# Contributions to the parallel simulation of two-phase flows and analysis of finite volume schemes on staggered grids

Katia Ait Ameur

**THÈSE DE DOCTORAT**

*présentée à*

**SORBONNE UNIVERSITÉ**

École doctorale : Sciences Mathématiques de Paris Centre (ED 386)

Par


# Katia AIT AMEUR


*Pour obtenir le grade de*

**DOCTEURE de SORBONNE UNIVERSITÉ**

Spécialité : Mathématiques Appliquées

## Contributions à la simulation parallèle d'écoulements diphasiques et analyse de schémas volumes finis sur grille décalée

Directeur de thèse : Yvon MADAY

Encadrant CEA : Marc TAJCHMAN

# Résumé

Dans cette thèse, l'apport le plus important a consisté en l'implémentation d'algorithmes modernes adaptés aux architectures massivement parallèles, dans un logiciel industriel dédié aux études de sûreté nucléaire, le code Cathare. Ce logiciel est dédié à la simulation des écoulements diphasiques au sein d'un réacteur nucléaire en conditions nominales ou accidentelles. L'implémentation de ces nouvelles techniques représentent en soi une contribution importante dans la physique des réacteurs car il permettra de déterminer, avec un temps de calcul réduit et de façon précise, l'état du cœur au cours d'accidents graves. Un effort particulier a été mené pour paralléliser de manière efficace la variable temporelle par l'algorithme pararéel. Pour cela, nous avons proposé une méthode pararéelle qui intègre de façon plus optimisée la présence de schémas en temps multi-pas. En effet, cette famille de schémas permet d'obtenir une approximation d'ordre supérieur à celui d'un schéma en temps à un pas. Cependant l'initialisation de la propagation en temps en chaque fenêtre doit être choisie avec soin. L'idée principale de ce nouveau schéma est de définir une approximation consistante des solutions permettant l'initialisation des propagations en temps, permettant ainsi à l'algorithme de converger vers la solution avec la précision voulue. Cette méthode a ensuite été appliquée sur deux cas tests représentatifs des défis numériques rencontrés dans la simulation des écoulements diphasiques dans le cadre des études de sûreté nucléaire.

La seconde partie de cette thèse est consacrée au développement de méthodes numériques permettant de traiter les difficultés numériques spécifiques aux modèles diphasiques avec un temps de calcul réduit. Dans cette partie, on développe un cadre d'analyse rigoureux pour l'étude des schémas volumes finis sur grille décalée comme celui utilisé dans le code Cathare. Les schémas décalés sont en pratique plus précis pour les fluides quasi incompressibles et sont couramment utilisés dans la communauté thermohydraulique. Cependant, pour les fluides compressibles, les études de stabilité ont été historiquement menées par une approche heuristique et par le réglage de paramètres numériques. Cette question est abordée par l'analyse des opérateurs de diffusion numérique qui permettent de porter un nouveau regard sur les schémas décalés. Cela nous permet de montrer que les schémas décalés classiques sont linéairement stables $L^2$ uniquement lorsque les vitesses sont de signe constant. On propose une classe de schémas décalés linéairement stables $L^2$ ainsi qu'une classe de schémas décalés entropiques. Ces nouvelles classes sont construites à l'aide d'un opérateur de diffusion numérique particulier et sont mieux adaptées aux modèles diphasiques pour lesquels les vitesses phasiques changent fréquemment de signe. Ces méthodes ont été appliquées au système d'Euler isentropique sur des cas tests analytiques et nous pensons que les développements actuels permettront à l'avenir son utilisation dans des cas plus réalistes et complexes, comme la simulation des écoulements diphasiques au sein d'une installation nucléaire.

# Abstract

In this thesis, the most important contribution has consisted in the implementation of modern algorithms that are well adapted for modern parallel architectures, in an industrial software dedicated to nuclear safety studies, the Cathare code. This software is dedicated to the simulation of two-phase flows within nuclear reactors under nominal or accidental situations. This work represents in itself an important contribution in nuclear safety studies thanks to the reduction of the computational time and the better accuracy that it can provide for the knowledge of the state of nuclear power plants during severe accidents. A special effort has been made in order to efficiently parallelise the time variable through the use of the parareal algorithm. For this, we have first designed a parareal scheme that takes more efficiently into account the presence of multi-step time schemes. This family of time schemes can potentially bring higher approximation orders than plain one-step methods but the initialisation of the time propagation in each time window needs to be appropriately chosen. The main idea consists in defining a consistent approximation of the solutions involved in the initialisation of the time propagations, allowing to reach convergence with the desired accuracy. Then, this method has been succesfully applied on test cases that are representative of the numerical challenges for the simulation of two-phase flows in the context of nuclear safety studies.

A second phase of our work has been to explore numerical methods that could handle better the numerical difficulties that are specific to two-phase flows with a lower computational cost. This part of the thesis has been devoted to the understanding of the theoretical properties of finite volume schemes on staggered grids such as the one used in the Cathare code. Staggered schemes are known to be more precise for almost incompressible flows in practice and are very popular in the thermal hydraulics community. However, in the context of compressible flows, their stability analysis has historically been performed with a heuristic approach and the tuning of numerical parameters. This question has been addressed by analysing their numerical diffusion operator that gives new insight into these schemes. For classical staggered schemes, the stability is obtained only in the case of constant sign velocities. We propose a class of linearly $L^2$-stable staggered schemes and a class of entropic staggered schemes. These new classes are based on a carefully chosen numerical diffusion operator and are more adapted to two-phase flows where phasic velocities frequently change signs. These methods have been successfully applied in analytical cases (involving Euler equations) and we expect that the present developments will allow its use in more realistic and complex cases in the future, like the one of the simulation of two-phase flows within a nuclear reactor during an accidental scenario.

# Acknowledgements

# Contents

# List of Figures

# List of Tables

# Introduction (Version française)

## Contexte industriel

On appelle "accident grave" ou "accident de fusion du cœur" d'un réacteur nucléaire à eau pressurisé un accident au cours duquel le combustible du réacteur est significativement dégradé avec fusion plus ou moins étendue du cœur du réacteur. La fusion résulterait d'une absence prolongée de refroidissement du cœur par le fluide caloporteur et consécutivement à une augmentation importante de la température des crayons combustibles dénoyés. C'est un type d'accident qui, en raison des mesures de prévention mises en place, ne peut survenir qu'à la suite d'une accumulation de dysfonctionnements (défaillances multiples, humaines ou matérielles).

Si la dégradation du cœur ne peut pas être arrêtée dans la cuve du réacteur par refroidissement du cœur dégradé (renoyage dans la cuve par le fluide caloporteur), l'accident peut à terme conduire à une perte de l'intégrité du confinement et à des relâchements importants de substances radioactives dans l'environnement. En raison des répercussions importantes qu'aurait un tel rejet, des efforts importants sont consacrés à l'étude de ce type de scénario pour pouvoir en limiter les conséquences (approche déterministe). L'étude des accidents de fusion du cœur passe en premier lieu par l'identification des principaux scénarios pouvant conduire à ce type d'accident. En complément des études déterministes, des études probabilistes de sûreté sont également menées. La méthode consiste à analyser de façon exhaustive tous les scénarios accidentels envisageables, d'estimer, souvent en les regroupant par famille, leur probabilité d'occurrence et les conséquences associées à l'intérieur de l'installation (fusion du cœur) ou à l'extérieur (rejets radioactifs dans l'environnement).

Dans le domaine des accidents graves, les phénomènes physiques mis en jeu sont extrêmement complexes. Les objectifs de la recherche sont donc de parvenir à comprendre au mieux ces phénomènes physiques et de développer des modèles applicables aux réacteurs. Ces modèles, regroupés au sein de codes de calcul informatiques, doivent permettre de prévoir le déroulement d'un accident grave. Comme il est impossible d'effectuer, dans ce domaine, des essais à taille réelle et de reproduire toutes les situations envisageables, il est nécessaire de réaliser des essais élémentaires, permettant d'étudier séparément chaque phénomène physique. Le tout doit se faire à des échelles compatibles avec les capacités techniques et économiques des installations, tout en restant représentatives pour l'extrapolation à l'échelle du réacteur. Les réacteurs expérimentaux constituent notamment des équipements privilégiés pour étudier le comportement des combustibles nucléaires en régime accidentel.

Dans ce contexte, le logiciel Cathare est un code thermohydraulique décrivant le réacteur nucléaire à l'échelle système, développé par le CEA depuis 1979. Ce code est dédié aux études de

sûreté pour les Réacteurs à Eau sous Pression et la validation des prodédures post-accidentelles.

L'objectif de cette introduction n'est pas d'effectuer une liste exhaustive de tous les scénarios accidentels faisant l'objet d'études de sûreté mais de décrire un type de scénario appelé APRP (pour Accidents de Perte de Réfrigérant Primaire) afin d'illustrer de manière synthétique la démarche alliant campagnes expérimentales et codes de simulation numérique.

Accidents de perte de réfrigérant primaire (APRP):

L'événement initiateur de ces accidents est une brèche dans la paroi du circuit primaire, [1]. La brèche provoque une fuite de réfrigérant primaire et une dépressurisation du circuit primaire. Plusieurs scénarios sont à distinguer selon l'état initial du réacteur, l'emplacement et la taille de la brèche. En cas d'APRP, la dépressurisation du circuit primaire entraîne l'arrêt automatique du réacteur, puis le démarrage automatique de l'injection de sécurité.

Les fonctions à assurer par les systèmes de protection et de sauvegarde pour limiter les conséquences de l'accident sont les suivantes:

- la maîtrise de la réactivité
- le maintien de l'inventaire en eau dans la cuve du réacteur
- l'évacuation de la puissance résiduelle dégagée par le combustible

La maîtrise de la réactivité est assurée par l'arrêt automatique du réacteur et l'injection d'eau borée dans le cœur. Le maintien de l'inventaire en eau dans la cuve du réacteur est assuré par le système d'injection de sécurité. L'évacuation de la puissance résiduelle dégagée par le combustible est assurée par le refroidissement de l'eau circulant dans la cuve. Les scenarii accidentels menant à une fusion du cœur supposent la défaillance de l'un ou de plusieurs des systèmes de sauvegarde et sont toujours associés à une défaillance du maintien d'eau en quantité suffisante dans le circuit primaire pour refroidir le cœur.

Le réacteur de recherche PHÉBUS [2] est un réacteur expérimental construit en 1977 sur le Centre d'études de Cadarache. Il a été conçu pour étudier le comportement des combustibles des centrales nucléaires dans des situations accidentelles du type perte de réfrigérant primaire pouvant aller jusqu'à la fusion du combustible. La principale problématique associée à l'étude des accidents de perte de réfrigérant est celle de la dégradation du combustible et de ses conséquences: à partir de quelle température, au bout de combien de temps a-t-on rupture de la gaine du combustible ou pire, fusion du cœur? Quel est le relâchement de produits de fission associé à ces deux phénomènes? Le réacteur PHÉBUS entre dans la catégorie des réacteurs d'essais en sûreté. L'objectif de ce programme était l'étude du comportement du combustible des réacteurs à eau sous pression (REP) dans des situations de perte de réfrigérant primaire correspondant à une situation accidentelle faisant suite à un fonctionnement en conditions nominales. Cet accident était suivi de la mise en œuvre du refroidissement de secours. La phénoménologie étudiée était liée à l'accident de référence des REP, qui ne va pas jusqu'à la fusion du cœur. Deux objectifs étaient recherchés:

- évaluer les marges relatives aux deux principaux critères retenus, dans le cadre du dimensionnement du réacteur: la température maximale et l'oxydation maximale des gaines
- valider les codes de comportement du combustible utilisés par l'analyse de sûreté et, en particulier, le module combustible du code de calcul Cathare.

De même, le logiciel Cathare a été validé sur 20 scénarii accidentels basés sur 10 installations expérimentales.

Par ailleurs, le code Cathare est également utilisé au sein de simulateurs de réacteurs. Le simulateur est notamment l'outil de formation au fonctionnement normal et accidentel des réacteurs à eau pressurisée pour des ingénieurs de l'IRSN ainsi que pour des opérateurs. Le rôle de l'opérateur au sein d'une centrale est d'activer les procédures de conduite post-accidentelles et de maintenir le réacteur en conditions de fonctionnement nominales. Parmi les moyens disponibles pour effectuer des études de sûreté des réacteurs, SOFIA - Simulateur d'Observation du Fonctionnement Incidentel et Accidentel [3] est un système informatique permettant le calcul et le suivi en temps réel de l'évolution des paramètres physiques d'un réacteur nucléaire. Il permet de simuler les défaillances de matériel et les actions des opérateurs, d'arrêter le calcul pour examiner l'état de l'installation à un instant donné et de revenir en arrière, afin de modifier le scénario. Le simulateur SOFIA utilise le code de calcul Cathare en temps réel.

## Objectifs de la thèse

Le code Cathare est un code à l'échelle système et décrit l'ensemble d'un réacteur nucléaire à l'aide d'un assemblage de conduites, de cuves et de pompes dont la taille est d'environ 10 mètres (taille de la cuve: $13m \times 5m \times 5m$). Pour simuler l'ensemble de l'installation nucléaire, le code Cathare possèdent des restrictions au niveau de la taille des cellules du maillage: en prenant en compte la taille importante des circuits, la taille d'une maille peut être relativement grande, et peut aller de quelque centimètres à un mètre. Après discrétisation des variables, les simulations numériques font intervenir entre $10^2$ et $10^3$ inconnues et jusqu'à un million de pas de temps. Une méthode de décomposition de domaine en espace est implémentée dans le code Cathare et les performances atteignent actuellement un plateau malgré qu'il y ait des ressources informatiques supplémentaires. Les performances de cette méthode de décomposition de domaine en espace sont limitées car les maillages utilisés dans les simulations du code Cathare sont peu raffinés afin de représenter l'ensemble du réacteur nucléaire. Le premier objectif de cette thèse est de proposer une nouvelle stratégie de parallélisation, complémentaire à la méthode de décomposition de domaine en espace. Pour cette raison, on propose d'élaborer une méthode de décomposition de domaine en temps. De plus, on souhaite appliquer cette stratégie de manière non intrusive et utiliser le code Cathare en boîte noire. Pour cela, on se base sur l'algorithme pararéel. Les résultats à ce sujet sont présentés dans les chapitres 2 et 3.

Le restant des chapitres de cette thèse est motivé par le besoin en méthodes numériques nouvelles pour mieux traiter les difficultés numériques spécifiques aux modèles diphasiques avec un coût en temps de calcul raisonnable. Cette seconde partie est dédiée à l'analyse des aspects théoriques des schémas volumes finis sur grille décalée, comme celui utilisé dans le code Cathare. L'objectif est de développer une méthode d'analyse de stabilité rigoureuse des schémas décalés classiques et de proposer une nouvelle classe de schémas décalés stable $L^2$. De même, on souhaite développer une nouvelle classe de schémas décalés entropiques. La méthodologie développée dans le manuscrit est très générale et pourrait s'appliquer aux modèles diphasiques.

Dans les paragraphes suivants, nous présentons un bref résumé de chaque chapitre de ce manuscrit.

## Résumé des résultats par chapitres

### Partie I: Chapitre 1

Ce premier chapitre a essentiellement pour but de résumer les connaissances actuelles que l'on peut trouver dans la bibliographie au sujet du modèle diphasique utilisé dans le code Cathare: le modèle bifluide à 6 équations. La plupart de ce qui est donc présenté n'est pas nouveau, mais il nous a semblé intéressant de présenter cette compilation d'informations pour positionner le modèle et les méthodes numériques du code Cathare dans la littérature sur la simulation des écoulements diphasiques.

Après la présentation du modèle bifluide à 6 équations, nous rappellerons les principaux résultats théoriques concernant l'hyperbolicité du modèle. Nous présenterons ensuite la structure complexe de la solution du modèle qui est spécifique aux modèles diphasiques.

Les techniques de discrétisation des variables du modèle seront ensuite présentées en insistant particulièrement sur le traitement du terme de convection par les schémas décalés les plus répandus. Bien que les modèles diphasiques héritent des connaissances actuelles sur la modélisation et les méthodes numériques pour les fluides monophasiques, ils possèdent néanmoins plusieurs spécificités. Nous présenterons certaines d'entre elles avec notamment le traitement des produits non conservatifs, la configuration des phases évanescentes et le traitement des termes sources discontinus.

La dernière partie du chapitre est consacrée aux techniques d'accélération actuellement déployées dans le code Cathare. On présentera également les facteurs qui limitent les performances de ces méthodes malgré la disponibilité de ressources informatiques supplémentaires. On présente ensuite la stratégie de parallélisation en temps choisie pour le code Cathare. La parallélisation de la variable temporelle est particulièrement délicate étant donné que le temps est séquentiel par nature. Malgré cela, plusieurs stratégies ont été proposées à ce sujet-là dans la littérature (see [27], [50]). Nous nous sommes concentrés sur la méthode pararéelle car c'est celle qui donne les meilleures performances sur des applications complexes (voir notamment [13], [49], [104]).

### Partie I: Chapitre 2

Le deuxième chapitre résume les moyens mis en œuvre pour implémenter l'algorithme pararéel au code Cathare de manière non intrusive. Nous présenterons les deux outils que nous avons développé durant la thèse pour appliquer l'algorithme pararéel au code Cathare: d'une part, une maquette du code Cathare restreinte à un cas test et d'autre part une librairie qui utilise le code Cathare en boîte noire de manière parallèle. Les deux cas tests que nous avons étudiés sont représentatifs des défis numériques rencontrés dans la simulation des écoulements diphasiques dans le cadre des études de sûreté. Ces défis numériques comprennent notamment les phases évanescentes pour lesquelles une des phases liquide ou vapeur disparaît dans une partie du domaine ou encore la simulation d'une brèche dans le circuit primaire d'une réacteur causant ainsi une dépressurisation rapide dans le système.

La contribution principale de ce travail est l'adaptation de l'algorithme pararéel à l'architecture logicielle du code Cathare et sa discrétisation en temps de manière non intrusive, sans modifier les fichiers sources du code, dans le but de réduire le temps de calcul et de se rapprocher d'une

réponse en temps réel du code.

Finalement, dans les exemples numériques que nous avons traité, l'utilisation de l'algorithme pararéel peut accélérer les calculs d'environ un facteur 3 avec 25 processeurs. Du point de vue de l'efficacité, ces résultats ne sont pas aussi compétitifs que les méthodes de décomposition de domaine en espace fournies, mais comme il sera expliqué au chapitre 2, il existe des raisons théoriques qui expliquent la relativement basse efficacité de la méthode pararéelle. Pour cette raison, cette méthode devient intéressante pour atteindre des performances additionnelles dans un contexte où les autres techniques de parallélisation dont on peut disposer atteignent saturation.

### Partie I: Chapitre 3

Comme il sera expliqué aux chapitres 1 et 2, la discrétisation en temps dans le code Cathare est basée sur un schéma en temps à deux pas. Un schéma en temps multi-pas permet d'obtenir une approximation d'ordre supérieur à celui d'un schéma en temps à un pas cependant l'initialisation de la propagation en temps en chaque fenêtre doit être définie avec rigueur. Lorsque le solveur fin et/ou le solveur grossier est un schéma en temps multi-pas, il est nécessaire de définir une approximation consistante des solutions intervenant dans l'initialisation du schéma fin pour chaque fenêtre en temps. Autrement, l'erreur commise à l'initialisation sera propagée sur l'intervalle de temps et empêchera l'algorithme pararéel de converger vers la solution avec la précision souhaitée.

Dans l'objectif d'aborder ce problème, nous présentons dans ce chapitre une nouvelle variante de l'algorithme pararéel adaptée à ce type de discrétisation et qui permet de converger vers la solution cible avec un taux de convergence similaire à celui de l'algorithme pararéel classique. Un effort particulier a été réalisé afin de construire un algorithme adapté aux schémas multi-pas de manière non intrusive dans les solveurs grossier et fin. Cela permet au code Cathare d'être utilisé en boîte noire, assurant ainsi la portabilité de ce nouvel algorithme. Concernant la méthode d'initialisation, l'algorithme pararéel multi-pas est plus consistant avec le schéma sous-jacent. Nous montrons à l'aide de résultats théoriques et numériques que les propriétés de précision et de convergence de l'algorithme pararaéel multi-pas sont compétitives lorsque l'on initialise rigoureusement chaque fenêtre en temps.

### Part II: Chapitre 4

La seconde partie de cette thèse est consacrée au développement d'un cadre d'analyse rigoureux pour l'étude des schémas volumes finis sur grille décalée comme celui utilisé dans le code Cathare. En particulier, la méthode présentée pourrait être appliquée dans le futur au code Cathare pour traiter les spécificités numériques propres aux modèles diphasiques. Pour développer un tel outil, il a été nécessaire tout d'abord d'étudier préalablement certains aspects théoriques et c'est ce qui est présenté dans la deuxième partie de ce manuscrit. Plusieurs exemples numériques simples seront aussi présentés, notamment sur la résolution du système d'Euler isentropique, dans le but d'illustrer la technique proposée ainsi que ses performances.

Les schémas décalés sont en pratique plus précis pour les fluides quasi incompressibles et sont couramment utilisés dans la communauté thermohydraulique ([63, 98, 26]). Cependant, dans le contexte des fluides compressibles, les études de stabilité ont été historiquement menées par une approche heuristique et par le réglage de paramètres numériques ([70]). Dans [66, 65, 67], les auteurs construisent des schémas décalés conservatifs avec des preuves rigoureuse de stabilité: le

caractère entropique et l'inégalité discrète de l'énergie cinétique. Néanmoins, le caractère bornée de l'entropie n'implique pas nécessairement que la solution reste bornée et c'est particulièrement le cas pour le système d'Euler complet. Par conséquent, on étudie dans ce chapitre la stabilité $L^2$ linéaire des schémas décalés.

Cette question est abordée par l'analyse des opérateurs de diffusion numérique qui permet de porter un nouveau regard sur les schémas décalés. On développe d'abord la forme de l'opérateur de diffusion numérique pour les schémas décalés classiques, ce qui permet de montrer que ces schémas sont linéairement stables $L^2$ uniquement lorsque les vitesses sont de signe constant. On propose ensuite une classe de schémas décalés linéairement stables $L^2$. Cette nouvelle classe est construite à l'aide d'un opérateur de diffusion numérique particulier et est mieux adaptée aux modèles diphasiques pour lesquels les vitesses phasiques changent fréquemment de signe.

Un schéma numérique appartenant à cette nouvelle classe de schémas décalés a été implémenté avec succès pour la simulation d'un problème de Riemann et nous pensons que les développements actuels permettront à l'avenir son utilisation dans des cas plus réalistes et complexes.

### Part II: Chapitre 5

Dans le dernier chapitre, nous abordons la question du caractère entropique des schémas décalés. On écrit le bilan d'entropie discret pour une classe de schémas décalés. A partir de là, on propose des conditions explicites sur les coefficients de la matrice de diffusion numérique pour garantir la dissipation de l'entropie discrète. La méthodologie est très générale et pourrait s'appliquer aux modèles diphasiques. On applique en premier lieu cette méthode au système d'Euler isentropique. On implémente ensuite un schéma appartenant à cette nouvelle classe schémas décalés entropiques pour la simulation d'un problème de Riemann dont la solution est composée d'une onde de détente transonique. Ces résultats numériques illustrent bien que notre méthode capture la solution entropique correcte.

# Introduction (English version)

## Industrial context

We call "serious accident" or "meltdown accident" of a pressurised water reactor an accident where the fuel of the reactor is significantly degraded with a more or less extensive meltdown of the reactor core. This meltdown would be the consequence of a prolonged absence of the coolant in the core causing an increase of the dewater fuel rods temperature. Due to the actual prevention measures, this type of accidents may only occur after an accumulation of malfunctions (multiple failures, human or material).

If the damages can not be stopped in the vessel by the cooling of the damaged core (reflooding of the vessel by the coolant) then the accident could lead to a loss of containment and to significant releases of radioactive substances in the environment. Due to the major consequences of such a sequence of events, several studies of these accidents are conducted to limit their consequences (deterministic approach). These studies firstly start by the identification of the main scenarii that can lead to the reactor core meltdown. Complementing the deterministic studies, probabilistic safety studies are also conducted. They consist in analysing exhaustively every possible accidental scenarii, in estimating its probability of occurence and their consequences inside the nuclear power plant (core meltdown) or outside (release of radioactive substances in the environment).

The physical phenomena encountered in nuclear safety are extremely complex. This research field is dedicated to the understanding of these phenomena and to develop physical and mathematical models applicable to reactors. These models are grouped within softwares for numerical simulation and allow to predict the course of a serious accident. Since it is impossible in nuclear safety to reproduce all possible accidental situations on real size experimental installations, it is necessary to perform several elementary experiments allowing to study seperately each physical phenomenon. These steps must be at scales that are compatible with the technical and economical constraints while remaining close to the behavior of the nuclear power plant. The experimental reactors constitute in particular a privileged equipment to study the behavior of nuclear fuel in accidental situation and represent a considerable economic and human investment.

In this context, the Cathare code (Code for Analysis of THermalhydraulics during Accident and for Reactor safety Evaluation) is a thermalhydraulic code describing a nuclear reactor at the system scale, developed by CEA since 1979 as part of an agreement between CEA, EDF, AREVA and IRSN. This software is dedicated to the safety studies of Pressurised Water Reactor and the validation of emergency procedures during an accidental scenario.

The objective of this introduction is not to give an exhaustive account of all the possible

accidental scenarii studied in nuclear safety but to describe a type of accidents called LOCA (Loss Of Coolant Accident) to illustrate synthetically the approach that couples experimental campaigns and numerical simulation softwares.

Loss Of Coolant Accident (LOCA):

The initiating event of these accidents is a breach in the primary circuit, [1]. This breach causes a leak of the coolant and a depressurisation of the primary circuit. Many accidental scenarii are then possible according to the initial state of the reactor, the location and the size of the breach. In the case of a LOCA, the depressurisation in the primary circuit will generate an automatic shutdown of the reactor and an automatic start of backup cooling system.

To limit the consequences of these accidents, protection and backup systems will ensure the following functions:

- control of the reactivity
- maintaining the water inventory in the reactor vessel
- evacuate the residual heat released by the fuel rods

The control of the reactivity is ensured by the automatic shutdown of the reactor and by the injection of borated water within the core. The water inventory in the vessel is maintained by the safety injection of water. The residual heat released by the fuel rods is evacuated by the cooling of the water circulating in the vessel. Accidental scenarii leading to a core meltdown are the consequences of the failure of one or several backup systems and are always associated to a failure in maintaining enough coolant in the primary circuit to cool the reactor core.

The research reactor PHEBUS [2] is an experimental reactor built in 1977 in the CEA research Center of Cadarache. It is dedicated to the study of the fuel rods behavior in nuclear power plants under accidental conditions of LOCA type. The main issue associated to this type of accidents is the fuel rods degradation and its consequences: from which temperature, after how long do we have the break of the fuel rod cladding or worse core meltdown? How much fission products are released due to these two phenomena? The PHEBUS reactor is an experimental reactor for safety studies. The objective of this program is to study the behavior of fuel rods for pressurised water reactors under LOCA type accidents especially. There are two main goals:

- evaluate the relative margins for two selected criteria, in order to sizing the reactor: the maximum temperature and the maximum oxidation of the fuel rod cladding
- validate the softwares used in the nuclear safety studies, particularly the fuel module of the Cathare code.

Likewise, the Cathare code has been validated on 20 accidental scenarii based on 10 experimental installations.

In addition, the Cathare code is daily used on reactor simulators for the training of operators. During an accidental scenario, the role of the operator is to activate emergency procedures to keep the reactor in nominal working conditions. Reactor simulators are used by IRSN and other French and foreign organisations to train their engineers. Among the available tools to make nuclear safety studies, SOFIA - Simulateur d'Observation du Fonctionnement Incidentel et Accidentel [3], is a computer system allowing the real-time tracking of the evolution of many physical parameters

in a nuclear reactor. It simulates the material failures and actions of the operator, it also stops the computation to analyse the installation at a given time and step back to modify the scenario. The SOFIA simulator uses the Cathare code in real-time.

## Aims of this thesis

The Cathare code is a system code and describes the whole reactor as an assembly of pipes, vessels and pumps whose sizes are around 10 meters (size of the vessel: $13m \times 5m \times 5m$). To simulate the whole system, the Cathare code has restrictions on the mesh size: taking into account the important size of the circuits, the mesh size used can be relatively large, from a few centimeters to a meter. After the discretisation of all the variables, typical cases involve up to $10^2$ or $10^3$ cells with 3D elements and involve up to a million of numerical time steps. A space domain decomposition method is implemented in the Cathare code and reaches its limits in its ability to use the entire computational resources. The scalability properties of the space domain decomposition method are limited by the small number of cells in the meshes of the Cathare simulations. We seek in this work to investigate a novel strategy of parallelisation to complement the actual parallelism in the Cathare code. For this reason, if we have more processors at our disposal and wish additional speed-ups, the parallelisation of other variables needs to be addressed. Our purpose is to design a strategy of time domain decomposition. We would like to use a non intrusive approach where the Cathare code is used as a black box. In this context we investigate the ability of the parareal method to match our requirements. This work constitutes the first part of this thesis and is presented in chapters 1, 2 and 3.

The remaining chapters of this thesis are motivated by the need of novel numerical methods that could handle better the numerical difficulties that are specific to two-phase flows with a lower computational cost. This second part is dedicated to the understanding of the theoretical properties of finite volume schemes on staggered grids such as the one used in the Cathare code. We seek to develop a rigorous framework for the stability analysis of classical staggered schemes and to propose a class of $L^2$-stable staggered schemes. In addition, we seek to derive a class of entropic staggered schemes. The procedure derived in the thesis is very general and could be applied to two-phase flows models.

In the following section, a summary of every chapter will be provided.

## Summary of the results by chapters

### Part I: Chapter 1

The aim of this first chapter is twofold: first, it is intended to provide a bibliographical overview of the two-phase flow model used in the Cathare code, namely the six-equation two-fluid model. Most of what is stated here is not new but it seemed interesting to us to present this compilation of information to position the model and numerical methods in the Cathare code within the existing litterature of two-phase flows simulation.

After introducing the six-equation two-fluid model, the main theoretical results regarding the hyperbolicity of the model will be presented. We will then explain the complex nature of the

solution displayed by the model that is specific to two-phase flows models.

We will continue by recalling the existing discretisation techniques of the variables involved in the equations and a special emphasis will be put ont the treatment of the convection term for the most widespread staggered schemes. Even if two-fluid models inherit achievements performed in the modeling, mathematical theory and numerical methods for single-phase flows, they however display many specific difficulties. We will discuss some difficulties existing in the two-phase flow models such as the presence of non conservative products, the configuration of the vanishing phase and the handling of discontinuous source terms.

The last part of the chapter is devoted to the existing acceleration techniques that are actually available in the Cathare code. First, these methods will be presented as well as their limits in the ability to use the entire computational resources. We finally present the strategy followed in the Cathare code to parallelise the time variable. The parallelisation of the time variable is particularly involved given the sequential nature of time. Despite this, several strategies have been proposed in the literature (see [27], [50]). We have focused on the parareal in time method because it is the one that seems to provide the best performances with many applications (see [13], [49], [104] among many others).

### Part I: Chapter 2

The second chapter summarises the special efforts we made to implement the parareal algorithm to the Cathare code in a non intrusive way. We will present the two computational tools we developped during the PhD in order to apply the parareal algorithm to the Cathare code: firstly a numerical clone of Cathare that is restricted to one test case and secondly through a library that uses the Cathare code as a black box in a parallel way. The two test cases we investigate are representative of the numerical challenges for the simulation of two phase flows in the context of safety studies. Numerical challenges include for instance, the vanishing phase issue where one of the two phases liquid or gas disappears in some parts of the domain or the simulation of a breach in the primary circuit that causes a fast depressurisation within the reactor core.

The main contribution of this work has been to adapt the parareal algorithm to the architecture of the software and to its time discretisation in a non intrusive way, without modification of the source files of the Cathare code, in order to reduce the computational time and get closer to a real-time response of the code.

In the numerical examples treated, the use of the parareal algorithm can speed-up the calculations by a factor of about 3 with 25 processors. From an efficiency point of view, these results are not as competitive as the high efficiency that domain decomposition methods provide, but, as it will be explained in chapter 2, there are theoretical reasons that explain the relatively low efficiency of the parareal method. Because of this fact, parareal is a useful technique to obtain additional speed-ups in the context where other more efficient parallelisation techniques have reached saturation.

### Part I: Chapter 3

As will be presented in detail in chapters 1 and 2, the time discretisation of the Cathare code relies on a two-step time scheme. A multi-step time scheme can potentially bring higher approximation

orders than plain one-step methods but the initialisation of the time propagation in each time window needs to be appropriately chosen. When the fine and/or coarse propagators is a multi-step time scheme, we need to choose a consistent approximation of the solutions involved in the initialisation of the fine solver at each time window. Otherwise, an initialisation error would be propagated over the whole time interval and would prevent the parareal algorithm to converge towards the solution with the desired accuracy.

In an attempt to address this issue, this chapter presents a new variant of the parareal algorithm, adapted to this type of discretisation, and that ensures to recover the target solution with a convergence rate similar to the one of the classical parareal algorithm. A special effort has been made to design an algorithm adapted to this type of discretisation without being intrusive in the coarse or fine solvers. This allows to the Cathare code to be treated as a black box, which ensures the portability of this new algorithm. With regard to the initialisation procedure, the multi-step parareal algorithm is more consistent with the underlying time scheme. We show both theoretically and numerically that the accuracy and convergence of the multi-step parareal algorithm are very competitive when we choose carefully the initialisation of each time window.

### Part II: Chapter 4

The second part of this thesis is devoted to the development of a rigorous framework for the analysis of finite volume schemes on staggered grids such as the one used in the Cathare code. In particular, the family of schemes presented here could be applied in the future to the Cathare code to handle the numerical difficulties specific to two-phase flows models. The derivation of the method has required the analysis of some theoretical aspects beforehand and this is what is presented in this second part of the manuscript. Nevertheless, some analytical numerical examples will be presented on the solution of the isentropic Euler equations with the purpose of illustrating the technique and its performances.

Staggered schemes are known to be more precise for almost incompressible flows in practice and are very popular in the thermal hydraulics community ([63, 98, 26]). However, in the context of compressible flows, their stability analysis has historically been performed with a heuristic approach and the tuning of numerical parameters ([70]). Yet the conservative staggered schemes presented in [66, 65] are proven to be entropic and to satisfy a kinetic energy preservation [67]. Unfortunately, the boundedness of the entropy does not necessarily imply the boundedness of the solution and this is particularly the case for the full Euler system. Hence, we investigate in this chapter the linear $L^2$-stability of staggered schemes. This question has been addressed by analysing their numerical diffusion operator that gives a new insight into these schemes. We first derive the numerical diffusion operator for classical staggered schemes and show that the $L^2$ stability is obtained only in the case of constant sign velocities. We then propose a class of linearly $L^2$-stable staggered schemes. This new class is based on a carefully chosen numerical diffusion operator and is more adapted to two-phase flows where phasic velocities frequently change signs.

A scheme belonging to this new class of staggered schemes has been successfully applied to the simulation of a Riemann problem and we expect that the present developments will allow its use in more realistic and complex cases in the future.

### Part II: Chapter 5

In the last chapter, we address the question of the entropic character of staggered schemes. We derive a discrete entropy balance for a class of staggered schemes. On this basis, we give explicit

constraints on the coefficients of the numerical diffusion matrix to ensure the dissipation of the discrete entropy. The procedure is very general and could be applied to two-phase flows models. We first investigate this strategy on the isentropic Euler system. We then implement a scheme belonging to this new class of entropic staggered schemes for the simulation of a Riemann problem that displays transonic rarefaction waves. These numerical results illustrate the ability of our method to capture the correct entropic solution in a stable way.

# Part I

# Numerical models for two-phase flows for safety studies

# Chapter 1

# Challenges of two phase flows simulation for safety studies

**Contents**



This first chapter is intended to be a bibliographic summary about the six-equation two-fluid model: an overview of some theoretical results, discretisation, numerical and HPC methods to solve the model are presented. We will also show the numerical difficulties that are specific to two-phase flows and represent an obstruction to obtain a satisfactory simulation at a reasonnable computational cost. In this context, the main contribution of this work has been to explore parallel acceleration techniques to reduce this computational time (see chapters 2 and 3) and also to explore numerical methods that could handle better these numerical difficulties with a lower computational cost (see part II of this manuscript).

## 1.1 The six-equation two-fluid model

### 1.1.1 Balance equations

Considering a two-phase flow of vapour (subscript $_v$) and liquid (subscript $_l$) phases, we are interested in determining unknown physical quantities that are the vapour and liquid volume fractions ($\alpha_v, \alpha_l \in [0,1]$), average pressures ($p_v \in \mathbb{R}_+$ and $p_l \in \mathbb{R}_+$), densities ($\rho_v \in \mathbb{R}_+$ and $\rho_l \in \mathbb{R}_+$), temperatures ($T_v \in \mathbb{R}_+$ and $T_l \in \mathbb{R}_+$), velocities ($\vec{u}_v \in \mathbb{R}^3$ and $\vec{u}_l \in \mathbb{R}^3$), enthalpies ($h_v \in \mathbb{R}_+$ and $h_l \in \mathbb{R}_+$) and total energies ($E_v \in \mathbb{R}_+$ and $E_l \in \mathbb{R}_+$).
The two-fluid model equations are obtained by averaging the balance equations for each separated phase, using space, time or ensemble averaged quantities (see [39, 71, 72]). The mass, momentum and energy balance equations for each phase then read as follows:

$$
\begin{cases}
\dfrac{\partial \alpha_v \rho_v}{\partial t} + \nabla \cdot (\alpha_v \rho_v \vec{u}_v) = \Gamma_v, \\
\dfrac{\partial \alpha_l \rho_l}{\partial t} + \nabla \cdot (\alpha_l \rho_l \vec{u}_l) = \Gamma_l, \\
\dfrac{\partial \alpha_v \rho_v \vec{u}_v}{\partial t} + \nabla \cdot (\alpha_v \rho_v \vec{u}_v \otimes \vec{u}_v) + \alpha_v \vec{\nabla} p_v = \nabla \cdot (\alpha_v (\bar{\bar{\tau}}_v + {}^t\bar{\bar{\tau}}_v)) + \alpha_v \rho_v \vec{g} + \vec{F}_v^{int} + \vec{F}_v^{w} + \Gamma_v \vec{u}_v^{int}, \\
\dfrac{\partial \alpha_l \rho_l \vec{u}_l}{\partial t} + \nabla \cdot (\alpha_l \rho_l \vec{u}_l \otimes \vec{u}_l) + \alpha_l \vec{\nabla} p_l = \nabla \cdot (\alpha_l (\bar{\bar{\tau}}_l + {}^t\bar{\bar{\tau}}_l)) + \alpha_l \rho_l \vec{g} + \vec{F}_l^{int} + \vec{F}_l^{w} + \Gamma_l \vec{u}_l^{int}, \\
\dfrac{\partial \alpha_v \rho_v E_v}{\partial t} + p_v^{int} \dfrac{\partial \alpha_v}{\partial t} + \nabla \cdot (\alpha_v \rho_v \vec{u}_v (E_v + \dfrac{p_v}{\rho_v})) = \alpha_v \rho_v \vec{u}_v \cdot \vec{g} + \vec{F}_v^{int} \cdot \vec{u}_v^{int} + \sigma_v^Q \\
\qquad\qquad + \nabla \cdot (\alpha_v (q_v + {}^t q_v)) + \Gamma_v (h_v^{int} + \dfrac{(\vec{u}_v^{int})^2}{2}) + Q_v^w, \\
\dfrac{\partial \alpha_l \rho_l E_l}{\partial t} + p_l^{int} \dfrac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \rho_l \vec{u}_l (E_l + \dfrac{p_l}{\rho_l})) = \alpha_l \rho_l \vec{u}_l \cdot \vec{g} + \vec{F}_l^{int} \cdot \vec{u}_l^{int} + \sigma_l^Q \\
\qquad\qquad + \nabla \cdot (\alpha_l (q_l + {}^t q_l)) + \Gamma_l (h_l^{int} + \dfrac{(\vec{u}_l^{int})^2}{2}) + Q_l^w.
\end{cases}
\tag{1.1}
$$

One also has to consider the closure relation:

$$\alpha_v + \alpha_l = 1,$$

which expresses the fact that the whole available volume is filled up with the fluids.
The interface interaction transfers consist in the following terms:

- $\Gamma_k$ the interfacial mass transfer terms,
- $\vec{F}_k^{int}$ the interfacial momentum transfer terms, which are either of algebraic or differential nature. They result of the viscous stresses at the interface and pressure forces normal to the interface. They can be further divided into a non-viscous component $\vec{F}_k^{nv}$ containing for instance the interfacial pressure term, the virtual mass force and a viscous component such as the interfacial friction term. The interfacial pressure term takes the following form:

$$\vec{F}_k^{p} = (p_k^{int} - p_k) \vec{\nabla} \alpha_k,$$

- $\vec{u}^{int}$ the interfacial velocity,
- $\sigma_k^Q$ the interfacial heat transfer terms.

The wall interaction modeling, denoted by the subscript $w$, includes wall heat transfer and wall frictional forces:

- $Q_k^w$ the wall heat transfer rates,
- $\vec{F}_k^w$ the wall frictional forces.

$\bar{\bar{\tau}}_k$ are the Reynolds stresses and $q_k$ are the turbulent heat fluxes.

### Equations of state

The number of unknown variables is in general greater than the number of equations in the system of PDEs. Therefore, it is necessary to consider supplementary constitutive equations such as equations of state. In general, such equations depend on the specific two-phase flow model and on the flow regimes. The equations of state considered are: $\rho_k = \rho_k(p, e_k)$.

The Cathare code uses tabulated equations of state based on the industrial formulation IAPWS (The International Association for the Properties of Water and Steam, [116]). These equations of state compute all the thermodynamical variables necessary for the simulation thanks to experimental measurements of pressures and temperatures and polynomial interpolations.

## 1.1.2 Closure laws

The jump conditions at the interface are:

- Mass transfer:
$$\sum_{k=g,l} \Gamma_k = 0.$$
- Interface momentum transfer:
$$\sum_{k=g,l} \vec{F}_k^{int} = 0.$$
- Energy transfer:
$$\sum_{k=g,l} (\Gamma_k h_k^{int} + \sigma_k^Q) = 0.$$

In our work in this thesis, we have worked under the following assumptions:

- We assume here that the Reynolds stresses and turbulent heat fluxes are negligible, as we are considering convection driven flow where viscosity tensor play a minor role.
- There is only one bulk average pressure in the system. We assume that the pressures relaxation time is negligible, and pressure equilibrium is considered to be reached in the flow: $p_v(\rho_v, E_v) = p_l(\rho_l, E_l) = p$.
- We neglect surface tension phenomena in the two-fluid model: $p_v^{int} = p_l^{int} = p^{int}$.
- We neglect as many authors do the pressure default $(p - p^{int})$ in the energy equation.

Providing the general expressions of the momentum transfer terms is a difficult task with many ongoing works. We summarise in the sequel some expressions that are commonly employed:

**Interfacial pressure term**
If we neglect virtual mass force, the Cathare model for the interface pressure force is given by [20]

$$\Delta p = p - p^{int} = \delta \frac{\alpha_v \alpha_l \rho_v \rho_l}{\alpha_v \rho_l + \alpha_l \rho_v} ||\vec{u}_v - \vec{u}_l||^2.$$

The coefficient $\delta$ can be gauged such that the two-fluid model becomes hyperbolic ([89]). Let us note that the Cathare model for the interfacial pressure term is slightly different in the case of stratified flows. We will give more details about this point in the section dedicated to the hyperbolicity of the six equation two-fluid model.

**Virtual mass force**

The dynamic drag or the transient forces that result from relative acceleration of the phases, usually also called the added mass effects is modeled in Cathare with the following formula, issued from [20]

$$\vec{F}_v^{vm} = \vec{F}_l^{vm} = -\beta_{vm}\alpha_v\alpha_l\rho_m \left[ \left( \frac{\partial \vec{u}_v}{\partial t} + \vec{u}_v \cdot \vec{\nabla}\vec{u}_v \right) - \left( \frac{\partial \vec{u}_l}{\partial t} + \vec{u}_l \cdot \vec{\nabla}\vec{u}_l \right) \right],$$

where $\beta_{vm}$ is the virtual mass coefficient depending on the flow regime.

**Interfacial friction term or Drag force**

The drag model depends on the flow regime (dispersed, stratified, ...) and has the following general form:

$$\vec{F}^D = \frac{1}{2} C_D a_i \rho k(\alpha) ||u_v - u_l||^2,$$

where $C_D$ is the drag coefficient and $k(\alpha)$ is a function that strongly couples the phases when one of them tends to disappear. More details will be given in section 1.2.5 dedicated to one important numerical difficulty of the two-fluid model: the vanishing phase.

**Interfacial velocity**

Assuming that the no slip condition is satisfied, the interfacial velocities for momentum and energy transfer are equal for both phases $u_v^{int} = u_l^{int} = u^{int}$ . The formulation used in Cathare is a volume fraction averaged formulation:

$$\vec{u}^{int} = \alpha_v \vec{u}_l + \alpha_l \vec{u}_v.$$

### 1.1.3 Boundary conditions and initial condition

We will assume that the domain $\Omega$ is a bounded open set of $\mathbb{R}^3$. We denote the boundary $\partial\Omega$ with $\vec{n}(X)$ the outward unit normal to $\partial\Omega$ at point $X$. We define the following partitions of $\partial\Omega$:

$$\partial\Omega = \partial\Omega^{in} \cup \partial\Omega^{out} \cup \partial\Omega^{wall}.$$

We list here some of the usual boundary conditions that are associated to problem (1.1).

**Inlet and outlet boundary conditions**

The flow velocity profile is specified at inlet boundaries to model the incoming of liquid or vapour in the domain:

$$u_k(t, X) \cdot \vec{n}(X) = u_k^{in}(t), \forall t \in [0, T] \text{ and } X \in \partial\Omega^{in}.$$

The pressure is specified at outlet boundaries to model the outgoing of liquid or vapour in the domain. Outlet boundary conditions for the pressure give informations external to the domain: depending on the test case, the outlet pressure can be equal to the atmospheric pressure ($1 \times 10^5 Pa$) or to the pressure within a nuclear reactor under nominal working conditions ($155 \times 10^5 Pa$).

$$p(t, X) = p^{out}(t), \forall t \in [0, T] \text{ and } X \in \partial\Omega^{out}.$$

In practice, one generally uses a ghost cell formulation at the boundary. The inlet boundary condition consists in imposing a Dirichlet condition for the velocity and a Neumann condition for

the other variables (pressure, volume fractions, enthalpies). For Neumann boundary conditions, we prescribe the gradient normal to the boundary of a variable at the boundary. Usually, we take the state of the ghost cell equal to the internal state.

**Wall boundary conditions**

This is commonly known as no-slip boundary condition. It specifies the conditions for velocity components at the wall. The normal component is set to zero since the wall is static:

$$\vec{u}_k(t,X)\cdot\vec{n}(X)=0,\forall t\in[0,T] \text{ and } X\in\partial\Omega^{wall}.$$

The tangential component is set to the velocity of the wall:

$$\vec{u}_k(t,X)\cdot\vec{t}(X)=u_k^{wall}(t),\forall t\in[0,T] \text{ and } X\in\partial\Omega^{wall}.$$

Heat transfer through the wall can be specified or set to zero in the case of adiabatic walls.

**Periodic boundary conditions**

Periodic boundary conditions are often used to simulate a large system by modeling a small part that is far from its edge. These conditions consist in enforcing a relation of the form:

$$\vec{u}_k(t,X)=\vec{u}_k(t,X'),$$

For $X$ and $X'$ in $\partial\Omega$. For instance, if $\Omega=[0,L]$ in a 1D case, the condition would read:

$$\vec{u}_k(t,0)=\vec{u}_k(t,L),\quad \forall t\in[0,T].$$

The initial condition depends on the situation under consideration. In the analysis of reactor cores, what one wishes in the end is to understand the connection between a stationary state and some transient state. In practise, there are two ways of initialising a simulation:

- First one can start with constant values of the unknowns. However this can set the system in a state that is very far from the stationnary state. Furthermore the transient dynamics can be very different from the one observed when one studies nuclear accident.
- The second option is the one used in thermalhydraulics (in particular in the Cathare code). It consists in computing first a stationnary condition in a normal regime using the values of the boundary condition then changing the source term and the boundary condition to reflect the occurence of a nuclear accident.

## 1.2 Hyperbolicity of the model

When considering equal pressure law and neglecting the interfacial pressure term and the virtual mass term, the original two-fluid model (1.1) is not unconditionnally hyperbolic in the low Mach flow regime in nuclear safety studies:

$$u_r=u_v-u_l\ll c_{sm},\ \text{with the mixture sound speed: } c_{sm}=\sqrt{\frac{\alpha_v\rho_v+\alpha_l\rho_l}{\alpha_v\rho_l c_v+\alpha_l\rho_v c_l}},$$

where $c_v, c_l$ are the phasic sound speeds.
Taking into account interfacial pressure default or the virtual mass the system becomes hyperbolic

in the flow regime of interest. The reader is referred to [111, 20, 113, 9, 114, 90] for details. In the sequel, we only study the effect of the interfacial pressure correction on the hyperbolicity of the model.
In this section, we first give the Taylor expansion of the spectrum for the compressible six-equation model in 1.2.1. The complex expression of the eigenvalues makes difficult a rigorous analysis of the compressible model, hence we turn our attention to the study of the low Mach regime in 1.2.2. In section 1.2.2, we show the effect of the interfacial pressure term in the incompressible case and give closure laws that ensure the hyperbolicity of the incompressible model. In section 1.2.3, we list some closure laws for the interfacial pressure term that ensure the hyperbolicity of the compressible model. In section 1.2.4, we summarise the analysis of the Riemann problem for two incompressible phases. This analysis also takes into account the case of a vanishing phase where one of the volume fraction $\alpha_k$ goes to zero. In the last section 1.2.5, we show the singularity that appears in the case of a vanishing phase and the difficulty to ensure the positivity of the volume fractions $\alpha_k$.

### 1.2.1 Eigenstructure of the compressible six-equation model

In [91], the author gives the spectral properties of the three-dimensional two-fluid model: six real eigenvalues are trivially computed and the remaining eigenvalues are the roots of a 4-th degree polynomial that is exactly the characteristic polynomial of the one dimensional isentropic two-fluid system:

$$\begin{cases} \dfrac{\partial \alpha_v \rho_v}{\partial t} & + & \dfrac{\partial \alpha_v \rho_v u_v}{\partial x} = 0, \\ \dfrac{\partial \alpha_l \rho_l}{\partial t} & + & \dfrac{\partial \alpha_l \rho_l u_l}{\partial x} = 0, \\ \dfrac{\partial \alpha_v \rho_v u_v}{\partial t} & + & \dfrac{\partial \alpha_v \rho_v u_v^2}{\partial x} + \alpha_v \dfrac{\partial p}{\partial x} + \Delta p \dfrac{\partial \alpha_v}{\partial x} = 0, \\ \dfrac{\partial \alpha_l \rho_l u_l}{\partial t} & + & \dfrac{\partial \alpha_l \rho_l u_l^2}{\partial x} + \alpha_l \dfrac{\partial p}{\partial x} + \Delta p \dfrac{\partial \alpha_l}{\partial x} = 0. \end{cases} \tag{1.2}$$

Denoting the unknown variable $U = (\alpha_v \rho_v, \alpha_v \rho_v u_v, \alpha_l \rho_l, \alpha_l \rho_l u_l)$, the system (1.2) can be rewritten as the quasi-linear form:

$$\frac{\partial U}{\partial t} + A(U) \frac{\partial U}{\partial x} = 0$$

where $A(U)$ is the Jacobian matrix of system (1.2).
For practical purposes, one usually does not find the exact solution of the fourth degree characteristic polynomial. Instead, following the works in [113] and [42], the authors suggest using a perturbation method to compute approximate eigenvalues.
Denoting the perturbation parameter:

$$\epsilon = \frac{u_v - u_l}{c_{sm}} = \frac{u_r}{c_{sm}}$$

where $c_{sm}$ is an approximate mixture sound speed defined by:

$$c_{sm} = \sqrt{\frac{\alpha_v \rho_v + \alpha_l \rho_l}{\alpha_v \rho_l c_v + \alpha_l \rho_v c_l}}$$

The analysis of the eigenvalues for the six-equation two-fluid model is approximately made around the mechanical equilibrium, i.e. the eigenvalues are computed as a perturbation of the relative velocity in comparison to the mixture sound speed, ($\epsilon = \dfrac{u_v - u_l}{c_{sm}}$). The first order approximation

of the compressibel two-fluid system eigenvalues gives:

$$\begin{cases} \lambda_{1,4} &= \dfrac{\alpha_v\rho_l u_v + \alpha_l\rho_v u_l}{\alpha_v\rho_l + \alpha_l\rho_v} \mp c_{sm} + O(\epsilon^2), \text{ pressure or acoustic waves} \\ \lambda_{2,3} &= \dfrac{\alpha_v\rho_l u_l + \alpha_l\rho_v u_v}{\alpha_v\rho_l + \alpha_l\rho_v} \mp \sqrt{\dfrac{1}{\alpha_v\rho_l + \alpha_l\rho_v}\left(\Delta p - \dfrac{\alpha_v\alpha_l\rho_v\rho_l}{\alpha_v\rho_l + \alpha_l\rho_v}u_r^2\right)} + O(\epsilon^2), \text{ void waves} \end{cases} \tag{1.3}$$

We see from (1.3) that for small relative velocities, the two eigenvalues $\lambda_{1,4}$ are always real and have the order of magnitude of the mixture speed of sound $c_{sm}$. In contrast, if $\Delta p < \dfrac{\alpha_v\alpha_l\rho_v\rho_l}{\alpha_v\rho_l + \alpha_l\rho_v}u_r^2$, the two other eigenvalues $\lambda_{2,3}$ are complex. We will discuss the influence of the interfacial pressure term on the hyperbolicity of the system in the next section. Assuming that the four eigenvalues are real, we can see that the eigenvalues may easily change sign. The two-fluid model displays a complicated eigenstructure and its spectrum is more complex than the one of for single-phase flows.

For single-phase flows, both mathematical theory and numerical methods of the Euler system have been studying well by numerous authors in the litterature. This system is a system of hyperbolic conservation laws with two Genuinely Non Linear fields and one Linearly Degenerate field (in one dimension). Even if the two-fluid models inherit achievements obtained in the single phase flow including of modeling, mathematical theory and numerical methods, however, the two-phase flow models possess many of specific difficulties due to existence of two phases in the same domain of interest and their interactions as well. We will see in section 1.2.4 that for the incompressible limit of system (1.2) the study of the spectrum proves that the eigenvalues are not a priori ordered and that the characteristic fields are neither GNL nor LD.

### 1.2.2 Hyperbolicity of the incompressible model

From the system (1.3), we see that there is a critical $\Delta p$ which ensures the positivity of the value under the square root. In this section, we show the influence of the interfacial pressure term on the hyperbolicity of the two-fluid model in the incompressible case.

To understand the role of the interfacial pressure term to get the hyperbolicity of the system, we show the eigenstructure of a reduced system where both phases are assumed incompressible in (1.2), derived in [91].

$$\frac{\partial}{\partial t}\begin{pmatrix}\tilde{\rho}\\ \tilde{\rho u}\end{pmatrix} + A_{red}\frac{\partial}{\partial x}\begin{pmatrix}\tilde{\rho}\\ \tilde{\rho u}\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}, \text{ where: } \begin{pmatrix}\tilde{\rho}\\ \tilde{\rho u}\end{pmatrix} = \begin{pmatrix}\alpha_v\rho_l + \alpha_l\rho_v\\ \rho_v u_v - \rho_l u_l\end{pmatrix}, \tag{1.4}$$

with

$$A_{red} = \begin{pmatrix} \dfrac{\alpha_l\rho_v u_v + \alpha_v\rho_l u_l}{\alpha_v\rho_l + \alpha_l\rho_v} & -\dfrac{\alpha_v\alpha_l(\rho_v - \rho_l)}{\alpha_v\rho_l + \alpha_l\rho_v} \\ \dfrac{\rho_v\rho_l(u_v - u_l)^2}{(\rho_v - \rho_l)(\alpha_v\rho_l + \alpha_l\rho_v)} - \dfrac{\Delta p}{\alpha_v\alpha_l(\rho_v - \rho_l)} & \dfrac{\alpha_l\rho_v u_v + \alpha_v\rho_l u_l}{\alpha_v\rho_l + \alpha_l\rho_v} \end{pmatrix},$$

and its eigenvalues are:

$$\lambda_{inc} = \frac{\alpha_l\rho_v u_v + \alpha_v\rho_l u_l}{\alpha_v\rho_l + \alpha_l\rho_v} \pm \sqrt{\frac{1}{\alpha_v\rho_l + \alpha_l\rho_v}\left(\Delta p - \frac{\alpha_v\alpha_l\rho_v\rho_l(u_v - u_l)^2}{\alpha_v\rho_l + \alpha_l\rho_v}\right)}.$$

The incompressible limit of the two-fluid model is hyperbolic provided:

$$\Delta p \geq \frac{\alpha_v\alpha_l\rho_v\rho_l(u_v - u_l)^2}{\alpha_v\rho_l + \alpha_l\rho_v} \tag{1.5}$$

This type of coefficient is used in the industrial code Cathare (see [20]) to make the compressible two-fluid model hyperbolic.

