrangian coordinates and describe the construction of second, third and fourth-order accurate numerical schemes in section 4.1.2.

2. **Remap step**. At the end of the Lagrangian step, we have at our disposal a set of updated Lagrangian conservative variables on a non-uniform grid. These are remapped on the initial Cartesian grid using a conservative high-order accurate polynomial reconstruction which is detailed in section 4.1.3.

4.1.1 The Lagrangian variable change

In this section, we introduce the Lagrangian variable change and provide a general framework for writing hyperbolic systems of conservation laws in Lagrangian coordinates that we finally apply to the one-dimensional Euler system of equations (4.2). The $(x,t) \rightarrow (X,t)$ variable change given by:

$$dx(X,t) = J(X,t)\, dX + u(X,t)\, dt, \tag{4.3}$$

where J denotes the associated Jacobian:

$$J(X,t) = \partial_X x(X,t). \tag{4.4}$$

From (4.3), one can notice that the Eulerian coordinate x satisfies:

$$\partial_t x(X,t) = u(X,t).$$

We choose for this evolution equation the initial condition $x(X,0) = X$ which implies that Cartesian and Lagrangian meshes initially match and in particular that $\Delta x = \Delta X$. This choice also implies that $J(X,0) = 1$. Differentiating equation (4.4) with respect to time leads to the well-known Piola identity:

$$\partial_t J(X,t) = J(x,t) \cdot \partial_x u(x,t). \tag{4.5}$$

Using these results, one gets the following relation between Eulerian and Lagrangian time-derivatives for any function ϕ:

$$\partial_t \phi(X,t) = \left[\partial_t \phi + u \cdot \partial_x \phi\right](x,t), \tag{4.6}$$

and the following one between spatial derivatives:

$$\partial_X \phi(X,t) = \left[J \cdot \partial_x \phi\right](x,t). \tag{4.7}$$

Combining equations (4.5) and (4.6), one can show that:

$$\partial_t(J\phi)(X,t) = \left[J \cdot (\partial_t\phi + \partial_x(\phi u))\right](x,t). \tag{4.8}$$

Finally, adding equations (4.7) and (4.8) allows to state the following lemma.

Lemma 1. *For any functions ϕ and ψ, one has the following relation between Eulerian and Lagrangian derivatives:*

$$\left[\partial_t(J\phi) + \partial_X\psi\right](X,t) = \left[J \cdot (\partial_t\phi + \partial_x(\phi u + \psi))\right](x,t).$$

We now consider the four (ϕ, ψ) couples:

$$(\rho,\ 0),\ (\rho u,\ p),\ (\rho e,\ pu),\ (1,\ -u). \tag{4.9}$$

The one-dimensional Euler equations in Eulerian coordinates (4.2) write under the form $\partial_t\phi + \partial_x(\phi u + \psi) = 0$ with the three first couples whereas the last one corresponds to the trivial equation $\partial_t(1) + \partial_x(0) = 0$. Applying Lemma 1 with $(\phi, \psi) = (\rho, 0)$ leads to:

$$\partial_t(J\rho) = 0$$

which implies that:

$$(J\rho)(X,t) = (J\rho)(X,0) = \rho(X,0) = \rho_0(X) \text{ since } J(X,0) = 1.$$

and thus:

$$J = \rho_0/\rho \text{ where } \rho_0 \text{ denotes the density at time } t^n.$$

Using this result and applying Lemma 1 to the three other couples leads to the one-dimensional Euler equations in Lagrangian coordinates:

$$\partial_t \begin{pmatrix} \rho_0\tau \\ \rho_0 u \\ \rho_0 e \end{pmatrix} + \partial_X \begin{pmatrix} -u \\ p \\ pu \end{pmatrix} = \mathbf{0}, \tag{4.10}$$

where $\tau = 1/\rho$ denotes the specific volume.

Remark 5. Note that building schemes in the Lagrange-remap formalism consists in treating separately the advection part and the effect of pressure forces. The Lagrangian step only takes the latter into account and, as a matter of fact, variables which satisfy pure advection equations should only be updated during the remap step.

4.1.2 Lagrangian step

This section aims at describing the high-order numerical schemes that have been developed for solving the Lagrangian system of equations (4.10). To that end, it is more convenient to rewrite it under the following general form:

$$\partial_t\mathbf{U} + \partial_X\mathbf{F}(\mathbf{U}) = 0 \quad \text{with} \quad \mathbf{U} = \begin{pmatrix} \rho_0\tau \\ \rho_0 u \\ \rho_0 e \end{pmatrix} \quad \text{and} \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} -u \\ p \\ pu \end{pmatrix}. \tag{4.11}$$

Since we would like to solve system (4.11) in finite volume context, we integrate it over a control volume $[X_{i-\frac{1}{2}}; X_{i+\frac{1}{2}}]\times[t^n; t^{n+1}]$ and then divide the result by Δt and ΔX. These manipulations lead to the following well-known finite volume scheme:

$$\frac{\overline{\mathbf{U}}_i^{n+1} - \overline{\mathbf{U}}_i^n}{\Delta t} + \frac{\mathbf{F}^*_{i+\frac{1}{2}} - \mathbf{F}^*_{i-\frac{1}{2}}}{\Delta X} = 0, \tag{4.12}$$

where $\overline{\mathbf{U}}_i^n$ denotes the cell average of $\mathbf{U}$ over $[X_{i-\frac{1}{2}}; X_{i+\frac{1}{2}}]$ at time t^n:

$$\overline{\mathbf{U}}_i^{n+1} = \frac{1}{\Delta x}\int_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}} \mathbf{U}(X, t^n)\ dX,$$

and $\mathbf{F}^*_{i+\frac{1}{2}}$ the Lagrangian flux at the $X_{i+\frac{1}{2}}$ cell interface:

$$\mathbf{F}^*_{i+\frac{1}{2}} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \mathbf{F}(\mathbf{U})(X_{i+\frac{1}{2}}, \theta)\ d\theta. \tag{4.13}$$

Building an Nth-order scheme requires to approximate these fluxes at desired order. To that end, we first perform an $(N-1)$th-order Taylor expansion in time of the $\mathbf{F}(\mathbf{U})(X_{i+\frac{1}{2}}, \theta)$ term in (4.13):

$$\mathbf{F}(\mathbf{U})(X_{i+\frac{1}{2}}, \theta) = \sum_{k=0}^{N-1} \frac{(\theta - t^n)^k}{k!}\left(\frac{\partial^k \mathbf{F}(\mathbf{U})}{\partial t^k}\right)(X_{i+\frac{1}{2}}, t^n) + \mathcal{O}\left((\theta - t^n)^N\right),$$

and thus obtain the following approximation of $\mathbf{F}^*_{i+\frac{1}{2}}$:

$$\mathbf{F}^{*,N}_{i+\frac{1}{2}} = \sum_{k=0}^{N-1} \frac{\Delta t^k}{(k+1)!}\left(\frac{\partial^k \mathbf{F}(\mathbf{U})}{\partial t^k}\right)(X_{i+\frac{1}{2}}, t^n) = \mathbf{F}^*_{i+\frac{1}{2}} + \mathcal{O}\left(\Delta t^N\right). \tag{4.14}$$

Note that the approximate flux (4.14) guarantees high-order accuracy in time provided that its computation does not require any information from a time $t > t^n$. This is of course not the case in (4.14) due to the presence of time-derivatives. Fortunately, time-derivatives can be removed using the so-called Cauchy-Kovalevskaya procedure that we briefly presented in section 3.3. Indeed, this technique allows to replace time-derivatives with a non-linear combination of space-derivatives and thus to form an expression of the approximate flux (4.14) that only relies on estimations of quantities which can be computed at time t^n. These manipulations are described more precisely in the sequel. Once the Cauchy-Kovalevskaya procedure has been applied, quantities involved in (4.14) have to be evaluated at cell interfaces. This is achieved in three steps.

1. We first reconstruct cell-centered point-wise values of conservative variables using a centered conservative polynomial reconstruction. More precisely, for $\phi \in \{1, u, e\}$, we compute:

$$\begin{aligned}(\rho_0\phi)_i^n &= (\rho_0\phi)(X_i, t^n) + \mathcal{O}\left(\Delta X^N\right),\\ &= c_0^N \cdot \overline{(\rho_0\phi)}_i^n + \sum_{k=1}^{r} c_k^N \cdot \left(\overline{(\rho_0\phi)}_{i+k}^n + \overline{(\rho_0\phi)}_{i-k}^n\right),\end{aligned} \tag{4.15}$$

   where $r = \lfloor N/2 \rfloor$ (floor function), the c_k^N coefficients being given in Table 4.1. Note that this step is not necessary if $N < 3$ as explained in the sequel (see remark 6).

2. The previous step now allows us to compute cell-centered point-wise values of each primitive variable without loss of accuracy. This would not have been possible directly from cell averages since the computation of primitive variables generally resorts to non-linear arithmetic operations. Assume for example that we would like to compute cell-centered values of u. This can be done without loss of accuracy using the following formula that operates on point-wise values:

$$u_i^n = (\rho_0 u)_i^n \,/\, (\rho_0)_i^n$$

but not from cell averages (see remark 6 below):

$$\overline{u}_i^n = \overline{(\rho_0 u)}_i^n \Big/ \overline{(\rho_0)}_i^n + \mathcal{O}\left(\Delta x^2\right) \neq \overline{(\rho_0 u)}_i^n \Big/ \overline{(\rho_0)}_i^n \,.$$

3. We finally compute high-order accurate estimations of quantities involved in (4.14) on cell interfaces using centered finite difference formulae. Note that the effectively required accuracy varies for each term depending on *where* it appears in the truncated Taylor expansion. For example, quantities involved in the l-th degree term only need to be estimated at $(N-l)$-th order due to the presence of an Δt^l factor. We therefore introduce:

$$N_{\text{eff}} = (N-l) + \left[(N-l) \bmod 2\right].$$

Note that since we only resort to centered formulae, the N_{eff} order of accuracy is necessarily even. These finite difference operators write:

$$\begin{aligned}
\left(\frac{\partial^m \psi}{\partial x^m}\right)_{i+\frac{1}{2}}^n &= \left(\frac{\partial^m \psi}{\partial x^m}\right)\left(X_{i+\frac{1}{2}}, t^n\right) + \mathcal{O}\left(\Delta X^{N_{\text{eff}}}\right) \\
&= \begin{cases} \dfrac{1}{\Delta x^m} \displaystyle\sum_{k=1}^{s} d_{m,k}^{N_{\text{eff}}} \cdot \left(\psi_{i+k}^n + \psi_{i-k+1}^n\right) & \text{if } m \text{ is even,} \\ \dfrac{1}{\Delta x^m} \displaystyle\sum_{k=1}^{s} d_{m,k}^{N_{\text{eff}}} \cdot \left(\psi_{i+k}^n - \psi_{i-k+1}^n\right) & \text{if } m \text{ is odd,} \end{cases}
\end{aligned} \tag{4.16}$$

where $s = \lceil (N_{\text{eff}} + m)/2 \rceil$ (ceiling function), the $d_{m,k}^{N_{\text{eff}}}$ coefficients being given in Table 4.2.

Remark 6. In the case of smooth functions, cell averages are a second-order approximation of cell-centered values:

$$\begin{aligned}
\frac{1}{\Delta x}\int_{x_{i+\frac{1}{2}}}^{x_{i+\frac{1}{2}}} f(x)\, dx &= \frac{1}{\Delta x}\int_{x_{i+\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left(f(x_i) + (x-x_i)f'(x_i) + \mathcal{O}\left(\Delta x^2\right)\right)\, dx, \\
&= f(x_i) + \frac{1}{\Delta x}\left[\frac{(x-x_i)^2}{2}\right]_{x_{i+\frac{1}{2}}}^{x_{i+\frac{1}{2}}} f'(x_i) + \mathcal{O}\left(\Delta x^2\right), \\
&= f(x_i) + \frac{f'(x_i)}{\Delta x}\left(\frac{\Delta x^2}{8} - \frac{\Delta x^2}{8}\right) + \mathcal{O}\left(\Delta x^2\right), \\
&= f(x_i) + \mathcal{O}\left(\Delta x^2\right).
\end{aligned}$$

The above-detailed first step consequently does not need to be performed if $N \leq 3$.

To complete the Lagrangian step, positions of the Eulerian grid have to be updated. We have indeed seen in section 4.1.1 that the Eulerian coordinate x satisfies the evolution equation:

$$\partial_t x = u.$$

	$k=0$	$k=1$	$k=2$	$k=3$
$r=1$	$\frac{13}{12}$	$\frac{1}{24}$		
$r=2$	$\frac{1067}{960}$	$-\frac{29}{480}$	$\frac{3}{640}$	
$r=3$	$\frac{30251}{26880}$	$-\frac{7621}{107520}$	$\frac{159}{17920}$	$-\frac{5}{7168}$

Table 4.1: Coefficients c_k^N used in (4.15)

	$\mathbf{N_{eff}=2}$			$\mathbf{N_{eff}=4}$			$\mathbf{N_{eff}=6}$		
	$k=1$	$k=2$	$k=3$	$k=1$	$k=2$	$k=3$	$k=1$	$k=2$	$k=3$
$m=0$	$1/2$			$9/16$	$-1/16$		$75/128$	$-25/256$	$3/256$
$m=1$	1			$9/8$	$-1/24$		$75/64$	$-25/384$	$3/640$
$m=2$	$-1/2$	$1/2$		$-17/24$	$13/16$	$-5/48$			
$m=3$	-3	1		$-17/4$	$13/8$	$-1/8$			
$m=4$	1	$-3/2$	$1/2$						
$m=5$	10	-5	1						

Table 4.2: Coefficients $d_{m,k}^{N_{\text{eff}}}$ used in (4.16).

Integrating it on $[t^n;t^{n+1}]$ with $X=X_{i+\frac{1}{2}}$ leads to:

$$x_{i+\frac{1}{2}}^{n+1}-x_{i+\frac{1}{2}}=\int_{t^n}^{t^{n+1}}u(X_{i+\frac{1}{2}},\theta)\,d\theta.$$

High-order accurate position of the Eulerian cell interfaces at time t^{n+1} can therefore be computed using the following scheme:

$$x_{i+\frac{1}{2}}^{n+1}=x_{i+\frac{1}{2}}+u_{i+\frac{1}{2}}^{*,N}\Delta t \tag{4.17}$$

since it relies on the high-order accurate numerical flux $u^{*,N}$.

We now describe the practical Lagrangian fluxes construction (in particular how to apply the Cauchy-Kovalevskaya procedure) for achieving second, third and fourth-order accuracy.

Second-order accurate Lagrangian fluxes

The second-order accurate Lagrangian flux is given by:

$$\mathbf{F}_{i+\frac{1}{2}}^{*,2}=\mathbf{F}(X_{i+\frac{1}{2}},t^n)+\frac{\Delta t}{2}\left(\frac{\partial\mathbf{F}(\mathbf{U})}{\partial t}\right)(X_{i+\frac{1}{2}},t^n).$$

We thus have to determine expressions of the first time-derivatives of u and p. The time-derivative of u can easily be derived from system (4.11), indeed:

$$\partial_t(\rho_0 u) + \partial_X p = 0 \;\Rightarrow\; \partial_t u = -\frac{1}{\rho_0}\partial_X p.$$

The time-derivative of p could be obtained by differentiating the equation of state but there is a more generic way (that has been firstly proposed in [45]) to determine it as shown below.

Proposition 2. *For any equation of state, the pressure p satisfies the evolution equation:*

$$\partial_t p = -\frac{(\rho c)^2}{\rho_0}\partial_X u$$

where $(\rho c)^2 = \gamma p/\tau$ denotes the Lagrangian sound speed, the ratio of specific heats γ being defined by:

$$\gamma = -\frac{\tau}{p}\left.\frac{\partial p}{\partial \tau}\right|_S.$$

Proof. The second law of thermodynamics writes $TdS = d\epsilon + pd\tau$ and implies that the entropy S is not time-dependent for smooth solutions in the Lagrangian framework:

$$\begin{aligned}\rho_0 T\partial_t S &= \partial_t(\rho_0\epsilon) + p\partial_t(\rho_0\tau)\\ &= \partial_t(\rho_0 e) - u\partial_t(\rho_0 u) + p\partial_t(\rho_0\tau)\\ &= -\partial_X(pu) + u\partial_X p + p\partial_X u\\ &= 0.\end{aligned}$$

To be thermodynamically consistant, the equation of state has to satisfy (see [63] for more details):

$$dp = -(\rho c)^2 d\tau + \frac{\Gamma T}{\tau}dS,$$

where Γ denotes the Grüneisen coefficient defined by:

$$\Gamma = -\frac{\tau}{T}\left.\frac{\partial T}{\partial \tau}\right|_S$$

The time-derivative of p therefore writes:

$$\begin{aligned}\partial_t p &= -(\rho c)^2\partial_t\tau + \frac{\Gamma T}{\tau}\partial_t S\\ &= -\frac{(\rho c)^2}{\rho_0}\partial_X u \text{ since } \partial_t S = 0 \text{ and } \partial_t(\rho_0\tau) = \partial_X u.\end{aligned}$$

□

We are now able to form the second-order accurate numerical fluxes u^* and p^* (for the sake of simplicity, we drop the time step superscript, every variable being evaluated at time t^n):

$$\begin{aligned}u^{*,2}_{i+\frac{1}{2}} &= \frac{u_i + u_{i+1}}{2} - \frac{\Delta t}{(\rho_0)_i + (\rho_0)_{i+1}}\left(\frac{p_{i+1} - p_i}{\Delta x}\right),\\ p^{*,2}_{i+\frac{1}{2}} &= \frac{p_i + p_{i+1}}{2} - \frac{\Delta t}{2}\left(\frac{(\rho c)_i^2 + (\rho c)_{i+1}^2}{(\rho_0)_i + (\rho_0)_{i+1}}\right)\left(\frac{u_{i+1} - u_i}{\Delta x}\right),\\ (pu)^{*,2}_{i+\frac{1}{2}} &= p^{*,2}_{i+\frac{1}{2}}\cdot u^{*,2}_{i+\frac{1}{2}}.\end{aligned}$$

Note that computing $(pu)^*$ by multiplying p^* and u^* is accurate enough for building second-order schemes. We will see that this term must be treated more carefully at higher orders of accuracy.

Third-order accurate Lagrangian fluxes

The third-order accurate Lagrangian flux writes:

$$\mathbf{F}^{*,3}_{i+\frac{1}{2}} = \mathbf{F}(X_{i+\frac{1}{2}}, t^n) + \frac{\Delta t}{2}\left(\frac{\partial \mathbf{F}(\mathbf{U})}{\partial t}\right)(X_{i+\frac{1}{2}}, t^n) + \frac{\Delta t^2}{6}\left(\frac{\partial^2 \mathbf{F}(\mathbf{U})}{\partial t^2}\right)(X_{i+\frac{1}{2}}, t^n)$$

and we now have two determine the second time-derivatives of u and p. The second time-derivative of u can be easily expressed from the first one:

$$\begin{aligned}\partial_{tt}u = \partial_t\left(\partial_t u\right) = \partial_t\left(-\frac{1}{\rho_0}\partial_X p\right) &= -\frac{1}{\rho_0}\partial_X\left(\partial_t p\right), \\ &= -\frac{1}{\rho_0}\partial_X\left(-\frac{(\rho c)^2}{\rho_0}\partial_X u\right), \\ &= \frac{(\rho c)^2}{\rho_0^2}\partial_{XX}u + \frac{1}{\rho_0}(\partial_X u)\left(\partial_X\left(\frac{(\rho c)^2}{\rho_0}\right)\right).\end{aligned}$$

The same procedure can be applied to determine the second time-derivative of p using the following proposition which provides an equation of state independent expression for the first time-derivative of $(\rho c)^2$.

Proposition 3. *For any equation of state, the Lagrangian sound speed $(\rho c)^2$ satisfies the following evolution equation:*

$$\partial_t(\rho c)^2 = -\frac{2(\rho c)^2\mathcal{G}}{\tau\rho_0}\partial_X u,$$

where $\mathcal{G}$ denotes the fundamental derivative:

$$\mathcal{G} = \frac{\tau^2}{2\gamma\rho}\cdot\left.\frac{\partial^2 p}{\partial\tau^2}\right|_S.$$

Proof. Differentiating $(\rho c)^2$ with respect to τ for constant S leads to:

$$\left.\frac{\partial(\rho c)^2}{\partial\tau}\right|_S = \frac{p}{\tau}\cdot\left.\frac{\partial\gamma}{\partial\tau}\right|_S + \frac{\gamma}{\tau}\cdot\left.\frac{\partial p}{\partial\tau}\right|_S - \frac{\gamma p}{\tau^2}. \tag{4.18}$$

We recall that the ratio of specific heats γ and the fundamental derivative $\mathcal{G}$ are defined by:

$$\gamma = -\frac{\tau}{p}\cdot\left.\frac{\partial p}{\partial\tau}\right|_S \quad \text{and} \quad \mathcal{G} = \frac{\tau^2}{2\gamma p}\cdot\left.\frac{\partial^2 p}{\partial\tau^2}\right|_S.$$

One can therefore show that:

$$\left.\frac{\partial\gamma}{\partial\tau}\right|_S = \frac{\gamma}{\tau}\left(\gamma + 1 - 2\mathcal{G}\right).$$

Equation (4.18) thus rewrites:

$$\left.\frac{\partial(\rho c)^2}{\partial\tau}\right|_S = -\frac{2\mathcal{G}(\rho c)^2}{\tau} = -2\rho\mathcal{G}(\rho c)^2.$$

These results allow us to derive the evolution equation of the Lagrangian sound speed $(\rho c)^2$ since:

$$\frac{\partial(\rho c)^2}{\partial t} = \left.\frac{\partial(\rho c)^2}{\partial\tau}\right|_S\cdot\frac{\partial\tau}{\partial t} + \left.\frac{\partial(\rho c)^2}{\partial S}\right|_\tau\cdot\underbrace{\frac{\partial S}{\partial t}}_{=0} = -\frac{2\rho\mathcal{G}(\rho c)^2}{\rho_0\tau}\partial_X u.$$

□

Remark 7. In the specific case of perfect gases, $\mathcal{G}$ is a constant, its value being given by $\mathcal{G} = (\gamma+1)/2$. The time-derivative of $\mathcal{G}$ is therefore zero in this case.

Proposition 3 leads to the following expression for the second time-derivative of p:

$$\begin{aligned}\partial_{tt}p = \partial_t\left(\partial_t p\right) &= \partial_t\left(-\frac{(\rho c)^2}{\rho_0}\partial_X u\right),\\ &= -\frac{1}{\rho_0}\partial_t(\rho c)^2\partial_X u - \frac{(\rho c)^2}{\rho_0}\partial_X\left(\partial_t u\right),\\ &= \frac{2(\rho c)^2\mathcal{G}}{\tau\rho_0^2}\left(\partial_X u\right)^2 + \frac{(\rho c)^2}{\rho_0^2}\partial_{XX}p - \frac{(\rho c)^2}{\rho_0^3}(\partial_X\rho_0)(\partial_X p).\end{aligned}$$

In order to write the third-order accurate Lagrangian fluxes, we first define the approximate first time-derivatives of u and p:

$$\begin{aligned}(\partial_t u)^*_{i+\frac{1}{2}} &= \left(\frac{2}{(\rho_0)_i + (\rho_0)_{i+1}}\right)\left(\frac{p_i - p_{i+1}}{\Delta x}\right) = \left(\partial_t u\right)(x_{i+\frac{1}{2}}, t^n) + \mathcal{O}\left(\Delta x^2\right),\\ (\partial_t p)^*_{i+\frac{1}{2}} &= \left(\frac{(\rho c)_i^2 + (\rho c)_{i+1}^2}{(\rho_0)_i + (\rho_0)_{i+1}}\right)\left(\frac{u_i - u_{i+1}}{\Delta x}\right) = \left(\partial_t p\right)(x_{i+\frac{1}{2}}, t^n) + \mathcal{O}\left(\Delta x^2\right),\end{aligned}$$

and their approximate second time-derivatives:

$$\begin{aligned}(\partial_{tt}u)^*_{i+\frac{1}{2}} &= \left(\frac{(\rho c)_i^2 + (\rho c)_{i+1}^2}{(\rho_0)_i^2 + (\rho_0)_{i+1}^2}\right)\left(\frac{(u_{i+2} + u_{i-1}) - (u_i + u_{i+1})}{2\Delta x^2}\right)\\ &\quad + \frac{1}{2\Delta x}\left(\frac{1}{(\rho_0)_i + (\rho_0)_{i+1}}\right)\left(\frac{(\rho c)_{i+1}^2}{(\rho_0)_{i+1}} - \frac{(\rho c)_i^2}{(\rho_0)_i}\right)\left(\frac{u_{i+1} - u_i}{\Delta x}\right)\\ &= \left(\partial_{tt}u\right)(x_{i+\frac{1}{2}}, t^n) + \mathcal{O}\left(\Delta x^2\right),\\ (\partial_{tt}p)^*_{i+\frac{1}{2}} &= \left(\frac{(\rho c)_i^2 + (\rho c)_{i+1}^2}{(\rho_0)_i^3 + (\rho_0)_{i+1}^3}\right)\left(\frac{(\rho_0)_{i+1} - (\rho_0)_i}{\Delta x}\right)\left(\frac{u_i - u_{i+1}}{\Delta x}\right) + 2\left(\frac{(\rho c)_i^2\mathcal{G}_i + (\rho c)_{i+1}^2\mathcal{G}_{i+1}}{(\rho_0)_i + (\rho_0)_{i+1}}\right)\left(\frac{u_{i+1} - u_i}{\Delta x}\right)^2\\ &\quad + \left(\frac{(\rho c)_i^2 + (\rho c)_{i+1}^2}{(\rho_0)_i^2 + (\rho_0)_{i+1}^2}\right)\left(\frac{(u_{i+2} + u_{i-1}) - (u_i + u_{i+1})}{2\Delta x^2}\right)\\ &= \left(\partial_{tt}p\right)(x_{i+\frac{1}{2}}, t^n) + \mathcal{O}\left(\Delta x^2\right).\end{aligned}$$

Third-order accurate fluxes are then given by:

$$\begin{aligned}u^{*,3}_{i+\frac{1}{2}} &= \frac{9}{16}\left(u_i + u_{i+1}\right) - \frac{1}{16}\left(u_{i-1} + u_{i+2}\right) + \frac{\Delta t}{2}\left(\partial_t u\right)^*_{i+\frac{1}{2}} + \frac{\Delta t^2}{6}\left(\partial_{tt}u\right)^*_{i+\frac{1}{2}}\\ p^{*,3}_{i+\frac{1}{2}} &= \frac{9}{16}\left(p_i + p_{i+1}\right) - \frac{1}{16}\left(p_{i-1} + p_{i+2}\right) + \frac{\Delta t}{2}\left(\partial_t p\right)^*_{i+\frac{1}{2}} + \frac{\Delta t^2}{6}\left(\partial_{tt}p\right)^*_{i+\frac{1}{2}}\\ (pu)^{*,3}_{i+\frac{1}{2}} &= \frac{9}{16}\left(p_i u_i + p_{i+1}u_{i+1}\right) - \frac{1}{16}\left(p_{i-1}u_{i-1} + p_{i+2}u_{i+2}\right) + \frac{\Delta t^2}{3}\left(\partial_t p\right)^*_{i+\frac{1}{2}}\left(\partial_t u\right)^*_{i+\frac{1}{2}}\\ &\quad + \frac{\Delta t}{2}\left(\frac{p_i + p_{i+1}}{2}\right)\left(\left(\partial_t u\right)^*_{i+\frac{1}{2}} + \frac{\Delta t}{3}\left(\partial_{tt}u\right)^*_{i+\frac{1}{2}}\right) + \frac{\Delta t}{2}\left(\frac{u_i + u_{i+1}}{2}\right)\left(\left(\partial_t p\right)^*_{i+\frac{1}{2}} + \frac{\Delta t}{3}\left(\partial_{tt}p\right)^*_{i+\frac{1}{2}}\right)\end{aligned}$$

Fourth-order accurate Lagrangian fluxes

The fourth-order accurate Lagrangian flux writes:

$$\mathbf{F}^{*,4}_{i+\frac{1}{2}} = \mathbf{F}(X_{i+\frac{1}{2}}, t^n) + \frac{\Delta t}{2}\left(\frac{\partial\mathbf{F}(\mathbf{U})}{\partial t}\right)(X_{i+\frac{1}{2}}, t^n) + \frac{\Delta t^2}{6}\left(\frac{\partial^2\mathbf{F}(\mathbf{U})}{\partial t^2}\right)(X_{i+\frac{1}{2}}, t^n) + \frac{\Delta t^3}{24}\left(\frac{\partial^3\mathbf{F}(\mathbf{U})}{\partial t^3}\right)(X_{i+\frac{1}{2}}, t^n)$$

and we now have to determine the third time-derivatives of u and p. Since such calculations will obviously lead to complex expressions, we present in this section a method relying on the Maple algebraic calculator that provides expressions of time-derivatives for building arbitrarily high-order accurate Lagrangian fluxes. First of all, we define some differentiating rules with respect to time for each variable that may appear in approximate Lagrangian fluxes:

```
> DTTAU := diff(tau(x,t),t) =  1/rho0(x) * diff(u(x,t),x)
> DTU   := diff(u(x,t),t)   = -1/rho0(x) * diff(p(x,t),x);
> DTP   := diff(p(x,t),t)   = -rc2(x,t)/rho0(x,t) * diff(u(x,t),x);
> DTRC2 := diff(rc2(x,t),t) = -2*rc2(x,t)/(tau(x,t)*rho0(x))*G(x,t) * diff(u(x,t),x);
> DTG   := diff(G(x,t),t)   =  0;
```

Remark 8. For the sake of simplicity, we restrict to the case of perfect gas laws from $N = 4$. As we indicated previously (see remark 7), the fundamental derivative $\mathcal{G}$ is constant in this context and its time-derivative is zero.

We now initialize a table `flux` that contains the flux variables, in our case u and p. Note that there is no need to add (pu) to this table since its time-derivatives can be easily computed once the time-derivatives of u and p are known.

```
> for n from 1 to N-1 do
>   flux[n] := [-diff(u(x,t),t$n), diff(p(x,t),t$n)];
> end do;
```

We apply the Cauchy-Kovalevskaya procedure by replacing several times the time-derivatives with expressions given previously.

```
> for n from 1 to N-1 do
>   for i from 1 to 3 do
>     for j from 1 to n do
>       flux[n][i] := expand(subs([DTTAU, DTU, DTP, DTRC2, DTG], flux[n][i]));
>     end do;
>   end do;
> end do;
```

Now that the Cauchy-Kovalevskaya procedure has been applied, the fluxes' expressions can be slightly simplified since $\rho_0 = 1/\tau$ at time t^n:

```
> for n from 1 to N-1 do
>   for i from 1 to 3 do
>     flux[n][i] := expand(subs([rho0(x)=1/tau(x,t)], flux[n][i]));
>   end do;
> end do;
```

Note that this substitution step allows to avoid costly floating point division operations. At this stage, it would be convenient to automatically generate source code that performs the computation of the required time-derivatives. To that end, we first replace the `diff(phi(x,t),x$m)` expressions with their approximate values.

```
> for n from 1 to N-1 do
>   for i from 1 to 3 do
```

```
>     for m from n to 0
>       Neff := N-n + mod(N-n, 2);
>       flux[n][i] := subs([diff(tau(x,t),x$m), _tau[m][k]), flux[n][var]);
>       flux[n][i] := subs([diff(u(x,t)  ,x$m),   _u[m][k]), flux[n][var]);
>       flux[n][i] := subs([diff(p(x,t)  ,x$m),   _p[m][k]), flux[n][var]);
>       flux[n][i] := subs([diff(rc2(x,t),x$m), _rc2[m][k]), flux[n][var]);
>       flux[n][i] := subs([diff(G(x,t)  ,x$m),   _G[m][k]), flux[n][var]);
>     end do;
>   end do;
> end do;
```

We recall that the `_phi[m][k]` terms should be estimated using the finite difference formulae (4.16):

$$\texttt{\_phi[m][k]} = \begin{cases} \dfrac{1}{\Delta x^m} \displaystyle\sum_{k=1}^{s} d_{m,k}^{N_{\text{eff}}} \cdot \left(\phi_{i+k}^n + \phi_{i-k+1}^n\right) & \text{if } m \text{ is even,} \\ \dfrac{1}{\Delta x^m} \displaystyle\sum_{k=1}^{s} d_{m,k}^{N_{\text{eff}}} \cdot \left(\phi_{i+k}^n - \phi_{i-k+1}^n\right) & \text{if } m \text{ is odd,} \end{cases}$$

the $d_{m,k}^{N_{\text{eff}}}$ coefficients being given in Table 4.2. It is now possible to generate source code as desired (here in C language) using Maple's `codegen` package.

```
> with(codegen);
> C(flux, precision=double);
```

Note that this code generation function provides many available options. One of these is particularly efficient since it forces source code optimization: several intermediate variables are then created to avoid computing several times the same quantities. Fourth-order accurate Lagrangian fluxes can finally be easily built using the procedure described in this section with $N = 4$. For example, the fourth-order accurate flux term $p^{*,4}$ writes:

$$p^{*,4} = \texttt{\_p[0][4]} + \frac{\Delta t}{2}\texttt{flux[1][2]} + \frac{\Delta t^2}{6}\texttt{flux[2][2]} + \frac{\Delta t^3}{24}\texttt{flux[3][2]}.$$

Practically, this procedure has been exploited in order to build up to sixth-order accurate Lagrangian fluxes.

4.1.3 Remap step

Once the Lagrangian scheme has been applied, we have at our disposal a set of Lagrangian conservative variables $\overline{(\rho_0\phi)}$ for $\phi \in \{1, u, e\}$ at time t^{n+1}. These correspond to Eulerian conservative variables on the non-uniform $\{x_{i+\frac{1}{2}}^{n+1}\}$ grid. Indeed, since the Lagrangian variable change is given by $\rho_0 dX = \rho dx$, Eulerian and Lagrangian conservative variables satisfy the following *exact* formula:

$$\Delta X \; \overline{(\rho_0\phi)}_i^{n+1} = \int_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}} (\rho_0\phi)(X, t^{n+1})\, dX = \int_{x_{i-\frac{1}{2}}^{n+1}}^{x_{i+\frac{1}{2}}^{n+1}} (\rho\phi)(x, t^{n+1})\, dx. \tag{4.19}$$

The remap step aims at projecting these values on the initial regular Cartesian grid. It is based on the following integral splitting for $\phi \in \{1, u, e\}$:

$$\begin{aligned}\Delta x\ \overline{(\rho\phi)}_i^{n+1} &= \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (\rho\phi)(x, t^{n+1})\ dx, \\ &= \underbrace{\int_{x_{i-\frac{1}{2}}}^{x_{i-\frac{1}{2}}^{n+1}} (\rho\phi)(x, t^{n+1})\ dx}_{\mathcal{P}_{i-\frac{1}{2}}} + \underbrace{\int_{x_{i-\frac{1}{2}}^{n+1}}^{x_{i+\frac{1}{2}}^{n+1}} (\rho\phi)(x, t^{n+1})\ dx}_{\mathcal{P}_i} + \underbrace{\int_{x_{i+\frac{1}{2}}^{n+1}}^{x_{i+\frac{1}{2}}} (\rho\phi)(x, t^{n+1})\ dx}_{\mathcal{P}_{i+\frac{1}{2}}}.\end{aligned}$$

The $\mathcal{P}_{i\mp\frac{1}{2}}$ terms in the right hand side can be expressed from the so-called remap fluxes $(\rho\phi)^*_{i\mp\frac{1}{2}}$:

$$\mathcal{P}_{i\mp\frac{1}{2}} = \left(x_{i\mp\frac{1}{2}}^{n+1} - x_{i\mp\frac{1}{2}}\right)(\rho\phi)^*_{i\mp\frac{1}{2}} \quad \text{with} \quad (\rho\phi)^*_{i\mp\frac{1}{2}} = \frac{1}{x_{i\mp\frac{1}{2}}^{n+1} - x_{i\mp\frac{1}{2}}} \int_{x_{i\mp\frac{1}{2}}}^{x_{i\mp\frac{1}{2}}^{n+1}} (\rho\phi)(x, t^{n+1})\ dx.$$

$\mathcal{P}_i$ corresponds to the cell average of $(\rho\phi)$ on the non-uniform grid $\{x_{i+\frac{1}{2}}^{n+1}\}_i$ and can be estimated using (4.19). Since $x_{i+\frac{1}{2}}^{n+1} - x_{i+\frac{1}{2}} = u^*_{i+\frac{1}{2}}\Delta t$, the remap scheme for $\phi \in \{1, u, e\}$ finally writes:

$$\overline{(\rho\phi)}_i^{n+1} = \overline{(\rho_0\phi)}_i^{n+1} - \frac{\Delta t}{\Delta x}\left(u^*_{i+\frac{1}{2}}(\rho\phi)^*_{i+\frac{1}{2}} - u^*_{i-\frac{1}{2}}(\rho\phi)^*_{i-\frac{1}{2}}\right). \tag{4.20}$$

We now have to compute high-order accurate remap fluxes. To that end, we resort to the standard polynomial reconstruction that has been presented in section 3.2 with an additional upwinding. Let us introduce the Nth-order accurate numerical flux:

$$(\rho\phi)^{*,N}_{i+\frac{1}{2}} = \frac{1}{x_{i+\frac{1}{2}}^{n+1} - x_{i-\frac{1}{2}}^{n+1}} \int_{x_{i+\frac{1}{2}}}^{x_{i+\frac{1}{2}}^{n+1}} p^{\phi,N}_{\text{upwind}} \quad \text{with} \quad \text{upwind} = \begin{cases} i & \text{if } x_{i+\frac{1}{2}}^{n+1} \geq x_{i+\frac{1}{2}}, \\ i+1 & \text{if } x_{i+\frac{1}{2}}^{n+1} \leq x_{i+\frac{1}{2}}, \end{cases}$$

where $p_i^{\phi,N}$ is a polynom satisfying:

$$p_i^{\phi,N} = (\phi) + \mathcal{O}\left(\Delta x^N\right) \text{ on } \left[x_{i-\frac{1}{2}}^{n+1}; x_{i+\frac{1}{2}}^{n+1}\right] \quad \text{and} \quad \int_{x_{i-\frac{1}{2}}^{n+1}}^{x_{i+\frac{1}{2}}^{n+1}} p_i^{\phi,N} = \int_{x_{i-\frac{1}{2}}^{n+1}}^{x_{i+\frac{1}{2}}^{n+1}} (\rho\phi). \tag{4.21}$$

We now define $s = \lfloor N/2 \rfloor + 1$ and $S_i^N = \{x_{i-s+\frac{1}{2}}^{n+1}, \ldots, x_{i+s-\frac{1}{2}}^{n+1}\}$. S_i^N is a centered stencil of $2s$ points around $[x_{i-\frac{1}{2}}^{n+1}; x_{i+\frac{1}{2}}^{n+1}]$. Let H_i^ϕ be the following primitive of $(\rho\phi)$:

$$H_i^\phi(x) = \int_{x_{i-s+\frac{1}{2}}^{n+1}}^{x} (\rho\phi)(\xi, t^{n+1})\mathrm{d}\xi. \tag{4.22}$$

Using formula (4.19), one easily computes the values of H_i^ϕ on S_i^N:

$$\left|\begin{array}{l} H_i^\phi(x_{i-s+\frac{1}{2}}^{n+1}) = 0, \\ H_i^\phi(x_{i-s+\frac{3}{2}}^{n+1}) = \Delta X \overline{(\rho_0\phi)}_{i-s+1}^{n+1}, \\ H_i^\phi(x_{i-s+\frac{5}{2}}^{n+1}) = \Delta X \left(\overline{(\rho_0\phi)}_{i-s+1}^{n+1} + \overline{(\rho_0\phi)}_{i-s+2}^{n+1}\right), \\ \quad\vdots \\ H_i^\phi(x_{i+s-\frac{1}{2}}^{n+1}) = \Delta X \left(\overline{(\rho_0\phi)}_{i-s+1}^{n+1} + \ldots + \overline{(\rho_0\phi)}_{i+s-1}^{n+1}\right). \end{array}\right. \tag{4.23}$$

Let $P_i^{\phi,N}$ be the unique Lagrange interpolating polynom of H_i^{ϕ} on S_i^N. As explained in section 3.2, the following choice of $p_i^{\phi,N}$ satisfies (4.21):

$$p_i^{\phi,N} = \left[P_i^{\phi,N}\right]'. \tag{4.24}$$

Practically, there is though no need to compute p_i^{ϕ}. The approximate remap fluxes can be obtained directly from the Lagrange interpolating polynom P_i^{ϕ}:

$$(\rho\phi)_{i+\frac{1}{2}}^{*,N} = \frac{1}{x_{i+\frac{1}{2}}^{n+1} - x_{i-\frac{1}{2}}^{n+1}} \left(P_i^{\phi,N}\left(x_{i+\frac{1}{2}}^{n+1}\right) - P_i^{\phi,N}\left(x_{i+\frac{1}{2}}\right)\right) \tag{4.25}$$

which allow to update Eulerian conservative variables.

4.1.4 Stability condition

The time step is constrained by a classical CFL condition for both Lagrangian and remap steps:

$$\begin{cases} \Delta t \leq \dfrac{\Delta x}{\max c_i} & \text{for the Lagrangian scheme,} \\ \Delta t \leq \dfrac{\Delta x}{\max u_{i+\frac{1}{2}}^*} & \text{for the remap scheme,} \end{cases}$$

where c denotes the Eulerian sound speed. To insure the stability of our numerical schemes, we therefore consider the following time step constraint for the whole Lagrange-remap scheme:

$$\Delta t \leq \frac{\Delta x}{\max\left(|u_i| + c_i\right)}. \tag{4.26}$$

4.1.5 Summarized one-dimensional Lagrange-remap scheme

In order to complete the description of our one-dimensional Lagrange-remap schemes, we summarize hereafter the different tasks that have to be performed.

1. **Lagrangian step** [1].
   (a) Initialize Lagrangian conservative variables: since $\rho_0 = \rho$ at time t^n, one has $\rho_0 u = \rho u$ and $\rho_0 e = \rho e$.
   (b) Compute cell-centered point-wise values of Lagrangian conservative variables using (4.15).
   (c) Form the Nth order accurate Lagrangian fluxes according to the truncated Taylor expansion (4.14).
   (d) Replace time-derivatives involved in (4.14) with space-derivatives using the Cauchy-Kovalevskaya procedure.
   (e) Compute cell-centered point-wise values of each primitive variable required to form numerical fluxes. We recall that this step can be performed without loss of accuracy since we have at our disposal cell-centered point-wise values of conservative variables.
   (f) Compute each term appearing in the Lagrangian fluxes after the Cauchy-Kovalevskaya procedure has been applied using finite difference formulae, see (4.16).

[1] As we indicated previously (see remark 6), step (b) is only necessary for accuracy orders that are strictly greater than two

(g) Update the $(\rho_0 u)$ and $(\rho_0 e)$ Lagrangian conservative variables according to the finite volume scheme (4.12).

(h) Compute updated positions of the Eulerian grid using (4.17).

2. **Remap step**.

(a) Compute high-order accurate remap fluxes using a conservative polynomial reconstruction method as explained in section 4.1.3.

(b) Compute updated values of the Eulerian conservative variables ρ, (ρu) and (ρe) according to the remap scheme (4.20).

Remark 9. Practically, the Lagrangian flux u^* has to be computed so that the Eulerian grid can be updated but there is no need to solve the equation on $(\rho_0 \tau)$ in which it is involved. Indeed, the remap scheme given by (4.20) does not make any use of updated values of $(\rho_0 \tau)$. The effectively solved Lagrangian scheme therefore writes:

$$\begin{pmatrix} (\rho_0 u)_i^{n+1} \\ (\rho_0 e)_i^{n+1} \end{pmatrix} = \begin{pmatrix} (\rho_0 u)_i^{n} \\ (\rho_0 e)_i^{n} \end{pmatrix} - \frac{\Delta t}{\Delta x} \begin{pmatrix} p^*_{i+\frac{1}{2}} - p^*_{i-\frac{1}{2}} \\ (pu)^*_{i+\frac{1}{2}} - (pu)^*_{i-\frac{1}{2}} \end{pmatrix}. \tag{4.27}$$

We have been able to give formal proofs of accuracy for the second, third and fourth-order GoHy schemes with the help of the algebraic calculation software Maple. The formal proof for the third-order scheme is given in Appendix F.

4.2 Building high-order schemes for two-dimensional planar geometries

We show in this section how the high-order schemes that have been developed for one-dimensional geometries can be extended to the two-dimensional case. We here propose to resort to dimensional splitting techniques which present the advantage of allowing cache-oblivious implementations. Indeed, instead of operating on multidimensional datasets, dimensionally split schemes operate on one-dimensional datasets: several so-called dimensional sweeps are performed successively along the directions of the considered geometry. They are therefore compatible with the second principle that we stated in introduction (see page 13) in the sense that a proper implementation of dimensionally split methods operates mostly on memory contiguous data. Practically, this can be achieved by storing the considered variables in several tables with different memory alignments (which is of course a drawback in terms of memory requirements). In the two-dimensional case, variables that have to be updated need to be stored twice: once with a memory alignment along the x direction and once with another memory alignment along the y direction (see Figure 4.1). Switching from one of these storage tables to the other one can be done during the update step. This strategy supposes that the update step is performed on non-contiguous data (it is fortunately a very cheap step in terms of computational cost) but allows to save a more expensive transposition step. The cache-oblivious algorithm for implementing dimensionally split Lagrange-remap schemes is summarized by the following pseudo-code in which the different steps are written in blue if they operate on contiguous data and in red otherwise.

Input: $\mathbf{U}$ = conservative variables, N = number of dimensional sweeps
// store in U_x the x-aligned version of $\mathbf{U}$
$\mathbf{U}_x \leftarrow \mathbf{U}$
// loop on dimensional sweeps
for $n = 1 \rightarrow N$ **do**

if $n \bmod 2 = 1$ **then**
// apply the one-dimension scheme along the x direction
$\mathbf{F}_x^* = \text{ComputeLagrangianFluxes}(\mathbf{U}_x)$
$\mathbf{U}_x = \text{UpdateVariables}(\mathbf{U}_x,\ \mathbf{F}_x^*)$
$\mathbf{F}_x^* = \text{ComputeRemapFluxes}(\mathbf{U}_x)$
// put updated variables in an y-aligned table after the remap step
$\mathbf{U}_y = \text{UpdateVariables}(\mathbf{U}_x,\ \mathbf{F}_x^*)$
else
// apply the one-dimensional scheme along the y direction
$\mathbf{F}_y^* = \text{ComputeLagrangianFluxes}(\mathbf{U}_y)$
$\mathbf{U}_y = \text{UpdateVariables}(\mathbf{U}_y,\ \mathbf{F}_y^*)$
$\mathbf{F}_y^* = \text{ComputeRemapFluxes}(\mathbf{U}_y)$
// put updated variables in an x-aligned table after the remap step
$\mathbf{U}_x = \text{UpdateVariables}(\mathbf{U}_y,\ \mathbf{F}_y^*)$
end if
end for
// once all dimensional sweeps have been performed, put updated variables in $\mathbf{U}$
if $N \bmod 2 = 0$ **then**
// if N is even, updated variables are stored in $\mathbf{U}_x$
$\mathbf{U} \leftarrow \mathbf{U}_x$
else
// if N is odd, updated variables are stored in $\mathbf{U}_y$
$\mathbf{U} \leftarrow \mathbf{U}_y$
end if
Output: $\mathbf{U}$ = updated conservative variables

The outline of this section is the following. We first recall a few well-known theoretical elements about splitting techniques (these have already been widely discussed in the literature, see [77] for example) and propose several splitting sequences up to sixth-order accuracy. We then apply these methods to extend one-dimensional GoHy schemes to the two-dimensional case.

4.2.1 Dimensional splitting techniques

Let A and B denote two differential operators. We consider the evolution equation:

$$\partial_t \phi = \mathcal{A}\phi + \mathcal{B}\phi, \tag{4.28}$$

whose exact solution at time $t + \Delta t$ is given by:

$$\phi(t + \Delta t) = \exp\left(\Delta t(\mathcal{A} + \mathcal{B})\right) \phi(t).$$

Splitting techniques consists in solving alternatively the two following equations:

$$\partial_t \phi = \mathcal{A}\phi \quad \text{and} \quad \partial_t \phi = \mathcal{B}\phi, \tag{4.29}$$

instead of equation (4.28), leading to the following approximate solution at time $t + \Delta t$:

$$\widetilde{\phi}(t + \Delta t) = \left[\exp\left(\Delta t\mathcal{A}\right) \cdot \exp\left(\Delta t\mathcal{B}\right)\right] \phi(t). \tag{4.30}$$

but such an approach has consequences in terms of accuracy as shown in the following proposition.

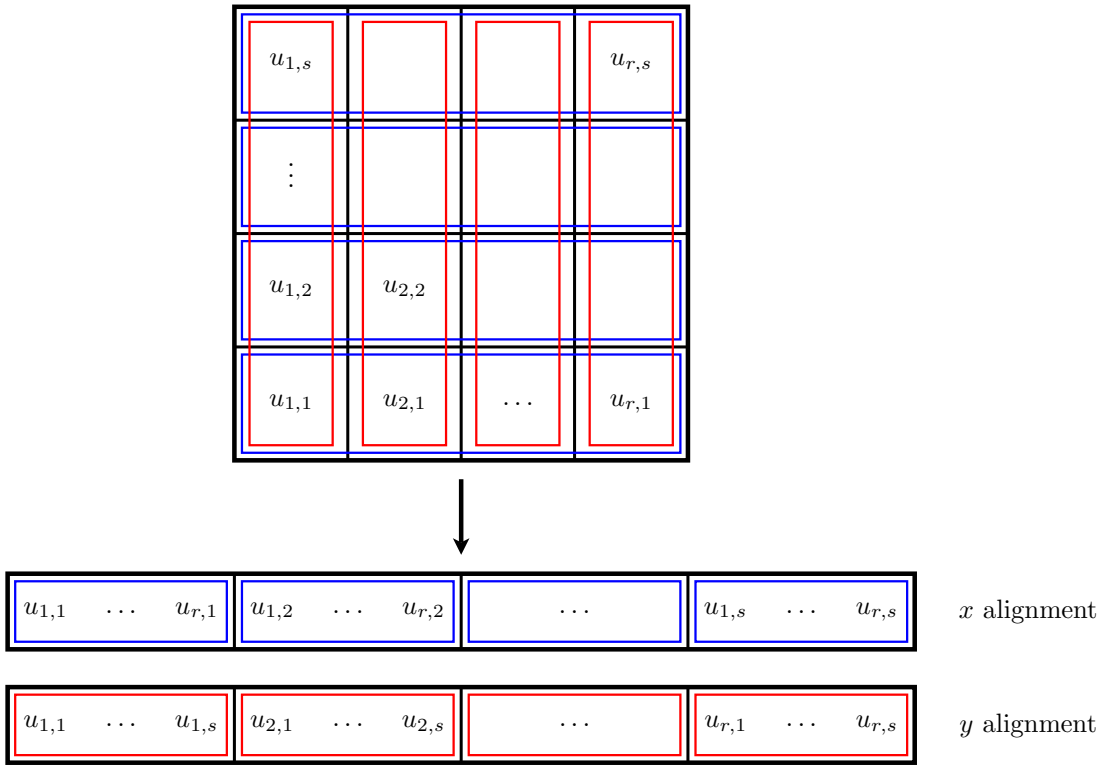


Figure 4.1: Memory alignment strategies

Proposition 4. *The approximate solution given by* (4.30) *is first-order accurate:*

$$\widetilde{\phi}(t+\Delta t) = \phi(t+\Delta t) + \mathcal{O}\left(\Delta t^2\right).$$

Proof. The accuracy of the approximate solution can be estimated by comparing Taylor expansions of $\exp\left(\Delta t(\mathcal{A}+\mathcal{B})\right)$ and $[\exp\left(\Delta t\mathcal{A}\right)\cdot\exp\left(\Delta t\mathcal{B}\right))]$. On the one hand, we have:

$$\exp\left(\Delta t(\mathcal{A}+\mathcal{B})\right) = 1 + \Delta t\left(\mathcal{A}+\mathcal{B}\right) + \frac{\Delta t^2}{2}\left(\mathcal{A}^2+\mathcal{B}^2+\mathcal{A}\mathcal{B}+\mathcal{B}\mathcal{A}\right) + \mathcal{O}\left(\Delta t^3\right), \tag{4.31}$$

whereas on the other hand one can show that:

$$\begin{aligned}\exp\left(\Delta t\mathcal{A}\right)\cdot\exp\left(\Delta t\mathcal{B}\right)) &= \left(1+\Delta t\mathcal{A}+\frac{\Delta t^2}{2}\mathcal{A}^2+\mathcal{O}\left(\Delta t^3\right)\right)\cdot\left(1+\Delta t\mathcal{B}+\frac{\Delta t^2}{2}\mathcal{B}^2+\mathcal{O}\left(\Delta t^3\right)\right)\\ &= 1+\Delta t\left(\mathcal{A}+\mathcal{B}\right)+\frac{\Delta t^2}{2}\left(\mathcal{A}^2+\mathcal{B}^2+2\mathcal{A}\mathcal{B}\right)+\mathcal{O}\left(\Delta t^3\right).\end{aligned}$$

The solution given by (4.30) is therefore obviously first-order accurate since $\mathcal{A}\mathcal{B}\neq\mathcal{B}\mathcal{A}$ generally. □

This first-order splitting is called the Godunov splitting. In order to increase accuracy, one has to solve the split equations (4.29) several times with appropriately weighted time steps, leading to the following solution at time $t+\Delta t$:

$$\widetilde{\phi}(t+\Delta t) = \prod_{i=1}^{p}\left[\exp\left(a_i\Delta t\mathcal{A}\right)\cdot\exp\left(b_i\Delta t\mathcal{B}\right)\right]\phi(t) \tag{4.32}$$

where a_i and b_i are real coefficients. Each step needed for computing $\phi(t+\Delta t)$ according to (4.32) is called a dimensional sweep. In the sequel, n denotes the number of dimensional sweeps, *i.e.* the number of non-zero a_i and b_i coefficients.

Remark 10. Note that the choice of dimensional splitting coefficients has an impact on the time step constraint. Let c denote a splitting coefficient, the effective time step in the multidimensional case is now given by $c\Delta t$. As a matter of fact, if c is less than 1, the time step Δt can be increased by a factor $1/c$ without violating the stability condition (4.26). In the case of of multidimensional GoHy schemes, we therefore determine the time step using the following adjusted formula that takes the value of splitting coefficients into account:

$$\Delta t = \frac{\Delta t_{\mathrm{CFL}}}{\max(|a_1|,\ldots|a_p|,|b_1|,\ldots,|b_p|)}$$

where Δt_{CFL} denotes the time step obtained from the CFL condition (4.26). This property gives a hint about how selecting sets of splitting coefficients: whereas these are usually chosen in order to minimize the error caused by the splitting strategy (see [13] for example), we propose to determine the sequence with the lowest $\max(|a_1|,\ldots|a_p|,|b_1|,\ldots,|b_p|)$ value in order to maximize the time step. Such splitting sequences will be called *optimal* in the sequel.

Remark 11. In practical cases, we are often interested in preserving symmetry and we therefore only consider symmetric splitting sequences, *i.e.* sequences that satisfy for all i:

$$\begin{cases} b_i = a_{n/2+1-i} & \text{if } n \text{ is even,}\\ a_i = a_{(n+1)/2+1-i} \text{ and } b_i = b_{(n-1)/2+1-i} & \text{if } n \text{ is odd.}\end{cases}$$

We now describe how to build second, third and fourth-order splitting sequences. Several sets of coefficients can be found in the literature on symplectic integrators for Hamiltonian systems, our contribution to this topic will therefore mainly consist in providing the optimal set. We finally summarize several splitting sequences of from third to sixth-order accuracy in Table 4.3.

Second-order accurate splitting method: the Strang splitting

Second-order accurate splitting methods can be obtained by taking $p = 2$ as shown below.

Proposition 5. *The approximate solution given by:*

$$\widetilde{\phi}(t + \Delta t) = \left[\prod_{i=1}^{2} \exp(a_i \Delta t \mathcal{A}) \cdot \exp(b_i \Delta t \mathcal{B})\right] \phi(t)$$

is second-order accurate if and only if the a_i and b_i coefficients satisfy the following system of non-linear equations:

$$\begin{cases} a_1 + a_2 &= 1, \\ b_1 + b_2 &= 1, \\ a_1 b_2 &= 1/2. \end{cases} \tag{4.33}$$

Proof. Once again, the proof of this proposition relies on Taylor expansions. Indeed, one can see that:

$$\begin{aligned} \prod_{i=1}^{2} \exp(a_i \Delta t \mathcal{A}) \cdot \exp(b_i \Delta t \mathcal{B}) = 1 &+ (a_1 + a_2)\, \Delta t \mathcal{A} + (b_1 + b_2)\, \Delta t \mathcal{B} \\ &+ (a_1 b_1 + a_1 b_2 + a_2 b_1)\, \Delta t^2 \mathcal{A}\mathcal{B} + b_1 a_2 \Delta t^2 \mathcal{B}\mathcal{A} \\ &+ \left(a_1^2 + a_2^2 + 2 a_1 a_2\right) \frac{\Delta t^2}{2} \mathcal{A} + \left(b_1^2 + b_2^2 + 2 b_1 b_2\right) \frac{\Delta t^2}{2} \mathcal{B} + \mathcal{O}\left(\Delta t^3\right). \end{aligned}$$

Comparing this expression with the Taylor expansion of $\exp\left(\Delta t(\mathcal{A} + \mathcal{B})\right)$ given in (4.31) leads to the following non-linear system of equations:

$$a_1 + a_2 = 1, \tag{4.34a}$$
$$b_1 + b_2 = 1, \tag{4.34b}$$
$$a_1^2 + a_2^2 + 2 a_1 a_2 = 1, \tag{4.34c}$$
$$b_1^2 + b_2^2 + 2 b_1 b_2 = 1, \tag{4.34d}$$
$$a_1 b_1 + a_1 b_2 + a_2 b_2 = 1/2, \tag{4.34e}$$
$$a_2 b_1 = 1/2. \tag{4.34f}$$

One can easily notice that (4.34a) $\Rightarrow$ (4.34c) and (4.34b) $\Rightarrow$ (4.34d) since:

$$a_1^2 + a_2^2 + 2 a_1 a_2 = (a_1 + a_2)^2 \quad \text{and} \quad b_1^2 + b_2^2 + 2 b_1 b_2 = (b_1 + b_2)^2.$$

Moreover, equation (4.34e) can be obtained from (4.34a), (4.34b) and (4.34f):

$$\begin{aligned} a_1 b_1 + a_1 b_2 + a_2 b_2 &= a_1(b_1 + b_2) + a_2 b_2, \\ &= a_1 + a_2 b_2, \\ &= 1 - a_2 + a_2 b_2, \\ &= 1 - a_2(1 - b_2), \\ &= 1 - a_2 b_1, \\ &= 1/2. \end{aligned}$$

The remaining equations finally write:

$$\left\{\begin{array}{rcl} a_1 + a_2 & = & 1, \\ b_1 + b_2 & = & 1, \\ a_1 b_2 & = & 1/2. \end{array}\right.$$

□

The optimal second-order accurate splitting method is obtained by taking either $a_1 = 0$ or $a_1 = 1/2$ which actually leads to the (symmetric) Strang splitting sequence [77]:

$$a_1 = a_2 = \frac{1}{2},\ b_1 = 1,\ b_2 = 0.$$

Third-order accurate splitting methods

In order to reach third-order accuracy, one has to consider the $p = 3$ case for which several sets of coefficients can be found as shown hereafter.

Proposition 6. *The approximate solution given by:*

$$\widetilde{\phi}(t + \Delta t) = \left[\prod_{i=1}^{3} \exp(a_i \Delta t \mathcal{A}) \cdot \exp(b_i \Delta t \mathcal{B})\right] \phi(t)$$

is third-order accurate if and only if the a_i and b_i coefficients satisfy the following system of non-linear equations:

$$\left\{\begin{array}{rcll} a_1 + a_2 + a_3 & = & 1, & \\ b_1 + b_2 + b_3 & = & 1, & \\ a_2 b_1 + a_3 b_1 + a_3 b_2 & = & 1/2, & (4.35) \\ a_2 b_1^2 + a_3 (b_1 + b_2)^2 & = & 1/3, & \\ b_3 + b_2 (a_1 + a_2)^2 + b_1 a_1^2 & = & 1/3. & \end{array}\right.$$

The proof is given in Appendix B. Some sequences satisfying system (4.35) can be found in the literature (see works of Forest and Ruth [36] and McLachlan [61] for example). We now would like to determine the optimal set of coefficients. To that end, we have solved system (4.35) numerically using Maple and it turns out that the optimal unsymmetric sequence is given by:

$$\begin{aligned} a_1 &= +0.28321919245984298, \\ a_2 &= +0.78867513459481288, \\ a_3 &= +0.07189432705465587, \\ b_1 &= +0.65323862309023541, \\ b_2 &= -0.44191375768504829, \\ b_3 &= +0.78867513459481288, \end{aligned}$$

whereas the optimal symmetric set of coefficients is actually the one proposed by McLachlan [61]:

$$\begin{aligned} a_1 &= +0.91966152301739986, \\ a_2 &= -0.18799161879915978, \\ a_3 &= +0.26833009578175993. \end{aligned}$$

Remark 12. Note the presence of negative coefficients. These can not be avoided from third-order accuracy and will generate negative time steps that could possibly be difficult to handle. Practically, we however did not encounter any particular issue. The only upwind-biased formula we resort to is used to determine the upwind cell for building remap fluxes and, since it depends on the sign of $(u^*\Delta t)$, it is automatically able to handle negative time steps. All other reconstruction methods operate on centered stencils and therefore do not require any special treatment. On the contrary, introducing slope or flux limiting techniques would be more complicated in such an unusual case.

Fourth-order and higher-order accurate splitting methods

For the case of fourth-order accuracy, we restrict to theoretical results given in the related literature. Chin [13] has shown that symmetric fourth-order accurate splitting methods can be built using nine dimensional sweeps with following a_i coefficients:

$$\begin{aligned} b_1 = b_4 &= \frac{1}{4}\left(1 \mp \sqrt{\frac{9a_2 - 4 \pm \sqrt{3 - 12a_2 + 9a_2^2}}{3a_2}}\right), \\ b_2 = b_3 &= 1/2 - b_1, \\ a_1 = a_5 &= 1/6 - 4a_2 b_2^2, \\ a_3 &= 1 - 2(a_1 + a_2), \end{aligned}$$

a_2 being a free negative parameter. The optimal symmetric set of coefficients is obtained with

$$a_2 = a_4 = -0.27516060407455222,$$

and it is given in Table 4.3. Forest and Ruth [36] propose the following symmetric fourth-order accurate sequence with $n = 7$:

$$a_1 = a_4 = \frac{2 + 2^{1/3} + 2^{-1/3}}{6}, \quad a_2 = a_3 = \frac{1 - 2^{1/3} - 2^{-1/3}}{6}, \quad b_1 = b_3 = \frac{1}{2 - 2^{1/3}}, \quad b_2 = \frac{1}{1 - 2^{2/3}}, \quad (4.36)$$

whereas McLachlan gives fourth ($n = 8$) and fifth-order ($n = 12$) unsymmetric sets of coefficients in [61] (these have been reported in Table 4.3).

Remark 13. (Yoshida's procedure) Regarding higher-order splitting sequences, Yoshida [95] proposes an algorithm for building symmetric dimensional splitting sequences with an arbitrary (but necessarily even) order of accuracy. Let S_{2m} denote a $(2m)$th order accurate splitting method, a $(2m+2)$th order one is given by:

$$S_{2m+2}(t) = S_{2m}(\alpha t) \cdot S_{2m}(\beta t) \cdot S_{2m}(\alpha t)$$

where the α and β coefficients satisfy:

$$\begin{cases} 2\alpha + \beta & = & 1, \\ 2\alpha^{2m+1} + \beta^{2m+1} & = & 0. \end{cases}$$

They are therefore given by:

$$\alpha = -\frac{2^{\frac{1}{2(2m+1)}}}{2-2^{\frac{1}{2(2m+1)}}} \quad \text{and} \quad \beta = \frac{1}{2-2^{\frac{1}{2m+1}}}.$$

Applying Yoshida's procedure with $m = 1$ from the second-order Strang sequence, one obtains Forest and Ruth's set of coefficients (4.36). Applying it one more time leads to a symmetric sixth-order accurate dimensional splitting sequence with $n = 15$. It is given in Table 4.3.

4.2.2 Extending one-dimensional schemes to the two-dimensional case

At this stage, we have at our disposal several splitting techniques that allow to achieve high-order accuracy. We now detail how one-dimensional schemes can practically be extended to the two-dimensional case. To that end, we first recall the two-dimensional Euler system of equations:

$$\partial_t \begin{pmatrix} \rho \\ \rho u_x \\ \rho u_y \\ \rho e \end{pmatrix} + \partial_x \begin{pmatrix} \rho u_x \\ \rho u_x^2 + p \\ \rho u_x u_y \\ (\rho e + p) u_x \end{pmatrix} + \partial_y \begin{pmatrix} \rho u_y \\ \rho u_x u_y \\ \rho u_y^2 + p \\ (\rho e + p) u_y \end{pmatrix} = \mathbf{0}.$$

According to (4.29), one has to solve several times the following split systems respectively along the x and y directions:

$$\partial_t \begin{pmatrix} \rho \\ \rho u_x \\ \rho u_y \\ \rho e \end{pmatrix} + \partial_x \begin{pmatrix} \rho u_x \\ \rho u_x^2 + p \\ \rho u_x u_y \\ (\rho e + p) u_x \end{pmatrix} = \mathbf{0} \quad \text{and} \quad \partial_t \begin{pmatrix} \rho \\ \rho u_x \\ \rho u_y \\ \rho e \end{pmatrix} + \partial_y \begin{pmatrix} \rho u_x \\ \rho u_x u_y \\ \rho u_y^2 + p \\ (\rho e + p) u_y \end{pmatrix} = \mathbf{0}$$

with appropriately weighted time steps. These split systems correspond to the one-dimensional set of equations that have been studied in section 4.1 with an additional pure advection equation for the transverse velocity component. As we indicated previously (see remark 5 page 60), this additional equation does not need any treatment during the Lagrangian step: it is solved during the remap step by applying the remap scheme (4.20) with $\phi = u_y$ or $\phi = u_x$ depending on the considered direction.

However, achieving high-order accuracy requires an additional step. Indeed, a trivial combination of high-order accurate one-dimensional schemes and high-order dimensional splitting sequences does not allow to go beyond second-order accuracy in practice. This is due to the fact that two-dimensional schemes naturally operate on two-dimensional cell averages whereas the one-dimensional Lagrange-remap schemes presented in section 4.1 operate on one-dimensional cell averages. We therefore propose in the sequel some conversion operations in order to switch between the one-dimensional and two-dimensional contexts. For the sake of simplicity, we now focus on the example of a dimensional sweep along the x direction (the treatment along the y direction being analog), more precisely along the j-th line, and introduce the following notations.

- $\left[\phi_{i,j}^n\right]^{xy}$ denotes the two-dimensional cell average of ϕ over $[x_{i-\frac{1}{2}}; x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}; y_{j+\frac{1}{2}}]$ at time t^n:

$$\left[\phi_{i,j}^n\right]^{xy} = \frac{1}{\Delta x\, \Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \left[\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \phi(x, y, t^n)\, dx \right] dy,$$

SYMMETRIC SEQUENCES		
Optimal 3rd-order ($n = 6$)	$a_1 = b_3 = +0.91966152301739986,$ $a_2 = b_2 = -0.18799161879915978,$ $a_3 = b_1 = +0.26833009578175993$	
Forest, Ruth [36] 4th-order ($n = 7$)	$a_1 = a_4 = (2 + 2^{1/3} + 2^{-1/3})/6,$ $a_2 = a_3 = (1 - 2^{1/3} - 2^{-1/3})/6,$	$b_1 = b_3 = (2 - 2^{1/3})^{-1},$ $b_2 = (1 - 2^{2/3})^{-1}$
Optimal 4th-order ($n = 9$)	$a_1 = a_5 = 0.5,$ $a_2 = a_4 = -0.27516060407455222,$ $a_3 = a_2$	$b_1 = b_4 = -0.05032120814910445,$ $b_2 = b_3 = +0.55032120814910445,$
Yoshida [95] 6th-order ($n = 15$)	$a_1 = a_8 = +0.3922568052387787,$ $a_2 = a_7 = +0.5100434119184577,$ $a_3 = a_6 = -0.4710533854097564,$ $a_4 = a_5 = +0.0687531682525201,$	$b_1 = b_7 = +0.7845136104775573,$ $b_2 = b_6 = +0.2355732133593581,$ $b_3 = b_5 = -1.1776799841788710,$ $b_4 = +1.3151863206839112$
UNSYMMETRIC SEQUENCES		
Forest, Ruth [36] 3rd-order ($n = 6$)	$a_1 = 7/24,$ $a_2 = 3/4,$ $a_3 = -1/24,$	$b_1 = 2/3,$ $b_2 = -2/3,$ $b_3 = 1$
Optimal 3rd-order ($n = 6$)	$a_1 = +0.28321919245984298,$ $a_2 = +0.78867513459481288,$ $a_3 = +0.07189432705465587,$	$b_1 = +0.65323862309023541,$ $b_2 = -0.44191375768504829,$ $b_3 = +0.78867513459481288$
McLachlan [61] 4th-order ($n = 8$)	$a_1 = +0.515352837431122936,$ $a_2 = -0.085782019412973646,$ $a_3 = +0.441583023616466524,$ $a_4 = +0.128846158365384185,$	$b_1 = +0.1344961992774310892,$ $b_2 = -0.2248198030794208058,$ $b_3 = +0.7563200005156682911,$ $b_4 = +0.3340036032863214255$
McLachlan [61] 5th-order ($n = 12$)	$a_1 = +0.339839625839110000,$ $a_2 = -0.088601336903027329,$ $a_3 = +0.585856476825962118,$ $a_4 = -0.603039356536491888,$ $a_5 = +0.323580796554697639,$ $a_6 = +0.442363794219749458,$	$b_1 = +0.119390029287567275,$ $b_2 = +0.698927370382475230,$ $b_3 = -0.171312358271600775,$ $b_4 = +0.401269502251353448,$ $b_5 = +0.010705081848235984,$ $b_6 = -0.058979625498031163$

Table 4.3: Some symmetric and unsymmetric splitting sequences from third to sixth-order accuracy.

