Development and implementation of numerical schemes for radiation hydrodynamics

Hélène Bloch

université PARIS-SACLAY

Development and implementation of numerical schemes for radiation hydrodynamics

Développement et implémentation de schémas numériques pour l'hydrodynamique radiative

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 127, Astronomie et Astrophysique d'Île de France (AAIF)
Spécialité de doctorat: Astronomie et Astrophysique
Unité de recherche : Université Paris-Saclay, UVSQ, Inria, CNRS, CEA, Maison de la Simulation, 91191, Gif-sur-Yvette, France
Référent : Faculté des Sciences d'Orsay

Thèse présentée et soutenue à Paris-Saclay, le 23/09/2021, par

Hélène BLOCH

Composition du Jury

Claire CHAINAIS-HILLAIRET Professeur, Université de Lille	Présidente
Thanh-Hà NGUYEN-BUI Cadre scientifique des EPIC HDR, CEA CESTA (Bordeaux)	Rapporteur & Examinatrice
Rodolphe TURPAULT Professeur, Université de Bordeaux	Rapporteur & Examinateur
Christophe CHALONS Professeur, Université Versailles Saint-Quentin-en-Yvelines	Examinateur

Direction de la thèse

Pascal TREMBLIN Cadre scientifique des EPIC, CEA Saclay	Directeur de thèse
Matthias GONZÁLEZ Maître de conférences, Université de Paris	Co-Directeur de thèse
Edouard AUDIT Cadre scientifique des EPIC, CEA Saclay	Co-Encadrant

Thèse de doctorat
NNT : 2021UPASP074

Remerciements

Je veux remercier toutes les personnes grâce à qui ces trois années se sont aussi bien déroulées.

Tout d'abord, un énorme merci à mes encadrants, Pascal Tremblin, Matthias González et Edouard Audit. J'ai découvert les problématiques liées au transfert radiatif en commençant mon stage de M2, avec des connaissances somme toute assez limitées. Vous avez su répondre avec bienveillance à toutes mes questions qui ont dû vous paraître un peu naïves. Pascal, j'ai tellement appris grâce à toi, que ce soit sur les schémas numériques, en physique, en HPC, mais aussi de façon plus large en méthodologie. Je pense notamment au théorème d'Astérix, que j'essaye d'appliquer au mieux. J'ai énormément apprécié que tu sois autant présent tout au long de ces trois années. Matthias, tu as toujours été disponible, tout au long de ces trois années, même si c'était parfois à distance ces derniers temps. Je ne sais pas s'il était prévu que cette thèse soit aussi fortement orientée maths appliquées plutôt que physique, merci de m'avoir laissé partir dans cette direction. Ces quelques lignes ne sont pas suffisantes pour exprimer toute ma gratitude.

Je voudrais également remercier les membres du jury, merci à Thanh-Hà Nguyen-Bui et Rodolphe Turpault d'avoir rapporté le manuscrit. Merci également à Christophe Chalons et Claire Chainais-Hillairet d'avoir accepté de participer au jury. J'ai découvert pendant mon TER de M1, encadré par Claire, que résoudre un système non linéaire avec la méthode de Newton peut ne pas être aussi compliqué qu'il y paraît.

Je voudrais remercier ensuite tous les membres de la Maison de la Simulation, actuels et anciens, permanents ou non. Les pauses café ont disparu ces derniers temps, mais j'y ai énormément appris, et pas que sur des sujets scientifiques. Je voudrais remercier les doctorants, anciens doctorants, ingénieurs sans qui ces trois années n'auraient pas été aussi agréables : Amal, Charles, Felix, Ksander, Rémi. En particulier un très grand merci à Thomas, qui a pris le temps de répondre à toutes mes questions concernant les volumes finis aussi bien que Kokkos. Je remercie chaleureusement Valérie, sans qui beaucoup de choses ne marcheraient pas aussi bien.

Je remercie également les membres de mon comité de suivi de thèse, Thomas Dufaud et Patrick Hennebelle, pour leurs conseils, en particulier concernant les méthodes multigrilles.

Un énorme merci à Raphaëlle Ichou pour nos discussions autour de mon avenir professionnel, mais également sur un tas d'autres sujets.

Une pensée aussi pour mes amis qui ont pu (ou pas) assister à la soutenance, malheureusement à distance : Laura, Victor, Dominique. Je voudrais aussi remercier les membres du groupe de demi-fond du club d'athlétisme de Palaiseau avec qui j'ai pu me défouler, ce qui m'a permis de bien vivre la rédaction du manuscrit.

Enfin, je voudrais remercier ma famille, en particulier mes parents pour leur soutien sans faille pendant ces trois années (et toutes les précédentes). Damien, je te souhaite de vivre dans les trois années à venir la même super expérience que moi.

Je ne sais pas si j'ai réussi à transcrire ici à quel point j'ai apprécié ces trois années, grâce à toutes les personnes que j'ai pu rencontrées, y compris celles que je n'ai pas nommées ici.

Contents

List of Figures

List of Tables

Introduction (version française)

Table des matières

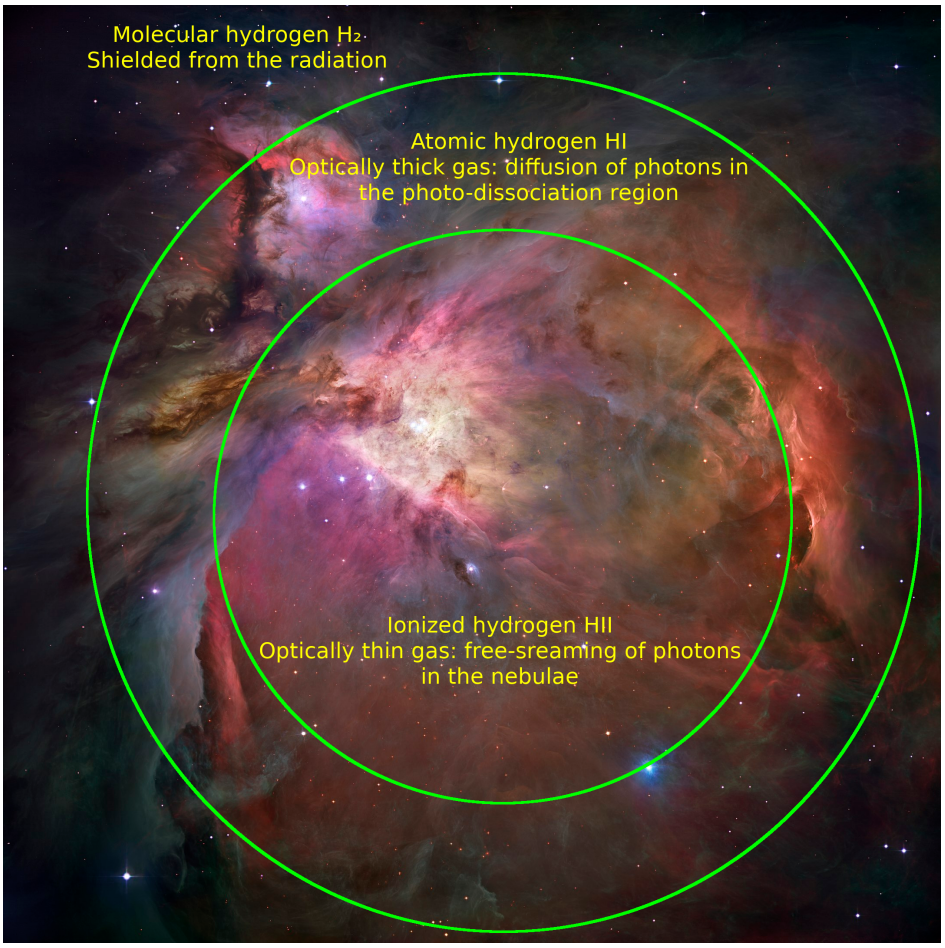


FIGURE 1.1 – Nébuleuse d'Orion et différentes zones optiques. Source : https://esahubble.org/news/heic0601

Le transfert radiatif décrit les interactions entre le rayonnement et le milieu dans lequel il se propage. Il apparaît dans une grande gamme d'applications physiques, de l'astrophysique à la médecine. Dans de nombreuses situations astrophysiques, le rayonnement est un processus important qui interagit avec le gaz environnant, par exemple dans les atmosphères d'(exo)planètes [Thomas and Stamnes, 2002], dans les étoiles massives [Kuiper et al., 2010 ; Mignon-Risse et al., 2020], les régions H II (Spitzer 1978 et fig. 1.1), jusqu'à la réionisation cosmique [Stiavelli, 2009]. La figure 1.1 montre la nébuleuse d'Orion en couleurs visibles. Au centre, une étoile massive émet de la lumière. Les photons les plus énergisants ionisent la région environnante, la région H II. Les photons moins énergisants traversent cette zone sans interaction et dissocient les molécules de dihydrogène dans la région de photodissociation (Photodissociation Region, PDR) extérieure. L'expansion du gaz chaud dans la région H II comprime la PDR ; les échanges d'énergie entre le rayonnement et la matière affectent l'hydrodynamique du système.

L'hydrodynamique radiative est le cas particulier où le rayonnement interagit avec un fluide. La lumière peut être absorbée, ce qui va chauffer le fluide. Du rayonnement peut également être émis par le fluide ou par une autre source. Le rayonnement peut être diffusé, ce qui va changer sa direction de propagation et éventuellement sa fréquence [Mihalas and Mihalas, 1984].

La distance parcourue par les photons avant d'être absorbés ou diffusés est modulée par l'opacité du milieu environnant. D'une part, le milieu peut être optiquement épais,

le rayonnement et l'hydrodynamique sont fortement couplés. D'autre part, le milieu peut être optiquement mince, la lumière et le fluide n'interagissent pas.

La théorie décrivant l'interaction entre le rayonnement et la matière a été formalisée par Chandrasekhar [1960]. L'intensité spécifique obéit à une équation intégro-différentielle, qui ne peut pas être résolue analytiquement dans la plupart des cas.

Comme les expériences et les observations, les simulations numériques peuvent améliorer la compréhension du système physique simulé. Parfois, les expériences ne peuvent pas être réalisées, car elles seraient trop dangereuses (accident), trop longues (climatologie), inatteignables (astrophysique) ou interdites (essais nucléaires). Les simulations numériques peuvent aussi être plus simples : par exemple, il est plus facile de réaliser des simulations en mécanique des fluides plutôt que des expériences en soufflerie.

Une étape supplémentaire est nécessaire, pour passer du modèle continu à une solution approchée. C'est la discrétisation de l'équation, pour obtenir un schéma numérique. Nous présentons brièvement quelques méthodes parmi les nombreuses techniques qui existent pour résoudre l'équation du transfert radiatif, ainsi que leurs limitations.

Le tracé de rayons résout le problème en suivant la propagation de rayons à travers le fluide [Wise and Abel, 2011]. Bien que cette méthode soit très précise, elle est très coûteuse quand elle est couplée à un code d'hydrodynamique. Le nombre de degrés de liberté est proportionnel au nombre de cellules spatiales multiplié par le nombre de sources de rayonnement.

En utilisant la méthode de Monte-Carlo, nous suivons la propagation de paquets de photons et leurs interactions avec le fluide [Roth and Kasen, 2014]. Elle est basée sur un processus stochastique, rendant la méthode précise, mais difficile à coupler avec un code d'hydrodynamique basé sur une grille. Quand le nombre de paquets de photons devient trop petit dans une région, un bruit numérique peut apparaître et polluer les résultats de la simulation, rendant la méthode inutilisable pour étudier les problèmes de stabilité d'interface.

Une autre approximation largement utilisée est la méthode des moments de l'intensité spécifique (en moyennant sur les directions de propagation des photons, Levermore 1984). Dans cette thèse, nous considérons les trois premiers moments de l'intensité spécifique : l'énergie radiative E_r, le flux radiatif $\boldsymbol{F}_\mathbf{r}$ et la pression radiative $\mathbb{P}_\mathrm{r}$. Une relation de fermeture est nécessaire pour exprimer la pression radiative en fonction de l'énergie radiative et du flux radiatif. Celle utilisée est le modèle M_1 [Levermore, 1984 ; Dubroca and Feugeas, 1999] pour ses bonnes propriétés dans les milieux optiquement épais et optiquement minces, ainsi que le compromis entre temps de calcul et précision. Remarquons que l'énergie radiative est positive $E_\mathrm{r} > 0$ et que le flux réduit vérifie $\frac{||\boldsymbol{F}_\mathbf{r}||}{cE_\mathrm{r}} \leqslant 1$, où c est la vitesse de la lumière. L'énergie radiative ne peut pas être transportée plus vite que la vitesse de la lumière.

Le modèle M_1 est un système hyperbolique, il peut être discrétisé avec la méthode des volumes finis. Tous les schémas numériques présentés dans cette thèse sont inspirés des schémas numériques développés pour les équations d'Euler compressibles. Cependant, les spécificités du transfert radiatif et du modèle M_1 complexifient les schémas.

On peut montrer que, dans la limite des longues échelles de temps et des hautes opacités (milieu optiquement épais), le modèle M_1 dégénère vers une équation de diffusion. Un schéma qui *préserve le régime asymptotique* capture ce comportement.

Un solveur explicite pour le transfert radiatif serait restreint par une condition de Courant-Friedrichs-Lewy (CFL), limité par la vitesse de la lumière. On obtiendrait un pas de temps très petit devant celui de l'hydrodynamique, qui est limité par la vitesse du son du fluide. Plusieurs méthodes ont été développées pour contourner ce problème.

Nous avons choisi d'utiliser un solveur implicite [González et al., 2007]. La temporalité du transfert radiatif est préservée, ce qui n'est pas le cas avec l'approximation de la vitesse de la lumière réduite (Reduced Speed of Light Approximation, RSLA, Gnedin and Abel 2001).

Notre but est de développer un schéma numérique avec ces trois propriétés :

— le schéma doit préserver le régime asymptotique pour capturer le comportement en temps long et à haute opacité ;
— la discrétisation en temps doit être implicite, pour éviter que le pas de temps ne soit limité par la vitesse de la lumière ;
— la solution doit être admissible, c'est-à-dire les états admissibles $E_\mathrm{r} > 0$ et $\frac{\|\boldsymbol{F}_\mathrm{r}\|}{cE_\mathrm{r}} \leqslant 1$ doivent être préservés. Ceci est nécessaire pour des raisons physiques et numériques.

De nombreuses heures de calcul peuvent être nécessaires pour étudier les problèmes astrophysiques. L'usage d'outils pour le calcul haute performance (High Performance Computing, HPC) est nécessaire. Ils permettent de réduire le coût de calcul et d'augmenter la taille du problème à résoudre. Plusieurs processeurs traitent différentes parties d'un programme ou différentes données : des problèmes plus grands peuvent être résolus plus vite.

Au cours des dernières décennies, différentes technologies ont émergé, telles que les processeurs multi-coeurs et les GPUs. Ceci soulève le problème de portabilité de performance. Un code peut être optimisé pour une architecture spécifique et l'exécuter sur une autre architecture donnera lieu à de mauvaises performances. Certaines bibliothèques, comme Kokkos, proposent des solutions.

Les solveurs implicites peuvent nécessiter la résolution de grands systèmes linéaires. Cela peut représenter la majorité du temps de calcul. Heureusement, l'algèbre linéaire pour le calcul haute performance a été étudiée au fil des années. Comme les systèmes linéaires que nous devons résoudre sont grands et creux, les méthodes directes sont inutilisables. Des solveurs itératifs préconditionnés ont été développés pour contourner ce problème [Saad, 2003]. La plupart de ces méthodes nécessitent de stocker la matrice. Dans cette thèse, nous utilisons la bibliothèque Trilinos [Heroux et al., 2005] car elle nous permet d'utiliser différentes architectures telles que les processeurs multi-coeurs et les GPUs. Elle fournit, entre autres, des préconditionneurs multigrille algébrique (Algebraic Multigrid, AMG). D'autres méthodes, qui ne nécessitent pas le stockage de la matrice, existent également. C'est notamment le cas des méthodes multigrille géométrique (Geometric Multigrid, GMG) qui ont été développées pour la résolution de systèmes, linéaires ou non linéaires, venant de la discrétisation d'équations différentielles [Briggs et al., 2000].

Cette thèse se concentre sur le développement et l'implémentation de schémas numériques pour l'hydrodynamique radiative, et non sur leurs applications à l'astrophysique.

1.1 Description des travaux

1.1.1 Transfert radiatif

Le premier chapitre est dédié à la présentation du modèle utilisé pour décrire le transfert radiatif. Nous établissons d'abord l'équation du transfert radiatif en effectuant un bilan d'énergie. Elle décrit l'évolution de l'intensité spécifique.

Cette équation peut être coûteuse à résoudre numériquement quand le transfert radiatif est couplé avec l'hydrodynamique. Pour réduire le temps de calcul, nous utilisons un modèle aux moments, où l'intensité spécifique est moyennée selon la direction de propagation et la fréquence des photons. Ceci mène à la résolution d'un système hyperbolique impliquant l'énergie radiative, le flux radiatif et la pression radiative. Ce modèle nécessite de spécifier une relation de fermeture. Celle que nous utilisons est le modèle M_1. La pression radiative est exprimée comme une fonction de l'énergie radiative et du flux radiatif en maximisant l'entropie radiative.

Ce modèle a le bon comportement dans les limites de transport et de diffusion. Nous explorons le système dans cette limite de diffusion, c'est-à-dire dans la limite des grandes opacités et des longues échelles de temps.

Pour des applications physiques, le modèle doit prendre en compte les échanges d'énergie entre le rayonnement et l'hydrodynamique. Nous modélisons ce phénomène en ajoutant des termes non conservatifs. Ces termes dépendent de l'opacité, nous explorons encore une fois le comportement du modèle dans la limite de diffusion, lorsque le couplage à l'hydrodynamique est pris en compte.

1.1.2 Un premier solveur préservant le régime asymptotique

Dans ce chapitre, nous présentons un premier schéma numérique, implicite et qui préserve le régime asymptotique. Il est basé sur un solveur HLL, les flux numériques sont corrigés pour capturer le régime asymptotique. Nous proposons également une modification *équilibre* (*well-balanced*) des termes sources pour atteindre un régime stationnaire constant avec un saut d'opacité.

En utilisant un solveur implicite, il faut résoudre un système non linéaire. Nous utilisons la méthode de Newton-Raphson et les systèmes linéaires sont résolus grâce à la bibliothèque Trilinos.

La résolution du problème d'hydrodynamique radiative est scindée en trois étapes. La première est la mise à jour des quantités hydrodynamiques. La deuxième est la mise à jour des quantités radiatives. La troisième est l'addition des termes sources de couplage.

Nous réalisons des tests standards, dans les limites de transport et de diffusion, incluant des tests purement radiatifs et des tests d'hydrodynamique radiative. Nous étudions également la propagation d'un front d'ionisation dans un cœur dense pré-stellaire massif, et nous montrons que le front d'ionisation est stable, même en présence de mouvements convectifs déstabilisants.

1.1.3 Un schéma tout-régime préservant le régime asymptotique

Inspirés par les schémas numériques pour l'hydrodynamique [Chalons et al., 2016], nous proposons dans ce chapitre un second schéma numérique basé sur un solveur *tout-régime*. Le modèle M_1 est d'abord réécrit sous une forme similaire à celle des équations d'Euler. Les phénomènes de transport et de diffusion sont séparés en deux sous-systèmes. Nous utilisons les flux numériques donnés par Buet and Despres [2008] pour discrétiser le système de diffusion. Nous utilisons un schéma décentré amont pour discrétiser l'étape de transport tout en assurant une mise à jour conservative.

Cependant, ceci conduit à un schéma de stencil 2, avec deux étapes. Nous proposons également un schéma tout-régime avec une seule étape.

Ces deux schémas (deux étapes et une étape) préservent le régime asymptotique si nous utilisons les flux donnés par Buet and Despres [2008]. Ces flux prennent en compte les termes sources. Nous montrons également que ces schémas sont entropiques et préservent les états admissibles, $E_\mathrm{r} > 0$ et $\frac{\|\boldsymbol{F}_\mathrm{r}\|}{cE_\mathrm{r}} \leqslant 1$.

Les tests numériques standards montrent le bon comportement des schémas dans les limites de transport et de diffusion.

1.1.4 Méthode de Jacobi non linéaire et multigrille géométrique

Dans ce chapitre, nous présentons une étape supplémentaire vers le développement d'un schéma implicite qui préserve le régime asymptotique et les états admissibles. Nous discrétisons le modèle M_1 avec un solveur HLL implicite sans terme source. Au lieu de résoudre le système non linéaire avec la méthode de Newton-Raphson, nous utilisons la méthode de Jacobi présentée par Pichard [2016]. Nous montrons que la solution construite avec ce processus itératif préserve les états admissibles.

Cependant, cette méthode peut converger lentement quand la résolution augmente. Pour s'attaquer à ce problème, nous utilisons un algorithme de multigrille géométrique [Briggs et al., 2000]. Le problème est résolu sur une grille plus grossière, où la méthode de Jacobi converge plus vite. Nous introduisons un pseudo-temps [Kifonidis and Müller, 2012] pour préserver les états admissibles.

Les premiers tests numériques dans le régime de transport montrent l'efficacité de la méthode.

1.1.5 Calcul haute performance et algèbre linéaire

Le schéma développé dans le chapitre 2 nécessite de résoudre de grands systèmes linéaires creux et mal conditionnés. Nous utilisons la bibliothèque Trilinos. Nous présentons les différents paquets que nous utilisons et nous montrons comment ils nous permettent d'utiliser différentes architectures, comme les processeurs multi-coeurs et les GPUs. Nous montrons également des résultats de performance et nous discutons l'impact du préconditionneur sur une application physique.

1.2 Publications et communications

Les travaux présentés dans cette thèse ont conduit aux publications et communications orales suivantes :

- H. Bloch, P. Tremblin, M. González, T. Padioleau, and E. Audit, A high-performance and portable asymptotic preserving radiation hydrodynamics code with the M_1 model, A&A, 646:A123, 2021.
- P. Tremblin, H. Bloch, M. González, E. Audit, S. Fromang, T. Padioleau, P. Kestener, and S. Kokh, Radiative Rayleigh-Taylor Instability and the structure of clouds in planetary atmospheres, accepté, A&A.
- Contributions orales :
  - Juin 2021 : Congrès de la SMAI 2021, La Grande Motte, France ;
  - Mai 2021 : Séminaire invité, Centre de Mathématiques Appliquées, École Polytechnique, Palaiseau, France
  - Décembre 2020 : Congrès d'Analyse Numérique pour les Jeunes, en ligne ;

  - Janvier 2020 : Radiation Hydrodynamics : Implementation and Application, Royal Astronomical Society Special Discussion Session, London, Royaume-Uni ;
  - Mai 2019 : International Workshop on Simulating Interactiong Particles, Bordeaux, France.
- Posters :
  - Juin 2019 : Workshop Numerical Methods for Multiscale Models arising in Physics and Biology, Nantes, France ;
  - Mai 2019 : Congrès de la SMAI 2019, Guidel, France.

De plus, les travaux présentés dans les chapitres 3 et 4 font l'objet d'articles en préparation.

1.3 Bibliographie

Gary E. Thomas and Knut Stamnes. *Radiative Transfer in the Atmosphere and Ocean*. 2002. 2

Rolf Kuiper, Hubert Klahr, Henrik Beuther, and Thomas Henning. CIRCUMVENTING THE RADIATION PRESSURE BARRIER IN THE FORMATION OF MASSIVE STARS VIA DISK ACCRETION. *The Astrophysical Journal*, 722(2) : 1556–1576, oct 2010. doi : 10.1088/0004-637x/722/2/1556. URL https://doi.org/10.1088/0004-637x/722/2/1556. 2

R. Mignon-Risse, M. González, B. Commerçon, and J. Rosdahl. A new hybrid radiative transfer method for massive star formation. *A&A*, 635 :A42, 2020. doi : 10.1051/0004-6361/201936605. URL https://doi.org/10.1051/0004-6361/201936605. 2

Lyman Spitzer. *Physical processes in the interstellar medium*. 1978. doi : 10.1002/9783527617722. 2

Massimo Stiavelli. *From First Light to Reionization : The End of the Dark Ages*. 2009. 2

D. Mihalas and B. W. Mihalas. *Foundations of radiation hydrodynamics*. 1984. 2

S. Chandrasekhar. *Radiative Transfer*. Dover Books on Intermediate and Advanced Mathematics. Dover Publications, 1960. ISBN 9780486605906. URL https://books.google.fr/books?id=CK3HDRwCT5YC. 3

John H Wise and Tom Abel. enzo+moray : radiation hydrodynamics adaptive mesh refinement simulations with adaptive ray tracing. *Monthly notices of the Royal Astronomical Society*, 414(4) :3458–3491, 2011. ISSN 0035-8711. 3

Nathaniel Roth and Daniel Kasen. Monte carlo radiation hydrodynamics with implicit methods. 2014. 3

C.D. Levermore. Relating eddington factors to flux limiters. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 31(2) :149 – 160, 1984. ISSN 0022-4073. doi : https://doi.org/10.1016/0022-4073(84)90112-2. URL http://www.sciencedirect.com/science/article/pii/0022407384901122. 3

Bruno Dubroca and Jean-Luc Feugeas. Etude théorique et numérique d'une hiérarchie de modèles aux moments pour le transfert radiatif. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 329(10) :915 – 920, 1999. ISSN 0764-4442. doi : https://doi.org/10.1016/S0764-4442(00)87499-6. URL http://www.sciencedirect.com/science/article/pii/S0764444200874996. 3

M. González, E. Audit, and P. Huynh. Heracles : a three-dimensional radiation hydrodynamics code. *A&A*, 464(2) :429–435, 2007. doi : 10.1051/0004-6361:20065486. URL https://doi.org/10.1051/0004-6361:20065486. 4

Nickolay Y Gnedin and Tom Abel. Multi-dimensional cosmological radiative transfer with a variable eddington tensor formalism. *New astronomy*, 6(7) :437–455, 2001. ISSN 1384-1076. 4

Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003. ISBN 0898715342. 4

Michael A. Heroux, Roscoe A. Bartlett, Vicki E. Howle, Robert J. Hoekstra, Jonathan J. Hu, Tamara G. Kolda, Richard B. Lehoucq, Kevin R. Long, Roger P. Pawlowski, Eric T. Phipps, Andrew G. Salinger, Heidi K. Thornquist, Ray S. Tuminaro, James M. Willenbring, Alan Williams, and Kendall S. Stanley. An overview of the trilinos project. *ACM Trans. Math. Softw.*, 31(3) :397–423, 2005. ISSN 0098-3500. doi : http://doi.acm.org/10.1145/1089014.1089021. 4

William Briggs, Van Henson, and Steve McCormick. *A Multigrid Tutorial, 2nd Edition*. 01 2000. ISBN 978-0-89871-462-3. 4, 6

Christophe Chalons, Pierre Kestener, Samuel Kokh, and Maxime Stauffert. A large time-step and well-balanced Lagrange-Projection type scheme for the shallow-water equations. July 2016. URL https://hal.archives-ouvertes.fr/hal-01297043. working paper or preprint. 5

Christophe Buet and Bruno Despres. A gas dynamics scheme for a two moments model of radiative transfer. 11 2008. 5, 6

Teddy Pichard. *Mathematical modelling for dose depositon in photontherapy*. PhD thesis, 2016. URL http://www.theses.fr/2016BORD0177. Thèse de doctorat dirigée par Dubroca, Bruno et Frank, Martin Mathematiques appliquees et calcul scientifique Bordeaux 2016. 6

K. Kifonidis and E. Müller. On multigrid solution of the implicit equations of hydrodynamics - experiments for the compressible euler equations in general coordinates. *A&A*, 544 :A47, 2012. doi : 10.1051/0004-6361/201116979. URL https://doi.org/10.1051/0004-6361/201116979. 6

Introduction (english version)

Contents

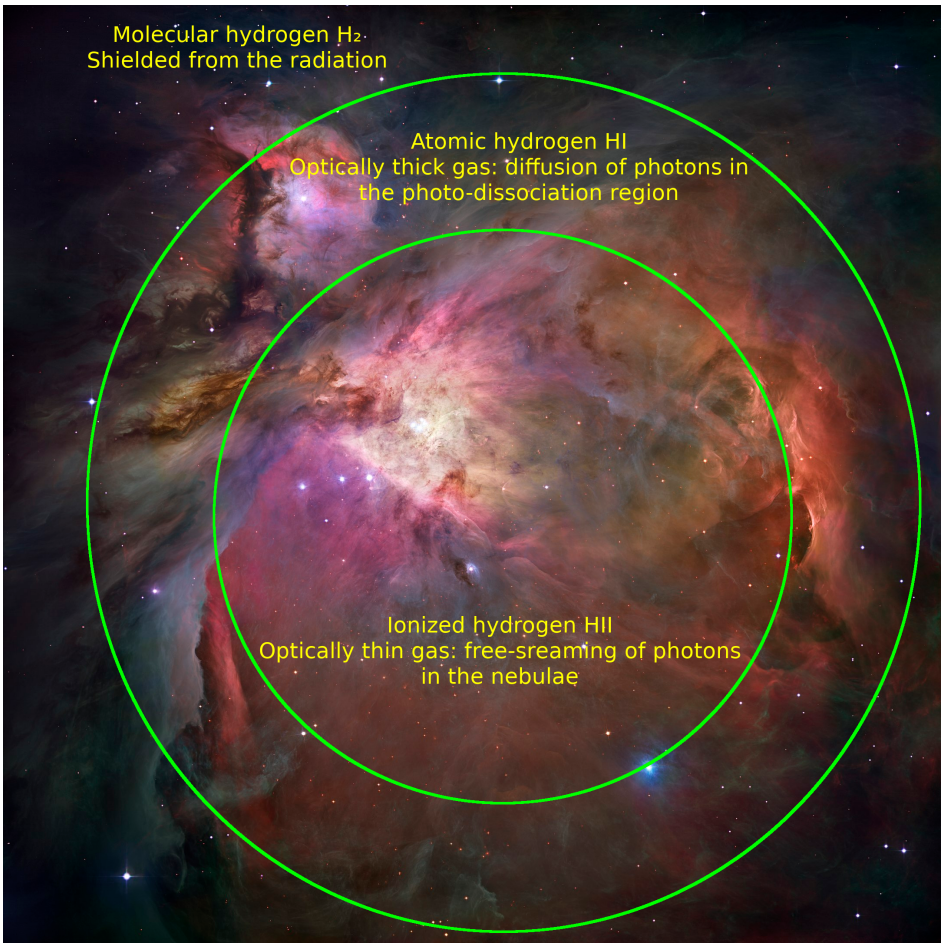


Figure 1.2 – Orion nebula and different optical zones. Source: https://esahubble.org/news/heic0601

Radiative transfer describes the interactions between the radiation and the surrounding medium. It appears in a wild range of physical situations, from astrophysics, up to medicine. In many astrophysical situations, radiation is an important process that interacts with the surrounding gas, e.g., in (exo) planet's atmospheres (e.g., Thomas and Stamnes 2002), massive stars (e.g., Kuiper et al. 2010; Mignon-Risse et al. 2020), H II regions (e.g., Spitzer 1978 and fig. 1.2), up to the cosmic reionization (e.g., Stiavelli 2009). Figure 1.2 shows the Orion nebula in visible colors. In the middle, a massive star emits light. Photons with the highest energy ionize the surrounding zone, the H II region. Less energetic photons travel across this area without interaction and dissociate dihydrogen molecules in the outer photodissociation region (PDR). The expanding hot gas in the H II region compresses the PDR; energy exchanges between light and matter affect the hydrodynamics of the system.

Radiation hydrodynamics is the special case where the radiation interacts with a fluid. Light can be absorbed, thus heating the fluid. The radiation could also be emitted by the fluid or by another source. Radiation could be scattered, this will change its direction and perhaps its frequency [Mihalas and Mihalas, 1984].

The distance traveled by the photons before being absorbed or scattered is shaped by the opacity of the surrounding medium. On one hand, the medium can be optically thick, the radiation and the hydrodynamics are strongly coupled. On the other hand, the medium can be optically thin, the light and the fluid do not interact with each other.

The theory describing the interaction between radiation and matter was formalized by Chandrasekhar [1960]. The specific intensity obeys an integro-differential equation, which cannot be solved analytically in most cases.

As experiments and observations, numerical simulations can improve the understanding of the simulated physical system. Sometimes, experiments cannot be performed, should they be too dangerous (crash), too long (climatology), unreachable (astrophysics), or forbidden (nuclear test). Numerical simulations can also be simpler: for example, it is easier to perform fluid simulations instead of experiments in a wind tunnel.

Another step is required, to go from the continuous model to an approximate solution. It is the discretization of the equation, to obtain a numerical scheme. Let us now briefly present some methods and their limitations, among the numerous techniques that exist to solve the radiative transfer equation.

Ray tracing solves the problem by following the propagation of beams through the fluid (e.g., Wise and Abel 2011). Although this method is very precise, it is highly costly when coupled to hydrodynamics codes. The number of degrees of freedom scales with the number of spatial cells multiplied by the number of radiation sources.

Using a Monte-Carlo method, we follow the propagation of "photon energy packets" and their interactions with the fluid (e.g., Roth and Kasen 2014). It is based on a stochastic process, making this method accurate, but difficult to couple with a grid based hydrodynamics code. When the number of photon packets becomes too small in a region, a numerical noise can arise and pollute the simulation results, making the method unusable to study interface stability problems.

Another widely used approximation is moment models [Levermore, 1984] of the specific intensity (by averaging over the direction of propagation of photons). In this thesis, we consider the three first moments of the specific intensity: the radiative energy E_r, the radiative flux $\boldsymbol{F}_\mathbf{r}$, and the radiative pressure $\mathbb{P}_\mathrm{r}$. A closure relation is needed to express the radiative pressure as a function of the radiative energy and the radiative flux. The one we use is the M_1 model [Levermore, 1984; Dubroca and Feugeas, 1999] for its good properties in both optically thick and thin media and the compromise between computational cost and precision. Let us notice that the radiative energy is positive $E_\mathrm{r} > 0$ and the reduced flux verifies $\frac{||\boldsymbol{F}_\mathbf{r}||}{cE_\mathrm{r}} \leqslant 1$, where c is the speed of light. It ensures that the radiative energy cannot be transported faster than the speed of light.

The M_1 model is a hyperbolic system, it can be discretized with a finite volume method. All numerical schemes presented in this thesis are inspired by numerical schemes developed for the compressible Euler equations. However, the specific features of the radiative transfer and the M_1 model increase the complexity of the schemes.

One can show that, in the limit of long timescale and high opacity (optically thick medium), the M_1 model degenerates towards a diffusion equation. An *asymptotic preserving* scheme captures this behavior.

An explicit solver for the radiative transfer would be restricted by a Courant-Friedrichs-Lewy (CFL) condition, limited by the speed of light. This will result in a very low time step compared to the hydrodynamics one, which is limited by the speed of sound of the fluid. Several methods have been developed to get around this problem. The one we have chosen is a time-implicit solver (e.g., González et al. 2007). The temporality of the radiative transfer will be preserved, which is not the case with the reduced speed of light approximation (RSLA, e.g, Gnedin and Abel 2001).

We aim at building a numerical scheme with these three properties:

— the scheme should be asymptotic preserving to capture the behavior in long

timescale and high opacity;
— the discretization in time should be implicit, to avoid a time step restricted by the speed of light;
— the solution should be admissible, i.e., the admissible states $E_\mathrm{r} > 0$ and $\frac{||\boldsymbol{F}_\mathrm{r}||}{cE_\mathrm{r}} \leqslant 1$ should be preserved. This is needed for both physical and numerical reasons.

To study astrophysical problems, many computational hours can be needed. This requires the use of High Performance Computing (HPC) tools. It allows reducing the computational cost and increasing the size of the problems to be solved. Multiple processing units handle different parts of the same program or data simultaneously: bigger problems can be solved faster.

In the last decades, different architectures have emerged, such as multi-cores, many-cores, and GP-GPUs. This raises the problem of performance portability. A code can be optimized for a specific architecture and running it on a different architecture will result in bad performance. Some libraries, such as Kokkos, address this issue.

Implicit solvers can require solving large linear systems. It can represent most of the computational time. Fortunately, linear algebra for HPC has been investigated over the years. Because the linear system we will have to solve is large and sparse, direct methods are out of reach. Iterative solvers with preconditioners have been developed to tackle this issue (e.g., Saad 2003). Most of these methods require storing the matrix. In this work, we use the library Trilinos [Heroux et al., 2005] because it allows us to target different architectures, such as multi-core, many-core, and GP-GPUs. It provides, among others, Algebraic Multigrid (AMG) preconditioners. Other methods that do not require to store the matrix also exist. For example, Geometric Multigrid (GMG) methods have been developed to solve linear and nonlinear systems arising from the discretization of differential equations (e.g., Briggs et al. 2000).

This thesis focuses on the development and the implementation of numerical schemes for radiation hydrodynamics, not their application to astrophysics.

1.4 Description of the work

1.4.1 Radiative transfer

The first chapter is dedicated to the presentation of the model used to describe the radiative transfer. We first derive the radiative transfer equation by computing the energy balance. It describes the evolution of the specific intensity.

This equation can be costly to solve numerically when the radiative transfer is coupled with the hydrodynamics. To reduce the computational cost, we use a moment method, in which the specific intensity is averaged over the direction of propagation and the frequency of the photons. This leads to solving a hyperbolic system involving radiative energy, radiative flux, and radiative pressure. This model requires specifying a closure relation. The one we use is the M_1 model. We express the radiative pressure as a function of the radiative energy and radiative flux by maximizing the radiative entropy.

This model has the correct behavior in both free-streaming and diffusive limits. We look at this system in the diffusive limit, i.e., large opacity and long timescale.

For physical applications, the model has to take into account the exchange of energy between the radiation and the hydrodynamics. We model this phenomenon by adding nonconservative terms. These terms depend on the opacity, we investigate again the

behavior of the model in the diffusive limit, when the coupling to the hydrodynamics is taken into account.

1.4.2 A first asymptotic preserving solver

In this chapter, we present a first numerical scheme, time-implicit, and asymptotic preserving. It is based on an HLL solver, numerical fluxes are corrected to capture the asymptotic behavior. We also suggest a *well-balanced* modification of the source terms to reach a constant steady state with a jump of opacity.

Using a time-implicit solver, a nonlinear system has to be solved. It is done with the Newton-Raphson method and linear systems are solved using the library Trilinos.

The resolution of the radiation hydrodynamics problem is split into three steps. The first one is the update of the hydrodynamics quantities. The second one is the update of the radiative quantities. The third one is the addition of coupling source terms.

We perform standard tests in both free-streaming and diffusive limits, including purely radiative tests and radiation hydrodynamics ones. We also study the propagation of an ionization front in a massive pre-stellar dense core, and we show that the ionization front is strongly stable against perturbations even with destabilizing convective motions.

1.4.3 An all-regime-like asymptotic preserving scheme

Inspired by numerical schemes for hydrodynamics [Chalons et al., 2016], we propose in this chapter a second numerical scheme based on an *all-regime* solver. The M_1 model is first written in a form similar to the Euler equations. Diffusion and transport phenomena are then split into two subsystems. We use numerical fluxes given by Buet and Despres [2008] to discretize the diffusion system. We use an upwind scheme to discretize the transport step while ensuring a conservative update.

However, this leads to a scheme of stencil 2, with two steps. We also propose an all-regime scheme with only one step.

Both schemes (two steps and one step) are asymptotic preserving if we use numerical fluxes given by Buet and Despres [2008]. These fluxes take into account source terms. We also show that these schemes are entropic and preserve the admissible states, $E_r > 0$ and $\frac{||\boldsymbol{F}_r||}{cE_r} \leqslant 1$.

Standard numerical tests show the good behavior of the schemes in both free-streaming and diffusive limits.

1.4.4 Nonlinear Jacobi method and geometric multigrid

In this chapter, we present an additional step towards the development of an asymptotic preserving and time-implicit solver that preserves the admissible states. We discretize the M_1 model with a time-implicit HLL solver without source terms. Instead of solving the nonlinear system with the Newton-Raphson method, we use the Jacobi method presented in Pichard [2016]. We show that the solution built with this iterative process preserves the admissible states.

However, this method can be slow to converge when the resolution increases. To tackle this issue, we use a geometric multigrid algorithm [Briggs et al., 2000]. The

problem is solved on a coarser grid, where the Jacobi method converges faster. Introducing a pseudo-time [Kifonidis and Müller, 2012] allows us to preserve the admissible states.

First numerical tests in the free-streaming regime show the efficiency of the method.

1.4.5 High performance computing and linear algebra

The scheme developed in chapter 2 requires solving large sparse ill-conditioner linear systems. We use the library Trilinos to do it. We present the different packages we used and we show how they allow us to target different architectures, such as multi-cores, many-cores, and GP-GPUs. We also show some performance results and we discuss the impact of the preconditioner on a physical application.

1.5 Publications and communications

The work presented in this thesis led to the following publications and oral communications:

- H. Bloch, P. Tremblin, M. González, T. Padioleau, and E. Audit, A high-performance and portable asymptotic preserving radiation hydrodynamics code with the M_1 model, A&A, 646:A123, 2021.
- P. Tremblin, H. Bloch, M. González, E. Audit, S. Fromang, T. Padioleau, P. Kestener, and S. Kokh, Radiative Rayleigh-Taylor Instability and the structure of clouds in planetary atmospheres, accepted, A&A.
- Oral contributions:
  - June 2021: Congrès de la SMAI 2021, La Grande Motte, France;
  - May 2021: Séminaire invité, Centre de Mathématiques Appliquées, École Polytechnique, Palaiseau, France
  - December 2020: Congrès d'Analyse Numérique pour les Jeunes, online;
  - January 2020: Radiation Hydrodynamics: Implementation and Application, Royal Astronomical Society Special Discussion Session, London, UK;
  - May 2019: International Workshop on Simulating Interactiong Particles, Bordeaux, France.
- Posters:
  - June 2019: Workshop Numerical Methods for Multiscale Models arising in Physics and Biology, Nantes, France;
  - May 2019: Congrès de la SMAI 2019, Guidel, France.

Furthermore, the work presented in chapters 3 and 4 is the subject of articles in preparation.

1.6 Bibliography

Gary E. Thomas and Knut Stamnes. *Radiative Transfer in the Atmosphere and Ocean*. 2002. 10

Rolf Kuiper, Hubert Klahr, Henrik Beuther, and Thomas Henning. CIRCUMVENTING THE RADIATION PRESSURE BARRIER IN THE FORMATION OF MASSIVE STARS VIA DISK ACCRETION. *The Astrophysical Journal*, 722(2): 1556–1576, oct 2010. doi: 10.1088/0004-637x/722/2/1556. URL https://doi.org/10.1088/0004-637x/722/2/1556. 10

R. Mignon-Risse, M. González, B. Commerçon, and J. Rosdahl. A new hybrid radiative transfer method for massive star formation. *A&A*, 635:A42, 2020. doi: 10.1051/0004-6361/201936605. URL https://doi.org/10.1051/0004-6361/201936605. 10

Lyman Spitzer. *Physical processes in the interstellar medium*. 1978. doi: 10.1002/9783527617722. 10

Massimo Stiavelli. *From First Light to Reionization: The End of the Dark Ages*. 2009. 10

D. Mihalas and B. W. Mihalas. *Foundations of radiation hydrodynamics*. 1984. 10

S. Chandrasekhar. *Radiative Transfer*. Dover Books on Intermediate and Advanced Mathematics. Dover Publications, 1960. ISBN 9780486605906. URL https://books.google.fr/books?id=CK3HDRwCT5YC. 11

John H Wise and Tom Abel. enzo+moray: radiation hydrodynamics adaptive mesh refinement simulations with adaptive ray tracing. *Monthly notices of the Royal Astronomical Society*, 414(4):3458–3491, 2011. ISSN 0035-8711. 11

Nathaniel Roth and Daniel Kasen. Monte carlo radiation hydrodynamics with implicit methods. 2014. 11

C.D. Levermore. Relating eddington factors to flux limiters. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 31(2):149 – 160, 1984. ISSN 0022-4073. doi: https://doi.org/10.1016/0022-4073(84)90112-2. URL http://www.sciencedirect.com/science/article/pii/0022407384901122. 11

Bruno Dubroca and Jean-Luc Feugeas. Etude théorique et numérique d'une hiérarchie de modèles aux moments pour le transfert radiatif. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 329(10):915 – 920, 1999. ISSN 0764-4442. doi: https://doi.org/10.1016/S0764-4442(00)87499-6. URL http://www.sciencedirect.com/science/article/pii/S0764444200874996. 11

M. González, E. Audit, and P. Huynh. Heracles: a three-dimensional radiation hydrodynamics code. *A&A*, 464(2):429–435, 2007. doi: 10.1051/0004-6361:20065486. URL https://doi.org/10.1051/0004-6361:20065486. 11

Nickolay Y Gnedin and Tom Abel. Multi-dimensional cosmological radiative transfer with a variable eddington tensor formalism. *New astronomy*, 6(7):437–455, 2001. ISSN 1384-1076. 11

Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003. ISBN 0898715342. 12

Michael A. Heroux, Roscoe A. Bartlett, Vicki E. Howle, Robert J. Hoekstra, Jonathan J. Hu, Tamara G. Kolda, Richard B. Lehoucq, Kevin R. Long, Roger P. Pawlowski, Eric T. Phipps, Andrew G. Salinger, Heidi K. Thornquist, Ray S. Tuminaro, James M. Willenbring, Alan Williams, and Kendall S. Stanley. An overview of the trilinos project. *ACM Trans. Math. Softw.*, 31(3):397–423, 2005. ISSN 0098-3500. doi: http://doi.acm.org/10.1145/1089014.1089021. 12

William Briggs, Van Henson, and Steve McCormick. *A Multigrid Tutorial, 2nd Edition*. 01 2000. ISBN 978-0-89871-462-3. 12, 13

Christophe Chalons, Pierre Kestener, Samuel Kokh, and Maxime Stauffert. A large time-step and well-balanced Lagrange-Projection type scheme for the shallow-water equations. July 2016. URL https://hal.archives-ouvertes.fr/hal-01297043. working paper or preprint. 13

Christophe Buet and Bruno Despres. A gas dynamics scheme for a two moments model of radiative transfer. 11 2008. 13

Teddy Pichard. *Mathematical modelling for dose depositon in photontherapy*. PhD thesis, 2016. URL http://www.theses.fr/2016BORD0177. Thèse de doctorat dirigée par Dubroca, Bruno et Frank, Martin Mathematiques appliquees et calcul scientifique Bordeaux 2016. 13

K. Kifonidis and E. Müller. On multigrid solution of the implicit equations of hydrodynamics - experiments for the compressible euler equations in general coordinates. *A&A*, 544:A47, 2012. doi: 10.1051/0004-6361/201116979. URL https://doi.org/10.1051/0004-6361/201116979. 14

Chapter 1

Radiative transfer

Contents

Sections 1.4, 1.5 and 1.C are the adaptation of an article published in Astronomy&Astrophysics, see Bloch et al. 2021.

This chapter is dedicated to the presentation of radiative transfer equation (section 1.1). The quantity of interest is the specific intensity, which describes the rate of radiative transfer of energy, at a point $\boldsymbol{x}$ and time t. It is used in the study of processes involving propagation of radiation and its interaction with the surrounding medium. It relies on classical electromagnetic radiation, whereas it is conceptually different from the description of Maxwell electromagnetic fields.

Due to the high number of degrees of freedom (i.e., the time, the position, the direction of propagation, and the frequency of photons), only a few problems can be solved analytically [Chandrasekhar, 1960] and, if coupled with hydrodynamics, direct simulations are out of reach for modern computers. Different models have been developed to reduce the computational cost. We will focus on the moment models [Levermore, 1984] in section 1.2: the specific intensity is averaged over the direction of propagation of photons. It presents several advantages: the computational cost is lower than other methods such as Monte-Carlo method and, mostly, it is easy to couple it with a grid-based hydrodynamics code. Furthermore, we derive a specific closure relation, the M_1 model, by maximizing the radiative entropy (section 1.3). This closure relation allows us to treat accurately both free-streaming and diffusive regimes. In particular, we investigate the diffusive limit in section 1.4. Finally, we enhance this model for a moving fluid in section 1.5 to derive the radiation hydrodynamics equations.

1.1 Radiative transfer equation

In the vacuum, photons propagate in a straight line at velocity $c\boldsymbol{\Omega}$, where c is the speed of light and $\boldsymbol{\Omega}$ is the direction of propagation of photons. At a time t, a photon is determined by its position $\boldsymbol{x}$, its direction of propagation $\boldsymbol{\Omega}$, and its frequency ν.

To derive the radiative transfer equation, we follow Chandrasekhar [1960]. Let us define the specific intensity I such that the flux of radiative energy Φ crossing an oriented elementary surface $\mathrm{d}\boldsymbol{S}$ around the point $\boldsymbol{x}$ in the elementary solid angle $\mathrm{d}\Omega$ around the direction $\boldsymbol{\Omega}$ with frequency in $[\nu, \nu + \mathrm{d}\nu]$, at time t, is

$$\Phi\left(\boldsymbol{x}, t, \boldsymbol{\Omega}, \nu\right) = I\left(\boldsymbol{x}, t, \boldsymbol{\Omega}, \nu\right) \boldsymbol{\Omega} \cdot \mathrm{d}\boldsymbol{S} \mathrm{d}\Omega \mathrm{d}\nu. \tag{1.1}$$

Figure 1.1 summarizes the notations.

Let us now derive the radiative transfer equation by computing the energy balance in an elementary volume $\mathrm{d}V$ with curvilinear length $\mathrm{d}l$ (see fig. 1.2). We recall that the radiation can be absorbed by the surrounding gas. Photons could also be emitted

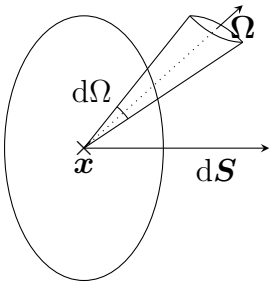


Figure 1.1 – Geometry used to define the specific intensity.

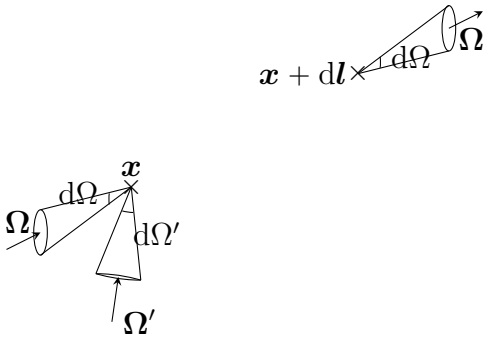


Figure 1.2 – Elementary volume in which the energy balance is computed.

by the gas or by another source. Photons could be scattered, which will change their direction of propagation and perhaps their frequency. Let us now focus on each of these phenomena.

We first investigate absorption. We consider the flux of radiative energy propagating in an absorbing medium. A part $\mathrm{d}\Phi_\mathrm{a}$ of the radiative energy is absorbed:

$$\mathrm{d}\Phi_\mathrm{a}\left(\boldsymbol{x} + \mathrm{d}\boldsymbol{l}, t, \boldsymbol{\Omega}, \nu\right) = -\sigma_\nu^\mathrm{a}\Phi\left(\boldsymbol{x}, t, \boldsymbol{\Omega}, \nu\right)\mathrm{d}l, \tag{1.2}$$

with σ_ν^a the absorption coefficient. Its inverse, $\frac{1}{\sigma_\nu^\mathrm{a}}$, is the absorption mean free path.

We now look at thermal emission. A part

$$\mathrm{d}\Phi_\mathrm{e}\left(\boldsymbol{x} + \mathrm{d}\boldsymbol{l}, t, \boldsymbol{\Omega}, \nu\right) = \eta_\nu \boldsymbol{\Omega} \cdot \mathrm{d}\boldsymbol{S}\mathrm{d}\Omega\mathrm{d}\nu\mathrm{d}l, \tag{1.3}$$

with η_ν the emission coefficient, is emitted by the surrounding matter. From Chandrasekhar [1960], at local thermodynamic equilibrium (LTE),

$$\eta_\nu = \sigma_\nu^\mathrm{a} B\left(\nu, T_\mathrm{g}\right) = \sigma_\nu^\mathrm{a}\frac{2h\nu^3}{c}\frac{1}{\exp\left(\frac{h\nu}{k_\mathrm{b}T_\mathrm{g}}\right) - 1}, \tag{1.4}$$

where B is the black body specific intensity, h is the Planck constant, k_b is the Boltzmann constant and T_g is the temperature of the matter. From now on, we always assume LTE. At global thermodynamic equilibrium, Φ does not depend on $\boldsymbol{x}$, therefore

$$0 = \frac{\partial\Phi}{\partial l} = \mathrm{d}\Phi_\mathrm{e} + \mathrm{d}\Phi_\mathrm{a}, \tag{1.5}$$

which leads to

$$I\left(\boldsymbol{x}, t, \boldsymbol{\Omega}, \nu\right) = B\left(\nu, T_\mathrm{g}\right). \tag{1.6}$$

We will recover eq. 1.6 in section 1.B.

Finally, let us consider scattering. On one hand, part of the energy propagating initially in the direction $\boldsymbol{\Omega}$ is scattered into a different direction $\boldsymbol{\Omega}'$. This will decrease the flux of radiative energy in the direction $\boldsymbol{\Omega}$. By writing σ_ν^s the scattering coefficient, the energy lost by scattering is

$$-\sigma_\nu^\mathrm{s}\Phi\left(\boldsymbol{x}, t, \boldsymbol{\Omega}, \nu\right)\mathrm{d}l. \tag{1.7}$$

The scattering mean free path is $\frac{1}{\sigma_\nu^\mathrm{s}}$.

On the other hand, energy propagating initially in a direction $\boldsymbol{\Omega}'$, different from $\boldsymbol{\Omega}$, can be scattered and then is propagated in the direction $\boldsymbol{\Omega}$. This will increase

the flux of radiative energy in the direction $\boldsymbol{\Omega}$. Let us introduce the phase function $p_\nu(\boldsymbol{x}, \boldsymbol{\Omega}, \boldsymbol{\Omega}')$ which represents the part of the flux of radiative energy that is scattered from the direction $\boldsymbol{\Omega}'$ into the direction $\boldsymbol{\Omega}$ at the point $\boldsymbol{x}$. Then, the gain is

$$\sigma_\nu^{\mathrm{s}} \int_{S^2} p_\nu(\boldsymbol{x}, \boldsymbol{\Omega}, \boldsymbol{\Omega}')\, \Phi(\boldsymbol{x}, t, \boldsymbol{\Omega}', \nu)\, \mathrm{d}\Omega' \mathrm{d}l. \tag{1.8}$$

If the medium is homogeneous, which we assume from now on, then p_ν does not depend on the position $\boldsymbol{x}$. Furthermore, we assume that p_ν depends only on the angle between $\boldsymbol{\Omega}$ and $\boldsymbol{\Omega}'$. Therefore, we write $p_\nu(\boldsymbol{\Omega} \cdot \boldsymbol{\Omega}')$. With our choice of normalization, we have

$$\int_{S^2} p_\nu(\boldsymbol{\Omega} \cdot \boldsymbol{\Omega}')\, \mathrm{d}\Omega = 1, \tag{1.9}$$

but other choices are possible.

The difference of energy in the elementary volume between the moment t and the moment $t + \mathrm{d}t$ is then

$$\begin{aligned}
&\left(I(\boldsymbol{x} + \mathrm{d}\boldsymbol{l}, t + \mathrm{d}t, \boldsymbol{\Omega}, \nu) - I(\boldsymbol{x}, t, \boldsymbol{\Omega}, \nu)\right) \boldsymbol{\Omega} \cdot \mathrm{d}\boldsymbol{S} \mathrm{d}\Omega \mathrm{d}\nu \\
&\qquad = -\left(\sigma_\nu^{\mathrm{a}} + \sigma_\nu^{\mathrm{s}}\right) I(\boldsymbol{x}, t, \boldsymbol{\Omega}, \nu)\, \boldsymbol{\Omega} \cdot \mathrm{d}\boldsymbol{S} \mathrm{d}\Omega \mathrm{d}\nu \mathrm{d}l \\
&\qquad\quad + \sigma_\nu^{\mathrm{a}} B(\nu, T_{\mathrm{g}})\, \boldsymbol{\Omega} \cdot \mathrm{d}\boldsymbol{S} \mathrm{d}\Omega \mathrm{d}\nu \mathrm{d}l \\
&\quad + \sigma_\nu^{\mathrm{s}} \int_{S^2} p_\nu(\boldsymbol{\Omega} \cdot \boldsymbol{\Omega}')\, I(\boldsymbol{x}, t, \boldsymbol{\Omega}', \nu)\, \mathrm{d}\Omega' \boldsymbol{\Omega} \cdot \mathrm{d}\boldsymbol{S} \mathrm{d}\Omega \mathrm{d}\nu \mathrm{d}l.
\end{aligned} \tag{1.10}$$

Let us notice that $\mathrm{d}t = \frac{\mathrm{d}l}{c}$. Then, using a Taylor expansion at first order with $\mathrm{d}l$ that goes to 0, one has

$$I(\boldsymbol{x} + \mathrm{d}\boldsymbol{l}, t + \mathrm{d}t, \boldsymbol{\Omega}, \nu) = I(\boldsymbol{x}, t, \boldsymbol{\Omega}, \nu) + \mathrm{d}l \left(\frac{\partial I}{\partial l} + \frac{1}{c}\frac{\partial I}{\partial t}\right)(\boldsymbol{x}, t, \boldsymbol{\Omega}, \nu) + o(\mathrm{d}l). \tag{1.11}$$

Furthermore, $\frac{\partial I}{\partial l} = \boldsymbol{\Omega} \cdot \nabla I$ and

$$\begin{aligned}
\left(\frac{1}{c}\partial_t + \boldsymbol{\Omega} \cdot \nabla\right) I(\boldsymbol{x}, t, \boldsymbol{\Omega}, \nu) = &-\left(\sigma_\nu^{\mathrm{a}} + \sigma_\nu^{\mathrm{s}}\right) I(\boldsymbol{x}, t, \boldsymbol{\Omega}, \nu) + \sigma_\nu^{\mathrm{a}} B(\nu, T_{\mathrm{g}}) \\
&+ \sigma_\nu^{\mathrm{s}} \int_{S^2} p_\nu(\boldsymbol{\Omega} \cdot \boldsymbol{\Omega}')\, I(\boldsymbol{x}, t, \boldsymbol{\Omega}', \nu)\, \mathrm{d}\Omega'.
\end{aligned} \tag{1.12}$$

Equation 1.12 is an integro-differential equation that describes the transport of the specific intensity. One can notice the analogy between eq. 1.12 and the Boltzmann equation that describes the statistical behavior of a thermodynamic system out of equilibrium.

Two main regimes can arise, depending on the mean free path of photons compared to the characteristic length of the system, written L [Mihalas and Mihalas, 1984]. On one hand, in the diffusive limit, the medium is optically thick (mean free path of photons much smaller than the characteristic length, $\frac{1}{\sigma_\nu^{\mathrm{a}} + \sigma_\nu^{\mathrm{s}}} \ll L$), the radiation and the matter strongly interact with each other. On the other hand, in the free-streaming regime, the radiation does not affect the gas, and the medium is optically thin (mean free path of photons greater than the characteristic length, $\frac{1}{\sigma_\nu^{\mathrm{a}} + \sigma_\nu^{\mathrm{s}}} \gg L$).

Because solving eq. 1.12 within the scope of radiation hydrodynamics will be costly, we present now the moment method used in this work.

1.2 Moment model

One can consider only the moment of order 0 (the radiative energy), leading to the flux-limited diffusion (FLD) approximation [Levermore and Pomraning, 1981]. Because this model considers only the moment of order 0, its computational cost is quite low, but it is very diffusive in the free-streaming regime. To tackle this issue, one can use a two-moment model (radiative energy and radiative flux), with the M_1 closure relation [Dubroca and Feugeas, 1999]. However, this method can suffer from artifacts when multiple beams cross in the free-streaming regime [González, 2006]. One can solve this issue by using a three-moment model (radiative energy, radiative flux, and radiative pressure) with the M_2 closure relation [Pichard et al., 2016]. However, because of the increase of unknowns, the computational cost also increases. In this work, we have chosen to use the two-moment model with the M_1 closure relation because the computational cost remains affordable, and we do not encounter in our applications the problem of beams crossing in the free-streaming regime.

We limit ourselves to gray radiative transfer, but the following can be generalized to multigroup radiative transfer (e.g., Turpault 2005).

Let us consider the three first moments of the specific intensity: the gray radiative energy E_r, the gray radiative flux $\boldsymbol{F}_\mathbf{r}$ and the gray radiative pressure $\mathbb{P}_\mathrm{r}$ defined as

$$\begin{cases} E_\mathrm{r} = \dfrac{1}{c}\displaystyle\int_0^\infty\int_{S^2} I\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\mathrm{d}\Omega\mathrm{d}\nu \\ \boldsymbol{F}_\mathbf{r} = \displaystyle\int_0^\infty\int_{S^2} I\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\boldsymbol{\Omega}\mathrm{d}\Omega\mathrm{d}\nu \\ \mathbb{P}_\mathrm{r} = \dfrac{1}{c}\displaystyle\int_0^\infty\int_{S^2} I\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\boldsymbol{\Omega}\otimes\boldsymbol{\Omega}\mathrm{d}\Omega\mathrm{d}\nu. \end{cases} \tag{1.13}$$

Let us notice that

$$||\boldsymbol{F}_\mathbf{r}|| \leqslant \int_0^\infty\int_{S^2} \underbrace{I\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)}_{\geqslant 0}\underbrace{||\boldsymbol{\Omega}||}_{=1}\mathrm{d}\Omega\mathrm{d}\nu = cE_\mathrm{r}. \tag{1.14}$$

This condition rewrites $f \leqslant 1$, where $f = ||\boldsymbol{f}|| = \frac{||\boldsymbol{F}_\mathbf{r}||}{cE_\mathrm{r}}$ is the reduced flux. It ensures that the radiative energy cannot be transported faster than the speed of light.