If we remove the interfacial pressure term by setting $\Delta p = 0$, the matrix $A_{red}$ of system (1.4) becomes:

$$A_{red} = \begin{pmatrix} \dfrac{\alpha_l\rho_v u_v + \alpha_v\rho_l u_l}{\alpha_v\rho_l + \alpha_l\rho_v} & -\dfrac{\alpha_v\alpha_l(\rho_v - \rho_l)}{\alpha_v\rho_l + \alpha_l\rho_v} \\ \dfrac{\rho_v\rho_l(u_v - u_l)^2}{(\rho_v - \rho_l)(\alpha_v\rho_l + \alpha_l\rho_v)} & \dfrac{\alpha_l\rho_v u_v + \alpha_v\rho_l u_l}{\alpha_v\rho_l + \alpha_l\rho_v} \end{pmatrix} \tag{1.6}$$

In (1.6), $A_{red}$ is a $2 \times 2$ matrix with two diagonal coefficients that are equal. Such a matrix has real eigenvalues provided the extradiagonal terms have the same sign. The efficiency of the parameter $\Delta p$ in making the system hyperbolic comes from the fact that it removes the component: $\dfrac{\rho_v\rho_l(u_v - u_l)^2}{(\rho_v - \rho_l)(\alpha_v\rho_l + \alpha_l\rho_v)}$ that gives rise to complex eigenvalues, as we can see in (1.4). Thus to make the system hyperbolic, another possibility is to cancel this component by assuming an algebraic pressure disequilibrium. Without the assumption of pressure equality, the evolution equation of $\tilde{\rho u}$ would be:

$$\frac{\partial\tilde{\rho u}}{\partial t} + \frac{\partial}{\partial x}\frac{(\tilde{\rho u})^2}{2(\rho_v - \rho_l)} + \frac{\partial}{\partial x}\left(p_v - p_l - \frac{\rho_v\rho_l(u_v - u_l)^2}{(\rho_v - \rho_l)(\alpha_v\rho_l + \alpha_l\rho_v)}\right) = 0$$

An alternative to the use of an interfacial pressure correction $\Delta p$ is the assumption that there is an algebraic pressure disequilibrium having the form:

$$p_v - p_l = \frac{\rho_v\rho_l(u_v - u_l)^2}{(\rho_v - \rho_l)(\alpha_v\rho_l + \alpha_l\rho_v)} \tag{1.7}$$

Hence, one can use a pressure disequilibrium of the form (1.7) or a parameter $\Delta p$ of the form (1.5) to ensure the hyperbolicity of the isentropic two-fluid model for two incompressible phases. In [93], the authors prove the existence and uniqueness of an admissible solution to the Riemann problem for the isentropic two-fluid model for two incompressible phases with a pressure disequilibrium of the form (1.7). In section 1.2.4, we detail the properties of this model and the results obtained in [93].

### 1.2.3 Hyperbolicity of the compressible model

In this section, we give some closures laws for the isentropic two-fluid model (1.2) in the compressible case that ensure the hyperbolicity of the system for a range of relative velocities $u_r$.
For small relative velocities $(u_v - u_l)$, closure laws were proposed in [111] and [20] that ensure the hyperbolicity of the system (1.2) with the following form:

$$\Delta p = \delta\frac{\alpha_v\alpha_l\rho_v\rho_l(u_v - u_l)^2}{\alpha_v\rho_l + \alpha_l\rho_v}, \text{ with: } \delta > 1$$

In the more general case of large relative velocities, the hyperbolicity of the isentropic system (1.2) has been studied in [9], [90], [91], [115] with:

$$\Delta p = \frac{\alpha_v\alpha_l\rho_v\rho_l}{\alpha_v\rho_l + \alpha_l\rho_v}(u_v - u_l)^2 + \frac{1}{c_v^2}\left(\rho_v - \frac{\alpha_v\alpha_l\rho_v\rho_l}{\alpha_v\rho_l + \alpha_l\rho_v}\right)(u_v - u_l)^4.$$

This closure law guarantees the hyperbolicity for the relative velocities up to the sound speed of vapour phase $c_v$.
On the other hand, the authors in [9, 91] introduces the interfacial of the form:

$$\Delta p = \rho_v(u_v - u_l)^2.$$

This closure law guarantees the hyperbolicity in the region where $|u_v - u_l| \leq c_v$.

In the next section, we will present some recent existence results obtained on a system for two incompressible phases.

### 1.2.4 The Riemann problem for two incompressible phases

In [93], the authors study a $2 \times 2$ system of conservation laws that models the dynamic of two incompressible phases. They establish their model from the following system:

$$\begin{cases} \partial_t \alpha_v \rho_v + \partial_x(\alpha_v \rho_v) = 0 \\ \partial_t \alpha_l \rho_l + \partial_x(\alpha_l \rho_l) = 0 \\ \partial_t(\alpha_v \rho_v u_v) + \partial_x(\alpha_v \rho_v u_v^2) + \alpha_v \partial_x p_v = \alpha_v \rho_v g \\ \partial_t(\alpha_l \rho_l u_l) + \partial_x(\alpha_l \rho_l u_l^2) + \alpha_l \partial_x p_l = \alpha_l \rho_l g \end{cases} \tag{1.8}$$

where the non zero pressure difference $p_v - p_l$ takes the form (1.7).
Assuming in (1.8) that both phases are incompressible, the resulting system focuses on the study of the void waves that determine the composition of the mixture:

$$\begin{cases} \partial_t \alpha + \partial_x \left( \dfrac{\alpha(1-\alpha)\omega}{\alpha(\rho_l - \rho_v) + \rho_v} \right) = 0 \\ \partial_t \omega + \partial_x \left( \dfrac{\omega^2}{2(\rho_v - \rho_l)} \right) = (\rho_v - \rho_l) g \end{cases} \tag{1.9}$$

We introduce the unknown variable vector $U = \begin{pmatrix} \alpha \\ \omega \end{pmatrix}$, and the space of admissible states $H = \{(\alpha, \omega), \alpha \in [0,1], \omega \in \mathbb{R}\}$. We also denote the sets $H_+$ and $H_-$ such that:

$$H_\pm = \{(\alpha, \omega), \omega \in \mathbb{R}_\pm \setminus \{0\}, \alpha \in (0,1)\}$$

Many specificities arise from this incompressible model:

- the two eigenvalues $\lambda_1$ and $\lambda_2$ are not a priori ordered:

$$\lambda_1 = \frac{\omega}{\rho_v - \rho_l} \left( 1 - \frac{\rho_v \rho_l}{(\alpha(\rho_l - \rho_v) + \rho_v)^2} \right), \quad \lambda_2 = \frac{\omega}{\rho_v - \rho_l}$$

- the characteristic fields associated to $\lambda_1$ and $\lambda_2$ are genuinely nonlinear in each domain $H_+$ and $H_-$ but are neither genuinely nonlinear nor linearly degenerate in general:

$$\vec{\nabla}\lambda_1 \cdot \vec{r}_1 = \frac{-2\rho_v \rho_l \omega}{(\alpha(\rho_l - \rho_v) + \rho_v)^3}, \quad \vec{\nabla}\lambda_2 \cdot \vec{r}_2 = \frac{\rho_v \rho_l \omega}{(\rho_v - \rho_l)}$$

  where $r_1, r_2$ are the eigenvectors of $\nabla F$ associated to $\lambda_1, \lambda_2$.

- the system is strictly hyperbolic in $H_\pm$ and in general weakly hyperbolic on the domain $H$

Considering the Riemann problem for the conservative system (1.9) in the case $g = 0$ with a piecewise constant initial data:

$$U_0(x) = \begin{cases} U_L(\alpha_L, \omega_L) & \text{if } x \leq 0 \\ U_R(\alpha_R, \omega_R) & \text{if } x > 0 \end{cases}$$

the authors prove the existence and uniqueness of an admissible solution, satisfying the Liu criterion.

Due to the complex structure of the shock and rarefaction curves, numerous cases are studied. In some cases, the Riemann problem does not give rise to classical weak solutions made of two waves of different families separated by an intermediate state $U_I(\alpha_I, \omega_I)$. In the sequel, four examples of the non classical solutions to the Riemann problem are given in order to illustrate the originality of this system that describes the void waves of the two-fluid model:

- solution made of two rarefactions of the same family, the 2-family
- solution made of three waves connected by two intermediate states $U^\star$ and $U^{\star\star}$
- solution made of three shocks. In this case, a pure phase ($\alpha$) is observed and the velocity of the vanishing phase does not necessarily equal the one of the remaining phase
- solution made of two waves of different families where one is a non classical shock wave. This shock wave is called non classical because a left state which is on a branch of a hyperbola that goes out of the domain $H$ is connected to a right state through an intermediate state $U_I$ located on the other branch of the hyperbola where it comes back in $H$ (see Figure 3a in [93]).

### 1.2.5 Vanishing phase

In the two-fluid model, the total boiling or condensation of one phase will arise a singularity. The absent phase is called vanishing phase or ghost phase. It poses a difficulty in the two-fluid model owing to its independent velocities. The singularity arises when one computes the absent phase velocity using the conservative variables $u_k = \dfrac{\alpha_k\rho_k u_k}{\alpha_k\rho_k}$ as $\alpha_k\rho_k \to 0$.
Studying the one dimensional two-fluid model, we are interested in the mathematical properties of the vanishing phase which is assumed to be vapour: $\alpha = \alpha_v = 0$. Let us take into account the isentropic model (1.2), when $\alpha = 0$, the Jacobian matrix becomes:

$$A_{\alpha=0} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -u_v^2 & 0 & 2u_v & 0 \\ \dfrac{\rho_l c_l^2}{\rho_v} & c_l^2 - u_l^2 & 0 & 2u_l \end{pmatrix}$$

which has four real eigenvalues:

$$u_v, u_v, u_l + c_l, u_l - c_l$$

but the matrix is not diagonalizable because there are only three eigenvectors:

$$\vec{v}_{u_l \pm c_l} = {}^t\left(0, 0, 1, u_l \pm c_l\right), \quad \vec{v}_{u_v} = {}^t\left(\frac{\rho_v}{\rho_l}\left(\frac{(u_v - u_l)^2}{c_l^2} - 1\right), u_v\frac{\rho_v}{\rho_l}\left(\frac{(u_v - u_l)^2}{c_l^2} - 1\right), 1, u_v\right)$$

The hyperbolicity of system (1.2) is then broken for $\alpha_v = 0$.
One of the objectives of [93] was to prove that the positiveness of the volume fractions $\alpha_k$ is inherent to the model. At the discrete level, it is not always the case that a numerical method capture positive volume fraction. In the thermalhydraulic platform Cathare, an interfacial friction term is used to ensure the positiveness of the volume fractions. We explain here the role of the

interfacial friction term in ensuring that the volume fractions $\alpha_k(t) \in [0,1], \forall t \in [0,T]$. We consider in the sequel that the unknowns are smooth enough:

$$\begin{cases} \partial_t u_v + u_v \partial_x u_v + \dfrac{\partial_x p}{\rho_v} + \dfrac{\Delta p}{\alpha_v \rho_v} \partial_x \alpha_v = \dfrac{C_D}{\rho_v} ||u_v - u_l|| (u_v - u_l) \\ \partial_t u_l + u_l \partial_x u_l + \dfrac{\partial_x p}{\rho_l} + \dfrac{\Delta p}{\alpha_l \rho_l} \partial_x \alpha_l = \dfrac{C_D}{\rho_l} ||u_l - u_v|| (u_l - u_v) \end{cases} \tag{1.10}$$

We now write the momentum equation for the relative velocity $u_r = u_v - u_l$:

$$\partial_t (u_v - u_l) + u_v \partial_x u_v - u_l \partial_x u_l + \left( \frac{1}{\rho_v} - \frac{1}{\rho_l} \right) \partial_x p = -C_D \left( \frac{1}{\rho_v} + \frac{1}{\rho_l} \right) ||u_v - u_l|| (u_v - u_l)$$

Since the variables $u_v$, $u_l$, $p$ and their derivatives are assumed bounded, when the drag coefficient $C_D$ goes to infinity, the relative velocity $u_r$ verifies the ODE:

$$\partial_t u_r = -C_D ||u_v - u_l|| (u_v - u_l)$$

Hence, the system tends to an equilibrium of the velocities with $u_r \to 0$. The numerical difficulty of the vanishing phase is thus handled by the Cathare code by imposing the equality of the two phasic velocities through an interfacial friction term that dominates the momentum equations when one of the volume fraction goes to 0.
Assuming that the vapour phase is the vanishing phase and neglecting the interfacial friction coefficient $C_D$, the corresponding momentum equation in (1.10) shows that the velocity of the vanishing phase follows a Burgers equation and does not have to be equal to the velocity of the remaining phase.

## 1.3 Discretisation of the two-fluid model

In general, there exists two families of numerical methods for the simulation of two-phase flows. Firstly, colocated schemes are generally used on unstructured meshes where the unknowns are located in the same place (cell-centered). In the litterature, many authors developed Riemann solvers (either Godunov-type methods or Roe-type schemes or Osher schemes or AUSM schemes) for the simulation of two-phase flows dealing with the numerical challenges encountered: vanishing phase [32, 31, 88, 107, 8, 42, 93], non conservative products [88, 87, 42, 113] and stiff source terms [93, 94]. We can also mention the VFFC scheme ([56, 57]) that has a generic formulation by contrast with the Roe scheme ([100]) that is applied under some algebraic conditions on the system. This category of schemes is robust but present a lack of accuracy for low Mach number/almost incompressible flows. Corrections are proposed in [36, 37] to overcome this issue but generate an instability with checker-board type oscillations. We can also mention the pressure-based methods with colocated variables such as the one used in the platform Neptune-CFD (developed by EDF and CEA, [10]). This type of schemes does not suffer from a lack of precision for low Mach number flows but show checker-board type oscillations. A way to overcome this issue is to use Rhie and Chow type corrections to avoid spurious oscillations, [43].
On the other hand, staggered schemes are used on structured meshes with unknows located either on edges or cell centers. This category of schemes has a good behaviour for almost incompressible flows. The space discretisation in the Cathare code is based on a staggered scheme. The main drawback of this family of schemes is the handling of complex geometries since its use is limited to structured meshes. In this section, we first introduce the family of staggered schemes as initially

proposed in [63, 62] (see section 1.3.1.1). We also discuss the properties of these numerical methods depending on the discretisation of the convection term for the isentropic Euler system (see section 1.3.1.2). In section 1.3.1.3, we present a family of staggered schemes derived with a new approach that discretises the conservative form of the Euler system. In section 1.3.2, we detail the discretisation of the one dimensional Cathare model to compare it with the existing litterature on staggered schemes. In the last section, we explain some numerical challenges of two-phase flows simulation by addressing:

- strategies of discretisation for non conservative products occuring in two-fluid models (see section 1.3.3.1),
- simulations made with the Cathare scheme on test cases showing vanishing phase (see section 1.3.3.2,
- strategies of discretisation for discontinuous source terms occuring in two-fluid models (see section 1.3.3.3).

### 1.3.1 An introduction to ICE schemes

We refer here to the numerical methods designed on staggered grids to simulate compressible flows at low Mach number. The velocity unknowns are located at cell interfaces whilst the density and pressure unknowns are located at cell centers. The expression of the products $\rho u$, $\rho \partial_t u$ and $\rho u^2$ then raises an issue since velocity and density are located in different places that is adressed in a different way by different authors.

Historically the Marker and Cell numerical scheme (MAC) [64] was designed for incompressible flows. Then the Implicit Continuous-fluid Eulerian (ICE) [63, 62] method was designed for compressible flows at low Mach numbers as well as high Mach numbers. The historical ICE method [63, 62] discretises the conservative Navier-Stokes equations in a way that reduces to the MAC method for incompressible flows. In the seminal papers [63, 62], the velocity unknowns are first eliminated then the pressure unknowns are determined in an iterative process and the velocity unknowns follow. For the velocity elimination to be valid the momentum fluxes are treated in a semi-explicit way (see section 1.3.1.1 for more details).

The ICE method encountered a considerable success with numerous variants (explicit/implicit, with/without prediction correction steps) and became popular in the thermal hydraulics community [97, 98]. There the Navier-Stokes equations are discretised in non conservative form, which makes the velocity elimination easier. The mass flux is upwinded for better stability but the approach still consists in eliminating the velocity unknowns in order to first retrieve the pressure (see 1.3.1.2 for more details). However for the elimination to be rigourously valid the momentum flux should be entirely explicit, which yields a restriction on the time step.

Herbin et al ([65], [66]) proposed an approach that does not rely on velocity elimination and enables full implicitation. They were able to derive rigourous proofs of stability (see section 1.3.1.3 for more details).

We will consider in the sequel the 1D isentropic Euler equations in conservative form

$$\begin{cases} \partial_t \rho + \partial_x q = 0 \\ \partial_t q + \partial_x \dfrac{q^2}{\rho} + \partial_x p = 0. \end{cases} \tag{1.11}$$

and in non conservative form

$$\left\{ \begin{array}{rcl} \partial_t \rho & + & \partial_x(\rho u) = 0 \\ \rho \partial_t u & + & \frac{1}{2}\rho \partial_x u^2 + \partial_x p = 0 \end{array} \right. , \tag{1.12}$$

We will give the expression of each scheme on the Euler equations which gives a first insight into the schemes. The discretisation of the momentum convective term $u\dfrac{\partial u}{\partial x}$ should verify the conservation of $h = p + \dfrac{1}{2}\rho u^2$. Some non conservative schemes discretise the form $\dfrac{1}{2}\partial_x u^2$ and satisfy the Bernoulli principle. Other non conservative schemes discretise the form $u\partial_x u$ and do not naturally recover the Bernoulli principle at the discrete level. In the sequel, we give the expression of several schemes proposed in the litterature in the single-phase case and the extension to two-phase flows can be found in references therein.

#### 1.3.1.1 The original scheme of Harlow and Amsden

We summarise here the numerical method for all flow speeds described in the seminal article [63]. In [63], the stability analysis used the heuristic approach of [70] by estimating the numerical diffusion of the scheme and tuning the numerical viscosity to make it positive. A mass diffusion coefficient $\tau$ and artificial viscosity coefficients $\lambda$ and $\mu$ are considered for example in [62]. For simplicity of the exposure, we neglect the various artificial viscosity terms, and present the scheme only in 1D. The discrete 1D isentropic Euler equations take the form

$$\frac{\rho_i^{n+1} - \rho_i^n}{\triangle t} + \frac{(\rho u)_{i+\frac{1}{2}}^{n+1} - (\rho u)_{i-\frac{1}{2}}^{n+1}}{\triangle x} = 0 \tag{1.13}$$

$$\frac{(\rho u)_{i+\frac{1}{2}}^{n+1} - (\rho u)_{i+\frac{1}{2}}^{n}}{\triangle t} + \frac{\rho_{i+1}^{n+1}(u^2)_{i+1}^n - \rho_i^{n+1}(u^2)_i^n}{\triangle x} + \frac{p_{i+1}^{n+1} - p_i^{n+1}}{\triangle x} = 0 \tag{1.14}$$

The expression of the cell centered velocity $u_i^2$ required in the momentum equation raises an issue of interpolation between face and cell that is adressed in different ways by different authors. There are at least four historical types of interpolation formula for $u_i$ (see [63] page 207) :

- Centered: $u_i^2 = \left( \dfrac{u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}}}{2} \right)^2$
- ZIP: $u_i^2 = u_{i-\frac{1}{2}} u_{i+\frac{1}{2}}$
- Partial Donor: $u_i^2 = \left\{ \begin{array}{ll} u_{i-\frac{1}{2}} \dfrac{u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}}}{2} & \text{if} \quad u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}} > 0 \\ u_{i+\frac{1}{2}} \dfrac{u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}}}{2} & \text{if} \quad u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}} < 0 \end{array} \right.$
- Complete Donor: $u_i^2 = \left\{ \begin{array}{ll} u_{i-\frac{1}{2}}^2 & \text{if} \quad u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}} > 0 \\ u_{i+\frac{1}{2}}^2 & \text{if} \quad u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}} < 0 \end{array} \right.$

In the next section, we list some choices of discretisation for the convection term and discuss the properties of the resulting schemes.

#### 1.3.1.2 The traditional non conservative discretisation

Other forms of the ICE method used in the thermal hydraulics community ([98] section 11.2, Cathare, Sabena [97]) discretise the non conservative version (1.12) of the Euler equations. The schemes take the generic form

$$\left\{\begin{array}{lcl} \dfrac{\rho_i^{n+1}-\rho_i^n}{\triangle t} & + & \dfrac{1}{\triangle x}(\rho^{up}_{i+\frac{1}{2}}u^{n+1}_{i+\frac{1}{2}}-\rho^{up}_{i-\frac{1}{2}}u^{n+1}_{i-\frac{1}{2}})=0 \\ \bar{\rho}_{i+\frac{1}{2}}\dfrac{u^{n+1}_{i+\frac{1}{2}}-u^n_{i+\frac{1}{2}}}{\triangle t} & + & \dfrac{1}{2}(\rho\partial_x u^2)_{i+\frac{1}{2}}+\dfrac{1}{\triangle x}(p^{n+1}_{i+1}-p^{n+1}_i)=0 \end{array}\right. . \tag{1.15}$$

with

$$\bar{\rho}_{i+\frac{1}{2}} = \frac{\rho_i+\rho_{i+1}}{2}$$

For stability reasons, the mass flux $\rho u$ at the cell interfaces is generally defined using an upwind density $\rho^{up}$ (see [98] section 11.2) defined on faces as :

$$\rho^{up}_{i+\frac{1}{2}} = \left\{\begin{array}{ll} \rho_i & \text{if} \quad u_{i+\frac{1}{2}}>0 \\ \rho_{i+1} & \text{if} \quad u_{i+\frac{1}{2}}\leq 0 \end{array}\right.$$

Here are some examples of expressions for the momentum convection term:

- semi implicit discretisation (Cathare 3D module section 11.5 page 339 of [74], [25])
  In this case the density is explicit in the mass flux $\rho u$, in the velocity evolution terms $\rho\partial_t u$ and the convection term $\rho u\partial_x u$. The scheme takes the form

$$\left\{\begin{array}{lcl} \dfrac{\rho_i^{n+1}-\rho_i^n}{\triangle t} & + & \dfrac{1}{\triangle x}(\rho^{up,n}_{i+\frac{1}{2}}u^{n+1}_{i+\frac{1}{2}}-\rho^{up,n}_{i-\frac{1}{2}}u^{n+1}_{i-\frac{1}{2}})=0 \\ \bar{\rho}^n_{i+\frac{1}{2}}\dfrac{u^{n+1}_{i+\frac{1}{2}}-u^n_{i+\frac{1}{2}}}{\triangle t} & + & \dfrac{1}{2}\rho^{up,n}_{i+\frac{1}{2}}(\partial_x u^2)_{i+\frac{1}{2}}+\dfrac{1}{\triangle x}(p^{n+1}_{i+1}-p^{n+1}_i)=0 \end{array}\right. , \tag{1.16}$$

  with

$$(\partial_x u^2)_{i+\frac{1}{2}} = \left\{\begin{array}{ll} \dfrac{1}{\triangle x_{i+\frac{1}{2}}}((u^2)^{n+1}_{i+\frac{1}{2}}-(u^2)^n_{i-\frac{1}{2}}) & \text{if} \quad u_{i+\frac{1}{2}}>0 \\ \dfrac{1}{\triangle x_{i+\frac{1}{2}}}((u^2)^{n+1}_{i+\frac{3}{2}}-(u^2)^n_{i+\frac{1}{2}}) & \text{if} \quad u_{i+\frac{1}{2}}\leq 0 \end{array}\right. \tag{1.17}$$

  The Bernouilli principle is recovered at the discrete level in the case of stationary incompressible non viscous flows. Note that in 2D/3D, the density is upwinded in axial momentum terms of the form $\rho u^2$ but centered in non axial terms of the form $\rho uv$.
  The treatment in (1.17) allows for a stronger implicitation of the velocity and larger time steps whilst retaining the elimination of the velocity unknowns. However this approach introduces a time consistency error as we detail in the following.

  We apply a standard truncation error analysis to the discrete momentum equation in (1.16) assuming constant densities. This corresponds to the term: $\dfrac{\partial u}{\partial t}+u\dfrac{\partial u}{\partial x}$, in the case $u_{i+\frac{1}{2}}>0$:

$$\frac{u(x_{i+1/2},t_{n+1})-u(x_{i+1/2},t_n)}{\Delta t}+\frac{1}{2}\frac{u^2(x_{i+1/2},t_{n+1})-u^2(x_{i-1/2},t_n)}{\Delta x}$$

$$= \frac{\partial u}{\partial t}(x_{i+1/2},t_n) + \mathcal{O}(\Delta t) + \frac{1}{2}\frac{\partial}{\partial x}u^2(x_{i+1/2},t_n) + \mathcal{O}(\Delta x)$$
$$+\frac{\mathbf{\Delta t}}{\mathbf{\Delta x}}\left(\mathbf{u(x_{i+1/2},t_n)}\frac{\partial \mathbf{u}}{\partial \mathbf{t}}\mathbf{(x_{i+1/2},t_n)} + \mathcal{O}(\mathbf{\Delta t})\right)$$

When the time step $\Delta t$ and the mesh size $\Delta x$ goes to zero, the consistency error tends to $\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x}$ with an additional term in bold that prevent the scheme to be consistent with the momentum equation.

- fully implicit discretisation (Cathare 1D module section 11.3 page 320 of [74], [26])
  In this case the density is treated implicitly in the discrete mass flux $\rho u$, in the discrete evolution terms $\rho\partial_t u$ and in the discrete convection term $\rho u\partial_x u$. The scheme takes the generic form

$$\left\{\begin{array}{lcl} \dfrac{\rho_i^{n+1}-\rho_i^n}{\triangle t} & + & \dfrac{1}{\triangle x}(\rho_{i+\frac{1}{2}}^{up,n+1}u_{i+\frac{1}{2}}^{n+1} - \rho_{i-\frac{1}{2}}^{up,n+1}u_{i-\frac{1}{2}}^{n+1}) = 0 \\ \bar{\rho}_{i+\frac{1}{2}}^{n+1}\dfrac{u_{i+\frac{1}{2}}^{n+1}-u_{i+\frac{1}{2}}^{n}}{\triangle t} & + & \bar{\rho}_{i+\frac{1}{2}}^{n+1}(u\partial_x u)_{i+\frac{1}{2}} + \dfrac{1}{\triangle x}(p_{i+1}^{n+1}-p_i^{n+1}) = 0 \end{array}\right. ,$$

  with

$$(u\partial_x u)_{i+\frac{1}{2}} = \left\{\begin{array}{ll} \dfrac{1}{\triangle x_{i+\frac{1}{2}}}u_{i+\frac{1}{2}}^{n+1}(u_{i+\frac{1}{2}}^{n+1}-u_{i-\frac{1}{2}}^{n+1}) & \text{if} \quad u_{i+\frac{1}{2}}^{n+1} > 0 \\ \dfrac{1}{\triangle x_{i+\frac{1}{2}}}u_{i+\frac{1}{2}}^{n+1}(u_{i+\frac{3}{2}}^{n+1}-u_{i+\frac{1}{2}}^{n+1}) & \text{if} \quad u_{i+\frac{1}{2}}^{n+1} \leq 0 \end{array}\right.$$

  This treatment allows for a stronger implicitation of the velocity and larger time steps, whilst retaining the elimination of the velocity unknowns. The Bernouilli principle is not recovered at the discrete level in the case of stationary incompressible non viscous flows.

- explicit discretisation (Sabena [97])
  In this case the density is explicit in the mass flux $\rho u$, in the velocity evolution terms $\rho\partial_t u$ and the convection term $\rho u\partial_x u$. The scheme takes the form

$$\left\{\begin{array}{lcl} \dfrac{\rho_i^{n+1}-\rho_i^n}{\triangle t} & + & \dfrac{1}{\triangle x}(\rho_{i+\frac{1}{2}}^{up,n}u_{i+\frac{1}{2}}^{n+1} - \rho_{i-\frac{1}{2}}^{up,n}u_{i-\frac{1}{2}}^{n+1}) = 0 \\ \bar{\rho}_{i+\frac{1}{2}}^{n}\dfrac{u_{i+\frac{1}{2}}^{n+1}-u_{i+\frac{1}{2}}^{n}}{\triangle t} & + & \bar{\rho}_{i+\frac{1}{2}}^{n}(\dfrac{1}{2}\partial_x u^2)_{i+\frac{1}{2}}^{n} + \dfrac{1}{\triangle x}(p_{i+1}^{n+1}-p_i^{n+1}) = 0 \end{array}\right. , \tag{1.18}$$

  with

$$\frac{1}{2}(\rho\partial_x u^2)_{i+\frac{1}{2}} = \left\{\begin{array}{ll} \dfrac{1}{2}\rho_i^n\dfrac{1}{\triangle x_{i+\frac{1}{2}}}((u^2)_{i+\frac{1}{2}}^n - (u^2)_{i-\frac{1}{2}}^n) & \text{if} \quad u_{i+\frac{1}{2}} > 0 \\ \dfrac{1}{2}\rho_{i+1}^n\dfrac{1}{\triangle x_{i+\frac{1}{2}}}((u^2)_{i+\frac{3}{2}}^n - (u^2)_{i+\frac{1}{2}}^n) & \text{if} \quad u_{i+\frac{1}{2}} \leq 0 \end{array}\right.$$

  The Bernouilli principle is recovered at the discrete level in the case of stationary incompressible non viscous flows.
  The explicit treatment introduces constraints on the time step.

- explicit convection ([98] section 11.2.1, equation 11.16)
  In this particular case the density is explicit in the mass flux $\rho u$, in the velocity evolution terms $\rho\partial_t u$ and the convection term $\rho u\partial_x u$. The scheme takes the form

$$\left\{\begin{array}{lcl} \dfrac{\rho_i^{n+1}-\rho_i^n}{\triangle t} & + & \dfrac{1}{\triangle x}(\rho^{up,n}_{i+\frac{1}{2}}u^{n+1}_{i+\frac{1}{2}}-\rho^{up,n}_{i-\frac{1}{2}}u^{n+1}_{i-\frac{1}{2}})=0 \\ \bar\rho^n_{i+\frac{1}{2}}\dfrac{u^{n+1}_{i+\frac{1}{2}}-u^n_{i+\frac{1}{2}}}{\triangle t} & + & \bar\rho^n_{i+\frac{1}{2}}(u\partial_x u)^n_{i+\frac{1}{2}}+\dfrac{1}{\triangle x}(p^{n+1}_{i+1}-p^{n+1}_i)=0 \end{array}\right. ,$$

  with

$$(\rho u\partial_x u)_{i+\frac{1}{2}} = \left\{\begin{array}{ll} \dfrac{1}{2}\dfrac{\rho_i^n+\rho_{i+1}^n}{2}\dfrac{1}{\triangle x_{i+\frac{1}{2}}}u^n_{i+\frac{1}{2}}(u^n_{i+\frac{1}{2}}-u^n_{i-\frac{1}{2}}) & \text{if} \quad u^n_{i+\frac{1}{2}}>0 \\ \dfrac{1}{2}\dfrac{\rho_i^n+\rho_{i+1}^n}{2}\dfrac{1}{\triangle x_{i+\frac{1}{2}}}u^n_{i+\frac{1}{2}}(u^n_{i+\frac{3}{2}}-u^n_{i+\frac{1}{2}}) & \text{if} \quad u^n_{i+\frac{1}{2}}\leq 0 \end{array}\right.$$

  With this treatment the velocity unknowns are easily eliminated using the momentum equation. This numerical scheme differs from the previous explicit discretisation (1.18) in the treatment of the convection term. In (1.18), the form $\dfrac{1}{2}\partial_x u^2$ is used and in the present scheme the form $u\partial_x u$ is discretised. The Bernouilli principle is not recovered at the discrete level in the case of stationary incompressible non viscous flows.

In most of the methods the velocity unknowns are first eliminated, and the resulting system is solved in a way that is compatible with the incompressible regime. The drawback of this approach is that there are constraints on the discretisation of the momentum flux $\rho\vec{u}\otimes\vec{u}$ and of the viscous terms $\mu\triangle\vec{u}$ for the elimination of the velocity unknowns to be possible. The elimination can take place rigorously speaking when the convective flux $\rho\vec{u}\otimes\vec{u}$ and the viscous terms $\mu\triangle\vec{u}$ are explicit in time. However explicit discretisations yield time step limitations.

#### 1.3.1.3 The recent scheme of Herbin, Latché et al

In the past decade, Herbin, Latché and their coauthors have proposed a new approach with rigorous proofs of stability. They discretise the conservative form of the Euler equations (equation 1.11) with a conservative scheme. Their approach does not rely on velocity elimination and thus explicit and implicit variants are possible.
The different variants include one step ([65] section 2.1, [66] section 3.1) and prediction/correction steps ([65] section 2.2, [66] section 4.1) variants, fully implicit ([66] section 3, [65] section 2.1), semi implicit and almost explicit [66] (all but the pressure gradient are explicit-in-time) variants.

For simplicity we present the discrete equation of the fully implicit variant ([66] section 3, [65] section 2.1) for the 1D isentropic Euler equations in conservative form:

$$\frac{\rho_i^{n+1}-\rho_i^n}{\triangle t} + \frac{\rho^{up,n+1}_{i+\frac{1}{2}}u^{n+1}_{i+\frac{1}{2}}-\rho^{up,n+1}_{i-\frac{1}{2}}u^{n+1}_{i-\frac{1}{2}}}{\triangle x}=0 \tag{1.19}$$

$$\frac{\bar\rho^{n+1}_{i+\frac{1}{2}}u^{n+1}_{i+\frac{1}{2}}-\bar\rho^n_{i+\frac{1}{2}}u^n_{i+\frac{1}{2}}}{\triangle t} + \frac{\overline{\rho u}^{n+1}_{i+1}u^{up,n+1}_{i+1}-\overline{\rho u}^{n+1}_i u^{up,n+1}_i}{\triangle x}+\frac{p^{n+1}_{i+1}-p^{n+1}_i}{\triangle x}=0. \tag{1.20}$$

The pressure $p_i$ and the density $\rho_i$ are located at the cell centers whereas the velocity $u_{i+\frac{1}{2}}$ are located at the cell interfaces. The expression of the products $\rho u$, $\rho\partial_t u$ and $\rho u^2$ between the

velocity located at cell interfaces and the density located at cell centers thus has to be defined through interpolation formula.

The mass flux $\rho u$ at the cell interfaces is defined using an upwind density $\rho^{up}_{i+\frac{1}{2}}$ defined as :

$$\begin{aligned}\rho^{up}_{i+\frac{1}{2}} &= \begin{cases} \rho_i & \text{if} \quad u_{i+\frac{1}{2}} > 0 \\ \rho_{i+1} & \text{if} \quad u_{i+\frac{1}{2}} \leq 0 \end{cases} \\ &= \frac{\rho_i + \rho_{i+1}}{2} + sign(u_{i+\frac{1}{2}})\frac{\rho_i - \rho_{i+1}}{2}, \end{aligned} \tag{1.21}$$

which is the sum of a centered and an upwind terms.
The expression of $\bar{\rho}_{i+\frac{1}{2}}$ in the discrete momentum equation accounts for an average of the neighbouring densities

$$\bar{\rho}_{i+\frac{1}{2}} = \frac{1}{2}(\rho_i + \rho_{i+1}). \tag{1.22}$$

The expression of $\overline{\rho u}$ in the discrete momentum equation is

$$\overline{\rho u}_i = \frac{1}{2}(\rho^{up}_{i-\frac{1}{2}} u_{i-\frac{1}{2}} + \rho^{up}_{i+\frac{1}{2}} u_{i+\frac{1}{2}}). \tag{1.23}$$

The upwind velocity $u^{up}_i$ at cell centers is defined as :

$$\begin{aligned} u^{up}_i &= \begin{cases} u_{i-\frac{1}{2}} & \text{if} \quad \overline{\rho u}_i > 0 \\ u_{i+\frac{1}{2}} & \text{if} \quad \overline{\rho u}_i \leq 0 \end{cases} \\ &= \frac{u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}}}{2} + sign(\overline{\rho u}_i)\frac{u_{i-\frac{1}{2}} - u_{i+\frac{1}{2}}}{2}, \end{aligned} \tag{1.24}$$

which is the sum of a centered and an upwind terms.
It is possible to use a centered velocity $\bar{u}$ instead of the upwind velocity $u^{up}$ (see [66]).

We presented the conservative staggered schemes studied in [65], [66] that is proven to be entropic and to satisfy a kinetic energy preservation. In the next section, we introduce the one dimensional Cathare scheme in the two-phase flows configuration.

### 1.3.2 The one dimensional Cathare scheme

In this section, we first give some details on the one dimensional two-fluid model before giving the discrete equations of Cathare. We presented a generic two-fluid model in the section 1.1 with the system 1.1. In the sequel, we write the one dimensional Cathare model which is a variable cross section area model.

$$\begin{cases} \dfrac{\partial A\alpha_k\rho_k}{\partial t} + \dfrac{\partial A\alpha_k\rho_k u_k}{\partial x} & = \Gamma_k \\ A\alpha_k\rho_k\left[\dfrac{\partial u_k}{\partial t} + u_k\dfrac{\partial u_k}{\partial x}\right] + A\alpha_k\dfrac{\partial p}{\partial x} & = A\alpha_k\rho_k g + F^{int}_k + F^w_k + A\sigma^M_k u^{int} \\ A\dfrac{\partial}{\partial t}\left[\alpha_k\rho_k\left(H_k + \dfrac{u_k^2}{2}\right)\right] + \dfrac{\partial}{\partial x}\left[A\alpha_k\rho_k u_k\left(H_k + \dfrac{u_k^2}{2}\right)\right] & = A\alpha_k\dfrac{\partial p}{\partial t} + A\alpha_k\rho_k u_k g + \Gamma_k\left(H^{int}_k + \dfrac{(u^{int}_k)^2}{2}\right) \\ & + Q^w_k + \sigma^Q_k \end{cases} \tag{1.25}$$

where $A$ is the cross section area of the channel.
In the sequel, we specify the discretisation of some terms of the system (1.25):

- In the mass balance laws, the void fraction $(\alpha_k^{up})_{i+1/2}$ and density $(\rho_k^{up})_{i+1/2}$ at the cell edge $i+1/2$ for the term:

$$\left(\frac{\partial A\alpha_k\rho_k u_k}{\partial x}\right)_{i+1/2}$$

are determined with an upwinding approach. $(\alpha_k^{up})_{i+1/2}$ is upwinded according to the velocity $(u_k)_{i+1/2}$ and the density $(\rho_k^{up})_{i+1/2}$ according to the quantity $(\alpha_k^{up})_{i+1/2}(u_k)_{i+1/2}$,

$$\begin{array}{rcl} (\alpha_k^{up})_{i+1/2} & = & \left\{ \begin{array}{ll} (\alpha_k)_i - \alpha_k^{inf}, & (u_k)_{i+1/2} > 0 \\ (\alpha_k)_{i+1} - \alpha_k^{inf}, & (u_k)_{i+1/2} < 0 \end{array} \right. \\ (\rho_k^{up})_{i+1/2} & = & \left\{ \begin{array}{ll} (\rho_k)_i, & (\alpha_k^{up})_{i+1/2}(u_k)_{i+1/2} > 0 \\ (\rho_k)_{i+1}, & (\alpha_k^{up})_{i+1/2}(u_k)_{i+1/2} < 0 \end{array} \right. \end{array} \tag{1.26}$$

where $\alpha_k^{inf}$ are constants for the minimal values of $\alpha_k$. The Cathare code considers that the volume fraction of each phase admit a non-zero minimum value and therefore, also a maximum value very slightly lower than one. It is therefore accepted that a flow is never completely single-phase, the residual phase represents respectively bubbles or tiny drops in a practically single-phase liquid or vapour zone.

- The mass and energy balance equations in the one dimensional Cathare model (1.25) are written in a conservative form, whereas the momentum balance equation is written in a non conservative form.
To evaluate $(\bar{\rho}_k)_{i+1/2}$, $(\bar{\alpha}_k)_{i+1/2}$ the cell-centered values at the edge $i+1/2$ for the term:

$$A\alpha_k\rho_k\left[\frac{\partial u_k}{\partial t} + u_k\frac{\partial u_k}{\partial x}\right]_{i+1/2},$$

an average of the adjacent values is taken weighted by the cell volumes.

$$(\bar{\rho}_k)_{i+1/2} = \frac{Vol_i\rho_i + Vol_{i+1}\rho_{i+1}}{Vol_i + Vol_{i+1}}$$

- The convection term:

$$\left(u_k\frac{\partial u_k}{\partial x}\right)_{i+1/2},$$

is determined with an upwinding approach,

$$\left(u_k\frac{\partial u_k}{\partial x}\right)_{i+1/2} = \left\{ \begin{array}{ll} (u_k)_{i+1/2}((u_k)_{i+1/2} - (u_k)_{i-1/2}), & (u_k)_{i+1/2} > 0 \\ (u_k)_{i+1/2}((u_k)_{i+3/2} - (u_k)_{i+1/2}), & (u_k)_{i+1/2} < 0 \end{array} \right.$$

The value of the cross section area is carefully chosen in the term

$$A\alpha_k\rho_k u_k\frac{\partial u_k}{\partial x}$$

to ensure the Bernoulli principle for single phase flows with varying cross section.

$$A_{Bern,i+1/2} = \left\{ \begin{array}{ll} \dfrac{\bar{A}_{i+1/2}}{2}\dfrac{A_{i+1/2} + A_{i-1/2}}{A_{i-1/2}}, & (u_k)_{i+1/2} > 0 \\ \dfrac{\bar{A}_{i+1/2}}{2}\dfrac{A_{i+1/2} + A_{i+3/2}}{A_{i+3/2}}, & (u_k)_{i+1/2} < 0 \end{array} \right.$$

with $\bar{A}_{i+1/2} = \dfrac{Vol_i + Vol_{i+1}}{2\Delta x}$

- In the energy balance laws, the velocity $(\bar{u}_k)_i^{n+1}$ at the cell $i$ for the time derivative term:

$$\frac{\partial}{\partial t}\left[\alpha_k\rho_k\left(H_k+\frac{u_k^2}{2}\right)\right]_i=\frac{(\alpha_k)_i^{n+1}(\rho_k)_i^{n+1}\left((H_k)_i^{n+1}+\frac{1}{2}((\bar{u}_k)_i^{n+1})^2\right)-(\alpha_k)_i^n(\rho_k)_i^n\left((H_k)_i^n+\frac{1}{2}((\bar{u}_k)_i^n)^2\right)}{\Delta t}$$

  is computed with an interpolation using the adjacent velocity at edges and the void fractions. In this interpolation, the void fractions are explicit in time

$$(\bar{u}_k)_i^{n+1}=\frac{((\alpha_k)_i^n+(\alpha_k)_{i-1}^n)A_{i-1/2}(u_k)_{i-1/2}^{n+1}+((\alpha_k)_i^n+(\alpha_k)_{i+1}^n)A_{i+1/2}(u_k)_{i+1/2}^{n+1}}{((\alpha_k)_i^n+(\alpha_k)_{i-1}^n)A_{i-1/2}+((\alpha_k)_i^n+(\alpha_k)_{i+1}^n)A_{i+1/2}} \tag{1.27}$$

  In the time derivative term of the energy equations the approximation involves the terms $(\bar{u}_k)_i^{n+1}$ and $(\bar{u}_k)_i^n$. From the interpolation formula (1.27), we see that the evaluation of $(\bar{u}_k)_i^{n+1}$ is made with the adjacent volume fractions at time $T^n$ and with the volume fractions at time $T^{n-1}$ for the term $(\bar{u}_k)_i^n$. Hence, the approximation of the time derivative at time $T^{n+1}$ in the energy equations is a function of the unknowns at times $T^{n+1}$, $T^n$ and $T^{n-1}$. From this reason, the Cathare time scheme is a two-step time scheme. This point will raise additional difficulties in the application of the time parallel algorithm, the parareal method. This requires an adaptation of the parareal algorithm to this type of multi-step time schemes that should not be intrusive in the Cathare code. We will detail this aspect in the next chapter.

- All the terms are implicit in time in the system (1.25) except for the interpolated velocity at nodes in the energy equations.

### 1.3.3 Difficulties of two-phase flow models

Even if two-fluid models inherit achievements obtained in the single-phase flow modeling, mathematical theory and numerical methods, however, they possess many specific difficulties due to the existence of two phases in the same domain and their interactions. In this section we will discuss some difficulties in general existing in the two-phase flow models such as the presence of non conservative products, the configuration of the vanishing phase and the handling of discontinuous source terms. We illustrate the mathematical challenge in discretising non conservative products on the term $\alpha_k\dfrac{\partial p}{\partial x}$ appearing in the the momentum equations of the six-equation two-fluid model (1.1) (see section 1.3.3.1). Then we comment in section 1.3.3.2 how the Cathare code handles numerically vanishing phases. Finally in section 1.3.3.3, we present the challenges coming from the discretisation of stiff source terms.

#### 1.3.3.1 Non conservative products in the two-fluid model

A theory of hyperbolic conservation laws, studied in depth in the literature, can be found in [106, 76, 77, 75, 21, 24, 33, 38, 58]. Such a theory gives a fundamental understanding and main ideas for plenty of numerical methods to solve a hyperbolic system of conservation laws, i.e. find a weak solution in the sense of distributions. However, our system (1.1) possesses non conservative products and is not therefore a conservative system. A discontinuous solution would lead to the product of two distributions that is not well-defined. In order to study the weak solutions of a non conservative hyperbolic system, one may consider different approaches.

In general, the most popular approach to deal with non conservative products is the theory of non conservative hyperbolic systems studied by Dal Maso et al in [84]. In the classical theory of

conservation laws, a shock wave depends merely on its left state and right state, the definition of Dal Maso et al depends on the choice of a specific path which connects a left state to a right state around a shock wave for non conservative products. Defining an appropriate path requires realistic physical information which is not easy, especially in a complicated two-fluid model. Moreover, once the appropriate path is chosen, different numerical methods may converge to different solutions, see comments in [4] and references therein.
Therefore, one may prefer to solve the two-fluid model by using a simpler consideration of the non conservative product. For example, in [113], the authors rewrite the 1D two-fluid model (1.2) and choose jump conditions based on a particular case where the system has a conservative form. The resulting two-fluid model includes two conservation laws of mass of each phase, one conservation law of mixture of momentum and one equation for the liquid velocity assumed incompressible. More precisely, it consists in the following equations:

$$\frac{\partial \alpha_v \rho_v}{\partial t} + \frac{\partial \alpha_v \rho_v u_v}{\partial x} = 0 \tag{1.28}$$

$$\frac{\partial \alpha_l \rho_l}{\partial t} + \frac{\partial \alpha_l \rho_l u_l}{\partial x} = 0 \tag{1.29}$$

$$\frac{\partial \alpha_v \rho_v u_v + \alpha_l \rho_l u_l}{\partial t} + \frac{\partial \alpha_v \rho_v u_v^2 + \alpha_l \rho_l u_l^2 + p}{\partial x} = 0 \tag{1.30}$$

$$\frac{\partial u_l}{\partial t} + \frac{\partial}{\partial x}\left(\frac{u_l^2}{2} + \frac{p}{\rho_l}\right) = 0. \tag{1.31}$$

In [113], they propose a formula to locally linearize the term $\alpha_k$ in the product $\alpha_k \dfrac{\partial p}{\partial x}$ around a discontinuity. This local linearisation denoted $\tilde{\alpha}_k$ is chosen such that the original isentropic model (1.2) (neglecting $\Delta p \partial_x \alpha_k$) and the system (1.28-1.31) have the same Rankine-Hugoniot relation. After calculations, [113] finds:

$$\tilde{\alpha}_l = \frac{2\alpha_l^L \alpha_l^R}{\alpha_l^L + \alpha_l^R}, \quad \tilde{\alpha}_v = 1 - \tilde{\alpha}_l.$$

This formula is then applied to the simulation of two-fluid model, [114]. It is important to note that although $\Delta p$ was neglected in the identification of jump conditions for the systems (1.2) and (1.28-1.31), its contribution is fundamental in practice to obtain real characteristic waves.
On the other hand the non conservative term in the energy equations is written as a spacial derivative in [113] using the assumption of incompressible liquiq together with the liquid mass equation:

$$\partial_t \alpha_l = -\partial_x(\alpha_l u_l).$$

Other authors focusing on the numerical methods usually neglect the product of $p\partial_t \alpha_k$ in the energy equations, see for example [88, 87].

#### 1.3.3.2 Cathare treatment of the vanishing phase

**An elementary test case: the oscillating manometer:**

In this section, we illustrate the role of the interfacial friction term on a two-phase test case where one of the phases disappears. The equations used are the simplified Cathare equations, (1.25). Transfers between phases (mass and heat) and wall friction are neglected in the system (1.25). The only source terms we consider in this test case are the gravity force and the interfacial forces $F_k^{int}$.

In [69], the oscillating manometer is proposed as a numerical benchmark test for system codes to test the ability of each numerical sheme to preserve system mass and to retain the gas-liquid interface. It consists in a U-shaped tube manometer which is connected at the top, so that a closed system is formed. The system contains initially gas and liquid with the liquid forming

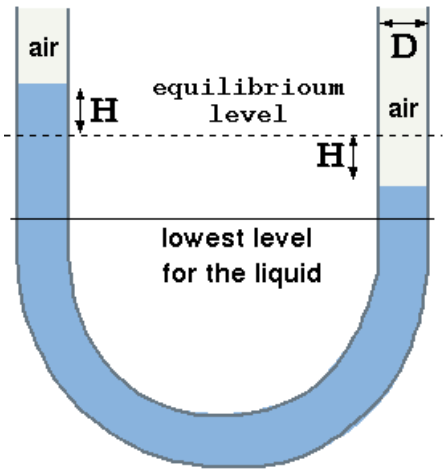


Figure 1.1: Oscillating manometer test case

equal levels in each arm of the manometer. Further, all parts of the fluid system have initially a uniform non zero velocity, but zero acceleration. Under these initial conditions, a hydrostatic pressure hypothesis is made throughout the system. Also, the system is isothermal at 50°C with $10^5$ Pa pressure at the vapour-liquid interfaces. Distance in the direction of the flow is measured by $x$ in meters. The length of the manometer is 20 m and the diameter is $D = 1$ m. The initial position of the vapour-liquid interface is 5m from the bottom of each manometer leg and the fluid initially has a velocity of $u_0 = 2.1$ m/s. This initial velocity will cause the interface to oscillate approximately $\pm 1.5$ m in height from the initial location.

We seek to show the influence of the interfacial friction term on the behaviour of the scheme when one of the phases disappears in some parts of the domain. Previously in section (1.2.5), we saw that the interfacial friction coefficient $C_D$ goes to infinity in the configuration of a vanishing phase, as a consequence the system tends to an equilibrium of the velocities. The expression of the interfacial friction term depends on the flow regime (bubbly, annular, dispersed,...) and on the geometry:

$$C_D = f(\alpha_k, \rho_k, \sigma, \mu_k, D_h)$$

where: $\sigma$ is the surface tension, $D_h$ the hydraulic diameter and $\mu_k$ are liquid and vapour dynamical viscosities.

The equation of motion of the vapour-liquid interface in each leg of the manometer is the following:

$$\begin{aligned} &\frac{d^2x}{dt^2} + \frac{2gx}{L} = 0 \\ &x(t=0) = 0, \quad \frac{dx}{dt}(t=0) = u_0 \end{aligned} .$$

The solution to this equation is: $x(t) = 2u_0\sqrt{\frac{L}{2g}}sin\left(\sqrt{\frac{2g}{L}}t\right)$ and the liquid velocity is :

$$u_L(t) = u_0 cos\left(\sqrt{\frac{2g}{L}}t\right), \tag{1.32}$$

where $g$ designates the acceleration of gravity and L designates the water length equal to 12 m.

For these simulations, the variation versus time of the liquid velocity at the bottom of the manometer (at one edge) is plotted. In Figure 1.2, we compare the reference solution of the

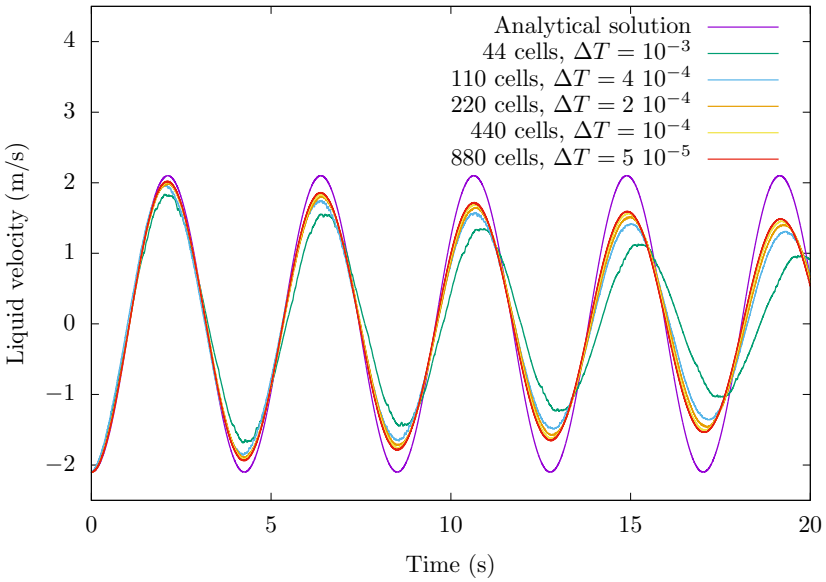


Figure 1.2: Convergence of the 1D Cathare scheme

oscillating manometer (1.32) with several numerical solutions computed with the one dimensionnal Cathare scheme. These results show that the numerical solution converge to another solution that is damped. This can be explained by the treatment of the vanishing phase used in Cathare. When the interfacial friction term dominates the system (section 1.2.5), it behaves as if a diffusion term was present in the model that damped the solution and prevent the numerical solution to converge towards the reference solution (1.32) of the test case.

### An industrial test case: the Water-Packing

The treatment of the vanishing phase is a challenge for every software that simulates two-phase flows. In the context of nuclear safety studies, a well known problem linked to vanishing phases is the Water-Packing. It occurs during the simulation of a water level rise due to condensation as well as in other more complex situations. The typical situation in which this problem was studied is a vertical tube initially filled with superheated steam, connected to a steam tank at its top, is gradually filled with very cold liquid from below. As the liquid front progresses upwards, the vapour condenses and the resulting local depression aspirates steam from the tank and the liquid front gradually heats up with the condensation. Provided you know the rate of phase change by condensation, the analytical solution to this problem is simple with regard to the pressure at

the bottom of the tube. Assuming that the front rises slowly, the problem is quasi-static and the pressure at the bottom of the tube then increases regularly due to the weight of the liquid column which increases by the addition of liquid injection and condensation. On the other hand, if one wishes to simulate this flow numerically, one runs up against difficulties: the field of physical pressure is disturbed by parasitic waves of strong amplitudes. This is the phenomenon of Water-Packing. The simulation of this kind of stratified flow is of great interest. In fact this type of situation is encountered in a more complex form in accidental transient calculations (for example, the filling phase of the reactor core after a large breach). The calculation of the water level rise in a simple situation is therefore used to analyse and solve a problem occurring in much more complex situations. This problem occurs in all the codes of thermalhydraulics dedicated to the simulation of accidental transients in the circuits of a nuclear reactor. It has been the subject of numerous studies and it is used to evaluate codes in test batteries called benchmarks. The Water-Packing benchmark was formalized by V. H. Ransom in 1987, in [68], under the title "expulsion of steam by cold water" (see Figure 1.3). To understand the origin of this phenomenon, we recall how the Cathare code

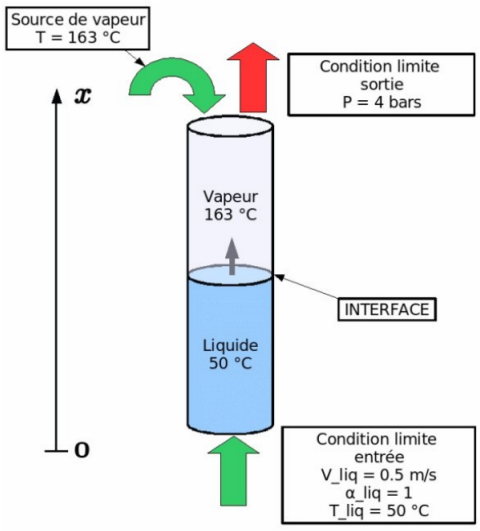


Figure 1.3: Setting of the Water-Packing benchmark in [68]

handles vanishing phase. The principle consists in considering that the volume fraction of each phase admit a non-zero minimum value and therefore, also a maximum value very slightly lower than one. It is therefore accepted that a flow (or a flow zone) is never completely single-phase, the residual phase represents respectively bubbles or tiny drops in a practically single-phase liquid or vapour zone. So even when a single-phase flow (or flow zone) is simulated, all the equations are solved. Also the mechanical equilibrium is forced: the residual phase has the same velocity as the dominant phase, consequence of an interfacial friction made artificially very large (section 1.2.5). This choice was made in the Cathare code since its initiation and has been valid for a large number of use cases. However it raises problems in the representative test case of Water-Packing. Physically, this choice amounts to say that the residual bubbles (resp. drops) in the liquid (resp. vapour) are entrained by the dominant liquid phase (resp. vapour).

In the context of test case (1.3), the liquid front is rising thus the liquid velocity is positive below the interface. The steam condenses at the interface with cold liquid, thereby creating an intake of additional steam from the top of the tube. The vapour velocity is negative when above the interface. The residual liquid velocity in the vapour phase is then negative. Therefore, when

a cell close to the interface is filled up with liquid, liquid velocity suddenly changes direction. A vector node located above the interface at a negative liquid velocity will become positive when it is joined by the interface. Artificial pressure spikes are then observed whenever the liquid level fills up a cell (see Figure 1.5, red curve).

Here is an explanatory diagram of the behaviour of the liquid velocity during the advance of the liquid front(see Figure 1.4).

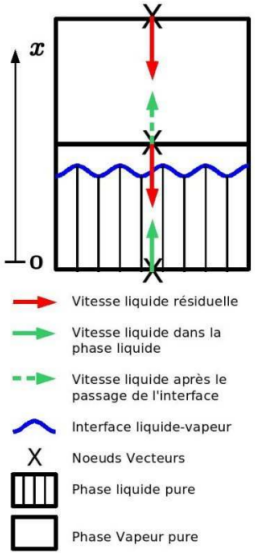


Figure 1.4: Behaviour of the liquid velocity during the Water-Packing phenomenon, from [86]

Currently, the Cathare code uses an Anti-Water-Packing correction, specially dedicated to the treatment of this problem. However, it comes with a large number of shortcomings with respect to the balance laws and returns sometimes erroneous values. The principle consists in locating the cell(s) in which the interface is located, then greatly reduce the rate of condensation. As a consequence, the pressure shows artificial spikes with lower amplitude (see Figure 1.5, blue curve) because by decreasing the rate of condensation, the velocity at which the vapour is aspirated decreases too and the velocity of the residual liquid phase just above the interface decreases too. The inversion of the liquid velocity still takes place but it goes from a weakly negative velocity to a stronger positive one. In summary, the inversion of the liquid velocity occurs more gradually. The upwinding approach in the Cathare numerical scheme plays an important role in solving this problem. Indeed, during the simulation of a counter-current flow, the upwinding strategy can be inconsistent with the dynamics we want to capture. Each momentum equation is upwinded according to the sign of its phasic velocity (reference section schema de Cathare). This approach is consistent when bot phasic velocities have the same sign but may fail to capture the void waves when the phasic velocities have opposite signs. In [93], the authors propose colocated schemes to capture void waves without forcing a mechanical equilibrium with a large interfacial friction. This scheme is based on upwinding according to the volume fraction wave speed that is different from the phasic velocities.