- $\left[\phi_{i,j}^n\right]^x$ and $\left[\phi_{i,j}^n\right]^y$ respectively denote one-dimensional cell averages of ϕ over $[x_{i-\frac{1}{2}};x_{i+\frac{1}{2}}]$ with $y=y_j$ and over $[y_{j-\frac{1}{2}};y_{j+\frac{1}{2}}]$ with $x=x_i$, both at time t^n:

$$\left[\phi_{i,j}^n\right]^x = \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \phi(x,y_j,t^n)\,dx \quad \text{and} \quad \left[\phi_{i,j}^n\right]^y = \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \phi(x_i,y,t^n)\,dy.$$

Remark 14. The manipulations described in this section only need to be performed for orders of accuracy that are strictly greater than two. This is related to the fact that cell averages are a second-order approximation of cell-centered values in the case of smooth functions (see remark 6 page 62) which implies that:

$$\left[\phi_{i,j}^n\right]^{xy} = \left[\phi_{i,j}^n\right]^x + \mathcal{O}\left(\Delta y^2\right) \quad \text{and} \quad \left[\phi_{i,j}^n\right]^{xy} = \left[\phi_{i,j}^n\right]^y + \mathcal{O}\left(\Delta x^2\right).$$

Reconstruction on conservative variables

In the two-dimensional case, the approximate solution is known along the j-th line at time t^n as a set of two-dimensional cell averages of conserved variables:

$$\left[\mathbf{U}_{i,j}^n\right]^{xy} \text{ for } i \in \{1,\dots,\mathrm{I}\}$$

where $\mathbf{U}=(\rho,\rho\mathbf{u},\rho e)$. But the one-dimensional Lagrange-remap schemes that have been developed operate on the following set of one-dimensional cell averages:

$$\left[\mathbf{U}_{i,j}^n\right]^{x} \text{ for } i \in \{1,\dots,\mathrm{I}\}$$

and they return:

$$\left[\mathbf{U}_{i,j}^{n+1}\right]^{x} \text{ for } i \in \{1,\dots,\mathrm{I}\}.$$

For each dimensional sweep, the Lagrange-remap scheme therefore has to be encapsulated between an interpolation step and a reconstruction step that both resort to *conservative* high-order accurate operators, respectively denoted by Π and Π^{-1} in the sequel. These aim to transform the input two-dimensional cell averages into one-dimensional cell averages and then revert the output one-dimensional cell averages back to two-dimensional cell averages. More precisely, they have to satisfy the following accuracy and conservativity conditions (note that the conservativity condition does not take boundary conditions into account):

$$\text{(accuracy condition)} \qquad \begin{cases} \left[\mathbf{U}_{i,j}^n\right]^x = \Pi\left(\left[\mathbf{U}_{i,j}^n\right]^{xy}\right) + \mathcal{O}\left(\Delta x^N\right) \\ \left[\mathbf{U}_{i,j}^{n+1}\right]^{xy} = \Pi^{-1}\left(\left[\mathbf{U}_{i,j}^{n+1}\right]^{x}\right) + \mathcal{O}\left(\Delta x^N\right) \end{cases} \quad \forall i \in \{1,\dots,\mathrm{I}\}, \tag{4.37}$$

$$\text{(conservativity condition)} \qquad \sum_{i,j}\left[\mathbf{U}_{i,j}^{n+1}\right]^{xy} = \sum_{i,j}\left[\mathbf{U}_{i,j}^{n}\right]^{xy}, \tag{4.38}$$

where N is the desired order of accuracy. Moreover, the interpolation and reconstruction operators have to be compatible with negative time steps. We therefore propose the following discrete *centered* operators that both satisfy these conditions:

$$\Pi\left(\left[\phi_{i,j}^n\right]^{xy}\right) = p_0\left[\phi_{i,j}^n\right]^{xy} + \sum_{k=1}^{r} p_k\left(\left[\phi_{i+k,j}^n\right]^{xy} + \left[\phi_{i-k,j}^n\right]^{xy}\right), \tag{4.39a}$$

$$\Pi^{-1}\left(\left[\phi_{i,j}^n\right]^{x}\right) = q_0\left[\phi_{i,j}^n\right]^{x} + \sum_{k=1}^{r} q_k\left(\left[\phi_{i+k,j}^n\right]^{x} + \left[\phi_{i-k,j}^n\right]^{x}\right), \tag{4.39b}$$

Order	p coefficients				q coefficients			
	$k=0$	$k=1$	$k=2$	$k=3$	$k=0$	$k=1$	$k=2$	$k=3$
$N=3$	$\frac{13}{12}$	$-\frac{1}{24}$			$\frac{11}{12}$	$\frac{1}{24}$		
$N=5$	$\frac{1067}{960}$	$-\frac{29}{480}$	$\frac{3}{640}$		$\frac{863}{960}$	$\frac{77}{1440}$	$-\frac{17}{5760}$	
$N=7$	$\frac{30251}{26880}$	$-\frac{7621}{107520}$	$\frac{159}{17920}$	$-\frac{5}{7168}$	$\frac{215641}{241920}$	$\frac{6361}{107520}$	$-\frac{281}{53760}$	$\frac{367}{967680}$

Table 4.4: p and q coefficients used to build the Π and Π^{-1} operators in (4.39a) and (4.39b).

where $r = \lfloor N/2 \rfloor$, coefficients p and q being given in Table (4.2.2). Note that the Π interpolator has already been used to transform cell averages into cell centered point-wise values in the construction of Lagrangian fluxes, see (4.15).

This approach however presents a major drawback. Like any high-order polynomial reconstruction, the discrete Π operator may return unphysical values in the vicinity of discontinuities due to Gibbs oscillations (this may also be true for the inverse operator). In this case, it would be temptating to apply the following procedure:

$$\left[\mathbf{U}_{i,j}^n\right]^x = \begin{cases} \left[\mathbf{U}_{i,j}^n\right]^{xy} & \text{if any physical criterion is not satisfied,} \\ \Pi\left(\left[\mathbf{U}_{i,j}^n\right]^{xy}\right) & \text{else,} \end{cases} \tag{4.40}$$

in order to prevent pathologic values from appearing but this is in fact a bad solution for two reasons.

- First, such a procedure is quite expensive in terms of computational time since a conditional test must be performed on each cell.
- Above all, the conservativity condition (4.38) would not be satisfied anymore (practically, it would be satisfied with second-order accuracy), making our finite volume schemes lose one of their most important features. This point is detailed in Proposition 7 below.

Proposition 7. *The conservativity condition* (4.38) *is exactly preserved by the discrete operators* Π *and* Π^{-1} *but it is only preserved at second-order accuracy if the procedure* (4.40) *is used in one or more cells.*

Proof. For the sake of simplicity, we restrict this proof to the case of a one-dimensional dataset denoted by (u_i) and we do not discuss the issue of boundary conditions. Let S denote the sum of the variable u over the considered dataset:

$$S = \sum u_i.$$

We now define $\hat{u} = \Pi(u)$ and $\tilde{u}$ that is obtained by computing $\Pi(u)$ in all cells except one (denoted by the j subscript in the sequel) where the procedure (4.40) is used. We also introduce the corresponding sums $\widehat{S}$ and $\widetilde{S}$. On the one hand, one can notice that:

$$\widehat{S} = \sum_i \hat{u}_i = \sum_i \left(\sum_{k=-r}^{r} p_{|k|} u_{i+k} \right) = \sum_{k=-r}^{r} p_{|k|} \left(\sum_i u_{i+k} \right) = \sum_{k=-r}^{r} p_{|k|} S = S$$

since the sum of all p_k coefficients is always equal to 1. This shows that the conservativity condition is preserved if the discrete operator Π is used in all cells. On the other hand, the sum $\widetilde{S}$ of the $\tilde{u}$ dataset is given by:

$$\begin{aligned}
\tilde{S} = \sum_i \tilde{u}_i &= u_j + \sum_{i \neq j} \left(\sum_{k=-r}^{r} p_{|k|} u_{i+k} \right) \\
&= u_j + \sum_{k=-r}^{r} p_{|k|} \left(\sum_{i \neq j} u_{i+k} \right) \\
&= u_j + \sum_{k=-r}^{r} p_{|k|} \left[\left(\sum_{i} u_{i+k} \right) - u_{j+k} \right] \\
&= \left(u_j - \sum_{k=-r}^{r} p_{|k|} u_{j+k} \right) + \sum_{k=-r}^{r} p_{|k|} S \\
&= \left(u_j - \sum_{k=-r}^{r} p_{|k|} u_{j+k} \right) + S \\
&= (u_j - \Pi(u_j)) + S.
\end{aligned}$$

Clearly, $\tilde{S}$ is not equal to S. The sum $\tilde{S}$ is actually a second-order approximation of S if u is a smooth function since (see remark 6 page 62):

$$\Pi(u_j) = u_j + \mathcal{O}\left(\Delta x^2\right).$$

As a matter of fact, the conservativity condition (4.38) is preserved at best at second-order accuracy if the procedure (4.40) is applied. □

Note that the issue described in Proposition 7 would appear with any interpolation method that does not operate on the same stencil for all cells. It therefore disqualifies all adaptative methods and in particular ENO and WENO techniques.

Reconstruction on fluxes

Another less intuitive approach consists in performing a reconstruction step on fluxes. Indeed, since finite volume schemes are conservative by construction, such an approach guarantees conservativity preservation whatever polynomial reconstruction method is chosen. More precisely, the idea consists is directly updating two-dimensional cell averages in both Lagrangian and remap steps by integrating the underlying one-dimensional schemes (4.27) and (4.20) over $[y_{j-\frac{1}{2}}; y_{j+\frac{1}{2}}]$. The resulting Lagrangian scheme then writes:

$$\left[\begin{pmatrix} (\rho_0 u_x)_{i,j}^{n+1} \\ (\rho_0 e)_{i,j}^{n+1} \end{pmatrix} \right]^{xy} = \left[\begin{pmatrix} (\rho_0 u_x)_{i,j}^{n} \\ (\rho_0 e)_{i,j}^{n} \end{pmatrix} \right]^{xy} - \frac{\Delta t}{\Delta x} \left(\mathbf{F}^*_{i+\frac{1}{2},j} - \mathbf{F}^*_{i-\frac{1}{2},j} \right) \tag{4.41}$$

whereas the remap scheme is given by:

$$\left[(\rho\phi)_{i,j}^{n+1} \right]^{xy} = \left[(\rho_0\phi)_{i,j}^{n+1} \right]^{xy} - \frac{\Delta t}{\Delta x} \left(\mathbf{R}^*_{i+\frac{1}{2},j} - \mathbf{R}^*_{i-\frac{1}{2},j} \right). \tag{4.42}$$

with $\phi \in \{1, \mathbf{u}, e\}$. The flux terms involved in (4.41) and (4.42) write:

$$\mathbf{F}^*_{i+\frac{1}{2},j} = \frac{1}{\Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \mathbf{F}^*_{i+\frac{1}{2}}(y)\, dy, \tag{4.43}$$

$$\mathbf{R}^*_{i+\frac{1}{2},j} = \frac{1}{\Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \left(u^*_{i+\frac{1}{2}} (\rho\phi)^*_{i+\frac{1}{2}} \right)(y)\, dy, \tag{4.44}$$

i.e. these are cell averages of the one-dimensional Lagrangian and remap fluxes $\mathbf{F}^*$ and $u^*(\rho\phi)^*$ along the transverse y direction. They can thus be computed using the Π^{-1} conversion function:

$$\mathbf{F}^*_{i+\frac{1}{2},j} = q_0 \mathbf{F}^*_{i+\frac{1}{2}}(y_j) + \sum_{k=1}^{r} q_k \left(\mathbf{F}^*_{i+\frac{1}{2}}(y_{j+k}) + \mathbf{F}^*_{i+\frac{1}{2}}(y_{j-k}) \right), \tag{4.45}$$

$$\mathbf{R}^*_{i+\frac{1}{2},j} = q_0 \left(u^*_{i+\frac{1}{2}} (\rho\phi)^*_{i+\frac{1}{2}} \right)(y_j) + \sum_{k=1}^{r} q_k \left[\left(u^*_{i+\frac{1}{2}} (\rho\phi)^*_{i+\frac{1}{2}} \right)(y_{j+k}) + \left(u^*_{i+\frac{1}{2}} (\rho\phi)^*_{i+\frac{1}{2}} \right)(y_{j-k}) \right]. \tag{4.46}$$

This approach is much more flexible than the previous reconstruction on conservative variables. Indeed, assume that the two-dimensional fluxes generate unphysical values due to one of the reconstruction steps (4.45) - (4.46), their computation can easily be avoided so that conservative variables are updated using one-dimensional fluxes on pathologic interfaces and, even in this case, conservativity will not be affected. Such a treatment does not solve the performance issue mentioned previously but at least improves robustness.

Summarized two-dimensional scheme with reconstruction on fluxes

To complete the description of dimensionally split GoHy schemes, we finally summarize the different steps that have to be performed for each dimensional sweep.

1. Transform the input two-dimensional cell averages of conservative variables into one-dimensional cell averages by applying the interpolating function Π along the transverse direction. Note that the resulting set of data will only be used to compute Lagrangian fluxes since we perform the reconstruction step on Lagrangian fluxes.

2. Compute the one-dimensional Lagrangian fluxes using the previously computed one-dimensional cell averages as explained in section 4.1.2.

3. Compute updated positions of the Eulerian grid using one-dimensional flux terms u^* according to (4.17).

4. Reconstruct two-dimensional Lagrangian fluxes according to (4.45).

5. Update conservative variables by applying the Lagrangian scheme (4.41).

6. Transform updated two-dimensional conservative variables into one-dimensional cell averages by applying the interpolation function Π along the transverse direction. Note that the resulting set of data will only be used to compute remap fluxes since we perform the reconstruction step on remap fluxes.

7. Compute one-dimensional remap fluxes as explained in section 4.1.3.

8. Reconstruct two-dimensional remap fluxes according to (4.46).

9. Update conservative variables by applying the remap scheme (4.42).

4.3 Improving robustness and symmetry preservation: artificial viscosity techniques

High-order dimensionally split numerical schemes encounter two major issues in practical computations that we describe below. We then propose an original method for dealing with them.

1. First of all, it is well-known that high-order schemes are subject to unphysical Gibbs oscillations in presence of discontinuities which may cause robustness issues. These are often controlled by slope or flux limiting techniques but such approaches usually impose to perform many conditional tests that have a negative impact on computational performance. That's why some sophisticated interpolation methods (like WENO techniques that have been presented in section 3.2) have been designed. These aim to replace conditional tests by numerical computations, for example by computing smoothness indicators in the case of WENO interpolations. This strategy is however still very expensive, especially in the multidimensional case.

2. The second issue that we may encounter comes from dimensional splitting techniques on Cartesian grids. Numerical experiments have indeed shown that these may cause symmetry losses when dealing with cylindrically symmetric problems, especially along the horizontal and vertical directions. This is a major issue for inertial confinement fusion applications since some unphysical instabilities may develop due to these symmetry flaws.

In this study, we have chosen to treat these two phenomena by adding artificial viscosity terms to the Lagrangian step. More precisely, Gibbs oscillations are controlled using an hyperviscosity model inspired from the large eddy simulations literature [21, 20] while symmetry preservation is achieved thanks to a first-order artificial viscosity model. This approach presents two main advatanges. On the one hand, its computational cost is relatively low as we will see in the sequel. On the other hand, artificial viscosity provides a fully centered scheme that easily handles negative time steps.

Practically, the idea is to consider the Navier-Stokes equations, *i.e.* the Euler system of equations with an additional viscous right hand side:

$$\partial_t \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ \rho e \end{pmatrix} + \nabla\cdot \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\otimes\mathbf{u} + p\underline{\mathbf{I}} \\ (\rho e + p)\mathbf{u} \end{pmatrix} = \nabla\cdot \begin{pmatrix} 0 \\ \underline{\boldsymbol{\tau}} \\ \underline{\boldsymbol{\tau}}\mathbf{u} + \kappa\nabla T \end{pmatrix}, \tag{4.47}$$

$\underline{\boldsymbol{\tau}}$ denoting the symmetric viscous stress tensor:

$$\begin{aligned} \underline{\boldsymbol{\tau}} &= \nu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) + \left(\beta - \frac{2}{3}\nu\right)(\nabla\cdot\mathbf{u})\underline{\mathbf{I}} \\ &= \begin{bmatrix} a\,\partial_x u_x + b\,\partial_y u_y & \nu(\partial_x u_y + \partial_y u_x) & \nu\partial_x u_z \\ & a\,\partial_y u_y + b\,\partial_x u_x & \nu\partial_y u_z \\ & & b\,(\partial_x u_x + \partial_y u_y) \end{bmatrix} \text{ where } \begin{cases} a = \left(\beta + \frac{4}{3}\nu\right), \\ b = \left(\beta - \frac{2}{3}\nu\right). \end{cases} \end{aligned}$$

One can notice that the right hand side involves three physical constants: the bulk viscosity β, the shear viscosity ν and the thermal conductivity κ. Artificial viscosity techniques consist in replacing these constants by artificial ones (marked with stars in the sequel) that vanish when the mesh size goes to zero so that consistency with the initial sytem of equations (4.1) is preserved. We have focused on two different methods for designing such artificial constants that we detail below.

4.3.1 Controlling oscillations: high-order hyperviscosity

We introduce in this section an hyperviscosity model inspired from previous works by Cook [20] in the area of large eddy simulations that allows to control Gibbs oscillations without sacrifying high-order accuracy. Defining:

$$S = \left\| \nabla \mathbf{u} + \nabla \mathbf{u}^T \right\|_2$$

the artificial constants of the hyperviscosity model are given by:

$$\begin{aligned} \beta^\star &= C_\beta \cdot \langle \rho \left| \nabla^r S \right| \rangle \cdot \Delta^{r+2}, \\ \nu^\star &= C_\nu \cdot \langle \rho \left| \nabla^r S \right| \rangle \cdot \Delta^{r+2}, \\ \kappa^\star &= C_\kappa \cdot \left\langle \tfrac{\rho c}{T} \left| \nabla^r \epsilon \right| \right\rangle \cdot \Delta^{r+1}, \end{aligned} \tag{4.48}$$

where Δ is the mesh size along the considered direction and r is an integer parameter.

Remark 15. We determine the r parameter so that high-order accuracy is preserved. Practically, the following values are taken:

$$\begin{cases} r = 2 & \text{for the second and third-order schemes,} \\ r = 4 & \text{for the fourth-order scheme.} \end{cases}$$

Remark 16. The artificial constants given in (4.48) tend to zero when the mesh is refined:

$$\beta^\star,\ \nu^\star,\ \kappa^\star \underset{\Delta \to 0}{\longrightarrow} 0.$$

The choice of the r parameter given by remark 15 imposes to compute the Laplacian (if $r = 2$) or the bi-Laplacian (if $r = 4$) of S and ϵ. We practically use the following centered discrete operators:

$$\left(\nabla^r \phi\right) i, j = \begin{cases} \dfrac{\phi_{i+1,j} + \phi_{i-1,j} - 2\phi_{i,j}}{\Delta x^2} + \dfrac{\phi_{i,j+1} + \phi_{i,j-1} - 2\phi_{i,j}}{\Delta y^2} & \text{if } r = 2, \\ \left(\nabla^2 \left(\nabla^2 \phi\right)\right)_{i,j} & \text{if } r = 4. \end{cases}$$

The $\langle \cdot \rangle$ operator denotes a truncated Gaussian filter:

$$\langle \phi(\mathbf{x}_0) \rangle = \int_{-3\Delta}^{+3\Delta} \mathcal{G}(|\mathbf{x}|) \cdot \phi\left(|\mathbf{x} - \mathbf{x}_0|\right) d\mathbf{x} \quad \text{with} \quad \mathcal{G}(x) = \exp\left(-2x/\Delta\right)^2 \Big/ \int_{-3\Delta}^{+3\Delta} \exp\left(-2x/\Delta\right)^2 \, dx$$

which can be discretized using the following formula [20]:

$$\langle \phi \rangle_{i,j} = \sum_{|k| \leq 4} \sum_{|l| \leq 4} f_{|k|} f_{|l|} \phi_{i+k,j+l} \text{ with } f_0 = \frac{3565}{10368},\ f_1 = \frac{3091}{12960},\ f_2 = \frac{1997}{25920},\ f_3 = \frac{149}{12960},\ f_4 = \frac{107}{103680}.$$

Constants C_β, C_ν and C_κ are user-defined coefficients that can be adjusted to increase or decrease the impact of hyperviscosity. Since the time scheme is fully explicit, this high-order viscosity model implies the following time step constraint:

$$\Delta t \leq \frac{1}{2} \min\left(\frac{\rho \Delta^2}{\beta^\star}, \frac{\rho \Delta^2}{\nu^\star}, \frac{\rho c^2 \Delta^2}{\kappa^\star T} \right) \tag{4.49}$$

but numerical experiments have shown that it usually does not lead to significantly lower time steps than the stability condition (4.26). We therefore do not take constraint (4.49) into account in most practical computations.

4.3.2 Symmetry preservation: first-order artificial viscosity

The first-order artificial viscosity model presented hereafter that we use to treat symmetry issues relies on the stability condition that naturally arises with explicit Navier-Stokes solvers. We have seen in the previous section that it writes:

$$\Delta t \leq \min\left(\frac{\rho\Delta^2}{\beta}, \frac{\rho\Delta^2}{\nu}, \frac{\rho c^2\Delta^2}{\kappa T}\right). \tag{4.50}$$

where Δ denotes the mesh size along the considered direction. We would like to design artificial constants $\beta^\star$, $\nu^\star$ and $\kappa^\star$ so that stability condition (4.50) is automatically satisfied provided that the CFL condition for the Lagrangian scheme holds:

$$\Delta t \leq \frac{1}{2}\min\left(\frac{\Delta}{c}\right).$$

Stability will necessarily be insured if:

$$\min\left(\frac{\Delta}{c}\right) \leq \min\left(\frac{\rho\Delta^2}{\beta^\star}, \frac{\rho\Delta^2}{\nu^\star}, \frac{\rho c^2\Delta^2}{\kappa^\star T}\right).$$

The following choice of $\beta^\star$, $\nu^\star$ and $\kappa^\star$ is therefore suitable:

$$\beta^\star = C_\beta(\rho c)\Delta, \quad \nu^\star = C_\nu(\rho c)\Delta, \quad \kappa^\star = C_\kappa\left(\frac{\rho c^3}{T}\right)\Delta, \tag{4.51}$$

C_β, C_ν and C_κ being user-defined constants that must be taken between 0 and 1. Such an artificial viscosity model is of course very dissipative and may noticeably degrade the shock-capturing features of GoHy schemes. In order to keep sharp shock profiles, first-order artificial viscosity may be reduced in compression zones by using a filter. The one used in this work writes:

$$\begin{aligned}
\beta^\star &= F(\nabla\cdot\mathbf{u})\cdot C_\beta(\rho c)\Delta \\
\nu^\star &= F(\nabla\cdot\mathbf{u})\cdot C_\nu(\rho c)\Delta \\
\kappa^\star &= F(\nabla\cdot\mathbf{u})\cdot C_\kappa\left(\tfrac{\rho c^3}{T}\right)\Delta
\end{aligned} \qquad \text{with} \quad F(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ \exp\left(-400\left(\frac{x}{\max|x|}\right)^2\right) & \text{if } x \leq 0. \end{cases}$$

4.3.3 Detailed viscosity fluxes

Now that the artificial constants have been designed, we can integrate them into viscosity terms which will be added to the Lagrangian step. Like Lagrangian and remap fluxes, viscosity fluxes are treated in a dimensionally split way, leading to the following modified Lagrangian scheme along the x direction:

$$\begin{pmatrix} \overline{(\rho_0 u_x)}^{n+1}_{i,j} \\ \overline{(\rho_0 u_y)}^{n+1}_{i,j} \\ \overline{(\rho_0 e)}^{n+1}_{i,j} \end{pmatrix} = \begin{pmatrix} \overline{(\rho_0 u_x)}^{n}_{i,j} \\ \overline{(\rho_0 u_y)}^{n}_{i,j} \\ \overline{(\rho_0 e)}^{n}_{i,j} \end{pmatrix} - \frac{\Delta t}{\Delta x}\left[\begin{pmatrix} p^*_{i+\frac{1}{2},j} \\ 0 \\ (pu_x)^*_{i+\frac{1}{2},j} \end{pmatrix} - \begin{pmatrix} p^*_{i-\frac{1}{2},j} \\ 0 \\ (pu_x)^*_{i-\frac{1}{2},j} \end{pmatrix} + \mathbf{V}^*_{i+\frac{1}{2},j} - \mathbf{V}^*_{i-\frac{1}{2},j}\right],$$

and to the following one along the y direction:

$$\begin{pmatrix} \overline{(\rho_0 u_x)}^{n+1}_{i,j} \\ \overline{(\rho_0 u_y)}^{n+1}_{i,j} \\ \overline{(\rho_0 e)}^{n+1}_{i,j} \end{pmatrix} = \begin{pmatrix} \overline{(\rho_0 u_x)}^{n}_{i,j} \\ \overline{(\rho_0 u_y)}^{n}_{i,j} \\ \overline{(\rho_0 e)}^{n}_{i,j} \end{pmatrix} - \frac{\Delta t}{\Delta y}\left[\begin{pmatrix} 0 \\ p^*_{i,j+\frac{1}{2}} \\ (pu_y)^*_{i,j+\frac{1}{2}} \end{pmatrix} - \begin{pmatrix} 0 \\ p^*_{i,j-\frac{1}{2}} \\ (pu_y)^*_{i,j-\frac{1}{2}} \end{pmatrix} + \mathbf{V}^*_{i,j+\frac{1}{2}} - \mathbf{V}^*_{i,j-\frac{1}{2}}\right].$$

We recall that the artificial constants $\beta^\star$, $\nu^\star$ and $\kappa^\star$ given by (4.48) have been designed so that they tend quickly to zero when the mesh is refined (see remark 16). The viscosity fluxes therefore satisfy the following property:

$$\mathbf{V}^*_{i+\frac{1}{2},j} = \mathcal{O}\left(\Delta^{r+1}\right) \quad \text{and} \quad \mathbf{V}^*_{i,j+\frac{1}{2}} = \mathcal{O}\left(\Delta^{r+1}\right).$$

As a matter of fact, there is no need to build high-order accurate viscosity fluxes. We rather propose the following viscosity terms which are physically suitable and can be efficiently computed:

$$\mathbf{V}^*_{i+\frac{1}{2},j} = \frac{1}{2}\begin{pmatrix} \underline{\boldsymbol{\tau}}^{xx}_{i,j} + \underline{\boldsymbol{\tau}}^{xx}_{i+1,j} \\ \underline{\boldsymbol{\tau}}^{xy}_{i,j} + \underline{\boldsymbol{\tau}}^{xy}_{i+1,j} \\ \left(\underline{\boldsymbol{\tau}}^{xx}u_x + \underline{\boldsymbol{\tau}}^{xy}u_y\right)_{i,j} + \left(\underline{\boldsymbol{\tau}}^{xx}u_x + \underline{\boldsymbol{\tau}}^{xy}u_y\right)_{i+1,j} + \kappa^\star_{i,j}\left(\frac{T_{i+1,j}-T_{i-1,j}}{2\Delta x}\right) + \kappa^\star_{i+1,j}\left(\frac{T_{i+2,j}-T_{i,j}}{2\Delta x}\right) \end{pmatrix},$$

$$\mathbf{V}^*_{i,j+\frac{1}{2}} = \frac{1}{2}\begin{pmatrix} \underline{\boldsymbol{\tau}}^{xy}_{i,j} + \underline{\boldsymbol{\tau}}^{xy}_{i,j+1} \\ \underline{\boldsymbol{\tau}}^{yy}_{i,j} + \underline{\boldsymbol{\tau}}^{yy}_{i,j+1} \\ \left(\underline{\boldsymbol{\tau}}^{xy}u_x + \underline{\boldsymbol{\tau}}^{yy}u_y\right)_{i,j} + \left(\underline{\boldsymbol{\tau}}^{xy}u_x + \underline{\boldsymbol{\tau}}^{yy}u_y\right)_{i,j+1} + \kappa^\star_{i,j}\left(\frac{T_{i,j+1}-T_{i,j-1}}{2\Delta y}\right) + \kappa^\star_{i,j+1}\left(\frac{T_{i,j+2}-T_{i,j}}{2\Delta y}\right) \end{pmatrix},$$

the approximate strain rate tensor being given by:

$$\begin{aligned}
(\underline{\boldsymbol{\tau}}_{xx})_{i,j} &= \left(\beta^\star_{i,j} + \frac{4}{3}\nu^\star_{i,j}\right)\frac{(u_x)_{i+1,j}-(u_x)_{i-1,j}}{2\Delta x} + \left(\beta^\star_{i,j} - \frac{2}{3}\nu^\star_{i,j}\right)\frac{(u_y)_{i,j+1}-(u_y)_{i,j-1}}{2\Delta y} \\
(\underline{\boldsymbol{\tau}}_{yy})_{i,j} &= \left(\beta^\star_{i,j} + \frac{4}{3}\nu^\star_{i,j}\right)\frac{(u_y)_{i,j+1}-(u_y)_{i,j-1}}{2\Delta y} + \left(\beta^\star_{i,j} - \frac{2}{3}\nu^\star_{i,j}\right)\frac{(u_x)_{i+1,j}-(u_x)_{i-1,j}}{2\Delta x} \\
(\underline{\boldsymbol{\tau}}_{xy})_{i,j} &= \nu^\star_{i,j}\left(\frac{(u_y)_{i+1,j}-(u_y)_{i-1,j}}{2\Delta x} + \frac{(u_x)_{i,j+1}-(u_x)_{i,j-1}}{2\Delta y}\right) \\
(\underline{\boldsymbol{\tau}}_{xz})_{i,j} &= \nu^\star_{i,j}\left(\frac{(u_z)_{i+1,j}-(u_z)_{i-1,j}}{2\Delta x}\right) \\
(\underline{\boldsymbol{\tau}}_{yz})_{i,j} &= \nu^\star_{i,j}\left(\frac{(u_z)_{i,j-1}-(u_z)_{i,j-1}}{2\Delta y}\right).
\end{aligned}$$

Practically, one has to compute both artificial constants and viscosity fluxes at the beginning of each dimensional sweep. These manipulations depart from the cache-oblivious implementation of Lagrangian and remap fluxes since they involve estimations along both directions (e.g. approximate space derivatives and filtering operators for hyperviscosity techniques) but they revealed quite cheap in practice. Artificial viscosity increases the computational cost of GoHy schemes of about 25% for the second-order scheme and 20% for the third and fourth-order schemes.

Remark 17. We only have detailed viscosity fluxes for the two-dimensional case but these can of course be adapted to the one-dimensional case by removing all y-wise discrete terms.

4.4 High performance computing aspects

The Lagrange-remap schemes proposed in the previous sections have been built so that they are able to take advantage of modern and massively parallel computer architectures. We focused our attention on the following two main points to reach this objective.

1. **Allowing cache-oblivious implementation**. Dimensional splitting has made it possible to implement the GoHy schemes in a cache-oblivious way. Indeed, most computations are performed on data that are memory-contiguous and therefore benefit from low-latency cache memory accesses. We propose several single core performance tests in section 4.4.1 in order to illustrate this feature.

2. **Restricting the number of communications sweeps in parallel context.** As we will see in section 4.4.2, the code that implements GoHy schemes has been parallelized using a domain decomposition method and the Message Passing Interface (MPI) library. Some efforts have been made to lower the number of communications phases between processes and thus increase parallel efficiency. In particular, unlike Runge-Kutta methods, GoHy schemes rely on a direct temporal scheme which does not require subcycling (and thus additional communications) to achieve high-order accuracy in time.

4.4.1 Performance figures

We first focus on estimating the single core performance of GoHy schemes. To that end, we measure the number of floating point operations that are performed for a given problem. There are differents methods for counting floating point operations practically. The most commonly used consists in resorting to hardware counters but these sometimes reveal unreliable and we therefore adopt a different approach. We have chosen to overload all functions that operate on floating point values with modified functions that increment a counter according to the scale given in [62] and recalled in Table 4.5. Of course, overloading these operators noticeably modifies the execution time and we consequently need to run our solver twice to measure its performance:

- once without overloading to get the restitution time;
- once with overloaded operators to count floating point operations.

Practical tests have been performed on an Intel Xeon E5650 CPU @ 2.66 Ghz based on the Nehalem architecture (theoretical peak performance: 10.64 GFLOPS). We have run a 100 cycles long simulation with different mesh sizes on the advected vortex test problem described in section 4.5.1. The hyperviscosity model has been enabled so that the considered benchmark is as representative as possible of real life applications. Both execution times and measured performance have been reported in Table 4.7. These show satisfying results: the GoHy solver is indeed able to exploit 20% to 30% of a modern processor's peak performance whereas most scientific codes reach values of about 10%.

Remark 18. Note that the efficiency of GoHy schemes increases with their accuracy, which is a predictable result. Indeed, the number of floating point operations required for applying the whole Lagrange-remap scheme increases noticeably with the order of accuracy as shown in Table 4.6. The solver therefore spends more time in functions that respect the principles we stated in introduction (see page 13): these do not require any test and operate mostly on contiguous data, thus increasing the global efficiency.

4.4.2 Parallelization technique and parallel efficiency

We now describe how the GoHy solver has been parallelized and then provide several parallel efficiency figures. We recall that the implementation has been performed using the MPI library.

Domain decomposition method

We have chosen to resort to a classical domain decomposition method. Basically, the idea consists in splitting the computational domain in several smaller ones (preferably of the same size) and to assign each of

Operator	Description	Cost
`+ - += -=`	addition	1
`* *=`	product	1
`/ /=`	division	4
`== < > <= >=`	conditional tests	1
`fabs`	absolute value	1
`sqrt`	square root	4
`pow`	power	4
`sin`	sinus	8
`cos`	cosinus	8
`exp`	exponential	8
`log`	logarithm	8

Table 4.5: Chosen Weights for counting floating point operations (the cost is given in terms of CPU instructions). These are taken from [62].

Scheme	Number of FLOPS per cell
2nd-order	≈ 2900
3rd-order	≈ 6500
4th-order	≈ 14000

Table 4.6: Number of floating point operations required to apply the two-dimensional second, third and fourth-order GoHy schemes in a single cell.

SECOND-ORDER SCHEME				
Mesh size	**Execution time**		**Performance**	
N × N cells	wall time (s)	μs / cell / cycle	GFLOPS	% peak
N = 100	1.29	1.29	2.27	21.3%
N = 200	4.92	1.23	2.34	21.9%
N = 400	20.93	1.31	2.17	20.4%
N = 800	94.09	1.47	1.93	18.1%
THIRD-ORDER SCHEME				
Mesh size	**Execution time**		**Performance**	
N × N cells	wall time (s)	μs / cell / cycle	GFLOPS	% peak
N = 100	2.54	2.54	2.65	24.9%
N = 200	9.98	2.50	2.63	24.7%
N = 400	41.87	2.62	2.48	23.2%
N = 800	190.57	2.98	2.16	20.3%
FOURTH-ORDER SCHEME				
Mesh size	**Execution time**		**Performance**	
N × N cells	wall time (s)	μs / cell / cycle	GFLOPS	% peak
N = 100	4.99	4.99	2.99	28.0%
N = 200	19.93	4.98	2.89	27.1%
N = 400	82.03	5.13	2.76	25.9%
N = 800	365.55	5.71	2.45	23.0%

Table 4.7: Single core performance figures of the two-dimensional Lagrange-remap schemes. Benchmark: 100 simulation steps on the advected vortex test problem, hyperviscosity model enabled. Test platform: Intel Xeon E5650 CPU @ 2.66 GHz (theoretical peak performance: 2.66 GHz × 4 double precision floating point operations per cycle = 10.64 GFLOPS).

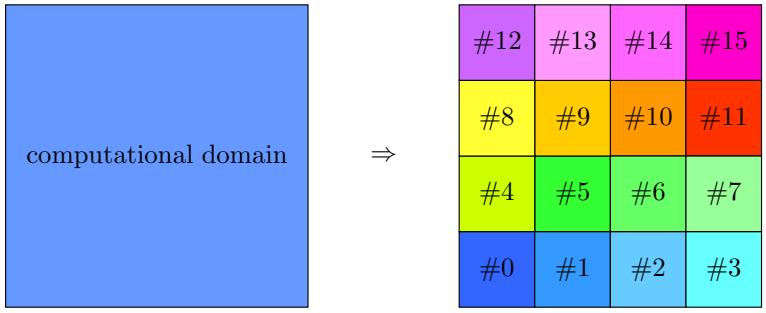


Figure 4.2: Principle of the domain decomposition method: the computational domain is here distributed over 16 different processes.

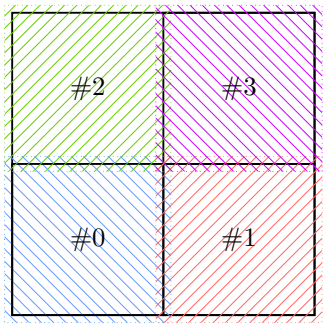


Figure 4.3: Example of overlapping zones with 4 processes. The physical computational domain is bounded by black lines but data must be allocated on larger domains on each process. There are consequently some overlapping zones on boundaries shared by several processes.

these subdomains to a different process as shown in Figure 4.2. Such a decomposition method corresponds to a SPMD (Single Programm, Multiple Data) parallel programming model: a single application is running, its data being distributed over all processes. Since GoHy schemes require ghost cells for performing some operations (e.g. high-order interpolations), processes share practically a few overlapping layers of cells as shown in Figure 4.3. The parallelized code finally mainly consists in the following communications phases:

- point-to-point communications (using `MPI_Send` / `MPI_Recv` instructions) with neighbour processes in order to fill ghost cells layers;
- one collective reducing operation (using `MPI_Allreduce` instructions) per cycle to set the time step and share it over all processes.

Parallel efficiency figures

In order to check if GoHy schemes are well-suited to parallel architectures, we propose some parallel efficiency results in the sequel. There are practically two methods for estimating parallel efficiency.

- The first one - the so-called *strong* measurement - consists in running the same computation in both sequential and parallel modes. Let T_{seq} denote the restitution time of the sequential run and T_p the

restitution time of the parallel one on p processes. Since the theoretical speed-up of the parallel computation over the sequential one is p, the strong parallel efficiency is given by:

$$\text{strong parallel efficiency} = \frac{pT_p}{T_{\text{seq}}}.$$

- The second one measures the so-called *weak* parallel efficiency. It consists in running on the one hand a sequential computation for a problem of size s and on the other hand a parallel computation on p processes for a problem of size $s \times p$. In this case, the theoretical speed-up is 1 and, using the above-defined notations, the weak parallel efficiency is given by:

$$\text{weak parallel efficiency} = \frac{T_p}{T_{\text{seq}}}.$$

In this study, we have restricted to the weak parallel efficiency measurement. Indeed, since the parallel efficiency measurements have been performed with up to 256 processes, we were not able to solve a large enough problem on a single process because of too high memory requirements. The test protocol is the following. Computations have been run on a calculator that is equipped with Intel Xeon E5462 @ 2.8 Ghz processors (Core 2 architecture) and an Infiniband interconnect @ 20 Gbit/s. We have run a 100 cycles long simulation with different mesh sizes on the advected vortex test problem described in section 4.5.1. The hyperviscosity model has been enabled so that the considered benchmark is as representative as possible of real life applications. The measured parallel efficiencies have been reported in Table 4.8. These are very satisfying: they are indeed above 95% in nearly all cases.

4.5 Numerical results

To conclude the presentation of GoHy high-order dimensionally split Lagrange-remap schemes, we provide in this section numerical results on various test problems taken from the literature. More precisely, we first exhibit convergence results on smooth test problems in order to show that the experimental order of convergence matches the theoretical one. We then propose simulation results obtained for several non-smooth test problems to prove the robustness of the schemes we developed.

4.5.1 Experimental order of convergence

We first focus on smooth test problems and carry out a convergence analysis. To that end, we consider two two-dimensional problems: Kidder's isentropic compression problem [53] and an advected vortex test problem proposed by Yee *et al.* [94]

Kidder's isentropic compression problem [53]

Kidder's test problem represents the isentropic compression of an ideal gas volume which is initially at rest (*i.e.* the initial velocity is zero). We here focus on the two-dimensional case in cylindrically symmetric geometry which actually describes the compression of a cylinder of gas. Let r_1 and r_2 denote the internal and external radiuses of the shell and ρ_l and p_l the initial densitiy and pressure at $r = r_l$ with $l = 1, 2$. The initial density profile is given by:

$$\rho_0(r) = \left(\frac{r_2^2 - r^2}{r_2^2 - r_1^2} \rho_1^{\gamma-1} + \frac{r^2 - r_1^2}{r_2^2 - r_1^2} \rho_2^{\gamma-1} \right)^{\frac{1}{\gamma-1}}$$

SECOND-ORDER SCHEME						
Mesh size / process	**Parallel efficiency for p processes**					
N × N cells	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
N = 200	100%	100%	99%	98%	98%	97%
N = 400	100%	99%	99%	98%	97%	96%
N = 800	100%	99%	98%	98%	98%	97%
THIRD-ORDER SCHEME						
Mesh size / process	**Parallel efficiency for p processes**					
N × N cells	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
N = 200	100%	99%	98%	98%	97%	97%
N = 400	100%	99%	99%	98%	97%	96%
N = 800	100%	100%	99%	99%	97%	96%
FOURTH-ORDER SCHEME						
Mesh size / process	**Parallel efficiency for p processes**					
N × N cells	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
N = 200	100%	98%	96%	96%	96%	95%
N = 400	99%	97%	96%	95%	95%	94%
N = 800	99%	98%	96%	96%	95%	95%

Table 4.8: Parallel efficiency of the two-dimensional Lagrange-remap schemes. Benchmark: 100 simulation steps on the advected vortex test problem, hyperviscosity model enabled. Test platform: Intel Xeon E5462 processors @ 2.8 GHz, InfiniBand interconnect @ 20 Gbit/s.

where $r^2 = x^2 + y^2$ and $\gamma = 2$ is the specific heats ratio. Since the compression is isentropic, the p/ρ^γ ratio is constant and thus $p_1\rho_2^\gamma = p_2\rho_1^\gamma$. Let c_l denote the Eulerian sound speed at $r = r_l$, we now define the shell focalization time τ:

$$\tau = \sqrt{\frac{\gamma-1}{2}\frac{r_2^2 - r_1^2}{c_2^2 - c_1^2}}.$$

which allows to write the complete analytical solution. Defining $h(t) = \sqrt{1-(t/\tau)^2}$, it is given by:

$$\rho(r,t) = \rho_0 \left(\frac{r}{h(t)}\right) \cdot h(t)^{\frac{2}{1-\gamma}}, \quad u(r,t) = -\frac{tr}{\tau^2 h(t)^2}, \quad p(r,t) = p_2 \left(\frac{\rho(r,t)}{\rho_2}\right)^\gamma.$$

Here, u denotes the velocity along the radial direction, the velocity components along the x and y directions therefore write:

$$\begin{aligned} u_x(r,t) &= \cos\theta \; u(r,t), \\ u_y(r,t) &= \sin\theta \; u(r,t), \end{aligned} \quad \text{with} \quad \theta = \operatorname{atan}(y/x).$$

For practical computations, we take $p_1 = 1$, $p_2 = 100$ and $\rho_2 = 1$. We have run simulations for $r \in [r_1; r_2] = [0; 1]$ until $t = 0.5\tau$ with a CFL coefficient of 0.9. The computational domain has wall boundary conditions along the x and y axis, the boundary conditions being imposed by the analytical solution elsewhere. In order to estimate the experimental order of convergence, we analyse how the error between the analytical and numerical solutions evolves when the mesh is refined. More precisely, the error is computed using a L^1 norm in space and time:

$$\text{error} = (\Delta x \cdot \Delta y \cdot \Delta t) \times \sum_{\phi \in \{1, u_x, u_y, e\}} \sum_{i,j} \sum_n \left| \overline{(\rho\phi)}_{i,j}^n - \left(\overline{(\rho\phi)}_{\text{exact}}\right)_{i,j}^n \right|.$$

Results have been reported in Table 4.5.1. These computations have been carried out with and without hyperviscosity. We notice that the experimental order of convergence perfectly matches the theoretical order (results obtained for the fourth-order scheme on 1600×1600 cells can not be trusted since the error is too close to the machine epsilon). Another satisfying point is that it is not affected by the presence of hyperviscosity.

WITHOUT HYPERVISCOSITY						
Mesh size	**2nd-order**		**3rd-order**		**4th-order**	
(N × N cells)	error	order	error	order	error	order
N = 100	4.26e-05		5.81e-07		2.92e-09	
N = 200	1.05e-05	2.02	7.10e-08	3.03	1.77e-10	4.04
N = 400	2.61e-06	2.01	8.79e-09	3.02	1.09e-11	4.02
N = 800	6.50e-07	2.00	1.09e-09	3.01	6.74e-13	4.02
N = 1600	1.62e-07	2.00	1.36e-10	3.00	5.37e-14	3.65*
WITH HYPERVISCOSITY ($C_\beta = C_\nu = C_\kappa = 1$)						
Mesh size	**2nd-order**		**3rd-order**		**4th-order**	
(N × N cells)	error	order	error	order	error	order
N = 100	5.97e-05		5.78e-07		2.85e-09	
N = 200	1.26e-05	2.24	7.09e-08	3.03	1.74e-10	4.04
N = 400	2.87e-06	2.13	8.78e-09	3.01	1.07e-11	4.02
N = 800	6.82e-07	2.07	1.09e-09	3.01	6.62e-13	4.02
N = 1600	1.66e-07	2.04	1.36e-10	3.00	5.27e-14	3.65*

Table 4.9: Error between the analytical and numerical solutions for Kidder's isentropic compression test problem. ($\star$) These results should not be trusted: they do not match the theoretical order because of machine round-off errors.

Isentropic advected vortex

We now carry out the same convergence study on a two-dimensional isentropic advected vortex test problem proposed by Yee *et al* [94]. The analytical solutions writes:

$$\rho(x,y,t) = \left(1 - \frac{25(\gamma-1)}{8\gamma\pi^2}\exp(1-r^2)\right)^{\frac{1}{\gamma-1}},$$
$$u_x(x,y,t) = u_0 - \frac{5}{2\pi}\exp\left(\frac{1-r^2}{2}\right)(y-u_0 t),$$
$$u_y(x,y,t) = v_0 + \frac{5}{2\pi}\exp\left(\frac{1-r^2}{2}\right)(x-v_0 t)),$$

and $p = \rho^\gamma$ where $r = \sqrt{(x-u_0 t)^2 + (y-v_0 t)^2}$. This problem is set on the $[-10;10]^2$ domain with periodic boundary conditions and $u_0 = v_0 = 1$ (the vortex is therefore advected along an oblique direction) and $\gamma = 1.4$. We have run computations until $t = 20$ with a CFL coefficient of 0.9 and reported the error between the analytical and numerical solutions in Table 4.10. Once again, the experimental order of convergence matches the theoretical one and it is not affected by the presence of hyperviscosity.

WITHOUT HYPERVISCOSITY						
Mesh size	**2nd-order**		**3rd-order**		**4th-order**	
(N × N cells)	error	order	error	order	error	order
N = 100	2.98e+01		3.16e+01		6.36e+00	
N = 200	5.60e+00	2.41	4.88e+00	2.69	3.92e-01	4.02
N = 400	1.08e+00	2.38	6.42e-01	2.93	2.42e-02	4.02
N = 800	2.34e-01	2.20	8.09e-02	2.99	1.50e-03	4.01
N = 1600	5.59e-02	2.06	1.01e-02	3.00	9.36e-05	4.00
WITH HYPERVISCOSITY ($C_\beta = C_\nu = C_\kappa = 1$)						
Mesh size	**2nd-order**		**3rd-order**		**4th-order**	
(N × N cells)	error	order	error	order	error	order
N = 100	4.23e+01		3.36e+01		8.09e+00	
N = 200	6.82e+00	2.63	4.94e+00	2.77	4.13e-01	4.29
N = 400	1.18e+00	2.53	6.43e-01	2.94	2.45e-02	4.08
N = 800	2.43e-01	2.28	8.09e-02	2.99	1.51e-03	4.02
N = 1600	5.68e-02	2.10	1.01e-02	3.00	9.37e-05	4.01

Table 4.10: Error between the analytical and numerical solutions for the isentropic advected vortex problem.

4.5.2 Non-smooth test problems

We now consider several non-smooth test problems in order to study the behaviour of GoHy schemes on such benchmaks and illustrate their robustness features.

Remark 19. In the case of non-smooth test problems, the numerical results provided with GoHy schemes are often compared in the sequel with a reference solution. This solution has actually been obtained by performing computations with a one-dimensional first-order accurate Lagrange-remap scheme [30] on 100,000 cells.

One-dimensional Sod shock tube

The Sod shock tube [76] is a well-known test problem whose initial condition is given by the following left and right states:

$$(\rho,\ p) = \begin{cases} (1,\ 1) & \text{if } 0 \leq x \leq 0.5, \\ (0.125,\ 0.1) & \text{if } 0.5 \leq x \leq 1 \end{cases},$$

the fluid being initially at rest. We have run computations with the fourth-order scheme until $t = 0.14$ on 400 cells, the CFL coefficient being set to 0.7. Figure 4.4 plots the density and pressure obtained with and without hyperviscosity. When hyperviscosity is enabled, the artifial coefficients are: $C_\beta = 2$, $C_\kappa = 5$ and

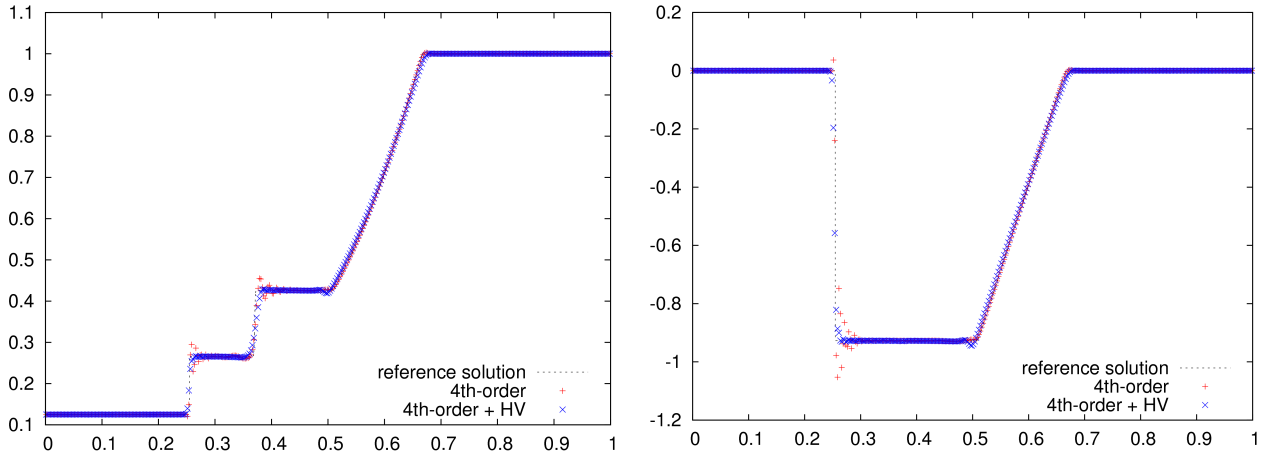


Figure 4.4: One-dimensional Sod shock tube at time $t = 0.14$. Density (left) and velocity (right) obtained on 400 cells with the fourth-order scheme. The reference solution has been obtained using a first-order accurate Lagrange-remap scheme on 100,000 cells (see remark 19 page 96).

$C_\nu = 0$. These results highlight the impact of hyperviscosity on oscillations: these have been noticeably reduced, leading to a satisfying approximation of the exact solution.

One-dimensional Shu-Osher problem

The next test problem has been proposed by Shu and Osher [75] and consists in a shock wave crossing a sinusoidal density field that generates turbulences. The initial condition is given by:

$$(\rho,\ u,\ p) = \begin{cases} (3.857143,\ 2.629369,\ 31/3) & \text{if } -5 \leq x \leq -4, \\ (1 + 0.2\sin(5x),\ 0,\ 1) & \text{if } -4 \leq x \leq 5. \end{cases}$$

The computational domain is $[-5; 5]$ and has open boundary conditions on both sides. We run this test problem until time $t = 1.8$ with a CFL coefficient of 0.7.

Figure 4.5 plots the density and pressure obtained with the fourth-order scheme at final time on 800 cells. Computations have been performed with and without hyperviscosity (when hyperviscosity is enabled, the artificial constants are $C_\beta = C_\kappa = 1$ and $C_\nu = 0$). They show that hyperviscosity provides important qualitative improvements by reducing Gibbs oscillations.

Figure 4.6 plots the density obtained on 400 cells with the first-order accurate acoustic solver (whose description can be found in [45]) and the second, third and fourth-order accurate GoHy schemes. For GoHy schemes, the hyperviscosity model has been turned on with $C_\beta = 1$ in all cases, the other artificial constants being set to zero. Figure 4.6 shows that the acoustic solver is not able to provide a satisfying numerical solution on 400 cells, especially in the turbulent zone behind the shock. GoHy schemes produce far better results. The second and third-order schemes provide similar numerical solutions in the turbulent zone but the second-order one is more subject to Gibbs oscillations in the left part of the computational domain. The fourth-order scheme clearly outperforms both of them in the turbulent zone though it generates a greater oscillation in the vicinity of $x = 0.6$.

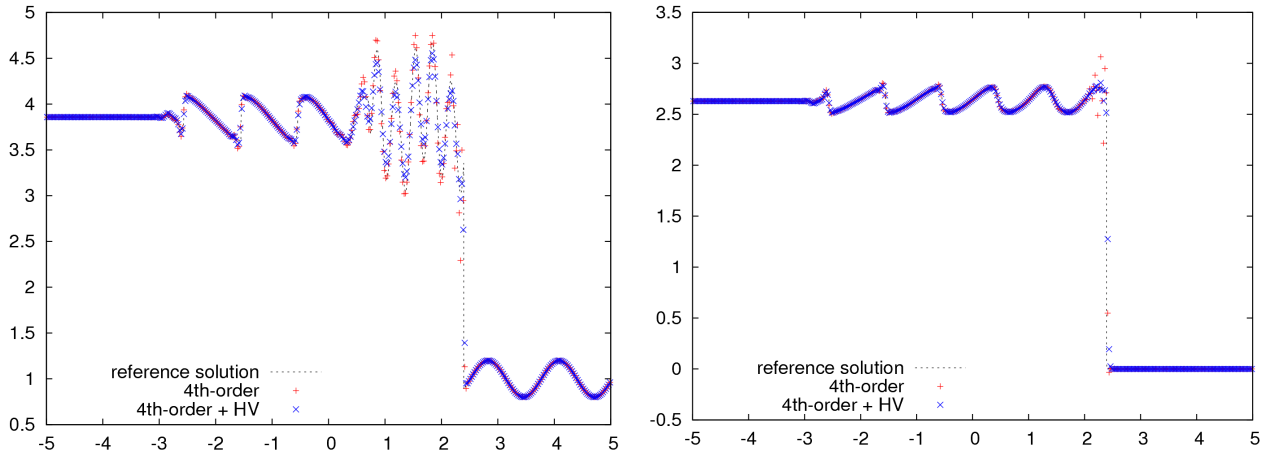


Figure 4.5: One-dimensional Shu-Osher problem at time $t = 1.8$. Density (top) and velocity (right) obtained on 800 cells with the fourth-order scheme. The reference solution has been obtained using a first-order accurate Lagrange-remap scheme on 100,000 cells (see remark 19 page 96).

One-dimensional Noh problem

The Noh problem [67] consists in a shock tube where a cold uniform perfect gas is driven by a piston with constant speed into a rigid wall (see Figure 4.7). Numerically, we set the following initial condition on the $[0; 0.4]$ domain:

$$\rho = 1, \ p = 10^{-6}, \ u = -1$$

with a reflective boundary on the left side and an open one on the right side. This problem is known to be particularly stringent since the initial pressure is very close to zero: any oscillation that may form would probably produce unphysical values. We have run computations until time $t = 0.6$ with the fourth-order scheme and a CFL coefficient of 0.4, the hyperviscosity model being enabled with following parameters: $C_\beta = 10$, $C_\kappa = 3$. Figure 4.8 plots the density and pressure obtained at final time on 200 cells. Like most numerical methods, the fourth-order GoHy scheme is subject to the wall heating phenomenon in the vicinity of the reflective boundary. Nevertheless, we noticed that artificial viscosity can reveal helpful for reducing wall heating. Indeed, taking large C_κ constants creates an overtemperature on the wall boundary and allows to catch more accurate solutions as shown in Figure 4.9. But for such C_κ values, we cannot ignore the stability condition (4.49): computations therefore last longer in this case. Practically, the calculation with $C_\kappa = 50$ lasts 50% longer than the one with $C_\kappa = 3$.

Note that the solution to the one-dimensional Noh problem is known analytically, it is for example given in [67]. At time $t = 0.6$, the exact density and pressure write:

$$\rho = \begin{cases} 4 & x < 0.2, \\ 1 & x > 0.2, \end{cases} \quad \text{and} \quad p = \begin{cases} 4/3 & x < 0.2, \\ 0 & x > 0.2, \end{cases}$$

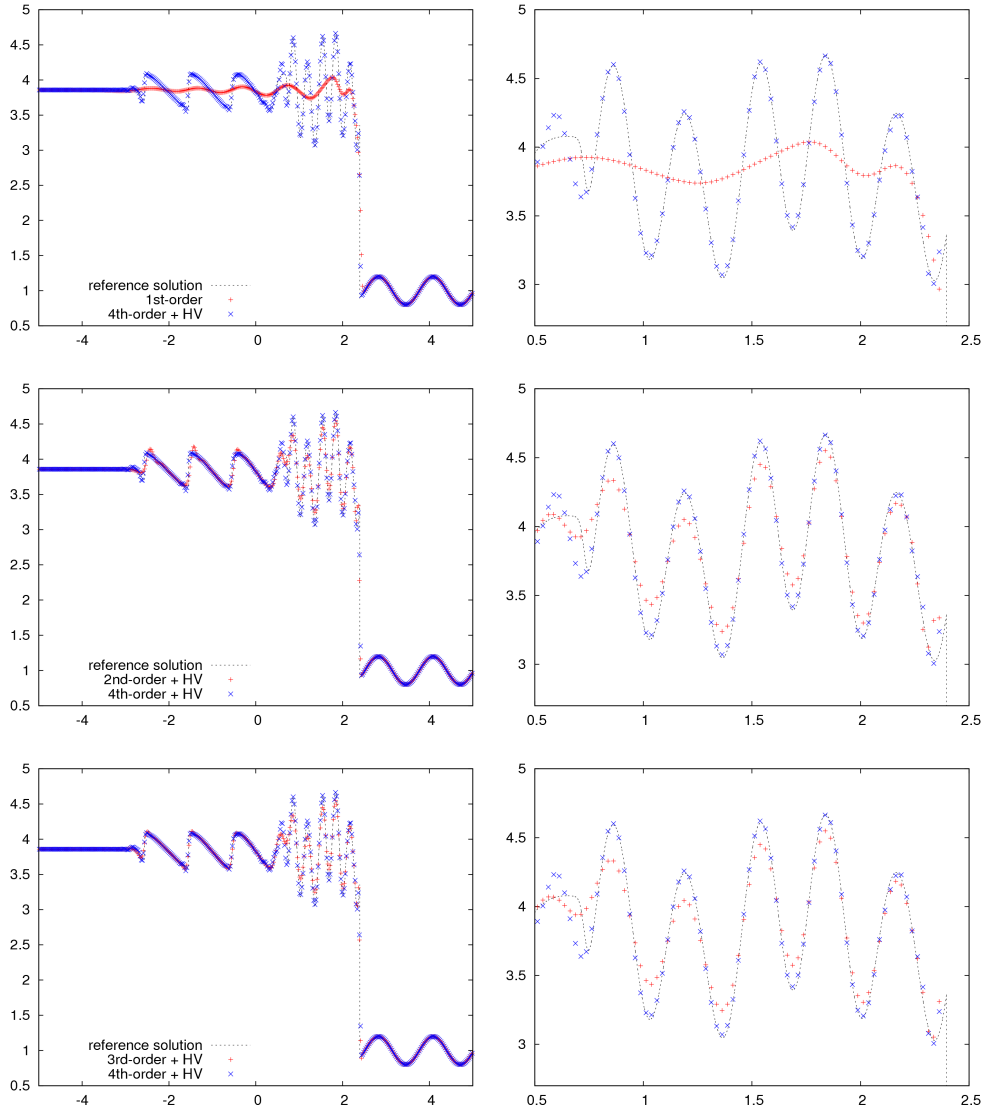


Figure 4.6: Comparison of the acoustic solver, second, third and fourth-order accurate GoHy schemes on Shu-Osher's problem: density at time $t = 1.8$ on 400 cells.

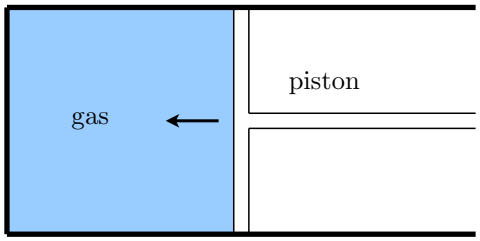


Figure 4.7: Physical setup of the Noh problem.

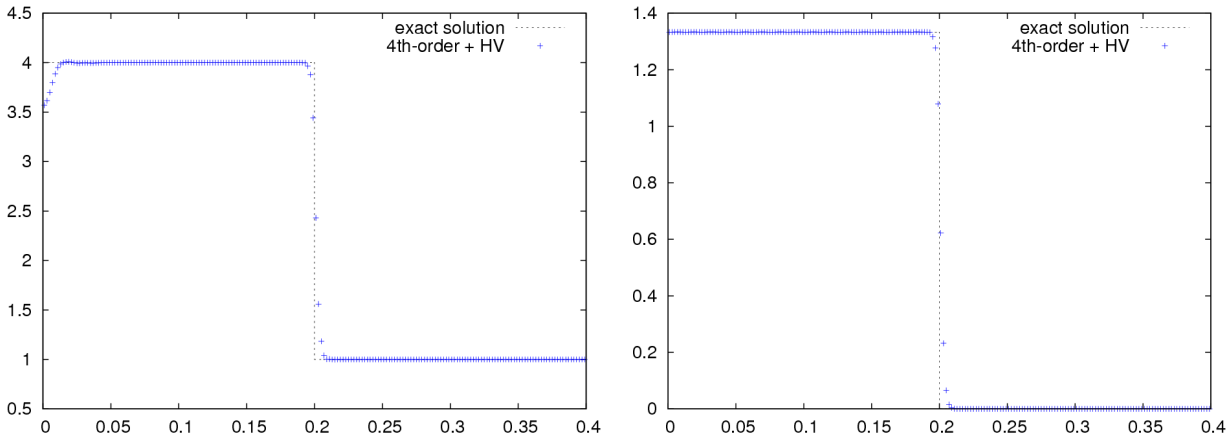


Figure 4.8: Density (left) and pressure (right) for the Noh problem at time $t = 0.6$ on 200 cells.

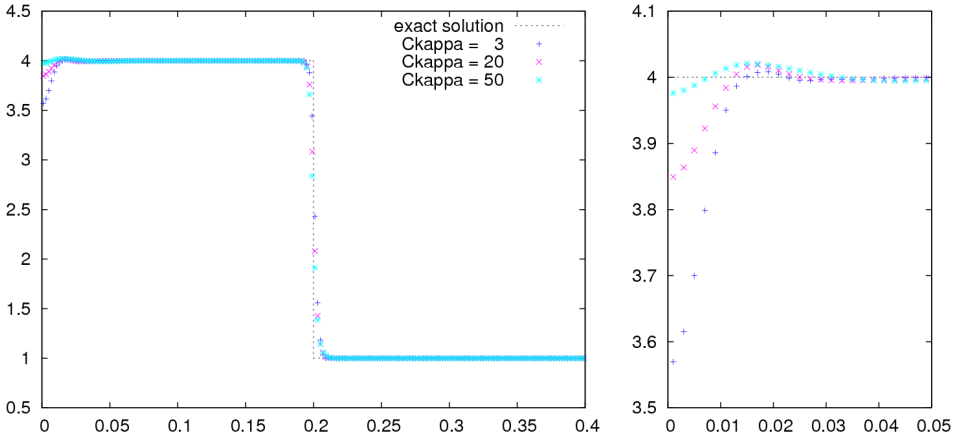


Figure 4.9: Density for the Noh problem at time $t = 0.6$ on 200 cells with different values of C_κ: whole computational domain (left) and zoom on the $x \in [0; 0.05]$ region (right).

Two-dimensional Sod shock tube

We now consider the Sod shock tube in a two-dimensional cylindrically symmetric geometry. The test problem is now set on the $[0;1]^2$ square and the initial condition is given by:

$$(\rho,\ p) = \begin{cases} (1,\ 1) & \text{if } 0 \leq r \leq 0.5, \\ (0.125,\ 0.1) & \text{if } 0.5 \leq r \leq 1 \end{cases},$$

The fluid is again initially at rest. The computational domain has reflective boundary conditions along the x and y axis and open boundary conditions elsewhere. We have run computations until a later time than the usual $t = 0.14$ limit. More precisely, this test problem has been run until $t = 0.5$ in order to let the shock focalize, be reflected as a divergent shock wave and crosses the contact discontinuity. This approach will make potential symmetry losses appear very clearly at final time. Figure 4.10 presents the results obtained for three computations on 1000×1000 cells with the third-order scheme (we restrict to third-order accuracy here because the fourth-order computation crashes without artificial viscosity) and three different configurations.

- The first results have been obtained with the base scheme, all artificial viscosity models being disabled. Note the presence of unsymmetric structures along the $x = 0$, $y = 0$ and $y = x$ axis. Some unphysical Gibbs oscillations also appear on discontinuities.

- The second set of results has been obtained by adding hyperviscosity to the previous configuration with the following parameters: $C_\beta = 0.5$, $C_\nu = 0.1$, $C_\kappa = 2$. Hyperviscosity noticeably reduces oscillations on discontinuities but does unfortunately not completely solve symmetry issues observed previously.

- The last set of results has been generated with both artificial viscosity models. More precisely, $\beta^\star$ and $\kappa^\star$ follow the hyperviscosity model with $C_\beta = 0.5$ and $C_\kappa = 2$ whereas the first-order artificial viscosity model has been used to determine $\nu^\star$ with $C_\nu = 0.1$. The two-dimensional density plot exhibits nice additional improvements in terms of symmetry compared to previous configurations.

