Contribution to numerical methods for all Mach flow regimes and to fluid-porous coupling for the simulation of homogeneous two-phase flows in nuclear reactors

Chady Zaza

École Doctorale 184 – Mathématiques et Informatique

Doctorat Aix-Marseille Université

THÈSE

pour obtenir le grade de docteur délivré par

Aix-Marseille Université

Discipline doctorale "Mathématiques"

présentée et soutenue publiquement par

Chady ZAZA

le 2 février 2015

Contribution à la résolution numérique d'écoulements à tout nombre de Mach et au couplage fluide-poreux en vue de la simulation d'écoulements diphasiques homogénéisés dans les composants nucléaires.

Jury

Mme. Ann ALMGREN,	Chercheur, Lawrence Berkeley National Laboratory	*Rapporteur*
M. Benoît GOYEAU,	Professeur, Ecole Centrale Paris	*Rapporteur*
M. Cédric GALUSINSKI,	Professeur, Université du Sud Toulon Var	*Examinateur*
M. Thierry GOUDON,	Directeur de Recherche, INRIA Sophia Antipolis	*Examinateur*
M. Jean-Marc HÉRARD,	Ingénieur Sénior, EDF R&D	*Examinateur*
Mme. Raphaèle HERBIN,	Professeur, Aix-Marseille Université	*Directeur*
M. Philippe ANGOT,	Professeur, Aix-Marseille Université	*Codirecteur*
M. Michel BELLIARD,	Ingénieur, CEA Cadarache	*Encadrant*

في ذكرى حرم أمينة [أمين مصطفى] ظاظا، اسم قبل الزواج خريبه

In the memory of Amina [Amin Mustapha] Zaza, née Kheraiba

Our doubts are traitors,
And make us loose the good we oft might win
By fearing to attempt.

William Shakespeare, Measure for Measure; I, 4

TABLE OF CONTENTS

INTRODUCTION

Industrial problem

The most widely built Generation II nuclear reactors are the *Pressurized Water Reactors* (PWR). Figure 1 gives a simplified overview of the interaction between the different components of a PWR. Water circulates inside the two independent loops with the help of two pumps. The water of the *primary loop* (contaminated), in liquid state, is heated by the *core's fuel rods*. This heat is transferred to the *secondary loop* (uncontaminated) using a *steam generator*. The latter generates steam at high pressure which later expands in a *turbine*, thereby producing electricity injected into the power grid. The high pressure steam comes back to liquid state after passing through a *condenser*. The nominal electric power generated by a PWR is about 1 GWe.

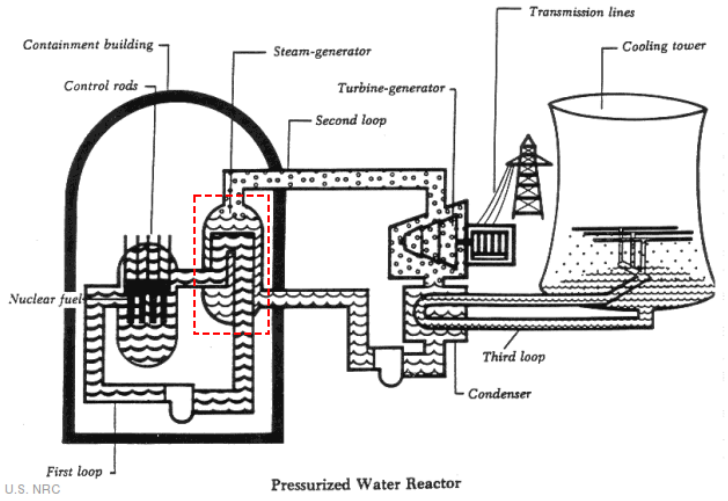


FIG. 1 – *Diagram of a Pressurized Water Reactor (courtesy of U.S. NRC). The steam generator is framed in red.*

Within the LMEC (Component Scale Modelling Laboratory) at CEA and more broadly at the STMF (Thermohydraulics and Fluid Mechanics Section) there is particular focus on steam generators, either at the component scale (eg. design of steam generators with the GENEPI code) or at the system scale (eg. nuclear safety analysis with the CATHARE code). Therefore we are interested in global energy balances of steam generators at the component scale rather than in the complex structure of the flow at the smallest scales therein.

An example of steam generator is depicted in figure 2. The main elements of a steam generator are the *evaporator* and the *dryers*. The evaporator is constituted of about 3000 *U*-

tubes made of copper. This array of tubes is held by horizontal support plates and it features anti-vibration bars at its outflow. The liquid water of the primary loop, at high temperature, flows inside these tubes. Liquid water from the secondary loop is injected at the top of the evaporator reaches the U-tubes at the bottom, perpendicularly to them. As the flow goes up, a bubbly flow develops and at the top of the evaporator steam is generated, high pressure and temperature. The dryers then further improve the vapor quality.

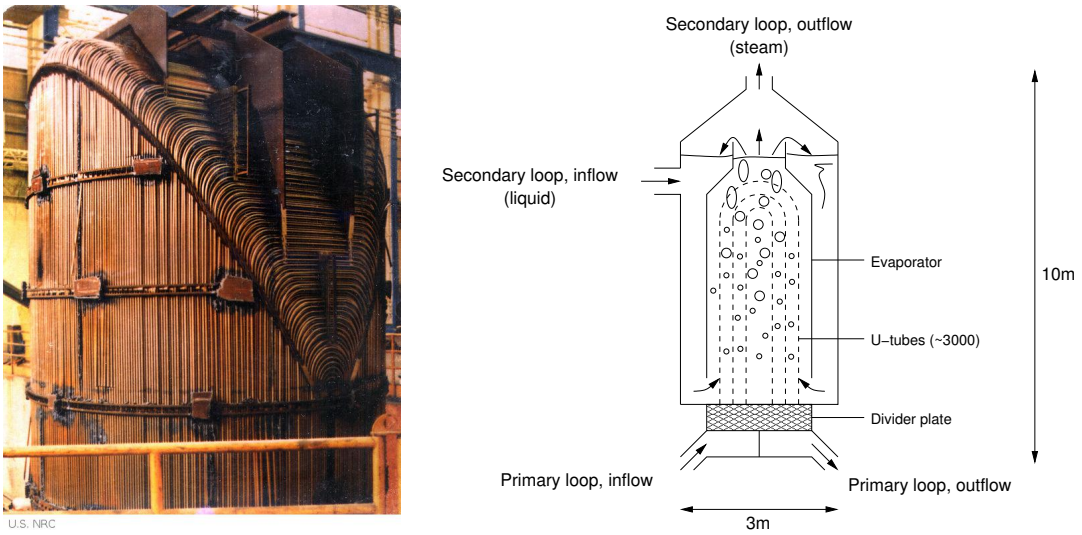


FIG. 2 – *Left: the upper part of a steam generator (courtesy of U.S. NRC). Right: simplified representation of a steam generator.*

The geometry of a steam generator is thus very complex, and it is not affordable nor of interest for our industrial applications to compute the flow at the typical scale separating the U-tubes. In nuclear codes, in particular in the GENEPI code, the array of tubes is represented by a *porous medium*. This equivalent porous medium has two interfaces with the plain fluid regions: at the bottom where the liquid water of the secondary loop attacks the tubes perpendicularly and then on the upper part (also called "chignon") where a bubbly flow leaves the array of U-tubes.

Layout of the thesis

Our aim is to contribute some efficient methods for the numerical simulation of pressurized water reactors. We hope to do so by developing some efficient all Mach solvers with adaptive mesh refinement, and by studying the modelling of the free fluid– porous interface. This thesis thus consists in three parts, namely the development of all-Mach solver, followed by a construction of of an adaptive mesh refinement procedure to implement it efficiently, and finally a study of the fluid-porous interface problem. Let us now give a short description of each part.

All-Mach solver

The simulation of compressible two-phase flows in PWR is a very complex problem, involving different flow regimes and several space scales. When dealing with accident scenarios such as SBLOCA (Small Break Loss Of Coolant Accident) flow variables can exhibit very fast and important variations and supersonic regions may appear. On the other hand, at the nominal regime, the flow is in the low-Mach regime and is almost at steady state.

It is therefore of great interest to develop numerical methods that would be able to handle any of these regimes, i.e. at all Mach numbers. Such a class of methods was recently introduced for staggered finite volumes [87, 76]. Following the original ideas of [76], we introduce in this first part a new pressure-correction scheme for cell-centered finite volumes for solving the compressible Navier-Stokes and Euler equations at all Mach number. This numerical scheme is proved to be well posed and to converge to weak form of the compressible Euler equations. The pressure-correction scheme handles naturally low-Mach flows, and this is verified numerically. More importantly, the scheme is tested on highly compressible flows with 1D and 2D Riemann problems. The cell-centered discretization allows more easily the formulation of two-phase flow models, which is the purpose of the last chapter of this part. Our pressure-correction scheme is extended to the compressible two-phase flow models of the GENEPI code, and validated on a dedicated benchmark.

Adaptive Mesh Refinement

Adaptive resolution techniques are of major interest in the industry, as they allow the simulation of large scale problem with an optimal resolution in relevant regions. As a result, numerical simulations can be performed with an accuracy beyond the limits which would be imposed by available computing resources without adaptive refinement.

In this second part, we present an extension of our pressure-correction scheme to handle adaptive grids. More precisely, block structured adaptive mesh refinement (SAMR) will be considered. The first part of this work is directed towards algorithms for adaptive mesh generation. A clustering algorithm is tested and improved for the needs of our numerical methods. The modified algorithms are assessed on several standard tests. Then the issue of the discretization and of the resolution of semi-implicit schemes on adaptive grids is addressed. The classical techniques for managing partitioned level grid are interpreted using domain decomposition concepts. Finally a multigrid-AMR solver is presented for solving our pressure correction scheme on adaptive grids. The numerical method is implemented in a numerical code "MNFD" developed from scratch for the purpose of this work. The validation of the subsequent all-Mach adaptive pressure-correction scheme is tested on a 2D Riemann problem and on a Mach reflection problem. An extensive analysis of the numerical results is provided in order to ultimately have an insight on the suitability of our the adaptive scheme for complex flow regimes in nuclear reactors.

Fluid-porous interface problem

As previously explained, porous models play an important role in the simulation of steam generators. A particular problem of interest is the modelling at the interface between the porous medium representing the evaporator and the free-fluid at the inflow and outflow.

In a bid to improve existing fluid-porous modelling in GENEPI, we focus on the classical issue of fluid-porous interface model at the macroscopic scale. This is a two fold problem. On

the one hand, interface conditions have to be relevant with respect to the physical regimes under consideration. On the other hand, these transmission conditions must be compatible with the partial differential equation governing the free-fluid flow and the flow in the porous medium, so as to the well-posedness of the coupled problem. In this part, we address the modelling side of the problem for highly convective flow governed by the incompressible Navier-Stokes equations. An interface condition derived from a kinetic energy balance is proposed and verified with direct simulations on about a hundred of flow configurations.

Part I

All-Mach flow solver

Chapter 1

A CELL-CENTERED NAVIER-STOKES SOLVER FOR ALL-MACH FLOWS

1.1 Introduction

In this work we address the issue of solving the compressible Navier-Stokes equations:

$$\partial_t \rho + \operatorname{div}(\rho \boldsymbol{u}) = 0 \tag{1.1.1a}$$

$$\partial_t(\rho \boldsymbol{u}) + \mathbf{div}(\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p - \mathbf{div}(\boldsymbol{\tau}(\boldsymbol{u})) = 0 \tag{1.1.1b}$$

$$\partial_t(\rho E) + \operatorname{div}(\rho E \boldsymbol{u}) + \operatorname{div}(p\boldsymbol{u}) - \operatorname{div}(\boldsymbol{\tau}(\boldsymbol{u}) \cdot \boldsymbol{u}) = 0 \tag{1.1.1c}$$

$$\rho \geq 0, e \geq 0, p = (\gamma - 1)\rho e, \quad E = \frac{1}{2}|\boldsymbol{u}|^2 + e, \tag{1.1.1d}$$

$$\boldsymbol{\tau}(\boldsymbol{u}) = \mu(\nabla \boldsymbol{u} + \nabla^t \boldsymbol{u}) - \frac{2}{3}\mu \operatorname{div} \boldsymbol{u}\, I \tag{1.1.1e}$$

where $\boldsymbol{u}$, ρ, p, E, e, μ and γ denote the velocity, the density, the pressure, the total energy, the internal energy, the dynamic viscosity and the heat capacity ratio, with suitable boundary and initial conditions on p, ρ and e. In the case of $\mu = 0$ the system reduces to the compressible Euler equations, which is the main concern of this work.

The problem is defined over $\Omega \times (0, T)$, where Ω is an open bounded connected subset of $\mathbb{R}^d$, $d = 1, 2$ or 3 and $(0, T)$ is a finite time interval. The system is complemented by initial conditions for ρ, e and $\boldsymbol{u}$, which are denoted by ρ_0, e_0 and $\boldsymbol{u}_0$, with $\rho_0 > 0$ and $e_0 > 0$. For simplicity, in the exposition and in the study of the numerical scheme we shall consider the boundary condition $\boldsymbol{u} \cdot \boldsymbol{n} = 0$, where $\boldsymbol{n}$ stands for the normal vector to the boundary, but other conditions are also implemented (see section 2.2).

Defining a robust scheme for the numerical solution of the compressible Euler equations at all Mach regimes is a challenging issue. Indeed in the zero Mach limit the pressure gradient has a singular limit and the acoustic time scale vanishes [4]. As a result, approximate Riemann solvers face severe limitations, among which the loss of accuracy of the pressure gradient approximation and the time step limitation [42, 128, 73]. Pressure-correction methods may be relevant for addressing this issue, in particular because of their built-in stability properties.

Pressure-correction schemes were originally introduced in the late 60's by Chorin [43] and Temam [123, 124] for the incompressible Navier-Stokes equations. One of the first attempts to extend projection methods to compressible flows also dates back to 1968 with the *ICE* method [74] of Harlow and Amsden. The implicitation of the incompressible terms avoids a CFL time step restriction with respect to the speed of sound but because the method is not conservative, shock speeds are miscalculated. Further approaches, this time limited to low-Mach numbers, include the method of Colella and Pao [47] with a Hodge decomposition on all

flow variables and the *SIMPLE* algorithm of Karki and Patankar [85] with a projection step on the mass balance. More recently Degond and Tang [48] introduced a scheme for solving the isentropic Euler equations based on a splitting of the pressure gradient into an explicit part and an implicit part. Their scheme is proved to be asymptotically preserving and numerical experiments show correct shock speeds.

Having also in mind the extension of incompressible projection schemes to high Mach regimes, we introduce an original pressure-correction scheme for cell-centered finite-volumes, extending the ideas of [71, 77] developed for staggered finite volumes. We choose to formulate the compressible Euler equations with the internal energy rather than with the total energy:

$$\partial_t \rho + \operatorname{div}(\rho \boldsymbol{u}) = 0 \qquad \text{in } \Omega \times [0,T] \tag{1.1.2a}$$

$$\partial_t (\rho \boldsymbol{u}) + \mathbf{div}(\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p = 0 \qquad \text{in } \Omega \times [0,T] \tag{1.1.2b}$$

$$\partial_t (\rho e) + \operatorname{div}(\rho e \boldsymbol{u}) + p \operatorname{div}(\boldsymbol{u}) = 0 \qquad \text{in } \Omega \times [0,T] \tag{1.1.2c}$$

$$\rho \geq 0, e \geq 0, p = (\gamma - 1)\rho e \tag{1.1.2d}$$

with suitable boundary conditions on p, ρ and e. Using an internal energy balance is often more convenient for engineering applications and makes easier the extension of the scheme to homogeneous two-phase flow models. But the main advantage of dealing with the internal energy formulation lies in the ability to control the positivity of the internal energy through the numerical scheme, thanks to an upwind procedure. However, it is well known that a blunt discretization in non-conservative variables generally leads to non-entropic solutions. Indeed a straightforward upwind discretization of system (1.1.2) introduces a numerical viscosity which creates source terms in the internal energy balance and in the kinetic energy balance; these terms are localized at shocks and their measure does not vanish as the space and time discretization steps tend to zero. This becomes clear when deriving a discrete kinetic energy inequality from equations (1.1.2b) and (1.1.2a), following the continuous method. A positive residual term appears, yielding an L^2 estimate on the solution which does not tend to zero with the mesh size. A careful choice of a corrective term in the discrete internal energy balance must then be performed to compensate this residual term in the discrete total energy balance and thus ensure the consistency of the scheme.

Our scheme follows two steps. First a tentative velocity is computed from the momentum balance. As in [77, 71], a scaling is introduced on the pressure gradient of the momentum balance in order to allow the derivation a discrete kinetic energy balance. In a second step, a non-linear system is solved to find a pressure correction to the velocity such that the mass balance and the internal energy balance are satisfied.

We prove the existence of a discrete solution in the multi-dimensional case. The positivity of the internal energy in the pressure-correction scheme is proved under the assumption that the corrective source term is positive, which is ensured by construction. For a solution featuring shocks we prove the consistency, in the sense that a limit of a converging sequence of solutions is shown to satisfy the weak form of the Euler equations (1.1.1). Therefore our scheme preserves the energy of the flow (*i.e.* the integral of the total energy over the computational domain), and keeps the velocity and pressure constant across the 1-dimensional contact discontinuity.

This article is organized as follows. In section 2, we introduce the cell-centered finite-volume discretization of the compressible Euler equations. Our pressure-correction scheme is given in section 3 along with the derivation of the source term of the internal energy balance. In section 4, we show several discrete properties of the numerical scheme. Section 5 introduces another pressure correction scheme called SLK [102] which will be compared to our method in

the numerical experiments of section 6.

1.2 Space and time discretization

1.2.1 Discretization of Ω

Let $\mathcal{T}$ be a family of disjoint convex polygonal subsets of Ω, called *control volumes*, such that $\bar{\Omega} = \cup_{K\in\mathcal{T}}\bar{K}$. For a control volume K, we denote $\partial K = \bar{K} \setminus K$ its boundary and $|K|$ its d-dimensional measure.

The edges ($d = 2$) or faces ($d = 3$) of all control volumes of $\mathcal{T}$ form a family $\mathcal{E}$ of disjoint subsets of $\bar{\Omega}$ such that for all $\sigma \in \mathcal{E}$ there exists $H \subset \mathbb{R}^{d-1}$ and $K \in \mathcal{T}$ with $\bar{\sigma} = \partial K \cap H \neq \emptyset$. The $(d-1)$-dimensional measure of an edge $\sigma \in \mathcal{E}$ is denoted $|\sigma|$. We define $\mathcal{E}_K$ the subset of edges $\sigma \in \mathcal{E}$ verifying $\sigma \cap \partial K \neq \emptyset$. Given two control volumes $K, L \in \mathcal{T}^2$ with $\bar{K} \cap \bar{L} = \bar{\sigma} \neq \emptyset$ we denote their common edge $\sigma = K|L$. The normal vector to a face σ pointing outwards the control volume $K \in \mathcal{T}$ is denoted by $\boldsymbol{n}_{K,\sigma} = \boldsymbol{n}_{K|L} = -\boldsymbol{n}_{L|K}$.

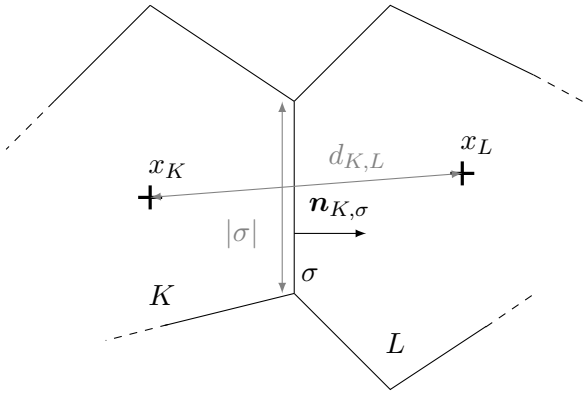


FIG. 1.1 – *Cell-centered finite-volume discretization.*

We define the family $\mathcal{P} = (\boldsymbol{x}_K)_{K\in\mathcal{T}}$ of points of Ω, where $\boldsymbol{x}_K \in K$. For an interface $\sigma \in \mathcal{E}_{int}$ separating cells K and L, the point $\boldsymbol{x}_\sigma$ of $\sigma \in \mathcal{E}$ is defined as the intersection between the line segment $\boldsymbol{x}_K\boldsymbol{x}_L$ and the hyperplane containing σ We define

$$\mathrm{d}_{K,L} = |\boldsymbol{x}_K - \boldsymbol{x}_L| \text{ and } \mathrm{d}_{K,\sigma} = |\boldsymbol{x}_K - \boldsymbol{x}_\sigma|, \tag{1.2.1a}$$

$$\alpha_{K,L} = \frac{d_{L,\sigma}}{d_{K,L}} \text{ and } \alpha_{L,K} = \frac{d_{K,\sigma}}{d_{K,L}}. \tag{1.2.1b}$$

where $|\cdot|$ is the L^2 norm in $\mathbb{R}^d$.

The set of strictly interior edges is denoted by $\mathcal{E}_{\mathrm{int}}$, and the set of edges lying on the boundary is denoted by $\mathcal{E}_{\mathrm{ext}}$. For simplicity, we shall assume in the exposition of the scheme that $\boldsymbol{u}\cdot\boldsymbol{n} = 0$ on the boundary $\partial\Omega$.

We also introduce the size of a control volume $h_K = \mathrm{diam}(K)$, the size of an edge $h_{K|L} = (h_K + h_L)/2$ and the global mesh size $h_\mathcal{T} = \sup_{K\in\mathcal{T}} h_K$.

1.2.2 Discrete gradient and divergence

The discrete velocity divergence and pressure gradient operators are defined by duality; indeed, we look for a discrete equivalent of the following statement: if $\boldsymbol{u} : \Omega \to \mathbb{R}^d$ and $p : \Omega \to \mathbb{R}$ are

sufficiently regular functions such that $\boldsymbol{u}\cdot\boldsymbol{n}=0$ on $\partial\Omega$ then

$$\int_\Omega \left(p\,\mathrm{div}\,\boldsymbol{u}+\boldsymbol{u}\cdot\nabla p\right)\,\mathrm{d}\boldsymbol{x}=0.$$

Lemma 1.2.1 (Discrete divergence and discrete gradient)**.** *Let $\mathcal{T}$ be a finite volume mesh as defined in the previous section, and for $K,L\in\mathcal{T}$, let $\alpha_{K,L}$ and $\alpha_{L,K}$ be defined by* (1.2.1b)*. Let $(\boldsymbol{u}_K)_{K\in\mathcal{T}}$ be a discrete velocity field, $(p_K)_{K\in\mathcal{T}}$ be a discrete pressure field. We define the discrete divergence of the velocity field by:*

$$\mathrm{div}_K(\boldsymbol{u})=\frac{1}{|K|}\sum_{\sigma\in\mathcal{E}(K)}|\sigma|u_{K,\sigma},\quad\forall K\in\mathcal{T},\tag{1.2.2}$$

where

$$u_{K,\sigma}=\begin{cases}\boldsymbol{u}_{\sigma,c}\cdot\boldsymbol{n}_{K,\sigma} & \forall\sigma\in\mathcal{E}_{\mathrm{int}},\\ 0 & \forall\sigma\in\mathcal{E}_{ext},\end{cases}\ \textit{and}\ \boldsymbol{u}_{\sigma,c}=\alpha_{L,K}\boldsymbol{u}_K+\alpha_{K,L}\boldsymbol{u}_L\ \textit{for}\ \sigma=\sigma_{KL}\in\mathcal{E}_{\mathrm{int}}.\tag{1.2.3}$$

with σ_{KL} denoting the interface between cells K and L. Next we define the discrete gradient of the pressure field by

$$\nabla_K p=\frac{1}{|K|}\sum_{L\in\mathcal{N}(K)}|\sigma_{KL}|(\alpha_{K,L}p_K+\alpha_{L,K}p_L)\boldsymbol{n}_{K,L}+\sum_{\sigma\in\mathcal{E}_{ext}\cap\mathcal{E}_K}|\sigma|p_K\boldsymbol{n}_{K,\sigma},\quad\forall K\in\mathcal{T},\tag{1.2.4}$$

where $\mathcal{N}(K)$ denotes the set of the neighbouring cells to K. Then:

$$\sum_{K\in\mathcal{T}}|K|\left(\boldsymbol{u}_K\cdot\nabla_K p+p_K\,\mathrm{div}_K\,\boldsymbol{u}\right)=0.\tag{1.2.5}$$

Furthermore,

$$\begin{aligned}&\boldsymbol{u}_K\cdot\nabla_K p+p_K\,\mathrm{div}_K\,\boldsymbol{u}=\widetilde{\mathrm{div}}_K(p\boldsymbol{u}),\\ &\textit{with}\ \widetilde{\mathrm{div}}_K(p\boldsymbol{u})=\frac{1}{|K|}\sum_{\sigma\in\mathcal{E}_K}|\sigma|(\widetilde{p\boldsymbol{u}})_\sigma\cdot\boldsymbol{n}_{K,\sigma}\\ &\textit{and}\ (\widetilde{p\boldsymbol{u}})_\sigma=\begin{cases}\alpha_{L,K}p_L\boldsymbol{u}_K+\alpha_{K,L}p_K\boldsymbol{u}_L\ \textit{for}\ \sigma=K|L\in\mathcal{E}_{\mathrm{int}},\\ 0\ \textit{if}\ \sigma\in\mathcal{E}_{\mathrm{ext}}.\end{cases}\end{aligned}\tag{1.2.6}$$

Proof. From the definition (1.2.2) of the discrete divergence operator, we have

$$\begin{aligned}\sum_{K\in\mathcal{T}}|K|p_K\,\mathrm{div}_K\,\boldsymbol{u}&=\sum_{K\in\mathcal{T}}p_K\sum_{L\in\mathcal{N}(K)}|\sigma_{KL}|(\alpha_{L,K}\boldsymbol{u}_K+\alpha_{K,L}\boldsymbol{u}_L)\cdot\boldsymbol{n}_{K,L}\\&=\sum_{K\in\mathcal{T}}p_K\sum_{L\in\mathcal{N}(K)}|\sigma_{KL}|\alpha_{L,K}\boldsymbol{u}_K\cdot\boldsymbol{n}_{K,L}+\sum_{K\in\mathcal{T}}p_K\sum_{L\in\mathcal{N}(K)}|\sigma_{KL}|\alpha_{K,L}\boldsymbol{u}_L\cdot\boldsymbol{n}_{K,L}\end{aligned}$$

We rewrite the second sum of the right hand side as:

$$\begin{aligned}\sum_{K\in\mathcal{T}}p_K\sum_{L\in\mathcal{N}(K)}|\sigma|\alpha_{K,L}\boldsymbol{u}_L\cdot\boldsymbol{n}_{K,L}&=\sum_{\substack{\sigma\in\mathcal{E}_{int}\\ \sigma=K|L}}|\sigma|(p_K\alpha_{K,L})\boldsymbol{u}_L\boldsymbol{n}_{K,L}+p_L\alpha_{L,K}\boldsymbol{u}_K\cdot\boldsymbol{n}_{L,K})\\&=\sum_{K\in\mathcal{T}}\sum_{L\in\mathcal{N}(K)}|\sigma_{KL}|p_L\alpha_{L,K}\boldsymbol{u}_K\cdot\boldsymbol{n}_{L,K},\end{aligned}$$

and therefore

$$\sum_{K\in\mathcal{T}} |K| p_K \operatorname{div}_K \boldsymbol{u} = \sum_{K\in\mathcal{T}} \sum_{L\in\mathcal{N}(K)} |\sigma_{KL}|(\alpha_{L,K}p_K - \alpha_{L,K}p_L)\boldsymbol{u}_K\cdot\boldsymbol{n}_{K,L}$$
$$= -\sum_{K\in\mathcal{T}} \left[\sum_{L\in\mathcal{N}(K)} |\sigma_{KL}|(\alpha_{K,L}p_K + \alpha_{L,K}p_L)\boldsymbol{u}_K\cdot\boldsymbol{n}_{K,L} - \sum_{\sigma\in\mathcal{E}_{ext}\cap\mathcal{E}_K} |\sigma| p_K \boldsymbol{n}_{K,\sigma} \right],$$

thanks to the fact that $\alpha_{L,K} = 1-\alpha_{K,L}$ and that $\sum_{\sigma\in\mathcal{E}_K} |\sigma| p_K \boldsymbol{u}_K\cdot\boldsymbol{n}_{K,\sigma} = 0$. This concludes the proof of (1.2.5).