#### 1.3.3.3 Discontinuous source terms

In the thermalhydraulics of nuclear reactors, two-fluid models display stiff source terms $S(U, x)$. The stiffness of these source terms has different origins. First, the heat source $\Phi$ is localised on the

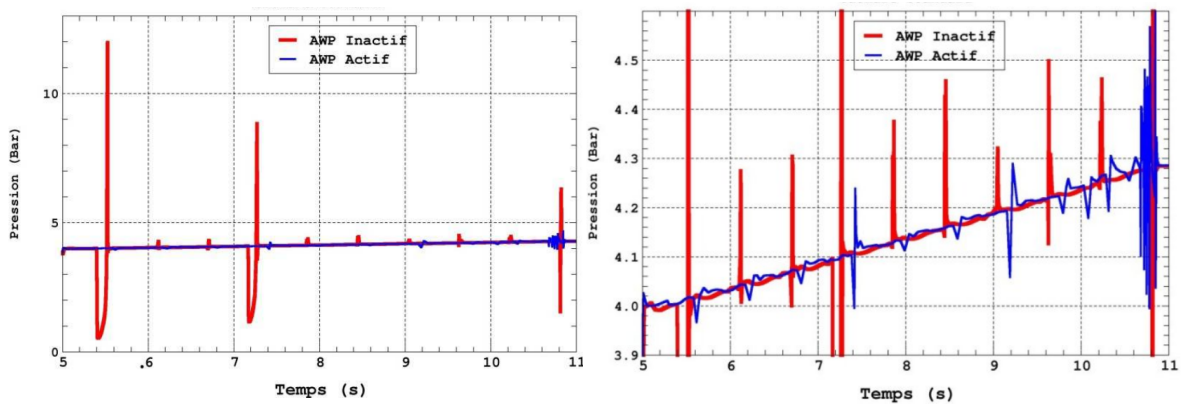


Figure 1.5: Pressures spikes with or without Cathare anti-Water-Packing correction, from [86]

core of the reactor which yields a discontinuity in space. Moreover, the dry-out of the Uranium rod when the temperature reaches a threshold (the critical heat flux) yields a discontinuity of $\Phi$ as a function of the temperature. Secondly, the boiling of the fluid is a stiff phenomena with a threshold that is the boiling temperature (or enthalpy). For these reasons the simulation of boiling of two-phase flows is challenging both from a mathematical and a numerical point of view. The source term $S(U,x)$ being discontinuous both in $U$ and $x$ makes it impossible to use Cauchy- Lipschitz type of theorems for the existence of solutions even for ODEs. However, there are particular cases where a unique solution may exist (see [23]). The source term $S(U,x)$ being discontinuous in $U$ makes numerical approximation more difficult in the numerical simulation because of the stiffness of the solution. Classical approaches to deal with stiff source term assume it is Lipschitz in the variable $U$.

In this section, we introduce non-homogeneous hyperbolic systems of conservation laws, i.e. we take into account a non zero source term $S$ as below

$$\frac{\partial U}{\partial t} + \partial_x F(U) = S(U,x), \quad x \in \mathbb{R}, t \geq 0, \tag{1.33}$$

The source term $S$ is usually a function of the unknown vector $U$ and spacial variable $x$, $S(U,x)$. A simple approach to solve the non homogeneous system of conservation laws is to include a source term in the right hand side as follows:

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} + \frac{F_{i+1/2} - F_{i-1/2}}{\Delta x} = S_i^n, \tag{1.34}$$

where $S_i^n$ is an approximation of

$$S_i = \frac{1}{\Delta x}\frac{1}{\Delta t}\int_{t^n}^{t^{n+1}}\int_{x_i}^{x_{i+1}} S dx dt \tag{1.35}$$

and $F_{i\pm1/2}$ is some interfacial flux function.
The numerical scheme (1.34) is a classical one solving the non homogeneous system of conservation laws. Sometimes both the flux divergence $\partial_x F(U)$ and the source term $S(U,x)$ are discretised independently and in practice, $S_i^n$ is simply considered as the source function at the average value $U_i$, i.e. $S_i^n = S(U_i^n, x_i)$. However this approach may generate instabilities in the simulation of the

system (1.33), especially for stiff source terms functions $S$.
We are interested in the capture of the stationary regime of a two-phase flow characterised by the stationary equation:

$$\partial_x F(U) = S. \tag{1.36}$$

In some cases, for example with stiff source terms, using $S_i^n = S(U_i^n, x_i)$ in (1.35) implies the instability of the numerical solution at the stationary state. In order to improve the numerical simulation, one suggests either upwinding the source terms ([17]) or developping well-balanced schemes ([22]) in the sense that it preserves the stationary state. In a two-fluid model, due to the complexity of the flux function and the lack of regularity of source terms, it seems difficult to construct a well-balanced scheme.

## 1.4 Acceleration techniques for the simulation of two phase flows

In the numerical resolution of partial differential equations, we generally have to solve linear or nonlinear systems arising from the discretisation. The large size of these systems and the fact that they are ill conditionned make a global resolution difficult. In this section, we firstly summarise in 1.4.1 the resolution method used by the Cathare code to solve the nonlinear system arising from the discretisation of the two-fluid model. Then we present the actual acceleration methods available in the Cathare code in 1.4.2. Finally, we introduce in 1.4.3 the time parallel algorithm that we consider in this thesis, the parareal algorithm.

### 1.4.1 Solution algorithm for the two-fluid model

After discretisation in space and time of the two-fluid model, we obtain a non linear system that is solved with a Newton method. Before describing the implementation of this method, we recall here the dependance of the different equations according to the implicited variables:

- the mass and energy balance equations at cell $i$ depend on the pressure, void fraction and enthalpies defined at cell $i$ and on the velocities defined at the edges.
- the momentum balance equations at edge $i+1/2$ depend on the velocities defined at edges $i+1/2$, $i-1/2$ and $i+3/2$ and on the cell centered unknowns defined at the adjacent cells of the edge $i+1/2$.

After applying the semi-implicit time scheme of the Cathare code, we obtain a system of the following form:

$$\frac{U^{n+1}-U^n}{\Delta t} + A(U^{n+1}, U^n) = S(U^n)$$

This non linear system is solved by a Newton method:

$$\begin{array}{l} \dfrac{\delta U^{k+1}}{\Delta t} + J(U^k, U^n)\delta U^{k+1} = \tilde{S}(U^n, U^k), \text{ where: } \delta U^{k+1} = U^{k+1} - U^k \\ \text{and : } U^{k+1} = (p^{k+1}, \alpha_v^{k+1}, H_l^{k+1}, H_v^{k+1}, u_l^{k+1}, u_v^{k+1}) \end{array} \tag{1.37}$$

The increments of the principal variables in the Cathare code are denoted $\delta U^{k+1}$ and the terms of the Jacobian matrix $J(U^k, U^n)$ are computed as follows:

$$J_{i,j} = \frac{\partial E_i}{U_j^k}$$

Since the Cathare time discretisation is semi-implicit, this term corresponds to the derivative of the balance equation $E_i$ with respect to the principal variable $U_j^k$ that is implicited, for the $k$-th Newton iteration. Hence, the Cathare code uses the derivatives with respect to the variables $(p, \alpha_v, H_l, H_v, u_l, u_v)$ instead of the conservative variables $(\alpha_v\rho_v, \alpha_l\rho_l, u_v, u_l, \alpha_v\rho_v E_v, \alpha_l\rho_l E_l)$.
The resulting linear system (2.1) at each Newton iteration is solved by a Gauss elimination to obtain a system with pressure increments only. Another specificity of the Cathare code is that it uses a semi-implicit pressure solver without splitting techniques for the velocity (no prediction/correction steps), and no explicit construction of an elliptic problem on the pressure.

In the Cathare software, the junction is an element of the geometry that links two other elements of the geometry: a 1D element (a pipe, ...) with a 3D element (vessel, ...) or a 1D/3D element with a boundary condition element (see Figure 1.6). The junctions store the informations about the edge that is common to the two elements linked by this junction (two velocities and derivatives of the momentum equations associated to this edge with respect to the principal unknowns). Hence the system depends on the $N_i$ pressure increments belonging to the internal cells and on the $N_j$ pressure increments belonging to the junctions. The unknowns associated to the junctions are called external variables. The dependance on the external pressure increments appears when we write the momentum balance equations associated to the junctions. The resulting system is of the following form:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \Delta p(N_i) \\ \Delta p(N_j) \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \tag{1.38}$$

The method adopted in the Cathare code consists in eliminating the internal pressure increments $\Delta p(N_i)$ and then solve the problem on the external pressure increments $\Delta p(N_j)$:

$$\left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right)\Delta p(N_j) = S_2 - A_{21}A_{11}^{-1}S_1$$

This linear system is then solved by a direct method with the library LAPACK BLAS.

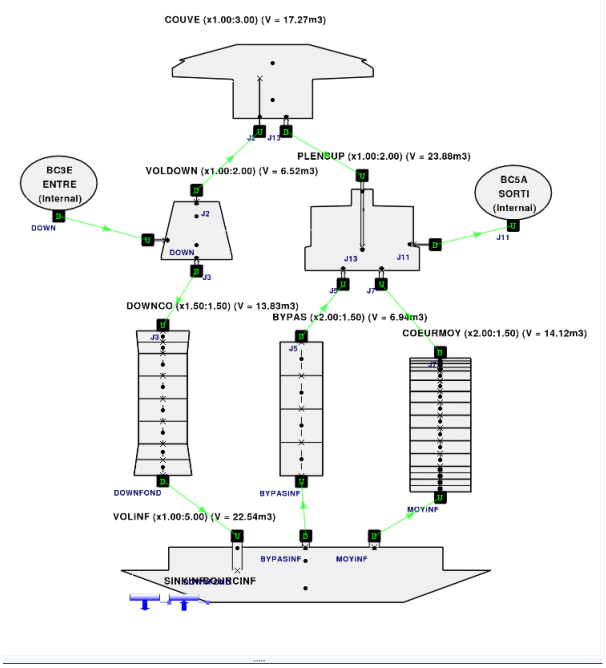


Figure 1.6: Example of a geometry in the Cathare code with elements and junctions

### 1.4.2 Actual acceleration methods in Cathare

In [101], the authors present the strategy implemented in the Cathare code in order to make its use compatible with a real-time response. This work was carried out within the context of the integration of the Cathare code in the SIPA simulator (Post-Accident Simulator) for training and engineering studies on nuclear PWR reactors under nominal conditions or accidental transients. In the sequel, we list the steps of the solution algorithm described in section 1.4.1, that were parallelised:

- Each element of the Cathare geometry (see figure 1.6) assembles a block of the Jacobian matrix in parallel. Hence, for each element, the matrix block depends on the $N_i$ pressure increments belonging to the internal cells of the element and on the $N_j$ pressure increments belonging to the junctions associated to the element (step $(i)$ in figure 1.7).
  An element of the Cathare geometry can be seen as a subdomain of a spacial domain decomposition method.

- Each element eliminates the internal variables of system (1.38) and obtains a system depending on the junction variables $\Delta p(N_j)$ only (step $(ii)$ in figure 1.7).

- The pressure increments $\Delta p(N_j)$ for all the junctions of the Cathare geometry are computed by a Gauss elimination. It is performed by an iterative algorithm that successively eliminates the blocks of the Jacobian matrix corresponding to a junction common to two elements. This process goes on, as long as the non-eliminated junctions are common to at least one other element. The order used to eliminate the block matrices is called an elimination tree. It depends on the reactor meshing and is optimised by the Cathare code for a sequential resolution (step $(iii)$ in figure 1.7).

- The increments of the other principal variables (velocities, volume fractions and enthalpies) are computed in parallel over all the elements (step $(iv)$ in figure 1.7).

In figure 1.7, we see the performances of the actual parallel method implemented in the Cathare code. We see in the left figure that the computational cost of the step $(i)$ in the solution algorithm represents 65% of the global computational time of the simulation. Hence, the parallelisation of step $(i)$ over the elements of the geometry offers good speed up performances: a speed up of 10 is obtained for the parallelisation of step $(i)$ on 12 processors where the global speed up is 6.

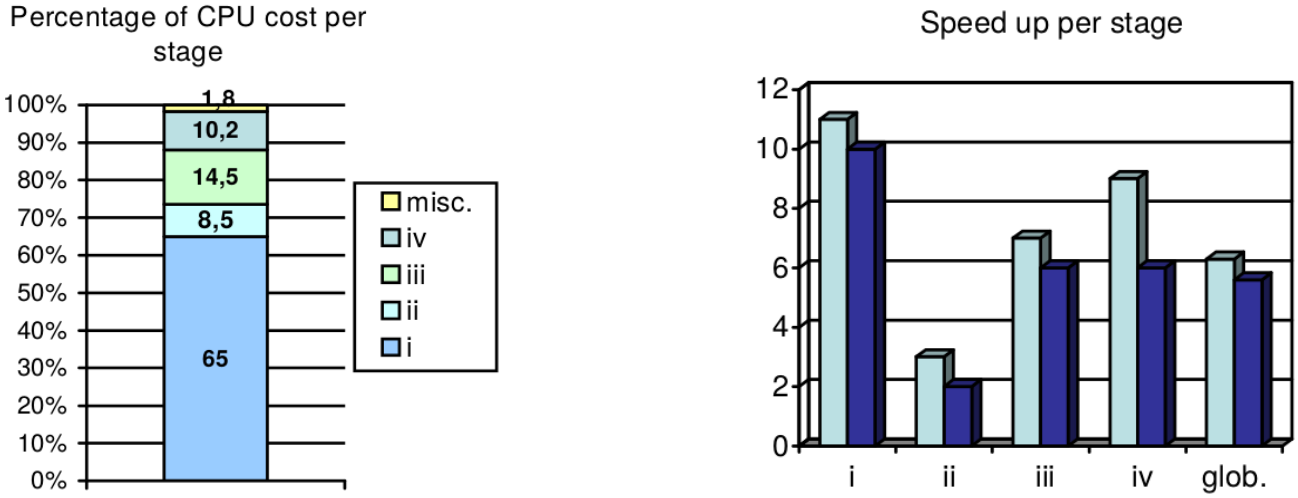


Figure 1.7: Performances with the actual parallelism in Cathare for two industrial test cases on 12 processors, [101]

This actual acceleration method has been successfully applied on industrial test cases for the simulation of a nuclear reactor under an accidental scenario. However, these parallel methods are implemented with OpenMP that allows shared memory parallelism. Their use is limited to a level of parallelism of about 20 processors or of at most all the processors of a standard desk computer. Moreover, the performances of this strategy reaches its limit when the geometry of the simulation includes 1D elements with $\sim 10^2$ cells and 3D elements with $\sim 10^3$. This imbalance of tasks between the processors can damage the speed up performances. We illustrate this behaviour in the left figure 1.8 where the computational time of one step of the Jacobian assembling is measured for one Newton iteration in one time step.
An optimisation of the parallel method that we have previously described has been proposed to overcome the issue of tasks imbalance between processors. This new algorithm allows to assign elements of the geometry to the processors with load balancing, knowing the computational time of the previous time step. In the right figure 1.8, we see that this algorithm allows to significantly improve the load balancing between processors and thus the performances of the parallel method.

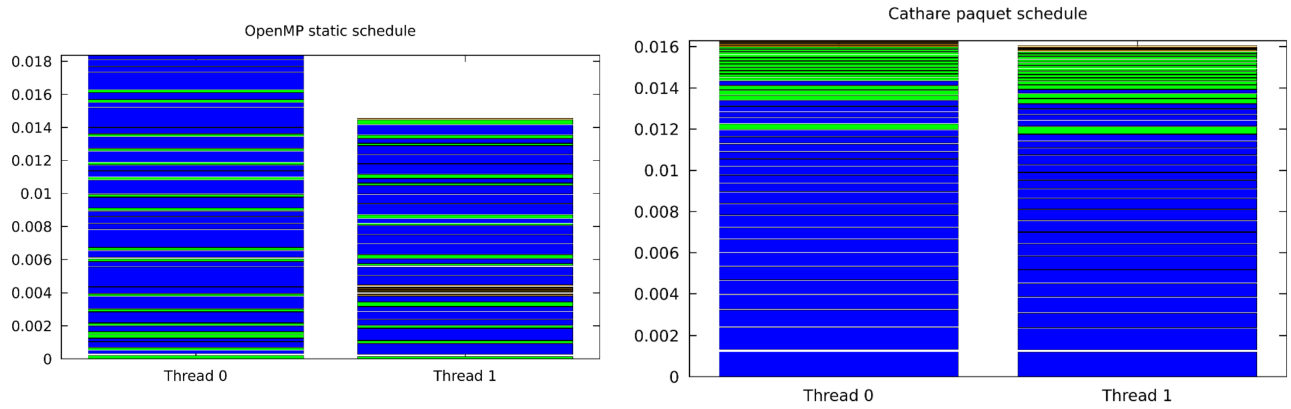


Figure 1.8: Load balancing between two threads in a Cathare simulation

### 1.4.3 Time domain decomposition: the parareal algorithm

The scalability properties of the space domain decomposition method implemented in the Cathare code are limited by the small number of cells in the meshes of the Cathare simulations. Several works are in progress to propose novel parallel algorithms in the Cathare development group. For example, the possibility to assign a 3D element to many threads by dividing the element on subdomains is actually investigated. Since the number of elements in a Cathare simulation is usually smaller than the number of available processors, we seek in this work to investigate a novel strategy of parallelisation to complement the actual parallelism in the Cathare code. For this reason, if we have more processors at our disposal and wish additional speed-ups, the parallelisation of other variables needs to be addressed.
Several approaches have been proposed over the years to decompose the time direction when solving a partial differential equation (see [96], [27], [41], [40], [51], [30], [44], [50]). Of these, the parareal algorithm, whose performances we explore in this work, was proposed two decades ago by [78] and has received an increasing amount of attention in the last years. The parareal method can also be cast into the category of multiple shooting type methods that were introduced in [96] (we refer to [55] for a detailed discussion about the several possible interpretations of the parareal method). The parareal method has been applied successfully to a number of applications (see [13], [49], [104], [79], [105] among many others), demonstrating its versatility. Theoretical advances on this method include stability analysis ( [16], [108], [15], [35]), its coupling with spatial domain decomposition methods ( [82], [61]) and control problems ( [81], [82]).
To see how the method works and how it has been applied to the two-fluid model, we write the system (1.1), after the discretisation in space on $\mathscr{N}$ degrees of freedom:

$$\frac{\partial U}{\partial t} + \mathcal{A}(t, U) = 0, \quad t \in [0, T], \quad U(t = 0) = U^0 \tag{1.39}$$

$$\mathcal{A} : \mathbb{R} \times \mathbb{R}^{\mathscr{N}} \to \mathbb{R}^{\mathscr{N}}, U \in \mathbb{R}^{\mathscr{N}} \tag{1.40}$$

where $U$ is the unknown, in our case, $U = (p, \alpha_v, h_l, h_v, u_v, u_l)$. Let us assume that we have two propagators $G$ and $F$ to solve (1.39). For any given $t \in [0, T]$, $s \in [0, T - t]$ and any function $w$ in a Banach space, $G(t, s, w)$ (respectively $F(t, s, w)$) takes $w$ as an initial value at time $t$ and propagates it at time $t + s$.

- $G(T^n, \Delta T, U^n)$ computes a coarse approximation of $U(T^{n+1})$ with initial condition $U(T^n) \simeq U^n$ (low computational cost)
- $F(T^n, \Delta T, U^n)$ computes a more accurate approximation of $U(T^{n+1})$ with initial condition $U(T^n) \simeq U^n$ (high computational cost)

The fine propagator $F$ can perform the propagation of the phenomenon with small time steps $\delta t$ and with very accurate physics described by $\mathcal{A}$. On the other hand, the coarse approximation $G$ does not need to be as accurate as $F$ and can be chosen much less expensive, by the use of a scheme with a much larger time step $\Delta T \gg \delta t$ or by treating reduced physics.
In addition to these two propagators $F$ and $G$, the parareal in time algorithm is based on the division of the time interval $[0, T]$ into $N$ sub-intervals $[0, T] = \cup_{n=0}^{N-1}[T^n, T^{n+1}]$ that will each be assigned to a processor $P^n$. The parareal algorithm applied to (1.39) is an iterative technique where, at each iteration $k$, the value $U(T^n)$ is approximated by $U_k^n$ with an accuracy that tends to the one achieved by the fine solver when $k$ increases. $U_k^n$ is obtained by the recurrence relation:

$$U_{k+1}^{n+1} = G(T^n, \Delta T, U_{k+1}^n) + F(T^n, \Delta T, U_k^n) - G(T^n, \Delta T, U_k^n) \tag{1.41}$$

Starting from $U_0^{n+1} = G(T^n, \Delta T, U_0^n)$.
From formula (1.41), one can see by recursion that the method is exact after enough iterations. Indeed, for any $n > 0, U_n^n = F(T^0, T^n - T^0, U^0)$. However, convergence of $U_k^n$ to $F(T^0, T^n - T^0, U^0)$ goes much faster than this.
While the main results about the convergence properties of the method were studied in depth several years ago (see [78], [13], [16]), more recent efforts ([85], [11], [40], [18], [80]) focus on the algorithmics to implement it in order to improve the speed-up provided by the original algorithm.

### 1.4.4 Time domain decomposition for hyperbolic problems

An instability of the parareal algorithm may appear when it is applied to convection dominated problems. In [15], the author points out the need of a strong regularity on the initial condition to ensure the stability of the parareal algorithm. When the problem is parabolic, the smoothing character of the problem prevents the appearance of instabilities even if the initial condition is not regular enough but when the problem is hyperbolic, an initial condition with high frequencies components may trigger the instability. Moreover in [15], the author shows the influence of the numerical diffusion on this instability. This aspect has also been explored in [102], [49], [79], [103].

Others strategies have been studied to treat the instability of the original parareal. They propose to improve the coarse approximations at every parareal iteration using the previous fine solutions to overcome the instability issues with hyperbolic equations. In [48] and [54], the parareal solution is projected in a Krylov subspace generated by the set of fine solutions from the previous parareal iterations. In [29], the authors use a subspace thanks to a reduced basis built from the matrix made of the previous fine solutions. In the same spirit, in [35], the authors propose a parareal algorithm with an additional step that project the parareal solution in a manifold to ensure the conservation of invariants (for example, the conservation of the Hamiltonian).

Other contributions propose cures to the parareal algorithm with an algebraic viewpoint. In [28], the authors formulate the method with an iteration matrix and propose preconditionners to improve the behaviour of the parareal algorithm.

In [52], the author shows the difficulties of the parareal algorithm to converge in a reasonable number of iterations on the advection equation and the wave equation using the method of characteristics.

In [95], the authors propose an algorithm called Communication Aware Adaptive Parareal (CAAP) to speed up the non linear shallow water equation beyond what is possible using spatial domain decomposition methods alone.

We will also note that the application of other parallel in time algorithms is not straightforward on convection dominated problems. In [109], the authors propose MGRIT algorithms that improve stability and scalability for the resolution of the advection equation with 1st-order numerical scheme. In [110], analysis tools are proposed to understand the source of the instability issues on MGRIT algorithms and on the parareal algorithm using the multigrid interpretation. In [73], a convergence analysis is conducted to propose criteria for coarse-grid operators involved in the MGRIT and parareal algorithms. This strategy can ensure stability and scalability for the time parallelisation of the advection equations with high order discretisations.

# Chapter 2

# Parareal algorithm for two phase flows simulation

**Contents**



In this chapter, we present the two strategies we developed during the PhD in order to apply the parareal algorithm to the Cathare code: a numerical clone of Cathare that is restricted to one test case and a library that uses the Cathare code in a non intrusive way. The main contribution of this work has been to adapt the parareal algorithm to the architecture of the software and to its time discretisation in a non intrusive way, without any changes of the source files of the Cathare code, in order to reduce the computational time and get closer to a real-time response of the code. In section 2.1, we present the challenges of implementing the parareal algorithm on an industrial software, the Cathare code, in a non intrusive way. In section 2.2, we introduce the new algorithm we designed to handle multi-step time schemes such as the one used within the Cathare code. In the last section, we report the speed up performances obtained on two test cases: the oscillating manometer using the numerical clone of the Cathare code and the Omega test case using the Parareal library that uses the Cathare software. Each test case is representative of the numerical challenges for the simulation of two phase flows in the context of safety studies such as vanishing phases and discontinuous source terms.

## 2.1 The Parareal library for the Cathare code

The development of the Cathare code started 30 years ago for the simulation of nuclear reactors under nominal or accidental situations. The software was designed for engineers in nuclear energy and experts in the thermalhydraulics of accidental scenarii in nuclear power plants. Hence, the terminology used in Cathare is directly linked to the physics of nuclear reactors. For example, to define the geometry of the simulation, the user has to define a reactor, a primary and a secondary circuit and hydraulic elements representing pumps or steam generators. Generally, the definition of the simulation one seeks to run is composed of two main blocks:

- First, informations are given for the description of the circuits. Each element is defined with the reference to a hydraulic element or Cathare element (1-D, 0-D, 3-D, junction, etc...). Geometrical parameters and meshing are defined. Elements are connected to constitute elementary circuits (one primary and several secondary circuits). Heat exchangers between elementary circuits are defined.
- The characteristics of the calculation are then specified. Successive directives are given corresponding to the different steps of the simulation:
  - operation of the initial state process,
  - time propagation called transient calculation,
  - time step control,
  - events occuring during the calculation (safety injection, break opening, valves, ...).

The actual parallel methods available in the Cathare code (see section 1.4 chapter 1) are implemented with OpenMP that allows shared memory parallelism. Their use is limited to a level of parallelism of about 20 processors or of at most all the processors of a standard desk computer. In this work, all the developments of the parareal algorithm applied to the Cathare code were made with MPI that allows distributed memory parallelism. Hence, the parareal algorithm allows to run a Cathare simulation in parallel on many processors located in different computers, including supercomputers.

In the sequel, we summarise the obstructions linked to the Cathare structure and the strategies we chose to handle them. These cures were of two natures: computer science and algorithmic. We think that the experience of implementing a parallel algorithm to an industrial software in a non intrusive way will be instructive for the future developments of parallel techniques in the Cathare code. This is why we list the obstructions that are only of a computer science nature. In the sequel, we list the adaptations and adjustments necessary to develop a library that applies the parareal algorithm to the Cathare code where it is used as a black box without modifying the source files of the software.

### 2.1.1 Obstructions linked to the data structure of the Cathare code

#### 2.1.1.1 Data structure

The data structure in the Cathare code mainly lies on an array of values containing the informations related to the mesh, the unknowns, the coefficients of the Jacobian matrix and of the right hand side. This array stores the principal variables of pressure, volume fraction and liquid and vapour velocities and enthalpies for every cell of the mesh. The Cathare array also contains auxiliary variables that are necessary to carry on the simulation. These auxiliary values are of different natures:

- variables computed from the principal unknowns
- variables tracked during the simulation: for example, the water level in an element of the system
- boolean variables giving the flow regime in an area of the system: bubbly, annular, dispersed, seperated phases, etc.

The first obstruction we met is that we can only define one system (one reactor and one array of values) in a Cathare simulation while in the context of the parareal algorithm we use two systems: a coarse one and a fine one. The first difficulty is to run a simulation with coarse and fine systems in the same processor. There are two possible approaches:

- When running the simulation over $N$ processors, we assign one processor ("the coarse processor") to the coarse propagation for all time windows $[T^n, T^{n+1}]$ and the remaining processors ("the fine processors") run the fine propagation only, over one time window. The coarse processor will also transmit the coarse approximation at times $T^n$ to the corresponding fine processor. The main drawback of this approach is the multiplication of communications and the use of $N+1$ ressources for a parallelism over $N$ time windows.
- We use the same array of values for the coarse and the fine propagations in each processor. This is made possible by the use of the Parareal library that is independent of the Cathare code and contains the parareal algorithm. This library collects the coarse and fine propagations made by the Cathare code then apply the parareal corrections and finally send to the Cathare code the updated initial conditions for each processor $[T^n, T^{n+1}]$. Depending on whether the Cathare will make a coarse or a fine propagation, the Parareal library will send the data array associated to the coarse or the fine solver (for example, the coarse or fine time step and the suitable initial condition). For the implementation of the parareal algorithm applied to the Cathare in a non intrusive way, we chose this option.

The data structure of the Cathare code represents a challenge for memory storage with up to $10^6$ variables and a size of about 8 Go. To reduce the cost of communication, we choose to exchange only the principal variables between processors. Some of the auxiliary variables are then computed from the principal unknowns. This choice will have consequences on the accuracy due to the error made on the reconstruction of the auxiliary variables. This question is addressed in section 2.3.2.

#### 2.1.1.2 Data set

A Cathare data set can be split in different blocks of instructions:

- definition of the geometry and the mesh
- calculation of an initial condition: computation of a stationary state starting from the instructions of the user: pressures and temperatures for some specific cells, for example, or rotating speed of a pump or flow direction
- a loop for the time integration from $T^0$ to $T^f$

The first two points are called "Initialisation" step and the last one "Time integration" step.
For the application of the parareal algorithm, we split the Cathare data file into two files for the "Initialisation" step and for the "Time integration" step. Each processor $P_n$ has an initialisation file and a file for the time integration between $T^n$ and $T^{n+1}$. The initialisation file of processor $P^0$ is

particular since it builds the geometry and mesh, calculates the initial state of the simulation and sends them to the other processors. The initial state will not be used in the remaining processors but this communication allows to initialise the size of the Cathare arrays in each processor. After that, $P_0$ makes a coarse propagation and sends $G(T^0, T^1, U_0^0)$ to $P_1$. The other processors receive the informations for the geometry and mesh and their local initial condition $G(T^{n-1}, T^n, U_0^{n-1})$ to run their own coarse propagation. We met two obstructions at this stage:

- The calculation in $[T^1, T^2]$ and generally for $[T^n, T^{n+1}]$, with $n \neq 0$, has to start from an initial time $T^n$ different from 0. It is unusual for the Cathare code to start a simulation from a time $T^n \neq 0$. The consequence is: instead of beginning the calculation with the coarse solution sent from the previous time window as an initial condition, the processor uses the initial state sent from rank 0 processor.
  To treat this difficulty, the time window makes one time step and delete the computed solution. This allows to fix the initial time $T^n$ of the time window to the correct value.

- The size of the arrays for every processors is computed at the moment processor $P_0$ sends to $P_i, i = 1, \cdots, N-1$ the geometry, mesh and initial state. If some variables are declared between the initialisation step and the time loop then the array in processor $P_0$ will have the correct size but the arrays in the remaining processors will still have the size of the state sent by $P_0$ during the initialisation step. Hence, the arrays in processors $P_i, i = 1, \cdots, N-1$ have a different size from the one in processor $P_0$. This will generate conflicts during the communications between processors and the copy of arrays. Thus we need to adapt the "Initialisation" data file of processor $P_0$ to handle this.

### 2.1.2 Obstructions linked to the time discretisation

The Cathare time discretisation is based on a two-step time scheme (see Chapter 1, section 1.3.2 on the Cathare discretisation methods). This leads us to design a new variant of the parareal algorithm that takes more efficiently into account the presence of multi-step time schemes. We detail this aspect in section 2.2. Moreover, the Cathare time scheme uses an adaptive time step.

In a Cathare simulation, the user imposes an initial time step $\Delta t_0$ and a maximal time step $\Delta t_{max}$. At each iteration, the code gets closer to $\Delta t_{max}$ by multiplying by 2 the current time step while respecting the CFL condition integrated in the code. If in a time iteration, the Newton method does not converge after the maximal number of iterations, the current time step is reduced by multiplying it by $\frac{2}{3}$. Thus, we need to communicate the fine and coarse time steps between processors. For the coarse propagation, we can transfer the time step computed at the current parareal iteration because it is a sequential step. Since the fine propagation is made in parallel, the time window $[T^n, T^{n+1}]$ at the $k$-th parareal iteration uses the fine time step from the time window $[T^{n-1}, T^n]$ computed at the $(k-1)$-th parareal iteration.

### 2.1.3 A numerical clone of the Cathare code

In this section, we present the numerical clone of the Cathare code, called MiniCathare, implemented during the PhD. This tool allowed us to make a first trial on the efficiency of the parareal algorithm to speed up a two-phase test case using the Cathare model and numerical scheme. The test case we consider here is the oscillating manometer (see section 2.3.1 for the description of the test case and the numerical results obtained with the numerical clone). Hence, MiniCathare is restricted to one test case. We give in the sequel some details about this numerical tool:

- MiniCathare only makes the time propagation in a time intervall $[T^0, T^f]$ and does not compute an initial state like the Cathare code. The initial condition in MiniCathare is extracted from the Cathare code after the initial state process.

- MiniCathare is implemented in C++ and the parareal algorithm in MPI.

- MiniCathare allows to make convergence tests which is not possible with the Cathare code since there is a limitation on the mesh size in Cathare. In Chapter 1, section 1.3.3.2, we show the convergence properties of the Cathare numerical scheme on the oscillating manometer by comparing several numerical solutions, obtained by MiniCathare, to the reference solution of this test case.

- The only difference between MiniCathare and the Cathare code is the solution of the non linear problem at each time step:

$$\frac{U^{n+1} - U^n}{\Delta t} + A(U^{n+1}, U^n) = S(U^n)$$

In both cases, this non linear system is solved by a Newton method:

$$\begin{array}{l} \dfrac{\delta U^{k+1}}{\Delta t} + J(U^k, U^n)\delta U^{k+1} = \tilde{S}(U^n, U^k), \text{ where: } \delta U^{k+1} = U^{k+1} - U^k \\ \text{and : } U^{k+1} = (p^{k+1}, \alpha_v^{k+1}, H_l^{k+1}, H_v^{k+1}, u_l^{k+1}, u_v^{k+1}) \end{array} \tag{2.1}$$

The increments of the principal variables in Cathare are denoted $\delta U^{k+1}$ and the terms of the Jacobian matrix $J(U^k, U^n)$ are computed as follows:

$$J_{i,j} = \frac{\partial E_i}{U_j^k}$$

Since the Cathare time discretisation is semi-implicit, this term corresponds to the derivative of the balance equation $E_i$ with respect to the principal variable $U_j^k$ that is implicited for the $k$-th Newton iteration. The resulting linear system (2.1) is solved differently in Cathare and MiniCathare:

  - Cathare, by a Gauss elimination, obtains a system with pressure increments $\delta p^{k+1}$ only and then solves the problem in pressure with a direct linear solver (Lapack-Blas)
  - MiniCathare assembles the whole Jacobian matrix and solves the complete linear system with an iterative linear solver (PETSC library)

## 2.2 Multi-step variant of the parareal algorithm

Several approaches have been proposed over the years to decompose the time direction when solving a partial differential equation (see [50] for an overview). Of these, the parareal in time algorithm, which performances we explore in this work, has received an increasing amount of attention in the last twenty years with many applications (see [13], [49], [104] among many others). In the sequel, we recall the classical parareal algorithm ([78], [13], [16]) and present the multi-step variant we will apply to the Cathare code.

### 2.2.1 Original parareal algorithm and notations

After the discretisation of a PDE in space, we obtain an ODE system of the form:

$$\frac{\partial u}{\partial t} + \mathcal{A}(t,u) = 0, \quad t \in [0,T], \quad u(t=0) = u_0 \tag{2.2}$$

where $\mathcal{A} : \mathbb{R} \times \mathbb{R}^{\mathscr{N}} \to \mathbb{R}^{\mathscr{N}}$, and $\mathscr{N}$ denotes the number of degrees of freedom.
Let $G$ and $F$ be two propagators such that, for any given $t \in [0,T]$, $s \in [0, T-t]$ and any function $w$ in a Banach space, $G(t,s,w)$ (respectively $F(t,s,w)$) takes $w$ as an initial value at time $t$ and propagates it at time $t+s$. The full time interval is divided into $N^c$ sub-intervals $[T^n, T^{n+1}]$ of size $\Delta T$ that will each be assigned to a processor. The algorithm is defined using two propagation operators:

- $G(T^n, \Delta T, u^n)$ computes a coarse approximation of $u(T^{n+1})$ with initial condition $u(T^n) \simeq u^n$ (low computational cost)
- $F(T^n, \Delta T, u^n)$ computes a more accurate approximation of $u(T^{n+1})$ with initial condition $u(T^n) \simeq u^n$ (high computational cost)

Starting from a coarse approximation $u_0^n$ at times $T^0, T^1, \cdots, T^{N^c}$, obtained using $G$, the parareal algorithm performs for $k = 0, 1, \cdots$ the following iteration:

$$u_{k+1}^{n+1} = G(T^n, \Delta T, u_{k+1}^n) + F(T^n, \Delta T, u_k^n) - G(T^n, \Delta T, u_k^n)$$

In the parareal algorithm, the value $u(T^n)$ is approximated by $u_k^n$ at each iteration $k$ with an accuracy that tends rapidly to the one achieved by the fine solver, when $k$ increases. The coarse approximation $G$ can be chosen much less expensive than the fine solver $F$ by the use of a scheme with a much larger time step (even $\delta T = \Delta T$) $\delta T \gg \delta t$ (time step of the fine solver) or by using a reduced model. All the fine propagations are made in parallel over the time windows and the coarse propagations are computed in a sequential way but have a low computational cost. We refer to [80] about the parallel efficiency of parareal and a recent work offering a new formulation of the algorithm to improve the parallel efficiency of the original one. The main convergence properties were studied in [55] and stability analysis was made in [108], [14].

### 2.2.2 Adaptation to multi-step time schemes

#### 2.2.2.1 Case of a two-step fine time scheme

In the sequel, we will consider the case where the fine solver is a two-step time scheme. Hence we will use the following notation for the fine solver $F$ that takes two initial values:

$$F(t,s,x,y), \quad t \in [0,T], \quad s \in [0, T-t[$$

and the initial values $x, y$ are in a Banach space $\mathbb{U}$.

**Example 2.1.** *If one solves* (2.2) *with a multi-step time scheme as fine propagator $F$ like the second-order BDF method:*

$$\frac{3}{2}u^{j+1} - 2u^j + \frac{1}{2}u^{j-1} = -\delta t \mathcal{A}(u^{j+1}, t^{j+1}), \quad j = 1, \cdots, N^f, t^{j+1} - t^j = \delta t$$

*Here the fine solver reads as: $u^{j+1} = F(t^j, \delta t, u^{j-1}, u^j)$. Now, we apply the parareal algorithm with a coarse grid: $T^0, \cdots, T^{N^c}$ where:*

$T^{n+1} - T^n = \Delta T = N^f \delta t$.
*Then we can write:* $u(T^n + j\delta t) \simeq u^{n,j}, j = 1, \cdots, N^f, n = 1, \cdots, N^c$.
*In order to perform the fine propagation, in a given time window* $[T^n, T^{n+1}]$*, we only need the local initial condition* $u_k^n$ *and a consistent approximation of* $u(T^n - \delta t)$.

In [12], the authors propose a consistent approximation in the context of the simulation of molecular dynamics. The proposed method was linked to the nature of the model and the symplectic character of their algorithm is shown, which is an important property to verify for molecular dynamics.
In the context of our application to the thermalhydraulic code Cathare, we want to derive a multi-step variant of parareal that will not be intrusive in the software. We seek to derive a consistent approximation of $u(T^n - \delta t)$. The only fine trajectory at our disposal is $F(T^{n-1}, \Delta T, u_k^{n-2,N^f-1}, u_k^{n-1})$. Its final value at time $T^n$ is:
$F(T^{n-1}, \Delta T, u_k^{n-2,N^f-1}, u_k^{n-1})(T^n)$ from which we compute $u_{k+1}^n$ by the parareal correction. Hence, we translate the solution:
$F(T^{n-1}, \Delta T - \delta t, u_k^{n-2,N^f-1}, u_k^{n-1})(T^n - \delta t)$ by the same correction:
$u_{k+1}^n - F(T^{n-1}, \Delta T, u_k^{n-2,N^f-1}, u_k^{n-1})$ and obtain the so called consistent approximation $u_{k+1}^{n-1,N^f-1}$ to initialise the fine propagation in $[T^n, T^{n+1}]$. We now detail our algorithm:

$$\begin{cases} u_0^{n+1} = G(T^n, \Delta T, u_0^n), \quad 0 \leq n \leq N-1 \\ u_0^{n-1,N^f-1} = u_0^n \\ u_{k+1}^{n+1} = G(T^n, \Delta T, u_{k+1}^n) + F(T^n, \Delta T, u_k^{n-1,N^f-1}, u_k^n) \\ \qquad\qquad -G(T^n, \Delta T, u_k^n), \quad 0 \leq n \leq N-1, \quad k \geq 0 \\ u_{k+1}^{n,N^f-1} = F(T^n, \Delta T - \delta t, u_k^{n-1,N^f-1}, u_k^n) + u_{k+1}^{n+1} \\ \qquad\qquad -F(T^n, \Delta T, u_k^{n-1,N^f-1}, u_k^n), \quad 0 \leq n \leq N-1, \quad k \geq 0 \end{cases} \tag{2.3}$$

Another option to treat this issue is to use a one-step time scheme to initialise the fine computation. We will see from the numerical results that this choice generates an error greater than the target accuracy and prevents the parareal algorithm to converge towards the solution with the desired accuracy.
The algorithm (2.3) adds consistency with the fine scheme. Also, this strategy can be generalised to multi-step time schemes involving several fine time steps preceding the time $T^n$ by applying the same correction to terms taking the form: $u_{k+1}^{n,N^f-i}, \quad i = 1, \cdots, I$.

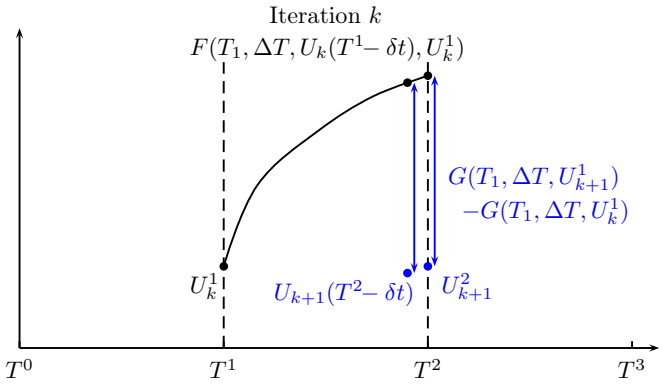


Figure 2.1: Correction of $u_{k+1}^{1,N^f-1}$ at time $T^2 - \delta t$ in $[T^1, T^2]$ for the initialisation of the fine propagation in $[T^2, T^3]$

#### 2.2.2.2 Case of two-step coarse and fine time schemes

In the sequel, we will consider the case where the fine solver and the coarse solver are both two-step time schemes. Hence we will use the following notation for the coarse $G$ solver that takes two initial values:

$$G(t, s, x, y), \text{ for } t \in [0, T], s \in [0, T - t[ \text{ and } x, y \in \mathbb{U}$$

We propose to add a correction for the solution at time $T^n - \delta T$, where $\delta T$ is the coarse time step, to initialise the coarse propagation in each time window in a consistent way. The parareal solution at time $T^n - \delta T$ and $k$-th parareal iteration is:

$$u_k^{n-1,N^f-R} \simeq u((n-1)\Delta T + (N^f - R)\delta t) = u(T^n - \delta T)$$

where $R$ is the ratio between the coarse and the fine time steps: $R = \dfrac{\delta T}{\delta t}$.
The full multi-step parareal algorithm (2.4) makes two additionnal corrections compared to the classical parareal algorithm when the coarse and fine propagators are based on one two-step time schemes: one at times $T^n - \delta t$ (see figure 2.1) and the other at times $T^n - \delta T$ (see figure 2.2).

$$\begin{cases} u_{k+1}^{n+1} &= G(T^n, \Delta T, u_k^{n-1,N^f-R}, u_{k+1}^n) + F(T^n, \Delta T, u_k^{n-1,N^f-1}, u_k^n), \\ &- G(T^n, \Delta T, u_k^{n-1,N^f-R}, u_k^n), \quad 0 \le n \le N-1, \quad k \ge 0 \\ u_{k+1}^{n,N^f-1} &= F(T^n, \Delta T - \delta t, u_k^{n-1,N^f-1}, u_k^n) + u_{k+1}^{n+1} - F(T^n, \Delta T, u_k^{n-1,N^f-1}, u_k^n), \quad 0 \le n \le N-1, \quad k \ge 0 \\ u_{k+1}^{n,N^f-R} &= G(T_n, \Delta T - \delta T, u_{k+1}^{n-1,N^f-R}, u_{k+1}^n) + F(T_n, \Delta T - \delta T, u_k^{n-1,N^f-1}, u_k^n) \\ &- G(T_n, \Delta T - \delta T, u_k^{n-1,N^f-R}, u_k^n), \quad 0 \le n \le N-1, \quad k \ge 0 \end{cases} \tag{2.4}$$

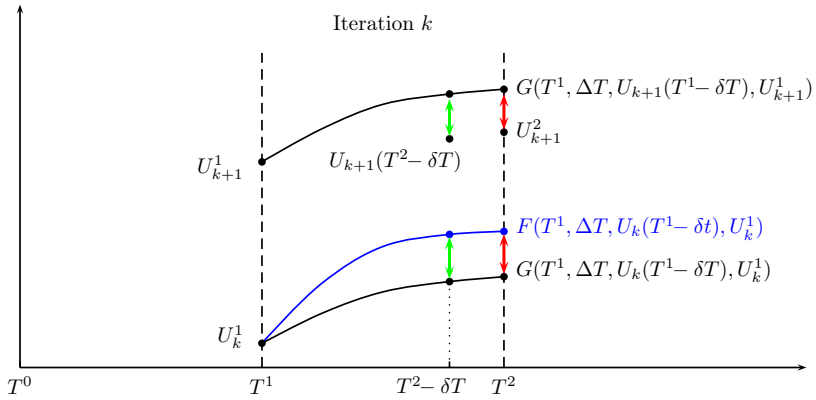


Figure 2.2: Correction of $u^{1,N^f-R}_{k+1}$ at time $T^2 - \delta T$ in $[T^1, T^2]$ for the initialisation of the coarse propagation in $[T^2, T^3]$

The full multi-step version (2.4) of the algorithm does not take into account the adaptive time stepping of the Cathare time scheme. A consequence of the adaptive time steppping can be seen on the update of the solution $u^{n,N^f-R}_{k+1}$. This quantity is corrected at each iteration as follows:

$$\begin{aligned} u_{k+1}(T^{n+1} - \delta T^n_{k+1}) &= G_{k+1}(T_n, \Delta T - \delta T^n_{k+1}, u^{n-1,N^f-R}_{k+1}, u^n_{k+1}) + F(T_n, \Delta T - \delta T^n_k, u^{n-1,N^f-1}_k, u^n_k) \\ &- G_k(T_n, \Delta T - \delta T^n_k, u^{n-1,N^f-R}_k, u^n_k) \end{aligned} \tag{2.5}$$

We distinguish the coarse solver $G$ at iterations $k$ and $k+1$ by the subscript $G_k$ and $G_{k+1}$ because the solver $G$ does not have the same sequence of time steps at iterations $k$ and $k+1$. Hence: $\delta T^n_{k+1} \neq \delta T^n_k$ and the correction (2.5) combines a quantity at time $T^{n+1} - \delta T^n_{k+1}$ with other quantities at time $T^{n+1} - \delta T^n_k$. This correction becomes incoherent. To correct this inconsistency, we propose to store the times $T^n - \delta T^n_0$ for each time window, at the parareal initialisation. Then, we impose to the coarse solver to pass by the point $T^n - \delta T^n_0$ in $[T^n, T^{n+1}]$, for every parareal iterations.

In the next section, we apply the multi-step parareal algorithm to two-phase test cases: the oscillating manometer and the simulation of a breach in the primary circuit of a nuclear reactor. The convergence analysis of this new algorithm is the subject of the next chapter.

## 2.3 Application to the Cathare code

### 2.3.1 The oscillating manometer

Here we apply the multi-step parareal algorithm to the resolution of an oscillating manometer ([6, 7]). This test case is proposed in [69] for system codes to test the ability of each numerical scheme to preserve system mass and to retain the gas-liquid interface. In [69], the oscillating manometer is proposed as a numerical benchmark test for system codes to test the ability of each numerical sheme to preserve system mass and to retain the gas-liquid interface. It consists in a U-shaped tube manometer which is connected at the top, so that a closed system is formed. The system contains initially gas and liquid with the liquid forming equal levels in each arm of the manometer. Further, all parts of the fluid system have initially a uniform non zero velocity, but zero acceleration. Under these initial conditions, a hydrostatic pressure hypothesis is made

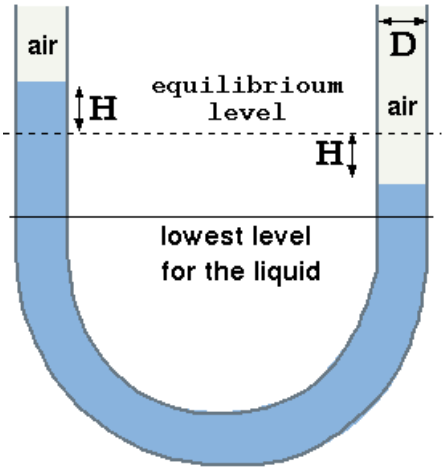


Figure 2.3: Oscillating manometer test case

throughout the system. Also, the system is isothermal at 50°C with $10^5$ Pa pressure at the vapour-liquid interfaces. Distance in the direction of the flow is measured by $x$ in meters. The length of the manometer is 20 m and the diameter is $D = 1$ m. The initial position of the vapour-liquid interface is 5m from the bottom of each manometer leg and the fluid initially has a velocity of $u_0 = 2.1$ m/s. This initial velocity will cause the interface to oscillate approximately $\pm 1.5$ m in height from the initial location.

In this test case, the phases are separated and the interfacial friction term will be important in this configuration.

#### 2.3.1.1 Model

The model used in Cathare is the 6 equation two-fluid model that considers a set of balance laws (mass, momentum and energy) for each phase, liquid and vapor. It assumes independent velocities and a pressure equilibrium.
The unknowns are the volume fraction $\alpha_k \in [0,1]$, the pressure $p \geq 0$, the velocity $u_k$ and the enthalpy $H_k$ of each phase. The subscript $k$ stands for $l$ if it is the liquid phase and $g$ for the gas phase. For the sake of simplicity, we write the terms of the model involved in our test case, studied in section 2.3.1.

$$\begin{cases} \partial_t(\alpha_k\rho_k) + \partial_x(\alpha_k\rho_k u_k) = 0 \\ \alpha_k\rho_k\partial_t u_k + \alpha_k\rho_k u_k\partial_x u_k + \alpha_k\partial_x p = \alpha_k\rho_k g + F_k^{\text{int}} \\ \partial_t\left[\alpha_k\rho_k\left(H_k + \frac{u_k^2}{2}\right)\right] + \partial_x\left[\alpha_k\rho_k u_k\left(H_k + \frac{u_k^2}{2}\right)\right] = \alpha_k\partial_t p + \alpha_k\rho_k u_k g \end{cases} \tag{2.6}$$

with $\alpha_v + \alpha_l = 1$ and the two equations of state : $\rho_k = \rho_k(p, H_k)$.

The interfacial forces $F_k^{\text{int}}$ are of 2 types. The first ensures hyperbolicity of the system (see [89] for the well-posedness of the 6 equation model). The second is the interfacial friction term that has an important role for our test case. For the oscillating manometer, the phases are separated which means that one of the two phases vanishes in some parts of the domain. It is numerically challenging to compute the velocity of the vanishing phase (see [93]). For this reason, the Cathare scheme forces the two velocities to be equal with the use of a damping term: the interfacial friction term.

#### 2.3.1.2 Numerical method

The Cathare scheme is based on a finite volume method on a staggered grid (MAC scheme) and on a two step time scheme. In a staggered scheme the i-th component of the velocity is located at the center of the edge orthogonal to the i-th unit vector. Pressures, void fractions and enthalpies are cell-centered. Given a time discretisation $T^0, T^1, T^2, \cdots$ of the full time interval $[0, T)$, we use the following notations: $(\alpha_k \rho_k)^n$ is an approximation of $(\alpha_k \rho_k)$ at time $T^n$. Here, we write the time discretisation of the Cathare scheme:

$$\begin{cases} \dfrac{(\alpha_k\rho_k)^{n+1} - (\alpha_k\rho_k)^n}{\Delta t} + \partial_x(\alpha_k\rho_k u_k)^{n+1} = 0 \\ (\alpha_k\rho_k)^{n+1}\dfrac{u_k^{n+1} - u_k^n}{\Delta t} + (\alpha_k\rho_k u_k)^{n+1}\partial_x u_k^{n+1} + \alpha_k^{n+1}\partial_x p^{n+1} = (\alpha_k\rho_k)^{n+1} g + F_k^{n,n+1} \\ \dfrac{1}{\Delta t}\left[(\alpha_k\rho_k)^{n+1}\left(H_k + \dfrac{u_k^2}{2}\right)^{n,n+1} - (\alpha_k\rho_k)^n\left(H_k + \dfrac{u_k^2}{2}\right)^{n-1,n}\right] \\ +\partial_x\left[\alpha_k\rho_k u_k\left(H_k + \dfrac{u_k^2}{2}\right)\right]^{n+1} = \alpha_k^{n+1}\dfrac{p^{n+1} - p^n}{\Delta t} + (\alpha_k\rho_k u_k)^{n+1} g \end{cases} \tag{2.7}$$

Where the notation $F_k^{n,n+1}$ shows that the time discretisation of $F_k$ is a function of the numerical solution at times $T^n$ and $T^{n+1}$. After discretisation, the non linear system is solved by a Newton method. In this test case, a numerical difficulty of two-phase flows simulations arises, namely the vanishing phase. An important issue is to guarantee the positivity of the volume fraction. Many schemes were designed to ensure this property (like [93] for two incompressible phases). The Cathare code uses a high interfacial friction to deal numerically with this difficulty.

#### 2.3.1.3 About the convergence

In this section, we apply the multi-step parareal algorithm (2.4) to the simulation of the oscillating manometer. We use the same physical model and the same mesh (110 cells) for both the coarse and the fine solvers: the only difference is the size of the time steps, $\delta t$ for $F$ and $\Delta T$ for G. All the calculations have been evaluated with a stopping criteria where the tolerance is fixed to the fine solver accuracy, $\epsilon = 5 \cdot 10^{-2}$. With this threshold, parareal convergence is achieved after 2 or 3 iterations.
In the sequel, after giving a numerical proof of the convergence of the parareal algorithm in our test case, some results about measured speed-up will be presented.

Figure 2.4 illustrates that the multi-step parareal algorithm effectively converges when applied to the problem of the oscillating manometer. For a given time step $T^n$ and parareal iteration $k$, the relative error in $L^2$ norm between the parareal solution and the sequential fine solver decreases beyond our given convergence threshold $\epsilon$. In the figure, the test case has been solved with the multi-step parareal algorithm when $\delta t = 10^{-5}$ and $\Delta T = 10\delta t$.
These results are obtained on 16 time windows.

#### 2.3.1.4 Speed-up performances

In the following strong scaling tests, the same setting is used for the multi-step parareal algorithm. The test case has been solved on an increasing number $N_{proc}$ of processes $N_{proc} = 5, 10, 15, \cdots, 70$. In figure 2.5, with 25 processes, we obtain a speed up of 3.4 and of 3.7 with 50 processes. Here, we observe two global trends:

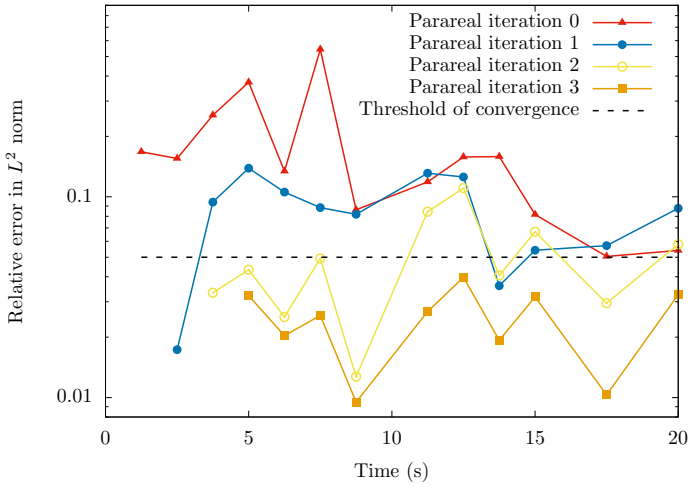


Figure 2.4: Convergence of the multi-step parareal algorithm when $\delta t = 10^{-5}$ and $\Delta T = 10\delta t$

- For $N_{proc} = \{5, 10, 15, 20, 25, 40, 50\}$, the speed up first monotonically increases until reaching 25 processes and then increase again with 40 and 50 processes. This is due to the number of parareal iterations that is equal to 2 in this case

- For $N_{proc} = \{30, 35, 45, 55, 60, 65, 70\}$, the speed up is drastically reduced because the parareal algorithm converges in 3 iterations in this case

In the sequel, we recall the well-known dependence of the computational cost of the parareal algorithm on the number of iterations. Let $T_{fine}$ be the CPU time to run the fine solver in a sequential way on the whole time interval $[0, T)$. Since the coarse time step is ten times greater than the fine time step we suppose that the cpu time of the coarse solver $T_{coarse} = \dfrac{T_{fine}}{10}$. This ratio between coarse and fine solvers should be as high as possible to minimise the computational cost of the coarse solver which is launched in a sequential way. When the algorithm converges in $N_{it}$ iterations, the coarse solver is launched $N_{it}$ times and the fine solver $N_{it} - 1$ times in parallel over the number of processes $N_{proc}$. Hence, we can write the cpu time in parallel $T_{para}$ in terms of $T_{fine}$:

$$T_{para} = (N_{it} - 1)\frac{T_{fine}}{N_{proc}} + N_{it}T_{coarse} + \tau = \left(\frac{N_{it} - 1}{N_{proc}} + \frac{N_{it}}{10}\right) T_{fine} + \tau$$

where $\tau$ contains the time of communication between processes and the cpu time for the computation of the parareal corrections and of the error. Now, we can deduce an upper bound of the speed up $S$ when the parareal algorithm converges in 2 or 3 iterations by neglecting $\tau$:

$$S = \frac{T_{fine}}{T_{para}} \leq \frac{1}{\frac{N_{it}-1}{N_{proc}} + \frac{N_{it}}{10}}$$

*Example:* On 25 processes, the algorithm converges in 2 iterations: $S \approx 4$ when the measured speed up is 3.4.
On 35 processes, the algorithm converges in 3 iterations: $S \approx 2.8$ when the measured speed up is 2.3.