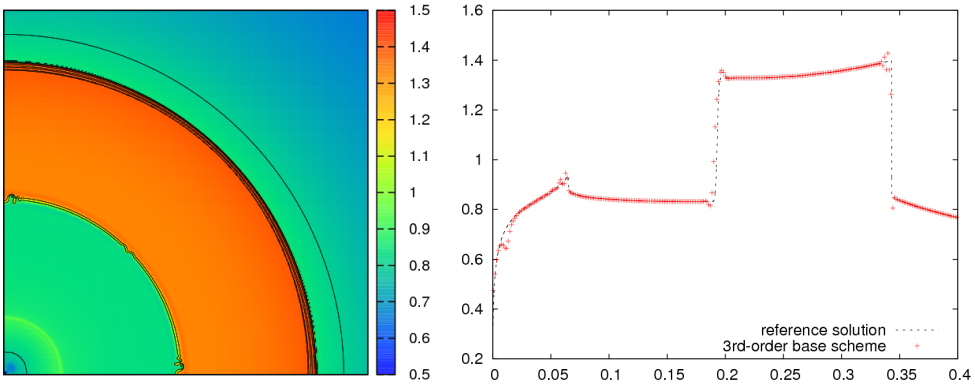
Two-dimensional Noh problem

We now consider the Noh problem in cylindrically symmetric geometry. Numerically, the problem is set up on the $[0;0.4]^2$ computational domain with reflective left and bottom boundary conditions and open boundaries elsewhere. We have run computations until time $t = 0.6$ with the fourth-order scheme, the CFL coefficient being set to 0.4 and the hyperviscosity model being enabled with following parameters: $C_\beta = 2$, $C_\nu = 2$, $C_\kappa = 20$. Figure 4.12 plots the density and pressure at final time. Contrarily to the two-dimensional Sod problem, we do not observe any symmetry loss here.

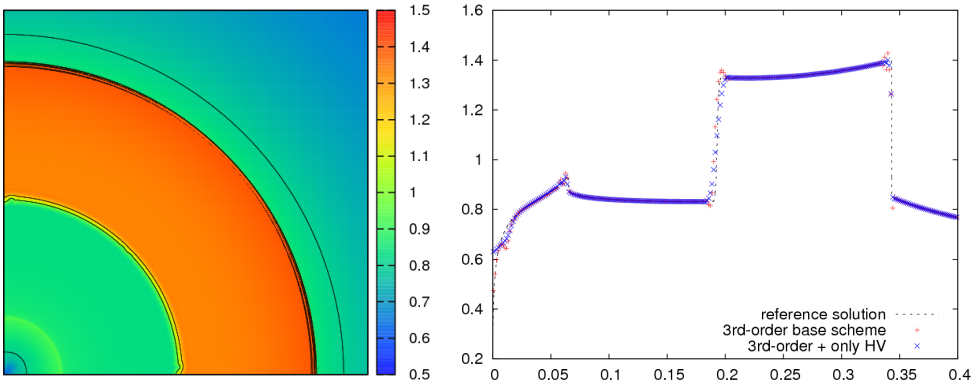
Once again, the density field is degraded by a wall heating phenomenon in the vicinity of reflective boundaries but, as in the one-dimensional case, taking high C_κ values can help to get a more accurate numerical solution. Figure 4.11 plots the density obtained at time $t = 0.6$ with the third-order GoHy scheme, the hyperviscosity model being enabled with $C_\beta = C_\nu = 5$. It shows that increasing C_κ allows to noticeably reduce the wall heating effect. But for high C_κ values, the stability condition (4.49) cannot be ignored and thus increases the restitution time. Practically, the simulation with $C_\kappa = 70$ lasts about 30% longer than the one with $C_\kappa = 25$.

Note that the solution to the two-dimensional Noh problem is known analytically, it is for example given

3rd-order scheme without artificial viscosity



3rd-order scheme + hyperviscosity



3rd-order scheme + both artificial viscosity models

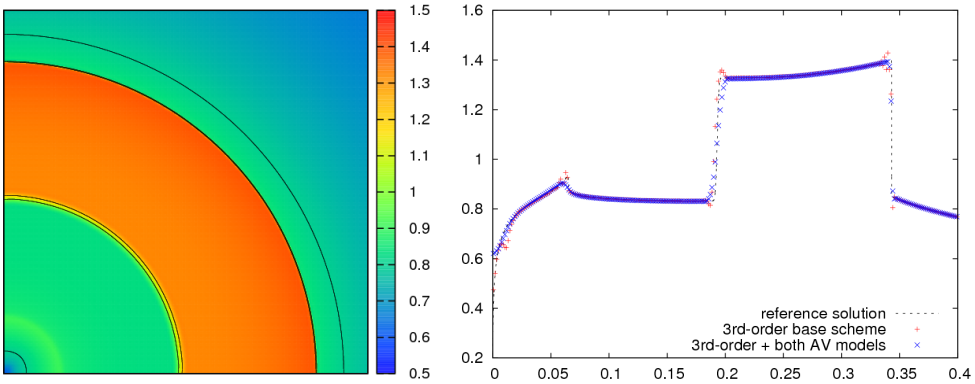


Figure 4.10: Two-dimensional Sod shock tube in cylindrically symmetric planar geometry at time $t = 0.5$ on 1000×1000 cells. Left: zoom on the $[0; 0.4]^2$ domain. Right: slice along the $y = x$ axis. The reference solution has been obtained using a one-dimensional first-order accurate Lagrange-remap scheme in cylindrical geometry on 100,000 cells (see remark 19 page 96).

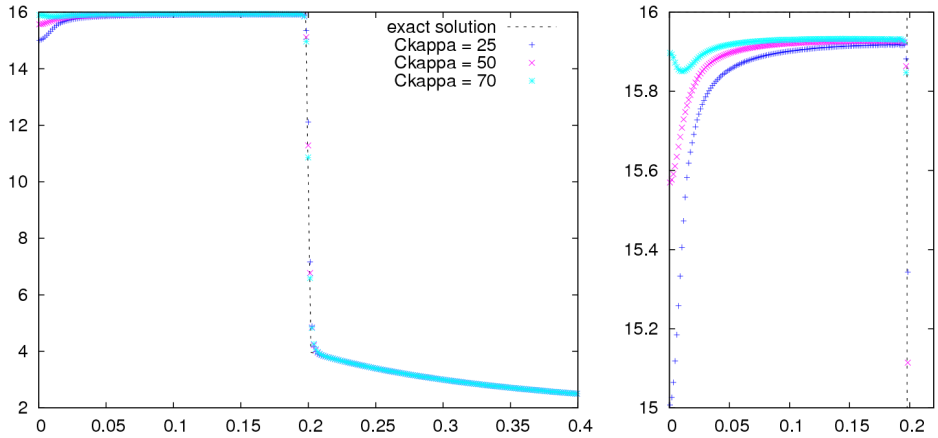


Figure 4.11: Density for the two-dimensional Noh problem obtained at time $t = 0.6$ on 400 cells with the third-order scheme and different values of C_κ: whole computational domain (left) and zoom on the $[0; 0.022]$ region (right).

in [67]. At time $t = 0.6$, the exact density and pressure write:

$$\rho = \begin{cases} 16 & r < 0.2, \\ 1 + \frac{0.6}{r} & r > 0.2, \end{cases} \quad \text{and} \quad p = \begin{cases} 16/3 & r < 0.2, \\ 0 & r > 0.2, \end{cases}$$

where r denotes the radial coordinate.

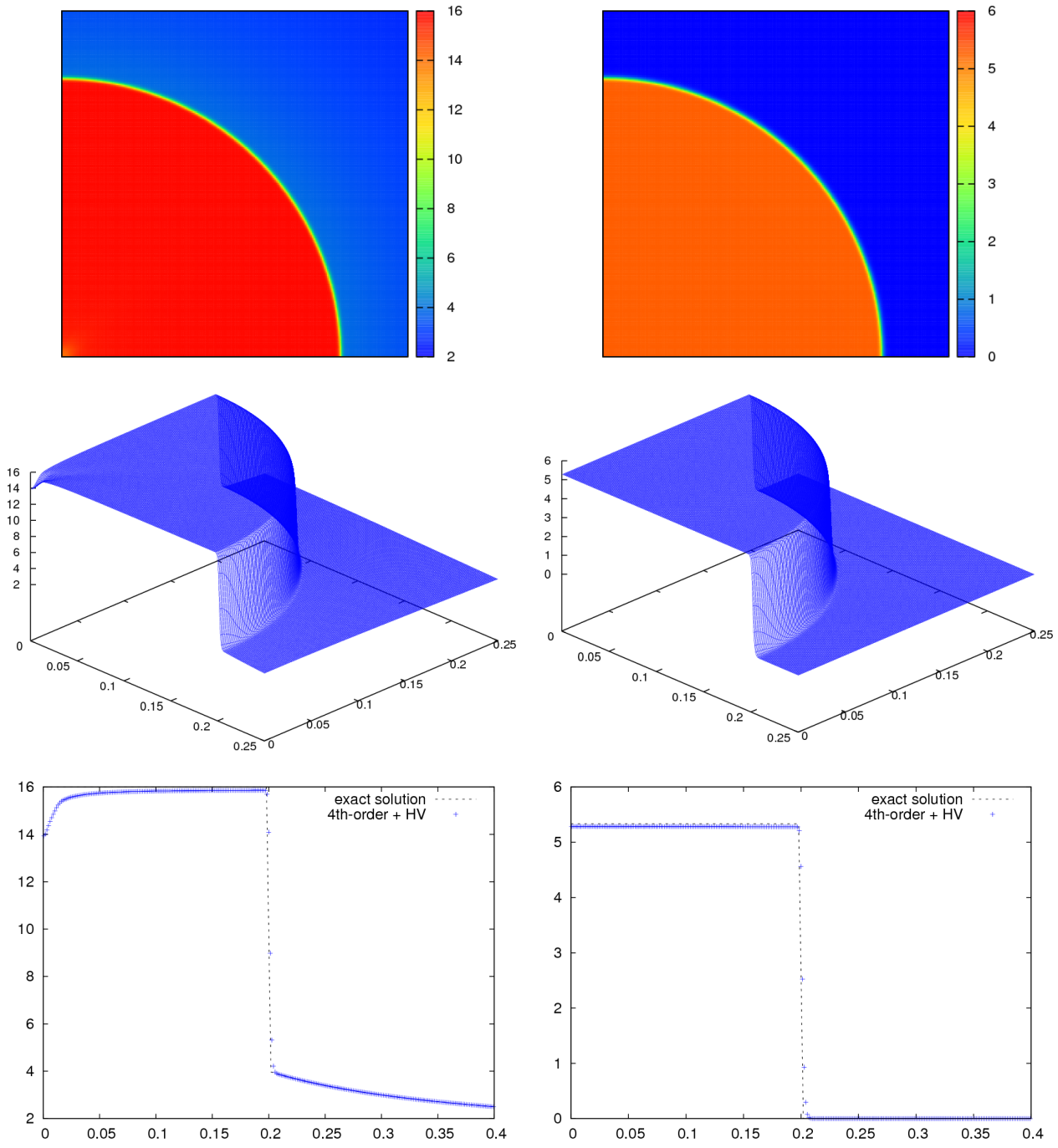


Figure 4.12: Two-dimensional Noh problem in cylindrically symmetric planar geometry: density (left) and pressure (right) obtained at time $t = 0.6$. From top to bottom: map plot, 2-D plot and slice along the $y = x$ axis.

Chapter 5

High-order dimensionally split Lagrange-remap schemes for ideal magnetohydrodynamics in planar geometry

In this chapter, we extend the high-order dimensionally split Lagrange-remap schemes presented previously in the case of hydrodynamics to the ideal magnetohydrodynamics system given by:

$$\partial_t \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ \mathbf{B} \\ \rho e \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\otimes\mathbf{u} + P\underline{\mathbf{I}} - \frac{(\mathbf{B}\otimes\mathbf{B})}{\mu} \\ \mathbf{u}\otimes\mathbf{B} - \mathbf{B}\otimes\mathbf{u} \\ (\rho e + P)\mathbf{u} - \frac{\mathbf{u}\cdot\mathbf{B}}{\mu}\mathbf{B} \end{pmatrix} = \mathbf{0}, \tag{5.1}$$

where $\mathbf{B}$ denotes the magnetic field and P the total pressure which is obtained by adding the gas pressure and the magnetic pressure:

$$P = p + \frac{\mathbf{B}\cdot\mathbf{B}}{2\mu}.$$

To that end, we first apply the strategy described in section 4.1 in order to build GoHy schemes for the one-dimensional ideal MHD system. We then combine these one-dimensional schemes with dimensional splitting techniques and a reconstruction step on fluxes as explained in section 4.2 in order to form high-order schemes for the two-dimensional ideal MHD system. We nevertheless have to consider an additional equation when dealing with multidimensional MHD simulations. Indeed, the magnetic field has to satisfy the following divergence constraint in this case:

$$\nabla \cdot \mathbf{B} = 0.$$

Multidimensional numerical solvers which do not guarantee a divergence-free evolution of the magnetic field may produce unphysical results due to non-zero $\nabla \cdot \mathbf{B}$ terms (see [10]) or even crash before the end of the computation. It is therefore crucial to enforce the divergence constraint numerically. Several techniques for preserving $\nabla \cdot \mathbf{B} = 0$ in practical simulations have been proposed in the literature in the last decades. A few of these methods are summarized below and described more precisely in this chapter.

- Brackbill and Barnes' projection scheme [10] consists in projecting the obtained magnetic field after an iteration of an arbitrary numerical scheme onto a divergence-free space.
- The so-called constrained transport methods which have been firstly introduced by Evans and Hawley [35] consist in building a numerical scheme that preserves at the accuracy of round-off errors a certain definition of the numerical divergence of the magnetic field.
- Powell [72] proposes an alternative non-conservative form of the ideal MHD system which has been derived without assuming that $\nabla \cdot \mathbf{B} = 0$. This approach - called the 8-wave formulation - aims at advecting divergence errors in order to prevent them from accumulating on a fixed grid point and producing unphysical results.
- Dedner *et al.* introduce in [29] the so-called hyperbolic divergence cleaning technique which can be seen as a genelization of Powell's idea. It consists in adding a new unknown to the ideal MHD system that satisfies an advection equation whose effect is to propagate divergence errors out of the computational domain. The improvement provided by the hyperbolic divergence cleaning technique upon the 8-wave formulation lies in the fact that the resulting system of equations is still in conservative form.

We propose a short study of the advantages and drawbacks of these techniques and focus in particular on their compatibility with dimensionally split approaches in order to determine which method could be coupled with GoHy schemes. It turns out that hyperbolic divergence cleaning techniques seem to be the best suited. We therefore describe how they can be combined with GoHy schemes and in particular how high-order accuracy can be achieved, which is an innovative feature of this work.

The outline of this chapter is the following. We first describe the construction of the one-dimensional GoHy Lagrange-remap scheme and its extension to the two-dimensional case. We then focus on how to enforce a divergence-free evolution of the magnetic field in the two-dimensional case without losing high-order accuracy. Finally, we provide some performance and parallel efficiency figures and exhibit numerical results on both smooth and non-smooth test problems taken from the related literature in order to illustrate the convergence and robustness features of these MHD-GoHy schemes.

5.1 One-dimensional Lagrange-remap schemes for ideal MHD

We first focus on extending GoHy schemes described in the previous chapter to the one-dimensional ideal MHD system which writes:

$$\partial_t \begin{pmatrix} \rho \\ \rho u_x \\ \rho u_y \\ \rho u_z \\ B_x \\ B_y \\ B_z \\ \rho e \end{pmatrix} + \partial_x \begin{pmatrix} \rho u_x \\ \rho u_x^2 + P - \frac{B_x^2}{\mu} \\ \rho u_x u_y - \frac{B_x B_y}{\mu} \\ \rho u_x u_z - \frac{B_x B_z}{\mu} \\ 0 \\ u_x B_y - u_y B_x \\ u_x B_z - u_z B_x \\ (\rho e + P) u_x - \frac{\mathbf{u}\cdot\mathbf{B}}{\mu} B_x \end{pmatrix} = \mathbf{0} \tag{5.2}$$

where P is the total pressure:

$$P = p + \frac{\mathbf{B}\cdot\mathbf{B}}{2\mu}.$$

To that end, we first rewrite system (5.2) in Lagrangian coordinates and present how to build high-order accurate Lagrangian and remap fluxes.

Remark 20. In the one-dimensional case, since $\partial_t B_x = 0$, the complete finite volume scheme for B_x simply writes:

$$\overline{(B_x)}_i^{n+1} = \overline{(B_x)}_i^{n} \quad \text{where} \quad \overline{(B_x)}_i^{n} = \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} B_x(x,t)\,dx.$$

Note that this approach is only valid in the one-dimensional case: it should not be applied in the multidimensional case as explained in the sequel (see remark 23).

5.1.1 Lagrangian step

We first derive the ideal MHD system in Lagrangian coordinates using the variable change presented in section 4.1.1. Note that ρ satisfies the same advection equation than in the hydrodynamical case. The Jacobian of the $(x,t) \to (X,t)$ variable change thus still writes $J = \rho_0/\rho$ where ρ_0 denotes the initial density. One can then apply Lemma 1 (see page 60) to the eight following (ϕ,ψ) couples:

$$(1,\ -u_x),\ (\rho u_x,\ P - B_x^2/\mu),\ (\rho u_y,\ -B_xB_y/\mu),\ (\rho u_z,\ -B_xB_z/\mu),\ (\mathbf{B},\ -B_x\mathbf{u}),\ (\rho e,\ Pu_x - B_x(\mathbf{u}\cdot\mathbf{B})/\mu),$$

which leads to the one-dimensional ideal MHD system of equations in Lagrangian coordinates:

$$\partial_t \mathbf{U} + \partial_X \mathbf{F}(\mathbf{U}) = \mathbf{0} \quad \text{with} \quad \mathbf{U} = \begin{pmatrix} \rho_0\tau \\ \rho_0 u_x \\ \rho_0 u_y \\ \rho_0 u_z \\ \rho_0\tau B_x \\ \rho_0\tau B_y \\ \rho_0\tau B_z \\ \rho_0 e \end{pmatrix} \quad \text{and} \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} -u_x \\ P - \frac{B_x^2}{\mu} \\ -\frac{B_xB_y}{\mu} \\ -\frac{B_xB_z}{\mu} \\ -u_xB_x \\ -u_yB_x \\ -u_zB_x \\ Pu_x - \frac{\mathbf{u}\cdot\mathbf{B}}{\mu}B_x \end{pmatrix}. \tag{5.3}$$

At this stage, we build a finite volume scheme for the Lagrangian ideal MHD equations by applying the strategy described in section 4.1.2 to system (5.3). More precisely, we integrate it over $[X_{i-\frac{1}{2}}; X_{i+\frac{1}{2}}] \times [t^n; t^{n+1}]$ and thus get the following numerical scheme:

$$\frac{\overline{\mathbf{U}}_i^{n+1} - \overline{\mathbf{U}}_i^{n}}{\Delta t} + \frac{\mathbf{F}^*_{i+\frac{1}{2}} - \mathbf{F}^*_{i-\frac{1}{2}}}{\Delta X} = 0,$$

where $\overline{\mathbf{U}}_i^n$ is the cell average of $\mathbf{U}$ over $[X_{i-\frac{1}{2}}; X_{i+\frac{1}{2}}]$ at time t^n and and $\mathbf{F}^*_{i+\frac{1}{2}}$ the Lagrangian flux at the $X_{i+\frac{1}{2}}$ boundary defined by (4.13). Flux terms are then replaced by truncated Taylor expansions and the resulting time-derivatives are replaced by space-derivatives using the Cauchy-Kovalevskaya procedure. Expressions for the time-derivatives of τ, $\mathbf{u}$ and $\mathbf{B}$ can be easily derived from system (5.3) but once again, one has to pay attention to the pressure term. The following proposition gives an expression of the first time-derivative of p that is suitable for any equation of state.

Proposition 8. *For any equation of state, the pressure p satisfies the following evolution equation:*

$$\partial_t p = -\frac{(\rho c)^2}{\rho_0}\partial_X u_x - \frac{\Gamma(\mathbf{u}\cdot\mathbf{B})}{\rho_0\tau\mu}\partial_X B_x, \tag{5.4}$$

where $(\rho c)^2$ denotes the Lagrangian sound speed and Γ the Grüneisen coefficient (see their definitions page 64).

Proof. The proof is similar to the one given in section 4.1.2 for the hydrodynamic set of equations. It again relies on the second law of thermodynamics which writes $TdS = d\epsilon + pd\tau$. In the case of ideal MHD, the internal energy is given by:

$$\epsilon = e - \frac{\mathbf{u}\cdot\mathbf{u}}{2} - \frac{\mathbf{B}\cdot\mathbf{B}}{2\rho\mu}.$$

By combining this definition with the second law of thermodynamics and the time-derivatives' expressions that can be derived from system (5.3), one can see that:

$$\begin{aligned}\rho_0 T\partial_t S &= \partial_t(\rho_0\epsilon) + p\partial_t(\rho_0\tau)\\ &= \partial_t(\rho e) - \partial_t(\rho_0\mathbf{u})\cdot\mathbf{u} - \frac{1}{\mu}\partial_t(\rho_0\tau\mathbf{B})\cdot\mathbf{B} + p\partial_t(\rho_0\tau)\\ &= -\frac{\mathbf{u}\cdot\mathbf{B}}{\mu}\partial_X B_x.\end{aligned}$$

Since the equation of state satisfies $dp = -(\rho c)^2 d\tau + (\Gamma T/\tau)dS$, the first time-derivative of p then writes:

$$\partial_t p = -(\rho c)^2\partial_t\tau + \frac{\Gamma T}{\tau}\partial_t S = -\frac{(\rho c)^2}{\rho_0}\partial_X u_x - \frac{\Gamma(\mathbf{u}\cdot\mathbf{B})}{\rho_0\tau\mu}\partial_X B_x.$$

□

Remark 21. In the case of ideal gases, the Grüneisen coefficient is a constant and its value is $\Gamma = \gamma - 1$.

Nevertheless, the result given by Proposition 8 can lead to different approximate Lagrangian fluxes depending on the considered context. It can indeed be simplified in several specific cases in order to reduce the cost of Lagrangian fluxes computations.

- In the one-dimensional case, the divergence constraint implies that $\partial_X B_x = 0$. The expression of the pressure's time-derivatives given in section 4.1.2 for hydrodynamics thus holds in the case of one-dimensional ideal MHD. As a matter of fact, the expression of the Lagrangian sound speed's time-derivative which had been derived in section 4.1.2 holds too. These write:

$$\partial_t p = -\frac{(\rho c)^2}{\rho_0}\partial_X u_x, \quad \text{and} \quad \partial_t(\rho c)^2 = -\frac{2(\rho c)^2\mathcal{G}}{\tau\rho_0}\partial_X u_x. \tag{5.5}$$

  We recall that these expressions are valid for any equation of state and allow to build Lagrangian fluxes up to third-order accuracy. From fourth-order accuracy, we will assume for the sake of simplicity that $\partial_t\mathcal{G} = 0$, which is true for ideal gases (see remark 7 page 66).

- In the multidimensional orthogonal MHD framework, the only possibly non-zero component of the magnetic field is the one that is perpendicular to the computational domain, *i.e.* B_z here. B_x and B_y are thus identically zero in this context and the spatial derivative of the magnetic field included in (5.4) can be removed for both dimensional sweeps along the x and y directions. The above-given time-derivatives' expressions can therefore be used.

- In the case of multidimensional MHD computations, the divergence constraint does not imply that $\partial_X B_x = 0$ and the expression of the pressure's time-derivative given by (5.4) thus can not be simplified. It allows to build Lagrangian fluxes that are valid for any equation of state up to second-order accuracy. From third-order accuracy, we restrict for the sake of simplicity to ideal gases. In this case, the Lagrangian sound speed's time-derivative can be obtained by differentiating its definition:

$$\begin{aligned}\partial_t(\rho c)^2 &= \gamma\partial_t\left(\frac{p}{\tau}\right) \text{ since } \gamma \text{ is a constant for ideal gases}\\ &= \frac{\gamma}{\tau}\partial_t p - \frac{(\rho c)^2}{\tau}\partial_t\tau\\ &= -\frac{(\gamma+1)(\rho c)^2}{\rho_0\tau}\partial_X u_x - \frac{\gamma\Gamma(\mathbf{u}\cdot\mathbf{B})}{\rho_0\tau^2\mu}\partial_X B_x\end{aligned}$$

  whereas the time-derivative of $\mathcal{G}$ is zero (see remark 7 page 66).

Once the Cauchy-Kovalevskaya procedure has been applied, cell-centered point-wise values of conservative variables have to be computed by applying formula (4.15) with $\phi \in \{1, \mathbf{u}, \tau\mathbf{B}, e\}$. One can then determine estimations of quantities involved in the truncated Taylor expansion using finite difference formulae (4.16) and form the high-order accurate Lagrangian fluxes. The Lagrangian scheme finally writes:

$$\begin{pmatrix} \overline{(\rho_0 u_x)}_i^{n+1} \\ \overline{(\rho_0 u_y)}_i^{n+1} \\ \overline{(\rho_0 u_z)}_i^{n+1} \\ \overline{(\rho_0\tau B_y)}_i^{n+1} \\ \overline{(\rho_0\tau B_z)}_i^{n+1} \\ \overline{(\rho_0 e)}_i^{n+1} \end{pmatrix} = \begin{pmatrix} \overline{(\rho_0 u_x)}_i^{n} \\ \overline{(\rho_0 u_y)}_i^{n} \\ \overline{(\rho_0 u_z)}_i^{n} \\ \overline{(\rho_0\tau B_y)}_i^{n} \\ \overline{(\rho_0\tau B_z)}_i^{n} \\ \overline{(\rho_0 e)}_i^{n} \end{pmatrix} - \frac{\Delta t}{\Delta x}\left[\begin{pmatrix} \left(P - \frac{B_x^2}{\mu}\right)^*_{i+\frac{1}{2}} \\ -\frac{1}{\mu}\left(B_x B_y\right)^*_{i+\frac{1}{2}} \\ -\frac{1}{\mu}\left(B_x B_z\right)^*_{i+\frac{1}{2}} \\ -\left(u_y B_x\right)^*_{i+\frac{1}{2}} \\ -\left(u_z B_x\right)^*_{i+\frac{1}{2}} \\ \left(P u_x - \frac{B_x(\mathbf{u}\cdot\mathbf{B})}{\mu}\right)^*_{i+\frac{1}{2}} \end{pmatrix} - \begin{pmatrix} \left(P - \frac{B_x^2}{\mu}\right)^*_{i-\frac{1}{2}} \\ -\frac{1}{\mu}\left(B_x B_y\right)^*_{i-\frac{1}{2}} \\ -\frac{1}{\mu}\left(B_x B_z\right)^*_{i-\frac{1}{2}} \\ -\left(u_y B_x\right)^*_{i-\frac{1}{2}} \\ -\left(u_z B_x\right)^*_{i-\frac{1}{2}} \\ \left(P u_x - \frac{B_x(\mathbf{u}\cdot\mathbf{B})}{\mu}\right)^*_{i-\frac{1}{2}} \end{pmatrix}\right]. \tag{5.6}$$

We recall that the equation on $(\rho_0\tau)$ does not need to be solved practically (see remark 9 page 71). Hereagain, Lagrangian fluxes are automatically generated with the algebraic calculation software Maple.

5.1.2 Remap step

High-order accurate remap fluxes for the ideal MHD system can be built by applying the method presented in section 4.1.3 for the hydrodynamics equations. The remap scheme then writes:

$$\overline{(\rho\phi)}_i^{n+1} = \overline{(\rho_0\phi)}_i^{n+1} - \frac{\Delta t}{\Delta x}\left(u^*_{i+\frac{1}{2}}(\rho\phi)^*_{i+\frac{1}{2}} - u^*_{i-\frac{1}{2}}(\rho\phi)^*_{i-\frac{1}{2}}\right)$$

with $\phi \in \{1, \mathbf{u}, \tau B_y, \tau B_z, e\}$.

5.1.3 Stability condition

In the case of ideal MHD, the stability condition for the whole Lagrange-remap scheme writes:

$$\Delta t \leq \frac{\Delta x}{\max\left(|u_i| + (c_f)_i\right)}. \tag{5.7}$$

where c_f denotes the fast magnetosonic wave speed:

$$c_f = \sqrt{\frac{1}{2}\left(\frac{\mathbf{B}\cdot\mathbf{B}}{\rho} + c^2\right) + \sqrt{\frac{1}{4}\left(\frac{\mathbf{B}\cdot\mathbf{B}}{\rho} + c^2\right)^2 - \frac{c^2}{\rho^2}\min\left(B_x^2, B_y^2\right)}}$$

This completes the description of the one-dimensional Lagrange-remap scheme. We now detail its extension to the two-dimensional case.

5.2 Two-dimensional extension: the divergence constraint

Now that the GoHy high-order Lagrange-remap schemes have been described in the one-dimensional case, we present their extension to two-dimensional geometries. Basically, it relies on the same recipes than in the hydrodynamical case: combining dimensional splitting techniques and a reconstruction step on fluxes, one can easily build high-order accurate numerical schemes for the ideal MHD equations (see sections 4.2.1 and 4.2.2 for more details). Nevertheless, some additional issues arise when dealing with multidimensional ideal MHD equations. It is indeed well-known that numerical schemes which do not provide a divergence-free evolution of the magnetic field may lead to unphysical solutions [10]. We thus present in the sequel a few well-known methods for enforcing numerically $\nabla\cdot\mathbf{B} = 0$ and adapt one of these to the Lagrange-remap schemes presented previously.

5.2.1 Projection scheme

The projection scheme has been proposed by Brackbill and Barnes [10] as a correction to the magnetic field after the time step has been completed by some arbitrary numerical scheme. It consists in projecting the $\mathbf{B}^\star$ field provided by the base scheme onto a divergence-free space and relies on the following decomposition of $\mathbf{B}^\star$ into the sum of a curl and a gradient (Helmholtz decomposition) :

$$\mathbf{B}^\star = \nabla\times\mathbf{A} + \nabla\phi. \tag{5.8}$$

where the curl of the vector potential $\mathbf{A}$ contains the physically meaningfull part of $\mathbf{B}^\star$. Taking the divergence of equation (5.8) leads to the Poisson equation:

$$\nabla\cdot\mathbf{B}^\star = \Delta\phi \tag{5.9}$$

and it is then easy to form a divergence-free $\mathbf{B}$ field:

$$\mathbf{B} = \mathbf{B}^\star - \nabla\phi. \tag{5.10}$$

All in all, the projection scheme consists in solving the Poisson equation (5.9) in order to compute ϕ and the corrected magnetic field $\mathbf{B}$. Noce that the numerical divergence of $\mathbf{B}$ will be exactly zero only if the Laplacian operator in (5.9) is evaluated in two steps as the divergence of a gradient with the same discrete gradient and divergence operators used for calculating $\nabla\cdot\mathbf{B}^\star$ and $\nabla\phi$. Let N denote the desired order of accuracy, the following high-order accurate operators can be used when dealing with finite volume schemes on two-dimensional Cartesian meshes:

$$(\nabla f)_{i,j} = \begin{pmatrix} (D_x f)_{i,j} \\ (D_y f)_{i,j} \end{pmatrix} \quad \text{and} \quad (\nabla\cdot\mathbf{f})_{i,j} = (D_x f_x)_{i,j} + (D_y f_y)_{i,j},$$

where D_x and D_y are the following discrete operators:

$$(D_x\psi)_{i,j} = \begin{cases} \dfrac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta x} & \text{if } N \leq 2, \\ \dfrac{8(\psi_{i+1,j} - \psi_{i-1,j}) - (\psi_{i+2,j} - \psi_{i-2,j})}{12\Delta x} & \text{if } 3 \leq N \leq 4, \end{cases}$$

$$(D_y\psi)_{i,j} = \begin{cases} \dfrac{\psi_{i,j+1} - \psi_{i,j-1}}{2\Delta y} & \text{if } N \leq 2, \\ \dfrac{8(\psi_{i,j+1} - \psi_{i,j-1}) - (\psi_{i,j+2} - \psi_{i,j-2})}{12\Delta y} & \text{if } 3 \leq N \leq 4. \end{cases}$$

These lead to the following second-order accurate discrete Laplacian operator:

$$(\Delta\psi)_{i,j} = \frac{2\psi_{i,j} - (\psi_{i+2,j} + \psi_{i-2,j})}{4\Delta x^2} + \frac{2\psi_{i,j} - (\psi_{i,j+2} + \psi_{i,j-2})}{4\Delta y^2}$$

and to the following fourth-order accurate one:

$$(\Delta\psi)_{i,j} = \sum_{k=0}^{4} \omega_k \left(\frac{\psi_{i+k,j} + \psi_{i-k,j}}{\Delta x^2} + \frac{\psi_{i,j+k} + \psi_{i,j-k}}{\Delta y^2} \right)$$

with:

$$\omega_0 = \frac{65}{144},\ \omega_1 = -\omega_3 = -\frac{1}{9},\ \omega_2 = -\frac{4}{9},\ \omega_4 = -\frac{1}{144}.$$

Note that the required stencil for building a fourth-order accurate estimation of $\Delta\psi$ is very wide (it is indeed made of 17 cells), which presents two majors drawbacks. On the one hand, it makes the projection scheme expensive for achieving high-order accuracy since the linear system resulting from the discretization of the Poisson equation (5.9) will have lots of extra-diagonal terms. On the other hand, this approach generates many unphysical oscillations on discontinuities and thus impacts negatively the global scheme robustness. We therefore do not resort to the projection scheme for divergence cleaning purposes.

5.2.2 Constrained transport

We now describe the so-called constrained transport (CT) method which basically relies on a staggered representation of the magnetic field on cell interfaces denoted by $\mathbf{b}$ hereafter. Note that we consider the following form of the magnetic induction equation in this section:

$$\partial_t \mathbf{B} + \nabla \times \mathbf{\Omega} = 0 \quad \text{with} \quad \mathbf{\Omega} = -\mathbf{u} \wedge \mathbf{B}.$$

Original CT method

The initial CT method has been introduced by Evans and Hawley [35] in a finite differences framework. It consists in insuring $\nabla \cdot \mathbf{b} = 0$ by updating $\mathbf{b}$ using a cell corner representation of Ω:

$$\begin{aligned} (b_x)^{n+1}_{i+\frac{1}{2},j} &= (b_x)^{n}_{i+\frac{1}{2},j} - \frac{\Delta t}{\Delta y}\left(\Omega_{i+\frac{1}{2},j+\frac{1}{2}} - \Omega_{i+\frac{1}{2},j-\frac{1}{2}}\right), \\ (b_y)^{n+1}_{i,j+\frac{1}{2}} &= (b_y)^{n}_{i,j+\frac{1}{2}} + \frac{\Delta t}{\Delta x}\left(\Omega_{i+\frac{1}{2},j+\frac{1}{2}} - \Omega_{i-\frac{1}{2},j+\frac{1}{2}}\right). \end{aligned} \tag{5.11}$$

which allows to state the following proposition.

Proposition 9. *Let the discrete divergence of* $\mathbf{b}$ *be defined by:*

$$(\nabla \cdot \mathbf{b})_{i,j} = \frac{(b_x)_{i+\frac{1}{2},j} - (b_x)_{i-\frac{1}{2},j}}{\Delta x} + \frac{(b_y)_{i,j+\frac{1}{2}} - (b_y)_{i,j-\frac{1}{2}}}{\Delta y}. \tag{5.12}$$

The finite difference scheme (5.11) *proposed by Evans and Hawley [35] implies that*

$$(\nabla \cdot \mathbf{b})_{i,j}^{n+1} = (\nabla \cdot \mathbf{b})_{i,j}^{n}$$

to the accuracy of round-off errors.

Proof. Indeed:

$$\begin{aligned}(\nabla \cdot \mathbf{b})_{i,j}^{n+1} - (\nabla \cdot \mathbf{b})_{i,j}^{n} = {} & \frac{\Delta t}{\Delta x \Delta y}\left(\left(\Omega_{i-\frac{1}{2},j+\frac{1}{2}} - \Omega_{i-\frac{1}{2},j-\frac{1}{2}}\right) - \left(\Omega_{i+\frac{1}{2},j+\frac{1}{2}} - \Omega_{i+\frac{1}{2},j-\frac{1}{2}}\right)\right) \\ & + \frac{\Delta t}{\Delta x \Delta y}\left(\left(\Omega_{i+\frac{1}{2},j+\frac{1}{2}} - \Omega_{i-\frac{1}{2},j+\frac{1}{2}}\right) - \left(\Omega_{i+\frac{1}{2},j-\frac{1}{2}} - \Omega_{i-\frac{1}{2},j-\frac{1}{2}}\right)\right) \\ = {} & 0.\end{aligned}$$

□

Extension of the original CT method to the finite volume framework

Several successful attempts to adapt the CT approach to finite volume schemes have been presented in the literature (see the review of CT methods proposed by Tòth in [88] and references therein). We detail one of these proposed by Dai and Woodward [26] called the *field-interpolated* CT scheme. Let $\mathbf{u}^\star$ and $\mathbf{B}^\star$ denote the updated values of the velocity and magnetic field returned by an arbitrary Godunov-type base scheme. Dai and Woodward's approach relies on the following cell corner centered representations of $\mathbf{u}$ and $\mathbf{B}$ given by:

$$\begin{aligned}\mathbf{B}_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} &= \frac{1}{8}\left(\mathbf{B}_{i,j}^{n} + \mathbf{B}_{i+1,j}^{n} + \mathbf{B}_{i,j+1}^{n} + \mathbf{B}_{i+1,j+1}^{n} + \mathbf{B}_{i,j}^{\star} + \mathbf{B}_{i+1,j}^{\star} + \mathbf{B}_{i,j+1}^{\star} + \mathbf{B}_{i+1,j+1}^{\star}\right), \\ \mathbf{u}_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} &= \frac{1}{8}\left(\mathbf{u}_{i,j}^{n} + \mathbf{u}_{i+1,j}^{n} + \mathbf{u}_{i,j+1}^{n} + \mathbf{u}_{i+1,j+1}^{n} + \mathbf{u}_{i,j}^{\star} + \mathbf{u}_{i+1,j}^{\star} + \mathbf{u}_{i,j+1}^{\star} + \mathbf{u}_{i+1,j+1}^{\star}\right).\end{aligned}$$

These values are then used to form cell corner centered representations of Ω:

$$\mathbf{\Omega}_{i+\frac{1}{2},j+\frac{1}{2}} = -\mathbf{u}_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} \wedge \mathbf{B}_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}},$$

and allow to apply the finite differences scheme given in (5.11). The finite volume representation of the magnetic field is finally updated by interpolating $\mathbf{b}$:

$$\begin{aligned}(B_x)_{i,j}^{n+1} &= \frac{1}{2}\left((b_x)_{i-\frac{1}{2},j}^{n+1} + (b_x)_{i+\frac{1}{2},j}^{n+1}\right), \\ (B_y)_{i,j}^{n+1} &= \frac{1}{2}\left((b_y)_{i,j-\frac{1}{2}}^{n+1} + (b_y)_{i,j+\frac{1}{2}}^{n+1}\right).\end{aligned}$$

One can easily notice that the field-interpolated CT scheme does not preserve the cell centered numerical divergence of $\mathbf{B}$ given by:

$$(\nabla \cdot \mathbf{B})_{i,j} = \frac{(B_x)_{i+1,j} - (B_x)_{i-1,j}}{2\Delta x} + \frac{(B_y)_{i,j+1} - (B_y)_{i,j-1}}{2\Delta y} \tag{5.13}$$

but it turns out that another representation of $\nabla \cdot \mathbf{B}$ is conserved numerically.

Proposition 10. *Let the discrete divergence of* $\mathbf{B}$ *be defined by the following cell corner centered representation:*

$$(\nabla\cdot\mathbf{B})_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{(B_x)_{i+1,j}+(B_x)_{i+1,j+1}-(B_x)_{i,j}-(B_x)_{i,j+1}}{2\Delta x} + \frac{(B_y)_{i,j+1}+(B_y)_{i+1,j+1}-(B_y)_{i,j}-(B_y)_{i+1,j}}{2\Delta y}.$$

The field-interpolated CT scheme proposed by Dai and Woodward [26] implies that:

$$(\nabla\cdot\mathbf{B})^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} = (\nabla\cdot\mathbf{B})^{n}_{i+\frac{1}{2},j+\frac{1}{2}}$$

to the accuracy of round-off errors.

Proof. Indeed, from the definition of the magnetic field at time t^{n+1}, one can see that:

$$(\nabla\cdot\mathbf{B})^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{4}\left((\nabla\cdot\mathbf{b})^{n+1}_{i,j}+(\nabla\cdot\mathbf{b})^{n+1}_{i+1,j}+(\nabla\cdot\mathbf{b})^{n+1}_{i,j+1}+(\nabla\cdot\mathbf{b})^{n+1}_{i+1,j+1}\right)$$

where $(\nabla\cdot\mathbf{b})_{i,j}$ is given by (5.12). Since $(\nabla\cdot\mathbf{b})$ is preserved in each cell by the underlying constrained transport method (5.11), one necessarily has:

$$(\nabla\cdot\mathbf{B})^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} = (\nabla\cdot\mathbf{B})^{n}_{i+\frac{1}{2},j+\frac{1}{2}}\,.$$

□

Tòth proposes in [88] an alternative CT technique that is more accurate and does not involve any cell interface representation of the magnetic field anymore. This method, called the field-interpolated central difference (CD) scheme, consists in computing a time centered averaging of $\mathbf{\Omega}$:

$$\mathbf{\Omega}_{i,j} = -\frac{\mathbf{u}^n_{i,j}\wedge\mathbf{B}^n_{i,j}+\mathbf{u}^\star_{i,j}\wedge\mathbf{B}^\star_{i,j}}{2}.$$

in order to update directly the cell-centered representation of the magnetic field according to the following finite volume scheme:

$$(B_x)^{n+1}_{i,j} = (B_x)^n_{i,j} - \frac{\Delta t}{\Delta y}\left(F^y_{i,j+\frac{1}{2}} - F^y_{i,j-\frac{1}{2}}\right),$$
$$(B_y)^{n+1}_{i,j} = (B_y)^n_{i,j} + \frac{\Delta t}{\Delta x}\left(F^x_{i+\frac{1}{2},j} - F^y_{i-\frac{1}{2},j}\right),$$

the approximate fluxes being given by:

$$F^y_{i,j+\frac{1}{2}} = \frac{\mathbf{\Omega}_{i,j}+\mathbf{\Omega}_{i,j+1}}{2},$$
$$F^x_{i+\frac{1}{2},j} = \frac{\mathbf{\Omega}_{i,j}+\mathbf{\Omega}_{i+1,j}}{2}.$$

It turns out that the resulting scheme preserves the numerical divergence of $\mathbf{B}$ in a more classical sense than the initial CT method and the field-interpolated CT scheme.

Proposition 11. *Let the discrete divergence of* $\mathbf{B}$ *be defined by the following cell centered representation:*

$$(\nabla \cdot \mathbf{B})_{i,j} = \frac{(B_x)_{i+1,j} - (B_x)_{i-1,j}}{2\Delta x} + \frac{(B_y)_{i,j+1} - (B_y)_{i,j-1}}{2\Delta y}.$$

The field-interpolated CD scheme proposed by Tóth [88] implies that:

$$(\nabla \cdot \mathbf{B})^{n+1}_{i,j} = (\nabla \cdot \mathbf{B})^{n}_{i,j}$$

to the accuracy of round-off errors.

Proof. Indeed, one can notice that:

$$\begin{aligned}
(\nabla \cdot \mathbf{B})^{n+1}_{i,j} - (\nabla \cdot \mathbf{B})^{n}_{i,j} = & -\frac{\Delta t}{2\Delta x \Delta y}\left(F^y_{i+1,j+\frac{1}{2}} - F^y_{i+1,j-\frac{1}{2}} + F^y_{i-1,j-\frac{1}{2}} - F^y_{i-1,j+\frac{1}{2}}\right) \\
& + \frac{\Delta t}{2\Delta x \Delta y}\left(F^x_{i+\frac{1}{2},j+1} - F^x_{i-\frac{1}{2},j+1} + F^x_{i-\frac{1}{2},j-1} - F^x_{i+\frac{1}{2},j-1}\right), \\
= & -\frac{\Delta t}{4\Delta x \Delta y}\left(\mathbf{\Omega}_{i+1,j+1} - \mathbf{\Omega}_{i+1,j-1} + \mathbf{\Omega}_{i-1,j-1} - \mathbf{\Omega}_{i-1,j+1}\right) \\
& + \frac{\Delta t}{4\Delta x \Delta y}\left(\mathbf{\Omega}_{i+1,j+1} - \mathbf{\Omega}_{i-1,j+1} + \mathbf{\Omega}_{i-1,j-1} - \mathbf{\Omega}_{i+1,j-1}\right), \\
= & \ 0.
\end{aligned}$$

□

Higher-order CT methods with divergence-free reconstructions

Balsara presents in [1] a method for building divergence-free second-order accurate polynomial approximations of the magnetic field. This technique is the first step for improving the accuracy of CT-type schemes that rely on a face-centered representation of the magnetic field. We here focus on the computation of the divergence-free reconstruction in the (i,j) cell. Let $B^{\pm}_x$ denote the magnetic field at the cell's left and right faces and $B^{\pm}_y$ the magnetic field at the cell's bottom and top faces:

$$\begin{aligned}
B^{\pm}_x &= (B_x)_{i\pm\frac{1}{2},j}, \\
B^{\pm}_y &= (B_y)_{i,j\pm\frac{1}{2}}.
\end{aligned}$$

In order to achieve second-order accuracy, one has to approximate the face-centered representation of the magnetic field using any linear fitting method:

$$\begin{aligned}
B_x(x = x_{i\pm\frac{1}{2}}, y) &= B^{\pm}_x + \Delta B^{\pm}_x y, \\
B_y(x, y = y_{j\pm\frac{1}{2}}) &= B^{\pm}_y + \Delta B^{\pm}_y x.
\end{aligned}$$

Note that due to Ostrogradsky's theorem, the divergence constraint implies that:

$$\left(B^+_x - B^-_x\right)\Delta y + \left(B^+_y - B^-_y\right)\Delta x = 0. \tag{5.14}$$

For the sake of simplicity, we assume from now on that $x_{i\pm\frac{1}{2}} = \pm\Delta x/2$ and $y_{j\pm\frac{1}{2}} = \pm\Delta y/2$. Balsara proposes to build a polynomial approximation of the magnetic field in the whole cell, its general form writes:

$$B_x(x,y) = a_0 + a_x x + a_y y + a_{xx}x^2 + a_{xy}xy + a_{yy}y^2, \tag{5.15}$$

$$B_y(x,y) = b_0 + b_x x + b_y y + b_{xx}x^2 + b_{xy}xy + b_{yy}y^2. \tag{5.16}$$

Since these reconstructions must fit the face-centered linear fittings, one necessarily has $a_{yy} = b_{xx} = 0$ and:

$$a_0 \pm a_x \frac{\Delta x}{2} + a_{xx} \frac{\Delta x^2}{4} = B_x^{\pm}, \tag{5.16a}$$

$$b_0 \pm b_y \frac{\Delta y}{2} + b_{yy} \frac{\Delta y^2}{4} = B_y^{\pm}, \tag{5.16b}$$

$$a_y \pm a_{xy} \frac{\Delta x}{2} = \Delta B_x^{\pm}, \tag{5.16c}$$

$$b_x \pm b_{xy} \frac{\Delta y}{2} = \Delta B_y^{\pm}. \tag{5.16d}$$

At this stage, we have ten unknowns left but only eight linear equations. These are completed by imposing the divergence constraint in a continuous sense to the approximate magnetic field. Indeed, requiring that

$$\partial_x(5.15) + \partial_y(5.16) = 0$$

leads to the following system of equations:

$$\begin{aligned} a_x + b_y &= 0, \\ 2a_{xx} + b_{xy} &= 0, \\ a_{xy} + 2b_{yy} &= 0. \end{aligned}$$

Note that the $a_x + b_y = 0$ constraint is automatically satisfied provided that (5.14) is satisfied. Indeed, from (5.16a) and (5.16b), one can show that:

$$B_x^+ - B_x^- = a_x \Delta x \quad \text{and} \quad B_y^+ - B_y^- = b_y \Delta y$$

and thus

$$\left(B_x^+ - B_x^-\right) \Delta y + \left(B_y^+ - B_y^-\right) \Delta x = (a_x + b_y) \Delta x \Delta y.$$

We finally have a well-posed linear system of ten equations whose solution is given by:

$$\begin{aligned} a_x &= -b_y = \frac{B_x^+ - B_x^-}{\Delta x} = -\frac{B_y^+ - B_y^-}{\Delta y} \\ a_y &= \frac{1}{2\Delta y} \left(\Delta B_x^+ + \Delta B_x^-\right) \\ b_x &= \frac{1}{2\Delta x} \left(\Delta B_y^+ + \Delta B_y^-\right) \\ a_{xy} &= -2b_{yy} = \frac{1}{\Delta x \Delta y} \left(\Delta B_x^+ - \Delta B_x^-\right) \\ b_{xy} &= -2a_{xx} = \frac{1}{\Delta x \Delta y} \left(\Delta B_y^+ - \Delta B_y^-\right) \\ a_0 &= \frac{B_x^+ + B_x^-}{2} - a_{xx} \frac{\Delta x^2}{4} \\ b_0 &= \frac{B_y^+ + B_y^-}{2} - b_{yy} \frac{\Delta y^2}{4}. \end{aligned}$$

The resulting approximate magnetic field satisfies the divergence constraint in both discrete and continuous senses. The reconstruction that is generated by the algorithm presented in this section is second-order accurate but Balsara has extended it to third and fourth-order accuracy (see [3]).

Ability of CT methods to be coupled with GoHy schemes

CT-type methods are globally badly suited to dimensionally split schemes. Indeed, they basically rely on the presevation of the following relation between the divergence and curl operators at a discrete level:

$$\nabla \cdot (\nabla \times \mathbf{v}) = 0,$$

which can hardly be interpreted in a dimensionally split framework. We therefore do not resort to CT methods for enforcing the divergence constraint in GoHy schemes.

5.2.3 The 8-wave formulation

Numerical solvers which do not guarantee a divergence-free evolution of the magnetic field may crash when dealing with multidimensional problem because errors generated by high values of $\nabla \cdot \mathbf{B}$ tend to accumulate and produce unphysical results. Powell's 8-wave formulation aims at propagating these errors, thus preventing them from accumulating on a fixed grid point. To that end, he introduces an alternative form of the ideal MHD system which can be obtained by deriving the MHD equations without assuming that $\mathbf{B}$ satisfies the divergence constraint $\nabla \cdot \mathbf{B} = 0$. In this context, the $\mathbf{J} \wedge \mathbf{B}$ term involved in the momentum equation (1.15) rewrites:

$$\mathbf{J} \wedge \mathbf{B} = \frac{1}{\mu} \left(\nabla \cdot (\mathbf{B} \otimes \mathbf{B}) - (\nabla \cdot \mathbf{B}) \mathbf{B} \right)$$

and the new momentum equation is thus given by:

$$\partial_t (\rho \mathbf{u}) + \nabla \cdot \left(\rho \mathbf{u} \otimes \mathbf{u} + P - \frac{\mathbf{B} \otimes \mathbf{B}}{\mu} \right) + \frac{\nabla \cdot \mathbf{B}}{\mu} \mathbf{B} = \mathbf{0}.$$

Moreover, according to [90], Faraday's equation rewrites:

$$\nabla \times \mathbf{E} = -\partial_t \mathbf{B} - (\nabla \cdot \mathbf{B}) \mathbf{u}. \tag{5.17}$$

Injecting equation (5.17) in the generalized Ohm's law (1.19) then leads to the following magnetic field equation:

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

Finally, Powell's alternative ideal MHD system is given by:

$$\partial_t \begin{pmatrix} \rho \\ \rho \mathbf{u} \\ \mathbf{B} \\ \rho e \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + P \underline{\mathbf{I}} - \frac{(\mathbf{B} \otimes \mathbf{B})}{\mu} \\ \mathbf{u} \otimes \mathbf{B} - \mathbf{B} \otimes \mathbf{u} \\ (\rho e + P) \mathbf{u} - \frac{\mathbf{u} \cdot \mathbf{B}}{\mu} \mathbf{B} \end{pmatrix} = -\frac{\nabla \cdot \mathbf{B}}{\mu} \begin{pmatrix} 0 \\ \mathbf{B} \\ \mu \mathbf{u} \\ \mathbf{u} \cdot \mathbf{B} \end{pmatrix}. \tag{5.18}$$

It is in fact the classical ideal MHD system with an additional right hand side that is proportional to $\nabla \cdot \mathbf{B}$. Numerical experiments show that solving the alternative system (5.18) instead of the initial one (5.1) helps to improve robustness because the divergence errors do not accumulate on a fixed grid point anymore.

The major drawback of this method is that the improved robustness can only be obtained at the cost of conservativity due to the presence of the right hand side term in system (5.18). The 8-wave formulation is thus not well-suited to our approach since one of the main features of finite volume schemes is conservativity preservation.

5.2.4 Hyperbolic divergence cleaning

Divergence cleaning techniques have been proposed by Dedner *et al.* [29] for the ideal MHD equations after having been applied in a similar way to the Maxwell equations (see works by Munz *et al.* [65]). These consists in adding a new unknown function ψ that is coupled to system (5.1) through the following equations:

$$\partial_t \mathbf{B} + \nabla \cdot (\mathbf{u} \otimes \mathbf{B} - \mathbf{B} \otimes \mathbf{u}) + \nabla \psi = 0, \tag{5.19}$$

$$\mathcal{D}(\psi) + \nabla \cdot \mathbf{B} = 0, \tag{5.20}$$

where $\mathcal{D}$ is a linear differential operator. Using (5.19) and (5.20), one can show that ψ and $\nabla \cdot \mathbf{B}$ satisfy the same evolution equation provided that both are sufficiently smooth and that $\mathcal{D}$ commutes with spatial and temporal differential operators. Indeed, notice that:

$$\nabla \cdot (5.19) \Rightarrow \partial_t (\nabla \cdot \mathbf{B}) + \Delta \psi = 0, \tag{5.21}$$

$$\mathcal{D}(5.21) \Rightarrow \partial_t \mathcal{D}(\nabla \cdot \mathbf{B}) + \Delta \mathcal{D}(\psi) = 0, \tag{5.22}$$

$$\partial_t(5.20) \Rightarrow \partial_t \mathcal{D}(\psi) + \partial_t (\nabla \cdot \mathbf{B}) = 0, \tag{5.23}$$

$$\Delta(5.20) \Rightarrow \Delta \mathcal{D}(\psi) + \Delta (\nabla \cdot \mathbf{B}) = 0, \tag{5.24}$$

and thus:

$$\partial_t \mathcal{D}(\nabla \cdot \mathbf{B}) - \Delta (\nabla \cdot \mathbf{B}) = 0, \tag{5.25}$$

$$\partial_t \mathcal{D}(\psi) - \Delta(\psi) = 0. \tag{5.26}$$

Divergence cleaning techniques consists in choosing $\mathcal{D}$ so that divergence errors are advected out of the computational domain. The authors propose different such operators but we only consider the so-called hyperbolic correction here which is the best-suited to GoHy Lagrange-remap schemes since it leads to a conservative equation for ψ. It consists in taking:

$$\mathcal{D}(\psi) = \frac{1}{c_h^2} \partial_t \psi$$

so that equation (5.26) is the wave equation:

$$\partial_{tt} \psi - c_h^2 \Delta \psi = 0.$$

Divergence errors are thus advected to the boundaries with the finite speed $c_h > 0$ that now has to be determined. Dedner *et al.* propose to set c_h uniformly over the computational domain, its value being imposed at each cycle by the time step so that c_h is as great as possible but does not break the stability condition of the underlying scheme. This choice of c_h may therefore vary in time if the time step varies. Nevertheless, in the case of GoHy schemes, numerical experiments have shown that such an approach does not allow to achieve high-order accuracy. The reason for this issue is that $\mathcal{D}$ has to commute with both spatial and temporal differential operators: the only suitable value for c_h consequently is a constant one in space and time. We therefore take:

$$c_h = \omega \max \left(c_f(t=0) \right),$$

where c_f denotes the fast magnetosonic wave speed and ω a real coefficient between 0 and the CFL coefficient.

Remark 22. The ω parameter should be set very carefully to insure stability throughout the whole computation since it can theoretically not be adjusted during the execution. More precisely, if the maximum

value of c_f at time $t > 0$ is greater than its value at $t = 0$ by a factor λ, c_h should be set according to the following formula:

$$c_h = \frac{\omega}{\lambda} \max\left(c_f(t=0)\right).$$

Of course, it is generally not possible to determine an accurate value of λ. The advection speed c_h should therefore be initialized carefully in order to keep a reasonable safety margin regarding the stability condition. As a last resort, c_h could also be adjusted dynamically during the execution but only a few times, otherwise high-order accuracy would be lost.

Though the approach described in remark 22 seems unsatisfying at first glance, it reveals reliable in most practical cases and presents the advantage of being quite cheap in terms of computational cost.

In order to combine the hyperbolic divergence cleaning with GoHy schemes, one has to apply the following steps. First of all, the Lagrangian step described in section 5.1 has to be completed by the resolution of the two following equations in order to compute updated values of B_x and ψ.

Remark 23. Due to the presence of a $\nabla\psi$ term in (5.19), the evolution equation for the tangential component of the magnetic field (*i.e.* B_x when performing the resolution along the x-direction and B_y when dealing with the y-direction) has to be solved in the multidimensional context, contrarily to the one-dimensional case (see remark 20).

These Lagrangian equations are obtained by applying Lemma 1 (see page 60) to the $(B_x,\ \psi - u_x B_x)$ and $\left(\psi,\ c_h^2 B_x - u_x\psi\right)$ couples, leading to:

$$\partial_t(\rho_0\tau B_x) + \partial_X\left(\psi - u_x B_x\right) = 0, \tag{5.27}$$

$$\partial_t(\rho_0\tau\psi) + \partial_X\left(c_h^2 B_x - u_x\psi\right) = 0. \tag{5.28}$$

Lagrangian fluxes for these two equations can then be built as explained in section 5.1. Note that the Cauchy-Kovalevskaya procedure is slightly modified due to the additional equation and to the presence of the ψ unknown (whose time-derivative can easily be derived from (5.28)) in the magnetic field equation. Finally, the additional Lagrangian schemes for B_x and ψ write:

$$\overline{(\rho_0\tau B_x)}_{i,j}^{n+1} = \overline{(\rho_0\tau B_x)}_{i,j}^{n} - \frac{\Delta t}{\Delta x}\left[(\psi - u_x B_x)^*_{i+\frac{1}{2},j} - (\psi - u_x B_x)^*_{i-\frac{1}{2},j}\right],$$

$$\overline{(\rho_0\tau\psi)}_{i,j}^{n+1} = \overline{(\rho_0\tau\psi)}_{i,j}^{n} - \frac{\Delta t}{\Delta x}\left[\left(c_h^2 B_x - u_x\psi\right)^*_{i+\frac{1}{2},j} - \left(c_h^2 B_x - u_x\psi\right)^*_{i-\frac{1}{2},j}\right].$$

Once the Lagrangian step has been performed, one can compute remap fluxes as explained in section (4.1.3) and apply the following remap schemes:

$$\overline{(B_x)}_{i,j}^{n+1} = \overline{(\rho_0\tau B_x)}_{i,j}^{n+1} - \frac{\Delta t}{\Delta x}\left[u^*_{i+\frac{1}{2},j}(B_x)^*_{i+\frac{1}{2},j} - u^*_{i-\frac{1}{2},j}(B_x)^*_{i-\frac{1}{2},j}\right],$$

$$\overline{(\psi)}_{i,j}^{n+1} = \overline{(\rho_0\tau\psi)}_{i,j}^{n+1} - \frac{\Delta t}{\Delta x}\left[u^*_{i+\frac{1}{2},j}(\psi)^*_{i+\frac{1}{2},j} - u^*_{i-\frac{1}{2},j}(\psi)^*_{i-\frac{1}{2},j}\right].$$

The above-described hyperbolic divergence cleaning technique has been used for performing the numerical computations described in section 5.5. Experiments have shown that it increases the restitution time of the whole Lagrange-remap scheme of about 10% whatever the order of accuracy, which seems reasonable in terms of computational cost.

5.3 Artificial viscosity for the ideal MHD equations

As in the hydrodynamics case, the high-order Lagrange-remap schemes that we developed for the ideal MHD equations are subject to Gibbs oscillations on discontinuities. To improve robustness, we again add hyperviscosity to the Lagrangian step. We thus consider the following resistive MHD system:

$$\partial_t \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ \mathbf{B} \\ \rho e \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\otimes\mathbf{u} + P\underline{\mathbf{I}} - \frac{\mathbf{B}\otimes\mathbf{B}}{\mu} \\ \mathbf{u}\otimes\mathbf{B} - \mathbf{B}\otimes\mathbf{u} \\ (\rho e + P)\mathbf{u} - \frac{\mathbf{u}\cdot\mathbf{B}}{\mu}\mathbf{B} \end{pmatrix} = \begin{pmatrix} 0 \\ \nabla\cdot\underline{\boldsymbol{\tau}} \\ -\eta\nabla\times(\nabla\times\mathbf{B}) \\ \nabla\cdot\left(\underline{\boldsymbol{\tau}}\mathbf{u} + \frac{\eta}{\mu}\mathbf{B}\wedge(\nabla\times\mathbf{B}) + \kappa\nabla T\right) \end{pmatrix} \tag{5.29}$$

and replace the physical constants β, ν, κ and η (η denotes the resistive conductivity) with artificial ones (marked with stars) so that consistency with the initial ideal MHD system (5.1) is preserved. We already described how $\beta^\star$, $\nu^\star$ and $\kappa^\star$ have been designed (see section 4.3) and we thus focus on the $\eta^\star$ constant in this section. We propose an hyperviscosity model for $\eta^\star$ which is inspired from previous works by Haugen *et al.* [43]. With notations defined in section 4.3, it writes:

$$\eta^\star = C_\eta \cdot \langle|\nabla^r J|\rangle \cdot \Delta^{r+2}$$

where J denotes the norm of the current density. We recall that C_η is a user-defined constant and r is an integer parameter that is set to 2 for the second and third-order schemes and to 4 for the fourth-order one. In two-dimensional planar geometries, the $\nabla\times(\nabla\times\mathbf{B})$ and $\mathbf{B}\wedge(\nabla\times\mathbf{B})$ terms respectively write:

$$\nabla\times(\nabla\times\mathbf{B}) = \partial_x\begin{pmatrix} 0 \\ -\Lambda \\ -\partial_x B_z \end{pmatrix} + \partial_y\begin{pmatrix} \Lambda \\ 0 \\ -\partial_y B_z \end{pmatrix} \quad \text{and} \quad \mathbf{B}\wedge(\nabla\times\mathbf{B}) = \begin{pmatrix} B_z\partial_x B_z + B_y\Lambda \\ B_z\partial_y B_z - B_x\Lambda \\ -B_x\partial_x B_z - B_y\partial_y B_z \end{pmatrix}$$

with:

$$\Lambda = (\nabla\times\mathbf{B})\cdot\mathbf{e}_z = \partial_x B_y - \partial_y B_x.$$

We thus propose the following discrete cell-centered artificial viscosity terms:

$$\mathbf{V}_{i,j}^{x,*} = \begin{pmatrix} \underline{\boldsymbol{\tau}}_{i,j}^{xx} \\ \underline{\boldsymbol{\tau}}_{i,j}^{xy} \\ \underline{\boldsymbol{\tau}}_{i,j}^{xz} \\ \eta_{i,j}^\star\Lambda_{i,j} \\ \eta_{i,j}^\star\left(\frac{(B_z)_{i+1,j}-(B_z)_{i-1,j}}{2\Delta x}\right) \\ (\underline{\boldsymbol{\tau}}^{xx}u_x)_{i,j} + (\underline{\boldsymbol{\tau}}^{xy}u_y)_{i,j} + (\underline{\boldsymbol{\tau}}^{xz}u_z)_{i,j} + \frac{1}{\mu}\eta_{i,j}^\star\left(\mathbf{B}\wedge(\nabla\times\mathbf{B})\right)_{i,j}^x + \kappa_{i,j}^\star\left(\frac{T_{i+1,j}-T_{i-1,j}}{2\Delta x}\right) \end{pmatrix},$$

$$\mathbf{V}_{i,j}^{y,*} = \begin{pmatrix} \underline{\boldsymbol{\tau}}_{i,j}^{xy} \\ \underline{\boldsymbol{\tau}}_{i,j}^{yy} \\ \underline{\boldsymbol{\tau}}_{i,j}^{yz} \\ -\eta_{i,j}^{\star}\Lambda_{i,j} \\ \eta_{i,j}^{\star}\left(\frac{(B_z)_{i,j+1}-(B_z)_{i,j-1}}{2\Delta y}\right) \\ (\underline{\boldsymbol{\tau}}^{xy}u_x)_{i,j} + (\underline{\boldsymbol{\tau}}^{yy}u_y)_{i,j} + (\underline{\boldsymbol{\tau}}^{yz}u_z)_{i,j} + \frac{1}{\mu}\eta_{i,j}^{\star}\left(\mathbf{B}\wedge(\nabla\times\mathbf{B})\right)_{i,j}^{y} + \kappa_{i,j}^{\star}\left(\frac{T_{i,j+1}-T_{i,j-1}}{2\Delta y}\right) \end{pmatrix}$$

with notations defined in section 4.3 and:

$$\begin{aligned}
\underline{\boldsymbol{\tau}}_{i,j}^{xz} &= \nu_{i,j}^{\star}\frac{(u_z)_{i+1,j}-(u_z)_{i-1,j}}{2\Delta x}, \\
\underline{\boldsymbol{\tau}}_{i,j}^{yz} &= \nu_{i,j}^{\star}\frac{(u_z)_{i,j+1}-(u_z)_{i,j-1}}{2\Delta y}, \\
\Lambda_{i,j} &= \frac{(B_y)_{i+1,j}-(B_y)_{i-1,j}}{2\Delta x} - \frac{(B_x)_{i,j+1}-(B_x)_{i,j-1}}{2\Delta y}, \\
\left(\mathbf{B}\wedge(\nabla\times\mathbf{B})\right)_{i,j}^{x} &= (B_z)_{i,j}\frac{(B_z)_{i+1,j}-(B_z)_{i-1,j}}{2\Delta x} + (B_y)_{i,j}\Lambda_{i,j}, \\
\left(\mathbf{B}\wedge(\nabla\times\mathbf{B})\right)_{i,j}^{y} &= (B_z)_{i,j}\frac{(B_z)_{i,j+1}-(B_z)_{i,j-1}}{2\Delta y} - (B_x)_{i,j}\Lambda_{i,j}.
\end{aligned}$$

These allow to build the following artificial viscosity fluxes:

$$\mathbf{V}_{i+\frac{1}{2},j}^{*} = \frac{1}{2}\left(\mathbf{V}_{i,j}^{x,*} + \mathbf{V}_{i+1,j}^{x,*}\right) \quad \text{and} \quad \mathbf{V}_{i,j+\frac{1}{2}}^{*} = \frac{1}{2}\left(\mathbf{V}_{i,j}^{y,*} + \mathbf{V}_{i,j+1}^{y,*}\right).$$

that have to be added to the right hand side of the Lagrangian scheme (5.6). In terms of computational cost, artificial viscosity increases the restitution time of about 25% for the second-order scheme, 20% for the third-order scheme and 10% for the fourth-order scheme, which are reasonable values.

5.4 High performance computing aspects

We have seen previously (see section 4.4) that GoHy schemes for compressible hydrodynamics have been built so that they are able to take advantage of modern and massively parallel computer architectures. The same strategy has of course been adopted for ideal MHD schemes. We illustrate this feature in this section by providing both single core performance and parallel efficiency figures.

5.4.1 Performance figures

We first focus on estimating the single core performance of GoHy schemes. To that end, we applied the procedure described for the hydrodynamical schemes (see section 4.4.1) to a MHD test problem, more precisely to the advected MHD vortex problem described in section 5.5. Note that the hyperviscosity model and the divergence cleaning have been turned on for performance measurements in order to perform tests that are as representative as possible of real life applications. Both execution times and measured performances have been reported in Table 5.2. These results are as expected better than those obtained with the hydrodynamic solver and thus reveal very satisfying. Indeed, the ideal MHD system being far more complex than the Euler system of equations, the number of floating point operations required for applying

Scheme	Number of FLOPS per cell (Euler)	Number of FLOPS per cell (MHD)
2nd-order	≈ 2900	≈ 5500
3rd-order	≈ 6500	≈ 16500
4th-order	≈ 14000	≈ 70000

Table 5.1: Number of floating point operations required to apply the two-dimensional second, third and fourth-order GoHy schemes in a single cell in both hydrodynamical and magnetohydrodynamical cases.

the whole Lagrange-remap scheme in the ideal MHD case is greater than for the hydrodynamics equations as shown in Table 5.1. As we explained previously (see remark 18 page 88), this implies that the solver spends more time in functions that respect the principles we stated in introduction (see page 13): these do not require any test and operate mostly on contiguous data, thus increasing the global efficiency.

5.4.2 Parallel efficiency figures

We now provide several weak parallel efficiency figures for the second, third and fourth-order accurate MHD-GoHy schemes. Computations have been performed on the advected MHD vortex test problem described in section 5.5 using the same testing procedure than in the hydrodynamical case (see page 91). The hyperviscosity model and the divergence cleaning method have been turned on for parallel efficiency measurements in order to perform tests that are as representative as possible of real life applications. The measured parallel efficiencies have been reported in Table 5.3. These are very satisfying: they are indeed above 95% in all cases.