Let us now turn to the proof of (1.2.6). By definition,

$$|K|(\boldsymbol{u}_K\cdot\nabla_K p + p_K \operatorname{div}_K \boldsymbol{u}) = \sum_{L\in\mathcal{N}(K)} |\sigma_{KL}| \left[(\alpha_{K,L}p_K + \alpha_{L,K}p_L)\boldsymbol{u}_K\cdot\boldsymbol{n}_{K,L} + p_K(\alpha_{L,K}\boldsymbol{u}_K + \alpha_{K,L}\boldsymbol{u}_L)\cdot\boldsymbol{n}_{K,L}\right]$$
$$+ \boldsymbol{u}_K \cdot \sum_{\sigma\in\mathcal{E}_{ext}\cap\mathcal{E}_K} |\sigma| p_K \boldsymbol{n}_{K,\sigma}.$$

Therefore,

$$|K|(\boldsymbol{u}_K\cdot\nabla_K p + p_K \operatorname{div}_K \boldsymbol{u}) = \sum_{L\in\mathcal{N}(K)} |\sigma_{KL}| \left(\alpha_{L,K}p_L\boldsymbol{u}_K\cdot\boldsymbol{n}_{K,L} + p_K\alpha_{K,L}\boldsymbol{u}_L\cdot\boldsymbol{n}_{K,L}\right)$$
$$+ \sum_{L\in\mathcal{N}(K)} |\sigma| p_K \boldsymbol{u}_K\cdot\boldsymbol{n}_{K,L} + \sum_{\sigma\in\mathcal{E}_{ext}\cap\mathcal{E}_K} |\sigma| p_K \boldsymbol{u}_K\cdot\boldsymbol{n}_{K,\sigma}$$
$$= \widetilde{\operatorname{div}}_K(p\boldsymbol{u}),$$

which concludes the proof. □

1.2.3 Upwind choice and discrete divergence operators

We need to discretize the term $\operatorname{div}(\rho\boldsymbol{u})$ in the mass equation (1.1.2a) and $\operatorname{div}(e\rho\boldsymbol{u})$ in the momentum equation (1.1.2b). If $\rho:\Omega\to\mathbb{R}$, $e:\Omega\to\mathbb{R}$, and $\boldsymbol{u}\to\mathbb{R}^d$ are regular functions on the cells of a given mesh $\mathcal{T}$ of Ω then the Stokes formula yields

$$\int_\Omega \operatorname{div}(\rho\boldsymbol{u}) = \sum_{\sigma\in\mathcal{E}_K} \int_\sigma \rho\boldsymbol{u}\cdot\boldsymbol{n}_{K,\sigma}\,\mathrm{d}\gamma = 0,$$

(where $\mathrm{d}\gamma$ denotes the integration with respect to the $d-1$ dimensional measure on $\partial\Omega$).

Consider now the functions $\rho:\Omega\to\mathbb{R}$, $e:\Omega\to\mathbb{R}$ and $\boldsymbol{u}\to\mathbb{R}^d$ which are piecewise constant on the cells of a given mesh $\mathcal{T}$ of Ω. The numerical flux associated with the normal flux in the Stokes formula reads:

$$F_{K,\sigma} = |\sigma|\rho_\sigma u_{K,\sigma} \tag{1.2.7}$$

where $u_{K,\sigma}$ is defined by (1.2.3) and the edge value ρ_σ at $\sigma = K|L\in\mathcal{E}_{\rm int}$ is the upwind value of ρ on edge σ. We recall that for a given piecewise constant function a, the upwind choice a_σ on an edge σ of the mesh is defined by:

$$a_\sigma = \varepsilon_{K,\sigma}a_K + (1-\varepsilon_{K,\sigma})a_L \quad \text{with } \varepsilon_{K,\sigma} = \begin{cases} 1 \text{ if } u_{K,\sigma} > 0 \\ 0 \text{ else.} \end{cases} \tag{1.2.8}$$

We may thus define a discrete divergence operator of the functions $\rho\boldsymbol{u}$ and $\rho e\boldsymbol{u}$ as:

$$\mathrm{div}_K(\rho\boldsymbol{u}) = \sum_{\sigma\in\mathcal{E}_K} |\sigma| F_{K,\sigma} \text{ and } \mathrm{div}_K(\rho e\boldsymbol{u}) = \sum_{\sigma\in\mathcal{E}_K} |\sigma| e_\sigma F_{K,\sigma}, \tag{1.2.9}$$

where e_σ is the upwind value of e on the edge σ of the mesh, as defined by (1.2.8).

In the same spirit, we discretize the non linear convection term $\mathbf{div}(\rho\boldsymbol{u}\otimes\boldsymbol{v})$ in the following way. Let $\rho : \Omega \to \mathbb{R}$, $\boldsymbol{u}, \boldsymbol{v} : \Omega \to \mathbb{R}^d$ be piecewise constant functions on the cells of a given mesh $\mathcal{T}$ of Ω

$$\mathbf{div}_K(\rho\boldsymbol{u}\otimes\boldsymbol{v}) = \sum_{\sigma\in\mathcal{E}_K} |\sigma| \boldsymbol{v}_\sigma \rho_\sigma u_{K,\sigma},$$

where $\boldsymbol{v}_\sigma$ is the upwind edge value of $\boldsymbol{v}$ defined by (1.2.8) (componentwise).

1.2.4 Time discretization

We are now ready to introduce the space-time discretization of Problem (1.1.2). For the sake of clarity, we start with an implicit-in-time scheme. Let δt be a time discretization step, which we assume to be constant for the sake of simplicity. Let $N \in \mathbb{N}$ be the number of time discretization steps, and let $\delta t = \frac{T}{N}$ be the time step; the discrete time is defined as $t^n = n\delta t$ for $n \in [\![0, N]\!]$. The scheme is colocated; the discrete unknowns are therefore the discrete density, internal energy and velocity fields $\{\rho_K^n, e_K^n, \boldsymbol{u}_K^n; K \in \mathcal{T}, n \in [\![1, N]\!]\}$. Assuming a given initial state $(\rho_K^0, \boldsymbol{u}_K^0, e_K^0)_{K\in\mathcal{T}}$ and $p_K^0 = (\gamma-1)\rho_K^0 e_K^0$, the implicit-in-time discretization of the original problem (1.1.2) reads:

$$\forall K \in \mathcal{T},\ \forall n \in [\![0, N-1]\!],$$

$$\frac{|K|}{\delta t}(\rho_K^{n+1} - \rho_K^n) + \mathrm{div}_K(\rho^{n+1}\boldsymbol{u}^{n+1}) = 0, \tag{1.2.10a}$$

$$\frac{|K|}{\delta t}(\rho_K^n \boldsymbol{u}_K^{n+1} - \rho_K^{n-1}\boldsymbol{u}_K^n) + \mathbf{div}_K(\rho^n \boldsymbol{u}^n \otimes \boldsymbol{u}^{n+1}) + \nabla_K p^n = 0, \tag{1.2.10b}$$

$$\frac{|K|}{\delta t}(\rho_K^{n+1} e_K^{n+1} - \rho_K^n e_K^n) + \mathrm{div}_K(e^{n+1}\rho^{n+1}\boldsymbol{u}^{n+1}) + p_K^{n+1}\mathrm{div}_K(\boldsymbol{u}^{n+1}) = S_K^{n+1} \tag{1.2.10c}$$

$$p_K^{n+1} = \wp(\rho_K^{n+1}, e_K^{n+1}), \tag{1.2.10d}$$

where div_K and ∇_K denote the values of the discrete divergence and discrete gradient operators, defined in Lemma 1.2.1. S_K^{n+1} is a corrective term needed for the scheme to converge to a weak solution of the Euler equations and which is defined in the sequel.

In real-life applications, the implicit scheme is too expensive in terms of memory and CPU requirements, and semi-implicit schemes are often preferred. We choose here a projection scheme which was recently developed for staggered schemes in [77].

As in incompressible pressure-correction schemes, a tentative velocity is computed using the momentum balance (4.2.3). Then a non-linear problem is solved in order to find a pressure correction to the velocity (4.2.4) such that the mass balance (4.2.5) and the internal energy balance (4.2.6) are verified. In some respect, the velocity update (4.2.4) can be interpreted as a Hodge decomposition [91].

Initialization

$$\forall K \in \mathcal{T}, \quad \rho_K^0,\ \boldsymbol{u}_K^0,\ p_K^0 \text{ given}; \qquad \rho_K^{-1} = \rho_K^0; \qquad e_K^0 = p_K^0/(\gamma-1)\rho_K^0; \tag{1.2.11}$$

Iterations for $n = 0, 1, \ldots, N-1$:

- *Prediction*: compute $\tilde{\boldsymbol{u}}_K^{n+1}$ by solving for all $K \in \mathcal{T}$,

$$\frac{|K|}{\delta t}(\rho_K^n \tilde{\boldsymbol{u}}_K^{n+1} - \rho_K^{n-1}\boldsymbol{u}_K^n) + \mathbf{div}_K(\rho^n \boldsymbol{u}^n \otimes \tilde{\boldsymbol{u}}^{n+1}) + \widetilde{\nabla_K p}^{\,n} = 0, \tag{1.2.12}$$

$$\text{with } \widetilde{\nabla_K p}^{\,n} = \sqrt{\frac{\rho_K^n}{\rho_K^{n-1}}}\nabla_K p^n \tag{1.2.13}$$

- *Projection-correction*: compute $\boldsymbol{u}_K^{n+1}$, p_K^{n+1}, e_K^{n+1}, ρ_K^{n+1} by solving the non-linear system of equations for all $K \in \mathcal{T}$,

$$\frac{|K|}{\delta t}\rho_K^n\left(\boldsymbol{u}_K^{n+1} - \tilde{\boldsymbol{u}}_K^{n+1}\right) + |K|\left(\nabla_K p^{n+1} - \widetilde{\nabla_K p}^{\,n}\right) = 0, \tag{1.2.14a}$$

$$\frac{|K|}{\delta t}(\rho_K^{n+1} - \rho_K^n) + \mathrm{div}_K(\rho^{n+1}\boldsymbol{u}^{n+1}) = 0, \tag{1.2.14b}$$

$$\frac{|K|}{\delta t}(\rho_K^{n+1}e_K^{n+1} - \rho_K^n e_K^n) + \mathrm{div}_K(e^{n+1}\rho^{n+1}\boldsymbol{u}^{n+1}) + p_K^{n+1}\,\mathrm{div}_K\,\boldsymbol{u}^{n+1} = S_K^{n+1} \tag{1.2.14c}$$

$$p_K^{n+1} = (\gamma - 1)\rho_K^{n+1}e_K^{n+1}. \tag{1.2.14d}$$

1.3 Stability of the scheme and existence of a solution

Proposition 1.3.1 (Positivity of the internal energy)**.** *Assume that* $\forall K \in \mathcal{T}$, $e_K^n \geq 0$, $S_K^{n+1} \geq 0$ *and* $\rho_K^n > 0$*, and let* S_K^{n+1} *and* S_K^{n+1} *satisfy* (4.2.3)–(1.2.14d) *then*

$$\forall K \in \mathcal{T}, \rho_K^{n+1} \text{ and } e_K^{n+1} \geq 0,$$

Proof. The positivity of the density is a consequence of the upwind discretization of the mass balance equation, see e.g. [65, Lemma 2.1]. We now show that the internal energy remains positive as long as the source term S_K^{n+1} of the internal energy balance is positive.

Multiplying the internal energy equation (4.2.6) by $(e_K^{n+1})^-$ we get:

$$T_1 + T_2 + T_3 = 0,$$

with:

$$T_1 = -\sum_{K\in\mathcal{T}}(e_K^{n+1})^-|K|\left[\frac{1}{\delta t}(\rho_K^{n+1}e_K^{n+1} - \rho_K^n e_K^n) + \mathrm{div}_K(e^{n+1}\rho^{n+1}\boldsymbol{u}^{n+1})\right]$$

$$T_2 = -\sum_{K\in\mathcal{T}}|K|\left[p_K^{n+1}(e_K^{n+1})^-\,\mathrm{div}_K(\rho^{n+1}\boldsymbol{u}^{n+1})\right]$$

$$T_3 = \sum_{K\in\mathcal{T}}\left[(e_K^{n+1})^- S_K^{n+1}\right].$$

The term T_2 is equal to zero, thanks to the form of the EOS (1.2.14d) and to the fact that $(e_K^{n+1})^- = -\min(0, e_K^{n+1})$. The positivity of S_K^{n+1} ensures that $T_3 \geq 0$. By applying Lemma 1.3.2 — which we recall below — on T_1 we get:

$$T_1 \geq \frac{1}{2}\sum_{K\in\mathcal{T}}\frac{|K|}{\delta t}\left\{\rho_K^{n+1}[(e_K^{n+1})^-]^2 - \rho_K^n[(e_K^n)^-]^2\right\} = \frac{1}{2}\sum_{K\in\mathcal{T}}\frac{|K|}{\delta t}\left(\rho_K^{n+1}(e_K^{n+1})^-\right)^2$$

thanks to the fact that $e_K^n \geq 0$. Gathering all terms yields

$$\sum_{K\in\mathcal{T}} \frac{|K|}{\delta t}\rho_K^{n+1}[(e_K^{n+1})^-]^2 \leq 0$$

As a result, for all $K \in \mathcal{T}$, $\min(e_K^{n+1}, 0) = 0$ and therefore $e_K^{n+1} \geq 0$.

Lemma 1.3.2 (Lemma 2.2, [65])**.** *Let $(\rho_K)_{K\in\mathcal{T}} \subset \mathbb{R}_+$, $(\rho_K^*)_{K\in\mathcal{T}} \subset \mathbb{R}_+$ $(u_{K,\sigma})_{K\in\mathcal{T},\sigma\in\mathcal{E}_K} \subset \mathbb{R}$ be three families of real numbers satisfying:*

$$\sum_{K\in\mathcal{T}} \left[\frac{|K|}{\delta t}(\rho_K - \rho_K^*) + \sum_{\sigma\in\mathcal{E}_K} |\sigma|\rho_\sigma u_{K,\sigma}\right] = 0$$

Then, for all real number y_K we have:

$$-\sum_{K\in\mathcal{T}} (y_K)^- \left[\frac{|\sigma|}{\delta t}(\rho_K y_K - \rho_K^* y_K^*) + \sum_{\sigma\in\mathcal{E}_K} |\sigma| y_\sigma \rho_\sigma u_{K,\sigma}\right] \geq \frac{1}{2}\sum_{K\in\mathcal{T}} \frac{|K|}{\delta t}\left\{\rho_K[(y_K)^-]^2 - \rho_K^*[(y_K^*)^-]^2\right\}$$

□

Let us now give a lemma which is a direct consequence of Lemma A.2 (ii) from [77], and which is used to construct a positive corrective source term in the internal energy balance.

Lemma 1.3.3. *Let $\mathcal{T}$ be a mesh of Ω, $\rho = (\rho_K)_{K\in\mathcal{T}}$, $\rho^* = (\rho_K^*)_{K\in\mathcal{T}}$, $\boldsymbol{u} = (\boldsymbol{u}_K)_{K\in\mathcal{T}}$, $\boldsymbol{u}^* = (\boldsymbol{u}_K^*)_{K\in\mathcal{T}}$. Let $F_{K,\sigma}$ and div_K be defined by* (1.2.2)–(1.2.7)*; then the following result holds:*

$$\left[\frac{|K|}{\delta t}(\rho_K \boldsymbol{u}_K - \rho_K^* \boldsymbol{u}_K^*) + \mathrm{div}_K(\rho\boldsymbol{u})\right]\cdot \boldsymbol{u}_K = \frac{|K|}{2\delta t}\left(\rho_K|\boldsymbol{u}_K|^2 - \rho_K^*|\boldsymbol{u}_K^*|^2\right) + \frac{1}{2}\sum_{\sigma\in\mathcal{E}_K} \boldsymbol{u}_K\cdot\boldsymbol{u}_L F_{K,\sigma} + \frac{|K|}{2\delta t}\rho_K^*(\boldsymbol{u}_K - \boldsymbol{u}_K^*)^2 + \frac{1}{4}\sum_{\sigma\in\mathcal{E}_K}(\boldsymbol{u}_K - \boldsymbol{u}_L)^2\ |F_{K,\sigma}|$$

Proof. First we apply Lemma A.2 (ii) from [77] with $\psi(s) = s^2/2$, which yields for all $K \in \mathcal{T}$:

$$\left[\frac{|K|}{\delta t}(\rho_K \boldsymbol{u}_K - \rho_K^* \boldsymbol{u}_K^*) + \sum_{\sigma\in\mathcal{E}_K} \boldsymbol{u}_\sigma F_{K,\sigma}\right]\cdot\boldsymbol{u}_K = -\left[\frac{|K|}{2\delta t}\left(\rho_K|\boldsymbol{u}_K|^2 - \rho_K^*|\boldsymbol{u}_K^*|^2\right) + \frac{1}{2}\sum_{\sigma\in\mathcal{E}_K}|\boldsymbol{u}_\sigma|^2 F_{K,\sigma}\right] + \frac{|K|}{2\delta t}\rho_K^*(\boldsymbol{u}_K - \boldsymbol{u}_K^*)^2 - \frac{1}{2}\sum_{\sigma\in\mathcal{E}_K}(\boldsymbol{u}_\sigma - \boldsymbol{u}_K)^2 F_{K,\sigma}.$$

Introducing the following decomposition of the upwind value $\boldsymbol{u}_\sigma$ with respect to the numerical flux $F_{K,\sigma}$:

$$F_{K,\sigma}\boldsymbol{u}_\sigma = \frac{F_{K,\sigma}}{2}(\boldsymbol{u}_K + \boldsymbol{u}_L) + \frac{|F_{K,\sigma}|}{2}(\boldsymbol{u}_K - \boldsymbol{u}_L),\ \forall \sigma = K|L \in \mathcal{E}_{int}, \tag{1.3.1}$$

we can rewrite the two edge values as:

$$\frac{1}{2}\sum_{\sigma\in\mathcal{E}_K}|\boldsymbol{u}_\sigma|^2 F_{K,\sigma} - \frac{1}{2}\sum_{\sigma\in\mathcal{E}_K}(\boldsymbol{u}_\sigma - \boldsymbol{u}_K)^2 F_{K,\sigma} = \frac{1}{2}\sum_{\substack{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}\\ \sigma=K|L}} \boldsymbol{u}_K\cdot\boldsymbol{u}_L F_{K,\sigma} + \frac{1}{2}\sum_{\substack{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}\\ \sigma=K|L}} (\boldsymbol{u}_K - \boldsymbol{u}_L)\cdot\boldsymbol{u}_K|F_{K,\sigma}|.$$

We thus get

$$\left[\frac{|K|}{\delta t}(\rho_K \boldsymbol{u}_K - \rho_K^* \boldsymbol{u}_K^*) + \sum_{\sigma\in\mathcal{E}_K} \boldsymbol{u}_\sigma F_{K,\sigma}\right]\cdot \boldsymbol{u}_K = \frac{|K|}{2\delta t}\left(\rho_K|\boldsymbol{u}_K|^2 - \rho_K^*|\boldsymbol{u}_K^*|^2\right) + \frac{1}{2}\sum_{\substack{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}\\ \sigma=K|L}} \boldsymbol{u}_K\cdot\boldsymbol{u}_L F_{K,\sigma} + \frac{|K|}{2\delta t}\rho_K^*(\boldsymbol{u}_K-\boldsymbol{u}_K^*)^2 + \frac{1}{4}\sum_{\substack{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}\\ \sigma=K|L}} (\boldsymbol{u}_K-\boldsymbol{u}_L)^2\,|F_{K,\sigma}| + T_K \tag{1.3.2}$$

with the residual term:

$$\begin{aligned} T_K &= \frac{1}{2}\sum_{\substack{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}\\ \sigma=K|L}} |\sigma|(\boldsymbol{u}_K-\boldsymbol{u}_L)\cdot\boldsymbol{u}_K\rho_\sigma|u_{K,\sigma}| - \frac{1}{4}\sum_{\substack{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}\\ \sigma=K|L}} (\boldsymbol{u}_K-\boldsymbol{u}_L)^2|F_{K,\sigma}| \\ &= \frac{1}{4}\sum_{\substack{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}\\ \sigma=K|L}} (|\boldsymbol{u}_K|^2-|\boldsymbol{u}_L|^2)|F_{K,\sigma}|. \end{aligned}$$

Hence $\sum_{K\in\mathcal{T}} T_K = 0$, and we conclude the proof by summing (1.3.2) over $K\in\mathcal{T}$. □

Thanks to this lemma, we now derive a discrete kinetic energy balance from the momentum balance (4.2.3), the mass balance (4.2.5) and the velocity update (4.2.4) in the same fashion as in the staggered discretization [77].

Proposition 1.3.4 (Discrete kinetic energy balance)**.** *Let* $(\rho_K^n, e_K^n, \boldsymbol{u}_K^n, \tilde{\boldsymbol{u}}_K^n, p_K^n)_{\substack{K\in\mathcal{T}\\ n\in[\![0,N-1]\!]}}$ *be a solution to* (4.2.3)-(1.2.14)*; then the following local discrete kinetic energy balance holds for all* $K\in\mathcal{T}$, $n\in[\![0,N-1]\!]$*:*

$$\frac{|K|}{2\delta t}\left(\rho_K^n|\boldsymbol{u}_K^{n+1}|^2 - \rho_K^{n-1}|\boldsymbol{u}_K^n|^2\right) + \frac{1}{2}\sum_{\substack{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}\\ \sigma=K|L}} \tilde{\boldsymbol{u}}_K^{n+1}\cdot\tilde{\boldsymbol{u}}_L^{n+1}F_{K,\sigma}^n + \boldsymbol{u}_K^{n+1}\cdot\nabla_K p^{n+1} + P_K^{n+1} = R_K^{n+1} \tag{1.3.3}$$

where:

$$R_K^{n+1} = -\frac{|K|}{2\delta t}\rho_K^{n-1}(\tilde{\boldsymbol{u}}_K^{n+1}-\boldsymbol{u}_K^n)^2 - \frac{1}{4}\sum_{\substack{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}\\ \sigma=K|L}} |\tilde{\boldsymbol{u}}_K^{n+1}-\tilde{\boldsymbol{u}}_L^{n+1}|^2|F_{K,\sigma}^n|, \tag{1.3.4}$$

$$P_K^{n+1} = \frac{\delta t}{2|K|\rho_K^n}\left|\nabla_K p^{n+1}\right|^2 - \frac{\delta t}{2|K|\rho_K^{n-1}}\left|\nabla_K p^n\right|^2. \tag{1.3.5}$$

Proof. We start from the momentum balance (4.2.3) multiplied by the predicted velocity $\tilde{\boldsymbol{u}}_K^{n+1}$, for a given control volume $K\in\mathcal{T}$:

$$\left[\frac{|K|}{\delta t}(\rho_K^n\tilde{\boldsymbol{u}}_K^{n+1} - \rho_K^{n-1}\boldsymbol{u}_K^n) + \sum_{\sigma\in\mathcal{E}_K}|\sigma|\tilde{\boldsymbol{u}}_\sigma^{n+1}\rho_\sigma^n u_{K,\sigma}^n\right]\cdot\tilde{\boldsymbol{u}}_K^{n+1} + \sqrt{\frac{\rho_K^n}{\rho_K^{n-1}}}\tilde{\boldsymbol{u}}_K^{n+1}\cdot\nabla_K p^n = 0$$

applying Lemma 1.3.3 we have:

$$\frac{|K|}{2\delta t}\left[\rho_K^n|\tilde{\boldsymbol{u}}_K^{n+1}|^2-\rho_K^{n-1}|\boldsymbol{u}_K^n|^2\right]+\frac{1}{2}\sum_{\substack{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}\\ \sigma=K|L}}|\sigma|\tilde{\boldsymbol{u}}_K^{n+1}\cdot\tilde{\boldsymbol{u}}_L^{n+1}\rho_\sigma^n u_{K,\sigma}^n$$
$$+\frac{|K|}{2\delta t}\rho_K^{n-1}(\tilde{\boldsymbol{u}}_K^{n+1}-\boldsymbol{u}_K^n)^2+\frac{1}{4}\sum_{\substack{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}\\ \sigma=K|L}}|\sigma|(\tilde{\boldsymbol{u}}_K^{n+1}-\tilde{\boldsymbol{u}}_L^{n+1})^2\rho_\sigma^n|u_{K,\sigma}^n|$$
$$+\sqrt{\frac{\rho_K^n}{\rho_K^{n-1}}}\tilde{\boldsymbol{u}}_K^{n+1}\nabla_K p^n=0 \quad (1.3.6)$$

The velocity update (4.2.4) can be rewritten as:

$$\frac{|K|}{\delta t}\sqrt{\rho_K^n}\boldsymbol{u}_K^{n+1}+\frac{1}{\sqrt{\rho_K^n}}\nabla_K p^{n+1}=\frac{|K|}{\delta t}\sqrt{\rho_K^n}\tilde{\boldsymbol{u}}_K^{n+1}+\frac{1}{\sqrt{\rho_K^{n-1}}}\nabla_K p^n$$

By taking the square of the previous equality and then multiplying by $\frac{\delta t}{2|K|}$ we get:

$$\frac{|K|}{2\delta t}\rho_K^n|\tilde{\boldsymbol{u}}_K^{n+1}|^2+\sqrt{\frac{\rho_K^n}{\rho_K^{n-1}}}\tilde{\boldsymbol{u}}_K^{n+1}\cdot\nabla_K p^n=\frac{|K|}{2\delta t}\rho_K^n|\boldsymbol{u}_K^{n+1}|^2+\boldsymbol{u}_K^{n+1}\cdot\nabla_K p^{n+1}$$
$$+\frac{\delta t}{2|K|\rho_K^n}\left[\nabla_K p^{n+1}\right]^2-\frac{\delta t}{2|K|\rho_K^{n-1}}\left[\nabla_K p^n\right]^2 \quad (1.3.7)$$

We conclude the proof by summing equations (1.3.7) and (1.3.6). □

Let us now define the source term of the internal energy balance as $S_K^{n+1}=-R_K^{n+1}$:

$$S_K^{n+1}=\frac{|K|}{2\delta t}\rho_K^{n-1}(\tilde{\boldsymbol{u}}_K^{n+1}-\boldsymbol{u}_K^n)^2+\frac{1}{4}\sum_{\substack{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}\\ \sigma=K|L}}|\sigma|\left(\tilde{\boldsymbol{u}}_K^{n+1}-\tilde{\boldsymbol{u}}_L^{n+1}\right)^2\rho_\sigma^n|u_{K,\sigma}^n| \quad (1.3.8)$$

which is obviously positive.