The mean over solid angles and frequency of eq. 1.12 and its product by $\boldsymbol{\Omega}$ give the following system:

$$\begin{cases} \partial_t E_\mathrm{r} + \nabla\cdot\boldsymbol{F}_\mathbf{r} = c\sigma^\mathrm{a}\left(a_\mathrm{r}T_\mathrm{g}^4 - E_\mathrm{r}\right) & \text{(1.15a)} \\ \partial_t\boldsymbol{F}_\mathbf{r} + c^2\nabla\cdot\mathbb{P}_\mathrm{r} = -c\sigma^\mathrm{s}\boldsymbol{F}_\mathbf{r}. & \text{(1.15b)} \end{cases}$$

The computation of the source terms is made explicit in section 1.A. σ^a and σ^s are weighted means of σ_ν^a and σ_ν^s respectively. More details can be found in section 1.A and González [2006].

The fluid and the radiation exchange energy and momentum through emission and absorption. To ensure the conservation of the total energy when the hydrodynamics is frozen, the energy exchange term is given by

$$\partial_t(\rho c_v T_\mathrm{g}) = -c\sigma^\mathrm{a}\left(a_\mathrm{r}T_\mathrm{g}^4 - E_\mathrm{r}\right). \tag{1.16}$$

$\rho c_v T_\mathrm{g}$ is the gas internal energy, with ρ the density of the fluid and c_v the heat capacity, defined by $c_v = \frac{k_\mathrm{b}}{\mu m_H(\gamma-1)}$ for a perfect gas, where μ is the mean molecular weight, m_H is the mass of hydrogen and γ is the adiabatic index of the gas.

In the following section, we specify the closure relation, which is a way to express $\mathbb{P}_\mathrm{r}$ as a function of E_r and $\boldsymbol{F}_\mathbf{r}$.

1.3 Closure relation

We have chosen here to use the M_1 closure relation for its good properties. To derive it, Levermore [1984] applies a Lorentz transform to an isotropic distribution of photons. The same relation can also be obtained by maximizing the radiative entropy [Dubroca and Feugeas, 1999; Buet and Despres, 2004]. We present the latter because some results will be used in chapter 3.

We only give here the main idea of the derivation of the model, detailed computations can be found in section 1.B.

From Buet and Despres [2004], we first define the gray radiative entropy as

$$S_\mathrm{r} = -\frac{2k_\mathrm{b}}{c^3}\int_0^\infty\int_{S^2}\nu^2\left(n\log n-(n+1)\log\left(n+1\right)\right)\mathrm{d}\Omega\mathrm{d}\nu, \tag{1.17}$$

where $n=\frac{c^2}{2h}\frac{I}{\nu^3}$ is the photon occupation number at the frequency ν. At a given time t, in a unit volume centered in $\boldsymbol{x}$ of volume d^3x, there is $\int_0^\infty\int_{S^2} n\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\mathrm{d}\Omega\mathrm{d}\nu\mathrm{d}^3x$ photons. S_r is the entropy of a photon gas. The derivation of eq. 1.17 relies on Gibbs statistical mechanics (e.g., ter Haar 1995) and is beyond the scope of this work.

S_r is the physical entropy, it reaches a maximum. One can also define the mathematical entropy, with the opposite sign (e.g., Dubroca and Feugeas 1999). In that case, the radiative entropy reaches a minimum.

One can show that the radiative entropy S_r is a strictly concave function of the specific intensity I. We write $\bar{n}$ the occupation number that maximizes the radiative entropy. Using the Lagrange multiplier method (e.g., Boyd et al. 2004), there exists θ_r and $\boldsymbol{\beta}$ such that

$$\bar{n}=\frac{1}{\exp\left(\frac{h\nu}{k_\mathrm{b}\theta_\mathrm{r}}\left(1-\frac{\boldsymbol{\beta}\cdot\boldsymbol{\Omega}}{c}\right)\right)-1}. \tag{1.18}$$

θ_r is homogeneous to a temperature, and we have $\theta_\mathrm{r}\geqslant 0$. Let us emphasize that θ_r is not the gas temperature T_g. Similarly, one can show that $\boldsymbol{\beta}$ is homogeneous to a velocity and $\beta=||\boldsymbol{\beta}||\leqslant c$. Let us notice that if $\boldsymbol{\beta}=0$, then $I\left(\bar{n}\right)$ is the black body specific intensity at temperature θ_r.

After computation (see section 1.B), one has

$$\begin{cases} E_\mathrm{r} = c^4\dfrac{3c^2+\beta^2}{3\left(c^2-\beta^2\right)^3}a_\mathrm{r}\theta_\mathrm{r}^4 \\ \boldsymbol{F}_\mathbf{r} = \dfrac{4}{3}\dfrac{c^6a_\mathrm{r}\theta_\mathrm{r}^4}{\left(c^2-\beta^2\right)^3}\boldsymbol{\beta} \\ \quad\ = \left(q+E_\mathrm{r}\right)\boldsymbol{\beta} \\ \mathbb{P}_\mathrm{r} = \dfrac{\boldsymbol{\beta}\otimes\boldsymbol{F}_\mathbf{r}}{c^2}+q\mathbb{I} \\ \quad\ = \left(\dfrac{1-\chi}{2}\mathbb{I}+\dfrac{3\chi-1}{2}\boldsymbol{n}\otimes\boldsymbol{n}\right)E_\mathrm{r}, \end{cases} \tag{1.19}$$

with $q=\frac{c^2-\beta^2}{3c^2+\beta^2}E_\mathrm{r}$, $\mathbb{I}$ the identity matrix, and $\boldsymbol{n}=\frac{\boldsymbol{f}}{f}$ is a unit vector aligned with the radiative flux and $\chi=\frac{3+4f^2}{5+2\sqrt{4-3f^2}}$ is the Eddington factor [Levermore, 1984].

Using eq. 1.19 to express $\mathbb{P}_\mathrm{r}$ as a function of E_r and $\boldsymbol{F_r}$ is the closure relation of the M_1 model. It preserves both free-streaming and diffusive limits. On one hand, if $f = 1$, then $\mathbb{P}_\mathrm{r} = E_\mathrm{r}\boldsymbol{n} \otimes \boldsymbol{n}$, only the transport regime remains. On the other hand, if $f = 0$, the model in the diffusive regime simplifies into the P_1 model, with $\mathbb{P}_\mathrm{r} = \frac{1}{3}E_\mathrm{r}\mathbb{I}$. The radiative pressure tensor becomes diagonal and isotropic. We will look into the diffusion limit more precisely in section 1.4.

Finally, one can also show that the maximum radiative entropy obeys the following hyperbolic equation

$$\partial_t S_\mathrm{r} + \nabla \cdot (\boldsymbol{\beta} S_\mathrm{r}) = \frac{c\sigma^\mathrm{a}}{\theta_\mathrm{r}} \left(a_\mathrm{r} T_\mathrm{g}^4 - E_\mathrm{r}\right) + \frac{\sigma^\mathrm{s}}{c\theta_\mathrm{r}} \boldsymbol{\beta} \cdot \boldsymbol{F_r}. \tag{1.20}$$

This equation will be used in chapter 3. It allows us to prove that the numerical scheme developed there preserves the admissible states $E_\mathrm{r} > 0$ and $f \leqslant 1$.

1.4 Diffusive limit

Let us now focus on the diffusive regime with the hydrodynamics frozen, i.e., the limit of large opacity and long timescale. From Audit et al. [2002], when the grid does not sample the mean free path of the photons, the numerical solution is dominated by the numerical diffusion. In order to understand this phenomena, we explore here the diffusive limit at the continuous level. Computations done here will lead to the development of so-called asymptotic preserving schemes in chapters 2 and 3. These schemes will capture the behavior described in the section, independently of the resolution.

Following Berthon and Turpault [2011], we introduce a rescaling parameter ε to write the time (resp. the opacity) as $\tilde{t} = \varepsilon t$ (resp. $\tilde{\sigma^\mathrm{a}} = \varepsilon\sigma^\mathrm{a}$ and $\tilde{\sigma^\mathrm{s}} = \varepsilon\sigma^\mathrm{s}$). The radiative energy, the radiative flux, and the gas temperature are expanded with ε, e.g., $E_\mathrm{r} = E_{\mathrm{r},0} + \varepsilon E_{\mathrm{r},1} + \mathcal{O}(\varepsilon^2)$. System 1.15 and eq. 1.16 become

$$\begin{cases} \varepsilon^2 \partial_{\tilde{t}} E_\mathrm{r} + \varepsilon \nabla \cdot \boldsymbol{F_r} = c\tilde{\sigma^\mathrm{a}} \left(a_\mathrm{r} T_\mathrm{g}^4 - E_\mathrm{r}\right) & \text{(1.21a)} \\ \varepsilon^2 \partial_{\tilde{t}} \boldsymbol{F_r} + \varepsilon c^2 \nabla \cdot \mathbb{P}_\mathrm{r} = -c\tilde{\sigma^\mathrm{s}} \boldsymbol{F_r} & \text{(1.21b)} \\ \varepsilon^2 \partial_{\tilde{t}} \left(\rho c_v T_\mathrm{g}\right) = -c\tilde{\sigma^\mathrm{a}} \left(a_\mathrm{r} T_\mathrm{g}^4 - E_\mathrm{r}\right). & \text{(1.21c)} \end{cases}$$

By expanding eqs. 1.21a and 1.21b at order 0, we have

$$\begin{cases} E_{\mathrm{r},0} = a_\mathrm{r} T_{\mathrm{g},0}^4 \\ \boldsymbol{F_{r,0}} = \boldsymbol{0}. \end{cases} \tag{1.22}$$

Because $\boldsymbol{F_{r,0}} = \boldsymbol{0}$, let us consider the P_1 closure relation instead of the M_1 closure relation. We use $\mathbb{P}_\mathrm{r} = \frac{1}{3}E_\mathrm{r}\mathbb{I}$ instead of eq. 1.19. This model is known to have the correct behavior only in the diffusive regime. It does not capture properly the free-streaming regime and should be used carefully. However, we use it here for its simplicity.

Expanding eq. 1.21b at order 1 leads to

$$\boldsymbol{F_{r,1}} = -\frac{c}{3\tilde{\sigma^\mathrm{s}}} \nabla E_{\mathrm{r},0}. \tag{1.23}$$

Finally, expanding the sum of eqs. 1.21a and 1.21c at order 2 gives

$$\partial_{\tilde{t}} \left(E_{\mathrm{r},0} + \rho c_v T_{\mathrm{g},0}\right) - \nabla \left(\frac{c}{3\tilde{\sigma^\mathrm{s}}} \nabla E_{\mathrm{r},0}\right) = 0. \tag{1.24}$$

In the diffusive limit, the total energy $E_\mathrm{r} + \rho c_v T_\mathrm{g}$ at order 0 obeys the diffusion equation given by eq. 1.24. Chapters 2 and 3 present our design of two asymptotic preserving schemes, that is, numerical schemes that will degenerate to the discretization of eq. 1.24 in the diffusive regime.

In this section, we only have considered the case where the hydrodynamics is frozen. In the next one, we present the radiation hydrodynamics equations, which take into account the evolution of the fluid.

1.5 Radiation hydrodynamics

We now consider the radiation hydrodynamics equations. The fluid evolution is described by the Euler equations expressing conservation of mass, balance of momentum, and balance of energy. Because the hydrodynamics is no longer frozen and both photons and the gas are moving, we have to evaluate the quantities in the laboratory frame or the comoving frame, which is moving with the fluid. On one hand, using the comoving frame introduces non-conservative terms in the left-hand side of the equations. On the other hand, the hyperbolic part of the system remains simple in the laboratory frame, but some source terms have to be incorporated to describe the interactions between the matter and the radiation. We have chosen the second approach. These source terms characterize the momentum and energy exchanges between the fluid and the radiation:

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0 & \text{(1.25a)} \\ \partial_t (\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u} + p\mathbb{I}) = \mathbf{F}_\mathrm{ext} + \dfrac{\sigma^\mathrm{s}}{c} \boldsymbol{F}_\mathbf{r} - \dfrac{1}{c} \boldsymbol{S}_{\boldsymbol{F}_\mathbf{r}}(\boldsymbol{u}) & \text{(1.25b)} \\ \partial_t (\rho E) + \nabla \cdot ((\rho E + p)\boldsymbol{u}) = \mathbf{F}_\mathrm{ext} \cdot \boldsymbol{u} - c\sigma^\mathrm{a} \left(a_\mathrm{r} T_\mathrm{g}^4 - E_\mathrm{r}\right) - S_{E_\mathrm{r}}(\boldsymbol{u}) & \text{(1.25c)} \\ \partial_t E_\mathrm{r} + \nabla \cdot \boldsymbol{F}_\mathbf{r} = c\sigma^\mathrm{a} \left(a_\mathrm{r} T_\mathrm{g}^4 - E_\mathrm{r}\right) + S_{E_\mathrm{r}}(\boldsymbol{u}) & \text{(1.25d)} \\ \partial_t \boldsymbol{F}_\mathbf{r} + c^2 \nabla \cdot \mathbb{P}_\mathrm{r} = -c\sigma^\mathrm{s} \boldsymbol{F}_\mathbf{r} + c \boldsymbol{S}_{\boldsymbol{F}_\mathbf{r}}(\boldsymbol{u}), & \text{(1.25e)} \end{cases}$$

where $\boldsymbol{u}$ is the material velocity, p is the pressure of the fluid, $\rho E = \rho e + \frac{1}{2}\rho \boldsymbol{u}^2$ is the density of total matter energy with e the specific internal energy. $\mathbf{F}_\mathrm{ext}$ is the external force, for example gravitation, self-gravitation or Lorentz force. The terms $S_{E_\mathrm{r}}(\boldsymbol{u})$ and $\boldsymbol{S}_{\boldsymbol{F}_\mathbf{r}}(\boldsymbol{u})$ depend on the velocity $\boldsymbol{u}$. Using eq. 29 and eq. 31 from Lowrie et al. [1999], we have

$$\begin{cases} S_{E_\mathrm{r}}(\boldsymbol{u}) = \dfrac{2\sigma^\mathrm{a} - \sigma^\mathrm{s}}{c} \boldsymbol{u} \cdot \boldsymbol{F}_\mathbf{r} + \dfrac{\sigma^\mathrm{s}}{c} E_\mathrm{r} \boldsymbol{u} \cdot \boldsymbol{u} + \dfrac{\sigma^\mathrm{s}}{c} \boldsymbol{u} \cdot (\boldsymbol{u} \cdot \mathbb{P}_\mathrm{r}) \\ \boldsymbol{S}_{\boldsymbol{F}_\mathbf{r}}(\boldsymbol{u}) = \sigma^\mathrm{s} \boldsymbol{u} \cdot \mathbb{P}_\mathrm{r} + \sigma^\mathrm{a} a_\mathrm{r} T_\mathrm{g}^4 \boldsymbol{u} + \dfrac{2\sigma^\mathrm{a}}{c^2} \boldsymbol{u} \cdot (\boldsymbol{u} \cdot \boldsymbol{F}_\mathbf{r}) + (\sigma^\mathrm{s} - \sigma^\mathrm{a}) E_\mathrm{r} \boldsymbol{u}, \end{cases} \tag{1.26}$$

at first order in $\frac{\boldsymbol{u}}{c}$. To close the system, we also add the equation of state of an ideal gas: $p = \rho e(\gamma - 1)$.

An asymptotic development for the radiation hydrodynamics case is presented in section 1.C.

1.6 Bibliography

H. Bloch, P. Tremblin, M. González, T. Padioleau, and E. Audit. A high-performance and portable asymptotic preserving radiation hydrodynamics code with the m1

model. *A&A*, 646:A123, 2021. doi: 10.1051/0004-6361/202038579. URL https://doi.org/10.1051/0004-6361/202038579. 18

S. Chandrasekhar. *Radiative Transfer*. Dover Books on Intermediate and Advanced Mathematics. Dover Publications, 1960. ISBN 9780486605906. URL https://books.google.fr/books?id=CK3HDRwCT5YC. 18, 19

C.D. Levermore. Relating eddington factors to flux limiters. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 31(2):149 – 160, 1984. ISSN 0022-4073. doi: https://doi.org/10.1016/0022-4073(84)90112-2. URL http://www.sciencedirect.com/science/article/pii/0022407384901122. 18, 22, 32

D. Mihalas and B. W. Mihalas. *Foundations of radiation hydrodynamics*. 1984. 20

C. D. Levermore and G. C. Pomraning. A flux-limited diffusion theory. *The Astrophysical Journal*, 248:321–334, August 1981. doi: 10.1086/159157. 21

Bruno Dubroca and Jean-Luc Feugeas. Etude théorique et numérique d'une hiérarchie de modèles aux moments pour le transfert radiatif. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 329(10):915 – 920, 1999. ISSN 0764-4442. doi: https://doi.org/10.1016/S0764-4442(00)87499-6. URL http://www.sciencedirect.com/science/article/pii/S0764444200874996. 21, 22

Matthias González. *Contribution to the numerical study of radiation hydrodynamics: from radiative shocks experiments to astrophysical jets*. Theses, Université Paris Sud - Paris XI, October 2006. URL https://tel.archives-ouvertes.fr/tel-00110290. 21, 27

Teddy Pichard, G. Alldredge, Stéphane Brull, B. Dubroca, and M. Frank. The m 2 model for dose simulation in radiation therapy. *Journal of Computational and Theoretical Transport*, pages 1–10, 04 2016. doi: 10.1080/23324309.2016.1150855. 21

Rodolphe Turpault. A consistent multigroup model for radiative transfer and its underlying mean opacities. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 94:357–371, 09 2005. doi: 10.1016/j.jqsrt.2004.09.042. 21

Christophe Buet and Bruno Despres. Asymptotic analysis of fluid models for the coupling of radiation and hydrodynamics. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 85(3):385 – 418, 2004. ISSN 0022-4073. doi: https://doi.org/10.1016/S0022-4073(03)00233-4. URL http://www.sciencedirect.com/science/article/pii/S0022407303002334. 22, 32

D. ter Haar. *Elements of Statistical Mechanics (Third edition)*. Butterworth-Heinemann, Oxford, third edition edition, 1995. ISBN 978-0-7506-2347-6. doi: https://doi.org/10.1016/B978-075062347-6/50017-X. URL https://www.sciencedirect.com/science/article/pii/B978075062347650017X. 22

Stephen Boyd, Stephen P. Boyd, and Lieven Vandenberghe. *Convex Optimization*. 03 2004. ISBN 9780521833783. 22, 29

E. Audit, P. Charrier, J. P. Chièze, and B. Dubroca. A radiation-hydrodynamics scheme valid from the transport to the diffusion limit. *arXiv e-prints*, art. astro-ph/0206281, Jun 2002. 23

Christophe Berthon and Rodolphe Turpault. Asymptotic preserving hll schemes. *Numerical Methods for Partial Differential Equations*, 27:1396 – 1422, 11 2011. doi: 10.1002/num.20586. 23

R. B. Lowrie, J. E. Morel, and J. A. Hittinger. The coupling of radiation and hydrodynamics. 521(1):432–450, aug 1999. doi: 10.1086/307515. URL https://doi.org/10.1086/307515. 24

Mark R. Krumholz, Richard I. Klein, Christopher F. McKee, and John Bolstad. Equations and algorithms for mixedframe fluxlimited diffusion radiation hydrodynamics. *The Astrophysical Journal*, 667(1):626643, Sep 2007. ISSN 1538-4357. doi: 10.1086/520791. URL http://dx.doi.org/10.1086/520791. 34

1.A Computation of source terms

The mean over solid angles and frequency of eq. 1.12 and its product by $\mathbf{\Omega}$ give the following system:

$$\begin{cases} \partial_t E_\mathrm{r} + \nabla \cdot \boldsymbol{F}_\mathbf{r} = S_\mathrm{E} \\ \dfrac{1}{c}\partial_t \boldsymbol{F}_\mathbf{r} + c\nabla \cdot \mathbb{P}_\mathrm{r} = \boldsymbol{S}_\mathbf{F}, \end{cases} \tag{1.27}$$

where

$$\begin{cases} S_\mathrm{E} = -\displaystyle\int_0^\infty \int_{S^2} \left(\sigma_\nu^\mathrm{a} + \sigma_\nu^\mathrm{s}\right) I\left(\boldsymbol{x}, t, \mathbf{\Omega}, \nu\right) \mathrm{d}\Omega \mathrm{d}\nu + \int_0^\infty \int_{S^2} \sigma_\nu^\mathrm{a} B\left(\nu, T_\mathrm{g}\right) \mathrm{d}\Omega \mathrm{d}\nu \\ \qquad + \displaystyle\int_0^\infty \int_{S^2} \left(\sigma_\nu^\mathrm{s} \int_{S^2} p_\nu \left(\mathbf{\Omega} \cdot \mathbf{\Omega}'\right) I\left(\boldsymbol{x}, t, \mathbf{\Omega}', \nu\right) \mathrm{d}\Omega' \right) \mathrm{d}\Omega \mathrm{d}\nu \\ \boldsymbol{S}_\mathbf{F} = -\displaystyle\int_0^\infty \int_{S^2} \left(\sigma_\nu^\mathrm{a} + \sigma_\nu^\mathrm{s}\right) I\left(\boldsymbol{x}, t, \mathbf{\Omega}, \nu\right) \mathbf{\Omega} \mathrm{d}\Omega \mathrm{d}\nu + \displaystyle\int_0^\infty \int_{S^2} \sigma_\nu^\mathrm{a} B\left(\nu, T_\mathrm{g}\right) \mathbf{\Omega} \mathrm{d}\Omega \mathrm{d}\nu \\ \qquad + \displaystyle\int_0^\infty \int_{S^2} \left(\sigma_\nu^\mathrm{s} \int_{S^2} p_\nu \left(\mathbf{\Omega} \cdot \mathbf{\Omega}'\right) I\left(\boldsymbol{x}, t, \mathbf{\Omega}', \nu\right) \mathrm{d}\Omega' \right) \mathbf{\Omega} \mathrm{d}\Omega \mathrm{d}\nu \end{cases} \tag{1.28}$$

are the source terms. We now make them explicit.

Let us consider first the term for the radiative energy equation. We compute each term individually.

First,

$$\begin{aligned} &\int_0^\infty \int_{S^2} \sigma_\nu^\mathrm{s} \int_{S^2} p_\nu \left(\mathbf{\Omega} \cdot \mathbf{\Omega}'\right) I\left(\boldsymbol{x}, t, \mathbf{\Omega}', \nu\right) \mathrm{d}\Omega' \mathrm{d}\Omega \mathrm{d}\nu \\ &= \int_0^\infty \sigma_\nu^\mathrm{s} \int_{S^2} \underbrace{\left(\int_{S^2} p_\nu \left(\mathbf{\Omega} \cdot \mathbf{\Omega}'\right) \mathrm{d}\Omega \right)}_{=1, \forall \mathbf{\Omega}'} I\left(\boldsymbol{x}, t, \mathbf{\Omega}', \nu\right) \mathrm{d}\Omega' \mathrm{d}\nu \\ &= \int_0^\infty \int_{S^2} \sigma_\nu^\mathrm{s} I\left(\boldsymbol{x}, t, \mathbf{\Omega}, \nu\right) \mathrm{d}\Omega \mathrm{d}\nu. \end{aligned} \tag{1.29}$$

Therefore,

$$\begin{aligned} &-\int_0^\infty \int_{S^2} \left(\sigma_\nu^\mathrm{a} + \sigma_\nu^\mathrm{s}\right) I\left(\boldsymbol{x}, t, \mathbf{\Omega}, \nu\right) \mathrm{d}\Omega \mathrm{d}\nu + \int_0^\infty \int_{S^2} \left(\sigma_\nu^\mathrm{s} \int_{S^2} p_\nu \left(\mathbf{\Omega} \cdot \mathbf{\Omega}'\right) I\left(\boldsymbol{x}, t, \mathbf{\Omega}', \nu\right) \mathrm{d}\Omega' \right) \mathrm{d}\Omega \mathrm{d}\nu \\ &= -\int_0^\infty \int_{S^2} \sigma_\nu^\mathrm{a} I\left(\boldsymbol{x}, t, \mathbf{\Omega}, \nu\right) \mathrm{d}\Omega \mathrm{d}\nu \\ &= -\sigma_\mathrm{E} c E_\mathrm{r}, \end{aligned} \tag{1.30}$$

where σ_{E} is the mean over frequency of σ_ν^{a} weighted by the radiative energy. Second,

$$\int_0^\infty \int_{S^2} \sigma_\nu^{\mathrm{a}} B\left(\nu, T_{\mathrm{g}}\right) \mathrm{d}\Omega \mathrm{d}\nu = \frac{2h}{c^2} \int_0^\infty \int_{S^2} \sigma_\nu^{\mathrm{a}} \frac{\nu^3}{\exp\left(\frac{h\nu}{k_{\mathrm{b}} T_{\mathrm{g}}}\right) - 1} \mathrm{d}\Omega \mathrm{d}\nu. \tag{1.31}$$

Using the change of variable $r = \frac{h\nu}{k_{\mathrm{b}} T_{\mathrm{g}}}$,

$$\int_0^\infty \int_{S^2} \sigma_\nu^{\mathrm{a}} B\left(\nu, T_{\mathrm{g}}\right) \mathrm{d}\Omega \mathrm{d}\nu = \frac{2k_{\mathrm{b}}^4}{c^2 h^3} T_{\mathrm{g}}^4 \int_{S^2} \mathrm{d}\Omega \int_0^\infty \sigma_{\nu(r)}^{\mathrm{a}} \frac{r^3}{\exp\left(r\right) - 1} \mathrm{d}r. \tag{1.32}$$

$\int_{S^2} \mathrm{d}\Omega$ is the area of the unit sphere, therefore $\int_{S^2} \mathrm{d}\Omega = 4\pi$. Using Bose-Einstein function, one has

$$\int_0^\infty \frac{r^3}{\exp\left(r\right) - 1} \mathrm{d}r = \zeta(4)\Gamma(4) = \frac{\pi^4}{15}, \tag{1.33}$$

where ζ is the Riemann zeta function and Γ is the gamma function. Finally,

$$\begin{aligned} S_{\mathrm{E}} &= \sigma_{\mathrm{P}} \frac{8\pi^5 k_{\mathrm{b}}^4}{15 h^3 c^2} T_{\mathrm{g}}^4 - \sigma_{\mathrm{E}} c E_{\mathrm{r}} \\ &= c\left(\sigma_{\mathrm{P}} a_{\mathrm{r}} T_{\mathrm{g}}^4 - \sigma_{\mathrm{E}} E_{\mathrm{r}}\right), \end{aligned} \tag{1.34}$$

where σ_{P} is the mean over frequency of σ_ν^{a} weighted by the Planck function and $a_{\mathrm{r}} = \frac{8\pi^5 k_{\mathrm{b}}^4}{15 c^3 h^3}$ is the Stefan-Boltzmann constant.

Likewise, we compute each term of $\boldsymbol{S}_{\mathbf{F}}$. First,

$$\int_0^\infty \int_{S^2} \left(\sigma_\nu^{\mathrm{a}} + \sigma_\nu^{\mathrm{s}}\right) I\left(\boldsymbol{x}, t, \boldsymbol{\Omega}, \nu\right) \boldsymbol{\Omega} \mathrm{d}\Omega \mathrm{d}\nu = \left(\sigma_{\mathrm{F}} + \sigma_{\mathrm{R}}\right) \boldsymbol{F}_{\mathbf{r}}, \tag{1.35}$$

where σ_{F} (resp. σ_{R}) is the mean over frequency of σ_ν^{a} (resp. σ_ν^{s}) weighted by the radiative flux. Second,

$$\int_0^\infty \int_{S^2} \int_{S^2} p_\nu \left(\boldsymbol{\Omega} \cdot \boldsymbol{\Omega}\right) \boldsymbol{\Omega} \mathrm{d}\Omega I\left(\boldsymbol{x}, t, \boldsymbol{\Omega}, \nu\right) \mathrm{d}\Omega' \mathrm{d}\nu = \sigma_{\mathrm{R}} g \boldsymbol{F}_{\mathbf{r}}, \tag{1.36}$$

with g the first moment of the phase function. Third, because B depends only on the temperature and the frequency, it is isotropic and

$$\int_{S^2} B\left(\boldsymbol{\Omega}, \nu, T_{\mathrm{g}}\right) \boldsymbol{\Omega} \mathrm{d}\Omega = 0. \tag{1.37}$$

Finally,

$$\boldsymbol{S}_{\mathbf{F}} = -\left(\sigma_{\mathrm{F}} + (1 - g)\, \sigma_{\mathrm{R}}\right) \boldsymbol{F}_{\mathbf{r}}. \tag{1.38}$$

The means of opacity σ_{E}, σ_{P}, σ_{F} and σ_{R} can be connected with Rosseland and Planck mean opacities. See González [2006] for more details. In the following, we always make the approximation $\sigma_{\mathrm{E}} = \sigma_{\mathrm{P}}$ and we write $\sigma_{\mathrm{E}} = \sigma_{\mathrm{P}} = \sigma^{\mathrm{a}}$. Furthermore, we note $\sigma^{\mathrm{s}} = \sigma_{\mathrm{F}} + (1 - g)\, \sigma_{\mathrm{R}}$ to write eq. 1.27 as

$$\begin{cases} \partial_t E_{\mathrm{r}} + \nabla \cdot \boldsymbol{F}_{\mathbf{r}} = c\sigma^{\mathrm{a}} \left(a_{\mathrm{r}} T_{\mathrm{g}}^4 - E_{\mathrm{r}}\right) \\ \partial_t \boldsymbol{F}_{\mathbf{r}} + c^2 \nabla \cdot \mathbb{P}_{\mathrm{r}} = -c\sigma^{\mathrm{s}} \boldsymbol{F}_{\mathbf{r}}. \end{cases} \tag{1.39}$$

1.B Maximization of radiative entropy

1.B.1 Radiative entropy

Let us recall that we have defined the radiative entropy as

$$S_{\mathrm{r}} = -\frac{2k_{\mathrm{b}}}{c^3}\int_0^\infty\int_{S^2}\nu^2\left(n\log n-(n+1)\log\left(n+1\right)\right)\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\mathrm{d}\Omega\mathrm{d}\nu. \tag{1.40}$$

We also define the entropy flux as

$$\boldsymbol{Q}_{\mathbf{r}} = -\frac{2k_{\mathrm{b}}}{c^2}\int_0^\infty\int_{S^2}\nu^2\left(n\log n-(n+1)\log\left(n+1\right)\right)\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\boldsymbol{\Omega}\mathrm{d}\Omega\mathrm{d}\nu. \tag{1.41}$$

Let us first compute the equation of evolution for the radiative entropy. From now on, we assume that eq. 1.12 has a solution smooth enough. This hypothesis allows us to compute the derivatives without further justification. Then, because $n = \frac{c^2}{2h}\frac{I}{\nu^3}$,

$$\begin{aligned}\partial_t S_{\mathrm{r}} &= -\frac{2k_{\mathrm{b}}}{c^3}\int_0^\infty\int_{S^2}\nu^2\partial_t\left(n\log n-(n+1)\log\left(n+1\right)\right)\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\mathrm{d}\Omega\mathrm{d}\nu\\ &= -\frac{2k_{\mathrm{b}}}{c^3}\int_0^\infty\int_{S^2}\nu^2\log\left(\frac{n}{n+1}\right)\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\partial_t n\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\mathrm{d}\Omega\mathrm{d}\nu\\ &= -\frac{k_{\mathrm{b}}}{ch}\int_0^\infty\int_{S^2}\frac{1}{\nu}\log\left(\frac{n}{n+1}\right)\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\partial_t I\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\mathrm{d}\Omega\mathrm{d}\nu.\end{aligned} \tag{1.42}$$

Likewise, one can show that

$$\nabla\cdot\boldsymbol{Q}_{\mathbf{r}} = -\frac{k_{\mathrm{b}}}{h}\int_0^\infty\int_{S^2}\frac{1}{\nu}\log\left(\frac{n}{n+1}\right)\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\boldsymbol{\Omega}\cdot\nabla I\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\mathrm{d}\Omega\mathrm{d}\nu. \tag{1.43}$$

Therefore,

$$\begin{aligned}\partial_t S_{\mathrm{r}}+\nabla\cdot\boldsymbol{Q}_{\mathbf{r}} &= -\frac{k_{\mathrm{b}}}{h}\int_0^\infty\int_{S^2}\frac{1}{\nu}\log\left(\frac{n}{n+1}\right)\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\left(\frac{1}{c}\partial_t I+\boldsymbol{\Omega}\cdot\nabla I\right)\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\mathrm{d}\Omega\mathrm{d}\nu\\ &= \frac{k_{\mathrm{b}}}{h}\int_0^\infty\int_{S^2}\frac{1}{\nu}\log\left(\frac{n}{n+1}\right)\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\left(\sigma_\nu^{\mathrm{a}}+\sigma_\nu^{\mathrm{s}}\right)I\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\mathrm{d}\Omega\mathrm{d}\nu\\ &\quad-\frac{k_{\mathrm{b}}}{h}\int_0^\infty\int_{S^2}\frac{1}{\nu}\log\left(\frac{n}{n+1}\right)\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\sigma_\nu^{\mathrm{a}}B\left(\nu,T_{\mathrm{g}}\right)\mathrm{d}\Omega\mathrm{d}\nu\\ &\quad-\frac{k_{\mathrm{b}}}{h}\int_0^\infty\int_{S^2}\frac{1}{\nu}\log\left(\frac{n}{n+1}\right)\left(\boldsymbol{x},t,\boldsymbol{\Omega},\nu\right)\sigma_\nu^{\mathrm{s}}\int_{S^2}p_\nu\left(\boldsymbol{\Omega}\cdot\boldsymbol{\Omega}'\right)I\left(\boldsymbol{x},t,\boldsymbol{\Omega}',\nu\right)\mathrm{d}\Omega'\mathrm{d}\Omega\mathrm{d}\nu.\end{aligned} \tag{1.44}$$

The radiative entropy obeys a hyperbolic equation. The source terms will be made explicit in section 1.B.4.

1.B.2 Maximization of entropy

Let us now compute $\bar{n}$ that maximizes the radiative entropy. We first show that the radiative entropy S_{r} is a strictly concave as a function of the specific intensity I. Let us recall that a function is strictly concave if and only if its second derivative is strictly negative.

After some computations similar to what has been done in the previous section, one can show that

$$\begin{cases} \dfrac{\partial S_\mathrm{r}}{\partial I} = -\dfrac{k_\mathrm{b}}{ch}\displaystyle\int_0^\infty\int_{S^2}\frac{1}{\nu}\log\left(\frac{n}{n+1}\right)(\boldsymbol{x},t,\boldsymbol{\Omega},\nu)\,\mathrm{d}\Omega\mathrm{d}\nu \\ \dfrac{\partial^2 S_\mathrm{r}}{\partial I^2} = -\dfrac{ck_\mathrm{b}}{2h^2}\displaystyle\int_0^\infty\int_{S^2}\frac{1}{\nu^4}\frac{1}{(n\,(n+1))\,(\boldsymbol{x},t,\boldsymbol{\Omega},\nu)}\mathrm{d}\Omega\mathrm{d}\nu. \end{cases} \tag{1.45}$$

The second equation shows that S_r is strictly concave. Because $\max S_\mathrm{r}(n) = -\min(-S_\mathrm{r}(n))$, computing the maximum of radiative entropy leads to solve the convex optimization problem $-\min(-S_\mathrm{r}(n))$ under the constraints $-E_\mathrm{r}(n) \leqslant 0$ and $||\boldsymbol{F}_\mathbf{r}(n)|| - cE_\mathrm{r}(n) \leqslant 0$.

Using the Karush-Kuhn-Tucker (KKT) conditions (e.g., Boyd et al. 2004), there exists $\lambda_1 \geqslant 0$ and $\lambda_2 \geqslant 0$ such that

$$\frac{\partial S_\mathrm{r}}{\partial n}(\bar{n}) - \lambda_1\frac{\partial E_\mathrm{r}}{\partial n}(\bar{n}) + \lambda_2\left(\frac{\partial\,||\boldsymbol{F}_\mathbf{r}||}{\partial n}(\bar{n}) - c\frac{\partial E_\mathrm{r}}{\partial n}(\bar{n})\right) = 0. \tag{1.46}$$

Let us introduce two variables, $\theta_\mathrm{r} \in \mathbb{R}$ and $\boldsymbol{\beta} \in \mathbb{R}^3$ defined as

$$\begin{cases} \theta_\mathrm{r} = \dfrac{1}{\lambda_1 + c\lambda_2} \\ \boldsymbol{\beta} = c^2\theta_\mathrm{r}\lambda_2\dfrac{\boldsymbol{F}_\mathbf{r}(\bar{n})}{||\boldsymbol{F}_\mathbf{r}(\bar{n})||}. \end{cases} \tag{1.47}$$

θ_r is obviously non-negative and homogeneous to a temperature. Some easy computations show that $\boldsymbol{\beta}$ is homogeneous to a velocity and $\beta = ||\boldsymbol{\beta}|| \leqslant c$:

$$\begin{aligned} &\lambda_1 \geqslant 0 \\ \Leftrightarrow\ &\beta = \frac{c^2\lambda_2}{\lambda_1 + c\lambda_2} \leqslant \frac{c^2\lambda_2}{c\lambda_2} = c. \end{aligned} \tag{1.48}$$

Equation 1.46 rewrites

$$\frac{\partial S_\mathrm{r}}{\partial n}(\bar{n}) - \frac{1}{\theta_\mathrm{r}}\frac{\partial E_\mathrm{r}}{\partial n}(\bar{n}) + \frac{\boldsymbol{\beta}}{c^2\theta_\mathrm{r}}\cdot\frac{\partial \boldsymbol{F}_\mathbf{r}}{\partial n}(\bar{n}) = 0. \tag{1.49}$$

Using eq. 1.45 and the definition of the radiation energy and the radiative flux, one has

$$0 = -\int_0^\infty\int_{S^2}\left(\frac{k_\mathrm{b}}{ch\nu}\log\left(\frac{\bar{n}}{\bar{n}+1}\right)(\boldsymbol{x},t,\boldsymbol{\Omega},\nu) + \frac{1}{c\theta_\mathrm{r}} - \frac{\boldsymbol{\beta}\cdot\boldsymbol{\Omega}}{c^2\theta_\mathrm{r}}\right)\mathrm{d}\Omega\mathrm{d}\nu. \tag{1.50}$$

Because S_r is strictly concave, it reaches exactly one maximum. Therefore, if we find one $\bar{n}$ that verifies eq. 1.50, it will maximize the radiative entropy. $\bar{n}$ such that

$$\frac{k_\mathrm{b}}{ch\nu}\log\left(\frac{\bar{n}}{\bar{n}+1}\right) + \frac{1}{c\theta_\mathrm{r}} - \frac{\boldsymbol{\beta}\cdot\boldsymbol{\Omega}}{c^2\theta_\mathrm{r}} = 0, \tag{1.51}$$

verifies eq. 1.50 and therefore maximizes S_r. Equation 1.51 leads to

$$\bar{n} = \frac{1}{\exp\left(\frac{h\nu}{k_\mathrm{b}\theta_\mathrm{r}}\left(1 - \frac{\boldsymbol{\beta}\cdot\boldsymbol{\Omega}}{c}\right)\right) - 1}. \tag{1.52}$$

$\bar{n}$ given by eq. 1.52 maximizes the radiative entropy S_r.

At global thermodynamic equilibrium, $\boldsymbol{\beta} = \mathbf{0}$ and the specific intensity given by eq. 1.52 is the black body specific intensity (eq. 1.6), by identifying θ_r with the gas temperature, T_g.

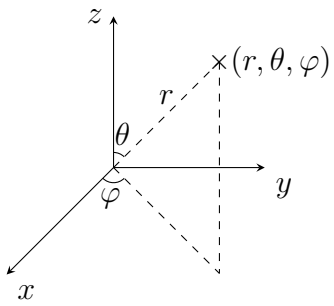


Figure 1.3 – Spherical coordinates.

1.B.3 M_1 closure relation

To derive $\mathbb{P}_\mathrm{r}$ as a function of E_r and $\boldsymbol{F}_\mathbf{r}$, we express the three moments of the specific intensity as functions of θ_r and $\boldsymbol{\beta}$.

1.B.3.1 Radiative energy

Let us start with the radiative energy. Using $I = \frac{2h\nu^3\bar{n}}{c^2} = \frac{2h}{c^2}\frac{\nu^3}{\exp(-\frac{\nu}{\alpha}(1+\boldsymbol{A}\cdot\boldsymbol{\Omega}))-1}$ with $\alpha = \frac{k_\mathrm{b}\theta_\mathrm{r}}{h}$ and $\boldsymbol{A} = -\frac{\boldsymbol{\beta}}{c}$, one has

$$E_\mathrm{r} = \frac{2h}{c^3}\int_0^\infty\int_{S^2}\frac{\nu^3}{\exp\left(-\frac{\nu}{\alpha}\left(1+\boldsymbol{A}\cdot\boldsymbol{\Omega}\right)\right)-1}\mathrm{d}\Omega\mathrm{d}\nu. \tag{1.53}$$

Using the change of variable $r = -\frac{\nu}{\alpha}\left(1+\boldsymbol{A}\cdot\boldsymbol{\Omega}\right)$, one has

$$E_\mathrm{r} = \frac{2h\alpha^4}{c^3}\int_0^\infty\frac{r^3}{\exp\left(r\right)-1}\mathrm{d}r\int_{S^2}\frac{1}{\left(1+\boldsymbol{A}\cdot\boldsymbol{\Omega}\right)^4}\mathrm{d}\Omega. \tag{1.54}$$

The integral over the frequency has already been made explicit in section 1.A. Let us now focus on the integral over the unit sphere. We choose a reference frame such that the z-axis is parallel to the vector $\boldsymbol{A}$ and θ is the angle between $\boldsymbol{A}$ and $\boldsymbol{\Omega}$ (see fig. 1.3). Therefore,

$$\boldsymbol{A}\cdot\boldsymbol{\Omega} = A\underbrace{||\boldsymbol{\Omega}||}_{=1}\cos\theta, \tag{1.55}$$

with $A = ||\boldsymbol{A}||$. Furthermore, we write the units vector $\boldsymbol{e_x}$, $\boldsymbol{e_y}$ and $\boldsymbol{e_z}$. In particular, $\boldsymbol{e_z} = \frac{\boldsymbol{A}}{A}$.

We can now compute the integral over the unit sphere:

$$\int_{S^2}\frac{1}{\left(1+\boldsymbol{A}\cdot\boldsymbol{\Omega}\right)^4}\mathrm{d}\Omega = \int_0^{2\pi}\mathrm{d}\varphi\int_0^\pi\frac{\sin\theta}{\left(1+A\cos\theta\right)^4}\mathrm{d}\theta = \frac{4\pi\left(3+A^2\right)}{3\left(1-A^2\right)^3}. \tag{1.56}$$

Finally,

$$E_\mathrm{r} = c^4\frac{3c^2+\beta^2}{3\left(c^2-\beta^2\right)^3}a_\mathrm{r}\theta_\mathrm{r}^4. \tag{1.57}$$

1.B.3.2 Radiative flux

Likewise, one can compute $\boldsymbol{F}_\mathbf{r}$:

$$\begin{aligned}\boldsymbol{F}_\mathbf{r} &= \frac{2h}{c^2}\int_0^\infty\int_{S^2}\frac{\nu^3}{\exp\left(-\frac{\nu}{\alpha}\left(1+\boldsymbol{A}\cdot\boldsymbol{\Omega}\right)\right)-1}\boldsymbol{\Omega}\mathrm{d}\Omega\mathrm{d}\nu\\ &= \frac{2h\alpha^4}{c^2}\frac{\pi^4}{15}\int_{S^2}\frac{\boldsymbol{\Omega}}{\left(1+\boldsymbol{A}\cdot\boldsymbol{\Omega}\right)^4}\mathrm{d}\Omega.\end{aligned} \tag{1.58}$$

Again, let us focus on the integral over the sphere. Using spherical coordinates, we can write

$$\boldsymbol{\Omega} = \begin{pmatrix} \sin\theta\cos\varphi \\ \sin\theta\sin\varphi \\ \cos\theta \end{pmatrix}. \tag{1.59}$$

Therefore, the x-coordinate of $\int_{S^2} \frac{\boldsymbol{\Omega}}{(1+\boldsymbol{A}\cdot\boldsymbol{\Omega})^4}\mathrm{d}\Omega$ is

$$\int_0^{2\pi} \cos\varphi \mathrm{d}\varphi \int_0^{\pi} \frac{\sin^2\theta}{(1+A\cos\theta)^4}\mathrm{d}\theta = 0. \tag{1.60}$$

Likewise, the y-coordinate of the integral is also 0. The z-coordinate is

$$\int_0^{2\pi} \mathrm{d}\varphi \int_0^{\pi} \frac{\cos\theta\sin\theta}{(1+A\cos\theta)^4}\mathrm{d}\theta = -16\pi\frac{A}{3\left(1-A^2\right)^3}. \tag{1.61}$$

Finally,

$$\begin{aligned} \boldsymbol{F_r} &= -\frac{\pi^4}{15}\frac{2h\alpha^4}{c^2}16\pi\frac{A}{3\left(1-A^2\right)^3}\boldsymbol{e_z} \\ &= \frac{4}{3}\frac{c^6 a_\mathrm{r}\theta_\mathrm{r}^4}{\left(c^2-\beta^2\right)^3}\boldsymbol{\beta} \\ &= \left(q+E_\mathrm{r}\right)\boldsymbol{\beta}, \end{aligned} \tag{1.62}$$

with $q = \frac{c^2-\beta^2}{3c^2+\beta^2}E_\mathrm{r} = \frac{1-\chi}{2}E_\mathrm{r}$, where χ is the Eddington factor, defined by $\chi = \frac{3+4f^2}{5+2\sqrt{4-3f^2}}$. The last expression for $\boldsymbol{F_r}$ will be used in chapter 3.

Equations 1.57 and 1.62 are equivalent to

$$\begin{aligned} \boldsymbol{\beta} &= \frac{c}{f}\frac{3\chi-1}{2}\boldsymbol{f} \\ \theta_\mathrm{r} &= \frac{2}{f}\left(-1+\sqrt{4-3f^2}\right)^{\frac{1}{4}}\sqrt{f^2-2+\sqrt{4-3f^2}}\left(\frac{E_\mathrm{r}}{a_\mathrm{r}}\right)^{\frac{1}{4}}. \end{aligned} \tag{1.63}$$

Finally, using $I = \frac{h\nu^3 n}{c^2}$ and eq. 1.52, the distribution of photons associated to the M_1 model is

$$\frac{2h\nu^3}{c^2}\frac{1}{\exp\left(\frac{h\nu}{k_\mathrm{b}\theta_\mathrm{r}}\left(1-\frac{2-\sqrt{4-3f^2}}{f^2}\boldsymbol{f}\cdot\boldsymbol{\Omega}\right)\right)}. \tag{1.64}$$

1.B.3.3 Radiative pressure

Finally,

$$\begin{aligned} \mathbb{P}_\mathrm{r} &= \frac{2h}{c^3}\int_0^{\infty}\int_{S^2}\frac{\nu^3}{\exp\left(-\frac{\nu}{\alpha}\left(1+\boldsymbol{A}\cdot\boldsymbol{\Omega}\right)\right)-1}\boldsymbol{\Omega}\otimes\boldsymbol{\Omega}\mathrm{d}\Omega\mathrm{d}\nu \\ &= \frac{2h\alpha^4}{c^3}\frac{\pi^4}{15}\int_{S^2}\frac{\boldsymbol{\Omega}\otimes\boldsymbol{\Omega}}{\left(1+\boldsymbol{A}\cdot\boldsymbol{\Omega}\right)^4}\mathrm{d}\Omega. \end{aligned} \tag{1.65}$$

As before, we focus on the integral over the unit sphere. $\boldsymbol{\Omega}\otimes\boldsymbol{\Omega}$ is a symmetric 3×3 matrix and, using eq. 1.59, we have

$$\boldsymbol{\Omega}\otimes\boldsymbol{\Omega} = \begin{pmatrix} \sin^2\theta\cos^2\varphi & \sin^2\theta\cos\varphi\sin\varphi & \sin\theta\cos\theta\cos\varphi \\ \sin^2\theta\cos\varphi\sin\varphi & \sin^2\theta\sin^2\varphi & \sin\theta\cos\theta\sin\varphi \\ \sin\theta\cos\theta\cos\varphi & \sin\theta\cos\theta\cos\varphi & \cos^2\theta \end{pmatrix}. \tag{1.66}$$

Because $\mathbb{P}_\mathrm{r}$ is a tensor, we write its components $\mathbb{P}_\mathrm{r}^{i,j}$ for $i,j \in \{x,y,z\}$. With this notation, $\mathbb{P}_\mathrm{r}^{x,y} = \mathbb{P}_\mathrm{r}^{y,x} = \mathbb{P}_\mathrm{r}^{x,z} = \mathbb{P}_\mathrm{r}^{z,x} = \mathbb{P}_\mathrm{r}^{y,z} = \mathbb{P}_\mathrm{r}^{z,y} = 0$. One can easily compute individually the non-zero coefficients:

$$\begin{aligned}\mathbb{P}_\mathrm{r}^{x,x} &= \mathbb{P}_\mathrm{r}^{y,y} = \frac{2h\alpha^4}{c^3}\frac{\pi^4}{15}\frac{4\pi}{3\left(1-A^2\right)^2} \\ \mathbb{P}_\mathrm{r}^{z,z} &= \frac{2h\alpha^4}{c^3}\frac{\pi^4}{15}\frac{4\pi\left(1+3A^2\right)}{3\left(1-A^2\right)^3} = \frac{2h\alpha^4}{c^3}\frac{\pi^4}{15}\left(\frac{4\pi}{3\left(1-A^2\right)^2} + \frac{4\pi\times 4A^2}{3\left(1-A^2\right)^3}\right).\end{aligned} \tag{1.67}$$

Therefore,

$$\begin{aligned}\mathbb{P}_\mathrm{r} &= \frac{c^4 a_\mathrm{r}\theta_\mathrm{r}^4}{3\left(c^2-\beta^2\right)^2}\left(\boldsymbol{e_x}\otimes\boldsymbol{e_x}+\boldsymbol{e_y}\otimes\boldsymbol{e_y}+\boldsymbol{e_z}\otimes\boldsymbol{e_z}\right) + \frac{4c^4 a_\mathrm{r}\theta_\mathrm{r}^4\beta^2}{3\left(c^2-\beta^2\right)^3}\boldsymbol{e_z}\otimes\boldsymbol{e_z} \\ &= \frac{c^4 a_\mathrm{r}\theta_\mathrm{r}^4}{3\left(c^2-\beta^2\right)^2}\mathbb{I} + \frac{4c^4 a_\mathrm{r}\theta_\mathrm{r}^4\beta^2}{3\left(c^2-\beta^2\right)^3}\boldsymbol{e_z}\otimes\boldsymbol{e_z}.\end{aligned} \tag{1.68}$$

Nevertheless,

$$\boldsymbol{e_z} = \frac{\boldsymbol{A}}{A} = -\frac{\boldsymbol{\beta}}{\beta} = -\frac{3c^2+\beta^2}{4c\beta}\boldsymbol{f}. \tag{1.69}$$

Finally,

$$\begin{aligned}\mathbb{P}_\mathrm{r} &= \left(\frac{c^2-\beta^2}{3c^2+\beta^2}\mathbb{I} + \frac{3c^2+\beta^2}{4c^2}\boldsymbol{f}\otimes\boldsymbol{f}\right)E_\mathrm{r} \\ &= \frac{\boldsymbol{\beta}\otimes\boldsymbol{F_r}}{c^2} + q\mathbb{I}.\end{aligned} \tag{1.70}$$

One can also check that eq. 1.70 is equivalent to $\mathbb{P}_\mathrm{r} = \mathbb{D}E_\mathrm{r}$, where $\mathbb{D}$ is the Eddington tensor, defined by $\mathbb{D} = \frac{1-\chi}{2}\mathbb{I} + \frac{3\chi-1}{2}\boldsymbol{n}\otimes\boldsymbol{n}$, with $\boldsymbol{n} = \frac{\boldsymbol{f}}{f}$ a unit vector aligned with the radiative flux [Levermore, 1984].

1.B.4 Back to the radiative entropy

In the previous section, we have expressed the moments of the specific intensity as functions of θ_r and $\boldsymbol{\beta}$. Following Buet and Despres [2004, Appendix B], we do the same for the maximum of radiative entropy. Using eq. 1.17, the maximum of entropy is given by

$$S_\mathrm{r} = -\frac{2k_\mathrm{b}}{c^3}\int_0^\infty\int_{S^2}\nu^2\left(\bar{n}\log\bar{n} - (\bar{n}+1)\log(\bar{n}+1)\right)\mathrm{d}\Omega\mathrm{d}\nu. \tag{1.71}$$

Using the change of variable $r = -\frac{\nu}{\alpha}(1+\boldsymbol{\Omega}\cdot\boldsymbol{A})$,

$$S_\mathrm{r} = -\frac{8\pi k_\mathrm{b}^4\theta_\mathrm{r}^3 c}{h^3\left(c^2-\beta^2\right)^2}\int_{S^2}\frac{1}{\left(1+\boldsymbol{\Omega}\cdot\boldsymbol{A}\right)^3}\mathrm{d}\Omega\int_0^\infty r^2\left(m\log m-(m+1)\log(m+1)\right)\mathrm{d}r, \tag{1.72}$$

where $m = \frac{1}{\exp(r)-1}$. We introduce a function h such that

$$h(\alpha) = \int_0^\infty r^2\left(M_\alpha\log M_\alpha - (M_\alpha+1)\log(M_\alpha+1)\right)\mathrm{d}r, \tag{1.73}$$

with $M_\alpha = \frac{1}{\exp(\frac{r}{\alpha})-1}$ to write

$$S_\mathrm{r} = -\frac{8\pi k_\mathrm{b}^4\theta_\mathrm{r}^3 c}{h^3\left(c^2-\beta^2\right)^2}h(1). \tag{1.74}$$

We now have to compute $h(\alpha)$. Let us notice that $\frac{M_\alpha}{M_\alpha+1} = \exp\left(\frac{\xi}{\alpha}\right)$. We then have

$$h'(\alpha) = -\alpha^2 \int_0^\infty \frac{\xi^3 \exp(\xi)}{(\exp(\xi)-1)} \mathrm{d}\xi. \tag{1.75}$$

By integrating by parts, we recognize the Bose-Einstein function:

$$h'(\alpha) = -\alpha^2 \int_0^\infty \frac{4\xi^3}{\exp(\xi)-1} \mathrm{d}\xi = -\frac{4\pi^4\alpha^2}{15}. \tag{1.76}$$

Because the entropy is defined up to a constant, integrating this equation leads to

$$h(\alpha) = -\frac{4\pi^4\alpha^3}{45}. \tag{1.77}$$

Finally,

$$S_\mathrm{r} = \frac{4c^4 a_\mathrm{r} \theta_\mathrm{r}^3}{3\left(c^2-\beta^2\right)^2}. \tag{1.78}$$

Likewise, we compute the entropy flux, defined by eq. 1.41:

$$\begin{aligned}
\boldsymbol{Q}_\mathbf{r} &= -\frac{2k_\mathrm{b}}{c^2} \int_0^\infty \int_{S^2} \nu^2 \boldsymbol{\Omega} \left(\bar{n}\log\bar{n} - (\bar{n}+1)\log(\bar{n}+1)\right) \mathrm{d}\Omega\mathrm{d}\nu \\
&= -\frac{2k_\mathrm{b}^4\theta_\mathrm{r}^3}{c^2h^3} g(1) \int_{S^2} \frac{\boldsymbol{\Omega}}{(1+\boldsymbol{\Omega}\cdot\boldsymbol{A})^3} \mathrm{d}\Omega \\
&= \frac{4c^4 a_\mathrm{r}\theta_\mathrm{r}^3}{3\left(c^2-\beta^2\right)^2}\boldsymbol{\beta} \\
&= S_\mathrm{r}\boldsymbol{\beta}.
\end{aligned} \tag{1.79}$$

Equation 1.44 simplifies into

$$\begin{aligned}
\partial_t S_\mathrm{r} + \nabla\cdot(S_\mathrm{r}\boldsymbol{\beta}) = &\frac{k_\mathrm{b}}{h}\int_0^\infty\int_{S^2} \frac{1}{\nu}\log\left(\frac{n}{n+1}\right)\left(\sigma_\nu^\mathrm{a}+\sigma_\nu^\mathrm{s}\right) I(\boldsymbol{x},t,\boldsymbol{\Omega},\nu)\,\mathrm{d}\Omega\mathrm{d}\nu \\
&- \frac{k_\mathrm{b}}{h}\int_0^\infty\int_{S^2} \frac{1}{\nu}\log\left(\frac{n}{n+1}\right)\sigma_\nu^\mathrm{a} B(\nu,T_\mathrm{g})\,\mathrm{d}\Omega\mathrm{d}\nu \\
&- \frac{k_\mathrm{b}}{h}\int_0^\infty\int_{S^2} \frac{1}{\nu}\log\left(\frac{n}{n+1}\right)\sigma_\nu^\mathrm{s}\int_{S^2} p_\nu(\boldsymbol{\Omega}\cdot\boldsymbol{\Omega}') I(\boldsymbol{x},t,\boldsymbol{\Omega}',\nu)\,\mathrm{d}\Omega'\mathrm{d}\Omega\mathrm{d}\nu.
\end{aligned} \tag{1.80}$$

We can now specify the source term of eq. 1.80. Let us recall that $\frac{k_\mathrm{b}}{h\nu}\log\left(\frac{\bar{n}}{\bar{n}+1}\right) = \frac{\boldsymbol{\beta}\cdot\boldsymbol{\Omega}}{c\theta_\mathrm{r}} - \frac{1}{\theta_\mathrm{r}}$. Therefore,

$$\begin{aligned}
\partial_t S_\mathrm{r} + \nabla\cdot(S_\mathrm{r}\boldsymbol{\beta}) = &\int_0^\infty\int_{S^2} \left(\frac{\boldsymbol{\beta}\cdot\boldsymbol{\Omega}}{c\theta_\mathrm{r}} - \frac{1}{\theta_\mathrm{r}}\right)\left(\sigma_\nu^\mathrm{a}+\sigma_\nu^\mathrm{s}\right) I(\boldsymbol{x},t,\boldsymbol{\Omega},\nu)\,\mathrm{d}\Omega\mathrm{d}\nu \\
&- \int_0^\infty\int_{S^2} \left(\frac{\boldsymbol{\beta}\cdot\boldsymbol{\Omega}}{c\theta_\mathrm{r}} - \frac{1}{\theta_\mathrm{r}}\right)\sigma_\nu^\mathrm{a} B(\nu,T_\mathrm{g})\,\mathrm{d}\Omega\mathrm{d}\nu \\
&- \int_0^\infty\int_{S^2} \left(\frac{\boldsymbol{\beta}\cdot\boldsymbol{\Omega}}{c\theta_\mathrm{r}} - \frac{1}{\theta_\mathrm{r}}\right)\sigma_\nu^\mathrm{s}\int_{S^2} p_\nu(\boldsymbol{\Omega}\cdot\boldsymbol{\Omega}') I(\boldsymbol{x},t,\boldsymbol{\Omega}',\nu)\,\mathrm{d}\Omega'\mathrm{d}\Omega\mathrm{d}\nu.
\end{aligned} \tag{1.81}$$

Because θ_r and $\boldsymbol{\beta}$ do not depend on $\boldsymbol{\Omega}$ and ν, we recognize computations done in section 1.A and

$$\begin{aligned}
\partial_t S_\mathrm{r} + \nabla\cdot(S_\mathrm{r}\boldsymbol{\beta}) &= -\frac{\boldsymbol{\beta}\cdot\boldsymbol{S}_\mathbf{F}}{c\theta_\mathrm{r}} + \frac{S_\mathrm{E}}{\theta_\mathrm{r}} \\
&= \frac{c\sigma^\mathrm{a}}{\theta_\mathrm{r}}\left(a_\mathrm{r}T_\mathrm{g}^4 - E_\mathrm{r}\right) + \frac{\sigma^\mathrm{s}}{c\theta_\mathrm{r}}\boldsymbol{\beta}\cdot\boldsymbol{F}_\mathbf{r}.
\end{aligned} \tag{1.82}$$

1.C Diffusive limit for radiation hydrodynamics

As in section 1.4, we consider the diffusive limit with the P_1 closure relation, but with hydrodynamics. We introduce a rescaling parameter ε to write the time (resp. the opacity) as $\tilde{t} = \varepsilon t$ (resp. $\tilde{\sigma^{\mathrm{a}}} = \varepsilon\sigma^{\mathrm{a}}$ and $\tilde{\sigma^{\mathrm{s}}} = \varepsilon\sigma^{\mathrm{s}}$). Because the velocity of the fluid is smaller than the speed of light ($\frac{u}{c} \ll 1$), we also rescale it as $\tilde{\boldsymbol{u}} = \frac{\boldsymbol{u}}{\varepsilon}$. Let us focus on the equations describing the evolution of the radiative variables:

$$\begin{cases} \varepsilon^2\partial_{\tilde{t}}E_{\mathrm{r}} + \varepsilon\nabla\cdot\boldsymbol{F_{\mathrm{r}}} = c\tilde{\sigma^{\mathrm{a}}}\left(a_{\mathrm{r}}T_{\mathrm{g}}^4 - E_{\mathrm{r}}\right) + \varepsilon\dfrac{2\tilde{\sigma^{\mathrm{a}}} - \tilde{\sigma^{\mathrm{s}}}}{c}\tilde{\boldsymbol{u}}\cdot\boldsymbol{F_{\mathrm{r}}} + \varepsilon^2\dfrac{4}{3}\dfrac{\tilde{\sigma^{\mathrm{s}}}}{c}E_{\mathrm{r}}\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}} & \text{(1.83a)} \\ \varepsilon^2\partial_{\tilde{t}}\boldsymbol{F_{\mathrm{r}}} + \varepsilon\dfrac{c^2}{3}\nabla E_{\mathrm{r}} = -\tilde{\sigma^{\mathrm{s}}}c\boldsymbol{F_{\mathrm{r}}} + \varepsilon c\left(\dfrac{4}{3}\tilde{\sigma^{\mathrm{s}}} - \tilde{\sigma^{\mathrm{a}}}\right)\tilde{\boldsymbol{u}}E_{\mathrm{r}} \\ \qquad\qquad + \varepsilon c\tilde{\sigma^{\mathrm{a}}}\tilde{\boldsymbol{u}}a_{\mathrm{r}}T_{\mathrm{g}}^4 + \varepsilon^2\dfrac{2\tilde{\sigma^{\mathrm{a}}}}{c}\tilde{\boldsymbol{u}}\cdot(\tilde{\boldsymbol{u}}\cdot\boldsymbol{F_{\mathrm{r}}}). & \text{(1.83b)} \end{cases}$$

By expanding eqs. 1.83a and 1.83b at order 0, we have

$$\begin{cases} E_{\mathrm{r},0} = a_{\mathrm{r}}T_{\mathrm{g},0}^4 \\ \boldsymbol{F_{\mathrm{r},0}} = \boldsymbol{0}. \end{cases} \tag{1.84}$$

Expanding eq. 1.83b at order 1 leads to

$$\boldsymbol{F_{\mathrm{r},1}} = -\frac{c}{3\tilde{\sigma^{\mathrm{s}}}}\nabla E_{\mathrm{r},0} + \frac{4}{3}E_{\mathrm{r},0}\widetilde{\boldsymbol{u}_0}. \tag{1.85}$$

Finally, looking at the radiative energy and the gas internal energy at order 2, source terms cancel each other, only the divergence of the radiative flux at order 1 remains, and we have

$$\partial_{\tilde{t}}\left(\rho c_v T_{\mathrm{g},0} + E_{\mathrm{r},0}\right) - \nabla\cdot\left(\frac{c}{3\tilde{\sigma^{\mathrm{s}}}}\nabla E_{\mathrm{r},0}\right) = -\frac{4}{3}\nabla\left(E_{\mathrm{r},0}\widetilde{\boldsymbol{u}_0}\right). \tag{1.86}$$

One can also look at eq. 1.83a at order 2. Expanding eq. 1.83a at order 2 gives

$$\begin{aligned} \partial_{\tilde{t}}E_{\mathrm{r},0} - \nabla\left(\frac{c}{3\tilde{\sigma^{\mathrm{s}}}}\nabla E_{\mathrm{r},0}\right) = c\tilde{\sigma^{\mathrm{a}}}\left(6a_{\mathrm{r}}T_{\mathrm{g},0}^2T_{\mathrm{g},1}^2 + 4a_{\mathrm{r}}T_{\mathrm{g},0}^3T_{\mathrm{g},2} - E_{\mathrm{r},0}\right) - \frac{4}{3}\nabla\cdot\left(E_{\mathrm{r},0}\widetilde{\boldsymbol{u}_0}\right) \\ - \frac{2\sigma^{\mathrm{a}} - \sigma^{\mathrm{s}}}{3\sigma^{\mathrm{s}}}\widetilde{\boldsymbol{u}_0}\cdot\nabla E_{\mathrm{r},0} + 2\frac{4}{3}\frac{\sigma^{\mathrm{a}}}{c}E_{\mathrm{r},0}\widetilde{\boldsymbol{u}_0}^2. \end{aligned} \tag{1.87}$$

We recover Eq. 43 of Krumholz et al. [2007]. The term $c\tilde{\sigma^{\mathrm{a}}}\left(6a_{\mathrm{r}}T_{\mathrm{g},0}^2T_{\mathrm{g},1}^2 + 4a_{\mathrm{r}}T_{\mathrm{g},0}^3T_{\mathrm{g},2} - E_{\mathrm{r},2}\right)$ is the development at second order of the term $\kappa_0(4\pi B - cE)$. Because we do not neglect any terms $\mathcal{O}\left(\frac{\tilde{\boldsymbol{u}}}{c}\right)$, some coefficients are slightly different. Krumholz et al. [2007] discuss the importance of the term $\frac{4}{3}\frac{\tilde{\sigma^{\mathrm{a}}}}{c}E_{\mathrm{r},0}\widetilde{\boldsymbol{u}_0}^2$. It is interpreted as a "relativistic work", as opposed to the term of "diffusion work", $\widetilde{\boldsymbol{u}_0}\cdot\nabla E_{\mathrm{r},0}$. Therefore, it is important for non-equilibrium non-uniform dynamic diffusion problem, whereas it is not important for static diffusion problems.