5.5 Numerical results

To conclude the description of MHD-GoHy high-order dimensionally split Lagrange-remap schemes, we provide in this section numerical results on various test problems taken from the related literature. More precisely, we first compute the experimental order of convergence on smooth solutions and then present results obtained for several non-smooth test problems to illustrate the robustness of the schemes we developed. The divergence cleaning has been turned on for computations on two-dimensional non-smooth problems and revealed particularly useful: in most cases, disabling divergence cleaning causes the computation to crash.

5.5.1 Experimental order of convergence

We first focus on smooth test problems and carry out a convergence analysis. To that end, we consider a one-dimensional analytical solution proposed by Picard [70] and a two-dimensional one - Balsara's advected MHD vortex benchmark [2] - on which we will illustrate the benefits provided by divergence cleaning techniques.

SECOND-ORDER SCHEME				
Mesh size	**Execution time**		**Performance**	
N × N cells	wall time (s)	μs / cell / cycle	GFLOPS	% peak
N = 100	2.32	2.32	2.42	22.7%
N = 200	9.56	2.39	2.31	21.6%
N = 400	40.83	2.55	2.14	20.0%
N = 800	186.36	2.91	1.86	17.5%
THIRD-ORDER SCHEME				
Mesh size	**Execution time**		**Performance**	
N × N cells	wall time (s)	μs / cell / cycle	GFLOPS	% peak
N = 100	5.13	5.13	3.35	31.4%
N = 200	20.98	5.23	3.19	29.9%
N = 400	88.85	5.55	2.96	27.8%
N = 800	406.61	6.35	2.58	24.1%
FOURTH-ORDER SCHEME				
Mesh size	**Execution time**		**Performance**	
N × N cells	wall time (s)	μs / cell / cycle	GFLOPS	% peak
N = 100	18.55	18.55	4.02	37.7%
N = 200	72.32	18.08	3.93	36.9%
N = 400	291.36	18.21	3.82	35.8%
N = 800	1230.38	19.22	3.57	33.5%

Table 5.2: Single core performance figures of the two-dimensional MHD-GoHy schemes. Benchmark: 100 simulation steps on the advected MHD vortex test problem, hyperviscosity model enabled. Test platform: Intel Xeon E5650 CPU @ 2.66 GHz (theoretical peak performance: 2.66 GHz × 4 double precision floating point operations per cycle = 10.64 GFLOPS).

SECOND-ORDER SCHEME						
Mesh size / process	**Parallel efficiency for p processes**					
N × N cells	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
N = 200	100%	97%	97%	97%	97%	96%
N = 400	100%	99%	99%	98%	97%	97%
N = 800	100%	100%	100%	100%	99%	98%
THIRD-ORDER SCHEME						
Mesh size / process	**Parallel efficiency for p processes**					
N × N cells	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
N = 200	100%	98%	98%	98%	98%	97%
N = 400	100%	99%	99%	99%	98%	97%
N = 800	100%	100%	100%	99%	98%	97%
FOURTH-ORDER SCHEME						
Mesh size / process	**Parallel efficiency for p processes**					
N × N cells	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
N = 200	100%	99%	99%	98%	98%	98%
N = 400	100%	100%	99%	99%	98%	97%
N = 800	100%	100%	99%	99%	99%	98%

Table 5.3: Parallel efficiency of the two-dimensional MHD-GoHy schemes. Benchmark: 100 simulation steps on the advected MHD vortex test problem, hyperviscosity model and divergence cleanoing enabled. Test platform: Intel Xeon E5462 processors @ 2.8 GHz, InfiniBand interconnect @ 20 Gbit/s

Picard's one-dimensional analytical solution

We first consider a one-dimensional analytical solution to the ideal MHD system proposed by Picard [70]. Defining:

$$\begin{aligned} R(t) &= \exp\left(\frac{5}{2} - \frac{5+3\ln(t+1)}{2(t+1)}\right), \\ W(t) &= 1 + \frac{3}{2}\ln(t+1), \\ \phi(x,t) &= -\ln\left((t+1)R(t)\right) - \frac{x}{t+1}, \end{aligned}$$

it is given by:

$$\begin{aligned} \rho(x,t) &= \exp\left(\frac{x}{t+1}\right) R(t), \\ p(x,t) &= \rho(x,t)^\gamma, \\ u_x(x,t) &= \frac{x - W(t)}{t+1}, \\ u_y(x,t) &= \sin\left(\phi(x,t)\right), \\ u_z(x,t) &= \cos\left(\phi(x,t)\right), \\ B_y(x,t) &= \sqrt{\frac{\rho(x,t)}{t+1}} \sin\left(\phi(x,t)\right), \\ B_z(x,t) &= \sqrt{\frac{\rho(x,t)}{t+1}} \cos\left(\phi(x,t)\right). \end{aligned}$$

This problem has been set on the $[0;5]$ computational domain with $\gamma = 2$ and $B_x = 0$, boundary conditions being imposed by the analytical solution. We have run computations until time $t = 10$ with the MHD-GoHy schemes and reported the error measurements between the exact and numerical solutions (estimated using a L^1 norm in space and time) in Table 5.4. These show as expected that the experimental order of convergence matches the theoretical one and that it is not affected by hyperviscosity.

Two-dimensional advected MHD vortex

We now focus on the two-dimensional advected MHD vortex problem proposed by Balsara [2]. Defining $r^2 = (x - u_0 t)^2 + (y - v_0 t)^2$, the analytical solution writes:

$$\begin{aligned} \rho(x,y,t) &= 1, \\ \mathbf{u}(x,y,t) &= \frac{1}{2\pi}\exp\left(\frac{1-r^2}{2}\right)(-y,\ x,\ 0)^t + (u_0,\ v_0, 0)^t, \\ \mathbf{B}(x,y,t) &= \frac{\mu}{2\pi}\exp\left(\frac{1-r^2}{2}\right)(-y,\ x,\ 0)^t, \\ p(x,y,t) &= 1 + \left(\frac{\mu^2(1-r^2) - 4\pi}{32\pi^3}\right)\exp\left(1-r^2\right), \end{aligned}$$

with $\gamma = 5/3$ and $\mu = 4\pi$. This test problem has been set on the $[-10;10]^2$ computational domain with periodic boundary conditions and $u_0 = v_0 = 1$ so that the vortex is advected along the $y = x$ axis. We have run computations until time $t = 20$ with a CFL coefficient of 0.9. Note that divergence cleaning has

WITHOUT HYPERVISCOSITY						
Mesh size	**2nd-order**		**3rd-order**		**4th-order**	
(N × N cells)	error	order	error	order	error	order
N = 100	5.42e-02		8.70e-03		2.12e-04	
N = 200	1.30e-02	2.06	1.08e-03	3.01	1.30e-05	4.03
N = 400	3.20e-03	2.03	1.35e-04	3.00	8.07e-07	4.01
N = 800	7.93e-04	2.01	1.69e-05	3.00	5.02e-08	4.01
N = 1600	1.98e-04	2.00	2.11e-06	3.00	3.13e-09	4.00
WITH HYPERVISCOSITY ($C_\beta = C_\nu = C_\kappa = C_\eta = 1$)						
Mesh size	**2nd-order**		**3rd-order**		**4th-order**	
(N × N cells)	error	order	error	order	error	order
N = 100	5.40e-02		8.60e-03		2.12e-04	
N = 200	1.30e-02	2.06	1.08e-03	3.00	1.30e-05	4.03
N = 400	3.20e-03	2.02	1.35e-04	3.00	8.07e-07	4.01
N = 800	7.93e-04	2.01	1.68e-05	3.00	5.02e-08	4.01
N = 1600	1.98e-04	2.00	2.11e-06	3.00	3.13e-09	4.00

Table 5.4: Error in space and time between the analytical and numerical solutions for Picard's one-dimensional analytical solution.

to be enabled to prevent the solver from crashing due to the accumulation of divergence errors as we will see in the sequel. Table 5.5 reports the error (measured using a L^1 norm in space and time) between the analytical and numerical solutions. Once again, we see that the experimental order of convergence matches the theoretical one and that it is affected neither by hyperviscosity nor by divergence cleaning.

In order to illustrate the improvements provided by the divergence cleaning technique, we have plotted in Figure 5.1 the following representation of $\nabla \cdot \mathbf{B}$:

$$(\nabla \cdot \mathbf{B})_{i,j} = \frac{(B_x)_{i+1,j} - (B_x)_{i-1,j}}{2\Delta x} + \frac{(B_y)_{i,j+1} - (B_y)_{i,j-1}}{2\Delta y}.$$

at time $t = 4$, $t = 5$, $t = 6$ and $t = 7$ for the advected vortex test problem on 200×200 cells with and without divergence cleaning. When the divergence cleaning is disabled, some growing unphysical perturbations appear in the tail of the moving vortex and make the computation crash at time $t \approx 7.5$. No such unphysical structures can be observed when divergence cleaning is enabled (in this case, divergence errors are advected and can not accumulate) and the computation can be performed until final time. These results are confirmed by the plot of the L^1 norm of $\nabla \cdot \mathbf{B}$ versus time in Figure 5.2: when divergence cleaning is disabled, the L^1 norm of $\nabla \cdot \mathbf{B}$ grows exponentially from time $t \approx 5.5$ whereas it stays low when divergence cleaning is activated.

WITHOUT HYPERVISCOSITY						
Mesh size	**2nd-order**		**3rd-order**		**4th-order**	
(N × N cells)	error	order	error	order	error	order
N = 100	8.27e+01		5.27e+01		6.79e+00	
N = 200	2.07e+01	2.00	7.65e+00	2.78	4.07e-01	4.06
N = 400	4.93e+00	2.07	9.81e-01	2.96	2.47e-02	4.04
N = 800	1.20e+00	2.03	1.23e-01	2.99	1.53e-03	4.01
N = 1600	2.99e-01	2.01	1.54e-02	3.00	9.55e-05	4.00
WITH HYPERVISCOSITY ($C_\beta = 1,\ C_\nu = C_\kappa = C_\eta = 0.1$)						
Mesh size	**2nd-order**		**3rd-order**		**4th-order**	
(N × N cells)	error	order	error	order	error	order
N = 100	8.51e+01		5.52e+01		7.94e+00	
N = 200	2.10e+01	2.02	8.00e+00	2.79	4.77e-01	4.06
N = 400	4.95e+00	2.08	1.02e+00	2.97	3.07e-02	4.01
N = 800	1.21e+00	2.04	1.28e-01	3.00	1.96e-03	4.01
N = 1600	2.99e-01	2.01	1.59e-02	3.00	1.22e-04	4.00

Table 5.5: Error in space and time between the analytical and numerical solutions for the two-dimensional MHD vortex benchmark.

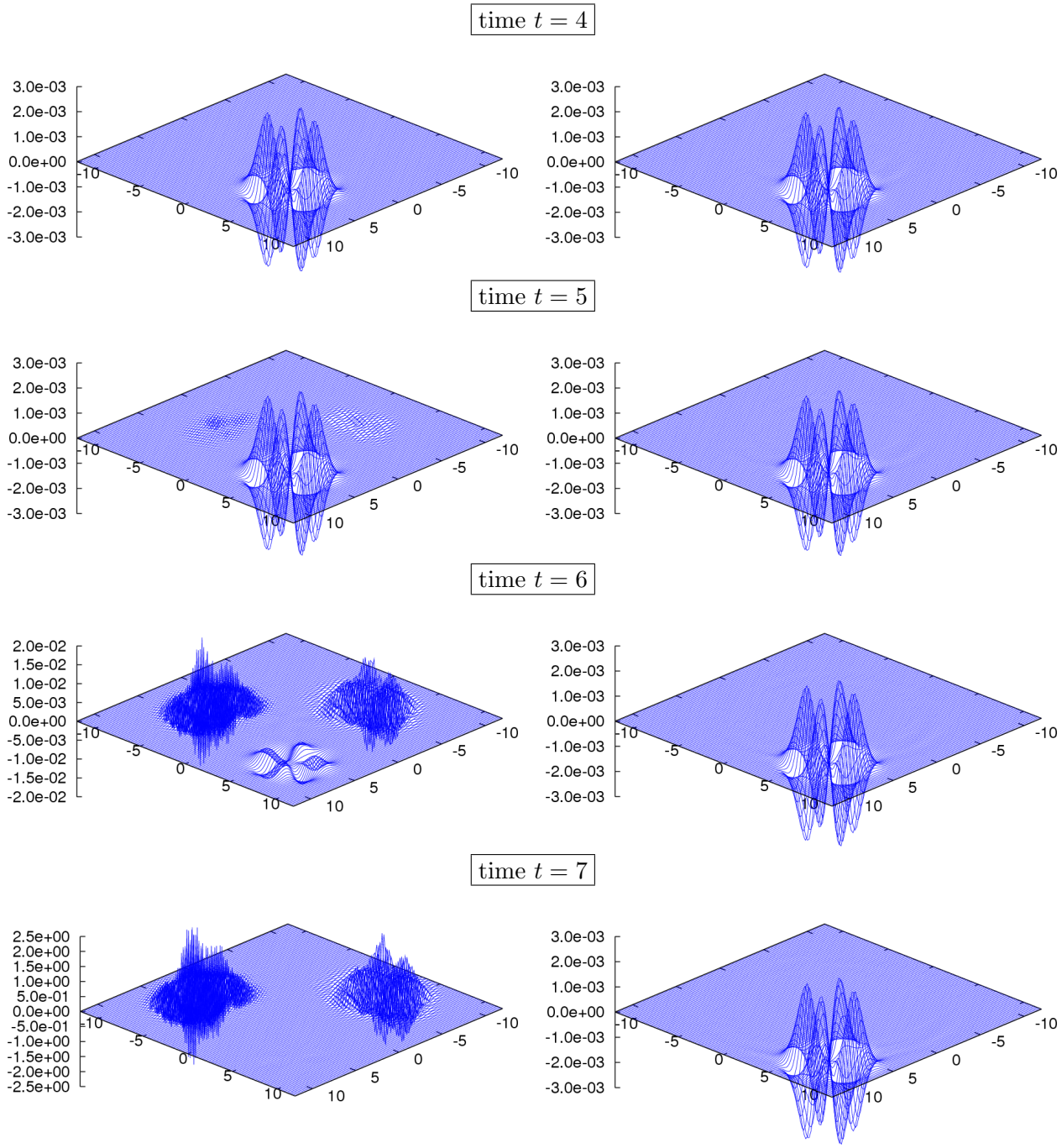


Figure 5.1: Representation of $\nabla \cdot \mathbf{B}$ at different times without (left) and with (right) divergence cleaning for the advected vortex test problem.

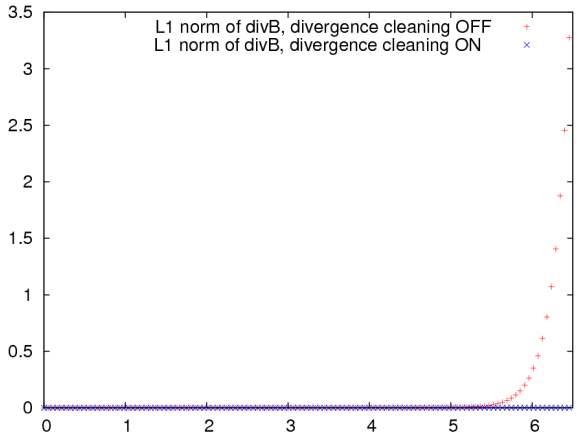


Figure 5.2: Evolution of the L^1 norm of $\nabla \cdot \mathbf{B}$ versus time with and without divergence cleaning for the advected MHD vortex test problem.

5.5.2 Ryu-Jones "all seven waves" Riemann problem

We now consider a one-dimensional Riemann problem proposed by Ryu and Jones [73] with the initial left and right states given by:

$$(\rho,\ u_x,\ u_y,\ u_z,\ B_y,\ B_z,\ p)_L = (1.08,\ ,1.2,\ 0.01,\ 0.5,\ 3.6,\ 2,\ 0.95),$$
$$(\rho,\ u_x,\ u_y,\ u_z,\ B_y,\ B_z,\ p)_R = (1,\ ,0,\ 0,\ 0,\ 4,\ 2,\ 1),$$

This test problem has been set on the $[-0.5; 0.5]$ domain with open boundary conditions and the following parameters: $B_x = 2$, $\gamma = 5/3$, $\mu = 1$. We have performed resolution until time $t = 0.2$ on 600 cells with the fourth-order scheme and a CFL coefficient of 0.7. Figure 5.5.2 plots several primitive variables at final time with and without hyperviscosity (when hyperviscosity is enabled, parameters are $C_\beta = C_\nu = 1$, $C_\kappa = 5$ and $C_\eta = 2$). This benchmark is interesting in the sense that it exhibits all kind of MHD waves: three left-going and right-going waves (a fast shock, a rotational discontinuity and a slow shock) on each side of the contact discontinuity. Whereas the base scheme generates Gibbs oscillations, we can notice that these are strongly reduced thanks to hyperviscosity, leading to a satisfying approximation of the exact solution on each kind of wave.

5.5.3 Orszag-Tang vortex

First proposed for incompressible flows [69], the Orszag-Tang vortex test problem has been adapted for compressible flows by Dahlburg and Picone [24, 25]. It consists in an initially smooth fluid that evolves gradually towards turbulence and thus becomes very complex. The initial vortex structure is given by $\rho = 25/9$, $p = 5/3$, $\mathbf{u} = (-\sin y,\ \sin x,\ 0)^t$ and $\mathbf{B} = (-\sin y,\ \sin 2x,\ 0)^t$. The ratio of specific heats is set to $5/3$ and the magnetic permeability to 1. Numerically, we set this test on the $[0; 2\pi]^2$ domain with periodic boundary conditions. The resolution is performed on 400×400 cells until time $t = 3.14$ with a CFL coefficient of 0.7, the hyperviscosity model being enabled with following parameters: $C_\beta = 5$, $C_\nu = 1$, $C_\kappa = 0.01$ and $C_\eta = 0.001$. Figure 5.4 plots the density, gas pressure, velocity and magnetic pressure obtained at final time with the fourth-order scheme. It shows a good agreement with numerical founds given in the literature (see [88, 78] for example).

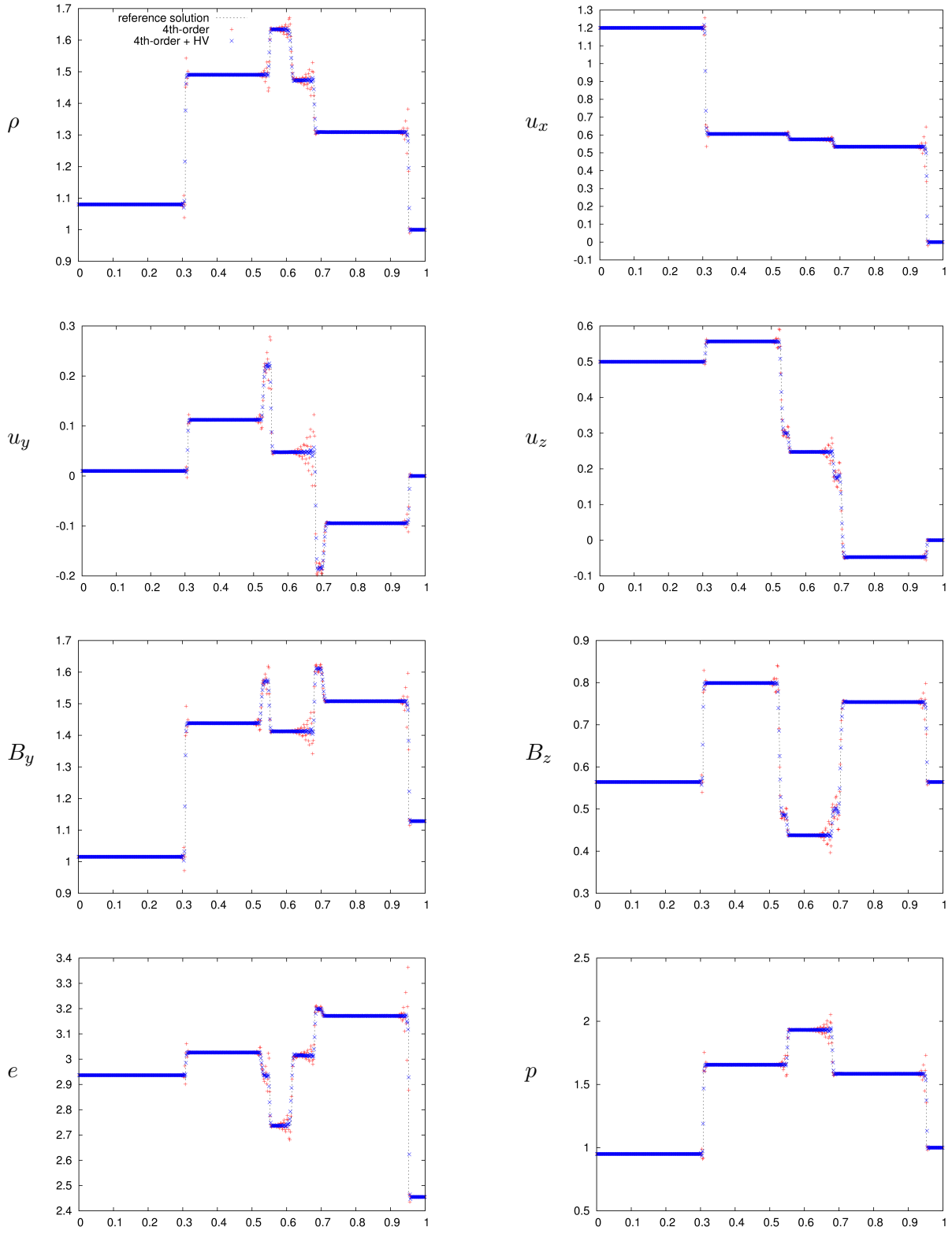


Figure 5.3: Numerical solution of the "all seven waves" MHD Riemann problem at $t = 0.2$ obtained with the fourth-order scheme on 600 cells. The reference solution has been obtained using a first-order accurate solver for the ideal MHD equations proposed by Bezard *et al.* [8] on a mesh of 100,000 cells.

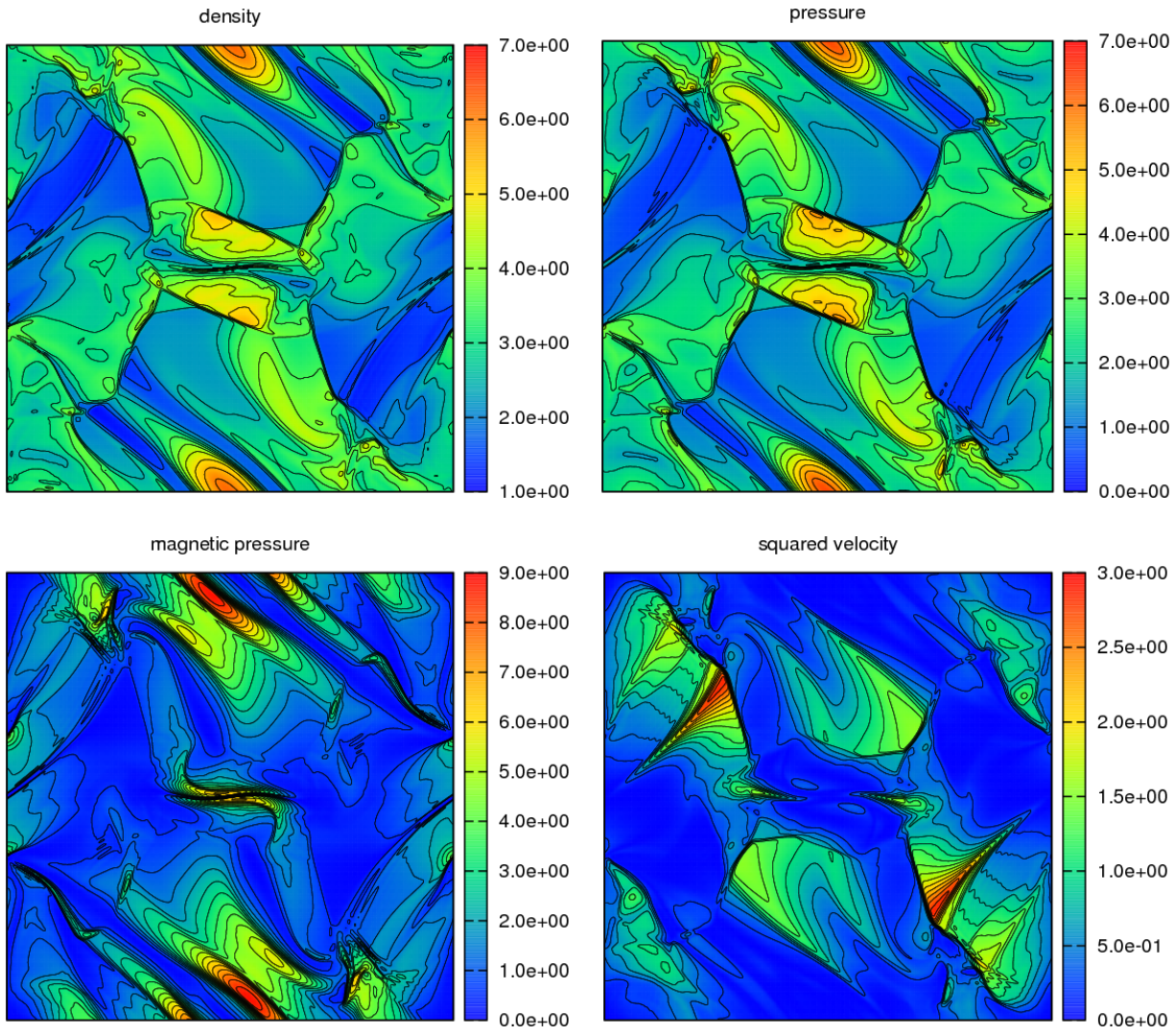


Figure 5.4: Numerical solution of the Orszag-Tang vortex test problem at time $t = 3.14$ obtained with the fourth-order scheme on 400×400 cells.

5.5.4 Rotor problem

We now consider Tóth's second rotor problem [88]. It consists in a dense and rapidly spinning disk of fluid in the center of the domain that launches torsional Alfvén waves and is progressively confined in an oblong shape due to the presence of a magnetic field. Defining $r^2 = (x-0.5)^2+(y-0.5)^2$, the initial density and velocities are given by:

$$(\rho,\ u_x,\ u_y) = \begin{cases} (10,\ -u_0(y-0.5)/r_0,\ u_0(x-0.5)/r_0) & \text{if } r \leq r_0, \\ (1+9f(r),\ -u_0 f(r)(y-0.5)/r,\ u_0 f(r)(x-0.5)/r) & \text{if } r_0 < r \leq r_1, \\ (1,\ 0,\ 0) & \text{if } r \geq r_1, \end{cases}$$

where f is a taper function that smoothens the initial profiles:

$$f(r) = \frac{r_1 - r}{r_1 - r_0}.$$

The initial pressure and magnetic field are uniform: $p = 0.5$, $B_x = 2.5$, $B_y = 0$. Numerically, this test problem is set on the $[0;1]^2$ square with open boundary conditions and the following parameters: $r_0 = 0.1$, $r_1 = 0.115$, $u_0 = 1$, $\mu = \sqrt{4\pi}$ and $\gamma = 5/3$. We have run computations on 400×400 cells until time $t = 0.295$ with the fourth-order scheme and a CFL coefficient of 0.7. Hyperviscosity is enabled with the following coeffcients: $C_\beta = 5$, $C_\nu = 1$, $C_\kappa = 0.01$, $C_\eta = 0.001$. Figure 5.5 plots the density, gas pressure, velocity and magnetic pressure obtained at final time and shows a good agreement with numerical results given in the literature (see [56, 7] for example).

5.5.5 Blast problem

We conclude this chapter with a stringent test problem that has been proposed by Balsara [2]. The initial condition consists in a central disk of radius 0.1 where the fluid has a large overpressure ($p = 1000$) compared to the ambient fluid ($p = 0.1$), leading to very strong magnetosonic shocks that spread out in every angular direction. Moreover, the fluid is initially at rest, its density is set to 1 and a magnetic field with a magnitude of 100 is initialized along the x direction. This benchmark has been set on the $[0;1]^2$ unit square with $\gamma = 1.4$ and $\mu = 1$. We have run computations on 400×400 cells with the fourth-order scheme and a CFL coefficient of 0.6 until time $t = 0.01$. Hyperviscosity is enabled with $C_\beta = 10$, $C_\nu = 1$ and $C_\kappa = C_\eta = 0.1$. Figure 5.6 plots the density, gas pressure, velocity and magnetic pressure obtained at final time and shows a good agreement with numerical results given in the literature (see [3, 5] for example). Note that the computation of fourth-order accurate point-wise values of conservative variables has been disabled in a few pathologic cells because it sometimes generated negative pressure values. This could be done without losing conservativity contrarily to a previous version of MHD-GoHy schemes which had been presented in [93] whose reconstruction step (see section 4.2.2) was performed on conservative variables. This highlights the flexibility gain (and the increased robustness) obtained by moving transverse reconstruction operations on fluxes.

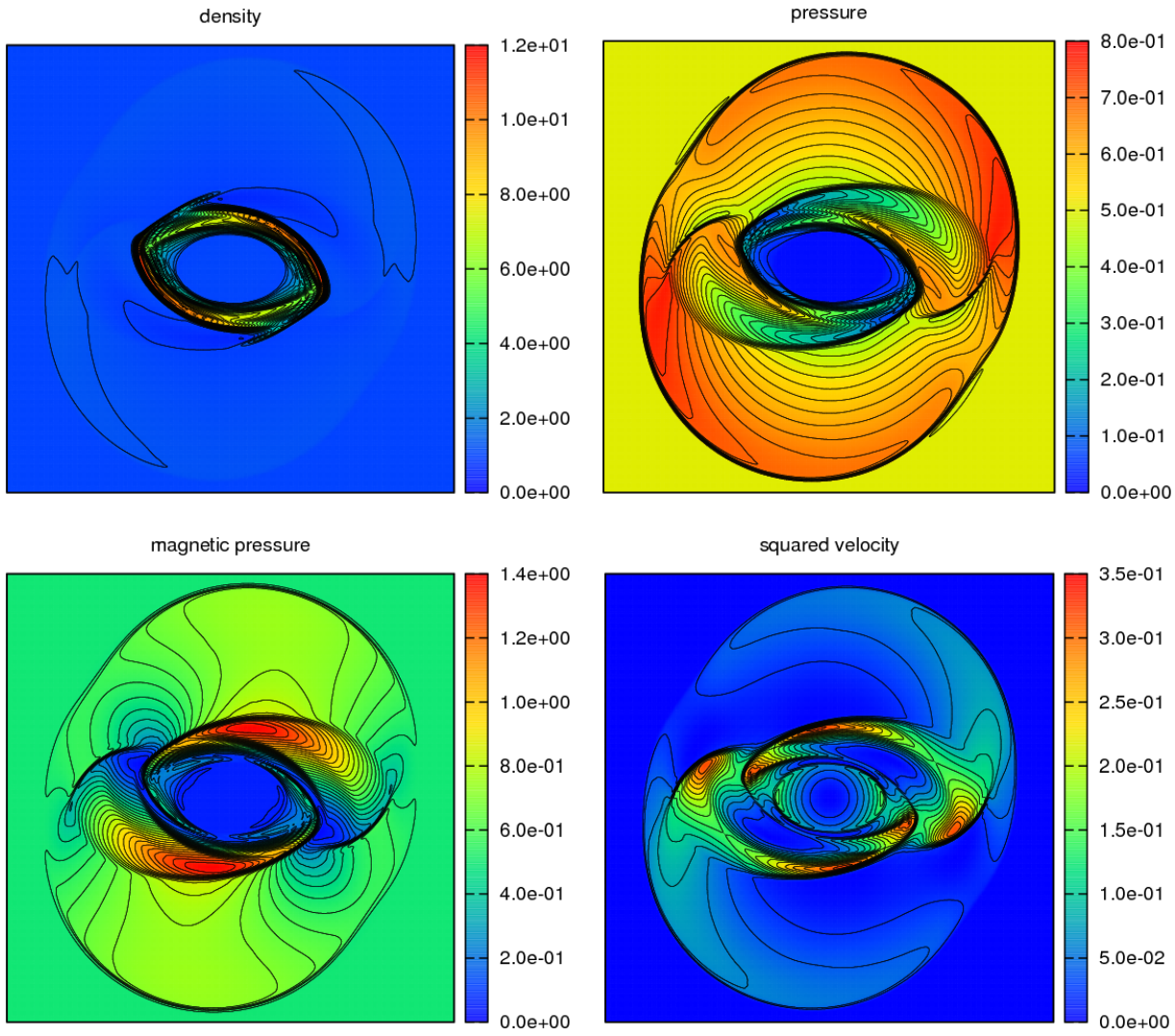


Figure 5.5: Numerical solution of the rotor problem at time $t = 0.295$ obtained with the fourth-order scheme on 400×400 cells.

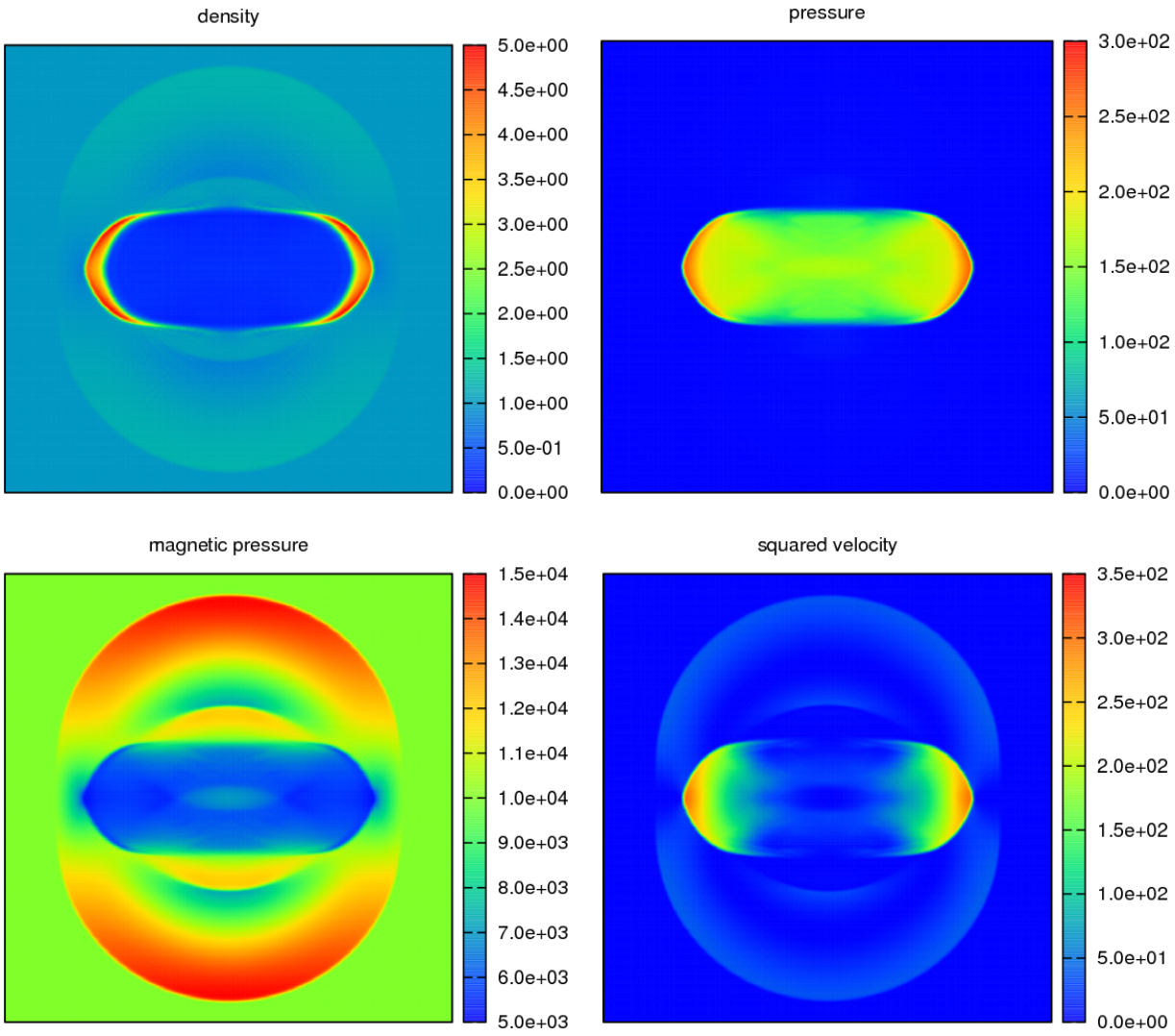


Figure 5.6: Numerical solution of Balsara's blast wave problem at time $t = 0.01$ obtained with the fourth-order scheme on 400×400 cells.

Chapter 6

Extension to axisymmetric geometries

We propose in this chapter an extension of GoHy schemes to one-dimensional and two-dimensional axisymmetric geometries. More precisely, we consider the (z, r, θ) system of coordinates in the sequel and assume that the flow is invariant by rotation around the z-axis, thus cancelling the space derivatives along the θ-direction. Note that we only describe the construction of numerical schemes for the axisymmetric ideal MHD equations: schemes for hydrodynamics can easily be obtained by removing the magnetic field's contribution.

We encounter two main difficulties when dealing with axisymmetric geometries. On the one hand, we will see that both hydrodynamics and ideal MHD systems of equations involve a non-conservative source term in this context. This additional term requires a particular attention since it has to be evaluated in a sufficiently accurate way in order to build high-order numerical schemes. In this study, we propose to approximate it using a methodology that is similar to the one used for building high-order accurate Lagrangian fluxes. On the other hand, we will point out an issue that arises for the computation of high-order Lagrangian fluxes in the axisymmetric case. These indeed involve $1/r$ terms that have to be computed at cell interfaces and thus have to be evaluated carefully along the $r = 0$ axis. This is not a problem from a theoretical point of view - solutions that are defined along the $r = 0$ axis naturally lead to well-defined Lagrangian fluxes - but from an algorithmic point of view since we resort to an automated procedure (see section 4.1.2) for computing approximate fluxes. We propose in this chapter a modification of this procedure that avoids the computation of unphysical values along the $r = 0$ axis.

The outline of this chapter is the following. We first apply the methodology described in section 4.1.2 to the one-dimensional axisymmetric ideal MHD system. As indicated previously, we insist above all on the approximation of the non-conservative term and on the automated computation of Lagrangian fluxes along the $r = 0$ axis. We then give a few details about the two-dimensional case, in particular about the axisymmetric extension of the reconstruction step, divergence cleaning techniques and artificial viscosity. Several numerical results are finally provided on both smooth and non-smooth test problems in order to illustrate the convergence and robustness features of GoHy schemes. Note that we do not present performance and parallel efficiency figures in this chapter: these are indeed similar to the ones given in chapter 5 for the planar case.

6.1 One-dimensional axisymmetric ideal MHD equations

The one-dimensional ideal MHD equations in axisymmetric geometry write:

$$\partial_t \begin{pmatrix} r\rho \\ r\rho u_r \\ r\rho u_\theta \\ r\rho u_z \\ rB_r \\ B_\theta \\ rB_z \\ r\rho e \end{pmatrix} + \partial_r \begin{pmatrix} r\rho u_r \\ r\rho u_r^2 + rP - r\frac{B_r^2}{\mu} \\ r\rho u_r u_\theta - r\frac{B_r B_\theta}{\mu} \\ r\rho u_r u_z - r\frac{B_r B_z}{\mu} \\ 0 \\ u_r B_\theta - u_\theta B_r \\ r u_r B_z - r u_z B_r \\ r(\rho e + P)u_r - r\frac{\mathbf{u}\cdot\mathbf{B}}{\mu} B_r \end{pmatrix} = \begin{pmatrix} 0 \\ \rho u_\theta^2 + P - \frac{B_\theta^2}{\mu} \\ -\rho u_r u_\theta + \frac{B_r B_\theta}{\mu} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{6.1}$$

We recall that P denotes the total pressure defined by:

$$P = p + \frac{\mathbf{B}\cdot\mathbf{B}}{2\mu}.$$

We describe in this section how to build GoHy schemes for the one-dimensional system (6.1). To that end, it is first rewritten in Lagrangian coordinates and then discretized in the finite volume framework. We insist in particular on the high-order accurate approximation of the non-conservative term and on the practical computation of Lagrangian fluxes along the $r = 0$ axis. We will also see that the equation on B_θ has to be treated in a specific way since its structure slightly differs from the other ones.

Remark 24. In the one-dimensional case, since $\partial_t(rB_r) = 0$, the complete finite volume scheme for B_r simply writes:

$$\overline{(B_r)}_i^{n+1} = \overline{(B_r)}_i^n \quad \text{with} \quad \overline{(B_r)}_i^n = \frac{1}{r_i \Delta r} \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} rB_r(r,t)\, dr.$$

Note that this approach is only valid in the one-dimensional case: it should not be applied in the multidimensional case as explained in the sequel (see section 6.2.2).

6.1.1 Lagrangian step

We first rewrite system (6.1) in Lagrangian coordinates. To that end, we introduce the the $(r,t) \to (R,t)$ variable change given by:

$$dr(R,t) = J(R,t)\, dR + u_r(R,t)\, dt$$

which is actually similar to the planar variable change (4.3). The previously mentioned relations between partial derivatives (see Lemma 1 page 60) are thus suitable in the axisymmetric case. We nevertheless write them alternatively in order to add the source term's contribution and obtain the following reference relation for any (ϕ, ψ, χ) triple:

$$\left[\partial_t(J\phi) + \partial_R \psi - J\chi\right](R,t) = \left[J \cdot (\partial_t \phi + \partial_r(\phi u_r + \psi) - \chi)\right](r,t). \tag{6.2}$$

The Jacobian can be determined by applying (6.2) to the $(r\rho, 0, 0)$ triple, leading to:

$$\partial_t (r\rho J) = 0.$$

J is thus given by:

$$J = \frac{R\rho_0}{r\rho}$$

where ρ_0 denotes the initial density. Applying relation (6.2) to appropriately chosen triples, one finally obtains the one-dimensional ideal MHD system in Lagrangian coordinates:

$$\partial_t \begin{pmatrix} R\rho_0\tau \\ R\rho_0 u_r \\ R\rho_0 u_\theta \\ R\rho_0 u_z \\ R\rho_0\tau B_r \\ JB_\theta \\ R\rho_0\tau B_z \\ R\rho_0 e \end{pmatrix} + \partial_R \begin{pmatrix} -ru_r \\ rP - r\frac{B_r^2}{\mu} \\ -r\frac{B_r B_\theta}{\mu} \\ -r\frac{B_r B_z}{\mu} \\ -ru_r B_r \\ -u_\theta B_r \\ -ru_z B_r \\ rPu_r - r\frac{\mathbf{u}\cdot\mathbf{B}}{\mu}B_r \end{pmatrix} = J \begin{pmatrix} 0 \\ \rho u_\theta^2 + P - \frac{B_\theta^2}{\mu} \\ -\rho u_r u_\theta + \frac{B_r B_\theta}{\mu} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{6.3}$$

Remark 25. One can notice that the B_θ equation differs from the other ones: its structure is indeed similar to planar equations studied in previous chapters. It is thus discretized and solved using the planar Lagrange-remap procedure. Further details about this point are given in section 6.1.3.

We now consider the following set of Lagrangian equations:

$$\partial_t(R\mathbf{U}) + \partial_R\mathbf{F}(\mathbf{U}) = \mathbf{S}(\mathbf{U}) \tag{6.4}$$

with:

$$\mathbf{U} = \begin{pmatrix} \rho_0 u_r \\ \rho_0 u_\theta \\ \rho_0 u_z \\ \rho_0\tau B_z \\ \rho_0 e \end{pmatrix},\ \mathbf{F}(\mathbf{U}) = \begin{pmatrix} rP - r\frac{B_r^2}{\mu} \\ -r\frac{B_r B_\theta}{\mu} \\ -r\frac{B_r B_z}{\mu} \\ -ru_z B_r \\ rPu_r - r\frac{\mathbf{u}\cdot\mathbf{B}}{\mu}B_r \end{pmatrix},\ \mathbf{S}(\mathbf{U}) = \begin{pmatrix} \rho u_\theta^2 + P - \frac{B_\theta^2}{\mu} \\ -\rho u_r u_\theta + \frac{B_r B_\theta}{\mu} \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

System (6.4) does not include:

- the equation on $(\rho_0\tau)$ since there is no need to compute updated values of $(\rho_0\tau)$ as we indicated previously (see remark 9 page 71);
- the equation on B_r since the complete scheme for B_r is trivial in the one-dimensional case (see remark 24);
- the equation on B_θ since it is treated differently (see remark 25).

We discretize system (6.4) on the $[R_{i-\frac{1}{2}}; R_{i+\frac{1}{2}}]$ regular grid (*i.e.* with a constant $\Delta R = \Delta r$ mesh size) using a finite volume method. This approach leads to the following numerical scheme:

$$\frac{\overline{\mathbf{U}}_i^{n+1} - \overline{\mathbf{U}}_i^n}{\Delta t} + \frac{\mathbf{F}^*_{i+\frac{1}{2}} - \mathbf{F}^*_{i-\frac{1}{2}}}{R_i\Delta R} = \frac{\Delta t}{R_i}\mathbf{S}_i^*$$

where $\overline{\mathbf{U}}_i^n$ denotes the following cell average of $\mathbf{U}$ over $[R_{i-\frac{1}{2}}; R_{i+\frac{1}{2}}]$ at time t^n:

$$\overline{\mathbf{U}}_i^n = \frac{1}{R_i \Delta R} \int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} R\mathbf{U}(R, t^n)\, dR,$$

$\mathbf{F}^*_{i+\frac{1}{2}}$ the Lagrangian flux at the $R_{i+\frac{1}{2}}$ boundary:

$$\mathbf{F}^*_{i+\frac{1}{2}} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathbf{F}(\mathbf{U})(R_{i+\frac{1}{2}}, t)\, dt,$$

and $\mathbf{S}_i^*$ the time-space average of the Lagrangian source term:

$$\mathbf{S}_i^* = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \left[\frac{1}{\Delta R} \int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} \mathbf{S}(\mathbf{U})(R, t)\, dR \right] dt.$$

In the axisymmetric case, building a high-order Lagrangian scheme supposes to evaluate high-order accurate approximations of fluxes and source terms. We propose such approximate terms in the sequel.

Lagrangian fluxes computation

As in the planar case, approximate flux terms are obtained thanks to a truncated Taylor expansion and the resulting time-derivatives are replaced by space-derivatives using the Cauchy-Kovalevskaya procedure. While the time-derivatives of τ, $\mathbf{u}$ and $\mathbf{B}$ can be easily derived from system (6.3), one has once again to pay attention to the pressure term. From the second law of thermodynamics, one can show that the pressure satisfies the following evolution equation for any equation of state:

$$\partial_t p = -\frac{(\rho c)^2}{R\rho_0} \partial_R(r u_r) - \frac{\Gamma(\mathbf{u} \cdot \mathbf{B})}{R\rho_0 \tau \mu} \partial_R(r B_r). \tag{6.5}$$

This result can nevertheless lead to different approximate Lagrangian fluxes depending on the considered context. It can indeed be simplified in several specific cases in order to reduce the cost of Lagrangian fluxes' computation.

- In the one-dimensional case, the divergence constraint implies that $\partial_R(r B_r) = 0$. The expression of the pressure's time-derivative is thus similar to the one given in section 4.1.2 for planar hydrodynamics. As a matter of fact, the expression of the Lagrangian sound speed's time-derivative is also similar to the one that had been derived in the planar hydrodynamical case. These write:

$$\partial_t p = -\frac{(\rho c)^2}{R\rho_0} \partial_R(r u_r) \quad \text{and} \quad \partial_t (\rho c)^2 = -\frac{2\rho(\rho c)^2 \mathcal{G}}{R\rho_0 \tau} \partial_R(r u_r). \tag{6.6}$$

  We recall that these expressions are valid for any equation of state and allow to build Lagrangian fluxes independently from the equation of state up to third-order accuracy. From fourth-order accuracy, we assume for the sake of simplicity that $\partial_t \mathcal{G} = 0$, which is an assumption satisfied by ideal gases for example (see remark 7 page 66).

- In the case of multidimensional computations in the orthogonal MHD framework, the only possibly non-zero component of the magnetic field is the one that is perpendicular to the computational domain, *i.e.* B_θ here. B_r and B_z are thus identically zero in this context and the spatial derivative of the magnetic field included in (6.5) can be removed for both dimensional sweeps along the z and r directions. The above-given time-derivatives' expressions can therefore be used.

- In the case of multidimensional MHD computations, the divergence constraint does not imply that the spatial derivative of the magnetic field included in (6.5) is zero. The expression of the pressure's time-derivative given by (6.5) thus cannot be simplified. It allows to build Lagrangian fluxes that are valid for any equation of state up to second-order accuracy. From third-order accuracy, we restrict for the sake of simplicity to ideal gases. In this case, the Lagrangian sound speed's time-derivative can be obtained by differentiating its definition:

$$\begin{aligned}\partial_t(\rho c)^2 &= \gamma\partial_t\left(\frac{p}{\tau}\right) \text{ since } \gamma \text{ is a constant for ideal gases}\\ &= \frac{\gamma}{\tau}\partial_t p - \frac{(\rho c)^2}{\tau}\partial_t\tau\\ &= -\frac{(\gamma+1)(\rho c)^2}{R\rho_0\tau}\partial_R(ru_r) - \frac{\gamma\Gamma(\mathbf{u}\cdot\mathbf{B})}{R\rho_0\tau^2\mu}\partial_R(rB_r)\end{aligned}$$

  whereas the time-derivative of $\mathcal{G}$ is zero (see remark 7 page 66).

Once the Cauchy-Kovalevskaya procedure has been applied, cell-centered point-wise values of conservative variables have to be computed according to the following high-order accurate formula:

$$\begin{cases}(\rho_0\phi)_i^n = c_0^N\cdot\overline{(\rho_0\phi)}_i^n + \displaystyle\sum_{k=1}^{\lfloor N/2\rfloor} c_k^N\cdot\left(\frac{r_{i+k}}{r_i}\overline{(\rho_0\phi)}_{i+k}^n + \frac{r_{i-k}}{r_i}\overline{(\rho_0\phi)}_{i-k}^n\right) & \text{for } \phi\in\{1,\mathbf{u},\tau B_r,\tau B_z,e\}\\ (B_\theta)_{i,j}^n = c_0^N\cdot\overline{(JB_\theta)}_i^n + \displaystyle\sum_{k=1}^{\lfloor N/2\rfloor} c_k^N\cdot\left(\overline{(JB_\theta)}_{i+k}^n + \overline{(JB_\theta)}_{i-k}^n\right) & \text{(see section 6.1.3),}\end{cases} \tag{6.7}$$

where N is the desired order of accuracy. One can then determine estimations of quantities involved in the truncated Taylor expansion using finite difference approximations given by (4.16) and evaluate the high-order accurate Lagrangian fluxes as explained in the previous chapters.

Practical computation of Lagrangian fluxes along the $r=0$ axis

Computing high-order accurate Lagrangian fluxes with the help of the Cauchy-Kovalevskaya procedure leads to expressions that involve $1/r$ terms on cell interfaces and thus have to be evaluated carefully along the $r=0$ axis. This distinctive characteristic of the axisymmetric case does not pose a problem in terms of definition: solutions that are well-defined at $r=0$ theoretically lead to well-defined Lagrangian fluxes. But it represents a difficulty in terms of discretization: fluxes have to be properly evaluated along the $r=0$ axis so that they do not lead to unphysical numerical solutions. This is all the more an issue when Lagrangian fluxes are generated using an automated procedure, which is the case for GoHy schemes since time-derivatives' calculations can hardly be done by hand from a given order of accuracy. We thus propose in this section a method for computing approximate fluxes as properly as possible along the $r=0$ axis. It consists in three modifications of the procedure given in section 4.1.2. Note that though the strategy described below reveals efficient in practice, there is no theoretical evidence that validates it.

1. First of all, time-derivatives involved in numerical fluxes should be computed in one step instead of evaluating the time-derivative of each variable separately. For example, one should try to evaluate directly the whole $\partial_t(rP)$ at the $r=0$ cell interface instead of computing approximations of $\partial_t r$ and $\partial_t P$ separately and then mixing them together. Such an approach obviously reduces the number of $1/r$ terms in approximate Lagrangian fluxes but it also makes their computation more expensive. This technique should therefore only be used along the $r=0$ axis.

2. There however still may be some $1/r$ terms left, especially when dealing with high-order Lagrangian fluxes. Another possible modification consists in interpolating pathological zero-degree terms in one step instead of interpolating each variable separately. Assume for example that one has to compute a high-order accurate estimation of p/r^α at the $r = 0$ cell interface from cell-centered point-wise values, α being a positive exponent. This can be achieved using the following interpolation formula:

$$\left(\frac{p}{r^\alpha}\right)_{r=0} \approx \sum_{k=1}^{s} d_{0,k}^{N_{\text{eff}}} \cdot \left[\left(\frac{p}{r^\alpha}\right)_{r=\frac{(2k-1)\Delta r}{2}} + \left(\frac{p}{r^\alpha}\right)_{r=\frac{(1-2k)\Delta r}{2}}\right]$$

with notations defined page 62. Such a manipulation may look trivial but its application becomes challenging with automatic Cauchy-Kovalevskaya procedures. Practically, we apply it by post-processing the automatically generated flux terms using a Python script.

3. The last modification concerns the u^* flux that has to be computed so that the position of Eulerian cell interfaces can be updated: one has to set u^* to zero along the $r = 0$ axis. Indeed, since we are considering axisymmetric geometries, the $r = 0$ boundary can be treated as if it was a reflective boundary along which the normal component of the velocity has to be zero.

Remark 26. Note that it is also possible to simply degrade Lagrangian fluxes at second-order along the $r = 0$ axis in order to avoid pathological cases since second-order fluxes do not involve $1/r$ terms provided that the first above-described modification has been applied. Though this approach cannot be considered as a satisfying solution, numerical experiments have shown that it does not affect the order of convergence in practice.

Approximate source term computation

We present in this section how the $\mathbf{S}_i^*$ source term can be evaluated high-order accurately. This can be achieved using a truncated Taylor expansion in both space and time around (R_i, t^n) in the definition of $\mathbf{S}_i^*$, leading to the following $\mathbf{S}_i^{*,N}$ approximate non-conservative term:

$$\begin{aligned}
\mathbf{S}_i^{*,N} &= \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \left[\frac{1}{\Delta R}\int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} \left[\sum_{k+l<N} \frac{(t-t^n)^k}{k!}\cdot\frac{(R-R_i)^l}{l!}\cdot\left(\frac{\partial^{k+l}\mathbf{S}(\mathbf{U})}{\partial t^k \partial R^l}\right)(R_i,t^n)\right] dR\right] dt,\\
&= \sum_{k+l<N}\left[\frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \frac{(t-t^n)^k}{k!}\,dt\right]\cdot\left[\frac{1}{\Delta R}\int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} \frac{(R-R_i)^l}{l!}\right]\cdot\left(\frac{\partial^{k+l}\mathbf{S}(\mathbf{U})}{\partial t^k \partial R^l}\right)(R_i,t^n),\\
&= \sum_{k+l<N}\frac{\Delta t^k}{(k+1)!}\cdot\underbrace{\left[\frac{(R_{i+\frac{1}{2}}-R_i)^{l+1}}{\Delta R(l+1)!} - \frac{(R_{i-\frac{1}{2}}-R_i)^{l+1}}{\Delta R(l+1)!}\right]}_{=0 \text{ if } l \text{ is odd}}\cdot\left(\frac{\partial^{k+l}\mathbf{S}(\mathbf{U})}{\partial t^k \partial R^l}\right)(R_i,t^n).
\end{aligned}$$

The approximate non-conservative term thus writes:

$$\mathbf{S}_i^{*,N} = \sum_{k+2l<N}\frac{\Delta t^k}{(k+1)!}\cdot\frac{\Delta R^{2l}}{2^{2l+1}(2l+1)!}\cdot\frac{\partial^{2l}\mathbf{S}^k}{\partial R^{2l}}(R_i,t^n) \quad \text{where} \quad \mathbf{S}^k = \frac{\partial^k \mathbf{S}(\mathbf{U})}{\partial t^k}. \tag{6.8}$$

We propose the following strategy for computing $\mathbf{S}_i^{*,N}$.

- Replace the time-derivatives involved in (6.8) by applying the Cauchy-Kovalevskaya procedure in order to form expressions of $\mathbf{S}^k$ that can be computed using quantities known at time t^n.

- For the sake of robustness, estimate high-order accurate approximations of the resulting $\mathbf{S}^k$ terms at cell interfaces using the finite difference formula (4.16).
- Evaluate the high-order accurate cell-centered non-conservative term (6.8) by interpolating the cell interface representation of $\mathbf{S}^k$. This step can be achieved using the following formula:

$$\frac{\partial^{2l}\mathbf{S}^k}{\partial R^{2l}}(R_i,t^n)=\frac{1}{\Delta r^{2l}}\sum_{m=1}^{\lfloor N_{\text{eff}}/2\rfloor} d_{2l,m}^{N_{\text{eff}}}\left(\mathbf{S}^k_{i-m-\frac{1}{2}}+\mathbf{S}^k_{i+m+\frac{1}{2}}\right),$$

the $d_{2l,m}^{N_{\text{eff}}}$ coefficients being given in Table 4.2. N_{eff} denotes here the effectively required order of accuracy for each term appearing in $\mathbf{S}^{*,N}$ depending on its position in the Taylor expansion. It is given by:

$$N_{\text{eff}}=N-k-2l+\left[(N-k-2l)\bmod 2\right].$$

We illustrate the above-described algorithm with the example of the second-order accurate $\mathbf{S}^{*,2}$ term in the case of hydrodynamics where the only non-zero component of $\mathbf{S}$ appears in the momentum equation and writes (Jp). The expression of $(Jp)_i^{*,2}$ is given by:

$$\begin{aligned}(Jp)_i^{*,2}&=(Jp)(R_i,t^n)+\frac{\Delta t}{2}\left(\frac{\partial(Jp)}{\partial t}\right)(R_i,t^n)\\&=p(R_i,t^n)+\frac{\Delta t}{2}\left(\frac{\partial Jp}{\partial t}\right)(R_i,t^n)\text{ since }J(t=t^n)=1.\end{aligned}$$

At this stage, we first have to determine an expression of the time-derivative of (Jp). After a few simplifications, it writes:

$$\partial_t(Jp)=-\frac{(\rho c)^2}{R\rho_0}\partial_R(ru_r)+p\partial_R u_r.$$

We now have to evaluate $\partial_t(Jp)$ at cell interfaces. This can be achieved using the following second-order accurate approximation:

$$(\partial_t(Jp))^*_{i+\frac{1}{2}}=-\frac{1}{2}\left(\frac{(\rho c)_i^2}{R_i(\rho_0)_i}+\frac{(\rho c)_{i+1}^2}{R_{i+1}(\rho_0)_{i+1}}\right)\cdot\left(\frac{r_{i+1}(u_r)_{i+1}-r_i(u_r)_i}{\Delta R}\right)+\left(\frac{p_i+p_{i+1}}{2}\right)\cdot\left(\frac{(u_r)_{i+1}-(u_r)_i}{\Delta R}\right).$$

Note that we have dropped the n superscript in the previous formula for the sake of simplicity, all quantities being evaluated at time t^n. The approximate non-conservative term is finally given by:

$$(Jp)_i^{*,2}=p(R_i,t^n)+\frac{\Delta t}{4}\left((\partial_t(Jp))^*_{i+\frac{1}{2}}+(\partial_t(Jp))^*_{i-\frac{1}{2}}\right).$$

Remark 27. Several approximate time-derivatives appear in both Lagrangian fluxes and non-conservative terms. These are computed only once in order to avoid unnecessary calculations.

6.1.2 Remap step

Once the Lagrangian scheme has been applied, we have at our disposal a set of Lagrangian conservative variables $\overline{(\rho_0\phi)}$ for $\phi\in\{1,\mathbf{u},\tau B_z,e\}$ at time t^{n+1} (we recall that the case of B_θ is discussed in section 6.1.3). These correspond to Eulerian conservative variables on the non-uniform $\{r^{n+1}_{i+\frac{1}{2}}\}$ grid given by:

$$r^{n+1}_{i+\frac{1}{2}}=r_{i+\frac{1}{2}}+(u_r)^*_{i+\frac{1}{2}}\Delta t\quad\text{since }\partial_t r=u_r.$$

Indeed, since the Lagrangian variable change is given by $R\rho_0 dR = r\rho dr$, Eulerian and Lagrangian conservative variables satisfy the following exact formula:

$$R_i \Delta R \, \overline{(\rho_0\phi)}_i^{n+1} = \int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} R(\rho_0\phi)(R, t^{n+1}) \, dR = \int_{r_{i-\frac{1}{2}}^{n+1}}^{r_{i+\frac{1}{2}}^{n+1}} r(\rho\phi)(r, t^{n+1}) \, dr. \tag{6.9}$$

The remap step aims at projecting these values on the initial regular Cartesian grid. It is based on the following integral splitting

$$\begin{aligned} r_i \Delta r \, \overline{(\rho\phi)}_i^{n+1} &= \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} r(\rho\phi)(r, t^{n+1}) \, dr \\ &= \underbrace{\int_{r_{i-\frac{1}{2}}}^{r_{i-\frac{1}{2}}^{n+1}} r(\rho\phi)(r, t^{n+1}) \, dr}_{\mathcal{P}_{i-\frac{1}{2}}} + \underbrace{\int_{r_{i-\frac{1}{2}}^{n+1}}^{r_{i+\frac{1}{2}}^{n+1}} r(\rho\phi)(r, t^{n+1}) \, dr}_{\mathcal{P}_i} + \underbrace{\int_{r_{i+\frac{1}{2}}^{n+1}}^{r_{i+\frac{1}{2}}} r(\rho\phi)(r, t^{n+1}) \, dr}_{\mathcal{P}_{i+\frac{1}{2}}}. \end{aligned}$$

We recall that $\mathcal{P}_i$ corresponds to the cell average of $(\rho\phi)$ on the non-uniform grid $\{r_{i+\frac{1}{2}}^{n+1}\}$ which is given by (6.9) and that the $\mathcal{P}_{i\mp\frac{1}{2}}$ terms in the right hand side can be expressed from the so-called remap fluxes $(\rho\phi)^*_{i\mp\frac{1}{2}}$:

$$\mathcal{P}_{i\mp\frac{1}{2}} = \left(r_{i\mp\frac{1}{2}}^{n+1} - r_{i\mp\frac{1}{2}} \right) (\rho\phi)^*_{i\mp\frac{1}{2}} \quad \text{with} \quad (\rho\phi)^*_{i\mp\frac{1}{2}} = \frac{1}{r_{i\mp\frac{1}{2}}^{n+1} - r_{i\mp\frac{1}{2}}} \int_{r_{i\mp\frac{1}{2}}}^{r_{i\mp\frac{1}{2}}^{n+1}} r(\rho\phi)(r, t^{n+1}) \, dr.$$

Since $r_{i+\frac{1}{2}}^{n+1} - r_{i+\frac{1}{2}} = u^*_{i+\frac{1}{2}} \Delta t$, the remap scheme finally writes:

$$\overline{(\rho\phi)}_i^{n+1} = \overline{(\rho_0\phi)}_i^{n+1} - \frac{\Delta t}{r_i \Delta r} \left(u^*_{i+\frac{1}{2}} (\rho\phi)^*_{i+\frac{1}{2}} - u^*_{i-\frac{1}{2}} (\rho\phi)^*_{i-\frac{1}{2}} \right).$$

High-order accurate remap fluxes can be computed as explained in section 4.1.3. The only difference with the planar case comes from the values of the H_i^ϕ primitive (4.22). Indeed, in the axisymmetric case, these are given by:

$$\begin{aligned} H_i^\phi(x_{i-s+\frac{1}{2}}^{n+1}) &= 0, \\ H_i^\phi(x_{i-s+\frac{3}{2}}^{n+1}) &= r_{i-s+1} \Delta r \overline{(\rho_0\phi)}_{i-s+1}^{n+1}, \\ H_i^\phi(x_{i-s+\frac{5}{2}}^{n+1}) &= \Delta r \left(r_{i-s+1} \overline{(\rho_0\phi)}_{i-s+1}^{n+1} + r_{i-s+2} \overline{(\rho_0\phi)}_{i-s+2}^{n+1} \right), \\ &\vdots \\ H_i^\phi(x_{i+s-\frac{1}{2}}^{n+1}) &= \Delta r \left(r_{i-s+1} \overline{(\rho_0\phi)}_{i-s+1}^{n+1} + \ldots + r_{i+s-1} \overline{(\rho_0\phi)}_{i+s-1}^{n+1} \right). \end{aligned}$$

with notations defined in section 4.1.3.

Stability condition

In axisymmetric geometry, the stability condition for the whole Lagrange-remap schemes slightly differs from the planar case. It is indeed given by:

$$\Delta t \leq \max \frac{r_i \Delta r}{r_{i+\frac{1}{2}} (|u_i| + (c_f)_i)}.$$

where c_f denotes the fast magnetosonic wave speed[1].

6.1.3 Solving the B_θ equation

We now focus on the particular treatment of the B_θ equation which writes:

$$\partial_t B_\theta + \partial_r(u_r B_\theta - u_\theta B_r) = 0. \tag{6.10}$$

We have seen previously (see remark 25) that it is similar to planar equations studied in previous chapters. It is thus reasonable to solve it using the planar Lagrange-remap procedure described in sections 4.1.2 and 4.1.3. Introducing the $(r,t) \to (R,t)$ Lagrangian variable change, equation (6.10) rewrites:

$$\partial_t(JB_\theta) + \partial_R(-u_\theta B_r) = 0. \tag{6.11}$$

We integrate equation (6.11) over $[R_{i-\frac{1}{2}}; R_{i+\frac{1}{2}}] \times [t^n; t^{n+1}]$ and thus obtain the following finite volume scheme:

$$\overline{(JB_\theta)}_i^{n+1} = \overline{(JB_\theta)}_i^{n} - \frac{\Delta t}{\Delta r}\left[(-u_\theta B_r)^*_{i+\frac{1}{2}} - (-u_\theta B_r)^*_{i-\frac{1}{2}}\right] \tag{6.12}$$

where the $\overline{(JB_\theta)}_{i,j}^{n}$ denotes the *planar* cell average:

$$\overline{(JB_\theta)}_{i,j}^{n} = \frac{1}{\Delta R}\int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} (JB_\theta)(R,t^n)\, dR.$$

Note that computing cell-centered point-wise values of B_θ has to be performed using the *planar* formula (4.15):

$$\begin{aligned}(B_\theta)_{i,j}^n &= (JB_\theta)_{i,j}^n \text{ since } J(t=t^n)=1 \\ &= c_0^N \cdot \overline{(JB_\theta)}_i^n + \sum_{k=1}^{\lfloor N/2 \rfloor} c_k^N \cdot \left(\overline{(JB_\theta)}_{i+k}^n + \overline{(JB_\theta)}_{i-k}^n\right).\end{aligned}$$

At this stage, Lagrangian fluxes have to be computed using the procedure described in section 6.1.1 in order to update cell averages of (JB_θ) according to the finite volume scheme (6.12). The resulting values now have to be projected on the initial grid. To that end, we consider the following integral splitting:

$$\Delta r\overline{(B_\theta)}_i^{n+1} = \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} B_\theta(r,t^{n+1})\, dr = \int_{r_{i-\frac{1}{2}}}^{r^{n+1}_{i-\frac{1}{2}}} B_\theta(r,t^{n+1})\, dr + \int_{r^{n+1}_{i-\frac{1}{2}}}^{r^{n+1}_{i+\frac{1}{2}}} B_\theta(r,t^{n+1})\, dr + \int_{r^{n+1}_{i+\frac{1}{2}}}^{r_{i+\frac{1}{2}}} B_\theta(r,t^{n+1})\, dr.$$

The second right hand side term is evaluated using an exact formula which is similar to (6.9). It writes:

$$\int_{r^{n+1}_{i-\frac{1}{2}}}^{r^{n+1}_{i+\frac{1}{2}}} B_\theta(r,t^{n+1})\, dr = \int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} JB_\theta(R,t^{n+1})\, dR = \Delta R\overline{(JB_\theta)}_i^{n+1},$$

since the $(r,t) \to (R,t)$ variable change satisfies $dr = JdR$. The remaining terms are remap fluxes:

$$(B_\theta)^*_{i+\frac{1}{2}} = \frac{1}{r^{n+1}_{i+\frac{1}{2}} - r_{i+\frac{1}{2}}}\int_{r_{i+\frac{1}{2}}}^{r^{n+1}_{i+\frac{1}{2}}} B_\theta(r,t^{n+1})\, dr,$$

which have to be computed using the *planar* procedure described in section 4.1.3. The remap scheme for B_θ is finally given by:

$$\overline{(B_\theta)}_i^{n+1} = \overline{(JB_\theta)}_i^{n+1} - \frac{\Delta t}{\Delta r}\left(u^*_{i+\frac{1}{2}}(B_\theta)^*_{i+\frac{1}{2}} - u^*_{i-\frac{1}{2}}(B_\theta)^*_{i-\frac{1}{2}}\right). \tag{6.13}$$

[1] In the case of hydrodynamics, c_f is the sound speed.