Furthermore with this definition of the source term S_K^{n+1}, the following discrete energy balance holds:

Proposition 1.3.5 (Local total energy balance)**.** *Assume that $\wp$ is given by* (1.2.14d)*, that $\rho_0\geq 0$ and $e_0\geq 0$, and that $\{\rho_K^n,e_K^n,\boldsymbol{u}_K^n,K\in\mathcal{T},n\in[\![0,N]\!]\}$ is a solution to the scheme* (4.2.1)–(1.2.14)*. Then the following local total energy balance holds for all $K\in\mathcal{T}$ and $n\in\{0,\dots,N\}$:*

$$\frac{|K|}{\delta t}\left((\widetilde{\rho_K E_K})^{n+1}-(\widetilde{\rho_K E_K})^n\right)+\sum_{\sigma\in\mathcal{E}_K}|\sigma|e_\sigma^{n+1}\rho_\sigma^{n+1}u_{K,\sigma}^{n+1}+\frac{1}{2}\sum_{\substack{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}\\ \sigma=K|L}}|\sigma|\tilde{\boldsymbol{u}}_K^{n+1}\cdot\tilde{\boldsymbol{u}}_L^{n+1}\rho_\sigma^n u_{K,\sigma}^n$$
$$+|K|\widetilde{\mathrm{div}}_K(p\boldsymbol{u})+P_K^{n+1}=0, \quad (1.3.9)$$

where

$$(\widetilde{\rho_K E_K})^n=\rho_K^n e_K^n+\frac{1}{2}\rho_K^{n-1}|\boldsymbol{u}_K^n|^2 \quad (1.3.10)$$

and where P_K^{n+1} is defined by (1.3.5)*.*

Proof. From the definition of the source terms R_K^{n+1} and S_K^{n+1}, summing the kinetic energy balance (1.3.3) and the internal energy balance (4.2.6) and using the property (1.2.6) yields the total energy balance (1.3.9). □

Remark 1.3.6 (Staggered scheme). A total energy balance can be recovered for staggered finite volumes [77], but this balance only holds globally on Ω whereas we get a local balance from equation (4.2.6).

Remark 1.3.7 (Error pressure term). The additional source term P_K^{n+1} is not compensated in the internal energy balance but it does not impact the convergence of the scheme since P_K^{n+1} vanishes when the time step tends to 0. Indeed formally it behaves like $\delta t^2(\partial_t \nabla p)^2$. One may avoid this pressure source term in the kinetic energy balance by removing the scaling of the pressure gradient introduced in the momentum balance (4.2.3) and in the velocity update (4.2.4). This would yield to a similar kinetic energy balance with a source term R_K^{n+1} which would now read:

$$R_K^{n+1} = -\frac{|K|}{2\delta t}\rho_K^{n-1}\left(\tilde{\mathbf{u}}_K^{n+1} - \boldsymbol{u}_K^n\right)^2 + \frac{|K|}{2\delta t}\rho_K^n\left(\boldsymbol{u}_K^{n+1} - \tilde{\mathbf{u}}_K^{n+1}\right)^2 - \frac{1}{4}\sum_{\Sigma\in\mathcal{E}_K}|\sigma|\left(\tilde{\boldsymbol{u}}_K^{n+1} - \tilde{\boldsymbol{u}}_L^{n+1}\right)^2\rho_\sigma^n\left|u_{K,\sigma}^n\right|$$

However this formulation is dangerous because the corresponding source term in the internal energy balance $S_K^{n+1} = -R_K^{n+1}$ *may* take negative values, and therefore we can no longer assert that the internal energy remains positive as shown in proposition 1.3.1 hereafter.

Remark 1.3.8 (Source term for the centered scheme). If we were to use a centered scheme instead of an upwind scheme for the edge value $\boldsymbol{u}_\sigma$ of the advected velocity, the resulting corrective source term would read:

$$S_K^{n+1} = \frac{|K|}{2\delta t}\rho_K^{n-1}(\tilde{\boldsymbol{u}}_K^{n+1} - \boldsymbol{u}_K^n)^2,\ K\in\mathcal{T}, n\in\mathbb{N}.$$

Regarding the non-linear projection step, the proof of existence relies on the topological degree theory. The reader may find an introduction in [49] and an application to finite volume discretizations of PDE in [56], chapter 6. We proceed in the same way as in section 6.4.1 of [56].

Theorem 1.3.9 (Existence of a solution to the scheme)**.** *Under the assumptions of Proposition 1.3.5, there exists a solution to the scheme* (4.2.3)–(1.2.14d).

Proof. We have to show the existence of a solution to the linear system given by the prediction step (4.2.3) and to the non-linear system (4.2.4)–(1.2.14d). We proceed by induction. Assume that the existence of $(\rho_K^k, p_K^k, \tilde{\boldsymbol{u}}_K^k, \boldsymbol{u}_K^k, e_K^k)_{K\in\mathcal{T}}$ is proved for $0\le k\le n$, and let us prove the existence of $(\rho_K^{n+1}, p_K^{n+1}, \tilde{\boldsymbol{u}}_K^{n+1}, \boldsymbol{u}_K^{n+1}, e_K^{n+1})_{K\in\mathcal{T}}$. The momentum balance (4.2.3) for a control volume $K\in\mathcal{T}$ can be rewritten as follows:

$$A_K^n\tilde{\boldsymbol{u}}_K^{n+1} + \sum_{\sigma=K|L\in\mathcal{E}_K} B_{K,\sigma}^n\tilde{\boldsymbol{u}}_L^{n+1} = C_K^n$$

with the diagonal, off-diagonal and right-hand-side contributions to the linear system:

$$\begin{aligned}
A_K^n &= \frac{|K|}{\delta t}\rho_K^n + \sum_{\sigma\in\mathcal{E}_K}|\sigma|\rho_\sigma^n u_{K,\sigma}^n\alpha_{K,\sigma}^n\\
B_{K,\sigma}^n &= |\sigma|\rho_\sigma^n u_{K,\sigma}^n(1-\alpha_{K,\sigma}^n)\\
C_K^n &= \frac{|K|}{\delta t}\rho_K^{n-1}\boldsymbol{u}_K^n - \sqrt{\frac{\rho_K^n}{\rho_K^{n-1}}}\nabla_K p^n
\end{aligned}$$

and $\alpha_{K,\sigma}^n$ the scalar used to compute the upwind value $\tilde{\boldsymbol{u}}_{K,\sigma}^n$. Owing to the definition of the scalar $\alpha_{K,\sigma}$ in (1.2.8), we have:

$$|A_K^n| - \sum_{\sigma\in\mathcal{E}_K} |B_{K,\sigma}^n| = A_K^n + \sum_{\sigma\in\mathcal{E}_K} B_{K,\sigma}^n$$

Using the discrete mass balance (4.2.5) and assuming the positivity of the density yields:

$$|A_K^n| - \sum_{\sigma\in\mathcal{E}_K} |B_{K,\sigma}^n| = \frac{|K|}{\delta t}\rho_K^{n-1} > 0$$

hence the diagonal dominance of the linear system associated with the prediction equation (4.2.3), which is subsequently invertible and thus the existence of $\tilde{\boldsymbol{u}}_K^{n+1}$.

Let us now prove the existence of $(\rho_K^{n+1}, p_K^{n+1}, \boldsymbol{u}_K^{n+1}, e_K^{n+1})_{K\in\mathcal{T}}$ which satisfy (4.2.4)–(1.2.14d). Let λ be a real number in $[0,1]$. We consider the following non-linear system of equations, which reads for all $K\in\mathcal{T}$:

$$U_{\mathcal{T}}^{N+1} + \lambda\mathcal{F}(U_{\mathcal{T}}^{N+1}) = V_{\mathcal{T}}^N \tag{1.3.11}$$

where

$$\left.\begin{array}{l} U_{\mathcal{T}}^{N+1} = [(\boldsymbol{u}_K^{N+1})_{K\in\mathcal{T}}\ (\rho_K^{N+1})_{K\in\mathcal{T}}\ (\rho_K^{N+1}e_K^{N+1})_{K\in\mathcal{T}}]^t \\ V_{\mathcal{T}}^N = [(\tilde{\boldsymbol{u}}_K^{N+1})_{K\in\mathcal{T}}\ (\rho_K^N)_{K\in\mathcal{T}}\ (\rho_K^N e_K^N)_{K\in\mathcal{T}}]^t \end{array}\right\} \in \mathbb{R}^M, \text{ with } M = 3\,\mathrm{card}\,\mathcal{T}, \text{ and}$$

$$\mathcal{F} : W = ((\boldsymbol{u}_K)_{K\in\mathcal{T}}, (\rho_K)_{K\in\mathcal{T}}, (\rho_K e_K)_{K\in\mathcal{T}}) \mapsto \mathcal{F}(W) = \delta t \begin{bmatrix} \frac{1}{\rho_K^n}\left(\nabla_K p - \widetilde{\nabla_K p}\right) \\ \mathrm{div}_K(\rho\boldsymbol{u}) \\ \left[\mathrm{div}_K(\rho e\boldsymbol{u}) + p_K\,\mathrm{div}_K\,\boldsymbol{u} - S_K^{n+1}\right] \end{bmatrix}$$

with $p_K = (\gamma-1)\rho_K e_K$ and S_K^{n+1} defined by (1.3.8).

For $\lambda = 1$, (1.3.11) is the original projection-correction step at $n = N$. For $\lambda = 0$, the system (1.3.11) is an invertible linear system. Using the second equation of (1.3.11) and summing over all $K\in\mathcal{T}$, we get by conservativity:

$$\sum_{K\in\mathcal{T}} |\rho_K| \le \sum_{K\in\mathcal{T}} |\rho_K^n|$$

which yields a uniform (in λ) estimate on ρ_K^{n+1}. Moreover, with the same arguments as in the proof of Proposition 1.3.1, we get that $\rho_K \ge 0$ and $e_K \ge 0$.

We then take the inner product of the first equation of (1.3.11) with $\rho_K\boldsymbol{u}_K$ and sum over $K\in\mathcal{T}$; we sum the result with the summation of the third equation of (1.3.11) over the mesh and obtain, thanks to the div-∇ duality and to conservativity, the following uniform (in λ) estimate:

$$\sum_{K\in\mathcal{T}} \rho_K e_K + \frac{1}{2}\rho_K^n|\boldsymbol{u}_K|^2 \le C$$

where $C\ge 0$ depends only on known quantities. The map $\mathcal{H}(\lambda,\cdot) = Id - \lambda\mathcal{F}$ defines a homotopy between the map $\mathcal{H}(1,\cdot)$ associated with the original system $U_{\mathcal{T}}^{N+1} + \mathcal{F}(U_{\mathcal{T}}^{N+1}) = V_{\mathcal{T}}^N$ (for $\lambda = 1$) and the identity function (obtained with $\lambda = 0$). Thanks to the uniform estimates, we can can

define a closed ball $\mathcal{B}$ of $\mathbb{R}^m$ with radius large enough to include the set of the solution of Problem (1.3.11), such that the following condition is satisfied:

$$V_{\mathcal{T}}^N \notin \mathcal{H}(\lambda, \partial\mathcal{B}).$$

We can now define the *topological degree* $d(\mathcal{H}(\lambda,\cdot), \mathcal{B}, V_{\mathcal{T}}^N)$ of the map $\mathcal{H}(\lambda,\cdot)$ on the set $\mathcal{B}$, associated with the problem $\mathcal{H}(\lambda, U_{\mathcal{T}}) = V_{\mathcal{T}}^N$. Using the invariance of the topological degree for an homotopy, we have:

$$d(\mathcal{H}(1,\cdot), \mathcal{B}, V_{\mathcal{T}}^N) = d(\mathcal{H}(0,\cdot), \mathcal{B}, V_{\mathcal{T}}^N) = d(Id, \mathcal{B}, V_{\mathcal{T}}^N) = 1$$

We deduce that topological degree of the map $Id - \mathcal{F}$ is non zero. As a result the non-linear problem has at least one solution in $\mathcal{B} \subset \mathbb{R}^m$. □

1.4 Passing to the limit

In this section we are interested in the problem of showing that the limit of the discrete solution to the pressure-correction scheme is a solution to the weak form of the continuous problem when the time step and space discretization step tend to zero. In some respect, it means that the shocks are correctly computed by the pressure-correction scheme since the Rankine-Hugoniot conditions can be readily derived from the weak form the Euler equations.

First we need to introduce further notations and assumptions with respect to the discretization of the problem:

- *Discrete unknown functions and norms.* Let $(\mathcal{T}, \delta t)$ be a space-time discretization. To a set of discrete values $\{z_K^n, K \in \mathcal{T}, n \in [\![0, N-1]\!]\}$, we associate the following piecewise constant function:

$$z_{\mathcal{T},\delta t}(x,t) = \sum_{n=0}^{N-1} \sum_{K\in\mathcal{T}} z_K^n \chi_K(x)\chi_n(t), \forall x \in \Omega,\ \forall t \in [0,T),$$

with χ_K and χ_n defined by

$$\chi_K(x) = \begin{cases} 1 \text{ if } x \in K, \\ 0 \text{ otherwise,} \end{cases} \quad \text{and } \chi_n(t) = \begin{cases} 1 \text{ if } t \in [t^n, t^{n+1}), \\ 0 \text{ otherwise.} \end{cases} \tag{1.4.1}$$

For such a function $z_{\mathcal{T},\delta t}$, we denote its $L^\infty(\Omega \times [0,T)$ norm by

$$\|z_{\mathcal{T},\delta t}\|_\infty = \max\{z_K^n, K \in \mathcal{T}, n \in \mathbb{N}\},$$

and introduce the following BV discrete semi-norms:

$$|z_{\mathcal{T},\delta t}|_{BV_x} = \sum_{n=0}^{N} \delta t \sum_{\sigma=K|L} |z_L^n - z_K^n|,\ |z_{\mathcal{T},\delta t}|_{BV_t} = \sum_{K\in\mathcal{T}} |K| \sum_{n=0}^{N-1} |z_K^{n+1} - z_K^n|$$

Given a solution $\{\rho_K^n, \boldsymbol{u}_K^n, e_K^n, K \in \mathcal{T}, n \in [\![0, N-1]\!]\}$ of the scheme (4.2.1)–(1.2.14), we may thus define the piecewise constant functions $\rho_{\mathcal{T},\delta t}$, $p_{\mathcal{T},\delta t}$, $\boldsymbol{u}_{\mathcal{T},\delta t}$, $e_{\mathcal{T},\delta t}$ and their BV semi-norms.

- *Interpolates and discrete derivative operators.* Let $\varphi \in \mathcal{C}_c^\infty(\Omega \times [0,T])$ be a given test function. We denote by $\phi_{\mathcal{T},\delta t}$ its interpolate for the space–time discretization $(\mathcal{T}, \delta t)$, defined by:

$$\phi_{\mathcal{T},\delta t}(x,t) = \sum_{n=0}^{N-1} \sum_{K\in\mathcal{T}} \phi_K^n \chi_K(x)\chi_n(t), \text{ where } \phi_K^n = \varphi(x_K, t_n), \forall K \in \mathcal{T}, \forall n \in [\![0, N-1]\!]. \tag{1.4.2}$$

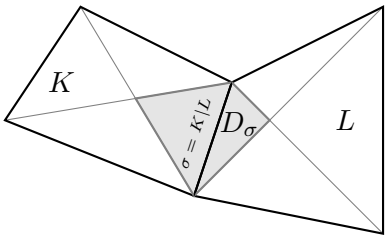


FIG. 1.2 – *Primal mesh and dual mesh.*

We may then define a discrete time-derivative operator $\eth_t$ on the primal grid by:

$$\eth_t \phi_{\mathcal{T},\delta t}(x,t) = \sum_{n=0}^{N-1} \sum_{K\in\mathcal{T}} \frac{\phi_K^{n+1} - \phi_K^n}{\delta t} \chi_K(x)\chi_n(t), \forall x \in \Omega, \forall t \in [0,T).$$

Owing to the regularity of φ, the quantity $\eth_t \phi_{\mathcal{T},\delta t}$ converges uniformly to $\partial_t \varphi$ as the mesh size and time step tend to 0. It is convenient in the convergence analysis to define a discrete gradient of the interpolate of a smooth function on a dual mesh which is composed of the so called "diamond cells", represented on Figure 1.2 and defined as follows. For a control volume $K \in \mathcal{T}$, $\sigma \in \mathcal{E}_K$, we define the half-diamond cell $D_{K,\sigma}$ by the cone with base σ and vertex x_K:

$$D_{K,\sigma} = \{t x_K + (1-t)y, \ t \in [0,1], \ y \in \sigma\} \tag{1.4.3}$$

We denote $|D_{K,\sigma}|$ its d-dimensional measure. We then define the diamond cells D_σ as

$$D_\sigma = \begin{cases} D_{K,\sigma} \cup D_{L,\sigma} \text{ if } \sigma = K|L \in \mathcal{E}_{int}, \\ D_{K,\sigma} \text{ if } \sigma \in \mathcal{E}_{ext} \text{ and } \bar{\sigma} = \bar{K} \cap \bar{\Omega}. \end{cases} \tag{1.4.4}$$

A discrete gradient operator $\nabla_{\mathcal{E}} \phi_{\mathcal{T},\delta t}$ of the interpolate $\phi_{\mathcal{T},\delta t}$ may then be defined as:

$$\nabla_{\mathcal{E}} \phi_{\mathcal{T},\delta t}(x,t) = \sum_{n=0}^{N-1} \sum_{\substack{\sigma\in\mathcal{E}_{int} \\ \sigma = K|L}} \frac{|\sigma|}{|D_\sigma|} (\phi_L^n - \phi_K^n) \boldsymbol{n}_{K,L} \chi_\sigma(x)\chi_n(t), \ \forall x \in \Omega, \ \forall t \in [0,T), \tag{1.4.5}$$

where χ_σ is the characteristic function of the dual cell D_σ, $\sigma \in \mathcal{E}_{int}$, defined by

$$\chi_\sigma(x) = \begin{cases} 1 \text{ if } x \in D_\sigma, \\ 0 \text{ otherwise.} \end{cases} \tag{1.4.6}$$

The discrete gradient $\nabla_{\mathcal{E}} \phi_{\mathcal{T},\delta t}$ is known to converge only weakly to $\nabla\varphi$ in $L^p(\Omega \times (0,T))$ with $p < +\infty$: see [55, Proof of Lemma 8] for the case $p = 2$ of interest here, and [67, Lemma 4.1] for the general case.

- *Assumptions on the estimates.* Let $(\mathcal{T}_m, \delta t_m)_{m\in\mathbb{N}}$ be a sequence of space-time discretizations of $\Omega\times[0,T)$. Let us denote by $(\rho^{(m)}, p^{(m)}, e^{(m)}, \tilde{\boldsymbol{u}}^{(m)}, \boldsymbol{u}^{(m)})_{m\in\mathbb{N}} = (\rho_{\mathcal{T}_m,\delta t_m}, p_{\mathcal{T}_m,\delta t_m}, e_{\mathcal{T}_m,\delta t_m}, \tilde{\boldsymbol{u}}_{\mathcal{T}_m,\delta t_m}, \boldsymbol{u}_{\mathcal{T}_m,\delta t_m})_{m\in\mathbb{N}}$ the piecewise constant functions reconstructed from the solutions of the scheme (4.2.1)–(1.2.14) for $(\mathcal{T},\delta t) = (\mathcal{T}_m,\delta t_m)$.

  We assume the following estimates on the sequence $(\rho^{(m)}, p^{(m)}, e^{(m)}, \tilde{\boldsymbol{u}}^{(m)}, \boldsymbol{u}^{(m)})_{m\in\mathbb{N}}$:

$$\begin{aligned}&\exists C>0, \forall K\in\mathcal{T}, \text{ for } 0\le n\le N^{(m)},\\ &\left|\frac{1}{(\rho^{(m)})_K^n}\right| + \left|(\rho^{(m)})_K^n\right| + \left|(p^{(m)})_K^n\right| + \left|(e^{(m)})_K^n\right| + \left|(\tilde{\boldsymbol{u}}^{(m)})_K^n\right| + \left|(\boldsymbol{u}^{(m)})_K^n\right| < C \end{aligned} \tag{1.4.7}$$

  In addition we require for all $m\in\mathbb{N}$:

$$|\rho^{(m)}|_{BV_x} + |e^{(m)}|_{BV_x} + |\tilde{\boldsymbol{u}}^{(m)}|_{BV_x} < C \tag{1.4.8}$$

$$|\rho^{(m)}|_{BV_t} + |\boldsymbol{u}^{(m)}|_{BV_t} < C \tag{1.4.9}$$

Theorem 1.4.1 (Consistency of the pressure-correction scheme for the total energy balance)**.** *Let $(\mathcal{T}_m,\delta t_m)$ be a sequence of discretizations such that $h_{\mathcal{T}^{(m)}}$ and $\delta t^{(m)}$ tend to 0 when $m\to\infty$. Let $(\rho^{(m)}, p^{(m)}, \tilde{\boldsymbol{u}}^{(m)}, \boldsymbol{u}^{(m)}, e^{(m)})_{m\in\mathbb{N}}$ be the sequence of piecewise constant functions corresponding to the solution of the scheme (4.2.1)–(1.2.14) for $\mathcal{T},\delta t = (\mathcal{T}_m,\delta t_m)$; we assume that these functions satisfy (1.4.7) and (1.4.8), and that the sequence $(\rho^{(m)}, p^{(m)}, \tilde{\boldsymbol{u}}^{(m)}, \boldsymbol{u}^{(m)}, e^{(m)})_{m\in\mathbb{N}}$ converges in $L^p(\Omega\times(0,T))^{3+2d}$ for $1\le p<\infty$ to a limit $(\bar\rho,\bar p,\bar{\tilde{\boldsymbol{u}}},\bar{\boldsymbol{u}},\bar e)\in L^\infty(\Omega\times(0,T))^{3+2d}$. Then $\bar{\tilde{\boldsymbol{u}}} = \bar{\boldsymbol{u}}$ and $(\bar\rho,\bar p,\bar{\boldsymbol{u}},\bar e)$ is a weak solution of the Euler equations, i.e. it satisfies*

$$\begin{aligned}&\forall\varphi\in C_c^\infty(\Omega\times[0,T),\mathbb{R}),\ \forall\boldsymbol{\varphi}\in\in C_c^\infty(\Omega\times[0,T),\mathbb{R}^d),\\ &\int_0^T\int_\Omega (\bar\rho\partial_t\varphi + \bar\rho\bar{\boldsymbol{u}}\cdot\nabla\varphi)\,\mathrm{d}x\,\mathrm{d}t \ + \int_\Omega \rho_0(x)\varphi(x,0)\,\mathrm{d}x = 0, \end{aligned}\tag{1.4.10}$$

$$\int_0^T\int_\Omega (\bar\rho\bar{\boldsymbol{u}}\cdot\partial_t\boldsymbol{\varphi} + (\bar\rho\bar{\boldsymbol{u}}\otimes\bar{\boldsymbol{u}}):\nabla\boldsymbol{\varphi})\,\mathrm{d}x\,\mathrm{d}t \ + \int_\Omega \rho_0(x)\boldsymbol{u}_0(x)\cdot\boldsymbol{\varphi}(x,0)\,\mathrm{d}x = 0, \tag{1.4.11}$$

$$\int_0^T\int_\Omega \big(\bar\rho\bar E\partial_t\varphi + (\bar\rho\bar E+\bar p)\bar{\boldsymbol{u}}\cdot\nabla\varphi\big)\,\mathrm{d}x\,\mathrm{d}t + \int_\Omega \bar\rho_0(x)\bar E_0(x)\varphi(x,0)\,\mathrm{d}x = 0, \tag{1.4.12}$$

$$\bar E = \bar e + \frac{1}{2}\bar\rho|\bar{\boldsymbol{u}}|^2.$$

Proof. The velocity update (4.2.4) yields the following estimate for all $K\in\mathcal{T}$, $n\in\mathbb{N}^*$

$$\left|\boldsymbol{u}_K^{n+1} - \tilde{\boldsymbol{u}}_K^{n+1}\right| < \delta t\left\|p^{(m)}\right\|_\infty\left\|\frac{1}{\rho^{(m)}}\right\|_\infty.$$

Thanks to the assumptions (1.4.7), we may pass to the limit in the above inequality and obtain: $\bar{\tilde{\boldsymbol{u}}} = \bar{\boldsymbol{u}}$.

We now turn to the process of taking the limit. Let $\varphi\in C_c^\infty(\Omega\times[0,T))$ and $\boldsymbol{\varphi}\in C_c^\infty(\Omega\times[0,T),\mathbb{R}^d)$; for a given discretization $(\mathcal{T}^{(m)},\delta t^{(m)})$, the interpolate of φ and $\boldsymbol{\varphi}$ defined by (1.4.2) are respectively denoted by $\phi^{(m)}$ and $\boldsymbol{\phi}^{(m)}$ (componentwise for $\boldsymbol{\varphi}$). For the sake of simplicity, we shall sometimes drop the index $^{(m)}$ when it is does not hinder comprehension.