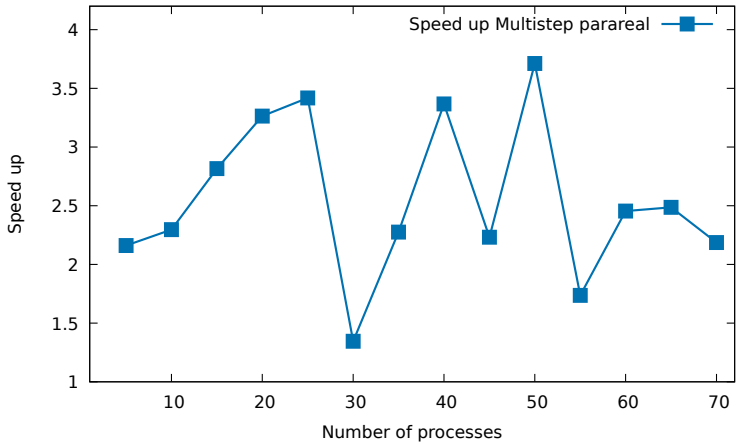


Figure 2.5: Strong scaling results with the multi-step variant of the parareal algorithm

### 2.3.2 An industrial test case

#### 2.3.2.1 Description of the test case

In this section, we report our efforts to apply the parareal algorithm to the time parallelisation of an industrial test case representative of the numerical difficulties met in nuclear safety studies. This test case simulates a breach in the primary circuit downstrean on the reactor core of a Pressurised Water Reactor. The size of this breach is $512mm^2$. This accidental scenario is studied to simulate numerically the behaviour of the nuclear reactor under a hypothetic break of a pipe welding and during the emergency procedure following the accident. In our test case, we only simulates the reactor core before the emergency procedure. Here, the system is composed of:

- the primary circuit
- the downcomer where the fluid transits from the primary circuit to the reactor core
- the reactor core, modeled by two sets of uranium rods

The boundary conditions replicate the effects of the breach on the system:

- Inlet boundary conditions: at $t = 0$ there is a single-phase liquid flow then the liquid flowrate decreases from $t = 2.8s$, time of the breach appearance, until reaching 0 at time $t = 7.9s$. After the breach, the volume fraction of the vapour phase increases until reaching the maximal value 1 at time $t = 19s$.
- Outlet boundary conditions: the pressure within the system decreases since there is a leak of liquid.

Concerning the power generated by the uranium rods in the reactor core, it is set to zero at time $t = 7.9s$ to model the effect of the control rods on reducing the reactivity.
After the breach and the disappearance of liquid in the reactor core, the fuel rods cladding is not anymore in contact with the coolant. Hence, the heat exchanges are limited and this can drive to the meltdown of the rod cladding.

#### 2.3.2.2 Application of the parareal algorithm

The Parareal library is an intermediate between the solutions computed by Cathare and the corrections made by the parareal algorithm. The library collects the fine and coarse approximations from Cathare, extracts from the Cathare arrays the principal unknows (liquid and vapour velocities, pressure, enthalpies, void fraction) and stores them in other arrays belonging to the Parareal library. From there, the local initial conditions are updated with the parareal correction by combining the solutions coming from the coarse and fine propagations. Then, the updated initial conditions are copied in Cathare arrays. Before making new coarse and fine propagations the Cathare code needs auxiliary values in addition to the principal unknowns. These auxiliary values are of different natures:

- variables computed from the principal unknowns
- variables tracked during the simulation: for example, the water level in an element of the system
- boolean variables giving the flow regime in an area of the system: bubbly, annular, dispersed, seperated phases, etc.

For the first category of auxiliary variables, they can be recalculated using the updated initial conditions. However, the others are the last saved quantities before the copy of the corrected initial conditions. Hence, there is an inconsistency between the updated variables depending on the principal unknowns and the other variables that are unchanged. We show the consequences of this inconsistency by the following numerical experiment.

We denote $F^{ref}$ the target fine solution we seek to approximate with the parareal algorithm. This solution is computed in a sequential way with a finer time step than the one of the fine solver used within the parareal algorithm, denoted $F^{para}$. The propagator $F^{ref}$ replaces the exact propagator since we do not have the expression of the exact solution for this test case and will be called the reference solution in this section.
On the one hand, we initialise in the Parareal library every time window with the principal variables of the reference solution $F^{ref}$ at times $T^n, \quad n = 0, \cdots, N$. Then we transfer these initial conditions to the Cathare code to make $N$ parallel fine propagations $F^{ref}_{|[T^n,T^{n+1}]}(T^n, \Delta T, F^{ref}(T^0, T^n - T^0, u^0)(T^n))$, where $F^{ref}_{|[T^n,T^{n+1}]}$ is the restriction of the reference solution $F^{ref}$ to the time interval $[T^n, T^{n+1}]$. From this information, the Cathare code builds the Cathare array composed of principal unknowns and auxiliary variables and propagates this initial state over the time window $[T^n, T^{n+1}]$. We plot in figure 2.6 the following quantity for every time windows $[T^n, T^{n+1}]$:

$$\frac{||F^{ref}_{|[T^n,T^{n+1}]}(T^n, \Delta T, F^{ref}(T^0, T^n - T^0, u^0)(T^n)) - F^{ref}(T^0, T^n - T^0, u^0)||_{L^2}}{||F^{ref}(T^0, T^n - T^0, u^0)||_{L^2}}(T^i), \quad i = 0, \cdots, N^f \tag{2.8}$$

where $N^f = \dfrac{\Delta T}{\delta t}$ is the number of fine time steps in a time window of size $\Delta T$. This numerical experiment will allow us to see the error made on the recalculation of the auxiliary variables. If there was no error at this stage, the quantity (2.8) would be equal to zero. In figure 2.6, we see that the inconsistency between principal variables and auxiliary variables in the Cathare array leads to an additionnal error. This error prevents the parareal algorithm to recover the fine sequential

solution with the target accuracy. This target accuracy is computed in the following way:

$$\frac{max_{n=1,\cdots,N}||F^{para}(T^0, T^n - T^0, u^0) - F^{ref}(T^0, T^n - T^0, u^0)||_{L^2(\Omega)}}{max_{n=1,\cdots,N}||F^{ref}(T^0, T^n - T^0, u^0)||_{L^2(\Omega)}}$$

where $F^{para}$ is the fine solver used within the parareal algorithm. This target accuracy is thus the accuracy $\epsilon_F$ of the fine solver compared to the reference solution $F^{ref}$.

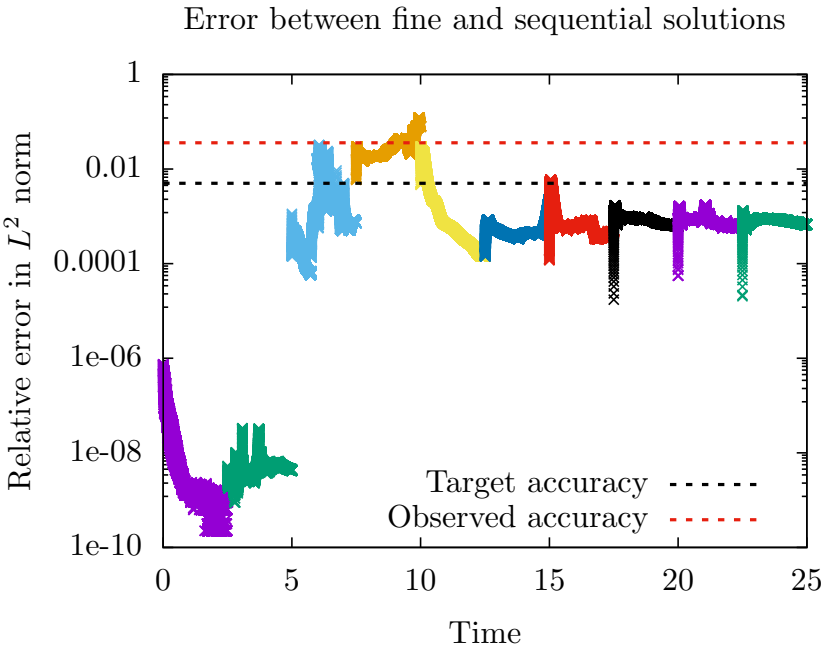


Figure 2.6: Sending only the principal variables to the Cathare code

On the other hand, the Parareal library transfers to the Cathare code the whole initial state with principal and auxiliary variables computed by the reference solution $F^{ref}(T^0, T^n - T^0, u^0)$. Then from these initial states, the Cathare code makes $N$ parallel fine propagations $F^{ref}_{|[T^n,T^{n+1}]}(T^n, \Delta T, F^{ref}(T^0, T^n - T^0, u^0)(T^n))$ spread over the $N$ time windows $[T^n, T^{n+1}]$. In figure 2.7, we plot the error (2.8) for this new configuration. In figure 2.7, we see that sending all the state to initialise the fine propagations in $[T^n, T^{n+1}]$ improves the accuracy of the fine solutions computed by the Cathare code, compared to figure 2.6 where we only initialise the time propagations with the principal variables. However, we still observe a non negligible error of about $10^{-4}$, especially starting from time $t = 5$, after the appearance of the breach in the system that occurs at time $t = 2.8$.

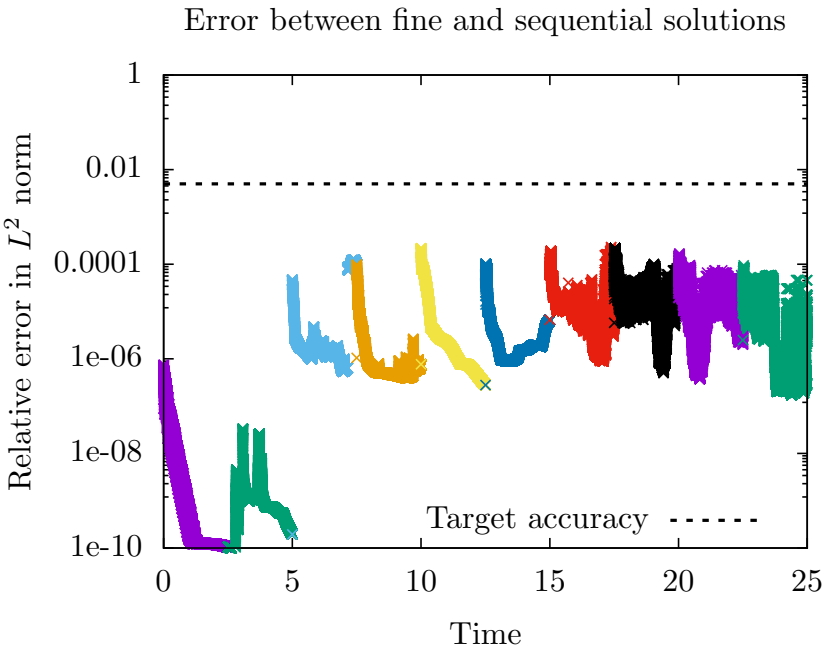


Figure 2.7: Sending all the state to the Cathare code on 10 time windows

We propose a strategy to improve the transfer of the auxiliary variables between time windows. The method consists in correcting at each parareal iteration the vector of auxiliary values in a specific way. Firstly, we need to define $V^{aux}$, the vector of auxiliary variables and $V^p$ the vector of principal variables. We distinguish:

- $V^{aux}(V^p(T^n))$ the auxiliary variables computed from the principal variables at time $T^n$.
- $V^{aux}(T^n)$ the auxiliary variables computed by the Cathare code at time $T^n$.

Secondly, we denote $T^n_-$ the point belonging to the interval $[T^{n-1}, T^n]$ and $T^n_+$ the one belonging to $[T^n, T^{n+1}]$. We propose the following correction for the auxiliary variables at time $T^n$ and $(k+1)$-th parareal iteration:

$$V^{aux}_{k+1}(T^n_+) = V^{aux}(V^p_{k+1}(T^n_+)) - V^{aux}(V^p_k(T^n_-)) + V^{aux}_k(T^n_-). \tag{2.9}$$

Since the Cathare time scheme is a two-step time scheme, we also apply a correction to the auxiliary variables at time $T^n - \delta t$.

$$V^{aux}_{k+1}(T^n - \delta t) = V^{aux}(V^p_{k+1}(T^n - \delta t)) - V^{aux}(V^p_k(T^n - \delta t)) + V^{aux}_k(T^n - \delta t). \tag{2.10}$$

This choice for the correction of the auxiliary values is motivated by the two numerical experiments 2.6 and 2.7. The figure 2.6 illustrates the error we make by computing the auxiliary variables with:

$$V^{aux}(T^n_+) = V^{aux}(V^p(T^n_+)).$$

In this configuration, the error is greater than the accuracy of the fine solver and is the main actual barrier to efficiently apply the parareal algorithm to the Cathare code in a non intrusive way.
On the other hand, the figure 2.7 illustrates the error we make by computing the auxiliary variables with:

$$V^{aux}(T^n_+) = V^{aux}(T^n_-).$$

In this case, the error is lower than the accuracy of the fine solver but is still non negligible. The correction we propose (2.9) combines the two approaches with a third term $V^{aux}(V^p(T^n))$. It is important to mention that when the parareal algorithm converges, we obtain:

$$V^{aux}(T^n_+) = V^{aux}(T^n_-),$$

since we have:

$$V^p(T^n_+) = V^p(T^n_-), \text{ and then : } V^{aux}(V^p(T^n_+)) = V^{aux}(V^p(T^n_-))$$

We have not implemented this promising strategy yet to handle specifically the auxiliary variables of the Cathare code. The actual option to reconstruct the auxiliary variables is to compute them from the principal variables. In figures 2.6 and 2.7, we see that the reconstruction error is greater than the fine solver accuracy on 10 time windows: in figure 2.6, the reconstruction error is about $4 \times 10^{-2}$ while the fine solver accuracy is $\epsilon_F = 5 \times 10^{-3}$. Hence we can not reach the fine solver accuracy with the parareal algorithm since the reconstruction error of the auxiliary variables will dominate and pollute the simulation. However, the reconstruction error of the auxiliary variables is in the order of the fine solver accuracy $\epsilon_F$ on 5 time windows. In figures 2.8 and 2.9, we make the same numerical experiment as in figures 2.6 and 2.7 on 5 time windows. We see in figure 2.8 that, in this particular case, the reconstruction of the auxiliary variables from the principal variables generates an error in the order of $\epsilon_F$ at the initial times of the 5 time windows. In figure 2.9, we observe the same behaviour as on 10 processors: sending all the state to initialise the fine propagations in $[T^n, T^{n+1}]$ improves the accuracy of the fine solutions computed by the Cathare code.
Hence, in the case of 5 processors, the reconstruction error of the auxiliary variables from the principal variables is in the order of the accuracy of the fine solver and may not pollute the parareal algorithm.

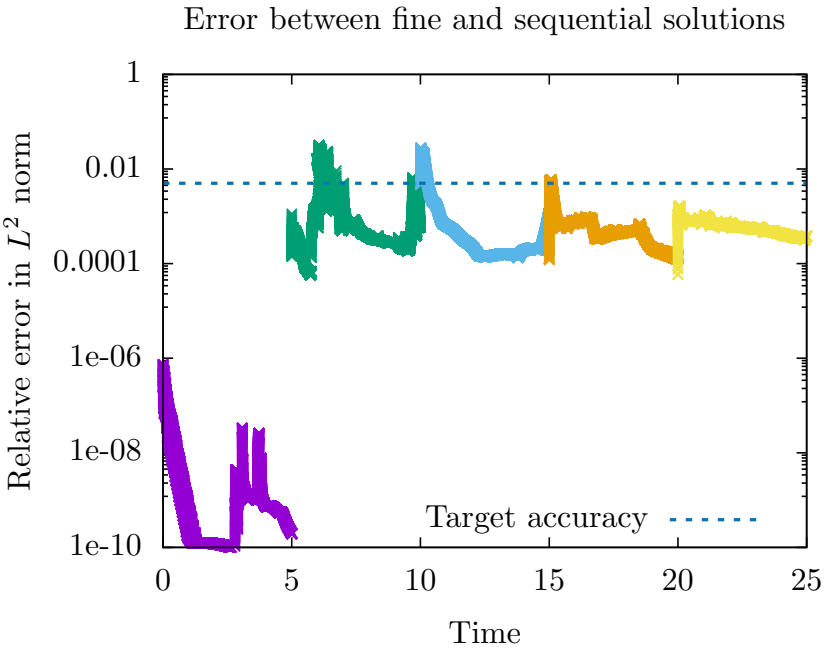


Figure 2.8: Sending only the principal variables to the Cathare code on 5 time windows

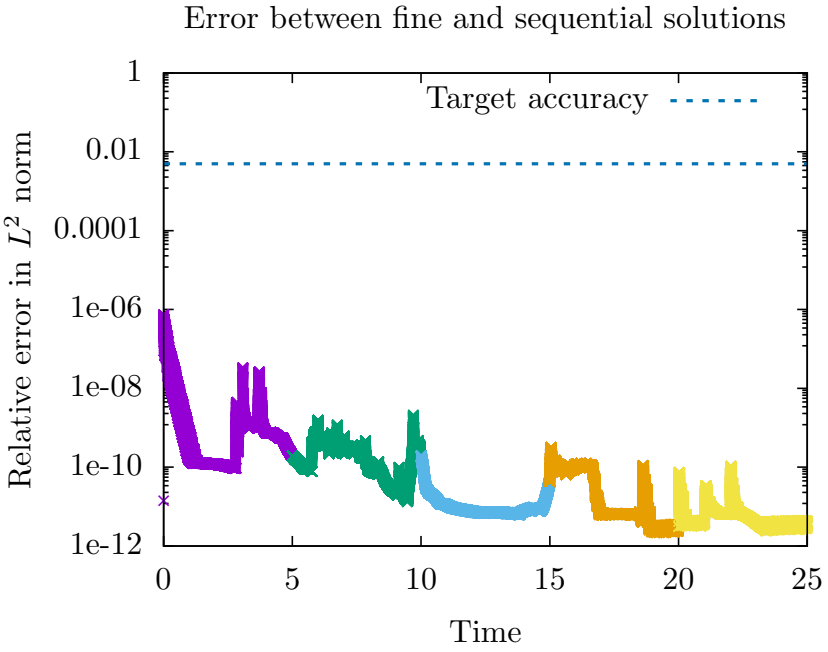


Figure 2.9: Sending all the state to the Cathare code on 5 time windows

In the sequel, we investigate the performances of the parareal algorithm when the reconstruction error is close to the fine solver accuracy to illustrate the behaviour of the parareal method when we will implement the specific treatment of the auxiliary values (2.9-2.10). Hence, we apply the parareal algorithm to the simulation of a breach in the primary circuit on 5 processors. The reference solution $F^{ref}$ is computed with a fine time step $\delta t_{ref} = 10^{-4}$. The fine solver $F^{para}$ within the parareal algorithm has a time step $\delta t = 10^{-4}$ and an accuracy $\epsilon_F = 5 \times 10^{-3}$, hence the target accuracy of the parareal algorithm is fixed to $5 \times 10^{-3}$. The coarse solver has an accuracy $\epsilon_G = 10^{-1}$ with a coarse time step $\delta T = 0.5 = 500\delta t$. We apply the full multi-step parareal (2.4) that corrects both the coarse and the fine solvers to properly initialise the time propagations. The simulation lasts 25 seconds and we split the time interval over 5 processors. In figure 2.10, we plot the following error at each time $T^n = n\Delta T$ and each parareal iteration $k$, where $n = 0, \cdots, 5$ and $\Delta T$ is the size of the time window :

$$\frac{||F^{para}(T^n, \Delta T, u_k^n) - F^{ref}(T^0, T^n - T^0, u^0)||_{L^2(\Omega)}}{||F^{ref}(T^0, T^n - T^0, u^0)||_{L^2(\Omega)}}(T^n), \quad n = 0, \cdots, 5$$

In this case, the parareal algorithm reaches the fine solver accuracy at the initial times of the 5 time windows after 3 iterations. Hence, when the reconstruction error is close to the fine solver accuracy the parareal algorithm can reach the target accuracy for the solution at times $T^n$.

In figures 2.11, 2.12 and 2.13, we see the volume fraction $\alpha_v$ in one cell of the mesh computed by the fine solver in the multi-step parareal algorithm for the three first iterations, compared to the reference solution. After the first parareal iteration, we clearly distinguish the fine propagations made by the parareal algorithm and the reference solution, over the different time windows. Then for the second and the third iteration, the reference and the fine parareal solutions are very close. We capture the dynamic of the test case with $\alpha_v \simeq 0$ at time $t = 0$ since the system is initially filled with water. Then the vapour volume fraction increases from $t = 2.8s$, time of the breach appearance until reaching its maximal value 1 at time $t = 19s$.

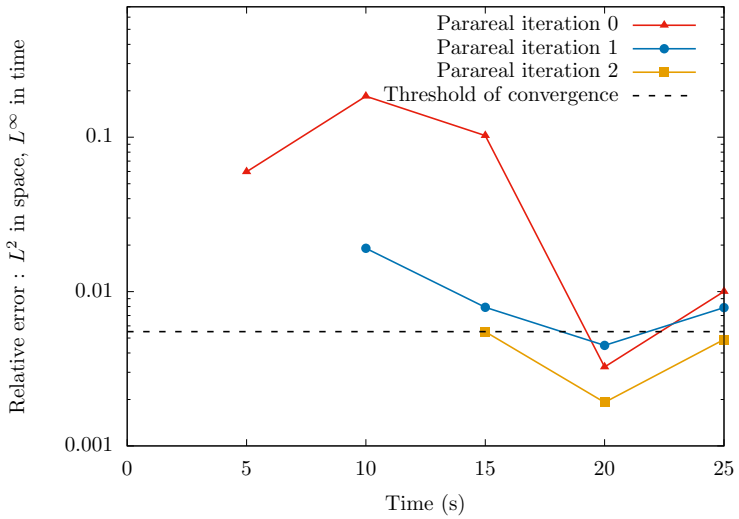


Figure 2.10: Convergence of the multi-step parareal algorithm when $\delta t = 10^{-4}$ and $\Delta T = 0.5$ for an industrial test case on 5 processors

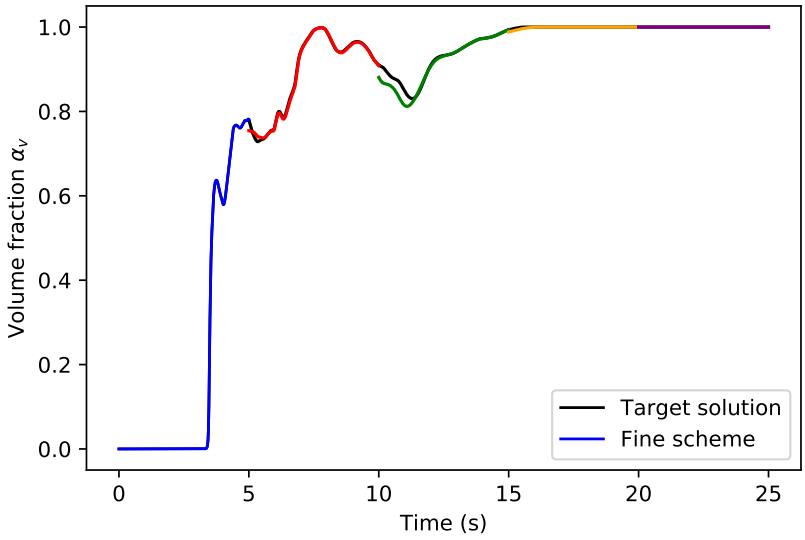


Figure 2.11: The reference solution and the fine solution after 1 parareal iteration

## 2.4 Conclusion

In this chapter, we developed two strategies to apply the parareal algorithm to the Cathare code: a numerical clone of Cathare that is restricted to one test case and a Parareal library that uses the Cathare code in a non intrusive way. The main contribution of this work has been to adapt the parareal algorithm to the architecture of the software and to its time discretisation in a non intrusive way. The results obtained with the numerical clone on the oscillating manometer show that the parareal algorithm can effectively speed-up two-phase flows simulations. These preliminary results illustrate the behaviour of the multi-step parareal algorithm on a test case that is representative of the numerical challenges for two phase flows. However, the Parareal library that uses the Cathare

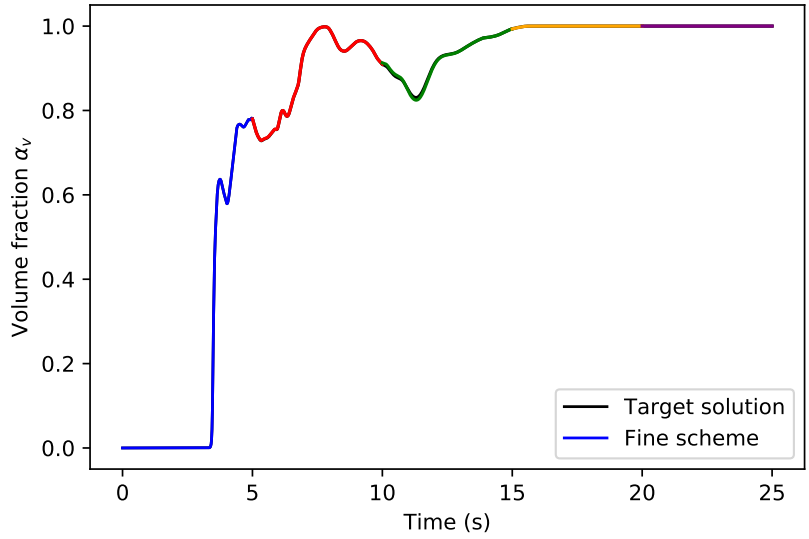


Figure 2.12: The reference solution and the fine solution after 2 parareal iterations

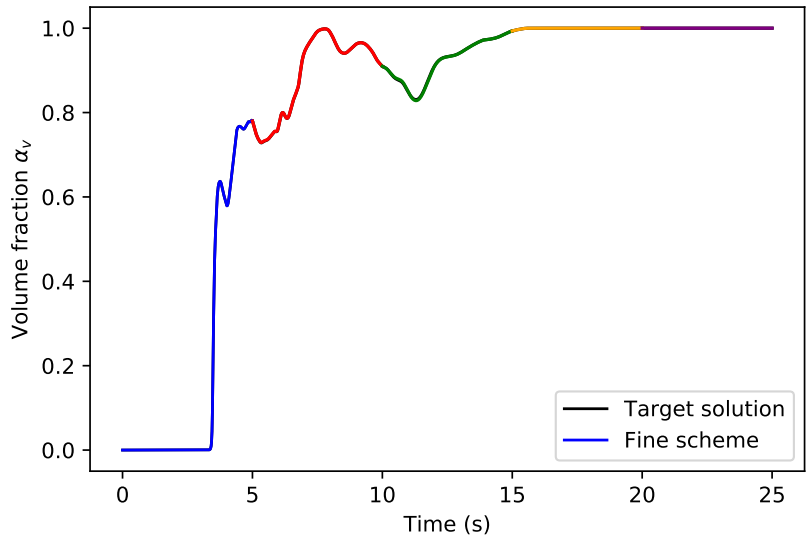


Figure 2.13: The reference solution and the fine solution after 3 parareal iterations

code as a black box suffers from a lack of accuracy. This is due to the data structure of the Cathare code that depends on principal and auxiliary variables. Hence, a reconstruction of the auxiliary values is necessary after the parareal update of the principal variables. The actual reconstruction error is greater than the target accuracy and prevents the parareal algorithm to converge towards the desired solution. We derived a strategy to accurately reconstruct the auxiliary variables by adding a parareal update specific to them and the efficiency of this method will be the subject of future works.

# Chapter 3

# Convergence analysis of the multi-step variant of the parareal algorithm

**Contents**



In this paper, we consider the problem of accelerating the numerical simulation of time dependent problems involving a multi-step time scheme by the parareal algorithm. A multi-step time scheme can potentially bring higher approximation orders than plain one-step methods but the initialisation of each time window needs to be appropriately chosen. Our main contribution is the design and analysis of an algorithm adapted to this type of discretisation without being intrusive in the coarse or fine propagators. This property allows to apply this variant of the parareal algorithm on a software as a black box and ensures the portability of the method. The parareal method is based on combining predictions made by two propagators: an accurate and expensive one used in a parallel way over the time windows and a coarse and cheaper one used in a sequential way. At convergence, the parareal algorithm provides a solution that has the fine solver's accuracy. In the classical version of parareal, the local initial condition of each time window is corrected at every iteration. When the fine and/or coarse propagators is a multi-step time scheme, we need to choose a consistent approximation of the solutions involved in the initialisation of the fine solver at each time window. Otherwise, we could loose one of the well known property of the parareal method: to recover the fine solution at the machine precision after $N$ iterations, where $N$ is the number of time windows. In this paper, we develop a variant of the algorithm that overcomes this obstacle. Thanks to this, the parareal algorithm is more coherent with the underlying time scheme and we

recover the properties of the original version. We show both theoretically and numerically that the accuracy and convergence of the multi-step variant of parareal algorithm are very competitive when we carefully choose the initialisation of each time window.

## 3.1 Introduction

Solving complex models with high accuracy and within a reasonable computing time has motivated the search for numerical schemes that exploit efficiently parallel computing architectures. In this paper, the model consists of a Partial Differential Equation (PDE) set on a domain D. In this context, one of the main ideas to parallelize a simulation is to break the problem into subproblems defined over subdomains of a partition of D. The domain can potentially have high dimensionality and be composed of different variables like space, time, velocity or even more specific variables for some problems. There exist algorithms with very good scalability properties for the decomposition of the spatial variable (see [99] or [112] for an overview) and time domain decomposition is more and more considered to complement this strategy when the speed up performances stagnates despite remaining computing resources. Research on time parallel algorithms is currently very active and has by now a history of at least 50 years (back to at least [96]) during which several algorithms have been explored (see [50] for an overview).

In this work, we report our recent effort to adapt one particular time-parallel algorithm: the parareal in time algorithm, to multi-step time schemes. The method was first introduced in [78] and has been well accepted by the community because it is easily applicable to a relatively large spectrum of problems (some specific difficulties are nevertheless encountered on certain types of PDEs as reported in [35, 47] for hyperbolic systems or [34] for hamiltonian problems). Some limitations persist for the classical version of the parareal algorithm like the parallel efficiency that decreases with the final number of iterations $K$ as $1/K$. This limitation is addressed in [80] that proposes an adaptive variant of the parareal method where the only remaining factor limiting high performance becomes the cost of the coarse solver. Without entering into very specific details of the algorithm at this stage, we can summarize the procedure by saying that we build iteratively a sequence to approximate the exact solution of the problem by a predictor-corrector algorithm. At every iteration, predictions are made by a solver which has to be as numerically inexpensive as possible since it is run on the full time interval. It usually involves coarse physics and/or coarse resolution. Corrections involve an expensive solver with high- fidelity physics and high resolution which is propagated in parallel over small time subdomains. In the classical version of parareal, the fine solver has a fixed high accuracy across all iterations. It is set to the one that we would use to solve the dynamics at the desired accuracy with a purely sequential solver. At each iteration, the local initial conditions are corrected for every time windows until convergence. Multi-step time schemes require several previous steps to compute the solution at a new point in time. When the fine and/or coarse propagators is a multi-step time scheme, we need to choose a consistent approximation of the solutions at previous steps involved in the initialisation of the fine and/or coarse solver at each time windows. Otherwise, the initialisation error will prevent the parareal algorithm to converge towards the solution with fine solver's accuracy. This point was addressed in the context of multigrid in time method in [46, 45]. Here, the authors adapt the MGRIT algorithm framework to the use of multi-step time schemes, the BDF methods. In this paper, we propose a variant of the algorithm that overcome this obstacle. Thanks to this, the parareal algorithm is more coherent with the underlying time scheme and we recover the properties of the original version.

We present in section 2 the variant of the parareal algorithm adapted to multi-step time schemes. This method includes additional corrections at previous steps involved in the intialisation of the fine and/or coarse solver at each time window. This choice has the benefit to be non intrusive into the code we seek to parallelise by a time domain decomposition.

In the last section, we illustrate the performance of the algorithm on numerical examples: the damped oscillator and the Brusselator. We show that this variant allows the parareal algorithm to converge towards the solution with fine solver's accuracy.

## 3.2 A multi-step variant of the parareal algorithm

In this section, after introducing some preliminary notations in section 2.1, we formulate the new variant of the parareal algorithm adapted to multi-step time schemes (section 2.2). We then present the hypothesis we consider in this article and restrict ourselves to two-step time schemes for the convergence analysis (section 2.3). We prove that the multi-step variant converges with a convergence rate similar to that of the classical parareal algorithm. Finally, we discuss how the new paradigm can be generalised to multi-step time schemes, not only two-step times schemes, used in the fine and/or the coarse solver (section 2.4).

### 3.2.1 Setting and preliminary notations

Let $\mathbb{U}$ be a Banach space of functions defined over a domain $\Omega \subset \mathbb{R}^d$ ($d \geq 1$). Let

$$S : [0,T] \times [0,T] \times \mathbb{U} \to \mathbb{U} \tag{3.1}$$

be a solver, that is, an operator such that, for any given time $t \in [0,T]$, $s \in [0, T-t]$ and any function $w \in \mathbb{U}$ takes an initial value at time $t$ and propagates it at time $t+s$. We further assume that $S$ is defined through the discretisation of the time-dependent problem:

$$\begin{cases} u'(t) + \mathcal{A}(t, u(t)) = 0, \quad t \in [0,T] \\ u(0) \in \mathbb{U} \end{cases} \tag{3.2}$$

where $\mathcal{A}$ is a locally Lipschitz operator from $[0,T] \times \mathbb{U}$. This ODE system can also be obtained from the discretisation of a PDE in space with $\mathcal{A} : \mathbb{R} \times \mathbb{R}^{\mathcal{N}} \to \mathbb{R}^{\mathcal{N}}$ and $\mathcal{N}$ denotes the number of degrees of freedom. We seek to approximate the solution of problem (3.2) at a given target accuracy by a solver $S$. We denote $\varepsilon(t,s,w)$ the propagator giving the exact solution of system (3.2), for any initial value $w \in \mathbb{U}$, any $t \in [0,T]$ and any $s \in [0, T-t]$. Thus, $S(t,s,w)$ approximates $\varepsilon(t,s,w)$ with an accuracy $\eta > 0$ such that we have:

$$||\delta S(t,s,w)|| = ||\varepsilon(t,s,w) - S(t,s,w)|| \leq \eta s(1 + ||w||), \forall t \in [0,T], s \in [0, T-t], w \in \mathbb{U} \tag{3.3}$$

where $||\cdot||$ denotes the norm in $\mathbb{U}$.
The choice of the solver $S$ determines the quality of the approximation and the computational cost of its implementation. One can potentially bring higher approximation orders than plain one-step methods by using a multi-step time discretisation method. Multi-step time schemes require several previous steps to compute the solution at a new point in time. Hence, the notation (3.1) does not hold when the solver is based on a multi-step time scheme. Here and in the following, we consider only two-step time schemes and we will use the following notation:

$$S : [0,T] \times [0,T] \times \mathbb{U} \times \mathbb{U} \to \mathbb{U} \tag{3.4}$$

such that $S(t, s, w^1, w^2)$ for any given time $t \in [0, T]$, $s \in [0, T-t]$ and any functions $w^1, w^2 \in \mathbb{U}$ takes two initial values at times $t$ and $t - \delta t$ and propagates them at time $t + s$, where $\delta t$ is the time step of the two-step time scheme underlying in $S$.

### 3.2.2 A multi-step variant of the parareal algorithm

We consider a given decomposition of the time interval $[0, T]$ into $N$ subintervals $[T^n, T^{n+1}]$, $n = 0, \cdots, N-1$. Without loss of generality, we will take them of uniform size $\Delta T = T/N$ which means that $T^n = n\Delta T$ for $n = 0, \cdots, N$. For a given target accuracy $\eta > 0$, the goal of the parareal algorithm is to accelerate the computation of an approximation $\tilde{u}(T^n)$ of $u(T^n)$ such that:

$$max_{1 \leq n \leq N} ||u(T^n) - \tilde{u}(T^n)|| \leq \eta$$

The classical way to compute such an approximation is to set $\tilde{u}(T^n) = S_{seq}(0, T^n, u(0)), 1 \leq n \leq N$, where $S_{seq}$ is some sequential solver in $[0, T]$. On the other hand, the strategy of the parareal algorithm follows the following steps, using two propagation operators:

- $G(T^n, \Delta T, u^n)$ computes a coarse approximation of $u(T^{n+1})$ with initial condition $u(T^n) \simeq u^n$. The coarse propagation is sequential but have a low computational cost.
- $F(T^n, \Delta T, u^n)$. computes a more accurate approximation of $u(T^{n+1})$ with initial condition $u(T^n) \simeq u^n$. The action of $F$ is distributed over $N$ time windows and $N$ processors solve over each interval $[T^n, T^{n+1}]$ of size $\Delta T$ instead of solving over $[0, T]$.

In the sequel, we analyse the convergence rate of the multi-step variant of parareal algorithm when the coarse solver is a one-step time scheme and the fine one is a two-step time scheme.
**Hypotheses (H)**: There exists $\epsilon_G, C_d, C > 0$ such that for any functions $x, y \in \mathbb{U}$ and for any $t \in [0, T]$ and $s \in [0, T-t]$,

$$||\varepsilon(t, s, x) - G(t, s, x)|| \leq s(1 + ||x||)\epsilon_G \Leftrightarrow ||\delta G(t, s, x)|| \leq s\epsilon_G(1 + ||x||) \tag{3.5}$$

$$||G(t, s, x) - G(t, s, y)|| \leq (1 + Cs)||x - y|| \tag{3.6}$$

$$||F(t, s, x_1, y_1) - F(t, s, x_2, y_2)|| \leq (1 + Cs)\left(||x_1 - x_2|| + ||y_1 - y_2||\right) \tag{3.7}$$

$$||\delta G(t, s, x) - \delta G(t, s, y)|| \leq C_d s\epsilon_G ||x - y|| \tag{3.8}$$

$$|| \left(F(t, s, \varepsilon(t, -\delta t, y_1), y_1) - \varepsilon(t, s, y_1)\right) - \left(F(t, s, \varepsilon(t, -\delta t, y_2), y_2) - \varepsilon(t, s, y_2)\right) || \leq Cs\delta t||y_1 - y_2|| \tag{3.9}$$

$$\begin{aligned} ||(F(t, s - \delta t, y_1 - \delta_1, y_1) &- F(t, s - \delta t, y_2 - \delta_2, y_2)) - (F(t, s, y_1 - \delta_1, y_1) - F(t, s, y_2 - \delta_2, y_2))|| \\ &\leq (1 + Cs)\delta t||\delta_1 - \delta_2|| + (1 + Cs)\delta t||y_1 - y_2|| \end{aligned} \tag{3.10}$$

$$||F(t, s, \varepsilon(t, -\delta t, y), y) - \varepsilon(t, s, y)|| \leq s\epsilon_F(1 + ||y||) \tag{3.11}$$

Note that the hypothesis (3.5)-(3.8) are the classical properties of numerical schemes related to stability and accuracy. Hypotheses (3.6) and (3.7) are Lipschitz conditions and the quantity $\epsilon_G$ is a small constant which, in the case of the explicit Euler scheme, would be proportionnal to the time step size. Hypotheses (3.9) and (3.10) are specific to two-step time schemes. There are two sources of error for two-step time schemes:

- the error from the discretisation of the time derivative, common to one-step time schemes.
- the error from the inconsistency between the two initial conditions $x_1$ and $y_1$ in $F(t, s, x_1, y_1)$.

In hypothesis (3.9), assuming the term $x_1 = \varepsilon(t, -\delta t, y_1), y_1)$ is defined, there is no inconsistency between the two initial values since $x_1$ is computed with the exact propagator starting from $y_1$. Hence, the only remaining errors are:

- the difference between $y_1$ and $y_2$.
- the error from the time propagation over a time window of size $s$.
- the error between the fine and the exact propagators that is proportionnal to the fine time step $\delta t$.

On the other hand, we assume hypothesis (3.10) holds for $s \geq \Delta T$, the time window size. This hypothesis includes the inconsistency between the two initial values and is denoted $\delta_1$ and $\delta_2$. Hence, we describe here the errors coming from:

- the inconsistency $\delta_i$ between the two initial values of the fine solver.
- the difference between the principal initial values $y_1$ and $y_2$.
- the time propagation over time windows of size $s$.

**Example 3.1.** *Here, we illustrate the validity of hypothesis (3.10) on a simple linear ODE. The parameters involved in hypothesis (3.10) are: $s$, $y_1$, $y_2$, $\delta_1$ and $\delta_2$. In the proof of convergence, we apply this hypothesis for $y_1 = u_{k-1}^n$, the parareal solution at $k-1$ iteration and time $T^n$, and $y_2 = u(T^n)$, the exact solution at time $T^n$, hence these two parameters are very close. On the other hand, $\delta_1 = u_{k-1}^n - u_{k-1}^{n-1,N^f-1}$, where $u_{k-1}^{n-1,N^f-1}$ is the parareal solution at $(k-1)$ iteration and time $T^n - \delta t$, and $\delta_2 = u(T^n) - u(T^n - \delta t)$, where $u(T^n - \delta t)$ is the exact solution at time $T^n - \delta t$ and $s$ is equal to the time window size.*

$$\begin{cases} y'(t) = y(t), \quad t \in [0, T] \\ y^{0a}, y^{0b} \text{ given} \end{cases}$$

*where $y^{0a}, y^{0b}$ are the two seed values to initialise the time propagation with the second-order BDF method:*

$$y^{0a} = (y_1 - \delta_1) - (y_2 - \delta_2), \quad y^{0b} = y_1 - y_2$$

*We solve this system by a second-order BDF method:*

$$\frac{3}{2}y^{n+1} - 2y^n + \frac{1}{2}y^{n-1} = \delta t y^{n+1} \tag{3.12}$$

*From (3.12), we have the expression of the numerical solution $y^n$ for $n = 0, \cdots, N^f$ with $N^f = \dfrac{T}{\delta t}$:*

$$y^n = \alpha r_1^n + \beta r_2^n$$

*such that:*

- $r_1 = \dfrac{2 + \sqrt{1 + 2\delta t}}{3 - 2\delta t} = 1 + \delta t + \mathcal{O}(\delta t^2)$
- $r_2 = \dfrac{2 - \sqrt{1 + 2\delta t}}{3 - 2\delta t} = \dfrac{1}{3} - \dfrac{\delta t}{9} + \mathcal{O}(\delta t^2)$.
  *In (3.12), the term $r_2^n$ tends rapidly to zero when $n$ goes to infinity. Thus we neglect its contribution.*

- $\alpha = \dfrac{r_2(\delta_1 - \delta_2) + (1 - r_2)(y_1 - y_2)}{r_1 - r_2}$
- $\beta = \dfrac{(r_1 - 1)(y_1 - y_2) - r_1(\delta_1 - \delta_2)}{r_1 - r_2}$

*In the linear case, we can write hypothesis (3.10):*

$$||y^{N+1} - y^N|| \leq (1 + Cs)\delta t||y^{0a} - y^{0b}|| + (1 + Cs)\delta t||y^{0b}||$$

*where:* $y^N = F(0, s - \delta t, y^{0a}, y^{0b}), \quad y^{N+1} = F(0, s, y^{0a}, y^{0b})$ *and* $(N + 1)$ *is the number of fine time steps in a time window of size* $s$*:* $N + 1 = \dfrac{s}{\delta t} = \dfrac{\Delta T}{\delta t}$*. From the expression of* $y^n$ *in (3.12):*

$$y^{N+1} - y^N = \alpha r_1^N(r_1 - 1) + \beta r_2^N(r_2 - 1)$$

*Neglecting the term* $r_2^N$*, we obtain:*

$$y^{N+1} - y^N = \frac{r_2}{r_1 - r_2}(1 + \Delta T)\delta t(\delta_1 - \delta_2) + \frac{1 - r_2}{r_1 - r_2}(1 + \Delta T)\delta t(y_1 - y_2) + \mathcal{O}(\delta t^2)$$

*Hence, the second-order BDF method verify hypothesis (3.10).*

In the following example, we explain the problem of initialising the fine solver in a time window when the fine propagator is a two-step time scheme:

**Example 3.2.** *If one solves (3.2) with a multi-step time scheme as fine propagator* $F$ *like the second-order BDF method:*

$$\frac{3}{2}u^{j+1} - 2u^j + \frac{1}{2}u^{j-1} = -\delta t\mathcal{A}(u^{j+1}, t^{j+1}), \quad j = 1, \cdots, N^f, t^{j+1} - t^j = \delta t$$

*Here the fine solver reads as:* $u^{j+1} = F(t^j, \delta t, u^{j-1}, u^j)$*. Now, we apply the parareal algorithm with a coarse grid:* $T^0, \cdots, T^N$ *where:*
$T^{n+1} - T^n = \Delta T = N^f \delta t$.
*Then we can write:* $u(T^n + j\delta t) \simeq u^{n,j}, j = 1, \cdots, N^f, n = 1, \cdots, N^c$.
*In order to perform the fine propagation, in a given time window* $[T^n, T^{n+1}]$*, we only need the local initial condition* $u_k^n$ *and a consistent approximation of* $u(T^n - \delta t)$*.*

In [12], the authors propose a consistent approximation in the context of the simulation of molecular dynamics. The proposed method was linked to the nature of the model and the symplectic character of their algorithm is shown, which is an important property to verify for molecular dynamics.
We now detail our algorithm:

$$\begin{cases} u_0^{n+1} &= G(T^n, \Delta T, u_0^n), \quad 0 \leq n \leq N - 1 \\ u_0^{n,N^f-1} &= u_0^{n+1}, \quad 0 \leq n \leq N - 1 \\ u_{k+1}^{n+1} &= G(T^n, \Delta T, u_{k+1}^n) + F(T^n, \Delta T, u_k^{n-1,N^f-1}, u_k^n) \\ & \quad -G(T^n, \Delta T, u_k^n), \quad 0 \leq n \leq N - 1, \quad k \geq 0 \\ u_{k+1}^{n,N^f-1} &= F(T^n, \Delta T - \delta t, u_k^{n-1,N^f-1}, u_k^n) + u_{k+1}^{n+1} \\ & \quad -F(T^n, \Delta T, u_k^{n-1,N^f-1}, u_k^n), \quad 0 \leq n \leq N - 1, \quad k \geq 0 \end{cases} \tag{3.13}$$

At this point, several comments are in order. To derive a consistent approximation of $u(T^n - \delta t)$, we use the only fine trajectory at our disposal which is $F(T^{n-1}, \Delta T, u_k^{n-2,N^f-1}, u_k^{n-1})$. Its final value at $T^n$ is:
$F(T^{n-1}, \Delta T, u_k^{n-2,N^f-1}, u_k^{n-1})(T^n)$ from which we compute $u_{k+1}^n$ by the parareal correction. Hence, we translate the solution:
$F(T^{n-1}, \Delta T - \delta t, u_k^{n-2,N^f-1}, u_k^{n-1})(T^n - \delta t)$ by the same correction:
$u_{k+1}^n - F(T^{n-1}, \Delta T, u_k^{n-2,N^f-1}, u_k^{n-1})$ and obtain the so called consistent approximation $u_{k+1}^{n-1,N^f-1}$ to initialize the fine propagation in $[T^n, T^{n+1}]$.
Moreover, an important feature of this new algorithm is to preserve a well known property of the parareal algorithm:

$$u_k^n = F(T^0, T^n - T^0, u^0), \text{ for: } k \geq n, \quad n = 0, \cdots, N \tag{3.14}$$

This comes from the term:

$$G(T^n, \Delta T, u_{k+1}^n) - G(T^n, \Delta T, u_k^n)$$

that is equal to zero when $k \geq n, \quad n = 0, \cdots, N$.
In our case, the multi-step variant of the parareal algorithm verifies (3.14) and the additionnal correction of the solution at time $T^n - \delta t$ leads to:

$$\begin{aligned} u_{k+1}^{n,N^f-1} &= F(T^n, \Delta T - \delta t, u_k^{n-1,N^f-1}, u_k^n) + u_{k+1}^{n+1} \\ &\quad -F(T^n, \Delta T, u_k^{n-1,N^f-1}, u_k^n), \quad 0 \leq n \leq N-1, \quad k \geq 0 \\ &= F(T^n, \Delta T - \delta t, u_k^{n-1,N^f-1}, u_k^n) + G(T^n, \Delta T, u_{k+1}^n) \\ &\quad -G(T^n, \Delta T, u_k^n), \quad 0 \leq n \leq N-1, \quad k \geq 0 \end{aligned}$$

Hence, the multi-step parareal method satisfies the same property (3.14) at time $T^n - \delta t$:

$$u_k^{n,N^f-1} = F(T^0, T^n - \delta t - T^0, u^0), \text{ for: } k \geq n, \quad n = 0, \cdots, N \tag{3.15}$$

The convergence result of theorem 3.3 and its proof are helpful to understand the main mechanisms driving the convergence of the algorithm and explaining its behavior. To present it, we introduce the shorthand notation for the error norm:

$$E_k^n := u_k^n - \varepsilon(T^0, T^n - T^0, u^0), k \geq 0, 0 \leq n \leq N,$$

We introduce the following quantities:

$$\left\{ \begin{array}{lcl} \alpha & := & C_d \epsilon_G \Delta T \\ \mu & := & C \Delta T \delta t \\ \beta & := & 1 + C_c \Delta T \\ \gamma_G & := & \Delta T \epsilon_G max_{0 \leq n \leq N}(1 + ||u(T^n)||) \\ \gamma_F & := & \Delta T \epsilon_F max_{0 \leq n \leq N}(1 + ||u(T^n)||) \end{array} \right. \tag{3.16}$$

as shorthand notations for the proof of convergence.

**Theorem 3.3** (Convergence of the multi-step parareal algorithm)**.** *Let $G$, $F$ and $\delta G$ satisfy Hypotheses (3.5)–(3.11). Let $k \geq 0$ be any given positive integer. If the time step $\delta t$ of the fine solver verifies:*

$$\delta t \leq \Delta T^2 \epsilon_G^2, \tag{3.17}$$

*then the* $(u_k^n)_n$ *of the multi-step parareal scheme (3.13) satisfy:*

$$\begin{cases} max_{0\leq n\leq N}||u_0^n - u(T^n)|| & \leq & \frac{\tilde{\gamma}_G}{\gamma_G}e^{CT}max_n(1+||u(T^n)||)Te^{-C\Delta T}\epsilon_G, \quad n\geq 1 \\ max_{0\leq n\leq N}||u_k^n - u(T^n)|| & \leq & \lambda\frac{\tilde{\tau}^{k+1}}{k+1!}\left(\frac{f_k}{2^{k+1}}\frac{\tilde{\gamma}_G}{\gamma_G}\left(\frac{\tilde{\alpha}}{\alpha}\right)^k + \frac{k+1}{\tau}\frac{f_{k-1}}{2^{k+1}}\frac{\tilde{\gamma}_F}{\gamma_G}\left(\frac{\tilde{\alpha}}{\alpha}\right)^{k-1}\right), \quad n\geq k+1, \quad k\geq 1 \end{cases} \tag{3.18}$$

*where:*

$$\lambda = \frac{e^{CT}max_{0\leq n\leq N}(1+||u(T^n)||)}{C_d}, \quad \tilde{\tau} = 2\tau = 2C_dTe^{-C\Delta T}\epsilon_G, \quad f_k = \frac{(1+\sqrt{5})^{k+1}-(1-\sqrt{5})^{k+1}}{2^{k+1}\sqrt{5}}.$$

*and* $\tilde{\alpha}$*,* $\tilde{\gamma}_G$ *and* $\tilde{\gamma}_F$ *are perturbations of the coefficients* $\alpha$*,* $\gamma_G$ *and* $\gamma_F$ *respectively, such that:*

$$\frac{\tilde{\gamma}_G}{\gamma_G} = 1+\mathcal{O}(\Delta T\epsilon_G), \quad \frac{\tilde{\alpha}}{\alpha} = 1+\mathcal{O}(\Delta T\epsilon_G), \quad \frac{\tilde{\gamma}_F}{\gamma_G} = \mathcal{O}(\Delta T\epsilon_G)$$

Let us make a couple of remarks before giving the proof of the theorem. First, the convergence rate of the multi-step parareal algorithm is similar to the one of the classical parareal algorithm with the factor $\frac{\tilde{\tau}^{k+1}}{k+1!}$, since in the classical version the convergence rate is $\frac{\tau^{k+1}}{k+1!}$. The remaining factors $\frac{\tilde{\gamma}_G}{\gamma_G}$, $\left(\frac{\tilde{\alpha}}{\alpha}\right)^k$ and $\frac{\tilde{\gamma}_F}{\gamma_G}$ are close to 1 and their contributions are negligible. The term $f_k$ is specific to the multi-step variant and have the following asymptotic behaviour when $k$ tends to infinity:

$$\frac{f_k}{2^{k+1}} \underset{k\to+\infty}{\sim} \left(\frac{1+\sqrt{5}}{4}\right)^{k+1}$$

*Proof.* The proof is in the spirit of existing results from the litterature [78, 14, 53].
If $k=0$, using definition (3.13) for $u_0^n$, we have for $0\leq n\leq N-1$,

$$\begin{aligned} E_0^{n+1} &= u_0^{n+1} - \varepsilon(T^0, T^{n+1}-T^0, u^0) \\ E_0^{n+1} &= G(T^n, \Delta T, u_0^n) - \varepsilon(T^n, \Delta T, u(T^n)) \\ ||E_0^{n+1}|| &\leq ||G(T^n, \Delta T, u_0^n) - G(T^n, \Delta T, u(T^n))|| + ||G(T^n, \Delta T, u(T^n)) - \varepsilon(T^n, \Delta T, u(T^n))|| \\ &\leq (1+C\Delta T)||E_0^n|| + \Delta T\epsilon_G(1+||u(T^n)||) \\ &\leq \beta||E_0^n|| + \gamma_G \end{aligned}$$

where we have used (3.5) and (3.6) to derive the second to last inequality.
For $k\geq 1$, starting from (3.13), we have

$$\begin{aligned} E_k^{n+1} &= u_k^{n+1} - \varepsilon(T^0, T^{n+1}-T^0, u^0) \\ &= G(T^n, \Delta T, u_k^n) + F(T^n, \Delta T, u_{k-1}^{n-1,N^f-1}, u_{k-1}^n) - G(T^n, \Delta T, u_{k-1}^n) - \varepsilon(T^n, \Delta T, u(T^n)) \end{aligned}$$

In the sequel, we add and substract the following quantites to $E_k^{n+1}$:

- $G(T^n, \Delta T, u(T^n))$ and $\varepsilon(T^n, \Delta T, u_{k-1}^n)$
- $\varepsilon(T^n, \Delta T, u(T^n))$ and $F(T^n, \Delta T, u(T^n)-\delta x, u(T^n))$

- $F(T^n, \Delta T, u_{k-1}^n - \delta x, u_{k-1}^n)$ and $F(T^n, \Delta T, u_{k-1}^n - \hat{\delta x}, u_{k-1}^n)$

where:

$$\delta x = u(T^n) - u(T^n - \delta t), \quad \tilde{\delta x} = u_{k-1}^n - u_{k-1}^{n-1,N^f-1}, \quad \hat{\delta x} = u_{k-1}^n - \varepsilon(T^n, -\delta t, u_{k-1}^n)$$

$$\begin{aligned} E_k^{n+1} &= G(T^n, \Delta T, u_k^n) - G(T^n, \Delta T, u(T^n)) + \delta G(T^n, \Delta T, u(T^n)) - \delta G(T^n, \Delta T, u_{k-1}^n) \\ &+ F(T^n, \Delta T, \varepsilon(T^n, -\delta t, u_{k-1}^n), u_{k-1}^n) - \varepsilon(T^n, \Delta T, u_{k-1}^n) \\ &- (F(T^n, \Delta T, \varepsilon(T^n, -\delta t, u(T^n)), u(T^n)) - \varepsilon(T^n, \Delta T, u(T^n))) \\ &+ F(T^n, \Delta T, u_{k-1}^n - \delta x, u_{k-1}^n) - F(T^n, \Delta T, u_{k-1}^n - \hat{\delta x}, u_{k-1}^n) \\ &+ F(T^n, \Delta T, u_{k-1}^n - \tilde{\delta x}, u_{k-1}^n) - F(T^n, \Delta T, u_{k-1}^n - \delta x, u_{k-1}^n) \\ &+ F(T^n, \Delta T, u(T^n - \delta t), u(T^n)) - \varepsilon(T^n, \Delta T, u(T^n)) \end{aligned}$$

Taking norms and using (3.6), (3.7), (3.8), (3.9), (3.11), we derive:

$$\begin{aligned} ||E_k^{n+1}|| \leq (1 + C\Delta T)||E_k^n|| + C\Delta T \epsilon_G ||E_{k-1}^n|| + C\Delta T \delta t ||E_{k-1}^n|| + C||\hat{\delta x} - \delta x|| + C||\delta x - \tilde{\delta x}|| \\ + \Delta T \epsilon_F (1 + ||u(T^n)||) \end{aligned}$$

On the one hand, the term $\delta x - \tilde{\delta x}$ becomes:

$$\delta x - \tilde{\delta x} = u_{k-1}^{n-1,N^f-1} - u(T^n - \delta t) - (u_{k-1}^n - u(T^n)) = E_k^{n,N^f-1} - E_k^{n+1} = \delta E_k^{n+1}$$

On the other hand, we derive a bound for the term: $||\delta x - \hat{\delta x}||$:

$$||\delta x - \hat{\delta x}|| = ||u(T^n - \delta t) - \varepsilon(T^n, -\delta t, u_{k-1}^n) - (u(T^n) - u_{k-1}^n)||$$

Writing the Taylor expansions of $u(T^n - \delta t)$ and $\varepsilon(T^n, -\delta t, u_{k-1}^n)$ around $T^n$ and $u_{k-1}^n$ respectively, we obtain formally:

$$u(T^n - \delta t) - u(T^n) = \delta t \mathcal{A}(T^n, u(T^n)) + \frac{\delta t^2}{2}\left(\frac{\partial \mathcal{A}}{\partial t} + \frac{\partial \mathcal{A}}{\partial u}\mathcal{A}\right)(T^n, u(T^n)) + \mathcal{O}(\delta t^3)$$

$$\varepsilon(T^n, -\delta t, u_{k-1}^n) - u_{k-1}^n = \delta t \mathcal{A}(T^n, u_{k-1}^n) + \frac{\delta t^2}{2}\left(\frac{\partial \mathcal{A}}{\partial t} + \frac{\partial \mathcal{A}}{\partial u}\mathcal{A}\right)(T^n, u_{k-1}^n) + \mathcal{O}(\delta t^3)$$

Hence, assuming the operator $\mathcal{A}$ and its derivatives $\dfrac{\partial \mathcal{A}}{\partial t}$, $\dfrac{\partial \mathcal{A}}{\partial u}$ are locally Lipschitz:

$$||\delta x - \hat{\delta x}|| \leq \left(C\delta t + \frac{C\delta t^2}{2}\right)||E_{k-1}^n|| + C\delta t^3$$

We recall: $\gamma_F = \Delta T \epsilon_F max_{0 \leq n \leq N}(1 + ||u(T^n)||)$. Since, the fine solver is based on a two-step time then $\epsilon_F \approx \delta t^2$. Hence, we neglect in the sequel the contribution $C\delta t^3$:

$$||E_k^{n+1}|| \leq \beta ||E_k^n|| + (\alpha + \mu + C\delta t + \frac{C\delta t^2}{2})||E_{k-1}^n|| + C||\delta E_{k-1}^n|| + \gamma_F$$

In the sequel, we derive an upper bound for the error terms $E_k^{n,N^f-1}$ and $\delta E_k^{n+1} = E_k^{n,N^f-1} - E_k^{n+1}$.

$$\delta E_k^{n+1} = E_k^{n,N^f-1} - E_k^{n+1}$$

$$\delta E_k^{n+1} = u_k^{n,N^f-1} - \varepsilon(T^0, T^{n+1} - \delta t - T^0, u^0) - u_k^{n+1} + \varepsilon(T^0, T^{n+1} - T^0, u^0)$$

In the sequel, we add and substract the following quantites to $\delta E_k^{n+1}$:

- $F(T^n, \Delta T - \delta t, u(T^n) - \delta x, u(T^n))$
- $F(T^n, \Delta T, u(T^n) - \delta x, u(T^n))$

$$\begin{aligned}
\delta E_k^{n+1} &= F(T^n, \Delta T - \delta t, u(T^n) - \delta x, u(T^n)) - \varepsilon(T^n, \Delta T - \delta t, u(T^n)) \\
&- (F(T^n, \Delta T, u(T^n) - \delta x, u(T^n)) - \varepsilon(T^n, \Delta T, u(T^n))) \\
&+ F(T^n, \Delta T - \delta t, u_{k-1}^n - \tilde{\delta x}, u_{k-1}^n) - F(T^n, \Delta T - \delta t, u(T^n) - \delta x, u(T^n)) \\
&- \Big(F(T^n, \Delta T, u_{k-1}^n - \tilde{\delta x}, u_{k-1}^n) - F(T^n, \Delta T, u(T^n) - \delta x, u(T^n))\Big)
\end{aligned}$$

Taking norms and using (3.10), (3.11), we derive:

$$\begin{aligned}
||\delta E_k^{n+1}|| &\leq 2\Delta T \epsilon_F (1 + ||u(T^n)||) + C\delta t ||u_{k-1}^{n-1,N^f-1} - u(T^n - \delta t) - (u_{k-1}^n - u(T^n))|| \\
&+ C\delta t ||u_{k-1}^n - u(T^n)|| \\
||\delta E_k^{n+1}|| &\leq C\delta t ||\delta E_{k-1}^n|| + C\delta t ||E_{k-1}^n|| + 2\gamma_F
\end{aligned}$$

$$\begin{aligned}
E_k^{n,N^f-1} &= u_k^{n,N^f-1} - \varepsilon(T^0, T^{n+1} - \delta t - T^0, u^0) \\
&= F(T^n, \Delta T - \delta t, u_{k-1}^{n-1,N^f-1}, u_{k-1}^n) + u_k^{n+1} - F(T^n, \Delta T, u_{k-1}^{n-1,N^f-1}, u_{k-1}^n) - \varepsilon(T^n, \Delta T - \delta t, u(T^n))
\end{aligned}$$

In the sequel, we add and substract the same quantites to $E_k^{n,N^f-1}$ as those for the term $\delta E_k^{n+1}$.

$$\begin{aligned}
E_k^{n,N^f-1} &= u_k^{n+1} - \varepsilon(T^n, \Delta T, u(T^n)) + F(T^n, \Delta T - \delta t, u(T^n) - \delta x, u(T^n)) - \varepsilon(T^n, \Delta T - \delta t, u(T^n)) \\
&- (F(T^n, \Delta T, u(T^n) - \delta x, u(T^n)) - \varepsilon(T^n, \Delta T, u(T^n))) \\
&+ F(T^n, \Delta T - \delta t, u_{k-1}^n - \tilde{\delta x}, u_{k-1}^n) - F(T^n, \Delta T - \delta t, u(T^n) - \delta x, u(T^n)) \\
&- \Big(F(T^n, \Delta T, u_{k-1}^n - \tilde{\delta x}, u_{k-1}^n) - F(T^n, \Delta T, u(T^n) - \delta x, u(T^n))\Big)
\end{aligned}$$

Taking norms and using (3.10), (3.11), we derive:

$$||E_k^{n,N^f-1}|| \leq \beta ||E_k^n|| + (C + C\delta t)||\delta E_{k-1}^n|| + (\alpha + \mu + 2C\delta t + \frac{C\delta t^2}{2})||E_{k-1}^n|| + 3\gamma_F$$

We summarise the obtained inequalities:

$$||E_k^{n+1}|| \leq \beta ||E_k^n|| + (\alpha + \mu + C\delta t + \frac{C\delta t^2}{2})||E_{k-1}^n|| + C||\delta E_{k-1}^n|| + \gamma_F \tag{3.19}$$

$$||E_k^{n,N^f-1}|| \leq \beta ||E_k^n|| + (\alpha + \mu + 2C\delta t + \frac{C\delta t^2}{2})||E_{k-1}^n|| + C(1+\delta t)||\delta E_{k-1}^n|| + 3\gamma_F \tag{3.20}$$

$$||\delta E_k^{n+1}|| \leq C\delta t ||\delta E_{k-1}^n|| + C\delta t ||E_{k-1}^n|| + 2\gamma_F \tag{3.21}$$

Since the upper bound of the error term $||E_k^{n,N^f-1}||$ depends on the error terms $||E_k^{n+1}||$ and $||\delta E_k^{n+1}||$ we focus on the inequalities (3.19)-(3.21). Hence, we can write by induction:

$$\begin{aligned}
||E_k^n|| &\leq \beta ||E_k^{n-1}|| + (\alpha + \mu + C\delta t + \frac{C\delta t^2}{2})||E_{k-1}^{n-1}|| + C^2\delta t \sum_{j=2}^{k} (C\delta t)^{j-2} ||E_{k-j}^{n-j}|| + C(C\delta t)^{k-1} ||\delta E_0^{n-k}|| \\
&+ \left(1 + 2C\frac{1 - (C\delta t)^{k-1}}{1 - (C\delta t)}\right)\gamma_F
\end{aligned} \tag{3.22}$$

The governing term in the sum $C^2\delta t\sum_{j=2}^{k}(C\delta t)^{j-2}||E_{k-j}^{n-j}||$ is the term $C^2\delta t||E_{k-2}^{n-2}||$. To ensure that it does not dominate the term $(\alpha+\mu+C\delta t+\frac{C\delta t^2}{2})||E_{k-1}^{n-1}||$, we suppose that the fine time step verifies: $\delta t\leq\Delta T^2\epsilon_G^2$ (see hypothesis (3.17)).
In the sequel, we show that the residual terms $\delta E_0^{n-k}$, $\left(1+2C\frac{1-(C\delta t)^{k-1}}{1-(C\delta t)}\right)\gamma_F$ and all the terms of the sum for $j\geq 3$ can be distributed over the terms: $||\tilde{E}_k^n||$, $||\tilde{E}_k^{n-1}||$, $||\tilde{E}_{k-1}^{n-1}||$ and $||\tilde{E}_{k-2}^{n-2}||$, where $||\tilde{E}_k^n||$ is a perturbation of $||E_k^n||$.