6.2 Two-dimensional extension

Now that the GoHy high-order Lagrange-remap schemes have been described in the one-dimensional case, we present their extension to the two-dimensional axisymmetric ideal MHD system which is given by:

$$\partial_t \begin{pmatrix} r\rho \\ r\rho u_r \\ r\rho u_\theta \\ r\rho u_z \\ rB_r \\ rB_z \\ r\rho e \end{pmatrix} + \partial_r \begin{pmatrix} r\rho u_r \\ r\rho u_r^2 + rP - r\frac{B_r^2}{\mu} \\ r\rho u_r u_\theta - r\frac{B_r B_\theta}{\mu} \\ r\rho u_r u_z - r\frac{B_r B_z}{\mu} \\ 0 \\ ru_r B_z - ru_z B_r \\ r(\rho e + P)u_r - r\frac{\mathbf{u}\cdot\mathbf{B}}{\mu} B_r \end{pmatrix} + r\partial_z \begin{pmatrix} \rho u_z \\ \rho u_z u_r - \frac{B_z B_r}{\mu} \\ \rho u_z u_\theta - \frac{B_z B_\theta}{\mu} \\ \rho u_z^2 + P - \frac{B_z^2}{\mu} \\ u_z B_r - u_r B_z \\ 0 \\ (\rho e + P)u_z - \frac{\mathbf{u}\cdot\mathbf{B}}{\mu} B_z \end{pmatrix} = \begin{pmatrix} 0 \\ \rho u_\theta^2 + P - \frac{B_\theta^2}{\mu} \\ -\rho u_r u_\theta + \frac{B_r B_\theta}{\mu} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\partial_t B_\theta + \partial_r(u_r B_\theta - u_\theta B_r) + \partial_z(u_z B_\theta - u_\theta B_z) = 0.$$

We rewrite it in the following general form:

$$\begin{cases} \partial_t \mathbf{U} + \partial_r \mathbf{F}_r + r\partial_z \mathbf{F}_z = \mathbf{S}, \\ \partial_t B_\theta + \partial_r(u_r B_\theta - u_\theta B_r) + \partial_z(u_z B_\theta - u_\theta B_z) = 0. \end{cases} \tag{6.14}$$

Once again, the two-dimensional extension basically relies on high-order accurate dimensional splitting techniques which here consist in solving successively:

$$\begin{cases} \partial_t \mathbf{U} + \partial_r \mathbf{F}_r = \mathbf{S} \\ \partial_t B_\theta + \partial_r(u_r B_\theta - u_\theta B_r) = 0 \end{cases} \quad \text{and} \quad \begin{cases} \partial_t \mathbf{U} + r\partial_z \mathbf{F}_z = 0 \\ \partial_t B_\theta + \partial_z(u_z B_\theta - u_\theta B_z) = 0 \end{cases}$$

with appropriately weighted time steps. As in the planar case, this strategy has to be combined with a reconstruction step and the resolution of an additional equation in order to advect divergence errors. Both the fluxes reconstruction and hyperbolic divergence cleaning slightly differ from the planar case. We present their axisymmetric version in this section.

6.2.1 Fluxes reconstruction

First of all, we recall that the B_θ equation is discretized in a planar framework: the planar reconstruction step presented in section 4.2.2 can therefore be applied without any modification to the Lagrangian and remap schemes (6.12) and (6.13). We now focus on the reconstruction step in the axisymmetric case for the remaining variables. We recall that the Lagrangian and remap fluxes that operate on two-dimensional cell averages actually are cell averages of the one-dimensional fluxes. More precisely, one has to compute:

$$\mathbf{F}_{i+\frac{1}{2},j} = \frac{1}{r_j \Delta r} \int_{r_{j-\frac{1}{2}}}^{r_{j+\frac{1}{2}}} r\mathbf{F}_{i+\frac{1}{2}}(r)\, dr \qquad \text{when solving system (6.14) along the } z \text{ direction,}$$

$$\mathbf{F}_{i,j+\frac{1}{2}} = \frac{1}{\Delta z} \int_{z_{i-\frac{1}{2}}}^{z_{i+\frac{1}{2}}} \mathbf{F}_{j+\frac{1}{2}}(z)\, dz \qquad \text{when solving system (6.14) along the } r \text{ direction,}$$

where $\mathbf{F}$ denotes either the Lagrangian or the remap flux. The reconstruction formula given in the planar case (4.39b) can obviously be used to compute the $\mathbf{F}_{i,j+\frac{1}{2}}$ fluxes along the r direction. On the contrary, an alternative formula is required for determining the z-wise $\mathbf{F}_{i+\frac{1}{2},j}$ fluxes. We propose the following reconstruction function:

$$\begin{aligned}\Pi^{-1}\left(\mathbf{F}_{i+\frac{1}{2}}\right) &= q_0\mathbf{F}_{i+\frac{1}{2}}(r_j) + \sum_{k=1}^{\lfloor N/2\rfloor} q_k\left(\frac{r_{j+k}}{r_j}\mathbf{F}_{i+\frac{1}{2}}(r_{j+k}) + \frac{r_{j-k}}{r_j}\mathbf{F}_{i+\frac{1}{2}}(r_{j-k})\right)\\ &= \mathbf{F}_{i+\frac{1}{2},j} + \mathcal{O}\left(\Delta r^N\right)\end{aligned}$$

which allows to build high-order accurate two-dimensional fluxes, the q coefficients being given in Table 4.2.2. This formula has been obtained by applying the planar reconstruction operator (4.39b) to $(r\mathbf{F})$ and then dividing the result by r_j.

Remark 28. Note that the reconstruction procedure for the r direction not only has to be applied to Lagrangian and remap fluxes but also to the approximate source terms.

6.2.2 Hyperbolic divergence cleaning in the axisymmetric case

Following the method proposed by Dedner *et al.* [29], as explained in section 5.2.4, hyperbolic divergence cleaning techniques consist in adding an unknown denoted by ψ which is coupled to the ideal MHD system through:

$$\partial_t\mathbf{B} + \nabla\cdot(\mathbf{u}\otimes\mathbf{B} - \mathbf{B}\otimes\mathbf{u}) + \nabla\psi = 0, \tag{6.15}$$

$$\partial_t\psi + c_h^2\nabla\cdot\mathbf{B} = 0. \tag{6.16}$$

We have seen in section 5.2.4 that ψ satisfies the same evolution equation than $\nabla\cdot\mathbf{B}$ which actually is the wave equation:

$$\partial_{tt}\psi - c_h^2\Delta\psi = 0.$$

As a matter of fact, the additional equation on ψ allows to advect divergence errors to the boundaries with the finite speed $c_h > 0$. In this study, c_h is defined from the fast magnetosonic wave speed at $t = 0$ (see remark 22 page 117). In the axisymmetric case, equations (6.15) and (6.16) rewrite:

$$\partial_t(rB_r) + \partial_r(r\psi) + r\partial_z\left(u_zB_r - u_rB_z + \psi\right) = \psi, \tag{6.17a}$$

$$\partial_t(rB_z) + \partial_r\left(ru_rB_z - ru_zB_r + r\psi\right) + r\partial_z\psi = 0, \tag{6.17b}$$

$$\partial_t(r\psi) + c_h^2\partial_r(rB_r) + c_h^2r\partial_z(B_z) = 0. \tag{6.17c}$$

Equations (6.17a)-(6.17c) can be solved along the z-direction using the planar MHD-GoHy schemes that have been described in chapter 5. We thus focus in the sequel on the resolution of (6.17a)-(6.17c) along the r-direction. The corresponding r-wise Lagrangian schemes write:

$$\begin{aligned}\overline{(\rho_0\tau B_r)}_{i,j}^{n+1} &= \overline{(\rho_0\tau B_r)}_{i,j}^{n} - \frac{\Delta t}{R_i\Delta r}\left[(r\psi - ru_rB_r)^*_{i+\frac{1}{2},j} - (r\psi - ru_rB_r)^*_{i-\frac{1}{2},j}\right] + \frac{\Delta t}{R_i}(\psi)^*_{i,j},\\ \overline{(\rho_0\tau B_z)}_{i,j}^{n+1} &= \overline{(\rho_0\tau B_z)}_{i,j}^{n} - \frac{\Delta t}{R_i\Delta r}\left[(-ru_zB_r)^*_{i+\frac{1}{2},j} - (-ru_zB_r)^*_{i-\frac{1}{2},j}\right],\\ \overline{(\rho_0\tau\psi)}_{i,j}^{n+1} &= \overline{(\rho_0\tau\psi)}_{i,j}^{n} - \frac{\Delta t}{R_i\Delta r}\left[\left(c_h^2rB_r - ru_r\psi\right)^*_{i+\frac{1}{2},j} - \left(c_h^2rB_r - ru_r\psi\right)^*_{i-\frac{1}{2},j}\right].\end{aligned}$$

Once updated values of the Lagrangian conservative variables have been computed, the following remap schemes has to be applied for $\phi \in \{B_r,\ B_z,\ \psi\}$:

$$\overline{\phi}_{i,j}^{n+1} = \overline{(\rho_0 \tau \phi)}_{i,j}^{n+1} - \frac{\Delta t}{r_i \Delta r}\left[(u_r)^*_{i+\frac{1}{2},j}(\phi)^*_{i+\frac{1}{2},j} - (u_r)^*_{i-\frac{1}{2},j}(\phi)^*_{i-\frac{1}{2},j}\right].$$

Remark 29. The B_r equation could actually be treated in a conservative way using the following formulation:

$$\partial_t B_r + \partial_r \psi + \partial_z \left(u_z B_r - u_r B_z + \psi\right) = 0. \tag{6.18}$$

In this case, the structure of the evolution equation (6.18) would be similar to the one for the B_θ equation that we studied in section 6.1.3. One could therefore imagine to solve both B_r and B_θ equations in the same framework, *i.e.* using *planar* GoHy schemes. Nevertheless, we have not tested this approach yet and thus restrict to the non-conservative form of the B_r equation given previously.

The above-described hyperbolic divergence cleaning technique has been used for performing the numerical computations described in section 6.4. Experiments have shown that, as in the planar case, it increases the restitution time of the whole Lagrange-remap scheme of about 10% whatever the order of accuracy, which seems reasonable in terms of computational cost.

6.3 Viscosity fluxes in the axisymmetric case

As in the case of hydrodynamics, the high-order Lagrange-remap schemes that we developed for the axisymmetric ideal MHD equations are subject to Gibbs oscillations on discontinuities. To improve robustness, we therefore again add hyperviscosity fluxes to the Lagrangian step. We recall that we thus consider the following resistive MHD system:

$$\partial_t \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ \mathbf{B} \\ \rho e \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\otimes\mathbf{u} + P\underline{\mathbf{I}} - \frac{\mathbf{B}\otimes\mathbf{B}}{\mu} \\ \mathbf{u}\otimes\mathbf{B} - \mathbf{B}\otimes\mathbf{u} \\ (\rho e + P)\mathbf{u} - \frac{\mathbf{u}\cdot\mathbf{B}}{\mu}\mathbf{B} \end{pmatrix} = \begin{pmatrix} 0 \\ \nabla\cdot\underline{\boldsymbol{\tau}} \\ -\eta\nabla\times(\nabla\times\mathbf{B}) \\ \nabla\cdot\left(\underline{\boldsymbol{\tau}}\mathbf{u} + \frac{\eta}{\mu}\mathbf{B}\wedge(\nabla\times\mathbf{B}) + \kappa\nabla T\right) \end{pmatrix} \tag{6.19}$$

and replace the physical constants β, ν, κ and η with artificial ones (marked with stars) so that consistency with the initial ideal MHD system (5.1) is preserved. In the (z, r, θ) system of coordinates, the symmetric viscous stress tensor $\underline{\boldsymbol{\tau}}$ writes:

$$\begin{aligned}\underline{\boldsymbol{\tau}} &= \nu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) + \left(\beta - \frac{2}{3}\nu\right)(\nabla\cdot\mathbf{u})\underline{\mathbf{I}}, \\ &= \begin{bmatrix} a\,\partial_z u_z + b\left(\partial_r u_r + \frac{u_r}{r}\right) & \nu(\partial_r u_z + \partial_z u_r) & \nu\partial_z u_\theta \\ & a\,\partial_r u_r + b\left(\frac{u_r}{r} + \partial_z u_z\right) & \nu\left(\partial_r u_\theta - \frac{u_\theta}{r}\right) \\ & & a\frac{u_r}{r} + b(\partial_r u_r + \partial_z u_z) \end{bmatrix},\end{aligned}$$

where $a = \left(\beta + \frac{4}{3}\nu\right)$ and $b = \left(\beta - \frac{2}{3}\nu\right)$. Discrete expressions of the $\underline{\boldsymbol{\tau}}$ tensor's components in the two-dimensional axisymmetric case are given by:

$$
\begin{aligned}
\underline{\boldsymbol{\tau}}_{i,j}^{zz} &= \left(\beta_{i,j}^{\star} + \frac{4}{3}\nu_{i,j}^{\star}\right)\left(\frac{(u_z)_{i+1,j} - (u_z)_{i-1,j}}{2\Delta z}\right) + \left(\beta_{i,j}^{\star} - \frac{2}{3}\nu_{i,j}^{\star}\right)\left(\frac{(u_r)_{i,j+1} - (u_r)_{i,j+1}}{2\Delta r} + \frac{(u_r)_{i,j}}{r_j}\right), \\
\underline{\boldsymbol{\tau}}_{i,j}^{zr} &= \underline{\boldsymbol{\tau}}_{i,j}^{rz} = \nu_{i,j}^{\star}\left(\frac{(u_r)_{i+1,j} - (u_r)_{i-1,j}}{2\Delta z} + \frac{(u_z)_{i,j+1} - (u_z)_{i,j-1}}{2\Delta r}\right), \\
\underline{\boldsymbol{\tau}}_{i,j}^{z\theta} &= \nu_{i,j}^{\star}\left(\frac{(u_\theta)_{i+1,j} - (u_\theta)_{i-1,j}}{2\Delta z}\right), \\
\underline{\boldsymbol{\tau}}_{i,j}^{rr} &= \left(\beta_{i,j}^{\star} + \frac{4}{3}\nu_{i,j}^{\star}\right)\left(\frac{(u_r)_{i,j+1} - (u_r)_{i,j-1}}{2\Delta r}\right) + \left(\beta_{i,j}^{\star} - \frac{2}{3}\nu_{i,j}^{\star}\right)\left(\frac{(u_z)_{i+1,j} - (u_z)_{i-1,j}}{2\Delta z} + \frac{(u_r)_{i,j}}{r_j}\right), \\
\underline{\boldsymbol{\tau}}_{i,j}^{r\theta} &= \nu_{i,j}^{\star}\left(\frac{(u_\theta)_{i,j+1} - (u_\theta)_{i,j-1}}{2\Delta r} - \frac{(u_\theta)_{i,j}}{r_j}\right), \\
\underline{\boldsymbol{\tau}}_{i,j}^{\theta\theta} &= \left(\beta_{i,j}^{\star} + \frac{4}{3}\nu_{i,j}^{\star}\right)\frac{(u_r)_{i,j}}{r_j} + \left(\beta_{i,j}^{\star} - \frac{2}{3}\nu_{i,j}^{\star}\right)\left(\frac{(u_r)_{i,j+1} - (u_r)_{i,j-1}}{2\Delta r} + \frac{(u_z)_{i+1,j} - (u_z)_{i-1,j}}{2\Delta z}\right).
\end{aligned}
$$

In this context, the $\nabla \times (\nabla \times \mathbf{B})$ and the $\mathbf{B} \wedge (\nabla \times \mathbf{B})$ terms respectively write:

$$
\nabla \times (\nabla \times \mathbf{B}) = \begin{pmatrix} \frac{1}{r}\partial_r\left(r\Lambda\right) \\ -\partial_z\Lambda \\ -\partial_r\left(\partial_r B_\theta + \frac{B_\theta}{r}\right) - \partial_z(\partial_z B_\theta) \end{pmatrix},
$$

and:

$$
\mathbf{B} \wedge (\nabla \times \mathbf{B}) = \begin{pmatrix} B_\theta \partial_z B_\theta + B_r \Lambda \\ B_\theta\left(\frac{B_\theta}{r} + \partial_r B_\theta\right) - B_z\Lambda \\ -B_z\partial_z B_\theta - B_r\left(\frac{B_\theta}{r} + \partial_r B_\theta\right) \end{pmatrix},
$$

with:

$$
\Lambda = (\nabla \times \mathbf{B}) \cdot \mathbf{e}_\theta = \partial_z B_r - \partial_r B_z.
$$

Considering the unknown vector $\mathbf{U} = (\rho_0 u_z,\ \rho_0 u_r,\ \rho_0 u_\theta,\ \rho_0 \tau B_z\ \rho_0 \tau B_r,\ \rho_0 \tau B_\theta,\ \rho_0 e)$, we propose the following discrete cell-centered artificial viscosity terms:

$$
\mathbf{V}_{i,j}^{z,*} = \begin{pmatrix} \underline{\boldsymbol{\tau}}_{i,j}^{zz} \\ \underline{\boldsymbol{\tau}}_{i,j}^{zr} \\ \underline{\boldsymbol{\tau}}_{i,j}^{z\theta} \\ 0 \\ \eta_{i,j}^{\star}\Lambda_{i,j} \\ \eta_{i,j}^{\star}\left(\frac{(B_\theta)_{i+1,j} - (B_\theta)_{i-1,j}}{2\Delta z}\right) \\ (\underline{\boldsymbol{\tau}}^{zz}u_z)_{i,j} + (\underline{\boldsymbol{\tau}}^{zr}u_r)_{i,j} + (\underline{\boldsymbol{\tau}}^{z\theta}u_\theta)_{i,j} + \frac{1}{\mu}\eta_{i,j}^{\star}\left(\mathbf{B} \wedge (\nabla \times \mathbf{B})\right)_{i,j}^{z} + \kappa_{i,j}^{\star}\left(\frac{T_{i+1,j} - T_{i-1,j}}{2\Delta z}\right) \end{pmatrix},
$$

$$\mathbf{V}_{i,j}^{r,*} = \begin{pmatrix} r_j \underline{\boldsymbol{\tau}}_{i,j}^{rz} \\ r_j \underline{\boldsymbol{\tau}}_{i,j}^{rr} \\ r_j \underline{\boldsymbol{\tau}}_{i,j}^{r\theta} \\ -r_j \eta_{i,j}^{\star} \Lambda_{i,j} \\ 0 \\ \eta_{i,j}^{\star} \left(\frac{(B_\theta)_{i,j+1} - (B_\theta)_{i,j-1}}{2\Delta r} + \frac{(B_\theta)_{i,j}}{r_j} \right) \\ r_j (\underline{\boldsymbol{\tau}}^{rz} u_z)_{i,j} + r_j (\underline{\boldsymbol{\tau}}^{rr} u_r)_{i,j} + r_j (\underline{\boldsymbol{\tau}}^{r\theta} u_\theta)_{i,j} + \frac{r_j}{\mu} \eta_{i,j}^{\star} \left(\mathbf{B} \wedge (\nabla \times \mathbf{B}) \right)_{i,j}^{r} + r_j \kappa_{i,j}^{\star} \left(\frac{T_{i,j+1} - T_{i,j-1}}{2\Delta r} \right) \end{pmatrix},$$

with the following notations:

$$\begin{aligned} \Lambda_{i,j} &= \frac{(B_r)_{i+1,j} - (B_r)_{i-1,j}}{2\Delta z} - \frac{(B_z)_{i,j+1} - (B_z)_{i,j-1}}{2\Delta r}, \\ \left(\mathbf{B} \wedge (\nabla \times \mathbf{B}) \right)_{i,j}^{z} &= (B_\theta)_{i,j} \left(\frac{(B_\theta)_{i+1,j} - (B_\theta)_{i-1,j}}{2\Delta z} \right) + (B_r)_{i,j} \Lambda_{i,j}, \\ \left(\mathbf{B} \wedge (\nabla \times \mathbf{B}) \right)_{i,j}^{r} &= (B_\theta)_{i,j} \left(\frac{(B_\theta)_{i,j+1} - (B_\theta)_{i,j-1}}{2\Delta r} + \frac{(B_\theta)_{i,j}}{r_j} \right) - (B_z)_{i,j} \Lambda_{i,j}. \end{aligned}$$

These allow to build the following artificial viscosity fluxes along z and r directions:

$$\mathbf{V}_{i+\frac{1}{2},j}^{*} = \frac{1}{2} \left(\mathbf{V}_{i,j}^{z,*} + \mathbf{V}_{i+1,j}^{z,*} \right) \quad \text{and} \quad \mathbf{V}_{i,j+\frac{1}{2}}^{*} = \frac{1}{2} \left(\mathbf{V}_{i,j}^{r,*} + \mathbf{V}_{i,j+1}^{r,*} \right).$$

Moreover, a viscous source term $\mathbf{VS}^{\star}$ has to be added to $\mathbf{S}^{\star}$ in the axisymmetric case when performing the resolution along the r direction. It writes:

$$\mathbf{VS}_{i,j}^{\star} = \left(0,\ -\underline{\boldsymbol{\tau}}_{i,j}^{\theta\theta},\ \underline{\boldsymbol{\tau}}_{i,j}^{r\theta},\ 0,\ 0,\ 0,\ 0 \right)^{t}.$$

Note that there is no need to design new artificial constants: these proposed in sections 4.3 and 5.3 are suitable for axisymmetric geometries.

6.4 Numerical results in axisymmetric geometry

To conclude the description of MHD-GoHy schemes in axisymmetric geometry, we provide in this section numerical results on various test problems taken from the related literature. More precisely, we first compute the experimental order of convergence on smooth test problems and then present results obtained for several non-smooth test problems to illustrate the robustness of the schemes we developed.

6.4.1 Experimental order of convergence

We first focus on smooth test problems and carry out a convergence analysis. To that end, we consider Kidder's isentropic compression problem [53] in spherically symmetric geometry and a two-dimensional analytical solution proposed by Picard [71].

Two-dimensional axisymmetric hydrodynamics: Kidder's isentropic compression problem in spherically symmetric geometry

We first consider Kidder's isentropic compression problem in spherically symmetric geometry which provides an analytical solution to the two-dimensional axisymmetric hydrodynamics equations. Since this benchmark has already been studied in the planar case (see section 4.5.1), we here simply recall the density, pressure and velocity profiles. Defining $R = \sqrt{r^2 + z^2}$, $\phi = \text{atan}(z/r)$ and:

$$\rho_0(R) = \left(\frac{r_2^2 - R^2}{r_2^2 - r_1^2} \rho_1^{\gamma-1} + \frac{R^2 - r_1^2}{r_2^2 - r_1^2} \rho_2^{\gamma-1} \right)^{\frac{1}{\gamma-1}},$$

the exact solution is given by:

$$\begin{aligned}
\rho(R,t) &= \rho_0 \left(\frac{R}{h(t)} \right) \cdot h(t)^{\frac{2}{1-\gamma}}, \\
p(R,t) &= p_2 \left(\frac{\rho(R,t)}{\rho_2} \right)^{\gamma}, \\
u_r(R,t) &= -\frac{tR\cos\phi}{\tau^2 h(t)^2}, \\
u_z(R,t) &= -\frac{tR\sin\phi}{\tau^2 h(t)^2}.
\end{aligned}$$

The spherical symmetry is achieved by taking $\gamma = 5/3$ and we choose following parameters: $p_1 = 1$, $p_2 = 100$ and $\rho_2 = 1$. Computations have been run for $r \in [r_1; r_2] = [0; 1]$ until time $t = 0.5\tau$ with a CFL coefficient of 0.9. The computational domain has reflective boundary conditions along the z and r axis, the boundary conditions being imposed by the analytical solution elsewhere. We have reported the error between the analytical and numerical solutions (measured using a L^1 norm in space and time) in Table 6.1 with and without hyperviscosity. One can notice that these results are very satisfying: the experimental order of convergence perfectly matches the theoretical one and it is not affected by hyperviscosity.

Two-dimensional axisymmetric ideal MHD: Picard's [70] analytical solution

We now focus on an analytical solution to the two-dimensional axisymmetric ideal MHD equations provided by Picard [71]. Defining $R = \sqrt{r^2 + z^2}$, $C_0 = (3\gamma - 1)/2$ and $f(t) = (C_0 t + 1)^{-1}$, it writes:

$$\begin{aligned}
\rho(z,r,t) &= \left[\frac{\gamma}{\gamma - 1} \left(f(t)^{\frac{3(\gamma-1)}{C_0}} + \frac{f(t)^2 \sqrt{r^2 + z^2}}{2(C_0 - 1)} \right) \right]^{\frac{1}{\gamma-1}}, \\
p(z,r,t) &= \rho(z,r,t)^{\gamma}, \\
(u_z,\ u_r,\ u_\theta)(z,r,t) &= f(t) \cdot (z,\ r,\ 0)^t, \\
(B_z,\ B_r,\ B_\theta)(z,r,t) &= (r^z + z^2)^{-\frac{3}{2}} \cdot (z,\ r,\ 0)^t.
\end{aligned}$$

This test problem has been set on the $[1; 6] \times [0; 5]$ computational domain with $\gamma = 2$, boundary conditions being imposed by the analytical solution. We have run computations until time $t = 0.1$ and reported the error measurements (evaluated using a L^1 norm in space and time) between the exact and numerical solutions in Table 6.2. These show that the experimental order of convergence matches the theoretical one and that it is affected neither by hyperviscosity nor by divergence cleaning.

WITHOUT HYPERVISCOSITY						
Mesh size	**2nd-order**		**3rd-order**		**4th-order**	
(N × N cells)	error	order	error	order	error	order
N = 100	1.96e-04		1.27e-05		1.68e-07	
N = 200	4.92e-05	1.99	1.55e-06	3.03	1.03e-08	4.03
N = 400	1.23e-05	2.00	1.91e-07	3.02	6.40e-10	4.01
N = 800	3.07e-06	2.00	2.38e-08	3.01	4.01e-11	4.00
N = 1600	7.69e-07	2.00	2.96e-09	3.00	2.53e-12	3.99
WITH HYPERVISCOSITY ($C_\beta = C_\nu = C_\kappa = 1$)						
Mesh size	**2nd-order**		**3rd-order**		**4th-order**	
(N × N cells)	error	order	error	order	error	order
N = 100	2.21e-04		1.29e-05		1.71e-07	
N = 200	5.22e-05	2.08	1.58e-06	3.03	1.06e-08	4.02
N = 400	1.27e-05	2.04	1.96e-07	3.01	6.58e-10	4.01
N = 800	3.12e-06	2.02	2.44e-08	3.01	4.13e-11	4.00
N = 1600	7.74e-07	2.01	3.05e-09	3.00	2.60e-12	3.99

Table 6.1: Error measured using a L^1 norm in space and time between the analytical and numerical solutions for Kidder's isentropic compression test problem in two-dimensional axisymmetric geometry.

WITHOUT HYPERVISCOSITY						
Mesh size	**2nd-order**		**3rd-order**		**4th-order**	
(N × N cells)	error	order	error	order	error	order
N = 100	1.11e-03		2.03e-05		6.29e-08	
N = 200	2.71e-04	2.03	2.21e-06	3.01	3.86e-09	4.02
N = 400	6.71e-05	2.01	3.12e-07	3.01	2.41e-10	4.00
N = 800	1.67e-05	2.00	3.89e-08	3.00	1.50e-11	4.00
N = 1600	4.16e-06	2.00	4.86e-09	3.00	9.40e-13	4.00
WITH HYPERVISCOSITY ($C_\beta = C_\nu = C_\kappa = C_\eta = 1$)						
Mesh size	**2nd-order**		**3rd-order**		**4th-order**	
(N × N cells)	error	order	error	order	error	order
N = 100	1.09e-03		7.81e-06		6.29e-08	
N = 200	2.69e-04	2.02	9.70e-07	3.01	3.86e-09	4.02
N = 400	6.69e-05	2.01	1.21e-07	3.01	2.41e-10	4.00
N = 800	1.67e-05	2.01	1.51e-08	3.00	1.50e-11	4.00
N = 1600	4.16e-06	2.00	1.88e-09	3.00	9.40e-13	4.00

Table 6.2: Error measured using a L^1 norm in space and time between the analytical and numerical solutions for Picard's analytical solution to the axisymmetric two-dimensional ideal MHD system.

6.4.2 Non-smooth test problems

Now that the convergence features of GoHy schemes for the axisymmetric case have been illustrated, we focus on two non-smooth test problems - the Sod shock tube and the Noh problem - in both cylindrically and spherically symmetric geometries which respectively correspond to one-dimensional and two-dimensional axisymmetric geometries. Note that these two benchmarks have been already been described in the planar case (see section 4.5.2), we thus do not recall initial conditions in the sequel.

Sod shock tube (axisymmetric hydrodynamics)

We first consider the one-dimensional Sod shock tube problem in cylindrically symmetric geometry that we set on the $r \in [0;1]$ domain with a reflective left boundary and an open right one. Figure 6.1 plots the density and pressure obtained at time $t = 0.14$ with the fourth-order scheme and a CFL coefficient of 0.7. When hyperviscosity is enabled, the artifial coefficients are $C_\beta = 2$ and $C_\kappa = 5$. These results highlight the impact of hyperviscosity on oscillations (these have been noticeably reduced, leading to a satisfying approximation of the exact solution) and tend to validate the usage of artificial viscosity in the axisymmetric case.

We now consider the two-dimensional Sod shock tube problem in spherically symmetric geometry that we set on the $[0;1]^2$ computational domain with reflective bottom and left boundary conditions and open boundaries elsewhere. We have run this benchmark until time $t = 0.5$ with the fourth-order scheme and a CFL coefficient of 0.7 on 1000×1000 cells. Figure 6.2 plots the density at final time with two different artificial viscosity configurations.

- The first one only resorts to hyperviscosity with following parameters: $C_\beta = 1$, $C_\nu = 0.1$ and $C_\kappa = 2$. The corresponding results show that Gibbs oscillations have been reduced but they also reveal symmetry losses, especially in the vicinity of the z and r axis.
- The second one consists in designing $\beta^\star$ and $\kappa^\star$ with the hyperviscosity model whereas $\nu^\star$ is determined by the first-order artificial viscosity model, the artificial constants being the same than for the first configuration. This strategy clearly improves symmetry preservation, leading to a satisfying approximate solution.

Noh problem (axisymmetric hydrodynamics)

We now examine the behaviour of GoHy schemes on the stringent Noh problem that we first set in cylindrically symmetric geometry on the $[0;0.4]$ computational domain with a reflective left boundary and an open right one. Figure 6.3 plots the density and pressure obtained at time $t = 0.6$ on 400 cells with the fourth-order scheme and a CFL coefficient of 0.4, the hyperviscosity model being enabled with $C_\beta = 5$ and $C_\kappa = 7$. Like its planar version, the axisymmetric Lagrange-remap scheme encounters some wall heating issues in the vicinity of the reflective boundary that can be treated by combining the hyperviscosity model with high values of C_κ (see Figure 6.4).

The Noh problem is now considered in spherically symmetric geometry. We set it on the $[0;0.4]^2$ domain with reflective bottom and left boundaries and open boundaries elsewhere. We have run computations until time $t = 0.6$ with the fourth-order GoHy scheme and a CFL coefficient of 0.4, the hyperviscosity model being enabled with following parameters: $C_\beta = 5$, $C_\nu = 1$, $C_\kappa = 20$. Figure 6.5 plots the density and pressure obtained at final time on 400 cells. We observe the same behaviour than in the one-dimensional case: one the one hand, taking large values of C_κ noticeably reduces wall heating (see Figure 6.7) and,

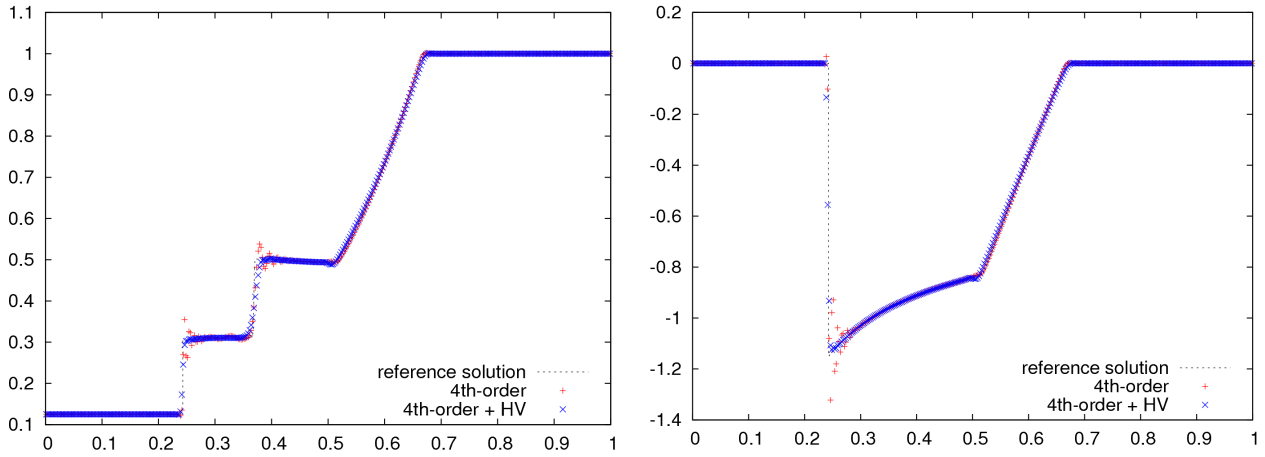


Figure 6.1: One-dimensional Sod shock tube in cylindrically symmetric geometry at time $t = 0.14$. Density (left) and velocity (right) obtained on 400 cells with the fourth-order scheme. The reference solution has been obtained using a first-order accurate Lagrange-remap scheme cylindrical geometry [30] on 100,000 cells.

on the other hand, the approximate solution converges towards the exact one when refining the mesh (see Figure 6.6).

As we have seen in the planar case, the solution to the Noh problem is known analytically [67]. At time $t = 0.6$, the exact density and pressure write for the one-dimensional problem write:

$$\rho = \begin{cases} 16 & r < 0.2, \\ 1 + \frac{0.6}{r} & r > 0.2, \end{cases} \quad \text{and} \quad p = \begin{cases} 16/3 & r < 0.2, \\ 0 & r > 0.2, \end{cases}$$

whereas they are given in the two-dimensional case by:

$$\rho = \begin{cases} 64 & \sqrt{z^2 + r^2} < 0.2, \\ \left(1 + \frac{0.6}{\sqrt{z^2+r^2}}\right)^2 & \sqrt{z^2 + r^2} > 0.2, \end{cases} \quad \text{and} \quad p = \begin{cases} 64/3 & \sqrt{z^2 + r^2} < 0.2, \\ 0 & \sqrt{z^2 + r^2} > 0.2. \end{cases}$$

Two-dimensional axisymmetric MHD implosion problem

We conclude this chapter with a test problem taken from [59] that represents an imploding sphere in axisymmetric geometry. The initial spherically symmetric configuration is split between an outer low-density and high-pressure shell and an inner high-density and low-pressure region. More precisely, it is given by:

$$(\rho,\ p) = \begin{cases} (10,\ 1) & \text{if } 0 \leq \sqrt{z^2 + r^2} \leq 0.8, \\ (1,\ 1000) & \text{if } 0.8 \leq \sqrt{z^2 + r^2} \leq 2. \end{cases}$$

The fluid is initially at rest. A uniform magnetic field is set along the z-direction with a magnitude of 40. The other parameters are $\gamma = 5/3$ and $\mu = 4\pi$. Numerically, this test problem is set on the $(z, r) \in [-2; 2] \times [0; 2]$ domain with a reflective boundary condition along the $r = 0$ axis and open boundaries elsewhere. We have run computations on 1600×800 cells until time $t = 0.07$ with the third-order GoHy scheme and a CFL

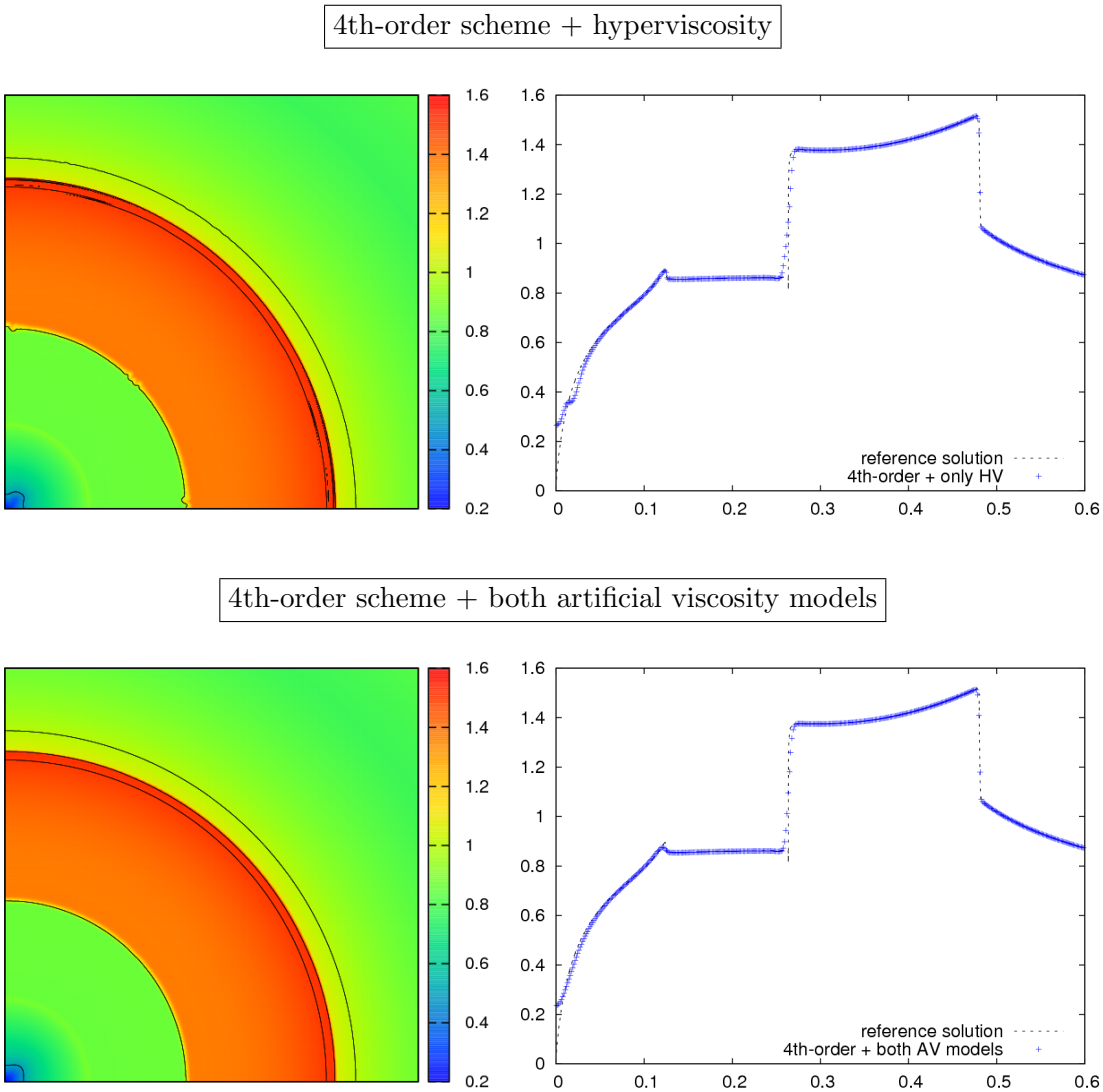


Figure 6.2: Density obtained for the two-dimensional Sod shock tube in spherically symmetric geometry at time $t = 0.5$ on 1000×1000 cells. Left: zoom on the $[0; 0.6]^2$ domain. Right: slice along the $z = r$ axis. The reference solution has been obtained using a one-dimensional first-order accurate Lagrange-remap scheme in spherical geometry [30] on 100,000 cells.

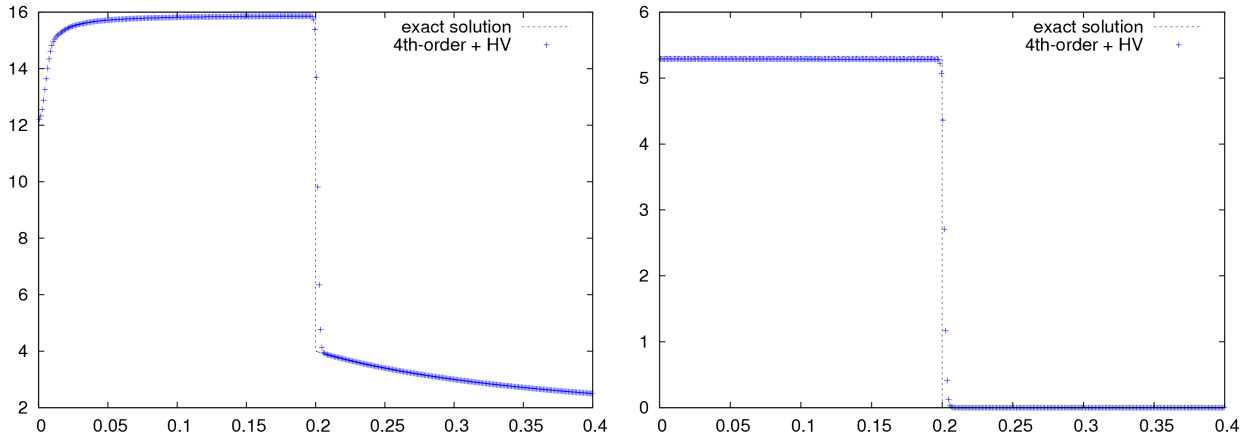


Figure 6.3: Density (left) and pressure (right) for the Noh problem in cylindrically symmetric geometry at time $t = 0.6$ on 400 cells.

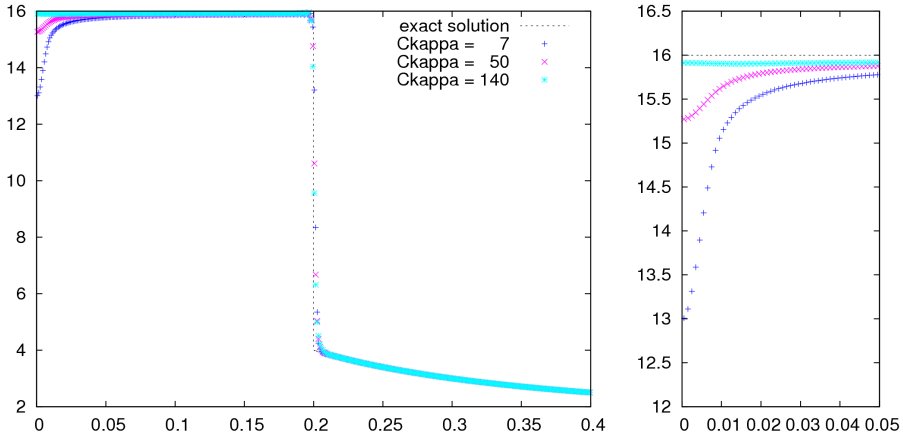


Figure 6.4: Density for the cylindrically symmetric Noh problem at time $t = 0.6$ on 400 cells with different values of C_κ: whole computational domain (left) and zoom on the $[0; 0.05]$ region (right).

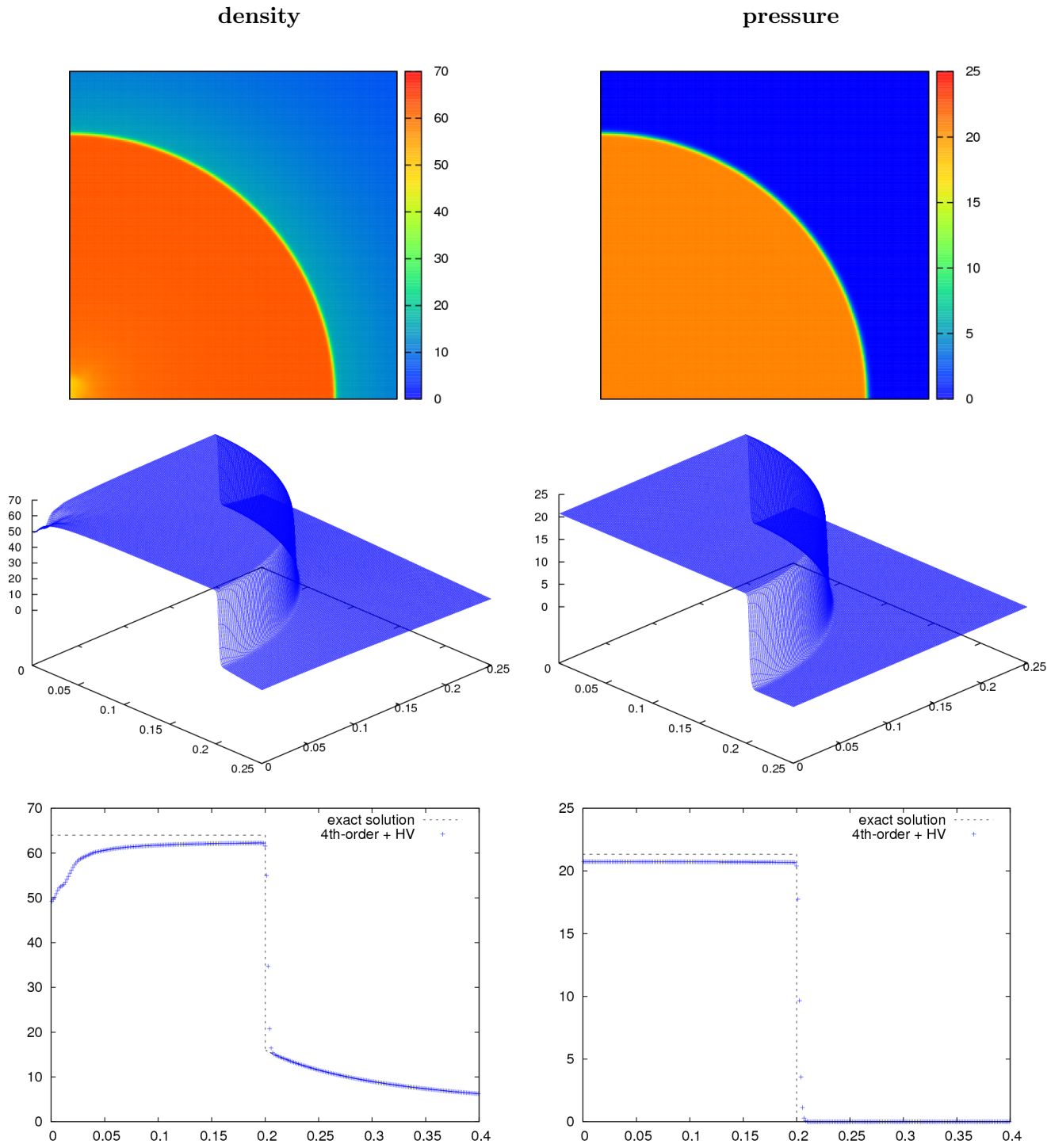


Figure 6.5: Two-dimensional Noh problem in spherically symmetric geometry: density (left) and pressure (right) obtained at time $t = 0.6$. From top to bottom: map plot, surface plot and slice along the $z = r$ axis.

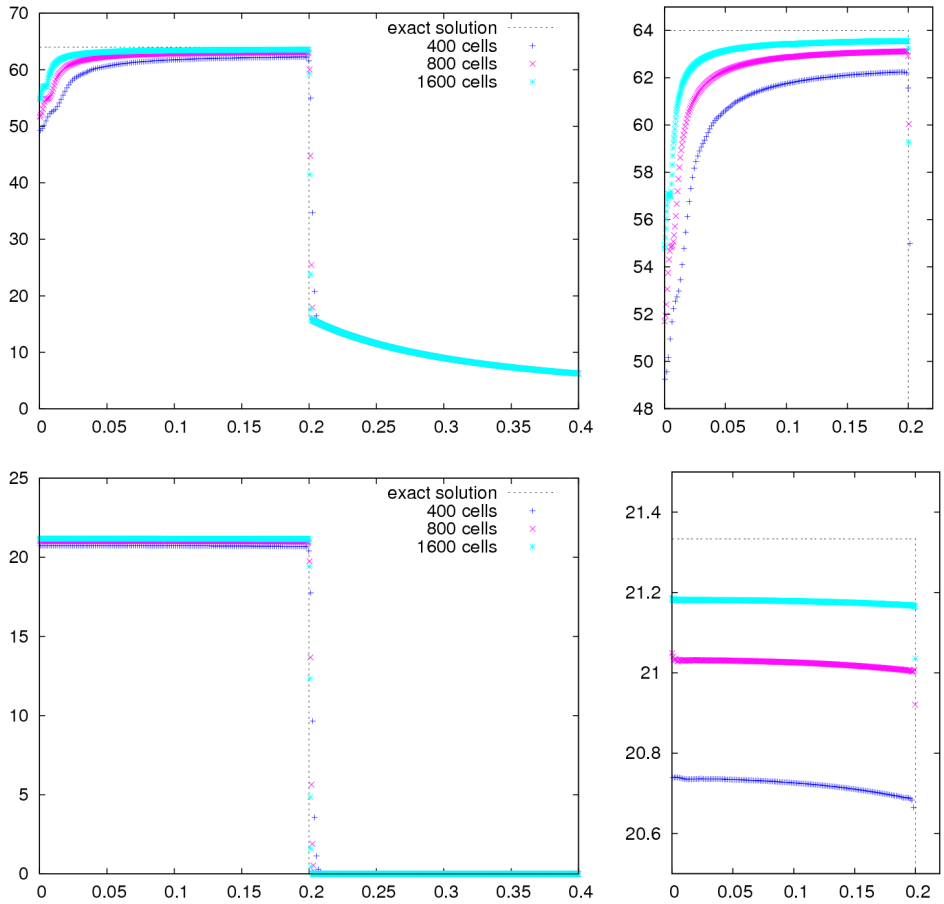


Figure 6.6: Density (top) and pressure (bottom) along the $z = r$ axis for the Noh problem in spherically symmetric geometry at time $t = 0.6$ on different meshes: whole computational domain (left) and zoom on the left state (right).

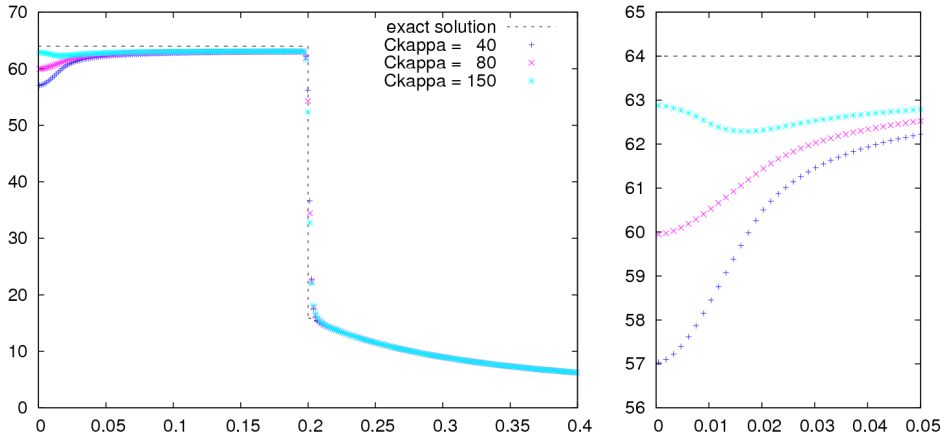


Figure 6.7: Density along the $z = r$ axis for the two-dimensional Noh problem at time $t = 0.6$ on 400 cells with different values of C_κ: whole computational domain (left) and zoom on the $[0; 0.05]$ region (right).

coefficient of 0.5. The fourth-order scheme was unfortunately not able to pass this test, showing that robustness still has to be improved in the axisymmetric case. Figures 6.8 and 6.9 plot the density and pressure distributions as well as the r and z-components of the magnetic field, respectively at time $t = 0.05$ and $t = 0.07$. These show a good agreement with numerical results given in the literature [59]. One can also notice that the symmetry properties of B_z and the antisymmetry properties of B_r with respect to the $z = 0$ axis are preserved.

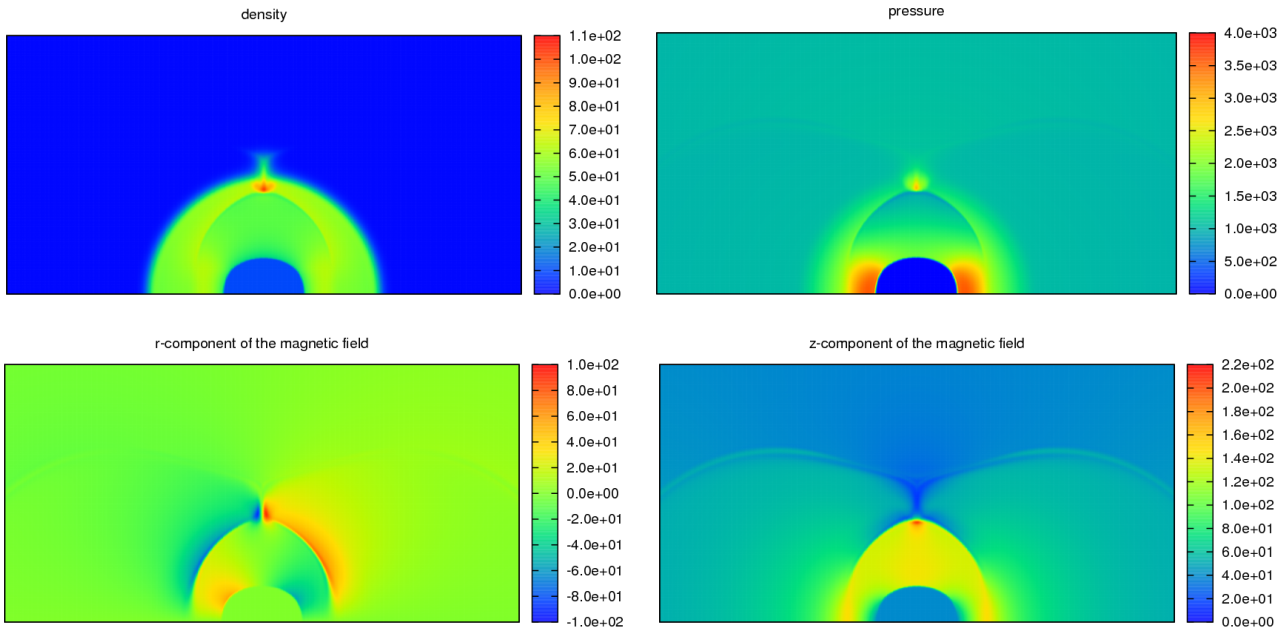


Figure 6.8: Numerical results for the implosion test problem obtained at time $t = 0.05$ with the third-order MHD-GoHy scheme (zoom on the $(z, r) \in [-1; 1] \times [0; 1]$ domain).

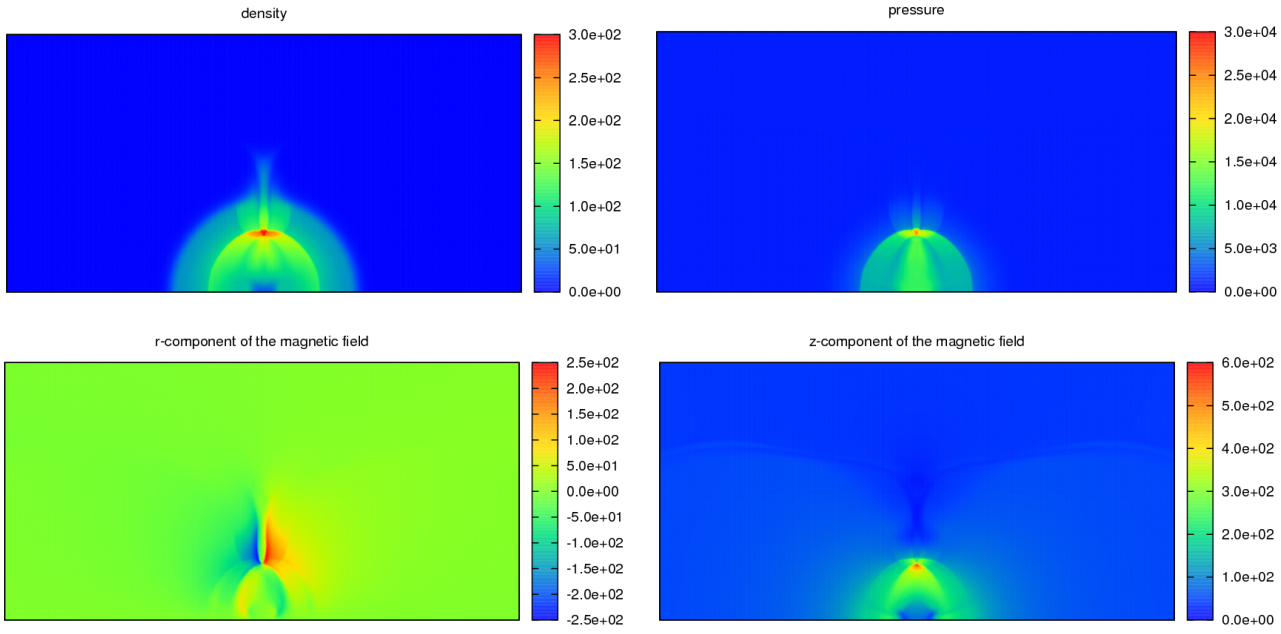


Figure 6.9: Numerical results for the implosion test problem obtained at time $t = 0.07$ with the third-order MHD-GoHy scheme (zoom on the $(z, r) \in [-1; 1] \times [0; 1]$ domain).

Part III

Towards direct drive ICF computations

Chapter 7

Discretization of diffusion and source terms operators

The two-temperature resistive MHD model described in chapter 1 that has been derived for performing direct drive ICF simulations relies on mathematical operators of different natures. While the previous part has been devoted to designing numerical methods for the hyperbolic operator, we discuss in this chapter the discretization of the remaining operators, namely the thermal and resistive conduction operators and the source terms that take magnetic field generation as well as Hall and Nernst effects into account. In order to be coupled with GoHy schemes through an operator splitting strategy, numerical methods presented in the sequel are built in the same finite volume framework. In particular, the treatment of the B_θ equation in the axisymmetric case is coherent with the approach described in section 6.1.3.

This study being devoted to the case of two-dimensional geometries, we will see in section 7.2 that the only possibly non-zero component of the magnetic field is the one that is orthogonal to the computational domain, *i.e.* B_z in the planar case and B_θ in the axisymmetric one. We thus restrict to the orthogonal MHD framework in the sequel.

The numerical schemes described in this chapter do not aim to achieve high-order accuracy like the previously presented GoHy schemes. We actually propose classical first-order accurate finite volume discretizations of the considered diffusion and source term operators. Our objective simply is to allow the numerical resolution of the complete two-temperature resistive MHD model derived in chapter 1 at a reasonable computational cost. In practice, even second-order accurate schemes would be too costly. Indeed, the operator splitting strategy proposed in chapter 2 should be second-order accurate too - and thus more expensive - and the implementation of diffusion schemes would reveal very difficult due to the complexity of Braginskii's conductivity and resistivity tensors given in section 1.6. A part of these developments have been performed during the 2010 CEMRACS summer school and have been published as part of the CEMRACS proceedings [92].

The outline of this chapter is the following. We first propose finite volume numerical methods for the systems of equations that govern thermal and resistive conduction. In this context, attention will be paid to the estimation of conductivity and resistivity tensors at cell interfaces. Then we give a few details about the implementation of these schemes and present several numerical results on smooth and non-smooth test problems in order to validate diffusion schemes. We finally describe the strategy that has been adopted for dealing with magnetic field generation and Hall and Nernst effects.

7.1 Diffusion schemes

7.1.1 Thermal conduction

We first consider in this section the system of equations that governs thermal conduction. According to the operator splitting presented in chapter 2, it is given by (2.3) and writes:

$$\begin{cases} \partial_t \rho = 0, \\ \partial_t(\rho \mathbf{u}) = 0, \\ \partial_t(\rho_e \epsilon_e) + \nabla \cdot \mathbf{q}_e = 3\frac{m_e}{m_i} n_e \nu_{ei} k_B (T_i - T_e), \\ \partial_t(\rho_i \epsilon_i) + \nabla \cdot \mathbf{q}_i = 3\frac{m_e}{m_i} n_e \nu_{ei} k_B (T_e - T_i), \\ \partial_t \mathbf{B} = 0. \end{cases}$$

The density, velocity and magnetic field components being constant, we now focus on the electronic and ionic temperatures equations. Since $\epsilon = C_v T$ for both ions and electrons (where C_v is a constant), these write:

$$\rho_e C_{v_e} \partial_t T_e + \nabla \cdot \mathbf{q}_e = 3\frac{m_e}{m_i} n_e \nu_{ei} k_B (T_i - T_e), \tag{7.1a}$$

$$\rho_i C_{v_i} \partial_t T_i + \nabla \cdot \mathbf{q}_i = 3\frac{m_e}{m_i} n_e \nu_{ei} k_B (T_e - T_i). \tag{7.1b}$$

We recall that the electronic and ionic heat fluxes are defined by:

$$\mathbf{q}_e = -k_B \left(\overline{\overline{\kappa}}_e \cdot \nabla T_e\right) - \frac{k_B}{e}\left(\overline{\overline{\beta}} \cdot \mathbf{J}\right) T_e, \tag{7.2a}$$

$$\mathbf{q}_i = -k_B \left(\overline{\overline{\kappa}}_i \cdot \nabla T_i\right), \tag{7.2b}$$

where the thermoelectric tensor $\overline{\overline{\beta}}$ and the conductivity tensors $\overline{\overline{\kappa}}_e$ and $\overline{\overline{\kappa}}_i$ are given by Braginskii's closure relations (see section 1.6). Equations (7.1a) and (7.1b) being very similar, we only discuss from now on the case of the electronic temperature equation whose detailed form writes:

$$\rho_e C_{v_e} \partial_t T_e + \partial_1 \left(\mathbf{q}_e \cdot \mathbf{e}_1\right) + \frac{1}{x_2^\alpha} \partial_2 \left(x_2^\alpha \mathbf{q}_e \cdot \mathbf{e}_2\right) = 3\frac{m_e}{m_i} n_e \nu_{ei} k_B (T_i - T_e) \tag{7.3}$$

with notations introduced in Appendix C. Equation (7.3) is now discretized in the finite volume framework as indicated previously. To that end, we first multiply it by x_2^α and then integrate it over the (i,j) cell $[(x_1)_{i-\frac{1}{2}}; (x_1)_{i+\frac{1}{2}}] \times [(x_2)_{j-\frac{1}{2}}; (x_2)_{j+\frac{1}{2}}]$, which leads to the following semi-discrete numerical scheme:

$$\begin{aligned} C_{v_e} \cdot (x_2^\alpha)_j \cdot (\rho_e)_{i,j} \cdot \partial_t \overline{(T_e)}_{i,j} + \frac{1}{\Delta x_2}\left[(x_2^\alpha)_{j+\frac{1}{2}} (\mathbf{q}_e \cdot \mathbf{e}_2)_{i,j+\frac{1}{2}} - (x_2^\alpha)_{j-\frac{1}{2}} (\mathbf{q}_e \cdot \mathbf{e}_2)_{i,j-\frac{1}{2}}\right] \\ + \frac{(x_2^\alpha)_j}{\Delta x_1}\left[(\mathbf{q}_e \cdot \mathbf{e}_1)_{i+\frac{1}{2},j} - (\mathbf{q}_e \cdot \mathbf{e}_1)_{i-\frac{1}{2},j}\right] = 3\frac{k_B m_e}{m_i} \cdot (x_2^\alpha)_j \cdot (n_e)_{i,j} \cdot (\nu_{ei})_{i,j} \cdot \left[\overline{(T_i)}_{i,j} - \overline{(T_e)}_{i,j}\right] \end{aligned} \tag{7.4}$$

where $\overline{(T_e)}_{i,j}$ denotes the cell average of T_e over the (i,j) cell:

$$\overline{(T_e)}_{i,j} = \frac{1}{(x_2^\alpha)_j \Delta x_2} \int_{(x_2)_{j-\frac{1}{2}}}^{(x_2)_{j+\frac{1}{2}}} x_2^\alpha \left[\frac{1}{\Delta x_1} \int_{(x_1)_{i-\frac{1}{2}}}^{(x_1)_{i+\frac{1}{2}}} T_e(x_1, x_2, t)\, dx_1\right] dx_2.$$

In the specific two-dimensional case with $\mathbf{B} = B\mathbf{e}_3$, the expression of the electronic heat flux using notations introduced in Appendix C is given by:

$$\mathbf{q}_e = -k_B \begin{pmatrix} \kappa_e^{\perp}\partial_1 T_e - \kappa_e^{\wedge}\partial_2 T_e \\ \kappa_e^{\perp}\partial_2 T_e + \kappa_e^{\wedge}\partial_1 T_e \\ 0 \end{pmatrix} - \frac{ck_B T_e}{e\mu} \begin{pmatrix} \frac{\beta^{\perp}}{x_2^{\alpha}}\partial_2\left(x_2^{\alpha}B\right) + \beta^{\wedge}\partial_1 B \\ -\beta^{\perp}\partial_1 B - \frac{\beta^{\wedge}}{x_2^{\alpha}}\partial_2\left(x_2^{\alpha}B\right) \\ 0 \end{pmatrix}.$$

We now combine the finite volume scheme (7.4) with a semi-implicit time discretization as suggested in introduction (see page 15). More precisely, the relaxation term is written implicitly and the electronic temperature terms that appear in (7.2a) are taken at time t^{n+1}. The resulting numerical scheme writes:

$$\begin{aligned} &C_{v_e}\cdot(x_2^{\alpha})_j\cdot(\rho_e)_{i,j}\cdot\frac{\overline{(T_e)}_{i,j}^{n+1}-\overline{(T_e)}_{i,j}^{n}}{\Delta t}+\frac{1}{\Delta x_2}\left[(x_2^{\alpha})_{j+\frac{1}{2}}(\mathbf{q}_e)^*_{i,j+\frac{1}{2}}-(x_2^{\alpha})_{j-\frac{1}{2}}(\mathbf{q}_e)^*_{i,j-\frac{1}{2}}\right] \\ &+\frac{(x_2^{\alpha})_j}{\Delta x_1}\left[(\mathbf{q}_e)^*_{i+\frac{1}{2},j}-(\mathbf{q}_e)^*_{i-\frac{1}{2},j}\right]=3\frac{k_B m_e}{m_i}\cdot(x_2^{\alpha})_j\cdot(n_e)^n_{i,j}\cdot(\nu_{ei})^n_{i,j}\cdot\left[\overline{(T_i)}_{i,j}^{n+1}-\overline{(T_e)}_{i,j}^{n+1}\right]. \end{aligned} \tag{7.5}$$

Remark 30. The time discretization given in (7.5) is only semi-implicit because the conductivity tensor $\overline{\overline{\kappa}}_e$ which is evaluated at time t^n depends on T_e as indicated in its definition (see page 32).