Let us first prove that $(\bar\rho, \bar{\boldsymbol{u}})$ satisfies the mass equation (1.4.10). Multiplying (4.2.5) by $\delta t \phi_K^{(m)}$ and summing for $n \in [\![0, N^{(m)}]\!]$ and $K \in \mathcal{T}_m$ yields

$$\underbrace{\sum_{n=0}^{N^{(m)}-1} \sum_{K\in\mathcal{T}} \frac{|K|}{\delta t}\left(\rho_K^{n+1} - \rho_K^n\right)\phi_K^n}_{T_1^{(m)}} + \underbrace{\sum_{n=0}^{N^{(m)}-1} \sum_{K\in\mathcal{T}} \sum_{\sigma\in\mathcal{E}_K} \left(F_{K,\sigma}^n\right)\delta t \phi_K^n}_{T_2^{(m)}} = 0$$

Reordering the summation in $T_1^{(m)}$ yields

$$\begin{aligned} T_1^{(m)} &= -\sum_{n=0}^{N^{(m)}-1} \delta t \sum_{K\in\mathcal{T}} |K| {\rho_K}^n \frac{{\phi_K}^{n+1} - {\phi_K}^n}{\delta t} \\ &= -\int_0^T \int_\Omega \rho^{(m)} \eth_t \phi^{(m)} \,\mathrm{d}x\,\mathrm{d}t \; - \int_\Omega (\rho^{(m)})^0(x)\phi^{(m)}(x,0)\,\mathrm{d}x \\ &\to -\int_0^T \int_\Omega \bar\rho \partial_t \varphi \,\mathrm{d}x\,\mathrm{d}t \; - \int_\Omega \rho_0(x)\varphi(x,0)\,\mathrm{d}x \text{ as } m \to +\infty \end{aligned}$$

thanks to the assumptions on the approximate solutions and the definition of the initial conditions in the algorithm.

Recall that $F_{K,\sigma}^n = |\sigma|\rho_\sigma^n u_{K,\sigma}^n$; therefore reordering the summation in $T_2^{(m)}$ yields

$$\begin{aligned} T_2^{(m)} &= -\sum_{n=0}^{N-1} \delta t \sum_{\sigma=K|L\in\mathcal{E}_{int}} |D_\sigma| \rho_\sigma^n \boldsymbol{u}_\sigma^n \cdot \frac{|\sigma|}{|D_\sigma|}({\phi_L}^n - {\phi_K}^n)\boldsymbol{n}_{K,\sigma} \\ &= -\int_0^T \int_\Omega \rho_{\mathcal{E}}^{(m)} \boldsymbol{u}_{\mathcal{E}}^{(m)} \nabla_{\mathcal{E},\delta t}^{(m)} \phi^{(m)} \,\mathrm{d}x\,\mathrm{d}t \end{aligned}$$

where $\rho_{\mathcal{E}}^{(m)}$ and $\boldsymbol{u}_{\mathcal{E}}^{(m)}$ are piecewise constant functions on the dual mesh $\mathcal{T}^{(m)}$ respectively equal to ρ_σ^n (the upwind choice) and $\boldsymbol{u}_\sigma^n$ (average value defined by (1.2.3) on each dual cell D_σ). Thanks to the assumptions on the approximate solutions, the function $\rho_{\mathcal{E}}^{(m)}$ (resp. $\boldsymbol{u}_{\mathcal{E}}^{(m)}$) converges to $\bar{\boldsymbol{u}}$ (resp. $\bar\rho$) in L^p, $p \in [1, +\infty)$; hence, thanks to the weak convergence of the discrete gradient, we get

$$\int_0^T \int_\Omega \rho^{(m)} \boldsymbol{u}_{\mathcal{E}}^{(m)} \nabla_{\mathcal{E},\delta t}^{(m)} \phi^{(m)} \,\mathrm{d}x\,\mathrm{d}t \to \int_0^T \int_\Omega \bar\rho \bar{\boldsymbol{u}} \cdot \nabla\varphi \,\mathrm{d}x\,\mathrm{d}t \text{ as } m \to +\infty.$$

Let us then prove that $(\bar\rho, \bar p, \bar{\boldsymbol{u}})$ satisfies the weak form of the momentum balance equation (1.4.11). Multiplying (4.2.3) by $\delta t \boldsymbol{\phi}_K^n$ and summing for $n \in [\![0, N^{(m)} - 1]\!]$ and $K \in \mathcal{T}$ yields:

$$\underbrace{\sum_{n=0}^{N^{(m)}-1} \sum_{K\in\mathcal{T}} \frac{|K|}{\delta t}\left(\rho_K^n \tilde{\boldsymbol{u}}^{n+1} - \rho_K^{n-1}\tilde{\boldsymbol{u}}^n\right)\boldsymbol{\phi}_K^n}_{T_1^{(m)}} + \underbrace{\sum_{n=0}^{N^{(m)}-1} \sum_{K\in\mathcal{T}} |K|\,\mathbf{div}_K(\rho^n \boldsymbol{u}^n \otimes \tilde{\boldsymbol{u}}^{n+1})\delta t \cdot \boldsymbol{\phi}_K^n}_{T_2^{(m)}}$$
$$+ \underbrace{\sum_{n=0}^{N^{(m)}-1} \sum_{K\in\mathcal{T}} |K| \widetilde{\nabla_K p}^{\,n} \delta t \boldsymbol{\phi}_K^n}_{T_3^{(m)}} = 0.$$

With arguments that are similar to those used for the term $T_1^{(m)}$ in the mass equation, we get

$$T_1^{(m)} \to -\int_0^T \int_\Omega \bar{\rho}\bar{\boldsymbol{u}} \cdot \partial_t \boldsymbol{\varphi} - \int_\Omega \rho_0(x)\boldsymbol{u}_0(x) \cdot \boldsymbol{\varphi}(x,0)\,\mathrm{d}x.$$

By the definition of the discrete **div** operator and of the numerical flux $F_{K,\sigma}$, reordering the summation in $T_2^{(m)}$, we get:

$$\begin{aligned} T_2^{(m)} &= \sum_{n=0}^{N^{(m)}-1} \delta t \sum_{K\in\mathcal{T}} \sum_{\sigma\in\mathcal{E}_K} F_{K,\sigma}\tilde{\boldsymbol{u}}_\sigma^{n+1} \cdot \boldsymbol{\phi}_K^n \\ &= \sum_{i=1}^{d} \sum_{n=1}^{N^{(m)}-1} \sum_{\sigma=K|L\in\mathcal{E}} |\sigma|\rho_\sigma^{n+1}(\boldsymbol{u}_\sigma^{n+1}\cdot\boldsymbol{n}_{K,\sigma})\delta t(\phi_{K,i}^n - \phi_{L,i}^n)\tilde{u}_{\sigma,i}^{n+1} \end{aligned}$$

where $\phi_{K,i}^n$ (resp. $\tilde{u}_{\sigma,i}^{n+1}$) represents the i-th component of $\boldsymbol{\phi}_K^n$ (resp. $\tilde{\boldsymbol{u}}_\sigma^{n+1}$). Therefore, $T_2^{(m)} = \sum_{i=1}^d (\mathcal{T}_{2,i}^{(m)} + R_{2,i}^{(m)})$, with

$$\mathcal{T}_{2,i}^{(m)} = \sum_{n=1}^{N^{(m)}-1} \delta t \sum_{\sigma=K|L\in\mathcal{E}} \left[|D_{K,\sigma}| \left(\tilde{u}_{K,i}^{n+1}\rho_K^{n+1}\boldsymbol{u}_K^{n+1} \right) + |D_{L,\sigma}| \left(\tilde{u}_{L,i}^{n+1}\rho_L^{n+1}\boldsymbol{u}_L^{n+1} \right) \right] \cdot \frac{|\sigma|}{|D_\sigma|}(\phi_{K,i}^n - \phi_{L,i}^n)\boldsymbol{n}_{K,\sigma}$$

$$\begin{aligned} R_2^{(m)} = \sum_{n=1}^{N^{(m)}-1} \delta t \sum_{\sigma=K|L\in\mathcal{E}} \Big[& |D_{K,\sigma}|(\tilde{u}_{\sigma,i}^{n+1}\rho_\sigma^{n+1}u_{K,\sigma}^{n+1} - u_{K,i}^{n+1}\rho_K^{n+1}\boldsymbol{u}_K^{n+1}\cdot\boldsymbol{n}_{K,\sigma}) \\ & + |D_{L,\sigma}| \left(\tilde{u}_{\sigma,i}^{n+1}\rho_\sigma^{n+1}u_{K,\sigma}^{n+1} - u_{L,i}^{n+1}\rho_L^{n+1}\boldsymbol{u}_L^{n+1}\cdot\boldsymbol{n}_{K,\sigma} \right) \Big] (\phi_{K,i}^n - \phi_{L,i}^n). \end{aligned}$$

We may then rewrite $\mathcal{T}_{2,i}^{(m)}$ as

$$\mathcal{T}_{2,i}^{(m)} = \int_0^T \int_\Omega \tilde{u}_i^{(m)}\rho^{(m)}\boldsymbol{u}^{(m)} \cdot \nabla_{\mathcal{E}^{(m)}}\phi_i^{(m)}\; dx,$$

where $\nabla_{\mathcal{E}^{(m)}}\varphi^{(m)}$ is the weak gradient defined by (1.2.4) which converges weakly to $\nabla\varphi_i$, as previously stated. Thanks to the assumptions of strong convergence in any L^p, $1 \le p < +\infty$ of the approximate solutions $e^{(m)}$, $\rho^{(m)}$ and $\boldsymbol{u}^{(m)}$, and to the fact that $\tilde{\boldsymbol{u}}^{(m)}$ also converges to $\bar{\boldsymbol{u}}$, we get

$$\mathcal{T}_2^{(m)} \to \sum_{i=1}^d \int_0^T \int_\Omega \bar{u}_i\bar{\rho}\bar{\boldsymbol{u}} \cdot \nabla\varphi_i \;\mathrm{d}\boldsymbol{x} = \int_0^T \int_\Omega (\bar{\rho}\bar{\boldsymbol{u}} \otimes \bar{\boldsymbol{u}}) : \nabla\boldsymbol{\varphi}\,\mathrm{d}\boldsymbol{x} \text{ as } m \to +\infty.$$

It now remains to be proved that the remainder term $R_2^{(m)}$ vanishes when $m \to \infty$. For the sake of clarity, we drop the exponents related to the discrete time indexation. Thanks to the L^∞ and BV estimates, we have (dropping the time indexes for short)

$$\begin{aligned} |Y_{K,\sigma}| &:= |\tilde{u}_{\sigma,i}\rho_\sigma u_{K,\sigma} - u_{K,i}\rho_K\boldsymbol{u}_K\cdot\boldsymbol{n}_{K,\sigma}| \\ &= |\tilde{u}_{\sigma,i}\rho_\sigma(\boldsymbol{u}_\sigma - \boldsymbol{u}_K)\cdot\boldsymbol{n}_{K,\sigma}| + |\tilde{u}_{\sigma,i}(\rho_\sigma - \rho_K)\boldsymbol{u}_K| + |\rho_K(\tilde{u}_{\sigma,i} - \tilde{u}_{K,i}\boldsymbol{u}_K)\cdot\boldsymbol{n}_{K,\sigma})| \\ &\le \|\tilde{u}_i\|_\infty \left(\|\rho\|_\infty |\boldsymbol{u}|_{BV_x} + \|\boldsymbol{u}\|_\infty |\rho|_{BV_x} \right) + |\tilde{u}_i|_{BV_x}\|\boldsymbol{u}\|_\infty\|\rho\|_\infty \end{aligned}$$

and therefore $R_2^{(m)} \to 0$ as $m \to +\infty$.

Using the definition (1.2.4) of the discrete pressure gradient and Lemma 1.4.2 given below (see [57, Proposition 2.1], [58, Proof of Lemma 5.7] for similar results), we get that

$$T_3^{(m)} \to -\int_0^T \int_\Omega \bar{p}\,\mathrm{div}\,\boldsymbol{\varphi}\;\mathrm{d}\boldsymbol{x}\;\mathrm{d}t \text{ as } m \to +\infty.$$

Let us finally prove that the limit $(\bar{\rho}, \bar{p}, \bar{\boldsymbol{u}})$ satisfies a weak form of the total energy balance (1.4.12). Let $\varphi \in C_c^\infty(\Omega \times [0,T))$; for a given discretization $(\mathcal{T}^{(m)}, \delta t^{(m)})$, we denote by $\phi^{(m)}$ the interpolate of φ as defined by (1.4.2). We momentarily drop for short the index (m), and denote by $\phi_K^n = \varphi(x_K, t_n)$, for $K \in \mathcal{T}$ and $n \in \{0, \ldots, N\}$. We assume that the mesh is fine enough so that $\phi_K^n = 0$ if K is a boundary cell. Multiplying the local total energy balance (1.3.9) by $\delta t \phi_K^n$ and summing over the mesh cells and the time steps, we get (still dropping the index (m) in the summations):

$$\underbrace{\sum_{n=0}^{N^{(m)}-1} \sum_{K\in\mathcal{T}} \frac{|K|}{\delta t}\left((\widetilde{\rho_K E_K})^{n+1} - (\widetilde{\rho_K E_K})^n\right)\delta t \phi_K^n}_{T_1^{(m)}}$$
$$+ \underbrace{\sum_{n=0}^{N^{(m)}-1} \sum_{K\in\mathcal{T}} \sum_{\sigma\in\mathcal{E}_K} \left(|\sigma| e_\sigma^{n+1} \rho_\sigma^{n+1} u_{K,\sigma}^{n+1}\right)\delta t \phi_K^n}_{T_2^{(m)}} + \underbrace{\sum_{n=0}^{N^{(m)}-1} \sum_{K\in\mathcal{T}} |K| \widetilde{\mathrm{div}}_K(p\boldsymbol{u})\phi_K^n}_{T_3^{(m)}}$$
$$+ \underbrace{\frac{1}{2}\sum_{n=0}^{N^{(m)}-1} \sum_{K\in\mathcal{T}} \sum_{\sigma\in\mathcal{E}_K} \left(|\sigma| \tilde{\boldsymbol{u}}_K^{n+1} \cdot \tilde{\boldsymbol{u}}_L^{n+1} \rho_\sigma^n u_{K,\sigma}^n\right)\delta t \phi_K^n}_{T_4^{(m)}} + \underbrace{\sum_{n=0}^{N^{(m)}-1} \sum_{K\in\mathcal{T}} P_K^n \delta t \phi_K^n}_{T_5^{((m)}} = 0$$

where $(\widetilde{\rho_K E_K})^n$ is defined in Proposition 1.3.5. The time-dependent term can be rewritten as:

$$\begin{aligned} T_1^{(m)} &= -\sum_{K\in\mathcal{T}} |K| (\widetilde{\rho_K E_K})^0 \phi_K^0 - \sum_{n=1}^{N-1} \sum_{K\in\mathcal{T}} |K| (\widetilde{\rho_K E_K})^n \delta t \left(\frac{\phi_K^n - \phi_K^{n-1}}{\delta t}\right) \\ &= -\int_\Omega (\widetilde{\rho E})^{(m)}(x,0)\phi^{(m)}(x,0)\,\mathrm{d}x - \int_0^T \int_\Omega (\widetilde{\rho E})^{(m)}(x,t) \eth_t \phi^{(m)}(x,t)\,\mathrm{d}x\,\mathrm{d}t \end{aligned}$$

where $(\widetilde{\rho E})^{(m)}$ is the space-time piecewise constant (with respect to the discretization $(\mathcal{T}^{(m)}, \delta t^{(m)})$) function defined by (1.3.10). Thanks to the regularity of φ, $\eth_t \phi^{(m)}$ tends to $\partial_t \varphi$ uniformly. Thanks to the strong convergence of the discrete solution and to assumptions (1.4.7)–(1.4.8), we thus get that for all $\varphi \in \mathcal{C}_c^\infty(\Omega \times [0,T])$,

$$\lim_{m\to\infty} T_1^{(m)} = -\int_0^T \int_\Omega \rho E(x,t)\;\partial_t \varphi(x,t)\;\mathrm{d}x\,\mathrm{d}t - \int_\Omega \rho_0(x) E_0(x) \varphi(x,0)\,\mathrm{d}x.$$

Reordering the summation in $T_2^{(m)}$, we get:

$$\begin{aligned} T_2^{(m)} &= \sum_{n=1}^{N^{(m)}-1} \sum_{\sigma=K|L\in\mathcal{E}} |\sigma| e_\sigma^{n+1} \rho_\sigma^{n+1} (\boldsymbol{u}_\sigma^{n+1} \cdot \mathbf{n}_{K,\sigma}) \delta t (\phi_K^n - \varphi_L^n) \\ &= \mathcal{T}_2^{(m)} + R_2^{(m)} \end{aligned}$$

with

$$\mathcal{T}_2^{(m)} = \sum_{n=1}^{N^{(m)}-1} \delta t \sum_{\sigma=K|L\in\mathcal{E}} \left[|D_{K,\sigma}| \left(e_K^{n+1}\rho_K^{n+1}\boldsymbol{u}_K^{n+1} \right) + |D_{L,\sigma}| \left(e_L^{n+1}\rho_L^{n+1}\boldsymbol{u}_L^{n+1} \right) \right] \cdot \frac{|\sigma|}{|D_\sigma|}(\phi_K^n - \varphi_L^n)\boldsymbol{n}_{K,\sigma}$$

$$R_2^{(m)} = \sum_{n=1}^{N^{(m)}-1} \delta t \sum_{\sigma=K|L\in\mathcal{E}} \Big[|D_{K,\sigma}|(e_\sigma^{n+1}\rho_\sigma^{n+1}u_{K,\sigma}^{n+1} - e_K^{n+1}\rho_K^{n+1}\boldsymbol{u}_K^{n+1}\cdot\boldsymbol{n}_{K,\sigma}) + |D_{L,\sigma}| \left(e_\sigma^{n+1}\rho_\sigma^{n+1}u_{K,\sigma}^{n+1} - e_L^{n+1}\rho_L^{n+1}\boldsymbol{u}_L^{n+1}\cdot\boldsymbol{n}_{K,\sigma} \right) \Big] (\phi_K^n - \varphi_L^n).$$

We may then rewrite $\mathcal{T}_2^{(m)}$ as

$$\mathcal{T}_2^{(m)} = \int_0^T \int_\Omega e^{(m)}\rho^{(m)}\boldsymbol{u}^{(m)} \cdot \nabla_{\mathcal{E}^{(m)}}\varphi^{(m)} \, dx,$$

where $\varphi^{(m)}$ is a piecewise constant function on the cells of the mesh $\mathcal{T}^{(m)}$ and $\nabla_{\mathcal{E}^{(m)}}\varphi^{(m)}$ is the weak gradient defined by (1.2.4), which converges weakly to $\nabla\varphi$ as previously stated. Thanks to the assumptions of strong convergence in any L^p, $1 \le p < +\infty$ of the approximate solutions $e^{(m)}$, $\rho^{(m)}$ and $\boldsymbol{u}^{(m)}$, we get

$$\mathcal{T}_2^{(m)} \to \int_0^T \int_\Omega \bar{e}\bar{\rho}\bar{\boldsymbol{u}} \cdot \nabla\varphi \, dx \text{ as } m \to +\infty.$$

With similar arguments as in the case of its namesake in the momentum balance equation, it is easily seen that the remainder term $R_2^{(m)}$ vanishes when $m \to \infty$. Using the definition (1.2.6) and reordering the summation, we may write the term $T_3^{(m)}$ as

$$\begin{aligned} T_3^{(m)} &= \sum_{n=1}^{N-1} \delta t \sum_{K\in\mathcal{T}} \sum_{L\in\mathcal{N}(K)} |\sigma_{KL}|(\widetilde{p\boldsymbol{u}})_\sigma \phi_K^n \\ &= \sum_{n=1}^{N-1} \delta t \sum_{\substack{\sigma\in\mathcal{E}_{int} \\ \sigma=K|L}} |D_\sigma|(\widetilde{p\boldsymbol{u}})_\sigma \frac{|\sigma|}{|D_\sigma|}(\phi_K^n - \varphi_L^n) \\ &= -\int_0^T \int_\Omega (\widetilde{p\boldsymbol{u}})^{(m)} \widetilde{\nabla}_{\mathcal{E}^{(m)}}\phi^{(m)} \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\boldsymbol{x}, \end{aligned}$$

where $(\widetilde{p\boldsymbol{u}})^{(m)}$ is the piecewise constant function on the dual mesh (with respect to the discretization $(\mathcal{T}^{(m)}, \delta t^{(m)})$) defined by:

$$(\widetilde{p\boldsymbol{u}})^{(m)}(x,t) = (\widetilde{p^n\boldsymbol{u}^n})_\sigma \text{ for } x \in K \text{ and } t \in [t_n, t_{n+1}),$$

with $(\widetilde{p\boldsymbol{u}})_\sigma$ defined by (1.2.6) and $\widetilde{\nabla}_{\mathcal{E}^{(m)}}\phi^{(m)}$ the weakly converging discrete gradient defined by (1.4.5). Thanks to assumptions (1.4.7–1.4.8), we have $(\widetilde{p\boldsymbol{u}})_{\mathcal{T}}^n \to p\boldsymbol{u}$ in $L^2(\Omega\times(0,T))$ as $h^{(m)}$ and $\delta t^{(m)}$ tend to 0. By [55, Proof of Lemma], we have $\widetilde{\nabla}_{\mathcal{T}^{(m)}}\varphi^{(m)} \to \nabla\varphi$ weakly in $L^2(\Omega)$ as $h^{(m)}$ and $\delta t^{(m)}$ tend to 0. Hence,

$$T_3^{(m)} \to -\int_0^T \int_\Omega p\boldsymbol{u}\varphi \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\boldsymbol{x} \text{ as } h \text{ and } \delta t \to 0.$$

As for the second advection term, we can rewrite $T_4^{(m)}$ as:

$$T_4^{(m)} = \sum_{n=1}^{N^{(m)}-1} \sum_{\sigma=K|L\in\mathcal{E}} \delta t |D_\sigma| \tilde{\boldsymbol{u}}_{D_\sigma}^{n+1} \cdot \tilde{\boldsymbol{u}}_{D_\sigma}^{n+1} \rho_{D_\sigma}^n \boldsymbol{u}_{D_\sigma}^n \left(\frac{|\sigma|}{|D_\sigma|} (\phi_K^n - \phi_L^n) \mathbf{n}_{K,\sigma} \right) + R_4^{(m)}$$

with the residual:

$$R_4^n = \sum_{n=1}^{N^{(m)}-1} \sum_{\sigma=K|L\in\mathcal{E}} \big[|D_{K,\sigma}| \tilde{\boldsymbol{u}}_K^{n+1} \cdot (\tilde{\boldsymbol{u}}_L^{n+1} \rho_\sigma^n \boldsymbol{u}_\sigma^n - \tilde{\boldsymbol{u}}_K^{n+1} \rho_K^n \boldsymbol{u}_K^n) + |D_{L,\sigma}| \tilde{\boldsymbol{u}}_L^{n+1} \cdot (\tilde{\boldsymbol{u}}_K^{n+1} \rho_\sigma^n \boldsymbol{u}_\sigma^n - \tilde{\boldsymbol{u}}_L^{n+1} \rho_L^n \boldsymbol{u}_L^n) \big] \cdot \delta t \frac{\phi_L^n - \phi_K^n}{h_\sigma} \mathbf{n}_{K|L}.$$

Proceeding in the same way as for the residual $R_2^{(m)}$ with the term $(\tilde{\boldsymbol{u}}_L \rho_\sigma \boldsymbol{u}_\sigma - \tilde{\boldsymbol{u}}_K \rho_K \boldsymbol{u}_K)$:

$$\begin{aligned} \tilde{\boldsymbol{u}}_L \rho_\sigma \boldsymbol{u}_\sigma - \tilde{\boldsymbol{u}}_K \rho_K \boldsymbol{u}_K = {} & (\varepsilon_{K,\sigma}^2 - 1)(\tilde{\boldsymbol{u}}_L + \tilde{\boldsymbol{u}}_K) \left[(\rho_K - \rho_L)(\boldsymbol{u}_K + \boldsymbol{u}_L) + (\rho_K + \rho_L)(\boldsymbol{u}_K - \boldsymbol{u}_L) \right] \\ & + (\tilde{\boldsymbol{u}}_L - \tilde{\boldsymbol{u}}_K) \left[(\varepsilon_{K,\sigma} + 1)^2 (\rho_K + \rho_L)(\boldsymbol{u}_K + \boldsymbol{u}_L) + (\varepsilon_{K,\sigma} - 1)^2 (\rho_K - \rho_L)(\boldsymbol{u}_K - \boldsymbol{u}_L) \right] \end{aligned}$$

A similar development can be made for the term $(\tilde{\boldsymbol{u}}_K \rho_\sigma \boldsymbol{u}_\sigma - \tilde{\boldsymbol{u}}_L \rho_K \boldsymbol{u}_K)$ which yields eventually to:

$$|R_4^n| \leq C_\varphi (h_\mathcal{T})^d \left[|\tilde{\boldsymbol{u}}|_\infty (|\rho|_{BV_x} |\boldsymbol{u}|_\infty + |\rho|_\infty |\boldsymbol{u}|_{BV_x}) + |\tilde{\boldsymbol{u}}|_{BV_x} (|\rho|_\infty |\boldsymbol{u}|_{BV_x} + |\rho|_{BV_x} |\boldsymbol{u}|_\infty) \right]$$

Using the same arguments as for the term T_2, we have:

$$\begin{aligned} \lim_{m\to\infty} T_4^{(m)} &= \lim_{m\to\infty} \int_0^T \int_\Omega \tilde{\boldsymbol{u}}^{(m)} \cdot \tilde{\boldsymbol{u}}^{(m)} \rho^{(m)} \boldsymbol{u}^{(m)} \nabla_{\mathcal{E}^{(m)}} \phi_{\mathcal{T}^{(m)},\delta t^{(m)}} \, \mathrm{d}x \, \mathrm{d}t \\ &= \int_0^T \int_\Omega \bar{\boldsymbol{u}} \cdot \bar{\boldsymbol{u}} \bar{\rho} \boldsymbol{u} \cdot \nabla \varphi \, \mathrm{d}x \, \mathrm{d}t \end{aligned}$$

The last term $T_5^{(m)}$ tied to the pressure residual P_K^n of the discrete kinetic energy balance vanishes as the time step and the space discretization step tend to 0, thanks to the L^∞ estimate on $\rho^{(m)}$ and $\frac{1}{\rho^{(m)}}$, and the time.