Chapter 2

A first asymptotic preserving solver

Contents

This chapter is the adaptation of an article published in Astronomy&Astrophysics, see Bloch et al. 2021.

Even though the M_1 model is accurate in both free-streaming and diffusive regimes at the continuous level, numerical schemes also need to properly capture both limits. Several approaches have been developed. For example, Berthon and Turpault [2011] presented a scheme based on an HLL solver with source terms modified with a free parameter. Following this idea, we propose a new so-called asymptotic preserving scheme, also based on an HLL solver. Nevertheless, to obtain a simpler solver, we have chosen another parameter to recover the asymptotic behavior, in the diffusive limit. Furthermore, our integration of source terms is different. In many physical applications (e.g., clouds), optically thick regions are found next to optically thin zones. We propose a well-balanced modification of the source term, which allows us to accurately reach steady states in the presence of sharp transitions.

This chapter is organized as follows. We go through our new numerical scheme, well-balanced and asymptotic preserving in the diffusive limit in section 2.1. In section 2.2, we present some numerical test cases to show the importance of the asymptotic preserving and well-balanced properties. We also present a physical application about the stability of the ionization front in an H II region in a massive pre-stellar dense core. Finally, we reach our conclusion and discuss the limitations of the scheme in section 2.3.

2.1 Numerical scheme and algorithm

2.1.1 Radiation transport in a static fluid

Let us first introduce some notations: we note Δx the step along the x-direction. Δt is the time interval between the current time t^n and t^{n+1}. We write x_i the center of the cell i and $x_{i+\frac{1}{2}}$ the interface between the cell i and the cell $i+1$. We use the notation u_i^n to represent the averaged quantity associated with the field u at time t^n in the cell i (finite volume). Finally, we note $u_{i+\frac{1}{2}}^n$ to represent the quantity associated with the field u at time t^n and at the interface between cells i and $i+1$.

The development of the numerical scheme is presented only in the one-dimensional case, but its extension to higher dimensions is straightforward. To ease notations, we drop the indices r for all radiative variables.

The time step given by the CFL condition is much smaller for radiation than for hydrodynamics. Indeed, for the radiation, it is limited by the speed of light, whereas it is limited by the speed of sound of the fluid for the hydrodynamics. Because we are interested in radiation hydrodynamics, we will consider a long timescale for the radiative transfer. Therefore, we use a time-implicit integration for the radiative transfer. A similar development can be done with a semi-implicit solver: source terms remain implicit, but the hyperbolic part is time-explicit.

2.1.1.1 Hyperbolic system

Following González et al. [2007], we discretize the hyperbolic part of eq. 1.15 using a first-order Godunov type solver [Toro, 2009]. From Berthon and Turpault [2011], we

also introduce an extra parameter α which will be specified in section 2.1.1.3:

$$\begin{cases} E_i^{n+1} = E_i^n - \dfrac{\Delta t}{\Delta x}\left(\alpha_{i+\frac{1}{2}}\mathcal{F}^*_{i+\frac{1}{2}} - \alpha_{i-\frac{1}{2}}\mathcal{F}^*_{i-\frac{1}{2}}\right) + c\sigma_i^{\mathrm{a}}\Delta t\left(a_{\mathrm{r}}\left(T_i^{n+1}\right)^4 - E_i^{n+1}\right) \\ F_i^{n+1} = F_i^n - \dfrac{\Delta t}{\Delta x}\left(\mathcal{P}^*_{i+\frac{1}{2}} - \mathcal{P}^*_{i-\frac{1}{2}}\right) - c\Delta t\{\sigma^{\mathrm{s}}F_{\mathrm{r}}\}_i^{n+1} \\ \rho c_v T_i^{n+1} = \rho c_v T_i^n - c\sigma_i^{\mathrm{a}}\Delta t\left(a_{\mathrm{r}}\left(T_i^{n+1}\right)^4 - E_i^{n+1}\right), \end{cases} \tag{2.1}$$

where $\mathcal{F}^*_{i+\frac{1}{2}}$ and $\mathcal{P}^*_{i+\frac{1}{2}}$ are the numerical fluxes given by

$$\begin{aligned} \mathcal{F}^*_{i+\frac{1}{2}} &= \frac{\lambda^+_{i+\frac{1}{2}}F_i^{n+1} - \lambda^-_{i+\frac{1}{2}}F_{i+1}^{n+1}}{\lambda^+_{i+\frac{1}{2}} - \lambda^-_{i+\frac{1}{2}}} + \frac{\lambda^+_{i+\frac{1}{2}}\lambda^-_{i+\frac{1}{2}}}{\lambda^+_{i+\frac{1}{2}} - \lambda^-_{i+\frac{1}{2}}}\left(E_{i+1}^{n+1} - E_i^{n+1}\right) \\ \mathcal{P}^*_{i+\frac{1}{2}} &= c^2\frac{\lambda^+_{i+\frac{1}{2}}P_i^{n+1} - \lambda^-_{i+\frac{1}{2}}P_{i+1}^{n+1}}{\lambda^+_{i+\frac{1}{2}} - \lambda^-_{i+\frac{1}{2}}} + \frac{\lambda^+_{i+\frac{1}{2}}\lambda^-_{i+\frac{1}{2}}}{\lambda^+_{i+\frac{1}{2}} - \lambda^-_{i+\frac{1}{2}}}\left(F_{i+1}^{n+1} - F_i^{n+1}\right), \end{aligned} \tag{2.2}$$

with $\lambda^+_{i+\frac{1}{2}} = \max(0, \lambda_{max})$ and $\lambda^-_{i+\frac{1}{2}} = \min(0, \lambda_{min})$, where λ_{max} and λ_{min} are the eigenvalues of eq. 1.15. From Berthon et al. [2007], we have

$$\lambda_{max,min} = c\left(\frac{f_x}{\xi} \pm \frac{\sqrt{2}\sqrt{(\xi-1)(\xi+2)(2(\xi-1)(\xi+2)+3f_y^2)}}{\sqrt{3}\xi(\xi+2)}\right), \tag{2.3}$$

with $\xi = \sqrt{4-3f^2}$. See Fig. 1 of González et al. [2007] for more details about the structure of the eigenvalues. $\{\sigma^{\mathrm{s}}F_{\mathrm{r}}\}_i^{n+1}$ is a well-chosen discretization of the term $\sigma^{\mathrm{s}}\boldsymbol{F}_{\mathbf{r}}$ in the cell i and at time t^{n+1}, and is specified in the next section.

2.1.1.2 Well-balanced modification of the source term

From Berthon et al. [2015], a well-balanced scheme catches the correct steady regime. The steady state, if it exists, is given by

$$\begin{cases} E_{\mathrm{r}} = a_{\mathrm{r}}T_{\mathrm{g}}^4 & \text{(2.4a)} \\ \nabla\cdot\boldsymbol{F}_{\mathbf{r}} = \mathbf{0} & \text{(2.4b)} \\ c\nabla\cdot\mathbb{P}_{\mathrm{r}} = -\sigma^{\mathrm{s}}\boldsymbol{F}_{\mathbf{r}}. & \text{(2.4c)} \end{cases}$$

Equation 2.4c is discretized by $\frac{c}{2}\left((\nabla\cdot P)_{i+\frac{1}{2}}^{n+1} - (\nabla\cdot P)_{i-\frac{1}{2}}^{n+1}\right) = -\{\sigma^{\mathrm{s}}F\}_i^{n+1}$, with $(\nabla\cdot P)_{i+\frac{1}{2}}^{n+1} = \frac{P_{i+1}^{n+1}-P_i^{n+1}}{\Delta x}$. An obvious choice for $\{\sigma^{\mathrm{s}}F\}_i^n$ is

$$\{\sigma^{\mathrm{s}}F\}_i^{n+1} = \sigma_i^{\mathrm{s}}F_i^{n+1}. \tag{2.5}$$

However, using this formulation, eq. 2.4c is discretized as

$$-\frac{c}{2}\left((\nabla\cdot P)_{i+\frac{1}{2}}^{n+1} + (\nabla\cdot P)_{i-\frac{1}{2}}^{n+1}\right) = \sigma_i^{\mathrm{s}}F_i^{n+1}. \tag{2.6}$$

The radiative flux remains cell-centered and is equal to the divergence of radiative pressure, defined at the interfaces of the cells. This can create some spurious flux at the interface when looking for a steady state with a constant flux in the box (see section 2.2.2). Inspired by well-balanced schemes for hydrodynamics (e.g., Padioleau

et al. 2019) which preserve the hydrostatic balance between the pressure forces and the gravitational force (and the similarity of this balance with the balance between radiative pressure and radiative flux source term in eq. 2.4c), we choose to use an average of a face discretization of the radiative flux source term:

$$\{\sigma^{\mathrm{s}}F\}_i^{n+1} = \frac{1}{2}\left(\sigma^{\mathrm{s}}_{i+\frac{1}{2}}F^{n+1}_{i+\frac{1}{2}} + \sigma^{\mathrm{s}}_{i-\frac{1}{2}}F^{n+1}_{i-\frac{1}{2}}\right), \tag{2.7}$$

with

$$\begin{cases} \sigma^{\mathrm{s}}_{i+\frac{1}{2}} = \dfrac{1}{2}\left(\sigma^{\mathrm{s}}_i + \sigma^{\mathrm{s}}_{i+1}\right) \\ F^{n+1}_{i+\frac{1}{2}} = \dfrac{1}{2}\left(F^{n+1}_i + F^{n+1}_{i+1}\right). \end{cases} \tag{2.8}$$

One way to interpret this equation is to remember that

$$\{\sigma^{\mathrm{s}}F\}_i^{n+1} = \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \sigma^{\mathrm{s}}(x)\boldsymbol{F}\left(t^{n+1}, x\right)\mathrm{d}x. \tag{2.9}$$

Equation 2.5 is obtained with the rectangle rule for numerical integration of eq. 2.9:

$$\begin{aligned} \{\sigma^{\mathrm{s}}F\}_i^{n+1} &= \frac{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}}{\Delta x}\left(\sigma^{\mathrm{s}}\left(\frac{x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}}{2}\right)\boldsymbol{F}\left(t^{n+1}, \frac{x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}}{2}\right)\right) \\ &= \sigma^{\mathrm{s}}(x_i)\boldsymbol{F}(t^{n+1}, x_i) \\ &= \sigma^{\mathrm{s}}_i F^{n+1}_i, \end{aligned} \tag{2.10}$$

whereas eq. 2.7 is given by the trapezoidal rule:

$$\begin{aligned} \{\sigma^{\mathrm{s}}F\}_i^{n+1} &= \frac{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}}{2\Delta x}\left(\sigma^{\mathrm{s}}\left(x_{i-\frac{1}{2}}\right)\boldsymbol{F}\left(t^{n+1}, x_{i-\frac{1}{2}}\right) + \sigma^{\mathrm{s}}\left(x_{i+\frac{1}{2}}\right)\boldsymbol{F}\left(t^{n+1}, x_{i+\frac{1}{2}}\right)\right) \\ &= \frac{1}{2}\left(\sigma^{\mathrm{s}}_{i-\frac{1}{2}}F^{n+1}_{i-\frac{1}{2}} + \sigma^{\mathrm{s}}i + \frac{1}{2}F^{n+1}_{i+\frac{1}{2}}\right). \end{aligned} \tag{2.11}$$

To have

$$\sigma^{\mathrm{s}}_{i+\frac{1}{2}}F^{n+1}_{i+\frac{1}{2}} = -c\frac{P^{n+1}_{i+1} - P^{n+1}_i}{\Delta x} \tag{2.12}$$

in the whole domain, we also impose it as boundary condition:

$$\sigma^{\mathrm{s}}_{\frac{1}{2}}F^{n+1}_{\frac{1}{2}} = -c\frac{P^{n+1}_1 - P^{n+1}_0}{\Delta x}, \tag{2.13}$$

where P^{n+1}_0 is the radiative pressure given by the boundary condition. In that way, the radiative flux is centered at the interfaces of the cells, as well as the divergence of radiative pressure. A von Neumann stability analysis of the modified scheme is done in section 2.A, which shows that this discretization for the source terms is unconditionally stable.

2.1.1.3 Asymptotic preserving scheme

Now that the choice for $\{\sigma^{\mathrm{s}}F\}_i^{n+1}$ is determined, we still have to specify our choice for $\alpha_{i+\frac{1}{2}}$ in eq. 2.1. $\alpha_{i+\frac{1}{2}} = 1$ corresponds to a classic HLL scheme. However, the solution given by an asymptotic preserving scheme has to approximate the solution of eq. 1.24 as soon as the asymptotic regime is reached, i.e., large opacity and long

timescale. Unfortunately, a standard HLL scheme does not have this property (see section 2.2.1). To tackle this issue and get an asymptotic preserving scheme, we choose

$$\alpha_{i+\frac{1}{2}} = \frac{1}{1 - 3\sigma^{\mathrm{s}}_{i+\frac{1}{2}}\Delta x\left(1 - f^2_{i+\frac{1}{2}}\right)\dfrac{\lambda^+_{i+\frac{1}{2}}\lambda^-_{i+\frac{1}{2}}}{c\left(\lambda^+_{i+\frac{1}{2}} - \lambda^-_{i+\frac{1}{2}}\right)}}, \tag{2.14}$$

with $f_{i+\frac{1}{2}} = \frac{1}{2}\left(f^n_i + f^n_{i+1}\right)$. The derivation of eq. 2.14 is done in section 2.B. Other choices can be done, which leads to other schemes, with other properties. For example, the choice done by Berthon and Turpault [2011] leads to a solver more difficult to write, but that benefits from the known properties of an approximate Riemann solver. If $\sigma_{i+\frac{1}{2}}\Delta x$ goes to 0, $\alpha_{i+\frac{1}{2}}$ goes to 1, and we recover a standard HLL scheme. Considering the diffusive limit, we prove that the scheme is asymptotic preserving in section 2.B. We show that

$$\begin{cases} E^{n+1}_{i,0} = a_{\mathrm{r}}\left(T^{n+1}_{i,0}\right)^4 \\ F^{n+1}_{i,0} = 0 \\ \sigma^{\mathrm{s}}_{i+\frac{1}{2}}F^{n+1}_{i+\frac{1}{2},1} = -\dfrac{c}{3\Delta x}\left(E^{n+1}_{i+1,0} - E^{n+1}_{i,0}\right) \\ E^{n+1}_{i,0} + \rho c_v T^{n+1}_{i,0} = E^n_{i,0} + \rho c_v T^n_{i,0} + \dfrac{c\Delta t}{3\Delta x^2}\left(\dfrac{E^{n+1}_{i+1,0} - E^{n+1}_{i,0}}{\sigma^{\mathrm{s}}_{i+\frac{1}{2}}} - \dfrac{E^{n+1}_{i,0} - E^{n+1}_{i-1,0}}{\sigma^{\mathrm{s}}_{i-\frac{1}{2}}}\right), \end{cases} \tag{2.15}$$

which is a standard discretization of eqs. 1.22 to 1.24.

Unfortunately, we cannot prove that this scheme will preserve the admissible states $f < 1$ and, indeed, numerical experiments with this scheme have shown that we can get $f > 1$ when we are close to the free-streaming regime. In these situations, depending on the test case, we can either enforce $f < 1$ (section 2.2.6) or come back to a centered discretization of the source term (section 2.2.4). Furthermore, the development of the asymptotic preserving scheme with the well-balanced modification of the source term is only done in the case of a static fluid. Using the asymptotic correction eq. 2.14 is only the first step to have an asymptotic preserving scheme in the case of a moving fluid (see section 2.3.2).

2.1.2 Coupling to hydrodynamics

Following González et al. [2007], the resolution of the whole system 1.25 describing radiation hydrodynamics is split into three steps:

1. update of the hydrodynamics quantities (eqs. 1.25a to 1.25c without the terms of energy and momentum exchange) using the well-balanced and all-regime solver developed in Padioleau et al. [2019];
2. update of the radiative quantities and gas temperature (eqs. 1.15 and 1.16) using the solver developed in section 2.1.1. During this step, the hydrodynamics quantities are frozen;
3. addition of source terms $S_{E_{\mathrm{r}}}(\boldsymbol{u})$ and $\boldsymbol{S}_{\boldsymbol{F}_{\mathrm{r}}}(\boldsymbol{u})$. For simplicity, all source terms which depend on the velocity are treated explicitly. The term $\frac{\sigma^{\mathrm{s}}}{c}\boldsymbol{F}_{\mathrm{r}}$ in eqs. 1.25b and 1.25e is discretized using the well-balanced scheme proposed in section 2.1.1.2. All the other terms remain cell-centered.

This splitting allows reducing the number of equations solved implicitly, making the method more efficient.

2.1.3 Algorithm for nonlinear implicit solver

2.1.3.1 Newton-Raphson method and linear solver

Because of the Eddington tensor, the eigenvalues in the numerical fluxes, and the $a_r T_g^4$ factor, the system is nonlinear. It is solved using a Newton-Raphson method. At each iteration, we have to solve a linear system. Because the system is large ($(2+d)N$ unknowns, where d is the number of dimensions and N the total number of cells) and sparse, it cannot be solved using a direct method. Because of the numerical fluxes, the matrix is not symmetric, and we use the biconjugate gradient stabilized method [Van der Vorst, 1992].

2.1.3.2 Preconditioner

Using large time steps for the radiative transfer, the matrix is ill-conditioned and iterative methods might not converge. One way to deal with this issue is to use a preconditioner. Instead of solving the original linear system $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$, we solve the right preconditioned system $\mathbb{A}\mathbb{K}^{-1}\mathbb{K}\boldsymbol{x} = \boldsymbol{b}$ via solving $\mathbb{A}\mathbb{K}^{-1}\boldsymbol{y} = \boldsymbol{b}$ to compute $\boldsymbol{y}$ and then $\mathbb{K}\boldsymbol{x} = \boldsymbol{y}$. As long as the matrix $\mathbb{K}$ is invertible, this gives the same solution as the original system. If $\mathbb{K}$ is well-chosen, the condition number of the matrix $\mathbb{A}\mathbb{K}^{-1}$ is lower than $\mathbb{A}$'s one. Preconditioners used here are presented in section 5.1.

In the next section, we use several numerical tests to show that the scheme developed in section 2.1 is well suited for the study of radiation hydrodynamics problems.

2.2 Numerical results

The implementation of the scheme developed in section 2.1 has been done in the code ARK-RT [1], a fork of the code ARK developed in Padioleau et al. [2019]. The hydrodynamics and gravity part of the solver is similar to ARK and is solved with a well-balanced and all-regime solver. More details about ARK-RT can be found in chapter 5.

We performed a series of verification tests to validate different properties of the scheme: the asymptotic correction with a Marshak wave, the well-balanced property to reach a steady state with a jump of opacity, the properties of the M_1 model with a beam test and a shadow test, and the coupling to the hydrodynamics with radiative shocks. We also present a physical application about the stability of the ionization front in an H II region in a massive pre-stellar dense core. To ease notations, we define the radiative temperature as $T_\mathrm{r} = \left(\frac{E_\mathrm{r}}{a_\mathrm{r}}\right)^{\frac{1}{4}}$.

2.2.1 Marshak wave

From Mihalas and Mihalas [1984], a Marshak wave is the propagation of hot radiation into a cold medium. We consider a one-dimensional case in the diffusive limit, to test the asymptotic preserving scheme developed in section 2.1.1.3.

The length of the computational domain is 1 cm; it is discretized with 400 points. Initially, the medium is at equilibrium with the radiation: $T_0 = T_\mathrm{r} = 300$ K, the initial radiative flux is $\boldsymbol{F}_\mathbf{r} = \mathbf{0}$. We consider a perfect gas with $\gamma = \frac{5}{3}$. The hydrodynamics is frozen. The density is constant, such that $\rho c_v = 1$ J K^{-1} cm^{-3}, the opacity is also

1. https://gitlab.erc-atmo.eu/erc-atmo/ark-rt/tree/v1.0.0

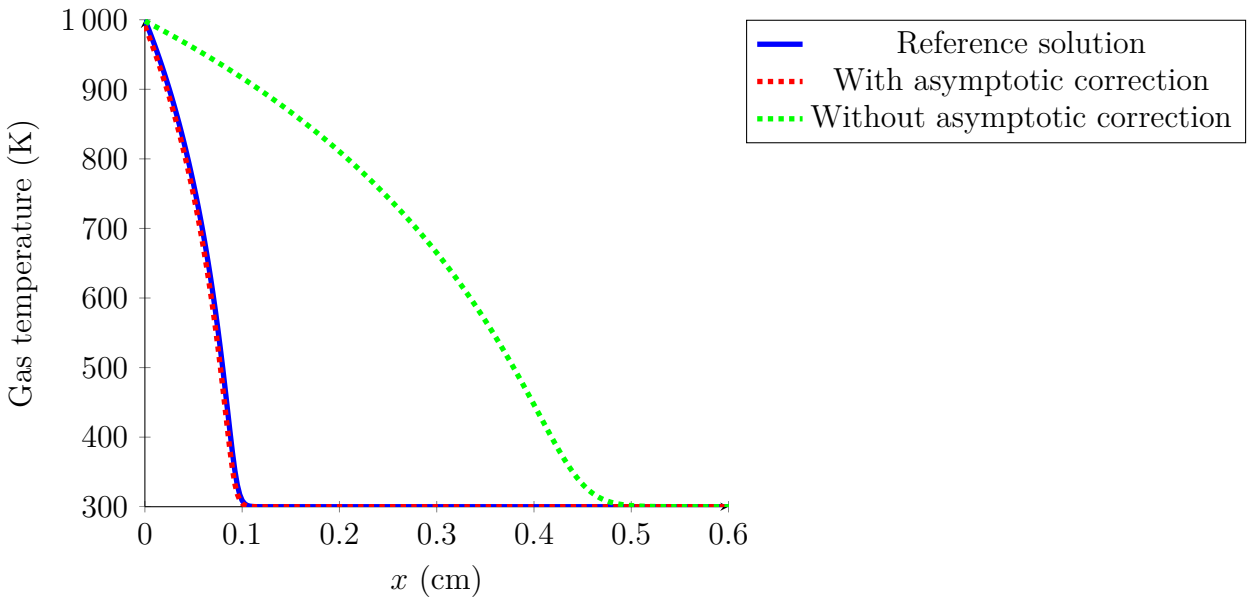


Figure 2.1 – Marshak wave simulation. This figure shows a snapshot of the gas temperature at time $t_f = 2 \times 10^{-4}$ s, with and without the asymptotic correction and the reference solution. Spatial resolution is 400 points and the opacity is $\sigma = 10\ 000\ \text{cm}^{-1}$.

constant, with $\sigma^{\text{a}} = \sigma^{\text{s}} = \sigma = 10\ 000\ \text{cm}^{-1}$, therefore $\sigma\Delta x = 25$. At time $t = 0$, a source is lit at the left boundary with $T_{\text{r}} = 1\ 000$ K.

The results are shown in fig. 2.1 at time $t_f = 2 \times 10^{-4}$ s. We compare different solutions: a reference solution, the solution given by our asymptotic preserving scheme, and the solution given by a standard scheme. The reference solution is given by a standard discretization of eq. 1.24. The relative L^2 error between the reference solution and the solution with $\alpha_{i+\frac{1}{2}}$ given by eq. 2.14 is 1.1%, whereas with the standard HLL scheme the relative L^2 error is 84%. Using the asymptotic correction, we recover the correct behavior in the asymptotic limit.

2.2.2 Steady state with a jump of opacity

In the previous case, the opacity is constant, we now consider a test with a jump of opacity, still in the one-dimensional case. We use this test to highlight the need for the well-balanced modification of the source term.

The length of the computational domain is 1 cm; it is discretized with 100 points. Initially, the medium is at equilibrium with the radiation: $T_0 = T_{\text{r}} = 300$ K, the initial radiative flux is $\boldsymbol{F}_{\mathbf{r}} = \mathbf{0}$. The opacity $\sigma^{\text{a}} = \sigma^{\text{s}} = \sigma$ is now a function of space:

$$\sigma(x) = \begin{cases} 10\ 000\ \text{cm}^{-1} \text{ if } & x < 0.5, \\ 0 \text{ if } & x > 0.5. \end{cases} \tag{2.16}$$

At time $t = 0$, a source is lit at the left boundary with $T_{\text{r}} = 1\ 000$ K.

Figure 2.2 shows the radiative flux at time $t_f = 10^{-3}$ s. From eq. 2.4b, when the steady state is reached, we expect the radiative flux to be constant in the box. Using a standard discretization of the source term, such as eq. 2.5, a spurious peak located at the discontinuity of opacity is observed (orange curve). The value taken by the radiative flux is more than 20 times the expected value. This seems to be

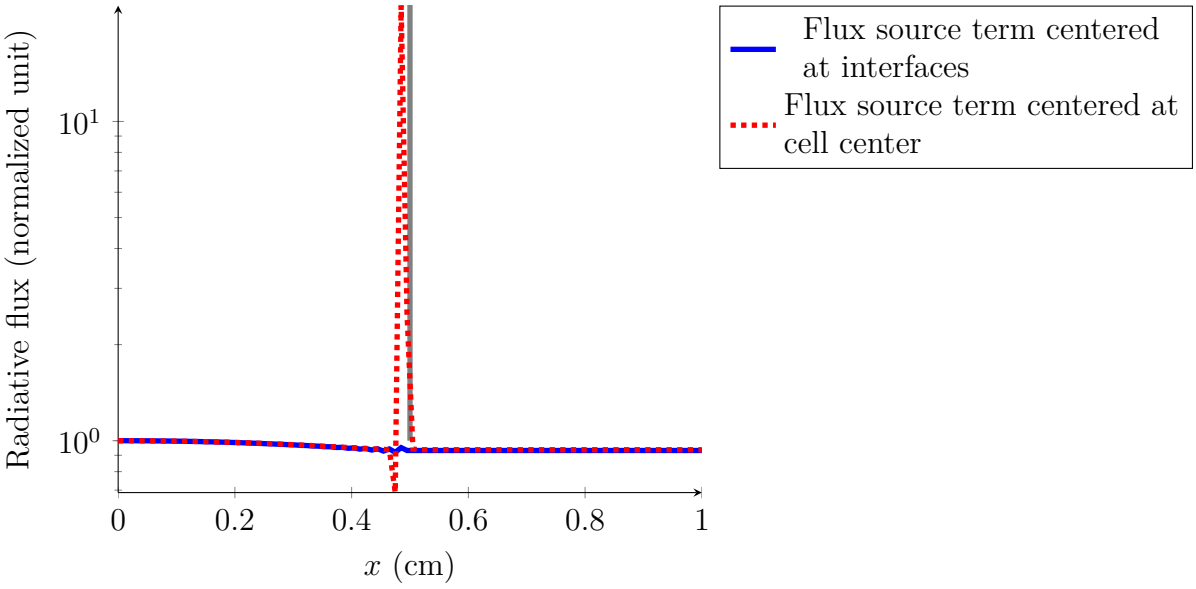


Figure 2.2 – Simulation of a steady state with a jump of opacity. The opacity is piecewise constant, a jump is located at $x = 0.5$ cm (gray line). This figure shows a snapshot of the radiative flux at time $t_f = 10^{-3}$ s.

caused by a numerical instability. This can result in $f > 1$ during the iterations of our Newton-Raphson implicit scheme, which is not physically admissible. However, using the well-balanced modification of the source term proposed by eq. 2.7 (blue curve), the spurious peak does not appear anymore and the constant steady state is reached.

Using the standard discretization of the source term eq. 2.5, one can show that the numerical scheme is unconditionally stable, in that the error between the numerical solution and the exact solution goes to 0 as Δx and Δt go to 0. The spurious peak seems to be due to a lack of precision in the integration of the source term. Using eq. 2.7, the source term is defined at the interfaces of the cells and balances the divergence of radiative pressure, also defined at the interfaces.

2.2.3 Beam

We now perform the same two-dimensional test as in González et al. [2007]; Richling et al. [2001]. The domain $[-1, 1] \times [-1, 1]$ is discretized with 128×128 cells. The initial temperature is $T_0 = T_\mathrm{r} = 300$ K, the initial radiative flux is $\boldsymbol{F}_\mathbf{r} = \mathbf{0}$, the opacity is $\sigma^\mathrm{a} = \sigma^\mathrm{s} = 0$. At time $t = 0$, a beam with $T_\mathrm{g} = T_\mathrm{r} = 1\,000$ K is introduced with an angle of 45°. The beam is located at $x = -1$ and $y \in [-0.875, -0.75]$. Because we are in the free-streaming regime, the propagation of the photons has to be followed, we cannot use large time steps. For performance reasons, we use the semi-implicit scheme.

Because there is no opacity, the beam propagates in the vacuum, and we expect it to cross the box without dispersion. The direction of the beam is not along the mesh axis; we use this test to quantify the numerical diffusion. Figure 2.3 shows the radiative energy at steady state. The eigenvalues in eq. 2.2 can either be fixed to $\pm c$ or computed using eq. 2.3. Because there is no opacity, the asymptotic correction nor the well-balanced source term affect the result, and we recover the same result as in González et al. [2007]. Figure 2.4 shows the horizontal cut at the middle height. The beam introduced at the boundary is sampled over 8 cells. Ideally, without any

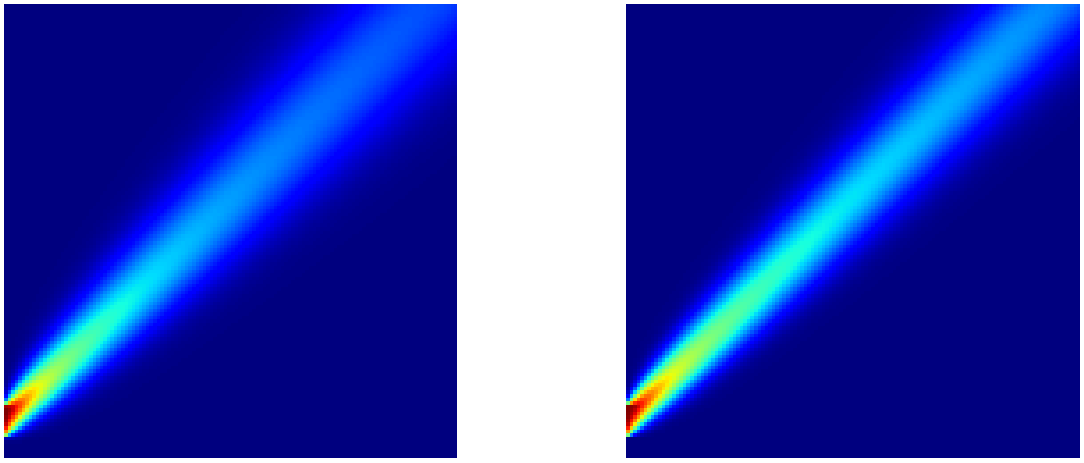


Figure 2.3 – Beam simulation. The figure shows the radiative energy. The eigenvalues are fixed to $\pm c$ (left panel) or calculated with eq. 2.3 (right panel).

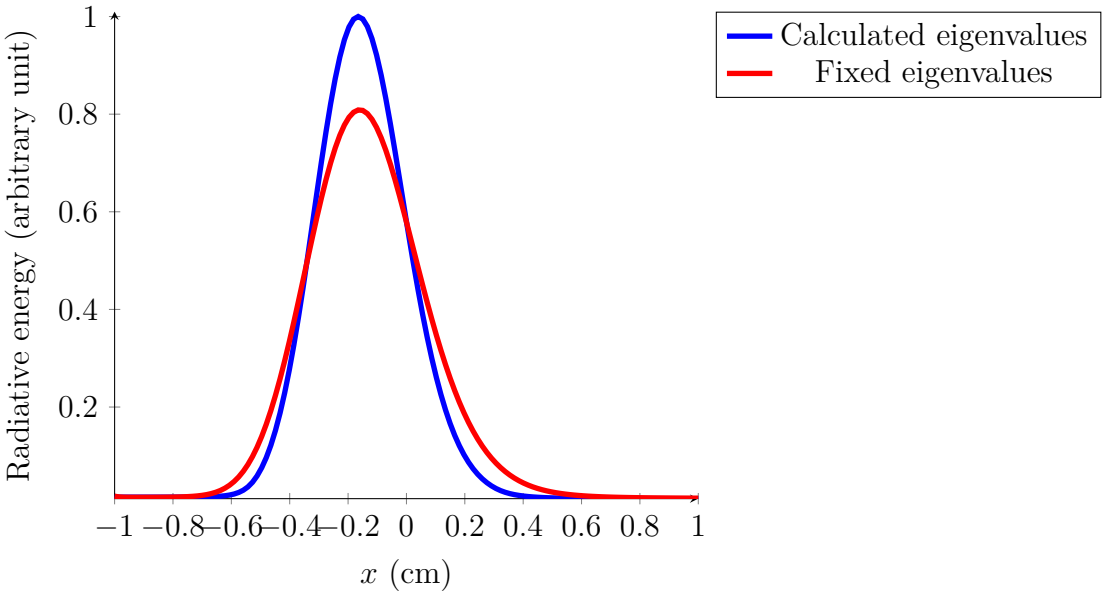


Figure 2.4 – Beam simulation. The figure shows a horizontal cut in fig. 2.3 at the middle height.

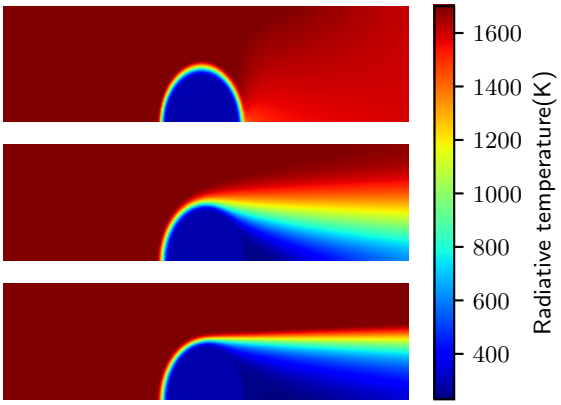


Figure 2.5 – Shadow simulation. Snapshots of the radiative temperature at time $t_f = 10^{-10}$ s with different closure relations: P_1 model (upper panel), M_1 model with fixed eigenvalues (middle panel) and M_1 model with computed eigenvalues (lower panel).

numerical diffusion, we would expect the width of the beam to stay exactly 8 cells. With the computed eigenvalues, we can keep the numerical diffusion under control. Using fixed eigenvalues, the width of the beam at middle height is approximately 30 cells, whereas it is only 24 cells with calculated eigenvalues.

The main difference in this test between ARK-RT and HERACLES [González et al., 2007] is the computation of the eigenvalues. We use the exact eigenvalues given by eq. 2.3 from Berthon et al. [2007], whereas in González et al. [2007], to save computational time, the eigenvalues are computed once at the beginning of the simulation and then interpolated. However, this approximation does not impact the result.

2.2.4 Shadow

Let us now consider a two-dimensional test with source terms. Following González et al. [2007]; Hayes and Norman [2003], we consider a shadow test. The computational domain is a cylinder of length $L = 1$ cm and radius $R = 0.12$ cm. It is discretized with 280×80 cells. A spheroid clump is located at the center of the box, on the symmetric axis: $(z_c, r_c) = (0.5, 0)$. The extension of the clump is $(z_0, r_0) = (0.1, 0.06)$. Initially, the medium is at equilibrium with the radiation, with $T_0 = T_\mathrm{r} = 290$ K. We consider a homogeneous gas, with $\rho_0 = 1$ g cm^{-3}, except for the clump with density $\rho_1 = 100\rho_0$. The boundary of the clump is smoothed: $\rho(r, z) = \rho_0 + \frac{\rho_1 - \rho_0}{1+\exp\Delta}$ with $\Delta = 10\left(\left(\frac{z-z_c}{z_0}\right)^2 + \left(\frac{r-r_c}{r_0}\right)^2 - 1\right)$. The opacity in the medium is $\sigma^\mathrm{a} = \sigma^\mathrm{s} = \sigma = \sigma_0 \left(\frac{T}{T_0}\right)^{-3.5} \left(\frac{\rho}{\rho_0}\right)^2$ with $\sigma_0 = 0.1$ cm^{-1}. At time $t = 0$, a source is lit at the left boundary with $T_\mathrm{r} = 1\,740$ K and the reduced flux is set to $f = 1$. Because f is close to 1 in the free-streaming regime, we encounter $f > 1$ in the simulation. To tackle this issue, we use the non-well-balanced scheme: the radiative flux source term is discretized using eq. 2.5. Because we are in the free-streaming regime, we cannot use large time steps. For performance reasons, we use the semi-implicit scheme. To recover the same result as in González et al. [2007], we use $\lambda^+_{i+\frac{1}{2}} = \max(0.1 \times c, \lambda_{max})$ and $\lambda^-_{i+\frac{1}{2}} = \min(-0.1 \times c, \lambda_{min})$, where λ_{max} and λ_{min} are given by eq. 2.3.

Figure 2.5 shows the radiative temperature at the final time $t_f = 10^{-10}$ s with different closure relations: the P_1 model, the M_1 model with fixed eigenvalues, and the

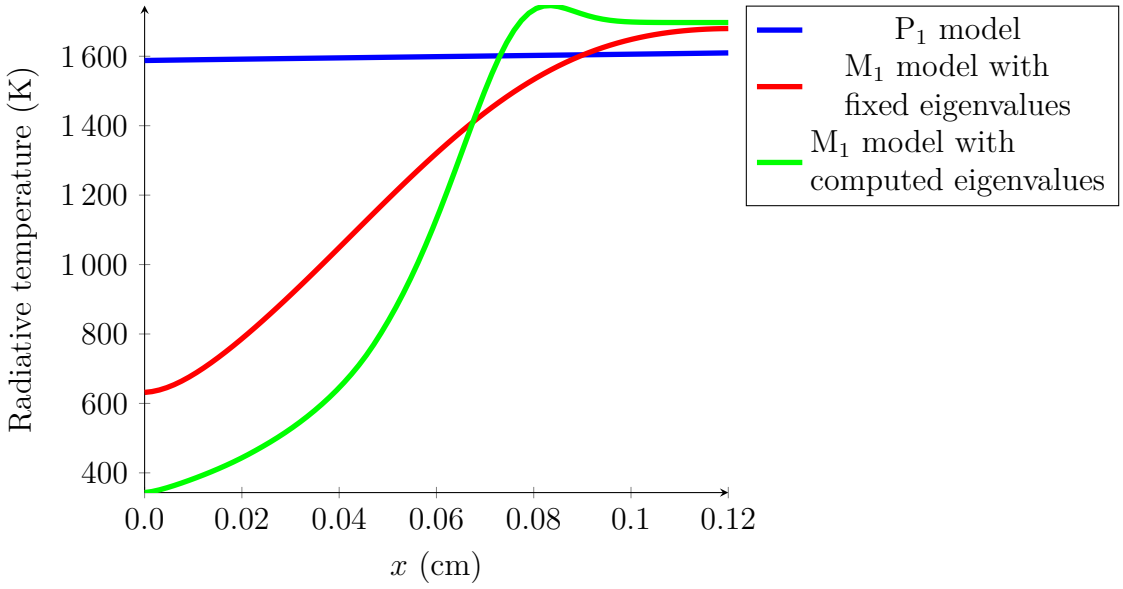


Figure 2.6 – Shadow simulation. The figure shows the radial profiles of the radiative temperature at time $t_f = 10^{-10}$ s with different closure relations: P_1 model, M_1 model with fixed eigenvalues and M_1 model with computed eigenvalues.

M_1 model with computed eigenvalues. Because of the high opacity in the clump, the light does not cross it and we expect the shadow behind it to remain stable.

As in González et al. [2007]; Hayes and Norman [2003], we plot the radial profile of the radiative temperature at the right boundary (fig. 2.6). Using the P_1 model, the radiative pressure is isotropic, therefore the photons go around the obstacle immediately, heating the whole domain. Using the M_1 closure relation, the shadow is better preserved, the temperature behind the obstacle remains at its initial value, 290 K. Because we are not in the diffusion regime outside of the clump and the light has not crossed the obstacle, the asymptotic correction has no impact on the result. Because the boundary of the clump is smoothed, the transition between the optically thick and thin medium is less sharp than in section 2.2.2 and the well-balanced modification of the source term is not necessary.

2.2.5 Radiative shocks

Now that we have verified the properties of our scheme with the hydrodynamics frozen, we perform numerical tests to validate the coupling with hydrodynamics. We consider radiative shocks: the gas and the radiation exchange energy and momentum. Following González et al. [2007]; Hayes and Norman [2003]; Ensman [1994], we consider a one-dimensional homogeneous medium, with $\rho = 7.78 \times 10^{-10}$ g cm^{-3} and $\sigma^{\mathrm{a}} = \sigma^{\mathrm{s}} = \sigma = 3.1 \times 10^{-10}$ cm^{-1}. We consider a perfect gas with an adiabatic coefficient $\gamma = \frac{7}{5}$ and a mean molecular weight $\mu = 1$. The length of the domain is 7×10^{10} cm. It is discretized with 400 cells. The initial temperature at the left boundary is set to 10 K and is increased by 0.25 K per cell. Initially, the radiation is at equilibrium with the gas. The left boundary condition is reflective, the initial velocity of the fluid is set to u_0. According to the value of u_0, the shock will be subcritical or supercritical. See González et al. [2007] for more details. To compare our results with González et al. [2007]; Hayes and Norman [2003]; Ensman [1994], we plot the temperature as a function of $x_i = x - u_0 t$.

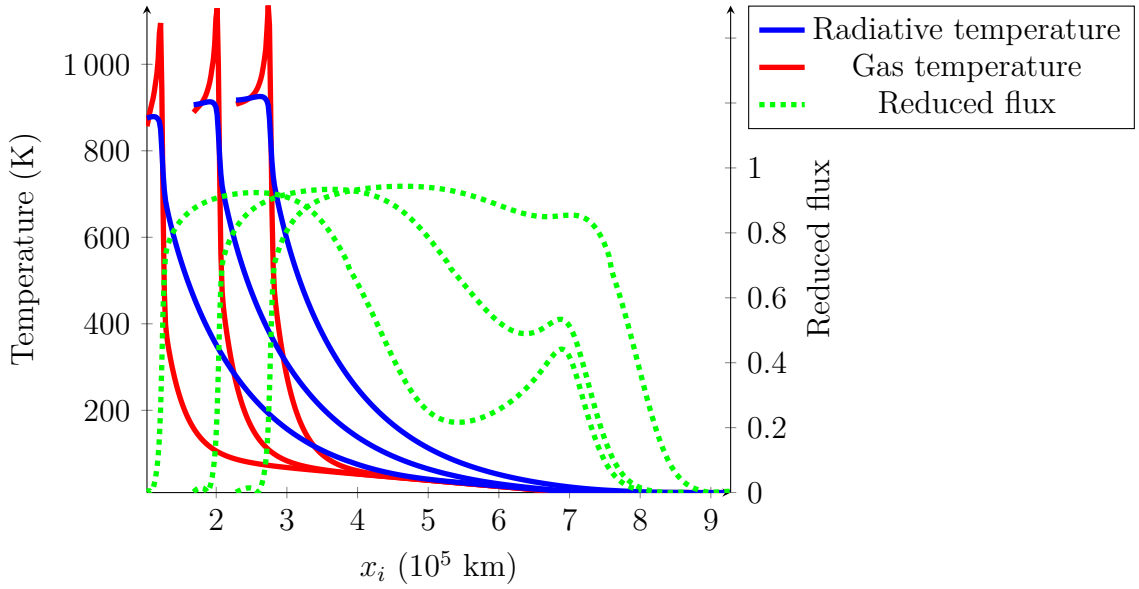


Figure 2.7 – Subcritical shock simulation. The figure shows snapshots of gas temperature, radiative temperature, and reduced flux at different times: 1.7×10^4 s, 2.8×10^4 s, and 3.8×10^4 s.

2.2.5.1 Subcritical shock

We first consider a subcritical shock, the initial velocity is set to $u_0 = -6$ km s^{-1}. Figure 2.7 shows the gas temperature, the radiative temperature, and the reduced flux at three different times: 1.7×10^4 s, 2.8×10^4 s, and 3.8×10^4 s. As expected, the gas and the radiation are not at equilibrium, before nor after the shock. The gas temperature reaches 1 135 K, as in González et al. [2007], whereas it is only 850 K in Ensman [1994].

2.2.5.2 Supercritical shock

We consider now a supercritical shock, where the initial velocity is set to $u_0 = -20$ km s^{-1}. Figure 2.8 shows the gas temperature, the radiative temperature, and the reduced flux at three different times: 4×10^3 s, 7.5×10^3 s, and 1.3×10^4 s. As in González et al. [2007], the radiative temperature is the same as the matter temperature on both sides of the shock. The gas and the radiation are therefore at equilibrium. The radiative precursor is larger than the subcritical shock's radiative precursor, as intended, and the temperature reaches 5 000 K, as in Ensman [1994]. We also recover the Zel'dovich spike.

We recover the expected results with both subcritical and supercritical shocks, therefore our code is well-suited to study radiation hydrodynamics problems.

2.2.6 Expansion of H II region

Now that we have confirmed the good behavior of the numerical scheme with both the asymptotic preserving and the well-balanced properties, we can apply it to a physical situation: the propagation of the ionization front in a massive pre-stellar dense core.

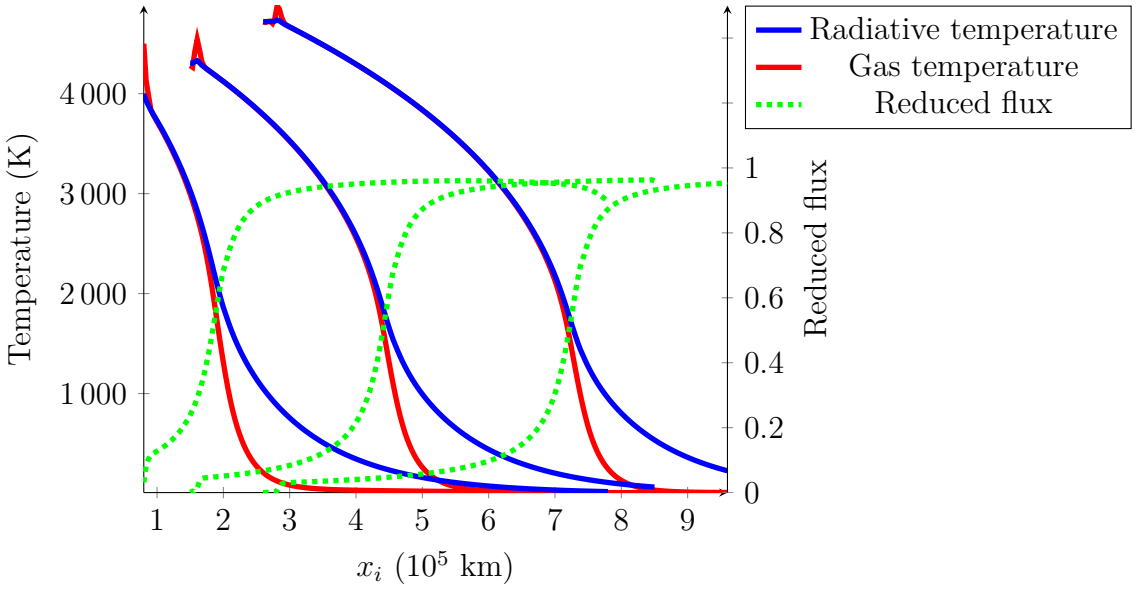


Figure 2.8 – Supercritical shock simulation. The figure shows snapshots of gas temperature, radiative temperature, and reduced flux at different times: 4×10^3 s, 7.5×10^3 s, and 1.3×10^4 s.

2.2.6.1 Model

We consider the early stage of the development of an H II region in a massive pre-stellar dense core [Churchwell, 2002]. We focus on a region of the dense core at about 100 AU from the massive young stellar object (YSO). This region has been heated by the YSO during the pre-main sequence phase, i.e., the early stage of the development of the massive star. The temperature reached at this location by infrared heating is of the order of 1 000 K and the transport of energy in this region can be dominated by convection. We have inferred the convective state of this region by computing thermal and adiabatic gradients based on observations of Herpin et al. 2009 (Fig. 7). High-energy photons emitted by the YSO when entering the main sequence will start to ionize the surrounding gas. This will trigger the propagation of an ionization front in a convective medium, and we are interested in the stability of such a front perturbed by the pre-existing convective motions.

The interaction of the ionizing photons with the gas is described by eq. 1.25. The only photons able to ionize the gas are emitted by the YSO, i.e., there is no local source of ionizing photons. Following Tremblin [2012], we need to modify this model to take into account photo-chemistry and thermal balance. We define the fraction of ionization $X = n_{H^+}/n_H$ where $n_H = n_{H^+} + n_{H^0}$, n_{H^+} is the number of ionized atoms and n_{H^0} is the number of cold atoms. The evolution of the number of ionized atoms is just the number of incoming photons that interact with the gas minus the number of ionized atoms that recombine (on the spot approximation, see Lesaffre 2002). Therefore,

$$\partial_t(\rho X) + \nabla \cdot (\rho X \boldsymbol{u}) = \sigma_\gamma F_\gamma n_H (1 - X) - m_H \beta X^2 n_H^2, \tag{2.17}$$

where F_γ is the number of incoming photons per unit of surface and time, σ_γ is the average cross-section at the temperature of the star, and β gives the recombination rate: $\beta = 2 \times 10^{-10} \left(\frac{T}{1\text{ K}}\right)^{-0.75}$ cm^3 s^{-1} with T the temperature of thermodynamic equilibrium [Black, 1981].

The thermal balance is the difference between the heating rate and the cooling rate. The extra energy of the absorbed photons is converted into kinetic energy of

electrons. It is the only source of heating during the ionization, hence the heating rate is given by $(1-X)n_H F_\gamma \sigma_\gamma e_\gamma$. In this simplified model, the equilibrium temperature is obtained from the balance between the heating from the ionization and the cooling from the recombination. We do not consider any other effects, such as metal cooling. Therefore, we take $e_\gamma = 1$ eV [Lesaffre, 2002] to recover the observed temperature around 1 000 K. From Tremblin [2012], the cooling rate is given by $\beta X^2 n_H^2 k_b T/(\gamma-1)$. We also add a term of Newtonian forcing $\partial_t T_g = \frac{T_g - T_{\text{forcing}}}{\tau_{\text{forcing}}}$ to trigger convection. T_{forcing} is the equilibrium temperature profile, depending on space, and τ_{forcing} is the relaxation timescale. The gas temperature will relax toward the equilibrium temperature profile T_{forcing}.

By writing $cE_\text{r} = F_\gamma e_\gamma$, $\rho = n_H m_H$ and $\sigma^\text{a} = \sigma^\text{s} = \sigma = \sigma_\gamma n_H$, we finally have to solve the following system:

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0 \\ \partial_t(\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u} + p\mathbb{I}) = \rho \boldsymbol{g} + \dfrac{\sigma(1-X)}{c} \boldsymbol{F}_\mathbf{r} \\ \partial_t(\rho E) + \nabla \cdot ((\rho E + p)\boldsymbol{u}) = \rho \boldsymbol{g} \cdot \boldsymbol{u} + c\sigma(1-X)E_\text{r} \\ \qquad\qquad - \beta \dfrac{\rho^2 X^2}{m_H^2} \dfrac{k_\text{b} T_\text{g}}{\gamma - 1} - \rho c_v \dfrac{T_\text{g} - T_\text{forcing}}{\tau_\text{forcing}} \\ \partial_t E_\text{r} + \nabla \cdot \boldsymbol{F}_\mathbf{r} = -c\sigma(1-X)E_\text{r} \\ \partial_t \boldsymbol{F}_\mathbf{r} + \nabla \cdot \mathbb{P}_\text{r} = -c\sigma(1-X)\boldsymbol{F}_\mathbf{r} \\ \partial_t(\rho X) + \nabla \cdot (\rho X \boldsymbol{u}) = \dfrac{\sigma(1-X)cE_\text{r} m_H}{e_\gamma} - \dfrac{\beta \rho^2 X^2}{m_H}. \end{cases} \tag{2.18}$$

In this test, we will use the M_1 solver with the asymptotic correction presented in section 2.1.1.3, but we do not use the well-balanced discretization of the source term because of stability issues that will be discussed in section 2.3.1.

2.2.6.2 Setup

We consider a square domain with a side 5 AU long. We use a setup similar to Padioleau et al. [2019] for compressible convection simulations. The temperature is set at the top and the bottom of the box at 500 K and 1 000 K, respectively. The initial temperature is a linear interpolation between the top and the bottom of the box. It is also the forcing temperature profile T_{forcing}. We take $\tau_{\text{forcing}} = 10^7$ s. These parameters are chosen to trigger the initial convective motions. We also set the pressure at the bottom: 10^{-3} dyn cm^{-2} [Herpin et al., 2009]. The density and the pressure are linked by the ideal gas law: $p = \frac{\rho k_b T_g}{m_H \mu}$, where μ is the mean molecular weight. The non-ionized medium is made of hydrogen, with $\mu_1 = 1$. When the medium is fully ionized, it is made of atomic nucleus and electrons, so twice as many particles for the same mass. Because the distribution of nucleus and electrons is homogeneous, the mean molecular weight is $\mu_2 = 0.5$. When the medium is partially ionized, we take μ as the mean of the previous values balanced by the fraction of ionization, i.e., $\mu = (1-X)\mu_1 + X\mu_2$. The density is initialized with the recursive formula $p_{i+1} - p_i = \frac{1}{2}(\rho_i + \rho_{i+1})\, g \Delta z$, which is the discrete version of the hydrostatic balance $\nabla p = -\rho \boldsymbol{g}$.

We impose Neumann boundary conditions for the temperature. The pressure and density are imposed by an extrapolation of the hydrostatic balance. Because the hydrodynamics solver is well-balanced for the gravity, this configuration will remain static,

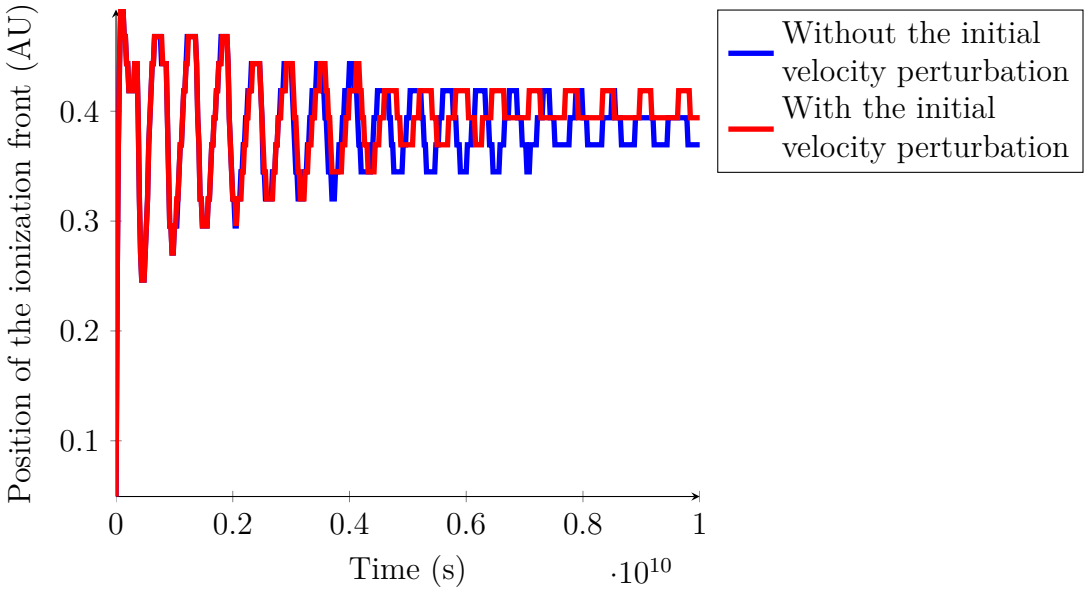


Figure 2.9 – Evolution of the position of the ionization front as a function of time, with and without the initial velocity perturbation.

even if the initial condition is unstable. The hydrostatic equilibrium is destabilized with a velocity mode perturbation of the form

$$\begin{cases} u(x,y) = 2 \cdot 10^{-4} c_s \sin\left(2\pi \dfrac{x - x_{\text{mid}}}{L_x}\right) \sin\left(\pi \dfrac{y - y_{\text{mid}}}{L_y}\right) \\ v(x,y) = 2 \cdot 10^{-4} c_s \cos\left(2\pi \dfrac{x - x_{\text{mid}}}{L_x}\right) \cos\left(\pi \dfrac{y - y_{\text{mid}}}{L_y}\right), \end{cases} \tag{2.19}$$

with c_s the speed of sound, $x_{\text{mid}} = y_{\text{mid}} = 2.5$ AU and $L_x = L_y = 5$ AU. Without any interaction with the ionizing photons, the convective motions are stationary.

The opacity is set to $\sigma = \frac{\sigma_\gamma \rho}{m_H}$ with $\sigma_\gamma = 6 \times 10^{-18}$ cm^2 [Lesaffre, 2002]. The radiative energy and flux are set to 0 and the medium is not ionized ($X = 0$). We initialize the hydrodynamics variables with the steady state described previously. At time $t = 0$, the bottom boundary of the region is ionized: the reduced flux is set to 1 and the radiative energy is set to $\frac{F_* e_\gamma}{c}$ with $F_* = 3 \times 10^{17}$ cm^{-2} s^{-1} in the boundary. The boundary conditions for the hydrodynamics variables remain unchanged.

2.2.6.3 Results

As the initial condition is such that E_{r} is close to 0, this can easily create some spurious values such that $f > 1$. This is clearly a numerical artifact induced by the very low value of the radiative energy in regions where no ionizing photons are present. Even with a centered discretization of the radiative flux source term and without the asymptotic correction, we still encounter $f > 1$ during the simulation. Because of this problem, we impose $f = 1$ in the computation of the Eddington tensor in the cells where $f > 1$.

Figure 2.9 shows the evolution of the position of the ionization front as a function of time. With and without the initial convective rolls, the position of the ionization front oscillates around an equilibrium position, between 0.3 AU and 0.4 AU. The oscillations around the equilibrium are expected and have been observed with simpler models [Tremblin et al., 2012].