The approximate heat fluxes along the x_1 and x_2 directions are respectively given by:

$$\begin{aligned} (\mathbf{q}_e)^*_{i+\frac{1}{2},j} &\approx (\mathbf{q}_e\cdot\mathbf{e}_1)_{i+\frac{1}{2},j}, \\ &= -k_B\cdot\left(\kappa_e^{\perp}\right)^n_{i+\frac{1}{2},j}\cdot\left(\frac{(T_e)^{n+1}_{i+1,j}-(T_e)^{n+1}_{i,j}}{\Delta x_1}\right)-\frac{k_B}{e}\cdot\left(\overline{\overline{\beta}}\cdot\mathbf{J}\right)_{i+\frac{1}{2},j}\cdot\left(\frac{(T_e)^{n+1}_{i,j}+(T_e)^{n+1}_{i+1,j}}{2}\right) \\ &\quad +k_B\cdot\left(\kappa_e^{\wedge}\right)^n_{i+\frac{1}{2},j}\cdot\left(\frac{(T_e)^{n+1}_{i+1,j+1}+(T_e)^{n+1}_{i,j+1}-(T_e)^{n+1}_{i+1,j-1}-(T_e)^{n+1}_{i,j-1}}{4\Delta x_2}\right), \\ (\mathbf{q}_e)^*_{i,j+\frac{1}{2}} &\approx (\mathbf{q}_e\cdot\mathbf{e}_2)_{i,j+\frac{1}{2}}, \\ &= -k_B\cdot\left(\kappa_e^{\perp}\right)^n_{i,j+\frac{1}{2}}\cdot\left(\frac{(T_e)^{n+1}_{i,j+1}-(T_e)^{n+1}_{i,j}}{\Delta x_2}\right)-\frac{k_B}{e}\cdot\left(\overline{\overline{\beta}}\cdot\mathbf{J}\right)_{i,j+\frac{1}{2}}\cdot\left(\frac{(T_e)^{n+1}_{i,j}+(T_e)^{n+1}_{i,j+1}}{2}\right) \\ &\quad -k_B\cdot\left(\kappa_e^{\wedge}\right)^n_{i,j+\frac{1}{2}}\cdot\left(\frac{(T_e)^{n+1}_{i+1,j+1}+(T_e)^{n+1}_{i+1,j}-(T_e)^{n+1}_{i-1,j+1}-(T_e)^{n+1}_{i-1,j}}{4\Delta x_1}\right), \end{aligned}$$

with:

$$\begin{aligned} \left(\overline{\overline{\beta}}\cdot\mathbf{J}\right)_{i+\frac{1}{2},j} &= +\frac{c}{\mu}\cdot\left(\beta^{\wedge}\right)^n_{i+\frac{1}{2},j}\cdot\left(\frac{B^n_{i+1,j}-B^n_{i,j}}{\Delta x_1}\right) \\ &\quad +\frac{c}{\mu}\cdot\left(\beta^{\perp}\right)^n_{i+\frac{1}{2},j}\cdot\left(\frac{(x_2^{\alpha})_{j+1}\left(B^n_{i+1,j+1}+B^n_{i,j+1}\right)-(x_2^{\alpha})_{j-1}\left(B^n_{i+1,j-1}+B^n_{i,j-1}\right)}{4(x_2^{\alpha})_j\Delta x_2}\right), \\ \left(\overline{\overline{\beta}}\cdot\mathbf{J}\right)_{i,j+\frac{1}{2}} &= -\frac{c}{\mu}\cdot\left(\beta^{\wedge}\right)^n_{i,j+\frac{1}{2}}\cdot\left(\frac{(x_2^{\alpha})_{j+1}B^n_{i,j+1}-(x_2^{\alpha})_j B^n_{i,j}}{(x_2^{\alpha})_{j+\frac{1}{2}}\Delta x_2}\right) \\ &\quad -\frac{c}{\mu}\cdot\left(\beta^{\perp}\right)^n_{i,j+\frac{1}{2}}\cdot\left(\frac{B^n_{i+1,j+1}+B^n_{i+1,j}-B^n_{i-1,j+1}-B^n_{i-1,j}}{4\Delta x_1}\right). \end{aligned}$$

Remark 31. The value of the heat flux being imposed by boundary conditions along the $x_2 = 0$ axis, the $(x_2^\alpha)_{j+\frac{1}{2}}$ denominator in the expression of $(\overline{\overline{\beta}} \cdot \mathbf{J})_{i,j+\frac{1}{2}}$ does not lead to pathologic approximations in the axisymmetric case.

The cell interface representations of the conductivity tensors' components have been determined so that the heat flux is continuous across cell interfaces as explained in Appendix D. They are given by harmonic means:

$$\kappa^\perp_{i+\frac{1}{2},j} = 2\frac{\kappa^\perp_{i,j} \cdot \kappa^\perp_{i+1,j}}{\kappa^\perp_{i,j} + \kappa^\perp_{i+1,j}}, \qquad \kappa^\wedge_{i+\frac{1}{2},j} = \frac{\kappa^\perp_{i,j} \cdot \kappa^\wedge_{i+1,j} + \kappa^\wedge_{i,j} \cdot \kappa^\perp_{i+1,j}}{\kappa^\perp_{i,j} + \kappa^\perp_{i+1,j}},$$
$$\kappa^\perp_{i,j+\frac{1}{2}} = 2\frac{\kappa^\perp_{i,j} \cdot \kappa^\perp_{i,j+1}}{\kappa^\perp_{i,j} + \kappa^\perp_{i,j+1}}, \qquad \kappa^\wedge_{i,j+\frac{1}{2}} = \frac{\kappa^\perp_{i,j} \cdot \kappa^\wedge_{i,j+1} + \kappa^\wedge_{i,j} \cdot \kappa^\perp_{i,j+1}}{\kappa^\perp_{i,j} + \kappa^\perp_{i,j+1}},$$

Regarding the thermoelectric tensor, the following cell interface representations that resort to arithmetic means have been implemented:

$$\beta^\perp_{i+\frac{1}{2},j} = \frac{\beta^\perp_{i,j} + \beta^\perp_{i+1,j}}{2}, \qquad \beta^\wedge_{i+\frac{1}{2},j} = \frac{\beta^\wedge_{i,j} + \beta^\wedge_{i+1,j}}{2},$$
$$\beta^\perp_{i,j+\frac{1}{2}} = \frac{\beta^\perp_{i,j} + \beta^\perp_{i,j+1}}{2}, \qquad \beta^\wedge_{i,j+\frac{1}{2}} = \frac{\beta^\wedge_{i,j} + \beta^\wedge_{i,j+1}}{2}.$$

Remark 32. In order to guarantee the continuity of the heat flux across cell interfaces, one should use the cell interfaces representations of the thermoelectric tensor given by:

$$\beta^\perp_{i+\frac{1}{2},j} = -\frac{\beta^\perp_{i,j} \cdot \beta^\wedge_{i+1,j} + \beta^\wedge_{i,j} \cdot \beta^\perp_{i+1,j}}{\beta^\wedge_{i,j} + \beta^\wedge_{i+1,j}}, \qquad \beta^\wedge_{i+\frac{1}{2},j} = -2\frac{\beta^\wedge_{i,j} \cdot \beta^\wedge_{i+1,j}}{\beta^\wedge_{i,j} + \beta^\wedge_{i+1,j}},$$
$$\beta^\perp_{i,j+\frac{1}{2}} = -\frac{\beta^\perp_{i,j} \cdot \beta^\wedge_{i,j+1} + \beta^\wedge_{i,j} \cdot \beta^\perp_{i,j+1}}{\beta^\wedge_{i,j} + \beta^\wedge_{i,j+1}}, \qquad \beta^\wedge_{i,j+\frac{1}{2}} = -2\frac{\beta^\wedge_{i,j} \cdot \beta^\wedge_{i,j+1}}{\beta^\wedge_{i,j} + \beta^\wedge_{i,j+1}}.$$

as explained in Appendix D. These have not been tested in our implementation.

Solving the evolution equations (7.1a) and (7.1b) using the finite volume scheme (7.4) finally consists in solving a linear system whose unknowns are the cell averages of T_i and T_e. In the general case, this linear system does not seem to have any interesting property that could help its numerical resolution. It is in particular unsymmetric as shown below.

Proposition 12. *In the general case, the semi-implicit finite volume scheme* (7.4) *leads to an unsymmetric linear system.*

Proof. Let $(m_e)^{i,j}$ denote the non-zero matrix coefficients of the line that corresponds to the (i,j) cell for the T_e unknown. One can notice that:

$$(m_e)^{i,j}_{i+1,j+1} = \frac{k_B}{4\Delta x_1 \Delta x_2}\left((x_2^\alpha)_j \left(\kappa_e^\wedge\right)^n_{i+\frac{1}{2},j} - (x_2^\alpha)_{j+\frac{1}{2}} \left(\kappa_e^\wedge\right)^n_{i,j+\frac{1}{2}}\right)$$

whereas:

$$(m_e)^{i+1,j+1}_{i,j} = \frac{k_B}{4\Delta x_1 \Delta x_2}\left((x_2^\alpha)_{j+1} \left(\kappa_e^\wedge\right)^n_{i+\frac{1}{2},j+1} - (x_2^\alpha)_{j+\frac{1}{2}} \left(\kappa_e^\wedge\right)^n_{i+1,j+\frac{1}{2}}\right).$$

This shows that the considered matrix is unsymmetric since:

$$(m_e)^{i,j}_{i+1,j+1} \neq (m_e)^{i+1,j+1}_{i,j}.$$

□

We therefore resort to a stabilized biconjugate gradient algorithm (BiCGStab) [74] for solving the heat conduction system. Nevertheless, when all of the $\kappa^\wedge$ terms are zero (*i.e.* when the conductivity tensor is diagonal, which occurs in particular in the absence of magnetic field), one can show that the above-mentioned linear system is symmetric and definite positive, allowing the usage of a more classical conjugate gradient (CG) algorithm [74].

Proposition 13. *If the conductivity tensor is diagonal, solving the semi-implicit finite volume scheme* (7.4) *with the heat flux* $\mathbf{q}$ *defined by* $\mathbf{q} = -\overline{\overline{\kappa}} \cdot \nabla T_e$ *leads to a symmetric and definite positive linear system.*

Proof. If $\overline{\overline{\kappa}}$ is a diagonal tensor, the heat conduction scheme for the (i, j) cell writes:

$$\begin{aligned} C_{v_e} \cdot (x_2^\alpha)_j \cdot (\rho_e)_{i,j}^n \cdot \left(\overline{(T_e)}_{i,j}^{n+1} - \overline{(T_e)}_{i,j}^{n} \right) - \nu_1 \left[\kappa^\perp_{i+\frac{1}{2},j} \left(\overline{(T_e)}_{i+1,j}^{n+1} - \overline{(T_e)}_{i,j}^{n+1} \right) - \kappa^\perp_{i-\frac{1}{2},j} \left(\overline{(T_e)}_{i,j}^{n+1} - \overline{(T_e)}_{i-1,j}^{n+1} \right) \right] \\ -\nu_2 \left[(x_2^\alpha)_{j+\frac{1}{2}} \kappa^\perp_{i,j+\frac{1}{2}} \left(\overline{(T_e)}_{i,j+1}^{n+1} - \overline{(T_e)}_{i,j}^{n+1} \right) - (x_2^\alpha)_{j-\frac{1}{2}} \kappa^\perp_{i,j-\frac{1}{2}} \left(\overline{(T_e)}_{i,j}^{n+1} - \overline{(T_e)}_{i,j-1}^{n+1} \right) \right] \\ = 3 \frac{k_B m_e}{m_i} \cdot (x_2^\alpha)_j \cdot (n_e)_{i,j}^n \cdot (\nu_{ei})_{i,j}^n \cdot \left(\overline{(T_i)}_{i,j}^{n+1} - \overline{(T_e)}_{i,j}^{n+1} \right) \end{aligned}$$

with:

$$\nu_1 = \frac{(x_2^\alpha)_j k_B \Delta t}{\Delta x_1^2} \quad \text{and} \quad \nu_2 = \frac{k_B \Delta t}{\Delta x_2^2}.$$

Using notations introduced in the previous proof, the off-diagonal matrix terms that operate on T_e write:

$$\begin{aligned} (m_e)_{i\pm1,j}^{i,j} &= -\nu_1 \kappa^\perp_{i\pm\frac{1}{2},j}, \\ (m_e)_{i,j\pm1}^{i,j} &= -\nu_2 (x_2^\alpha)_{j\pm\frac{1}{2}} \kappa^\perp_{i,j\pm\frac{1}{2}}. \end{aligned}$$

The off-diagonal term $(m_{ei})_{i,j}^{i,j}$ that operates on T_i (the coupling term) is given by:

$$(m_{ei})_{i,j}^{i,j} = \frac{k_B m_e}{m_i} \cdot (x_2^\alpha)_j \cdot (n_e)_{i,j}^n \cdot (\nu_{ei})_{i,j}^n.$$

The matrix is thus clearly symmetric since:

$$\begin{aligned} (m_e)_{i\pm1,j}^{i,j} &= (m_e)_{i,j}^{i\pm1,j}, \\ (m_e)_{i,j\pm1}^{i,j} &= (m_e)_{i,j}^{i,j\pm1}, \\ (m_{ei})_{i,j}^{i,j} &= (m_{ie})_{i,j}^{i,j}. \end{aligned}$$

Note that all of these off-diagonal matrix coefficients are negative. Moreover, the diagonal term writes:

$$(m_e)_{i,j}^{i,j} = C_{v_e} (x_2^\alpha)_j (\rho_e)_{i,j}^n - \left((m_e)_{i+1,j}^{i,j} + (m_e)_{i-1,j}^{i,j} + (m_e)_{i,j+1}^{i,j} + (m_e)_{i,j-1}^{i,j} + (m_{ei})_{i,j}^{i,j} \right),$$

it is thus necessarily positive. The matrix that is associated to the thermal conduction solver is consequently diagonally dominant, indeed:

$$\left| (m_e)_{i,j}^{i,j} \right| - \left(\left| (m_e)_{i+1,j}^{i,j} \right| + \left| (m_e)_{i-1,j}^{i,j} \right| + \left| (m_e)_{i,j+1}^{i,j} \right| + \left| (m_e)_{i,j-1}^{i,j} \right| + \left| (m_{ei})_{i,j}^{i,j} \right| \right) = C_{v_e} (x_2^\alpha)_j (\rho_e)_{i,j}^n$$

which is a strictly positive term. Being strictly diagonally dominant with positive diagonal terms, it is thus definite positive. □

7.1.2 Resistive conduction

We now deal with the system of equations that governs resistive conduction. According to the operator splitting presented in chapter 2, it is given by (2.4) and writes:

$$\begin{cases} \partial_t \rho = 0, \\ \partial_t(\rho \mathbf{u}) = 0, \\ \partial_t(\rho_e \epsilon_e) - (\overline{\overline{\rho}} \cdot \mathbf{J}) \cdot \mathbf{J} = 0, \\ \partial_t(\rho_i \epsilon_i) = 0, \\ \partial_t \mathbf{B} + c\, \nabla \times (\overline{\overline{\rho}} \cdot \mathbf{J}) = 0, \end{cases}$$

but we rather consider in this study its total energy formulation (2.5) given below in order to preserve conservativity:

$$\begin{cases} \partial_t \rho = 0, \\ \partial_t(\rho \mathbf{u}) = 0, \\ \partial_t(\rho_i \epsilon_i) = 0, \\ \partial_t(\rho e) + \frac{c}{\mu} \nabla \cdot \big((\overline{\overline{\rho}} \cdot \mathbf{J}) \wedge \mathbf{B} \big) = 0, \\ \partial_t \mathbf{B} + c\, \nabla \times (\overline{\overline{\rho}} \cdot \mathbf{J}) = 0. \end{cases}$$

The density, velocity and ionic internal energy being here constant, we now focus on the magnetic field and total energy evolution equations:

$$\partial_t \mathbf{B} + c\, \nabla \times (\overline{\overline{\rho}} \cdot \mathbf{J}) = 0, \tag{7.6a}$$

$$\partial_t(\rho e) + \frac{c}{\mu} \nabla \cdot \big((\overline{\overline{\rho}} \cdot \mathbf{J}) \wedge \mathbf{B} \big) = 0. \tag{7.6b}$$

We recall that the current density $\mathbf{J}$ writes:

$$\mathbf{J} = \frac{c}{\mu} \nabla \times \mathbf{B}.$$

As suggested in introduction (see page 15), we propose to solve the resistive conduction system thanks to a semi-implicit time scheme, more precisely using the following discretization:

$$\frac{\mathbf{B}^{n+1} - \mathbf{B}^n}{\Delta t} + c\, \nabla \times \big(\overline{\overline{\rho}}^n \cdot \mathbf{J}^{n+1} \big) = 0, \tag{7.7a}$$

$$\frac{(\rho e)^{n+1} - (\rho e)^n}{\Delta t} + \frac{c}{\mu} \nabla \cdot \big((\overline{\overline{\rho}} \cdot \mathbf{J})^{n+1} \wedge \mathbf{B}^{n+1} \big) = 0. \tag{7.7b}$$

Remark 33. Note that the magnetic field and the total energy are not coupled anymore in (7.7a)-(7.7b) due to the semi-implicit time scheme. We therefore first solve the magnetic field equation and then update the total energy using values of the magnetic field at time t^{n+1}.

We now rewrite equations (7.6a)-(7.6b) in the two-dimensional orthogonal MHD framework, *i.e.* with $\mathbf{B} = B\mathbf{e}_3$. In this specific case, the expressions of the $\overline{\overline{\rho}} \cdot \mathbf{J}$ and $(\overline{\overline{\rho}} \cdot \mathbf{J}) \wedge \mathbf{B}$ terms using notations introduced

in Appendix C are respectively given by:

$$\overline{\overline{\rho}}\cdot\mathbf{J}=\frac{c}{\mu}\begin{pmatrix}\frac{\rho^{\perp}}{x_2^{\alpha}}\partial_2\left(x_2^{\alpha}B\right)+\rho^{\wedge}\partial_1 B\\ -\rho^{\perp}\partial_1 B-\frac{\rho^{\wedge}}{x_2^{\alpha}}\partial_2\left(x_2^{\alpha}B\right)\\ 0\end{pmatrix}\quad\text{and}\quad(\overline{\overline{\rho}}\cdot\mathbf{J})\wedge\mathbf{B}=-\frac{cB}{\mu}\begin{pmatrix}\rho^{\perp}\partial_1 B+\frac{\rho^{\wedge}}{x_2^{\alpha}}\partial_2\left(x_2^{\alpha}B\right)\\ \frac{\rho^{\perp}}{x_2^{\alpha}}\partial_2\left(x_2^{\alpha}B\right)+\rho^{\wedge}B\partial_1 B\\ 0\end{pmatrix}.$$

In this context, equations (7.6a) and (7.6b) thus rewrite:

$$\partial_t B+\frac{c^2}{\mu}\left[\partial_1\left(-\rho^{\perp}\partial_1 B-\frac{\rho^{\wedge}}{x_2^{\alpha}}\partial_2\left(x_2^{\alpha}B\right)\right)-\partial_2\left(\frac{\rho^{\perp}}{x_2^{\alpha}}\partial_2\left(x_2^{\alpha}B\right)+\rho^{\wedge}\partial_1 B\right)\right]=0, \tag{7.8a}$$

$$\partial_t(x_2^{\alpha}\rho e)-\frac{c^2}{\mu^2}\left[x_2^{\alpha}\partial_1\left(B\rho^{\perp}\partial_1 B+B\frac{\rho^{\wedge}}{x_2^{\alpha}}\partial_2\left(x_2^{\alpha}B\right)\right)+\partial_2\left(B\rho^{\perp}\partial_2\left(x_2^{\alpha}B\right)+x_2^{\alpha}B\rho^{\wedge}\partial_1 B\right)\right]=0. \tag{7.8b}$$

These will now be discretized using a finite volume method combined with the semi-implicit time discretization introduced in (7.7a)-(7.7b).

Magnetic field equation

At this stage, we integrate equation (7.8a) over $[(x_1)_{i-\frac{1}{2}};(x_1)_{i+\frac{1}{2}}]\times[(x_2)_{j-\frac{1}{2}};(x_2)_{j+\frac{1}{2}}]$ in order to form the following semi-implicit conservative finite volume scheme:

$$\frac{\overline{B}_{i,j}^{n+1}-\overline{B}_{i,j}^{n}}{\Delta t}+\frac{c^2\Delta t}{\mu\Delta x_1}\left(R^*_{i+\frac{1}{2},j}-R^*_{i-\frac{1}{2},j}\right)-\frac{c^2\Delta t}{\mu\Delta x_2}\left(R^*_{i,j+\frac{1}{2}}-R^*_{i,j-\frac{1}{2}}\right)=0 \tag{7.9}$$

where $\overline{B}_{i,j}^{n}$ denotes the *planar* cell average of B over the (i,j) cell at time t^n (note that this choice is coherent with the particular treatment of the B_θ equation in the axisymmetric case, see section 6.1.3):

$$\overline{B}_{i,j}^{n}=\frac{1}{\Delta x_1\Delta x_2}\int_{(x_1)_{i-\frac{1}{2}}}^{(x_1)_{i+\frac{1}{2}}}\int_{(x_2)_{j-\frac{1}{2}}}^{(x_2)_{j+\frac{1}{2}}}B(x_1,x_2,t^n)\,dx_2\,dx_1. \tag{7.10}$$

The resistivity flux is denoted R^*, its value along the x_1 and x_2 directions being respectively given by:

$$\begin{aligned}R^*_{i+\frac{1}{2},j}=&-(\rho^{\perp})^n_{i+\frac{1}{2},j}\left(\frac{B^{n+1}_{i+1,j}-B^{n+1}_{i,j}}{\Delta x_1}\right)\\&-(\rho^{\wedge})^n_{i+\frac{1}{2},j}\left(\frac{(x_2^{\alpha})_{j+1}\left(B^{n+1}_{i+1,j+1}+B^{n+1}_{i,j+1}\right)-(x_2^{\alpha})_{j-1}\left(B^{n+1}_{i+1,j-1}+B^{n+1}_{i,j-1}\right)}{4(x_2^{\alpha})_j\Delta x_2}\right),\\ R^*_{i,j+\frac{1}{2}}=&(\rho^{\perp})^n_{i,j+\frac{1}{2}}\left(\frac{(x_2^{\alpha})_{j+1}B^{n+1}_{i,j+1}-(x_2^{\alpha})_jB^{n+1}_{i,j}}{(x_2^{\alpha})_{j+\frac{1}{2}}\Delta x_2}\right)+(\rho^{\wedge})^n_{i,j+\frac{1}{2}}\left(\frac{B^{n+1}_{i+1,j+1}+B^{n+1}_{i+1,j}-B^{n+1}_{i-1,j+1}-B^{n+1}_{i-1,j}}{4\Delta x_1}\right).\end{aligned}$$

Remark 34. The value of the resistivity flux being imposed by boundary conditions along the $x_2=0$ axis, the $(x_2^{\alpha})_{j+\frac{1}{2}}$ denominator in the expression of $R^*_{i,j+\frac{1}{2}}$ does not lead to pathologic approximations in the axisymmetric case.

The cell interface representations of the resistivity tensor have been determined so that the resistivity flux is continuous across cell interfaces as explained in Appendix D. They are given by:

$$\rho^{\perp}_{i+\frac{1}{2},j} = 2\frac{\rho^{\perp}_{i,j}\cdot\rho^{\perp}_{i+1,j}}{\rho^{\perp}_{i,j}+\rho^{\perp}_{i+1,j}}, \qquad \rho^{\wedge}_{i+\frac{1}{2},j} = \frac{\rho^{\perp}_{i,j}\cdot\rho^{\wedge}_{i+1,j}+\rho^{\wedge}_{i,j}\cdot\rho^{\perp}_{i+1,j}}{\rho^{\perp}_{i,j}+\rho^{\perp}_{i+1,j}},$$
$$\rho^{\perp}_{i,j+\frac{1}{2}} = 2\frac{\rho^{\perp}_{i,j}\cdot\rho^{\perp}_{i,j+1}}{\rho^{\perp}_{i,j}+\rho^{\perp}_{i,j+1}}, \qquad \rho^{\wedge}_{i,j+\frac{1}{2}} = \frac{\rho^{\perp}_{i,j}\cdot\rho^{\wedge}_{i,j+1}+\rho^{\wedge}_{i,j}\cdot\rho^{\perp}_{i,j+1}}{\rho^{\perp}_{i,j}+\rho^{\perp}_{i,j+1}}.$$

Solving the evolution equation (7.8a) using the finite volume scheme (7.9) finally consists in solving a linear system whose unknowns are the cell averages of B. As in the case of thermal conduction, this linear system does not seem to have any interesting property that could help its numerical resolution. It is in particular unsymmetric. We therefore resort again to a BiCGStab algorithm for solving the resistive conduction system.

Total energy equation

We finally build a semi-implicit finite volume scheme for the total energy equation. To that end, we integrate equation (7.8b) over $[(x_1)_{i-\frac{1}{2}};(x_1)_{i+\frac{1}{2}}]\times[(x_2)_{j-\frac{1}{2}};(x_2)_{j+\frac{1}{2}}]$ and thus obtain the following conservative numerical scheme that can be used to compute updated values of the total energy once the magnetic field equation has been solved:

$$\frac{\overline{(\rho e)}^{n+1}_{i,j}-\overline{(\rho e)}^{n}_{i,j}}{\Delta t}-\frac{c^2}{\mu^2}\left[\frac{1}{\Delta x_1}\left(E^*_{i+\frac{1}{2},j}-E^*_{i-\frac{1}{2},j}\right)+\frac{1}{(x_2^\alpha)_j\Delta x_2}\left(E^*_{i,j+\frac{1}{2}}-E^*_{i,j-\frac{1}{2}}\right)\right]=0. \tag{7.11}$$

Here, $\overline{(\rho e)}^{n}_{i,j}$ denotes the following cell average of (ρe) over the (i,j) cell at time t^n:

$$\overline{(\rho e)}^{n}_{i,j}=\frac{1}{(x_2^\alpha)_j\Delta x_2}\int_{(x_2)_{j-\frac{1}{2}}}^{(x_2)_{j+\frac{1}{2}}}x_2^\alpha\left[\frac{1}{\Delta x_1}\int_{(x_1)_{i-\frac{1}{2}}}^{(x_1)_{i+\frac{1}{2}}}(\rho e)(x_1,x_2,t^n)\,dx_1\right]dx_2.$$

E^* denotes the total energy flux, its value along the x_1 and x_2 direction is respectively given by:

$$\begin{aligned}
E^*_{i+\frac{1}{2},j} &= (\rho^{\perp})^n_{i+\frac{1}{2},j}\left(\frac{B^{n+1}_{i+1,j}+B^{n+1}_{i,j}}{2}\right)\left(\frac{B^{n+1}_{i+1,j}-B^{n+1}_{i,j}}{\Delta x_1}\right)\\
&\quad (\rho^{\wedge})^n_{i+\frac{1}{2},j}\left(\frac{B^{n+1}_{i+1,j}+B^{n+1}_{i,j}}{2}\right)\left(\frac{(x_2^\alpha)_{j+1}\left(B^{n+1}_{i+1,j+1}+B^{n+1}_{i,j+1}\right)-(x_2^\alpha)_{j-1}\left(B^{n+1}_{i+1,j-1}+B^{n+1}_{i,j-1}\right)}{4(x_2^\alpha)_j\Delta x_2}\right),\\
E^*_{i,j+\frac{1}{2}} &= (\rho^{\perp})^n_{i,j+\frac{1}{2}}\left(\frac{(x_2^\alpha)_{j+1}B^{n+1}_{i,j+1}+(x_2^\alpha)_jB^{n+1}_{i,j}}{2}\right)\left(\frac{(x_2^\alpha)_{j+1}B^{n+1}_{i,j+1}-(x_2^\alpha)_jB^{n+1}_{i,j}}{(x_2^\alpha)_{j+\frac{1}{2}}\Delta x_2}\right)\\
&\quad (\rho^{\wedge})^n_{i,j+\frac{1}{2}}\left(\frac{(x_2^\alpha)_{j+1}B^{n+1}_{i,j+1}+(x_2^\alpha)_jB^{n+1}_{i,j}}{2}\right)\left(\frac{B^{n+1}_{i+1,j+1}+B^{n+1}_{i+1,j}-B^{n+1}_{i-1,j+1}-B^{n+1}_{i-1,j}}{4\Delta x_1}\right).
\end{aligned}$$

Remark 35. The value of the E^* flux being imposed by boundary conditions along the $x_2=0$ axis, the $(x_2^\alpha)_{j+\frac{1}{2}}$ denominator in the expression of $E^*_{i,j+\frac{1}{2}}$ does not lead to pathologic approximations in the axisymmetric case.

Joule heating

As indicated previously, we here solve the total energy formulation of the resistive conduction system instead of its internal energy formulation in order to preserve conservativity. This choice departs from classical approaches which usually deal with the electronic internal energy equation:

$$\partial_t(\rho_e\epsilon_e) = (\overline{\overline{\rho}}\cdot\mathbf{J})\cdot\mathbf{J}. \tag{7.12}$$

Equation (7.12) represents the Joule heating phenomenon: due to resistivity, a part of the magnetic energy is transformed into heat and increases the electronic temperature. The electronic internal energy variation is indeed positive since:

$$(\overline{\overline{\rho}}\cdot\mathbf{J})\cdot\mathbf{J} = \begin{pmatrix} \rho^{\perp}J_1 - \rho^{\wedge}J_2 \\ \rho^{\perp}J_2 + \rho^{\wedge}J_1 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} J_1 \\ J_2 \\ 0 \end{pmatrix} = \rho^{\perp}J_1^2 + \rho^{\perp}J_2^2.$$

This property should be preserved by the considered discretization. In our case, the electronic internal energy is updated from the total energy and magnetic field values:

$$(\rho_e\epsilon_e)^{n+1} = (\rho e)^{n+1} - \frac{\left(B^{n+1}\right)^2}{2\mu}$$

and we state the following proposition.

Proposition 14. *If the magnetic field and total energy are updated using the finite volume schemes* (7.9) *and* (7.11) *and if the resistivity tensor is diagonal (i.e. if the $\rho^{\wedge}$ component is identically zero), the electronic internal energy variation given by*

$$(\rho_e\epsilon_e)^{n+1} - (\rho_e\epsilon_e)^{n} = (\rho e)^{n+1} - (\rho e)^{n} - \left(\frac{\left(B^{n+1}\right)^2}{2\mu} - \frac{(B^{n})^2}{2\mu}\right) \tag{7.13}$$

is positive.

The proof of this proposition is given in Appendix E. If the resistivity tensor is not diagonal, we show in Appendix E that the electronic internal energy variation (7.13) may be positive provided that the space step is small enough and that both the $\rho^{\wedge}$ component and the magnetic field are smooth enough.

7.1.3 Diffusion schemes implementation

The diffusion schemes that have been presented in this chapter rely on semi-implicit time schemes. Dealing with diffusion operators therefore consists in solving a linear system whose size S is equal to:

$$S = \text{number of cells} \times \text{number of variables}$$

The number of variables is equal to two for the thermal conduction operator (the T_i and T_e temperatures) and to one for the resistive conduction term (the third magnetic field component). The size S being possibly very large (we have reached up to $S \approx 1.3 \cdot 10^8$ in numerical experiments), these linear systems will be implemented using sparse storage schemes [81].

All in all, the resolution of the thermal and resistive conduction terms requires a parallel sparse linear system solver which is able to deal with large problems and to provide a high performance level. Several existing linear algebra libraries may fit our needs but none of the ones we have examined revealed suitable. We have therefore chosen to develop our own iterative solvers, more precisely CG and BiCGStab algorithms whose descriptions can for example be found in [74]. We give in this section a few details about the practical implementations of these solvers.

Matrix storage format

The sparse matrices' implementation is based on the so-called sparse coordinate format. In this case, matrices are represented by three tables that respectively store the row index, the column index and the value of non-zero coefficient.

Iterative solvers implementation

Iterative solvers mainly rely on two linear algebra functions: the matrix vector product and the dot product. In order to get a high performance level, these are performed using Intel's Math Kernel Library [47] that provides very efficient implementations of the matrix vector and dot products. More precisely, we use the sequential `mkl_dcoomv` (double precision matrix vector product) and `ddot` (double precision dot product) functions. The MKL functions have then been adapted to the domain decomposition method we resort to.

- The matrix vector product is preceded by a boundary conditions call that fill ghost cells layers for all artificial boundaries (*i.e.* for boundaries that are shared by two processes, not for the physical ones). It simply consists in a data sharing step (using `MPI_Send` / `MPI_Recv` instructions) with immediate neighbours. This allows each process to perform a local matrix vector product on variables that correspond to its own physical cells. The linear system is thus completely distributed - the only redundancy indeed comes from ghost cell layers - and our implementation is consequently eligible for massively parallel architectures.

- The dot product is followed by a collective reduction operation (using a `MPI_Allreduce` instruction) so that the dot product is computed over the whole domain and known on each process.

Preconditioning

Iterative solvers are combined with a Jacobi preconditioner which is not the most efficient one but presents the advantage of being very cheap (about 2% of the total computation time) and completely scalable. Indeed, in our implementation, no communication step is needed for applying the Jacobi preconditioner.

The preconditioning step should nevertheless be improved in future works in order to reduce the number of iterations in iterative solvers and thus increase performance. Several sophisticated preconditioners can be found in the related literature [89] (incomplete LU factorization, SSOR technique, multigrid preconditioner): these should be implemented and evaluated in terms of appropriateness to massively parallel computer architectures.

7.1.4 Accuracy of semi-implicit diffusion schemes

We discuss in this section the accuracy of the diffusion schemes that have been presented previously for the thermal and resistive conduction systems. For the sake of simplicity, we here consider the following generic diffusion equation in the two-dimensional planar case:

$$\partial_t T + \nabla \cdot \mathbf{q} = f \quad \text{with} \quad \mathbf{q} = -\overline{\overline{\kappa}} \cdot \nabla T. \tag{7.14}$$

Note that the explanations given below also hold for the resistive conduction system, we therefore do not discuss the accuracy of resistive conduction schemes separately.

Applying the methodology that has been introduced in section 7.1.1, one obtains the following finite volume discretization of the diffusion equation (7.14):

$$\frac{\overline{T}_{i,j}^{n+1}-\overline{T}_{i,j}^{n}}{\Delta t}+\frac{1}{\Delta x}\left(q_{i+\frac{1}{2},j}^{*}-q_{i-\frac{1}{2},j}^{*}\right)+\frac{1}{\Delta y}\left(q_{i,j+\frac{1}{2}}^{*}-q_{i,j-\frac{1}{2}}^{*}\right)=\overline{f}_{i,j}^{n+1} \tag{7.15}$$

where the approximate flux q^* is respectively given on vertical and horizontal cell interfaces by:

$$q_{i+\frac{1}{2},j}^{*}=-(\kappa^{\perp})_{i+\frac{1}{2},j}^{n}\left(\frac{T_{i+1,j}^{n+1}-T_{i,j}^{n+1}}{\Delta x}\right)+(\kappa^{\wedge})_{i+\frac{1}{2},j}^{n}\left(\frac{T_{i+1,j+1}^{n+1}+T_{i,j+1}^{n+1}-T_{i+1,j-1}^{n+1}-T_{i,j-1}^{n+1}}{4\Delta y}\right),$$
$$q_{i+\frac{1}{2},j}^{*}=-(\kappa^{\perp})_{i,j+\frac{1}{2}}^{n}\left(\frac{T_{i,j+1}^{n+1}-T_{i,j}^{n+1}}{\Delta y}\right)+(\kappa^{\wedge})_{i,j+\frac{1}{2}}^{n}\left(\frac{T_{i+1,j+1}^{n+1}+T_{i+1,j}^{n+1}-T_{i-1,j+1}^{n+1}-T_{i-1,j}^{n+1}}{4\Delta x}\right).$$

The finite volume scheme (7.14) is theoretically first-order accurate because of the semi-implicit time discretization which only allows to achieve first-order accuracy in time. In order to improve this point, one would have to consider the following discretization of equation (7.14):

$$\frac{\overline{T}_{i,j}^{n+1}-\overline{T}_{i,j}^{n}}{\Delta t}+\frac{1}{\Delta x}\left(q_{i+\frac{1}{2},j}^{*,2}-q_{i-\frac{1}{2},j}^{*,2}\right)+\frac{1}{\Delta y}\left(q_{i,j+\frac{1}{2}}^{*,2}-q_{i,j-\frac{1}{2}}^{*,2}\right)=\frac{1}{2}\left(\overline{f}_{i,j}^{n}+\overline{f}_{i,j}^{n+1}\right),$$

where the second-order accurate approximate heat flux $q^{*,2}$ on vertical cell interfaces (the expression of the approximate heat flux on horizontal cell interfaces being similar, we do not detail it) is given by:

$$\begin{aligned}q_{i+\frac{1}{2}}^{*,2}=&\frac{1}{2}\left[-(\kappa^{\perp})_{i+\frac{1}{2},j}^{n}\left(\frac{T_{i+1,j}^{n}-T_{i,j}^{n}}{\Delta x}\right)+(\kappa^{\wedge})_{i+\frac{1}{2},j}^{n}\left(\frac{T_{i+1,j+1}^{n}+T_{i,j+1}^{n}-T_{i+1,j-1}^{n}-T_{i,j-1}^{n}}{4\Delta y}\right)\right]+\\&\frac{1}{2}\left[-(\kappa^{\perp})_{i+\frac{1}{2},j}^{n+1}\left(\frac{T_{i+1,j}^{n+1}-T_{i,j}^{n+1}}{\Delta x}\right)+(\kappa^{\wedge})_{i+\frac{1}{2},j}^{n+1}\left(\frac{T_{i+1,j+1}^{n+1}+T_{i,j+1}^{n+1}-T_{i+1,j-1}^{n+1}-T_{i,j-1}^{n+1}}{4\Delta y}\right)\right].\end{aligned}$$

In this case, computing T at time t^{n+1} would suppose to solve a non-linear system of equations since $\overline{\overline{\kappa}}$ generally depends on T (see remark 30). This could for example be done by first linearizing the non-linear system with the help of a Newton-Raphson method and then solving the resulting linear system. This approach seems to be quite easy to set up if the conductivity tensor is a diagonal tensor but its implementation reveals tricky otherwise, in particular in our case due to the complexity of Braginskii's conductivity tensors expressions. It may also considerably increase the computational cost of the thermal conduction scheme. This is why we restrict in this study to first-order accurate semi-implicit discretizations.

Remark 36. The above described method for achieving second-order accuracy considerably differs in terms of implementation of the semi-implicit approach we adopted. In order to keep a clear and easily maintainable code, it therefore has not been implemented, even in simple cases like the two diffusion problems that are considered in section 7.1.5.

7.1.5 Numerical results

We conclude this section with some numerical results that aim at validating the diffusion schemes presented previously. We focus in particular on two smooth diffusion problems and carry out a convergence analysis. Note that the conductivity tensor is diagonal for both of these problems: the linear system that comes from the semi-implicit time discretization is thus symmetric definite positive thanks to Proposition 13 in these cases and it is solved using a CG algorithm.

Analytical solution to a thermal conduction equation

In this section, we deal with the following thermal conduction equation in two-dimensional axisymmetric geometry:

$$\partial_t T - \nabla \cdot \left(\nu T^{\nu-1} \nabla T\right) = 0. \tag{7.16}$$

Let $\rho = \sqrt{z^2 + r^2}$. A spherically symmetric analytical solution to equation (7.16) is given in [60], it writes:

$$T(\rho, t) = \frac{\phi(\rho, t)}{\lambda(t)^3} \quad \text{with} \quad \phi(\rho, t) = \left[1 - \left(\frac{\rho}{\lambda(t)}\right)^2\right]^{\frac{1}{\nu-1}} \quad \text{and} \quad \lambda(t) = \left(\frac{2\nu(3\nu-1)}{\nu-1} t + 1\right)^{\frac{1}{3\nu-1}}.$$

We here take $\nu = 3.5$ and set this test problem on the $(z, r) \in [0; 1]^2$ computational domain (the analytical solution is therefore well defined since $\rho/\lambda(t) \leq 1 \; \forall t$). with Neumann boundary conditions along the $z = 0$ and $r = 0$ axis and Dirichlet boundary conditions elsewhere (in this case, the exact solution is imposed along boundaries). We have carried out a convergence analysis: calculations have been performed until time $t = 0.1$ with the two-dimensional axisymmetric thermal conduction scheme on different meshes and the error between the analytical and numerical solutions has been measured using a L^1 norm in space and time. The time step is set to $\Delta t = 10^{-3}$ for the computation on 100×100 cells and is divided by 2 each time the mesh is refined. The error measurements and the corresponding experimental order of convergence have been summarized in Table 7.1. They show that the thermal conduction scheme is first-order accurate, which matches expectations as explained in section 7.1.4.

Analytical solution to a resistive conduction problem

We now deal with the following resistive conduction equation in two-dimensional planar geometry:

$$\partial_t \mathbf{B} + \nabla \times (\nabla \times \mathbf{B}) = 0. \tag{7.17}$$

An analytical solution to equation (7.17) is given by:

$$\mathbf{B} = \exp(-t) \sin(2\pi x) \sin(2\pi y) \, \mathbf{e}_z.$$

We have set this problem on the $(x, y) \in [0; 1]^2$ computational domain with homogeneous Dirichlet boundary conditions and we have carried out a convergence analysis using the procedure described for the previous benchmark. Error measurements between the exact and numerical solutions as well as the corresponding experimental order of convergence have been reported in Table 7.1. Once again, they exhibit a first-order convergence which matches expectations as explained in section 7.1.4.

A non-smooth anisotropic diffusion problem in planar geometry

We finally consider a diffusion test problem proposed by Hermeline [44] that is associated to the following anisotropic diffusion equation in planar geometry:

$$\partial_t u - \nabla \cdot (\kappa(u) \nabla u) = 0 \quad \text{with} \quad \kappa(u) = \begin{pmatrix} \kappa_1(x, y) & \kappa_2(x, y) u \\ -\kappa_2(x, y) u & \kappa_1(x, y) \end{pmatrix}$$

where the κ_1 and κ_2 values are given by:

$$(\kappa_1, \; \kappa_2) = \begin{cases} (10^{-2}, \; 10^{-1}) & \text{if } x \leq 0.5, \\ (10^{-3}, \; 10^{-5}) & \text{if } x > 0.5. \end{cases}$$

DIFFUSION SCHEMES				
Mesh size	**Thermal conduction**		**Resistive conduction**	
(N × N cells)	error	order	error	order
N = 100	2.61e-03		2.26e-05	
N = 200	1.36e-03	0.95	1.13e-05	1.00
N = 400	6.83e-04	0.99	5.72e-06	0.99
N = 800	3.46e-04	0.99	2.87e-06	0.99
N = 1600	1.74e-04	0.99	1.44e-06	0.99

Table 7.1: Error measured using a L^1 norm in space and time between the analytical and numerical solutions for the smooth thermal conduction and resistive conduction problems respectively described page 174 and page 174.

Due to the anisotropic nature of the $\kappa(u)$ tensor, the linear system that comes from the semi-implicit time discretization is non-symmetric. We therefore resort in this case to a BiCGStab iterative solver. As in [44], we consider the following initial and boundary conditions:

$$\text{IC} \begin{cases} u(t=0) = -1 & \text{if } y \geq 0.5, \\ u(t=0) = 0 & \text{if } y < 0.5, \end{cases} \qquad \text{BC} \begin{cases} u = -1 & \text{along the } y = 1 \text{ boundary}, \\ \nabla u \cdot \mathbf{n} = 0 & \text{along other boundaries}, \end{cases}$$

and set this test problem on the $[0;1]^2$ computational domain. Figure 7.1 presents numerical results obtained at time $t = 0.5$ on 100×100, 200×200, 400×400 and 800×800 cells with the finite volume thermal conduction scheme. These show a good agreement with the numerical results presented in [44]. Figure 7.2 shows a slice of the previous results along the $y = 0.4$ axis: it exhibits nice convergence properties.

7.2 Source terms treatment

We deal in this section with the source terms equations (2.7) that are responsible for the magnetic field generation and govern the Hall and Nernst effects. These write:

$$\begin{cases} \partial_t \rho = 0, \\ \partial_t (\rho \mathbf{u}) = 0, \\ \partial_t (\rho_i \epsilon_i) = 0, \\ \partial_t \mathbf{B} - c\, \nabla \times \mathbf{S} = 0, \\ \partial_t (\rho e) - \nabla \cdot \left(\frac{p_e + \rho_e \epsilon_e}{e n_e} \mathbf{J} + \frac{c}{\mu} \mathbf{S} \wedge \mathbf{B} \right) = 0, \end{cases}$$

where $\mathbf{S} = \mathbf{S}_{\text{self}} + \mathbf{S}_{\text{Hall}} + \mathbf{S}_{\text{Nernst}}$. We recall the definitions of the $\mathbf{S}_{\text{self}}$ (magnetic field generation), $\mathbf{S}_{\text{Hall}}$ (Hall effect) and $\mathbf{S}_{\text{Nernst}}$ (Nernst effect) terms below:

$$\mathbf{S}_{\text{self}} = \frac{1}{e n_e} \nabla p_e, \quad \mathbf{S}_{\text{Hall}} = -\frac{1}{e c n_e} \mathbf{J} \wedge \mathbf{B}, \quad \mathbf{S}_{\text{Nernst}} = \frac{1}{e} \overline{\overline{\beta}} \cdot \nabla \left(k_B T_e \right).$$

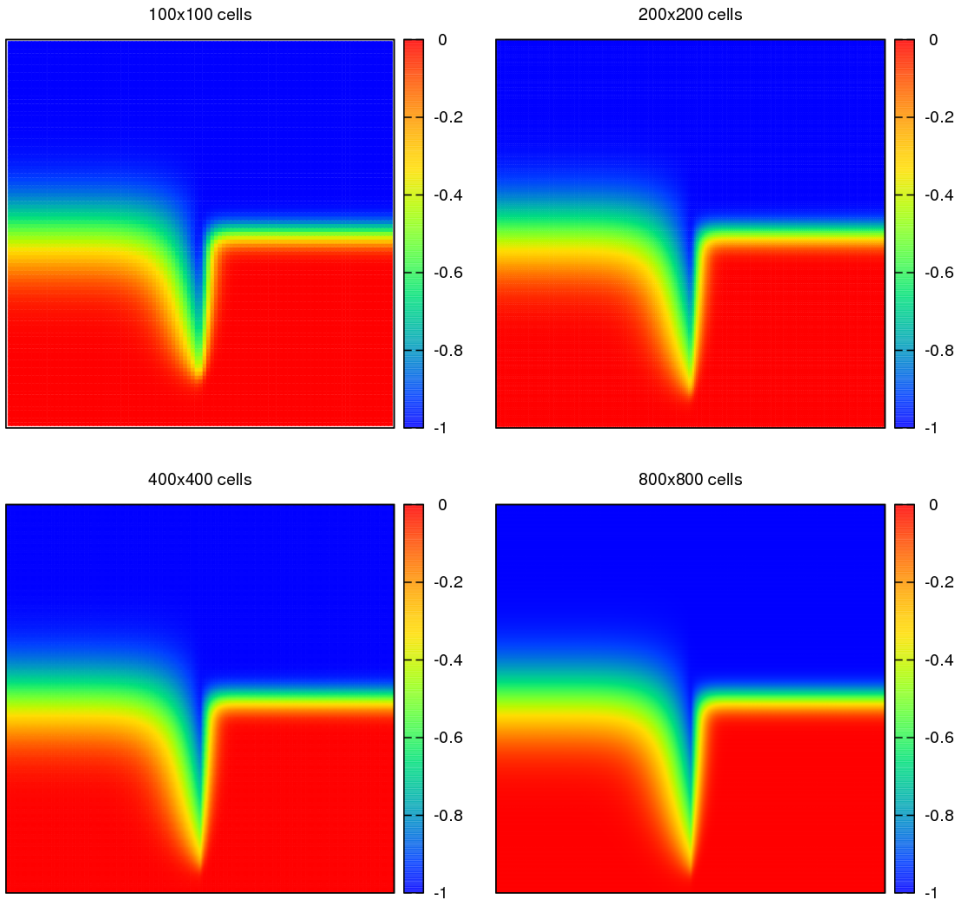


Figure 7.1: Numerical solution of Hermeline's diffusion problem at time $t = 0.5$ on several mesh sizes.

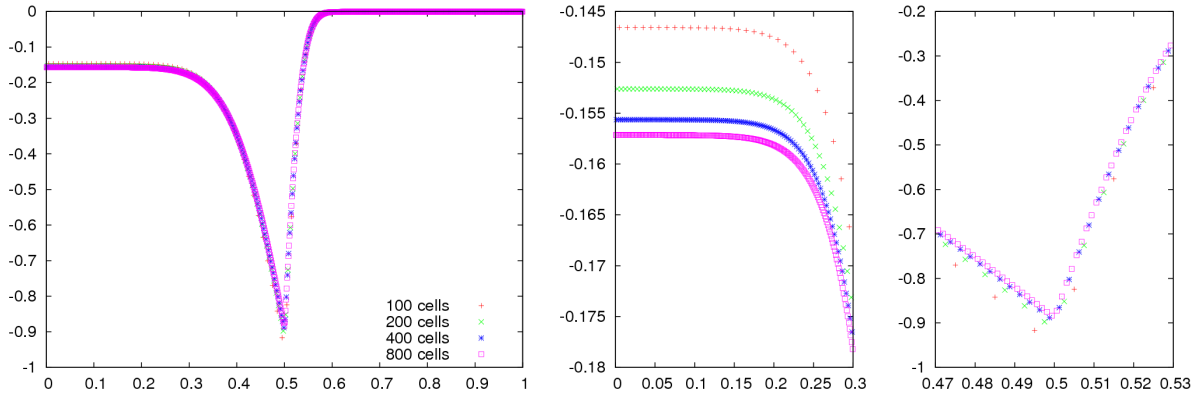


Figure 7.2: Slice along the $y = 0.4$ axis on the numerical solution of Hermeline's diffusion problem at time $t = 0.5$. From left to right: whole computational domain, zoom on the $[0; 0.3]$ region and zoom on the $[0.47; 0.53]$ region.

Remark 37. As in the case of resistive conduction, we consider the total energy formulation for the source terms equations. This approach departs from classical internal energy formulations (see [9] for example) but allows to preserve conservativity. In this context, once the magnetic field and total energy equations have been solved, the electronic internal energy is updated using the following formula:

$$(\rho_e \epsilon_e)^{n+1} = (\rho e)^{n+1} - \frac{\mathbf{B}^{n+1} \cdot \mathbf{B}^{n+1}}{2\mu},$$

since $\partial_t(\rho_i \epsilon_i) = 0$.

The density, velocity and ionic internal energy being here constant, we now focus on the magnetic field and total energy evolution equations:

$$\partial_t \mathbf{B} - c\, \nabla \times \mathbf{S} = 0, \tag{7.18a}$$

$$\partial_t(\rho e) - \nabla \cdot \left(\frac{p_e + \rho_e \epsilon_e}{e n_e} \mathbf{J} + \frac{c}{\mu} \mathbf{S} \wedge \mathbf{B} \right) = 0. \tag{7.18b}$$

In the two-dimensional case with $\mathbf{B} = B\mathbf{e}_3$ and using notations introduced in Appendix C, the $\mathbf{S}_{\text{self}}$, $\mathbf{S}_{\text{Hall}}$ and $\mathbf{S}_{\text{Nernst}}$ terms rewrite:

$$\mathbf{S}_{\text{self}} = \frac{1}{e n_e} \begin{pmatrix} \partial_1 p_e \\ \partial_2 p_e \\ 0 \end{pmatrix}, \quad \mathbf{S}_{\text{Hall}} = \frac{B}{e \mu n_e} \begin{pmatrix} \partial_1 B \\ \frac{1}{x_2^\alpha} \partial_2 (x_2^\alpha B) \\ 0 \end{pmatrix}, \quad \mathbf{S}_{\text{Nernst}} = \frac{k_B}{e} \begin{pmatrix} \beta^\perp \partial_1 T_e - \beta^\wedge \partial_2 T_e \\ \beta^\perp \partial_2 T_e + \beta^\wedge \partial_1 T_e \\ 0 \end{pmatrix}.$$

In this context, the magnetic field and total energy equations are given by:

$$\partial_t B - c\partial_1 (\mathbf{S} \cdot \mathbf{e}_2) + c\partial_2 (\mathbf{S} \cdot \mathbf{e}_1) = 0, \tag{7.19a}$$

$$\partial_t(\rho e) - \partial_1 \left[\left(\frac{p_e + \rho_e \epsilon_e}{e n_e} \mathbf{J} + \frac{c}{\mu} \mathbf{S} \wedge \mathbf{B} \right) \cdot \mathbf{e}_1 \right] - \frac{1}{x_2^\alpha} \partial_2 \left[x_2^\alpha \left(\frac{p_e + \rho_e \epsilon_e}{e n_e} \mathbf{J} + \frac{c}{\mu} \mathbf{S} \wedge \mathbf{B} \right) \cdot \mathbf{e}_2 \right] = 0. \tag{7.19b}$$

We now describe how these equations have been discretized. As suggested in introduction (see page 15), an explicit time scheme will be used.

7.2.1 Magnetic field equation

We deal in this section with the the magnetic field equation. We first discuss the discretization of the magnetic field generation source term and then extend the chosen approach to the complete equation.

Discretization of the magnetic field generation source term

We here focus on the following evolution equation for the magnetic field where we have neglected the Hall and Nernst effects:

$$\partial_t \mathbf{B} - c \nabla \times \mathbf{S}_{\text{self}} = 0. \tag{7.20}$$

Practically, self-generated magnetic field terms will appear as soon as ∇n_e and ∇p_e are not colinear since:

$$\nabla \times \mathbf{S}_{\text{self}} = \nabla \times \left(\frac{1}{e n_e} \nabla p_e \right) = -\frac{1}{e n_e^2} \nabla n_e \wedge \nabla p_e. \tag{7.21}$$

In the case of two-dimensional geometries, the only possibly non-zero component of $\nabla n_e \wedge \nabla p_e$ is the one that is orthogonal to the computational domain, namely the z component in the planar case and the θ

component in the axisymmetric case. This is what motivates our study of the specific case $\mathbf{B} = B\mathbf{e}_3$. Indeed, one can show that if $\mathbf{B}$ is initially zero, the only magnetic field component that may appear when solving the complete system of equations presented in chapter 1 is the orthogonal one.

The relation given by (7.21) naturally raises the question of the $\nabla \times \mathbf{S}_{\text{self}}$ term's discretization. Indeed, it seems a priori interesting to preserve the property of avoiding magnetic field generation when ∇n_e and ∇p_e are colinear at the discrete level. This can for example be achieved with the followig non-conservative finite difference discretization of equation (7.20) which is the one used in [9]:

$$\frac{B_{i,j}^{n+1} - B_{i,j}^{n}}{\Delta t} + \frac{c}{e\,(n_e^2)_{i,j}^{n}} \left[(\partial_1 n_e)_{i,j}^{n} \cdot (\partial_2 p_e)_{i,j}^{n} - (\partial_1 p_e)_{i,j}^{n} \cdot (\partial_2 n_e)_{i,j}^{n} \right] = 0 \tag{7.22}$$

where:

$$(\partial_1 \phi)_{i,j}^{n} = \frac{\phi_{i+1,j}^{n} - \phi_{i-1,j}^{n}}{\Delta x_1} \quad \text{and} \quad (\partial_2 \phi)_{i,j}^{n} = \frac{\phi_{i,j+1}^{n} - \phi_{i,j-1}^{n}}{\Delta x_2}. \tag{7.23}$$

The finite difference scheme (7.22) indeed clearly leads to $B_{i,j}^{n+1} = B_{i,j}^{n}$ if:

$$\begin{pmatrix} (\partial_1 n_e)_{i,j}^{n} \\ (\partial_2 n_e)_{i,j}^{n} \end{pmatrix} = \lambda \begin{pmatrix} (\partial_1 p_e)_{i,j}^{n} \\ (\partial_2 p_e)_{i,j}^{n} \end{pmatrix}.$$

Nevertheless, this property heavily depends on the gradient operator's discretization. Practically, it turns out in numerical experiments for which ∇n_e and ∇p_e are theoretically colinear that the magnetic field only remains zero in regions where the discrete gradients given by (7.23) are aligned with the mesh, *i.e.* along the horizontal, vertical and 45° directions. In all other directions, magnetic field terms that are only due to numerical approximations appear. There is consequently no need to use the finite difference scheme (7.22) and sacrifice conservativity.

Another way to proceed - which actually is the one that we have adopted - consists in resorting to a classical conservative finite volume discretization of (7.20) which does not guarantee that the magnetic field will remain zero if ∇n_e and ∇p_e are colinear:

$$\frac{\overline{B}_{i,j}^{n+1} - \overline{B}_{i,j}^{n}}{\Delta t} - \frac{c}{\Delta x_1} \left((S_{\text{self}})_{i+\frac{1}{2},j}^{*} - (S_{\text{self}})_{i-\frac{1}{2},j}^{*} \right) + \frac{c}{\Delta x_2} \left((S_{\text{self}})_{i,j+\frac{1}{2}}^{*} - (S_{\text{self}})_{i,j-\frac{1}{2}}^{*} \right) = 0. \tag{7.24}$$

Here, $\overline{B}_{i,j}^{n}$ denotes the following planar cell average of B over the (i,j) cell at time t^n:

$$\overline{B}_{i,j}^{n} = \frac{1}{\Delta x_1 \Delta x_2} \int_{(x_1)_{i-\frac{1}{2}}}^{(x_1)_{i+\frac{1}{2}}} \int_{(x_2)_{j-\frac{1}{2}}}^{(x_2)_{j+\frac{1}{2}}} B(x_1, x_2, t^n)\, dx_2\, dx_1,$$

and the approximate fluxes $(S_{\text{self}})_{i+\frac{1}{2},j}^{*}$ and $(S_{\text{self}})_{i,j+\frac{1}{2}}^{*}$ are given by:

$$\begin{aligned}
(S_{\text{self}})_{i+\frac{1}{2},j}^{*} &\approx (\mathbf{S}_{\text{self}} \cdot \mathbf{e}_2)_{i+\frac{1}{2},j}, \\
&= \frac{1}{e} \left(\frac{2}{(n_e)_{i,j}^{n} + (n_e)_{i+1,j}^{n}} \right) \left(\frac{(p_e)_{i,j+1}^{n} + (p_e)_{i+1,j+1}^{n} - (p_e)_{i,j-1}^{n} - (p_e)_{i,j-1}^{n}}{4\Delta x_2} \right), \\
(S_{\text{self}})_{i,j+\frac{1}{2}}^{*} &\approx (\mathbf{S}_{\text{self}} \cdot \mathbf{e}_1)_{i,j+\frac{1}{2}}, \\
&= \frac{1}{e} \left(\frac{2}{(n_e)_{i,j}^{n} + (n_e)_{i,j+1}^{n}} \right) \left(\frac{(p_e)_{i+1,j}^{n} + (p_e)_{i+1,j+1}^{n} - (p_e)_{i-1,j}^{n} - (p_e)_{i-1,j+1}^{n}}{4\Delta x_1} \right).
\end{aligned}$$

Discretization of the complete magnetic field equation

Applying the finite volume discretization to the complete magnetic field equation (7.19a) leads to the following numerical scheme:

$$\frac{\overline{B}_{i,j}^{n+1} - \overline{B}_{i,j}^{n}}{\Delta t} - \frac{c}{\Delta x_1}\left(S^*_{i+\frac{1}{2},j} - S^*_{i-\frac{1}{2},j}\right) + \frac{c}{\Delta x_2}\left(S^*_{i,j+\frac{1}{2}} - S^*_{i,j-\frac{1}{2}}\right) = 0.$$

Introducing:

$$\begin{aligned}
(S_{\text{Hall}})^*_{i+\frac{1}{2},j} &\approx (\mathbf{S}_{\text{Hall}}\cdot\mathbf{e}_2)_{i+\frac{1}{2},j},\\
&= \frac{1}{e\mu}\left(\frac{B^n_{i,j}+B^n_{i+1,j}}{(n_e)^n_{i,j}+(n_e)^n_{i+1,j}}\right)\left(\frac{(x_2^\alpha)_{j+1}\left(B^n_{i+1,j+1}+B^n_{i,j+1}\right)-(x_2^\alpha)_{j-1}\left(B^n_{i+1,j-1}+B^n_{i,j-1}\right)}{4(x_2^\alpha)_j\Delta x_2}\right),\\
(S_{\text{Hall}})^*_{i,j+\frac{1}{2}} &\approx (\mathbf{S}_{\text{Hall}}\cdot\mathbf{e}_1)_{i,j+\frac{1}{2}},\\
&= \frac{1}{e\mu}\left(\frac{B^n_{i,j}+B^n_{i,j+1}}{(n_e)^n_{i,j}+(n_e)^n_{i,j+1}}\right)\left(\frac{B^n_{i+1,j+1}+B^n_{i+1,j}-B^n_{i-1,j+1}-B^n_{i-1,j}}{4\Delta x_1}\right),\\
(S_{\text{Nernst}})^*_{i+\frac{1}{2},j} &\approx (\mathbf{S}_{\text{Nernst}}\cdot\mathbf{e}_2)_{i+\frac{1}{2},j},\\
&= +\frac{k_B}{e}(\beta^\perp)^n_{i+\frac{1}{2},j}\left(\frac{(T_e)^n_{i+1,j+1}+(T_e)^n_{i,j+1}-(T_e)^n_{i+1,j-1}-(T_e)^n_{i,j-1}}{4\Delta x_2}\right)\\
&\quad+\frac{k_B}{e}(\beta^\wedge)^n_{i+\frac{1}{2},j}\left(\frac{(T_e)^n_{i+1,j}-(T_e)^n_{i,j}}{\Delta x_1}\right),\\
(S_{\text{Nernst}})^*_{i,j+\frac{1}{2}} &\approx (\mathbf{S}_{\text{Nernst}}\cdot\mathbf{e}_1)_{i,j+\frac{1}{2}},\\
&= +\frac{k_B}{e}(\beta^\perp)^n_{i,j+\frac{1}{2}}\left(\frac{(T_e)^n_{i+1,j+1}+(T_e)^n_{i+1,j}-(T_e)^n_{i-1,j+1}-(T_e)^n_{i-1,j}}{4\Delta x_1}\right)\\
&\quad-\frac{k_B}{e}(\beta^\wedge)^n_{i,j+\frac{1}{2}}\left(\frac{(T_e)^n_{i,j+1}-(T_e)^n_{i,j}}{\Delta x_2}\right),
\end{aligned}$$

the approximate fluxes $S^*_{i+\frac{1}{2},j}$ and $S^*_{i,j+\frac{1}{2}}$ are respectively given by:

$$\begin{aligned}
S^*_{i+\frac{1}{2},j} &= (S_{\text{self}})^*_{i+\frac{1}{2},j} + (S_{\text{Hall}})^*_{i+\frac{1}{2},j} + (S_{\text{Nernst}})^*_{i+\frac{1}{2},j},\\
S^*_{i,j+\frac{1}{2}} &= (S_{\text{self}})^*_{i,j+\frac{1}{2}} + (S_{\text{Hall}})^*_{i,j+\frac{1}{2}} + (S_{\text{Nernst}})^*_{i,j+\frac{1}{2}}.
\end{aligned}$$

7.2.2 Total energy equation

We finally discuss the case of the total energy equation (7.19b). Like the magnetic field equation, it is discretized using a conservative finite volume method combined with an explicit time scheme which writes:

$$\frac{\overline{(\rho e)}^{n+1}_{i,j} - \overline{(\rho e)}^{n}_{i,j}}{\Delta t} - \frac{c}{\mu\Delta x_1}\left(E^*_{i+\frac{1}{2},j} - E^*_{i-\frac{1}{2},j}\right) - \frac{c}{\mu(x_2^\alpha)_j\Delta x_2}\left((x_2^\alpha)_{j+\frac{1}{2}}E^*_{i,j+\frac{1}{2}} - (x_2^\alpha)_{j-\frac{1}{2}}E^*_{i,j-\frac{1}{2}}\right) = 0.$$

Here, $\overline{(\rho e)}^{n}_{i,j}$ denotes the following cell average of (ρe) over the (i,j) cell at time t^n:

$$\overline{(\rho e)}^{n}_{i,j} = \frac{1}{\Delta x_1}\int_{(x_1)_{i-\frac{1}{2}}}^{(x_1)_{i+\frac{1}{2}}}\left[\frac{1}{(x_2^\alpha)_j\Delta x_2}\int_{(x_2)_{j-\frac{1}{2}}}^{(x_2)_{j+\frac{1}{2}}} x_2^\alpha(\rho e)(x_1,x_2,t^n)\,dx_2\right]dx_1,$$

and the approximate fluxes for the total energy equation are given by:

$$
\begin{aligned}
E^{*}_{i+\frac{1}{2},j} = &+ \left(\frac{(p_e + \rho_e \epsilon_e)^n_{i,j}}{2e(n_e)^n_{i,j}} + \frac{(p_e + \rho_e \epsilon_e)^n_{i+1,j}}{2e(n_e)^n_{i+1,j}} \right) \times \\
&\left(\frac{(x_2^{\alpha})_{j+1} \left(B^n_{i+1,j+1} + B^n_{i,j+1} \right) - (x_2^{\alpha})_{j-1} \left(B^n_{i+1,j-1} + B^n_{i,j-1} \right)}{4(x_2^{\alpha})_j \Delta x_2} \right) + S^{*}_{i+\frac{1}{2},j} \left(\frac{B^n_{i,j} + B^n_{i+1,j}}{2} \right), \\
E^{*}_{i,j+\frac{1}{2}} = &- \left(\frac{(p_e + \rho_e \epsilon_e)^n_{i,j}}{2e(n_e)^n_{i,j}} + \frac{(p_e + \rho_e \epsilon_e)^n_{i,j+1}}{2e(n_e)^n_{i,j+1}} \right) \left(\frac{B^n_{i+1,j+1} + B^n_{i+1,j} - B^n_{i-1,j+1} - B^n_{i-1,j}}{4\Delta x_1} \right) \\
&- S^{*}_{i,j+\frac{1}{2}} \left(\frac{B^n_{i,j} + B^n_{i,j+1}}{2} \right).
\end{aligned}
$$

Chapter 8

A direct drive ICF deceleration phase benchmark

We conclude this study with a test problem taken from [80] that aims at simulating the deceleration phase of an ICF shell in the direct drive context. It is thus particularly representative of concrete plasma physics applications. Several numerical results obtained on this ICF test problem with different configurations are given in the sequel. We present in particular simulation results for the complete two-temperature resistive MHD model derived in chapter 1 using a combination of all previously presented schemes. In order to emulate the presence of initial sphericity deviations and to see the development of self-generated magnetic field terms, a numerical perturbation is introduced in the considered initial condition.

The outline of this chapter is the following. We first describe the considered ICF test problem - in particular its initial condition and numerical setup - and explain how a slight sphericity deviation can be introduced numerically. We then compare our numerical results for an hydrodynamics + heat conduction configuration with those provided by a one-dimensional reference code (the so-called LP code [49, 80]) in spherical geometry in order to validate the thermal conduction scheme and the underlying physical model. We also illustrate the benefits of the first-order artificial viscosity model described in section 4.3.2 in terms of symmetry preservation on a purely hydrodynamical problem. Numerical results for the complete two-temperature resistive MHD equations are finally proposed for the perturbed ICF test problem in order to study self-generated magnetic field terms and to evaluate their effects on the flow.

Part of the simulation results given in the sequel have already been presented in the proceedings of the 2010 CEMRACS summer school [92].

8.1 Benchmark description

We first describe the test problem that we consider in the sequel for simulating the deceleration phase of an ICF shell in the direct drive context. This benchmark has been taken from [80]. The initial condition describes a spherical ICF capsule of radius 0.0313 cm at the beginning of the deceleration phase. More precisely, the capsule is at this stage made of a cryogenic DT layer of 0.0113 cm thickness which encloses an inner sphere of radius 0.021 cm filled with DT gas. It is initially surrounded by an isotemperature low-density and low-pressure gas that actually aims at emulating vacuum. According to the assumptions stated in chapter 1, both electrons and ions follow here a perfect gas law with $\gamma_{e,i} = 5/3$ and the plasma is assumed to be fully ionized. The average mass of DT is set to $\frac{3}{2}m_p$ where m_p is the mass of a proton.