□

Lemma 1.4.2 (Weak consistency of the pressure gradient)**.** *For a given polygonal mesh $\mathcal{T}$ of Ω, consider a scalar field $q = (q_K)_{K\in\mathcal{T}}$ and let $G_\mathcal{T} q$ be a piecewise constant discrete gradient defined by*

$$G_\mathcal{T} q(\boldsymbol{x}) = G_K q, \ \forall \boldsymbol{x} \in K, \ \text{with } G_K q = \sum_{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}} |\sigma| (q_\sigma - q_K) \boldsymbol{n}_{K,\sigma}.$$

where $(q_\sigma)_{\sigma\in\mathcal{E}_{int}}$ is a family of values such that if $\sigma = K|L$, then q_σ is a convex combination of q_K and q_L. Let $\mathcal{T}_m$ be a sequence of meshes such that $h_{\mathcal{T}_m} \to 0$ as $m \to 0$, and let $q^{(m)} = (q_K^{(m)})_{K\in\mathcal{T}_m}$ be a family of corresponding discrete scalar fields. We assume that

1. *$q^{(m)} \to \bar{q}$ in $L^1_{\mathrm{loc}}(\Omega)$ as $m \to +\infty$,*
2. *there exists $C \in \mathbb{R}_+$ such that $|u^{(m)}|_{BV_x} \leq C$ for any $m \in \mathbb{N}$.*

Let $\varphi \in \mathcal{C}_c^\infty(\Omega)$ be a given test function; denote by $\phi^{(m)}$ its interpolate on the mesh $\mathcal{T}_m$, defined by:

$$\phi^{(m)}(\boldsymbol{x}) = \varphi(\boldsymbol{x}_K) \text{ for } \boldsymbol{x} \in K.$$

Then

$$\int_\Omega G_{\mathcal{T}^{(m)}} q^{(m)}(\boldsymbol{x})\phi^{(m)}(\boldsymbol{x})d\boldsymbol{x} \to -\int_\Omega \bar{q}(\boldsymbol{x})\nabla\varphi(\boldsymbol{x})d\boldsymbol{x} \text{ as } m \to +\infty.$$

Proof. We have

$$\begin{aligned}\int_\Omega G_{\mathcal{T}^{(m)}} q^{(m)}(\boldsymbol{x})\phi^{(m)}(\boldsymbol{x})d\boldsymbol{x} &= \sum_{K\in\mathcal{T}}\sum_{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}} |\sigma|(q_\sigma - q_K)\boldsymbol{n}_{K,\sigma}\phi_K^{(m)} \\ &= T + R,\end{aligned}$$

with

$$\begin{aligned}T^{(m)} &= \sum_{K\in\mathcal{T}}\sum_{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}} |\sigma|(q_\sigma - q_K)\boldsymbol{n}_{K,\sigma}\phi_\sigma^{(m)}, \text{ where } \phi_\sigma^{(m)} = \frac{1}{|\sigma|}\int_\sigma \varphi(\boldsymbol{x})d\gamma(\boldsymbol{x}) \\ R^{(m)} &= \sum_{K\in\mathcal{T}}\sum_{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}} |\sigma|(q_\sigma - q_K)\boldsymbol{n}_{K,\sigma}(\phi_K^{(m)} - \phi_\sigma^{(m)})\end{aligned}$$

We have

$$T^{(m)} = \sum_{\sigma\in\mathcal{E}_{int}} |\sigma|(q_L - q_K)\boldsymbol{n}_{K,\sigma}\varphi_\sigma = -\sum_{K\in\mathcal{T}}\sum_{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}} |\sigma|(\phi_\sigma)\boldsymbol{n}_{K,\sigma}q_K = -\sum_{K\in\mathcal{T}} q_K\int_K \nabla\varphi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x},$$

so that thanks to Assumption (1), $T^{(m)} \to \int_\Omega q(\boldsymbol{x})\nabla\varphi d\boldsymbol{x}$ as $m \to +\infty$. Now

$$R^{(m)} \le \sum_{\sigma\in\mathcal{E}_{int}} |\sigma||q_L - q_K||\boldsymbol{n}_{K,\sigma}||\phi_K - \phi_\sigma| \le C_\varphi h_m \sum_{\sigma\in\mathcal{E}_{int}} |\sigma||q_L - q_K| \le C_\varphi h_m |q|_{BV_x},$$

so that, thanks to Assumption (2), $R^{(m)} \to 0$ as $m \to +\infty$, □

Chapter 2

APPLICATION TO SHOCK HYDRODYNAMICS

In our numerical experiments, our pressure-correction scheme is compared with another fractional step scheme for compressible flows (compressible Euler equations, $\mu = 0$), the SLK scheme. This scheme was originally introduced in 2004 [102] and it is available in the Code_Saturne *CFD code developed at EDF R&D.*

2.1 The SLK scheme

In our numerical experiments, our pressure-correction scheme is compared with another fractional step scheme for compressible flows, the SLK scheme. This scheme was originally introduced in 2004 [102] and it is available in the *Code_Saturne* CFD code developed at EDF R&D. Among the differences between our scheme and the SLK scheme, it should be stressed first that the latter is derived from the compressible Euler equations written with the *total energy balance* (1.1.1). Furthermore thanks to a decomposition of the variation of the pressure into the variations of the density and of the entropy (2.1.3), the *projection-correction* step (here taking the form of an "acoustic step") is linear.

2.1.1 Time discretization

Among the several variants of this algorithm we will consider the following semi-discrete algorithm [12], which reads for all $n \in \mathbb{N}$

Acoustic step (compute $\rho^{n+1}, \mathbf{q}_{ac}^{n+1}$)

$$\frac{\rho^{n+1} - \rho^n}{\delta t} + \operatorname{div} \mathbf{q}^n - \operatorname{div}(\delta t (c^2)^n \nabla \rho^{n+1}) - \operatorname{div}(\delta t \beta^n \nabla s^n) = 0 \tag{2.1.1a}$$

$$\mathbf{q}_{ac}^{n+1} = \mathbf{q}^n - \delta t (c^2)^n \nabla \rho^{n+1} - \delta t \beta^n \nabla s^n \tag{2.1.1b}$$

Momentum and total energy steps (compute $\boldsymbol{u}^{n+1}, E^{n+1}$)

$$\rho^n \frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n}{\delta t} - \boldsymbol{u}^{n+1} \operatorname{div} \mathbf{q}_{ac}^{n+1} + \mathbf{div}(\boldsymbol{u}^{n+1} \otimes \mathbf{q}_{ac}^{n+1}) + \nabla p^n = 0 \tag{2.1.2a}$$

$$\rho^n \frac{E^{n+1} - E^n}{\delta t} - E^{n+1} \operatorname{div} \mathbf{q}_{ac}^{n+1} + \operatorname{div} \left[\left(E^{n+1} + \frac{p^n}{\rho^{n+1}} \right) \mathbf{q}_{ac}^{n+1} \right] = 0 \tag{2.1.2b}$$

Pressure update (compute p^{n+1})

$$p^{n+1} = (\gamma - 1) \rho^{n+1} \left(E^{n+1} - \frac{1}{2} \boldsymbol{u}^{n+1} \cdot \boldsymbol{u}^{n+1} \right)$$

with $\mathbf{q} = \rho \boldsymbol{u}$ the mass flux, $\mathbf{q}_{\rm ac}$ the *acoustic mass flux*, s the entropy, c the speed of the sound and $\beta = \rho^\gamma$. The derivation of the *acoustic step* is given in [102]. The variation of the pressure is decomposed into the variation of the density and of the entropy:

$$\nabla p = c^2 \nabla \rho + \beta \nabla s \tag{2.1.3}$$

During the acoustic step — and the acoustic step only — an isentropic flow is considered. As a result the classical acoustic system can be written:

$$\begin{cases} \partial_t \rho + \operatorname{div} \mathbf{q} = 0 \\ \partial_t \mathbf{q} + \nabla p = 0 \end{cases}$$

with semi-discretization:

$$\frac{\rho^{n+1} - \rho^n}{\delta t} + \operatorname{div} \mathbf{q}_{ac}^{n+1} = 0 \tag{2.1.4}$$

$$\frac{\mathbf{q}_{ac}^{n+1} - \mathbf{q}_{ac}^n}{\delta t} + \nabla p^{n+1} = 0 \tag{2.1.5}$$

Combining the above equations yields:

$$\operatorname{div}\left((c^2)^n \nabla \rho^{n+1}\right) = \frac{\operatorname{div} \mathbf{q}_{ac}^n + \frac{\rho^{n+1} - \rho^n}{\delta t}}{\delta t}$$

If $\mathbf{q}_{ac}^n$ is replaced by its more up-to-date value $\mathbf{q}^n$ then we recover step (2.1.1).

Remark 2.1.1. Assuming a homogeneous sound speed and using the mass balance (2.1.4) from the previous time step we can obtain the wave equation at the discrete level:

$$\frac{\rho^{n+1} - 2\rho^n + \rho^{n-1}}{\delta t^2} - (c^2)^n \Delta \rho^{n+1} = 0$$

Remark 2.1.2. In some respect the SLK algorithm may be viewed as a projection scheme. Using equations (2.1.4) and (2.1.5), we have:

$$\mathbf{q}_{ac}^{n+1} = \mathbf{q}_{ac}^n - \delta t \nabla p^{n+1} \tag{2.1.6a}$$

$$\Delta p^{n+1} = \frac{\operatorname{div} \mathbf{q}_{ac}^n + \frac{\rho^{n+1} - \rho^n}{\delta t}}{\delta t} \tag{2.1.6b}$$

The first equation is close to the Hodge decomposition for the mass flux which is at the basis of incompressible projection methods. The second equation is similar to a *projection* step, which would be aimed at enforcing the mass conservation constraint on the mass flux. Likewise step (2.1.1) can be thought of as a projection of the last predicted mass flux $\boldsymbol{q}^n$ on the space of the mass fluxes satisfying the discrete mass balance (2.1.4). Using the density instead of the pressure in the acoustic step allows to control the positivity of the density through the numerical scheme.

2.1.2 Space discretization

We give the space-time discretization of the SLK scheme for cell-centered finite-volumes that we actually used in the numerical tests:

Initialization

$$\forall K \in \mathcal{T}, \quad \rho_K^0,\ \boldsymbol{u}_K^0,\ p_K^0 \text{ given} \quad ; \quad \rho^{-1} = \rho^0 \quad ; \quad E_K^0 = p_K^0/(\gamma - 1)\rho_K^0 - \frac{1}{2}\boldsymbol{u}_K^0 \cdot \boldsymbol{u}_K^0$$

Iterations for $n = 0, 1, \ldots, N-1$:

1. *Update passive scalars* for all $K \in \mathcal{T}$

$$\begin{aligned} F^n_{K,\sigma} &= \rho^n_\sigma \boldsymbol{u}^n_\sigma \cdot \mathbf{n}_{K,\sigma} \quad \forall \sigma \in \mathcal{E}_K \\ s^n_K &= p^n_K/(\rho^n_K)^\gamma \\ (c^2)^n_K &= \gamma p^n_K/\rho^n_K \\ \beta^n_K &= (\rho^n_K)^\gamma \end{aligned}$$

2. *Predict the density (ρ^{n+1}_K)* solve for all $K \in \mathcal{T}$

$$\frac{|K|}{\delta t}(\rho^{n+1}_K - \rho^n_K) + \sum_{\sigma\in\mathcal{E}_K} |\sigma| F^n_{K,\sigma} - \sum_{\sigma\in\mathcal{E}_K} |\sigma|\delta t (c^2)^n_\sigma (\partial_n \rho)^{n+1}_\sigma - \sum_{\sigma\in\mathcal{E}_K} |\sigma|\delta t \beta^n_\sigma (\partial_n s)^n_\sigma = 0$$

3. *Update the acoustic mass flux ($\tilde{F}^{n+1}_{K,\sigma}$)* for all $K \in \mathcal{T},\ \sigma \in \mathcal{E}_K$

$$\tilde{F}^{n+1}_{K,\sigma} = F^n_{K,\sigma} - \delta t (c^2)^n_\sigma (\partial_n \rho)^{n+1}_\sigma - \delta t \beta^n_\sigma (\partial_n s)^n_\sigma$$

4. *Compute the velocity ($\boldsymbol{u}^{n+1}_K$)* solve for all $K \in \mathcal{T}$

$$\frac{|K|}{\delta t}\rho^n_K(\boldsymbol{u}^{n+1}_K - \boldsymbol{u}^n_K) - \boldsymbol{u}^{n+1}_K \sum_{\sigma\in\mathcal{E}_K} |\sigma| \tilde{F}^{n+1}_{K,\sigma} + \sum_{\sigma\in\mathcal{E}_K} |\sigma| \boldsymbol{u}^{n+1}_\sigma \tilde{F}^{n+1}_{K,\sigma} + \nabla_K p^n = 0$$

5. *Compute the total energy (E^{n+1}_K)* solve for all $K \in \mathcal{T}$

$$\frac{|K|}{\delta t}\rho^n_K(E^{n+1}_K - E^n_K) - E^{n+1}_K \sum_{\sigma\in\mathcal{E}_K} |\sigma| \tilde{F}^{n+1}_{K,\sigma} + \sum_{\sigma\in\mathcal{E}_K} |\sigma| \left(E^{n+1}_\sigma + \frac{p^n_\sigma}{\rho^{n+1}_\sigma} \right) \tilde{F}^{n+1}_{K,\sigma} = 0$$

6. *Update the pressure (p^{n+1}_K)* for all $K \in \mathcal{T}$

$$p^{n+1}_K = (\gamma - 1)\rho^{n+1}_K \left(E^{n+1}_K - \frac{1}{2}\boldsymbol{u}^{n+1}_K \cdot \boldsymbol{u}^{n+1}_K \right)$$

$\tilde{F}^{n+1}_{K,\sigma}$ stands for the normal component of the discrete acoustic mass flux $\boldsymbol{q}_{\rm ac}$. The edge pressure p_σ of the discrete pressure gradient and the edge velocity $\mathbf{u}_\sigma$ are centered. The fluxes $\tilde{F}^{n+1}$ and p_σ are discretized with a centered scheme. All the advected fluxes are discretized with an upwind scheme with respect to the acoustic flux $\tilde{F}^{n+1}_{K,\sigma}$. The edge values of passive scalars β and c^2 are computed with a harmonic scheme. For instance for an edge $\sigma = K|L$

$$\beta_\sigma = \frac{2\beta_K \beta_L}{\beta_K + \beta_L}$$

Lastly the normal gradients denoted "∂_n" (for example for the density at an edge $\sigma = K|L$) are discretized as:

$$(\partial_n \rho)_\sigma = \frac{\rho_L - \rho_K}{h_\sigma}$$

2.2 Numerical results

In a first section, we check the accuracy of our scheme and of the SLK [102] scheme on one-dimensional Riemann problems. We observe an oscillatory behaviour near shocks though it does not affect the L^1 convergence of the error; it can be cured with an artificial viscosity method such as [90]. In a second section several two-dimensional Riemann problems are performed. The oscillatory behaviour only needs to be dealt with for strong shock problems. In all our tests, the non-linear system for the projection-correction step is solved with a simple fixed-point procedure. The sub-iterations are initialized with the most up-to-date flow variables. During a sub-iteration we successively update the velocity with (4.2.4), solve the density with (4.2.5), update the source term with (1.3.8), solve the internal energy with (4.2.6) and finally update the pressure with the equation of state. The stopping criteria for these sub-iterations is defined with respect to the relative $L^\infty(\Omega; BV(\delta t^*))$ norm of each flow variables for a sub-iteration over δt^*:

$$\max\left(\frac{|\boldsymbol{u}|_{\infty,t,BV}}{|\boldsymbol{u}|_\infty}, \frac{|\rho|_{\infty,t,BV}}{|\rho|_\infty}, \frac{|e|_{\infty,t,BV}}{|e|_\infty}, \frac{|p|_{\infty,t,BV}}{|p|_\infty}\right) < \varepsilon$$

With the choice $\varepsilon = 10^{-3}$, the number of sub-iterations in our numerical tests is always lower than 5.

2.2.1 One dimensional problems

Accuracy tests

The accuracy of our scheme and of the SLK scheme is evaluated on four classical 1D Riemann problems described in detail in [126] and for which an analytical solution is available. Test 1 is a Sod shock tube ; test 3 has a left rarefaction, a contact discontinuity and a right shock ; test 4 has a left shock, a contact discontinuity and a right rarefaction ; test 5 has a two strong shocks and a contact discontinuity. The domain is $\Omega = [-4, 4]$ with Dirichet boundary conditions on $\partial\Omega$. The tests are carried out on 6 grids with 2^m cells, $10 \leq m \leq 15$. The initial states are given in table 2.1 and the convergence orders for our scheme and for the SLK scheme in table 2.2. The final state for test 5 is shown in figure 2.1.

Test	$\rho_{\mathbf{L}}$	$\mathbf{u_L}$	$\mathbf{p_L}$	$\rho_{\mathbf{R}}$	$\mathbf{u_R}$	$\mathbf{p_R}$	**T**	$\delta\mathbf{t}$
1	1.0	0.0	1.0	0.125	0.0	0.1	0.25	$h/2$
3	1.0	0.0	1000.0	1.0	0.0	0.01	0.012	$h/40$
4	1.0	0.0	0.01	1.0	0.0	100.0	0.035	$h/14$
5	5.99924	19.5975	460.894	5.99242	-6.19633	46.0950	0.035	$h/40$

TAB. 2.1 – *1D Riemman problems used for accuracy tests.*

In all tests the order of convergence in L^1 norm is between 0.5 and 1. With both schemes, while shocks are sharply computed contact discontinuities are inaccurately calculated, which is the main source of error in all the tests. SLK is often slightly more accurate than our scheme but convergence orders are close. Both schemes suffer from oscillations near shocks. Nevertheless although the magnitude of the oscillations does not decrease with the mesh size, the measure of these oscillations vanishes. Therefore they do not have a significant impact on

the L^1 convergence of error. The issue of the oscillatory behaviour is addressed in the next subsection.

Variable	**Test 1**	**Test 3**	**Test 4**	**Test 5**
ρ (our scheme)	0.652	0.536	0.525	0.529
ρ (SLK)	0.640	0.514	0.498	0.568
p (our scheme)	0.834	0.836	0.824	0.974
p (SLK)	0.827	0.858	0.824	1.008
$\boldsymbol{u}$ (our scheme)	0.884	0.860	0.836	0.981
$\boldsymbol{u}$ (SLK)	0.847	0.872	0.837	1.009

TAB. 2.2 – *Convergence orders (L_1 norm) for 1D Riemman problems with our scheme and with the SLK scheme.*

Remark 2.2.1 (Importance of the term S_K). As outlined in the previous sections, the most critical component of our method lies in the discrete source term S_K added to the internal energy balance. As observed in figure 2.2 the source term is localized at flow discontinuities and its measure does not vanish when the space discretization step tend to zero. The density field at the final time for test 5 is given for our scheme with S_K (in blue) and without S_K (in green). The two profiles are similar, however the shocks speeds are different so are the intermediate states. Without a reference solution, it would not be obvious to assess which numerical solution matches best the entropy weak solution. This illustrates the fact that any discretization of the Euler equation in non-conservative variables must be derived very carefully.

Oscillatory behaviour

In order to damp the oscillations in our pressure-correction scheme without impacting the accuracy of the scheme we implement an adaptive artificial viscosity method, the *Weak Local Residual* method [90]. This method consists in adding an artificial viscosity proportional to a weak local residual (WLR), constructed from one of the balance equation in its weak form. The discrete WLR is $o(h)$ near shocks, $o(h^\alpha)$ near contact waves with $1 < \alpha \leq 2$ and $o(h^3)$ in smooth regions [90]. We choose to use the weak residual of the mass balance:

$$\mathrm{WR} = \int_0^T \int_\Omega \rho \partial_t \varphi + \int_0^T \int_\Omega \rho \boldsymbol{u} \cdot \nabla \varphi \quad \forall \varphi \in \mathcal{C}_c^\infty(\Omega \times [0,T])$$

The WLR as a localized indicator of flow features — shocks, contact waves — is deduced from the weak residual using a decomposition of the test function φ on a basis of B-spline functions [90].

Introducing WLR artificial viscosity with a strong diffusion factor $C = 100$ for our scheme in test 5 significantly damps the oscillations (see figure 2.3). Despite the added diffusion the global accuracy of the scheme is not noticeably affected. Indeed the convergence orders in L^1 norms are 0.565 for the density, 0.956 for the pressure and 0.936 for the velocity.

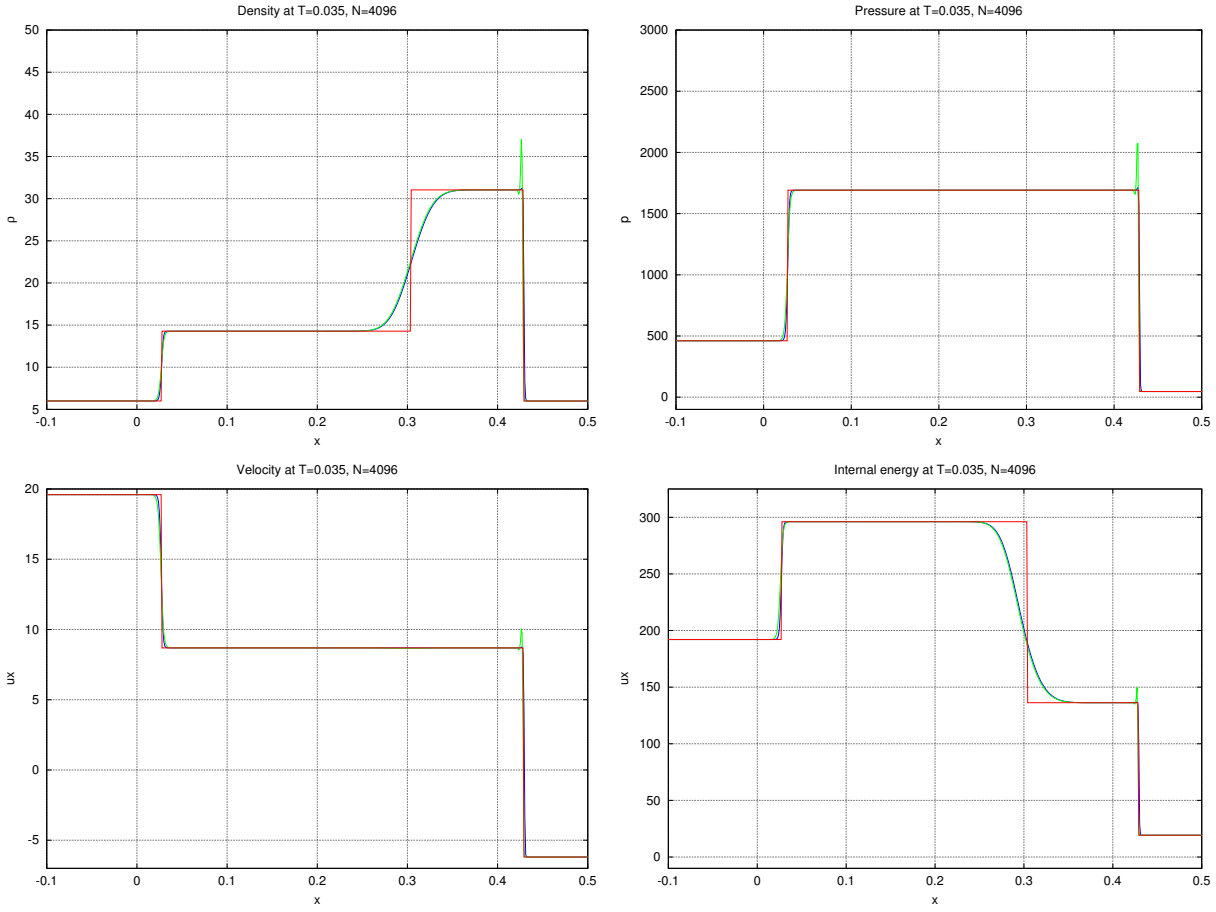


FIG. 2.1 – *Final state for test 5 with our scheme (blue),* SLK *(green) vs the exact solution (red).*

2.2.2 Two dimensional problems

Riemann problems

We now test a set of classical 2D Riemann problems, whose description can be found in [94]. The contours of the density for the 19 test cases are given in figures 2.4 to 2.22. The results obtained with SLK and with our scheme are similar except for problems with strong shocks as observed in configuration 3. Both schemes show an oscillatory behaviour near 1D shocks.