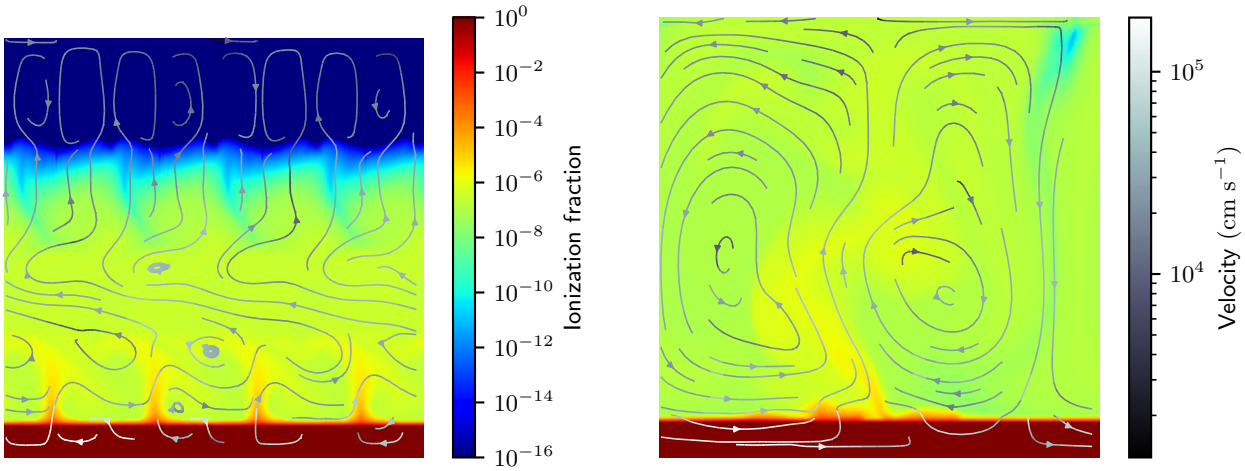


Figure 2.10 – Snapshots of the fraction of ionization and the velocity field at the final time $t_f = 10^{10}$ s without the initial velocity perturbation (left panel) and with it (right panel).

Figure 2.10 shows the ionization front at the final time $t_f = 10^{10}$ s. With and without the initial convective rolls, a numerical noise appears as a consequence of the long timescales. Because of the numerical noise, some lack of symmetry can appear. The fraction of ionization, which is always between 0 and 1, reaches values between 10^{-12} and 10^{-6}. The effect of the preconditioner and the MPI domain decomposition is discussed in section 5.2.3. However, the numerical noise does not affect the position of the ionization front.

The stability of the ionization front is an issue that has been discussed for a long time in the literature [Mizuta et al., 2005]. For example, 3D simulations of the expansion of a spherical ionization front in 3D Cartesian grids have shown instabilities either on the axis of the grid or in the diagonal depending on the numerical scheme (see Bisbas et al. 2015, fig. A3). The dependence of the instability on the grid cast doubts about a possible physical regime. Our test case shows that even with convective motions of large amplitude, the ionization front remains stable.

2.3 Discussion and conclusion

2.3.1 Well-balanced discretization of the source term

In section 2.1.1.2 we have proposed a well-balanced discretization of the source term on the radiative flux equation. This discretization allows us to properly capture the steady state with constant flux and with a discontinuity of opacity. However, this discretization can lead to spurious oscillations in the radiative flux, a problem that we have encountered in the test case for the expansion of H II regions. Although we have changed the discretization of the source term to achieve a well-balanced property, our integration of the hyperbolic part on the source term is still split into two steps. Such a splitting strategy might be unstable if the source term is not taken into account in the hyperbolic part. A possible solution to this problem would be to incorporate the source term in a Lagrange-projection-like scheme such as Buet and Despres [2008].

2.3.2 Asymptotic limit for radiation hydrodynamics

In section 2.1.1.3, we have presented an asymptotic correction which allows us to capture the asymptotic behavior, whereas the solution given by a standard scheme is dominated by numerical diffusion. The asymptotic correction uses the numerical diffusion to recover the physical one in a static fluid. Nevertheless, this scheme does not capture the asymptotic regime in a moving fluid, as presented in section 1.C. Most of the schemes proposed in the literature do not preserve this asymptotic regime (e.g., Berthon and Turpault 2011; González et al. 2007). The diffusive regime depends on the material velocity; our scheme cannot reach it. A possible solution would be to limit the numerical diffusion with a correction similar to a low Mach correction, as in Chalons et al. [2016], in conjunction with a cell-centered discretization of the source term, as proposed by the Lagrange-projection scheme of Buet and Despres [2008]. This Lagrange-projection-like strategy is explored in chapter 3 to treat both the asymptotic preserving and well-balanced issues.

2.4 Bibliography

H. Bloch, P. Tremblin, M. González, T. Padioleau, and E. Audit. A high-performance and portable asymptotic preserving radiation hydrodynamics code with the m1 model. *A&A*, 646:A123, 2021. doi: 10.1051/0004-6361/202038579. URL https://doi.org/10.1051/0004-6361/202038579. 36

Christophe Berthon and Rodolphe Turpault. Asymptotic preserving hll schemes. *Numerical Methods for Partial Differential Equations*, 27:1396 – 1422, 11 2011. doi: 10.1002/num.20586. 36, 39, 51

M. González, E. Audit, and P. Huynh. Heracles: a three-dimensional radiation hydrodynamics code. *A&A*, 464(2):429–435, 2007. doi: 10.1051/0004-6361:20065486. URL https://doi.org/10.1051/0004-6361:20065486. 36, 37, 39, 42, 44, 45, 46, 51

Eleuterio Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*, pages 163–212. 03 2009. ISBN 978-3-540-25202-3. doi: 10.1007/b79761_5. 36

Christophe Berthon, Pierre Charrier, and Bruno Dubroca. An hllc scheme to solve the m1 model of radiative transfer in two space dimensions. *Journal of Scientific Computing*, 31(3):347–389, Jun 2007. ISSN 1573-7691. doi: 10.1007/s10915-006-9108-6. URL https://doi.org/10.1007/s10915-006-9108-6. 37, 44

Christophe Berthon, Anaïs Crestetto, and Françoise Foucher. A Well-Balanced Finite Volume Scheme for a Mixed Hyperbolic/Parabolic System to Model Chemotaxis. *Journal of Scientific Computing*, 67:618–643, September 2015. URL https://hal.archives-ouvertes.fr/hal-01620749. 37

Thomas Padioleau, Pascal Tremblin, Edouard Audit, Pierre Kestener, and Samuel Kokh. A high-performance and portable all-mach regime flow solver code with well-balanced gravity. application to compressible convection. *The Astrophysical Journal*, 875(2):128, apr 2019. doi: 10.3847/1538-4357/ab0f2c. URL https://doi.org/10.3847/1538-4357/ab0f2c. 37, 39, 40, 48

H. Van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992. doi: 10.1137/0913035. URL https://doi.org/10.1137/0913035. 40

D. Mihalas and B. W. Mihalas. *Foundations of radiation hydrodynamics*. 1984. 40

S. Richling, E. Meinköhn, N. Kryzhevoi, and G. Kanschat. Radiative transfer with finite elements - i. basic method and tests. *A&A*, 380(2):776–788, 2001. doi: 10.1051/0004-6361:20011411. URL https://doi.org/10.1051/0004-6361:20011411. 42

John C. Hayes and Michael L. Norman. Beyond flux-limited diffusion: Parallel algorithms for multidimensional radiation hydrodynamics. *The Astrophysical Journal Supplement Series*, 147(1):197–220, jul 2003. doi: 10.1086/374658. URL https://doi.org/10.1086/374658. 44, 45

L Ensman. Test problems for radiation and radiation-hydrodynamics codes. *Astrophysical Journal*, 424(1), 3 1994. doi: 10.1086/173889. URL https://www.osti.gov/biblio/45745. 45, 46

Ed Churchwell. Ultra-compact hii regions and massive star formation. *Annual Review of Astronomy and Astrophysics*, 40(1):27–62, 2002. doi: 10.1146/annurev.astro.40.060401.093845. URL https://doi.org/10.1146/annurev.astro.40.060401.093845. 47

F. Herpin, M. Marseille, V. Wakelam, S. Bontemps, and D. C. Lis. S-bearing molecules in massive dense cores ***. *A&A*, 504(3):853–867, 2009. doi: 10.1051/0004-6361/200811257. URL https://doi.org/10.1051/0004-6361/200811257. 47, 48

Pascal Tremblin. *Ionization impact on molecular clouds and star formation : numerical simulations and observations*. PhD thesis, 2012. URL http://www.theses.fr/2012PA077217. 47, 48

Pierre Lesaffre. *Dynamical aspects of the interstellar medium*. Theses, Université Paris-Diderot - Paris VII, September 2002. URL https://tel.archives-ouvertes.fr/tel-00123712. 47, 48, 49

John H. Black. The physical state of primordial intergalactic clouds. *Monthly Notices of the Royal Astronomical Society*, 197(3):553–563, 12 1981. ISSN 0035-8711. doi: 10.1093/mnras/197.3.553. URL https://doi.org/10.1093/mnras/197.3.553. 47

P. Tremblin, E. Audit, V. Minier, W. Schmidt, and N. Schneider. Three-dimensional simulations of globule and pillar formation around hii regions: turbulence and shock curvature. *A&A*, 546:A33, 2012. doi: 10.1051/0004-6361/201219224. URL https://doi.org/10.1051/0004-6361/201219224. 49

Akira Mizuta, Jave O. Kane, Marc W. Pound, Bruce A. Remington, Dmitri D. Ryutov, and Hideaki Takabe. Hydrodynamic instability of ionization fronts in HiiRegions. *The Astrophysical Journal*, 621(2):803–807, mar 2005. doi: 10.1086/427677. URL https://doi.org/10.1086/427677. 50

T. G. Bisbas, T. J. Haworth, R. J. R. Williams, J. Mackey, P. Tremblin, A. C. Raga, S. J. Arthur, C. Baczynski, J. E. Dale, T. Frostholm, S. Geen, T. Haugbølle, D. Hubber, I. T. Iliev, R. Kuiper, J. Rosdahl, D. Sullivan, S. Walch, and R. Wünsch. starbench: the D-type expansion of an Hii region. *Monthly Notices of the Royal Astronomical Society*, 453(2):1324–1343, 08 2015. ISSN 0035-8711. doi: 10.1093/mnras/stv1659. URL https://doi.org/10.1093/mnras/stv1659. 50

Christophe Buet and Bruno Despres. A gas dynamics scheme for a two moments model of radiative transfer. 11 2008. 50, 51

Christophe Chalons, Mathieu Girardin, and Samuel Kokh. An all-regime lagrange-projection like scheme for the gas dynamics equations on unstructured meshes. *Communications in Computational Physics*, 20(1):188233, 2016. doi: 10.4208/cicp.260614.061115a. 51

John David Anderson. *Computational fluid dynamics : the basics with applications*. McGraw-Hill series in mechanical engineering. McGraw-Hill, New York, 1995. ISBN 0-07-001685-2. 53

2.A Von Neumann stability analysis for the well-balanced modification of the source term

For simplicity, we split eq. 1.15b into a pure hyperbolic problem $\partial_t \boldsymbol{F}_\mathbf{r} + c^2 \nabla \cdot \mathbb{P}_\mathrm{r} = 0$ and a source problem $\partial_t \boldsymbol{F}_\mathbf{r} = -c\sigma^\mathrm{s} \boldsymbol{F}_\mathbf{r}$. We focus on the one-dimensional source problem, with periodic boundary conditions on the domain $[0, T] \times [0, 1]$ with T the final time. The following can easily be extended to an arbitrary space interval. Because we use periodic boundary conditions, we can apply the von Neumann stability analysis (see e.g., Anderson 1995), based on the decomposition of the numerical solution into Fourier series. Let us recall that, using the well-balanced modification of the source term (eq. 2.7), the source problem is discretized as

$$F_j^{n+1} + r_{j-\frac{1}{2}} F_{j-1}^{n+1} + \left(r_{j-\frac{1}{2}} + r_{j+\frac{1}{2}} \right) F_j^{n+1} + r_{j+\frac{1}{2}} F_{j+1}^{n+1} = F_j^n, \tag{2.20}$$

with $r_{j+\frac{1}{2}} = \frac{c\sigma^\mathrm{s}_{j+\frac{1}{2}} \Delta t}{4}$. We define the function F^n, piecewise constant, such that

$$F^n(x) = \begin{cases} F_j^n \text{ if } x_{j-\frac{1}{2}} < x < x_{j+\frac{1}{2}} \\ 0 \text{ otherwise.} \end{cases} \tag{2.21}$$

This function is then extended to $\mathbb{R}$ by periodicity. F^n can now be expanded in a Fourier series:

$$F^n(x) = \sum_{k \in \mathbb{Z}} \hat{F}^n(k) e^{2ik\pi x}, \tag{2.22}$$

with

$$\hat{F}^n(k) = \int_0^1 F^n(x) e^{-2ik\pi x} \mathrm{d}x. \tag{2.23}$$

We can define the 2-norm of the function F^n:

$$||F^n||_2 = \left(\int_0^1 \left(F^n(x)\right)^2 \mathrm{d}x \right)^{\frac{1}{2}} = \left(\sum_{k \in \mathbb{Z}} \left| \hat{F}^n(k) \right|^2 \right)^{\frac{1}{2}} = \left(\sum_{j=1}^{J} \Delta x \left(F_j^n\right)^2 \right)^{\frac{1}{2}}. \tag{2.24}$$

We apply the Fourier transform to eq. 2.20:

$$\hat{F}^{n+1}(k)\left(1+r_{j-\frac{1}{2}}e^{-2ik\pi\Delta x}+r_{j-\frac{1}{2}}+r_{j+\frac{1}{2}}+r_{j+\frac{1}{2}}e^{2ik\pi\Delta x}\right)=\hat{F}^{n}(k). \tag{2.25}$$

We define the amplification factor $A(k)$ as

$$A(k)=\frac{1}{1+r_{j-\frac{1}{2}}e^{-2ik\pi\Delta x}+r_{j-\frac{1}{2}}+r_{j+\frac{1}{2}}+r_{j+\frac{1}{2}}e^{2ik\pi\Delta x}}, \tag{2.26}$$

and we have then $\hat{F}^{n+1}(k)=A(k)\hat{F}^{n}(k)$. By induction, we have $\hat{F}^{n}(k)=(A(k))^{n}\,\hat{F}^{0}(k)$. The coefficient $\hat{F}^{n}(k)$ remains bounded if and only if $|A(k)|\leqslant 1$. In this case, for all $k\in\mathbb{Z}$, $\left|\hat{F}^{n+1}(k)\right|\leqslant\left|\hat{F}^{n}(k)\right|$. Therefore, $||F^{n+1}||_2\leqslant||F^{n}||_2\leqslant||F^{0}||_2$ and the scheme is unconditionally stable. We now have to prove that $|A(k)|\leqslant 1$:

$$\begin{aligned}
\frac{1}{|A(k)|^2}&=\left(1+r_{j-\frac{1}{2}}+r_{j+\frac{1}{2}}\right)^2+r_{j-\frac{1}{2}}^2+r_{j+\frac{1}{2}}^2+2r_{j-\frac{1}{2}}r_{j+\frac{1}{2}}\cos(4k\pi\Delta x)\\
&\quad+2r_{j-\frac{1}{2}}\left(1+r_{j-\frac{1}{2}}+r_{j+\frac{1}{2}}\right)\cos(2k\pi\Delta x)+2r_{j+\frac{1}{2}}\left(1+r_{j-\frac{1}{2}}+r_{j+\frac{1}{2}}\right)\cos(2k\pi\Delta x)\\
&=\left(1+\underbrace{r_{j-\frac{1}{2}}\left(1-\cos(2k\pi\Delta x)\right)}_{\geqslant 0}+\underbrace{r_{j+\frac{1}{2}}\left(1-\cos(2k\pi\Delta x)\right)}_{\geqslant 0}\right)^2\\
&\quad+\underbrace{\left(r_{j-\frac{1}{2}}\sin(2k\pi\Delta x)-r_{j+\frac{1}{2}}\sin(2k\pi\Delta x)\right)^2}_{\geqslant 0}\\
&\geqslant 1.
\end{aligned} \tag{2.27}$$

Since $\frac{1}{|A(k)|^2}\geqslant 1$, we have $|A(k)|\leqslant 1$.

2.B Numerical scheme in the diffusive limit

We consider the numerical scheme developed in section 2.1.1.3 in the asymptotic regime, with $\sigma^{\mathrm{s}}_{i+\frac{1}{2}}\Delta x\to\infty$. Following section 1.4, we introduce the rescaling parameter ε to write the time (resp. the opacity) as $\widetilde{\Delta t}=\varepsilon\Delta t$ (resp. $\widetilde{\sigma^{\mathrm{a}}}=\varepsilon\sigma^{\mathrm{a}}$ and $\widetilde{\sigma^{\mathrm{s}}}=\varepsilon\sigma^{\mathrm{s}}$). Using the P_1 closure relation, we have $\lambda^{+}_{i+\frac{1}{2}}=-\lambda^{-}_{i+\frac{1}{2}}=\frac{c}{\sqrt{3}}$ and

$$\varepsilon^2E_i^{n+1}=\varepsilon^2E_i^{n}-\frac{\widetilde{\Delta t}}{\Delta x}\left(\varepsilon\alpha_{i+\frac{1}{2}}\mathcal{F}^{*}_{i+\frac{1}{2}}-\varepsilon\alpha_{i-\frac{1}{2}}\mathcal{F}^{*}_{i-\frac{1}{2}}\right)+c\widetilde{\Delta t}\widetilde{\sigma^{\mathrm{a}}_i}\left(a_{\mathrm{r}}\left(T_i^{n+1}\right)^4-E_i^{n+1}\right) \tag{2.28a}$$

$$\varepsilon^2F_i^{n+1}=\varepsilon^2F_i^{n}-\frac{\widetilde{\Delta t}}{\Delta x}\left(\varepsilon\mathcal{P}^{*}_{i+\frac{1}{2}}-\varepsilon\mathcal{P}^{*}_{i-\frac{1}{2}}\right)-\frac{c\widetilde{\Delta t}}{2}\left(\widetilde{\sigma^{\mathrm{s}}}_{i+\frac{1}{2}}F^{n+1}_{i+\frac{1}{2}}+\widetilde{\sigma^{\mathrm{s}}}_{i-\frac{1}{2}}F^{n+1}_{i-\frac{1}{2}}\right) \tag{2.28b}$$

$$\varepsilon^2\rho c_vT_i^{n+1}=\varepsilon^2\rho c_vT_i^{n}-c\widetilde{\Delta t}\widetilde{\sigma^{\mathrm{a}}_i}\left(a_{\mathrm{r}}\left(T_i^{n+1}\right)^4-E_i^{n+1}\right). \tag{2.28c}$$

Radiative variables are expanded, e.g., $E_i^n=E_{i,0}^n+\varepsilon E_{i,1}^n+\mathcal{O}\left(\varepsilon^2\right)$. Expanding eqs. 2.28a and 2.28b at order 0 leads to

$$\begin{cases}E_{i,0}^{n+1}=a_{\mathrm{r}}\left(T_{i,0}^{n+1}\right)^4\\ F_{i,0}^{n+1}=0.\end{cases} \tag{2.29}$$

At first order for eq. 2.28b, we have

$$\widetilde{\sigma^{\mathrm{s}}}_{i+\frac{1}{2}} F^{n+1}_{i+\frac{1}{2},1} + \widetilde{\sigma^{\mathrm{s}}}_{i-\frac{1}{2}} F^{n+1}_{i-\frac{1}{2},1} = -\frac{c}{3}\frac{E^{n+1}_{i+1,0} - E^{n+1}_{i-1,0}}{\Delta x} + \underbrace{\frac{F^{n+1}_{i+1,0} - 2F^{n+1}_{i,0} + F^{n+1}_{i-1,0}}{\sqrt{3}\Delta x}}_{=0}. \tag{2.30}$$

Using the boundary condition given by eq. 2.13, we have

$$\widetilde{\sigma^{\mathrm{s}}}_{i+\frac{1}{2}} F^{n+1}_{i+\frac{1}{2}} = -\frac{c}{3\Delta x}\left(E^{n+1}_{i+1,0} - E^{n+1}_{i,0}\right) \tag{2.31}$$

in the whole domain.

Now, we consider the sum of eqs. 2.28a and 2.28c expanded at second order. If $\alpha_{i+\frac{1}{2}} = 1$, we have

$$\begin{aligned}
\varepsilon^2\left(E^{n+1}_{i,0} + \rho c_v T^{n+1}_{i,0}\right) = \varepsilon^2\left(E^{n}_{i,0} + \rho c_v T^{n}_{i,0}\right) \\
+ \varepsilon\frac{\widetilde{\Delta t}}{\Delta x}\frac{c}{2\sqrt{3}}\left(\alpha_{i+\frac{1}{2}}\left(E^{n+1}_{i+1,0} - E^{n+1}_{i,0}\right) - \alpha_{i-\frac{1}{2}}\left(E^{n+1}_{i,0} - E^{n+1}_{i-1,0}\right)\right) \\
- \varepsilon^2\frac{\widetilde{\Delta t}}{2\Delta x}\alpha_{i+\frac{1}{2}}\left(F^{n+1}_{i+1,1} + F^{n+1}_{i,1} - \frac{c}{\sqrt{3}}\left(E^{n+1}_{i+1,1} - E^{n+1}_{i,1}\right)\right) \\
+ \varepsilon^2\frac{\widetilde{\Delta t}}{2\Delta x}\alpha_{i-\frac{1}{2}}\left(F^{n+1}_{i,1} + F^{n+1}_{i-1,1} - \frac{c}{\sqrt{3}}\left(E^{n+1}_{i,1} - E^{n+1}_{i-1,1}\right)\right),
\end{aligned} \tag{2.32}$$

whereas the asymptotic development of a standard discretization of eq. 1.24 would be

$$\varepsilon^2\left(E^{n+1}_{i,0} + \rho c_v T^{n+1}_{i,0}\right) = \varepsilon^2\left(E^{n}_{i,0} + \rho c_v T^{n}_{i,0}\right) + \varepsilon\frac{c}{3}\frac{\Delta t}{\Delta x^2}\left(\frac{E^{n+1}_{i+1,0} - E^{n+1}_{i,0}}{\sigma^{\mathrm{s}}_{i+\frac{1}{2}}} - \frac{E^{n+1}_{i,0} - E^{n+1}_{i-1,0}}{\sigma^{\mathrm{s}}_{i-\frac{1}{2}}}\right). \tag{2.33}$$

So, we are looking for $\alpha_{i+\frac{1}{2}}$ such that the term of order 1 in eq. 2.32 becomes a term of order 2 with the expected coefficient of diffusion $\frac{c}{3\sigma^{\mathrm{s}}_{i+\frac{1}{2}}}$ and the term of order 2 becomes a term of order 3 and therefore negligible. In other words, we want the asymptotic development of $\alpha_{i+\frac{1}{2}}$ to be $\frac{2\varepsilon}{\sqrt{3}\sigma^{\mathrm{s}}_{i+\frac{1}{2}}\Delta x}$. One way to achieve this is to take

$$\alpha_{i+\frac{1}{2}} = \frac{1}{1 + \sqrt{3}\sigma^{\mathrm{s}}_{i+\frac{1}{2}}\frac{\Delta x}{2}}. \tag{2.34}$$

However, in the general case, we do not have $\lambda^+_{i+\frac{1}{2}} = -\lambda^-_{i+\frac{1}{2}} = \frac{c}{\sqrt{3}}$. We can then replace eq. 2.34 by

$$\alpha_{i+\frac{1}{2}} = \frac{1}{1 - 3\sigma^{\mathrm{s}}_{i+\frac{1}{2}}\Delta x\dfrac{\lambda^+_{i+\frac{1}{2}}\lambda^-_{i+\frac{1}{2}}}{c\left(\lambda^+_{i+\frac{1}{2}} - \lambda^-_{i+\frac{1}{2}}\right)}}. \tag{2.35}$$

Unfortunately, in numerical tests with $\sigma^{\mathrm{s}}\Delta x$ close to 1, the condition $f < 1$ is not preserved. Because f is close to 1 in this case, we write

$$\alpha_{i+\frac{1}{2}} = \frac{1}{1 - 3\sigma^{\mathrm{s}}_{i+\frac{1}{2}}\Delta x\left(1 - f^2_{i+\frac{1}{2}}\right)\dfrac{\lambda^+_{i+\frac{1}{2}}\lambda^-_{i+\frac{1}{2}}}{c\left(\lambda^+_{i+\frac{1}{2}} - \lambda^-_{i+\frac{1}{2}}\right)}}. \tag{2.36}$$

We use $f_{i+\frac{1}{2}} = \frac{1}{2}\left(f_i^n + f_{i+1}^n\right)$ because numerical experiments have shown good results using this form. In the diffusion regime, because $F_{i,0}^{n+1} = 0$, we recover eq. 2.35.

Now that we have the form of $\alpha_{i+\frac{1}{2}}$, we can check that the proposed scheme is asymptotic preserving. We have

$$\alpha_{i+\frac{1}{2}} = \frac{2\varepsilon}{\sqrt{3}\widetilde{\sigma^{\mathrm{s}}}_{i+\frac{1}{2}}\Delta x} + \mathcal{O}\left(\varepsilon^2\right). \tag{2.37}$$

Therefore,

$$\alpha_{i+\frac{1}{2}}\mathcal{F}^*_{i+\frac{1}{2}} = -\varepsilon\frac{c}{3\widetilde{\sigma^{\mathrm{s}}}_{i+\frac{1}{2}}}\frac{E_{i+1,0}^{n+1} - E_{i,0}^{n+1}}{\Delta x} + \mathcal{O}\left(\varepsilon^2\right). \tag{2.38}$$

We finally have

$$E_{i,0}^{n+1} + \rho c_v T_{i,0}^{n+1} = E_{i,0}^{n} + \rho c_v T_{i,0}^{n} + \frac{c\widetilde{\Delta t}}{3\Delta x^2}\left(\frac{E_{i+1,0}^{n+1} - E_{i,0}^{n+1}}{\widetilde{\sigma^{\mathrm{s}}}_{i+\frac{1}{2}}} - \frac{E_{i,0}^{n+1} - E_{i-1,0}^{n+1}}{\widetilde{\sigma^{\mathrm{s}}}_{i-\frac{1}{2}}}\right). \tag{2.39}$$

Equations 2.29, 2.31 and 2.39 are standard discretization of eqs. 1.22 to 1.24, so this scheme is asymptotic preserving.

Chapter 3

An all-regime-like asymptotic preserving scheme

Contents

The numerical scheme we have shown in the previous chapter does not preserve the admissible states $E_\mathrm{r} > 0$ and $f \leqslant 1$. We present in this chapter the development of another numerical scheme that preserves these admissible states, as well as the asymptotic preserving property. This scheme being inspired by all-regime schemes for compressible hydrodynamics, we also call it "all-regime scheme". We derive it using a time-explicit integration. A possible extension for a time-implicit solver is presented in chapter 4. For the sake of simplicity, we only consider the case where the hydrodynamics is frozen. The coupling to hydrodynamics could be done as in section 2.1.2.

This chapter is organized as follows. We present our numerical scheme in section 3.1. Buet and Despres [2008] have derived a Lagrange-projection scheme for the M_1 model. Inspired by Chalons et al. [2016], we adapt this scheme with a "diffusion-transport" splitting. Because this leads to a scheme of stencil 2, we also present an all-regime scheme of stencil 1. In section 3.2, we present some numerical test cases to show the good properties of our solvers. Finally, we reach our conclusion in section 3.3 and we discuss the limits of the schemes.

3.1 Numerical scheme

We use the same notations as in chapter 2: we note Δx the step along the x-direction. Δt is the time interval between the current time t^n and t^{n+1}. We write x_i the center of the cell i and $x_{i+\frac{1}{2}}$ the interface between the cell i and the cell $i+1$. We use the notation u_i^n to represent the averaged quantity associated with the field u at time t^n in the cell i (finite volume). Finally, we note $u_{i+\frac{1}{2}}^n$ to represent the quantity associated with the field u at time t^n and at the interface between cells i and $i+1$.

Because we use a time-explicit scheme, the time step Δt is restricted by a CFL condition. For the sake of simplicity, we always use $\Delta t < \frac{\Delta x}{c}$ in the numerical tests. For physical reasons, the fastest phenomena occur at the speed of light, therefore we use this velocity to restrict the time step, even though slower phenomena could allow us to use a larger time step.

3.1.1 Rewriting the M_1 model

In this chapter, we adapt a scheme developed for hydrodynamics to the M_1 model. We first rephrase our model to have the same form as the Euler equation. Following Buet and Despres [2008], we introduce a variable $\varrho \in \mathbb{R}$ that obeys the transport equation

$$\partial_t \varrho + \nabla \cdot (\varrho \boldsymbol{\beta}) = 0, \tag{3.1}$$

with $\boldsymbol{\beta}$ already defined in section 1.B.2, given by

$$\boldsymbol{\beta} = \frac{1}{f}\frac{3\chi - 1}{2}\frac{\boldsymbol{F}_\mathbf{r}}{E_\mathrm{r}}. \tag{3.2}$$

Despite the resemblance with the mass conservation equation of Euler equations, ϱ is not the density of the fluid ρ. However, it will play a similar role in the derivation of the scheme.

In chapter 1, we have shown that

$$\begin{cases} \boldsymbol{F}_\mathbf{r} = (q + E_\mathrm{r})\,\boldsymbol{\beta} \\ \mathbb{P}_\mathrm{r} = \dfrac{\boldsymbol{\beta} \otimes \boldsymbol{F}_\mathbf{r}}{c^2} + q\mathbb{I}, \end{cases} \tag{3.3}$$

with $q = \frac{c^2-\beta^2}{3c^2+\beta^2}E_\mathrm{r}$, where $\beta = ||\boldsymbol{\beta}||$. Let us notice that $\boldsymbol{\beta}$ is a velocity and q is a pressure.

Let us introduce an specific energy e and a specific flux $\boldsymbol{v}$ such that

$$\begin{cases} E_\mathrm{r} = \varrho e \\ \boldsymbol{F}_\mathbf{r} = \varrho \boldsymbol{v}. \end{cases} \tag{3.4}$$

Despite the notation, $\boldsymbol{v}$ is not a velocity. Because of the similarly with Euler equations, we use the same vocabulary as hydrodynamics: we call ϱ a density. Let us notice that the admissible states $E_\mathrm{r} > 0$ and $\frac{||\boldsymbol{F}_\mathbf{r}||}{cE_\mathrm{r}} \leqslant 1$ are equivalent to $\varrho > 0$, $e > 0$ and $\frac{||\boldsymbol{v}||}{ce} \leqslant 1$.

Let us first consider the case without source terms, with $\sigma^\mathrm{a} = \sigma^\mathrm{s} = 0$. Using eqs. 3.1, 3.3 and 3.4, the M_1 model (eq. 1.15) can be written as

$$\begin{cases} \partial_t \varrho + \nabla \cdot (\varrho \boldsymbol{\beta}) = 0 \\ \partial_t (\varrho \boldsymbol{v}) + \nabla \cdot (\varrho \boldsymbol{v} \otimes \boldsymbol{\beta}) + c^2 \nabla q = 0 \\ \partial_t (\varrho e) + \nabla \cdot (\varrho e \boldsymbol{\beta} + q\boldsymbol{\beta}) = 0. \end{cases} \tag{3.5}$$

Equation 3.5 has the same form as the Euler equations. The main difference is the link between the momentum $\varrho\boldsymbol{v}$ and the velocity $\boldsymbol{\beta}$: in the hydrodynamics case, it is linear (the momentum is $\rho\boldsymbol{u}$ and the velocity is $\boldsymbol{u}$), whereas it is not linear for the M_1 model (eq. 3.2). Let us now focus on the discretization of eq. 3.5.

3.1.2 Diffusion-transport splitting

Following Chalons et al. [2016], we use a splitting strategy to solve eq. 3.5. In the hydrodynamics case, the Euler system is split into an acoustic step and a transport step. Here, we split eq. 3.5 into a diffusion step and a transport step. The choice of the terms "diffusion" and "transport" will be explained later. Transport phenomena of the form $\boldsymbol{\beta} \cdot \nabla$ are separated from the other terms. We first solve the diffusion step

$$\begin{cases} \partial_t \varrho + \varrho \nabla \cdot \boldsymbol{\beta} = 0 \\ \partial_t (\varrho \boldsymbol{v}) + \varrho \boldsymbol{v} \nabla \cdot \boldsymbol{\beta} + c^2 \nabla q = 0 \\ \partial_t (\varrho e) + \varrho e \nabla \cdot \boldsymbol{\beta} + \nabla \cdot (q\boldsymbol{\beta}) = 0, \end{cases} \tag{3.6}$$

followed by the transport step

$$\begin{cases} \partial_t \varrho + \boldsymbol{\beta} \cdot \nabla \varrho = 0 \\ \partial_t (\varrho \boldsymbol{v}) + \boldsymbol{\beta} \cdot \nabla (\varrho \boldsymbol{v}) = 0 \\ \partial_t (\varrho e) + \boldsymbol{\beta} \cdot \nabla (\varrho e) = 0. \end{cases} \tag{3.7}$$

On one hand, in the diffusive limit, we have $\boldsymbol{F}_\mathbf{r} = \mathbf{0}$ at first order. Because $\boldsymbol{F}_\mathbf{r}$ and $\boldsymbol{\beta}$ are colinear and $E_\mathrm{r} + q \neq 0$, we also have $\boldsymbol{\beta} = \mathbf{0}$ at first order. Therefore, the variables are not impacted by the transport step at first order. Furthermore, one can show that $q = \frac{E_\mathrm{r}}{3}$ at first order, which is equivalent to the P_1 closure relation. Let us notice the source terms play an important role in the diffusive limit, as shown in section 1.4, therefore, the analysis done here is not whole. Recovering the diffusion equation eq. 1.24 brings into play velocity $\boldsymbol{\beta}$ at order 1. More details about the integration of source terms can be found in section 3.1.6.

On the other hand, in the transport regime, $\boldsymbol{\beta} = c\boldsymbol{n}$, where $\boldsymbol{n}$ is a unit vector aligned with the radiative flux, and $q = 0$. The diffusion step simplifies into $\partial_t \vartheta + c\vartheta \nabla \cdot \boldsymbol{n} = 0$

for $\vartheta \in \{\varrho, \varrho\boldsymbol{v}, \varrho e\}$. In particular, if $\boldsymbol{n}$ is constant, all variables remain constant during the diffusion step. They are then transported at velocity c by the transport step.

Using eq. 3.6, one can deduce the equation of evolution of velocity $\boldsymbol{\beta}$ and pressure q:

$$\begin{cases} \partial_t \boldsymbol{\beta} - \dfrac{c^2-\beta^2}{3c^2-\beta^2}\boldsymbol{\beta}\nabla\cdot\boldsymbol{\beta} + \dfrac{c^2-\beta^2}{4q}\nabla q - \dfrac{c^2-\beta^2}{2q\left(3c^2-\beta^2\right)}\boldsymbol{\beta}\left(\boldsymbol{\beta}\cdot\nabla q\right) = 0 \\ \partial_t q + \dfrac{4c^2 q}{3c^2-\beta^2}\nabla\cdot\boldsymbol{\beta} - \dfrac{c^2-\beta^2}{3c^2-\beta^2}\boldsymbol{\beta}\cdot\nabla q = 0. \end{cases} \tag{3.8}$$

Detailed computations are done in section 3.A.

Let us now study the eigenstructure of eqs. 3.6 and 3.7. For the sake of simplicity, we focus on the one-dimensional case. We write λ^-, λ^0 and λ^+ the eigenvalues of the M_1 model, with $\lambda^- \leqslant \lambda^0 \leqslant \lambda^+$. From González et al. [2007],

$$\begin{cases} \lambda^- = c\dfrac{n_x\chi' - \sqrt{\chi'^2 + 4\chi - 4\chi' f}}{2} \\ \lambda^0 = c^2\dfrac{3\chi-1}{2}\dfrac{E_\mathrm{r}}{F_\mathrm{r}} \\ \lambda^+ = c\dfrac{n_x\chi' + \sqrt{\chi'^2 + 4\chi - 4\chi' f}}{2}, \end{cases} \tag{3.9}$$

where n_x is the x-coordinate of $\boldsymbol{n} = \frac{\boldsymbol{f}}{f}$ and $\chi' = \frac{\mathrm{d}\chi}{\mathrm{d}f}$. The eigenvalues for the transport system (eq. 3.7) are $\beta_x = \lambda^0$, with multiplicity 3. Let us now focus on the acoustic step (eq. 3.6). Equation 3.8 is introduced to ease computations. Equations 3.6 and 3.8 involve five eigenvalues: $\lambda^- - \lambda^0, 0, \lambda^+ - \lambda^0$, the multiplicity of the eigenvalue 0 is 3. The fields associated with 0 (resp. $\lambda^\pm - \lambda^0$) are linearly degenerate (resp. genuinely nonlinear). See section 3.B for more details. Both diffusion and transport systems are hyperbolic.

We now focus on the resolution of each system in the one-dimensional case. The extension to higher dimensions is straightforward.

3.1.3 Diffusion step

From now on, because we consider only the one-dimensional case, we slightly change our notations: β (resp. v) is no longer the norm of the vector $\boldsymbol{\beta}$ (resp. $\boldsymbol{v}$), but it is the x-coordinate of the vector.

Let us first write the diffusion system (eq. 3.6) using Lagrangian variables (τ, v, e), where $\tau = \frac{1}{\varrho}$. We introduce a mass variable $\mathrm{d}m = \varrho\left(t^n, x\right)\mathrm{d}x$ where the time is frozen at instant t^n. Equation 3.6 becomes

$$\begin{cases} \partial_t\tau - \partial_m\beta = 0 \\ \partial_t v + c^2\partial_m q = 0 \\ \partial_t e + \partial_m\left(q\beta\right) = 0, \end{cases} \tag{3.10}$$

which is discretized as

$$\begin{cases} \widetilde{\tau}_i = \tau_i^n + \dfrac{\Delta t}{\Delta m_i}\left(\beta^*_{i+\frac{1}{2}} - \beta^*_{i-\frac{1}{2}}\right) \\ \widetilde{v}_i = v_i^n - c^2\dfrac{\Delta t}{\Delta m_i}\left(q^*_{i+\frac{1}{2}} - q^*_{i-\frac{1}{2}}\right) \\ \widetilde{e}_i = e_i^n - \dfrac{\Delta t}{\Delta m_i}\left(q^*_{i+\frac{1}{2}}\beta^*_{i+\frac{1}{2}} - q^*_{i-\frac{1}{2}}\beta^*_{i-\frac{1}{2}}\right). \end{cases} \tag{3.11}$$

Numerical fluxes β^* and q^* will be specified later.

The update of the conservative variables $(\varrho, \varrho v, \varrho e)$ can be written as

$$\begin{cases} \widetilde{L}_i \widetilde{\varrho}_i = \varrho_i^n \\ \widetilde{L}_i \widetilde{(\varrho v)}_i = (\varrho v)_i^n - c^2 \dfrac{\Delta t}{\Delta x} \left(q^*_{i+\frac{1}{2}} - q^*_{i-\frac{1}{2}} \right) \\ \widetilde{L}_i \widetilde{(\varrho e)}_i = (\varrho e)_i^n - \dfrac{\Delta t}{\Delta x} \left(q^*_{i+\frac{1}{2}} \beta^*_{i+\frac{1}{2}} - q^*_{i-\frac{1}{2}} \beta^*_{i-\frac{1}{2}} \right), \end{cases} \tag{3.12}$$

where $\widetilde{L}_i = 1 + \frac{\Delta t}{\Delta x} \left(\beta^*_{i+\frac{1}{2}} - \beta^*_{i-\frac{1}{2}} \right)$.

The last point needed for the resolution of the diffusion step to be complete is the choice of the numerical fluxes $\beta^*_{i+\frac{1}{2}}$ and $q^*_{i+\frac{1}{2}}$. We have chosen to use the fluxes given by Buet and Despres [2008]:

$$\begin{cases} \beta^*_{i+\frac{1}{2}} = \dfrac{\frac{\varrho_i^n e_i^n}{3c^2 + (\beta_i^n)^2} \beta_i^n + \frac{\varrho_{i+1}^n e_{i+1}^n}{3c^2 + \left(\beta_{i+1}^n\right)^2} \beta_{i+1}^n}{\frac{\varrho_i^n e_i^n}{3c^2 + (\beta_i^n)^2} + \frac{\varrho_{i+1}^n e_{i+1}^n}{3c^2 + \left(\beta_{i+1}^n\right)^2}} + \dfrac{\sqrt{3}}{4c} \dfrac{q_i^n - q_{i+1}^n}{\frac{\varrho_i^n e_i^n}{3c^2 + (\beta_i^n)^2} + \frac{\varrho_{i+1}^n e_{i+1}^n}{3c^2 + \left(\beta_{i+1}^n\right)^2}} \\ q^*_{i+\frac{1}{2}} = \dfrac{\frac{3c^2 + (\beta_i^n)^2}{\varrho_i^n e_i^n} q_i^n + \frac{3c^2 + \left(\beta_{i+1}^n\right)^2}{\varrho_{i+1}^n e_{i+1}^n} q_{i+1}^n}{\frac{3c^2 + (\beta_i^n)^2}{\varrho_i^n e_i^n} + \frac{3c^2 + \left(\beta_{i+1}^n\right)^2}{\varrho_{i+1}^n e_{i+1}^n}} + \dfrac{4c}{\sqrt{3}} \dfrac{\beta_i^n - \beta_{i+1}^n}{\frac{3c^2 + (\beta_i^n)^2}{\varrho_i^n e_i^n} + \frac{3c^2 + \left(\beta_{i+1}^n\right)^2}{\varrho_{i+1}^n e_{i+1}^n}}. \end{cases} \tag{3.13}$$

The derivation of eq. 3.13 is done in section 3.C. Buet and Despres [2008] show that these numerical fluxes preserve the admissible states, i.e., $\widetilde{e}_i > 0$ and $\frac{|\widetilde{v}_i|}{c\widetilde{e}_i} \leqslant 1$ (see section 3.D).

Other choices for β^* and q^* can be done. Riemann invariants for the diffusion step (eq. 3.10) could be used. It has been explored with S.Bulteau:

$$\begin{cases} \beta^*_{i+\frac{1}{2}} = \dfrac{(q_i^n)^{\frac{\sqrt{3}}{4}} \sqrt{\left(c + \beta_i^n\right)\left(c + \beta_{i+1}^n\right)} - \left(q_{i+1}^n\right)^{\frac{\sqrt{3}}{4}} \sqrt{\left(c - \beta_i^n\right)\left(c - \beta_{i+1}^n\right)}}{(q_i^n)^{\frac{\sqrt{3}}{4}} \sqrt{\left(c + \beta_i^n\right)\left(c + \beta_{i+1}^n\right)} + \left(q_{i+1}^n\right)^{\frac{\sqrt{3}}{4}} \sqrt{\left(c - \beta_i^n\right)\left(c - \beta_{i+1}^n\right)}} \\ q^*_{i+\frac{1}{2}} = \sqrt{q_i^n q_{i+1}^n} \left(\dfrac{\left(c - \beta_{i+1}^n\right)\left(c + \beta_i^n\right)}{\left(c + \beta_{i+1}^n\right)\left(c - \beta_i^n\right)} \right)^{\frac{1}{\sqrt{3}}}. \end{cases} \tag{3.14}$$

First numerical experiments have shown results similar to those obtained with eq. 3.13, but with a higher computational cost.

Finally, Buet and Despres [2008] show that the solution $\left(\widetilde{E}_i, \widetilde{\boldsymbol{F}}_i\right)$ is independent of the initial value of the density ϱ_i^n.

3.1.4 Transport step

The scheme proposed by Buet and Despres [2008] uses a Lagrange-remap solver. As our all-regime solver, it is split into two steps. The first one, the Lagrangian step, has the same form as our diffusion step. The fields evolve using the Lagrangian formalism. The mesh is distorted to follow the matter. This new mesh is then projected onto the original mesh during the remap step. This approach, although robust, might be difficult to implement in the multi-dimensional case. Inspired by Chalons et al. [2016],

we propose here to replace the remap step by the resolution of the transport step eq. 3.7. The mesh is not changed during the resolution of the diffusion step, therefore solving the transport step is easier than computing the remap step of the Lagrange-remap solver.

Following Padioleau et al. [2019], the transport system (eq. 3.7) can be written

$$\partial_t \vartheta + \partial_x (\vartheta \beta) - \vartheta \partial_x \beta = 0, \tag{3.15}$$

for $\vartheta \in \{\varrho, \varrho v, \varrho e\}$. We discretize it as

$$\vartheta_i^{n+1} = \widetilde{\vartheta}_i - \frac{\Delta t}{\Delta x}\left(\widetilde{\vartheta}_{i+\frac{1}{2}}\beta^*_{i+\frac{1}{2}} - \widetilde{\vartheta}_{i-\frac{1}{2}}\beta^*_{i-\frac{1}{2}}\right) + \widetilde{\vartheta}_i \frac{\Delta t}{\Delta x}\left(\beta^*_{i+\frac{1}{2}} - \beta^*_{i-\frac{1}{2}}\right). \tag{3.16}$$

The term defined at the interface $\widetilde{\vartheta}_{i+\frac{1}{2}}$ is defined by the upwind choice with respect to the velocity $\beta^*_{i+\frac{1}{2}}$:

$$\widetilde{\vartheta}_{i+\frac{1}{2}} = \begin{cases} \widetilde{\vartheta}_i \text{ if } \beta^*_{i+\frac{1}{2}} \geqslant 0 \\ \widetilde{\vartheta}_{i+1} \text{ if } \beta^*_{i+\frac{1}{2}} \leqslant 0. \end{cases} \tag{3.17}$$

Using eq. 3.12, the update of the conservative quantities is then

$$\boldsymbol{w}_i^{n+1} = \boldsymbol{w}_i^n - \frac{\Delta t}{\Delta x}\left(\mathcal{F}^*\left(\widetilde{\boldsymbol{w}}_i, \widetilde{\boldsymbol{w}}_{i+1}\right) - \mathcal{F}^*\left(\widetilde{\boldsymbol{w}}_{i-1}, \widetilde{\boldsymbol{w}}_i\right)\right), \tag{3.18}$$

where $\boldsymbol{w} = (\varrho, F, E)^T$ and

$$\mathcal{F}^*\left(\begin{pmatrix}\varrho_L\\F_L\\E_L\end{pmatrix}, \begin{pmatrix}\varrho_R\\F_R\\E_R\end{pmatrix}\right) = \begin{cases} \begin{pmatrix}\varrho_L\beta^*\\F_L\beta^* + c^2 q^*\\E_L\beta^* + q^*\beta^*\end{pmatrix} \text{ if } \beta^* \geqslant 0, \\ \begin{pmatrix}\varrho_R\beta^*\\F_R\beta^* + c^2 q^*\\E_R\beta^* + q^*\beta^*\end{pmatrix} \text{ if } \beta^* \leqslant 0, \end{cases} \tag{3.19}$$

This choice of discretization allows the total update to be conservative. Furthermore, because the transport step is simply the transport of the quantities updated by the diffusion step, all properties preserved by the diffusion step are also preserved by the transport step. In particular, the transport step preserves the admissible states $e_i^{n+1} > 0$ and $\frac{\left|v_i^{n+1}\right|}{ce_i^{n+1}} \leqslant 1$, as well as the asymptotic preserving property. For the same reason, the solution at time t^{n+1} is also independent of the value of the density at time t^n. Therefore, in numerical tests, we set ϱ to 1 in each cell at the beginning of each time step.

3.1.5 Scheme of stencil 1

When looking at the total update using this diffusion-transport splitting, from time t^n to t^{n+1}, the resulting scheme is of stencil 2. Looking at eq. 3.18, the update of ϑ_i^{n+1} involves, among others, $\widetilde{\vartheta}_{i\pm1}$, which is computed thanks to ϑ_i^n, $\vartheta_{i\pm1}^n$ and $\vartheta_{i\pm2}^n$. To tackle this issue, eq. 3.17 can be replaced by

$$\widetilde{\vartheta}_{i+\frac{1}{2}} = \begin{cases} \vartheta_i^n \text{ if } \beta^*_{i+\frac{1}{2}} \geqslant 0 \\ \vartheta_{i+1}^n \text{ if } \beta^*_{i+\frac{1}{2}} \leqslant 0. \end{cases} \tag{3.20}$$

Equation 3.18 becomes

$$\boldsymbol{w}_i^{n+1} = \boldsymbol{w}_i^n - \frac{\Delta t}{\Delta x}\left(\mathcal{F}^*\left(\boldsymbol{w}_i^n, \boldsymbol{w}_{i+1}^n\right) - \mathcal{F}^*\left(\boldsymbol{w}_{i-1}^n, \boldsymbol{w}_i^n\right)\right), \tag{3.21}$$

which is indeed a scheme of stencil 1. This scheme can also be derived as an approximate Riemann solver. It is done in section 3.E.

We show in section 3.D that the admissible states $E_\mathrm{r} > 0$ and $\frac{|F_\mathrm{r}|}{cE_\mathrm{r}} \leqslant 1$ are preserved. Looking at eqs. 3.19 and 3.21, the update of E_i^{n+1} and F_i^{n+1} does not depend on ϱ. In the numerical tests, we set it to 1 in the whole domain, at the beginning of each time step.

3.1.6 Source terms

Now that we have presented two schemes for the hyperbolic part of the M_1 model, let us consider the addition of the source terms. We aim at designing an asymptotic preserving and well-balanced scheme, i.e., it should capture the diffusive limit eq. 1.24 and the correct steady state, as explained in section 2.1.1.2.

As in section 2.1.1, we use a splitting strategy to treat the source terms. Once we have obtained an intermediate solution E_i^{n+1}, F_i^{n+1} with the scheme described in sections 3.1.3 and 3.1.4 or with the one presented in section 3.1.5, we still have to solve

$$\begin{cases} \partial_t E = c\sigma^\mathrm{a}\left(a_\mathrm{r}T^4 - E\right) \\ \partial_t F = -c\sigma^\mathrm{s}F \\ \partial_t\left(\rho c_v T\right) = -c\sigma^\mathrm{a}\left(a_\mathrm{r}T^4 - E\right), \end{cases} \tag{3.22}$$

with initial condition $E_i^{n+1}, F_i^{n+1}, T_i^n$. To avoid restricting the time step, we use an implicit integration to solve this system, as done in section 2.1.1. This leads to a semi-implicit scheme.

As Buet and Despres [2008], we focus on the radiative flux source term. In section 2.1.1.2, we use an average of a face discretization of this source term. $\sigma^\mathrm{s}F$ is discretized in space as $\frac{1}{2}\left(\sigma^\mathrm{s}_{i+\frac{1}{2}}F_{i+\frac{1}{2}} + \sigma^\mathrm{s}_{i-\frac{1}{2}}F_{i-\frac{1}{2}}\right)$, with

$$\begin{cases} \sigma^\mathrm{s}_{i+\frac{1}{2}} = \dfrac{1}{2}\left(\sigma_i^\mathrm{s} + \sigma_{i+1}^\mathrm{s}\right) \\ F^{n+1}_{i+\frac{1}{2}} = \dfrac{1}{2}\left(F_i^{n+1} + F_{i+1}^{n+1}\right). \end{cases} \tag{3.23}$$

This leads to solve a linear system and might not preserve the admissible states. Because β and F are colinear, Buet and Despres [2008] suggest using a discretization proportional to $\beta^*_{i+\frac{1}{2}} + \beta^*_{i-\frac{1}{2}}$. However, this requires the resolution of a nonlinear system. An algorithm to solve the nonlinear system arising from the discretization of the M_1 model while preserving the admissible states is presented in chapter 4. Because of the issues raised here, we will use a cell-centered discretization for the radiative flux source term, while being aware of its limitations.

To write eq. 3.13, we first wrote in section 3.C some jump relations:

$$\begin{cases} \left(q^*_{i+\frac{1}{2}} - q_i^n\right) + \dfrac{4c}{\sqrt{3}}\dfrac{E_i^n}{3c^2 + \left(\beta_i^n\right)^2}\left(\beta^*_{i+\frac{1}{2}} - \beta_i^n\right) &= 0 \\ \left(q^*_{i+\frac{1}{2}} - q_{i+1}^n\right) - \dfrac{4c}{\sqrt{3}}\dfrac{E_{i+1}^n}{3c^2 + \left(\beta_{i+1}^n\right)^2}\left(\beta^*_{i+\frac{1}{2}} - \beta_{i+1}^n\right) &= 0. \end{cases} \tag{3.24}$$

One can check that eq. 3.13 is indeed the solution of this system.

To develop an asymptotic preserving scheme, Buet and Despres [2008] replaced eq. 3.24 with

$$\begin{cases} \left(q^*_{i+\frac{1}{2}} - q^n_i\right) + \dfrac{4c}{\sqrt{3}} \dfrac{E^n_i}{3c^2 + \left(\beta^n_i\right)^2} \left(\beta^*_{i+\frac{1}{2}} - \beta^n_i\right) & = -\dfrac{\sigma^{\mathrm{s}}_{i+\frac{1}{2}} \Delta x}{2c} F^n_i \\ \left(q^*_{i+\frac{1}{2}} - q^n_{i+1}\right) - \dfrac{4c}{\sqrt{3}} \dfrac{E^n_{i+1}}{3c^2 + \left(\beta^n_{i+1}\right)^2} \left(\beta^*_{i+\frac{1}{2}} - \beta^n_{i+1}\right) & = \dfrac{\sigma^{\mathrm{s}}_{i+\frac{1}{2}} \Delta x}{2c} F^n_{i+1}, \end{cases} \tag{3.25}$$

whose solution is

$$\begin{cases} \beta^*_{i+\frac{1}{2}} = \dfrac{\frac{\varrho^n_i e^n_i}{3c^2+\left(\beta^n_i\right)^2} \beta^n_i + \frac{\varrho^n_{i+1} e^n_{i+1}}{3c^2+\left(\beta^n_{i+1}\right)^2} \beta^n_{i+1}}{\frac{\varrho^n_i e^n_i}{3c^2+\left(\beta^n_i\right)^2} + \frac{\varrho^n_{i+1} e^n_{i+1}}{3c^2+\left(\beta^n_{i+1}\right)^2}} + \dfrac{\sqrt{3}}{4c} \dfrac{\left(q^n_i - \frac{\sigma^{\mathrm{s}}_{i+\frac{1}{2}} \Delta x}{2c} \varrho^n_i v^n_i\right) - \left(q^n_{i+1} + \frac{\sigma^{\mathrm{s}}_{i+\frac{1}{2}} \Delta x}{2c} \varrho^n_{i+1} v^n_{i+1}\right)}{\frac{\varrho^n_i e^n_i}{3c^2+\left(\beta^n_i\right)^2} + \frac{\varrho^n_{i+1} e^n_{i+1}}{3c^2+\left(\beta^n_{i+1}\right)^2}} \\ q^*_{i+\frac{1}{2}} = \dfrac{\frac{3c^2+\left(\beta^n_i\right)^2}{\varrho^n_i e^n_i} \left(q^n_i - \frac{\sigma^{\mathrm{s}}_{i+\frac{1}{2}} \Delta x}{2c} \varrho^n_i v^n_i\right) + \frac{3c^2+\left(\beta^n_{i+1}\right)^2}{\varrho^n_{i+1} e^n_{i+1}} \left(q^n_{i+1} + \frac{\sigma^{\mathrm{s}}_{i+\frac{1}{2}} \Delta x}{2c} \varrho^n_{i+1} v^n_{i+1}\right)}{\frac{3c^2+\left(\beta^n_i\right)^2}{\varrho^n_i e^n_i} + \frac{3c^2+\left(\beta^n_{i+1}\right)^2}{\varrho^n_{i+1} e^n_{i+1}}} \\ \qquad + \dfrac{4c}{\sqrt{3}} \dfrac{\beta^n_i - \beta^n_{i+1}}{\frac{3c^2+\left(\beta^n_i\right)^2}{\varrho^n_i e^n_i} + \frac{3c^2+\left(\beta^n_{i+1}\right)^2}{\varrho^n_{i+1} e^n_{i+1}}}. \end{cases} \tag{3.26}$$

We show in section 3.F that, using eq. 3.26 instead of eq. 3.13, the two schemes developed previously are asymptotic preserving. We also verify this property numerically in section 3.2.2.

3.2 Numerical results

We perform a series of verification tests to validate different properties of the two schemes developed in section 3.1. We compare them with a standard HLL solver. Furthermore, we refer to the scheme presented in sections 3.1.3 and 3.1.4 as the "split scheme" and to the solver presented in section 3.1.5 as the "unsplit scheme".

We first verify that both schemes are conservative, especially the split scheme, with a two-dimensional Riemann problem. We then look at numerical diffusion with a beam test. We also show some performance results. Afterward, we explore the asymptotic preserving property with a Marshak wave. Next, we study the well-balanced behavior by reaching a steady state with a jump of opacity. Finally, we highlight all these properties with a shadow test.

3.2.1 Numerical results without source terms

3.2.1.1 2D Riemann problem

Let us first consider the test described in Blachère and Turpault [2016]. It is a 2D Riemann problem with four states. The domain $[0, 1] \times [0, 1]$ is discretized with 256×256 cells. There is no opacity: $\sigma^{\mathrm{a}} = \sigma^{\mathrm{s}} = 0$. The initial temperature is $T_0 = T_{\mathrm{r}} = 1\,000$ K and the radiative flux is set to $(1 - 10^{-8})\, cE_{\mathrm{r}} \hat{\boldsymbol{F}}$, where $\hat{\boldsymbol{F}}$ is a unit

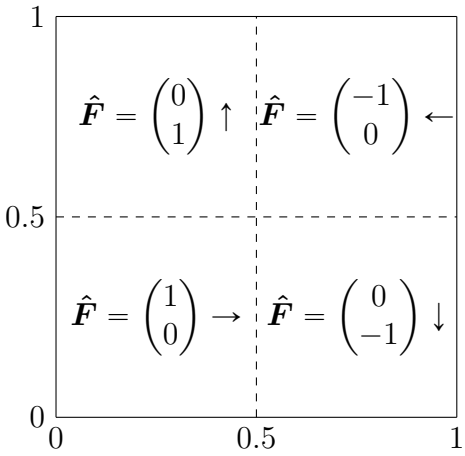


Figure 3.1 – Initial condition for 2D Riemann problem.

Scheme	Computational time (s)
HLL with computed eigenvalues	44
HLL with fixed eigenvalues	26
Split	79
Unsplit	36

Table 3.1 – Computational time with the HLL solver, with computed and fixed eigenvalues (eq. 2.2), the split scheme and the unsplit scheme.

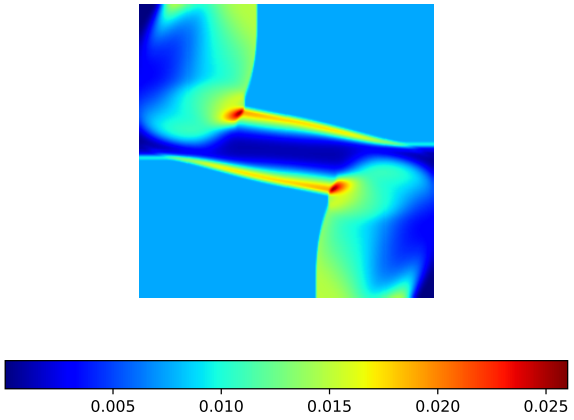
vector, piecewise constant. The domain is cut into four states. In each region, the direction of the radiative flux is constant, see fig. 3.1.

Figure 3.2 shows the radiative energy (figs. 3.2a, 3.2c and 3.2e) and the reduced flux (figs. 3.2b, 3.2d and 3.2f) at the final time 2×10^{-11} s. Solutions are obtained with a standard HLL solver (figs. 3.2a and 3.2b), the split scheme (figs. 3.2c and 3.2d) and the unsplit scheme (figs. 3.2e and 3.2f). As shown by figs. 3.2b, 3.2d and 3.2f, the reduced flux stays close to 1 during the simulation. Even for a very stiff problem, our all-regime schemes can preserve the admissible states $E_\mathrm{r} > 0$ and $f \leqslant 1$, as shown in section 3.D.

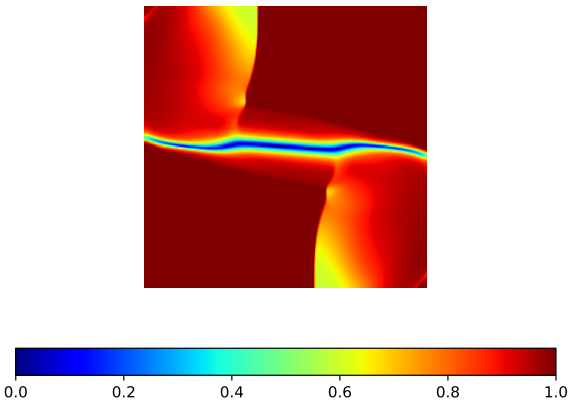
Figure 3.3 shows the evolution of the relative error between the expected total radiative energy and the one actually computed in the box at each step. For this test, we use periodic boundary conditions, so no energy should enter or leave the box, therefore, the error should remain at machine precision. Here, it oscillates around the value 10^{-12}. This shows that even with the split scheme, the total energy is conserved. Indeed, the choice of discretization for the transport step allows us to have a conservative scheme. The split scheme being conservative relies on, among others, the fact that we use the same β^* for the discretization of both diffusion and transport steps.