Setting the error perturbation to:

$$||\tilde{E}_k^n||=||E_k^n||+||\delta E_k^n||+C\delta t\left(||E_{k-1}^{n-1}||+||\delta E_{k-1}^{n-1}||\right)+\frac{2C+3+2C^2\delta t}{\beta-1+\alpha+3\mu+C\delta t+\frac{C\delta t^2}{2}+C^2\delta t}\gamma_F \tag{3.23}$$

The following inequality is satisfied by $||\tilde{E}_k^n||$:

$$||\tilde{E}_k^n||\leq\beta||\tilde{E}_k^{n-1}||+\tilde{\alpha}||\tilde{E}_{k-1}^{n-1}||+C^2\delta t||\tilde{E}_{k-2}^{n-2}|| \tag{3.24}$$

where the constant $\tilde{\alpha}$ is defined in (3.25).
Hence, for $k=0$, we have:

$$||\tilde{E}_0^n||\leq\beta||\tilde{E}_0^{n-1}||+\tilde{\gamma}_G$$

Then, for $k=1$, we have:

$$||\tilde{E}_1^n||\leq\beta||\tilde{E}_1^{n-1}||+\tilde{\alpha}||\tilde{E}_0^{n-1}||+\tilde{\gamma}_F$$

where: In the sequel, we use the following notations:

$$\left\{\begin{array}{lcl}\tilde{\alpha} & = & \alpha+3\mu+3C\delta t+\frac{C\delta t^2}{2}\\ \tilde{\gamma}_G & = & \gamma_G+C\delta t+\frac{2C+3+2C^2\delta t}{\beta-1+\alpha+3\mu+C\delta t+C\frac{\delta t^2}{2}+C^2\delta t}\gamma_F\\ \tilde{\gamma}_F & = & \left(\frac{2C+3+2C^2\delta t}{\beta-1+\alpha+3\mu+C\delta t+C\frac{\delta t^2}{2}+C^2\delta t}+3\right)\gamma_F+C\delta t+2C\delta t^2\end{array}\right. \tag{3.25}$$

Following [53], we consider the sequence $(\tilde{e}_k^n)_{n\geq 0,k\geq 0}$ defined recursively as follows. For $k=0$:

$$\tilde{e}_0^n=\left\{\begin{array}{ll}0 & \text{, if } \quad n=0\\ \beta\tilde{e}_0^{n-1}+\tilde{\gamma}_G & \text{, if } \quad n\geq 1\end{array}\right. \tag{3.26}$$

For $k=1$:

$$\tilde{e}_1^n=\left\{\begin{array}{ll}0 & \text{, if } \quad n=0,1\\ \beta\tilde{e}_1^{n-1}+\tilde{\alpha}\tilde{e}_0^{n-1}+\tilde{\gamma}_F & \text{, if } \quad n\geq 2\end{array}\right. \tag{3.27}$$

For $k\geq 2$:

$$\tilde{e}_k^n=\left\{\begin{array}{ll}0 & \text{, if } \quad n=0,1,2\\ \beta\tilde{e}_k^{n-1}+\tilde{\alpha}\tilde{e}_{k-1}^{n-1}+C^2\delta t\tilde{e}_{k-2}^{n-2} & \text{, if } \quad n\geq 3\end{array}\right. \tag{3.28}$$

Since $||E_k^n|| \leq ||\tilde{E}_k^n|| \leq \tilde{e}_k^n$, for $k \geq 0$, $n = 0, \cdots, N$, we analyse the behavior of $(\tilde{e}_k^n)$ to derive a bound for $\tilde{E}_k^n$. For this, we consider the generating function:

$$\tilde{\rho}_k(\xi) = \sum_{n \geq 0} \tilde{e}_k^n \xi^n$$

From (3.26), (3.27) and (3.28) we get:

$$\begin{cases} \tilde{\rho}_0(\xi) = \dfrac{\tilde{\gamma}_G \xi}{(1-\beta\xi)(1-\xi)} \\ \tilde{\rho}_1(\xi) = \dfrac{\tilde{\alpha}\xi}{1-\beta\xi}\tilde{\rho}_0(\xi) + \dfrac{\tilde{\gamma}_F \xi}{(1-\beta\xi)(1-\xi)} \\ \tilde{\rho}_k(\xi) = \dfrac{\tilde{\alpha}\xi}{1-\beta\xi}\tilde{\rho}_{k-1}(\xi) + \dfrac{C^2\delta t \xi^2}{1-\beta\xi}\tilde{\rho}_{k-2}(\xi), \quad k \geq 2 \end{cases} \tag{3.29}$$

From which we derive, for $k \geq 1$:

$$\tilde{\rho}_k(\xi) = \tilde{\gamma}_G \tilde{\alpha}^k \frac{\xi^{k+1}}{(1-\xi)} \sum_{j=0}^{[k/2]} \frac{(C^2\delta t)^j}{\tilde{\alpha}^{2j}} \binom{k-j}{j} \frac{1}{(1-\beta\xi)^{k+1-j}} + \tilde{\gamma}_F \tilde{\alpha}^{k-1} \frac{\xi^k}{(1-\xi)} \sum_{j=0}^{[k-1/2]} \frac{(C^2\delta t)^j}{\tilde{\alpha}^{2j}} \binom{k-1-j}{j} \frac{1}{(1-\beta\xi)^{k-j}} \tag{3.30}$$

For $k = 0$, we have:

$$\tilde{\rho}_0(\xi) = \tilde{\gamma}_G \xi \left( \sum_{p \geq 0} \xi^p \right) \left( \sum_{p \geq 0} \beta^p \xi^p \right) = \tilde{\gamma}_G \sum_{p \geq 0} \left( \sum_{l=0}^{p} \beta^l \right) \xi^{p+1}$$

By a change of variable $p = n - 1$, we obtain:

$$\tilde{e}_0^n = \tilde{\gamma}_G \left( \sum_{l=0}^{n-1} \beta^l \right) \leq \frac{\tilde{\gamma}_G}{\gamma_G} e^{CT} max_n(1 + ||u(T^n)||) T e^{-C\Delta T} \epsilon_G, \quad n \geq 1$$

For $k \geq 1$, using the binomial expansion in (3.30):

$$\frac{1}{(1-\beta\xi)^{k+1-j}} = \sum_{p \geq 0} \binom{k-j+p}{p} \beta^p \xi^p$$

and by a change of variable, we obtain:

$$\sum_{n \geq 0} \tilde{e}_k^n \xi^n = \tilde{\gamma}_G \tilde{\alpha}^k \sum_{n \geq k+1} K_{n-k-1} \xi^n + \tilde{\gamma}_F \tilde{\alpha}^{k-1} \sum_{n \geq k} K'_{n-k} \xi^n$$

Identifying the term $\xi^k$ in the expansion yields to:

$$\tilde{e}_k^k = \tilde{\gamma}_F \tilde{\alpha}^{k-1} K'_0$$

This gives an upper bound for the error terms $||\tilde{E}_k^k||, \quad k \geq 1$. We do not use this estimate since the parareal algorithm ensures $u_k^n = F(T^0, T^n - T^0, u^0)$ for $k \geq n$, which yields:

$$||E_k^k|| = \mathcal{O}(\epsilon_F), \quad k \geq 1$$

In the sequel, we identify the terms $\xi^n$ for $n \geq k+1$ in the expansion:

$$\tilde{e}_k^n = \tilde{\gamma}_G \tilde{\alpha}^k K_{n-k-1} + \tilde{\gamma}_F \tilde{\alpha}^{k-1} K'_{n-k}$$

We now compute the terms

$$K_p = \sum_{l=0}^{p} \sum_{j=0}^{[k/2]} \frac{(C^2\delta t)^j}{(\alpha + 3\mu + C\delta t)^{2j}} \binom{k-j}{j} \binom{k-j+l}{l} \beta^l$$

and

$$K'_p = \sum_{l=0}^{p} \sum_{j=0}^{[k-1/2]} \frac{(C^2\delta t)^j}{\tilde{\alpha}^{2j}} \binom{k-1-j}{j} \binom{k-1-j+l}{l} \beta^l$$

Using: $\binom{k-j+l}{l} \leq \binom{k+l}{l}$ and: $\dfrac{C^2\delta t}{(\alpha + 3\mu + C\delta t)^2} \leq 1$, from hypothesis (3.17): $\delta t \leq \Delta T^2 \epsilon_G^2$. We have:

$$K_p \leq \sum_{l=0}^{p} \binom{k+l}{l} \beta^l \sum_{j=0}^{[k/2]} \binom{k-j}{j} \leq f_k \binom{k+1+p}{p} \beta^p$$

$$K'_p \leq \sum_{l=0}^{p} \binom{k-1+l}{l} \beta^l \sum_{j=0}^{[k-1/2]} \binom{k-1-j}{j} \leq f_{k-1} \binom{k+p}{p} \beta^p$$

where $f_k$ is the general term of the Fibonacci sequence defined by $f_0 = f_1 = 1$ and $f_{k+1} = f_k + f_{k-1}, k \geq 1$:

$$f_k = \sum_{j=0}^{[k/2]} \binom{k-j}{j} = \frac{(1+\sqrt{5})^{k+1} - (1-\sqrt{5})^{k+1}}{2^{k+1}\sqrt{5}}$$

Hence, we derive the bound:

$$\begin{aligned}
||E_0^n|| &\leq ||\tilde{E}_0^n|| \leq \tilde{e}_0^n \leq \frac{\tilde{\gamma}_G}{\gamma_G} e^{CT} max_n(1 + ||u(T^n)||) T e^{-C\Delta T} \epsilon_G, \quad n \geq 1 \\
||E_k^k|| &= \mathcal{O}(\epsilon_F), \quad k \geq 1 \\
||E_k^n|| &\leq ||\tilde{E}_k^n|| \leq \tilde{e}_k^n \leq \tilde{\gamma}_G \tilde{\alpha}^k f_k \binom{n}{k+1} \beta^{n-k-1} + \tilde{\gamma}_F \tilde{\alpha}^{k-1} f_{k-1} \binom{n}{k} \beta^{n-k}, \quad n \geq k+1, \quad k \geq 1
\end{aligned} \tag{3.31}$$

which ends the proof of the theorem. □

## 3.3 Advantages of the multi-step parareal algorithm

We proposed in the last section a new variant of the parareal algorithm with a consistent approximation of the solution at time $T^n - \delta t$ in a non intrusive way. The initialisation of the fine propagation in each time window has te be appropriately chosen because an initialisation error would be propagated over the whole time interval and would prevent the parareal algorithm to converge towards the target solution. Another option to treat this issue is to use a one-step time scheme or a multi-stage Runge Kutta method to initialize the fine computation. This option is intrusive since we have to implement new time scheme for the initialisation. Morevover, we will see in section 3.4 that this strategy prevents the parareal to converge to the numerical solution with the target accuracy since the first-order scheme error will dominate.

This method adds consistency with the fine scheme. Also, this strategy can be applied to multi-step time schemes involving several fine time steps preceding the time $T^n$ by applying the same correction to terms taking the form: $u_{k+1}^{n,N^f-i}, \quad i = 1, \cdots, I$.

We detail the algorithm for a multi-step time scheme involving more than one fine time step preceding the time $T^n$.

$$\left\{\begin{array}{lcl} u_0^{n+1} & = & G(T^n,\Delta T,u_0^n),\quad 0\leq n\leq N-1\\ u_{k+1}^{n+1} & = & G(T^n,\Delta T,u_{k+1}^n)+F(T^n,\Delta T,u_k^{n-1,N^f-1},u_k^n)\\ & & -G(T^n,\Delta T,u_k^n),\quad 0\leq n\leq N-1,\quad k\geq 0\\ u_{k+1}^{n,N^f-i} & = & F(T^n,\Delta T-i\delta t,u_k^{n-1,N^f-I},u_k^{n-1,N^f-I+1},\cdots,u_k^n)+u_{k+1}^{n+1}\\ & & -F(T^n,\Delta T,u_k^{n-1,N^f-I},u_k^{n-1,N^f-I+1},\cdots,u_k^n),\quad i=0,\cdots,I,\quad 0\leq n\leq N-1,\quad k\geq 0 \end{array}\right. \tag{3.32}$$

where we denote $F(t,s,w^1,w^2,\cdots,w^I)$ the multi-step propagator for any given time $t\in[0,T]$, $s\in[0,T-t]$ and any function $w^1,\cdots,w^I\in\mathbb{U}$ takes $I$ initial values at times $t$ and $t-i\delta t$ and propagates it at time $t+s$, where $\delta t$ is the fine time step. We illustrate the good convegence properties in the next section by applying the parareal algorithm to an ODE system solved by a coarse solver based on a one-step time scheme and a fine solver based on a third-order BDF method.

When the coarse solver is a multi-step time scheme, there exists several options to initialise it on each time window:

- If the coarse time step $\delta T$ is equal to the size of the time window $\Delta T$, there is no additional correction in the parareal algorithm since the solution at every coarse time step are updated
- If $\delta T<\Delta T$, there are intermediate coarse time iterations in each time window. In [12], the initialisation of the coarse solver is addressed and the authors propose a parareal-type correction at time $T^n-\delta T$:

$$\begin{array}{rcl} u_{k+1}^{n+1-N_{int}^c} & = & G(T^n,\Delta T-\delta T,u_{k+1}^{n-N_{int}^c},u_{k+1}^n)+F(T^n,\Delta T-\delta T,u_k^{n-1,N^f-1},u_k^n)\\ & - & G(T^n,\Delta T-\delta T,u_k^{n-N_{int}^c},u_k^n),\quad N_{int}^c=\dfrac{\delta T}{\delta t}\quad 0\leq n\leq N-1,\quad k\geq 0 \end{array} \tag{3.33}$$

We illustrate the behavior of the full multi-step parareal algorithm with specific initialisation of the fine and coarse solver in the next section where the two solvers involved in the parareal method are the second-order BDF method.

## 3.4 Numerical tests

We apply the multi-step parareal algorithm to a simple ODE firstly, the damped oscillator and then to a stiff problem, the Brusselator system. Our results illustrate that our approach improves the convergence properties with respect to the classical parareal algorithm. We also show that the generalisation of this approach to third-order time schemes holds and the convergence properties derived in Theorem (3.3) are preserved. Finally, we address the question of the parallel efficiency of the multi-step parareal. In the last section, we apply the adaptive parareal algorithm (see [80]) where the accuracy of the fine solver is increased accross the iterations.

### 3.4.1 Numerical convergence results

#### 3.4.1.1 The damped oscillator

We consider the damped oscillator system:

$$u''(t)+2\lambda u'(t)+\omega_0^2u(t)=0,\quad t\in(0,T),\ \text{with } u(0)=u_0,\quad T=10,$$

We rewrite it as a first order ODE system:

$$X'(t) = AX(t), \quad t \in (0,T), \text{ with } X(0) = \begin{pmatrix} u_0 \\ u_0' \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & -2\lambda \end{pmatrix}$$

This system models the dynamic of a simple nonstiff harmonic oscillator under a frictional force. For our tests, we set $\lambda = 0.05$, $\omega_0 = 1$ and $X_0 = \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$.

The coarse solver is a Backward Euler method with a coarse time step:

$$\Delta T = 0.1$$

which corresponds to 100 time windows and the fine solver is a second-order BDF method with a fine time step $\delta t = 10^{-4}$ (respecting hypothesis (3.17)). In figure 3.1, the fine solver is based on a two-step time scheme where the computation of the solution $u^{n,j+1}$ at time $T^n + (j+1)\delta t$ depends on the solutions $u^{n,j}$ and $u^{n,j-1}$ at times $T^n + j\delta t$ and $T^n + (j-1)\delta t$, respectively. We use the multi-step variant of parareal (3.13) to initialise the fine solver in each time window, starting from the parareal iteration $k \geq 2$. At the parareal iteration $k = 1$, we use a Backward Euler method to initialise the fine solver since we did not use the fine propagator yet.

In this section, we analyse the evolution of two different errors accross the parareal iterations:

- the error between the fine solution computed in a sequential way and the parareal solution in $L^\infty(0,T)$ norm,
$$max_{1 \leq n \leq N} ||u_k^n - F(T^0, T^n - T^0, u^0)|| \tag{3.34}$$
- the error between the exact solution and the parareal solution in $L^\infty(0,T)$ norm
$$max_{1 \leq n \leq N} ||u_k^n - u(T^n)|| \tag{3.35}$$

In all the figures of this section, we plot the evolution of errors (3.35-3.34) in the two following cases:

- Without a multi-step adaptation (red curve): the error between the parareal solution where the Backward Euler method is used at each iteration for the initialisation of the fine solver and the fine solution computed in a sequential way for (3.34) (the exact solution for (3.35)), on one hand,
- With a multi-step adaptation (blue curve): the error between the solution given by the multi-step parareal algorithm the fine solution computed in a sequential way for (3.34) (the exact solution for (3.35)), on the other hand.

In figure 3.1, we see that without the multi-step adaptation the error (3.34) stagnates around $10^{-6}$ without recovering the fine solution at the machine precision, even after 100 iterations. On the other hand, using the multi-step parareal algorithm, the error continues to decrease until reaching the machine precision. Moreover, in the right figure, we see that the only way to recover the correct approximation of the exact solution is to use a multi-step adaptation, otherwise, without adaptation, the parareal algorithm will not reach the target accuracy. This result shows that making an initialisation error for a multi-step fine solver will prevent the parareal algorithm to obtain the approximation of the exact solution with the desired accuracy.

The convergence properties are illustrated in figure 3.1 on a fine solver based on the second-order

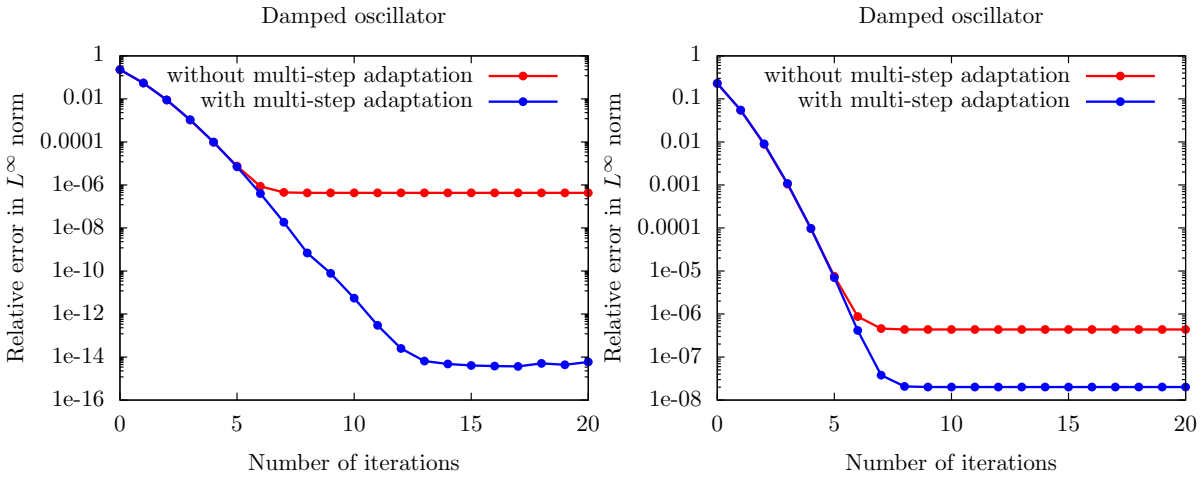


Figure 3.1: Convergence of the multi-step parareal for the second-order BDF method, $\delta t = 10^{-4}$ (left: error (3.34), right: error (3.35))

BDF method with time step $\delta t = 10^{-4}$.

In figure (3.2), we apply the extension of the multi-step parareal algorithm (3.32) to three-step time schemes by giving a consistent approximation of the solutions $u(T^n - \delta t)$ and $u(T^n - 2\delta t)$. We illustrate the convergence properties of this strategy by applying it on a fine solver based on the third-order BDF method with a time step $\delta t = 10^{-4}$ (see figure 3.2). We observe the same behaviour of the errors (3.35-3.34): without a multi-step adaptation, the fine propagation is initialised by two Backward Euler iterations and does not allow to recover the target approximation of the exact solution while the multi-step parareal converges to the exact solution with the desired accuracy.

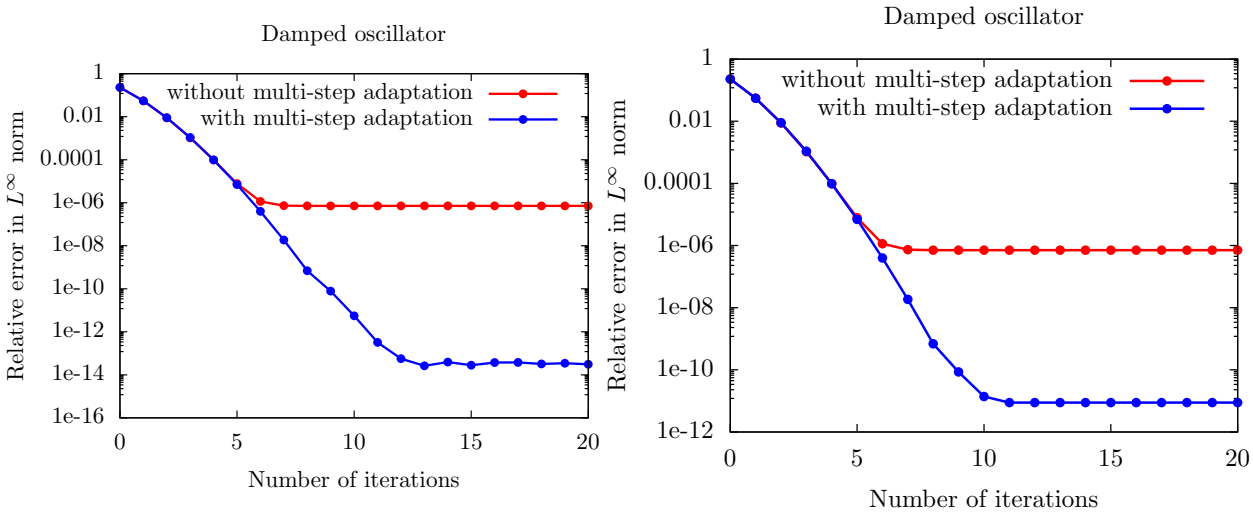


Figure 3.2: Convergence of the multi-step parareal for the third-order BDF method, $\delta t = 10^{-4}$ (left: error (3.34), right: error (3.35))

#### 3.4.1.2 The Brusselator system

We consider the Brusselator system:

$$\begin{cases} x' &= A + x^2 y - (B+1)x \\ y' &= Bx - x^2 y \end{cases}$$

with initial condition $x(0) = 0$ and $y(0) = 1$. This is a stiff ODE that models a chain of chemical reactions. It was already studied in previous works on the parareal algorithm ([53, 80]). The system has a fixed point at $x = A$ and $y = \dfrac{B}{A}$ which becomes unstable when $B > 1 + A^2$ and leads to oscillations. We place ourselves in this oscillatory regime by setting $A = 1$ and $B = 3$. The dynamics present large velocity variations in some time sub-intervals, making the use of high order time schemes particularly desirable for an appropriate treatment of the transient. The coarse solver is a Backward Euler method with a coarse time step:

$$\Delta T = 0.1$$

which corresponds to 180 time windows since $T = 18$. The fine solver is a second-order BDF method with a fine time step $\delta t = 10^{-4}$ (respecting hypothesis (3.17)). In figure 3.3, the fine solver is based on a two-step time scheme. We use the multi-step parareal algorithm (3.13) to initialise the fine solver in each time window.

Likewise, we analyse the evolution of the errors (3.34) and (3.35) accross the parareal iterations. In Figure 3.3, we see that without the multi-step adaptation the error (3.34) stagnates around $10^{-6}$ without recovering the fine solution at the machine precision, even after 180 iterations. On the other hand, using the multi-step parareal algorithm, the error continues to decrease until reaching the machine precision. Moreover, in the right figure, we see that the only way to recover the correct approximation of the exact solution is to use a multi-step adaptation, otherwise, without adaptation, the parareal algorithm will not reach the target accuracy. This result shows that making an initialisation error for a multi-step fine solver will prevent the parareal algorithm to obtain the approximation of the exact solution with the desired accuracy.

The convergence properties are illustrated in figure 3.3 on a fine solver based on the second-order BDF method with time step $\delta t = 10^{-4}$.

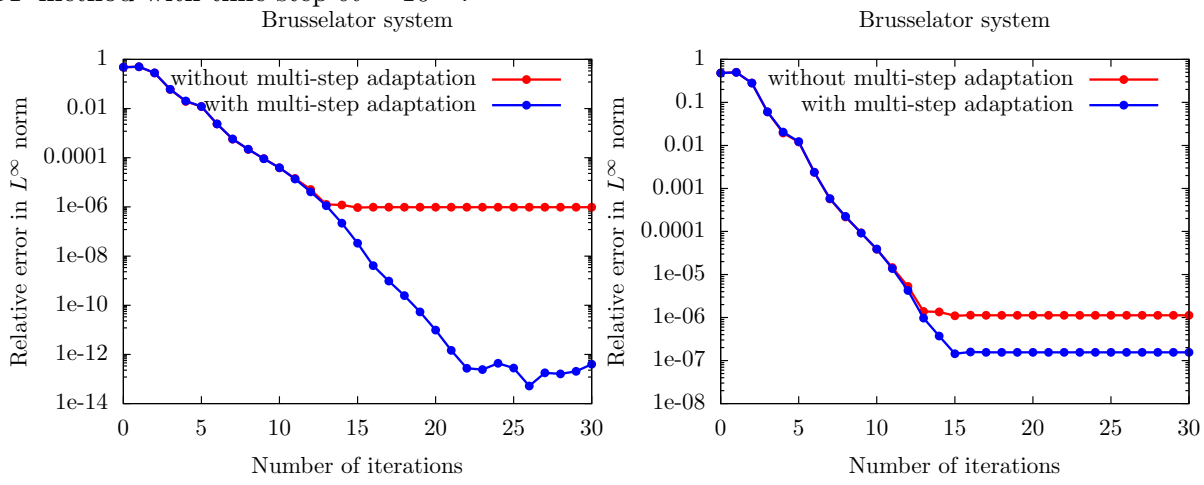


Figure 3.3: Convergence of the multi-step parareal for the second-order BDF method, $\delta t = 10^{-4}$ (left: error (3.34), right: error (3.35))

In the figure (3.4), we apply the extension of the multi-step parareal algorithm (3.32) to three-step time schemes by giving a consistent approximation of the solutions $u(T^n - \delta t)$ and $u(T^n - 2\delta t)$. We illustrate the convergence properties of this strategy by applying it on a fine solver based on the third-order BDF method with time steps $\delta t = 10^{-4}$ (see figure 3.4). We observe the same behavior of the errors (3.35-3.34): without a multi-step adaptation, the fine propagation is initialised by two Backward Euler iterations and does not allow to recover the target approximation of the exact solution while the multi-step parareal converges to the exact solution with the desired accuracy.

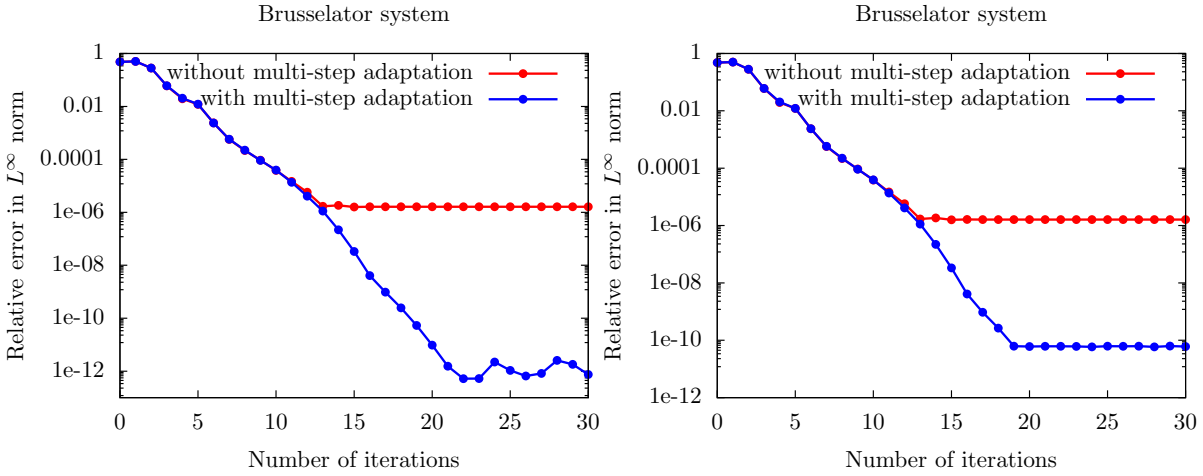


Figure 3.4: Convergence of the multi-step parareal for the third-order BDF method, $\delta t = 10^{-4}$ (left: error (3.34), right: error (3.35))

### 3.4.2 Parallel efficiency

We address in this section the question of the speed up performances for the multi-step parareal algorithm. The only additional operations in the multi-step variant compared to the classical parareal are the corrections of solutions involved in the initialisation of the fine solver in each time window (update of $u_{k+1}^{n,N^f-1}$ in (3.13) for example). Hence, we consider that the computational cost of the multi-step variant is the same as the one of the classical parareal. In a recent work [80], the authors propose a new method, the adaptive parareal algorithm, where the accuracy of the fine solver is increased across the iterations. This new point of view improves the speed up performances of the parareal method and the only remaining factor limiting high performance becomes the cost of the coarse solver. In this section, we seek to improve the parallel efficiency of the multi-step parareal method by increasing the accuracy of fine solver at each iteration. We first recall the parallel efficiency for the classical parareal (CP) and the adaptive parareal (AP) to obtain a solution with accuracy $\eta$ and a propagation over $[0, T]$:

$$
\begin{aligned}
eff_{CP}(\eta, [0,T]) &\sim \frac{1}{K(\eta)} \\
eff_{AP}(\eta, [0,T]) &\sim \frac{1}{1+\epsilon_G^{1/\alpha}}, \text{ under the hypothesis of Proposition 3.1 in [80]}
\end{aligned}
$$

where $K(\eta)$ is the number of parareal iterations to obtain the approximation of the exact solution with the target accuracy $\eta$ and $\alpha$, the order of the fine time scheme. To apply this approach

on the multi-step variant, we need to carefully initialise each time window. If the fine scheme is the second-order BDF method, the computation of $u^{n+1}$ depends on $u^n$ and $u^{n-1}$ and with the adaptive paradigm we have:

$$t^n - t^{n-1} \neq t^{n+1} - t^n$$

Hence, we initialise the fine solver with one variable step-size BDF method.

We apply this strategy to the damped oscillator system with the Backward Euler method as a coarse solver ($\Delta T = 0.1$) and the second-order BDF method as a fine solver with the sequence of time steps indicated in table 3.1.

| | Multi-step parareal | | Adaptive parareal | |
|---|---|---|---|---|
| Iteration | Time step | Error | Time step | Error |
| $k = 1$ | $10^{-4}$ | $5 \times 10^{-2}$ | $10^{-2}$ | $5 \times 10^{-2}$ |
| $k = 2$ | $10^{-4}$ | $9 \times 10^{-3}$ | $5 \times 10^{-3}$ | $2 \times 10^{-2}$ |
| $k = 3$ | $10^{-4}$ | $10^{-3}$ | $10^{-3}$ | $3 \times 10^{-3}$ |
| $k = 4$ | $10^{-4}$ | $9 \times 10^{-5}$ | $5 \times 10^{-4}$ | $3 \times 10^{-4}$ |
| $k = 5$ | $10^{-4}$ | $7 \times 10^{-6}$ | $4 \times 10^{-4}$ | $2 \times 10^{-5}$ |
| $k = 6$ | $10^{-4}$ | $4 \times 10^{-7}$ | $2.5 \times 10^{-4}$ | $3 \times 10^{-6}$ |
| $k = 7$ | $10^{-4}$ | $3.8 \times 10^{-8}$ | $2 \times 10^{-4}$ | $3 \times 10^{-7}$ |
| $k = 8$ | $10^{-4}$ | $2 \times 10^{-8}$ | $10^{-4}$ | $2.9 \times 10^{-8}$ |

Table 3.1: Convergence of the adaptive parareal and the multi-step parareal with a target accuracy $\eta = 3 \times 10^{-8}$

The multi-step parareal algorithm with adaptivity converges to the exact solution with an accuracy obtained by a sequential fine solution with time step $\delta t = 10^{-4}$ after 8 iterations such as the multi-step method without adaptivity (see table 3.1). With the sequence of fine time steps used in the adaptive parareal method, convergence is reached with the same number of iterations as the multi-step variant. The adaptive algorithm allows to obtain better speed-up performances compared to the nonadaptive version since the fine solver ($\delta t = 10^{-4}$) is used only one time instead of 8 times in the multi-step variant. In table 3.2, we give the speed-up and the efficiency of the

| Speed-up | Multi-step parareal | Adaptive parareal |
|---|---|---|
| With cost $G$ | 10.9 | 23.7 |
| Without cost $G$ | 12.5 | 32.2 |

| Efficiency | Multi-step parareal | Adaptive parareal |
|---|---|---|
| With cost $G$ | 10.9% | 23.7% |
| Without cost $G$ | 12.5% | 32.2% |

Table 3.2: Speed up and efficiency with $T = 10$, $\delta t = 10^{-4}$ and $N = 100$

adaptive and multi-step parareal algorithms applied to the damped oscillator. The speed-up is defined as the ratio:

$$S(\eta, [0, T]) := \frac{T_{seq}(\eta, [0, T])}{T_{par}(\eta, [0, T])},$$

between the cost to run a sequential fine solver achieving a target accuracy $\eta$ with the cost to run a parareal algorithm providing at the end the same target accuracy $\eta$. The parallel efficiency of

the method is then defined as the ratio of the above speed up with the number of processors which gives a target of 1 to any parallel solver:

$$eff(\eta, [0,T]) := \frac{S(\eta, [0,T])}{N}.$$

To compare the speed-up of the multi-step and adaptive parareal algorithms, we use the number of fine and coarse propagations involved in the numerical solution and the computational cost of the coarse and fine propagations (communication delays have not been taken into account). For example, in table 3.1, the cost of the multi-step parareal algorithm is equal to the cost of 9 coarse propagations over $[0,T]$ plus 8 fine propagations over $[T^n, T^{n+1}]$ with a fine time step $\delta t = 10^{-4}$. In [80], the authors show that the main element affecting the performance of the adaptive parareal method is no longer the cost of the fine solver but the cost of the coarse solver. Hence, we compare the speed-up and efficiency when we count or do not count the cost of the coarse solver in table 3.2. Obviously, when we do not count the cost of the coarse solver, the performance of both algorithms improves.

## 3.5 Conclusion

We have built a new variant of the parareal algorithm allowing to overcome the issue of initialising the fine and the coarse solvers when they are based on a multi-step time scheme ([5]). The convergence properties of the multi-step parareal are very close to that of the classical parareal algorithm in the case of two-step time schemes. An extension of our approach to generic multi-step time schemes is proposed and validated numerically on a three-step time scheme. In addition, the accuracy of the multi-step parareal algorithm is illustrated on the numerical solution of a stiff ODE such as the Brusselator system. Finally, we address the question of the parallel efficiency of our strategy by coupling it with the adaptive parareal algorithm proposed in [80]. The new adaptive formulation of the parareal algorithm opens the door to improve significantly the parallel efficiency of the method provided that the cost of the coarse solver is moderate.

# Part II

# Analysis of finite volume schemes on staggered grids

# Chapter 4

# $L^2$-stability of finite volume schemes on staggered grids

**Contents**



## 4.1 Introduction

As an introduction to the issue, we consider a 1D conservative non linear hyperbolic system

$$\partial_t U(x,t) + \partial_x F(U)(x,t) = 0, \tag{4.1}$$

with unknown vector $U \in \mathbb{R}^m$ and Lipschitz flux $F : \mathbb{R}^m \to \mathbb{R}^m$ with real-diagonalisable Jacobian matrix $A(U) = \nabla_U F(U) \in \mathbb{R}^{m\times m}$.

In the rest of the introduction we review some notions about the numerical diffusion.

### 4.1.1 Consistency analysis

When approximating smooth solutions $U$ of (4.1) by a consistent numerical method on a regular mesh with space step $\triangle x$, the semi-discrete equations approximate to the first order in $\triangle x$ the following perturbed version of equation (4.1) :

$$\partial_t U_{\Delta x} + \partial_x F(U_{\Delta x}) = \mathcal{D}(U_{\Delta x}, \triangle x) + o(\triangle x), \tag{4.2}$$

where $U_{\Delta x}$ is the numerical solution and $\mathcal{D}$ is a second order differential operator.

When the flux function $F$ is linear, linear numerical methods yield a linear diffusion operator $\mathcal{D}(U, \triangle x) = \triangle x \partial_x (D \partial_x U)$. The matrix $D$ comes from the upwind (off-centered) contributions of the discrete equations and gives a first insight into the scheme precision and stability. In the non linear case ($F$ Lipschitz), the numerical diffusion operator can often be approximated to the first order by a non linear diffusion operator :

$$\mathcal{D}(U, \triangle x) = \triangle x \partial_x (D(U) \partial_x U) + o(\triangle x). \tag{4.3}$$

This is the case for instance for colocated schemes based on characteristic upwinding such as Godunov [60], Roe [100], VFRoe [83] or VFFC [56] schemes where the non linear numerical diffusion tensor is $D(U) = |A(U)|$.

We recall that in the case of symmetric hyperbolic systems (${}^tA(U) = A(U)$), any entropy solution to (4.1) preserves the $L^2$ norm (see [59] Example 3.2 in the Introduction chapter) and we would like the discrete $L^2$ norm of any scheme to be bounded as well. In the case of non-symmetric systems, one first symmetrises the system using entropic variables $V(U) = \vec{\nabla} s(U)$ where $s$ a strictly convex entropy of the system (4.1) is assumed to exist (see [59] Theorem 3.2 in the introduction chapter). The new symmetric system:

$$\partial_t V + \bar{\bar{A}}(V) \partial_x V = 0, \text{ with } {}^t\bar{\bar{A}} = \bar{\bar{A}} \tag{4.4}$$

is linearly $L^2$-stable. Any numerical scheme yields a numerical diffusion $\bar{\bar{D}}(V)$ in the symmetrised basis and we require that the diffusion operator $\bar{\bar{D}}$ have positive symmetric part : ${}^t\bar{\bar{D}} + \bar{\bar{D}} \geq 0$.

The operator $D$ gives a first insight into the scheme precision since the smaller the operator $D$, the closer the approximate solution $U_{\Delta x}$ is to $U$. However the numerical diffusion operator gives also important informations about the scheme stability.

### 4.1.2 Stability analysis

In the case of symmetric hyperbolic systems (${}^tA(U) = A(U)$), the exact equation (4.1) yields the conservation of the $L^2$ norm (see [59] Example 3.2 in the Introduction chapter)

$$\forall t \in \mathbb{R}_+, \quad \partial_t \int_{\mathbb{R}} ||U||_2^2(x,t) \mathrm{d}x = 0, \tag{4.5}$$

whilst the perturbed equation (4.2) yields most of the time the first order estimate

$$\begin{aligned} \forall t \in \mathbb{R}_+, \quad \partial_t \int_{\mathbb{R}} ||U_{\Delta x}||_2^2(x,t) \mathrm{d}x &= \triangle x \int_{\mathbb{R}} {}^t U_{\Delta x} \partial_x (D(U_{\Delta x}) \partial_x U_{\Delta x}) \mathrm{d}x + o(\triangle x) \\ &= -\triangle x \int_{\mathbb{R}} {}^t (\partial_x U_{\Delta x}) D(U_{\Delta x}) (\partial_x U_{\Delta x}) \mathrm{d}x + o(\triangle x), \end{aligned} \tag{4.6}$$

In order to obtain an $L^2$ stable scheme, it is therefore usual to require that the diffusion operator $D$ have positive symmetric part : ${}^tD + D \geq 0$.

In the case of non symmetric systems, one first symmetrises the system using entropic variables $\vec{\xi}(U) = \nabla_U s(U)$ where $s$ is an entropy of the system. The new system is entropic and yields

$$\forall t \in \mathbb{R}_+, \quad \partial_t \int_{\mathbb{R}} ||\xi(U)||_2^2(x,t) \mathrm{d}x = 0, \tag{4.7}$$

We remark that a scheme that is entropic $\partial_t \int_{\mathbb{R}} s(U) \leq 0$ is not necessarily stable since $s$ is not necessarily bounded below as is the case with the full Euler system. We recall the expression of the entropy for the full Euler system:

$$s = C_v \left( \ln \left( \rho E - \frac{q^2}{2\rho} \right) - \gamma \ln \rho \right), \tag{4.8}$$

where $C_v$ is the specific heat and $\gamma$ the adiabatic constant such that: $p = (\gamma - 1)\rho e$, $e$ is the internal energy and $E$, the total energy. If a numerical scheme applied to the full Euler system is entropic then the quantity (4.8) decays. This does not imply the boundedness of the unknowns $\rho, u, E$. For example, assuming $\rho$ is constant, the variables $E$ and $q$ can grow infinitely while maintaining the difference $\rho E - \dfrac{q^2}{2\rho}$ constant.

We investigate in this chapter the $L^2$-stability of staggered schemes. In this first account of our research, we investigate the isentropic Euler system which raises an issue that will remain with more complex fluid model : the numerical treatment of the mass balance equation and of the momentum equation yields a non classical diffusion operator. In order to obtain a straightforwardly stable scheme we propose a new discretisation with positive numerical diffusion. In section 4.2, we determine the numerical diffusion of the staggered schemes and show it does not straightforwardly yield a linear stability. We then present a new class of staggered schemes and prove their linear stability in section 4.3. Some numerical results are given in section 4.4.

## 4.2 The numerical diffusion of staggered schemes for the Euler system

We address here the following system, the isentropic Euler equations, written in the following conservative form :

$$\begin{cases} \partial_t \rho + \partial_x(\rho u) = 0 \\ \partial_t(\rho u) + \partial_x(\rho u^2) + \partial_x p = 0 \end{cases} \tag{4.9}$$

This problem is posed over an open bounded connected subset $\Omega$ of $\mathbb{R}$, with boundary $\partial\Omega$, and a finite time interval $(0, T)$. The variable $t$ stands for the time, $\rho$, $u$ and $p$ are the density, velocity and pressure in the flow.
The results can be extended to the multidimensional Euler system but the calculations are lengthier and do not help the intuition.
The Euler system (4.9) can take the non conservative form

$$\partial_t U + A(U)\partial_x U = 0 \tag{4.10}$$

with $c^2 = \dfrac{\partial p}{\partial \rho}, u = \dfrac{q}{\rho}$ and

$$U = \begin{pmatrix} \rho \\ q \end{pmatrix}, \qquad A(U) = \begin{pmatrix} 0 & 1 \\ c^2 - u^2 & 2u \end{pmatrix}. \tag{4.11}$$

**Remark 4.1.** *In the sequel, we seek to show the upwind matrices for numerical schemes based on the principle of vector upwinding using the eigenbasis of the Jacobian matrix A such as: Godunov [60], Roe [100] and VFRoe [83]. The upwind matrices have two arguments: a left state $U_L$ and a*

*right state $U_R$. Here, we illustrate their behaviour with the Roe scheme. When the right and left states are equal, the upwind matrix has the following expression:*

$$D_{upw}(U,U) = |A(U)| = \frac{1}{c}\begin{pmatrix} c^2-u^2 & u \\ u(c^2-u^2) & u^2+c^2 \end{pmatrix}. \tag{4.12}$$

*If the flux $F$ in 4.1 is linear, the upwind matrices of the Godunov and VFRoe schemes have the same form (4.12). We remark that by a change of basis, we obtain a system of decoupled transport equation where the transport speed is positive, hence ensuring the $L^2$-stability for each equation and thus for the original problem.*

*Also for low Mach numbers ($\frac{|u|}{c} \ll 1$), for a fixed velocity $u$ and the sound speed $c$ tending to infinity, the upwind matrix of the Roe scheme converges towards the identity matrix:*

$$D_{upw} = c\mathbb{I}d + \mathcal{O}(\frac{1}{c}) \tag{4.13}$$

*The diffusion is evenly distributed on the mass and momentum equations. We see that there is problem in the order of magnitude:*

- *Considering the mass equation of the system (4.9), the discretisation introduced a perturbation of the order $\rho c$ in the right hand side while the left hand side is of the order $\rho u$:*

$$\partial_t \rho + \partial_x q = c\triangle x \partial_{xx}\rho + o(\triangle x). \tag{4.14}$$

  *Hence, the numerical scheme is too diffusive for the mass equation.*

- *Considering the momentum equation of the system (4.9), the discretisation introduced a perturbation of the order $\rho uc$ in the right hand side while the left hand side is of the order $\rho c^2$:*

$$\partial_t q + \partial_x \frac{q^2}{\rho} + c^2\partial_x\rho = c\triangle x\partial_{xx} q + o(\triangle x). \tag{4.15}$$

  *Hence, the numerical scheme is not diffusive enough for the mometum equation.*

The upwind type schemes can be proven to be linearly $L^2$ stable ([60], [100], [83]). However the amount of numerical diffusion is proportional to the sound speed $c$ and for low Mach number flows, the schemes based on characteristics upwinding are not able to capture nearly incompressible solutions (see [36, 37] for more details).

On the contrary, staggered schemes are known to be more precise for low Mach number flows in practice and are very popular in the thermal hydraulics community ([98]). However their stability analysis is historically based on heuristics ([70]). Yet the conservative staggered schemes presented in [66, 65] are proven to be entropic and to satisfy a kinetic energy preservation [67]. Likewise in [19], the authors present a kinetic scheme on staggered grids for the barotropic Euler equations, derive stability conditions which preserve both the positivity of the density and the decay of the discrete global entropy, and satisfy a kinetic energy preservation. Unfortunately the boundedness of the entropy does not necessarily imply the boundedness of the solution. Indeed a strictly convex function is not necessarily bounded below. This is in particular the case for the full Euler system since the entropy involves the function $-\ln$ which is strictly convex but not bounded below (see [59] Example 3.3 in the Introduction chapter). In the next subsection we show that the first order perturbed equation (4.2) associated to staggered schemes yields not the classical diffusion operator (4.3) but instead a strongly nonlinear numerical diffusion operator.

### 4.2.1 The staggered scheme of Herbin et al.

Using staggered schemes, the density and pressure are located on cells and the velocity on faces (nodes in 1D) [63, 62]. The momentum variable is usually split as a product between the density and the velocity : $\vec{q} = \rho\vec{u}$. The main difference between the various staggered schemes is the treatment of the convection term $\rho\vec{u} \otimes \vec{u}$ in the momentum equation. We consider the staggered scheme of [67] as a prototype of staggered schemes. Indeed, the main results of this section extends to other staggered schemes. This is because the mass discretisation is the same in all staggered schemes and therefore the mass diffusion operator will be non classical. In the past decade, Herbin, Latché and their coauthors have proposed a new approach with rigorous proofs of stability: discrete inequality for the kinetic energy and entropic character. They discretise the conservative form of the Euler equations (eq:euler syst cons 1D) with a conservative scheme.
The different variants include one step ([65] section 2.1, [66] section 3.1) and prediction/correction steps ([65] section 2.2, [66] section 4.1) variants, fully implicit ([66] section 3, [65] section 2.1), semi implicit and almost explicit [66] (all but the pressure gradient are explicit-in-time) variants.

For simplicity we present the discrete equation of the fully implicit variant ([66] section 3, [65] section 2.1) for the 1D isentropic Euler equations in conservative form.

$$\frac{\rho_i^{n+1} - \rho_i^n}{\triangle t} + \frac{\rho_{i+\frac{1}{2}}^{up,n+1} u_{i+\frac{1}{2}}^{n+1} - \rho_{i-\frac{1}{2}}^{up,n+1} u_{i-\frac{1}{2}}^{n+1}}{\triangle x} = 0 \tag{4.16}$$

$$\frac{\bar{\rho}_{i+\frac{1}{2}}^{n+1} u_{i+\frac{1}{2}}^{n+1} - \bar{\rho}_{i+\frac{1}{2}}^{n} u_{i+\frac{1}{2}}^{n}}{\triangle t} + \frac{\overline{\rho u}_{i+1}^{n+1} u_{i+1}^{up,n+1} - \overline{\rho u}_{i}^{n+1} u_{i}^{up,n+1}}{\triangle x} + \frac{p_{i+1}^{n+1} - p_i^{n+1}}{\triangle x} = 0. \tag{4.17}$$

The pressure $p_i$ and the density $\rho_i$ are located at the cell centers whereas the velocity $u_{i+\frac{1}{2}}$ are located at the cell interfaces. The expression of the products $\rho u$, $\rho\partial_t u$ and $\rho u^2$ between the velocity located at cell interfaces and the density located at cell centers thus has to be defined through interpolation formula.

The mass flux $\rho u$ at the cell interfaces is defined using an upwind density $\rho_{i+\frac{1}{2}}^{up}$ defined as :

$$\begin{aligned} \rho_{i+\frac{1}{2}}^{up} &= \begin{cases} \rho_i & \text{if} \quad u_{i+\frac{1}{2}} > 0 \\ \rho_{i+1} & \text{if} \quad u_{i+\frac{1}{2}} \leq 0 \end{cases} \\ &= \frac{\rho_i + \rho_{i+1}}{2} + sign(u_{i+\frac{1}{2}}) \frac{\rho_i - \rho_{i+1}}{2}, \end{aligned} \tag{4.18}$$

which is the sum of a centered and an upwind terms.
The expression of $\bar{\rho}_{i+\frac{1}{2}}$ in the discrete momentum equation accounts for an average of the neighbouring densities

$$\bar{\rho}_{i+\frac{1}{2}} = \frac{1}{2}(\rho_i + \rho_{i+1}). \tag{4.19}$$

The expression of $\overline{\rho u}$ in the discrete momentum equation is

$$\overline{\rho u}_i = \frac{1}{2}(\rho_{i-\frac{1}{2}}^{up} u_{i-\frac{1}{2}} + \rho_{i+\frac{1}{2}}^{up} u_{i+\frac{1}{2}}). \tag{4.20}$$

The upwind velocity $u_i^{up}$ at cell centers is defined as :

$$u_i^{up} = \begin{cases} u_{i-\frac{1}{2}} & \text{if} \quad \overline{\rho u}_i > 0 \\ u_{i+\frac{1}{2}} & \text{if} \quad \overline{\rho u}_i \leq 0 \end{cases}$$

$$= \frac{u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}}}{2} + sign(\overline{\rho u}_i)\frac{u_{i-\frac{1}{2}} - u_{i+\frac{1}{2}}}{2}, \tag{4.21}$$

which is the sum of a centered and an upwind terms.
It is possible to use a centered velocity $\bar{u}$ instead of the upwind velocity $u^{up}$ (see [66]).