Two dimensional stabilization

In order to damp the oscillations of our scheme near strong shocks, we run test 3 (most defavorable configuration) with WLR artificial diffusion. We observe that 1D shocks are still accurately computed and the oscillations are reduced significantly. However, 2D shock speeds are inaccurate. This issue becomes worse as the artificial diffusion is increased. In fact when adding artificial viscosity in the momentum prediction, a compensation term (positive) has to be added in the right hand side of the internal energy balance in order to preserve the consistency of the scheme. The source term S_K is localized at flow discontinuities, so is this compensation term. Maybe when the latter introduces too much diffusion the compensation term S_K is too

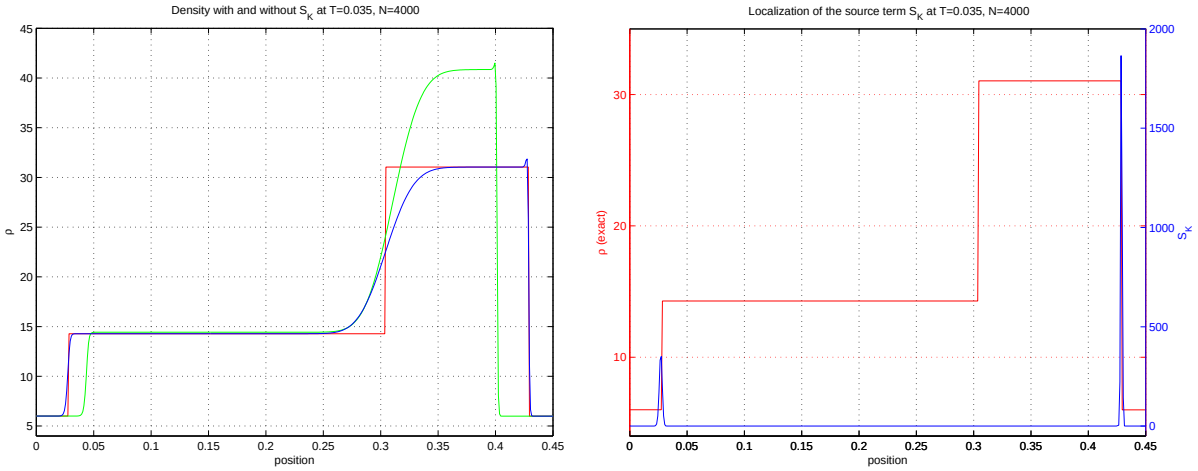


FIG. 2.2 – *Influence of the source term for test 5. Left: density at* $T = 0.035$ *with* S_K *(blue), without* S_K *(green) and exact solution (red). Right: localization of* S_K *(blue) with respect to the exact solution (red).*

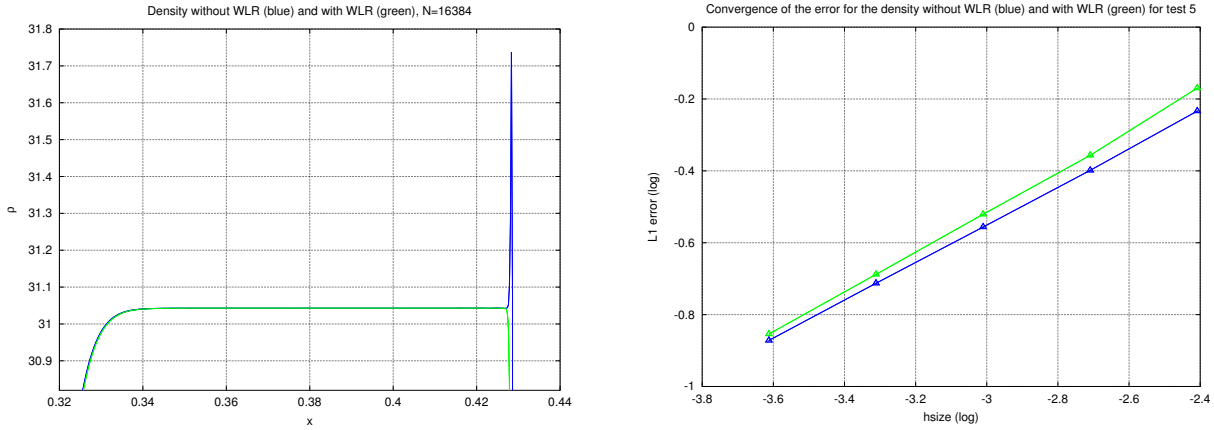


FIG. 2.3 – *Application of the WLR artificial viscosity method for our scheme in test 5.*

weakened.

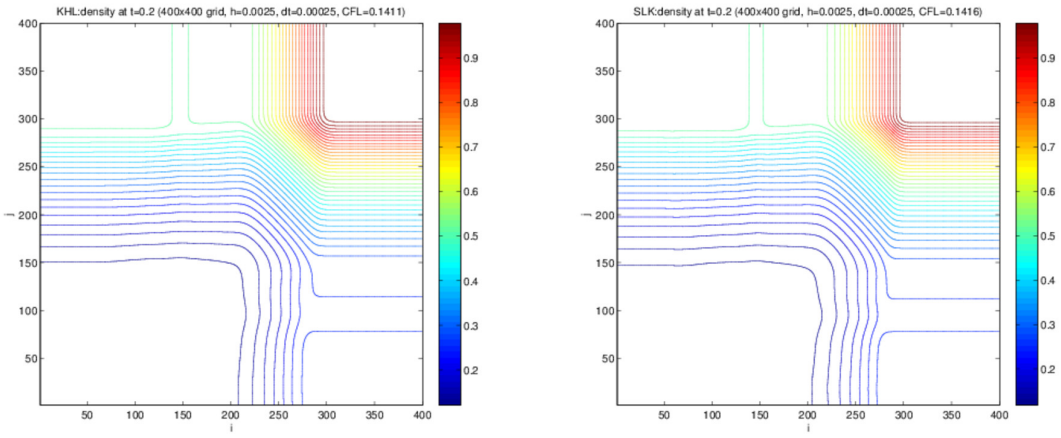


FIG. 2.4 – *Density contours (40) for 2D Riemann problem 1 with our scheme (left) and SLK (right).*

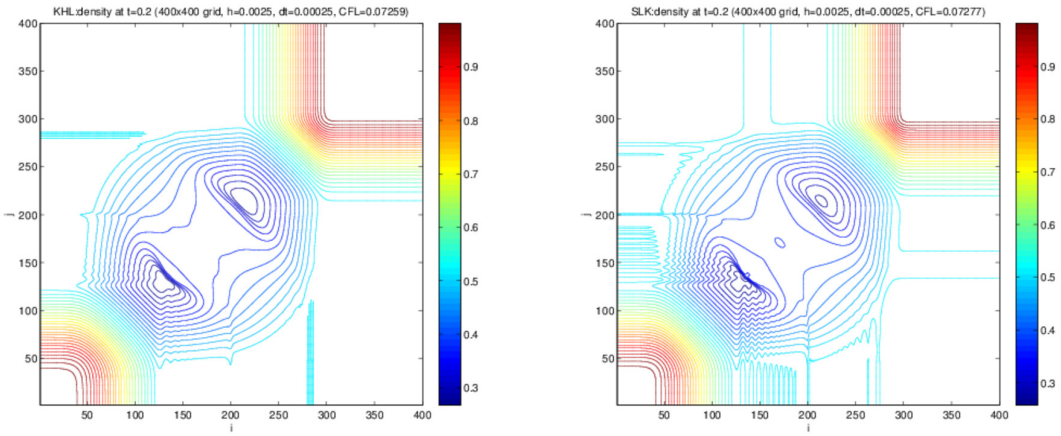


FIG. 2.5 – *Density contours (40) for 2D Riemann problem 2 with our scheme (left) and SLK (right).*

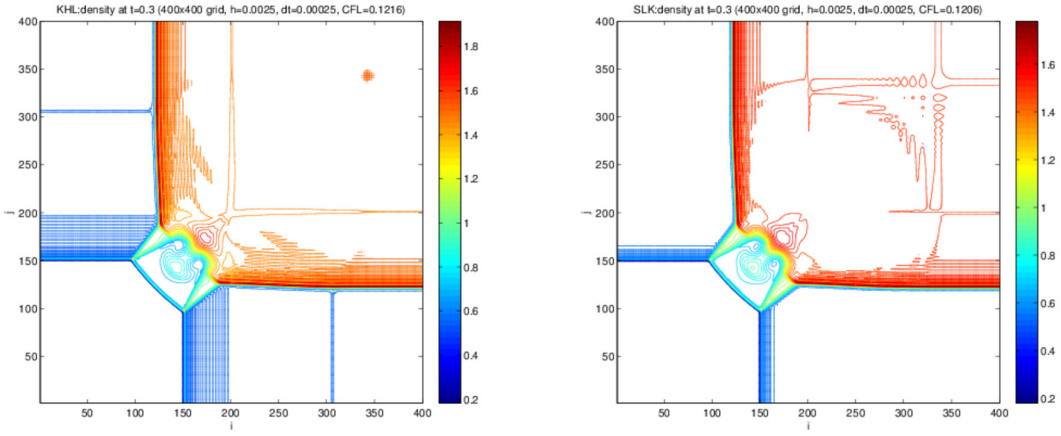


FIG. 2.6 – *Density contours (40) for 2D Riemann problem 3 with our scheme (left) and SLK (right).*

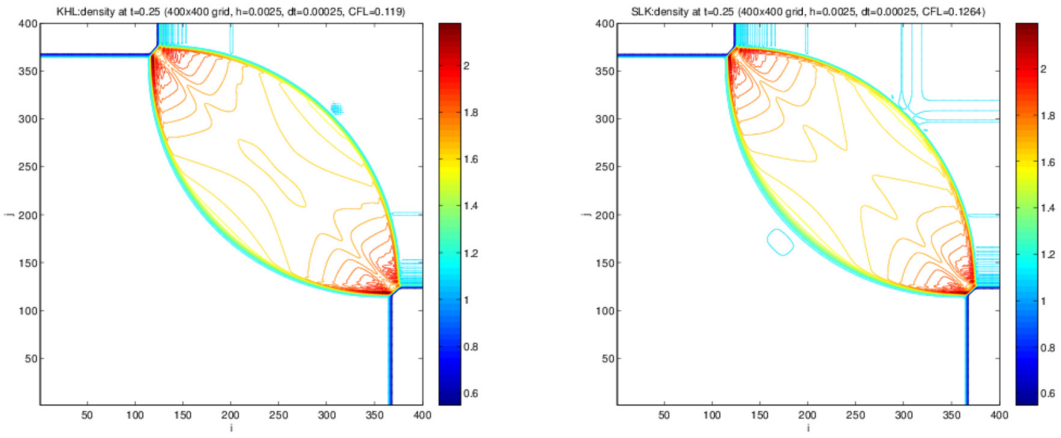


FIG. 2.7 – *Density contours (40) for 2D Riemann problem 4 with our scheme (left) and SLK (right).*

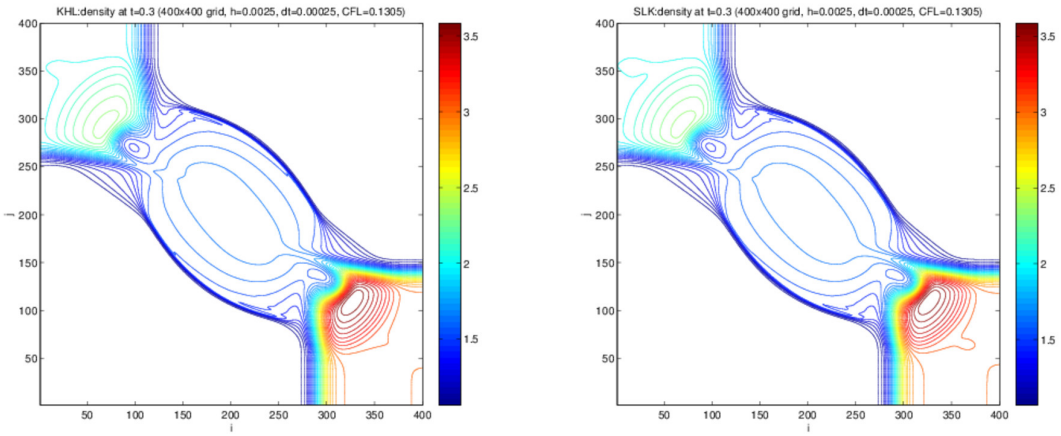


FIG. 2.8 – *Density contours (40) for 2D Riemann problem 5 with our scheme (left) and SLK (right).*

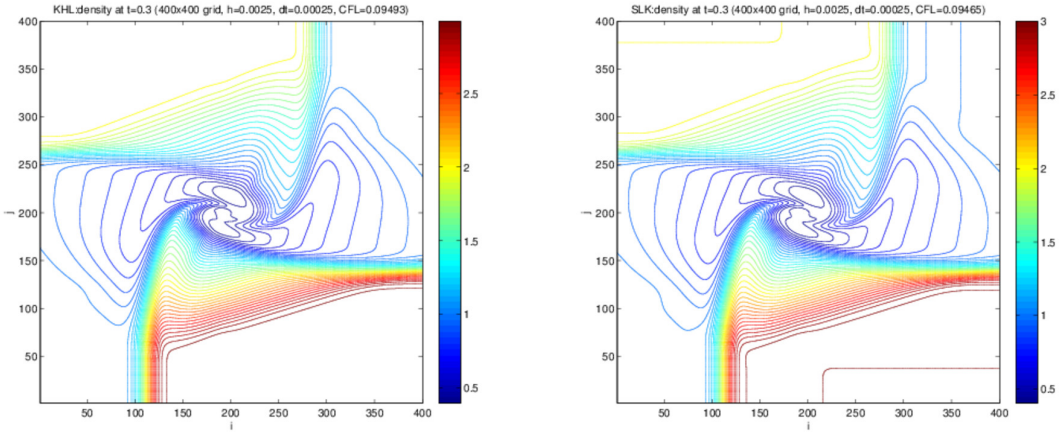


FIG. 2.9 – *Density contours (40) for 2D Riemann problem 6 with our scheme (left) and SLK (right).*

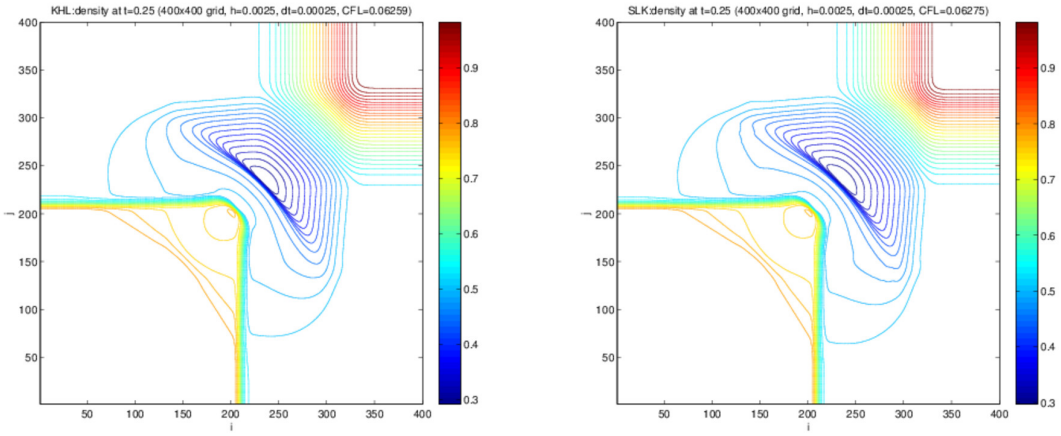


FIG. 2.10 – *Density contours (40) for 2D Riemann problem 7 with our scheme (left) and SLK (right).*

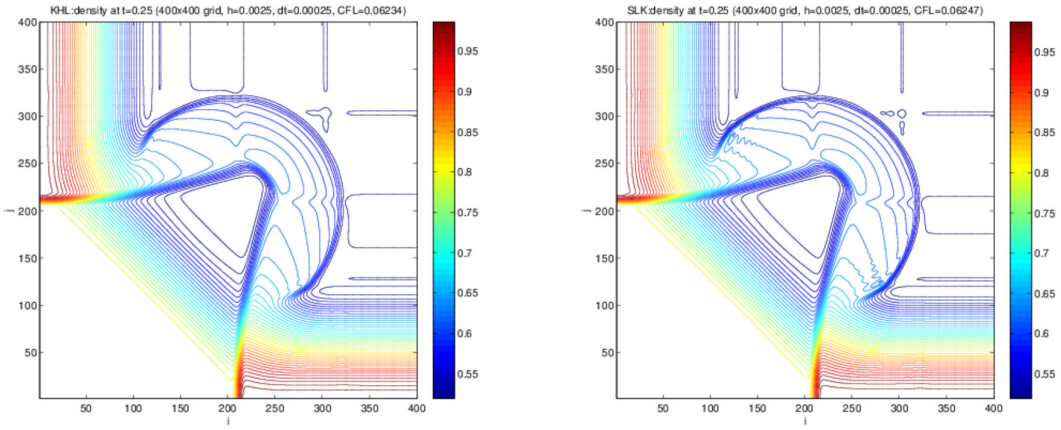


FIG. 2.11 – *Density contours (40) for 2D Riemann problem 8 with our scheme (left) and SLK (right).*

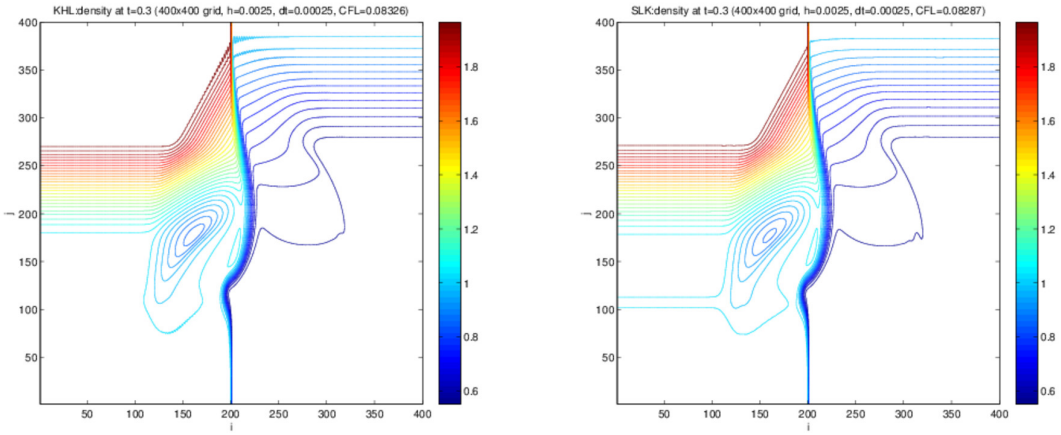


FIG. 2.12 – *Density contours (40) for 2D Riemann problem 9 with our scheme (left) and SLK (right).*

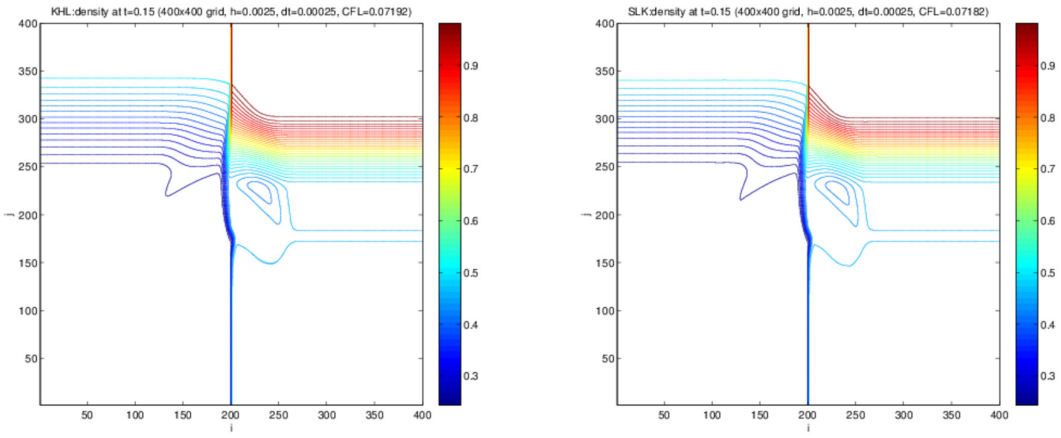


FIG. 2.13 – *Density contours (40) for 2D Riemann problem 10 with our scheme (left) and SLK (right).*

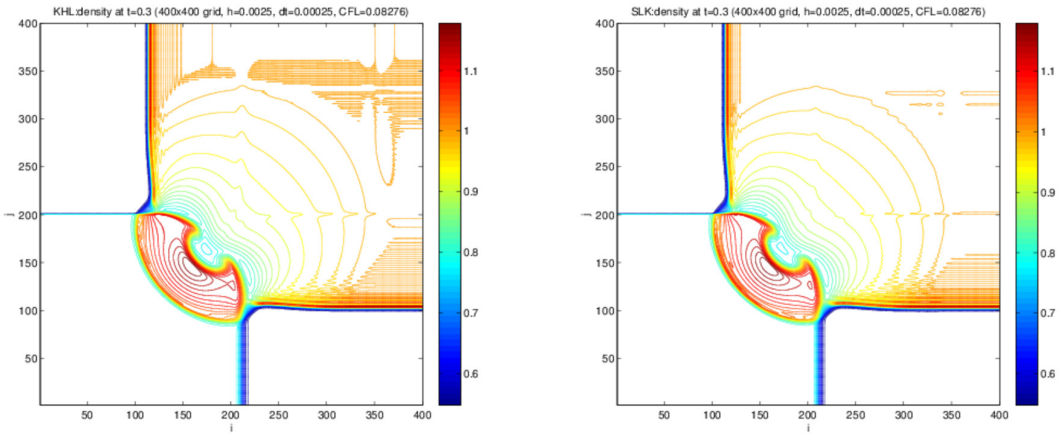


FIG. 2.14 – *Density contours (40) for 2D Riemann problem 11 with our scheme (left) and SLK (right).*

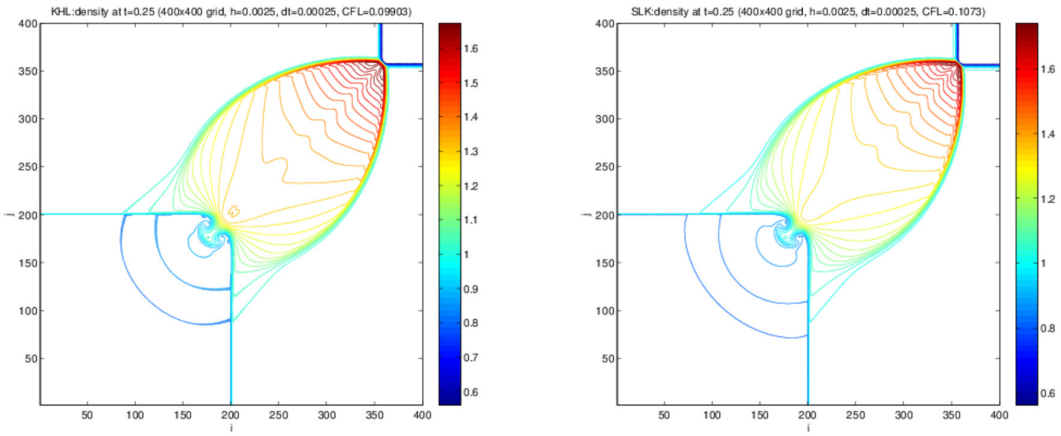


FIG. 2.15 – *Density contours (40) for 2D Riemann problem 12 with our scheme (left) and SLK (right).*

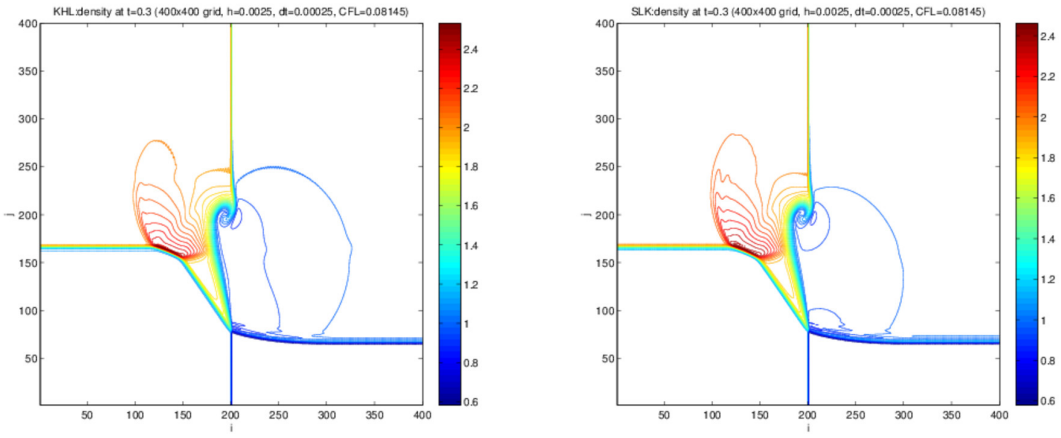


FIG. 2.16 – *Density contours (40) for 2D Riemann problem 13 with our scheme (left) and SLK (right).*

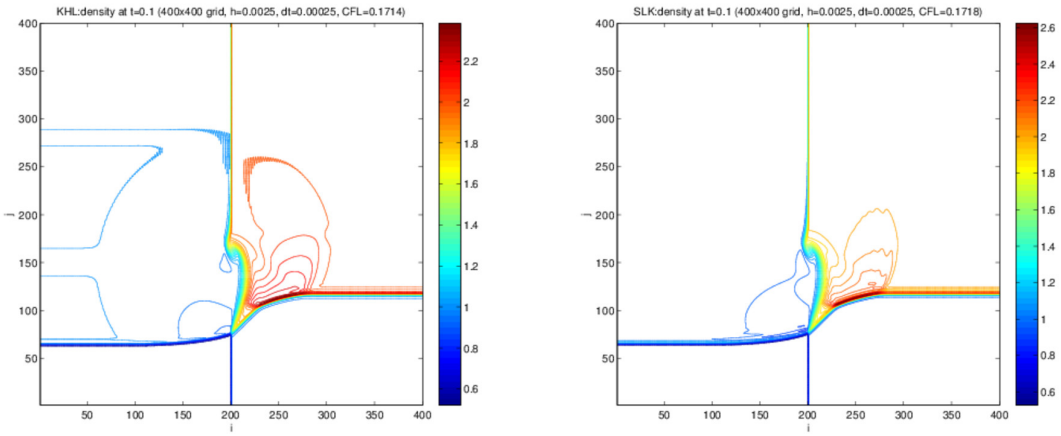


FIG. 2.17 – *Density contours (40) for 2D Riemann problem 14 with our scheme (left) and SLK (right).*

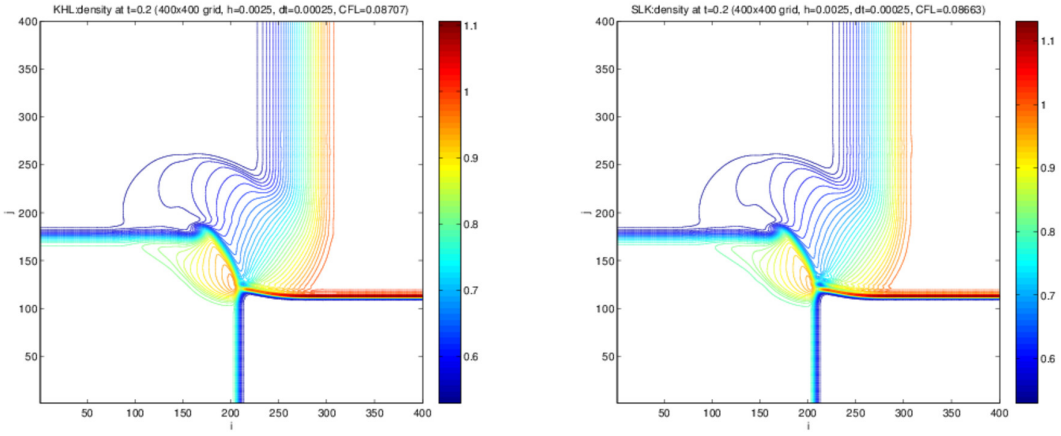


FIG. 2.18 – *Density contours (40) for 2D Riemann problem 15 with our scheme (left) and SLK (right).*