3.2.1.2 Beam

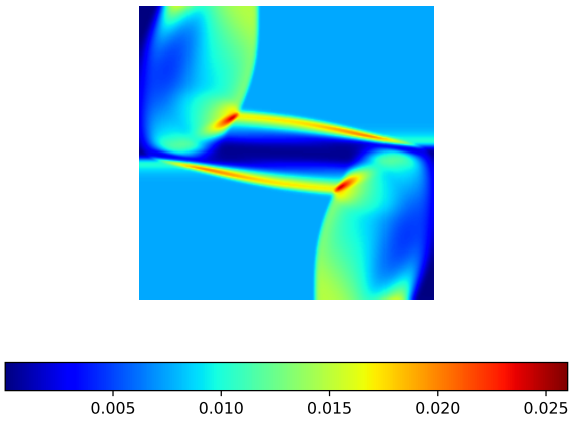
We perform the same test as in section 2.2.3, but we do not discuss the properties of the HLL solver, we use it here as a reference. It is the same test as in González et al. [2007]; Richling et al. [2001]. It is the propagation of a beam in the free-streaming regime, with $\sigma^\mathrm{a} = \sigma^\mathrm{s} = 0$. The domain $[-1, 1] \times [-1, 1]$ is discretized with 128×128 cells. The initial temperature is $T_0 = T_\mathrm{r} = 300$ K and the initial radiative flux is



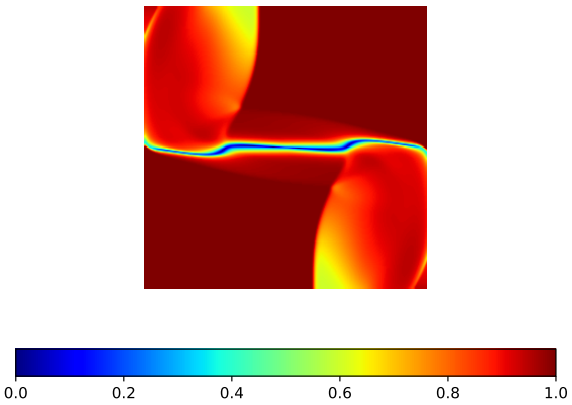
(a) Radiative energy, HLL solver



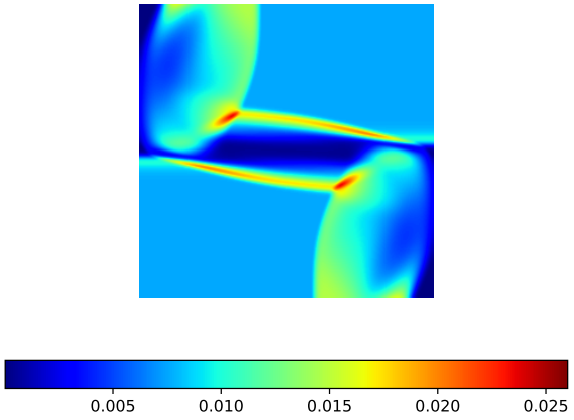
(b) Reduced flux, HLL solver



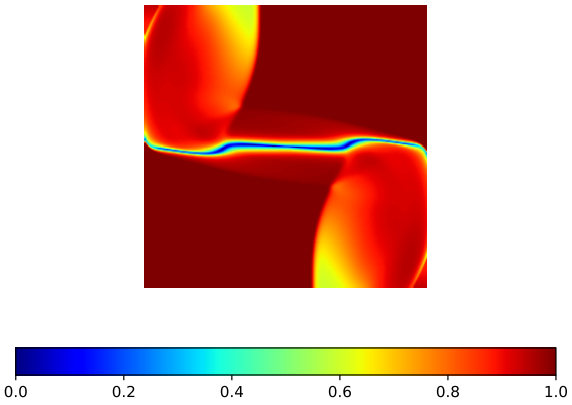
(c) Radiative energy, split solver



(d) Reduced flux, split solver



(e) Radiative energy, unsplit solver



(f) Reduced flux, unsplit solver

Figure 3.2 – Snapshots of radiative energy and reduced flux at final time $t_f = 2 \times 10^{-11}$ s with the HLL solver, the split scheme and the unsplit scheme.

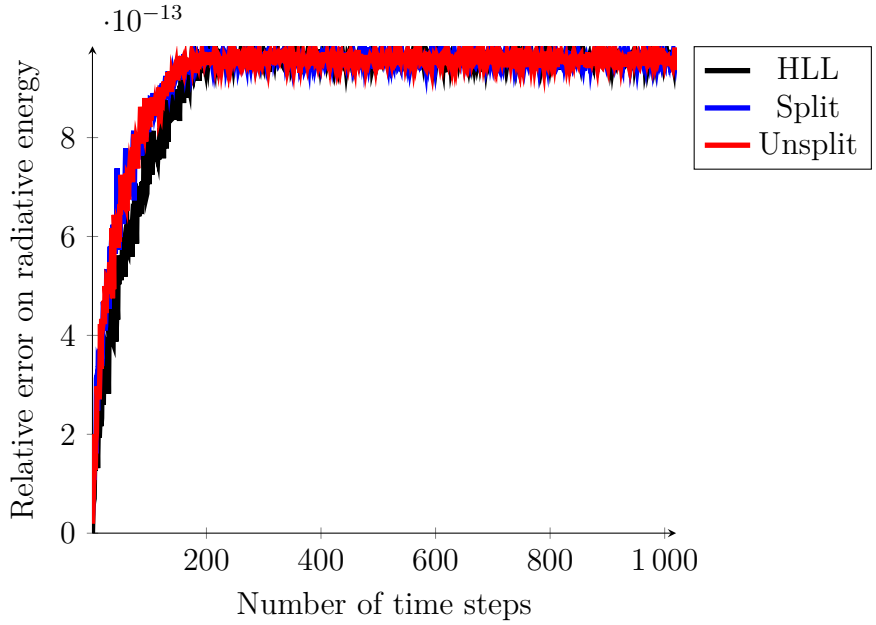


Figure 3.3 – Evolution of the relative error on radiative energy as a function of time. With periodic boundary conditions, the radiative energy is conserved at machine precision.

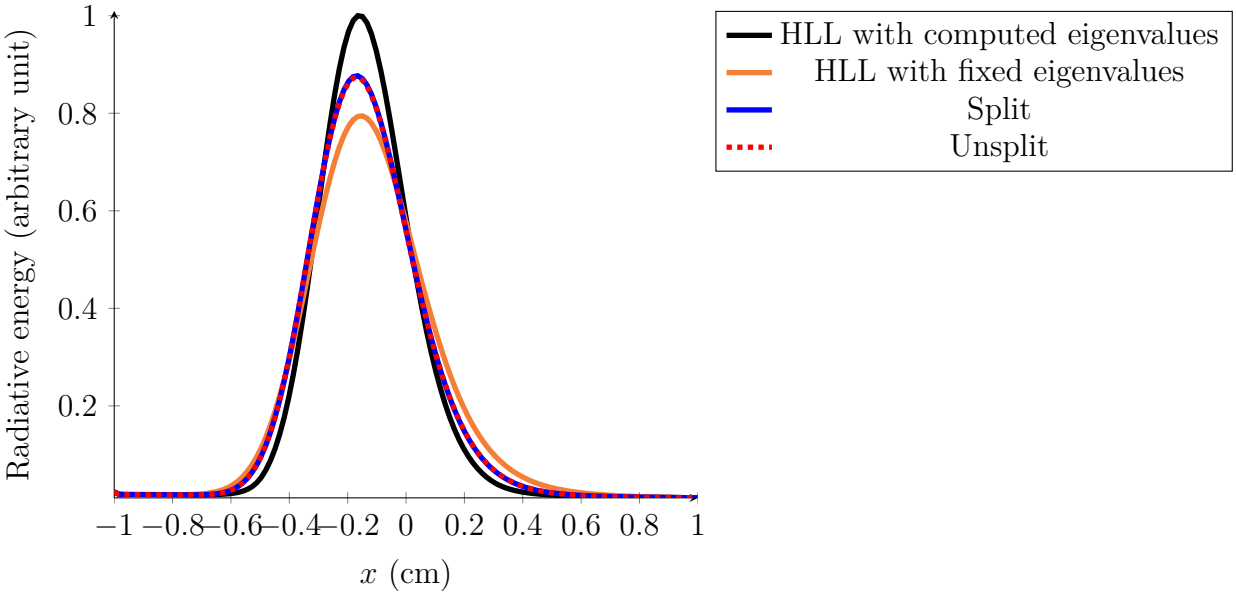


Figure 3.4 – Beam simulation. The figure shows a horizontal cut at the middle height with different solvers: HLL with computed and fixed eigenvalues (see eq. 2.2), split and unsplit solvers.

$\boldsymbol{F}_\mathrm{r} = \mathbf{0}$. At time $t = 0$, a beam with $T_\mathrm{g} = T_\mathrm{r} = 1\,000$ K is introduced with an angle of 45° at $x = -1$ and $y \in [-0.875, -0.75]$.

Because there is no opacity, the beam propagates in the vacuum, and we expect it to cross the box without dispersion. The direction of the beam is not along the mesh axis; we use this test to quantify the numerical diffusion. We compare both split and unsplit solvers with a standard HLL solver. The eigenvalues used in the HLL solver can be fixed to $\pm c$ or computed (see eq. 2.2). The impact of this choice has already been discussed in section 2.2.3.

Figure 3.4 shows a horizontal cut at the middle height once the steady state is reached. Using the HLL solver, the full width at half maximum is 22 (resp. 29) cells with computed (resp. fixed) eigenvalues. With both split and unsplit schemes, it is 26 cells.

Table 3.1 shows the computational time needed to reach the steady state with the different solvers: HLL with fixed and computed eigenvalues, and both split and unsplit schemes. Using fixed eigenvalues for the HLL solver allows us to save 40% of the computational time compared to using computed eigenvalues. To reduce computational time with computed eigenvalues, González et al. [2007] suggest tabulating them. They are computed once at the beginning of the simulation and the value needed is then interpolated. The split scheme requires solving two steps, the diffusion step followed by the transport step, hence the increase of 80% of computational cost compared to the HLL solver with computed eigenvalues. The unsplit scheme, which does not require solving these two steps, allows a decrease of 20% of the computational time compared to the HLL solver with computed eigenvalues and an increase of 40% compared to the HLL solver with fixed eigenvalues.

Overall, the unsplit scheme seems to be a good compromise between numerical diffusion and computational cost.

3.2.2 Numerical results with source terms

3.2.2.1 Marshak wave

Now that we have confirmed the good behavior of our schemes when there is no opacity, let us consider a one-dimensional Marshak wave, in the diffusive limit, to test their asymptotic behavior. We perform a test similar to section 2.2.1. Because we use time-explicit integration here, we cannot reach easily the same final time, therefore, we change some numerical values.

The interval $[0, 1]$ is discretized with 100 points. Initially, the medium is at equilibrium with the radiation: $T_0 = T_\mathrm{r} = 5.87 \times 10^8$ K, the radiative flux is $\boldsymbol{F}_\mathrm{r} = \mathbf{0}$, the opacity is constant, with $\sigma^\mathrm{a} = \sigma^\mathrm{s} = 9\,900$ m^{-1}, such that $\sigma^\mathrm{s}\Delta x = 100$ and $\rho c_v = 1.53$ J K^{-1} m^{-3}. At time $t = 0$, a source is lit at the left boundary with $T_\mathrm{r} = 5.87 \times 10^9$ K.

The results are shown in fig. 3.5 at time $t_f = 3 \times 10^{-10}$ s. We compare different solutions: a reference solution, given by a standard discretization of the diffusion equation 1.24, a standard HLL scheme, the split and unsplit schemes. We also discuss the impact of the source terms in the numerical fluxes β^* and q^*.

As shown in chapter 2, the solution obtained with the HLL solver is dominated by the numerical diffusion and does not capture the asymptotic behavior, the relative L^2 error between this solution and the reference solution is around 192%. Using eq. 3.13, without source terms in the numerical fluxes, leads to a relative L^2 error of 194% between the solutions obtained with both split and unsplit solvers and the reference

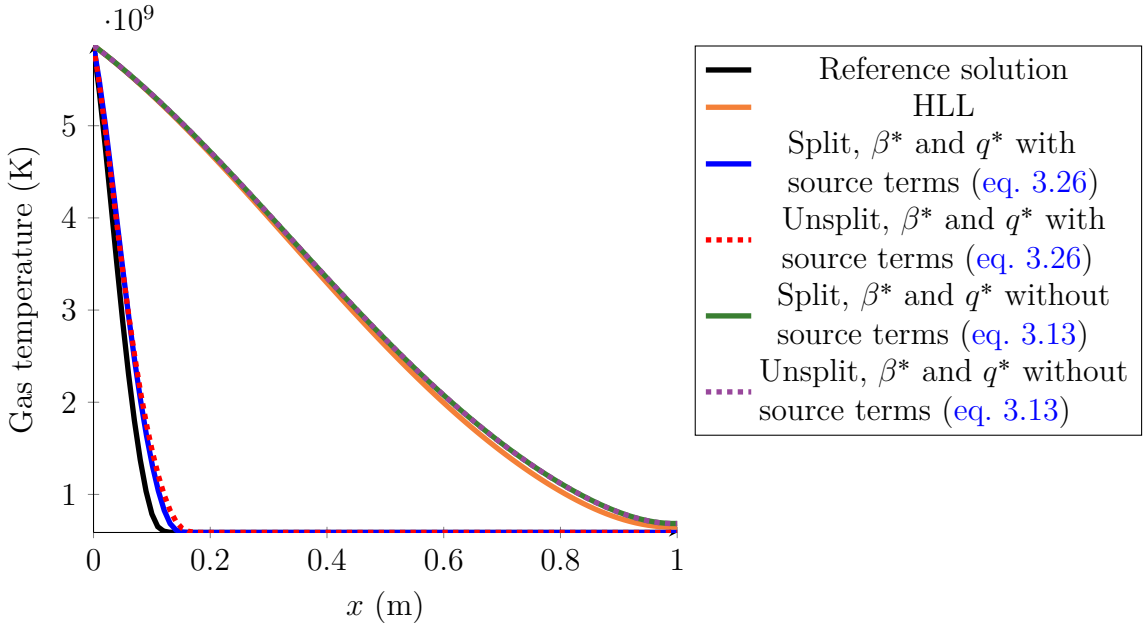


Figure 3.5 – Marshak wave simulation. This figure shows a snapshot of the gas temperature at time $t_f = 3 \times 10^{-10}$ s, with a reference solution and different solvers: a standard HLL scheme, the split and unsplit schemes, with and without the modification to take into account the source terms in β^* and q^*.

solution. These solutions are also dominated by numerical diffusion. To tackle this issue, pressure terms in the numerical fluxes are modified to take into account the source terms, as shown by eq. 3.26. With this choice, the relative L^2 error drops to 12.8% and 14.4% for the split and unsplit solvers, respectively. We recover numerically the result shown in section 3.F.

3.2.2.2 Steady state with a jump of opacity

In the previous cases, the opacity was constant; let us now consider a test with a jump of opacity, in the one-dimensional case, as done in section 2.2.2. We use this test to discuss the well-balanced behavior of the schemes.

The length of the computational domain is 1 cm, it is discretized with 100 cells. Initially, the medium is at equilibrium with the radiation: $T_0 = T_\mathrm{r} = 300$ K, the initial radiative flux is $\boldsymbol{F}_\mathbf{r} = \mathbf{0}$ and $\rho c_v = 1$ J K^{-1} cm^{-3}. The opacity is now a function of space: $\sigma^\mathrm{a}(x) = \sigma^\mathrm{s}(x) = \sigma(x)$, with

$$\sigma(x) = \begin{cases} 50 \text{ cm}^{-1} \text{ if } x < 0.5, \\ 0 \text{ if } x > 0.5. \end{cases} \tag{3.27}$$

At time $t = 0$, a source is lit at the left boundary, with $T_\mathrm{r} = 5\,000$ K.

Figure 3.6 shows the radiative flux at time $t_f = 10^{-6}$ s, once the steady state is reached. From section 2.1.1.2, when the steady state is reached, we expect the radiative flux to be constant in the box. Using a standard HLL scheme, the value reached in the second half of the box is 50% more than the expected value. As discussed in section 2.2.2, this is due to a lack of precision in the numerical integration of the source terms and can be tackled by using higher order integration of the source terms, which leads to an interface-centered discretization.

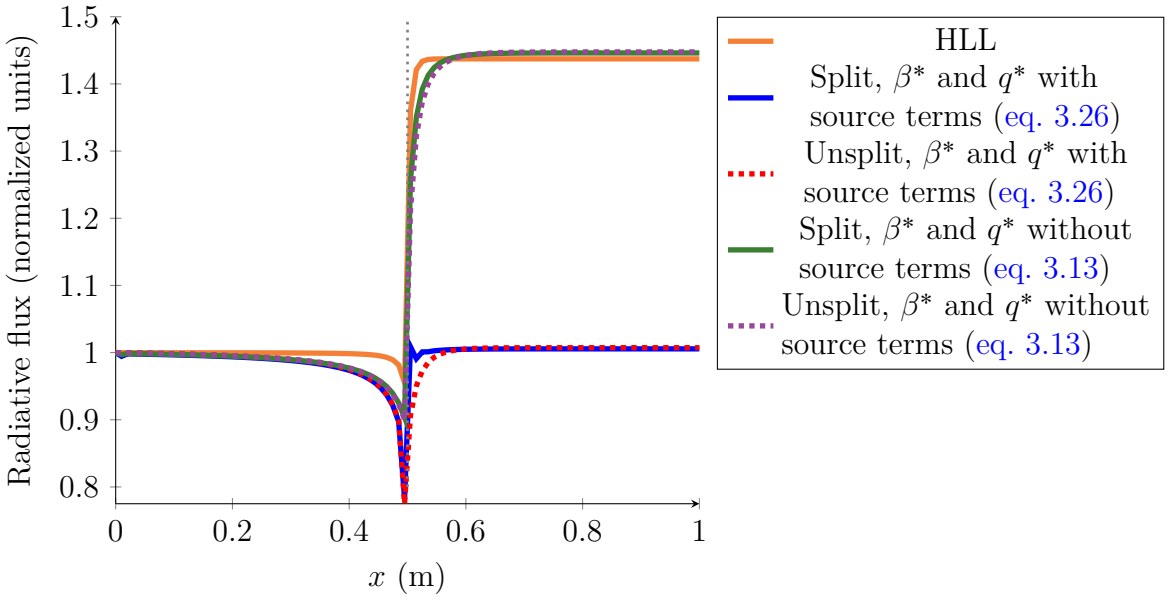
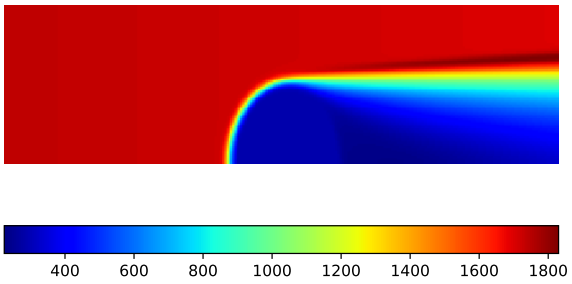


Figure 3.6 – Simulation of a steady state with a jump of opacity. The opacity is piecewise constant, a jump is located at $x = 0.5$ m (gray line). This figure shows a snapshot of the radiative flux at time $t_f = 10^{-6}$ s, with a standard HLL solver, and both the split and unsplit solvers.

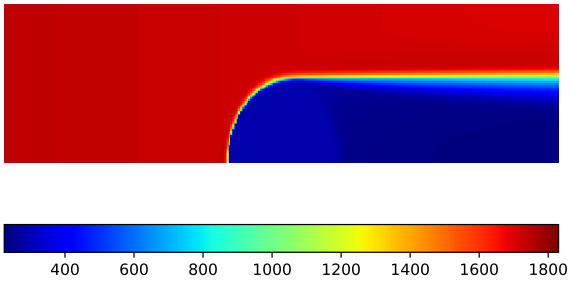
Using both split and unsplit solvers, when the numerical fluxes do not take into account the source terms (eq. 3.13), the radiative flux in the second half of the box is also 50% more than the expected value. However, using eq. 3.26, the error is much smaller. We only observe the same spurious peak located at the discontinuity of opacity as in section 2.2.2. Because we still use a splitting strategy to treat implicitly the source terms, this can be improved, either by using β^* to compute the source terms, as suggested by Buet and Despres [2008], or by using a discretization at the interfaces of the cells, as done in section 2.1.1.2. Both choices would require solving a (non)linear system while preserving the admissible states. An algorithm able to do that is explored in chapter 4.

3.2.2.3 Shadow test

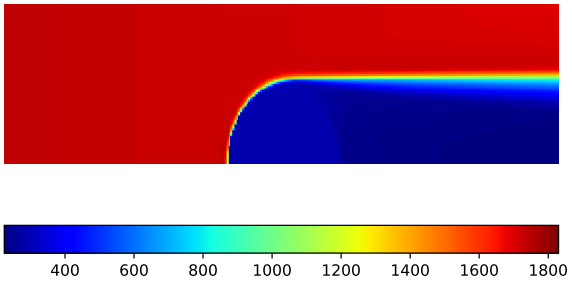
Finally, let us consider the same two-dimensional test as in section 2.2.4. Again, we do not discuss the properties of the HLL solver, nor the M_1 model, but we compare the results obtained with our all-regime scheme to those obtained with the HLL solver. Following González et al. [2007]; Hayes and Norman [2003], we consider a shadow test. The computational domain is a cylinder of length $L = 1$ cm and radius $R = 0.12$ cm. It is discretized with 280×80 cells. A spheroid clump is located at the center of the box, on the symmetric axis: $(z_c, r_c) = (0.5, 0)$. The extension of the clump is $(z_0, r_0) = (0.1, 0.06)$. We consider a homogeneous gas, with $\rho_0 = 1$ g cm^{-3}, expect for the clump with density $\rho_1 = 100\rho_0$. The boundary of the clump is smoothed: $\rho(z,r) = \rho_0 + \frac{\rho_1 - \rho_0}{1+\exp(\Delta)}$, with $\Delta = 10\left(\left(\frac{z-z_c}{z_0}\right)^2 + \left(\frac{r-r_c}{r_0}\right)^2 - 1\right)$. The opacity of the medium is $\sigma^a = \sigma^s = \sigma_0 \left(\frac{T}{T_0}\right)^{-3.5} \left(\frac{\rho}{\rho_0}\right)^2$, with $\sigma_0 = 0.1$ cm^{-1}. Initially, the medium is at equilibrium with the radiation, with $T_0 = T_r = 290$ K. At time $t = 0$, a source



(a) HLL



(b) Split



(c) Unsplit

Figure 3.7 – Shadow simulation, showing snapshots of radiative temperature at time $t_f = 6 \times 10^{-11}$ s with different solvers: HLL, split solver and unsplit solver.

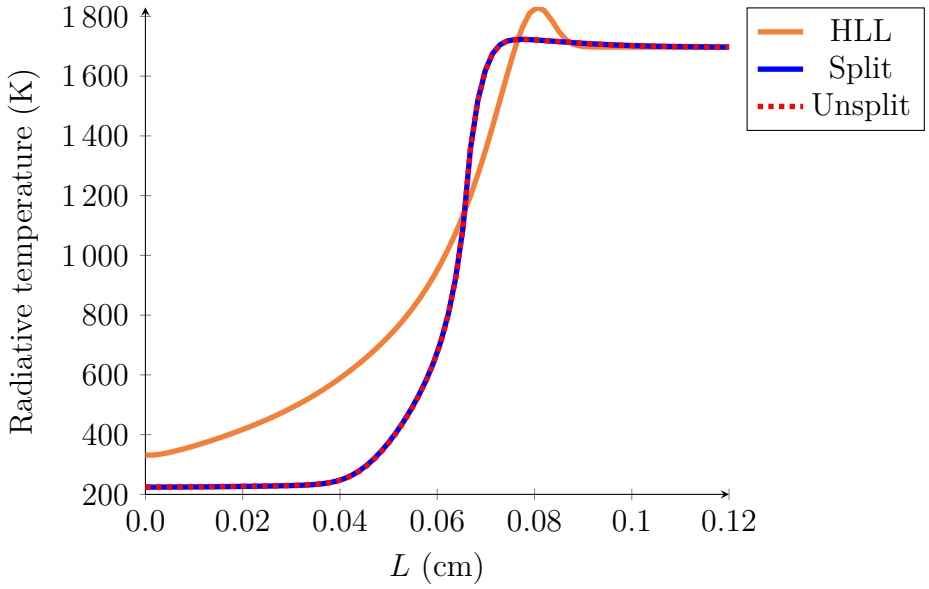


Figure 3.8 – Shadow simulation, showing the radial profiles of the radiative temperature at time $t_f = 6 \times 10^{-11}$ s with different solves.

is lit at the left boundary with $T_r = 1\ 740$ K and the reduced flux, aligned with the cylinder, is set to $f = 1$.

Figure 3.7 shows the radiative temperature at the final time $t_f = 6 \times 10^{-11}$ s with different solvers: a classical HLL scheme, the split and unsplit schemes. Because of the high opacity in the clump, the light does not cross it, and we expect the shadow behind it to remain stable.

To compare our results with González et al. [2007]; Hayes and Norman [2003], fig. 3.8 shows the radial profile of radiative temperature at the right boundary. Using both split and unsplit solvers, the shadow is better preserved than using the HLL solver. The solution obtained with these all-regime solvers does not overheat, unlike the one obtained with the HLL solver (see the discussion about the modification of the eigenvalues in section 2.2.4). As discussed in the previous tests, both split and unsplit solvers perform similarly. This all-regime strategy allows keeping numerical diffusion under control, at least when the radiation propagates along the mesh axis.

3.3 Discussion and conclusion

In this chapter, we have presented two numerical schemes. The first one is based on a diffusion-transport splitting, similar to the acoustic-transport splitting for Euler equations. However, this leads to a scheme of stencil two, with a high computational cost. The second one is an approximate Riemann solver and relies on a relaxation. Both solvers perform well on classical test cases. Nevertheless, they present some limitations.

3.3.1 Well-balanced discretization of the source terms

Numerical fluxes proposed by Buet and Despres [2008] take into account the source terms and are designed to capture correctly the asymptotic regime. For performance reasons, the source terms are still taken into the center of the cells. As shown in chapter 2, this does not lead to a well-balanced modification of the source terms. It would require computing the source terms at the interfaces of the cells, which calls for the resolution of a linear or nonlinear system, depending on the choice of discretization. However, with source terms taken into account in the numerical fluxes, the steady state with a constant radiative flux in the presence of a discontinuity of opacity is better captured than without source terms in the fluxes.

3.3.2 Time implicit integration

In order to study astrophysical problems, coupling to hydrodynamics is essential. Because of the difference between the timescale of the propagation of the fluid and the photons, solving the radiative transfer requires a time implicit solver. Buet and Despres [2008] show that the schemes proposed in this chapter preserve the admissible states $E_r > 0$ and $f \leqslant 1$. Using a Newton-Raphson algorithm to solve the nonlinear system coming from a time-implicit discretization with large time steps might not preserve these admissible states [Buet and Despres, 2006]. In chapter 4, we explore another strategy to solve this nonlinear system while still preserving the admissible states, based on the work of Pichard [2016].

3.4 Bibliography

Christophe Buet and Bruno Despres. A gas dynamics scheme for a two moments model of radiative transfer. 11 2008. 59, 62, 64, 65, 71, 73, 77, 80, 81, 85

Christophe Chalons, Pierre Kestener, Samuel Kokh, and Maxime Stauffert. A large time-step and well-balanced Lagrange-Projection type scheme for the shallow-water equations. July 2016. URL https://hal.archives-ouvertes.fr/hal-01297043. working paper or preprint. 59, 60, 62

M. González, E. Audit, and P. Huynh. Heracles: a three-dimensional radiation hydrodynamics code. *A&A*, 464(2):429–435, 2007. doi: 10.1051/0004-6361:20065486. URL https://doi.org/10.1051/0004-6361:20065486. 61, 66, 69, 71, 73, 76

Thomas Padioleau, Pascal Tremblin, Edouard Audit, Pierre Kestener, and Samuel Kokh. A high-performance and portable all-mach regime flow solver code with well-balanced gravity. application to compressible convection. *The Astrophysical Journal*, 875(2):128, apr 2019. doi: 10.3847/1538-4357/ab0f2c. URL https://doi.org/10.3847/1538-4357/ab0f2c. 63

F Blachère and R Turpault. An admissibility and asymptotic-preserving scheme for systems of conservation laws with source term on 2d unstructured meshes. *Journal of computational physics*, 315:98–123, 2016. ISSN 0021-9991. 65

S. Richling, E. Meinköhn, N. Kryzhevoi, and G. Kanschat. Radiative transfer with finite elements - i. basic method and tests. *A&A*, 380(2):776–788, 2001. doi: 10.1051/0004-6361:20011411. URL https://doi.org/10.1051/0004-6361:20011411. 66

John C. Hayes and Michael L. Norman. Beyond flux-limited diffusion: Parallel algorithms for multidimensional radiation hydrodynamics. *The Astrophysical Journal Supplement Series*, 147(1):197–220, jul 2003. doi: 10.1086/374658. URL https://doi.org/10.1086/374658. 71, 73

Christophe Buet and Bruno Despres. Asymptotic preserving and positive schemes for radiation hydrodynamics. *Journal of Computational Physics*, 215(2):717 – 740, 2006. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2005.11.011. URL http://www.sciencedirect.com/science/article/pii/S0021999105005127. 73

Teddy Pichard. *Mathematical modelling for dose depositon in photontherapy*. PhD thesis, 2016. URL http://www.theses.fr/2016BORD0177. Thèse de doctorat dirigée par Dubroca, Bruno et Frank, Martin Mathematiques appliquees et calcul scientifique Bordeaux 2016. 73

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, January 2017. ISSN 2376-5992. doi: 10.7717/peerj-cs.103. URL https://doi.org/10.7717/peerj-cs.103. 76

François Bouchut, Christian Klingenberg, and Knut Waagan. A multiwave approximate riemann solver for ideal mhd based on relaxation ii: numerical implementation with 3 and 5 waves. *Numerische Mathematik*, 115(4):647–679, 2010. ISSN 0029-599X. 77

François Bouchut. *Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws and Well-Balanced Schemes for Sources*. Frontiers in Mathematics. 1st ed. 2004.. edition, 2004. ISBN 3-7643-6665-6. 79

Eleuterio Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*, pages 163–212. 03 2009. ISBN 978-3-540-25202-3. doi: 10.1007/b79761_5. 79

Christophe Berthon and Rodolphe Turpault. Asymptotic preserving hll schemes. *Numerical Methods for Partial Differential Equations*, 27:1396 – 1422, 11 2011. doi: 10.1002/num.20586. 91

3.A Equation of evolution for pressure and velocity in the diffusion step

We derive the equations of evolution for the velocity and the pressure. Let us begin with the pressure q. From chapter 1, we have $q = \frac{c^2-\beta^2}{3c^2+\beta^2}E_\mathrm{r} = \frac{1-\chi}{2}E_\mathrm{r}$, where $\chi = \frac{3+4f^2}{5+2\sqrt{4-3f^2}}$ is the Eddington factor. Because $\boldsymbol{f} = \frac{\boldsymbol{F}_\mathrm{r}}{cE_\mathrm{r}}$, one has

$$\partial_t q = \frac{\partial q}{\partial E_\mathrm{r}}\partial_t E_\mathrm{r} + \nabla_{\boldsymbol{F}_\mathbf{r}} q \cdot \partial_t \boldsymbol{F}_\mathbf{r}, \tag{3.28}$$

where $\nabla_{\boldsymbol{F}_\mathbf{r}} q = \left(\frac{\partial q}{\partial F_\mathrm{r}^x}, \frac{\partial q}{\partial F_\mathrm{r}^y}, \frac{\partial q}{\partial F_\mathrm{r}^z}\right)^T$ and $\boldsymbol{F}_\mathbf{r} = (F_\mathrm{r}^x, F_\mathrm{r}^y, F_\mathrm{r}^z)^T$. Let us now compute each term individually:

$$\frac{\partial q}{\partial E_\mathrm{r}} = -\frac{E_\mathrm{r}}{2}\frac{\partial \chi}{\partial E_\mathrm{r}} + \frac{1-\chi}{2}. \tag{3.29}$$

Because

$$\frac{\partial \chi}{\partial E_\mathrm{r}} = \frac{\partial f^2}{\partial E_\mathrm{r}}\frac{\mathrm{d}\chi}{\mathrm{d}f^2} = -\frac{(3c^2+\beta^2)\,f^2}{(3c^2-\beta^2)\,E_\mathrm{r}}, \tag{3.30}$$

one has

$$\frac{\partial q}{\partial E_\mathrm{r}} = \frac{c^2+\beta^2}{3c^2-\beta^2}. \tag{3.31}$$

Likewise, one can show that

$$\frac{\partial q}{\partial F_\mathrm{r}^x} = -\frac{F_\mathrm{r}^x}{2c^2E_\mathrm{r}}\frac{3c^2+\beta^2}{3c^2-\beta^2} = -\frac{2\beta_x}{3c^2-\beta^2}, \tag{3.32}$$

with $\boldsymbol{\beta} = (\beta_x, \beta_y, \beta_z)^T$. Similar computations can be done for the x and y coordinates. Therefore,

$$\nabla_{\boldsymbol{F}_\mathbf{r}} q = -\frac{2}{3c^2-\beta^2}\boldsymbol{\beta}. \tag{3.33}$$

$\partial_t E_\mathrm{r}$ and $\partial_t \boldsymbol{F}_\mathbf{r}$ are given by the diffusion system (eq. 3.6). Therefore,

$$\partial_t q + \frac{4c^2 q}{3c^2-\beta^2}\nabla\cdot\boldsymbol{\beta} - \frac{c^2-\beta^2}{3c^2-\beta^2}\boldsymbol{\beta}\cdot\nabla q = 0. \tag{3.34}$$

Using the fact that $\boldsymbol{F}_\mathrm{r} = (E_\mathrm{r} + q)\,\boldsymbol{\beta}$, one has

$$\partial_t \boldsymbol{\beta} = \partial_t \left(\frac{\boldsymbol{F}_\mathrm{r}}{E_\mathrm{r} + q} \right). \tag{3.35}$$

By mean of the chain rule derivative, eqs. 3.6 and 3.34, one can show that

$$\partial_t \boldsymbol{\beta} - \frac{c^2 - \beta^2}{3c^2 - \beta^2} \boldsymbol{\beta} \nabla \cdot \boldsymbol{\beta} + \frac{c^2 - \beta^2}{4q} \nabla q - \frac{c^2 - \beta^2}{2q\,(3c^2 - \beta^2)} \boldsymbol{\beta}\,(\boldsymbol{\beta} \cdot \nabla q) = 0. \tag{3.36}$$

3.B Eigenstructure of the diffusion system

For the sake of simplicity, we focus on the one-dimensional case. We recall that the eigenvalues of the M_1 model are given by [González et al., 2007]:

$$\begin{cases} \lambda^- = c \dfrac{n_x \chi' - \sqrt{\chi'^2 + 4\chi - 4\chi' f}}{2} \\ \lambda^0 = c^2 \dfrac{3\chi - 1}{2} \dfrac{E_\mathrm{r}}{F_\mathrm{r}} \\ \lambda^+ = c \dfrac{n_x \chi' + \sqrt{\chi'^2 + 4\chi - 4\chi' f}}{2}, \end{cases} \tag{3.37}$$

where n_x is the x-coordinate of $\boldsymbol{n} = \frac{\boldsymbol{f}}{f}$ and $\chi' = \frac{\mathrm{d}\chi}{\mathrm{d}f} = \frac{2f}{\sqrt{4-3f^2}}$.

Let us write eqs. 3.6 and 3.8 as

$$\partial_t \boldsymbol{U} + \mathbb{A}\,(\boldsymbol{U})\,\partial_x \boldsymbol{U} = 0, \tag{3.38}$$

with $\boldsymbol{U} = (\varrho, \varrho v, \varrho e, \beta, q)^T$ and

$$\mathbb{A}\,(\boldsymbol{U}) = \begin{pmatrix} 0 & 0 & 0 & \varrho & 0 \\ 0 & 0 & 0 & \varrho v & c^2 \\ 0 & 0 & 0 & \frac{4c^2 q}{c^2-\beta^2} & \beta \\ 0 & 0 & 0 & -\frac{c^2-\beta^2}{3c^2-\beta^2}\beta & \frac{3}{4q}\frac{(c^2-\beta^2)^2}{3c^2-\beta^2} \\ 0 & 0 & 0 & \frac{4c^2 q}{3c^2-\beta^2} & -\frac{c^2-\beta^2}{3c^2-\beta^2}\beta \end{pmatrix}. \tag{3.39}$$

0 is eigenvalue with multiplicity 3. Let us write $\Lambda^\pm$ the two remaining eigenvalues. They are the eigenvalues of the 2×2 matrix

$$\mathbb{B}\,(\boldsymbol{U}) = \begin{pmatrix} -\frac{c^2-\beta^2}{3c^2-\beta^2}\beta & \frac{3}{4q}\frac{(c^2-\beta^2)^2}{3c^2-\beta^2} \\ \frac{4c^2 q}{3c^2-\beta^2} & -\frac{c^2-\beta^2}{3c^2-\beta^2}\beta \end{pmatrix}. \tag{3.40}$$

Its characteristic polynomial is $\Lambda^2 - \frac{2\beta(c^2-\beta^2)}{3c^2-\beta^2}\Lambda - \frac{c^2-\beta}{3c^2-\beta^2} = 0$. Its roots are $\Lambda^\pm = \frac{c^2-\beta^2}{\beta \pm \sqrt{3}c}$. Using, for example, the symbolic computation Python library SymPy [Meurer et al., 2017], one can check that $\Lambda^\pm = \lambda^\pm - \lambda^0$.

The three eigenvectors associated to the eigenvalue 0 are $(1,0,0,0,0)^T$, $(0,1,0,0,0)^T$ and $(0,0,1,0,0)^T$. The field associated to the stationary wave is linearly degenerated. One can check that $\boldsymbol{R}^+$ (resp. $\boldsymbol{R}^-$) is an eigenvector associated to the eigenvalue Λ^+ (resp. Λ^-), with

$$\boldsymbol{R}^+ = \begin{pmatrix} \frac{\sqrt{3}\varrho(\sqrt{3}c-\beta)}{4cq} \\ -\frac{c(c-\sqrt{3}\beta)(\sqrt{3}c-\beta)}{c^2-\beta^2} \\ \frac{(\sqrt{3}c-\beta)(3c^2-\beta^2)}{(\sqrt{3}c+\beta)(c^2-\beta^2)} \\ -\frac{\sqrt{3}(c^2-\beta^2)}{4cq} \\ 1 \end{pmatrix}, \quad \boldsymbol{R}^- = \begin{pmatrix} \frac{\sqrt{3}\varrho(\sqrt{3}c+\beta)}{4cq} \\ \frac{c(c+\sqrt{3}\beta)(\sqrt{3}c+\beta)}{c^2-\beta^2} \\ \frac{(\sqrt{3}c+\beta)(3c^2-\beta^2)}{(\sqrt{3}c-\beta)(c^2-\beta^2)} \\ \frac{\sqrt{3}(c^2-\beta^2)}{4cq} \\ 1 \end{pmatrix}. \tag{3.41}$$

The fields associated to $\Lambda^\pm$ are genuinely nonlinear under the condition $|\beta| < c$.

If all the waves were linearly degenerated, the Riemann problem could be solved exactly. Here, we use the result of Buet and Despres [2008].

3.C Derivation of β^* and q^*

For the sake of simplicity, we focus on the one-dimensional case. We do not present the extension to higher dimensions. Equations 3.6 and 3.8 become

$$\begin{cases} \partial_t \varrho + \varrho \partial_x \beta = 0 \\ \partial_t (\varrho v) + \varrho v \partial_x \beta + c^2 \partial_x q = 0 \\ \partial_t (\varrho e) + (\varrho e + q) \partial_x \beta + \beta \partial_x q = 0, \\ \partial_t \beta - \dfrac{c^2-\beta^2}{3c^2-\beta^2} \beta \partial_x \beta + \dfrac{3}{4q} \dfrac{(c^2-\beta^2)^2}{3c^2-\beta^2} \partial_x q = 0 \\ \partial_t q + \dfrac{4c^2 q}{3c^2-\beta^2} \partial_x \beta - \dfrac{c^2-\beta^2}{3c^2-\beta^2} \beta \partial_x q = 0. \end{cases} \tag{3.42}$$

Let us recall from section 3.B that the eigenvalues of eq. 3.42 are 0 with multiplicity 3 and $\Lambda^\pm = \frac{c^2-\beta^2}{\beta \pm \sqrt{3}c}$ with multiplicity 1. After some algebra, one can show that the two last equations from eq. 3.42 rewrite

$$\begin{cases} \partial_t \beta + \dfrac{\Lambda^+ + \Lambda^-}{2} \partial_x \beta + \dfrac{\sqrt{3}\,(c^2-\beta^2)}{4cq} \dfrac{\Lambda^+ - \Lambda^-}{2} \partial_x q = 0 \\ \partial_t q + \dfrac{4cq}{\sqrt{3}\,(c^2-\beta^2)} \dfrac{\Lambda^+ - \Lambda^-}{2} \partial_x \beta + \dfrac{\Lambda^+ + \Lambda^-}{2} \partial_x q = 0. \end{cases} \tag{3.43}$$

3.C.1 Relaxed set of equations

We now follow Bouchut et al. [2010]. Let us first introduce a relaxation parameter λ. We also introduce an approximation of the velocity $b \approx \beta$, pressure $\Pi \approx q$, and eigenvalues $l^\pm \approx \Lambda^\pm$. Finally, we introduce a coefficient a which is an approximation of $\frac{4cq}{\sqrt{3}(c^2-\beta^2)}$. We use these approximations in the flux terms:

$$\begin{cases} \partial_t \varrho + \varrho \partial_x b = 0 \\ \partial_t (\varrho v) + \varrho v \partial_x b + c^2 \partial_x \Pi = 0 \\ \partial_t (\varrho e) + (\varrho e + \Pi) \partial_x b + b \partial_x \Pi = 0 \\ \partial_t b + \dfrac{l^+ + l^-}{2} \partial_x b + \dfrac{l^+ - l^-}{2a} \partial_x \Pi = \lambda (b - \beta) \\ \partial_t \Pi + \dfrac{a\,(l^+ - l^-)}{2} \partial_x b + \dfrac{l^+ + l^-}{2} \partial_x \Pi = \lambda (\Pi - q). \end{cases} \tag{3.44}$$

We add the corresponding evolution equations to the system:

$$\begin{cases} \partial_t l^+ = \lambda \left(l^+ - \Lambda^+ \right) \\ \partial_t l^- = \lambda \left(l^- - \Lambda^- \right) \\ \partial_t a = \lambda \left(a - \dfrac{4cq}{\sqrt{3}\left(c^2 - \beta^2\right)} \right). \end{cases} \tag{3.45}$$

We formally recover eq. 3.42 when $\lambda \to \infty$. We mimic this behavior by setting λ to 0 in eq. 3.45. More details about this relaxation method can be found in section 3.E.1. We aim at building an exact solver for eqs. 3.44 and 3.45. This solver will be an approximate solver for eq. 3.42.

3.C.2 Eigenstructure

Let us now study the eigenstructure of eqs. 3.44 and 3.45. We write $\boldsymbol{W} = (\varrho, \varrho v, \varrho e, b, \Pi, l^+, l^-, a)^T$, then eqs. 3.44 and 3.45 can be written in the matrix form

$$\partial_t \boldsymbol{W} + \mathbb{M}(\boldsymbol{W})\, \partial_x \boldsymbol{W} = 0, \tag{3.46}$$

where

$$\mathbb{M}(\boldsymbol{W}) = \begin{pmatrix} 0 & 0 & 0 & \varrho & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \varrho v & c^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \varrho e + \Pi & b & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{l^+ + l^-}{2} & \frac{l^+ - l^-}{2a} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{a(l^+ - l^-)}{2} & \frac{l^+ + l^-}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3.47}$$

After some algebra, the eigenvalues of $\mathbb{M}$ are 0 with multiplicity 6 and $l^\pm$ with multiplicity 1. Let us notice that the eigenvalues of eqs. 3.44 and 3.45 are equal to the eigenvalues of the diffusion step whenever $l^\pm = \Lambda^\pm$. The eigenvectors associated with the eigenvalue 0 are

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \tag{3.48}$$

$$\overline{\boldsymbol{R}^+} = \begin{pmatrix} \frac{\varrho}{al^+} \\ \frac{1}{l^+}\left(\frac{\varrho v}{a} + c^2\right) \\ \frac{1}{l^+}\left(\frac{\varrho e + \Pi}{a} + b\right) \\ \frac{1}{a} \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ and } \overline{\boldsymbol{R}^-} = \begin{pmatrix} \frac{\varrho}{al^-} \\ \frac{1}{l^-}\left(\frac{\varrho v}{a} - c^2\right) \\ \frac{1}{l^-}\left(\frac{\varrho e + \Pi}{a} - b\right) \\ \frac{1}{a} \\ -1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.49}$$

are the eigenvectors associated with the eigenvalues $l^\pm$.

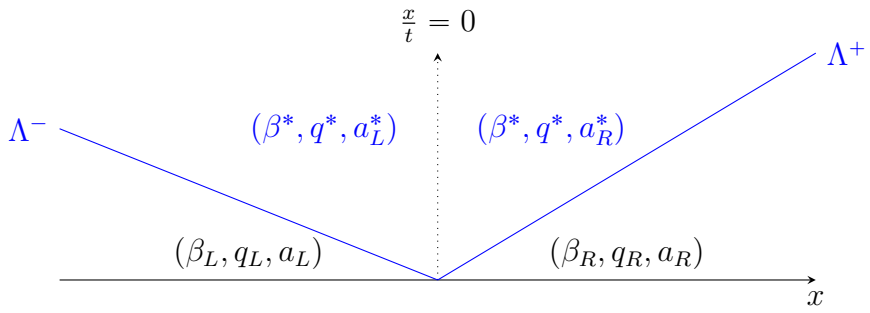


Figure 3.9 – Solution of the Riemann problem for b, Π, and a.

3.C.3 Riemann invariants for a hyperbolic system

Let us recall some notions defined by Bouchut [2004]. We consider a hyperbolic system of the form

$$\partial_t \boldsymbol{U} + \mathbb{A}(\boldsymbol{U}) \partial_x \boldsymbol{U} = 0, \tag{3.50}$$

where $\boldsymbol{U} \in \mathbb{R}^d$ and $\mathbb{A}$ is a $d \times d$ matrix. Because the system is hyperbolic, $\mathbb{A}$ has d real eigenvalues, λ_j, and d independent eigenvectors, $\boldsymbol{R}_j = \left(R_j^{(1)}, \cdots, R_j^{(d)}\right)^T$, for $j = 1, \cdots, d$.

A *weak j-Riemann invariant* is a scalar function $w(\boldsymbol{U})$ constant along the characteristic curves. In particular, from eq. 2.123 in Toro [2009],

$$\frac{\mathrm{d}U^{(k)}}{R_j^{(k)}} \tag{3.51}$$

is constant across the wave λ_j, for all $k = 1, \cdots, d$. For example, if $d = 2$, $\boldsymbol{U} = \left(U^{(1)}, U^{(2)}\right)^T$, $\boldsymbol{R}_1 = \left(R_1^{(1)}, R_1^{(2)}\right)^T$ and $\boldsymbol{R}_2 = \left(R_2^{(1)}, R_2^{(2)}\right)^T$. Then,

$$\frac{\mathrm{d}U^{(1)}}{R_1^{(1)}} = \frac{\mathrm{d}U^{(2)}}{R_1^{(2)}} \tag{3.52}$$

across the wave λ_1, and

$$\frac{\mathrm{d}U^{(1)}}{R_2^{(1)}} = \frac{\mathrm{d}U^{(2)}}{R_2^{(2)}} \tag{3.53}$$

across the wave λ_2.

A *strong j-Riemann invariant* is a scalar function $w(\boldsymbol{U})$ characterized by

$$\partial_t w(\boldsymbol{U}) + \lambda_j(\boldsymbol{U}) \partial_x w(\boldsymbol{U}) = 0. \tag{3.54}$$

Bouchut [2004] shows that $w(\boldsymbol{U})$ is a strong j-Riemann invariant if and only if $w(\boldsymbol{U})$ is a weak k-Riemann invariant, for $k = 1, \cdots, j-1, j+1, \cdots, d$.

3.C.4 Riemann invariants for the relaxed system

Figure 3.9 shows the wave structure for the system made of eqs. 3.44 and 3.45. We consider only one intermediate state for the velocity b^* and the pressure Π^*, and two intermediate states for a, a_L^* and a_R^*.

Using eqs. 3.49 and 3.51, we have

$$a \mathrm{d}b = \mathrm{d}\Pi \tag{3.55}$$

across the wave l^-. From the last equation of eq. 3.45, a is a strong Riemann invariant associated with the central wave. Therefore, a is a weak Riemann invariant associated with both waves l^- and l^+. Then, $a_L^* = a_L$ and $a_R^* = a_R$. After integration, eq. 3.55 becomes

$$\frac{4c}{\sqrt{3}}\frac{\Pi_L}{c^2 - b_L^2}(b_L - b^*) + (\Pi_L - \Pi^*) = 0. \tag{3.56}$$

Similar computations across the wave l^+ give

$$\frac{4c}{\sqrt{3}}\frac{\Pi_R}{c^2 - b_R^2}(b_R - b^*) - (\Pi_R - \Pi^*) = 0. \tag{3.57}$$

Equations 3.56 and 3.57 result in an exact solver for the relaxed system made of eqs. 3.44 and 3.45. We obtain an approximate solver for the original system, eq. 3.42. Using $q = \frac{c^2-\beta^2}{3c^2+\beta^2}E$, we recover the same jump relations as Buet and Despres [2008] (see also eq. 3.24). One can show that β^* and q^* given by eq. 3.13 are the solution of the system made of eqs. 3.56 and 3.57.

3.D Admissible states

Let us focus on the case without source terms. We follow Buet and Despres [2008]. To prove that the schemes developed in the chapter preserve the admissible states, we first prove that they are entropic. As done for the radiative energy and the radiative flux, let us introduce a specific entropy s such that $S_\mathrm{r} = \varrho s$.

3.D.1 Diffusion step

Using eq. 1.49, we have

$$\theta(t)\frac{\mathrm{d}S_\mathrm{r}}{\mathrm{d}t}(t) - \frac{\mathrm{d}E_\mathrm{r}}{\mathrm{d}t}(t) + \frac{\beta(t)}{c^2}\frac{\mathrm{d}F_\mathrm{r}}{\mathrm{d}t}(t) = 0. \tag{3.58}$$

Because $\frac{\mathrm{d}S_\mathrm{r}}{\mathrm{d}t}(t) = \frac{\mathrm{d}(\varrho s)}{\mathrm{d}t}(t)$ and $\tau = \frac{1}{\varrho}$, eq. 3.58 becomes

$$\theta(t)\frac{\mathrm{d}s}{\mathrm{d}t}(t) = \frac{\mathrm{d}e}{\mathrm{d}t}(t) - \frac{\beta(t)}{c^2}\frac{\mathrm{d}v}{\mathrm{d}t}(t) + \left(\theta(t)S_\mathrm{r}(t) - E_\mathrm{r}(t) + \frac{\beta(t)}{c^2}F_\mathrm{r}(t)\right)\frac{\mathrm{d}\tau}{\mathrm{d}t}(t). \tag{3.59}$$

Let us recall some results obtained in section 1.B:

$$\begin{cases} E_\mathrm{r} = c^4\dfrac{3c^2+\beta^2}{3\left(c^2-\beta^2\right)^3}a_\mathrm{r}\theta^4 \\ F_\mathrm{r} = \dfrac{4}{3}\dfrac{c^6 a_\mathrm{r}\theta^4}{\left(c^2-\beta^2\right)^3}\beta \\ S_\mathrm{r} = \dfrac{4c^4 a_\mathrm{r}\theta^3}{3\left(c^2-\beta^2\right)^2}. \end{cases} \tag{3.60}$$

Using eq. 3.60 and $q = \frac{c^2-\beta^2}{3c^2+\beta^2}E_\mathrm{r}$, eq. 3.59 becomes

$$\theta(t)\frac{\mathrm{d}s}{\mathrm{d}t}(t) = \frac{\mathrm{d}e}{\mathrm{d}t}(t) - \frac{\beta(t)}{c^2}\frac{\mathrm{d}v}{\mathrm{d}t}(t) + q(t)\frac{\mathrm{d}\tau}{\mathrm{d}t}(t). \tag{3.61}$$

Let us also recall eq. 1.82:

$$\partial_t S_\mathrm{r} + \partial_x \left(\beta S_\mathrm{r}\right) \geqslant 0, \tag{3.62}$$

with equality for smooth functions. This equation can be split into the diffusion step

$$\partial_t S_\mathrm{r} + S_\mathrm{r} \partial_x \beta \geqslant 0, \tag{3.63}$$

followed by the transport step

$$\partial_t S_\mathrm{r} + \beta \partial_x S_\mathrm{r} \geqslant 0. \tag{3.64}$$

Using $S_\mathrm{r} = \varrho s$ and the first equation of eq. 3.6, eq. 3.63 rewrites

$$\partial_t s \geqslant 0. \tag{3.65}$$

Following Buet and Despres [2008], let us now prove that the entropy s is non decreasing in the semi-discrete case for the diffusion step. The semi-discrete diffusion step is

$$\begin{cases} \Delta m_i \dfrac{\mathrm{d}\tau_i}{\mathrm{d}t}(t) = \beta^*_{i+\frac{1}{2}}(t) - \beta^*_{i-\frac{1}{2}}(t) \\ \Delta m_i \dfrac{\mathrm{d}v_i}{\mathrm{d}t}(t) = -c^2 \left(q^*_{i+\frac{1}{2}}(t) - q^*_{i-\frac{1}{2}}(t) \right) \\ \Delta m_i \dfrac{\mathrm{d}e_i}{\mathrm{d}t}(t) = -\left(q^*_{i+\frac{1}{2}}(t)\beta^*_{i+\frac{1}{2}}(t) - q^*_{i-\frac{1}{2}}(t)\beta^*_{i-\frac{1}{2}}(t) \right). \end{cases} \tag{3.66}$$

Using eq. 3.66 in eq. 3.61, and after some algebra, we have

$$\begin{aligned} \Delta m_i \theta_i(t) \frac{\mathrm{d}s_i}{\mathrm{d}t}(t) &= \left(-q^*_{i+\frac{1}{2}}(t)\beta^*_{i+\frac{1}{2}}(t) + \beta_i(t) q^*_{i+\frac{1}{2}}(t) + q_i(t)\beta^*_{i+\frac{1}{2}}(t) - q_i(t)\beta_i(t) \right) \\ &\quad - \left(-q^*_{i-\frac{1}{2}}(t)\beta^*_{i-\frac{1}{2}}(t) + \beta_i(t) q^*_{i-\frac{1}{2}}(t) + q_i(t)\beta^*_{i-\frac{1}{2}}(t) - q_i(t)\beta_i(t) \right) \\ &= \left(q^*_{i+\frac{1}{2}}(t) - q_i(t) \right) \left(\beta_i(t) - \beta^*_{i+\frac{1}{2}}(t) \right) - \left(q^*_{i-\frac{1}{2}}(t) - q_i(t) \right) \left(\beta_i(t) - \beta^*_{i-\frac{1}{2}}(t) \right). \end{aligned} \tag{3.67}$$

From the jump relations eq. 3.24, one can show that $q^*_{i+\frac{1}{2}}(t) - q_i(t)$ and $\beta_i(t) - \beta^*_{i+\frac{1}{2}}(t)$ have the same sign, while $q^*_{i-\frac{1}{2}}(t) - q_i(t)$ and $\beta_i(t) - \beta^*_{i-\frac{1}{2}}(t)$ have the opposite sign. One can deduce that $\Delta m_i \theta_i(t) \frac{\mathrm{d}s_i}{\mathrm{d}t}(t)$ is non negative. Because Δm_i and $\theta_i(t)$ are non negative, one has $\frac{\mathrm{d}s_i}{\mathrm{d}t}(t) \geqslant 0$ and the semi-discrete scheme for the diffusion step is entropic.

Let us notice that, because $F_\mathrm{r} = \left(E_\mathrm{r} + q\right)\beta$ and $q = \frac{c^2-\beta^2}{3c^2+\beta^2} E_\mathrm{r}$, then $\frac{|F_\mathrm{r}|}{cE_\mathrm{r}} = \frac{4c}{3c^2+\beta^2}|\beta|$. One can show that $\frac{|F_\mathrm{r}|}{cE_\mathrm{r}} \leqslant 1$ if and only if $|\beta| \leqslant c$ or $|\beta| \geqslant 3c$. For physical reasons, we only consider the case $|\beta| \leqslant c$.

From eq. 3.60, one can show that

$$S_\mathrm{r} = \frac{4}{3} c a_\mathrm{r}^{\frac{1}{4}} \left(\frac{3}{3c^2+\beta^2} \right)^{\frac{3}{4}} \left(c^2 - \beta^2\right)^{\frac{1}{4}} E_\mathrm{r}^{\frac{3}{4}}. \tag{3.68}$$

We assume that $|\beta_i| < c$ at initial time $t = 0$. Then, $s_i(t=0) > 0$. Because s is non-decreasing, $s_i(t) > 0$ for all t. This leads to

$$E_i^{\frac{3}{4}} \left(c^2 - \beta_i^2\right)^{\frac{1}{4}} > 0. \tag{3.69}$$

If $E_i > 0$, then $|\beta_i| < c$.

Buet and Despres [2008] show that

$$\left| \beta^*_{i+\frac{1}{2}} \right| \leqslant c, \tag{3.70}$$

under the Courant-Friedrichs-Lewy (CFL) condition $\Delta t \leqslant \frac{\Delta x}{2c}$ for the diffusion step.

3.D.2 Transport step

We consider now the transport step. We define $[\beta^*]^{\pm} = \frac{\beta^* \pm |\beta^*|}{2}$. If $\beta^* \geqslant 0$, then $[\beta^*]^+ = \beta^*$ and $[\beta^*]^- = 0$. On the contrary, if $\beta^* \leqslant 0$, then $[\beta^*]^+ = 0$ and $[\beta^*]^- = \beta^*$. With this notation, $\vartheta_{i+\frac{1}{2}}\beta^*_{i+\frac{1}{2}} = \vartheta_i \left[\beta^*_{i+\frac{1}{2}}\right]^+ + \vartheta_{i+1}\left[\beta^*_{i+\frac{1}{2}}\right]^-$, for $\vartheta \in \{\varrho, F, E\}$. After some algebra, eq. 3.16 becomes

$$\vartheta_i^{n+1} = \left(1 + \frac{\Delta t}{\Delta x}\left[\beta^*_{i+\frac{1}{2}}\right]^- - \frac{\Delta t}{\Delta x}\left[\beta^*_{i-\frac{1}{2}}\right]^+\right)\vartheta_i(t) - \frac{\Delta t}{\Delta x}\left[\beta^*_{i+\frac{1}{2}}\right]^-\vartheta_{i+1}(t) + \frac{\Delta t}{\Delta x}\left[\beta^*_{i-\frac{1}{2}}\right]^+\vartheta_{i-1}(t). \tag{3.71}$$

3.D.3 Split scheme

Let us consider the complete split scheme, the diffusion step followed by the transport step. Equation 3.71 becomes

$$\vartheta_i^{n+1} = \left(1 + \frac{\Delta t}{\Delta x}\left[\beta^*_{i+\frac{1}{2}}\right]^- - \frac{\Delta t}{\Delta x}\left[\beta^*_{i-\frac{1}{2}}\right]^+\right)\widetilde{\vartheta}_i - \frac{\Delta t}{\Delta x}\left[\beta^*_{i+\frac{1}{2}}\right]^-\widetilde{\vartheta}_{i+1} + \frac{\Delta t}{\Delta x}\left[\beta^*_{i-\frac{1}{2}}\right]^+\widetilde{\vartheta}_{i-1}. \tag{3.72}$$

Under the CFL condition eq. 3.70, ϑ_i^{n+1} is a convex combination of $\widetilde{\vartheta}_{i-1}$, $\widetilde{\vartheta}_i$ and $\widetilde{\vartheta}_{i+1}$. Similar computations can be done for the entropy, with inequality in eq. 3.72. By doing the same computations as for the diffusion step, we show that $E_i^{n+1} > 0$ and $\left|\beta_i^{n+1}\right| \leqslant c$.

3.D.4 Unsplit scheme

The unsplit scheme can be seen as the average of two steps:

$$\vartheta_i^{n+1} = \frac{1}{2}\left(\vartheta_i^D + \vartheta_i^T\right), \tag{3.73}$$

with ϑ_i^D given by the diffusion step and

$$\vartheta_i^T = \vartheta_i^n - \frac{\Delta t}{\Delta x}\left(\beta^*_{i+\frac{1}{2}}\vartheta^n_{i+\frac{1}{2}} - \beta^*_{i-\frac{1}{2}}\vartheta^n_{i-\frac{1}{2}}\right), \tag{3.74}$$

for $\vartheta \in \{\varrho, F, E\}$.

Using eq. 3.71, we have

$$\varrho_i^T = \left(1 + \frac{\Delta t}{\Delta x}\left[\beta^*_{i+\frac{1}{2}}\right]^- - \frac{\Delta t}{\Delta x}\left[\beta^*_{i-\frac{1}{2}}\right]^+\right)\varrho_i^n - \frac{\Delta t}{\Delta x}\left[\beta^*_{i+\frac{1}{2}}\right]^-\varrho^n_{i+1} + \frac{\Delta t}{\Delta x}\left[\beta^*_{i-\frac{1}{2}}\right]^+\varrho^n_{i-1}. \tag{3.75}$$

Under the CFL condition eq. 3.70, ϱ_i^{n+1} is a convex combination of ϱ^n_{i-1}, ϱ_i^n and ϱ^n_{i+1}. Therefore, $\varrho_i^T \geqslant 0$.

For $\bar{\vartheta} \in \{v, e\}$, we have

$$\begin{aligned}
\bar{\vartheta}_i^T &= \frac{\left(\varrho\bar{\vartheta}\right)_i^T}{\varrho_i^T} \\
&= \frac{\varrho_i^n}{\varrho_i^T}\left(1 + \frac{\Delta t}{\Delta x}\left[\beta^*_{i+\frac{1}{2}}\right]^+ - \frac{\Delta t}{\Delta x}\left[\beta^*_{i-\frac{1}{2}}\right]^-\right)\bar{\vartheta}_i^n - \frac{\varrho^n_{i+1}}{\varrho_i^T}\frac{\Delta t}{\Delta x}\left[\beta^*_{i+\frac{1}{2}}\right]^-\bar{\vartheta}^n_{i+1} + \frac{\varrho^n_{i-1}}{\varrho_i^T}\frac{\Delta t}{\Delta x}\left[\beta^*_{i-\frac{1}{2}}\right]^+\bar{\vartheta}^n_{i-1} \\
&= \frac{\alpha_i^0}{\varrho_i^T}\bar{\vartheta}_i^n + \frac{\alpha_i^+}{\varrho_i^T}\bar{\vartheta}^n_{i+1} + \frac{\alpha_i^-}{\varrho_i^T}\bar{\vartheta}^n_{i-1}.
\end{aligned} \tag{3.76}$$

After some algebra, one can show that $\alpha_i^0 \geqslant 0$, $\alpha_i^+ \geqslant 0$, $\alpha_i^- \geqslant 0$ and $\alpha_i^0 + \alpha_i^+ + \alpha_i^- = \varrho_i^T$. $\bar{\vartheta}_i^T$ is a convex combination of $\bar{\vartheta}^n$ in the neighbor cells. Similar computations can be done for the entropy, therefore $s_i^T \geqslant 0$ and $S_i^T = \varrho_i^T s_i^T \geqslant 0$. We can conclude as done in section 3.D.3 because $S_i^{n+1} = \frac{1}{2}\left(S_i^D + S_i^T\right) \geqslant 0$.