Numerically, the initial velocity, total gas pressure, density and electronic pressure profiles in CGS units are given by:

- for $r \leq 0.018$ cm:

$$
\begin{aligned}
u^{(1)}(r) &= -6.01121 \cdot 10^8\, r - 8.59652 \cdot 10^{11}\, r^2 + 9.00161 \cdot 10^{13}\, r^3 - 2.55503 \cdot 10^{15}\, r^4,\\
p^{(1)}(r) &= 1.1364 \cdot 10^{14} - 1.4741 \cdot 10^{15}\, r + 2.5482 \cdot 10^{18}\, r^2 - 2.6914 \cdot 10^{20}\, r^3 + 8.3432 \cdot 10^{21}\, r^4,\\
\rho^{(1)}(r) &= p^{(1)}(r) \cdot 2.586759 \cdot 10^{-15} \cdot \left(1 - 1.7226 \cdot 10^4\, r^{2.56061}\right)^{-1},\\
p_e^{(1)}(r) &= p^{(1)}(r) \cdot (9.44575 \cdot 10^{18} - 3.00077 \cdot 10^{20}\, r) \cdot (3.3336 \cdot 10^{19} - 1.00185 \cdot 10^{21}\, r - 2.02242 \cdot 10^{22}\, r^2)^{-1},
\end{aligned}
$$

- for $0.018 \text{ cm} < r \leq 0.021$ cm:

$$
\begin{aligned}
u^{(2)}(r) &= -1.176725 \cdot 10^{11} + 2.40071 \cdot 10^{13}\, r - 1.83229 \cdot 10^{15}\, r^2 + 6.19924 \cdot 10^{16}\, r^3 - 7.84614 \cdot 10^{17}\, r^4,\\
p^{(2)}(r) &= -9.90683 \cdot 10^{17} + 2.04304 \cdot 10^{20}\, r - 1.57663 \cdot 10^{22}\, r^2 + 5.39767 \cdot 10^{23}\, r^3 - 6.91788 \cdot 10^{24}\, r^4,\\
\rho^{(2)}(r) &= 0.251956 - 1.99589 \cdot 10^4\, r^3 + (97.3135 - 4.61145 \cdot 10^3\, r)^{-1},\\
p_e^{(2)}(r) &= (-3.812829 \cdot 10^{12}\, r^5 + 3.758976 \cdot 10^{11}\, r^4 - 1.481416 \cdot 10^{10}\, r^3 + 2.916974 \cdot 10^8\, r^2\\
&\qquad - 2.869374 \cdot 10^6\, r + 1.127981 \cdot 10^4) \cdot p^{(2)}(r),
\end{aligned}
$$

- for $0.021 \text{ cm} < r \leq 0.0313$ cm:

$$
\begin{aligned}
u^{(3)}(r) &= 2.4198 \cdot 10^9 - 3.73623 \cdot 10^{11}\, r + 2.10716 \cdot 10^{13}\, r^2 - 5.24689 \cdot 10^{14}\, r^3 + 4.88108 \cdot 10^{15}\, r^4,\\
p^{(3)}(r) &= 5.5673 \cdot 10^{16} - 8.3122 \cdot 10^{18}\, r + 4.5695 \cdot 10^{20}\, r^2 - 1.0932 \cdot 10^{22}\, r^3 + 9.619 \cdot 10^{22}\, r^4,\\
\rho^{(3)}(r) &= 270.52 - 4.3198 \cdot 10^4\, r + 2.3752 \cdot 10^6\, r^2 - 5.17461 \cdot 10^7\, r^3 + 3.6722 \cdot 10^8\, r^4,\\
p_e^{(3)}(r) &= 0.5 \cdot p^{(3)}(r),
\end{aligned}
$$

- for $0.0313 \text{ cm} < r \leq 0.04$ cm:

$$
\begin{aligned}
u^{(4)}(r) &= \min(u^{(3)}(r), 0),\\
\rho^{(4)}(r) &= \exp\left(-1.20269 \cdot 10^3\, (r - 0.0313)\right),\\
p^{(4)}(r) &= 2.443754 \cdot 10^{13}\, \rho^{(4)}(r),\\
p_e^{(4)}(r) &= (p_e^{(3)}/\rho_e^{(3)})(r = 0.0313) \cdot \rho_e^{(4)}(r).
\end{aligned}
$$

These initial profiles (which are plotted in Figure 8.1) actually have been obtained by computing non-linear fittings of discrete numerical results provided by a one-dimensional ICF code [12]. More precisely, this ICF code has been used for simulating the implosion of an ICF capsule up to the beginning of the deceleration phase starting from a configuration that is particularly representative of realistic experiments. At the moment where the velocity has reached its maximum value (which is about $4.5 \cdot 10^7$ cm/s as shown in Figure 8.1), the discrete one-dimensional density, velocity and pressure profiles have been saved. These have then been interpolated in order to generate the above given analytical profiles. This approach allows us to perform realistic ICF implosion simulations without taking into account all the physical phenomena involved in such processes, especially the laser absorption since laser beams have already been shut down at the beginning of the deceleration phase.

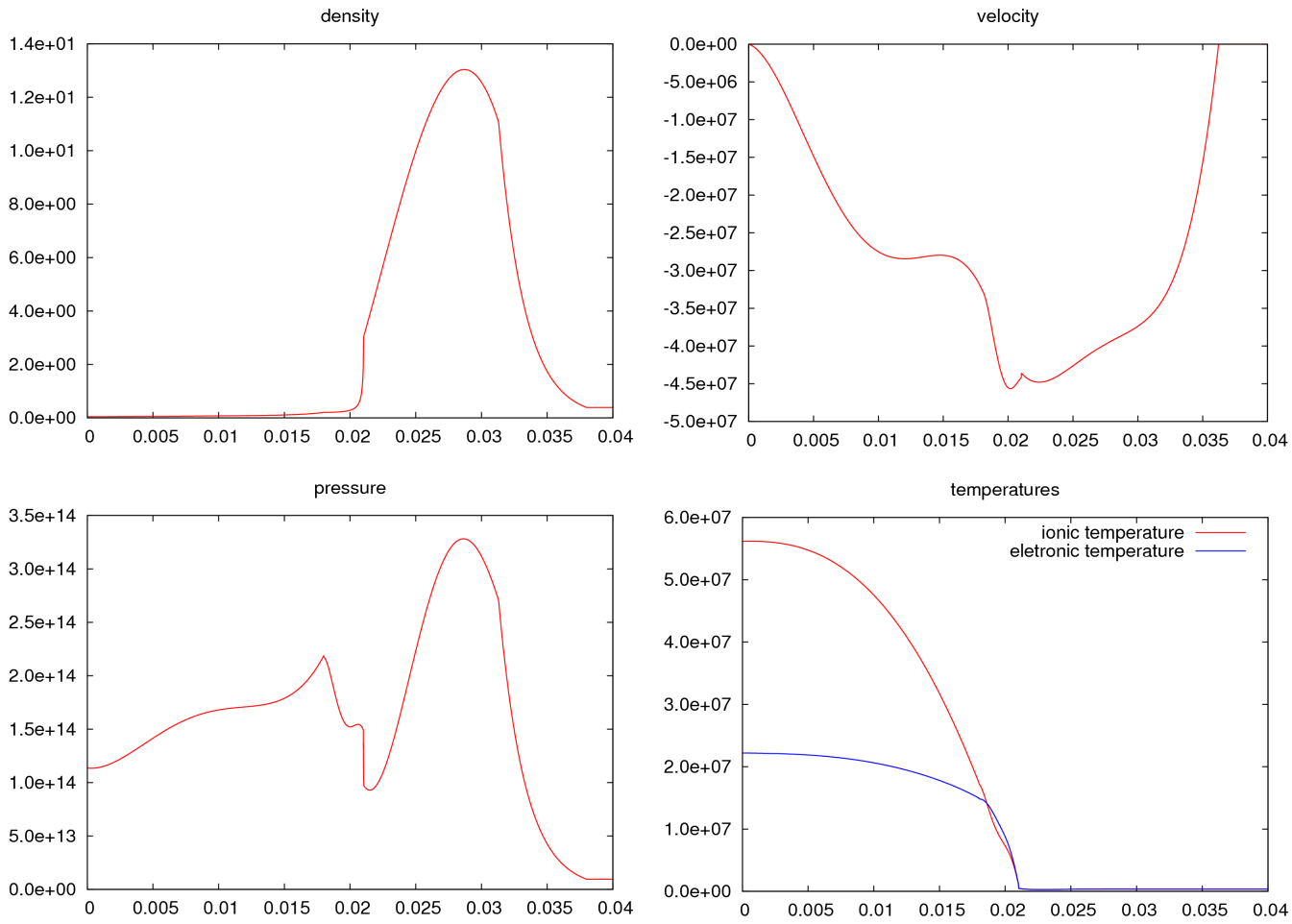


Figure 8.1: Initial profiles of the ICF benchmark taken from [80].

The deceleration phase of this spherically symmetric ICF implosion is here simulated in axisymmetric geometry until time $t = 6{\cdot}10^{-10}$ s (*i.e.* $t = 600$ ps) with the two-dimensional solver that has been developed. The considered problem is Rayleigh-Taylor unstable, especially for simulation times between 500 ps and 600 ps. The computational domain is a 0.04 cm wide square in the (z, r) plane with reflective boundary conditions along the $z = 0$ and $r = 0$ axis and open boundary conditions elsewhere. The following execution parameters are common to all simulation results presented in this chapter:

- The Lagrangian and remap fluxes used in GoHy schemes are computed at third-order accuracy.
- The polynomial reconstruction (4.15) that aims at computing high-order accurate cell-centered pointwise values of conservative variables is here disabled.
- The dimensional splitting set of coefficients used in multidimensional GoHy schemes is Godunov's first-order accurate splitting sequence (4.30).
- The Lagrangian and remap fluxes reconstruction steps (4.45) and (4.46) are disabled.
- Divergence cleaning is here unnecessary since computations are performed in the orthogonal MHD framework: the only possibly non-zero component of the magnetic field is the one that is orthogonal to the comptutional domain. The divergence constraint is thus automatically satisfied.

8.2 Introducing perturbations

We have seen in introduction that self-generated magnetic field terms may appear during direct drive ICF processes in case of sphericity deviations on the surface of the ICF shell, these being generally due either to the non-uniformity of the laser irradiation and/or to shell rugosity. In order to represent these sphericity deviations in numerical simulations, we introduce as in [80] a slight perturbation along the radial direction in the previously given initial condition. Let:

$$r^{\varepsilon}(r) = r + A_l(r)P_l(\cos\theta) \quad \text{with} \quad A_l(r) = A_0 \exp\left(-l\left|\frac{r}{r_i} - 1\right|\right),$$

where A_0 is the initial perturbation amplitude, $r_i = 0.021$ cm the location of the gas/shell interface, l the mode number and P_l the Legendre polynomial with θ the angle between the radial and symmetry axis. Perturbed initial conditions $\mathbf{U}^{\varepsilon}$ are then defined by:

$$\mathbf{U}^{\varepsilon}\left(r^{\varepsilon}(r)\right) = \mathbf{U}(r)$$

with $\mathbf{U}(r) = (\rho, p_e, p, u)^t(r)$ given in section 8.1.

8.3 Numerical results

8.3.1 Comparison with a one-dimensional reference code

In order to validate the solver that we have developed, in particular the heat conduction scheme and the underlying physical model, we first compare our two-dimensional axisymmetric simulations with converged one-dimensional numerical results obtained with the one-temperature LP code [49, 80] in spherical geometry. In this case, the test problem is initialized as indicated in section 8.1 but both ionic and electronic temperatures are here computed according to the following formula:

$$T_i = T_e = \frac{p}{\rho C_v(\gamma - 1)}$$

with $C_v = 6.471255 \cdot 10^7$ and $\gamma = 5/3$. Only the hyperbolic and thermal diffusion systems are considered here so that no magnetic field term may appear. For practical computations, the CFL coefficient is set to 0.7 and the hyperviscosity model is enabled with $C_\beta = 0.8$. Figure 8.2 plots the results obtained with both codes at time $t = 600$ ps. It shows a good agreement between both simulations, the results of our two-dimensional axisymmetric solver appearently converging towards those of the one-dimensional LP reference code.

8.3.2 Impact of the first-order artificial viscosity model

In order to illustrate the benefits provided by the first-order artificial viscosity model that has been described in section 4.3.2, we now present numerical results obtained with the hydrodynamical solver and the below given execution parameters.

- The hyperviscosity model is enabled with $C_\beta = 0.8$ and C_κ given by:

$$C_\kappa = \begin{cases} 0.1 & t \leq 230 \text{ ps}, \\ 0.2 & 230 \text{ ps} \leq t \leq 250 \text{ ps}, \\ 0.1 & 250 \text{ ps} \leq t \leq 600 \text{ ps}. \end{cases}$$

- The CFL coefficient is set according to the following table:

$$\text{CFL} = \begin{cases} 0.7 & t \leq 430 \text{ ps}, \\ 0.6 & 430 \text{ ps} \leq t \leq 480 \text{ ps}, \\ 0.4 & 480 \text{ ps} \leq t \leq 500 \text{ ps}, \\ 0.7 & 500 \text{ ps} \leq t \leq 600 \text{ ps}. \end{cases}$$

In Figure 8.3, we compare two unperturbed simulations on different meshes where the evanescent first-order artificial viscosity model is respectively disabled (left side) and enabled with $C_\nu = 0.4$ (right side). These numerical results clearly show that symmetry preservation is improved when the first-order artificial viscosity model is activated. Indeed, the unphysical high-frequency structures that appear with $C_\nu = 0$ are completely smeared out thanks to artificial viscosity. Figure 8.4 plots simulation results that have been obtained in the perturbed case with $A_0 = 10^{-4}$ cm. It shows even better improvements since the unphysical small scale structures grow faster in the presence of an initial perturbation but are still efficiently removed by introducing artificial viscosity.

8.3.3 Study of self-generated magnetic field terms

We now study numerical results obtained for the complete two-temperature resistive MHD model. To that end, the GoHy schemes are coupled with diffusion operators and source terms discretizations through an operator splitting strategy.

Impact of self-generated magnetic field terms

We first compare simulations for different amplitudes of the initial perturbation with and without magnetic field generation. Figure 8.5 plots the density and electronic temperature obtained for the perturbed ICF test problem at time $t = 600$ ps on 4000×4000 cells with two different configurations.

- The left hand side results have been obtained by solving only the hyperbolic and heat conduction systems so that no magnetic field terms appear.

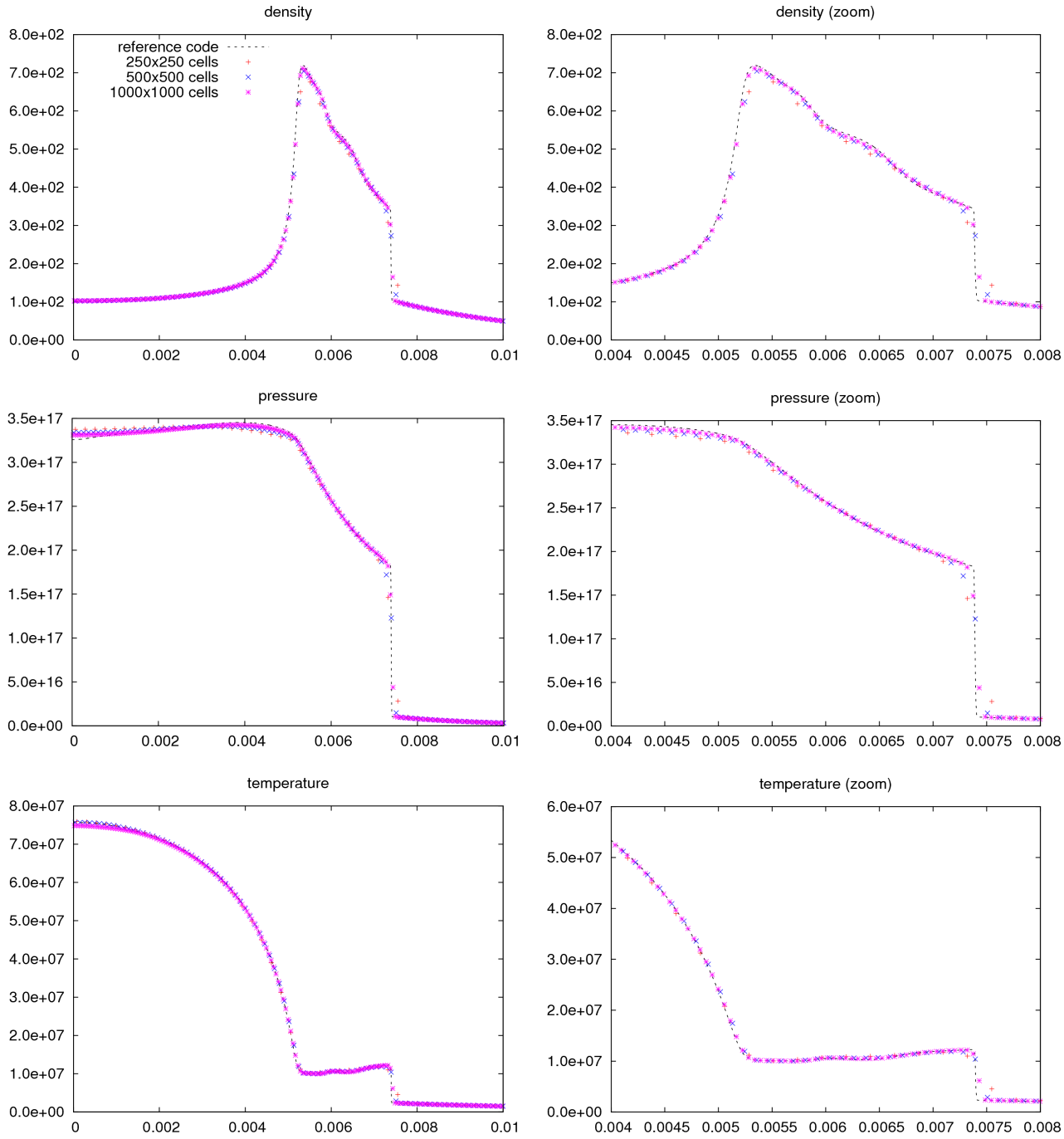


Figure 8.2: One-temperature ICF test problem. Comparison between one-dimensional LP code [49, 80] results and two-dimensional axisymmetric simulations (slice along the $z = r$ axis) at time $t = 600$ ps: zoom on the $[0; 0.01]$ region (left) and on the $[0.004; 0.008]$ region (right).

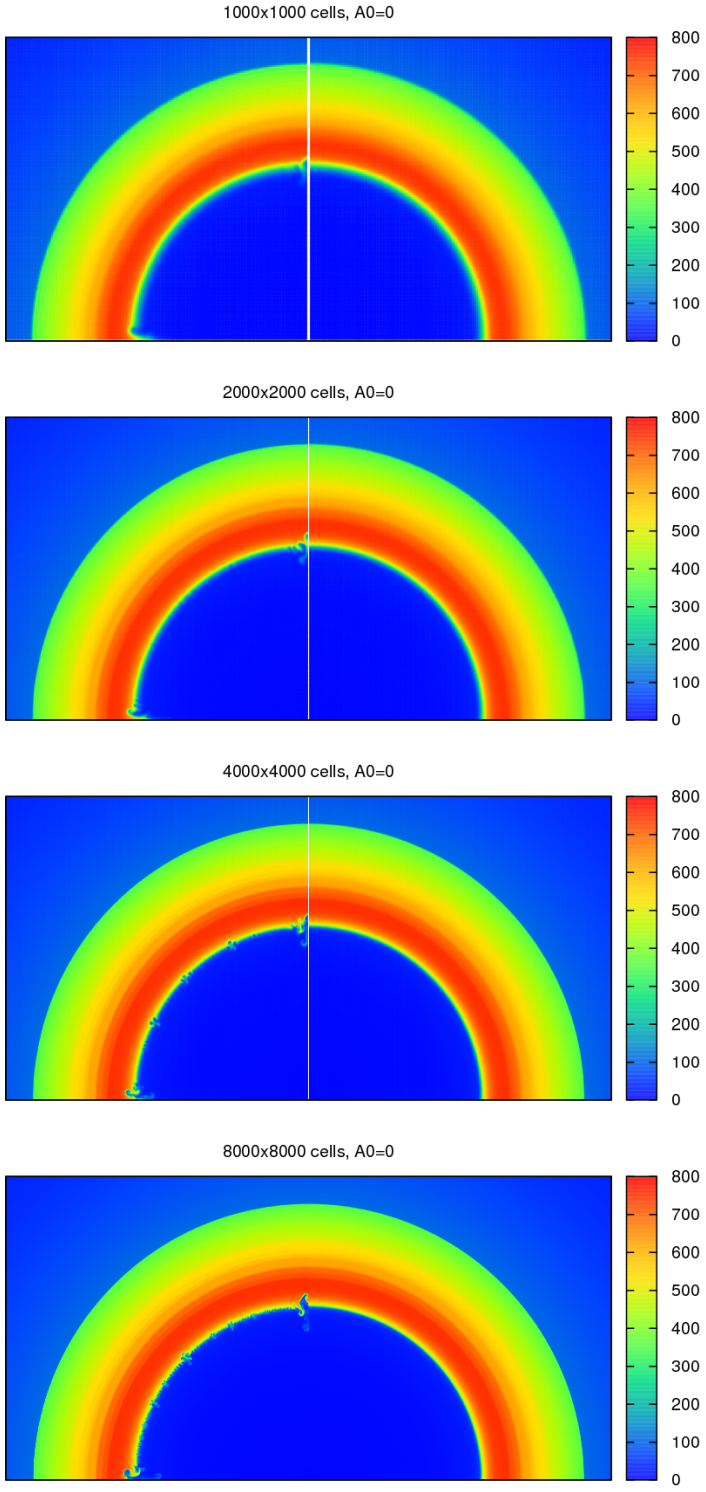


Figure 8.3: Effect of the evanescent first-order artificial viscosity $\nu^\star$ coefficient on the two-temperature ICF test problem at time $t = 600$ ps (zoom on the $[0; 0.008]^2$ square). Density obtained with the hydrodynamical solver (no heat conduction) on 1000×1000, 2000×2000, 4000×4000 and 8000×8000 cells for $A_0 = 0$ with $C_\nu = 0$ (left) and $C_\nu = 0.4$ (right).

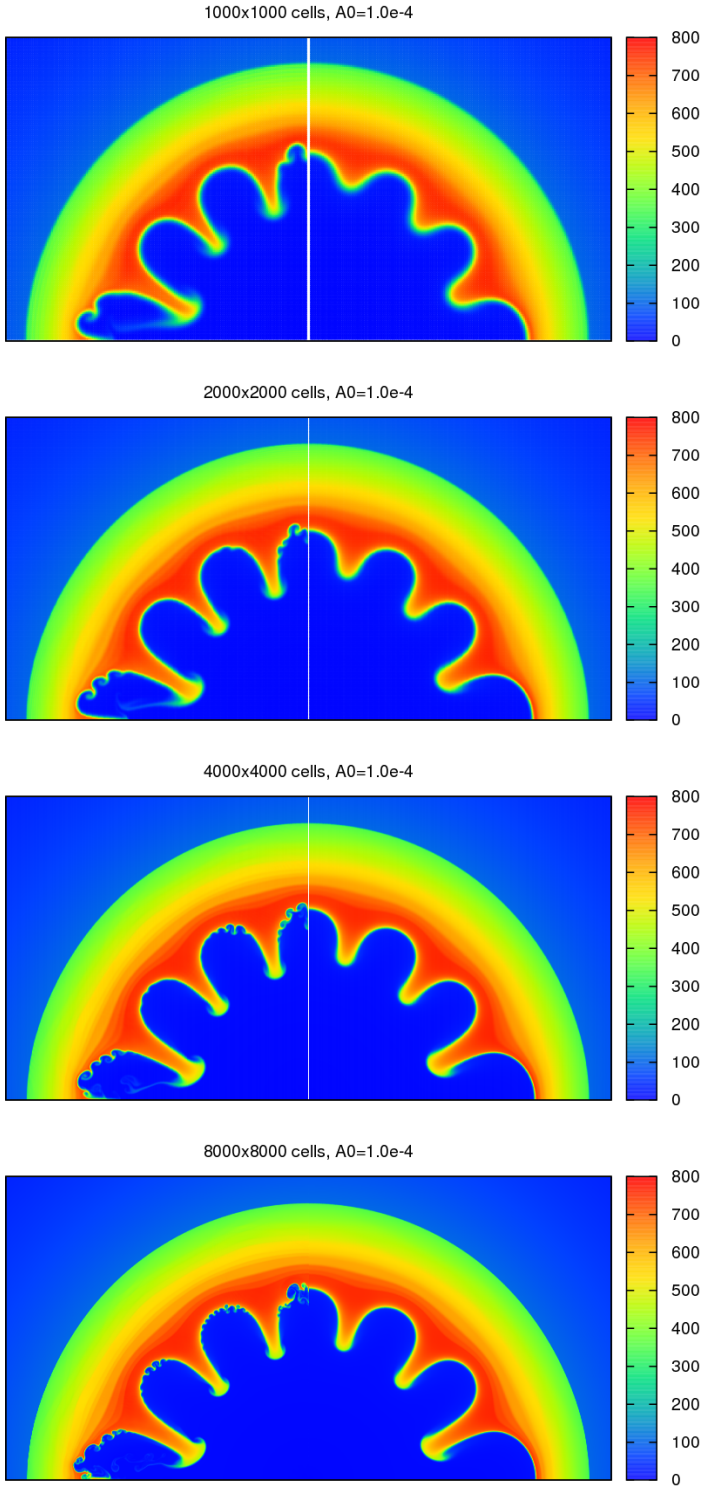


Figure 8.4: Effect of the evanescent first-order artificial viscosity $\nu^\star$ coefficient on the two-temperature ICF test problem at time $t = 600$ ps (zoom on the $[0; 0.008]^2$ square). Density obtained with the hydrodynamical solver (no heat conduction) on 1000×1000, 2000×2000, 4000×4000 and 8000×8000 cells for $A_0 = 10^{-4}$ cm with $C_\nu = 0$ (left) and $C_\nu = 0.4$ (right).

- The right hand side results have been obtained by considering the complete two-temperature resistive MHD model, *i.e.* the orthogonal MHD equations, the thermal and resistive conduction systems and the source terms' contributions. In this case, some self-generated magnetic field terms appear. They are plotted in Figure 8.6

In both cases, simulations have been performed with the hyperviscosity model (with $C_\beta = 0.8$, $C_\nu = 0.4$ and $C_\kappa = 0.05$) and a CFL coefficient of 0.7.

As expected, Figure 8.6 shows that magnetic fields are generated around both sides of spikes where the electronic temperature gradient is the steepest. It also indicates that the higher the initial perturbation amplitude is, the higher the magnitude of the magnetic field. For $A_0 = 10^{-3}$, the computed magnetic field is about 50 MG, a value that is comparable to magnitudes given in the related literature [42]. According to Figures 8.5 and 8.7, self-generated magnetic field terms have a very little impact: they seem to slightly reduce the perturbations growth rates (especially along the z axis as shown in Figure 8.7) but do not modify the topology of the flow.

Nernst effect contribution

The source terms system that has been studied in section 7.2 takes the Hall and Nernst effects' contributions into account. Practically, numerical experiments have shown that the Hall effect does not have a visible impact on numerical results in the case of the ICF test problem. On the contrary, we have observed that the Nernst effect noticeably modifies the magnetic field. Figure 8.8 plots simulations results obtained for the ICF test problem at time $t = 600$ ps on 4000×4000 cells. Computation parameters are the same than in the previous section. The left hand side results have been obtained by solving the complete two-temperature resistive MHD model while the Nernst effect has been disabled for the right hand side ones. Figure 8.8 shows that the Nernst effect convects the magnetic field into smaller regions and thus makes it reach significantly higher magnitudes. For example, starting from a perturbed initial condition with $A_0 = 10^{-3}$ cm, the maximum magnitude of the magnetic field at time $t = 600$ ps is about 30 MG when the Nernst effect is not taken into account whereas this value reaches more than 50 MG in the opposite case.

Convergence analysis

We finally carry out a short convergence analysis on numerical results obtained with the complete two-temperature resistive MHD model. Figure 8.10 plots the density, electronic temperature and magnetic field obtained at time $t = 600$ ps with an initial perturbation of amplitude $A_0 = 5 \cdot 10^{-4}$ cm for different mesh sizes. On the one hand, it shows that the numerical results on 1000×1000 cells seem slightly less accurate than those obtained on finer meshes, especially in zones where non-linear Rayleigh-Taylor instabilities develop. On the other hand, the numerical results obtained on 2000×2000 and 4000×4000 cells can hardly be distinguished in Figure 8.10, which tends to show that these are close to convergence.

It seems nevertheless difficult to state conclusions from the map plots given in Figure 8.10 since differences are very slight. We therefore plot in Figure 8.11 two slices of the above-described numerical results, one along the $z = r$ axis and the other along the $r = 0.00119$ cm axis where the magnetic field reaches its maximum value at $z \approx 0.003$ cm. On both slices, the density and electronic temperature profiles seem to exhibit convergent behaviours towards simulation results obtained on the finest mesh (4000×4000 cells here). But the magnetic field profiles have to be discussed more carefully.

- On the one hand, the behaviour observed on the density and temperature profiles seems to be confirmed by the slice of the magnetic field along the $z = r$ axis (though finer results would be needed in

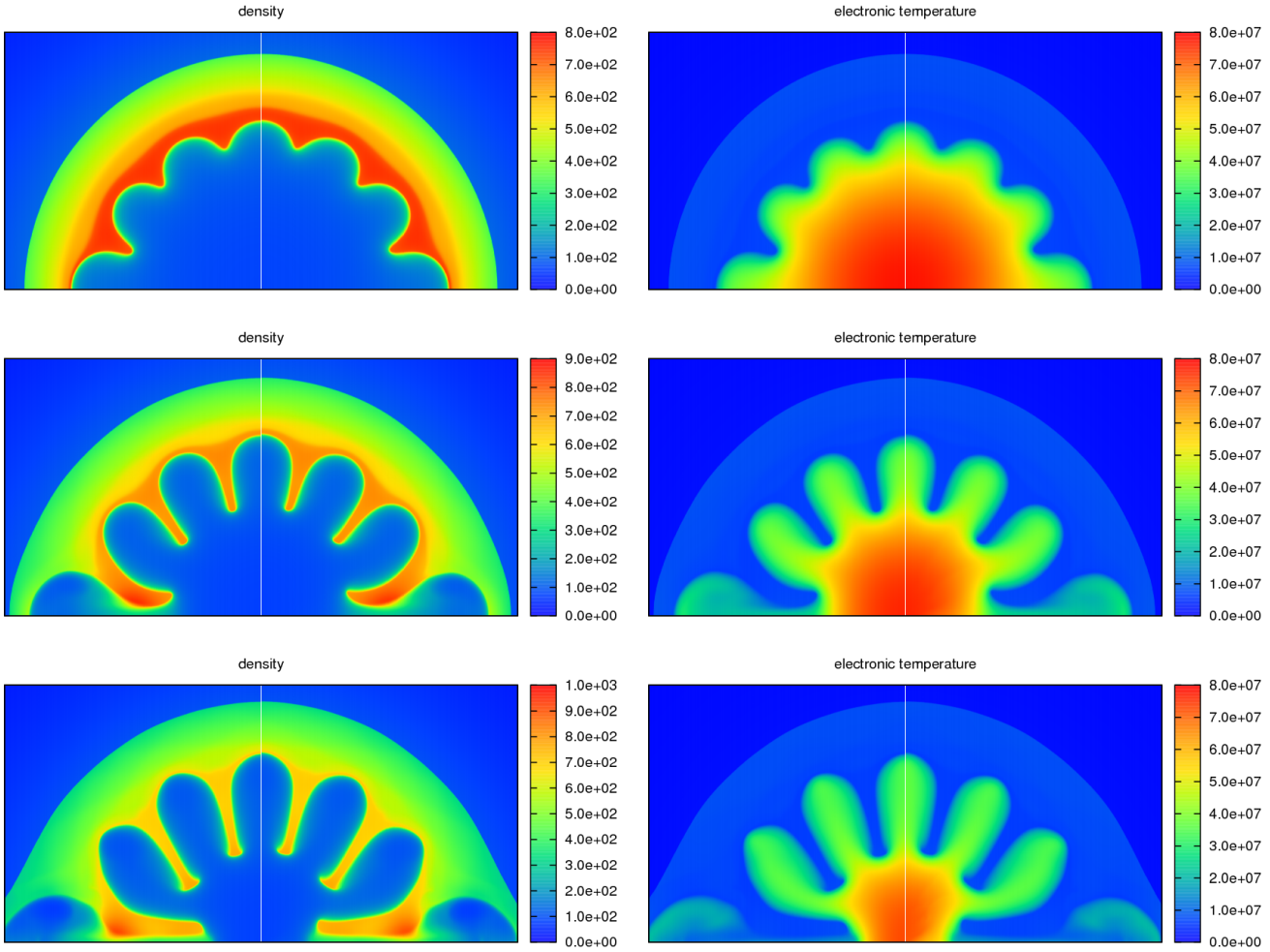


Figure 8.5: Two-temperature ICF test problem at time $t = 600$ ps on 4000×4000 cells (zoom on the $[0; 0.008]^2$ square). Density and electronic temperature without/with (left/right) self-generated magnetic field terms. From top to bottom: $A_0 = 10^{-4}$, $5 \cdot 10^{-4}$ and 10^{-3} cm.

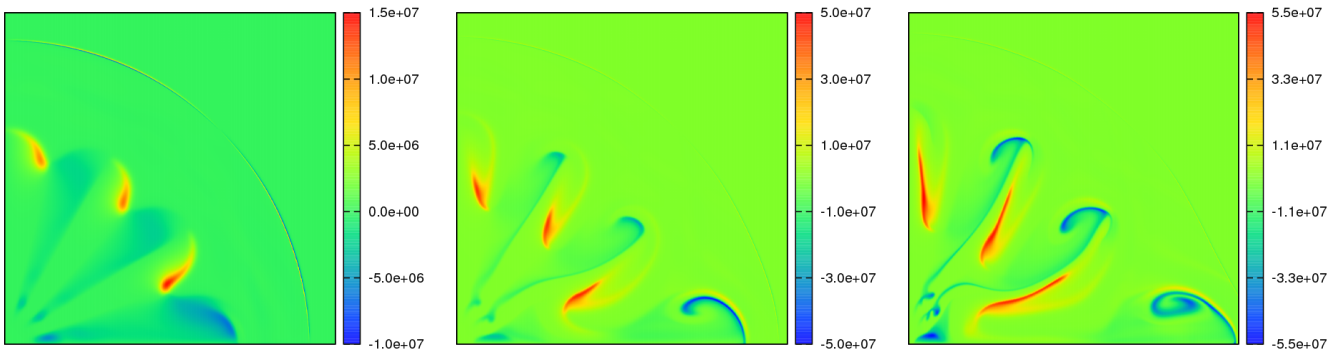


Figure 8.6: Magnetic field obtained for the perturbed two-temperature ICF test problem at time $t = 600$ ps on 4000×4000 cells (zoom on the $[0; 0.008]^2$ square). From left to right: $A_0 = 10^{-4}$, $5 \cdot 10^{-4}$ and 10^{-3} cm.

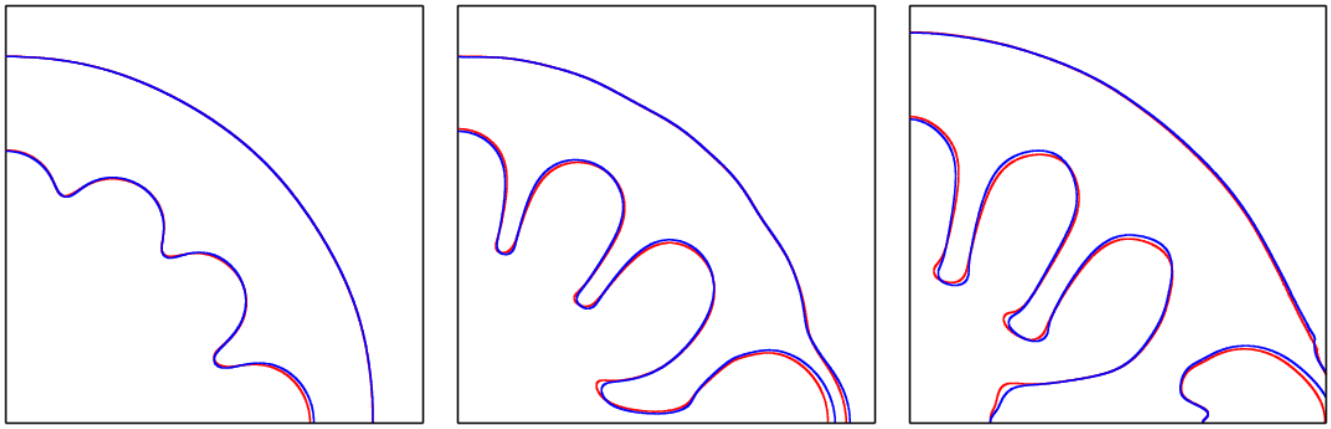


Figure 8.7: Contour plots of the density profiles shown in Figure 8.5. Results with and without self-generated magnetic field terms are respectively plotted in red and blue. From left to right: $A_0 = 10^{-4}$, $5 \cdot 10^{-4}$ and 10^{-3} cm.

order to state conclusions for the region located around 0.001 cm).

- On the other hand, the slice of the magnetic field map along the $r = 0.00119$ cm axis shows that numerical results clearly differ in the vicinity of $z = 0.006$ cm. However, this could simply be a consequence of the instability that develops along the z axis around $z = 0.006$ (see Figure 8.10). This structure may indeed modify the magnetic field profiles in a larger region on coarse meshes due to numerical dissipation and thus impact the magnetic field profiles along the $r = 0.00119$ cm axis.

- The magnetic field slice along the $r = 0.00119$ axis provides another interesting information. It indeed shows that the magnetic field reaches comparable magnitudes (about 45 MG) whatever the mesh size.

Convergence analysis for multidimensional datasets cannot rely exclusively on informations provided by slices since all multidimensional effects cannot be represented faithfully on one-dimensional plots. The previous study therefore has to be interpreted carefully. These results nevertheless seem to indicate that convergence has not been reached on 4000×4000 cells yet. This diagnostic is confirmed by the evolution of the L^1 norm of the magnetic field versus time shown in Figure 8.9 for the three considered mesh sizes. One can notice that the magnetic field norm plots seem to coincide up to $t \approx 450$ ps but then slightly differ, which actually was predictable since the deceleration phase is heavily Rayleigh-Taylor unstable for $500 \text{ ps} \leq t \leq 600$ ps.

Figure 8.9 shows another satisfying behaviour. It plots the L^1 norm of the magnetic field versus time in the absence of an initial perturbation. In this case, the generated magnetic field is unphysical in the sense that it is only due to numerical approximations. Fortunately, Figure 8.9 indicates that the norm of the magnetic field is reduced for unperturbed simulations when the mesh is refined.

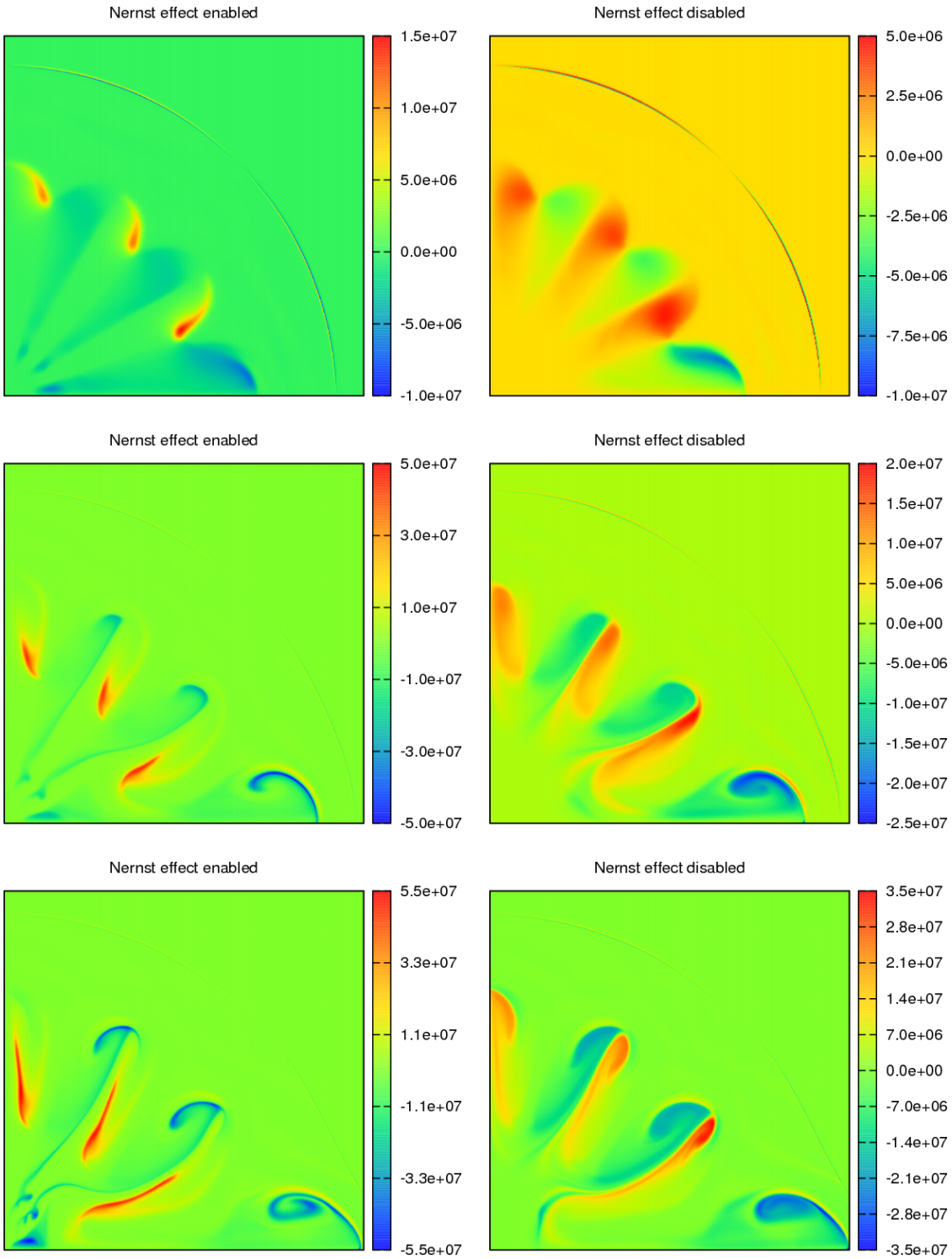


Figure 8.8: Magnetic field obtained for the perturbed two-temperature ICF test problem at time $t = 600$ ps on 4000×4000 cells (zoom on the $[0; 0.008]^2$ square) with (left) and without (right) Nernst effect. From top to bottom: $A_0 = 10^{-4}$, $5 \cdot 10^{-4}$ and 10^{-3} cm.

Number of cells	Number of processes	Grind time	Restitution time	Efficiency
1000×1000	25	3.19	1333	
2000×2000	100	3.32	2750	96%
4000×4000	400	3.41	6499	94%
8000×8000	1600	3.69	15379	86%

Table 8.1: Measured grind time (in microseconds per cell per time step) and restitution time (in seconds) for the simulations presented in section 8.3.2. The efficiency is given relatively to the computation on 1000×1000 cells.

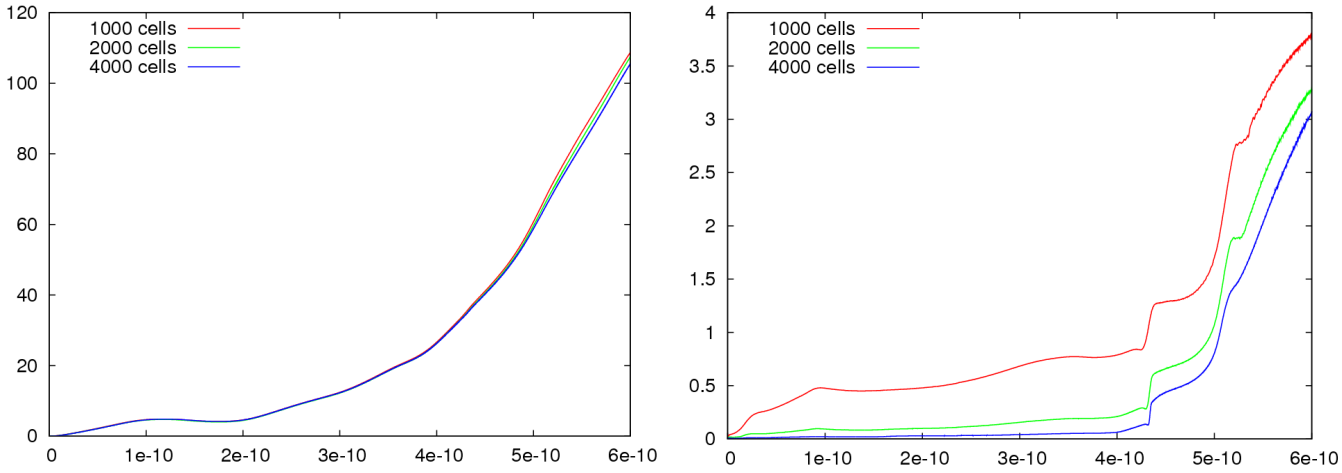


Figure 8.9: L^1 norm of the magnetic field versus time for the ICF test problem on 1000×1000, 2000×2000 and 4000×4000 cells: perturbed case with $A_0 = 5 \cdot 10^{-4}$ cm (left) and unperturbed case (right).

8.4 Performance and parallelism aspects

We conclude this chapter with a few remarks about performance and parallellism aspects. The simulations described in this chapter have been performed on the CEA Tera 100 supercomputer.

8.4.1 Hydrodynamics simulations

Hydrodynamics simulations presented in section 8.3.2 have been performed with 200×200 cells per process. The measured grind time has been summarized in Table 8.1 for the considered mesh sizes (from 1000×1000 to 8000×8000 cells). Note that we do not distinguish computations performed with or without first-order artificial viscosity since the restitution time is similar for both cases. These performance figures show that the parallel efficiency still reaches satisfying values on 1600 processes.

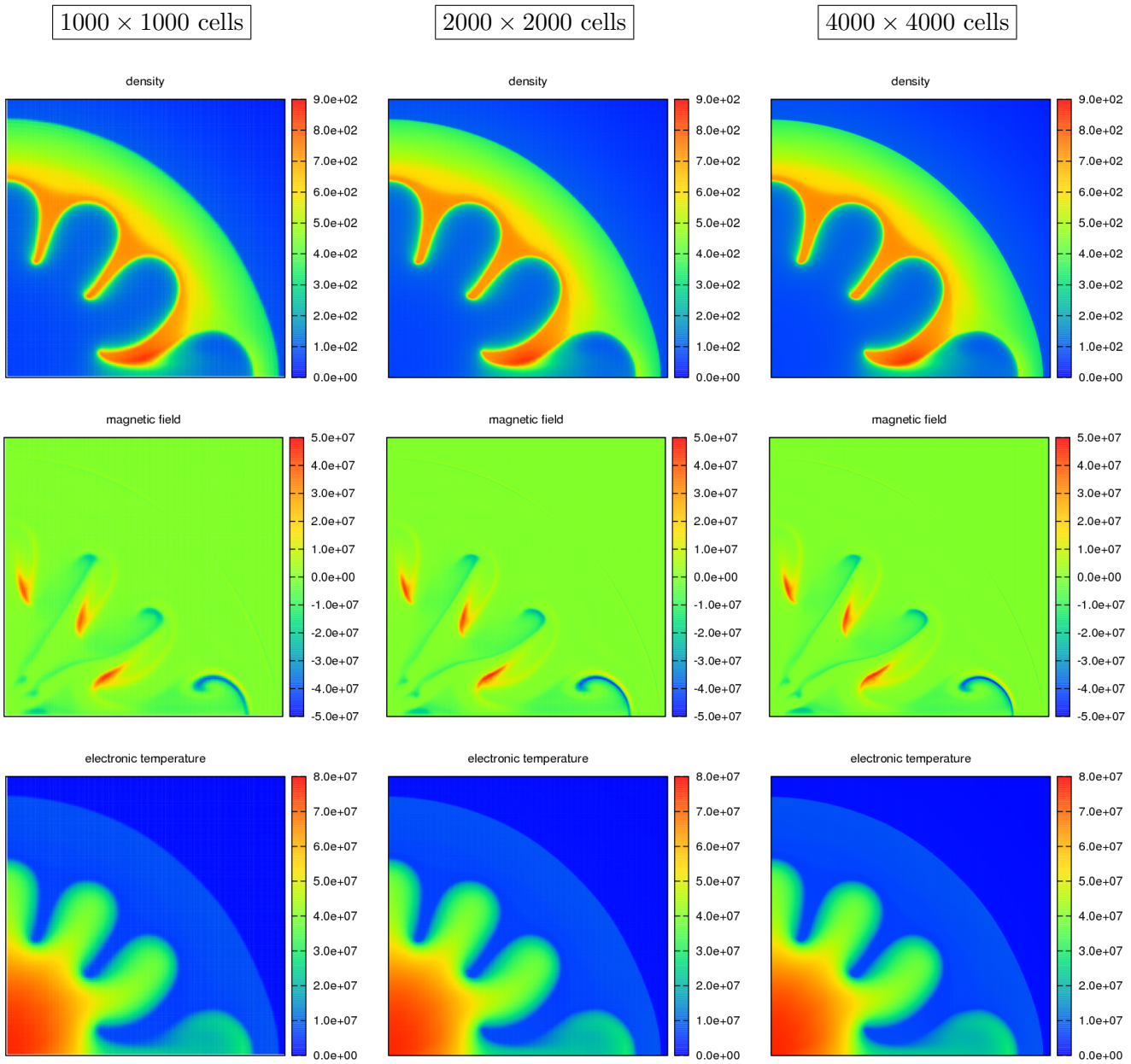


Figure 8.10: Numerical results to the two-temperature ICF test problem with $A_0 = 5 \cdot 10^{-4}$ cm at time $t = 600$ ps on 1000×1000, 2000×2000 and 4000×4000 cells (zoom on the $[0; 0.008]^2$ square). From top to bottom: density, magnetic field and electronic temperature.

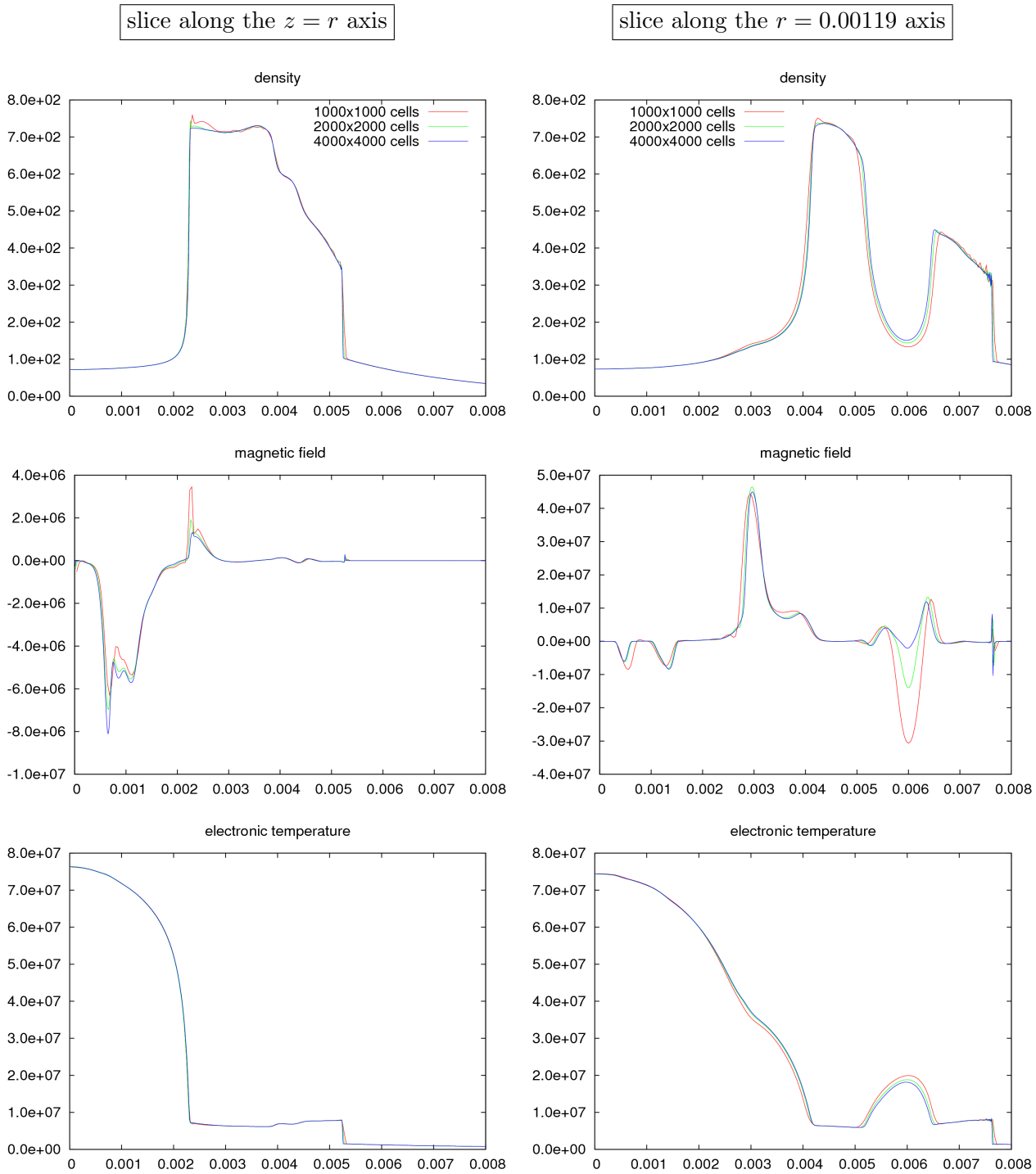


Figure 8.11: Slices along the $z = r$ and $r = 0.00119$cm axis of the numerical results presented in Figure 8.10.

Number of cells	Number of processes	Grind time	Restitution time
1000×1000	100	27.9	1260
2000×2000	400	35.2	3162
4000×4000	1600	52.5	9423
8000×8000	6400	93.7	33623

Table 8.2: Measured grind time (in microseconds per cell per time step) and restitution time (in seconds) for the simulations presented in section 8.3.3 on the perturbed ICF test problem with $A_0 = 10^{-4}$.

8.4.2 Simulations on the complete two-temperature resistive MHD model

Computations presented in section 8.3.3 for the complete model have been performed with 100×100 cells per process. The measured grind time has been summarized in Table 8.2 for the perturbed ICF test problem with $A_0 = 10^{-4}$ and mesh sizes going from 1000×1000 to 8000×8000 cells. Note that we do not compute the parallel efficiency here since the number of iterations in the BiCGStab algorithm varies with the mesh size: results given below therefore cannot be compared. We can nevertheless state the following remarks.

- The grind time is a lot greater than in the pure hydrodynamical case due to the semi-implicit treatment of diffusion terms which clearly has to be optimized.
- We infer that the grind time could be noticeably lowered using a more efficient preconditioner than the implemented Jacobi preconditioner. Another interesting point would be to decorrelate the number of iterations in the iterative solver from the number of cells, which seems possible using multigrid techniques [89].
- Even if performance still has to be improved, these results show that the solver that we have developed is able to deal with a huge number of processes and can provide results for the complete model on very fine meshes is a reasonable restitution time (approximately 9 hours for the 8000×8000 cells computation).

8.5 Conclusions on the ICF deceleration phase benchmark

We have presented in this section preliminary numerical results for the complete two-temperature resistive MHD model on an ICF test problem in both perturbed an unperturbed cases. On the one hand, we have seen that the magnetic field which is generated in the absence of an initial perturbation and is only due to numerical approximations is reduced in L^1 norm when the mesh is refined. On the other hand, numerical results obtained from a perturbed initial condition seem convincing and in accordance with published workds (see [42] for instance). Magnetic fields are generated around both sides of spikes of Rayleigh-Taylor modes, reach magnitudes up to 50 MG depending on the initial perturbation amplitude and seem to slighlty reduce the instabilities growth rates compared to classical two-temperature simulations.

These growth rates now have to be determined more accurately, which requires numerical results that are closer to convergence. This could be achieved by improving the preconditioning step so that the solver

that has been developed is able to run faster and thus to deal with even finer meshes. Convergence could also be accelerated by building second-order accurate schemes for the diffusion and source term operators.

Conclusions et perspectives

Conclusions

Nous avons établi dans cette étude un modèle de MHD résistive à deux températures avec prise en compte des termes de champ magnétique auto-généré basé sur des relations de fermeture de Braginskii [11] et sur un modèle de collisions de Decoster [28]. Afin de faciliter sa résolution, celui-ci a été décomposé en plusieurs sous-systèmes selon la nature de l'opérateur mathématique sous-jacent que nous avons reformulés en énergie totale afin de garantir la conservativité. Nous avons ensuite proposé des méthodes numériques adaptées à chaque opérateur permettant la résolution du modèle complet en deux dimensions d'espace sur grille cartésienne en géométries plane et axisymétrique. Celles-ci ont été implémentées dans un code de simulation qui a été parallélisé à l'aide d'une méthode de décomposition de domaine.

Nous avons tout d'abord traité le cas de l'opérateur hyperbolique - c'est-à-dire des équations de l'hydrodynamique ou de la MHD idéale selon que l'on tienne compte ou non des termes de champ magnétique - pour lequel nous avons proposé une nouvelle classe de schémas numériques GoHy d'ordre élevé en espace et en temps en régime non linéaire sur grille cartésienne dans le formalisme Lagrange + projection. Ceux-ci ont fait l'objet de plusieurs publications [32, 93].

La principale innovation de cette nouvelle souche de schémas est qu'elle a été conçue afin de tirer parti des calculateurs modernes massivement parallèles. Les schémas GoHy reposent ainsi sur une méthode monodimensionnelle particulièrement efficace faisant appel à un schéma en temps direct (ce qui limite le nombre d'appels aux conditions de bord - et donc les phases de communication - et à l'équation d'état) et à des méthodes d'interpolation centrées peu coûteuses. Celle-ci est étendue au cas bidimensionnel à l'aide de techniques de splitting directionnel d'ordre élevé. Le traitement par directions alternées constitue une approche novatrice pour les ordres strictement supérieurs à 2 qui présente un avantage significatif en termes de performances : celle-ci permet en effet la mise en œuvre d'implémentations dites *cache-oblivious* favorisant la réutilisation des données présentes dans le cache du processeur. Par ailleurs, le contrôle des oscillations inhérentes aux schémas d'ordre élevé et la préservation de la symétrie sont assurés par des méthodes de viscosité artificielle ne nécessitant aucun test - contrairement aux méthodes classiques de type limiteur - dont le coût est par conséquent modéré.

En pratique, nous avons montré que l'ordre expérimental de convergence des schémas GoHy correspond à l'ordre théorique et que leur robustesse est satisfaisante. Surtout, nous avons mis en évidence leur capacité à tirer parti des calculateurs modernes par des mesures de performance et d'efficacité parallèle. Les schémas GoHy sont capables d'exploiter jusqu'à 40% de la puissance crête des processeurs actuels tandis que l'efficacité parallèle de notre implémentation atteint 95% sur 256 processus, ce qui constitue des résultats tout à fait satisfaisants.

Afin de permettre la simulation de processus d'implosion de capsule FCI en attaque directe, nous avons combiné les schémas GoHy avec des méthodes de type volumes finis permettant la prise en compte des termes

de conduction thermique et résistive, des termes de champ magnétique auto-généré et des contributions des effets Hall et Nernst. Dans cette partie de notre étude, l'aspect ordre élevé a été mis de côté, l'objectif étant simplement de résoudre le modèle complet en un temps de calcul raisonnable. Une attention particulière a donc été portée à l'implémentation de ces schémas, notamment au traitement semi-implicite des termes de diffusion qui suppose la résolution de systèmes linéaires creux de grande taille.

Le code de calcul ainsi obtenu nous a permis d'effectuer des simulations numériques sur un cas test particulièrement représentatif de processus d'implosion concrets décrivant la phase de décélération d'une capsule FCI. Dans ce contexte, nous avons émulé la présence de défauts de sphéricité à la surface de la coquille en introduisant une perturbation numérique à l'aide de polynômes de Legendre. Plusieurs conclusions peuvent être tirées de ces expérimentations numériques.

- D'une part, les efforts consentis en termes d'implémentation ont permis de mener des simulations sur des maillages très fins. En pratique, nous avons été capables d'effectuer des calculs pour le modèle complet de MHD résistive à deux températures sur des maillages de 64 millions de mailles (soit près de 400 millions d'inconnues) distribués sur 6400 coeurs, ce qui montre la capacité du solveur développé à tirer parti des supercalculateurs modernes. Les résultats numériques ainsi obtenus ne sont toutefois pas tout à fait convergés.
- D'autre part, les simulations effectuées ont permis d'étudier les termes de champ magnétique auto-généré. Ceux-ci atteignent des magnitudes de plusieurs dizaines de mégagauss mais ne modifient pas la topologie du plasma. Le taux de croissance des instabilités est très légèrement réduit mais de tels niveaux de champ magnétique ne semblent pas modifier considérablement la distribution de température.

L'implémentation des schémas GoHy est désormais utilisée par plusieurs équipes comme benchmark permettant d'évaluer les performances des architectures récentes dans le domaine du calcul hautes performances.

Perspectives

À l'issue de cette étude, plusieurs points demandent encore à être approfondis. Du point de vue numérique, l'un des axes d'amélioration consisterait à discrétiser les opérateurs de diffusion et les termes sources à l'ordre 2. Une telle approche serait probablement très coûteuse et son implémentation se révèlerait difficile en raison de la complexité des relations de fermeture de Braginskii mais elle pourrait accélérer la convergence et permettre une estimation précise des taux de croissance des instabilités au cours de la phase de décélération à l'instar de ce qui a été fait dans [80].

Dans ce but, il serait par ailleurs bénéfique d'optimiser les performances de notre solveur, notamment dans le cadre de la résolution des termes de conduction thermique et résistive. En effet, le préconditionnement du système linéaire généré par notre approche semi-implicite est actuellement effectué par une méthode de Jacobi dont nous avons vu qu'elle est peu efficace et nécessiterait d'être remplacée par un préconditionneur plus sophistiqué, de type multigrille par exemple.

Enfin, nous avons vu que les termes de champ magnétique auto-généré atteignent des magnitudes très importantes mais n'ont pas d'effet visible sur le plasma. Ceux-ci devraient en revanche avoir une grande influence sur le transport des particules chargées en les faisant s'enrouler autour des lignes de champ magnétique. Le dépôt d'énergie serait alors plus important au voisinage du point chaud. Il semblerait donc intéressant de coupler le code de simulation que nous avons mis en place avec un code de transport. Des collaborations visant à réaliser ce couplage ont déjà été initiées.

Conclusions and perspectives

Conclusions

We have derived in this study a two-temperature resistive MHD model based on Braginskii's closure relations [11] and Decoster's collisions model [28] that takes self-generated magnetic field terms into account. In order to ease its resolution, the complete system has been split in several subsystems according to the nature of the underlying mathematical operator. These subsystems have been reformulated in total energy in order to guarantee conservativity. We then have discussed the discretization of each operator and proposed appropriate numerical methods in order to allow the resolution of the whole model on two-dimensional planar and axisymmetric geometries. The resulting schemes have been implemented in a simulation code which has been parallelized using a domain decomposition method.

We have first discussed the case of the hyperbolic operator - namely the hydrodynamics or ideal MHD equations depending on whether magnetic fields are taken into account or not - for which we have proposed the so-called GoHy class of schemes. Built in the Lagrange-remap formalism, these schemes are high-order accurate in both space and time in the non-linear regime on Cartesian grids. The principles of GoHy schemes have been described in several papers, firstly in the case of hydrodynamics [32] and then for the ideal MHD equations [93].

The most innovative feature of this new class of schemes is that it has been designed in order to take advantage of modern massively parallel computer architectures. GoHy schemes therefore rely on a particularly efficient one-dimensional scheme that uses a one-step temporal scheme (which limits the number of boundary conditions - and thus communication phases - and equation of state calls) and exclusively resorts to centered (and thus cheap) interpolation methods. They have been extended to the two-dimensional case thanks to high-order accurate dimensional splitting techniques. This dimensionally split strategy is an innovative approach for orders that are strictly greater than 2 that presents a major advantage from the performance point of view. It indeed allows the development of cache-oblivious implementations that mostly operate on data located in the processor's cache memory. Besides, GoHy schemes have been combined with artificial viscosity models for controlling oscillations phenomena that naturally occur with high-order schemes and preserving symmetry. Contrarily to classical limiting techniques, such artificial viscosity approaches do not impose to perform conditional tests and thus reveal relatively cheap in terms of computational cost.

We then have carried out some practical tests. In particular, we have seen that the experimental order of convergence of GoHy schemes matches the theoretical one and that their robustness features are satisfying. Above all, we have illustrated the ability of GoHy schemes to take advantage of modern computer architectures through performance and parallel efficiency measurements. These are indeed able to exploit up to 40% of recent processors' peak performance while the parallel efficiency of our implementation reaches 95% on 256 processes.

In order to allow the simulation of ICF implosion processes in the direct drive context, we have combined GoHy schemes with finite volume methods for solving the thermal and resistive conduction terms and taking the magnetic field generation as well as the Hall and Nernst effects into account. In this part of our study, we have not focused on achieving high-order accuracy but rather on solving the complete two-temperature resistive MHD model in reasonable restitution times. We therefore have proposed classical first-order accurate discretizations of the above-mentioned operators and have paid particular attention to their implementation, especially to the semi-implicit treatment of diffusion terms since it implies the resolution of large sparse linear systems.

The code that has been developed has finally been used for performing numerical simulations on a test problem that accurately describes the deceleration phase of an ICF capsule. In this context, we have emulated the presence of sphericity deviations on the outer surface of the shell by introducing numerical perturbations with the help of Legendre polynomials. Several conclusions can be stated about the numerical experiments that we have carried out.

- On the one hand, we have been able to perform computations for the complete two-temperature resistive MHD model with our simulation code on very fine meshes, more precisely up to meshes of 64 million cells (*i.e.* almost 400 million unknowns) distributed among 6400 processes, showing the ability of the solver we have developed to take advantage of modern supercomputers. However, it seems that the numerical results that have been obtained have still not reached convergence.

- On the other hand, we have been able to study self-generated magnetic field terms on the basis of preliminary simulation results. We have seen that these reach magnitudes of several tens megagauss but do not modify the topology of the plasma. The instabilities growth rates are slightly reduced but such levels of magnetic field do not seem to noticeably modify the temperature distribution.

The GoHy schemes' implementation is now used by several teams as a benchmark for assessing performances of recent computer architectures in the area of high performance computimg.

Perspectives

Several points could be improved in the work that we have presented. From the numerical point of view, one of the improvement paths would consist in building second-order schemes for solving the diffusion and source terms operators. This approach would probably lead to expensive numerical methods in terms of implementation and computational cost but it could help accelerating convergence and allow the accurate estimation of instabilities growth rates as in [80].

To that end, it would also be beneficial to carry out further performance optimizations, especially in the frame of the thermal and resistive conduction systems resolution. Indeed, we have seen that the Jacobi preconditioner used until now to reduce the number of iterations in iterative solvers in not efficient enough and should be replaced with a more sophisticated one (multigrid preconditioner for example).

Finally, we have seen that self-generated magnetic field terms reach very high magnitudes but only have slight effects on the plasma. Yet, these should have a strong impact on the transport of charged particles, trapping them around magnetic field lines and making them deliver more energy in the vicinity of the hot spot. It would therefore seem interesting to couple the simulation code that we have developed with a particle transport code. Collaborations that aim at doing so have already been initiated.

Appendix A

From kinetic to fluid equations

For the sake of simplicity, we drop the α subscript and take $c = 1$ hereafter. Let $f(\mathbf{x}, \mathbf{v}, t)$ denote a distribution function whose evolution is governed by the Botzmann equation:

$$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \frac{Ze}{m} (\mathbf{E} + \mathbf{v} \wedge \mathbf{B}) \cdot \nabla_{\mathbf{v}} f = \sum_{\beta} C(f, f_\beta). \tag{A.1}$$

Before deriving the fluid equations from (A.1), we recall a few definitions:

$n = \int f \, d\mathbf{v}$	density,
$n\mathbf{u} = \int f\mathbf{v} \, d\mathbf{v}$	momentum,
$ne = \frac{1}{2} \int f\|\mathbf{v}\|^2 \, d\mathbf{v}$	total energy,
$\overline{\overline{P}} = m \int f(\mathbf{v} - \mathbf{u}) \otimes (\mathbf{v} - \mathbf{u}) \, d\mathbf{v}$	constraint tensor,
$\mathbf{q} = \frac{m}{2} \int f\|\mathbf{v} - \mathbf{u}\|^2 (\mathbf{v} - \mathbf{u}) \, d\mathbf{v}$	heat flux,
$\mathbf{R}_\beta = m \int C_\beta(f, f_\beta)(\mathbf{v} - \mathbf{u}) \, d\mathbf{v}$	mean change in momentum due to collisions,
$Q_\beta = \frac{m}{2} \int C_\beta(f, f_\beta)\|\mathbf{v} - \mathbf{u}\|^2 \, d\mathbf{v}$	mean change in internal energy due to collisions.