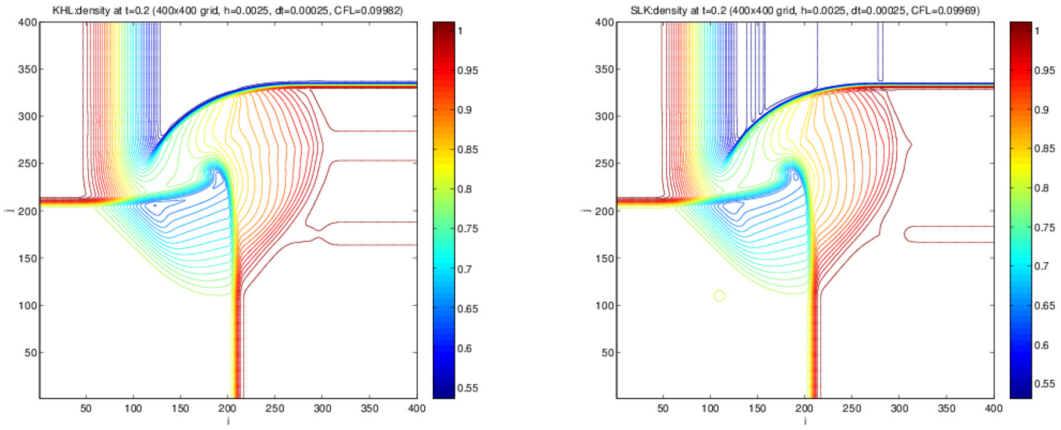


FIG. 2.19 – *Density contours (40) for 2D Riemann problem 16 with our scheme (left) and SLK (right).*

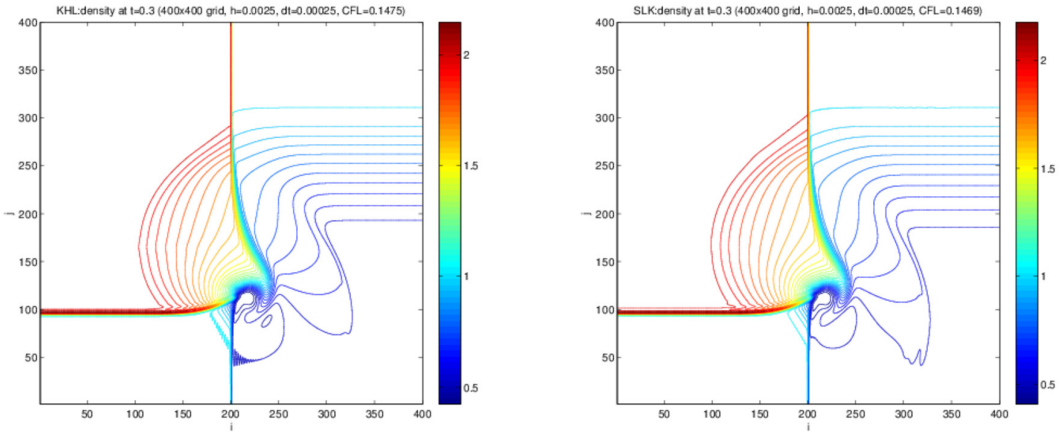


FIG. 2.20 – *Density contours (40) for 2D Riemann problem 17 with our scheme (left) and SLK (right).*

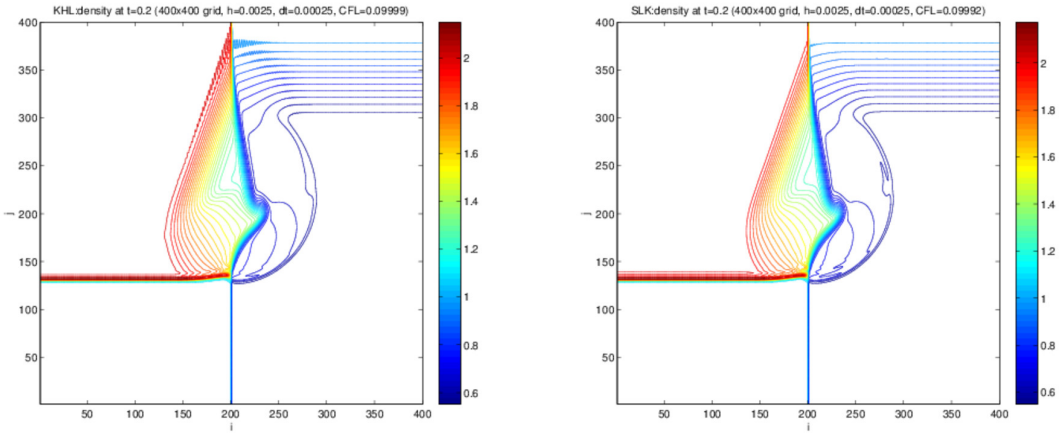


FIG. 2.21 – *Density contours (40) for 2D Riemann problem 18 with our scheme (left) and SLK (right).*

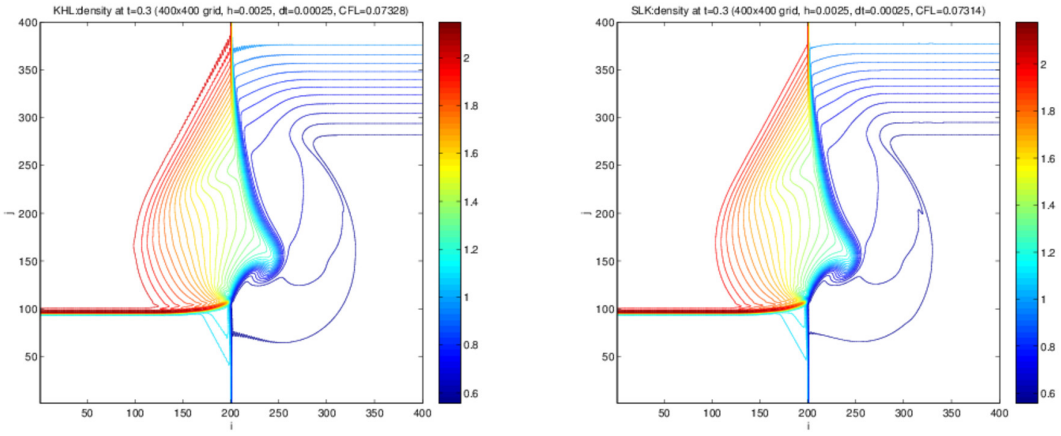


FIG. 2.22 – *Density contours (40) for 2D Riemann problem 19 with our scheme (left) and SLK (right).*

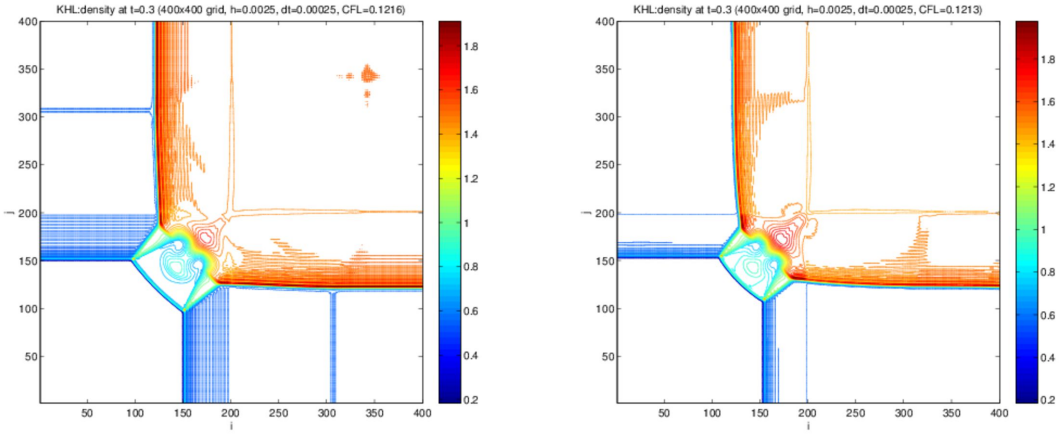


FIG. 2.23 – *Density contours (40) for 2D Riemann problems 3 without stabilization (left) and with WLR stabilization (right).*

Chapter 3

APPLICATION TO LOW-MACH FLOWS

We give here some numerical tests which were performed for low Mach number flows and compared with a staggered discretization. These results were presented at FVCA VII conference [125].

3.1 Introduction

We address in this Chapter the compressible Euler equations written with the internal energy as energy variable:

$$\partial_t \rho + \operatorname{div}(\rho \mathbf{u}) = 0, \tag{3.1.1a}$$

$$\partial_t (\rho \mathbf{u}) + \operatorname{div}(\rho \mathbf{u} \otimes \mathbf{u}) + \boldsymbol{\nabla} p = 0, \tag{3.1.1b}$$

$$\partial_t (\rho e) + \operatorname{div}(\rho e \mathbf{u}) + p \operatorname{div} \mathbf{u} = 0, \tag{3.1.1c}$$

$$p = (\gamma - 1)\rho e, \tag{3.1.1d}$$

where t stands for the time ; ρ, $\mathbf{u}$, p and e are the density, velocity, pressure and internal energy respectively, and $\gamma > 1$ is a coefficient specific to the fluid. The problem is defined over $\Omega \times (0, T)$, where Ω is an open bounded connected subset of $\mathbb{R}^d$, $1 \leq d \leq 3$, and $(0, T)$ is a finite time interval.

Defining a robust scheme for the numerical solution of the compressible Euler equations at all Mach number is a challenging issue. Indeed in the zero Mach limit, the pressure gradient has a singular limit and the acoustic time scale vanishes [4]. As a result approximate Riemann solvers face severe limitations, among which the loss of accuracy of the pressure gradient approximation and the time step limitation. Pressure-correction methods may be relevant for addressing this issue, in particular because of their built-in stability properties.

While pressure-correction schemes were originally introduced for the incompressible Navier-Stokes equations [43, 123] many extensions to compressible flows have been attempted [74, 48]. In this work we compare two finite volume discretizations — staggered and cell-centered — of an original pressure-correction scheme first introduced in [71, 76].

The use of the internal energy as energy variable is motivated by our will to control its positivity through the numerical scheme. The internal energy balance must be discretized carefully in order to force the scheme to be consistent with the total energy equation. Indeed, similarly to the continuous case, we obtain a (discrete) kinetic energy equation from the (discrete) momentum balance and the (discrete) mass balance in which there is a numerical diffusion term. This term must be compensated in the discrete internal energy balance so that the sum of the internal and kinetic energy equations yields the correct total energy equation.

3.2 Pressure correction scheme

We first introduce the pressure correction method in a semi-discrete time setting. Let δt be a time discretization step. We define the discrete time $t^n = n\delta t$ with $t^N = T$ and $N = \lfloor T/\delta t \rfloor$. The pressure-correction scheme reads:

- Solve for $\tilde{\mathbf{u}}^{n+1}$:

$$\frac{1}{\delta t}\left(\rho^n \tilde{\mathbf{u}}^{n+1} - \rho^{n-1}\tilde{\mathbf{u}}^n\right) + \operatorname{div}\left(\rho^n \tilde{\mathbf{u}}^{n+1} \otimes \mathbf{u}^n\right) + \sqrt{\frac{\rho^n}{\rho^{n-1}}}\boldsymbol{\nabla} p^n = 0.$$

- Solve for p^{n+1}, $\mathbf{u}^{n+1}$, ρ^{n+1} and e^{n+1} the non-linear system:

$$\begin{aligned}
&\frac{\rho^n}{\delta t}\left(\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1}\right) + \boldsymbol{\nabla} p^{n+1} - \sqrt{\frac{\rho^n}{\rho^{n-1}}}\boldsymbol{\nabla} p^n = 0,\\
&\frac{1}{\delta t}\left(\rho^{n+1} - \rho^n\right) + \operatorname{div}\left(\rho^{n+1}\mathbf{u}^{n+1}\right) = 0,\\
&\frac{1}{\delta t}\left(\rho^{n+1}e^{n+1} - \rho^n e^n\right) + \operatorname{div}\left(\rho^{n+1}e^{n+1}\mathbf{u}^{n+1}\right) + p^{n+1}\operatorname{div}\left(\mathbf{u}^{n+1}\right) = 0,\\
&p^{n+1} = (\gamma - 1)\rho^{n+1}e^{n+1}.
\end{aligned}$$

The first step is a classical semi-implicit discretization of the momentum balance to obtain a predicted velocity. The second step is a non-linear pressure correction step which combines the mass balance and the internal energy balance. This non-linear coupling is important to ensure the positivity of the energy. It is solved using Newton's method.

3.3 Spatial discretization

We suppose that the boundaries of the domain are sections of hyperplanes normal to a coordinate axis. Let $\mathcal{T}$ be a decomposition of Ω. The cells are either rectangles ($d = 2$) or rectangular parallelepipeds ($d = 3$). By $\mathcal{E}$ and $\mathcal{E}(K)$ we denote the set of all $(d-1)$-faces σ of the mesh and of the element $K \in \mathcal{T}$ respectively. The set of faces included in the boundary of Ω is denoted by $\mathcal{E}_{ext}$ and the set of internal faces (*i.e.* $\mathcal{E} \setminus \mathcal{E}_{ext}$) is denoted by $\mathcal{E}_{int}$; a face $\sigma \in \mathcal{E}_{int}$ separating the cells K and L is denoted by $\sigma = K|L$. The outward normal vector to a face σ of K is denoted by $\mathbf{n}_{K,\sigma}$. For $K \in \mathcal{T}$ and $\sigma \in \mathcal{E}$, we denote by $|K|$ the measure of K and by $|\sigma|$ the $(d-1)$-measure of the face σ. For $1 \le i \le d$, we denote by $\mathcal{E}^{(i)} \subset \mathcal{E}$ and $\mathcal{E}_{ext}^{(i)} \subset \mathcal{E}_{ext}$ the subset of the faces of $\mathcal{E}$ and $\mathcal{E}_{ext}$ respectively which are perpendicular to the i^{th} unit vector of the canonical basis of $\mathbb{R}^d$. The definition of the divergence operator is similar in both the cell-centered and the staggered scheme. For $(\mathbf{u}_\sigma^n)_{\sigma\in\mathcal{E}}$, we set:

$$\text{for } K \in \mathcal{T}, \qquad (\operatorname{div}\mathbf{u})_K^n = \frac{1}{|K|}\sum_{\sigma\in\mathcal{E}(K)} |\sigma|\; u_{K,\sigma}^n, \tag{3.3.1}$$

with $u_{K,\sigma}^n = \mathbf{u}_\sigma^n \cdot \mathbf{n}_{K,\sigma}$ the advecting velocity.

3.3.1 Cell-centered scheme

The unknowns are associated to the cells of the mesh $\mathcal{T}$ and are denoted by:

$$\{\rho_K,\ e_K,\ p_K,\ \mathbf{u}_K,\ K \in \mathcal{T}\}.$$

We first explain the initial discrete conditions: ρ^0, p^0 and $\mathbf{u}^0$ are given ; then we set for $K \in \mathcal{T}$ and $1 \leq i \leq d$:

$$\rho_K^0 = \frac{1}{|K|}\int_K \rho_0(\mathbf{x})\ \mathrm{d}\boldsymbol{x}, \quad e_K^0 = \frac{1}{|K|}\int_K e_0(\mathbf{x})\ \mathrm{d}\boldsymbol{x}, \quad \text{and} \quad u_{K,i}^0 = \frac{1}{|K|}\int_K (\mathbf{u}_0(\mathbf{x}))_i\ \mathrm{d}\boldsymbol{x}.$$

The fully discrete scheme then reads, for $n = 0, 1, \ldots, N-1$:

- *Velocity prediction step*:

$$\frac{|K|}{\delta t}(\rho_K^n \tilde{\mathbf{u}}_K^{n+1} - \rho_K^{n-1}\mathbf{u}_K^n) + \sum_{\sigma\in\mathcal{E}(K)} \tilde{\mathbf{u}}_\sigma^{n+1} F_{K,\sigma}^n + \sqrt{\frac{\rho_K^n}{\rho_K^{n-1}}}|K|(\boldsymbol{\nabla} p)_K^n = 0. \tag{3.3.2}$$

- *Projection step*: solve the non-linear system

$$\mathbf{u}_K^{n+1} = \tilde{\mathbf{u}}_K^{n+1} - \frac{\delta t}{\rho_K^n}\left((\boldsymbol{\nabla} p)_K^{n+1} - \sqrt{\frac{\rho_K^n}{\rho_K^{n-1}}}(\boldsymbol{\nabla} p)_K^n\right), \tag{3.3.3a}$$

$$\frac{|K|}{\delta t}(\rho_K^{n+1} - \rho_K^n) + \sum_{\sigma\in\mathcal{E}(K)} F_{K,\sigma}^{n+1} = 0, \tag{3.3.3b}$$

$$\frac{|K|}{\delta t}(\rho_K^{n+1} e_K^{n+1} - \rho_K^n e_K^n) + \sum_{\sigma\in\mathcal{E}(K)} e_\sigma^{n+1} F_{K,\sigma}^{n+1} + p_K^{n+1}\sum_{\sigma\in\mathcal{E}(K)} |\sigma| u_{K,\sigma}^{n+1} - S_K^n = 0, \tag{3.3.3c}$$

$$p_K^{n+1} = (\gamma - 1)\rho_K^{n+1} e_K^{n+1}, \tag{3.3.3d}$$

where $\tilde{\mathbf{u}}_\sigma^{n+1}$ in (3.3.2) is a centered interpolation of the velocity, $F_{K,\sigma}^{n+1} = |\sigma|\rho_\sigma^{n+1} u_{K,\sigma}^{n+1}$ is the mass flux and ρ_σ^{n+1}, e_σ^{n+1} are upwind interpolations with respect to the sign of $u_{K,\sigma}^{n+1}$ and $F_{K,\sigma}^{n+1}$ respectively. In the expression of the advecting velocity, we use a centered interpolation of the velocity at the face σ. In order to ensure the consistency of the scheme, the pressure gradient is constructed by duality with the discrete divergence of the velocity and reads:

$$(\boldsymbol{\nabla} p)_K^n = \frac{1}{|K|}\sum_{\sigma\in\mathcal{E}(K)} |\sigma| p_\sigma^n \mathbf{n}_{K,\sigma}, \tag{3.3.4}$$

with p_σ^n a centered interpolation of the pressure at face σ.

The corrective term S_K^n is defined as:

$$S_K^n = \frac{|K|}{2\delta t}\rho_K^{n-1}(\tilde{\mathbf{u}}_K^{n+1} - \mathbf{u}_K^n)^2. \tag{3.3.5}$$

3.3.2 Staggered scheme

The space discretization is staggered, using the Marker-And Cell (MAC) scheme. The degrees of freedom for scalar variables are still associated to the cells of the mesh, but the degrees of freedom for the i^{th} component of the velocity are defined at the center of the faces $\sigma \in \mathcal{E}^{(i)}$, so the whole set of discrete velocity unknowns reads:

$$\big\{u_{\sigma,i},\ \sigma \in \mathcal{E}^{(i)},\ 1 \leq i \leq d\big\}.$$

We introduce dual meshes for each direction i centered on $\sigma \in \mathcal{E}^{(i)}$, which are used for the finite volume approximation of the time derivative and convection terms in the momentum balance. For $\sigma = K|L \in \mathcal{E}^{(i)}$, we build a dual cell D_σ made of two half cells $D_{K,\sigma}$ and $D_{L,\sigma}$ included in K and L respectively. Each cell $D_{K,\sigma}$ is a rectangle or a rectangular parallelepiped of basis σ and of measure $|K|/2$. We denote by $|D_\sigma|$ the measure of D_σ and by $\varepsilon = D_\sigma|D_{\sigma'}$ the face separating D_σ and $D_{\sigma'}$. We denote by $\tilde{\mathcal{E}}$ the set of dual faces, $\tilde{\mathcal{E}}^{(i)}_{int}$ the internal faces in the direction i and $\tilde{\mathcal{E}}(D_\sigma)$ those belonging to D_σ.

We will only point out the major changes with respect to the cell-centered scheme. Initial conditions differ from the cell-centered scheme only for the velocities, which are now defined on the dual cells:

$$\forall \sigma \in \mathcal{E}_{\text{int}}, \quad u^0_{\sigma,i} = \frac{1}{|D_\sigma|}\int_{D_\sigma} (\mathbf{u}_0(\mathbf{x}))_i \ \mathrm{d}\boldsymbol{x}. \tag{3.3.6}$$

The definition of the divergence operator is the same as before but the discrete gradient is now defined on the dual mesh:

$$\forall \sigma = K|L \in \mathcal{E}_{\text{int}}, \quad (\boldsymbol{\nabla} p)^n_\sigma = \frac{|\sigma|}{|D_\sigma|}(p_L - p_K)\mathbf{n}_{K,\sigma}. \tag{3.3.7}$$

Equations for scalar variables have just minor changes. Unlike the cell-centered discretization the convective fluxes $u^{n+1}_{K,\sigma}$ are obtained without interpolation as the velocity unknowns are defined on the edges. We still use an upwind interpolation for ρ_σ and e_σ in (3.3.3b) and (3.3.3c) respectively. We need to rewrite the velocity updates (3.3.2) and (3.3.3a) on the dual mesh, which read for all $i \in [1, d]$, for all $\sigma \in \mathcal{E}^{(i)}_{\text{int}}$:

$$\frac{|D_\sigma|}{\delta t}(\rho^n_{D_\sigma}\tilde{u}^{n+1}_{\sigma,i} - \rho^{n-1}_{D_\sigma}u^n_{\sigma,i}) + \sum_{\varepsilon \in \tilde{\mathcal{E}}(D_\sigma)} \tilde{u}^{n+1}_{\varepsilon,i}F^n_{\sigma,\varepsilon} + \sqrt{\frac{\rho^n_{D_\sigma}}{\rho^{n-1}_{D_\sigma}}}|D_\sigma|(\boldsymbol{\nabla} p)^n_\sigma = 0, \tag{3.3.8}$$

$$u^{n+1}_{\sigma,i} = \tilde{u}^{n+1}_{\sigma,i} - \frac{\delta t}{\rho^n_{D_\sigma}}\left((\boldsymbol{\nabla} p)^{n+1}_\sigma - \sqrt{\frac{\rho^n_{D_\sigma}}{\rho^{n-1}_{D_\sigma}}}(\boldsymbol{\nabla} p)^n_\sigma\right). \tag{3.3.9}$$

The dual fluxes $F^n_{\sigma,\varepsilon}$ and densities ρ_{D_σ} are defined such that we recover a discrete mass balance over the dual cells. As mentioned in the introduction this is critical for obtaining a discrete kinetic balance. The corrective term S^n_K in the internal energy balance reads for all $K \in \mathcal{T}$:

$$S^n_K = \sum_{i=1}^d S^n_{K,i}, \quad \text{with} \quad S^n_{K,i} = \frac{1}{2}\rho^{n-1}_K \sum_{\sigma \in \mathcal{E}(K)\cap\mathcal{E}^{(i)}} \frac{|D_{K,\sigma}|}{\delta t}(\tilde{u}^{n+1}_{\sigma,i} - u^n_{\sigma,i})^2. \tag{3.3.10}$$

3.4 Discrete properties

Thanks to the upwind choice for the density in the mass balance both schemes preserve the positivity of the density, see [66, Lemma 2.2] for further details. With either discretization a kinetic energy balance can be derived from the momentum prediction equation:

Proposition 3.4.1. *(Discrete kinetic energy balance for the cell-centered discretization) A solution to the cell-centered (resp. staggered) scheme satisfies (3.4.1) (resp. (3.4.2)):*

$$\frac{|K|}{2\delta t}\left[\rho_K^n(\mathbf{u}_K^{n+1})^2-\rho_K^{n-1}(\mathbf{u}_K^n)^2\right]+\frac{1}{2}\sum_{\sigma\in\mathcal{E}(K)}\tilde{\mathbf{u}}_K^{n+1}\tilde{\mathbf{u}}_L^{n+1}F_{K,\sigma}^n,$$
$$+\mathbf{u}_K^{n+1}\cdot\sum_{\sigma\in\mathcal{E}(K)}|\sigma|p_\sigma^{n+1}\mathbf{n}_{K,\sigma}+P_K^{n+1}-R_K^{n+1}=0. \tag{3.4.1}$$

$$\frac{|D_\sigma|}{2\delta t}\left[\rho_{D_\sigma}^n(u_{\sigma,i}^{n+1})^2-\rho_{D_\sigma}^{n-1}(u_{\sigma,i}^n)^2\right]+\frac{1}{2}\sum_{\varepsilon\in\tilde{\mathcal{E}}(D_\sigma)}\tilde{u}_{\sigma,i}^{n+1}\tilde{u}_{\sigma',i}^{n+1}F_{\sigma,\varepsilon}^{n+1}$$
$$+\tilde{u}_{\sigma,i}^{n+1}|D_\sigma|(\boldsymbol{\nabla}p^{n+1})_\sigma^{(i)}+P_\sigma^{n+1}-R_{\sigma,i}^{n+1}=0, \tag{3.4.2}$$

with the following source terms:

$$R_K^{n+1}=-\frac{|K|}{2\delta t}\rho_K^{n-1}(\tilde{\mathbf{u}}_K^{n+1}-\mathbf{u}_K^n)^2,\quad R_{\sigma,i}^{n+1}=-\frac{|D_\sigma|}{2\delta t}\rho_{D_\sigma}^{n-1}(\tilde{u}_{\sigma,i}^{n+1}-u_{\sigma,i}^n)^2,$$
$$P_K^{n+1}=\frac{\delta t}{2}\left[\frac{1}{\rho_K^n}\left((\boldsymbol{\nabla}p)_K^{n+1}\right)^2-\frac{1}{\rho_K^{n-1}}\left((\boldsymbol{\nabla}p)_K^n\right)^2\right],$$
$$P_\sigma^{n+1}=\frac{\delta t}{2}\frac{|\sigma|^2}{|D_\sigma|^2}\left[\frac{1}{\rho_{D_\sigma}^n}\left((\boldsymbol{\nabla}p)_\sigma^{n+1}\right)^2-\frac{1}{\rho_{D_\sigma}^{n-1}}\left((\boldsymbol{\nabla}p)_\sigma^n\right)^2\right].$$

In both schemes, the corrective term S_K^n in the internal energy balance is intended to compensate the terms R_K^{n+1} and R_σ^{n+1} which tend to zero, hence the expression of the corrective term S_K given in (3.3.5) and (3.3.10). Note that S_K is always positive, which ensures the positivity of the internal energy thanks to the following proposition proved in Chapter 1:

Proposition 3.4.2. *(Positivity of the internal energy) If* $\forall K\in\mathcal{T},\ e_K^n\geq 0,\ S_K^n\geq 0$ *and* $\rho_K>0$ *then* $\forall K\in\mathcal{T},\ e_K^{n+1}\geq 0$

3.5 Numerical results

The two discretizations are tested with a recent benchmark introduced in [34]. This benchmark aims at testing numerical schemes for the compressible Euler equations against their incompressible limit when the Mach number M tends to zero. It consists in a Taylor vortex in a unit square cavity $\Omega=[0,1]\times[0,1]$. The initial solution verifies the incompressible Euler equations and reads in non-dimensional variables:

$$\rho_0(\mathbf{x})=1,\ \ \mathbf{u}_0(\mathbf{x})=\begin{pmatrix}\sin(\pi x)\cos(\pi y)\\-\cos(\pi x)\sin(\pi y))\end{pmatrix},\ \ p_0(\mathbf{x})=\frac{1}{\gamma M^2}+\frac{1}{4}(\cos(2\pi x)+\cos(2\pi y))$$

However it does not lead to a steady flow with the compressible Euler equations, as the homogeneous density induces variations of the entropy. The main idea is to study the behaviour

of the scheme at two scales: the macroscopic scale (slow variations associated with time variable t) and the acoustic scale (fast variations associated with time variable $\tau = t/M$). Each flow variable is decomposed as $X(\mathbf{x},\tau,t) = \bar{X}(\mathbf{x},t) + \delta X(\mathbf{x},\tau,t)$ with $\bar{X}(\mathbf{x},t)$ its time average over the acoustic scale and $\delta X(\mathbf{x},\tau,t)$ the fast time fluctuations. The asymptotic expansion of the non-dimensional flow variables with respect to the Mach number yields [34]:

$$p(\mathbf{x},t) = p_0(\mathbf{x}) + M\delta P_3(\mathbf{x},\tau,0) + M^2(\bar{P}_4(\mathbf{x},t) + \delta P_4(\mathbf{x},\tau,t)) + o(M^2),$$
$$\rho(\mathbf{x},t) = \rho_0(\mathbf{x}) + M^2\bar{\rho}_2(\mathbf{x},t) + M^3\delta\rho_3(\mathbf{x},\tau,0) + M^4(\bar{\rho}_4(\mathbf{x},t) + \delta\rho_4(\mathbf{x},\tau,t)) + o(M^4).$$

The particular field chosen for initialization allows the derivation of an analytic solution well suited for spectral analysis. We focus on two terms of the asymptotic expansion: $\bar{\rho}_2$, associated with the slow variations and $\mathcal{P}_3 = \delta P_3(\mathbf{x},\tau,0) + M(\bar{P}_4(\mathbf{x},t) + \delta P_4(\mathbf{x},\tau,t))$ associated with the fast variations. In practice these two terms are computed as $\bar{\rho}_2 = (\rho - \rho_0)/M^2$ and $\mathcal{P}_3 = (p - p_0)/M$.