3.E Derivation of the unsplit scheme

The goal of this section is to build an approximate Riemann solver for eq. 3.5. For the sake of simplicity, we focus on the one-dimensional case, but it can be extended to higher dimensions without any difficulty.

3.E.1 Relaxed set of equations

Let us introduce an approximation of the density $r \approx \varrho$, velocity $b \approx \beta$, radiative flux $\mu \approx \varrho v$, energy $\varphi \approx \varrho e$ and pressure $\Pi \approx q$ and a relaxation parameter λ in the flux terms:

$$\begin{cases} \partial_t \varrho + \partial_x (rb) = 0 \\ \partial_t (\varrho v) + \partial_x (b\mu) + c^2 \partial_x \Pi = 0 \\ \partial_t (\varrho e) + \partial_x (\varphi b) + \partial_x (\Pi b) = 0. \end{cases} \tag{3.77}$$

We add the corresponding evolution equations to the system:

$$\begin{cases} \partial_t r + b\partial_x r = \lambda (r - \varrho) \\ \partial_t \mu + b\partial_x \mu = \lambda (\mu - \varrho v) \\ \partial_t \varphi + b\partial_x \varphi = \lambda (\varphi - \varrho e) \\ \partial_t b - \dfrac{c^2 - b^2}{3c^2 - b^2} b\partial_x b + \dfrac{3}{4\Pi} \dfrac{(c^2 - b^2)^2}{3c^2 - b} \partial_x \Pi = \lambda (b - \beta) \\ \partial_t \Pi + \dfrac{4c^2 \Pi}{3c^2 - b^2} \partial_x b - \dfrac{c^2 - b^2}{3c^2 - b^2} b\partial_x \Pi = \lambda (\Pi - q). \end{cases} \tag{3.78}$$

The relaxed system is an approximation of the original system, which can be recovered in the limit $\lambda \to \infty$. We solve it using an operator splitting technique. We first solve $\partial_t \widetilde{\vartheta} = \lambda \left(\widetilde{\vartheta} - \vartheta\right)$ for $\left(\vartheta, \widetilde{\vartheta}\right) \in \{(\varrho, r), (\varrho v, \mu), (\varrho e, \varphi), (\beta, b), (q, \Pi)\}$, followed by the transport system without source terms, with $\lambda = 0$. We end up with the following system:

$$\begin{cases} \partial_t \varrho + b\partial_x r + r\partial_x b = 0 \\ \partial_t (\varrho v) + b\partial_x \mu + \mu \partial_x b + c^2 \partial_x \Pi = 0 \\ \partial_t (\varrho e) + b\partial_x \varphi + (\varphi + \Pi) \partial_x b + b\partial_x \Pi = 0 \\ \partial_t r + b\partial_x r = 0 \\ \partial_t \mu + b\partial_x \mu = 0 \\ \partial_t \varphi + b\partial_x \varphi = 0 \\ \partial_t b - \dfrac{c^2 - b^2}{3c^2 - b^2} b\partial_x b + \dfrac{3}{4\Pi} \dfrac{(c^2 - b^2)^2}{3c^2 - b} \partial_x \Pi = 0 \\ \partial_t \Pi + \dfrac{4c^2 \Pi}{3c^2 - b^2} \partial_x b - \dfrac{c^2 - b^2}{3c^2 - b^2} b\partial_x \Pi = 0. \end{cases} \tag{3.79}$$

3.E.2 Eigenstructure

Let us now study the eigenstructure of eq. 3.79. We write $\widetilde{\boldsymbol{W}} = (\varrho, \varrho v, \varrho e, r, \mu, \varphi, b, \Pi)^T$, then eq. 3.79 can be written in the matrix form

$$\partial_t \widetilde{\boldsymbol{W}} + \widetilde{\mathbb{M}}\left(\widetilde{\boldsymbol{W}}\right) \partial_x \widetilde{\boldsymbol{W}} = 0, \tag{3.80}$$

where

$$\widetilde{\mathbb{M}}\left(\widetilde{\boldsymbol{W}}\right) = \begin{pmatrix} 0 & 0 & 0 & b & 0 & 0 & r & 0 \\ 0 & 0 & 0 & 0 & b & 0 & \mu & c^2 \\ 0 & 0 & 0 & 0 & 0 & b & \varphi + \Pi & b \\ 0 & 0 & 0 & b & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & b & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{c^2-b^2}{3c^2-b^2} b & \frac{3}{4\Pi} \frac{\left(c^2-b^2\right)^2}{3c^2-b^2} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{4c^2\Pi}{3c^2-b^2} & -\frac{c^2-b^2}{3c^2-b^2} b \end{pmatrix}. \tag{3.81}$$

Using similar computations as in section 3.B, the eigenvalues of $\widetilde{\mathbb{M}}$ are 0 with multiplicity 3, b with multiplicity 3, $\widetilde{\Lambda}^+$ and $\widetilde{\Lambda}^-$, with $\widetilde{\Lambda}^\pm = \frac{c^2-b^2}{b \pm \sqrt{3}c}$. The three eigenvectors associated with the eigenvalue 0 are

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{3.82}$$

The field associated with the stationary wave is linearly degenerated. The three eigenvectors associated with the eigenvalue b are

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \tag{3.83}$$

The field is also linearly degenerated. Finally, $\widetilde{\boldsymbol{R}}^+$ and $\widetilde{\boldsymbol{R}}^-$ are the eigenvectors associated with the eigenvalues $\widetilde{\Lambda}^+$ and $\widetilde{\Lambda}^-$ respectively, with

$$\widetilde{\boldsymbol{R}}^+ = \begin{pmatrix} \frac{\sqrt{3}r(\sqrt{3}c-b)}{4c\Pi} \\ -\frac{c(c-\sqrt{3}b)(\sqrt{3}c-b)}{c^2-b} \\ \frac{(\sqrt{3}c-b)(3c^2-b^2)}{(\sqrt{3}c+b)(c^2-b^2)} \\ 0 \\ 0 \\ 0 \\ -\frac{\sqrt{3}(c^2-b^2)}{4c\Pi} \\ 1 \end{pmatrix}, \quad \widetilde{\boldsymbol{R}}^- = \begin{pmatrix} \frac{\sqrt{3}r(\sqrt{3}c+b)}{4c\Pi} \\ \frac{c(c+\sqrt{3}b)(\sqrt{3}c+b)}{c^2-b^2} \\ \frac{(\sqrt{3}c+b)(3c^2-b^2)}{(\sqrt{3}c-b)(c^2-b^2)} \\ 0 \\ 0 \\ 0 \\ \frac{\sqrt{3}(c^2-b^2)}{4c\Pi} \\ 1 \end{pmatrix}. \tag{3.84}$$

The field is genuinely nonlinear.

3.E.3 Jump conditions

Let us now write the jump conditions to solve the Riemann problem associated with eq. 3.79. From eq. 3.77, we only need the variables r, μ, φ, b and Π in an intermediate state to update the variables of interest ϱ, ϱv and ϱe. We are not interested in the update of the relaxed variables r, μ, φ, b, and Π. The intermediate state for b and Π will be given by Buet and Despres [2008], therefore we focus on the intermediate state for r, φ and μ.

We introduce the artificial density variable Θ to rewrite eq. 3.79 in conservative variables:

$$\begin{cases} \partial_t \varrho + \partial_x (br) = 0 \\ \partial_t (\varrho v) + \partial_x (b\mu) + c^2 \partial_x \Pi = 0 \\ \partial_t (\varrho e) + \partial_x (b\varphi) + \partial_x (b\Pi) = 0 \\ \partial_t (\Theta r) + \partial_x (\Theta b r) = 0 \\ \partial_t (\Theta \mu) + \partial_x (\Theta b \mu) = 0 \\ \partial_t (\Theta \varphi) + \partial_x (\Theta b \varphi) = 0 \\ \partial_t \Theta + \partial_x (b\Theta) = 0. \end{cases} \tag{3.85}$$

The corresponding Rankine-Hugoniot jump condition for a discontinuity of speed D are

$$\begin{cases} -D [\varrho] + [rb] = 0 \\ -D [\varrho v] + [b\mu] + c^2 [\Pi] = 0 \\ -D [\varrho e] + [b\varphi] + [b\Pi] = 0 \\ -D [\Theta r] + [\Theta b r] = 0 \\ -D [\Theta \mu] + [\Theta b \mu] = 0 \\ -D [\Theta \varphi] + [\Theta b \varphi] = 0 \\ -D [\Theta] + [b\Theta] = 0, \end{cases} \tag{3.86}$$

where $[\cdot]$ is the difference from either side of the discontinuity. We introduce the following quantify:

$$m = \Theta (b - D), \tag{3.87}$$

to write the jump condition as

$$\begin{cases} -D [\varrho] + [rb] = 0 \\ -D [\varrho v] + [b\mu] + c^2 [\Pi] = 0 \\ -D [\varrho e] + [b\varphi] + [b\Pi] = 0 \\ m [r] = 0 \\ m [\mu] = 0 \\ m [\varphi] = 0 \\ [m] = 0. \end{cases} \tag{3.88}$$

We now have to consider several cases:

1. If $D = 0$, then eq. 3.88 becomes

$$\begin{cases} [br] = 0 \\ [b\mu] + c^2 [\Pi] = 0 \\ [b\varphi] + [b\Pi] = 0 \\ m [r] = 0 \\ m [\mu] = 0 \\ m [\varphi] = 0 \\ [m] = 0, \end{cases} \tag{3.89}$$

which leads to

$$\begin{cases} b [r] = 0 \\ b [\mu] + c^2 [\Pi] = 0 \\ b [\varphi] + b [\Pi] = 0 \\ m [r] = 0 \\ m [\mu] = 0 \\ m [\varphi] = 0 \\ [m] = 0. \end{cases} \tag{3.90}$$

1.a. If $D = 0$ and $b = 0$, then we have $m = 0$ and $[r] \in \mathbb{R}$, $[\mu] \in \mathbb{R}$, $[\varphi] \in \mathbb{R}$.

1.b. If $D = 0$ and $b \neq 0$, then $[r] = 0$, $[\mu] = 0$, $[\varphi] = 0$.

2. If $D \neq 0$,

2.a. We first assume that $[b] = 0$. Then eq. 3.88 becomes

$$\begin{cases} -D [\varrho] + b [r] = 0 \\ -D [\varrho v] + b [\mu] + c^2 [\Pi] = 0 \\ -D [\varrho e] + b [\varphi] + b [\Pi] = 0 \\ m [r] = 0 \\ m [\mu] = 0 \\ m [\varphi] = 0 \\ (b - D) [\Theta] = 0. \end{cases} \tag{3.91}$$

2.a.i. If $D \neq 0$, $[b] = 0$ and $D = b$, then $m = 0$ and $[r] \in \mathbb{R}$, $[\mu] \in \mathbb{R}$, $[\varphi] \in \mathbb{R}$.

2.a.ii. If $D \neq 0$, $[b] = 0$ and $D \neq b$, then $[r] = 0$, $[\mu] = 0$, $[\varphi] = 0$.

2.b. If $D \neq 0$ and $[b] \neq 0$, we have to consider two cases:

2.b.i. If $D \neq 0$, $[b] \neq 0$ and $b \neq D$, then $m \neq 0$ and $[r] = 0$, $[\mu] = 0$, $[\varphi] = 0$.

2.b.ii. If $D \neq 0$, $[b] \neq 0$ and $b = D$, then $m = 0$ and $[r] \in \mathbb{R}$, $[\mu] \in \mathbb{R}$, $[\varphi] \in \mathbb{R}$

Table 3.2 summarizes the different cases.

3.E.4 Riemann problem solution

Let us consider a piecewise initial data defined by

$$\begin{cases} (r_L, \mu_L, \varphi_L, b_L, \Pi_L)^T \text{ if } \dfrac{x}{t} < \tilde{\Lambda}^-, \\ (r_R, \mu_R, \varphi_R, b_R, \Pi_R)^T \text{ if } \dfrac{x}{t} > \tilde{\Lambda}^+. \end{cases} \tag{3.92}$$

Value of D and b	$[r], [\mu], [\varphi]$	Wave
$D = 0,\ b = 0$	$\in \mathbb{R}$	Contact wave
$D = 0,\ b \neq 0$	$= 0$	-
$D \neq 0,\ [b] = 0,\ D = b$	$\in \mathbb{R}$	Contact wave
$D \neq 0,\ [b] = 0,\ D \neq b$	$= 0$	-
$D \neq 0,\ [b] \neq 0,\ D \neq b$	$= 0$	-
$D \neq 0,\ [b] \neq 0,\ D = b$	$\in \mathbb{R}$	Shock wave

Table 3.2 – Summary of the different cases, according to the value of D and b.

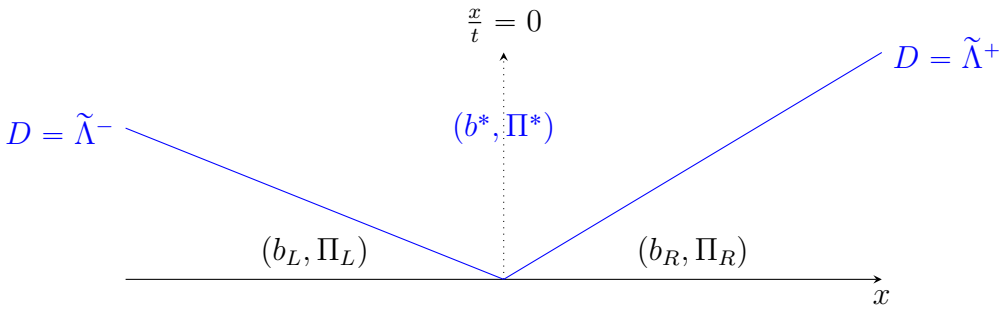


Figure 3.10 – Solution of the Riemann problem for b and Π.

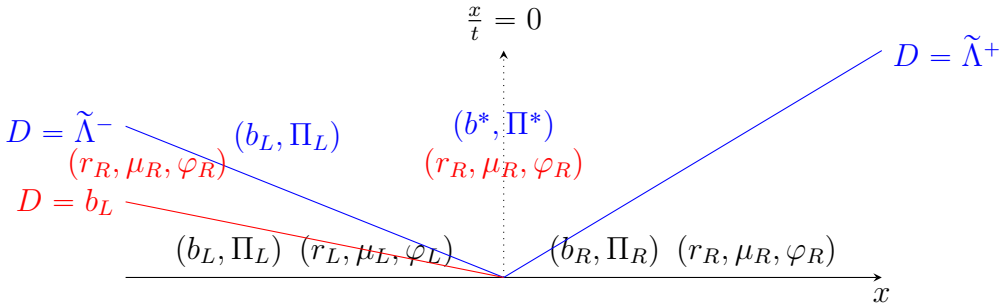


Figure 3.11 – Solution of the Riemann problem for b, Π, r, μ, φ with $D = b_L$.

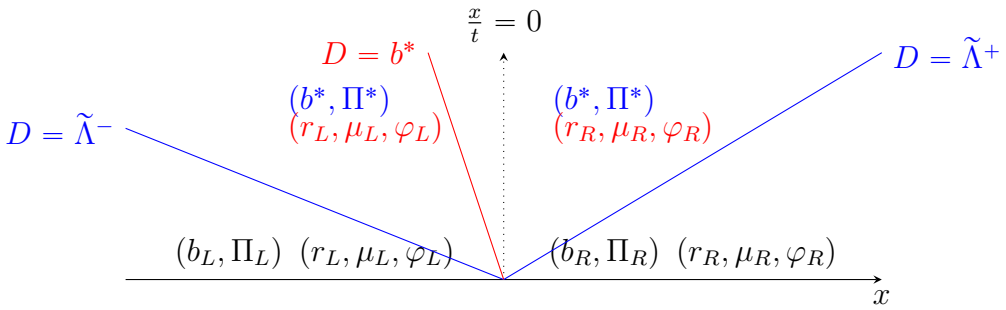


Figure 3.12 – Solution of the Riemann problem for b, Π, r, μ, φ with $D = b^* < 0$.

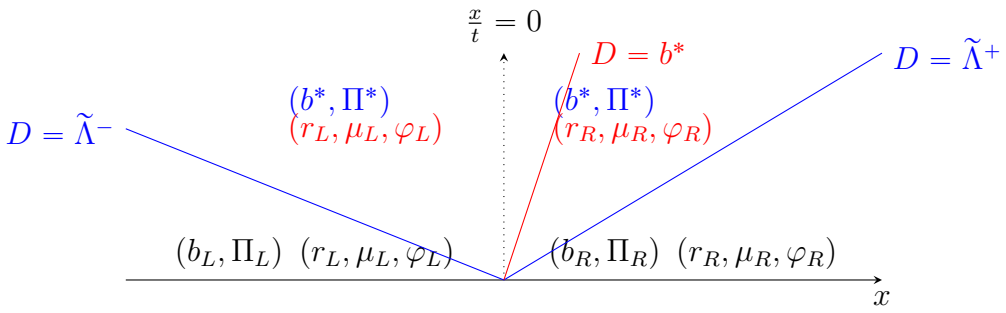


Figure 3.13 – Solution of the Riemann problem for b, Π, r, μ, φ with $D = b^* > 0$.

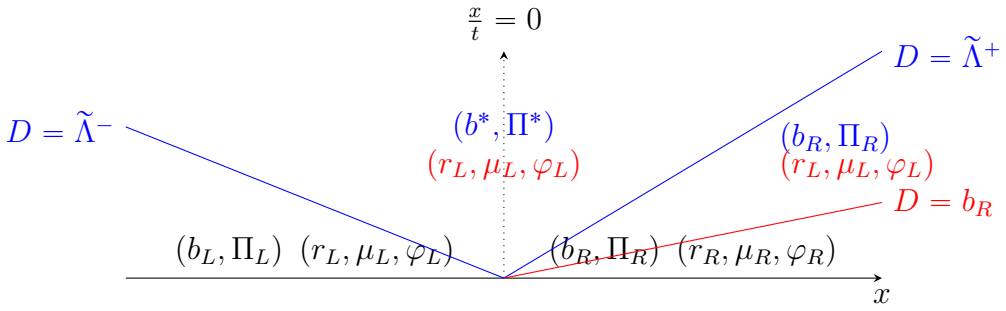


Figure 3.14 – Solution of the Riemann problem for b, Π, r, μ, φ with $D = b_R$.

We now build an approximate Riemann solver for eq. 3.79. We are looking for a function composed of three states separated by discontinuities as follows (see fig. 3.10):

$$\begin{cases} (r_L, \mu_L, \varphi_L, b_L, \Pi_L)^T \text{ if } \dfrac{x}{t} < \widetilde{\Lambda}^-, \\ (r^*, \mu^*, \varphi^*, b^*, \Pi^*)^T \text{ if } \widetilde{\Lambda}^- < \dfrac{x}{t} < \widetilde{\Lambda}^+, \\ (r_R, \mu_R, \varphi_R, b_R, \Pi_R)^T \text{ if } \dfrac{x}{t} > \widetilde{\Lambda}^+. \end{cases} \tag{3.93}$$

The computation of b^* and Π^* is independent of the other variables. We take b^* and Π^* given by eq. 3.26, but other choices can be done. Let us now show the different possible wave patterns. Then, we distinguish the different cases corresponding to the localization of the wave $D = b$:

— $D = b_L$: we take $(r^*, \mu^*, \varphi^*)^T = (r_R, \mu_R, \varphi_R)^T$, see fig. 3.11.

— $D = b^*$: we take

$$(r^*, \mu^*, \varphi^*)^T = \begin{cases} (r_L, \mu_L, \varphi_L)^T \text{ if } \dfrac{x}{t} < b^*, \\ (r_R, \mu_R, \varphi_R)^T \text{ if } \dfrac{x}{t} > b^*, \end{cases} \tag{3.94}$$

see figs. 3.12 and 3.13.

— $D = b_R$: we take $(r^*, \mu^*, \varphi^*)^T = (r_L, \mu_L, \varphi_L)^T$, see fig. 3.14.

3.E.5 Riemann flux computation

From the wave pattern shown previously, we can deduce the following algorithm to compute the numerical fluxes $\mathcal{F}^*$ on the density and radiative flux variables, ϱ and ϱv given by our relaxed system eq. 3.79. To do so, we evaluate the flux function from eq. 3.5 on the state obtained on $\frac{x}{t} = 0$:

1. Compute (b^*, Π^*) with eq. 3.26;
2. Compute the flux according to the sign of b^*:

$$\mathcal{F}^*\left(\begin{pmatrix}\varrho_L\\F_L\\E_L\end{pmatrix},\begin{pmatrix}\varrho_R\\F_R\\E_R\end{pmatrix}\right)=\begin{cases}\begin{pmatrix}\varrho_L b^*\\F_L b^*+c^2\Pi^*\\E_L b^*+\Pi^* b^*\end{pmatrix} & \text{if } b^*\geqslant 0\\ \begin{pmatrix}\varrho_R b^*\\F_R b^*+c^2\Pi^*\\E_R b^*+\Pi^* b^*\end{pmatrix} & \text{if } b^*\leqslant 0.\end{cases} \tag{3.95}$$

3.F Numerical schemes in the diffusive limit

As in section 2.B, we consider both numerical schemes developed in section 3.1 in the asymptotic regime, with $\sigma^{\mathrm{s}}_{i+\frac{1}{2}}\Delta x\to\infty$. Following section 1.4, we introduce the rescaling parameter ε to write the time (resp. the opacity) as $\widetilde{\Delta t}=\varepsilon\Delta t$ (resp. $\widetilde{\sigma^{\mathrm{a}}}=\varepsilon\sigma^{\mathrm{a}}$ and $\widetilde{\sigma^{\mathrm{s}}}=\varepsilon\sigma^{\mathrm{s}}$). This leads to

$$\varepsilon\varrho_i^{n+1}=\varepsilon\varrho_i^n-\frac{\widetilde{\Delta t}}{\Delta x}\left(\varrho^m_{i+\frac{1}{2}}\beta^*_{i+\frac{1}{2}}-\varrho^m_{i-\frac{1}{2}}\beta^*_{i-\frac{1}{2}}\right) \tag{3.96a}$$

$$\begin{aligned}\varepsilon^2F_i^{n+1}=&\varepsilon^2F_i^n-\varepsilon\frac{\widetilde{\Delta t}}{\Delta x}\left(F^m_{i+\frac{1}{2}}\beta^*_{i+\frac{1}{2}}+c^2q^*_{i+\frac{1}{2}}-F^m_{i-\frac{1}{2}}\beta^*_{i-\frac{1}{2}}-c^2q^*_{i-\frac{1}{2}}\right)\\&-c\widetilde{\sigma^{\mathrm{s}}}_i\widetilde{\Delta t}F_i^{n+1}\end{aligned} \tag{3.96b}$$

$$\begin{aligned}\varepsilon^2E_i^{n+1}=&\varepsilon^2E_i^n-\varepsilon\frac{\widetilde{\Delta t}}{\Delta x}\left(E^m_{i+\frac{1}{2}}\beta^*_{i+\frac{1}{2}}+q^*_{i+\frac{1}{2}}\beta^*_{i+\frac{1}{2}}-E^m_{i-\frac{1}{2}}\beta^*_{i-\frac{1}{2}}-q^*_{i-\frac{1}{2}}\beta^*_{i-\frac{1}{2}}\right)\\&+c\widetilde{\sigma^{\mathrm{a}}}_i\widetilde{\Delta t}\left(a_{\mathrm{r}}\left(T_i^{n+1}\right)^4-E_i^{n+1}\right)\end{aligned} \tag{3.96c}$$

$$\varepsilon^2\rho c_vT_i^{n+1}=\varepsilon^2\rho c_vT_i^n-c\widetilde{\sigma^{\mathrm{a}}}_i\widetilde{\Delta t}\left(a_{\mathrm{r}}\left(T_i^{n+1}\right)^4-E_i^{n+1}\right), \tag{3.96d}$$

where $\vartheta^m_{i+\frac{1}{2}}$ is given by eq. 3.17 for the split scheme ($\vartheta^m_{i+\frac{1}{2}}=\widetilde{\vartheta}_{i+\frac{1}{2}}$) and by eq. 3.20 for the unsplit scheme ($\vartheta^m_{i+\frac{1}{2}}=\vartheta^n_{i+\frac{1}{2}}$), for $\vartheta\in\{\varrho,F,E\}$.

Variables are expanded, e.g., $E_i^n=E_{i,0}^n+\varepsilon E_{i,1}^n+\mathcal{O}\left(\varepsilon^2\right)$. Expanding eq. 3.96c or eq. 3.96d at order 0 leads to

$$E_{i,0}^{n+1}=a_{\mathrm{r}}\left(T_{i,0}^{n+1}\right)^4. \tag{3.97}$$

Equation 3.96b at order 0 leads to

$$F_{i,0}^{n+1}=0. \tag{3.98}$$

Because $F_{i,0}^n=\left(E_{i,0}^n+q_{i,0}^n\right)\beta_{i,0}^n$ and $E+q>0$, one has

$$\beta_{i,0}^n=0. \tag{3.99}$$

At order 0 for eq. 3.96a, we have $\beta^*_{i+\frac{1}{2},0}-\beta^*_{i-\frac{1}{2},0}=0$ for all cells i. Therefore, $\beta^*_{i+\frac{1}{2},0}=C$, where C is a constant that does not depend on i. With well-chosen boundary conditions, we have $\beta^*_{0+\frac{1}{2},0}=0$, which leads to $\beta^*_{i+\frac{1}{2},0}=0$ for all cells i.

Moreover,

$$\begin{aligned} 0 = \beta^*_{i+\frac{1}{2},0} &= \frac{\sqrt{3}c}{4}\frac{\left(q^n_{i,0} - \frac{\widetilde{\sigma^{\mathrm{s}}}_{i+\frac{1}{2}}}{2c}F^n_{i,1}\right) - \left(q^n_{i+1,0} + \frac{\widetilde{\sigma^{\mathrm{s}}}_{i+\frac{1}{2}}}{2c}F^n_{i+1,0}\right)}{E^n_{i,0} + E^n_{i+1,0}} \\ \Leftrightarrow \frac{F^n_{i+1,1} + F^n_{i,1}}{2} &= -\frac{c}{\widetilde{\sigma^{\mathrm{s}}}_{i+\frac{1}{2}}}\frac{q^n_{i+1,0} - q^n_{i,0}}{\Delta x}. \end{aligned} \tag{3.100}$$

Using eq. 3.99 in the definition of q, we have $q^n_{i,0} = \frac{1}{3}E^n_{i,0}$ and

$$\frac{F^n_{i+1,1} + F^n_{i,1}}{2} = -\frac{c}{3\widetilde{\sigma^{\mathrm{s}}}_{i+\frac{1}{2}}}\frac{E^n_{i+1,0} - E^n_{i,0}}{\Delta x}. \tag{3.101}$$

Let us now consider the sum of eq. 3.96c and eq. 3.96d. The term of order 1 is 0 because $\beta^*_{i+\frac{1}{2},0} = 0$. We now look at the term of order 2 in the numerical fluxes. For the same reason, it is

$$E^m_{i+\frac{1}{2},0}\beta^*_{i+\frac{1}{2},1} + q^*_{i+\frac{1}{2},0}\beta^*_{i+\frac{1}{2},1} - E^m_{i-\frac{1}{2},0}\beta^*_{i-\frac{1}{2},1} - q^*_{i-\frac{1}{2},0}\beta^*_{i-\frac{1}{2},1}. \tag{3.102}$$

From now on, we assume that all functions are sufficiently smooth to write, for example,

$$E^n_{i+1,0} = E^n_{i,0} + \Delta x\frac{E^n_{i+1,0} - E^n_{i,0}}{\Delta x} = E^n_{i,0} + \mathcal{O}\left(\Delta x\right). \tag{3.103}$$

We can now look at the different terms involved in eq. 3.102. First,

$$\begin{aligned} \beta^*_{i+\frac{1}{2},1} &= \frac{E^n_{i,0}\beta^n_{i,1} + E^n_{i+1,0}\beta^n_{i+1,1}}{E^n_{i,0} + E^n_{i+1,0}} - \frac{3\sqrt{3}c}{4}\left(\underbrace{\frac{q^n_{i+1,1} - q^n_{i,1}}{E^n_{i,0} + E^n_{i+1,0}}}_{\mathcal{O}(\Delta x)} - \frac{\widetilde{\sigma^{\mathrm{s}}}_{i+\frac{1}{2}}\Delta x}{2c}\underbrace{\frac{F^n_{i+1,2} - F^n_{i,2}}{E^n_{i,0} + E^n_{i+1,0}}}_{\mathcal{O}(\Delta x)}\right) \\ &= \alpha_i\beta^n_{i,1} + \left(1-\alpha_i\right)\beta^n_{i+1,1} + \mathcal{O}\left(\Delta x\right), \end{aligned} \tag{3.104}$$

with $\alpha_i = \frac{E^n_{i,0}}{E^n_{i,0}+E^n_{i+1,0}}$. Second,

$$\begin{aligned} q^*_{i+\frac{1}{2},0} &= \frac{\frac{q^n_{i,0}}{E^n_{i,0}} + \frac{q^n_{i+1,0}}{E^n_{i+1,0}}}{\frac{1}{E^n_{i,0}} + \frac{1}{E^n_{i+1,0}}} + \frac{\widetilde{\sigma^{\mathrm{s}}}_{i+\frac{1}{2}}\Delta x}{2c}\frac{\frac{F^n_{i+1,1}}{E^n_{i+1,0}} - \frac{F^n_{i,1}}{E^n_{i,0}}}{\frac{1}{E^n_{i,0}} + \frac{1}{E^n_{i+1,0}}} \\ &= \frac{E^n_{i+1,0}q^n_{i,0} + E^n_{i,0}q^n_{i+1,0}}{E^n_{i,0} + E^n_{i+1,0}} + \mathcal{O}\left(\Delta x\right) \\ &= \left(1-\alpha_i\right)q^n_{i,0} + \alpha_i q^n_{i+1,0} + \mathcal{O}\left(\Delta x\right). \end{aligned} \tag{3.105}$$

Using eqs. 3.104 and 3.105, one has

$$\beta^n_{i+\frac{1}{2},1}q^n_{i+\frac{1}{2},0} = \alpha_i\beta^n_{i,1}q^n_{i,0} + \left(1-\alpha_i\right)\beta^n_{i+1,1}q^n_{i+1,0} + \mathcal{O}\left(\Delta x\right). \tag{3.106}$$

Furthermore,

$$\alpha_i = \frac{E^n_{i,0}}{E^n_{i,0} + E^n_{i+1,0}} = \frac{1}{2} + \mathcal{O}\left(\Delta x\right), \tag{3.107}$$

which leads to

$$\beta^n_{i+\frac{1}{2},1} q^n_{i+\frac{1}{2},0} = \frac{1}{2}\beta^n_{i,1} q^n_{i,0} + \frac{1}{2}\beta^n_{i+1,1} q^n_{i+1,0}. \tag{3.108}$$

Likewise,

$$\begin{aligned} E^m_{i+\frac{1}{2},0}\beta^n_{i+\frac{1}{2},1} &= \alpha_i E^m_{i+\frac{1}{2},0}\beta^n_{i,1} + (1-\alpha_i)\, E^m_{i+\frac{1}{2},0}\beta^n_{i+1,1} + \mathcal{O}(\Delta x) \\ &= \frac{1}{2}E^m_{i,0}\beta^n_{i,1} + \frac{1}{2}E^m_{i+1,0}\beta^n_{i+1,1} + \mathcal{O}(\Delta x). \end{aligned} \tag{3.109}$$

Finally,

$$\left(E^m_{i+\frac{1}{2},0} + q^n_{i+\frac{1}{2},0}\right)\beta^n_{i+\frac{1}{2},1} = \frac{1}{2}\left(E^m_{i,0} + q^n_{i,0}\right)\beta^n_{i,1} + \frac{1}{2}\left(E^m_{i+1,0} + q^n_{i+1,0}\right)\beta^n_{i+1,1} + \mathcal{O}(\Delta x). \tag{3.110}$$

Using the unsplit scheme, we have $E^m_i = E^n_i$ and eq. 3.110 becomes

$$\left(E^m_{i+\frac{1}{2},0} + q^n_{i+\frac{1}{2},0}\right)\beta^n_{i+\frac{1}{2},1} = \frac{F^n_{i,1} + F^n_{i+1,1}}{2} + \mathcal{O}(\Delta x), \tag{3.111}$$

which leads to

$$E^{n+1}_{i,0} + \rho c_v T^{n+1}_{i,0} = E^n_{i,0} + \rho c_v T^n_{i,0} + \frac{c\widetilde{\Delta t}}{3\Delta x^2}\left(\frac{E^n_{i+1,0} - E^n_{i,0}}{\widetilde{\sigma^s}_{i+\frac{1}{2}}} - \frac{E^n_{i,0} - E^n_{i-1,0}}{\widetilde{\sigma^s}_{i-\frac{1}{2}}}\right) + \mathcal{O}(\Delta x). \tag{3.112}$$

Using the split scheme, we can write $E^m_i = \widetilde{E}_i = E^n_i + \mathcal{O}(\Delta t)$ and eq. 3.110 becomes

$$E^{n+1}_{i,0} + \rho c_v T^{n+1}_{i,0} = E^n_{i,0} + \rho c_v T^n_{i,0} + \frac{c\widetilde{\Delta t}}{3\Delta x^2}\left(\frac{E^n_{i+1,0} - E^n_{i,0}}{\widetilde{\sigma^s}_{i+\frac{1}{2}}} - \frac{E^n_{i,0} - E^n_{i-1,0}}{\widetilde{\sigma^s}_{i-\frac{1}{2}}}\right) + \mathcal{O}(\Delta x) + \mathcal{O}(\Delta t). \tag{3.113}$$

Equations 3.112 and 3.113 are consistent with the diffusion equation eq. 1.24 because $\mathcal{O}(\Delta x)$ and $\mathcal{O}(\Delta t)$ go to 0 as Δx and Δt go to 0. To summarize, eqs. 3.97, 3.98, 3.101, 3.112 and 3.113 are standard discretization of eqs. 1.22 to 1.24, so these schemes are asymptotic preserving.

Unlike the scheme developed in chapter 2 or the one presented by Berthon and Turpault [2011], we do not need to choose a parameter to recover the asymptotic behavior.

Chapter 4

Nonlinear Jacobi method and geometric multigrid

Contents

In chapter 2, we presented an asymptotic preserving scheme, with a well-balanced modification of the source terms, that uses a time-implicit integration. Nevertheless, it does not preserve the admissible states $E_\mathrm{r} > 0$ and $||\boldsymbol{F}_\mathbf{r}|| \leqslant cE_\mathrm{r}$. To tackle this issue, we presented in chapter 3 an asymptotic preserving scheme that preserves the admissible states. For the sake of simplicity, we derived it using a time-explicit integration. This chapter is an additional step towards the derivation of a time-implicit and asymptotic preserving scheme that preserves the admissible states: we present an implicit solver that preserves the admissible states. In this chapter, we consider a time-implicit HLL solver, without source terms. Therefore, the asymptotic preserving and well-balanced properties are not considered.

This chapter is organized as follows. We first present in section 4.1 a Jacobi method to solve the nonlinear system arising from the discretization of the M_1 model. Because this method is iterative, its convergence rate can be improved thanks to multigrid acceleration. We explore this technique in section 4.2. In section 4.3, we perform some tests to validate both algorithms. We also show some performance results. Finally, we reach our conclusion in section 4.4.

4.1 Nonlinear Jacobi method

We use notations similar to the previous chapters. We note h the step along the x-direction. As previously, Δt is the time interval between the current time t^n and t^{n+1}. We write x_i the center of the cell i. We use the notation u_i^n to represent the averaged quantity associated with the field u at time t^n in the cell i (finite volume).

4.1.1 Time-implicit HLL solver

Let us consider a time-implicit HLL solver for the M_1 model without source terms. For the sake of simplicity, we use fixed eigenvalues in the numerical fluxes (eq. 2.2). This leads to solving the following nonlinear system:

$$\begin{cases} E_i^{n+1}\left(1+c\dfrac{\Delta t}{h}\right) - \dfrac{\Delta t}{2h}\left(cE_{i+1}^{n+1} - F_{i+1}^{n+1}\right) - \dfrac{\Delta t}{2h}\left(cE_{i-1}^{n+1} + F_{i-1}^{n+1}\right) & = E_i^n \\ F_i^{n+1}\left(1+c\dfrac{\Delta t}{h}\right) - \dfrac{c\Delta t}{2h}\left(F_{i+1}^{n+1} - cP_{i+1}^{n+1}\right) - \dfrac{c\Delta t}{2h}\left(F_{i-1}^{n+1} + cP_{i-1}^{n+1}\right) & = F_i^n. \end{cases} \tag{4.1}$$

As done in chapter 2, eq. 4.1 can be solved using a Newton-Raphson method. However, numerical tests, especially in the free-streaming regime, have shown that the admissible states are not preserved when large time steps are used. A solution is therefore to reduce the time step, but it leads to poor performances when the radiative transfer is coupled to hydrodynamics. Buet and Despres [2006] show how the resolution of eq. 4.1 can be reduced to the resolution of a linear system (by fixing the nonlinearity at time t^n) and why this approach does not preserve the admissible states either.

In the next section, we present another method to solve eq. 4.1 while preserving the admissible states.

4.1.2 Jacobi method

Let us follow the work of Pichard [2016]. We first define the set of admissible states

$$\mathcal{S} = \left\{\left(E_\mathrm{r}, \boldsymbol{F}_\mathbf{r}\right), E_\mathrm{r} > 0, ||\boldsymbol{F}_\mathbf{r}|| \leqslant cE_\mathrm{r}\right\}. \tag{4.2}$$

By writing $\boldsymbol{v} = (\cdots, E_i, \boldsymbol{F}_i, \cdots)^T \in \mathcal{S}^N$, where N is the number of cells, eq. 4.1 rewrites

$$-\mathcal{L}\left(v_{i-1}^{n+1}\right) + \mathcal{D}\left(v_i^{n+1}\right) - \mathcal{R}\left(v_{i+1}^{n+1}\right) = v_i^n. \tag{4.3}$$

The operators $\mathcal{L}$, $\mathcal{D}$, and $\mathcal{R}$ contain the terms depending on $\left(E_{i-1}^{n+1}, \boldsymbol{F}_{i-1}^{n+1}\right)$, $\left(E_i^{n+1}, \boldsymbol{F}_i^{n+1}\right)$ and $\left(E_{i+1}^{n+1}, \boldsymbol{F}_{i+1}^{n+1}\right)$ respectively.

Equation 4.1 can also be seen as

$$\mathcal{A}(\boldsymbol{v}) = \boldsymbol{f}, \tag{4.4}$$

where $\mathcal{A}$ is a nonlinear operator and $\boldsymbol{f} \in \mathcal{S}^N$ is a known vector.

From Pichard [2016], we solve eq. 4.4 using algorithm 1.

Algorithm 1 Nonlinear Jacobi method

Initialization: $\boldsymbol{v}^{n+1,(0)} = \boldsymbol{v}^n$
while $\left|\left|\mathcal{A}\left(\boldsymbol{v}^{n+1,(k)}\right) - \boldsymbol{f}\right|\right| > \varepsilon_J$ **do**
  for each cell i **do**

$$v_i^{n+1,(k+1)} = \mathcal{D}^{-1}\left(f_i + \mathcal{L}\left(v_{i-1}^{n+1,(k)}\right) + \mathcal{R}\left(v_{i+1}^{n+1,(k)}\right)\right) \tag{4.5}$$

  end for
  $k \leftarrow k + 1$
end while
$\boldsymbol{v}^{n+1} = \boldsymbol{v}^{n+1,(k)}$

Let us notice that if $\mathcal{A}$ is a linear operator, algorithm 1 simplifies into the classical Jacobi method for a tridiagonal matrix (see e.g., Saad 2003). Pichard [2016] also shows that this algorithm converges to the unique solution $\boldsymbol{v}^{n+1}$ because the operator $\mathcal{A}$ is contractant.

4.1.3 Preservation of the admissible states

We assume that $v_i^n \in \mathcal{S}$, and we show that v_i^{n+1} obtained with algorithm 1 is also admissible, i.e., $v_i^{n+1} \in \mathcal{S}$. Let us show that if $\boldsymbol{v}^{n+1,(k)} \in \mathcal{S}^N$, then $\boldsymbol{v}^{n+1,(k+1)} \in \mathcal{S}^N$.

Using Proposition 5.1 in Pichard [2016], one can show that $\mathcal{L}$ and $\mathcal{R}$ are stable, i.e., $\mathcal{L}(v) \in \mathcal{S}$ and $\mathcal{R}(v) \in \mathcal{S}$ if $v \in \mathcal{S}$.

Because $f_i = v_i^n$, it is admissible and $f_i + \mathcal{L}(v_{i-1}) + \mathcal{R}(v_{i+1}) \in \mathcal{S}$.

Finally, in this particular case, $\mathcal{D} = \left(1 + c\frac{\Delta t}{h}\right)\mathbb{I}$, where $\mathbb{I}$ is the identity matrix. So, if $v \in \mathcal{S}$, then $\mathcal{D}^{-1}(v) = \frac{1}{1+c\frac{\Delta t}{h}} v \in \mathcal{S}$.

We have shown that $\mathcal{D}^{-1}$, $\mathcal{L}$, and $\mathcal{R}$ are stable. Therefore, $v_i^{n+1,(k+1)}$ is admissible. By induction, the proof is complete.

4.2 Geometric multigrid (GMG)

4.2.1 Convergence problematic

Even though the algorithm described in section 4.1 does not include the treatment of source terms for the M_1 model, it is the early development of a time-implicit solver suitable for the study of astrophysical problems. Therefore, it is reasonable to investigate its performances with academic problems. The one considered here is the beam

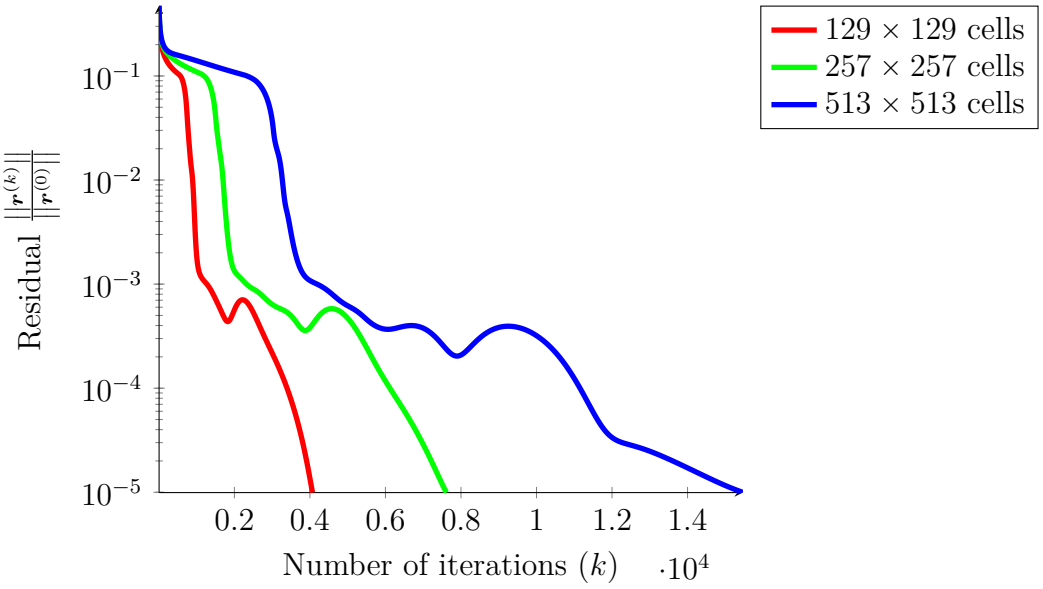


Figure 4.1 – Evolution of the residual as a function of the number of iterations of Jacobi method, with different resolutions.

Number of cells	Cell-updates/s
129 × 129	146
257 × 257	76
513 × 513	38

Table 4.1 – Number of cell-updates per second of Jacobi method, with different resolutions.

test already described in sections 2.2.3 and 3.2.1.2. It is the propagation of a beam in the vacuum, therefore it is well-suited to test our solver, as there is no opacity. A square two-dimensional domain is discretized with the same number of cells in both x and y-directions. Because the solver is time-implicit, we set a time step Δt such that the steady state is reached with one iteration. This test is discussed more precisely in section 4.3.1.

Figure 4.1 shows the evolution of the norm of the residual as a function of the number of iterations of the Jacobi method, with different resolutions. As the resolution increases, the number of iterations needed to reach the same residual also increases, from 4 000 iterations with 129 × 129 cells up to 15 000 iterations with 513 × 513 cells. As shown by table 4.1, performances decrease as the resolution increases, from around 150 cell-updates/s with 129 × 129 cells down to 40 cell-updates/s with 513 × 513 cells.

Indeed, Saad [2003] explains that the convergence rate decreases as the size of the problem increases, resulting in a decrease in performance. For linear problems, classical iterative methods are efficient to compute the high frequencies of the solution, but lack efficiency to compute its low frequencies. However, the computation is easier on a coarser grid with fewer unknowns. This observation has lead to the development of the geometric multigrid technique. The initial guess for the Jacobi algorithm (or any iterative method) is an interpolation of a solution computed on a coarser grid. This method requires information about the geometry of the problem, unlike preconditioners based on Krylov subspace. This additional information allows the multigrid method to be very efficient, but it lacks generality. To tackle this issue, algebraic multigrid methods have been developed, see section 5.1.5.

We first present the geometric multigrid algorithm in the linear case (section 4.2.2). Then, in section 4.2.3, we present its extension to the general nonlinear case, leading to the Full Approximation Scheme (FAS). Finally, in section 4.2.4 we apply it to the M_1 model, and we highlight the distinctive features of radiative transfer, i.e., the preservation of the admissible states $E_r > 0$ and $f \leqslant 1$.

4.2.2 Linear case

In this section, we present the main ideas of the geometric multigrid method. It does not intend to be a full review of existing work, see e.g., Briggs et al. [2000]; Brandt and Livne [2011].

We are interested in solving the linear system

$$\mathcal{A}\boldsymbol{v} = \boldsymbol{f}, \tag{4.6}$$

with an iterative method, such as Jacobi [Saad, 2003], where $\mathcal{A}$ is an invertible matrix. We assume that this system is obtained by the discretization of a PDE and therefore depends on a step h. For example, let us consider the Poisson equation:

$$\begin{cases} -\Delta u(\boldsymbol{x}) = f(\boldsymbol{x}) \text{ for } \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x}) = 0 \text{ for } \boldsymbol{x} \in \partial\Omega, \end{cases} \tag{4.7}$$

where $\partial\Omega$ is the boundary of Ω. In the one-dimensional case, a finite difference discretization of eq. 4.7 leads to solving a linear system of the form given by eq. 4.6, with

$$\mathcal{A} = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \text{ and } \boldsymbol{f} = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}. \tag{4.8}$$

As mentioned above, the initial guess used in the Jacobi algorithm is the interpolation of a solution obtained on a coarser grid. This solution computed on the coarse grid can be itself computed thanks to a solution obtained on a third grid, even coarser. This process can be applied recursively until the last grid has only a few unknowns and the problem could be solved with a direct method, which leads to the so-called "nested iterations". From Briggs et al. [2000], this requires the definition of two elements:

— a hierarchy of grids, completed with restriction and prolongation operators;
— a smoother to solve the system on a given mesh.

Our goal is to solve nonlinear systems coming from the discretization of the M_1 model, then we will use the Jacobi method presented in section 4.1 as smoother. Therefore, we also use a Jacobi method as smoother in the linear case, and we do not investigate the impact of the smoother anymore. In the linear case, one can study the eigenvalues and eigenvectors of $\mathcal{A}$, as well as the iterative method to understand how the smoother allows the quick decrease of the high frequencies of the error.

The hierarchy of grids is handled through recursiveness, we only need to describe the method with two grids. The domain Ω is discretized with two Cartesian meshes: the first one with step h, written Ω_h and the second one with step $2h$, written Ω_{2h}. Ω_h contains 2^d times more elements than Ω_{2h}, where d is the dimension of the problem.

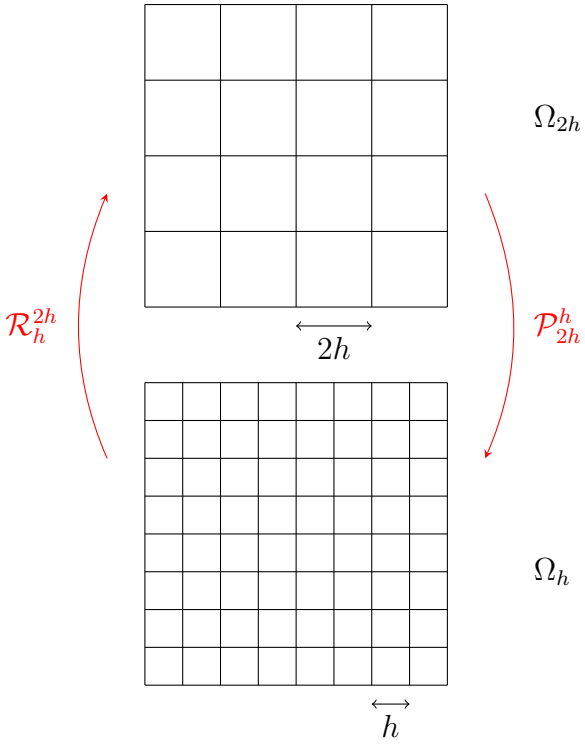


Figure 4.2 – Two-grid cycle.

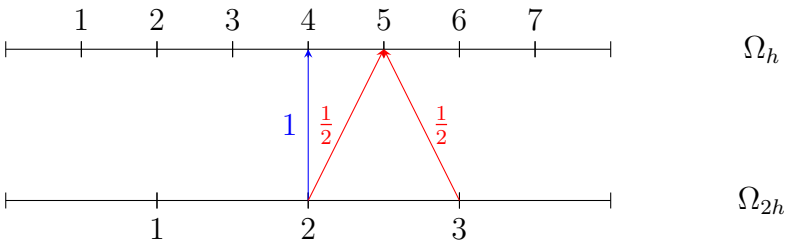


Figure 4.3 – Prolongation operator in the one-dimensional case. The blue arrow represents the operator if i^h is even and the red arrows represent the operator if i^h is odd.

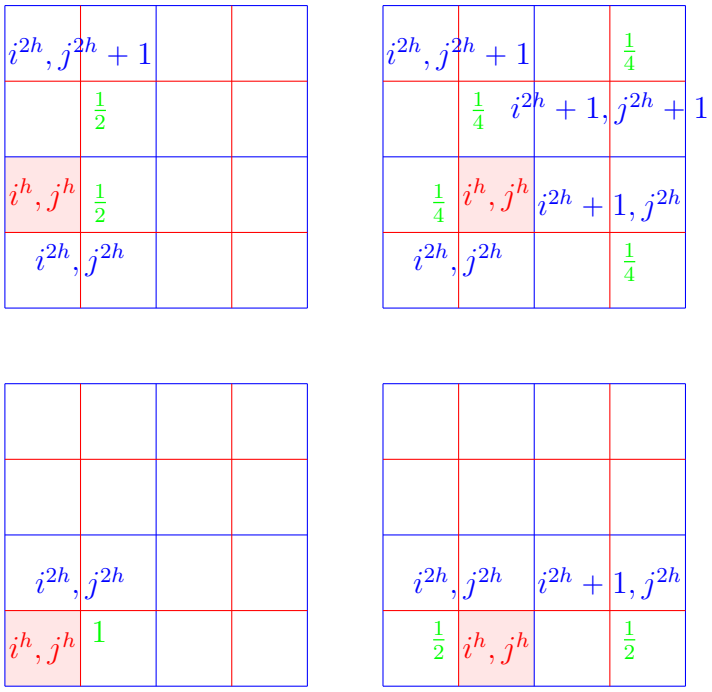


Figure 4.4 – Prolongation operator in the two-dimensional case. The coarse mesh Ω_{2h} is in blue, the fine mesh Ω_h is in red, the weight of the cells in the coarse mesh is in green. The computed cell is colored.

Let us now define the restriction and prolongation operators used for inter-grid operations. We write the prolongation operator $\mathcal{P}_{2h}^h : \Omega_{2h} \to \Omega_h$ and the restriction operator $\mathcal{R}_h^{2h} : \Omega_h \to \Omega_{2h}$, see fig. 4.2.

We begin with the prolongation operator. We use full weighting operators [Strang, 2006]. Let us consider the one-dimensional case, with an odd number of cells in the fine mesh. Indices i^h even in the fine mesh are the same as i^{2h} in the coarse mesh (see fig. 4.3). Values corresponding to these cells in the coarse mesh are just moved in the fine mesh. The other values in the fine mesh are obtained by linear interpolation. The prolongation operator writes

$$\left(\mathcal{P}_{2h}^h\left(\boldsymbol{v}^{2h}\right)\right)_{i^h} = \begin{cases} \boldsymbol{v}_{i^{2h}}^{2h} \text{ if } i^h \text{ is even,} \\ \dfrac{1}{2}\left(\boldsymbol{v}_{i^{2h}}^{2h} + \boldsymbol{v}_{i^{2h}+1}^{2h}\right) \text{ if } i^h \text{ is odd.} \end{cases} \tag{4.9}$$

We present now the prolongation operator in the two-dimensional case. A cell i^h, j^h in the fine mesh Ω_h can be mapped directly into the cell i^{2h}, j^{2h} in the coarse mesh Ω_{2h}, with $i^h = 2i^{2h}$ and $j^h = 2j^{2h}$. The value in a cell of the fine mesh is obtained by the interpolation in one direction, followed by the interpolation in the other direction. The interpolation in the x-direction gives

$$\left(\mathcal{P}_{2h}^h\left(\boldsymbol{v}^{2h}\right)\right)_{2i^{2h},2j^{2h}} = \boldsymbol{v}_{i^{2h},j^{2h}}^{2h} \text{ and } \left(\mathcal{P}_{2h}^h\left(\boldsymbol{v}^{2h}\right)\right)_{2i^{2h}+1,2j^{2h}} = \frac{1}{2}\left(\boldsymbol{v}_{i^{2h},j^{2h}}^{2h} + \boldsymbol{v}_{i^{2h}+1,j^{2h}}^{2h}\right). \tag{4.10}$$

The interpolation in the y-direction preserves these values and gives

$$\left(\mathcal{P}_{2h}^h\left(\boldsymbol{v}^{2h}\right)\right)_{2i^{2h},2j^{2h}+1} = \frac{1}{2}\left(\boldsymbol{v}_{i^{2h},j^{2h}}^{2h} + \boldsymbol{v}_{i^{2h},j^{2h}+1}^{2h}\right). \tag{4.11}$$

Finally, the last value is given by

$$\left(\mathcal{P}_{2h}^h\left(\boldsymbol{v}^{2h}\right)\right)_{2i^{2h}+1,2j^{2h}+1} = \frac{1}{4}\left(\boldsymbol{v}_{i^{2h},j^{2h}}^{2h} + \boldsymbol{v}_{i+1^{2h},j^{2h}}^{2h} + \boldsymbol{v}_{i^{2h},j^{2h}+1}^{2h} + \boldsymbol{v}_{i^{2h}+1,j^{2h}+1}^{2h}\right) \tag{4.12}$$

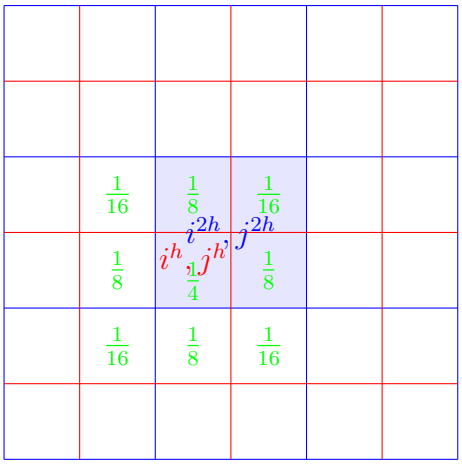


Figure 4.5 – Restriction operator. The coarse mesh Ω_{2h} is in blue, the fine mesh Ω_h is in red, the weight of the cells in the fine mesh is in green. The computed cell is colored.

Using eqs. 4.10 to 4.12 leads to writing the prolongation operator as

$$\left(\mathcal{P}_{2h}^{h}\left(\boldsymbol{v}^{2h}\right)\right)_{i^h,j^h} = \begin{cases} \boldsymbol{v}^{2h}_{i^{2h},j^{2h}} \text{ if } i^h \text{ is even and } j^h \text{ is even,} \\ \frac{1}{2}\left(\boldsymbol{v}^{2h}_{i^{2h},j^{2h}} + \boldsymbol{v}^{2h}_{i^{2h}+1,j^{2h}}\right) \text{ if } i^h \text{ is odd and } j^h \text{ is even,} \\ \frac{1}{2}\left(\boldsymbol{v}^{2h}_{i^{2h},j^{2h}} + \boldsymbol{v}^{2h}_{i^{2h},j+1^{2h}}\right) \text{ if } i^h \text{ is even and } j^h \text{ is odd,} \\ \frac{1}{4}\left(\boldsymbol{v}^{2h}_{i^{2h},j^{2h}} + \boldsymbol{v}^{2h}_{i+1^{2h},j^{2h}} + \boldsymbol{v}^{2h}_{i^{2h},j^{2h}+1} + \boldsymbol{v}^{2h}_{i^{2h}+1,j^{2h}+1}\right) \\ \quad \text{if } i^{2h} \text{ is odd and } j^{2h} \text{ is odd.} \end{cases} \tag{4.13}$$

See also fig. 4.4.

Let us now consider the restriction operator. We first consider the case where the mesh is given by fig. 4.3. The prolongation operator can be made explicit, with appropriate boundary conditions:

$$\boldsymbol{v}^h = \left(\mathcal{P}_{2h}^{h}\left(\boldsymbol{v}^{2h}\right)\right) = \begin{pmatrix} \boldsymbol{v}^h_1 \\ \boldsymbol{v}^h_2 \\ \boldsymbol{v}^h_3 \\ \boldsymbol{v}^h_4 \\ \boldsymbol{v}^h_5 \\ \boldsymbol{v}^h_6 \\ \boldsymbol{v}^h_7 \end{pmatrix} = \begin{pmatrix} \boldsymbol{v}^h_{2\times 0+1} \\ \boldsymbol{v}^h_{2\times 1} \\ \boldsymbol{v}^h_{2\times 1+1} \\ \boldsymbol{v}^h_{2\times 2} \\ \boldsymbol{v}^h_{2\times 2+1} \\ \boldsymbol{v}^h_{2\times 3} \\ \boldsymbol{v}^h_{2\times 3+1} \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \boldsymbol{v}^{2h}_1 \\ \boldsymbol{v}^{2h}_2 \\ \boldsymbol{v}^{2h}_3 \end{pmatrix}. \tag{4.14}$$

Even though the prolongation operator is written as an operator, it is linear and can therefore be seen as a matrix, $\mathcal{P}_{2h}^{h}$. Apart the first and the last rows that handle of boundary conditions, the sum of the coefficients in a row is 1. Values on the fine mesh are convex combinations of values on the coarse mesh.

We take the restriction operator as the transpose of the prolongation operator: $\mathcal{R}_h^{2h} = \frac{1}{2^d}\left(\mathcal{P}_{2h}^{h}\right)^T$, where d is the dimension of the problem ($d \in \{1,2,3\}$). Thanks to the factor $\frac{1}{2^d}$, values on the coarse mesh are also convex combinations of values on the fine mesh. In the one one-dimensional case given by fig. 4.3, the restriction operator

writes

$$\boldsymbol{v}^{2h} = \left(\mathcal{R}_h^{2h}\left(\boldsymbol{v}^{2h}\right)\right) = \begin{pmatrix} \boldsymbol{v}_1^{2h} \\ \boldsymbol{v}_2^{2h} \\ \boldsymbol{v}_3^{2h} \end{pmatrix} = \frac{1}{4}\begin{pmatrix} 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{v}_1^h \\ \boldsymbol{v}_2^h \\ \boldsymbol{v}_3^h \\ \boldsymbol{v}_4^h \\ \boldsymbol{v}_5^h \\ \boldsymbol{v}_6^h \\ \boldsymbol{v}_7^h \end{pmatrix}. \tag{4.15}$$

In the general one-dimensional case, we have

$$\left(\mathcal{R}_h^{2h}\left(\boldsymbol{v}^h\right)\right)_{i^{2h}} = \frac{1}{4}\boldsymbol{v}_{i^h-1}^h + \frac{1}{2}\boldsymbol{v}_{i^h}^h + \frac{1}{4}\boldsymbol{v}_{i^h+1}^h. \tag{4.16}$$

Numerical tests in section 4.3 are done in the two-dimensional case, we also make $\mathcal{R}_h^{2h}$ explicit with $d = 2$ (see also fig. 4.5):

$$\begin{aligned} \left(\mathcal{R}_h^{2h}\left(\boldsymbol{v}^h\right)\right)_{i^{2h},j^{2h}} = {} & \frac{1}{16}\boldsymbol{v}_{i^h-1,j^h-1}^h + \frac{1}{8}\boldsymbol{v}_{i^h-1,j^h}^h + \frac{1}{16}\boldsymbol{v}_{i^h-1,j^h+1}^h \\ & + \frac{1}{8}\boldsymbol{v}_{i^h,j^h-1}^h + \frac{1}{4}\boldsymbol{v}_{i^h,j^h}^h + \frac{1}{8}\boldsymbol{v}_{i^h,j^h+1}^h \\ & + \frac{1}{16}\boldsymbol{v}_{i^h+1,j^h-1}^h + \frac{1}{8}\boldsymbol{v}_{i^h+1,j^h}^h + \frac{1}{16}\boldsymbol{v}_{i^h+1,j^h+1}^h. \end{aligned} \tag{4.17}$$

Let us recall that we are interested in solving eq. 4.6 on the mesh Ω_h. We then rewrite it as

$$\mathcal{A}^h\boldsymbol{v}^h = \boldsymbol{f}^h, \tag{4.18}$$

to highlight the dependence on the mesh. To fix the notations, the discretization of eq. 4.7 on the coarse mesh Ω_{2h} leads to solving the following linear system:

$$\mathcal{A}^{2h}\boldsymbol{v}^{2h} = \boldsymbol{f}^{2h}. \tag{4.19}$$

Now that we have made explicit the restriction and prolongation operators used with two grids, we write the two-grid algorithm (algorithm 2).