### 4.2.2 The numerical diffusion of the scheme

The scheme presented in [66] section 3 and [65] section 2.1 is proven to be entropic. However the boundedness of the entropy does not necessarily imply the boundedness of the solution. Indeed a convex function is not necessarily bounded below. This is in particular the case for the full Euler system (not the isentropic one) because the entropy involves the function $-\ln$ which is convex but not bounded below.
We would like to study the $L^2$ stability of the scheme by analysing its numerical diffusion operator. This would give a new insight into the scheme and prove at least a linear stability and a nonlinear stability in the case of almost constant initial data.
In this section, we assume that the exact solution is smooth and we determine the numerical diffusion of the scheme (4.16-4.17). In the context of staggered schemes, we have two meshes: one for the cell centered unknowns at points $x_i, i = 1, \cdots, N$ and the other for the unknowns defined on edges at points $x_{i+1/2}$. Then we could write the consistency error for the momentum equation around $x = x_i$ or $x_{i+1/2}$. To stay in the spirit of the staggered schemes, we first develop the analysis around $x = x_{i+1/2}$ then in a second time around $x = x_i$ to be closer to the classical analysis of the consistency error. To obtain the consistency error for the mass equation, we perform the following Taylor expansions around $x = x_i$, assuming smooth solutions with $u \neq 0$ :

$$\rho_{i-1} = \rho(x_i) - \Delta x\partial_x\rho(x_i) + \frac{1}{2}(\Delta x)^2\partial_{xx}\rho(x_i) + \mathcal{O}(\Delta x^3) \tag{4.22}$$

$$\rho_{i+1} = \rho(x_i) + \Delta x\partial_x\rho(x_i) + \frac{1}{2}(\Delta x)^2\partial_{xx}\rho(x_i) + \mathcal{O}(\Delta x^3) \tag{4.23}$$

$$u_{i-\frac{1}{2}} = u(x_i) - \frac{\Delta x}{2}\partial_x u(x_i) + \frac{1}{2}\left(\frac{\Delta x}{2}\right)^2\partial_{xx}u(x_i) + \mathcal{O}(\Delta x^3) \tag{4.24}$$

$$u_{i+\frac{1}{2}} = u(x_i) + \frac{\Delta x}{2}\partial_x u(x_i) + \frac{1}{2}\left(\frac{\Delta x}{2}\right)^2\partial_{xx}u(x_i) + \mathcal{O}(\Delta x^3) \tag{4.25}$$

**Mass numerical diffusion** From (4.16), (A.1) and (A.2) the discrete mass flux is (we omit the time indices)

$$\rho^{up}_{i+\frac{1}{2}}u_{i+\frac{1}{2}} - \rho^{up}_{i-\frac{1}{2}}u_{i-\frac{1}{2}} = \Delta x\partial_x(\rho u)(x_i) - \frac{1}{2}(\Delta x)^2 sign(u(x_i))\partial_x(u(x_i)\partial_x\rho) + \mathcal{O}(\Delta x^3). \tag{4.26}$$

Hence the mass flux consistency is finally

$$\frac{\rho^{up}_{i+\frac{1}{2}}u_{i+\frac{1}{2}} - \rho^{up}_{i-\frac{1}{2}}u_{i-\frac{1}{2}}}{\Delta x} = \partial_x(\rho u)(x_i) - \frac{\Delta x}{2}sign(u(x_i))\partial_x(u\partial_x\rho)(x_i) + \mathcal{O}(\Delta x^2). \tag{4.27}$$

and the numerical diffusion associated to th mass conservation law is the strongly non linear diffusion term $sign(u)\partial_x(u\partial_x\rho)$. The linear stability analysis of such a strongly nonlinear diffusion is not classical and we are not aware of any reference.
If we assume that $u$ does not change sign then the diffusion term simplifies to the weakly non linear

diffusion term $\partial_x(|u|\partial_x\rho)$ which is positive diffusion coefficient $|u|$. The weakly non linear diffusion term $\partial_x(|u|\partial_x\rho)$ can be linearised around a constant state $(\rho_0, u_0 \neq 0$ as:

$$\partial_x(|u_0|\partial_x\rho) + \partial_x(|u|\partial_x\rho_0) = |u_0|\partial_{xx}\rho).$$

Hence if $u > 0$ or $u < 0$ the mass equation has a positive contribution on the diagonal of the numerical diffusion tensor $D$ and thus has a stabilising effect.
If we allow $u$ to change sign then the multiplication with $sign(u)$ makes things more complicate and we can not rule out potential instabilities. The linearisation is not trivial since the smoothness of $u$ does not imply even the continuity of $sign(u)$. The consistency analysis is only a first step that requires smooth solutions but the final goal of capturing discontinuous weak solutions with velocity that change sign will raise even more questions.

**Momentum numerical diffusion** For the momentum numerical diffusion we compute

- the contribution from the pressure

$$p_{i+1}-p_i = \Delta x\partial_x p(x_i)+\frac{1}{2}(\Delta x)^2\partial_{xx}p(x_i)+\mathcal{O}(\Delta x^3).p_{i+1}-p_i = \Delta x\partial_x p(x_{i+\frac{1}{2}})+\mathcal{O}(\Delta x^3). \quad (4.28)$$

- the contribution from the time evolution term (case of the conservative scheme (4.16-4.17))

$$\begin{aligned}\partial_t(\bar{\rho}_{i+\frac{1}{2}}u)(x_{i+\frac{1}{2}},t) &= \partial_t(\rho u)(x_{i+\frac{1}{2}},t) + \partial_t\left(\frac{\rho(x_{i+1}) - 2\rho(x_{i+\frac{1}{2}}) + \rho(x_i)}{2}u\right)(x_{i+\frac{1}{2}},t) & (4.29)\\ &= \partial_t(\rho u)(x_{i+\frac{1}{2}},t) + \frac{1}{2}\left(\frac{\Delta x}{2}\right)^2\partial_t\left((\partial_{xx}\rho)u\right)(x_{i+\frac{1}{2}},t) + \mathcal{O}(\Delta x^3) & (4.30)\end{aligned}$$

  The time evolution term will bring a perturbation in $(\Delta x)^2$ that we can neglect since we are interested in first order error terms in $(\Delta x)$.

To obtain the consistency error, we perform the following Taylor expansions around $x = x_{i+\frac{1}{2}}$, assuming smooth solutions with $u \neq 0$:

$$\begin{aligned}\rho_{i+1} &= \rho(x_{i+\frac{1}{2}}) + \frac{\Delta x}{2}\partial_x\rho(x_{i+\frac{1}{2}}) + \frac{\Delta x^2}{8}\partial_{xx}\rho(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^3)\\ \rho_{i+2} &= \rho(x_{i+\frac{1}{2}}) + \frac{3\Delta x}{2}\partial_x\rho(x_{i+\frac{1}{2}}) + \frac{9\Delta x^2}{8}\partial_{xx}\rho(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^3)\\ \rho_i &= \rho(x_{i+\frac{1}{2}}) - \frac{\Delta x}{2}\partial_x\rho(x_{i+\frac{1}{2}}) + \frac{\Delta x^2}{8}\partial_{xx}\rho(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^3)\\ \rho_{i-1} &= \rho(x_{i+\frac{1}{2}}) - \frac{3\Delta x}{2}\partial_x\rho(x_{i+\frac{1}{2}}) + \frac{9\Delta x^2}{8}\partial_{xx}\rho(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^3)\\ u_{i+\frac{3}{2}} &= u(x_{i+\frac{1}{2}}) + \Delta x\partial_x u(x_{i+\frac{1}{2}}) + \frac{1}{2}(\Delta x)^2\partial_{xx}u(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^3)\\ u_{i-\frac{1}{2}} &= u(x_{i+\frac{1}{2}}) - \Delta x\partial_x u(x_{i+\frac{1}{2}}) + \frac{1}{2}(\Delta x)^2\partial_{xx}u(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^3)\end{aligned} \quad (4.31)$$

**Momentum numerical diffusion** From (4.17), (A.7) and (A.8), the discrete momentum flux yields (we omit the time indices)

$$\begin{aligned}\overline{\rho u}_{i+1}u_{i+1}^{up} - \overline{\rho u}_i u_i^{up} &= \Delta x\partial_x(\rho u^2)(x_{i+\frac{1}{2}}) - \frac{\Delta x^2}{2}sign(\rho u(x_{i+\frac{1}{2}}))\partial_x(\rho u\partial_x u)(x_{i+\frac{1}{2}})\\ &- \frac{\Delta x^2}{2}sign(u(x_{i+\frac{1}{2}}))\partial_x(u^2\partial_x\rho)(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^3)\end{aligned} \quad (4.32)$$

Hence, from (4.32) and (4.28), the momentum flux consistency is finally:

$$\begin{aligned}\frac{\overline{\rho u}_{i+1}u_{i+1}^{up}-\overline{\rho u}_i u_i^{up}}{\Delta x}+\frac{p_{i+1}-p_i}{\Delta x} &= \partial_x(\rho u^2)(x_{i+\frac{1}{2}})+\partial_x p(x_{i+\frac{1}{2}})\\ &- \frac{\Delta x}{2}\left\{sign(u(x_{i+\frac{1}{2}}))\partial_x(u^2\partial_x\rho)(x_{i+\frac{1}{2}})+sign(\rho u(x_{i+\frac{1}{2}}))\partial_x(\rho u\partial_x u)(x_{i+\frac{1}{2}})\right\}\\ &+ \mathcal{O}(\Delta x^2)\end{aligned} \tag{4.33}$$

For smooth solutions $u$ and $\rho$:

$$\partial_x(\rho u\partial_x u)=\partial_x(\rho u\partial_x\frac{q}{\rho})=\partial_x(u\partial_x q-u^2\partial_x\rho)$$

We obtain:

$$\begin{aligned}\frac{\overline{\rho u}_{i+1}u_{i+1}^{up}-\overline{\rho u}_i u_i^{up}}{\Delta x}+\frac{p_{i+1}-p_i}{\Delta x} &= \partial_x(\rho u^2)(x_{i+\frac{1}{2}})+\partial_x p(x_{i+\frac{1}{2}})\\ &- \frac{\Delta x}{2}\left\{\left(sign(u(x_{i+\frac{1}{2}}))-sign(\rho u(x_{i+\frac{1}{2}}))\right)\partial_x(u^2\partial_x\rho)(x_{i+\frac{1}{2}})\right.\\ &\left.+sign(\rho u(x_{i+\frac{1}{2}}))\partial_x(u\partial_x q)(x_{i+\frac{1}{2}})\right\}+\mathcal{O}(\Delta x^2)\end{aligned} \tag{4.34}$$

Assuming that the scheme (4.16-4.17) preserves the positivity of $\rho$, we have:

$$sign(\rho u(x_{i+\frac{1}{2}}))=sign(u(x_{i+\frac{1}{2}}))$$

Hence:

$$\begin{aligned}\frac{\overline{\rho u}_{i+1}u_{i+1}^{up}-\overline{\rho u}_i u_i^{up}}{\Delta x}+\frac{p_{i+1}-p_i}{\Delta x} &= \partial_x(\rho u^2)(x_{i+\frac{1}{2}})+\partial_x p(x_{i+\frac{1}{2}})\\ &- \frac{\Delta x}{2}sign(u(x_{i+\frac{1}{2}}))\partial_x(u\partial_x q)(x_{i+\frac{1}{2}})+\mathcal{O}(\Delta x^2)\end{aligned} \tag{4.35}$$

The numerical diffusion associated to the momentum conservation law is the term $sign(u)\partial_x(u\partial_x q)$ and is decoupled from the mass diffusion. The linear stability analysis of such a strongly non linear diffusion is not classical and we are not aware of any reference in the litterature.
If we assume that $u$ does not change sign then the diffusion term simplifies to the weakly non linear diffusion term $\partial_x(|u|\partial_x q)$ which involves a positive diffusion coefficient $|u|$. The weakly non linear diffusion term $\partial_x(|u|\partial_x q)$ can be linearised around a constant state $(q_0, u_0 \neq 0$ as:

$$\partial_x(|u_0|\partial_x q)+\partial_x(|u|\partial_x q_0)=|u_0|\partial_{xx}q).$$

Hence if $u>0$ or $u<0$ the momentum equation has a positive contribution on the diagonal of the numerical diffusion tensor $D$ and thus has a stabilising effect.
If we allow $u$ to change sign then the multiplication with $sign(u)$ makes things more complicated and we can not rule out potential instabilities. The linearisation is not trivial, even taking $u$ smooth enough, since the function $sign(u)$ is not continuous. The consistency analysis is only a first step that requires smooth solutions but the final goal of capturing discontinuous weak solutions with velocity that change sign will raise even more issues.

Finally, the staggered scheme (4.16-4.17) have the following numerical diffusion operator:

$$\begin{aligned}&\partial_t U_{\Delta x}+\partial_x F(U_{\Delta x})=\mathcal{D}(U_{\Delta x},\Delta x)+\mathcal{O}(\Delta x^2)\\ &\mathcal{D}(U_{\Delta x},\Delta x)=\frac{\Delta x}{2}\begin{pmatrix}sign(u(x_i)) & 0\\ 0 & sign(u(x_{i+\frac{1}{2}}))\end{pmatrix}\partial_x\left(\begin{pmatrix}u(x_i) & 0\\ 0 & u(x_{i+\frac{1}{2}})\end{pmatrix}\partial_x\begin{pmatrix}\rho(x_i)\\ q(x_{i+\frac{1}{2}})\end{pmatrix}\right)\end{aligned} \tag{4.36}$$

Now that we derived the consistency error around the point $x = x_{i+1/2}$ for the momentum equation we turn to the classical way to analyse the consistency error around $x = x_i$.

- the contribution from the pressure

$$p_{i+1} - p_i = \Delta x \partial_x p(x_i) + \frac{1}{2}(\Delta x)^2 \partial_{xx} p(x_i) + \mathcal{O}(\Delta x^3). \tag{4.37}$$

- the contribution from the time evolution term (case of the conservative scheme (4.16-4.17))

$$\begin{aligned} \bar{\rho}_{i+\frac{1}{2}} u_{i+\frac{1}{2}} &= \frac{1}{2}(\rho_i + \rho_{i+1}) u_{i+\frac{1}{2}} \\ &= (\rho u)(x_i) + \frac{\Delta x}{2}\partial_x(\rho u)(x_i) + \frac{\Delta x^2}{8}\rho \partial_{xx} u(x_i) + \frac{\Delta x^2}{4}\partial_x(u\partial_x \rho)(x_i) + \mathcal{O}(\Delta x^3) \end{aligned} \tag{4.38}$$

Since we are interested in first order error terms in $(\Delta x)$, we neglect the perturbation in $(\Delta x)^2$ and obtain:

$$\partial_t(\bar{\rho}_{i+\frac{1}{2}} u)(x_i, t) = \partial_t(\rho u)(x_i, t) + \frac{\Delta x}{2}\partial_t(\partial_x(\rho u))(x_i, t) + \mathcal{O}(\Delta x^2) \tag{4.39}$$

Since $(\rho u)(x_i, t)$ is a solution of (4.9):

$$\partial_t(\partial_x(\rho u))(x_i, t) = \partial_x(\partial_t(\rho u))(x_i, t) = -\partial_{xx}(\rho u^2)(x_i, t) - \partial_{xx} p(x_i, t)$$

$$\partial_t(\bar{\rho}_{i+\frac{1}{2}} u)(x_i, t) = \partial_t(\rho u)(x_i, t) - \frac{\Delta x}{2}\partial_{xx}(\rho u^2)(x_i, t) - \frac{\Delta x}{2}\partial_{xx} p(x_i, t) + \mathcal{O}(\Delta x^2) \tag{4.40}$$

We seek to derive the numerical diffusion for the momentum equation around $x = x_i$:

$$\overline{\rho u}_{i+1} u^{up}_{i+1} - \overline{\rho u}_i u^{up}_i = \Delta x \left( \partial_x(\rho u^2)(x_i) + \frac{\Delta x}{2}\partial_{xx}(\rho u^2)(x_i) \right) - \frac{\Delta x^2}{2} sign(u(x_i)) \partial_x(u \partial_x q)(x_i) + \mathcal{O}(\Delta x^3) \tag{4.41}$$

Hence, from (4.41) and (4.37), the momentum flux consistency is finally:

$$\begin{aligned} \partial_t(\bar{\rho}_{i+\frac{1}{2}} u) + \frac{\overline{\rho u}_{i+1} u^{up}_{i+1} - \overline{\rho u}_i u^{up}_i}{\Delta x} + \frac{p_{i+1} - p_i}{\Delta x} &= \partial_t(\rho u)(x_i) + \partial_x(\rho u^2)(x_i) + \partial_x p(x_i) \\ &- \frac{\Delta x}{2}\{ sign(u(x_i)) \partial_x(u \partial_x q)(x_i) - \partial_{xx} p(x_i) \\ &- \partial_{xx}(\rho u^2)(x_i) - \partial_t(\partial_x(\rho u))(x_i) \} + \mathcal{O}(\Delta x^2) \end{aligned} \tag{4.42}$$

From (4.40), we obtain:

$$\begin{aligned} \partial_t(\bar{\rho}_{i+\frac{1}{2}} u) + \frac{\overline{\rho u}_{i+1} u^{up}_{i+1} - \overline{\rho u}_i u^{up}_i}{\Delta x} + \frac{p_{i+1} - p_i}{\Delta x} &= \partial_t(\rho u)(x_i) + \partial_x(\rho u^2)(x_i) + \partial_x p(x_i) \\ &- \frac{\Delta x}{2} sign(u(x_i)) \partial_x(u \partial_x q)(x_i) + \mathcal{O}(\Delta x^2) \end{aligned} \tag{4.43}$$

Finally, the staggered scheme (4.16-4.17) have the following numerical diffusion operator when the consistency error is derived around $x = x_i$ in the momentum equation:

$$\begin{aligned} &\partial_t U_{\Delta x} + \partial_x F(U_{\Delta x}) = \mathcal{D}(U_{\Delta x}, \Delta x) + \mathcal{O}(\Delta x^2) \\ &\mathcal{D}(U_{\Delta x}, \Delta x) = \frac{\Delta x}{2} \begin{pmatrix} sign(u(x_i)) & 0 \\ 0 & sign(u(x_i)) \end{pmatrix} \partial_x \left( \begin{pmatrix} u(x_i) & 0 \\ 0 & u(x_i) \end{pmatrix} \partial_x \begin{pmatrix} \rho(x_i) \\ q(x_i) \end{pmatrix} \right) \end{aligned} \tag{4.44}$$

Hence, we have the following result on the numerical diffusion operator of the staggered schemes.

**Theorem 4.2** (Numerical diffusion of staggered schemes)**.** *The second order perturbation operator associated to the staggered scheme (4.16-4.17) on a 1D regular mesh with space step $\Delta x$ is the strongly nonlinear operator :*

$$\mathcal{D}(U,\Delta x) = \triangle x \begin{pmatrix} sign(u) & 0 \\ 0 & sign(u) \end{pmatrix} \partial_x \left[ \begin{pmatrix} u & 0 \\ c^2 & u \end{pmatrix} \partial_x \begin{pmatrix} \rho \\ q \end{pmatrix} \right] + o(\triangle x). \tag{4.45}$$

The numerical diffusion associated to the mass conservation law is the same for every staggered scheme since they all have the same discretisation of the mass equation. Hence all the staggered schemes have a non classical diffusion operator at least for the mass equation. Each staggered scheme differs from the others with the discretisation of the momentum equation, especially the convective term (non conservative or conservative scheme, implicit or semi implicit, ...). Hence, theorem 4.2 holds for both the mass and the momentum equations only for the Herbin et al staggered scheme (4.16-4.17).

In the context of staggered schemes, there are two meshes, one for the density $\rho$ and one for the velocity $u$. Hence we have two options to write the consistency error for the momentum equation. This leads to two numerical diffusion operators (4.36) and (4.44). (4.36) writes the momentum consistency equation at point $x_{i+1/2}$ whereas (4.44) writes it at point $x_i$. Provided that the velocity $u$ has a constant sign then these operators are equivalent and the diffusion operator (4.36) obtained around $x_{i+1/2} = x_i + \dfrac{\Delta x}{2}$ converges towards the diffusion operator (4.44) obtained around $x_i$, when $\Delta x$ tends to zero. However, if the velocity changes sign, there is not enough regularity to have the equivalence of (4.36) and (4.44). In practice, the velocity $u$ changes sign on a finite number of points in the domain, hence, the numerical diffusion operators (4.36) and (4.44) will differ only in some points.

In the next section, we propose a new class of staggered schemes whose numerical diffusion operator has a classical form (4.3) and ensures the $L^2$-stability of the scheme.

## 4.3 A new class of staggered schemes for the Euler equations

The most advanced result regarding staggered schemes are the kinetic energy inequality and the entropic stability proved for the scheme (4.16-4.17) in [66]. These properties however do not guarantee the boundedness of the solution (see the discussion in section 4.1.2). In this section, we propose a class of staggered schemes for conservation laws (4.1) which are linearly $L^2$-stable. Unlike classical staggered schemes which have a non classical diffusion operator (see the discussion at the end of section 4.2.1 and therorem 4.2). We impose a classical diffusion operator $\mathcal{D}$ (4.3), such that the diffusion tensor verifies: $\bar{\bar{D}} + {}^t\bar{\bar{D}} \geq 0$, where $\bar{D}$ is the matrix $D$ in the basis that symmetrises the Euler system.
We specify this new class in the particular case of the following 2D isentropic Euler equations in conservative form:

$$\begin{cases} \partial_t \rho + \nabla \cdot \vec{q} = 0 \\ \partial_t \vec{q} + \nabla \cdot \dfrac{\vec{q} \otimes \vec{q}}{\rho} + \vec{\nabla} p = 0 \end{cases}. \tag{4.46}$$

This can be written:

$$\left\{\begin{array}{l} \partial_t\rho + \partial q_x + \partial q_y = 0 \\ \partial_t q_x + \partial_x\left(\frac{q_x^2}{\rho}\right) + \partial_y\left(\frac{q_x q_y}{\rho}\right) + c^2\partial_x\rho = 0 \\ \partial_t q_y + \partial_x\left(\frac{q_y^2}{\rho}\right) + \partial_x\left(\frac{q_x q_y}{\rho}\right) + c^2\partial_y\rho = 0 \end{array}\right. . \tag{4.47}$$

The isentropic Euler system takes the conservative form

$$\partial_t U + \nabla\cdot F(U) = 0, \tag{4.48}$$

where $U = (\rho, \vec{q})$ and $F$ is the flux matrix.
The isentropic Euler system takes the non conservative form

$$\partial_t U + A_x(U)\partial_x U + A_y(U)\partial_y U = 0. \tag{4.49}$$

It is usual to define the Jacobian of $F$ along vectors $\vec{n} = (n_x, n_y) \in \mathcal{R}^2$ as

$$\begin{array}{rcl} A(U,\vec{n}) & = & n_x A_x(U) + n_y A_y(U) \\ & = & \left(\begin{array}{cc} 0 & {}^t\vec{n} \\ c^2\vec{n} - (\vec{u}\cdot\vec{n})\vec{u} & \vec{u}\otimes\vec{n} + (\vec{u}\cdot\vec{n})\mathbb{I}_2 \end{array}\right), \end{array} \tag{4.50}$$

with $c^2 = \dfrac{\partial p}{\partial\rho}$ assumed constant.
We consider the class *Stag* of discrete staggered conservative schemes of the form:

$$U'_{i,j}(t) + \frac{F^x_+ - F^x_-}{\Delta x} + \frac{F^y_+ - F^y_-}{\Delta y} = 0, \quad \text{with: } U_{i,j} = \left(\begin{array}{c} \rho_{i,j} \\ q^x_{i+1/2,j} \\ q^y_{i,j+1/2} \end{array}\right), \text{ and:} \tag{4.51}$$

$$F^x_+ = \frac{F^x(U_{i,j}) + F^x(U_{i+1,j})}{2} + D_{Stag}(U_{i,j}, U_{i+1,j}, \vec{n}_x)\frac{U_{i,j} - U_{i+1,j}}{2}, \tag{4.52}$$

$$F^y_+ = \frac{F^y(U_{i,j}) + F^y(U_{i,j+1})}{2} + D_{Stag}(U_{i,j}, U_{i,j+1}, \vec{n}_y)\frac{U_{i,j} - U_{i,j+1}}{2}, \tag{4.53}$$

where $D_{Stag}$ is a $3\times 3$ matrix valued function. An example of scheme in the class *Stag* is the staggered centered scheme, which correspond to the case $D_{Stag} = 0$.
Schemes of the class *Stag* admit a classical diffusion operator (4.3).

**Theorem 4.3** (Classical diffusion of *Stag* schemes)**.** *Let $D_{Stag} : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ be a matrix valued Lipshitz function. A staggered conservative scheme (4.51) with a numerical flux $F^x_+$ and $F^x_-$ of the form (4.53) admits the following classical diffusion operator*

$$\mathcal{D}(U, \Delta x, \Delta_y) = \Delta x\partial_x(D_{Stag}(U, U, \vec{n}_x)\partial_x U) + \Delta y\partial_y(D_{Stag}(U, U, \vec{n}_y)(\partial_y U) + o(\triangle x, \triangle y).$$

*on a regular mesh with space steps $\Delta x$ and $\Delta y$.*

In the sequel, we analyse an element of the class of schemes *Stag* whose expression is:

$$\left\{\begin{array}{l} \partial_t\rho_{i,j} + \dfrac{\bar{q}^x_{i+1/2,j} - \bar{q}^x_{i-1/2,j}}{\Delta x} + \dfrac{\bar{q}^y_{i,j+1/2} - \bar{q}^y_{i,j-1/2}}{\Delta y} = 0 \\ \partial_t q^x_{i+1/2,j} + \dfrac{\left(\frac{\bar{q_x^2}}{\rho}\right)_{i+1,j} - \left(\frac{\bar{q_x^2}}{\rho}\right)_{i,j}}{\Delta x} + \dfrac{\left(\frac{\bar{\bar{q_x q_y}}}{\rho}\right)_{i+1/2,j+1/2} - \left(\frac{\bar{\bar{q_x q_y}}}{\rho}\right)_{i+1/2,j-1/2}}{\Delta y} + c^2\dfrac{\rho_{i+1,j} - \rho_{i,j}}{\Delta x} = 0 \\ \partial_t q^y_{i,j+1/2} + \dfrac{\left(\frac{\bar{q_y^2}}{\rho}\right)_{i,j+1} - \left(\frac{\bar{q_y^2}}{\rho}\right)_{i,j}}{\Delta y} + \dfrac{\left(\frac{\tilde{q_x q_y}}{\rho}\right)_{i+1/2,j+1/2} - \left(\frac{\tilde{q_x q_y}}{\rho}\right)_{i-1/2,j+1/2}}{\Delta x} + c^2\dfrac{\rho_{i,j+1} - \rho_{i,j}}{\Delta y} = 0 \end{array}\right. . \tag{4.54}$$

This numerical scheme (4.54) can be written in the form (4.51)-(4.53) based on the following numerical flux:

$$F_+^x = \begin{pmatrix} \bar{q}^x_{i+1/2,j} \\ \left(\frac{\bar{q_x^2}}{\rho}\right)_{i+1,j} + c^2\rho_{i+1,j} \\ \left(\frac{\tilde{q_x q_y}}{\rho}\right)_{i+1/2,j+1/2} \end{pmatrix}, \quad F_+^y = \begin{pmatrix} \bar{q}^y_{i,j+1/2} \\ \left(\frac{\bar{q_x q_y}}{\rho}\right)_{i+1/2,j+1/2} \\ \left(\frac{\bar{q_y^2}}{\rho}\right)_{i,j+1} + c^2\rho_{i,j+1} \end{pmatrix}. \tag{4.55}$$

Using the Roe average, the scheme takes a more compact form that follows:

$$\begin{aligned}
\bar{q}^x_{i+1/2,j} &= q^x_{i+1/2,j} + (|u_x| - u_x)\frac{\rho_{i,j} - \rho_{i+1,j}}{2} \\
\left(\frac{\bar{q_x^2}}{\rho}\right)_{i+1,j} &= \frac{(q^x_{i+1/2,j})^2}{\rho_{i,j}} + (|u_x| - u_x)\frac{q^x_{i+1/2,j} - q^x_{i+3/2,j}}{2} \\
\left(\frac{\tilde{q_x q_y}}{\rho}\right)_{i+1/2,j+1/2} &= \frac{q^x_{i+1/2,j} q^y_{i,j+1/2}}{\rho_{i,j}} + (|u_y| - u_y)\frac{q^x_{i+1/2,j} - q^x_{i+1/2,j+1}}{2}
\end{aligned}$$

$$\text{Where } u_x \text{ is a Roe-type average: } u_x = \frac{\frac{q^x_{i+1/2,j}}{\sqrt{\rho_{i,j}}} + \frac{q^x_{i+3/2,j}}{\sqrt{\rho_{i+1,j}}}}{\sqrt{\rho_{i,j}} + \sqrt{\rho_{i+1,j}}}$$

$$\begin{aligned}
\bar{q}^y_{i,j+1/2} &= \frac{(q^y_{i,j+1/2})^2}{\rho_{i,j}} + (|u_y| - u_y)\frac{q^y_{i,j+1/2} - q^y_{i,j+3/2}}{2} \\
\left(\frac{\bar{q_x q_y}}{\rho}\right)_{i+1/2,j+1/2} &= \frac{q^x_{i+1/2,j} q^y_{i,j+1/2}}{\rho_{i,j}} + (|u_x| - u_x)\frac{q^y_{i,j+1/2} - q^y_{i+1,j+1/2}}{2}
\end{aligned}$$

$$\text{Where } u_y \text{ is a Roe-type average: } u_y = \frac{\frac{q^y_{i,j+1/2}}{\sqrt{\rho_{i,j}}} + \frac{q^y_{i,j+3/2}}{\sqrt{\rho_{i,j+1}}}}{\sqrt{\rho_{i,j}} + \sqrt{\rho_{i,j+1}}}$$

Using the fact that the numerical scheme (4.54) has a numerical flux of the form (4.51-4.53), we can determine the scheme upwind operator: $D_{Stag}(U_{i,j}, U_{i+1,j}, \vec{n}_x)$.

$$\begin{aligned}
F_+^x - \frac{F^x(U_{i,j}) + F^x(U_{i+1,j})}{2} &= \begin{pmatrix} \bar{q}^x_{i+1/2,j} \\ \left(\frac{\bar{q_x^2}}{\rho}\right)_{i+1,j} + c^2\rho_{i+1,j} \\ \left(\frac{\tilde{q_x q_y}}{\rho}\right)_{i+1/2,j+1/2} \end{pmatrix} \\
&\quad -\frac{1}{2}\begin{pmatrix} q^x_{i+1/2,j} + q^x_{i+3/2,j} \\ \frac{(q^x_{i+1/2,j})^2}{\rho_{i,j}} + \frac{(q^x_{i+3/2,j})^2}{\rho_{i+1,j}} + c^2(\rho_{i,j} + \rho_{i+1,j}) \\ \frac{q^x_{i+1/2,j}}{\rho_{i,j}} q^y_{i,j+1/2} + \frac{q^x_{i+3/2,j}}{\rho_{i+1,j}} q^y_{i+1,j+1/2} \end{pmatrix} \\
&= -\frac{1}{2}\begin{pmatrix} |u_x| - u_x & 1 & 0 \\ -c^2 - u_x^2 & |u_x| + u_x & 0 \\ -u_x u_y & u_y & |u_x| \end{pmatrix}\begin{pmatrix} \rho_{i+1,j} - \rho_{i,j} \\ q^x_{i+3/2,j} - q^x_{i+1/2,j} \\ q^y_{i+1,j+1/2} - q^y_{i,j+1/2} \end{pmatrix}
\end{aligned} \tag{4.56}$$

We obtain $D_{Stag}(U_{i,j}, U_{i+1,j}, \vec{n}_x)$ and $D_{Stag}(U_{i,j}, U_{i,j+1}, \vec{n}_y)$ (with the same approach on $F_y^+$) the matrices coming from the upwind contributions of the discrete equations:

$$\begin{aligned} D_{Stag}(U_{i,j}, U_{i+1,j}, \vec{n}_x) &= \begin{pmatrix} |u_x| - u_x & 1 & 0 \\ -c^2 - u_x^2 & |u_x| + u_x & 0 \\ -u_x u_y & u_y & |u_x| \end{pmatrix}, \\ D_{Stag}(U_{i,j}, U_{i,j+1}, \vec{n}_y) &= \begin{pmatrix} |u_y| - u_y & 0 & 1 \\ -u_x u_y & |u_y| & u_x \\ -c^2 - u_y^2 & 0 & |u_y| + u_y \end{pmatrix} \end{aligned} \tag{4.57}$$

### 4.3.1 The linearised system

In order to simplify the stability analysis we consider the linearised Euler system around a state with density $\rho_0$, momentum $\vec{q}_0$ and velocity $\vec{u}_0 = \dfrac{1}{\rho_0}\vec{q}_0$. From the identity:

$$\nabla \cdot \frac{\vec{q} \otimes \vec{q}}{\rho} = (\nabla \cdot \vec{q})\frac{\vec{q}}{\rho} + (\vec{\nabla}\vec{q})\frac{\vec{q}}{\rho} - \frac{\vec{q} \otimes \vec{q}}{\rho^2}\vec{\nabla}\rho,$$

the linearised Euler system is obtained as the following constant coefficient PDE system

$$\begin{cases} \partial_t \rho + \nabla \cdot \vec{q} = 0 \\ \partial_t \vec{q} + \left( (\nabla \cdot \vec{q})\mathbb{I}_d + \vec{\nabla}\vec{q} \right) \vec{u}_0 + (c^2 \mathbb{I}_d - \vec{u}_0 \otimes \vec{u}_0)\vec{\nabla}\rho = 0 \end{cases} . \tag{4.58}$$

This can be written:

$$\begin{cases} \partial_t \rho + \partial_x q_x + \partial_y q_y = 0 \\ \partial_t q_x + u_0^x(\nabla \cdot q + \partial_x q_x) + u_0^y \partial_y q_x + (c^2 - (u_0^x)^2)\partial_x \rho - u_0^x u_0^y \partial_y \rho = 0 \\ \partial_t q_y + u_0^y(\nabla \cdot q + \partial_y q_y) + u_0^x \partial_x q_y + (c^2 - (u_0^y)^2)\partial_y \rho - u_0^x u_0^y \partial_x \rho = 0 \end{cases} . \tag{4.59}$$

The linearised Euler system takes the form

$$\partial_t U + \nabla \cdot \bar{F}(U) = 0, \tag{4.60}$$

where $U = (\rho, \vec{q})$ and $\bar{F}$ is the linearisation of the matrix flux $F$, and satisfies $\bar{F}(U)\vec{n} = \bar{A}(\vec{n})U$ where the jacobian matrix $\bar{A}(\vec{n}) = n_x \bar{A}_x + n_y \bar{A}_y$ has expression

$$\forall \vec{n} \in \mathcal{R}^2, \qquad \bar{A}(\vec{u}_0, \vec{n}) = \begin{pmatrix} 0 & {}^t\vec{n} \\ c^2\vec{n} - (\vec{u}_0 \cdot \vec{n})\vec{u}_0 & \vec{u}_0 \otimes \vec{n} + (\vec{u}_0 \cdot \vec{n})\mathbb{I}_d \end{pmatrix}, \tag{4.61}$$

The scheme (4.54) applied to the linearised Euler system writes:

$$\begin{cases} \partial_t \rho_{i,j} + \dfrac{\bar{q}^x_{i+1/2,j} - \bar{q}^x_{i-1/2,j}}{\Delta x} + \dfrac{\bar{q}^y_{i,j+1/2} - \bar{q}^y_{i,j-1/2}}{\Delta y} = 0 \\ \partial_t q^x_{i+1/2,j} + 2u_0^x \dfrac{\bar{q}^x_{i+1,j} - \bar{q}^x_{i,j}}{\Delta x} + u_0^x \dfrac{\bar{\bar{q}}^y_{i,j+1} - \bar{\bar{q}}^y_{i,j}}{\Delta y} + u_0^y \dfrac{\tilde{q}^x_{i+1/2,j+1/2} - \tilde{q}^x_{i+1/2,j-1/2}}{\Delta y} \\ \qquad + c^2 \dfrac{\rho_{i+1,j} - \rho_{i,j}}{\Delta x} - (u_0^x)^2 \dfrac{\rho_{i,j} - \rho_{i-1,j}}{\Delta x} - u_0^x u_0^y \dfrac{\rho_{i,j} - \rho_{i,j-1}}{\Delta y} = 0 \\ \partial_t q^y_{i,j+1/2} + 2u_0^y \dfrac{\bar{q}^y_{i,j+1} - \bar{q}^y_{i,j}}{\Delta y} + u_0^y \dfrac{\bar{\bar{q}}^x_{i+1,j} - \bar{\bar{q}}^x_{i,j}}{\Delta x} + u_0^x \dfrac{\tilde{q}^y_{i+1/2,j+1/2} - \tilde{q}^y_{i-1/2,j+1/2}}{\Delta x} \\ \qquad + c^2 \dfrac{\rho_{i,j+1} - \rho_{i,j}}{\Delta y} - (u_0^y)^2 \dfrac{\rho_{i,j} - \rho_{i,j-1}}{\Delta y} - u_0^x u_0^y \dfrac{\rho_{i,j} - \rho_{i-1,j}}{\Delta x} = 0 \end{cases} . \tag{4.62}$$

The numerical scheme (4.62) can be written in the form (4.51)-(4.53) as follows:

$$U'_{i,j}(t) + \frac{\bar{F}^x_+ - \bar{F}^x_-}{\Delta x} + \frac{\bar{F}^y_+ - \bar{F}^y_-}{\Delta y} = 0, \quad \text{with: } U_{i,j} = \begin{pmatrix} \rho_{i,j} \\ q^x_{i+1/2,j} \\ q^y_{i,j+1/2} \end{pmatrix}, \text{ and:} \tag{4.63}$$

$$\bar{F}^x_+ = \frac{\bar{F}^x(U_{i,j}) + \bar{F}^x(U_{i+1,j})}{2} + \bar{D}_{Stag}(\vec{u}_0, \vec{u}_0, \vec{n}_x)\frac{U_{i,j} - U_{i+1,j}}{2}, \tag{4.64}$$

$$\bar{F}^y_+ = \frac{\bar{F}^y(U_{i,j}) + \bar{F}^y(U_{i,j+1})}{2} + \bar{D}_{Stag}(\vec{u}_0, \vec{u}_0, \vec{n}_y)\frac{U_{i,j} - U_{i,j+1}}{2}, \tag{4.65}$$

based on the following numerical flux:

$$\bar{F}^x_+ = \begin{pmatrix} \bar{q}^x_{i+1/2,j} \\ 2u^x_0\bar{q}^x_{i+1,j} + c^2\rho_{i+1,j} - (u^x_0)^2\rho_{i,j} \\ u^y_0\bar{\bar{q}}_{i+1,j} - u^x_0u^y_0\rho_{i,j} + u^x_0\tilde{q}^y_{i+1/2,j+1/2} \end{pmatrix}, \quad \bar{F}^y_+ = \begin{pmatrix} \bar{q}^y_{i,j+1/2} \\ u^x_0\bar{\bar{q}}^y_{i,j+1} - u^x_0u^y_0\rho_{i,j} + u^y_0\tilde{q}^x_{i+1/2,j+1/2} \\ 2u^y_0\bar{q}^y_{i,j+1} + c^2\rho_{i,j+1} - (u^y_0)^2\rho_{i,j} \end{pmatrix}. \tag{4.66}$$

Using the Roe average, the scheme takes a more compact form that follows:

$$\begin{aligned}
\bar{q}^x_{i+1/2,j} &= q^x_{i+1/2,j} + (|u^x_0| - u^x_0)\frac{\rho_{i,j} - \rho_{i+1,j}}{2} \\
\bar{q}^x_{i+1,j} &= \frac{q^x_{i+1/2,j} + q^x_{i+3/2,j}}{2} + \frac{1 + sign(u^x_0)}{2}\frac{q^x_{i+1/2,j} - q^x_{i+3/2,j}}{2} \\
\bar{\bar{q}}^x_{i+1,j} &= q^x_{i+1/2,j} \\
\tilde{q}^y_{i+1/2,j+1/2} &= \frac{q^y_{i,j+1/2} + q^y_{i+1,j+1/2}}{2} + sign(u^x_0)\frac{q^y_{i,j+1/2} - q^y_{i+1,j+1/2}}{2} \\
\bar{q}^y_{i,j+1/2} &= q^y_{i,j+1/2} + (|u^y_0| - u^y_0)\frac{\rho_{i,j} - \rho_{i,j+1}}{2} \\
\bar{q}^y_{i,j+1} &= \frac{q^y_{i,j+1/2} + q^y_{i,j+3/2}}{2} + \frac{1 + sign(u^y_0)}{2}\frac{q^y_{i,j+1/2} - q^y_{i,j+3/2}}{2} \\
\bar{\bar{q}}^y_{i,j+1} &= q^y_{i,j+1/2} \\
\tilde{q}^x_{i+1/2,j+1/2} &= \frac{q^x_{i+1/2,j} + q^x_{i+1/2,j+1}}{2} + sign(u^y_0)\frac{q^x_{i+1/2,j} - q^x_{i+1/2,j+1}}{2}
\end{aligned}$$

Using the fact that the numerical scheme (4.62) has a numerical flux of the form (4.63-4.65), we can determine the scheme upwind operator: $\bar{D}_{Stag}(\vec{u}_0, \vec{u}_0, \vec{n}_x)$.

$$\begin{aligned}
\bar{F}^x_+ \;-\; \frac{\bar{F}^x(U_{i,j}) + \bar{F}^x(U_{i+1,j})}{2} &= \begin{pmatrix} \bar{q}^x_{i+1/2,j} \\ 2u^x_0\bar{q}^x_{i+1,j} + c^2\rho_{i+1,j} - (u^x_0)^2\rho_{i,j} \\ u^y_0\bar{\bar{q}}_{i+1,j} - u^x_0u^y_0\rho_{i,j} + u^x_0\tilde{q}^y_{i+1/2,j+1/2} \end{pmatrix} \\
&\quad -\frac{1}{2}\begin{pmatrix} q^x_{i+1/2,j} + q^y_{i+3/2,j} \\ 2u^x_0(q^x_{i+1/2,j} + q^x_{i+3/2,j}) + (c^2 - (u^x_0)^2)(\rho_{i,j} + \rho_{i+1,j}) \\ u^y_0(q^x_{i+1/2,j} + q^x_{i+3/2,j}) - u^x_0u^y_0(\rho_{i,j} + \rho_{i+1,j}) + u^x_0(q^y_{i,j+1/2} + q^y_{i+1,j+1/2}) \end{pmatrix} \\
&= -\frac{1}{2}\begin{pmatrix} |u^x_0| - u^x_0 & 1 & 0 \\ -c^2 - (u^x_0)^2 & |u^x_0| + u^x_0 & 0 \\ -u^x_0u^y_0 & u^y_0 & |u^x_0| \end{pmatrix}\begin{pmatrix} \rho_{i+1,j} - \rho_{i,j} \\ q^x_{i+3/2,j} - q^x_{i+1/2,j} \\ q^y_{i+1,j+1/2} - q^y_{i,j+1/2} \end{pmatrix}
\end{aligned} \tag{4.67}$$

We obtain $\bar{D}_{Stag}(\vec{u}_0, \vec{u}_0, \vec{n}_x)$ and $\bar{D}_{Stag}(\vec{u}_0, \vec{u}_0, \vec{n}_y)$ (with the same approach on $\bar{F}_y^+$) the matrices coming from the upwind contributions of the discrete equations:

$$\bar{D}_{Stag}(\vec{u}_0, \vec{u}_0, \vec{n}_x) = \begin{pmatrix} |u_0^x| - u_0^x & 1 & 0 \\ -c^2 - (u_0^x)^2 & |u_0^x| + u_0^x & 0 \\ -u_0^x u_0^y & u_0^y & |u_0^x| \end{pmatrix}, \quad \bar{D}_{Stag}(\vec{u}_0, \vec{u}_0, \vec{n}_y) = \begin{pmatrix} |u_0^y| - u_0^y & 0 & 1 \\ -u_0^x u_0^y & |u_0^y| & u_0^x \\ -c^2 - (u_0^y)^2 & 0 & |u_0^y| + u_0^y \end{pmatrix} \tag{4.68}$$

### 4.3.2 Linear stability of the class *Stag*

In order to study the stability of the *Stag* class of staggered schemes, the first step is to use the variables that symmetrise the continuous system and the second to prove that the energy ($L^2$ norm) of the new variables decreases with time. The reason why symmetrising is important is that the contribution of the first order derivatives to the energy balance:

$$\int_{\mathbb{R}^2} {}^tV A_x \partial_x V + {}^tV A_y \partial_y V dxdy$$

vanishes if $A$ is a symmetric matrix since in that case

$${}^tV A_x \partial_x V + {}^tV A_y \partial_y V = \partial_x \frac{1}{2} {}^tV A_x V + \partial_y \frac{1}{2} {}^tV A_y V,$$

which is the divergence of the vector field $\frac{1}{2} \begin{pmatrix} {}^tV A_x V \\ {}^tV A_y V \end{pmatrix}$.

The linearised Euler system can be symmetrised using the variable $V = \begin{pmatrix} c\rho \\ \vec{q} - \rho \vec{u}_0 \end{pmatrix}$. We obtain the following system that is equivalent to the linearised Euler system:

$$\begin{cases} \partial_t \tilde{\rho} + c \nabla \cdot \vec{u} + \vec{u}_0 \cdot \vec{\nabla} \tilde{\rho} = 0 \\ \partial_t \vec{u} + (\vec{\nabla} \vec{u}) \vec{u}_0 + c \vec{\nabla} \tilde{\rho} = 0 \end{cases}$$

with $\tilde{\rho} = \dfrac{c}{\rho_0} \rho$. The symmetrisation of the linearised Euler system therefore takes the form:

$$\partial_t V + \nabla \cdot \bar{\bar{F}}(V) = 0$$

where $\bar{\bar{F}}$ is the symmetrisation of the linearised matrix flux $\bar{F}$. The Jacobian matrix associated to $\bar{\bar{F}}$ is the symmetric operator $\bar{\bar{A}}$ with expression:

$$\bar{\bar{A}}(\vec{u}_0, \vec{n}) = \begin{pmatrix} \vec{u}_0 \cdot \vec{n} & c^t \vec{n} \\ c\vec{n} & \vec{u}_0 \cdot \vec{n} \end{pmatrix}$$

From the symmetrised Euler system, we have the following property of $L^2$ stability for the schemes of the class *Stag*.

**Theorem 4.4** ($L^2$-stability of *Stag* schemes)**.** *A staggered conservative scheme (4.51) with a numerical flux $F_+^x, F_+^y$ of the form (4.53) such that the diffusion operator $D_{Stag}$ verifies: $\bar{\bar{D}}_{Stag} + {}^t\bar{\bar{D}}_{Stag} \geq 0$, is linearly $L^2$-stable.*

*Proof.* After the linearisation around the state $V_0\mathbb{R}^3$ and symmetrisation, the Euler system takes the form:

$$\partial_t V + \bar{\bar{A}}(V_0)\nabla \cdot V = 0, \quad V = \begin{pmatrix} c\rho \\ \vec{q} - \rho\vec{u}_0 \end{pmatrix}$$

The values $V_{i,j}$ of $V$ in the cell $C_{i,j}$ are solutions of:

$$\partial_t V_{i,j} + \frac{\bar{\bar{F}}^x_+ - \bar{\bar{F}}^x_-}{\Delta x} + \frac{\bar{\bar{F}}^y_+ - \bar{\bar{F}}^y_-}{\Delta y} = 0$$

with fluxes:

$$\bar{\bar{F}}^x_+ = \bar{\bar{A}}(V_0, \vec{n}_x)\frac{V_{i,j} + V_{i+1,j}}{2} + \bar{\bar{D}}_{Stag}(V_0, V_0, \vec{n}_x)\frac{V_{i,j} - V_{i+1,j}}{2}$$
$$\bar{\bar{F}}^y_+ = \bar{\bar{A}}(V_0, \vec{n}_y)\frac{V_{i,j} + V_{i,j+1}}{2} + \bar{\bar{D}}_{Stag}(V_0, V_0, \vec{n}_y)\frac{V_{i,j} - V_{i,j+1}}{2}$$

We compute the evolution in time of $||V||_2^2 = \int_{\mathbb{R}^2} c^2\rho^2 + ||\vec{q} - \rho\vec{u}_0||^2 dxdy$ using the symmetry of $\bar{\bar{A}}$ and the positiveness of $\bar{\bar{D}}_{Stag}$:

$$\begin{aligned}
\frac{1}{2}\frac{d||V||_2^2}{dt} &= V \cdot \frac{dV}{dt} = \sum_i \sum_j |C_{ij}| V_{i,j} \cdot \frac{dV_{i,j}}{dt} \\
&= -\sum_i \sum_j |C_{ij}| V_{i,j} \cdot \left( \frac{\bar{\bar{F}}^x_+ - \bar{\bar{F}}^x_-}{\Delta x} + \frac{\bar{\bar{F}}^y_+ - \bar{\bar{F}}^y_-}{\Delta y} \right) \\
&= -\frac{1}{2}\sum_i \sum_j \frac{|C_{ij}|}{\Delta x} V_{i,j} \cdot \bar{\bar{A}}(V_0, \vec{n}_x)(V_{i+1,j} - V_{i-1,j}) \\
&- \frac{1}{2}\sum_i \sum_j \frac{|C_{ij}|}{\Delta x} V_{i,j} \cdot \left( \bar{\bar{D}}_{Stag}(V_0, V_0, \vec{n}_x)(V_{i,j} - V_{i+1,j}) - \bar{\bar{D}}_{Stag}(V_0, V_0, \vec{n}_x)(V_{i-1,j} - V_{i,j}) \right) \\
&- \frac{1}{2}\sum_i \sum_j \frac{|C_{ij}|}{\Delta y} V_{i,j} \cdot \bar{\bar{A}}(V_0, \vec{n}_y)(V_{i,j+1} - V_{i,j-1}) \\
&- \frac{1}{2}\sum_i \sum_j \frac{|C_{ij}|}{\Delta y} V_{i,j} \cdot \left( \bar{\bar{D}}_{Stag}(V_0, V_0, \vec{n}_y)(V_{i,j} - V_{i,j+1}) - \bar{\bar{D}}_{Stag}(V_0, V_0, \vec{n}_y)(V_{i,j-1} - V_{i,j}) \right)
\end{aligned}$$

Since:

$$\bar{\bar{A}}(V_0, \vec{n}_x) = {}^t\bar{\bar{A}}(V_0, \vec{n}_x) \text{ and } \bar{\bar{A}}(V_0, \vec{n}_y) = {}^t\bar{\bar{A}}(V_0, \vec{n}_y)$$

Then:

$$\sum_i \sum_j \frac{|C_{ij}|}{\Delta x} V_{i,j} \cdot \bar{\bar{A}}(V_0, \vec{n}_x)(V_{i+1,j} - V_{i-1,j}) = 0 \text{ and: } \sum_i \sum_j \frac{|C_{ij}|}{\Delta y} V_{i,j} \cdot \bar{\bar{A}}(V_0, \vec{n}_y)(V_{i,j+1} - V_{i,j-1}) = 0$$

$$\begin{aligned}
\frac{1}{2}\frac{d||V||_2^2}{dt} &= -\frac{\Delta y}{2}\sum_i \sum_j (V_{i,j} - V_{i+1,j}) \cdot \bar{\bar{D}}_{Stag}(V_0, V_0, \vec{n}_x)(V_{i,j} - V_{i+1,j}) \\
&- \frac{\Delta x}{2}\sum_i \sum_j (V_{i,j} - V_{i,j+1}) \cdot \bar{\bar{D}}_{Stag}(V_0, V_0, \vec{n}_y)(V_{i,j} - V_{i,j+1})
\end{aligned}$$

Since

$$(V_{i,j} - V_{i+1,j}) \cdot \bar{\bar{D}}_{Stag}(V_0, V_0, \vec{n}_x)(V_{i,j} - V_{i+1,j}) = {}^t\bar{\bar{D}}_{Stag}(V_0, V_0, \vec{n}_x)(V_{i,j} - V_{i+1,j}) \cdot (V_{i,j} - V_{i+1,j})$$

We have:

$$(V_{i,j}-V_{i+1,j})\cdot\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_x)(V_{i,j}-V_{i+1,j})=\frac{1}{2}(V_{i,j}-V_{i+1,j})\cdot\left(\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_x)+{}^t\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_x)\right)(V_{i,j}-V_{i+1,j})$$

$$\begin{aligned}\frac{1}{2}\frac{d||V||_2^2}{dt}&=-\frac{\Delta y}{4}\sum_i\sum_j(V_{i,j}-V_{i+1,j})\cdot\left(\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_x)+{}^t\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_x)\right)(V_{i,j}-V_{i+1,j})\\&-\frac{\Delta x}{4}\sum_i\sum_j(V_{i,j}-V_{i,j+1})\cdot\left(\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_y)+{}^t\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_y)\right)(V_{i,j}-V_{i,j+1})\\&=-\frac{\Delta y}{4}\left(\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_x)+{}^t\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_x)\right)\sum_i\sum_j||V_{i,j}-V_{i+1,j}||_2^2\\&-\frac{\Delta x}{4}\left(\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_y)+{}^t\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_y)\right)\sum_i\sum_j||V_{i,j}-V_{i,j+1}||_2^2\end{aligned}$$

Since: $\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n})+{}^t\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n})\geq 0$, we obtain $\frac{1}{2}\frac{d||V||_2^2}{dt}\leq 0$. □

**Corollary 4.5.** *The numerical scheme (4.54) is linearly $L^2$ stable.*

*Proof.* We determine the upwinding matrices $\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_x)$ and $\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_y)$:

We have: $V=PU$ with: $P=\begin{pmatrix}c&0\\-\vec{u}_0&\mathbb{I}_2\end{pmatrix}$, $P^{-1}=\begin{pmatrix}\frac{1}{c}&0\\\frac{\vec{u}_0}{c}&\mathbb{I}_2\end{pmatrix}$.

Thus, we have:

$$\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_x)=P\bar{D}_{Stag}(V_0,V_0,\vec{n}_x)P^{-1}=\begin{pmatrix}|\vec{u}_0\cdot\vec{n}_x|&c^t\vec{n}_x\\-c\vec{n}_x&|\vec{u}_0\cdot\vec{n}_x|\mathbb{I}_2\end{pmatrix}$$

$$\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_y)=P\bar{D}_{Stag}(V_0,V_0,\vec{n}_y)P^{-1}=\begin{pmatrix}|\vec{u}_0\cdot\vec{n}_y|&c^t\vec{n}_y\\-c\vec{n}_y&|\vec{u}_0\cdot\vec{n}_y|\mathbb{I}_2\end{pmatrix}$$

We have:

$$\begin{aligned}\frac{1}{2}\frac{d||V||_2^2}{dt}&=-\frac{\Delta y}{4}\sum_i\sum_j(V_{i,j}-V_{i+1,j})\cdot\left(\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_x)+{}^t\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_x)\right)(V_{i,j}-V_{i+1,j})\\&-\frac{\Delta x}{4}\sum_i\sum_j(V_{i,j}-V_{i,j+1})\cdot\left(\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_y)+{}^t\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_y)\right)(V_{i,j}-V_{i,j+1})\end{aligned}$$

Since:

$\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_x)+{}^t\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_x)=|\vec{u}_0\cdot\vec{n}_x|\mathbb{I}_3$ and: $\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_y)+{}^t\bar{\bar{D}}_{Stag}(V_0,V_0,\vec{n}_y)=|\vec{u}_0\cdot\vec{n}_y|\mathbb{I}_3$

$$\frac{1}{2}\frac{d||V||_2^2}{dt}=-\frac{\Delta y}{4}|\vec{u}_0\cdot\vec{n}_x|\sum_i\sum_j||V_{i,j}-V_{i+1,j}||_2^2-\frac{\Delta x}{4}|\vec{u}_0\cdot\vec{n}_y|\sum_i\sum_j||V_{i,j}-V_{i,j+1}||_2^2$$

$$\frac{1}{2}\frac{d||V||_2^2}{dt}\leq 0$$

Since $\bar{\bar{D}}_{Stag}$ satisfies the hypotheses of Theorem 4.4, this numerical scheme is linearly $L^2$ stable. □

**Remark 4.6.** *For one dimensionnal flows and low Mach numbers ($\frac{|u|}{c} \ll 1$), the numerical scheme (4.54) of the class Stag has the following numerical diffusion matrix:*

$$D_{Stag} = \begin{pmatrix} 0 & 1 \\ -c^2 & 0 \end{pmatrix} + \mathcal{O}(\frac{1}{c}),$$

*obtained for a fixed velocity $u$ and the sound speed $c$ tending to infinity in (4.57). We see that the diffusion is less important on the mass than on the momentum equation while it is evenly distributed for the Roe scheme (see 4.13). In the mass equation, the discretisation introduced a perturbation $\triangle x \partial_{xx} q$:*

$$\partial_t \rho + \partial_x q = \triangle x \partial_{xx} q + o(\triangle x). \tag{4.69}$$

*It can be expected that the scheme (4.54) yield a better accuracy with less numerical diffusion on the mass equation than the upwind scheme (4.14).*

Now that we proved that the scheme (4.54) is $L^2$-stable, we show in the next section some numerical results with an implicit version of the numerical scheme.

## 4.4 Numerical results

In this section, we assess the behaviour of our new staggered scheme ([92]) on a one dimensional Riemann problem. The robustness of the scheme is illustrated on a compressible fluid with isothermal equation of state $p = \rho c^2$ where the sound speed is $c = 300m/s$. We choose initial conditions such that the structure of the solution consists in a rarefaction wave followed by a shock wave, with sufficiently strong shock to allow an easy discrimination of correct numerical solutions. These initial conditions are:

$$\text{left state: } \begin{pmatrix} \rho_{left} \\ q_{left} \end{pmatrix} = \begin{pmatrix} 10 \\ 9 \\ 100\rho_{left} \end{pmatrix}, \quad \text{right state: } \begin{pmatrix} \rho_{right} \\ q_{right} \end{pmatrix} = \begin{pmatrix} \frac{\rho_{left}}{2} \\ -100\rho_{left} \end{pmatrix}$$

We consider this Riemann problem for the isentropic Euler system (1.11). The problem is posed over $\Omega = (0, 1)$ and the discontinuity is initially located at $x = 0.5$. In figures 4.1 and 4.2, the solution displays a rarefaction (smooth) wave followed by a (discontinuous) shock wave. Our new method is able to capture both waves in a distinct and stable way.