We also assume in the sequel that f is zero for infinite velocities and that the integral of the collision term over the velocity space is zero:

$$\int C(f, f_\beta) \, d\mathbf{v} = 0 \quad \forall \beta. \tag{A.2}$$

A.1 Mass continuity equation

We first integrate the Boltzmann equation (A.1) over the velocity space. Since $\mathbf{x}$ and $\mathbf{v}$ are independent variables, we thus obtain:

$$\partial_t \int f \, d\mathbf{v} + \nabla_{\mathbf{x}} \cdot \int f\mathbf{v} \, d\mathbf{v} + \frac{Ze}{m} \int (\mathbf{E} + \mathbf{v} \wedge \mathbf{B}) \cdot \nabla_{\mathbf{v}} f \, d\mathbf{v} = 0. \tag{A.3}$$

Note that $\mathbf{E}$ does not depend on $\mathbf{v}$ and that the i-th component of $\mathbf{v} \wedge \mathbf{B}$ does not depend on v_i. We therefore have the following relation:

$$(\mathbf{E} + \mathbf{v} \wedge \mathbf{B}) \cdot \nabla_{\mathbf{v}} f = \nabla_{\mathbf{v}} \cdot (f(\mathbf{E} + \mathbf{v} \wedge \mathbf{B})) \, . \tag{A.4}$$

Moreover, since f is zero for infinite velocities, equation (A.3) rewrites:

$$\partial_t \int f \, d\mathbf{v} + \nabla_{\mathbf{x}} \cdot \int f\mathbf{v} \, d\mathbf{v} = 0$$
$$i.e. \ \partial_t n + \nabla \cdot (n\mathbf{u}) = 0.$$

A.2 Momentum equation

We now multiply the Boltzmann equation (A.1) with $m\mathbf{v}$ and integrate it over the velocity space:

$$m\partial_t \int f\mathbf{v} \, d\mathbf{v} + m\nabla_{\mathbf{x}} \cdot \int f(\mathbf{v} \otimes \mathbf{v}) \, d\mathbf{v} + \frac{Ze}{m} \int ((\mathbf{E} + \mathbf{v} \wedge \mathbf{B}) \cdot \nabla_{\mathbf{v}} f) \, \mathbf{v} \, d\mathbf{v} = \sum_\beta \int C(f, f_\beta)\mathbf{v} \, d\mathbf{v}. \quad (A.5)$$

Using (A.4) and an integration by parts, one can show that:

$$\begin{aligned} \int ((\mathbf{E} + \mathbf{v} \wedge \mathbf{B}) \cdot \nabla_{\mathbf{v}} f) \, \mathbf{v} \, d\mathbf{v} &= \int (\nabla_{\mathbf{v}} \cdot f \, (\mathbf{E} + \mathbf{v} \wedge \mathbf{B})) \, \mathbf{v} \, d\mathbf{v}, \\ &= -\int (f\mathbf{E} + f\mathbf{v} \wedge \mathbf{B}) \, d\mathbf{v}, \\ &= -n \, (\mathbf{E} + \mathbf{u} \wedge \mathbf{B}) \, . \end{aligned}$$

Moreover, since:

$$\begin{aligned} m \int f(\mathbf{v} \otimes \mathbf{v}) \, d\mathbf{v} &= m \int f \, ((\mathbf{v} - \mathbf{u} + \mathbf{u}) \otimes (\mathbf{v} - \mathbf{u} + \mathbf{u})) \, d\mathbf{v}, \\ &= m \int f \, ((\mathbf{v} - \mathbf{u}) \otimes (\mathbf{v} - \mathbf{u})) \, d\mathbf{v} + mn\mathbf{u} \otimes \mathbf{u}, \\ &= \overline{\overline{P}} + mn\mathbf{u} \otimes \mathbf{u}, \end{aligned}$$

equation (A.5) rewrites:

$$\partial_t (mn\mathbf{u}) + \nabla_{\mathbf{x}} \cdot \left(mn\mathbf{u} \otimes \mathbf{u} + \overline{\overline{P}} \right) - \frac{Zne}{m} (\mathbf{E} + \mathbf{u} \wedge \mathbf{B}) = \sum_\beta \int C(f, f_\beta)\mathbf{v} \, d\mathbf{v}.$$

The right hand side can be expressed in terms of the mean change in momentum, indeed:

$$\begin{aligned} m \int C(f, f_\beta)\mathbf{v} \, d\mathbf{v} &= m \int C(f, f_\beta)(\mathbf{v} - \mathbf{u} + \mathbf{u}) \, d\mathbf{v}, \\ &= \mathbf{R}_\beta + m\mathbf{u} \sum_\beta \int C(f, f_\beta) \, d\mathbf{v}, \\ &= \mathbf{R}_\beta \text{ according to (A.2)}, \end{aligned}$$

so that the fluid momentum equation finally writes:

$$\partial_t (mn\mathbf{u}) + \nabla_{\mathbf{x}} \cdot \left(mn\mathbf{u} \otimes \mathbf{u} + \overline{\overline{P}} \right) = \frac{Zne}{m} (\mathbf{E} + \mathbf{u} \wedge \mathbf{B}) + \sum_\beta \mathbf{R}_\beta.$$

A.3 Total energy equation

We finally multiply equation (A.1) with $\frac{m|\mathbf{v}|^2}{2}$ and integrate it over the velocity space:

$$\frac{m}{2}\int f|\mathbf{v}|^2\,d\mathbf{v} + \frac{m}{2}\nabla_{\mathbf{x}}\cdot\int f|\mathbf{v}|^2\mathbf{v}\,d\mathbf{v} + \frac{Ze}{2m}\int f|\mathbf{v}|^2\nabla_{\mathbf{v}}\cdot(\mathbf{E}+\mathbf{v}\wedge\mathbf{B})\,d\mathbf{v} = \frac{m}{2}\sum_{\beta}\int C(f,f_\beta)|\mathbf{v}|^2\,d\mathbf{v}. \quad \text{(A.6)}$$

Once again, combining (A.4) and an integration by parts helps to show that:

$$\begin{aligned}\int f|\mathbf{v}|^2\nabla_{\mathbf{v}}\cdot(\mathbf{E}+\mathbf{v}\wedge\mathbf{B})\,d\mathbf{v} &= -2\int f\mathbf{v}\cdot(\mathbf{E}+\mathbf{v}\wedge\mathbf{B})\,d\mathbf{v},\\ &= -2\left(\int f\mathbf{v}\,d\mathbf{v}\right)\cdot\mathbf{E} - 2\int f\mathbf{v}\cdot(\mathbf{v}\wedge\mathbf{B})\,d\mathbf{v},\\ &= -2n\mathbf{u}\cdot\mathbf{E} \text{ since } \mathbf{v} \text{ and } \mathbf{v}\wedge\mathbf{B} \text{ are orthogonal vectors.}\end{aligned}$$

We now develop the following term:

$$\begin{aligned}\int f|\mathbf{v}|^2\mathbf{v}\,d\mathbf{v} &= \int f|\mathbf{v}|^2(\mathbf{v}-\mathbf{u}+\mathbf{u})\,d\mathbf{v},\\ &= 2ne\mathbf{u} + \int f|\mathbf{v}|^2(\mathbf{v}-\mathbf{u})\,d\mathbf{v},\\ &= 2ne\mathbf{u} + \int f|\mathbf{v}-\mathbf{u}+\mathbf{u}|^2(\mathbf{v}-\mathbf{u})\,d\mathbf{v},\\ &= 2ne\mathbf{u} + \int f|\mathbf{v}-\mathbf{u}|^2(\mathbf{v}-\mathbf{u})\,d\mathbf{v} + 2\left(\int f(\mathbf{v}-\mathbf{u})\otimes(\mathbf{v}-\mathbf{u})\,d\mathbf{v}\right)\mathbf{u} + |\mathbf{u}|^2\int f(\mathbf{v}-\mathbf{u})\,d\mathbf{v}.\end{aligned}$$

One can easily note that the last right hand side term is zero and thus:

$$\frac{m}{2}\int f|\mathbf{v}|^2\mathbf{v}\,d\mathbf{v} = mne\mathbf{u} + \overline{\overline{P}}\mathbf{u} + \mathbf{q}.$$

Equation (A.6) then rewrites:

$$\partial_t(mne) + \nabla_{\mathbf{x}}\cdot\left(mne\mathbf{u} + \overline{\overline{P}}\mathbf{u} + \mathbf{q}\right) - Zne\mathbf{u}\cdot\mathbf{E} = \sum_{\beta}\int C(f,f_\beta)|\mathbf{v}|^2\,d\mathbf{v}.$$

The right hand side can be expressed in terms of the mean changes in momentum and total energy, indeed:

$$\begin{aligned}\frac{m}{2}\int C(f,f_\beta)|\mathbf{v}|^2\,d\mathbf{v} &= \frac{m}{2}\int C(f,f_\beta)|\mathbf{v}-\mathbf{u}+\mathbf{u}|^2\,d\mathbf{v}\\ &= \frac{m}{2}\int C(f,f_\beta)|\mathbf{v}-\mathbf{u}|^2\,d\mathbf{v} + m\mathbf{u}\cdot\int C(f,f_\beta)(\mathbf{v}-\mathbf{u})\,d\mathbf{v} + |u|^2\int C(f,f_\beta)\,d\mathbf{v}\\ &= Q_\beta + \mathbf{u}\cdot\mathbf{R}_\beta \text{ according to (A.2)},\end{aligned}$$

so that the fluid total energy equation finally writes:

$$\partial_t(mne) + \nabla_{\mathbf{x}}\cdot\left(mne\mathbf{u} + \overline{\overline{P}}\mathbf{u} + \mathbf{q}\right) - Zne(\mathbf{u}\cdot\mathbf{E}) = \sum_{\beta}\left(Q_\beta + \mathbf{u}\cdot\mathbf{R}_\beta\right).$$

Appendix B

Building a third-order accurate dimensional splitting method

Third-order accurate splitting methods can be built by taking $p = 3$ in equation (4.32). In order to determine splitting sequences that allow to reach third-order accuracy, we compare the Taylor expansion of the exact operator:

$$\exp\left(\Delta t(\mathcal{A}+\mathcal{B})\right)$$

with the Taylor expansion of the approximate operator:

$$\prod_{i=1}^{3} \exp\left(a_i \Delta t \mathcal{A}\right) \cdot \exp\left(b_i \Delta t \mathcal{B}\right))$$

and identify the coefficients in both expressions. We thus obtain the following non-linear system ($C_{\mathcal{X}}$ denotes the coefficient that is in factor of the $\mathcal{X}$ term):

$$\begin{aligned}
C_{\mathcal{A}} &= a_1 + a_2 + a_3 &&= 1,\\
C_{\mathcal{B}} &= b_1 + b_2 + b_3 &&= 1,\\
2C_{\mathcal{A}^2} &= a_1^2 + a_2^2 + a_3^2 + 2a_1a_2 + 2a_1a_3 + 2a_2a_3 &&= 1,\\
2C_{\mathcal{B}^2} &= b_1^2 + b_2^2 + b_3^2 + 2b_1b_2 + 2b_1b_3 + 2b_2b_3 &&= 1,\\
C_{\mathcal{AB}} &= a_1b_1 + a_1b_2 + a_1b_3 + a_2b_2 + a_2b_3 + a_3b_3 &&= 1/2,\\
C_{\mathcal{BA}} &= b_1a_2 + b_1a_3 + b_2a_3 &&= 1/2,\\
6C_{\mathcal{A}^3} &= a_1^3 + a_2^3 + a_3^3 + 3a_1^2a_2 + 3a_1^2a_3 + 3a_2^2a_1 + 3a_2^2a_3 + 6a_1a_2a_3 &&= 1,\\
6C_{\mathcal{B}^3} &= b_1^3 + b_2^3 + b_3^3 + 3b_1^2b_2 + 3b_1^2b_3 + 3b_2^2b_1 + 3b_2^2b_3 + 6b_1b_2b_3 &&= 1,\\
2C_{\mathcal{A}^2\mathcal{B}} &= a_1^2b_1 + a_1^2b_2 + a_1^2b_3 + a_2^2b_2 + a_2^2b_3 + a_3^2b_3 + 2a_1a_2b_2 + 2a_1a_3b_3 + 2a_1a_2b_3 + 2a_2a_3b_3 &&= 1/3,\\
2C_{\mathcal{AB}^2} &= a_1b_1^2 + a_1b_2^2 + a_1b_3^2 + a_2b_2^2 + a_2b_3^2 + a_3b_3^2 + 2a_1b_1b_2 + 2a_1b_2b_3 + 2a_1b_1b_3 + 2a_2b_2b_3 &&= 1/3,\\
2C_{\mathcal{B}^2\mathcal{A}} &= b_1^2a_2 + b_1^2a_3 + b_2^2a_3 + 2b_1b_2a_3 &&= 1/3,\\
2C_{\mathcal{BA}^2} &= b_1a_2^2 + b_1a_3^2 + b_2a_3^2 + 2b_1a_2a_3 &&= 1/3,\\
C_{\mathcal{ABA}} &= a_1b_1a_2 + a_1b_1a_3 + a_2b_2a_3 + a_1b_2a_3 &&= 1/6,\\
C_{\mathcal{BAB}} &= b_1a_2b_2 + b_1a_2b_3 + b_1a_3b_3 + b_2a_3b_3 &&= 1/6.
\end{aligned}$$

We consider the equations coming from the $\mathcal{A}$, $\mathcal{B}$, $\mathcal{BA}$, $\mathcal{B}^2\mathcal{A}$ and $\mathcal{A}^2\mathcal{B}$ coefficients as reference equations:

$$\begin{cases} a_1 + a_2 + a_3 &= 1, \\ b_1 + b_2 + b_3 &= 1, \\ a_2 b_1 + a_3 b_1 + a_3 b_2 &= 1/2, \\ b_1^2 a_2 + b_1^2 a_3 + b_2^2 a_3 + 2 b_1 b_2 a_3 &= 1/3, \\ a_1^2 b_1 + a_1^2 b_2 + a_1^2 b_3 + a_2^2 b_2 + a_2^2 b_3 + a_3^2 b_3 + 2 a_1 a_2 b_2 + 2 a_1 a_3 b_3 + 2 a_1 a_2 b_3 + 2 a_2 a_3 b_3 &= 1/3. \end{cases} \tag{B.1}$$

Note that systems (B.1) and (4.35) are equivalent since:

$$\begin{aligned} 2C_{\mathcal{B}^2\mathcal{A}} &= b_1^2 a_2 + b_1^2 a_3 + b_2^2 a_3 + 2 b_1 b_2 a_3 \\ &= b_1^2 a_2 + a_3 (b_1 + b_2)^2, \end{aligned}$$

and:

$$\begin{aligned} 2C_{\mathcal{A}^2\mathcal{B}} &= a_1^2 b_2 + a_1^2 b_3 + a_2^2 b_2 + a_2^2 b_3 + a_3^2 b_3 + 2 a_1 a_2 b_2 + 2 a_1 a_3 b_3 + 2 a_1 a_2 b_3 + 2 a_2 a_3 b_3 \\ &= b_3 (a_1^2 + a_2^2 + a_3^2 + 2 a_1 a_2 + 2 a_1 a_3 + 2 a_2 a_3) + b_2 (a_1^2 + a_2^2 + 2 a_1 a_2) + b_1 a_1^2 \\ &= b_3 (a_1 + a_2 + a_3)^2 + b_2 (a_1 + a_2)^2 + b_1 a_1^2 \\ &= b_3 + b_2 (a_1 + a_2)^2 + b_1 a_1^2. \end{aligned}$$

We now would like to show that the other non-linear equations that have been obtained by identifying coefficients in the Taylor expansions of the exact and approximate operators can actually be obtained from the five reference equations given in (B.1). Indeed:

$$\begin{aligned} 2C_{\mathcal{A}^2} &= (a_1 + a_2 + a_3)^2 \\ &= 1, \\ 6C_{\mathcal{A}^3} &= (a_1 + a_2 + a_3)^3 \\ &= 1, \\ 2C_{\mathcal{B}^2} &= (b_1 + b_2 + b_3)^2 \\ &= 1, \\ 6C_{\mathcal{B}^3} &= (b_1 + b_2 + b_3)^3 \\ &= 1, \\ C_{\mathcal{AB}} &= a_1 b_1 + a_1 b_2 + a_1 b_3 + a_2 b_2 + a_2 b_3 + a_3 b_3 \\ &= a_1 (b_1 + b_2 + b_3) + a_2 (b_2 + b_3) + a_3 b_3 \\ &= a_1 + a_2 (1 - b_1) + a_3 (1 - b_1 - b_2) \\ &= (a_1 + a_2 + a_3) - (b_1 a_2 + b_1 a_3 + b_2 a_3) \\ &= 1 - 1/2 \\ &= 1/2, \end{aligned}$$

$$
\begin{aligned}
C_{\mathcal{BAB}} &= b_1a_2b_2 + b_1a_2b_3 + b_1a_3b_3 + b_2a_3b_3 \\
&= b_1a_2b_2 + (b_1a_2 + b_1a_3 + b_2a_3)b_3 \\
&= b_1a_2b_2 + (b_1a_2 + b_1a_3 + b_2a_3)(1 - b_1 - b_2) \\
&= (b_1a_2 + b_1a_3 + b_2a_3) - (b_1^2a_2 + b_1^2a_3 + 2b_1b_2a_3 + b_2^2a_3) \\
&= 1/2 - 1/3 \\
&= 1/6, \\
2C_{\mathcal{AB}^2} &= a_1b_1^2 + a_1b_2^2 + a_1b_3^2 + a_2b_2^2 + a_2b_3^2 + a_3b_3^2 + 2a_1b_1b_2 + 2a_1b_2b_3 + 2a_1b_1b_3 + 2a_2b_2b_3 \\
&= (a_1 + a_2 + a_3)(b_1 + b_2 + b_3)^2 - (a_2b_1^2 + a_3b_1^2 + a_3b_2^2 + +2a_3b_1b_2 + 2a_2b_1b_2 + 2a_2b_1b_3 + 2a_3b_1b_3 + 2a_3b_2b_3) \\
&= 1 - (a_3(b_1 + b_2)^2 + a_2b_1^2 + 2(a_2b_1b_2 + a_2b_1b_3 + a_3b_1b_3 + a_3b_2b_3)) \\
&= 1 - (1/3 + 1/3) \\
&= 1/3, \\
2C_{\mathcal{BA}^2} &= b_1a_2^2 + b_1a_3^2 + b_2a_3^2 + 2b_1a_2a_3 \\
&= (a_1 + a_2 + a_3)^2(b_1 + b_2 + b_3) - (2C_{\mathcal{A}^2\mathcal{B}} + 2C_{\mathcal{BAB}}) \\
&= 1 - (1/3 + 1/3) \\
&= 1/3, \\
C_{\mathcal{ABA}} &= a_1b_1a_2 + a_1b_1a_3 + a_2b_2a_3 + a_1b_2a_3 \\
&= a_2b_2a_3 + (b_1a_2 + b_1a_3 + b_2a_3)a_1 \\
&= a_2b_2a_3 + (b_1a_2 + b_1a_3 + b_2a_3)(1 - a_2 - a_3) \\
&= (b_1a_2 + b_1a_3 + b_2a_3) - (b_1a_2^2 + b_1a_3^2 + b_2a_3^2 + 2b_1a_2a_3) \\
&= 1/2 - 1/3 \\
&= 1/6.
\end{aligned}
$$

This shows that the a_i and b_i coefficients that allow to build a third-order accurate splitting method are solution of the following system of non-linear equations:

$$
\begin{cases}
a_1 + a_2 + a_3 &= 1, \\
b_1 + b_2 + b_3 &= 1, \\
a_2b_1 + a_3b_1 + a_3b_2 &= 1/2, \\
a_2b_1^2 + a_3(b_1 + b_2)^2 &= 1/3, \\
b_3 + b_2(a_1 + a_2)^2 + b_1a_1^2 &= 1/3.
\end{cases}
$$

Appendix C

Unified notations for planar and axisymmetric cases

This appendix aims at introducing a unified system of notations that is suitable for both planar and axisymmetric cases. First of all, we define the so-called geometry indicator parameter denoted by α whose value is given by:

$$\alpha = \begin{cases} 0 & \text{in the planar case,} \\ 1 & \text{in the axisymmetric case.} \end{cases}$$

We also propose to consider from now on the following vector basis:

$$(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) = \begin{cases} (\mathbf{e}_x,\ \mathbf{e}_y,\ \mathbf{e}_z) & \text{in the planar case,} \\ (\mathbf{e}_z,\ \mathbf{e}_r,\ \mathbf{e}_\theta) & \text{in the axisymmetric case.} \end{cases} \tag{C.1}$$

The integer subscripts introduced in (C.1) will be used for denoting spatial coordinates:

$$(x_1,\ x_2,\ x_3) = \begin{cases} (x,\ y,\ z) & \text{in the planar case,} \\ (z,\ r,\ \theta) & \text{in the axisymmetric case,} \end{cases}$$

as well as vector components:

$$\mathbf{u} = (u_1,\ u_2,\ u_3) = \begin{cases} (u_x,\ u_y,\ u_z) & \text{in the planar case,} \\ (u_z,\ u_r,\ u_\theta) & \text{in the axisymmetric case,} \end{cases}$$

and spatial derivatives along a given direction:

$$\partial_i f = \frac{\partial f}{\partial x_i},\ i \in \{1,\ 2,\ 3\}.$$

Remark 38. Note that $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ forms a direct basis in both planar and axisymmetric cases.

We finally provide the expression of several operators for the specific two-dimensional case in Table C.1 using the unified system of notations (C.1).

Operator definition	**Vector basis**		
	$(\mathbf{e}_x,\ \mathbf{e}_y,\ \mathbf{e}_z)$	$(\mathbf{e}_z,\ \mathbf{e}_r,\ \mathbf{e}_\theta)$	$(\mathbf{e}_1,\ \mathbf{e}_2,\ \mathbf{e}_3)$
$\nabla \cdot \mathbf{u}$ (divergence)	$\partial_x u_x + \partial_y u_y$	$\partial_z u_z + \frac{1}{r}\partial_r\left(r u_r\right)$	$\partial_1 u_1 + \frac{1}{x_2^\alpha}\partial_2\left(x_2^\alpha u_2\right)$
$\nabla \times \mathbf{u}$ (curl)	$\begin{pmatrix} \partial_y u_z \\ -\partial_x u_z \\ \partial_x u_y - \partial_y u_x \end{pmatrix}$	$\begin{pmatrix} \frac{1}{r}\partial_r\left(r u_\theta\right) \\ -\partial_z u_\theta \\ \partial_z u_r - \partial_r u_z \end{pmatrix}$	$\begin{pmatrix} \frac{1}{x_2^\alpha}\partial_2\left(x_2 u_3\right) \\ -\partial_1 u_3 \\ \partial_1 u_2 - \partial_2 u_1 \end{pmatrix}$
$\overline{\overline{T}} \cdot \mathbf{u}$ with $\mathbf{h} = \mathbf{e}_3$ (see page 30)	$\begin{pmatrix} T^\perp u_x - T^\wedge u_y \\ T^\perp u_y + T^\wedge u_y \\ 0 \end{pmatrix}$	$\begin{pmatrix} T^\perp u_z - T^\wedge u_r \\ T^\perp u_r + T^\wedge u_r \\ 0 \end{pmatrix}$	$\begin{pmatrix} T^\perp u_1 - T^\wedge u_2 \\ T^\perp u_2 + T^\wedge u_1 \\ 0 \end{pmatrix}$

Table C.1: Expression of several operators for the two-dimensional case in planar geometry, axisymmetric geometry and using the unified system of notations introduced in (C.1).

Appendix D

Estimation of the thermoelectric, conductivity and resistivity tensors at cell interfaces

We consider in this appendix a set of two adjacent cells Ω_1 and Ω_2 separated by a vertical edge $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$ and adopt the following notations:

- ϕ_i denotes the cell-centered value of ϕ in the Ω_i cell;
- $\phi_\star$ denotes the value of ϕ on the cell interface Γ;
- $\phi_\pm$ denote the value of ϕ on the vertices of Γ.

Remark 39. The approximations of the thermoelectric, conductivity and resistivity tensors on horizontal edges being similar to their approximations on vertical edges, we only discuss the case of vertical edges below.

D.1 Conductivity tensor and thermoelectric tensor approximations

We here deal with the following diffusion equation:

$$\partial_t T + \nabla \cdot \mathbf{q} = 0,$$

where the heat flux $\mathbf{q}$ writes:

$$\mathbf{q} = -\overline{\overline{\kappa}} \cdot \nabla T \quad \text{with} \quad \overline{\overline{\kappa}} = \begin{pmatrix} \kappa^O rth & -\kappa^\wedge \\ \kappa^\wedge & \kappa^\perp \end{pmatrix}.$$

In the two-dimensional case, using notations introduced in Appendix C, the heat flux writes:

$$\mathbf{q} = -\begin{pmatrix} \kappa^\perp \partial_1 T - \kappa^\wedge \partial_2 T \\ \kappa^\perp \partial_2 T + \kappa^\wedge \partial_1 T \\ 0 \end{pmatrix}.$$

Using basic approximations, the normal component of the heat flux in the Ω_1 and Ω_2 cells is given by:

$$(\mathbf{q}\cdot\mathbf{n})_1 = -\kappa_1^{\perp}\left(\frac{T_\star - T_1}{\Delta x_1/2}\right) + \kappa_1^{\wedge}\left(\frac{T_+ - T_-}{\Delta x_2}\right),$$
$$(\mathbf{q}\cdot\mathbf{n})_2 = -\kappa_2^{\perp}\left(\frac{T_2 - T_\star}{\Delta x_1/2}\right) + \kappa_2^{\wedge}\left(\frac{T_+ - T_-}{\Delta x_2}\right).$$

For the sake of conservativity, the normal component of the heat flux has to be continuous across Γ, imposing that:

$$(\mathbf{q}\cdot\mathbf{n})_1 = (\mathbf{q}\cdot\mathbf{n})_2,$$

which leads to the following expression of $T_\star$:

$$T_\star = \frac{\kappa_1^{\perp}T_1 + \kappa_2^{\perp}T_2}{\kappa_1^{\perp}+\kappa_2^{\perp}} - \frac{\Delta x_1}{2\Delta x_2}\left(\frac{\kappa_1^{\wedge}(T_+ - T_-) - \kappa_2^{\wedge}(T_+ - T_-)}{\kappa_1^{\perp}+\kappa_2^{\perp}}\right). \tag{D.1}$$

Injecting (D.1) in the heat flux definition leads to:

$$\mathbf{q}\cdot\mathbf{n} = -\left(2\frac{\kappa_1^{\perp}\kappa_2^{\perp}}{\kappa_1^{\perp}+\kappa_2^{\perp}}\right)\frac{T_2 - T_1}{\Delta x_1} + \left(\frac{\kappa_1^{\perp}\kappa_2^{\wedge} + \kappa_2^{\perp}\kappa_1^{\wedge}}{\kappa_1^{\perp}+\kappa_2^{\perp}}\right)\frac{T_+ - T_-}{\Delta x_2},$$

which motivates the choice of the following cell interface representation of the conductivity tensor $\overline{\overline{\kappa}}$:

$$\kappa_\Gamma^{\perp} = 2\frac{\kappa_1^{\perp}\kappa_2^{\perp}}{\kappa_1^{\perp}+\kappa_2^{\perp}}, \quad \kappa_\Gamma^{\wedge} = \frac{\kappa_1^{\perp}\kappa_2^{\wedge} + \kappa_2^{\perp}\kappa_1^{\wedge}}{\kappa_1^{\perp}+\kappa_2^{\perp}}.$$

We now apply the same methodology to the following term that is involved in the electronic heat flux:

$$T_e(\overline{\overline{\beta}}\cdot\mathbf{J}) = T_e\begin{pmatrix} \frac{\beta^{\perp}}{x_2^{\alpha}}\partial_2\left(x_2^{\alpha}B\right) + \beta^{\wedge}\partial_1 B \\ -\beta^{\perp}\partial_1 B - \frac{\beta^{\wedge}}{x_2^{\alpha}}\partial_2\left(x_2^{\alpha}B\right) \\ 0 \end{pmatrix}. \tag{D.2}$$

Using basic approximations, imposing the continuity of (D.2) leads to the following estimation of $B_\star$:

$$B_\star = \frac{\beta_1^{\wedge}B_1 + \beta_2^{\wedge}B_2}{\beta_1^{\wedge}+\beta_2^{\wedge}} - \frac{\Delta x_1}{2\Delta x_2}\left(\frac{\beta_1^{\perp}(B_+ - B_-) - \beta_2^{\perp}(B_+ - B_-)}{\beta_1^{\wedge}+\beta_2^{\wedge}}\right), \tag{D.3}$$

and thus to the following cell interface representation of the thermoelectric tensor $\overline{\overline{\beta}}$:

$$\beta_\Gamma^{\perp} = -\frac{\beta_1^{\perp}\beta_2^{\wedge} + \beta_2^{\perp}\beta_1^{\wedge}}{\beta_1^{\wedge}+\beta_2^{\wedge}}, \quad \beta_\Gamma^{\wedge} = -2\frac{\beta_1^{\wedge}\beta_2^{\wedge}}{\beta_1^{\wedge}+\beta_2^{\wedge}}.$$

D.2 Resistivity tensor approximations

We deal in this section with the following resistive conduction equation:

$$\partial_t\mathbf{B} + \nabla\times\left(\overline{\overline{\rho}}\cdot\mathbf{J}\right) = 0 \quad \text{with} \quad \mathbf{J} = \nabla\times\mathbf{B}$$

with $\mathbf{B} = B\mathbf{e}_3$. In the two-dimensional case, using notations introduced in Appendix C, the $\overline{\overline{\rho}} \cdot \mathbf{J}$ term writes:

$$\overline{\overline{\rho}} \cdot \mathbf{J} = \frac{c}{\mu} \begin{pmatrix} \frac{\rho^{\perp}}{x_2^{\alpha}} \partial_2 \left(x_2^{\alpha} B \right) + \rho^{\wedge} \partial_1 B \\ -\rho^{\perp} \partial_1 B - \frac{\rho^{\wedge}}{x_2^{\alpha}} \partial_2 \left(x_2^{\alpha} B \right) \\ 0 \end{pmatrix}$$

Using basic approximations, the tangential component of the resistivity flux in the Ω_1 and Ω_2 cells is given by:

$$\begin{aligned} \left((\overline{\overline{\rho}} \cdot \mathbf{J}) \cdot \mathbf{t} \right)_1 &= -\rho_1^{\perp} \left(\frac{B_{\star} - B_1}{\Delta x_1 / 2} \right) - \rho_1^{\wedge} \left(\frac{(x_2)_+^{\alpha} T_+ - (x_2)_-^{\alpha} T_-}{((x_2)_+^{\alpha} + (x_2)_-^{\alpha}) \Delta x_2 / 2} \right), \\ \left((\overline{\overline{\rho}} \cdot \mathbf{J}) \cdot \mathbf{t} \right)_2 &= -\rho_2^{\perp} \left(\frac{B_2 - B_{\star}}{\Delta x_1 / 2} \right) - \rho_2^{\wedge} \left(\frac{(x_2)_+^{\alpha} T_+ - (x_2)_-^{\alpha} T_-}{((x_2)_+^{\alpha} + (x_2)_-^{\alpha}) \Delta x_2 / 2} \right). \end{aligned}$$

For the sake of conservativity, the tangential component of the resistivity flux has to be continuous across Γ, imposing that:

$$\left((\overline{\overline{\rho}} \cdot \mathbf{J}) \cdot \mathbf{t} \right)_1 = \left((\overline{\overline{\rho}} \cdot \mathbf{J}) \cdot \mathbf{t} \right)_2,$$

which leads after a few manipulations to the following resistivity tensor expressions on the cell interface Γ:

$$\rho_{\Gamma}^{\perp} = 2 \frac{\rho_1^{\perp} \rho_2^{\perp}}{\rho_1^{\perp} + \rho_2^{\perp}}, \quad \rho_{\Gamma}^{\wedge} = \frac{\rho_1^{\perp} \rho_2^{\wedge} + \rho_2^{\perp} \rho_1^{\wedge}}{\rho_1^{\perp} + \rho_2^{\perp}}.$$

Appendix E

Positivity of the electronic internal energy variation for the resistive conduction operator

In this section, we aim at showing that the following electronic internal energy variation:

$$\mathcal{A}_{i,j} = \frac{(\rho_e \epsilon_e)_{i,j}^{n+1} - (\rho_e \epsilon_e)_{i,j}^{n}}{\Delta t} = \frac{(\rho e)_{i,j}^{n+1} - (\rho e)_{i,j}^{n}}{\Delta t} - \frac{1}{\Delta t}\left(\frac{\left(B_{i,j}^{n+1}\right)^2}{2\mu} - \frac{\left(B_{i,j}^{n}\right)^2}{2\mu} \right)$$

used for updating the $(\rho_e \epsilon_e)$ when dealing with resistive conduction is positive. In this context, we recall that the magnetic field B and the total energy E are updated according to the following finite volume schemes:

$$\frac{\overline{B}_{i,j}^{n+1} - \overline{B}_{i,j}^{n}}{\Delta t} + \frac{c^2 \Delta t}{\mu \Delta x_1}\left(R^*_{i+\frac{1}{2},j} - R^*_{i-\frac{1}{2},j} \right) - \frac{c^2 \Delta t}{\mu \Delta x_2}\left(R^*_{i,j+\frac{1}{2}} - R^*_{i,j-\frac{1}{2}} \right) = 0, \tag{E.1}$$

$$\frac{\overline{(\rho e)}_{i,j}^{n+1} - \overline{(\rho e)}_{i,j}^{n}}{\Delta t} - \frac{c^2}{\mu^2}\left[\frac{1}{\Delta x_1}\left(E^*_{i+\frac{1}{2},j} - E^*_{i-\frac{1}{2},j} \right) + \frac{1}{(x_2^{\alpha})_j \Delta x_2}\left(E^*_{i,j+\frac{1}{2}} - E^*_{i,j-\frac{1}{2}} \right) \right] = 0 \tag{E.2}$$

where the R^* and E^* fluxes are defined in section 7.1.2. One can notice that:

$$\begin{aligned}
\left(B_{i,j}^{n+1}\right)^2 - \left(B_{i,j}^{n}\right)^2 &= \left(B_{i,j}^{n+1} - B_{i,j}^{n}\right)\left(B_{i,j}^{n+1} + B_{i,j}^{n}\right) \\
&= \left(B_{i,j}^{n+1} - B_{i,j}^{n}\right)\left(2B_{i,j}^{n+1} - (B_{i,j}^{n+1} - B_{i,j}^{n})\right) \\
&= 2B_{i,j}^{n+1}\left(B_{i,j}^{n+1} - B_{i,j}^{n}\right) - \left(B_{i,j}^{n+1} - B_{i,j}^{n}\right)^2
\end{aligned}$$

so that $\mathcal{A}_{i,j}$ is given by:

$$\mathcal{A}_{i,j} = \underbrace{\frac{(\rho e)_{i,j}^{n+1} - (\rho e)_{i,j}^{n}}{\Delta t} - \frac{B_{i,j}^{n+1}}{\mu}\left(\frac{B_{i,j}^{n+1} - B_{i,j}^{n}}{\Delta t} \right)}_{\mathcal{A}'_{i,j}} + \frac{1}{2\mu \Delta t}\left(B_{i,j}^{n+1} - B_{i,j}^{n}\right)^2. \tag{E.3}$$

The last right hand side term in (E.3) being positive, we discuss the sign of $\mathcal{A}'_{i,j}$ from now on. Its detailed form is given by:

$$\begin{aligned}\mathcal{A}'_{i,j} = &+\frac{c^2}{\mu^2}\left[\frac{1}{\Delta x_1}\left(E^*_{i+\frac{1}{2},j} - E^*_{i-\frac{1}{2},j}\right) + \frac{1}{(x_2^\alpha)_j \Delta x_2}\left(E^*_{i,j+\frac{1}{2}} - E^*_{i,j-\frac{1}{2}}\right)\right] \\ &-\frac{B^{n+1}_{i,j}}{\mu}\left[\frac{c^2\Delta t}{\mu\Delta x_2}\left(R^*_{i,j+\frac{1}{2}} - R^*_{i,j-\frac{1}{2}}\right) - \frac{c^2\Delta t}{\mu\Delta x_1}\left(R^*_{i+\frac{1}{2},j} - R^*_{i-\frac{1}{2},j}\right)\right].\end{aligned} \tag{E.4}$$

Injecting the definition of the R^* and E^* fluxes in (E.4) leads after a few manipulations to the following expression of $\mathcal{A}'_{i,j}$:

$$\mathcal{A}'_{i,j} = \mathcal{A}^{\perp}_{i,j} + \frac{c^2}{2\mu^2 (x_2^\alpha)_j \Delta x_1 \Delta x_2}\mathcal{A}^{\wedge}_{i,j},$$

where $\mathcal{A}^{\perp}_{i,j}$ is the following positive term:

$$\begin{aligned}\mathcal{A}^{\perp}_{i,j} = &+\frac{c^2}{2\mu^2\Delta t\Delta x_1^2}\left[(\rho^\perp)^n_{i+\frac{1}{2},j}\left(B^{n+1}_{i+1,j} - B^{n+1}_{i,j}\right)^2 + (\rho^\perp)^n_{i-\frac{1}{2},j}\left(B^{n+1}_{i-1,j} - B^{n+1}_{i,j}\right)^2\right] \\ &+\frac{c^2(\rho^\perp)^n_{i,j+\frac{1}{2}}}{2\mu^2\Delta t (x_2^\alpha)_j (x_2^\alpha)_{j+\frac{1}{2}}\Delta x_2^2}\left((x_2^\alpha)_{j+1}B^{n+1}_{i,j+1} - (x_2^\alpha)_j B^{n+1}_{i,j}\right)^2 \\ &+\frac{c^2(\rho^\perp)^n_{i,j-\frac{1}{2}}}{2\mu^2\Delta t (x_2^\alpha)_j (x_2^\alpha)_{j-\frac{1}{2}}\Delta x_2^2}\left((x_2^\alpha)_{j-1}B^{n+1}_{i,j-1} - (x_2^\alpha)_j B^{n+1}_{i,j}\right)^2,\end{aligned}$$

and $\mathcal{A}^{\wedge}_{i,j}$ is given by:

$$\begin{aligned}\mathcal{A}^{\wedge}_{i,j} = &+(\rho^\wedge)^n_{i,j+\frac{1}{2}}\left((x_2^\alpha)_{j+1}B^{n+1}_{i,j+1} - (x_2^\alpha)_j B^{n+1}_{i,j}\right)\left(B^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} - B^{n+1}_{i-\frac{1}{2},j+\frac{1}{2}}\right) \\ &-(\rho^\wedge)^n_{i+\frac{1}{2},j}\left(B^{n+1}_{i+1,j} - B^{n+1}_{i,j}\right)\left((x_2^\alpha)_{j+\frac{1}{2}}B^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} - (x_2^\alpha)_{j-\frac{1}{2}}B^{n+1}_{i+\frac{1}{2},j-\frac{1}{2}}\right) \\ &+(\rho^\wedge)^n_{i,j-\frac{1}{2}}\left((x_2^\alpha)_j B^{n+1}_{i,j} - (x_2^\alpha)_{j-1}B^{n+1}_{i,j-1}\right)\left(B^{n+1}_{i+\frac{1}{2},j-\frac{1}{2}} - B^{n+1}_{i-\frac{1}{2},j-\frac{1}{2}}\right) \\ &-(\rho^\wedge)^n_{i-\frac{1}{2},j}\left(B^{n+1}_{i,j} - B^{n+1}_{i-1,j}\right)\left((x_2^\alpha)_{j+\frac{1}{2}}\left(B^{n+1}_{i-\frac{1}{2},j+\frac{1}{2}} - (x_2^\alpha)_{j-\frac{1}{2}}B^{n+1}_{i-\frac{1}{2},j-\frac{1}{2}}\right)\right).\end{aligned}$$

If the resistivity tensor is diagonal, (*i.e.* if $\rho^\wedge$ is identically zero), $\mathcal{A}_{i,j}$ is equal to the a sum of two positive terms:

$$\mathcal{A}_{i,j} = \frac{1}{2\mu\Delta t}\left(B^{n+1}_{i,j} - B^n_{i,j}\right)^2 + \mathcal{A}^{\perp}_{i,j},$$

it is thus necessarily positive. Otherwise, $\mathcal{A}_{i,j}$ writes:

$$\mathcal{A}_{i,j} = \frac{1}{2\mu\Delta t}\left(B^{n+1}_{i,j} - B^n_{i,j}\right)^2 + \mathcal{A}^{\perp}_{i,j} + \frac{c^2}{2\mu^2 (x_2^\alpha)_j \Delta x_1 \Delta x_2}\mathcal{A}^{\wedge}_{i,j}.$$

Assuming that Δt is of the order of the average mesh size Δ and that both the resistivity tensor component $(\rho^\wedge)^n$ and the magnetic field B^{n+1} are smooth enough, we now perform a Taylor expansion around $((x_1)_i;(x_2)_j)$ in the expression of $\mathcal{A}^{\wedge}_{i,j}$. After a few manipulations, it shows that:

$$\frac{c^2}{2\mu^2 (x_2^\alpha)_j \Delta x_1 \Delta x_2}\mathcal{A}^{\wedge}_{i,j} = \mathcal{O}\left(\Delta^2\right).$$

In this case, $\mathcal{A}_{i,j}$ is thus positive provided that Δ is small enough.

Appendix F

Formal derivation of the order of accuracy of the Lagrange-remap GoHy scheme using Maple (third-order hydrodynamics)

We first initialize Maple, state that Δt is of the order of Δx (according to the stability condition of GoHy schemes) and declare an `EOS` procedure for applying the equation of state, *i.e.* for calculating the pressure from the other quantites. Here, we consider the case of a perfect gas law.

```
> restart;
> # dt is of the order of dx
> dt:=alpha*dx;
> # equation of state procedure
> EOS:=proc(rho,u,rhoe)
>   return (gamma-1)*(rhoe-rho*u^2/2);
> end proc;
```

For the sake of simplicity, we define the following `myexpand` procedure that aims at transforming expressions in polynomials of the Δx parameter.

```
> # accuracy parameter for Taylor expansions
> # has to set high enough to avoid undesired simplifications
> myorder:=6;
> myexpand:=proc(expression)
>   return expand(convert(series(expression,dx,myorder),polynom));
> end proc;
```

The formal proof of accuracy consists in applying the whole Lagrange-remap scheme in a single cell which is here assumed to be $[-\Delta x/2; \Delta x/2]$ and whose mid-point is denoted `x`. We now initialize cell averages of conservative variables. The formula given in the `CellAverage` procedure has been obtained by calculating a truncated taylor expansion of cell averages around `x`.

```
> CellAverage:=proc(var,k)
>   c1:=k*dx;
```

```
>   c2:=(k^2/2+1/24)*dx^2;
>   c3:=(k/24+k^3/6)*dx^3;
>   return var + c1*diff(var,x) + c2*diff(var,x$2) + c3*diff(var,x$3);
> end proc;
>
>  rhoAV:=k->CellAverage(rho(x,t),k);
> rhouAV:=k->CellAverage(rho(x,t)*u(x,t),k);
> rhoeAV:=k->CellAverage(rhoe(x,t),k);
```

Cell-centered point wise values of conservative variables are computed according to formula (4.15).

```
> AVtoCC:=proc(var,k)
>   return expand(13/12*var(k)-var(k-1)/24-var(k+1)/24);
> end proc;
>
>  rhoCC:=k->AVtoCC( rhoAV,k);
> rhouCC:=k->AVtoCC(rhouAV,k);
> rhoeCC:=k->AVtoCC(rhoeAV,k);
```

These allow to compute cell-centered point-wise values of all the necessary primitive variables, *i.e.* τ, u, p and $(\rho c)^2$. Since we consider the case of an ideal gas, the fundamental derivative $\mathcal{G}$ is constant and equal to $(\gamma+1)/2$ (see remark 7 page 66).

```
> tauCC:=k->myexpand(1/rhoCC(k));
>   uCC:=k->myexpand(rhouCC(k)*tauCC(k));
>   pCC:=k->myexpand(EOS(rhoCC(k),uCC(k),rhoeCC(k)));
> rc2CC:=k->myexpand(gamma*rhoCC(k)*pCC(k));
>     G:=(gamma+1)/2;
```

Approximate time-derivatives now can be computed on cell interfaces as explained in section 4.1.2. To that end, we first introduce procedures for computing high-order accurate cell interface representations of quantities involved in approximate fluxes according to formula (4.16). These procedures are named `Dab` where `a` and `b` respectively correspond to the m and N_{eff} parameters in (4.16).

```
> D02:=proc(var,k)
>   return expand((var(k)+var(k-1))/2);
> end proc;
>
> D04:=proc(var,k)
>   return expand((9*var(k)+9*var(k-1)-var(k+1)-var(k-2))/16);
> end proc;
>
> D12:=proc(var,k)
>   return expand((var(k)-var(k-1))/dx);
> end proc;
>
> D22:=proc(var,k)
>   return expand((var(k+1)+var(k-2)-var(k)-var(k-1))/(2*dx^2));
> end proc;
```

We recall that (see section 4.1.2):

$$\begin{aligned}
\partial_t u &= -\frac{1}{\rho_0}\partial_X p,\\
\partial_t p &= -\frac{(\rho c)^2}{\rho_0}\partial_X u,\\
\partial_{tt} u &= \frac{(\rho c)^2}{\rho_0^2}\partial_{XX} u - \frac{(\rho c)^2}{\rho_0^3}(\partial_X u)(\partial_X \rho_0) + \frac{1}{\rho_0^2}(\partial_X u)\left(\partial_X (\rho c)^2\right),\\
\partial_{tt} p &= \frac{2(\rho c)^2 \mathcal{G}}{\tau \rho_0^2}\left(\partial_X u\right)^2 + \frac{(\rho c)^2}{\rho_0^2}\partial_{XX} p - \frac{(\rho c)^2}{\rho_0^3}(\partial_X \rho_0)(\partial_X p).
\end{aligned}$$

```
> dt1u:=k->myexpand(-D02(tauCC,k)*D12(pCC,k));
> dt1p:=k->myexpand(-D02(rc2CC,k)*D02(tauCC,k)*D12(uCC,k));
> dt2u:=k->myexpand(D02(tauCC,k)*(D02(rc2CC,k)*(D02(tauCC,k)*D22(uCC,k)+D12(uCC,k)*
                    D12(tauCC,k))+D02(tauCC,k)*D12(uCC,k)*D12(rc2CC,k)));
> dt2p:=k->myexpand(D02(rc2CC,k)*D02(tauCC,k)*(2*G*D12(uCC,k)^2+D02(tauCC,k)*D22(pCC,k)+
                    D12(tauCC,k)*D12(pCC,k)));
```

These time-derivatives can now be used to form Lagrangian fluxes.

```
>  uStar:=k->myexpand(D04(uCC,k)+dt/2*dt1u(k)+dt^2/6*dt2u(k));
>  pStar:=k->myexpand(D04(pCC,k)+dt/2*dt1p(k)+dt^2/6*dt2p(k));
> puStar:=k->myexpand(D04(pCC,k)*D04(uCC,k)+dt/2*(D02(uCC,k)*dt1p(k)+D02(pCC,k)*dt1u(k))
                      +dt^2/6*(D02(uCC,k)*dt2p(k)+D02(pCC,k)*dt2u(k)+2*dt1u(k)*dt1p(k)));
```

Updated values of Lagrangian conservative variables at the end of the Lagrangian step now can be computed.

```
> rhouLS:=k->expand(rhouAV(k)-dt/dx*( pStar(k+1)- pStar(k)));
> rhoeLS:=k->expand(rhoeAV(k)-dt/dx*(puStar(k+1)-puStar(k)));
```

The next step consists in computing remap fluxes. To that end, we first introduce `xN` and `xStar` which respectively represent the initial and updated positions of Eulerian cell interfaces.

```
> xN:=k->(2*k-1)*dx/2;
> xStar:=k->xN(k)+dt*uStar(k);
```

The Lagrange interpolating polynomial P defined in section 4.1.3 is built by the following procedure.

```
> P:=proc(x,var,k)
>   num1:=(x-xStar(k-1))*(x-xStar(k+1))*(x-xStar(k+2));
>   num2:=(x-xStar(k-1))*(x-xStar(k))  *(x-xStar(k+2));
>   num3:=(x-xStar(k-1))*(x-xStar(k))  *(x-xStar(k+1));
>   den1:=(xStar(k)  -xStar(k-1))*(xStar(k)  -xStar(k+1))*(xStar(k)  -xStar(k+2));
>   den2:=(xStar(k+1)-xStar(k-1))*(xStar(k+1)-xStar(k))  *(xStar(k+1)-xStar(k+2));
>   den3:=(xStar(k+2)-xStar(k-1))*(xStar(k+2)-xStar(k))  *(xStar(k+2)-xStar(k+1));
>   H1:=dx*var(k-1);
>   H2:=dx*var(k)+H1;
>   H3:=dx*var(k+1)+H2;
>   return num1/den1*H1 + num2/den2*H2 + num3/den3*H3;
> end proc;
```

It allows to compute high-order accurate remap fluxes according to (4.25). Note that we here assume that u^* is strictly positive on all cell interfaces for choosing the upwinding cell.

```
>  rhoRF:=k->myexpand(P(xStar(k), rhoAV,k-1) - P(xN(k), rhoAV,k-1));
> rhouRF:=k->myexpand(P(xStar(k),rhouLS,k-1) - P(xN(k),rhouLS,k-1));
> rhoeRF:=k->myexpand(P(xStar(k),rhoeLS,k-1) - P(xN(k),rhoeLS,k-1));
```

At this stage, updated Eulerian conservative variables obtained at the end of the remap step can be computed in the considered cell (whose index is 0).

```
>  rhoRS:=myexpand( rhoAV(0)-1/dx*( rhoRF(1)- rhoRF(0)));
> rhouRS:=myexpand(rhouLS(0)-1/dx*(rhouRF(1)-rhouRF(0)));
> rhoeRS:=myexpand(rhoeLS(0)-1/dx*(rhoeRF(1)-rhoeRF(0)));
```

These have to be compared with exact solutions. In order to compute exact solutions, we first determine expressions of the Eulerian conservative variables time-derivatives by applying the Cauchy-Kovalevskaya procedure to the Eulerian hydrodynamics equations.

```
> # rules for the Cauchy-Kovalevskaya procedure
> DTRHO :=diff(rho(x,t),t) =-diff(rho(x,t)*u(x,t),x);
> DTRHOE:=diff(rhoe(x,t),t)=-diff(rhoe(x,t)*u(x,t)+EOS(rho(x,t),u(x,t),rhoe(x,t))*
                            u(x,t),x);
> DTU   :=diff(u(x,t),t)   =(-diff(rho(x,t)*u(x,t)^2+EOS(rho(x,t),u(x,t),rhoe(x,t)),x)+
                            u(x,t)*diff(rho(x,t)*u(x,t),x))/rho(x,t);
> # time derivatives computation
> dt1rho :=expand(subs([DTRHO,DTU,DTRHOE],diff(rho(x,t),t)));
> dt2rho :=expand(subs([DTRHO,DTU,DTRHOE],diff(dt1rho,t)));
> dt3rho :=expand(subs([DTRHO,DTU,DTRHOE],diff(dt2rho,t)));
> dt1rhou:=expand(subs([DTRHO,DTU,DTRHOE],diff(rho(x,t)*u(x,t),t)));
> dt2rhou:=expand(subs([DTRHO,DTU,DTRHOE],diff(dt1rhou,t)));
> dt3rhou:=expand(subs([DTRHO,DTU,DTRHOE],diff(dt2rhou,t)));
> dt1rhoe:=expand(subs([DTRHO,DTU,DTRHOE],diff(rhoe(x,t),t)));
> dt2rhoe:=expand(subs([DTRHO,DTU,DTRHOE],diff(dt1rhoe,t)));
> dt3rhoe:=expand(subs([DTRHO,DTU,DTRHOE],diff(dt2rhoe,t)));
```

Exact solutions are finally computed using the following approximations which have been obtained by applying a Taylor expansion in both space and time around `(x,t)` to the exact cell average of (ρu) and (ρe) at time $t + \Delta t$.

```
>  rhoExact:=expand( rhoAV(0)+dt* dt1rho+dt^2/2* dt2rho+dt^3/6* dt3rho+
             dt*dx^2/24*diff( dt1rho,x$2));
> rhouExact:=expand(rhouAV(0)+dt*dt1rhou+dt^2/2*dt2rhou+dt^3/6*dt3rhou+
             dt*dx^2/24*diff(dt1rhou,x$2));
> rhoeExact:=expand(rhoeAV(0)+dt*dt1rhoe+dt^2/2*dt2rhoe+dt^3/6*dt3rhoe+
             dt*dx^2/24*diff(dt1rhoe,x$2));
```

The exact solutions are finally compared.

```
> subs([dx=0],expand((rhoRS-rhoExact)/dx));
  subs([dx=0],expand((rhoRS-rhoExact)/dx^2));
  subs([dx=0],expand((rhoRS-rhoExact)/dx^3));
```

```
                        0
                        0
                        0
> subs([dx=0],expand((rhouRS-rhouExact)/dx));
  subs([dx=0],expand((rhouRS-rhouExact)/dx^2));
  subs([dx=0],expand((rhouRS-rhouExact)/dx^3));
                        0
                        0
                        0
> subs([dx=0],expand((rhoeRS-rhoeExact)/dx));
  subs([dx=0],expand((rhoeRS-rhoeExact)/dx^2));
  subs([dx=0],expand((rhoeRS-rhoeExact)/dx^3));
                        0
                        0
                        0
```

Since the result of these expressions is zero, the Lagrange-remap scheme studied here is formally proved to be third-order accurate.

Bibliography

[1] D.S. Balsara. Divergence-free adaptative mesh refinement for magnetohydrodynamics. *J. Comp. Phys.*, 174:614–648, 2001.

[2] D.S. Balsara. Second order accurate schemes for magnetohydrodynamics with divergence-free reconstruction. *Astrophys. J. Suppl. Ser.*, 151:149–184, 2004.

[3] D.S. Balsara. Divergence-free reconstruction of magnetic fields and WENO schemes for magnetohydrodynamics. *J. Comp. Phys.*, 228(14):5040–5056, 2009.

[4] D.S. Balsara, C. Altmann, C.D. Munz, and M. Dumbser. A sub-cell based indicator for troubled zones in RKDG schemes and a novel class of hybrid RKDH + HWENO schemes. *J. Comp. Phys.*, 226:586–620, 2007.

[5] D.S. Balsara, T. Rumpf, M. Dumbser, and C.D. Munz. Efficient, high-accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics. *J. Comp. Phys.*, 228:2480–2516, 2009.

[6] D.S. Balsara and C.W. Shu. Monotonicity preserving weighted non-oscillatory schemes with increasingly high order of accuracy. *J. Comp. Phys.*, 160:405–452, 2000.

[7] D.S. Balsara and D.S. Spicer. A staggered mesh algorithm using high-order Godunov fluxes to ensure solenoidal magnetic fields in magnetohydrodynamic simulations. *J. Comp. Phys.*, 149:270–292, 1999.

[8] F. Bezard and B. Després. An entropic solver for ideal lagrangian magnetohydrodynamics. *J. Comp. Phys.*, 154(1):65–89, 1999.

[9] R.S. Bowers and J.R. Wilson. *Numerical modeling in applied physics and astrophysics.* Jones and Barlett Publishers, Boston, 1991.

[10] J.U. Brackbill and D.C. Barnes. The effect of nonzero $\nabla \cdot B$ on the numerical solution of the magnetohydrodynamics equations. *J. Comp. Phys.*, 35:426–430, 1980.

[11] S.I. Braginskii. Transport processes in a plasma. In M.A. Leontovich, editor, *Reviews of plasma physics.* Consultants Bureau - New-York, 1965.

[12] E. Buresi, J. Coutant, R. Dautray, M. Decroisette, B. Duborgel, P. Guillaneux, J. Launspach, P. Nelson, C. Patou, J.-M. Reisse, and J.-P. Watteau. Laser program development at CEL-V: overview of recent experimental results. *Laser and Particle Beams*, 4:531–544, 1986.

[13] S.A. Chin. Forward and non-forward symplectic integrators in solving classical dynamics problems. *Internat. J. of Comp. Math.*, 84:729–747, 2007.

[14] B. Cockburn, S. Hou, and C.W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Math. Comp.*, 54:545–581, 1990.

[15] B. Cockburn, S.Y. Lin, and C.W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems. *J. Comp. Phys.*, 84:90–113, 1989.

[16] B. Cockburn and C.W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: general framework. *Math. Comp.*, 52:411–435, 1989.

[17] B. Cockburn and C.W. Shu. The Runge-Kutta local projection P^1-discontinuous Galerkin method for scalar conservation laws. *Math. Model. and Num. An.*, 25:337–361, 1991.

[18] B. Cockburn and C.W. Shu. The Runge-Kutta discontinuous Galerkin methods for conservation laws V. *J. Comp. Phys.*, 141:199–224, 1998.

[19] B. Cockburn, C.W. Shu, C. Johnson, E. Tadmor, and C.W. Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, volume 1697 of *Lecture Notes in Mathematics*, pages 325–432. Springer Berlin / Heidelberg, 1998.

[20] A.W. Cook. Artificial fluid properties for large-eddy simulation of compressible turbulent mixing. *Phys. of Fluids*, 19, 2007.

[21] A.W. Cook and W.H. Cabot. Hyperviscosity for shock-turbulence interactions. *J. Comp. Phys.*, 203(2):379–385, 2005.

[22] F. Coquel and C. Marmignon. Numerical methods for weakly ionized gases. *Astrophys. Space Sci.*, 260:15–27, 1998.

[23] F. Coquel and C. Marmignon. *A Roe-type linearization for the Euler equations for weakly ionized gases.* Hermes Sci. Publ., Paris, 1999.

[24] R.B. Dahlburg and J.M. Picone. Evolution of the Orszag-Tang vortex system in a compressible medium. I. Initial average subsonic flow. *Phys. Fluids B*, 1:2153–2171, 1989.

[25] R.B. Dahlburg and J.M. Picone. Evolution of the Orszag-Tang vortex system in a compressible medium. II. Supersonic flow. *Phys. Fluids B*, 3:29–44, 1991.

[26] W. Dai and P.R. Woodward. On the divergence-free condition and conservation laws in numerical simulations for supersonic magnetohydrodynamic flows. *Astrophys. J.*, 494:317–335, 1998.

[27] R. Dautray and J.-P. Watteau. *La fusion thermonucléaire inertielle par laser : l'interaction laser-matière*, volume 2. Eyrolles, 1993.

[28] A. Decoster. Fluid equations and transport coefficient of plasmas. In P.-A. Raviart, editor, *Modelling of collisions*, Research in applied mathematics, pages 1–137. Masson, Paris, 1997.

[29] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, and M. Wesenberg. Hyperbolic divergence cleaning for MHD equations. *J. Comp. Phys.*, 175:317–335, 2002.

[30] B. Després. *Lois de conservations eulériennes, lagrangiennes et méthodes numériques.* Springer, 2010.

[31] F. Duboc. Extension de schémas de Godunov à la MHD orthogonale résistive bi-température et termes de champs magnétiques auto-générés. Technical report, CEA/DAM / Université de Bordeaux 1, 2002. Mémoire de DEA.

[32] F. Duboc, C. Enaux, S. Jaouen, H. Jourdren, and M. Wolff. High-order dimensionally split Lagrange-remap schemes for compressible hydrodynamics. *C.R. Acad. Sci. Paris*, 348:105–110, 2010.

[33] M. Dumbser, D.S. Balsara, E.F. Toro, and C.D. Munz. A unified framework for the construction of one-step finite volume and Discontinuous Galerkin schemes on unstructured meshes. *J. Comp. Phys.*, 227:8209–8253, 2008.

[34] M. Dumbser and C.D. Munz. Building blocks for arbitrary high-order discontinuous Galerkin schemes. *J. Sci. Comp.*, 27:215–230, 2006.

[35] C.R. Evans and J.F. Hawley. Simulation of magnetohydrodynamic flows: a constrained transport method. *Astrophys. J.*, 332:659–667, 1988.

[36] E. Forest and R.D. Ruth. Fourth-order symplectic integration. *Physica D*, 43:105–117, 1990.

[37] K.O. Friedrichs and P.D. Lax. Systems of conservation equations with a convex extension. *Proc. Nat. Acad. Sci. USA*, 68:1686–1688, 1971.

[38] G.A. Gerolymos, D. Sénéchal, and I. Vallet. Very high-order WENO schemes. *J. Comp. Phys.*, 228:8481–8524, 2009.

[39] N. Godel, S. Schomann, T. Warburton, and M. Clemens. GPU accelerated Adams-Bashforth multirate Discontinuous Galerkin simulation of high frequency electromagnetic fields. *IEEE Transactions on Magnetics*, 46:2735–2738, 2010.

[40] S.K. Godunov. A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sb.*, 47:271–290, 1959.

[41] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high-order essentially non-oscillatory schemes, III. *J. Comp. Phys.*, 71:231–303, 1987.

[42] A. Hata, K. Mima, A. Sunahara, H. Nagatomo, and A. Nishiguchi. Dynamics of self-generated magnetic fields in stagnation phase and their effects on hot spark formation. *Plasmas and Fusion Research*, 1(20):1–6, 2006.

[43] N.E.L. Haugen. Hydrodynamic and hydromagnetic energy spectra from large eddy simulations. *Phys. of Fluids*, 18, 2006.

[44] F. Hermeline. A finite volume method for the approximation of diffusion operators on distorted meshes. *J. Comp. Phys.*, 160:481–499, 2000.

[45] O. Heuzé, S. Jaouen, and H. Jourdren. Dissipative issue of high-order shock capturing schemes with non-convex equation of state. *J. Comp. Phys.*, 228:833–860, 2009.

[46] C. Hu and C.W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *J. Comp. Phys.*, 150:97–127, 1999.

[47] Intel math kernel library documentation. URL: http://software.intel.com/en-us/articles/intel-math-kernel-library-documentation.

[48] S. Jaouen. Solveur entropique d'ordre élevé pour les équations de l'hydrodynamique à deux températures. Technical report, CEA-DAM / Université de Bordeaux 1, 1997. Mémoire de DEA.

[49] S. Jaouen. A purely Lagrangian method for computing linearly-perturbed flows in spherical geometry. *J. Comp. Phys.*, 225:464–490, 2007.

[50] G.S. Jiang, D. Levy, C.T. Lin, S. Osher, and E. Tadmor. High-resolution non-oscillatory central schemes with non-staggered grids for hyperbolic conservation laws. *J. Comp. Phys.*, 160:241–282, 2000.

[51] G.S. Jiang and C.W. Shu. Efficient implementation of weighted ENO schemes. *J. Comp. Phys.*, 126:202–228, 1996.

[52] G.S. Jiang and C.C. Wu. A high-order WENO finite difference scheme for the equation of ideal magnetohydrodynamics. *J. Comp. Phys.*, 150:561–594, 1999.

[53] R.E. Kidder. Laser-driven compression of hollow shells; power requirements and stability limitations. *Nuclear Fusion*, 16:3–14, 1976.

[54] D. Levy, G. Puppo, and G. Russo. Central WENO schemes for hyperbolic systems of conservation laws. *Math. Model. and Num. An.*, 33:547–571, 1999.

[55] D. Levy, G. Puppo, and G. Russo. A fourth-order central WENO scheme for multidimensional hyperbolic systems of conservation laws. *SIAM J. Sci. Comp.*, 24:480–506, 2002.

[56] F. Li and C.W. Shu. Locally divergence-free discontinuous Galerkin methods for MHD equations. *J. Sci. Comp.*, 22:413–442, 2005.

[57] S. Li. High-order central scheme on overlapping cells for magnetohydrodynamic flows with and without constrained transport method. *J. Comp. Phys.*, 227:7368–7393, 2008.

[58] X.D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comp. Phys.*, 115:200–212, 1994.

[59] E. Livne, L. Dessart, A. Burrows, and C. A. Meakin. A two-dimensional magnetohydrodynamics scheme for general unstructured grids. *Astrophys. J. Suppl. Series*, 170:187–202, 2007.

[60] P.-H. Maire. Unpublished work.

[61] R.I. McLachlan and P. Atela. The accuracy of symplectic integrators. *Nonlinearity*, 5:541–562, 1992.

[62] F.H. McMahon. *The Livermore Fortran kernels: a computer test of the numerical performance range.* Lawrence Berkeley Nat. Lab., Berkeley, CA, 1986.

[63] R. Menikoff and B.J. Plohr. The Riemann problem for fluid flow of real materials. *Rev. Mod. Phys.*, 61(1):75, 1989.

[64] K. Mima, T. Tajima, and J.-N. LeBoeuf. Magnetic field generation by the Rayleigh-Taylor instability. *Phys. Rev. Let.*, 41(25):1715–1719, 1978.

[65] C.D. Munz, P. Omnes, R. Schneider, E. Sonnendrücker, and U. Voss. Divergence correction techniques for Maxwell solvers based on a hyperbolic model. *J. Comp. Phys.*, 161:484–511, 1999.

[66] H. Nessyahu and E. Tadmor. Non-oscillatory central differencing for hyperbolic conservation laws. *J. Comp. Phys.*, 87:408–463, 1990.

[67] W.F. Noh. Errors for calculations of strong shocks using artificial viscosity and an artificial heat flux. *J. Comp. Phys.*, 72:78–120, 1987.

[68] E. Novak and K. Ritter. Simple cubature formulas with high polynomial exactness. *Constructive Approximation*, 15:499–522, 1999.

[69] A. Orszag and C.M. Tang. Small-scale structure of two-dimensional magnetohydrodynamic turbulence. *J. Fluid. Mech.*, 90:129–143, 1979.

[70] P. Picard. Reduction and exact solutions of the ideal magnetohydrodynamics equations. *Mathemat. Phys. e-prints*, 2005. arXiv:math-ph/0509048.

[71] P. Picard. Some spherical solutions of ideal magnetohydrodynamic equations. *J. of Nonlinear Mathemat. Phys.*, 14:583–584, 2007.

[72] K.G. Powell. An approximate Riemann solver for MHD (that works in more than one dimension). ICASE Report 94-24, 1994.

[73] D. Ryu and T.W. Jones. Numerical magnetohydrodynamics in astrophysics: algorithm and tests for one-dimensional flow. *Astrophys. J.*, 442:228–258, 1995.

[74] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[75] C.W. Shu and S.J. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes, II. *J. Comp. Phys.*, 83:32–78, 1989.

[76] G.A. Sod. A survey of several finite difference methods for systems of non linear hyperbolic conservation laws. *J. Comp. Phys.*, 27:1–31, 1978.

[77] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:506, 1968.

[78] H.Z. Tang and K. Xu. A high-order gas-kinetic method for multidimensional ideal magnetohydrodynamics. *J. Comp. Phys.*, 165:69–88, 2000.

[79] A. Taube, M. Dumbser, D.S. Balsara, and C.D. Munz. Arbitrary high-order discontinuous Galerkin schemes for the magnetohydrodynamics equations. *J. Sci. Comp.*, 30:441–464, 2007.

[80] M. Temporal, S. Jaouen, L. Masse, and B. Canaud. Hydrodynamic instabilities in ablative tamped flows. *Phys. of Plasmas*, 13, 2006.

[81] R.P. Tewarson. *Sparse matrices*. Academic Press, New York, 1973.

[82] V.A. Titarev and E.F. Toro. ADER: arbitrary high-order Godunov approach. *J. Sci. Comp.*, 17:609–618, 2002.

[83] V.A. Titarev and E.F. Toro. Solution of the generalised Riemann problem for advection-reaction equations. *Proc. Roy. Soc. Lond.*, 458:271–281, 2002.

[84] V.A. Titarev and E.F. Toro. ADER schemes for scalar hyperbolic conservation laws in three dimensions. *J. Comp. Phys.*, 202:196–215, 2005.

[85] V.A. Titarev and E.F. Toro. ADER schemes for three-dimensional non-linear hyperbolic systems. *J. Comp. Phys.*, 204:715–736, 2005.

[86] E.F. Toro, R.C. Millington, and L.A.M. Nejad. Primitive upwind methods for hyperbolic partial differential equations. In *Sixteenth international conference on numerical methods, lecture notes in physics*, pages 421–426. Springer, 1998.

[87] E.F. Toro, R.C. Millington, and L.A.M. Nejad. Towards very high-order Godunov schemes. In *Godunov methods: theory and applications*, pages 907–940. Kluwer Academic Publishers/Plenum Press, 2001.

[88] G. Tóth. The $\nabla \cdot B = 0$ constraint in shock-capturing magnetohydrodynamics codes. *J. Comp. Phys.*, 161:605–652, 2000.

[89] H.A. van der Vorst. *Iterative Krylov Methods for Large Linear systems*. Cambridge University Press, Cambridge, 2003.

[90] M. Vinokur. A rigorous derivation of the MHD equations based only on Faraday's and Ampère's laws. Presentation at LANL MHD workshop, 1996.

[91] T. Warburton and G.E. Karniadakis. A Discontinuous Galerkin method for the viscous MHD equations. *J. Comp. Phys.*, 152:608–641, 1999.

[92] M. Wolff, S. Jaouen, and L.-M. Imbert-Gérard. Conservative numerical methods for a two-temperature resistive MHD model with self-generated magnetic field terms. *ESAIM Proceedings*, 2011. Accepted.

[93] M. Wolff, S. Jaouen, and H. Jourdren. High-order dimensionally split Lagrange-remap schemes for ideal magnetohydrodynamics. In *Discrete and Continuous Dynamical Systems Series S: proceedings of Numerical Models for Controlled Fusion (NMCF'09)*, Porquerolles, France, 2009.

[94] H.C. Yee, N.D. Sandham, and M.J. Djomehri. Low-dissipative high-order shock-capturing methods using characteristic-based filters. *J. Comp. Phys.*, 150:199–238, 1999.

[95] H. Yoshida. Construction of higher order symplectic integrators. *Phys. Letters A*, 150:262–267, 1990.

[96] S. Zhang and C.W. Shu. A new smoothness indicator for the WENO schemes and its effect on the convergence to steady state solutions. *J. Sci. Comp.*, 31:273–305, 2007.

[97] U. Ziegler. A central-constrained transport scheme for ideal magnetohydrodynamics. *J. Comp. Phys.*, 196:393–416, 2004.