Algorithm 2 Two-grid algorithm

Pre-smoother: relax ν_0 times $\mathcal{A}^h\boldsymbol{u}^h = \boldsymbol{f}^h$ with initial guess $\boldsymbol{v}^h$ on the fine mesh Ω_h
Restrict the current solution $\boldsymbol{v}^{2h} = \mathcal{R}_h^{2h}\left(\boldsymbol{u}^h\right)$ and the right-hand side $\boldsymbol{f}^{2h} = \mathcal{R}_h^{2h}\left(\boldsymbol{f}^h\right)$
Solve $\mathcal{A}^{2h}\boldsymbol{u}^{2h} = \boldsymbol{f}^{2h}$ with initial guess $\boldsymbol{v}^{2h}$ on the coarse mesh Ω_{2h}
Prolong the solution: $\bar{\boldsymbol{v}}^h = \mathcal{P}_{2h}^h\left(\boldsymbol{u}^{2h}\right)$
Post-smoother: relax ν_0 times $\mathcal{A}^h\boldsymbol{u}^h = \boldsymbol{f}^h$ with initial guess $\bar{\boldsymbol{v}}^h$ on the fine mesh Ω_h

This process can now be applied recursively until a mesh coarse enough is reached. This leads to the so-called V-cycle algorithm. Figure 4.6 shows the mesh size with different levels. Solving the linear system at a given level is less costly than the previous level because the number of unknowns is reduced. However, numerical experiments have shown that the best performances are obtained by performing a series of V-cycles. The algorithm is made explicit by algorithm 3. At the coarsest level, a direct solver could be used. In the case of radiative transfer, the development of a direct solver is out of reach. Therefore, we will also use the Jacobi method as a coarse grid solver.

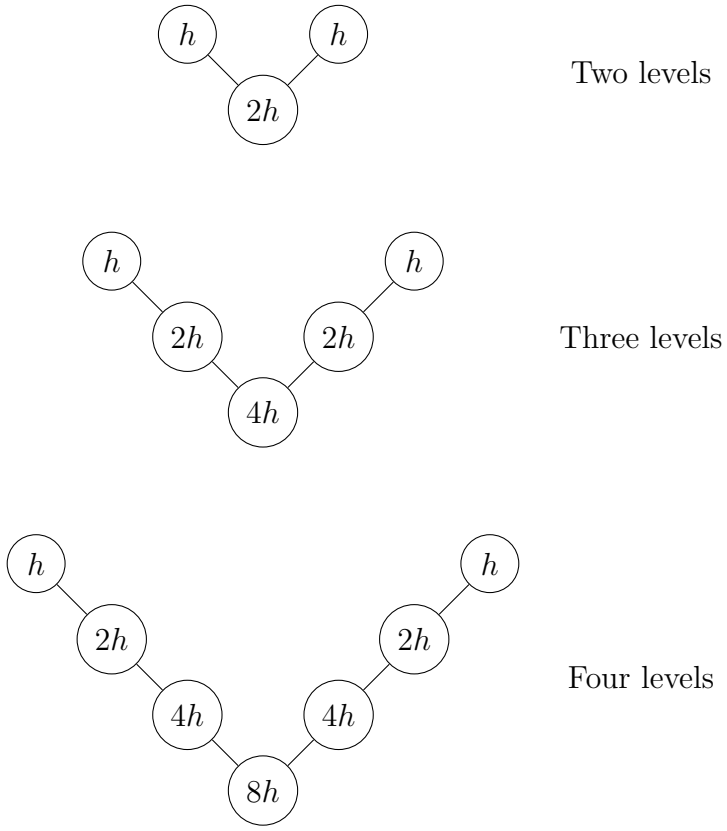


Figure 4.6 – V-cycle with two to four levels. Circles represent the different meshes, black lines represent the restriction and prolongation operators.

Algorithm 3 V-cycle

Choose a maximum number of levels $L_{\max}$
$\kappa = 0$
while $\left|\left|\mathcal{A}^h \boldsymbol{v}^h_{(\kappa)} - \boldsymbol{f}^h\right|\right| > \varepsilon$ **do**
  $\boldsymbol{v}^h_{(\kappa+1)} =$ V-CYCLE$(0, \boldsymbol{v}^h_{(\kappa)}, \boldsymbol{f}^h, L_{\max})$
  $\kappa \leftarrow \kappa + 1$
end while
function V-CYCLE$(l, \boldsymbol{v}^{2^l h}, \boldsymbol{f}^{2^l h}, L_{\max})$
  Pre-smoother: relax ν_l times $\mathcal{A}^{2^l h}\boldsymbol{u}^{2^l h} = \boldsymbol{f}^{2^l h}$ with initial guess $\boldsymbol{v}^{2^l h}$
  Restrict the current solution $\boldsymbol{v}^{2^{l+1}h} = \mathcal{R}^{2^{l+1}h}_{2^l h}\left(\boldsymbol{u}^{2^l h}\right)$ and the right-hand side $\boldsymbol{f}^{2^{l+1}h} = \mathcal{R}^{2^{l+1}h}_{2^l h}\left(\boldsymbol{f}^{2^l h}\right)$
  if $l + 1 = L_{\max}$ **then**
    Solve $\mathcal{A}^{2^{l+1}h}\boldsymbol{u}^{2^{l+1}h} = \boldsymbol{f}^{2^{l+1}h}$ with initial guess $\boldsymbol{v}^{2^{l+1}h}$
  else
    Recursion: $\boldsymbol{u}^{2^{l+1}h} =$V-CYCLE$(\ l+1, \boldsymbol{v}^{2^{l+1}h}, \boldsymbol{f}^{2^{l+1}h}, L_{\max}\ )$
  end if
  Prolong the solution: $\bar{\boldsymbol{v}}^{2^l h} = \mathcal{P}^{2^l h}_{2^{l+1}h}\left(\boldsymbol{u}^{2^{l+1}h}\right)$
  Post-smoother: relax ν_l times $\mathcal{A}^{2^l h}\boldsymbol{u}^{2^l h} = \boldsymbol{f}^{2^l h}$ with initial guess $\bar{\boldsymbol{v}}^{2^l h}$
  return $\boldsymbol{u}^{2^l h}$
end function

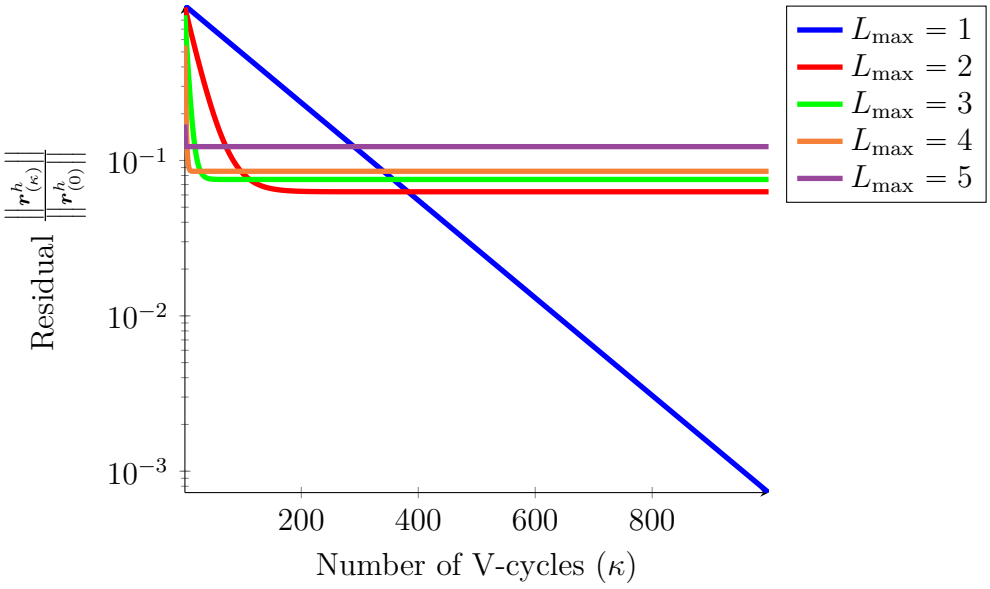


Figure 4.7 – Evolution of the residual as a function of the number of V-cycle (algorithm 2), with different values of $L_{\max}$.

Because we mimic this in the linear case, we do not investigate the effect of the coarse grid solver.

Now that we have presented the algorithm, let us consider a simple test. We solve eq. 4.7 in the two-dimensional case, with $\Omega = [0,1] \times [0,1]$ and $f(x,y) = 8\pi^2 \sin(2\pi x)\sin(2\pi y)$. Equation 4.7 has an exact solution: $\bar{u}(x,y) = \sin(2\pi x)\sin(2\pi y)$.

We discretize Ω with 129×129 cells and apply algorithm 3. Figure 4.7 shows the evolution of the residual $\left|\left|\mathcal{A}^h \boldsymbol{v}^h - \boldsymbol{f}^h\right|\right|$ as a function of the number of V-cycles, with different numbers of levels, $L_{\max}$. With $L_{\max} = 1$, we recover the classical Jacobi method. Apart from $L_{\max} = 1$, the residual drops quickly, as $L_{\max}$ increases, but it reaches a plateau, around 10^{-1}, depending on the value of $L_{\max}$. The same phenomena have been observed with different resolutions.

Algorithm 4 Two-grid correction scheme

Pre-smoother: relax ν_0 times $\mathcal{A}^h \boldsymbol{u}^h = \boldsymbol{f}^h$ with initial guess $\boldsymbol{v}^h$ on the fine mesh Ω_h
Restrict the residual $\boldsymbol{r}^{2h} = \mathcal{R}^{2h}_h \left(\boldsymbol{f}^h - \mathcal{A}^h \boldsymbol{u}^h\right)$
Solve $\mathcal{A}^{2h} \boldsymbol{e}^{2h} = \boldsymbol{r}^{2h}$ on the coarse grid with initial guess $\mathbf{0}$ on the coarse mesh Ω_{2h}
Correct the fine grid approximation $\bar{\boldsymbol{v}}^h = \boldsymbol{v}^h + \mathcal{P}^h_{2h}\left(\boldsymbol{e}^{2h}\right)$
Post-smoother: relax ν_0 times $\mathcal{A}^h \boldsymbol{u}^h = \boldsymbol{f}^h$ with initial guess $\bar{\boldsymbol{v}}^h$ on the fine mesh Ω_h

One can show that the presence of smooth modes in the error between the approximate and exact solution $\bar{u}$ can cause the relaxation to stall. The proof relies on the study of the eigenvalues and eigenvectors of $\mathcal{A}$, see e.g., Brandt and Livne 2011. As shown by fig. 4.8, a smooth function on the fine mesh will be less smooth on the coarse mesh. An iterative method as Jacobi will converge faster and will smooth the error on the coarse mesh. This gives the idea of restrict and prolong the error between the different meshes. If $\boldsymbol{v}$ is an approximate solution of eq. 4.6 and $\boldsymbol{u}$ is the exact solution, let us define the error as $\boldsymbol{e} = \boldsymbol{u} - \boldsymbol{v}$. It verifies $\mathcal{A}\boldsymbol{e} = \mathcal{A}\boldsymbol{u} - \mathcal{A}\boldsymbol{v} = \boldsymbol{f} - \mathcal{A}\boldsymbol{v}$. By defining the residual $\boldsymbol{r} = \boldsymbol{f} - \mathcal{A}\boldsymbol{v}$, the error verifies the following equation:

$$\mathcal{A}\boldsymbol{e} = \boldsymbol{r}. \tag{4.20}$$

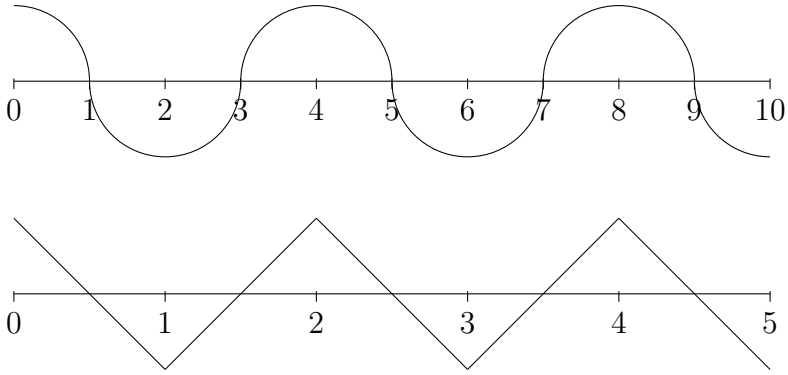


Figure 4.8 – Restriction of a smooth function.

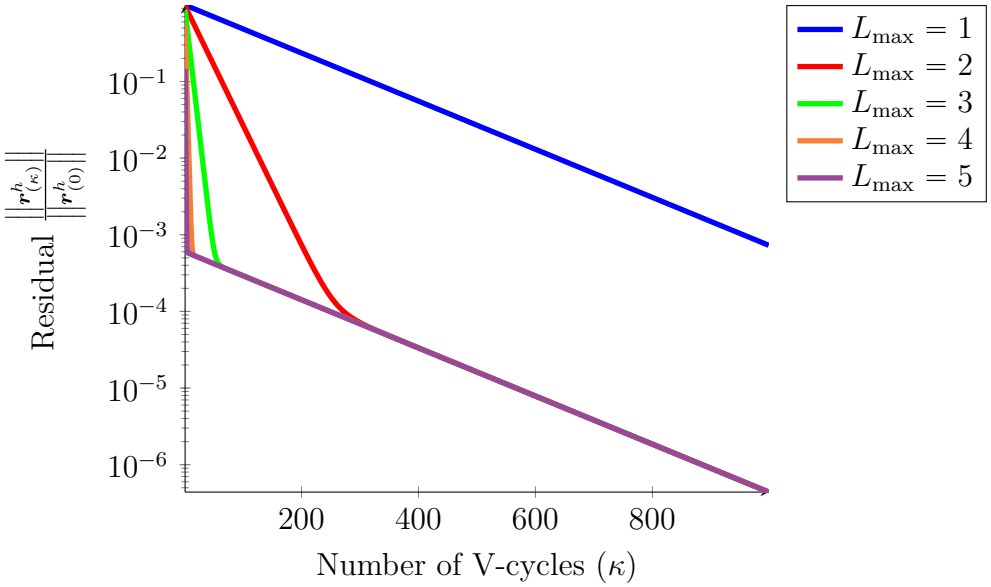


Figure 4.9 – Evolution of the residual as a function of the number of V-cycle (algorithm 4), with different values of $L_{\max}$.

We will solve the residual equation eq. 4.20 on the coarse mesh. Once the error is computed, a new approximate solution can be computed: $\boldsymbol{v}+\boldsymbol{e}$. We replace algorithm 2 with algorithm 4, which leads to the so-called "correction scheme".

Figure 4.9 shows the evolution of the norm of residual as a function of the number of V-cycles, using the correction scheme. Unlike fig. 4.7, the residual drops quickly to 0. It decreases faster as $L_{\max}$ increases, as expected. When the residual is small enough, around 10^{-4}, all curves converge with the same slope. The low resolution at the coarsest meshes is responsible for that.

We have first presented the algorithm without the error and the residual because it would have been easier to adapt for radiative transfer. As shown in section 4.1.3, if the initial guess and the right-hand side are admissible, then the solution obtained with the Jacobi method is also admissible. The restriction and prolongation operators used here (eqs. 4.13 and 4.17) are convex combinations of admissible states, therefore the restricted and prolonged vectors are also admissible. Unfortunately, as shown here, this method does not converge. Because the error and the residual are not admissible in the general case, the algorithm has to be modified for the M_1 model (section 4.2.4).

4.2.3 General nonlinear case

Now that we have presented the multigrid method in the linear case, let us focus on the nonlinear case. Instead of solving eq. 4.6, we solve

$$\mathcal{A}(\boldsymbol{v}) = \boldsymbol{f}, \tag{4.21}$$

where $\mathcal{A}$ is a nonlinear operator. We suppose that we have an iterative method to solve eq. 4.21, such as nonlinear Jacobi or Gauss-Seidel.

To mimic the linear case, we want to use the residual equation on the coarse mesh to approximate the error on the fine mesh $\boldsymbol{e}^h$. Let us recall that the error is defined as $\boldsymbol{e} = \boldsymbol{u} - \boldsymbol{v}$, where $\boldsymbol{u}$ is the exact solution of eq. 4.21 and $\boldsymbol{v}$ is an approximate solution. We also define the residual as $\boldsymbol{r} = \boldsymbol{f} - \mathcal{A}(\boldsymbol{v})$. The residual equation eq. 4.20 is no longer verified. The definition of the residual leads to

$$\begin{aligned} \boldsymbol{r} &= \boldsymbol{f} - \mathcal{A}(\boldsymbol{v}) \\ &= \mathcal{A}(\boldsymbol{u}) - \mathcal{A}(\boldsymbol{v}) \\ &= \mathcal{A}(\boldsymbol{v} + \boldsymbol{e}) - \mathcal{A}(\boldsymbol{v}). \end{aligned} \tag{4.22}$$

We can now solve eq. 4.22 on the coarse mesh:

$$\mathcal{A}^{2h}\left(\boldsymbol{v}^{2h} + \boldsymbol{e}^{2h}\right) - \mathcal{A}^{2h}\left(\boldsymbol{v}^{2h}\right) = \boldsymbol{r}^{2h}, \tag{4.23}$$

where $\boldsymbol{r}^{2h} = \mathcal{R}_h^{2h}\left(\boldsymbol{r}^h\right) = \mathcal{R}_h^{2h}\left(\boldsymbol{f}^h - \mathcal{A}^h\left(\boldsymbol{v}^h\right)\right)$ is an approximation of the residual on the coarse mesh. Likewise, $\boldsymbol{v}^{2h} = \mathcal{R}_h^{2h}\left(\boldsymbol{v}^h\right)$. By writing $\boldsymbol{u}^{2h} = \boldsymbol{v}^{2h} + \boldsymbol{e}^{2h}$, the error can be computed by solving

$$\mathcal{A}^{2h}\left(\boldsymbol{u}^{2h}\right) = \mathcal{A}^{2h}\left(\boldsymbol{v}^{2h}\right) + \boldsymbol{r}^{2h}, \tag{4.24}$$

whose right-hand side is known. Once $\boldsymbol{u}^{2h}$ is computed by solving eq. 4.24, the new approximate solution on the fine mesh can be computed:

$$\boldsymbol{v}^h + \mathcal{P}_{2h}^h\left(\underbrace{\boldsymbol{u}^{2h} - \boldsymbol{v}^{2h}}_{e^{2h}}\right). \tag{4.25}$$

This leads to the full approximation scheme (FAS) algorithm 5 [Briggs et al., 2000].

Algorithm 5 Two-grid full approximation scheme

Pre-smoother: relax ν_0 times $\mathcal{A}^h\left(\boldsymbol{u}^h\right) = \boldsymbol{f}^h$ with initial guess $\boldsymbol{v}^h$ on the fine mesh Ω_h
Restrict the residual $\boldsymbol{r}^{2h} = \mathcal{R}_h^{2h}\left(\boldsymbol{f}^h - \mathcal{A}^h\left(\boldsymbol{u}^h\right)\right)$ and the current solution $\boldsymbol{v}^{2h} = \mathcal{R}_h^{2h}\left(\boldsymbol{u}^h\right)$
Solve $\mathcal{A}^{2h}\left(\boldsymbol{u}^{2h}\right) = \boldsymbol{r}^{2h} + \mathcal{A}^{2h}\left(\boldsymbol{v}^{2h}\right)$ with initial guess $\boldsymbol{v}^{2h}$ on the coarse mesh Ω_{2h}
Correct the approximation on the fine mesh: $\bar{\boldsymbol{v}}^h = \boldsymbol{v}^h + \mathcal{P}_{2h}^h\left(\boldsymbol{u}^{2h} - \boldsymbol{v}^{2h}\right)$
Post-smoother: relax ν_0 times $\mathcal{A}^h\left(\boldsymbol{u}^h\right) = \boldsymbol{f}^h$ with initial guess $\bar{\boldsymbol{v}}^h$ on the fine mesh Ω_h

The solution of the problem solved on the coarse mesh is the full approximation $\boldsymbol{u}^{2h} = \boldsymbol{v}^{2h} + \boldsymbol{e}^{2h}$, and not the error $\boldsymbol{e}^{2h}$. Algorithm 5 is presented for only two meshes, but it can easily be extended to a higher number of meshes using a V-cycle, as done in section 4.2.2. If $\mathcal{A}$ is linear, one can show that algorithm 5 is equivalent to algorithm 4.

4.2.4 Application to the HLL solver for the M_1 model

Let us now apply algorithm 5 to the HLL solver for the M_1 model, without source terms. The operator $\mathcal{A}$ is the one described in section 4.1, and we use algorithm 1 as smoother and coarse grid solver. The system to be solved on the coarse mesh is

$$\mathcal{A}^{2h}\left(\boldsymbol{u}^{2h}\right) = \mathcal{A}^{2h}\left(\boldsymbol{v}^{2h}\right) + \boldsymbol{r}^{2h}. \tag{4.26}$$

As shown in section 4.1.3, if the initial guess for $\boldsymbol{v}^{2h}$ and the right-hand side $\mathcal{A}^{2h}\left(\boldsymbol{v}^{2h}\right) + \boldsymbol{r}^{2h}$ are admissible, then the solution obtained with algorithm 1 is also admissible. However, numerical experiments have shown that, in general, $\mathcal{A}^{2h}\left(\boldsymbol{v}^{2h}\right) + \boldsymbol{r}^{2h}$ is not admissible, which leads to a non-admissible solution.

To tackle this issue, we follow Kifonidis and Müller [2012] and we introduce a pseudo-time τ. Instead of solving eq. 4.26, we look for the steady state in pseudo-time of the following equation:

$$\frac{\mathrm{d}\boldsymbol{u}^{2h}}{\mathrm{d}\tau} + \mathcal{A}^{2h}\left(\boldsymbol{u}^{2h}\right) = \mathcal{A}^{2h}\left(\boldsymbol{v}^{2h}\right) + \boldsymbol{r}^{2h}. \tag{4.27}$$

When the steady state is reached, $\frac{\mathrm{d}\boldsymbol{u}^{2h}}{\mathrm{d}\tau} = 0$, and we recover eq. 4.26. Let us notice that the pseudo-time τ is completely independent of the physical time step Δt.

Equation 4.27 is a (nonlinear) system of ordinary differential equations in the variable τ. We use notation similar to the physical time for the discretization in pseudo-time. $\Delta\tau$ is the interval between the current pseudo-time τ^m and τ^{m+1}. We choose $\Delta\tau$ such that $\boldsymbol{u}^{2h}$ is admissible. Using the definition of the residual, the right-hand side of eq. 4.27 becomes $\mathcal{A}^{2h}\left(\boldsymbol{v}^{2h}\right) + \boldsymbol{f}^{2h} - \mathcal{R}_h^{2h}\left(\mathcal{A}^h\left(\boldsymbol{v}^h\right)\right)$. We want to solve eq. 4.27 with a stable scheme, for all $\Delta\tau$. Therefore, we use a splitting strategy. For the scheme to be stable, the left-hand side has to be taken implicitly. This leads to solving eq. 4.27 as

$$\begin{cases} \dfrac{\widetilde{\boldsymbol{u}^{2h}} - \left(\boldsymbol{u}^{2h}\right)^m}{\Delta\tau} = \mathcal{A}^{2h}\left(\boldsymbol{v}^{2h}\right) + \boldsymbol{f}^{2h} - \mathcal{R}_h^{2h}\left(\mathcal{A}^h\left(\boldsymbol{v}^h\right)\right) \\ \dfrac{\left(\boldsymbol{u}^{2h}\right)^{m+1} - \widetilde{\boldsymbol{u}^{2h}}}{\Delta\tau} + \mathcal{A}^{2h}\left(\left(\boldsymbol{u}^{2h}\right)^{m+1}\right) = 0. \end{cases} \tag{4.28}$$

The first equation in eq. 4.28 is explicit in pseudo-time. We can always choose a value for $\Delta\tau$ such that $\widetilde{\boldsymbol{u}^{2h}}$ is admissible. Let us notice that the right-hand side is fixed and the left-hand side is local to a cell.

The second equation in eq. 4.28 is implicit in pseudo-time. With arguments similar to section 4.1.3, $\left(\boldsymbol{u}^{2h}\right)^{m+1}$ is admissible as soon as $\widetilde{\boldsymbol{u}^{2h}}$ is admissible.

Unfortunately, choosing $\Delta\tau$ such that $\widetilde{\boldsymbol{u}^{2h}}$ is admissible can result in small values for $\Delta\tau$. To reduce the computational cost, we slightly change the algorithm. We do not use the same pseudo-time step in all cells and for the implicit step. We choose a pseudo-time step $\Delta\tau^{\mathrm{im}}$ for the implicit step and we iterate the explicit step with another pseudo-time step, local to each cell, until $\Delta\tau^{\mathrm{im}}$ is reached. This results in algorithm 6. We use this algorithm as smoother and coarse solver in algorithm 5.

The next question to be solved is how to choose $\Delta\tau^{\mathrm{im}}$. Numerical experiments have shown that a large value for $\Delta\tau^{\mathrm{im}}$ allows reducing the computational cost per V-cycle, but can result in a very low decrease of the norm of the residual. To avoid such a result, we use an adaptive pseudo-time step. When the norm of the residual decreases fast enough, $\Delta\tau^{\mathrm{im}}$ is increased. On the contrary, when the norm of the residual decreases slowly, $\Delta\tau^{\mathrm{im}}$ is decreased.

Algorithm 6 Resolution at coarse level using pseudo-time

Choose $\Delta\tau^{\text{im}}$
$m = 0$
while the steady state is not reached **do**
  for each cell i^{2h} in the coarse grid **do**
    $\tau^{\text{ex}}_{i^{2h}} = 0$
    $K = 0$
    $\left(\widetilde{\boldsymbol{u}^{2h}}\right)^{(0)}_{i^{2h}} = \left(\boldsymbol{u}^{2h}\right)^{m}_{i^{2h}}$
    while $\tau^{\text{ex}}_{i^{2h}} < \Delta\tau^{\text{im}}$ **do**
      $\left(\widetilde{\boldsymbol{u}^{2h}}\right)^{(K+1)}_{i^{2h}} = \left(\widetilde{\boldsymbol{u}^{2h}}\right)^{(K)}_{i^{2h}} + \Delta\tau^{\text{ex}}_{i^{2h}} \left(\left(\mathcal{A}^{2h}\left(\boldsymbol{v}^{2h}\right)\right)_{i^{2h}} + \boldsymbol{f}^{2h}_{i^{2h}} - \left(\mathcal{R}^{2h}_{h}\left(\mathcal{A}^{h}\left(\boldsymbol{v}^{h}\right)\right)\right)_{i^{2h}}\right)$
with $\Delta\tau^{\text{ex}}_{i^{2h}}$ such that $\left(\widetilde{\boldsymbol{u}^{2h}}\right)^{(K+1)}_{i^{2h}}$ is admissible
      $K \leftarrow K + 1$
      $\tau^{\text{ex}}_{i^{2h}} \leftarrow \tau^{\text{ex}}_{i^{2h}} + \Delta\tau^{\text{ex}}_{i^{2h}}$
    end while
    $\widetilde{\boldsymbol{u}^{2h}}_{i^{2h}} = \left(\widetilde{\boldsymbol{u}^{2h}}\right)^{(K)}_{i^{2h}}$
  end for
  Solve $\left(\boldsymbol{u}^{2h}\right)^{m+1} + \Delta\tau^{\text{im}}\mathcal{A}^{2h}\left(\left(\boldsymbol{u}^{2h}\right)^{m+1}\right) = \widetilde{\boldsymbol{u}^{2h}}$ with nonlinear Jacobi method (algorithm 1)
  $m \leftarrow m + 1$
end while

The update of the solution with the correction on the fine grid

$$\overline{\boldsymbol{v}^{h}} = \boldsymbol{v}^{h} + \mathcal{P}^{h}_{2h}\left(\boldsymbol{u}^{2h} - \boldsymbol{v}^{2h}\right) \tag{4.29}$$

can also result in non-admissible states. Let us introduce another pseudo-time $\overline{\tau}$ to compute the steady state of

$$\frac{\mathrm{d}\overline{\boldsymbol{v}^{h}}}{\mathrm{d}\overline{\tau}} + \overline{\boldsymbol{v}^{h}} = \boldsymbol{v}^{h} + \mathcal{P}^{h}_{2h}\left(\boldsymbol{u}^{2h} - \boldsymbol{v}^{2h}\right), \tag{4.30}$$

instead of using eq. 4.29. We write $\Delta\overline{\tau}$ the interval between the current pseudo-time $\overline{\tau}^{\overline{m}}$ and $\overline{\tau}^{\overline{m}+1}$. We can now discretize eq. 4.30 with a time-implicit solver:

$$\frac{\left(\overline{\boldsymbol{v}^{h}}\right)^{\overline{m}+1} - \left(\overline{\boldsymbol{v}^{h}}\right)^{\overline{m}}}{\Delta\overline{\tau}} + \left(\overline{\boldsymbol{v}^{h}}\right)^{\overline{m}+1} = \boldsymbol{v}^{h} + \mathcal{P}^{h}_{2h}\left(\boldsymbol{u}^{2h} - \boldsymbol{v}^{2h}\right). \tag{4.31}$$

We choose $\Delta\overline{\tau}$ such that

$$\left(\overline{\boldsymbol{v}^{h}}\right)^{\overline{m}+1} = \frac{\left(\overline{\boldsymbol{v}^{h}}\right)^{\overline{m}} + \Delta\overline{\tau}\left(\boldsymbol{v}^{h} + \mathcal{P}^{h}_{2h}\left(\boldsymbol{u}^{2h} - \boldsymbol{v}^{2h}\right)\right)}{1 + \Delta\overline{\tau}} \tag{4.32}$$

is admissible. This process can be applied locally, only in the cells where it is needed.

The whole algorithm, including these modifications, is presented in section 4.A.

In the next section, we present numerical results to show the gain in computational time.

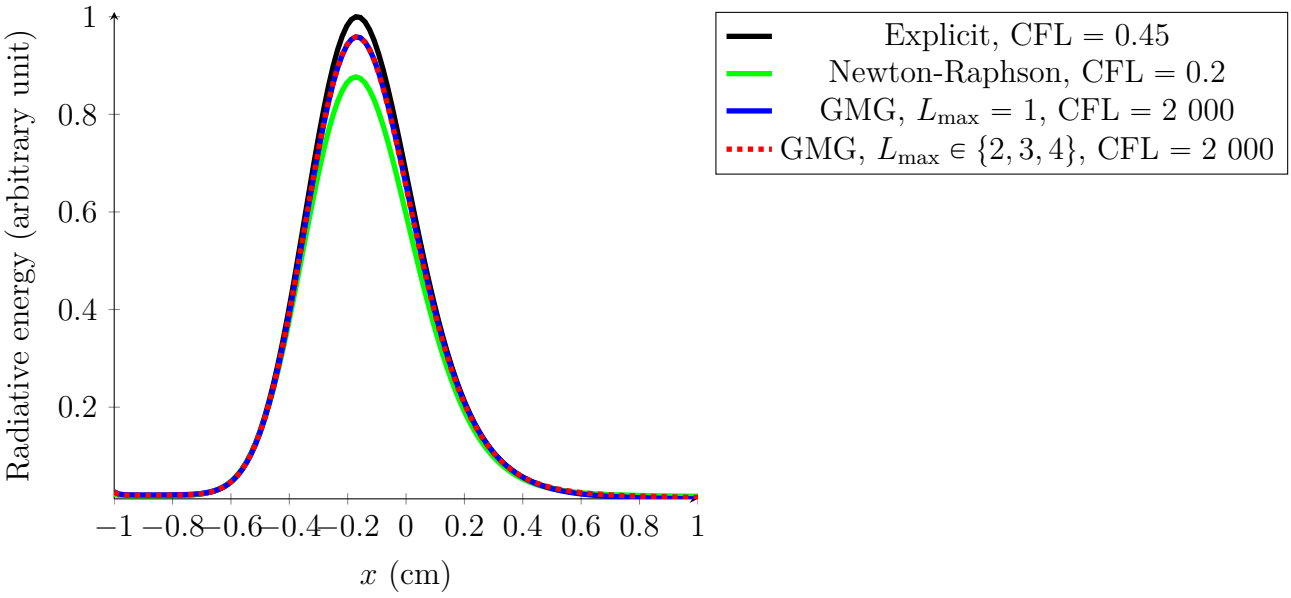


Figure 4.10 – Beam simulation. The figure shows a horizontal cut at the middle height with the explicit solver, the implicit solver using Newton-Raphson method and the geometric multigrid algorithm different values for $L_{\max}$.

4.3 Numerical results

We perform some verification tests to validate the performances of the algorithms presented in sections 4.1 and 4.2. Because we developed them without source terms, we focus on tests without opacity. We consider the same tests as in section 3.2.1. We compare our results with those obtained with a time-explicit HLL solver and a time-implicit HLL solver using a Newton-Raphson method (see section 2.1.3). We will compare the performances with different time steps. We write $\Delta t = \text{CFL}\frac{h}{c}$. For the time-explicit solver, one should have CFL $\leqslant 1$ to respect the Courant-Friedrichs-Lewy (CFL) condition. The eigenvalues in the numerical fluxes (eq. 2.2) are fixed to $\pm c$.

We always use the same parameters for the geometric multigrid method. At the finest level, the number of iterations for the pre- and post-smoothers is $\nu_0 = 3$. When using iterations in pseudo-time, the number of iterations in pseudo-time m is set to 3 and the number of iterations for the smoothers is $\nu_l = 1$, for $l \neq 0$. The initial value for $\Delta\tau^{\text{im}}$ is 10^{-3}. These parameters are chosen because they give reasonable performances in most cases and can easily be used in physical problems.

4.3.1 Beam

We first consider the same test as in sections 2.2.3 and 3.2.1.2. It is the same test as in González et al. [2007]; Richling et al. [2001]. We consider the propagation of a beam in the free-streaming regime. The domain is $[-1, 1] \times [-1, 1]$ and it is discretized with the same number of cells in the x and y-directions. The initial temperature is $T_0 = T_{\text{r}} = 300$ K and the initial radiative flux is $\boldsymbol{F}_{\mathbf{r}} = \mathbf{0}$. At time $t = 0$, a beam is introduced with $T_{\text{g}} = T_{\text{r}} = 1\,000$ K with an angle of 45° at $x = -1$ and $y \in [-0.875, -0.75]$.

The domain is first discretized with 129×129 cells. Figure 4.10 shows a horizontal cut at the middle height once the steady state is reached. With the Newton-Raphson method, we are not able to reach large time steps. Indeed, we have to use a smaller

Scheme	Computational time (s)
Explicit	549
Newton-Raphson	26 356
GMG, $L_{\max} = 1$	182
GMG, $L_{\max} = 2$	154
GMG, $L_{\max} = 3$	143
GMG, $L_{\max} = 4$	68

Table 4.2 – Computation time to reach the steady state with the explicit solver, the implicit solver using Newton-Raphson method and the geometric multigrid algorithm different values for $L_{\max}$.

$L_{\max}$	Computational time per V-cycle (s)
1	0.48
2	1.41
3	1.67
4	1.75

Table 4.3 – Computational time per V-cycle with different values of $L_{\max}$.

time step than the one required by the explicit scheme to preserve the admissible states. Despite that, the solution obtained with the Newton-Raphson method is more diffusive than the one obtained with the explicit scheme. It reaches only 89% of the maximum value of radiative energy. On the contrary, solutions obtained with the Jacobi method ($L_{\max} = 1$) and with the geometric multigrid algorithm ($L_{\max} = 2$) reach 96% of the maximum of radiative energy, with a much larger time step (CFL = 2 000).

Using the geometric multigrid method should reduce the computational cost. If we reach a low residual such as $\frac{\left|\left|\boldsymbol{r}^h_{(\kappa)}\right|\right|}{\left|\left|\boldsymbol{r}^h_{(0)}\right|\right|} = 10^{-5}$, numerical experiments have shown that there is no gain in computational cost. Therefore, we set it to 10^{-2} and we check some properties of the scheme in section 4.3.2.

Table 4.2 shows the computational time needed to reach the steady state with different methods: the explicit HLL solver, the implicit HLL solver using the Newton-Raphson method to solve nonlinear solver and the geometric multigrid algorithm with different values for $L_{\max}$. The resolution used is now 257×257 cells. Using the explicit solver and the Newton-Raphson method, the time step is restricted by the CFL condition, whereas the steady state is reached with only one iteration using the geometric multigrid method. With $L_{\max} = 1$, we recover the Jacobi method and the steady state is reached three times faster than using the explicit solver. Increasing the value of $L_{\max}$ leads again to a decrease in computational cost. The time needed to reach the steady state with $L_{\max} = 1$ is more than twice the time needed with $L_{\max} = 4$.

As shown by table 4.3, when $L_{\max}$ increases, the computational time per V-cycle also increases, from 0.48 s with $L_{\max} = 1$ to 1.75 s with $L_{\max} = 4$. With $L_{\max} > 1$, the computational time per V-cycle is more than three times the computational time per cycle with $L_{\max} = 1$. However, the total computational time (table 4.2) decreases when $L_{\max}$ increases. As shown by fig. 4.11, the number of V-cycles to reach the same residual decreases as $L_{\max}$ increases, from 1 267 with $L_{\max} = 1$ to 400 with $L_{\max} = 4$.

Table 4.4 shows the memory consumption with the different methods. As described in section 2.1.3, the Newton-Raphson method requires solving large sparse linear systems. For performance reasons, we store the matrix and its preconditioner. This leads

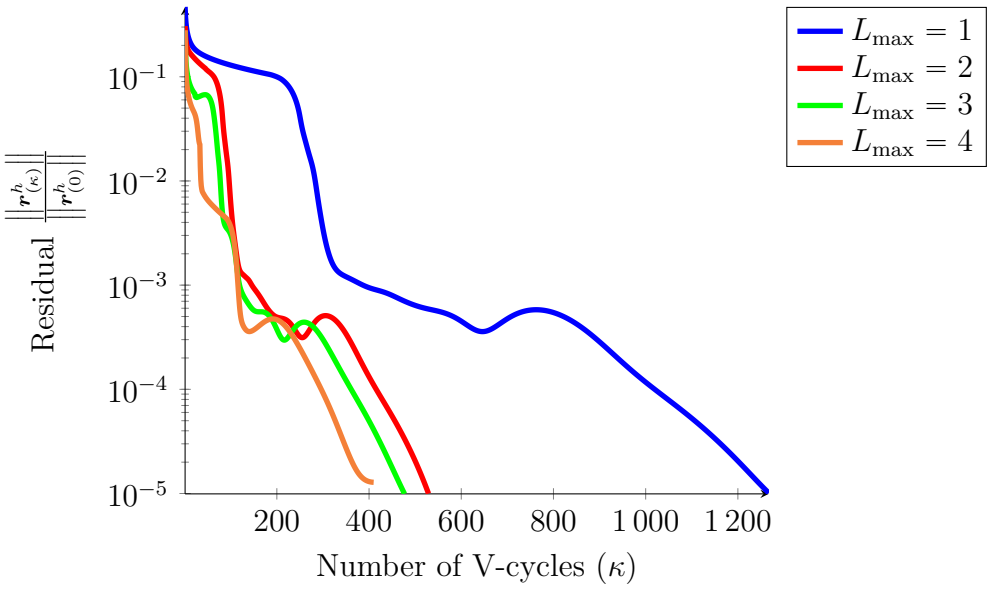


Figure 4.11 – Evolution of the residual as a function of the number of V-cycle, with different values for $L_{\max}$ to reach the steady state for the beam problem.

Scheme	Memory consumption (MB)
Explicit	25
Newton-Raphson	127
$L_{\max} = 1$	63
$L_{\max} = 2$	74
$L_{\max} = 3$	77
$L_{\max} = 4$	78

Table 4.4 – Memory consumption for the explicit solver, the implicit solver using Newton-Raphson method and the geometric multigrid algorithm different values for $L_{\max}$.

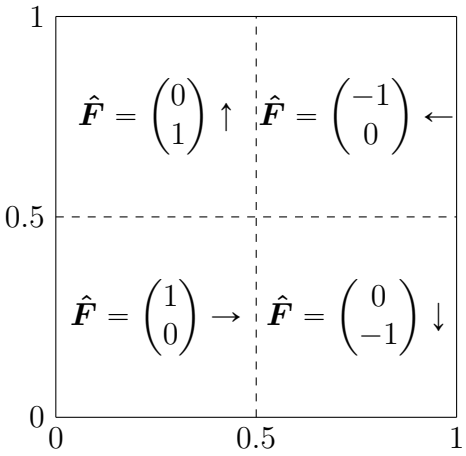


Figure 4.12 – Initial condition for 2D Riemann problem.

to a higher memory footprint than the other methods, 127 MB instead of 25 MB for the explicit solver. Furthermore, the choice of the preconditioner has an impact on the memory consumption of the method. This is discussed in section 5.2. We use a "matrix-free" approach for the Jacobi method, i.e., we do not store the operator $\mathcal{A}$, but we access it by computing $\mathcal{A}(\boldsymbol{v})$. However, using the geometric multigrid algorithm requires storing temporary values at coarse levels, hence the increase of memory consumption with $L_{\max}$, from 63 MB with $L_{\max} = 1$ to 78 MB with $L_{\max} = 4$. As $L_{\max}$ increases, more values have to be stored, but each level is coarser than the previous one, therefore fewer variables per additional level are needed. For example, with 257×257 cells at the fine level and $L_{\max} = 4$, there is only 32×32 additional cells compared to $L_{\max} = 3$.

Performance results obtained here are obtained on small configurations: 257×257 cells for a two-dimensional problem. One can expect a better speed-up in the three-dimensional case.

4.3.2 2D Riemann problem

Let us now consider the same two-dimensional Riemann problem as in section 3.2.1.1 and Blachère and Turpault [2016]. The domain $[0, 1] \times [0, 1]$ is discretized with 257×257 cells. The initial temperature is $T_0 = T_{\mathrm{r}} = 1\,000$ K. The domain is cut into four states, in each of them the initial radiative flux is constant. It is set to $(1 - 10^{-8})\, cE_{\mathrm{r}}\hat{\boldsymbol{F}}$, with $\hat{\boldsymbol{F}}$ given by fig. 4.12.

Using periodic boundary conditions, no energy should enter nor leave the box, therefore, the total radiative energy should be conserved at machine precision. Figure 4.13 shows the evolution of the relative error between the expected total radiative energy and the one actually computed in the box at each time step. Using both the explicit solver and the Jacobi method, with the same time step Δt, the relative error oscillates around the value 10^{-12}. For the Jacobi method, the residual $\frac{\left|\left|\boldsymbol{r}^h_{(\kappa)}\right|\right|}{\left|\left|\boldsymbol{r}^h_{(0)}\right|\right|}$ is set to 5×10^{-3}. Even for a quite high value of residual, the scheme is conservative.

We only checked that the Jacobi method ($L_{\max} = 1$) is conservative for performance reasons. As shown by fig. 4.14, with a large time step (CFL = 2 000), increasing the value of $L_{\max}$ does not help to reduce the number of V-cycle to reach the same residual. When increasing the value of $L_{\max}$, the computational time per V-cycle increases,

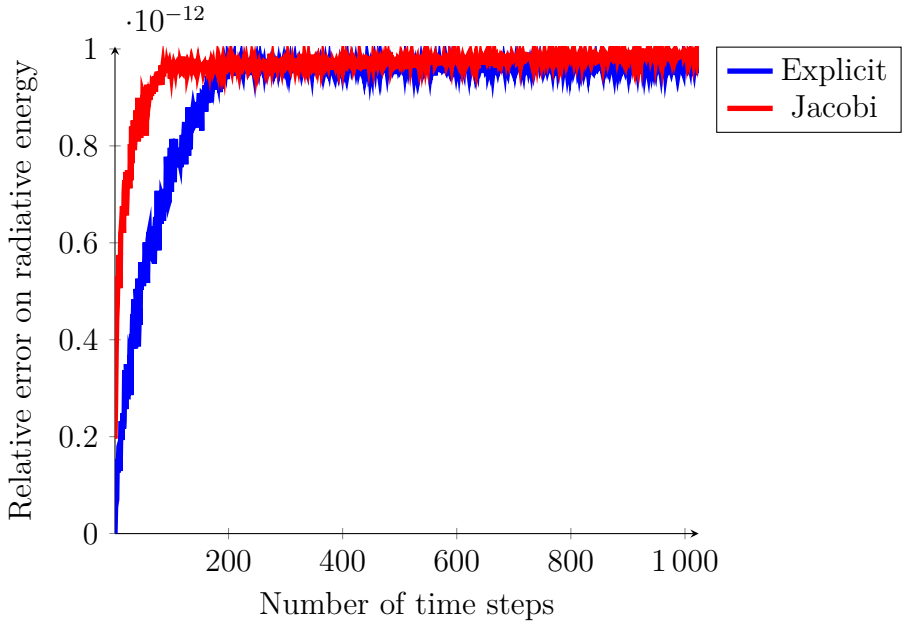


Figure 4.13 – Evolution of the relative error on radiative energy as a function of time. With periodic boundary conditions, the radiative energy is conserved at machine precision.

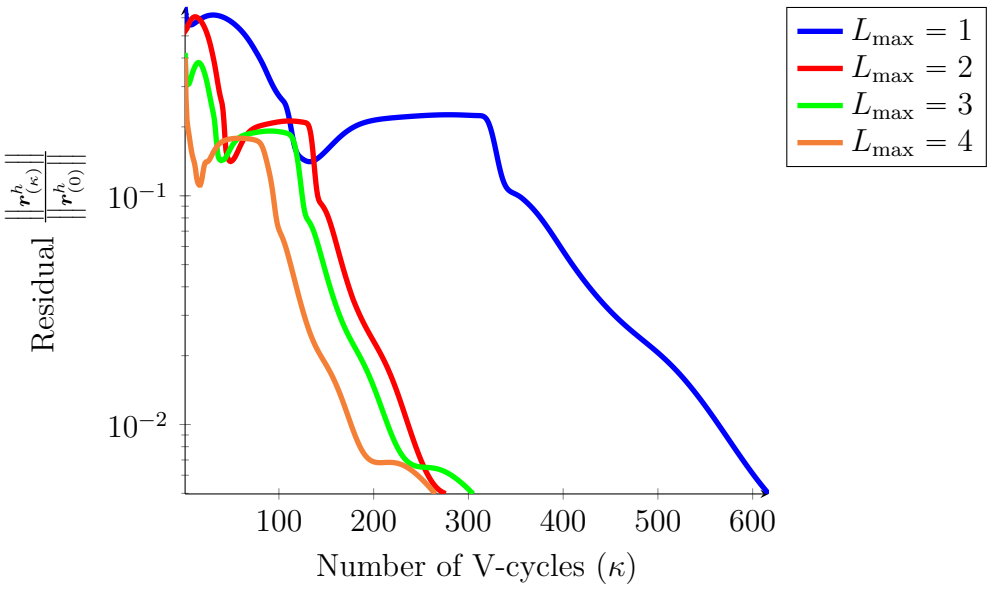
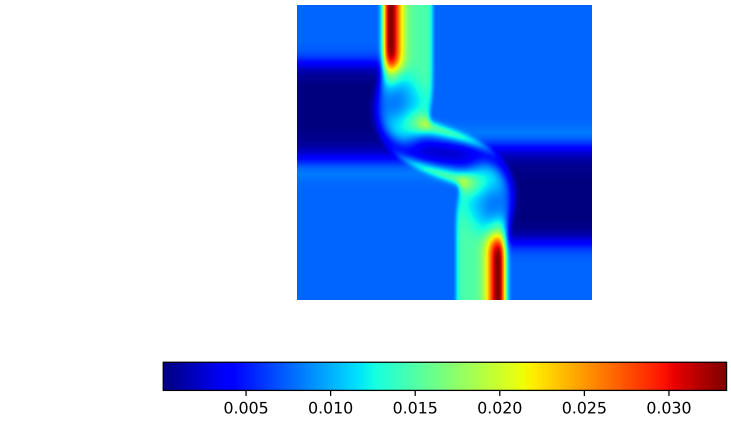
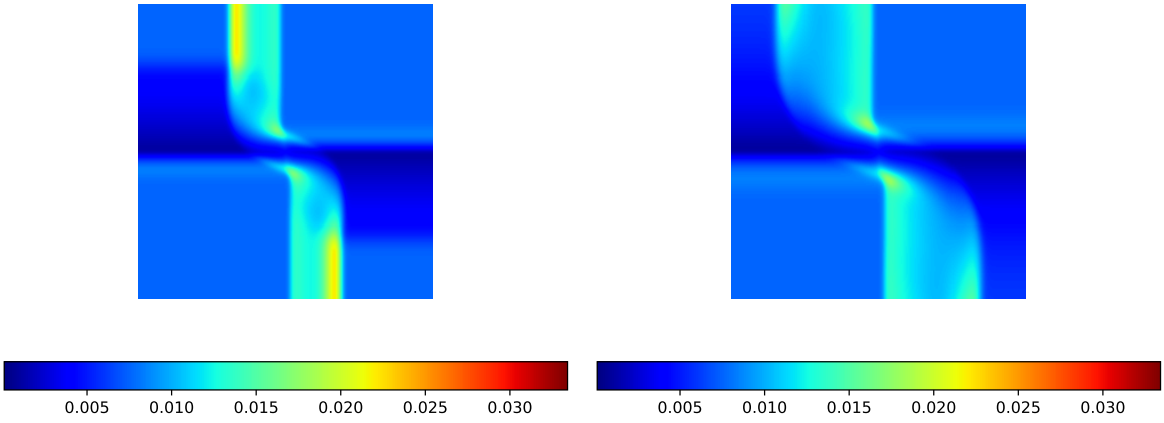


Figure 4.14 – Evolution of the residual as a function of the number of V-cycle, with different values for $L_{\max}$ for the two-dimensional Riemann problem.



(a) Explicit



(b) Jacobi, $\frac{\left|\left|\boldsymbol{r}^h_{(\kappa)}\right|\right|}{\left|\left|\boldsymbol{r}^h_{(0)}\right|\right|} = 2.9 \times 10^{-1}$

(c) Jacobi, $\frac{\left|\left|\boldsymbol{r}^h_{(\kappa)}\right|\right|}{\left|\left|\boldsymbol{r}^h_{(0)}\right|\right|} = 9.1 \times 10^{-2}$

Figure 4.15 – Snapshots of radiative energy at final time $t_f = 10^{-11}$ s with the explicit solver and the Jacobi method with different residuals.

therefore the total computational time also increases, unlike the beam test. Because we are in the free-streaming regime, the propagation of the photons has to be followed. In sections 2.2.3 and 2.2.4, we did it by reducing the time step. Here, we still use a large time step, but the Jacobi method needs lots of iterations to converge.

Figure 4.15 shows snapshots of radiative energy at the final time 10^{-11} s with the explicit solver (fig. 4.15a) and Jacobi method with different residuals (figs. 4.15b and 4.15c). Using the Jacobi method, we only need one time step to reach the same final time as the explicit solver with more than 150 time steps. Because we use a larger time step, the solution obtained with the Jacobi method is more diffusive than the one obtained with the explicit solver. However, in this test, numerical diffusion increases as the residual decreases.

As discussed in section 3.2.1.1, the reduced flux stays close to 1 during the simulation. Overall, the methods presented in this chapter are stable, even for very stiff problems. But it should be used carefully in physical problems that require following the propagation of photons.

4.4 Discussion and conclusion

In this chapter, we have first presented a Jacobi method to solve the nonlinear system arising from the discretization of the M_1 model without source terms with a time-implicit HLL solver. The admissible states are preserved, even with large CFL numbers. This method is iterative, the convergence rate decreases when the resolution increases. To tackle this issue, we use a nonlinear geometric multigrid algorithm. The Jacobi method first described is used as smoother and coarse grid solver. However, this algorithm relies on the residual equation, which does not preserve the admissible states. Instead of solving this equation, we introduce a pseudo-time, and we look for a steady state in pseudo-time. Numerical experiments have shown the good performances obtained with this algorithm.

4.4.1 Source terms

In order to study astrophysical problems, the integration of source terms is indispensable. It needs to be done carefully, regarding the asymptotic preserving property already discussed in chapters 2 and 3. The schemes developed in chapter 3 are asymptotic preserving and preserve the admissible states. The HLL solver used here could be replaced by the all-regime schemes presented in chapter 3. This will change the nonlinear operator $\mathcal{A}$. It requires to choose carefully which terms are taken at iteration k and $k+1$ in algorithm 1. The geometric multigrid algorithm presented in section 4.2 does not rely on the form of the nonlinear operator, as long as the iterations of the Jacobi method preserve the admissible states. Therefore, the geometric multigrid method can be used to study astrophysical problems with opacity as soon as the Jacobi algorithm is extended to the all-regime scheme.

4.4.2 Performances

Although performance results shown in section 4.3 are promising, some choices were made and others were not explored. For example, only a V-cycle is used, but there exist other possibilities: W-cycle, F-cycle,... The restriction and prolongation operators can also have an impact on the performances.

Another well-known method to reduce the number of iterations performed by the Jacobi method is to replace it with a Gauss-Seidel algorithm. Even though the tests presented in section 4.3 are obtained with a sequential code, we aim at using a parallel implementation to study astrophysical situations. In the linear case, the Jacobi method is known to be easier to make parallel than the Gauss-Seidel method.

4.5 Bibliography

Christophe Buet and Bruno Despres. Asymptotic preserving and positive schemes for radiation hydrodynamics. *Journal of Computational Physics*, 215(2):717 – 740, 2006. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2005.11.011. URL http://www.sciencedirect.com/science/article/pii/S0021999105005127. 94

Teddy Pichard. *Mathematical modelling for dose depositon in photontherapy*. PhD thesis, 2016. URL http://www.theses.fr/2016BORD0177. Thèse de doctorat dirigée par Dubroca, Bruno et Frank, Martin Mathematiques appliquees et calcul scientifique Bordeaux 2016. 94, 95

Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003. ISBN 0898715342. 95, 96, 97

William Briggs, Van Henson, and Steve McCormick. *A Multigrid Tutorial, 2nd Edition*. 01 2000. ISBN 978-0-89871-462-3. 97, 105

Achi Brandt and Oren Livne. Multigrid techniques. 1984 guide with applications to fluid dynamics. revised ed. 67, 01 2011. doi: 10.1137/1.9781611970753. 97, 103

Gilbert Strang. 18.086 mathematical methods for engineers ii. *Massachusetts Institute of Technology: MIT OpenCourseWare*, Spring 2006. URL https://ocw.mit.edu. Creative Commons BY-NC-SA. 99

K. Kifonidis and E. Müller. On multigrid solution of the implicit equations of hydrodynamics - experiments for the compressible euler equations in general coordinates. *A&A*, 544:A47, 2012. doi: 10.1051/0004-6361/201116979. URL https://doi.org/10.1051/0004-6361/201116979. 106

M. González, E. Audit, and P. Huynh. Heracles: a three-dimensional radiation hydrodynamics code. *A&A*, 464(2):429–435, 2007. doi: 10.1051/0004-6361:20065486. URL https://doi.org/10.1051/0004-6361:20065486. 108

S. Richling, E. Meinköhn, N. Kryzhevoi, and G. Kanschat. Radiative transfer with finite elements - i. basic method and tests. *A&A*, 380(2):776–788, 2001. doi: 10.1051/0004-6361:20011411. URL https://doi.org/10.1051/0004-6361:20011411. 108

F Blachère and R Turpault. An admissibility and asymptotic-preserving scheme for systems of conservation laws with source term on 2d unstructured meshes. *Journal of computational physics*, 315:98–123, 2016. ISSN 0021-9991. 111

4.A Overall algorithm

Algorithm 7 presents the whole algorithm to compute $\boldsymbol{v}^{n+1}$ with $\boldsymbol{v}^n$ known. When we presented the V-cycle in section 4.2.2, we used a recursive form. We present here a version based on loops.

Algorithm 7 Overall algorithm

Choose a maximum number of levels $L_{\max}$
$\kappa = 0$
$\boldsymbol{f}^h = \boldsymbol{v}^n$
$\boldsymbol{v}^h_{(0)} = \boldsymbol{v}^n$. From now on, the exponent is the mesh size, it is no longer the time
while $\left|\left|\boldsymbol{r}_{(\kappa)}\right|\right| > \varepsilon$ **do**
  for $l = 0, \cdots, L_{\max} - 1$ **do**
    if $l = 0$ **then**
      Pre-smoother: solve $\mathcal{A}^h\left(\boldsymbol{u}^h_{(\kappa)}\right) = \boldsymbol{f}^h$ with ν_0 iterations of the nonlinear Jacobi method (algorithm 1), with initial guess $\boldsymbol{v}^h_{(\kappa)}$
      Compute residual: $\overline{\boldsymbol{r}}^h_{(\kappa)} = \boldsymbol{f}^h - \mathcal{A}^h\left(\boldsymbol{u}^h_{(\kappa)}\right)$
    else

Pre-smoother: compute steady state in pseudo-time of $\frac{\mathrm{d}\boldsymbol{u}_{(\kappa)}^{2^l h}}{\mathrm{d}\tau}+\mathcal{A}^{2^l h}\left(\boldsymbol{u}_{(\kappa)}^{2^l h}\right)=\mathcal{A}^{2^l h}\left(\boldsymbol{v}_{(\kappa)}^{2^l h}\right)+\boldsymbol{r}_{(\kappa)}^{2^l h}$ using algorithm 6, with ν_l iterations of the nonlinear Jacobi method , with initial guess $\boldsymbol{v}_{(\kappa)}^{2^l h}$

Compute residual: $\overline{\boldsymbol{r}}_{(\kappa)}^{2^l h}=\mathcal{A}^{2^l h}\left(\boldsymbol{v}_{(\kappa)}^{2^l h}\right)+\boldsymbol{r}_{(\kappa)}^{2^l h}-\mathcal{A}^{2^l h}\left(\boldsymbol{u}_{(\kappa)}^{2^l h}\right)$

end if

Restriction: $\boldsymbol{v}_{(\kappa)}^{2^{l+1} h}=\mathcal{R}_{2^l h}^{2^{l+1} h}\left(\boldsymbol{v}_{(\kappa)}^{2^l h}\right)$ and $\boldsymbol{r}_{(\kappa)}^{2^{l+1} h}=\mathcal{R}_{2^l h}^{2^{l+1} h}\left(\overline{\boldsymbol{r}}_{(\kappa)}^{2^l h}\right)$

end for

Solver at coarsest level: compute steady state in pseudo-time of $\frac{\mathrm{d}\boldsymbol{u}_{(\kappa)}^{2^{L_{\max}} h}}{\mathrm{d}\tau}+\mathcal{A}^{2^{L_{\max}} h}\left(\boldsymbol{u}_{(\kappa)}^{2^{L_{\max}} h}\right)=\mathcal{A}^{2^{L_{\max}} h}\left(\boldsymbol{v}_{(\kappa)}^{2^{L_{\max}} h}\right)+\boldsymbol{r}_{(\kappa)}^{2^{L_{\max}} h}$ using algorithm 6, with $\nu_{L_{\max}}$ iterations of the nonlinear Jacobi method , with initial guess $\boldsymbol{v}_{(\kappa)}^{2^{L_{\max}} h}$

for $l=L_{\max}-2,\cdots,0$ by step -1 **do**

Prolongation: $\overline{\boldsymbol{v}}_{(\kappa)}^{2^l h}=\boldsymbol{v}_{(\kappa)}^{2^l h}+\mathcal{P}_{2^{l+1} h}^{2^l h}\left(\boldsymbol{u}_{(\kappa)}^{2^{l+1} h}-\boldsymbol{v}_{(\kappa)}^{2^{l+1} h}\right)$

for each cell l^i in the grid at level l **do**

if $\left(\overline{\boldsymbol{v}}_{(\kappa)}^{2^l h}\right)_{l^i}$ is not admissible **then**

use eq. 4.32 to update $\left(\overline{\boldsymbol{v}}_{(\kappa)}^{2^l h}\right)_{l^i}$

end if

end for

if $l=0$ **then**

Post-smoother: solve $\mathcal{A}^h\left(\boldsymbol{u}_{(\kappa)}^h\right)=\boldsymbol{f}^h$ with ν_0 iterations of the nonlinear Jacobi method (algorithm 1), with initial guess $\overline{\boldsymbol{v}}_{(\kappa)}^h$

else

Post-smoother: compute steady state in pseudo-time of $\frac{\mathrm{d}\boldsymbol{u}_{(\kappa)}^{2^l h}}{\mathrm{d}\tau}+\mathcal{A}^{2^l h}\left(\boldsymbol{u}_{(\kappa)}^{2^l h}\right)=\mathcal{A}^{2^l h}\left(\boldsymbol{v}_{(\kappa)}^{2^l h}\right)+\boldsymbol{r}_{(\kappa)}^{2^l h}$ using algorithm 6, with ν_l iterations of the nonlinear Jacobi method , with initial guess $\overline{\boldsymbol{v}}_{(\kappa)}^{2^l h}$

end if

end for

$\boldsymbol{v}_{(\kappa)}^h=\boldsymbol{u}_{(\kappa)}^h$

$\boldsymbol{r}_{(\kappa)}=\boldsymbol{v}^n-\mathcal{A}^h\left(\boldsymbol{v}_{(\kappa)}^h\right)$

if $\frac{|||\boldsymbol{r}_{(\kappa)}||-||\boldsymbol{r}_{(\kappa-1)}|||}{||\boldsymbol{r}_{(0)}||}>\varepsilon_i$ **then**

$\Delta\tau^{\mathrm{im}}\leftarrow 1.1\Delta\tau^{\mathrm{im}}$

end if

if $\frac{|||\boldsymbol{r}_{(\kappa)}||-||\boldsymbol{r}_{(\kappa-1)}|||}{||\boldsymbol{r}_{(0)}||}<\varepsilon_d$ **then**

$\Delta\tau^{\mathrm{im}}\leftarrow\Delta\tau^{\mathrm{im}}/2$

end if

$\kappa\leftarrow\kappa+1$

end while

$\boldsymbol{v}^{n+1}=\boldsymbol{v}_{(\kappa)}^h$

Chapter 5

High performance computing and linear algebra

Contents

Sections 5.1 and 5.2 are the adaptation of an article published in Astronomy&Astrophysics, see Bloch et al. 2021.