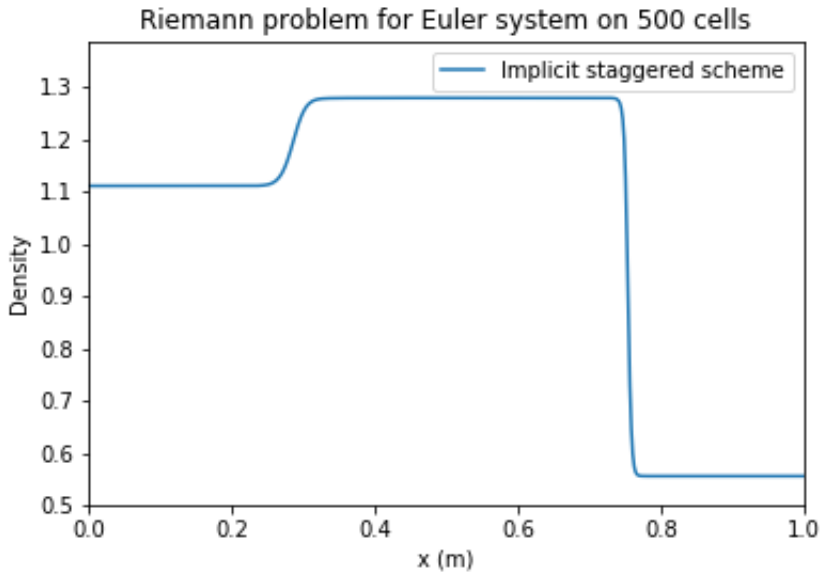


Figure 4.1: Density at time $t = 0.001$ with $\Delta x = 0.002$ and CFL= 0.99

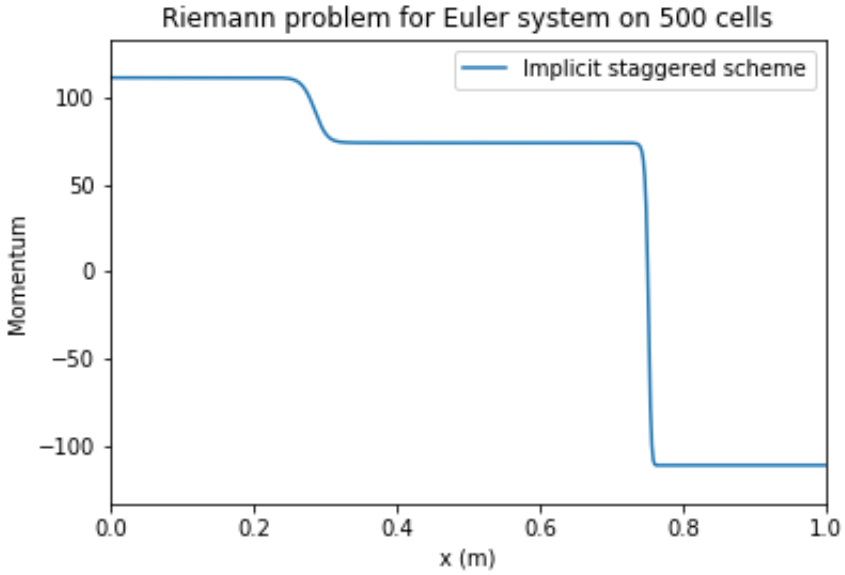


Figure 4.2: Momentum at time $t = 0.001$ with $\Delta x = 0.002$ and CFL= 0.99

## 4.5 Conclusion

In this chapter, we developed a rigorous framework for the $L^2$-stability analysis of finite volume schemes on staggered grids. The derivation of the method has required the analysis of some theoretical aspects beforehand. We also presented some analytical numerical examples on the solution of the isentropic Euler equations with the purpose of illustrating the technique and its performances. The family of schemes presented here could be applied in the future to the Cathare code to handle the numerical difficulties specific to two-phase flows models.
A major challenge for the simulation of two-phase flows is the configuration of the vanishing phase

where one of the phases disappears in some parts of the domain. The prediction of this complex dynamic mainly relies on the capture of the void waves that appear in the two-fluid model. Since the void waves have a complex structure with a propagation speed that frequently change signs, it is important for the numerical scheme to be stable regardless of the velocity sign. Our new class of staggered schemes is a promising alternative to the actual numerical treatment of the vanishing phase implemented in the Cathare code. The actual strategy relies on an interfacial friction coefficient that becomes high when one of the phases vanishes and has reached its limits for some test cases like the one of the Water-packing, that is relevant for nuclear safety studies. A first step toward the implementation of a new staggered scheme in the Cathare code could be the application to the reduced system of [93] that focuses on the study of the void waves.

# Appendix A

# The numerical diffusion of the Herbin et al staggered scheme

**Preliminary calculations for the mass equation** From (4.18) and (4.21) we have

$$
\begin{aligned}
\rho^{up}_{i+\frac{1}{2}} u_{i+\frac{1}{2}} &= \left(\frac{\rho_i+\rho_{i+1}}{2} + sign(u_{i+\frac{1}{2}})\frac{\rho_i-\rho_{i+1}}{2}\right) u_{i+\frac{1}{2}} \\
&= \left(\rho(x_i) + \frac{1}{2}\left(\Delta x \partial_x \rho(x_i) + \frac{1}{2}(\Delta x)^2 \partial_{xx}\rho(x_i) + \mathcal{O}(\Delta x^3)\right)(1 - sign(u(x_i)))\right) \\
&\quad \times \left(u(x_i) + \frac{\Delta x}{2}\partial_x u(x_i) + \frac{1}{2}\left(\frac{\Delta x}{2}\right)^2 \partial_{xx} u(x_i) + \mathcal{O}(\Delta x^3)\right) \\
&= \rho(x_i)u(x_i) + \frac{\Delta x}{2}\left(\rho(x_i)\partial_x u(x_i) + (1 - sign(u(x_i)))u(x_i)\partial_x\rho(x_i)\right) \\
&\quad + \frac{\Delta x^2}{4}\left(\rho(x_i)\frac{1}{2}\partial_{xx}u(x_i) + (1 - sign(u(x_i)))\left(u(x_i)\partial_{xx}\rho(x_i) + (\partial_x\rho)(x_i)(\partial_x u)(x_i)\right)\right) \\
&\quad + \mathcal{O}(\Delta x^3) \\
&= \rho(x_i)u(x_i) + \frac{\Delta x}{2}\left(\rho(x_i)\partial_x u(x_i) + (1 - sign(u(x_i)))u(x_i)\partial_x\rho(x_i)\right) \\
&\quad + \frac{\Delta x^2}{4}\left(\rho(x_i)\frac{1}{2}\partial_{xx}u(x_i) + (1 - sign(u(x_i)))\partial_x(u\partial_x\rho)(x_i)\right) \\
&\quad + \mathcal{O}(\Delta x^3) \qquad \text{(A.1)} \\
\rho^{up}_{i-\frac{1}{2}} u_{i-\frac{1}{2}} &= \left(\frac{\rho_{i-1}+\rho_i}{2} + sign(u_{i-\frac{1}{2}})\frac{\rho_{i-1}-\rho_i}{2}\right) u_{i-\frac{1}{2}} \\
&= \left(\rho(x_i) + \frac{1}{2}\left(-\Delta x \partial_x \rho(x_i) + \frac{1}{2}(\Delta x)^2 \partial_{xx}\rho(x_i) + \mathcal{O}(\Delta x^3)\right)(1 + sign(u(x_i)))\right) \\
&\quad \times \left(u(x_i) - \frac{\Delta x}{2}\partial_x u(x_i) + \frac{1}{2}\left(\frac{\Delta x}{2}\right)^2 \partial_{xx} u(x_i) + \mathcal{O}(\Delta x^3)\right) \\
&= \rho(x_i)u(x_i) - \frac{\Delta x}{2}\left(\rho(x_i)\partial_x u(x_i) + (1 + sign(u(x_i)))u(x_i)\partial_x\rho(x_i)\right) \\
&\quad + \frac{\Delta x^2}{4}\left(\rho(x_i)\frac{1}{2}\partial_{xx}u(x_i) + (1 + sign(u(x_i)))\left(u(x_i)\partial_{xx}\rho(x_i) + (\partial_x\rho)(x_i)(\partial_x u)(x_i)\right)\right) \\
&\quad + \mathcal{O}(\Delta x^3) \\
&= \rho(x_i)u(x_i) - \frac{\Delta x}{2}\left(\rho(x_i)\partial_x u(x_i) + (1 + sign(u(x_i)))u(x_i)\partial_x\rho(x_i)\right)
\end{aligned}
$$

$$+\frac{\Delta x^2}{4}\left(\rho(x_i)\frac{1}{2}\partial_{xx}u(x_i)+(1+sign(u(x_i)))\partial_x(u\partial_x\rho)(x_i)\right)$$
$$+\mathcal{O}(\Delta x^3). \tag{A.2}$$

**Preliminary calculations for the momentum equation**

$$\begin{aligned}
\overline{\rho u}_{i+1} &= \frac{1}{2}(\rho^{up}_{i+\frac{1}{2}}u_{i+\frac{1}{2}}+\rho^{up}_{i+\frac{3}{2}}u_{i+\frac{3}{2}})\\
&= \frac{1}{2}\left((\rho u)(x_{i+\frac{1}{2}})-\frac{\Delta x}{2}sign(u(x_{i+\frac{1}{2}}))u\partial_x\rho(x_{i+\frac{1}{2}})+\frac{\Delta x^2}{8}u\partial_{xx}\rho(x_{i+\frac{1}{2}})+\mathcal{O}(\Delta x^3)\right)\\
&\times \frac{1}{2}\left(\rho(x_{i+\frac{1}{2}})+\Delta x\partial_x\rho(x_{i+\frac{1}{2}})-\frac{\Delta x}{2}sign(u(x_{i+\frac{1}{2}}))\partial\rho(x_{i+\frac{1}{2}})+\frac{5\Delta x^2}{8}\partial_{xx}\rho(x_{i+\frac{1}{2}})\right.\\
&\quad \left.-\frac{\Delta x^2}{2}sign(u(x_{i+\frac{1}{2}}))\partial_{xx}\rho(x_{i+\frac{1}{2}})\right)\left(u(x_{i+\frac{1}{2}})+\Delta x\partial_x u(x_{i+\frac{1}{2}})+\frac{1}{2}(\Delta x)^2\partial_{xx}u(x_{i+\frac{1}{2}})\right)\\
&= (\rho u)(x_{i+\frac{1}{2}})+\frac{\Delta x}{2}\partial_x(\rho u)(x_{i+\frac{1}{2}})-\frac{\Delta x}{2}sign(u(x_{i+\frac{1}{2}}))u\partial_x\rho(x_{i+\frac{1}{2}})+\frac{3\Delta x^2}{8}u\partial_{xx}\rho(x_{i+\frac{1}{2}})\\
&+ \frac{\Delta x^2}{4}\rho\partial_{xx}u(x_{i+\frac{1}{2}})+\frac{\Delta x^2}{2}\partial_x\rho\partial_x u(x_{i+\frac{1}{2}})-\frac{\Delta x^2}{4}sign(u(x_{i+\frac{1}{2}}))\partial_x(u\partial_x\rho)(x_{i+\frac{1}{2}})+\mathcal{O}(\Delta x^3)
\end{aligned} \tag{A.3}$$

$$\begin{aligned}
\overline{\rho u}_{i} &= \frac{1}{2}(\rho^{up}_{i+\frac{1}{2}}u_{i+\frac{1}{2}}+\rho^{up}_{i-\frac{1}{2}}u_{i-\frac{1}{2}})\\
&= \frac{1}{2}\left((\rho u)(x_{i+\frac{1}{2}})-\frac{\Delta x}{2}sign(u(x_{i+\frac{1}{2}}))u\partial_x\rho(x_{i+\frac{1}{2}})+\frac{\Delta x^2}{8}u\partial_{xx}\rho(x_{i+\frac{1}{2}})+\mathcal{O}(\Delta x^3)\right)\\
&\times \frac{1}{2}\left(\rho(x_{i+\frac{1}{2}})-\Delta x\partial_x\rho(x_{i+\frac{1}{2}})-\frac{\Delta x}{2}sign(u(x_{i+\frac{1}{2}}))\partial\rho(x_{i+\frac{1}{2}})+\frac{5\Delta x^2}{8}\partial_{xx}\rho(x_{i+\frac{1}{2}})\right.\\
&\quad \left.+\frac{\Delta x^2}{2}sign(u(x_{i+\frac{1}{2}}))\partial_{xx}\rho(x_{i+\frac{1}{2}})\right)\left(u(x_{i+\frac{1}{2}})-\Delta x\partial_x u(x_{i+\frac{1}{2}})+\frac{1}{2}(\Delta x)^2\partial_{xx}u(x_{i+\frac{1}{2}})\right)\\
&= (\rho u)(x_{i+\frac{1}{2}})-\frac{\Delta x}{2}\partial_x(\rho u)(x_{i+\frac{1}{2}})-\frac{\Delta x}{2}sign(u(x_{i+\frac{1}{2}}))u\partial_x\rho(x_{i+\frac{1}{2}})+\frac{3\Delta x^2}{8}u\partial_{xx}\rho(x_{i+\frac{1}{2}})\\
&+ \frac{\Delta x^2}{4}\rho\partial_{xx}u(x_{i+\frac{1}{2}})+\frac{\Delta x^2}{2}\partial_x\rho\partial_x u(x_{i+\frac{1}{2}})+\frac{\Delta x^2}{4}sign(u(x_{i+\frac{1}{2}}))\partial_x(u\partial_x\rho)(x_{i+\frac{1}{2}})+\mathcal{O}(\Delta x^3)
\end{aligned} \tag{A.4}$$

$$\begin{aligned}
u^{up}_{i+1} &= \frac{u_{i+\frac{1}{2}}+u_{i+\frac{3}{2}}}{2}+sign(\overline{\rho u}_{i+1})\frac{u_{i+\frac{1}{2}}-u_{i+\frac{3}{2}}}{2}\\
&= u(x_{i+\frac{1}{2}})+\frac{1-sign(\rho u(x_{i+\frac{1}{2}}))}{2}\Delta x\partial_x u(x_{i+\frac{1}{2}})+\frac{1-sign(\rho u(x_{i+\frac{1}{2}}))}{4}\Delta x^2\partial_{xx}u(x_{i+\frac{1}{2}})+\mathcal{O}(\Delta x^3)
\end{aligned} \tag{A.5}$$

$$\begin{aligned}
u^{up}_{i} &= \frac{u_{i-\frac{1}{2}}+u_{i+\frac{1}{2}}}{2}+sign(\overline{\rho u}_{i})\frac{u_{i-\frac{1}{2}}-u_{i+\frac{1}{2}}}{2}\\
&= u(x_{i+\frac{1}{2}})-\frac{1+sign(\rho u(x_{i+\frac{1}{2}}))}{2}\Delta x\partial_x u(x_{i+\frac{1}{2}})+\frac{1+sign(\rho u(x_{i+\frac{1}{2}}))}{4}\Delta x^2\partial_{xx}u(x_{i+\frac{1}{2}})+\mathcal{O}(\Delta x^3)
\end{aligned} \tag{A.6}$$

From (A.3) and (A.5), we have:

$$\begin{aligned}
\overline{\rho u}_{i+1}u^{up}_{i+1} &= (\rho u^2)(x_{i+\frac{1}{2}})+\frac{\Delta x}{2}(1-sign(\rho u))\rho u\partial_x u+\frac{\Delta x^2}{4}(1-sign(\rho u))\rho u\partial_{xx}u+\frac{\Delta x}{2}u\partial_x\rho u\\
&+ \frac{\Delta x^2}{4}(1-sign(\rho u))\partial_x u\partial_x\rho u-\frac{\Delta x}{2}sign(u)u^2\partial_x\rho-\frac{\Delta x^2}{4}sign(u)(1-sign(\rho u))u\partial_x u\partial_x\rho\\
&+ \frac{3\Delta x^2}{8}u^2\partial_{xx}\rho+\frac{\Delta x^2}{4}\rho u\partial_{xx}u+\frac{\Delta x^2}{2}u\partial_x u\partial_x\rho-\frac{\Delta x^2}{4}sign(u)u\partial_x(u\partial_x\rho)+\mathcal{O}(\Delta x^3)
\end{aligned} \tag{A.7}$$

From (A.4) and (A.6), we have:

$$\begin{aligned}
\overline{\rho u}_i u_i^{up} &= (\rho u^2)(x_{i+\frac{1}{2}}) - \frac{\Delta x}{2}(1+sign(\rho u))\rho u\partial_x u + \frac{\Delta x^2}{4}(1+sign(\rho u))\rho u\partial_{xx}u - \frac{\Delta x}{2}u\partial_x\rho u \\
&+ \frac{\Delta x^2}{4}(1+sign(\rho u))\partial_x u\partial_x\rho u - \frac{\Delta x}{2}sign(u)u^2\partial_x\rho + \frac{\Delta x^2}{4}sign(u)(1+sign(\rho u))u\partial_x u\partial_x\rho \\
&+ \frac{3\Delta x^2}{8}u^2\partial_{xx}\rho + \frac{\Delta x^2}{4}\rho u\partial_{xx}u + \frac{\Delta x^2}{2}u\partial_x u\partial_x\rho + \frac{\Delta x^2}{4}sign(u)u\partial_x(u\partial_x\rho) + \mathcal{O}(\Delta x^3)
\end{aligned} \tag{A.8}$$

# Chapter 5

# A new class of entropic staggered schemes for the Euler equations

**Contents**



## 5.1 Introduction

In the previous chapter we have derived a linearly $L^2$ stable class of staggered schemes for the isentropic Euler equations. This linear stability has been characterised by using the properties of the numerical diffusion matrix. A $L^2$ stable scheme can however capture non entropic weak solutions as is the case with the Roe scheme ([38, 100]).
In this section, we focus on some non linear property of the scheme, namely the entropy property. We will characterise the staggered schemes that are entropic by analysing the properties of the diffusion matrix.
In section 5.1.1, we recall the derivation of the entropy-entropy flux pair for the isentropic Euler equations. The main result of this chapter is in section 5.2 with the definition of a new class of entropic staggered schemes. For the sake of simplicity, the analysis is performed in the one dimensional case with a linear state equation $p = \rho c^2$ but the strategy can be extended to the multidimensional case with a general state law $p(\rho)$. In the last section, we illustrate the performance of a prototype scheme belonging to the class of entropic staggered schemes on a one dimensional Riemann problem.

### 5.1.1 Entropy of the isentropic Euler system

The derivation of the entropy-entropy flux pair is recalled in this section for the particular case $p = \rho c^2$. The proof that the family of staggered schemes is entropic will follow the same lines

albeit using discrete quantities.

We recall that the mathematical entropy function in this case is

$$s = \rho\left(\frac{1}{2}||\vec{u}||^2 + c^2\ln\rho\right). \tag{5.1}$$

$s$ is strictly convex since its hessian matrix

$$H(s) = \begin{pmatrix} \frac{||\vec{q}||^2}{\rho^3} + \frac{c^2}{\rho} & -\frac{{}^t\vec{q}}{\rho^2} \\ -\frac{\vec{q}}{\rho^2} & \frac{1}{\rho} \end{pmatrix} \tag{5.2}$$

is positive definite since $c^2 > 0$.
Let us assume that $\vec{u}$ and $\rho$ are smooth functions solving

$$\left\{\begin{array}{lcl} \partial_t\rho & + & \nabla\cdot(\rho\vec{u}) = 0 \\ \rho\partial_t\vec{u} & + & \rho\vec{u}\cdot\nabla\vec{u} + \vec{\nabla}p = 0, \end{array}\right. \tag{5.3}$$

we compute

$$\begin{array}{rcl} \partial_t s & = & \left(\frac{1}{2}||\vec{u}||^2 + c^2\ln\rho\right)\partial_t\rho + \rho\left(\vec{u}\cdot\partial_t\vec{u} + \frac{c^2}{\rho}\partial_t\rho\right) \\ & = & \left(\frac{1}{2}||\vec{u}||^2 + c^2(1+\ln\rho)\right)\partial_t\rho + \rho\vec{u}\cdot\partial_t\vec{u}. \end{array} \tag{5.4}$$

Using the fact that $\vec{u}$ and $\rho$ solve (5.3), we obtain

$$\begin{array}{rcl} -\partial_t s & = & \left(\frac{1}{2}||\vec{u}||^2 + c^2(1+\ln\rho)\right)\nabla\cdot(\rho\vec{u}) + \rho\vec{u}\cdot(\vec{u}\cdot\vec{\nabla}u) + \vec{u}\cdot\vec{\nabla}p \\ & = & \frac{1}{2}||\vec{u}||^2\nabla\cdot(\rho\vec{u}) + c^2\ln\rho\nabla\cdot(\rho\vec{u}) + c^2\rho\nabla\cdot\vec{u} + c^2\vec{u}\cdot\vec{\nabla}\rho + \rho\vec{u}\cdot(\vec{u}\cdot\vec{\nabla}u) + \vec{u}\cdot\vec{\nabla}p. \end{array} \tag{5.5}$$

Since $p = \rho c^2$ and $c^2\rho\nabla\cdot\vec{u} + \vec{u}\cdot\vec{\nabla}p = \nabla\cdot(p\vec{u})$ we obtain

$$-\partial_t s = \frac{1}{2}||\vec{u}||^2\nabla\cdot(\rho\vec{u}) + c^2\ln\rho\nabla\cdot(\rho\vec{u}) + c^2\vec{u}\cdot\vec{\nabla}\rho + \rho\vec{u}\cdot(\vec{u}\cdot\vec{\nabla}u) + \nabla\cdot(p\vec{u}). \tag{5.6}$$

Since $\vec{u}\cdot\vec{\nabla}u = \frac{1}{2}\vec{\nabla}||\vec{u}||^2 + (\vec{\nabla}\times\vec{u})\times\vec{u}$ we obtain

$$\begin{array}{rcl} -\partial_t s & = & \frac{1}{2}||\vec{u}||^2\nabla\cdot(\rho\vec{u}) + c^2\ln\rho\nabla\cdot(\rho\vec{u}) + c^2\vec{u}\cdot\vec{\nabla}\rho + \frac{1}{2}\rho\vec{u}\cdot\vec{\nabla}||\vec{u}||^2 + \nabla\cdot(p\vec{u}) \\ & = & \frac{1}{2}\nabla\cdot(||\vec{u}||^2\rho\vec{u}) + c^2\ln\rho\nabla\cdot(\rho\vec{u}) + c^2\vec{u}\cdot\vec{\nabla}\rho + \nabla\cdot(p\vec{u}). \end{array} \tag{5.7}$$

Since $\ln\rho\nabla\cdot(\rho\vec{u}) + \vec{u}\cdot\vec{\nabla}\rho = \nabla\cdot(\rho\ln\rho\vec{u})$ we finally obtain

$$-\partial_t s = \nabla\cdot(\frac{1}{2}||\vec{u}||^2\rho\vec{u} + c^2\rho\ln\rho\vec{u} + p\vec{u}). \tag{5.8}$$

and finally

$$\partial_t s + \nabla\cdot((s+p)\vec{u}) = 0, \tag{5.9}$$

and $s$ is an entropy with the associated entropy flux

$$g(\vec{u},\rho) = \left(\rho\left(\frac{1}{2}||\vec{u}||^2 + c^2\ln\rho\right) + p\right)\vec{u}. \tag{5.10}$$

## 5.2 Entropy bound for a new class of staggered schemes

For the sake of pedagogy, we investigate the state law $p = \rho c^2$. For general 1D flows, we study the following scheme:

$$\partial_t \rho_i + \frac{\bar{q}_{i+\frac{1}{2}} - \bar{q}_{i-\frac{1}{2}}}{\triangle x} = 0, \tag{5.11}$$

$$\partial_t q_{i+\frac{1}{2}} + \frac{1}{\Delta x}\left(\frac{\bar{q}^2_{i+1}}{\rho_{i+1}} - \frac{\bar{q}^2_i}{\rho_i}\right) + \frac{p_{i+1} - p_i}{\triangle x} = 0, \tag{5.12}$$

In the sequel, we analyse the entropic character of this class of staggered schemes (5.11)-(5.12) that can be written in a more compact form as follows:

$$U_i'(t) + \frac{F_{i+1} - F_i}{\Delta x} = 0, \quad \text{with: } U_i = \begin{pmatrix} \rho_i \\ q_{i+1/2} \end{pmatrix}, \text{ and:} \tag{5.13}$$

$$F_{i+1} = \frac{F(U_i) + F(U_{i+1})}{2} + D_{Stag}(U_i, U_{i+1})\frac{U_i - U_{i+1}}{2} \tag{5.14}$$

where $D_{Stag}$ is a $2 \times 2$ matrix valued function:

$$D_{Stag}(U_i, U_{i+1}) = \begin{pmatrix} a_{i,i+1} & b_{i,i+1} \\ c_{i,i+1} & d_{i,i+1} \end{pmatrix}$$

Hence, the interpolated quantities $\bar{q}_{i+1/2}$, $\bar{q}_{i-1/2}$, $\frac{\bar{q}^2_{i+1}}{\rho_{i+1}}$ and $\frac{\bar{q}^2_i}{\rho_i}$ have the following expressions:

$$\begin{aligned}
\bar{q}_{i+1/2} &= \frac{q_{i+1/2} + q_{i+3/2}}{2} + a_{i,i+1}\frac{\rho_i - \rho_{i+1}}{2} + b_{i,i+1}\frac{q_{i+1/2} - q_{i+3/2}}{2} \\
&= q_{i+1/2} + a_{i,i+1}\frac{\rho_i - \rho_{i+1}}{2} + (b_{i,i+1} - 1)\frac{q_{i+1/2} - q_{i+3/2}}{2} \\
\frac{\bar{q}^2_{i+1}}{\rho_{i+1}} &= \frac{\frac{q^2_{i+1/2}}{\rho_i} + \frac{q^2_{i+3/2}}{\rho_{i+1}}}{2} + c_{i,i+1}\frac{\rho_i - \rho_{i+1}}{2} + d_{i,i+1}\frac{q_{i+1/2} - q_{i+3/2}}{2} \\
&= \frac{q^2_{i+1/2}}{\rho_i} - \frac{1}{2}\left(\frac{q^2_{i+1/2}}{\rho_i} - \frac{q^2_{i+3/2}}{\rho_{i+1}}\right) + c_{i,i+1}\frac{\rho_i - \rho_{i+1}}{2} + d_{i,i+1}\frac{q_{i+1/2} - q_{i+3/2}}{2} \\
&= \frac{q^2_{i+1/2}}{\rho_i} + (c_{i,i+1} + u^2_{i,i+1})\frac{\rho_i - \rho_{i+1}}{2} + (d_{i,i+1} - 2u_{i,i+1})\frac{q_{i+1/2} - q_{i+3/2}}{2}
\end{aligned}$$

where the Roe average is given by:

$$u_{i,i+1} = \frac{\frac{q_{i+1/2}}{\sqrt{\rho_i}} + \frac{q_{i+3/2}}{\sqrt{\rho_{i+1}}}}{\sqrt{\rho_i} + \sqrt{\rho_{i+1}}} \tag{5.15}$$

The coefficients $a_{i,i+1}, b_{i,i+1}, c_{i,i+1}$ and $d_{i,i+1}$ depend on the left state $U_i = \begin{pmatrix} \rho_i \\ q_{i+1/2} \end{pmatrix}$ and the right state $U_{i+1} = \begin{pmatrix} \rho_{i+1} \\ q_{i+3/2} \end{pmatrix}$. In the sequel, we will derive constraints on these coefficients to ensure the entropic character of a new class of staggered schemes.

From now on, we set the coefficient $b_{i,i+1} = 1$. This choice is motivated by the remarks 4.1 and 4.6 made in the previous chapter about the low Mach number accuracy of the upwind-type schemes and

the staggered schemes. Setting $b_{i,i+1} = 1$, we obtain the following expressions for the interpolated quantities:

$$\bar{q}_{i+1/2} = q_{i+1/2} + a_{i,i+1}\frac{\rho_i - \rho_{i+1}}{2}$$
$$\frac{\bar{q}^2_{i+1}}{\rho_{i+1}} = \frac{q^2_{i+1/2}}{\rho_i} + (c_{i,i+1} + u^2_{i,i+1})\frac{\rho_i - \rho_{i+1}}{2} + (d_{i,i+1} - 2u_{i,i+1})\frac{q_{i+1/2} - q_{i+3/2}}{2}$$

Hence, the class of staggered schemes we study in this section takes the form (5.13)-(5.14) where the matrix diffusion has the following generic expresssion:

$$D_{Stag}(U_i, U_{i+1}) = \begin{pmatrix} a_{i,i+1} & 1 \\ -c^2 + c_{i,i+1} & d_{i,i+1} \end{pmatrix} \tag{5.16}$$

where the term $-c^2$ comes from the discretisation of the pressure gradient.

**Theorem 5.1** (A class of entropic staggered schemes)**.** *A staggered conservative scheme (5.13) with a numerical flux $F_{i,i+1}$ of the form (5.14) such that, the coefficients $a_{i-1,i}$, $c_{i-1,i}$ and $d_{i-1,i}$ of the diffusion operator $D_{Stag}$ satisfy:*

$$\begin{aligned} E(a_{i-1,i}, c_{i-1,i}, d_{i-1,}) &= c^2 q_{i-1/2}\left(\frac{\rho_i}{\rho_{i-1}} - 1 - \ln\frac{\rho_i}{\rho_{i-1}}\right) + c^2\frac{a_{i-1,i}}{2}(\rho_{i-1} - \rho_i)\ln\frac{\rho_{i-1}}{\rho_i} \\ &+ \frac{1}{2}q_{i-1/2}\left(\frac{q_{i+1/2}}{\rho_i} - \frac{q_{i-1/2}}{\rho_{i-1}}\right)^2 + \frac{a_{i-1,i}}{4}(\rho_{i-1} - \rho_i)\left(\left(\frac{q_{i+1/2}}{\rho_i}\right)^2 - \left(\frac{q_{i-1/2}}{\rho_{i-1}}\right)^2\right) \\ &+ \frac{c_{i-1,i} + u^2_{i-1,i}}{2}(\rho_{i-1} - \rho_i)\left(\frac{q_{i-1/2}}{\rho_{i-1}} - \frac{q_{i+1/2}}{\rho_i}\right) \\ &+ \frac{d_{i-1,i} - 2u_{i-1,i}}{2}(q_{i+1/2} - q_{i-1/2})\left(\frac{q_{i+1/2}}{\rho_i} - \frac{q_{i-1/2}}{\rho_{i-1}}\right) \geq 0 \end{aligned} \tag{5.17}$$

*is entropic and the following discrete entropy dissipation estimate holds:*

$$\partial_t s_i + \frac{1}{\Delta x}\left(\tilde{g}\left(\frac{q_{i+1/2}}{\rho_i}, \rho_i\right) - \tilde{g}\left(\frac{q_{i-1/2}}{\rho_{i-1}}, \rho_{i-1}\right)\right) = -\frac{1}{\Delta x}E(a_{i-1,i}, c_{i-1,i}, d_{i-1,i}) \leq 0 \tag{5.18}$$

*where:*

$$\begin{aligned} \tilde{g}\left(\frac{q_{i+1/2}}{\rho_i}, \rho_i\right) &= \frac{1}{2}q_{i+1/2}\left(\frac{q_{i+1/2}}{\rho_i}\right)^2 + \frac{q_{i+1/2}}{\rho_i}p_{i+1} + c^2 q_{i+1/2}\ln\rho_i \\ &+ \frac{\rho_i - \rho_{i+1}}{2}\left((c_{i,i+1} + u^2_{i,i+1})\frac{q_{i+1/2}}{\rho_i} + a_{i,i+1}\left(c^2(\ln\rho_i + 1) - \frac{1}{2}\left(\frac{q_{i+1/2}}{\rho_i}\right)^2\right)\right) \\ &+ \frac{d_{i,i+1} - 2u_{i,i+1}}{2}\frac{q_{i+1/2}}{\rho_i}(q_{i+1/2} - q_{i+3/2}) \end{aligned} \tag{5.19}$$

*is the numerical entropy flux that is consistent with (5.10).*

*Proof.* Firstly, we seek to derive the discrete analog of the velocity evolution equation:

$$\rho\partial_t u + \rho u\partial_x u + \partial_x p = 0 \tag{5.20}$$

From (5.12), we derive:

$$\bar{\rho}_{i+1/2}\partial_t\frac{q_{i+1/2}}{\bar{\rho}_{i+1/2}} + \frac{q_{i+1/2}}{\bar{\rho}_{i+1/2}}\partial_t\bar{\rho}_{i+1/2} + \frac{1}{\Delta x}\left(\frac{\bar{q}^2_{i+1}}{\rho_{i+1}} - \frac{\bar{q}^2_i}{\rho_i}\right) + \frac{p_{i+1} - p_i}{\triangle x} = 0$$

Assuming $\bar{\rho}_{i+1/2} = \rho_i$, and from (5.11), we obtain:

$$\rho_i \partial_t \frac{q_{i+1/2}}{\rho_i} - \frac{q_{i+1/2}}{\rho_i} \frac{\bar{q}_{i+1/2} - \bar{q}_{i-1/2}}{\Delta x} + \frac{1}{\Delta x}\left(\frac{\bar{q}^2_{i+1}}{\rho_{i+1}} - \frac{\bar{q}^2_i}{\rho_i}\right) + \frac{p_{i+1} - p_i}{\triangle x} = 0$$

From the expression of the interpolated quantities $\bar{q}_{i+1/2}$ and $\dfrac{\bar{q}^2_{i+1}}{\rho_{i+1}}$, this yields to:

$$\begin{aligned}
\rho_i \partial_t \frac{q_{i+1/2}}{\rho_i} \quad &+ \quad q_{i-1/2} \frac{\frac{q_{i+1/2}}{\rho_i} - \frac{q_{i-1/2}}{\rho_{i-1}}}{\Delta x} + \frac{p_{i+1} - p_i}{\triangle x} \\
&+ \quad \frac{\rho_i - \rho_{i+1}}{2\Delta x}\left((c_{i,i+1} + u^2_{i,i+1}) - a_{i,i+1}\frac{q_{i+1/2}}{\rho_i}\right) + \frac{\rho_{i-1} - \rho_i}{2\Delta x}\left(a_{i-1,i}\frac{q_{i+1/2}}{\rho_i} - (c_{i-1,i} + u^2_{i-1,i})\right) \\
&+ \quad \frac{d_{i,i+1} - 2u_{i,i+1}}{2\Delta x}(q_{i+1/2} - q_{i+3/2}) + \frac{d_{i-1,i} - 2u_{i-1,i}}{2\Delta x}(q_{i+1/2} - q_{i-1/2}) = 0
\end{aligned} \tag{5.21}$$

Since $s = \rho\left(\frac{1}{2}||u||^2 + c^2 \ln \rho\right)$, we have:

$$\begin{aligned}
\partial_t s_i &= \partial_t \left(\rho_i \left(\frac{1}{2}\left(\frac{q_{i+1/2}}{\rho_i}\right)^2 + c^2 \ln \rho_i\right)\right) \\
&= \partial_t \rho_i \left(\frac{1}{2}\left(\frac{q_{i+1/2}}{\rho_i}\right)^2 + c^2 \ln \rho_i\right) + \rho_i \frac{q_{i+1/2}}{\rho_i} \partial_t \frac{q_{i+1/2}}{\rho_i} + c^2 \partial_t \rho_i
\end{aligned}$$

From (5.21) and (5.12), we obtain:

$$\begin{aligned}
-\partial_t s_i = &\left(\frac{1}{2}\left(\frac{q_{i+1/2}}{\rho_i}\right)^2 + c^2 \ln \rho_i\right) \frac{\bar{q}_{i+1/2} - \bar{q}_{i-1/2}}{\Delta x} + \frac{q_{i+1/2}}{\rho_i} q_{i-1/2} \frac{\frac{q_{i+1/2}}{\rho_i} - \frac{q_{i-1/2}}{\rho_{i-1}}}{\Delta x} + \frac{q_{i+1/2}}{\rho_i} \frac{p_{i+1} - p_i}{\triangle x} \\
&+ \frac{\rho_i - \rho_{i+1}}{2\Delta x} \frac{q_{i+1/2}}{\rho_i}\left((c_{i,i+1} + u^2_{i,i+1}) - a_{i,i+1}\frac{q_{i+1/2}}{\rho_i}\right) \\
&+ \frac{q_{i+1/2}}{\rho_i} \frac{\rho_{i-1} - \rho_i}{2\Delta x}\left(a_{i-1,i}\frac{q_{i+1/2}}{\rho_i} - (c_{i-1,i} + u^2_{i-1,i})\right) + \frac{d_{i,i+1} - 2u_{i,i+1}}{2\Delta x} \frac{q_{i+1/2}}{\rho_i}(q_{i+1/2} - q_{i+3/2}) \\
&+ \frac{d_{i-1,i} - 2u_{i-1,i}}{2\Delta x} \frac{q_{i+1/2}}{\rho_i}(q_{i+1/2} - q_{i-1/2}) + c^2 \frac{\bar{q}_{i+1/2} - \bar{q}_{i-1/2}}{\Delta x}
\end{aligned}$$

Using:

$$q_{i+1/2} - q_{i-1/2} = \frac{q_{i-1/2}}{\rho_{i-1}}(\rho_i - \rho_{i-1}) + \rho_i\left(\frac{q_{i+1/2}}{\rho_i} - \frac{q_{i-1/2}}{\rho_{i-1}}\right),$$

and:

$$c^2 \rho_i\left(\frac{q_{i+1/2}}{\rho_i} - \frac{q_{i-1/2}}{\rho_{i-1}}\right) + \frac{q_{i+1/2}}{\rho_i}(p_{i+1} - p_i) = \frac{q_{i+1/2}}{\rho_i} p_{i+1} - \frac{q_{i-1/2}}{\rho_{i-1}} p_i,$$

We obtain:

$$\begin{aligned}
-\Delta x \partial_t s_i = &\frac{1}{2} q_{i+1/2}\left(\frac{q_{i+1/2}}{\rho_i}\right)^2 - \frac{1}{2} q_{i-1/2}\left(\frac{q_{i-1/2}}{\rho_{i-1}}\right)^2 + \frac{1}{2} q_{i-1/2}\left(\frac{q_{i+1/2}}{\rho_i} - \frac{q_{i-1/2}}{\rho_{i-1}}\right)^2 \\
&+ \frac{q_{i+1/2}}{\rho_i} p_{i+1} - \frac{q_{i-1/2}}{\rho_{i-1}} p_i + c^2\left((q_{i+1/2} - q_{i-1/2}) \ln \rho_i + q_{i-1/2}\left(\frac{\rho_i}{\rho_{i-1}} - 1\right)\right) \\
&+ \frac{\rho_i - \rho_{i+1}}{2}\left((c_{i,i+1} + u^2_{i,i+1})\frac{q_{i+1/2}}{\rho_i} + a_{i,i+1}\left(c^2(\ln \rho_i + 1) - \frac{1}{2}\left(\frac{q_{i+1/2}}{\rho_i}\right)^2\right)\right)
\end{aligned}$$

$$+\frac{\rho_{i-1}-\rho_i}{2}\left(a_{i-1,i}\left(\frac{1}{2}\left(\frac{q_{i+1/2}}{\rho_i}\right)^2-c^2(\ln\rho_i+1)\right)-(c_{i-1,i}+u_{i-1,i}^2)\frac{q_{i+1/2}}{\rho_i}\right)$$
$$+\frac{d_{i,i+1}-2u_{i,i+1}}{2}\frac{q_{i+1/2}}{\rho_i}(q_{i+1/2}-q_{i+3/2})$$
$$+\frac{d_{i-1,i}-2u_{i-1,i}}{2}\frac{q_{i+1/2}}{\rho_i}(q_{i+1/2}-q_{i-1/2})$$

In order to deal with the term $(q_{i+1/2}-q_{i-1/2})\ln\rho_i$, we add and substract the quantity

$$c^2 q_{i-1/2}\ln\frac{\rho_i}{\rho_{i-1}}. \tag{5.22}$$

In order to absorb the terms containing the variables $\rho_{i+1}$ and $q_{i+3/2}$ into flux differences, we add and substract the quantities:

$$\frac{\rho_{i-1}-\rho_i}{2}\left((c_{i-1,i}+u_{i-1,i}^2)\frac{q_{i-1/2}}{\rho_{i-1}}+a_{i-1,i}\left(c^2(\ln\rho_{i-1}+1)-\frac{1}{2}\left(\frac{q_{i-1/2}}{\rho_{i-1}}\right)^2\right)\right) \tag{5.23}$$

and

$$\frac{d_{i-1,i}-2u_{i-1,i}}{2}\frac{q_{i-1/2}}{\rho_{i-1}}(q_{i-1/2}-q_{i+1/2}) \tag{5.24}$$

We obtain:

$$\begin{aligned}
-\Delta x\partial_t s_i=&\frac{1}{2}\left[q_{-1/2}\left(\frac{q_{-1/2}}{\rho}\right)^2\right]_i^{i+1}+\left[\frac{q_{-1/2}}{\rho}p_{+1}\right]_i^{i+1}+c^2\left[q_{+1/2}\ln\rho\right]_i^{i+1}\\
&+\frac{1}{2}q_{i-1/2}\left(\frac{q_{i+1/2}}{\rho_i}-\frac{q_{i-1/2}}{\rho_{i-1}}\right)^2+c^2q_{i-1/2}\left(\frac{\rho_i}{\rho_{i-1}}-1-\ln\frac{\rho_i}{\rho_{i-1}}\right)\\
&+\left[\frac{\rho_{i-1}-\rho_i}{2}\left((c_{i-1,i}+u_{i-1,i}^2)\frac{q_{i-1/2}}{\rho_{i-1}}+a_{i-1,i}\left(c^2(\ln\rho_{i-1}+1)-\frac{1}{2}\left(\frac{q_{i-1/2}}{\rho_{i-1}}\right)^2\right)\right)\right]_i^{i+1}\\
&+\left[\frac{d_{i-1,i}-2u_{i-1,i}}{2}\frac{q_{i-1/2}}{\rho_{i-1}}(q_{i-1/2}-q_{i+1/2})\right]_i^{i+1}\\
&+\frac{\rho_{i-1}-\rho_i}{2}a_{i-1,i}\left(\frac{1}{2}\left(\frac{q_{i+1/2}}{\rho_i}\right)^2-\frac{1}{2}\left(\frac{q_{i-1/2}}{\rho_{i-1}}\right)^2+c^2\ln\frac{\rho_{i-1}}{\rho_i}\right)\\
&+\frac{\rho_{i-1}-\rho_i}{2}(c_{i-1,i}+u_{i-1,i}^2)\left(\frac{q_{i-1/2}}{\rho_{i-1}}-\frac{q_{i+1/2}}{\rho_i}\right)\\
&+\frac{d_{i-1,i}-2u_{i-1,i}}{2}(q_{i+1/2}-q_{i-1/2})\left(\frac{q_{i+1/2}}{\rho_i}-\frac{q_{i-1/2}}{\rho_{i-1}}\right)
\end{aligned}$$

which becomes:

$$\begin{aligned}
-\Delta x\partial_t s_i=&\tilde{g}\left(\frac{q_{i+1/2}}{\rho_i},\rho_i\right)-\tilde{g}\left(\frac{q_{i-1/2}}{\rho_{i-1}},\rho_{i-1}\right)\\
&+c^2q_{i-1/2}\left(\frac{\rho_i}{\rho_{i-1}}-1-\ln\frac{\rho_i}{\rho_{i-1}}\right)+c^2\frac{a_{i-1,i}}{2}(\rho_{i-1}-\rho_i)\ln\frac{\rho_{i-1}}{\rho_i}\\
&+\frac{1}{2}q_{i-1/2}\left(\frac{q_{i+1/2}}{\rho_i}-\frac{q_{i-1/2}}{\rho_{i-1}}\right)^2+\frac{a_{i-1,i}}{4}(\rho_{i-1}-\rho_i)\left(\left(\frac{q_{i+1/2}}{\rho_i}\right)^2-\left(\frac{q_{i-1/2}}{\rho_{i-1}}\right)^2\right)
\end{aligned}$$

$$+\frac{c_{i-1,i}+u_{i-1,i}^2}{2}(\rho_{i-1}-\rho_i)\left(\frac{q_{i-1/2}}{\rho_{i-1}}-\frac{q_{i+1/2}}{\rho_i}\right)$$
$$+\frac{d_{i-1,i}-2u_{i-1,i}}{2}(q_{i+1/2}-q_{i-1/2})\left(\frac{q_{i+1/2}}{\rho_i}-\frac{q_{i-1/2}}{\rho_{i-1}}\right)$$

□

**Corollary 5.2.** *The numerical scheme (5.13) with a numerical flux $F_{i,i+1}$ of the form (5.14) such that the coefficients $a_{i-1,i}$, $c_{i-1,i}$ and $d_{i-1,i}$ of the diffusion operator $D_{Stag}$ satisfy:*

$$\left\{\begin{array}{rcl} c_{i-1,i} &=& 2u_{i-1,i}\left(\frac{q_{i-1/2}}{\rho_{i-1}}-\frac{q_{i+1/2}}{\rho_i}\right)-u_{i-1,i}^2 \\ d_{i-1,i} &=& a_{i-1,i}+2u_{i-1,i} \end{array}\right. \tag{5.25}$$

*where $u_{i-1,i}$ is the Roe average (5.15) and:*

$$a_{i-1,i}\geq max\left(-2q_{i-1/2}\frac{\frac{\rho_i}{\rho_{i-1}}-1-\ln\frac{\rho_i}{\rho_{i-1}}}{(\rho_{i-1}-\rho_i)\ln\frac{\rho_{i-1}}{\rho_i}},-\frac{4}{\rho_{i-1}+\rho_i}\left(\frac{1}{2}q_{i-1/2}+u_{i-1,i}(\rho_{i-1}-\rho_i)\right)\right) \tag{5.26}$$

*is entropic and the discrete entropy dissipation estimate (5.18) holds. The numerical entropy flux is given by (5.19) and is consistent with (5.10).*

*Proof.* Let us recall the entropy balance of our class of entropic staggered schemes:

$$\begin{array}{rcl} -\Delta x\partial_t s_i &=& \tilde{g}\left(\frac{q_{i+1/2}}{\rho_i},\rho_i\right)-\tilde{g}\left(\frac{q_{i-1/2}}{\rho_{i-1}},\rho_{i-1}\right) \\ &+& c^2 q_{i-1/2}\left(\frac{\rho_i}{\rho_{i-1}}-1-\ln\frac{\rho_i}{\rho_{i-1}}\right)+c^2\frac{a_{i-1,i}}{2}(\rho_{i-1}-\rho_i)\ln\frac{\rho_{i-1}}{\rho_i} \\ &+& \frac{1}{2}q_{i-1/2}\left(\frac{q_{i+1/2}}{\rho_i}-\frac{q_{i-1/2}}{\rho_{i-1}}\right)^2+\frac{a_{i-1,i}}{4}(\rho_{i-1}-\rho_i)\left(\left(\frac{q_{i+1/2}}{\rho_i}\right)^2-\left(\frac{q_{i-1/2}}{\rho_{i-1}}\right)^2\right) \\ &+& \frac{c_{i-1,i}+u_{i-1,i}^2}{2}(\rho_{i-1}-\rho_i)\left(\frac{q_{i-1/2}}{\rho_{i-1}}-\frac{q_{i+1/2}}{\rho_i}\right) \\ &+& \frac{d_{i-1,i}-2u_{i-1,i}}{2}\frac{q_{i-1/2}}{\rho_{i-1}}(\rho_i-\rho_{i-1})\left(\frac{q_{i+1/2}}{\rho_i}-\frac{q_{i-1/2}}{\rho_{i-1}}\right) \\ &+& \frac{d_{i-1,i}-2u_{i-1,i}}{2}\rho_i\left(\frac{q_{i+1/2}}{\rho_i}-\frac{q_{i-1/2}}{\rho_{i-1}}\right)^2 \end{array} \tag{5.27}$$

We define $\alpha_{i-1,i}$ as:

$$\alpha_{i-1,i}=a_{i-1,i}-|u_{i-1,i}|+u_{i-1,i}$$

Hence the coefficients $a_{i-1,i}$ and $d_{i-1,i}$ are defined as:

$$a_{i-1,i}=|u_{i-1,i}|-u_{i-1,i}+\alpha_{i-1,i}$$
$$d_{i-1,i}=|u_{i-1,i}|+u_{i-1,i}+\alpha_{i-1,i}$$

The entropy balance (5.27) becomes:

$$\begin{aligned}
-\Delta x \partial_t s_i &= \tilde{g}\left(\frac{q_{i+1/2}}{\rho_i}, \rho_i\right) - \tilde{g}\left(\frac{q_{i-1/2}}{\rho_{i-1}}, \rho_{i-1}\right) \\
&+ c^2 q_{i-1/2}\left(\frac{\rho_i}{\rho_{i-1}} - 1 - \ln\frac{\rho_i}{\rho_{i-1}}\right) + c^2 \frac{|u_{i-1,i}| - u_{i-1,i} + \alpha_{i-1,i}}{2}(\rho_{i-1} - \rho_i)\ln\frac{\rho_{i-1}}{\rho_i} \\
&+ \frac{1}{2} q_{i-1/2}\left(\frac{q_{i+1/2}}{\rho_i} - \frac{q_{i-1/2}}{\rho_{i-1}}\right)^2 + \frac{|u_{i-1,i}| - u_{i-1,i} + \alpha_{i-1,i}}{4}(\rho_{i-1} - \rho_i)\left(\left(\frac{q_{i+1/2}}{\rho_i}\right)^2 - \left(\frac{q_{i-1/2}}{\rho_{i-1}}\right)^2\right) \\
&+ \frac{c_{i-1,i} + u_{i-1,i}^2}{2}(\rho_{i-1} - \rho_i)\left(\frac{q_{i-1/2}}{\rho_{i-1}} - \frac{q_{i+1/2}}{\rho_i}\right) \\
&+ \frac{|u_{i-1,i}| - u_{i-1,i} + \alpha_{i-1,i}}{2}\frac{q_{i-1/2}}{\rho_{i-1}}(\rho_i - \rho_{i-1})\left(\frac{q_{i+1/2}}{\rho_i} - \frac{q_{i-1/2}}{\rho_{i-1}}\right) \\
&+ \frac{|u_{i-1,i}| - u_{i-1,i} + \alpha_{i-1,i}}{2}\rho_i\left(\frac{q_{i+1/2}}{\rho_i} - \frac{q_{i-1/2}}{\rho_{i-1}}\right)^2
\end{aligned} \tag{5.28}$$

From hypothesis (5.25), the coefficient $c_{i-1,i}$ is defined as:

$$c_{i-1,i} = 2u_{i-1,i}\left(\frac{q_{i-1/2}}{\rho_{i-1}} - \frac{q_{i+1/2}}{\rho_i}\right) - u_{i-1,i}^2 \tag{5.29}$$

and we finally obtain:

$$\begin{aligned}
-\Delta x \partial_t s_i &= \tilde{g}\left(\frac{q_{i+1/2}}{\rho_i}, \rho_i\right) - \tilde{g}\left(\frac{q_{i-1/2}}{\rho_{i-1}}, \rho_{i-1}\right) \\
&+ c^2\left(q_{i-1/2}\left(\frac{\rho_i}{\rho_{i-1}} - 1 - \ln\frac{\rho_i}{\rho_{i-1}}\right) + \frac{a_{i-1,i}}{2}(\rho_{i-1} - \rho_i)\ln\frac{\rho_{i-1}}{\rho_i}\right) \\
&+ \left(\frac{q_{i+1/2}}{\rho_i} - \frac{q_{i-1/2}}{\rho_{i-1}}\right)^2\left(\frac{1}{2}q_{i-1/2} + \frac{a_{i-1,i}}{4}(\rho_{i-1} + \rho_i) + u_{i-1,i}(\rho_{i-1} - \rho_i)\right)
\end{aligned} \tag{5.30}$$

This term is positive provided the upwinding coefficient $a_{i-1,i}$ is large enough. The threshold values are

$$a_{i-1,i} = |u_{i-1,i}| - u_{i-1,i} + \alpha_{i-1,i} \geq -2q_{i-1/2}\frac{\frac{\rho_i}{\rho_{i-1}} - 1 - \ln\frac{\rho_i}{\rho_{i-1}}}{(\rho_{i-1} - \rho_i)\ln\frac{\rho_{i-1}}{\rho_i}} \tag{5.31}$$

and

$$a_{i-1,i} = |u_{i-1,i}| - u_{i-1,i} + \alpha_{i-1,i} \geq -\frac{4}{\rho_{i-1} + \rho_i}\left(\frac{1}{2}q_{i-1/2} + u_{i-1,i}(\rho_{i-1} - \rho_i)\right) \tag{5.32}$$

□

**Remark 5.3.** ***Particular case:*** $\rho_i - \rho_{i-1} \to 0$*:*

*Here, we give the asymptotic behaviour, when* $(\rho_i - \rho_{i-1})$ *tends to zero, of the coefficient* $a_{i-1,i}$ *that satisfies the condition (5.26).*

*When* $\dfrac{\rho_i}{\rho_{i-1}} - 1 \to 0$*, we have:*

$$\frac{\frac{\rho_i}{\rho_{i-1}} - 1 - \ln\frac{\rho_i}{\rho_{i-1}}}{(\rho_{i-1} - \rho_i)\ln\frac{\rho_{i-1}}{\rho_i}} = \frac{\frac{1}{2}\left(\frac{\rho_i}{\rho_{i-1}} - 1\right)^2 + \mathcal{O}\left(\left(\frac{\rho_i}{\rho_{i-1}} - 1\right)^3\right)}{\left(\frac{\rho_i}{\rho_{i-1}} - 1\right)^2 + \mathcal{O}\left(\left(\frac{\rho_i}{\rho_{i-1}} - 1\right)^3\right)}$$

$$\leq \frac{1}{2} + C, \text{ where } C \text{ is a constant.}$$

*Hence, the first lower bound (5.31) of the coefficient* $a_{i-1,i}$ *in (5.26) is asymptotically bounded when* $(\rho_i - \rho_{i-1})$ *tends to zero.*

## 5.3 Numerical results

### 5.3.1 Entropy default of the $L^2$-stable staggered scheme from Corollary 4.5

In this section, we discuss the entropic character of the $L^2$-stable staggered scheme introduced in Chapter 4, (equation 4.54) in the one dimensional case. This staggered scheme has the following upwinding matrix:

$$D_{Stag}(U_i, U_{i+1}) = \begin{pmatrix} |u_{i,i+1}| - u_{i,i+1} & 1 \\ -c^2 - u_{i,i+1}^2 & |u_{i,i+1}| + u_{i,i+1} \end{pmatrix}$$

In practice, the lack of entropy decrease results in a shock wave being captured instead of a rarefaction wave. This is particularly the case for Riemann solvers ([38, 100]) where the violation of entropy decrease leads to the capture of a non entropic stationnary shock wave instead of a transonic rarefaction wave. Since the upwinding matrix of our class of staggered schemes is not based on the eigenvalues of the system, it is not straightforward to choose a numerical test that will show the entropy default. Hence, we need the entropy balance (5.33) to identify the configurations that could lead to the capture of non entropic solutions.
For the $L^2$-stable staggered scheme (4.54), we have:

$$\partial_t s_i + \frac{1}{\Delta x}\left(\tilde{g}\left(\frac{q_{i+1/2}}{\rho_i}, \rho_i\right) - \tilde{g}\left(\frac{q_{i-1/2}}{\rho_{i-1}}, \rho_{i-1}\right)\right) = -\frac{1}{\Delta x} E(a_{i-1,i}, c_{i-1,i}, d_{i-1,i})$$

with:

$$\begin{aligned} E(a_{i-1,i}, c_{i-1,i}, d_{i-1,i}) &= c^2\left(q_{i-1/2}\left(\frac{\rho_i}{\rho_{i-1}} - 1 - \ln\frac{\rho_i}{\rho_{i-1}}\right) + \frac{|u_{i-1,i}| - u_{i-1,i}}{2}(\rho_{i-1} - \rho_i)\ln\frac{\rho_{i-1}}{\rho_i}\right) \\ &+ \left(\frac{q_{i+1/2}}{\rho_i} - \frac{q_{i-1/2}}{\rho_{i-1}}\right)^2\left(\frac{1}{2}q_{i-1/2} + \frac{|u_{i-1,i}| - u_{i-1,i}}{4}(\rho_{i-1} + \rho_i)\right) \end{aligned} \tag{5.33}$$

From (5.33), we see that the scheme violates the entropy decrease property in the two following cases:

- For constant densities, if $q_{i-1/2} < 0$ and $u_{i-1,i} = 0$, then $E(a_{i-1,i}, c_{i-1,i}, d_{i-1,i}) < 0$. Hence, in this case, the $L^2$-stable scheme locally generates a positive contribution of the entropy in the order of $|q_{i-1/2}|^2$.
- For variable densities and high negative velocities $u_{i-1/2}$, then $E(a_{i-1,i}, c_{i-1,i}, d_{i-1,i}) < 0$. Hence, in this case, the $L^2$-stable scheme locally generates a positive contribution of the entropy in the order of $c^2$.

The largest amount of entropy creation occurs in the second case, because the entropy creation is proportional to $c^2$. In the sequel, we solve Riemann problems for the isentropic Euler system with initial conditions such that the scheme initially violates the entropy decrease.

#### 5.3.1.1 Entropy default for negative high velocities

In the sequel, we assess the behaviour of the $L^2$-stable staggered scheme (4.54) on a one dimensional Riemann problem. We choose initial conditions such that the structure of the solution consists in a shock followed by a rarefaction wave to allow an easy discrimination of correct numerical solutions. These initial conditions are:

$$\text{left state: } \begin{pmatrix}\rho_{left}\\ q_{left}\end{pmatrix} = \begin{pmatrix}1\\ \rho_{left}u_0\end{pmatrix}, \quad \text{right state: } \begin{pmatrix}\rho_{right}\\ q_{right}\end{pmatrix} = \begin{pmatrix}2\\ \rho_{right}u_0\end{pmatrix} \tag{5.34}$$

We consider this Riemann problem for the isentropic Euler system (1.11). The problem is posed over $\Omega = (0, 1)$ and the discontinuity is initially located at $x = 0.5$. This case illustrates the second configuration of entropy default for the $L^2$-stable scheme with variable densities and an entropy creation proportional to $c^2$.
In figures 5.1 and 5.2, the $L^2$-stable staggered scheme captures a shock wave followed by a rarefaction wave. Hence, the method captures the correct entropic solution despite the initial entropy creation proportional to $c^2$. Hence for long time simulations the entropy default we observe in theory in (5.33) does not lead to the capture of non entropic solutions.

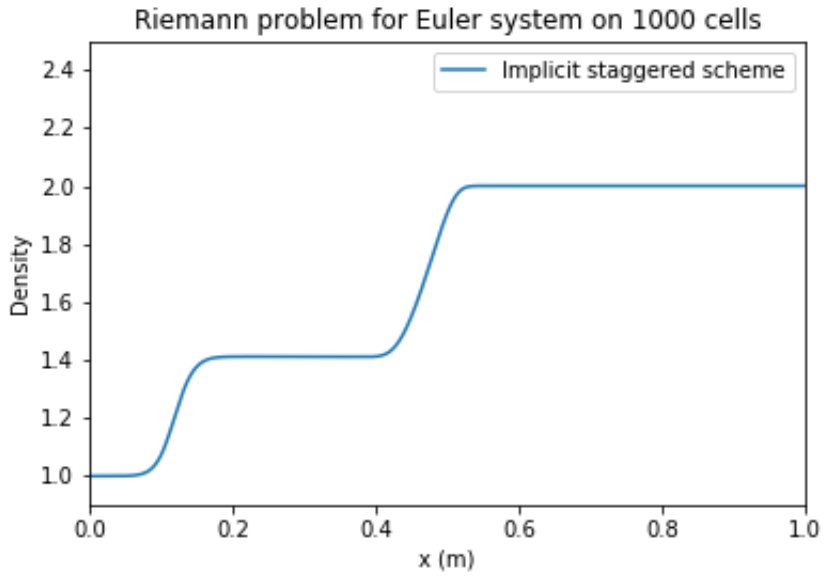


Figure 5.1: Density at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99 for $u_0 = -300$

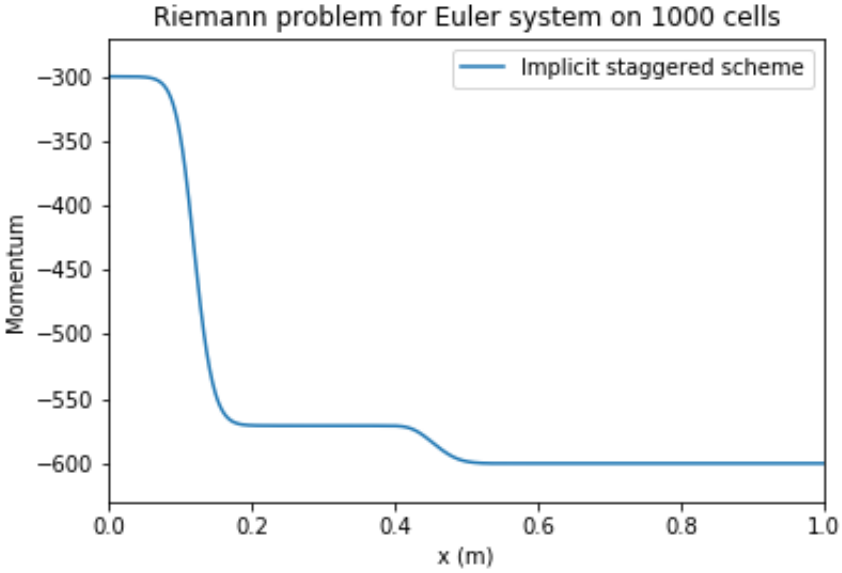


Figure 5.2: Momentum at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99 for $u_0 = -300$

#### 5.3.1.2 Case of high and low Mach number flows

In this section, the numerical tests we choose do not generate an entropy violation in the entropy balance (5.33) of the $L^2$-stable staggered scheme. They illustrate the behaviour of the $L^2$-stable scheme for high and low Mach number flows. Here, we solve the Riemann problem with the initial condition (5.34) where $u_0 = 300$ (see figures 5.3 and 5.4) and $u_0 = -1$ (see figures 5.5 and 5.6). In these two cases, our conservative staggered scheme is able to capture shock and rarefaction waves without any prior information on the characteristic fields.

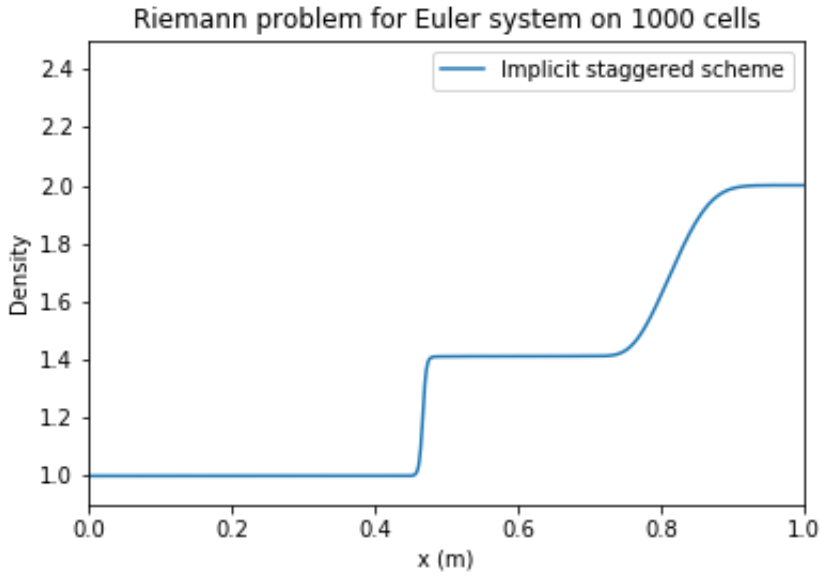


Figure 5.3: Density at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99 for $u_0 = 300$

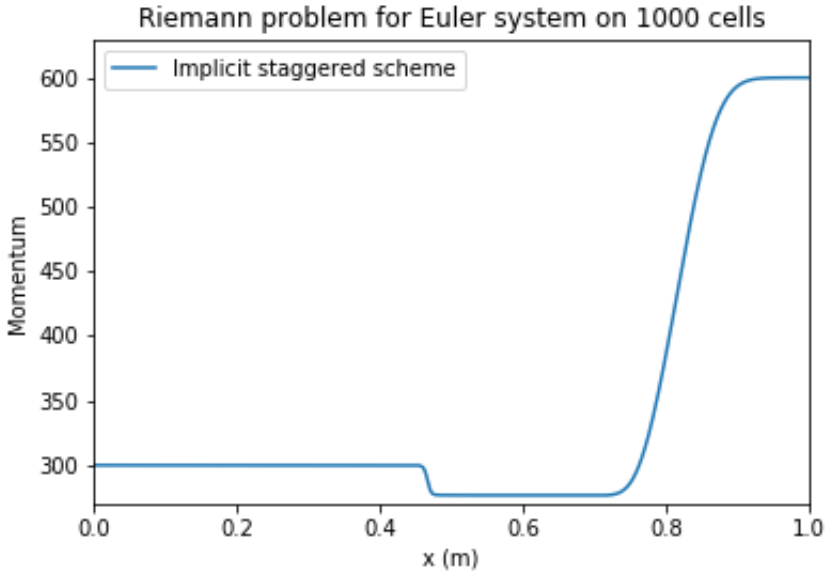


Figure 5.4: Momentum at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99 for $u_0 = 300$

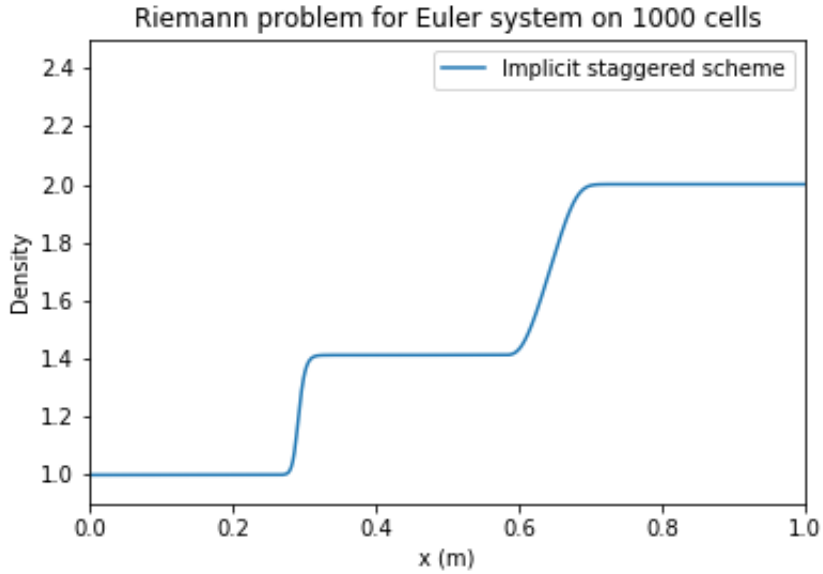


Figure 5.5: Density at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99 for $u_0 = -1$

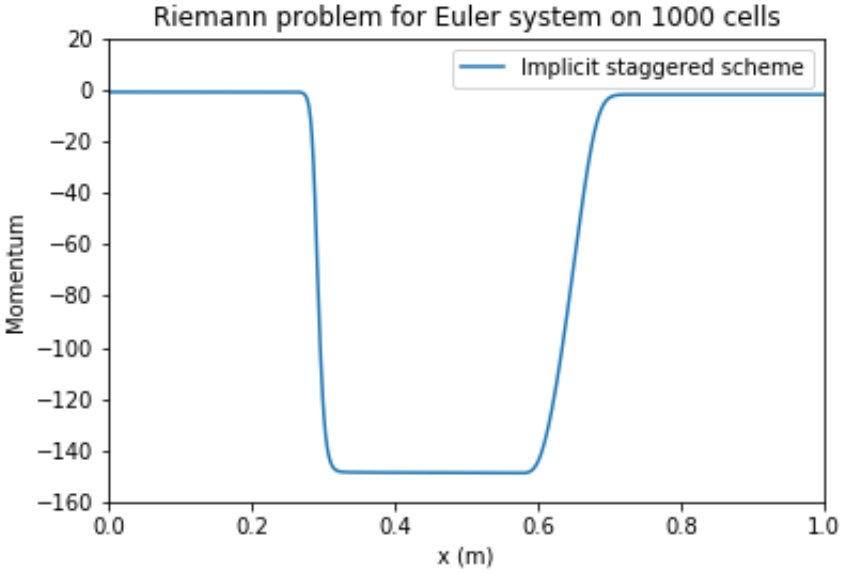


Figure 5.6: Momentum at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99 for $u_0 = -1$

#### 5.3.1.3 Entropy default for constant densities

In this section, we consider the Riemann problem where the initial conditions are chosen such that the structure of the solution consists in two transonic rarefaction waves:

$$\text{left state: } \begin{pmatrix} \rho_{left} \\ q_{left} \end{pmatrix} = \begin{pmatrix} 1 \\ -300 \end{pmatrix}, \quad \text{right state: } \begin{pmatrix} \rho_{right} \\ q_{right} \end{pmatrix} = \begin{pmatrix} 1 \\ 300 \end{pmatrix} \tag{5.35}$$

This case illustrates the first configuration of entropy default for the $L^2$-stable scheme with constant densities and an entropy creation proportional to $|q_{i-1/2}|^2$ (see introduction of section 5.1.1). In figures 5.7 and 5.8, we see that despite the initial entropy creation the scheme captures the correct entropic solution of the Riemann problem with two transonic rarefaction waves. Hence for long time simulations the entropy default we observe in theory in (5.33) does not lead to the capture of non entropic solutions.

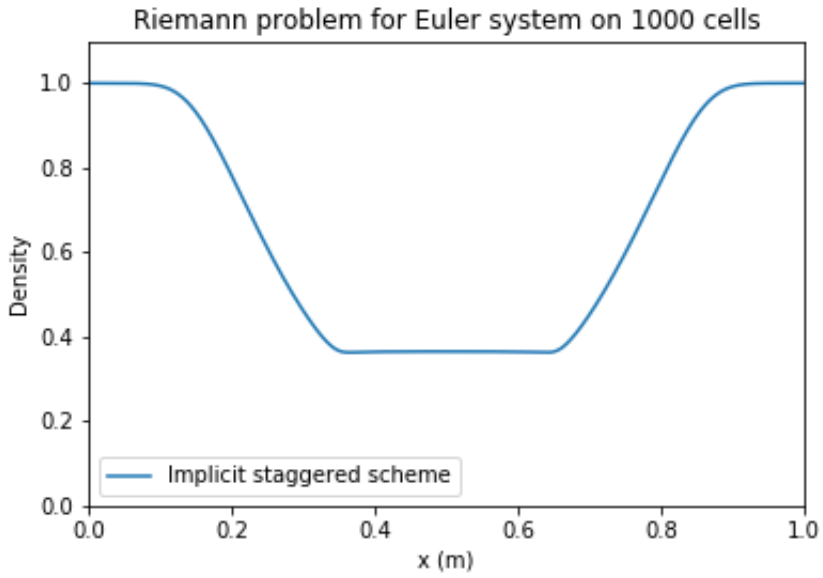


Figure 5.7: Density at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99

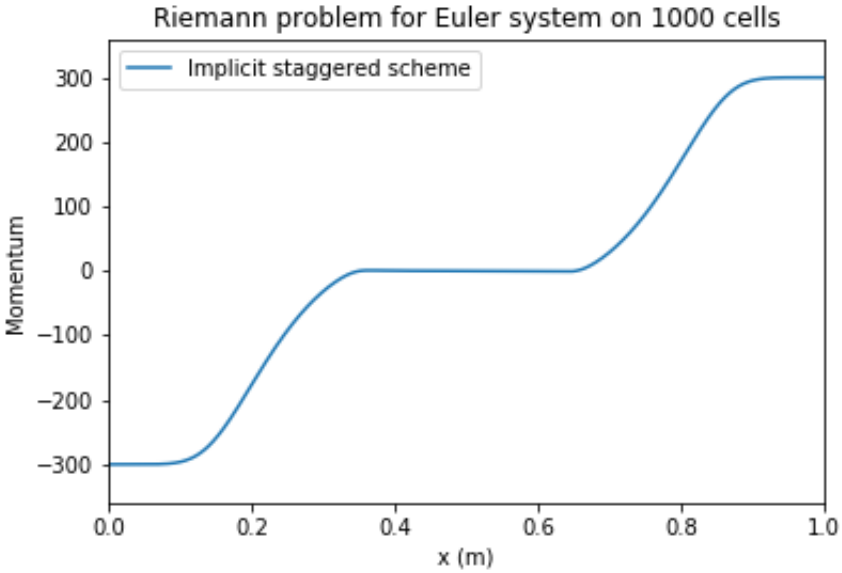


Figure 5.8: Momentum at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL$= 0.99$

### 5.3.2 The entropic staggered scheme from Corollary 5.2

In this section, we show some numerical results obtained with the entropic staggered scheme defined in Corollary 5.2. This staggered scheme has the following numerical diffusion matrix:

$$D_{Stag}(U_i, U_{i+1}) = \begin{pmatrix} a_{i,i+1} & 1 \\ -c^2 - u_{i,i+1}^2 + 2u_{i,i+1}\Delta v_{i,i+1} & a_{i,i+1} + 2u_{i,i+1} \end{pmatrix}$$

with $\Delta v_{i,i+1} = \dfrac{q_{i+1/2}}{\rho_i} - \dfrac{q_{i+3/2}}{\rho_{i+1}}$. The coefficient $a_{i,i+1}$ of this diffusion matrix is chosen such that the entropic character of this numerical method is ensured. For this numerical scheme, we have:

$$\partial_t s_i + \frac{1}{\Delta x}\left(\tilde{g}\left(\frac{q_{i+1/2}}{\rho_i}, \rho_i\right) - \tilde{g}\left(\frac{q_{i-1/2}}{\rho_{i-1}}, \rho_{i-1}\right)\right) = -\frac{1}{\Delta x} E(a_{i-1,i}, c_{i-1,i}, d_{i-1,i})$$

with:

$$\begin{aligned} E(a_{i-1,i}, c_{i-1,i}, d_{i-1,i}) &= c^2\left(q_{i-1/2}\left(\frac{\rho_i}{\rho_{i-1}} - 1 - \ln\frac{\rho_i}{\rho_{i-1}}\right) + \frac{a_{i-1,i}}{2}(\rho_{i-1} - \rho_i)\ln\frac{\rho_{i-1}}{\rho_i}\right) \\ &+ \left(\frac{q_{i+1/2}}{\rho_i} - \frac{q_{i-1/2}}{\rho_{i-1}}\right)^2\left(\frac{1}{2}q_{i-1/2} + \frac{a_{i-1,i}}{4}(\rho_{i-1} + \rho_i) + u_{i-1,i}(\rho_{i-1} - \rho_i)\right) \end{aligned} \tag{5.36}$$

The coefficient $a_{i-1,i}$ is set to ensure $E(a_{i-1,i}, c_{i-1,i}, d_{i-1,i}) \geq 0$ and hence the entropic character of this numerical method, according to the semi-discrete analysis of the entropy balance in section 5.2.

#### 5.3.2.1 Case of negative high velocities

In the sequel, we assess the behaviour of the entropic staggered scheme (5.2) on a one dimensional Riemann problem. We choose initial conditions such that the structure of the solution consists in a shock followed by a rarefaction wave to allow an easy discrimination of correct numerical

solutions. This initial condition is given by (5.34) with $u_0 = -300$. In figures 5.9 and 5.10, the numerical scheme (5.2) captures a shock wave followed by a rarefaction wave with spurious oscillations. Hence, the method captures the correct entropic solution of the Riemann problem but generates oscillations around the rarefaction wave. The entropic staggered scheme seems to have a lower numerical diffsion than the $L^2$-stable staggered scheme. The semi-discrete analysis gives a theoretical lower bound for the coefficient $a_{i-1,i}$ to ensure the entropic character of the scheme but seems not sufficient to avoid spurious oscillations.

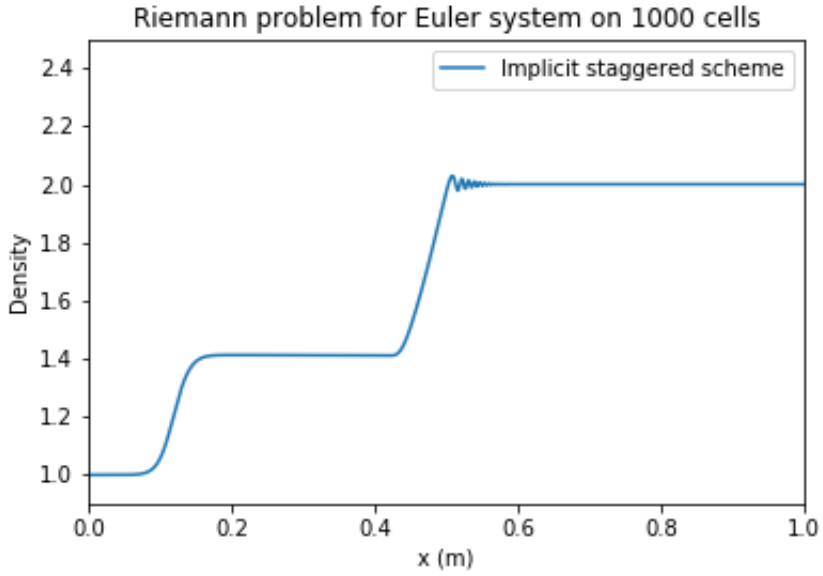


Figure 5.9: Density at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99 for $u_0 = -300$

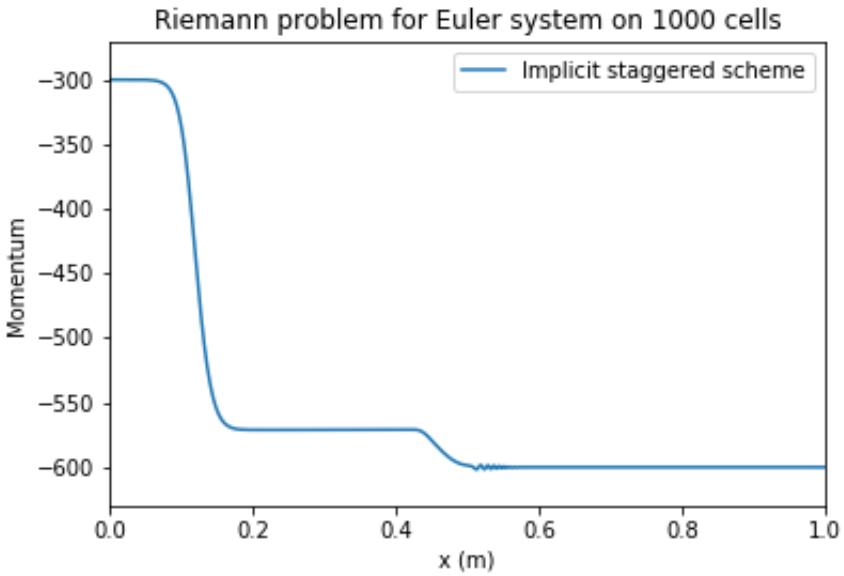


Figure 5.10: Momentum at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99 for $u_0 = -300$

#### 5.3.2.2 Case of high and low Mach number flows

In this section, we illustrate the behaviour of the entropic staggered (5.2) scheme for high and low Mach number flows. Here, we solve the Riemann problem with the initial condition (5.34) where $u_0 = 300$ (see figures 5.11 and 5.12) and $u_0 = -1$ (see figures 5.13 and 5.14). In these two cases, our conservative entropic staggered scheme (5.2) is able to capture shock and rarefaction waves without any prior information on the characteristic fields. In the case of high Mach number flow, the numerical method captures the correct entropic solution of the Riemann problem but generates oscillations around the stationnary shock wave. In the case of low Mach number flows, we do not observe spurious oscillations.

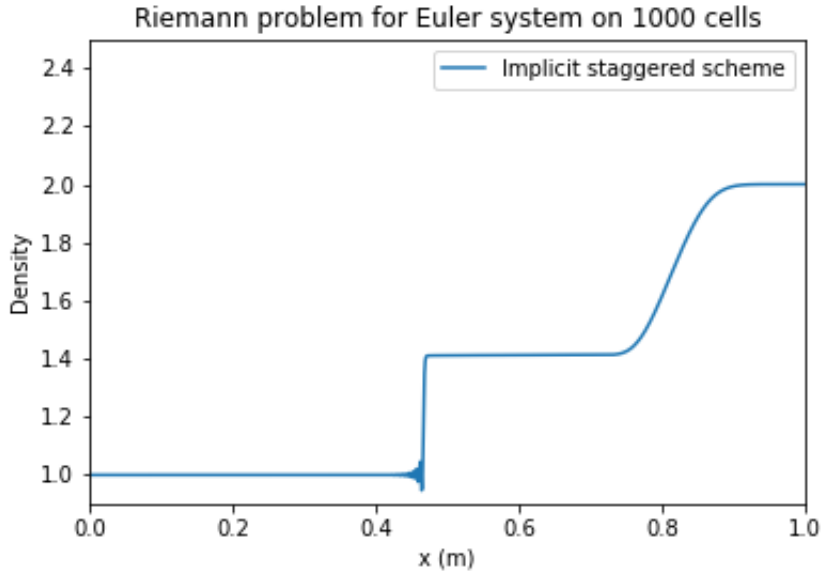


Figure 5.11: Density at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99 for $u_0 = 300$

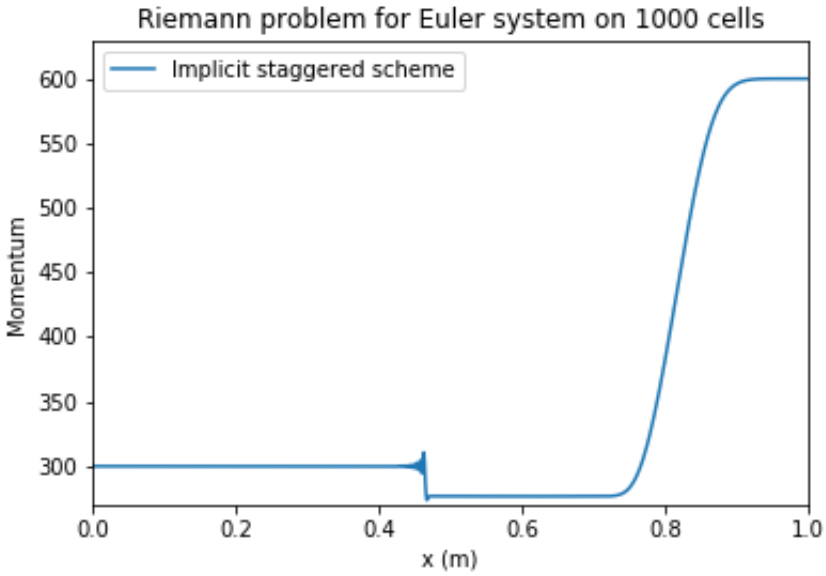


Figure 5.12: Momentum at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99 for $u_0 = 300$

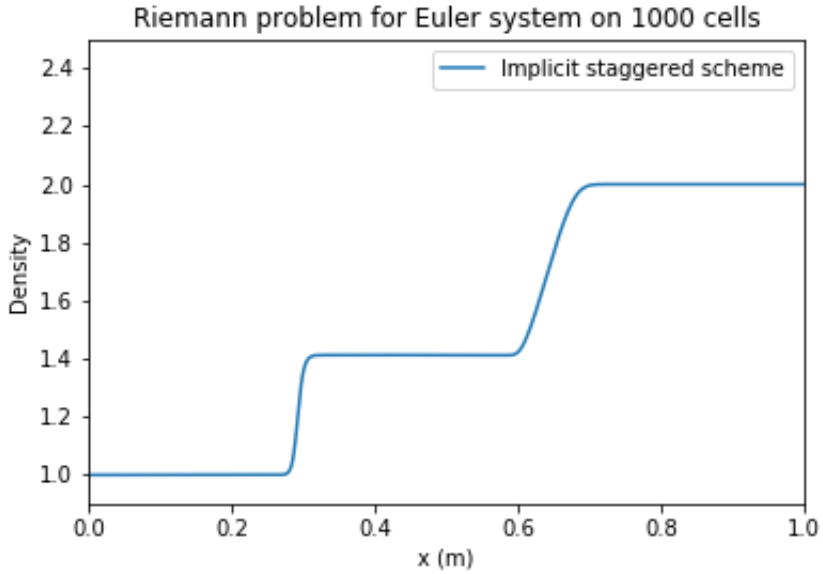


Figure 5.13: Density at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99 for $u_0 = -1$

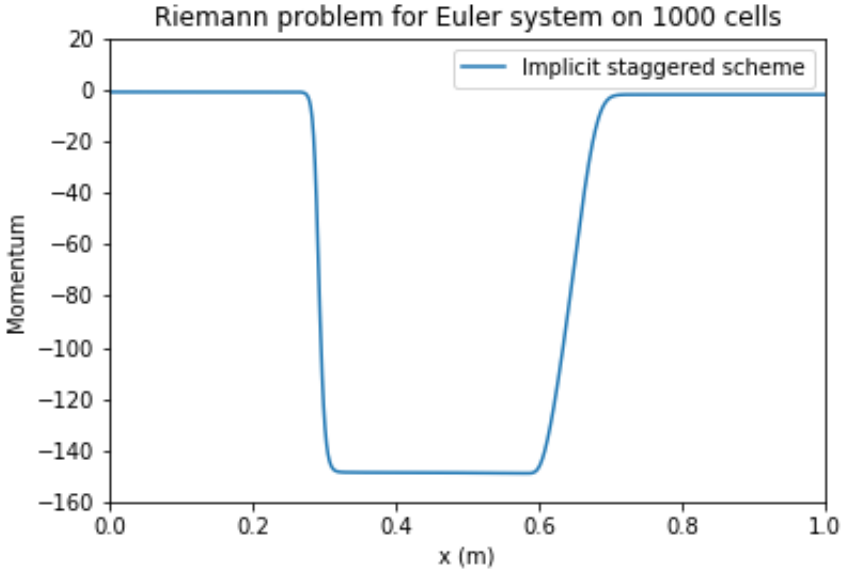


Figure 5.14: Momentum at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99 for $u_0 = -1$

#### 5.3.2.3 Case of constant densities

In this section, we consider the Riemann problem where the initial conditions are chosen such that the structure of the solution consists in two transonic rarefaction waves:

$$\text{left state: } \begin{pmatrix} \rho_{left} \\ q_{left} \end{pmatrix} = \begin{pmatrix} 1 \\ -300 \end{pmatrix}, \quad \text{right state: } \begin{pmatrix} \rho_{right} \\ q_{right} \end{pmatrix} = \begin{pmatrix} 1 \\ 300 \end{pmatrix} \tag{5.37}$$

In figures 5.15 and 5.16, we see that the entropic staggered scheme (5.2) captures the correct entropic solution of the Riemann problem with two transonic rarefaction waves. In this case, we do not observe spurious oscillations.

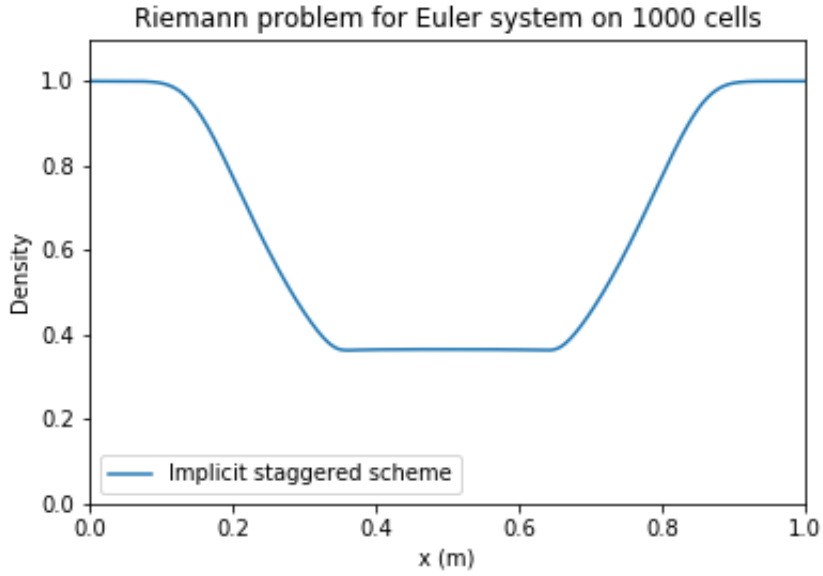


Figure 5.15: Density at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99

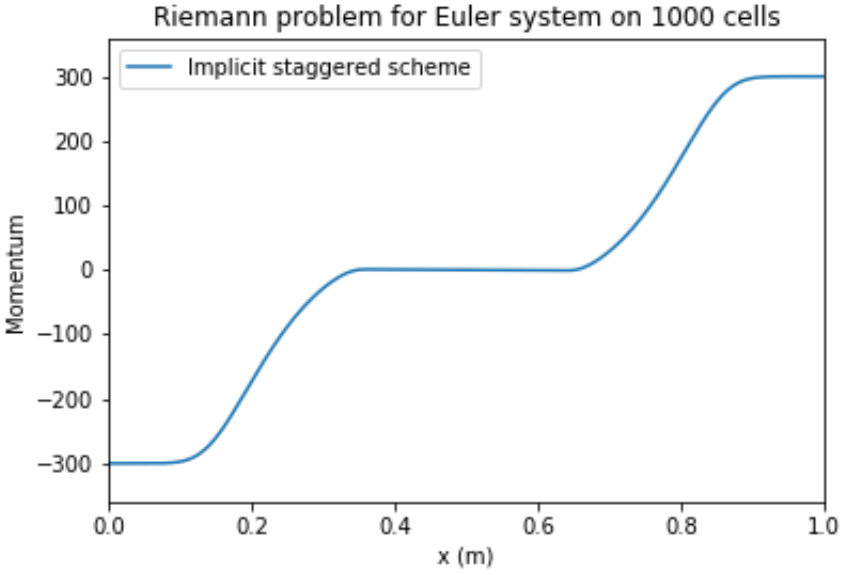


Figure 5.16: Momentum at time $t = 5 \times 10^{-4}$ with $\Delta x = 1 \times 10^{-3}$ and CFL= 0.99

## 5.4 Conclusion

In this chapter, we developed a rigorous framework for the analysis of the entropic character of finite volume schemes on staggered grids. We also presented some analytical numerical examples on the solution of the isentropic Euler equations. These numerical results illustrate the behaviour of the $L^2$-stable staggered scheme from Corollary 4.5 in configurations with a theoretical entropy violation, on the one hand. They also illustrate the behaviour of the entropic staggered scheme from Corollary (5.2) in the same configurations, on the other hand. The $L^2$-stable staggered scheme captures the correct entropic solution despite the initial entropy default. In future works, we will analyse in detail the local entropy violation in the first time steps for the $L^2$-stable scheme to have a better insight on the entropic character of this scheme. Depending on the conclusions from this first analysis, we will derive a fully discrete entropic analysis. On the other hand, the entropic staggered scheme from Corollary 5.2 captures the correct entropic solution but generates spurious oscillations. In future works, we will derive a strategy to limit the appearance of these spurious oscillations for the entropic staggered scheme. The multidimensionnal formulation of the new class of entropic schemes, introduced in this chapter, and its entropic stability analysis will be in a forthcoming paper. An extension of this strategy to the six-equation two-fluid model will be the subject of future works.

# Conclusions and perspectives

In the present work, we have first of all developed acceleration techniques in the Cathare code in order to deal with the computational complexity with reasonable computing times. These methods take advantage of modern computer architectures by the use of a time domain decomposition method. The latter has been implemented with the parareal in time algorithm. A very special stress has been put on the adaptation of this algorithm to the Cathare code in a non intrusive way allowing to use the Cathare code as a black box.

This development will be useful for several applications. First of all, the tool is important for safety calculations in the nuclear industry for the analysis of two-phase flows during accidental scenarii. A major challenge for the Cathare code is to produce real-time simulations when the software is governing a reactor simulator. A reactor simulator allows to reproduce the behaviour of a nuclear power plant under nominal or accidental conditions for the training of the operators and for the validation of the emergency procedures. Coupling the parareal algorithm with the actual acceleration techniques of the Cathare code represents a step toward a real-time response of the code.

Furthermore, since the time discretisation of the two-fluid model is done through a two-step time scheme, we designed a new variant of the parareal algorithm adapted to this family of methods. This work aims at solving the loss of accuracy in the parareal algorithm that can arise when an initialisation error is made at each time window. The parallel efficiency of this new variant is similar to the one achieved by the classical parareal algorithm and a way to improve these performances is to introduce adaptivity in the algorithm ([80]), by dynamically increasing the accuracy of the fine solver accross the parareal iterations. This adaptive formulation of the parareal algorithm offers new degrees of freedom to optimise the speed-up performances such as the choice of increasing target tolerances. A very interesting and challenging task would be the design of adaptive refinements based on a posteriori estimators. It would allow local time stepping adaptation in the parareal algorithm as well as spatial refinement if the problem involves also spatial variables. This strategy would be particularly interesting in the context of hyperbolic equations with whom the parareal algorithm may suffer from an instability. It has been shown ([102], [103], [79]) that the numerical diffusion of the coarse and the fine solvers has an impact on the appearance of the parareal instability. Hence, introducing local refinements in the algorithm may reduce the instability we may observe when the parareal algorithm is applied to hyperbolic problems.

The second main contribution of this work has been devoted to the understanding of the theoretical properties of finite volume schemes on staggered grids such as the one used in the Cathare code. The idea consists in analysing the properties of the numerical diffusion operator. After showing that the staggered schemes do not straightforwardly yield a linear stability, we derive a linearly $L^2$-stable class of staggered schemes for the isentropic Euler equations. We also

implemented a scheme of the $L^2$-stable class of schemes and perform a simulation of a Riemann problem with satisfactory results. Unlike classical staggered schemes that are $L^2$-stable for constant sign velocities, the new class is $L^2$-stable for variable sign velocities. This property is important in the context of the approximation of two-phase flows models since the phasic velocities frequently change signs. A major challenge for the simulation of two-phase flows is the configuration of the vanishing phase where one of the phases disappears in some parts of the domain. The prediction of this complex dynamic mainly relies on the capture of the void waves that appear in the two-fluid model. Since the void waves have a complex structure with a propagation speed that frequently change signs, it is important for the numerical scheme to be stable regardless of the velocity sign. Our new class of staggered schemes is a promising alternative to the actual numerical treatment of the vanishing phase implemented in the Cathare code. The actual strategy relies on an interfacial friction coefficient that becomes high when one of the phases vanishes and has reached its limits for some test cases like the one of the Water-packing, that is relevant for nuclear safety studies. A first step toward the implementation of a new staggered scheme in the Cathare code could be the application to the reduced system of [93] that focuses on the study of the void waves.

In a second time, we analyse a non linear property of the staggered schemes, namely the entropy property. We design a new class of entropic staggered schemes for the isentropic Euler equations by deriving conditions on the coefficients of the numerical diffusion operator. We implemented a scheme of this class of entropic schemes and perform with success a simulation of a Riemann problem that displays a transonic rarefaction wave with successful results. Hence, our conservative entropic staggered scheme is able to capture shock and rarefaction waves without any prior information on the characteristic fields. In the continuation of this analysis, there are several points that still need to be addressed for a better theoretical understanding of staggered schemes like the low Mach number accuracy. The multidimensionnal formulation of the new class of entropic schemes and its entropic stability analysis will be in a forthcoming work. An extension of this strategy to the six-equation two-fluid model will be the subject of future works.

# Bibliography


[1] *Etudes et approches de la gestion des accidents graves pour les réacteurs à eau sous pression du parc français.* https://www.irsn.fr/FR/Larecherche/publications-documentation/collection-ouvrages-IRSN/Documents/8_LAG_chap04.pdf.

[2] *Etudier les situations accidentelles.* http://www.cea.fr/Documents/monographies/R%C3%A9acteurs-nucl%C3%A9aires-exp%C3%A9rimentaux-situations-accidentelles.pdf.

[3] *Sofia, un simulateur pour améliorer la sûreté des réacteurs à eau sous pression.* https://www.irsn.fr/FR/base_de_connaissances/Installations_nucleaires/Les-centrales-nucleaires/Documents/IRSN_Plaquette-SOFIA_122011.pdf.

[4] R. Abgrall and S. Karni, *A comment on the computation of non-conservative products*, Journal of Computation Physics 229, pp. 2759–2763, (2010).

[5] K. Ait-Ameur and Y. Maday, *Multi-step variant of the parareal algorithm: convergence analysis and numerics*, in preparation for submission, (2020).

[6] K. Ait-Ameur, Y. Maday, and M. Tajchman, *Multi-step variant of the parareal algorithm*, to appear in: Domain Decomposition Methods in Science and Engineering XXV, Series Lecture Notes in Computational Science and Engineering, Editors: O. Widlund, X.-C. Cai, A. Klawonn, R. D. Haynes, H. H. Kim, L. Halpern, S. MacLachlan, (2020).

[7] ——, *Time-parallel algorithm for two phase flows simulation*, to appear in: Numerical Simulation in Physics and Engineering : Trends and Applications, Lecture Notes of the XVIII Jacques-Louis Lions Spanish-French School, Editors : Greiner, David ; Asensio, Maria Isabel ; Montenegro, Rafael, (2020).

[8] A. Ambroso, C. Chalons, and P. Raviart, *A Godunov-type method for the seven-equation model of compressible two-phase flow*, Computers and Fluids, 54, pp. 67-91, (2012).

[9] K. E. Amine, *Modélisation et analyse numérique des écoulements diphasiques en déséquilibre*, PhD Thesis. Université Paris 6, (1997).

[10] F. Archambeau, J.-M. Hérard, and J. Laviéville, *Comparative study of pressure correction and Godunov-type schemes on unsteady compressible cases*, Computers and Fluids, 38, pp. 1495-1509, (2009).

[11] E. Aubanel, *Scheduling of tasks in the parareal algorithm*, Parallel Computing, 37:172–182, (2011).

[12] C. Audouze, M. Massot, and S. Volz, *Symplectic multi-time step parareal algorithms applied to molecular dynamics*, (2009). http://hal.archives-ouvertes.fr/hal-00358459/fr/.

[13] L. Baffico, S. Bernard, Y. Maday, G. Turinici, and G. Zérah, *Parallel-in-time molecular-dynamics simulations*, Physical Review E, 66, pp. 057701 (2002).

[14] G. Bal, *Parallelization in time of (stochastic) ordinary differential equations*, (2003). `http://www.columbia.edu/gb2030/PAPERS/paralleltime.pdf`.

[15] G. Bal, *On the convergence and the stability of the Parareal algorithm to solve partial differential equations*, In Kornhuber, R. et et al, editors : Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering, Springer, 40, pp. 426–432 (2005).

[16] G. Bal and Y. Maday, *A "parareal" time discretization for non-linear pde's with application to the pricing of an american put*, Recent developments in domain decomposition methods, pp. 189-202, 23 (2002).

[17] A. Bermudez and M. E. Vazquez, *Upwind method for hyperbolic conservation laws with source terms*, Computers Fluids, 23.8, pp. 1049–1071, (1994).

[18] L. Berry, W. Elwasif, J. Reynolds-Barredo, D. Samaddar, R. Sanchez, and D. Newman, *Event-based parareal: A data-flow based implementation of parareal*, J. Comput. Phys., 231(17):5945 – 5954, (2012).

[19] F. Berthelin, T. Goudon, and S. Minjeaud, *Kinetic schemes on staggered grids for barotropic euler models : entropy-stability analysis*, Mathematics of Computation, Vol. 84 (295), pp. 2221-2262, (2015).

[20] D. Bestion, *The physical closure laws in the cathare code*, Nuclear Engineering and Design, 124, pp. 229-245 (1990).

[21] F. Bouchut, *Nonlinear stability of finite volume methods for hyperbolic conservation laws and well-balances scheme for sources*, Frontiers in Mathematics, 2000.

[22] ——, *Nonlinear stability of finite volume methods for hyperbolic conservation laws and well-balances scheme for sources*, Frontiers in Mathematics, 2000.

[23] A. Bressan, *Unique solutions for a class of discontinuous differential equations*, Proceedings of the AMS, 104, pp. 1753-1783, (1988).

[24] A. Bressan, *Hyperbolic Systems of Conservation Laws : The One-dimentional Cauchy Problem*, Oxford Lecture Series in Mathematics et Its Applications, 2000.

[25] M. Chandesris, *3d module of cathare : improvement of the momentum convective term discretization*, Technical Report DEN/CAD/DER/SSTH/LDLD/EM/NT/2010-035/A, CEA, (2010).

[26] ——, *Discrétisation des équations du module 3d de cathare 2 & comparaison avec le module 1d – etat des lieux pour la version v2.5_2 et perspectives*, Note Technique DEN/CAD/DER/SSTH/LDLD/NT/2011-046/A, (2011).

[27] P. Chartier and B. Philippe, *A parallel shooting technique for solving dissipative ODE's*, Computing, 51(3-4):209–236, (1993).

[28] F. Chen, J. S. Hesthaven, Y. Maday, and A. S. Nielsen, *An adjoint approach for stabilizing the Parareal method*, Rapport technique, EPFL-ARTICLE-211097, (2015).

[29] F. Chen, J. S. Hesthaven, and X. Zhu, *On the use of reduced basis methods to accelerate and stabilize the Parareal method*, In Reduced Order Methods for Modeling and Computational Reduction, Springer, 25, pp. 187–214 (2014).

[30] A. Christlieb, C. B. Macdonald, and B. W. Ong, *Parallel high-order integrators*, SIAM J. Sci. Comput., 32(2):818–835, (2010).

[31] F. Coquel, K. E. Amine, E. Godlewski, B. Perthame, and P. Rascle†, *A numerical method using upwind schemes for the resolution of two-phase flows*, Journal Computational Physics, 136, pp. 272-288, (1997).

[32] F. Cordier, P. Degond, and A. Kumbaro, *Phase Appearance or Disappearance in Two-Phase Flows*, J. Sci. Compu., 58, (2014).

[33] C. Dafermos, *Hyperbolic conservation laws in continuum physics*, T. 325. Grundlehren der mathematischen Wissenschaften, 2010.

[34] X. Dai, C. L. Bris, F. Legoll, and Y. Maday, *Symmetric parareal algorithms for hamiltonian systems*, ESAIM: Mathematical Modelling and Numerical Analysis, 47, pp. 717–742 (2013).

[35] X. Dai and Y. Maday, *Stable Parareal in time method for first-and second-order hyperbolic systems*, SIAM J. Sci. Comput., pp. A52–A78, 35(1) (2013).

[36] S. Dellacherie, *Analysis of Godunov type schemes applied to the compressible Euler system at low Mach number*, Journal of Computational Physics, 229(4), pp. 978-1016, (2010).

[37] S. Dellacherie, P. Omnes, and F. Rieper, *Analysis of godunov type schemes applied to the compressible euler system at low mach number*, J. Comp. Phys., 229(14), pp. 5315-5338, (2010).

[38] B. Després and F. Dubois, *Systèmes hyperboliques de lois de conservation*, Les Éditions de l'École Polytechnique, 2005.

[39] D. Drew and S. Passman, *Theory of multicomponent fluids*, Springer-Verlag, New-York, 1999.

[40] M. Emmet and M. Minion, *Toward an efficient parallel in time method for partial differential equations*, Comm. App. Math. and Comp. Sci., 1(1), (2012).

[41] J. Erhel and S. Raoult, *Algorithme parallèle pour le calcul d'orbites - Parallélisation à travers le temps*, Techniques et Sciences informatiques, 19(5), (2000).

[42] S. Evje and T. Flatten, *Hybrid flux-splitting schemes for a common two-fluid model*, Journal of Computational Physics, 192, pp. 175–210, (2003).

[43] R. Eymard, R. Herbin, J.-C. Latché, and B. Piar, *A class of collocated finite volume schemes for incompressible flow problems*, Proceedings of ALGORITMY, pp. 31-40, (2009).

[44] R. Falgout, S. Friedhoff, T. Kolev, S. MacLachlan, and J. Schroder, *Parallel time integration with multigrid*, SIAM J. Sci. Comput., 36(6):C635–C661, (2014).

[45] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, J. B. Schroder, and S. Vandewalle, *Multigrid methods with space-time concurrency*, Computing and Visualization in Science, 18, pp. 123-143 (2017). LLNL-JRNL-678572, `http://dx.doi.org/10.1007/s00791-017-0283-9`.

[46] R. D. Falgout, M. Lecouvez, and C. S. Woodward, *A parallel-in-time algorithm for variable step multistep methods*, (2017). Submitted, LLNL-JRNL-739759, `https://computing.llnl.gov/projects/parallel-time-integration-multigrid/2017_BDF_Paper_v1.pdf`.

[47] C. Farhat and M. Chandesris, *Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid–structure applications*, International Journal for Numerical Methods in Engineering, 58, pp. 1397–1434 (2003).

[48] C. Farhat, J. Cortial, and H. Dastillung, C.and Bavestrello, *Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses*, Internat. J. Numer. Methods Engrg., 67(5), pp. 697–724, (2006).

[49] P. Fischer, F. Hecht, and Y. Maday, *A parareal in time semi-implicit approximation of the Navier-Stokes equations*, In Kornhuber, R. and al, editors : Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering, 40, pp. 433-440 (2004).

[50] M. Gander, *50 years of time parallel time integration*, In Carraro, T., Geiger, M., Körkel, S. and Rannacher, R., editors : Multiple Shooting and Time Domain Decomposition Methods, 40, pp. 69-114 (2015).

[51] M. Gander and S. Guttel, *Paraexp : A parallel integrator for linear initial-value problems*, SIAM J. Sci. Comput., 35(2):C123–C142, (2013).

[52] M. J. Gander, *Analysis of the Parareal algorithm applied to hyperbolic problems using characteristics*, Bol. Soc. Esp. Mat. Apl., 42, pp. 21–35, (2008).

[53] M. J. Gander and E. Hairer, *Nonlinear convergence analysis for the parareal algorithm*, in Domain Decomposition Methods in Science and Engineering XVII, Springer, pp. 45–56, (2008).

[54] M. J. Gander and M. Petcu, *Analysis of a Krylov subspace enhanced Parareal algorithm for linear problems*, In ESAIM : Proceedings, EDP Sciences, pp. 114–129, 25 (2008).

[55] M. J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, SIAM J. Sci. Comput., 29(2), pp. 556-578 (2007).

[56] J. Ghidaglia, A. Kumbaro, and G. L. Coq, *Une méthode volumes finis à flux caractéristiques pour la résolution numérique des systèmes hyperboliques de lois de conservation*, Comptes Rendus de l'Acad. Sciences Paris, Série 1, vol 322, pp. 981 988, (1996).

[57] J.-M. Ghidaglia, A. Kumbaro, and G. L. Coq, *On the numerical solution to two fluid models via a cell centered finite volume method*, Eur. J. Mech. B-Fluids, (2001).

[58] E. Godlewski and P.-A. Raviart, *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, T. 118. Appllied Mathematical Sciences - Springer, 1996.

[59] E. Godlewski and P. A. Raviart, *Numerical approximation of hyperbolic systems of conservation laws*, Applied Mathematical Sciences, vol. 118., Springer-Verlag, New York, 1996.

[60] S. K. Godunov, *A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations*, Mat. Sbornik. 47: 271–306, (1959).

[61] R. Guetat, *Méthode de parallélisation en temps: Application aux méthodes de décomposition de domaine*, PhD thesis, Paris VI, (2012).

[62] F. Harlow and A. Amsden, *Numerical calculation of almost incompressible flow*, Journal of Computational Physics, 3:80–93, (1968).

[63] ——, *A numerical fluid dynamics calculation method for all flow speeds*, Journal of Computational Physics, 8:197–213, (1971).

[64] F. Harlow and J. Welsh, *Numerical calculation of time-dependent viscous incompressible flow with free surface*, Physics of Fluids, vol 8, pp 2182-2189, (1965).

[65] R. Herbin, W. Kheriji, and J.-C. Latché, *Staggered schemes for all speed flows*, ESAIM: Proceedings, EDP Sciences, Congrès National de Mathématiques Appliquées et Industrielles, 35, pp.122-150, (2011).

[66] ——, *On some implicit and semi-implicit staggered schemes for the shallow water and euler equations*, ESAIM: Mathematical Modelling and Numerical Analysis, EDP Sciences, 48 (6), pp.1807-1857, (2014).

[67] R. Herbin and J.-C. Latché, *A kinetic energy preserving convection operator for the mac discretization of compressible navier-stokes equations*, Mathematical Modelling and Numerical Analysis, (2010). `https://hal.archives-ouvertes.fr/hal-00477079/document`.

[68] G. Hewitt, J. Delhaye, and N. Zuber, *Multiphase Science and Technology*, chapter V.H. Ransom : Numerical Benchmark 2.3, Hemisphere Publishing Corporation and Springer-Verlag, 3, pp. 471-473 (1987).

[69] G. Hewitt, J. Delhaye, and N. Zuber, *Multiphase science and technology*, vol. 6, 1991.

[70] C. W. Hirt, *Heuristic stability theory for finite difference equations*, J. Comp. Phys., 2, pp. 339-355, (1968).

[71] M. Ishii, *Thermo-fluid dynamic theory of two-phase flow*, Paris : Eyrolles, (1975).

[72] M. Ishii and T. Hibiki, *Thermo-Fluid Dynamics of Two-Phase Flow*, Springer, (2011).

[73] O. A. Krzysik, H. D. Sterck, S. P. MacLachlan, and S. Friedhoff, *On selecting coarse-grid operators for Parareal and MGRIT applied to linear advection*, arXiv:1902.07757, (2019).

[74] G. Lavialle, *Cathare 2 v2.5_3mod3.1 code: General description (partners version)*, Note Technique DEN/DANS/DM2S/STMF/LMES/NT/13-018/A, (2013).

[75] P. LeFloch, *Hyperbolic Systems of Conservation Laws, the Theory of Classical and Non-classical Shock Waves*, Birkhauser, 2002.

[76] R. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser, 1992.

[77] ——, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, 2004.

[78] J.-L. Lions, Y. Maday, and G. Turinici, *Résolution par un schéma en temps "pararéel"*, C. R. Acad. Sci. Paris, 332(7), pp. 661-668 (2001).

[79] T. Lunet, J. Bodart, S. Gratton, and X. Vasseur, *Time-parallel simulation of the decay of homogeneous turbulence using Parareal with spatial coarsening*, Computing and Visualization in Science, 19, no. 1, pp. 31–44 (2018).

[80] Y. Maday and O. Mula, *An adaptive parareal algorithm*, Journal of Computational and Applied Mathematics, (2020). `https://arxiv.org/pdf/1909.08333.pdf`.

[81] Y. Maday, J. Salomon, and G. Turinici, *Monotonic time-discretized schemes in quantum control*, Numerische Mathematik, 103(2):323–338, (2006).

[82] Y. Maday and G. Turinici, *The Parareal in Time Iterative Solver: a Further Direction to Parallel Implementation*, In Domain Decomposition Methods in Science and Engineering, pp. 441–448. Springer Berlin Heidelberg, (2005).

[83] J. M. Masella, I. Faille, and T. Gallouët, *On an approximate godunov scheme*, Intl. J. Computational Fluid Dynamics,, Vol. 12, pp. 133-149, (1999).

[84] G. D. Maso, P. LeFloch, and F. Murat, *Definition and weak stability on non conservative products*, Journal of Math Pures Application 74, pp. 483–548, (1995).

[85] M. Minion, *A hybrid parareal spectral deferred corrections method*, Comm. App. Math. and Comp. Sci., 5(2), (2010).

[86] M. Mohaupt, *Étude du problème numérique de Water-Packing*, Master's Thesis. CEA-Grenoble/DEN/DER/SSTH/LMDL, (2008).

[87] A. Morin, T. Fltten, and S. T. Munkejord, *A roe scheme for a compressible six-equation two-fluid model*, Int. J. Numer. Meth. Fluids, pp. 1–28, (2011).

[88] S. Munkejord, S. Evje, and T. Fltten, *A musta scheme for a nonconservative two-fluid model*, SIAM J. Sci. Comput. 31, pp. 2587–2622, (2011).

[89] M. Ndjinga, *Influence of interfacial pressure on the hyperbolicity of the two-fluid model*, C. R. Acad. Sci. Paris, Ser. I 344, pp. 407-412 (2007).

[90] M. Ndjinga, *Influence of interfacial pressure on the hyperbolicity of the two-fluid model*, Comptes Rendus Mathematique, 344, pp. 407–412, (2007).

[91] ——, *Quelques aspects de modélisation et d'analyse des systèmes issus des écou- lements diphasiques*, Ecole Centrale de Paris, (2007).

[92] M. Ndjinga and K. Ait-Ameur, *A new class of $L^2$ -stable schemes for the isentropic Euler equations on staggered grids*, to appear in: Finite Volumes for Complex Applications IX , Editors: Robert Klofkorn, Erik Keilegavlen, Adrian Florin Radu, Jurgen Fuhrmann, (2020).

[93] M. Ndjinga, T. P. K. Nguyen, and C. Chalons, *A $2 \times 2$ hyperbolic system modelling incompressible two phase flows: theory and numerics*, Nonlinear Differential Equations and Applications, n. 36, vol. 24, issue 4, (2017).

[94] T. P. K. Nguyen, *On the mathematical analysis and the numerical simulation of boiling flow models in nuclear power plants thermal hydraulics*, Université Paris-Saclay, Université de Versailles Saint-Quentin-en-Yvelines, (2016).

[95] A. S. Nielsen, G. Brunner, and J. S. Hesthaven, *Communication-aware adaptive parareal with application to a nonlinear hyperbolic system of partial differential equations*, Journal of Computational Physics, 371, pp. 483–505, (2018).

[96] J. Nievergelt, *Parallel methods for integrating ordinary differential equations*, Commun. ACM, 7, pp. 731–733 (1964).

[97] H. Ninokata and T. Okano, *Sabena: Subassembly boiling evolution numerical analysis*, Nuclear Engineering and Design 120, pp 349 - 367, (1990).

[98] A. Prosperetti and G. Tryggvason, *Computational methods for multiphase flow*, Cambridge University press, 2009.

[99] A. Quarteroni and A. Valli, *Domain decomposition methods for partial differential equations*, Von Karman institute for fluid dynamics, 1996.

[100] P. L. Roe, *Approximate riemann solvers, parameter vectors and difference schemes*, J. of Comput. Phys., 43, 357, (1981).

[101] A. Ruby, O. Antoni, V. Créach, P. Dufeil, C. Rose, and F. Iffenecker, *Quest for the real-time for the safety analysis code cathare 2 used in the post-accident simulator sipa*, Technical Report, CEA, EDF, IRSN, (2003).

[102] D. Ruprecht, *Convergence of Parareal with spatial coarsening*, PAMM, 14(1), pp. 1031–1034, (2014).

[103] D. Ruprecht, *Wave propagation characteristics of Parareal*, Computing and Visualization in Science, 19, no. 1, pp. 1–17 (2018).

[104] D. Samaddar, D. E. Newman, and R. Sanchez, *Parallelization in time of numerical simulations of fully-developed plasma turbulence using the parareal algorithm*, Journal of Computational Physics, 229(18), pp. 6558-6573, (2010).

[105] G. Samaey and T. Slawig, *A micro/macro parallel-in-time (parareal) algorithm applied to a climate model with discontinuous non-monotone coefficients and oscillatory forcing.* `https://arxiv.org/abs/1806.04442`, 2018.

[106] D. Serre, *System of Conservation Laws 1 : Hyperbolicity, Entropies, Shock Waves*, Cambridge University Press, 1999.

[107] Y. Shekari and E. Hajidavalloo, *Application of Osher and PRICE-C schemes to solve compressible isothermal two-fluid models of two-phase flow*, Computers and Fluids, 86, pp. 363-379, (2013).

[108] G. Staff and E. Ronquist, *Stability of the parareal algorithm*, Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering, 40, pp. 449-456 (2005).

[109] H. D. Sterck, R. D. Falgout, A. J. M. Howse, S. P. MacLachlan, and J. B. Schroder, *Parallel-in-time multigrid with adaptive spatial coarsening for the linear advection and inviscid Burgers equations*, SIAM Journal on Scientific Computing, 41(1), pp. A538–A565, (2019).

[110] H. D. Sterck, S. Friedhoff, A. J. M. Howse, and S. P. MacLachlan, *Convergence analysis for parallel-in-time solution of hyperbolic systems*, arXiv:1903.08928, (2019).

[111] J. H. Stuhmiller, *The Influence of Interfacial Pressure Forces on the Character of Two-Phase Flow Model Equations*, Int. J. Multiphase Flow, 3, (1977).

[112] A. Toselli and O. Widlund, *Domain decomposition methods: algorithms and theory*, vol. vol. 3, Springer, 2005.

[113] I. Toumi and A. Kumbaro, *An Approximate Linearized Riemann Solver for a Two-Fluid Model*, Journal of Computational Physics, 124, pp. 286–300, (1996).

[114] I. Toumi, A. Kumbaro, and H. Paillere, *Approximate Riemann solvers and flux vector splitting schemes for two-phase flow*, 30th Computational Fluid Dynamics, (1999).

[115] A. Vernier, *Validation de Modèles Multichamps dans le Logiciel OVAP*, Rapp. tech. École Nationale Supérieure des Mines de Paris, (2006).

[116] W. Wagner, J. R. Cooper, A. Dittmann, J. Kijima, H.-J. Kretzschmar, A. Kruse, R. Mares, K. Oguchi, H. Sato, I. Stocker, O. Sifner, Y. Takaishi, I. Tanishita, J. Trubenbach, and T. Willkommen, *The IAPWS industrial formulation 1997 for the thermodynamic properties of water and steam*, Journal of Engineering for Gas Turbines and Power, pp. 31-40, (2000).