In the previous chapters, we presented numerical schemes to solve the M_1 model of radiative transfer. However, studying three-dimensional astrophysical problems requires the use of High Performance Computing (HPC) tools, to reach large resolutions.

In this chapter, we focus on the scheme developed in chapter 2. Its implementation has been done in the code ARK-RT [1], a fork of the code ARK developed in Padioleau et al. [2019]. The hydrodynamics and gravity part of the solver is similar to ARK and is solved explicitly with a well-balanced and all-regime solver. For the radiation part, the scheme is implicit and a Newton-Raphson method is used to solve the nonlinear system arising from the discretization of the M_1 model. At each iteration, a large sparse ill-conditioned linear system has to be solved. In order to do this, we use the library Trilinos [Trilinos Project Team], especially its second generation of packages.

We first present this library in section 5.1. In section 5.2, we provide details of some implementation features of ARK-RT. We also show some performance results and we discuss the impact of preconditioners and domain decomposition on a physical problem. Finally, we reach our conclusion in section 5.3.

5.1 Trilinos

The packages we used are Kokkos (section 5.1.1) for shared memory computation, Tpetra (section 5.1.2) for distributed vectors and matrices, Belos (section 5.1.3) for linear solvers, Ifpack2 (section 5.1.4) for classical preconditioners, and MueLu (section 5.1.5) for the algebraic multigrid (AMG) preconditioner. We now detail each of them. We only present the way we used Trilinos, it can be used for many others applications. This work does not intend to be a full review of the capabilities of Trilinos.

5.1.1 Kokkos

As new architectures have more and more cores, the distributed memory model is not enough to take advantage of all the computational power available. Therefore, we need to use a shared memory model inside the nodes. Furthermore, computational nodes are more and more heterogeneous, for example, multi-cores, many-cores, or accelerators such as GP-GPUs. Each architecture requires its own interface, such as OpenMP or C++11 threads for multi-cores and many-cores processors and CUDA or OpenACC for NVIDIA GPUs. This raises the problem of portability and performance portability: many HPC codes are optimized for some specific architectures, so running the code on a different architecture will result in bad performance.

The package Kokkos [Edwards et al., 2014] tackles this issue. The user has a unique code that can be compiled with different shared memory models such as OpenMP or CUDA. We call these programming models backends.

As CUDA programming model, Kokkos relies on a host/device abstraction. The host dispatches the parallel work to the device. The device can be a GPU or a multi-core CPU, for example.

1. https://gitlab.erc-atmo.eu/erc-atmo/ark-rt/tree/v1.0.0

The memory space abstraction defines where data are stored. It can be on the host (**Kokkos**::HostSpace) or on the device (e.g., **Kokkos**::CudaSpace for a GPU). Kokkos provides multidimensional arrays, whose storage is optimized according to the architecture. These arrays are called **Kokkos**::View.

The execution space specifies where a function is executed. The different possibilities are

— on the host with a sequential execution (**Kokkos**::Serial),
— on the host using POSIX threads (**Kokkos**::Threads),
— on the host using OpenMP (**Kokkos**::OpenMP),
— on the device, if a GPU is used (**Kokkos**::Cuda).

Finally, the execution policy dictates how the function is executed. Kokkos provides three of them. **Kokkos**::**parallel_for** is a loop pattern. Each iteration executes the body of the function. Kokkos gives access to an index, but there is no guarantee about the order of the iterations. The second pattern is **Kokkos**::**parallel_reduce** that implements a reduction. Each iteration produces a result, and they are accumulated into a single value. The last pattern is **Kokkos**::**parallel_scan**, which is a cumulative reduction. For example, if the reduction operation is a sum, all partial sums are stored.

```
1  int main(int argc, char* argv[])
2  {
3    // Initialize Kokkos
4    Kokkos::initialize(argc, argv);
5    // Declare a view with 3*15 elements
6    Kokkos::View <double*[3]> view("view_name", 15);
7    // Initialize the view through a parallel loop
8    Kokkos::parallel_for("loop_name", 15,
9        [=] (const int i) {
10       view(i, 0) = i;
11       view(i, 1) = i*i;
12       view(i, 2) = i*i*i;
13       });
14   // Shut down Kokkos
15   Kokkos::finalize();
16 }
```

Listing 5.1 – Example of code using Kokkos: declaration and initialization of a Kokkos::View

Listing 5.1 is an example of code showing how to declare a **Kokkos**::View and how to initialize it with a parallel loop.

5.1.2 Tpetra

The package Tpetra [Tpetra Project Team] handles distributed linear algebra objects, such as sparse matrices and vectors. There are two levels of parallelism: the Message Passing Interface (MPI) for distributed memory parallelism and a shared memory programming model, handled through Kokkos (see section 5.1.1).

Tpetra relies on another package, Teuchos [Teuchos Project Team], that provides some tools common to all packages of Trilinos. The only class we mention here is **Teuchos**::**RCP**<T>. This class provides smart pointers, with an automatic garbage collector. The memory is managed automatically, the user does not deallocate it manually.

Before presenting the different objects we used, let us briefly present the template parameters used by Tpetra.

5.1.2.1 Template parameters

Trilinos is a general-purpose framework, the user has to specify several data types. Therefore, most of the objects use template parameters, such as local_ordinal_type, global_ordinal_type, scalar_type, and node_type.

The scalar_type is the type of values in the matrix or vectors. Common cases are **float** or **double**. The choice of **float** or **double** has an impact on the performances and the precision of the code.

Because Tpetra handles distributed objects, the rows of the matrix or vector are distributed across the MPI processes. Each of them is associated with a unique global index, whose type is global_ordinal_type. This type indicates the maximum size of the object, for example, the maximal number of rows of the matrix. If we use **int** as global_ordinal_type, the maximum number of rows in the matrix would $2^{31}-1$, about two billion. In the three-dimensional case, there is five variables per cell (two scalars, the radiative energy and the gas temperature, and three components for the radiative flux). We are then restricted to 754^3 cells, so the total number of unknowns is less than $2^{31}-1$. Therefore, Trilinos suggest using more than 64 bits integers. The default type is **long long**.

Inside a MPI process, the row of the matrix or vector is also associated with a local index. Its type is local_ordinal_type and is usually **int**.

The last type shared by Tpetra's objects is the node_type. It specifies the shared memory programming model, such as OpenMP or CUDA.

Most of the classes defined by Tpetra use template parameters. Let us consider a matrix class, with the four template parameters presented previously. A class with template parameters does not generate any code until all the parameters are specified. Therefore, part of the code of Trilinos is compiled only when the application code is compiled. Each file of the application code which uses a matrix has to include the header file where the matrix class is declared. So, the matrix class will be compiled several times, even if the same template parameters are used. This leads to a large compilation time when the application code is built against Trilinos.

To tackle this issue, explicit template instantiation (ETI) is used. When Trilinos is built, the template parameters are explicitly specified, and the classes are compiled. This increases the compilation time of Trilinos, but significantly reduces the compilation time of the application code.

```
1 using scalar_type = double;
2 using global_ordinal_type = long long;
3 using local_ordinal_type = int;
4 using node_type = Kokkos::Compat::KokkosCudaWrapperNode;
```

Listing 5.2 – Definition of template parameters used by Tpetra.

Listing 5.2 shows a way to simplify future code by defining template parameters used by Tpetra. Let us notice that these types can also be defined using default values when Trilinos is compiled with ETI (e.g. **using** scalar_type = **Tpetra**::MultiVector <>::scalar_type;).

5.1.2.2 Map

Using Tpetra vocabulary, a map describes how data are distributed across the MPI processes. Let us notice that, despite the name, it is not the same concept as C++ Standard Template Library (STL) **std**::map, although there are some common ideas.

Let us consider a matrix with N rows, distributed across p processes. For the sake of simplicity, let us assume that N is a multiple of p. Using Tpetra's default behavior, each process handles $\frac{N}{p}$ contiguous rows. This case is optimized by Tpetra, it should provide the best performances.

```
using map_type = Tpetra::Map<local_ordinal_type, global_ordinal_type,
    node_type>;
// Create a communicator, such as MPI_COMM_WORLD
Teuchos::RCP<const Teuchos::Comm<int>> comm = Tpetra::getDefaultComm();
// Local number of elements
const local_ordinal_type numLocalElements = ...;
// Total number of elements
// Local number of elements * number of MPI processes
const global_ordinal_type numGlobalElements = numLocalElements *
    comm->getSize();
// Global indices start at 0
const global_ordinal_type indexBase = 0;
// Create the map
Teuchos::RCP<const map_type> map = Teuchos::rcp (new map_type
    (numGlobalElements, indexBase, comm));
```

Listing 5.3 – Demonstration of the construction of a Tpetra::Map

From Tpetra documentation the constructor of a map must be called collectively, i.e., all processes have to call it. Listing 5.3 shows how to build a map.

5.1.2.3 Multivector

A multivector is a set of distributed vectors. Their distribution across the MPI processes is the same, i.e., they all have the same map. A multivector can contain only one vector. Having a set of vectors can be useful, for example, to solve a linear system with several right-hand sides, but the same matrix. Local to each MPI process, the multivector can be seen as a view, in the sense given by Kokkos (section 5.1.1).

Let us briefly recall the algorithm to solve radiation hydrodynamics (section 2.1.2):

1. update of the hydrodynamics quantities;
2. update of the radiative quantities and gas temperature using the implicit solver developed in section 2.1.1;
3. addition of source terms.

The first and the last steps do not require the use of whole library Trilinos, only Kokkos and MPI are used. The code ARK implements only explicit solvers, it is not coupled with Trilinos. Views used to store the hydrodynamics quantities and views coming from a multivector have to interact. For performance reasons, it should be done through a **Kokkos::parallel_for**, to avoid data transfers between the device and the host. Listing 5.4 shows how it can be done.

```
using vec_type = Tpetra::MultiVector<scalar_type, local_ordinal_type,
    global_ordinal_type, node_type>;
// View containing hydrodynamics quantities
Kokkos::View<double*, Kokkos::DefaultExecutionSpace> hydroData = ...;
// Create multivectors containing one vector
Teuchos::RCP<vec_type> X = Teuchos::rcp(new vec_type (map, 1));
Teuchos::RCP<vec_type> B = Teuchos::rcp(new vec_type (map, 1));
// Set initial guess to 0
X->putScalar(0.);
// Initialize RHS
```

```
{
  // Get the view in write mode
  auto dataB = B->getLocalViewDevice(Tpetra::Access::OverwriteAll);
  // Write data in the view
  Kokkos::parallel_for("Initialize B", numLocalElements,
      [=] (const int i){
      dataB(i, 0) = someFunction(hydroData(i));
      });
}
// Do something to solve AX = B
// Update hydrodynamics quantities with the solution of the linear system
{
  // Get the view in read-only mode
  auto dataX = X->getLocalViewDevice(Tpetra::Access::ReadOnly);
  // Read data from the view
  Kokkos::parallel_for("Read X", numLocalElements,
      [=] (const int i){
      hydroData(i) = someOtherFunction(dataX(i, 0));
      });
}
```

Listing 5.4 – Accessing data of a multivector from the device

5.1.2.4 CrsMatrix

Using the scheme presented in chapter 2, there are only a few nonzero elements per row. Using a sparse matrix is essential to save both computational time and memory. The storage of the matrix is done with a compressed sparse row (CSR) or compressed row storage (CRS) format (see e.g., Buluç et al. 2009). The class **Tpetra**::CrsMatrix implements such storage. The rows of the matrix are distributed across the MPI processes. All the nonzero elements of a row are owned by the same process.

```
using graph_type =  Tpetra::CrsGraph<local_ordinal_type,
    global_ordinal_type, node_type>;
using matrix_type = Tpetra::CrsMatrix<scalar_type, local_ordinal_type,
    global_ordinal_type, node_type>;
// Graph with at most three nonzero entry per row
Teuchos::RCP<graph_type> G = Teuchos::rcp (new graph_type (map, 3));
// Tell the graph where are the nonzero entries
// The loop is sequential, but it is only done once
for(local_ordinal_type localIndex = 0; localIndex < numLocalElements;
    localIndex++)
{
  // Ask the map the global index corresponding to the local index
  const global_ordinal_type globalIndex =
   map->getGlobalElement(localIndex);
  // Use global index to build the graph
  if(globalIndex == 0)
    G->insertGlobalIndices(globalIndex, Teuchos::tuple(globalIndex,
    globalIndex+1));
  else if(globalIndex == numGlobalElements-1)
    G->insertGlobalIndices(globalIndex, Teuchos::tuple(globalIndex-1,
    globalIndex));
  else
    G->insertGlobalIndices(globalIndex, Teuchos::tuple(globalIndex-1,
    globalIndex, globalIndex+1));
}
// Tell the graph we are done changing its structure
```

```
20 G->fillComplete();
21 // Use the graph to build the matrix
22 Teuchos::RCP<matrix_type> A = Teuchos::rcp( new matrix_type (G));
23 // Tell the matrix we are done changing its structure
24 A->fillComplete();
```

Listing 5.5 – Demonstration of the construction of a tridiagonal matrix with a static graph

In our application, the values of the matrix have to be updated at each iteration of the Newton-Raphson method, but its graph (the localization of the nonzero elements in the matrix) does not change. To obtain the best performances, TRILINOS recommends building first the graph and then the matrix. The structure of the matrix is not allowed to change. The graph is distributed as the matrix, but values are not stored. Listings 5.5 to 5.7 show how to build the matrix obtained with the discretization of the Laplacian operator in the one-dimensional case. The matrix is

$$\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}. \tag{5.1}$$

In particular, listing 5.5 shows how to build this tridiagonal matrix with such a graph.

```
1  // Tell the matrix we are about to change its values
2  A->resumeFill();
3  // Update the coefficients
4  for(local_ordinal_type localIndex = 0; localIndex < numLocalElements;
5      localIndex++)
6  {
7    // Ask the map the global index corresponding to the local index
8    const global_ordinal_type globalIndex =
       map->getGlobalElement(localIndex);
9    // Use global index to access the elements of the matrix
10   if(globalIndex == 0)
11     A->replaceGlobalValues(globalIndex, Teuchos::tuple(globalIndex,
       globalIndex+1), Teuchos::tuple(2., -1.));
12   else if(globalIndex == numGlobalElements-1)
13     A->replaceGlobalValues(globalIndex, Teuchos::tuple(globalIndex-1,
       globalIndex), Teuchos::tuple(-1., 2.));
14   else
15     A->replaceGlobalValues(globalIndex, Teuchos::tuple(globalIndex-1,
       globalIndex, globalIndex+1), Teuchos::tuple(-1., 2., -1.));
16 }
17 // Tell the matrix we are done changing its values
18 A->fillComplete();
```

Listing 5.6 – Updating the coefficients of the matrix from the host (non-parallel update)

TPETRA provides several methods to update the coefficients of the matrix. One of them uses only global indices, which is the way recommended by TPETRA. Listing 5.6 shows how this can be done. However, this function can only be called from the host. This has two main consequences. First, because all the rows of the matrix have to be updated, we use a sequential loop over the rows of the matrix. Second, in the case where we are using GP-GPUs, we have to update the matrix with data coming from the device. Therefore, we have to transfer some data from the device to the host, update

the matrix with these data, and then transfer the matrix from the host to the device, this last step is done implicitly by TPETRA. This will increase the computational cost.

```
// Get the part of the matrix local to the MPI process
auto localMatrix = A->getLocalMatrix();
// Get the local map, which can be used from Kokkos kernel
auto localMap = map->getLocalMap();
// Use a parellel loop to update the coefficients of the matrix
Kokkos::parallel_for("Update matrix coefficients", numLocalElements,
    [=] (const int localIndex){
    // Ask the map the global index corresponding to the local index
    const global_ordinal_type globalIndex =
   localMap.getGlobalElement(localIndex);
    if(globalIndex == 0)
    {
      // Tell Tpetra which column(s) to update
      local_ordinal_type col[2];
      col[0] = localIndex;
      col[1] = localIndex+1;
      // Tell Tpetra which value(s) to set
      scalar_type val[2];
      val[0] = 2.;
      val[1] = -1.;
      // Update the matrix
      localMatrix.replaceValues(localIndex, col, 2, val);
    }
    else if(globalIndex == numGlobalElements-1)
    {
      // Tell Tpetra which column(s) to update
      local_ordinal_type col[2];
      col[0] = localIndex-1;
      col[1] = localIndex;
      // Tell Tpetra which value(s) to set
      scalar_type val[2];
      val[0] = -1.;
      val[1] = 2.;
      // Update the matrix
      localMatrix.replaceValues(localIndex, col, 2, val);
    }
    else
    {
      // Tell Tpetra which column(s) to update
      local_ordinal_type col[3];
      col[0] = localIndex-1;
      col[1] = localIndex;
      col[2] = localIndex+1;
      // Tell Tpetra which value(s) to set
      scalar_type val[3];
      val[0] = -1.;
      val[1] = 2.;
      val[2] = -1.;
      // Update the matrix
      localMatrix.replaceValues(localIndex, col, 3, val);
    }
    });
```

Listing 5.7 – Updating the coefficients of the matrix from the device (parallel update)

Another way is to use local indices. The package KOKKOS KERNEL, part of the KOKKOS ecosystem, provides several ways to update the coefficients of the matrix

through a kernel. This allows the use of a parallel loop (via **Kokkos**::**parallel_for**) and we avoid data transfers between the host and the device (see listing 5.7). For performance reasons, we use local indices.

5.1.3 Belos

As discussed in section 2.1.3.1, the matrix of the linear system to be solved using the scheme developed in chapter 2 is large, sparse, and not symmetric. We solve the linear system with a biconjugate gradient stabilized (BiCGSTAB) method [Van der Vorst, 1992]. The package Belos [Belos Project Team] provides an implementation of this algorithm. Belos gives also access to other linear solvers we did not explore. Listing 5.8 shows how to create the linear problem associated with AX=B and the associated solver manager. It also shows how to solve the problem.

```
using op_type = Tpetra::Operator<scalar_type, local_ordinal_type,
    global_ordinal_type, node_type>;
using problem_type = Belos::LinearProblem<scalar_type, vec_type,
    op_type>;
// Create, configure, and return the specified solver
Belos::SolverFactory<scalar_type, vec_type, op_type> factory;
Teuchos::RCP<Belos::SolverManager<scalar_type, vec_type, op_type>>
    solver = factory.create("BiCGSTAB");
// Linear system AX = B to be solved, and its associated informations
Teuchos::RCP<problem_type> problem =
Teuchos::rcp(new problem_type(A, X, B));
// If needed, do something with a preconditionner
// Prepare the problem to solve the linear system
problem->setProblem();
// Tell the solver what problem you want to solve
solver->setProblem(problem);
// Solve the linear system
// Belos::ReturnType can take the values Belos::Converged or
    Belos::Unconverged
Belos::ReturnType result = solver->solve();
// Get the number of iterations needed to solve the system
const int numIters = solver->getNumIters();
// Get the achieved tolerance reached by calling solve()
const scalar_type achievedTol = solver->achievedTol();
// Do something to read the solution X, to change the values of the
    matrix A and the right-hand side B
// Tell the solver manager to prepare the solver for the next solve
solver->reset(Belos::Problem);
// Solve again the linear system
result = solver->solve();
```

Listing 5.8 – Demonstration of the construction of a Belos solver

5.1.4 Ifpack2

In theory, the BiCGSTAB algorithm converges with a finite number of iterations. However, the convergence can be slow for physical applications. A similar phenomenon has already been observed in section 4.2.1. Instead of using a geometric multigrid method as in section 4.2, we use a preconditioning technique (see e.g., Saad 2003). The linear system to be solved is modified into another system, with the same solution, but easier to solve. Instead of solving $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$, we solve the right preconditioned system

$\mathbb{A}\mathbb{K}^{-1}\mathbb{K}\boldsymbol{x} = \boldsymbol{b}$ via solving $\mathbb{A}\mathbb{K}^{-1}\boldsymbol{y} = \boldsymbol{b}$ to compute $\boldsymbol{y}$ and then $\mathbb{K}\boldsymbol{x} = \boldsymbol{y}$. As long as the matrix $\mathbb{K}$ is invertible, this gives the same solution as the original system. If $\mathbb{K}$ is well-chosen, the condition number of the matrix $\mathbb{A}\mathbb{K}^{-1}$ is lower than $\mathbb{A}$'s one.

The package IFPACK2 [Ifpack2 Project Team] provides classical preconditioners. One of them, the Jacobi or diagonal preconditioner is given by

$$\mathbb{K}_{ij} = \begin{cases} \mathbb{A}_{ij} \text{ if } i = j \\ 0 \text{ otherwise.} \end{cases} \tag{5.2}$$

IFPACK2 also provides a relaxed incomplete LU factorization with level k fill (**Ifpack2**::RILUK). Instead of solving $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$, one can write $\mathbb{A} = \mathbb{L}\mathbb{U}$, where $\mathbb{L}$ is a lower triangular matrix and $\mathbb{U}$ is an upper triangular matrix and then solve $\mathbb{L}\boldsymbol{y} = \boldsymbol{b}$ followed by $\mathbb{U}\boldsymbol{x} = \boldsymbol{y}$. However, even if $\mathbb{A}$ is sparse, $\mathbb{L}$ and $\mathbb{U}$ are not sparse in general. This leads to a phenomenon called fill-in. Memory consumption can become a bottleneck. Therefore, other matrices $\mathbb{L}$ and $\mathbb{U}$ can be used, such that $\mathbb{A} \approx \mathbb{L}\mathbb{U}$. The product $\mathbb{L}\mathbb{U}$ is used as a preconditioner. For example, the nonzero elements of the new matrices $\mathbb{L}$ and $\mathbb{U}$ can be the same as $\mathbb{A}$, leading to the method ILU(0). A more accurate method, but with a higher memory footprint is ILU(k), where the nonzero elements of $\mathbb{L}$ and $\mathbb{U}$ are the same as $\mathbb{A}^{k+1}$. All tests in section 5.2 are performed with the ILU(0) method.

IFPACK2 implements another incomplete factorization (**Ifpack2**::ILUT) where the maximum number of entries to keep in each row of $\mathbb{L}$ and $\mathbb{U}$ is set according to the number of nonzero elements of $\mathbb{A}$ in the same row. Entries smaller than a threshold are dropped. See Saad [1994] for more details.

Finally, IFPACK2 gives access to an addition Schwarz domain decomposition method (**Ifpack2**::AdditiveSchwarz). The problem is divided into smaller domains and the local results are added to each other. Each domain corresponds exactly to one MPI process, to avoid extra communications. More details can be found, for example in Prokopenko et al. [2016].

```
1  using ifpack2_prec_type = Ifpack2::Preconditioner<scalar_type,
       local_ordinal_type, global_ordinal_type, node_type>;
2  // Create an instance of the preconditioner, here it is a diagonal
       preconditioner
3  Ifpack2::Factory factory;
4  Teuchos::RCP<ifpack2_prec_type> K =
       factory.create<matrix_type>("DIAGONAL", A);
5  // Set up the graph structure of the preconditioner
6  K->initialize();
7  // Set up the numerical values in the preconditioner
8  K->compute();
9  // Use K as right preconditioner for Belos linear problem
10 problem->setRightPrec(K);
11 // Solve the linear system
12 solver->solve();
13 // Do something to change the values of the matrix A and the right-hand
       side B
14 // Compute again numerical values in the preconditioner
15 K->compute();
16 // Tell the solver manager to prepare the solver for a next solve
17 solver->reset(Belos::Problem);
18 // Solve again the linear system
19 solver->solve();
```

Listing 5.9 – Demonstration of the construction of an IFPACK2 preconditioner

Listing 5.9 shows how to create a preconditioner using IFPACK2 and how it interacts with BELOS linear problem.

5.1.5 MueLu

The last package we present here is MUELU [Muelu Project Team]. We used it as algebraic multigrid (AMG) preconditioner. The AMG methods were first developed as linear solvers for symmetric positive definite matrices arising from the discretization of scalar elliptic PDEs. For such a matrix, classical iterative methods are efficient to compute the high frequencies of the solution, but lack efficiency to compute its low frequencies. However, the computation is easier on a coarser grid with fewer unknowns. We recover some ideas already developed in section 4.2. The idea of the multigrid solver is to build a coarser grid, then solve the problem on this coarse grid and finally interpolate the solution on the fine grid. We can then define a restriction operator $\mathbb{R}$ which transfers vectors from the fine grid to the coarse grid and an interpolation operator $\mathbb{P}$ used to return to the finer grid. $\mathbb{P}$ and $\mathbb{R}$ are non squared matrices. From Saad [2003], here are the main steps of the method:

1. pre-smoothing: a few iterations of a simple method such as Jacobi or an incomplete factorization are performed, to get the value $\tilde{\boldsymbol{x}}$;
2. the residual $\tilde{\boldsymbol{r}} = \boldsymbol{b} - \mathbb{A}\tilde{\boldsymbol{x}}$ is projected over the coarse grid with the restriction operator $\mathbb{R}$, to get the residual equation $\mathbb{R}\mathbb{A}\mathbb{P}\boldsymbol{y} = \mathbb{R}\tilde{\boldsymbol{r}}$;
3. this equation is solved, possibly with a direct solver;
4. the solution $\boldsymbol{y}$ is interpolated over the fine grid with the interpolation operator $\mathbb{P}$ and then $\bar{\boldsymbol{x}} = \tilde{\boldsymbol{x}} + \mathbb{P}\boldsymbol{y}$;
5. post-smoothing: a few iterations of a simple method are again performed to get the solution $\bar{\tilde{\boldsymbol{x}}}$.

The solution $\bar{\tilde{\boldsymbol{x}}}$ is used as a preconditioner result. If the coarse grid has too many unknowns to be solved directly, this process is applied recursively: the coarse grid becomes the fine grid and a coarser grid is built. Therefore, we have a hierarchy of grids. With a geometric multigrid solver, the restriction and interpolation operators are determined by the mesh (see section 4.2), whereas, with an algebraic multigrid solver, they are automatically generated, using data from the matrix. Algorithms provided by IFPACK2 (see section 5.1.4) are used as smoothers. Listing 5.10 shows how to create a preconditioner using MUELU and how it interacts with BELOS linear problem.

```
1  using muelu_prec_type = MueLu::TpetraOperator<scalar_type,
       local_ordinal_type, global_ordinal_type, node_type>;
2  // Create a MueLu preconditioner that can be used by Tpetra
3  Teuchos::RCP<muelu_prec_type> K =
       MueLu::CreateTpetraPreconditioner<scalar_type, local_ordinal_type,
       global_ordinal_type, node_type>(A);
4  // Use K as right preconditioner for Belos linear problem
5  problem->setRightPrec(K);
6  // Solve the linear system
7  solver->solve();
8  // Do something to change the values of the matrix A and the right-hand
       side B
9  // Update the preconditioner
10 MueLu::ReuseTpetraPreconditioner(A, *K);
11 // Tell the solver manager to prepare the solver for a next solve
12 solver->reset(Belos::Problem);
```

```
13 // Solve again the linear system
14 solver->solve();
```

Listing 5.10 – Demonstration of the construction of a MueLu preconditioner

In the next section, we present some performance results obtained with the different packages of Trilinos. Especially, we explore the impact of the different preconditioners.

5.2 Application to radiative transfer and performance results

5.2.1 Implementation

For the hydrodynamics step, Kokkos is used as an independent library for shared memory computation. Communications between the nodes are handled by the Message Passing Interface (MPI) programming model through a regular domain decomposition. Following Kestener [2017], inside each node, the domains are endowed with ghost cells used to implement physical boundary conditions, but also to contain values from neighbor domains. The code is organized with computational kernels, each kernel is a C++ functor. See Padioleau et al. [2019] for more details.

The second step is the time-implicit solver for radiative transfer. The values of the matrix and the right-hand side of the linear system have to be updated at each iteration of the Newton-Raphson method. See section 5.1.2 for the different ways to update the matrix and the right-hand side.

5.2.2 Performances

Thanks to Trilinos, we can use many preconditioners. Unfortunately, they do not behave the same way when the size of the system increases. All tests are performed on Poincare, our local cluster at Maison de la Simulation. Each node consists of two Sandy Bridge E5-2670 @ 2.6 GHz (2×8 cores, 32 Go RAM) processors. We use a hybrid configuration MPI/OpenMP, with one MPI process per socket to avoid NUMA effects.

We first performed a weak scaling test, where we consider a two-dimensional case with periodic boundary conditions and a hot source located at the center of each domain. Each MPI process is getting a piece of the whole domain of $1\,500^2$ cells, therefore the size of the system increases with the number of MPI tasks. The resolution is close to the one we are aiming for three-dimensional simulations. Figure 5.1 shows the mean number of iterations for the linear solver to converge as a function of the number of cells. For all preconditioners, the number of iterations remains constant, around 10 iterations for the AMG preconditioner, around 20 iterations for both incomplete factorizations and the additive Schwarz domain decomposition and around 250 iterations for the relaxation. Figure 5.2 shows the speed-up as a function of the number of MPI processes. The speed-up reaches a plateau of 80% to 90% of maximum performance, depending on the preconditioner.

Figures 5.3 and 5.4 shows the number of iterations and the speed-up as a function of the number of MPI processes for a strong scaling test. We now consider a Marshak wave propagation in the diffusive limit. The global resolution remains constant as the number of processes increases. It is set to $2\,048^2$ cells. Because the global resolution

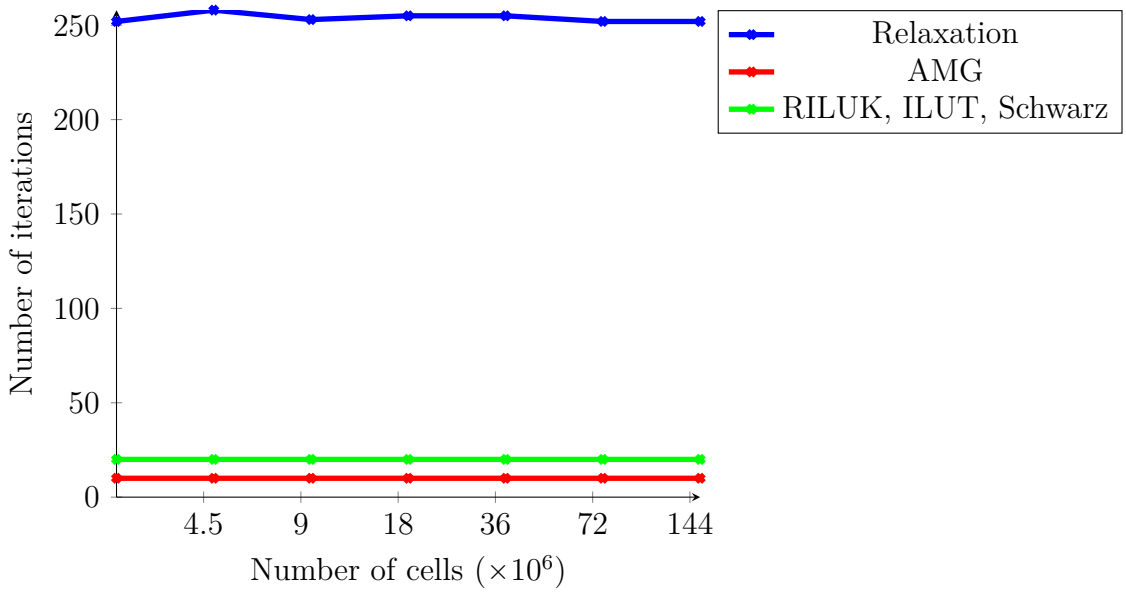


Figure 5.1 – Number of iterations in a weak scaling test. Each MPI process treats 1 500^2 cells. We have tested different preconditioners: Jacobi with damping (Relaxation), algebraic multigrid (AMG), standard ILU(k) factorization (RILUK), variant of the standard ILU factorization (ILUT) and additive Schwarz domain decomposition (Schwarz).

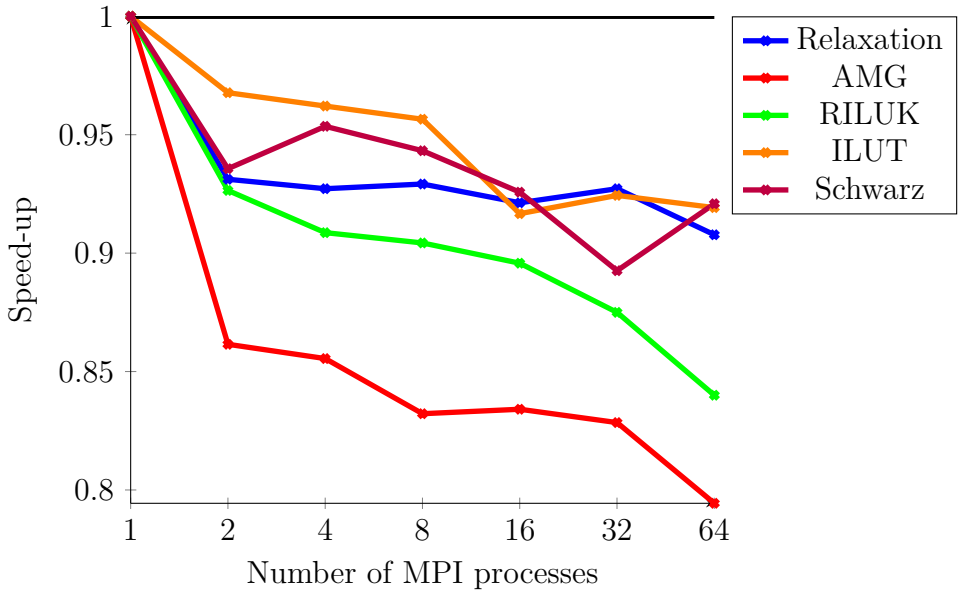


Figure 5.2 – Speed-up as a function of the number of MPI processes in a weak scaling test. Each MPI process treats 1 500^2 cells. We have tested different preconditioners: Jacobi with damping (Relaxation), algebraic multigrid (AMG), standard ILU(k) factorization (RILUK), variant of the standard ILU factorization (ILUT) and additive Schwarz domain decomposition (Schwarz).

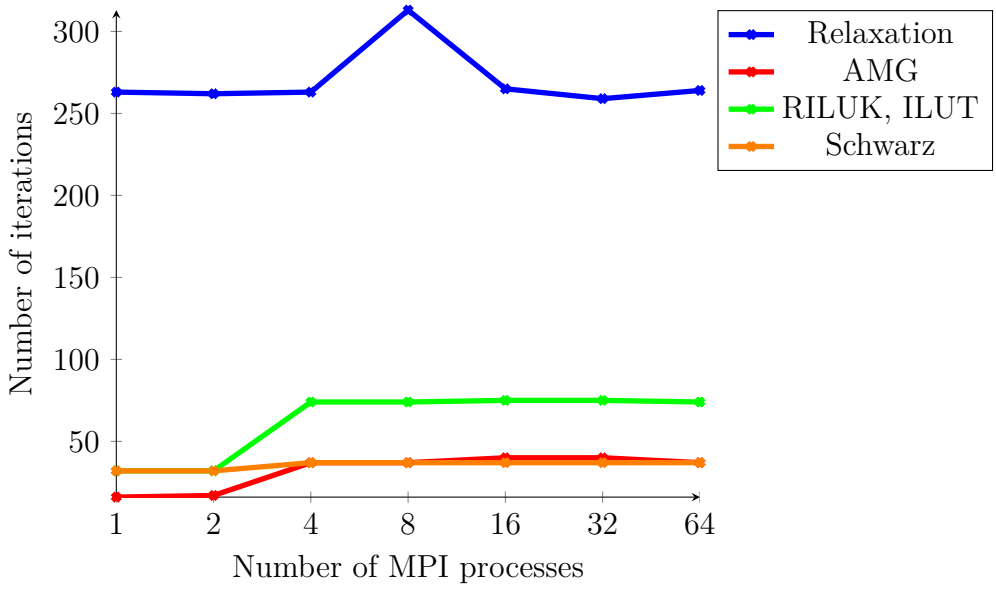


Figure 5.3 – Number of iterations in a strong scaling test. The global resolution is $2\,048^2$ cells. We have tested different preconditioners: Jacobi with damping (Relaxation), algebraic multigrid (AMG), standard ILU(k) factorization (RILUK), variant of the standard ILU factorization (ILUT) and additive Schwarz domain decomposition (Schwarz).

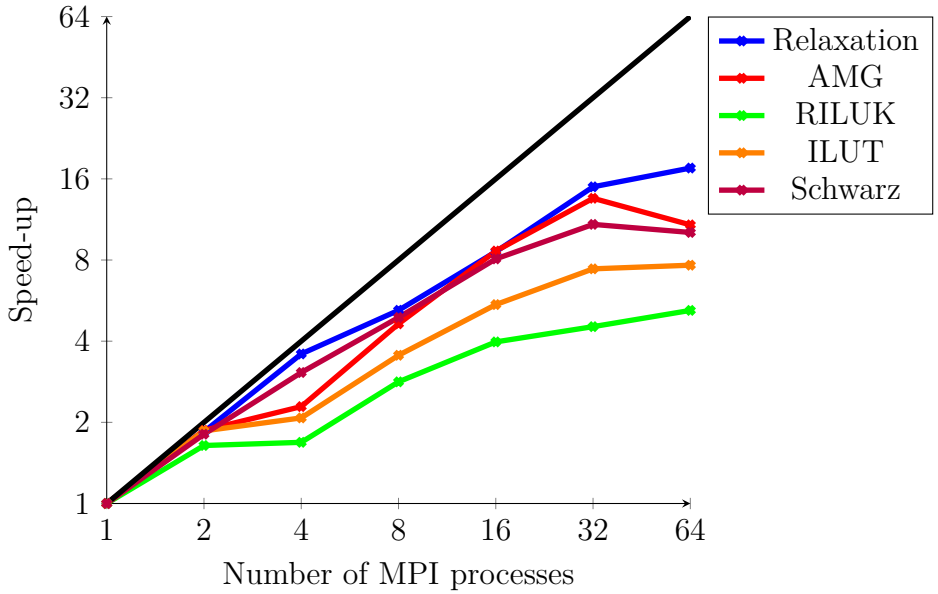


Figure 5.4 – Speed-up as a function of the number of MPI processes in a strong scaling test. The global resolution is $2\,048^2$ cells. We have tested different preconditioners: Jacobi with damping (Relaxation), algebraic multigrid (AMG), standard ILU(k) factorization (RILUK), variant of the standard ILU factorization (ILUT) and additive Schwarz domain decomposition (Schwarz).

Scheme	Number of time steps	Computational time OpenMP (s)	Computational time CUDA (s)
Explicit	73 823	6 991	839
Implicit (non-parallel update)	1	60	93
Implicit (parallel update)	1	44	77

Table 5.1 – Computational time with both explicit and implicit solvers on CPU and GPU. With the implicit solver, the matrix is updated in a parallel or a non-parallel way. The implicit solver uses the AMG preconditioner.

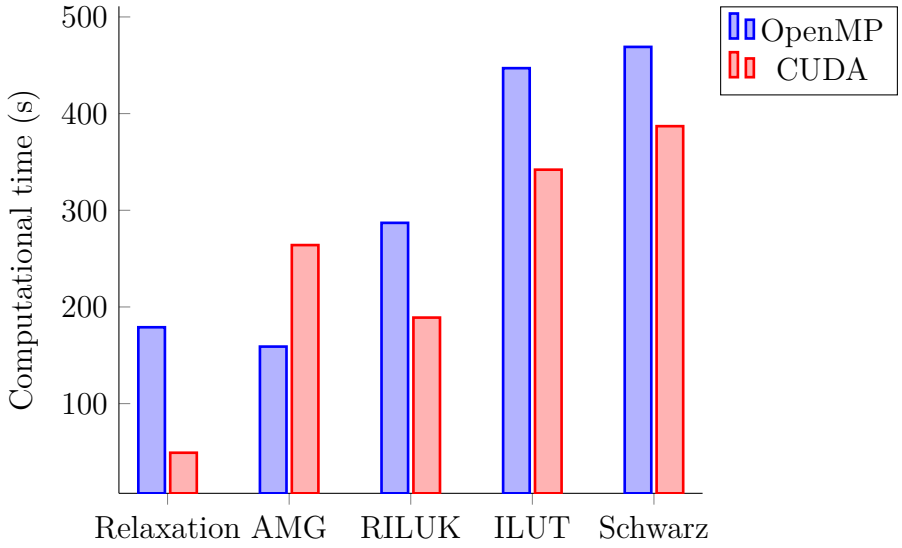


Figure 5.5 – Computational time for the implicit solver with different preconditioners (Jacobi with damping (Relaxation), algebraic multigrid (AMG), ILU(k) factorization (RILUK), slightly modified variant of ILU factorization (ILUT) and additive Schwarz domain decomposition (Schwarz)) on different architectures (Sandy Bridge CPU and K80 NVIDIA GPU). The resolution is 1000^2 cells.

is constant, one can expect the number of iterations to also remain constant when the number of MPI processes increases. However, using the algebraic multigrid (orange curve) and the incomplete factorizations (green curve), when four MPI processes or more are used, the number of iterations is twice the number of iterations reached with one or two MPI processes. Therefore, the computational time is the same using two or four MPI processes. Furthermore, all tested preconditioners and the linear solver requires several communications per iteration, which likely become the main cost when the local resolution decreases.

Thanks to Kokkos, we can use exactly the same code on different architectures like Sandy Bridge processors and NVIDIA GP-GPUs (e.g., K80). Unfortunately, the memory required by the AMG preconditioner with a $1\ 500^2$ simulation is larger than the memory available on a K80 GPU. For the next tests, we use a lower resolution of $1\ 000^2$ cells. Table 5.1 summarizes the computational time for a fixed problem with different schemes and different architectures. As the explicit solver is restricted by a CFL condition, it needs several thousands of time steps whereas the implicit solver only needs a few time steps to reach the same final time. Updating the matrix in parallel allows for a 25% reduction in computational time required. On CPU, the implicit solver is around 160 times faster than the explicit solver, whereas, on GPU, it is only 11 times faster.

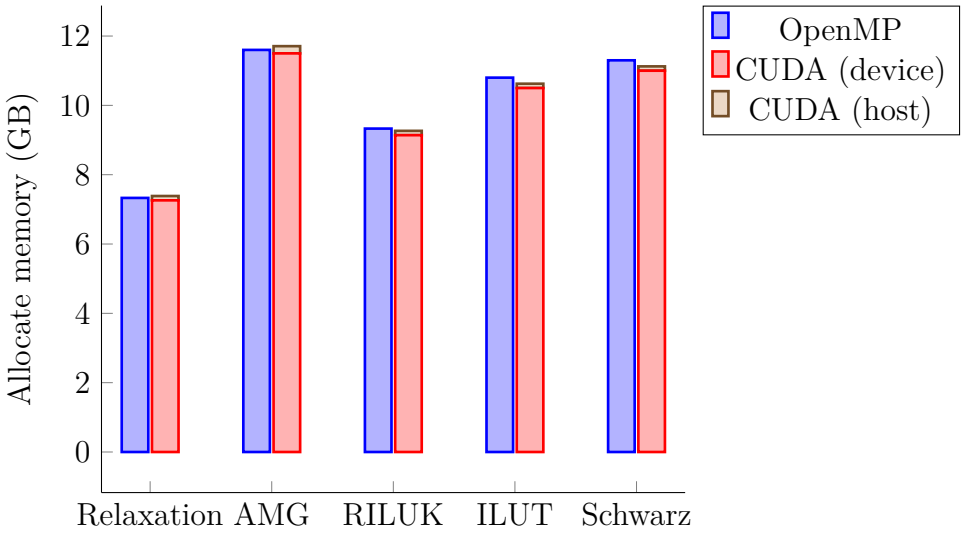


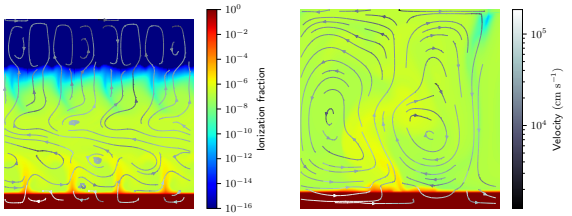
Figure 5.6 – Memory consumption for the implicit solver with different preconditioners (Jacobi with damping (Relaxation), algebraic multigrid (AMG), ILU(k) factorization (RILUK), slightly modified variant of ILU factorization (ILUT) and additive Schwarz domain decomposition (Schwarz)) on different architectures (Sandy Bridge CPU and K80 NVIDIA GPU). The resolution is 1000^2 cells.

Figure 5.5 compares the computational time with different preconditioners, on both CPU and GPU. Except for the implicit solver using the AMG preconditioner, all solvers are faster on GPU than CPU, up to three times faster for the relaxation preconditioner. Part of the AMG algorithm probably remains sequential. On CPU, the AMG preconditioner is faster than the relaxation preconditioner. The other preconditioners are slower, up to a factor 8 between the relaxation and the additive Schwarz domain decomposition on GPU.

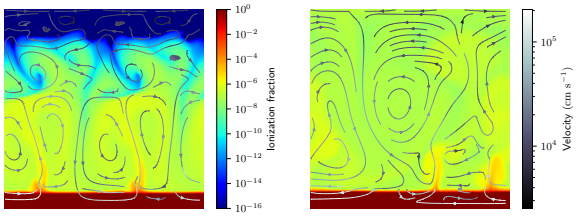
Figure 5.6 compares the memory consumption with different preconditioners. Using GP-GPU, most of the data are located on the device, but Trilinos still allocates some memory on the host, between 0.125 GB and 0.208 GB, unlike the explicit solver. Using the relaxation as a preconditioner, the amount of memory allocated is lower than with the other preconditioners (7.3 GB for the relaxation against 11.5 GB for the AMG).

Choosing a well-suited preconditioner can be challenging and problem dependent. Once the preconditioner is chosen, it depends on many parameters. For example, Trilinos allows the user to choose the damping factor ω for the relaxation method or the smoother and the coarse solver for the AMG. Performances and stability can largely depend on these choices. For example, the relaxation method seems to be well suited for this problem with low computational time, and memory consumption, but in many other test cases, the linear solver will not converge. The AMG preconditioner performs well on CPU but is less efficient on GPU. Both incomplete factorizations and the additive Schwarz domain decomposition are slightly less efficient than the AMG preconditioner. Overall we have found the AMG preconditioner or relaxation method are a good compromise between stability and performances.

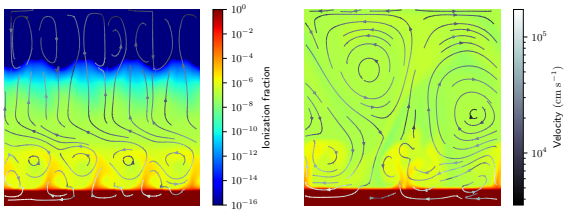
The performances we obtained thanks to Kokkos and Trilinos are encouraging for the study of astrophysical problems. The time step given by the hydrodynamics can be written as $CFL\frac{\Delta x}{c}$. Using a relaxation as a preconditioner, we need $CFL \geqslant 50$ on CPU and $CFL \geqslant 100$ on GPU to save computational time, whereas, using an incomplete factorization, we need $CFL \geqslant 250$ on CPU and $CFL \geqslant 1\,000$ on GPU. We need a larger CFL number on GPU because the explicit solver is more efficient on



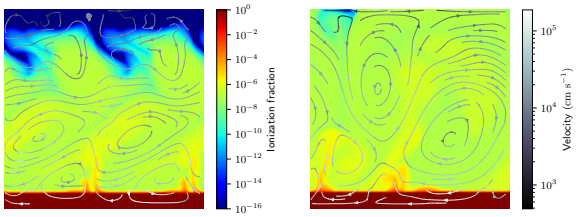
(a) 4 × 4 MPI processes, ILU(k) factorization.



(b) 2 × 2 MPI processes, ILU(k) factorization.



(c) 4 × 4 MPI processes, Schwarz domain decomposition.



(d) 2 × 2 MPI processes, Schwarz domain decomposition.

Figure 5.7 – Snapshots of the fraction of ionization and the velocity field at the final time $t_f = 10^{10}$ s without the initial velocity perturbation (left panel) and with it (right panel). The physical domain is distributed across different numbers of MPI processes and different preconditioners have been used. Figure 5.7a is the same figure as fig. 2.10.

GPU than CPU.

5.2.3 Effect of preconditioner and MPI domain decomposition on the expansion of H II region

In the test case described in section 2.2.6, some numerical noise appears, as a consequence of the long timescales. Let us recall that a time-implicit scheme is used, with large time steps for the radiative transfer. At each time step, the Newton-Raphson method is used and, at each iteration of this algorithm, an ill-conditioned linear system is solved, using an iterative process. This results in the appearance of some numerical noise.

We have performed the simulation described previously, with different numbers of MPI processes and different preconditioners. The physical domain is either distributed over 4×4 MPI processes (figs. 5.7a and 5.7c) or 2×2 MPI processes (figs. 5.7b and 5.7d). We have also tried two preconditioners which allowed us to reach the final time with reasonable computational time: a standard ILU(k) factorization (figs. 5.7a and 5.7b) and an additive Schwarz domain decomposition (figs. 5.7c and 5.7d).

Figure 5.7 shows snapshots of the fraction of ionization and the velocity field at the final time $t_f = 10^{10}$ s. The shape of the small structures produced by the numerical noise varies with the number of MPI processes and the preconditioner. Furthermore, the propagation of the ionization front creates some velocity that also depends on the number of MPI processes and preconditioners. However, the position of the ionization front is not affected by these parameters.

5.3 Discussion and conclusion

In this chapter, we explored some features of the library Trilinos for the resolution of linear systems. We also studied their performances on different architectures, such

as multi-cores and GPUs, thanks to the package Kokkos. Finally, we discussed the impact of the choice of the preconditioner and the MPI domain decomposition on a physical application.

5.3.1 Vectorization

We did not explore all the capabilities of Kokkos. In particular, Kokkos provides tools for vectorization through hierarchical parallelism. Some CPUs can perform operations on multiple floating points with the same number of cycles as the scalar version. Enabling vectorization through Kokkos should improve the performances [Padioleau, 2020].

5.3.2 Implementation of other numerical schemes

Numerical schemes developed in chapters 3 and 4 are only implemented in sequential codes. Parallelism in the code ARK, which implements an all-regime solver for hydrodynamics, is handled through Kokkos and MPI. Because our all-regime scheme (chapter 3) has the same structure as the scheme for hydrodynamics, its implementation in ARK should be easily done. However, we have not done it because we aim for an implicit solver.

The Jacobi method, presented in section 4.1, can easily be made parallel. It could also be done with Kokkos and MPI. However, the geometric multigrid algorithm (section 4.2) raises the problem of load balancing when the coarser grids, with only a few unknowns, are reached.

5.4 Bibliography

H. Bloch, P. Tremblin, M. González, T. Padioleau, and E. Audit. A high-performance and portable asymptotic preserving radiation hydrodynamics code with the m1 model. *A&A*, 646:A123, 2021. doi: 10.1051/0004-6361/202038579. URL https://doi.org/10.1051/0004-6361/202038579. 118

Thomas Padioleau, Pascal Tremblin, Edouard Audit, Pierre Kestener, and Samuel Kokh. A high-performance and portable all-mach regime flow solver code with well-balanced gravity. application to compressible convection. *The Astrophysical Journal*, 875(2):128, apr 2019. doi: 10.3847/1538-4357/ab0f2c. URL https://doi.org/10.3847/1538-4357/ab0f2c. 118, 128

The Trilinos Project Team. *The Trilinos Project Website*. 118

H. Carter Edwards, Christian R. Trott, and Daniel Sunderland. Kokkos: Enabling manycore performance portability through polymorphic memory access patterns. *Journal of Parallel and Distributed Computing*, 74(12):3202 – 3216, 2014. ISSN 0743-7315. doi: https://doi.org/10.1016/j.jpdc.2014.07.003. URL http://www.sciencedirect.com/science/article/pii/S0743731514001257. Domain-Specific Languages and High-Level Frameworks for High-Performance Computing. 118

The Tpetra Project Team. *The Tpetra Project Website*. 119

The Teuchos Project Team. *The Teuchos Project Website*. 119

Aydn Buluç, Jeremy T. Fineman, Matteo Frigo, John R. Gilbert, and Charles E. Leiserson. Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks. In *IN SPAA*, pages 233–244, 2009. 122

H. Van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992. doi: 10.1137/0913035. URL https://doi.org/10.1137/0913035. 125

The Belos Project Team. *The Belos Project Website*. 125

Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003. ISBN 0898715342. 125, 127

The Ifpack2 Project Team. *The Ifpack2 Project Website*. 126

Yousef Saad. Ilut: A dual threshold incomplete lu factorization. *Numerical linear algebra with applications*, 1(4):387–402, 1994. ISSN 1070-5325. 126

Andrey Prokopenko, Christopher M. Siefert, Jonathan J. Hu, Mark Hoemmen, and Alicia Klinvex. Ifpack2 Users Guide 1.0. Technical Report SAND2016-5338, Sandia National Labs, 2016. 126

The Muelu Project Team. *The Muelu Project Website*. 127

Pierre Kestener. Implementing high-resolution fluid dynamics solver in a performance portable way with kokkos. Technical report, GPU Technology Conf. GTC, 05 2017. 128

Thomas Padioleau. *Development of "all-régime" AMR simulation methods for fluid dynamics, application in astrophysics and two-phase flows*. Theses, Université Paris-Saclay, December 2020. URL https://tel.archives-ouvertes.fr/tel-03130146. 134

Conclusion and perspectives

Contents

The goal of this thesis is to develop an asymptotic preserving implicit scheme for the M_1 model for radiative transfer. This scheme should also preserve the admissible states, $E_r > 0$ and $\frac{\|\boldsymbol{F}_r\|}{cE_r} \leqslant 1$.

In chapter 1, we have presented the radiative transfer equation. This model is costly to solve numerically, we derive a two-moment model by averaging the specific intensity over the direction of propagation and the frequency of the photons. We have to specify a closure relation, to express the radiative pressure as a function of the radiative energy and the radiative flux. We have chosen to use the M_1 model for its good properties in both optically thick and thin media. We compute the radiative pressure by maximizing the radiative entropy. We also have investigated the behavior of this model in the diffusive limit, with long timescale and large opacity. Finally, we have presented the model used for radiation hydrodynamics, where the hydrodynamics is no longer frozen.

In chapter 2, we have developed a first numerical scheme. This scheme is asymptotic preserving and time-implicit. It is based on a standard HLL scheme, numerical fluxes are modified with an asymptotic correction that allows us to capture the correct behavior in the diffusive limit. We also propose a well-balanced modification of the source term, to capture the correct steady state with a discontinuity of opacity. The nonlinear system arising from this discretization is solved using the Newton-Raphson method. Using this numerical scheme, we study the stability of the ionization front in a massive pre-stellar dense core in the presence of convection. However, numerical experiments have shown that this scheme does not preserve the admissible states.

In chapter 3, we have derived a second numerical scheme, asymptotic preserving and entropic. Because this scheme is entropic, we have shown that it preserves the admissible states. It is based on a splitting of operator, first developed for compressible hydrodynamics. We have rewritten the M_1 model in a form similar to the Euler equations. The first scheme we presented in this chapter is an adaptation of the work of Chalons et al. [2016]. We have then modified this scheme to obtain an unsplit scheme, with stencil 1. For both schemes, we have used the numerical fluxes proposed by Buet and Despres [2008], developed for a Lagrange-remap method. These fluxes take into account source terms and allow the scheme to be asymptotic preserving. For the sake of simplicity, these two schemes are presented with a time-explicit integration.

In chapter 4, we have presented an algorithm to solve the nonlinear system arising from a time-implicit HLL solver. We used the Jacobi method developed by Pichard [2016]. It is an iterative process and each iteration results in an admissible state. Nevertheless, it should be used carefully in the free-streaming regime. Increasing the number of iterations can increase the numerical diffusion (see section 4.3.2). Furthermore, the Jacobi method can be slow to converge when the resolution increases. To tackle this issue, we have used a nonlinear geometric multigrid algorithm [Briggs et al., 2000]. To help the convergence on a fine grid, a coarser grid is built, and the problem is solved on the coarse grid. This process can be applied recursively to solve the problem on the coarse grid. The Jacobi method is used as a smoother and coarse grid solver. The quantities moved between the different grids are the residual of the equation and the error. These quantities are not admissible in most cases. Following Kifonidis and Müller [2012], we introduce a pseudo-time at the coarse level and we use it to preserve the admissible states.

Finally, in chapter 5, we have presented the library Trilinos, used for linear algebra in a high performance computing context. We used it to solve the linear systems involved in the scheme developed in chapter 2. The package Kokkos allows us to

target different architectures, such as multi-cores, many-cores, and GP-GPUs. We have discussed the performances obtained thanks to Kokkos and the different preconditioners provided by Trilinos. We have also investigated the impact of the MPI domain decomposition and the preconditioner on the stability of the ionization front in a massive pre-stellar dense core.

Several perspectives arise from this work. To study astrophysical problems, an implicit and asymptotic preserving scheme is required. It should also preserve the admissible states. We have presented several schemes, each of them with two of these three properties. Developing a scheme with all of these properties, using the schemes presented in chapters 3 and 4, has not been done because of time constraints.

Such a scheme could be altered with a well-balanced modification of the source terms to capture properly the steady state with a discontinuity of opacity. In chapter 2, we have presented such a modification. For performance reasons, we did not take it into account in the schemes presented in chapter 3.

So far, the schemes presented in chapters 3 and 4 are implemented in a sequential C++ code. To study astrophysical problems, high performance computing is mandatory. We can achieve portability and performance portability with the library Kokkos.

Inspired by the resolution of the Poisson equation for self-gravity [Guillet and Teyssier, 2011], the geometric multigrid algorithm could be extended in the context of Adaptive Mesh Refinement (AMR), but is beyond the scope of this thesis.

5.1 Bibliography

Christophe Chalons, Pierre Kestener, Samuel Kokh, and Maxime Stauffert. A large time-step and well-balanced Lagrange-Projection type scheme for the shallow-water equations. July 2016. URL https://hal.archives-ouvertes.fr/hal-01297043. working paper or preprint. 138

Christophe Buet and Bruno Despres. A gas dynamics scheme for a two moments model of radiative transfer. 11 2008. 138

Teddy Pichard. *Mathematical modelling for dose depositon in photontherapy.* PhD thesis, 2016. URL http://www.theses.fr/2016BORD0177. Thèse de doctorat dirigée par Dubroca, Bruno et Frank, Martin Mathematiques appliquees et calcul scientifique Bordeaux 2016. 138

William Briggs, Van Henson, and Steve McCormick. *A Multigrid Tutorial, 2nd Edition.* 01 2000. ISBN 978-0-89871-462-3. 138

K. Kifonidis and E. Müller. On multigrid solution of the implicit equations of hydrodynamics - experiments for the compressible euler equations in general coordinates. *A&A*, 544:A47, 2012. doi: 10.1051/0004-6361/201116979. URL https://doi.org/10.1051/0004-6361/201116979. 138

Thomas Guillet and Romain Teyssier. A simple multigrid scheme for solving the poisson equation with arbitrary domain boundaries. *Journal of computational physics*, 230 (12):4756–4771, 2011. ISSN 0021-9991. 139

université PARIS-SACLAY

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

Titre : Développement et implémentation de schémas numériques pour l'hydrodynamique radiative

Mots clés : hydrodynamique radiative, schémas préservant l'asymptotique, schéma tout-régime, multigrille géométrique, calcul haute performance, algèbre linéaire

Résumé : Le transfert radiatif décrit les interactions entre la lumière et la matière, il apparaît donc dans de nombreux systèmes astrophysiques. Ce travail vise à développer des méthodes numériques pour l'hydrodynamique radiative, lorsque le rayonnement interagit avec un fluide. Nous utilisons le modèle M_1 qui est capable de capturer précisément les deux régimes rencontrés dans les problèmes impliquant du transfert radiatif : le milieu optiquement fin dans lequel les photons circulent librement et le milieu optiquement épais dans lequel les photons interagissent constamment et obéissent à une équation de diffusion dans la limite asymptotique. Nous présentons plusieurs schémas numériques, avec différentes propriétés : le bon comportement dans la limite de diffusion, une intégration implicite en temps et la préservation des états admissibles. Leur implémentation vise à être prête pour le calcul haute performance, sur des architectures exascales. Nous appliquons l'un d'eux à l'étude de la propagation d'un front d'ionisation dans un cœur dense pré-stellaire massif.

Title : Development and implementation of numerical schemes for radiation hydrodynamics

Keywords : radiation hydrodynamics, asymptotic preserving scheme, all-regime scheme, geometric multigrid, high performance computing, linear algebra

Abstract : The problem of radiative transfer describes the interaction between light and matte, therefore it appears in many astrophysical systems. This work aims at developing numerical methods for radiation hydrodynamics, when the radiation interacts with a fluid. We use the M_1 model that is able to accurately capture the two main regimes arising in problems involving radiative transfer: the optically thin medium in which photons are free-streaming and the optically thick medium in which photons are constantly interacting and obey a diffusion equation in the asymptotic limit. We present several numerical schemes, with different properties: the correct behavior in the asymptotic limit, a time-implicit integration, and the preservation of the admissible states. Their implementation aims at being ready for high-performance computing, on exascale architectures. We apply one of them to the study of the propagation of an ionization front in a massive pre-stellar dense core.

Maison du doctorat de l'Université Paris-Saclay
2ème étage aile ouest, Ecole normale supérieure Paris-Saclay
4 avenue des Sciences,
91190 Gif sur Yvette, France