Our numerical simulations are carried out on a 400×400 grid with $M = 0.1$ and $M = 0.01$. We observe very similar results for both cell-centered and staggered discretizations. At the macroscopic scale, the upwind diffusion damps the main modes of the density, which looks smooth at $T = 8.8$ (figure 3.1). As for the term $\bar{\rho}_2$, the oscillations of the solution are completely damped after $t = 4$ (figure 3.2, left). The Mach number does not appear to have any influence on this term. At the acoustic scale, the fluctuations of the pressure $\mathcal{P}_3$ on the short time interval $(0, 5)$ are also close with both discretizations (figure 3.2, center). After $t = 0.5$, the amplitude of the main mode of $\mathcal{P}_3$ (frequency $f = \sqrt{10}/2$) is decreased by two orders of magnitude.

The results of this benchmark do not feature spurious pressure modes for the cell-centered discretization as we might have expected. Indeed the internal energy balance (3.3.3c) can be reformulated as a non-linear equation on the pressure using the velocity update (3.3.3a) and the equation of state (3.3.3d):

$$M^2 \left\{ \frac{|K|}{\delta t}(P_K^{n+1} - P_K^n) + \sum_{\sigma\in\mathcal{E}(K)} |\sigma|(P_\sigma^{n+1} - (\gamma-1)P_K^{n+1}) \left[\frac{\delta t}{2}(\mathbf{g}_{K,\sigma}^{n+1} + \mathbf{g}_{L,\sigma}^{n+1}) \cdot \mathbf{n}_{K,\sigma} + \tilde{u}_{K,\sigma}^{n+1} \right] - (\gamma-1)S_K^n \right\} + \sum_{\sigma\in\mathcal{E}(K)} |\sigma| \left[\frac{\delta t}{2}(\mathbf{g}_{K,\sigma}^{n+1} + \mathbf{g}_{L,\sigma}^{n+1}) \cdot \mathbf{n}_{K,\sigma} + \tilde{u}_{K,\sigma}^{n+1} \right] = 0$$

with the change of variables $P = p - 1/(\gamma M^2)$, P_σ^{n+1} the upwind interpolation with respect to $F_{K,\sigma}^{n+1}$ and $\mathbf{g}_{K,\sigma}^{n+1} = (\rho_K^{n-1}\rho_K^n)^{-1/2}(\nabla P)_K^n - (\rho_K^n)^{-1}(\nabla P)_K^{n+1}$ for the cell-centered discretization. In the zero Mach limit this equation degenerates to the classical Poisson equation of the projection step of incremental pressure-correction schemes for incompressible flows. For the cell-centered discretization the resulting discrete Laplace operator introduces a decoupling between neighboring pressure unknowns, which is not the case with the staggered discretization. We managed to introduce spurious pressure modes for the cell-centered discretization by adding artificially a Dirac to the right hand side of this pressure equation at $t = 0$. However, these oscillations are quickly damped by the boundary conditions. We expect sustained spurious pressure modes in the case of an open boundary.

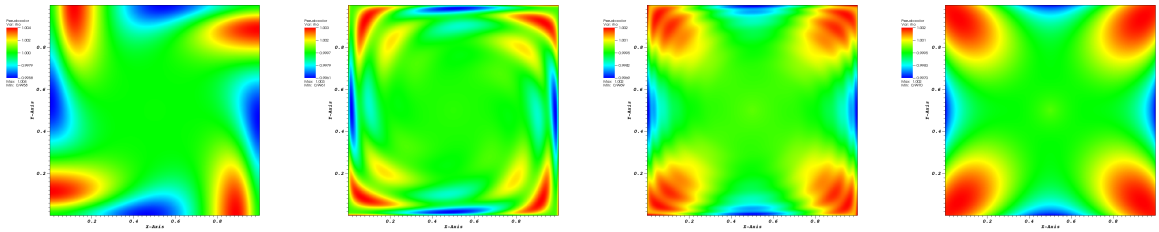


FIG. 3.1 – *Density field for the staggered discretization at* $t = 0.5$, $t = 2$, $t = 4$ *and* $t = 8.8$ *for* $M = 0.1$. *The density fields obtained with* $M = 0.01$ *and with the cell-centered discretization are the same.*

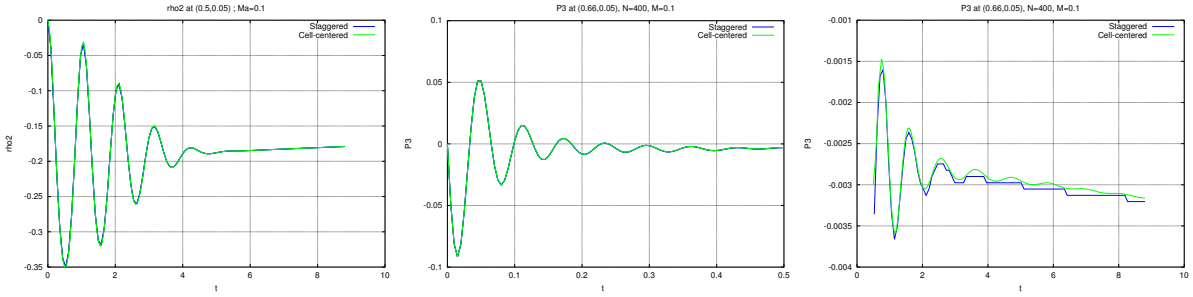


FIG. 3.2 – *Evolution of* $\bar{\rho}_2$ *(left) at position* $(0.5, 0.05)$ *and* $\mathcal{P}_3$ *(middle and right) at position* $(0.66, 0.05)$ *for* $M = 0.1$.

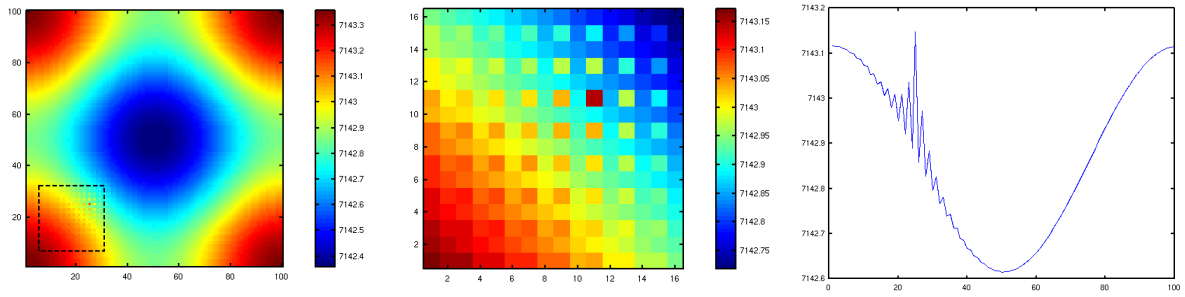


FIG. 3.3 – *Checkerboard pressure when the RHS of the projection equation is perturbed with a Dirac.*

Chapter 4

APPLICATION TO TWO-PHASE FLOWS

The GENEPI numerical code developed at CEA is used at industrial scale by AREVA to design steam generators for Pressurized Water Reactors (PWR). The objective of this Chapter is to extend the pressure-correction scheme introduced in Chapter 1 to the compressible homogeneous two-phase flow models of Genepi and assess its validity on simple 1D problems. We first review the two-phase flow model in GENEPI and its simplification. Then the projection scheme is derived and numerical results are presented for an original benchmark from the GENEPI test suite.

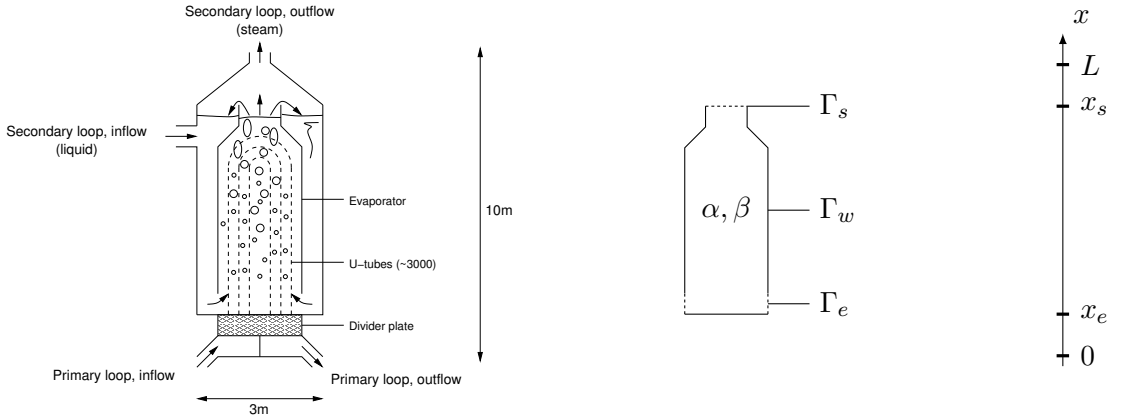


FIG. 4.1 – *Left: PWR Steam Generator. Center: macroscopic description using an equivalent porous medium and mixture variables. Right: simplified 1D model for the purpose of testing the two-phase flow model without the complications induced by the geometry.*

4.1 Homogeneous two-phase flow models

We denote by T the temperature, $\mathbf{u}$ the fluid velocity, e the internal energy, h the enthalpy, ρ density, p the pressure, q the heat flux from the primary fluid to the secondary fluid through the U-tubes and $\mathcal{L}$ the latent heat. The physical problem is defined in the evaporator Ω. The subscript K denotes a physical quantity belonging to phase K, either the liquid phase ($K = L$)

or gas phase ($K = G$) of the secondary fluid, or the U-tubes with the primary fluid ($K = P$). A flow variable without the aforementioned subscript is a mixture variable. For instance the mixture density ρ is related to phase variables as:

$$\rho = \alpha\rho_G + (1-\alpha)\rho_L \tag{4.1.1}$$

with α the void fraction. The *static* quality X is ratio of the steam flow rate to the total flow rate, defined as $X = \alpha\rho_G/\rho$. In the following all flow variables are assumed to be at saturation T^*.

We first present the general two-phase flow model implemented in GENEPI and then explain how the simplified model introduced hereafter is still relevant in complexity to assess the relevance of our projection scheme for industrial two-phase flow models.

4.1.1 GENEPI general model

The GENEPI physical model is intended for the simulation of the secondary fluid flow in the evaporator of the steam generator in the nominal regime. The latter considers a steady flow, weakly compressible, for low Mach numbers. The fluid of the secondary loop is heated by the primary fluid coming from the core and flowing in the U-tubes. This yields to a bubbly flow and ultimately to the generation of steam. GENEPI treats this problem at the macroscopic scale. The local Navier-Stokes equations are upscaled in such way that the complex geometry inside the evaporator (figure 4.1, left) is accounted for with an equivalent porous medium of porosity β (figure 4.1, center). After upscaling the flow is described using mixture variables only.

The original GENEPI model reads [39]:

$$\rho_P C_P \partial_t T_P + \rho_P C_P \mathbf{u}_P \cdot \nabla T_P - \operatorname{div}(C_P \chi_{TP} \nabla T_P) = -\frac{\gamma_0 h_{\text{eq}}}{\beta_{P_0}}(T_P - T_W) \tag{4.1.2}$$

$$\beta\cancel{\partial_t \rho} + \operatorname{div}(\beta\rho\mathbf{u}) = 0 \tag{4.1.3}$$

$$\begin{aligned}\beta\rho\partial_t \mathbf{u} + \beta\rho\mathbf{u}\cdot\nabla\,\mathbf{u} - \operatorname{div}(\beta 2\mu_T(\nabla\mathbf{u} + \nabla^t\mathbf{u})) + \beta\nabla p = \beta\rho\mathbf{g} - \beta\mathbf{\Lambda}\cdot\rho\mathbf{u} \\ - \operatorname{div}(\beta X(1-X)\rho\mathbf{u}_R\otimes\mathbf{u}_R)\end{aligned} \tag{4.1.4}$$

$$\begin{aligned}\beta\rho\partial_t h + \beta\rho\mathbf{u}\cdot\nabla h - \operatorname{div}(\beta\chi_T\nabla j) = \gamma_0 h_{\text{eq}}(T_P - T_W) + \cancel{\partial_t p} + \cancel{\mathbf{u}\cdot\nabla p} \\ - \operatorname{div}(\beta X(1-X)\rho\mathcal{L}\mathbf{u}_R)\end{aligned} \tag{4.1.5}$$

Equation (4.1.2) gives the primary fluid energy balance. C_P stands for the specific heat capacity, χ_{TP} the turbulent thermal conductivity, h_{eq} the equivalent exchange coefficient, β_{P_0} the bundle primary porosity, γ_0 the heating surface density between the primary loop and the secondary loop and T_W the temperature at the wall of the U-tubes. The right hand side models the heat transfer between the primary fluid and the secondary fluid. Equation (4.1.3) is the secondary fluid mass balance. Equation (4.1.4) gives the secondary fluid momentum balance. Here are denoted by μ_T the turbulent viscosity, $\mathbf{g}$ the gravitational constant, $\mathbf{\Lambda}$ a friction tensor and $\mathbf{u}_R$ a drift velocity. Finally equation (4.1.5) gives the secondary fluid energy balance. This system of equations is supplemented by a tabulated equation of state. The latter is accessed through high order polynomials fitting these tabulated values [40].

Remark 4.1.1. GENEPI computes steady state solutions as the limit a "pseudo-transient", hence the presence of some unsteady terms. The missing terms with respect to a proper unsteady

model are the time derivatives $\partial_t p$ and $\partial_t \rho$ and also the term $\mathbf{u} \cdot \nabla p$. The latter was deemed negligible by the authors of GENEPI for the targeted industrial problems.

Remark 4.1.2. The tensor $\mathbf{\Lambda}$ models the friction with the tubes, the tube support plates and the anti-vibration bars. It is determined through physical experiments.

Remark 4.1.3. The system always assumes a thermal equilibrium T^* though allowing a slight kinematic imbalance. The latter is represented by the drift velocity $\mathbf{u}_R \equiv \mathbf{u}_G - \mathbf{u}_L$. In the following we assume the so called *slip model* for the drift velocity: the gas and the fluid velocities are linked by tensor $\mathbf{S}$ with $\mathbf{u}_G = \mathbf{S} \cdot \mathbf{u}_L$.

4.1.2 GENEPI simplified model

We now derive a simplified model of (4.1.2) to (4.1.5), which is based upon the following assumptions:

1. The evaporator is represented by the 1D domain $[x_e, x_s]$ depicted on figure 4.1 (right).
2. The heat flux from the primary fluid is constant, hence no coupling through (4.1.2).
3. The porosity is taken to $\beta = 1$, i.e. no obstacle in the evaporator.
4. The gravity, the viscous stress and the friction with the internal obstacles of the evaporator are neglected.

The above assumptions may seem crude and oversimplifying w.r.t. the actual industrial applications but the very purpose of this chapter is to test the two-phase flow features of the GENEPI model only. Therefore any term adding further complexity (eg. more realistic geometry, unsteady flux from primary loop) and possibly substantial interference to the conclusion that could be drawn upon our numerical tests are removed. The drift flux terms and the equation of state — detailed in the next section — are unchanged. The problem is now governed by the following set of equations, identical to the compressible Euler equations but with the addition of specific source terms:

$$\partial_t \rho + \operatorname{div}(\rho \mathbf{u}) = 0 \tag{4.1.6a}$$

$$\partial_t (\rho \mathbf{u}) + \operatorname{div}(\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = -\operatorname{div}(X(1-X)\rho \mathbf{u_R} \otimes \mathbf{u_R}) \tag{4.1.6b}$$

$$\partial_t (\rho e) + \operatorname{div}(\rho e \mathbf{u}) + p \operatorname{div} \mathbf{u} = q - \operatorname{div}(X(1-X)\mathcal{L}\rho \mathbf{u_R}) \tag{4.1.6c}$$

with q the constant heat flux from the primary loop. The energy balance is written using the internal energy $e = h - p/\rho$ as energy variable in order to be in line with the models used for deriving our pressure-correction scheme. This is equivalent to solving:

$$\partial_t (\rho h) + \operatorname{div}(\rho h \mathbf{u}) - \mathbf{u} \cdot \nabla p - \partial_t p = q - \operatorname{div}(X(1-X)\mathcal{L}\rho \mathbf{V_R}) \tag{4.1.7}$$

The simplified model 4.1.6 suitable with GENEPI except for the term $p \operatorname{div} \mathbf{u}$ in (4.1.6c) which becomes $-\mathbf{u} \cdot \nabla p - \partial_t p$ in (4.1.7). In order to compare the numerical results of our pressure correction scheme implemented in our own code to those obtained with GENEPI, a particular heat source term $\tilde{q} = q - \mathbf{u}^\star \cdot \nabla p^\star$ is introduced. Here $\mathbf{u}^\star$ and $p^\star$ stand respectively for the velocity and for the pressure of the analytical solution at the steady state. As a result while the transient calculated by our scheme the one calculated by GENEPI will be different (ours is "physical"), the steady-state solution should match.

4.1.3 Equation of state

We consider the equation of state of Freon R-114 gas so as to match the conditions of the GENEPI test suite presented in reference [109]. A pressure-enthalpy diagram of Freon R-114 is given in figure 4.5. The equation of state for Freon R-114 in the GENEPI code is defined using polynomials fitting experiments carried out at CEA. The polynomials have the following form:

$$\begin{array}{ll} \rho_L(h_L) & = a_0 + a_1 h_L + a_2 h_L^2 \\ h_L(p) & = b_0 + b_1 p + b_2 p^2 + b_3 p^3 \\ \mathcal{L}(p) & = c_0 + c_1 p + c_2 p^2 \\ \rho_G(p) & = d_0 + d_1 p + d_2 p^2 \end{array} \tag{4.1.8}$$

The polynomial coefficients are found in the internal report [40]. Let us recall that all the thermodynamic variables above are at saturation. In order to derive an equation of state involving only the pressure, the density and the internal energy, we start from the definition of the enthalpy:

$$p - \rho(h - e) = 0$$

Using the classical definition of the static quality X

$$X = \frac{h - h_L}{\mathcal{L}}$$

yields:

$$p - \rho(h_L - e) + \rho\mathcal{L}X = 0$$

The definition of mixture variable ρ in (4.1.1) combined with the expression of X with the void fraction gives:

$$X = \frac{\rho_G(\rho_L - \rho)}{\rho(\rho_L - \rho_G)}$$

Hence the following equation of state:

$$\boxed{(p - (h_L(p) - e)\rho)(\rho_L(h_L(p)) - \rho_G(p)) - \rho_G(p)(\rho_L(h_L(p)) - \rho)\mathcal{L}(p) = 0}$$

This equation depends only on the mixture variables at saturation ρ, e and p. Using the polynomial fittings previously introduced, we have:

$$\begin{aligned} (p - \rho(b_3p^3 + b_2p^2 + b_1p + b_0 - e))(-d_2p^2 - d_1p - d_0 + a_2(b_3p^3 + b_2p^2 + b_1p + b_0)^2 \\ + a_1(b_3p^3 + b_2p^2 + b_1p + b_0) + a_0) - (a_2(b_3p^3 + b_2p^2 + b_1p + b_0)^2 \\ + a_1(b_3p^3 + b_2p^2 + b_1p + b_0) + a_0 - \rho)(c_2p^2 + c_1p + c_0)(d_2p^2 + d_1p + d_0) = 0 \end{aligned} \tag{4.1.9}$$

The equation of state which will be effectively used in our scheme is a highly non-linear law in the form of a multivariate polynomial with monomial terms $e^l\rho^m p^n$.

4.1.4 Drift velocity models

For a 1D problem, tensor $\mathbf{S}$ reduces to a scalar s and we have:

$$\mathbf{u}_G = s \cdot \mathbf{u}_L$$

Using the definition of the drift velocity $\mathbf{u}_R \equiv \mathbf{u}_G - \mathbf{u}_L$ and the definition of the mixture velocity:

$$\mathbf{u} = X\mathbf{u}_G + (1-X)\mathbf{u}_L$$

the following expression of $\mathbf{u}_R$ can be established:

$$\mathbf{u}_R = \frac{s-1}{(s-1)X+1}\mathbf{u}$$

Two models for the slip coefficient s will be considered. The first one assumes s constant:

$$s = C \tag{4.1.10}$$

with $C \in]0,2]$. The second one known as the *Chisholm model* [19] is more complex:

$$s = \min(s_1, s_2) \text{ with } \begin{cases} s_1 = \left[1 + \left(\dfrac{\rho_L}{\rho_G} - 1\right) X_d\right]^{1/2} \\ s_2 = \left(\dfrac{\rho_L}{\rho_G}\right)^{1/4} \end{cases} \tag{4.1.11}$$

with $X_d = X|\mathbf{u}_G|/|\mathbf{u}|$ the *dynamic* quality, which can rewritten as:

$$X_d = \frac{sX}{1+(s-1)X}$$

4.2 Numerical method

4.2.1 General projection algorithm

The same pressure correction scheme as in the first chapter for the compressible Euler equations is used with a few differences:

- It is assumed that our physical problems will not feature shocks. Therefore source term S_K^{n+1} is dropped.
- The drift flux is accounted for with explicit source terms.
- The static quality X is interpolated at cell edges with a centered scheme.
- The heat source term $\tilde{q}_K$ features the original heat flux from the primary fluid and the constant correction $\mathbf{u}_K^\star \cdot (\nabla p^\star)_K$ so that the steady state matches that of GENEPI.
- The slip coefficient s may be constant or calculated with a specific procedure in the case of the Chisholm model (as explained in the next section).

Finally the algorithm reads:

Initialization

$$h_K^0,\ \mathbf{u}_K^0,\ p_K^0 \text{ given} \quad ; \quad \rho_K^{-1} = \rho_K^0 \text{ deduced} \quad ; \quad e_K^0 = h_K^0 - p_K^0/\rho_K^0 \tag{4.2.1}$$

Iterations for $n = 0, 1, \ldots, N-1$:

1. *Explicit quantities*:

$$X_K^n = \frac{\rho_G(p_K^n)\left[\rho_L(h_L(p_K^n)) - \rho_K^n\right]}{\rho_K^n\left[\rho_L(h_L(p_K^n)) - \rho_G(p_K^n)\right]}$$
$$(\mathbf{u}_R)_\sigma^n = \frac{s-1}{(s-1)X_\sigma^n + 1}\mathbf{u}_\sigma^n \tag{4.2.2}$$
$$(u_R)_{K,\sigma}^n = \frac{s-1}{(s-1)X_\sigma^n + 1}\mathbf{u}_{K,\sigma}^n$$

2. *Prediction*: compute $\tilde{\mathbf{u}}_K^{n+1}$ by solving for all $K \in \mathcal{T}$,

$$\frac{|K|}{\delta t}(\rho_K^n \tilde{\mathbf{u}}_K^{n+1} - \rho_K^{n-1}\mathbf{u}_K^n) + \sum_{\sigma\in\mathcal{E}_K} |\sigma|\tilde{\mathbf{u}}_\sigma^{n+1}\rho_\sigma^n u_{K,\sigma}^n + \sqrt{\frac{\rho_K^n}{\rho_K^{n-1}}}\sum_{\sigma\in\mathcal{E}_K}|\sigma|p_\sigma^n\mathbf{n}_{K,\sigma}$$
$$= -\sum_{\sigma\in\mathcal{E}_K}|\sigma|X_\sigma^n(1-X_\sigma^n)\rho_\sigma^n(\mathbf{u}_R)_\sigma^n(u_R)_{K,\sigma}^n \tag{4.2.3}$$

3. *Projection-correction*: compute $\mathbf{u}_K^{n+1}$, p_K^{n+1}, e_K^{n+1}, ρ_K^{n+1} by solving the non-linear system of equations for all $K \in \mathcal{T}$,

   - Velocity update:

$$\mathbf{u}_K^{n+1} = \tilde{\mathbf{u}}_K^{n+1} - \frac{\delta t}{\rho_K^n|K|}\left(\sum_{\sigma\in\mathcal{E}_K}|\sigma|p_\sigma^{n+1}\mathbf{n}_{K,\sigma} - \sqrt{\frac{\rho_K^n}{\rho_K^{n-1}}}\sum_{\sigma\in\mathcal{E}_K}|\sigma|p_\sigma^n\mathbf{n}_{K,\sigma}\right) \tag{4.2.4}$$

   - Mass balance:

$$\frac{|K|}{\delta t}(\rho_K^{n+1} - \rho_K^n) + \sum_{\sigma\in\mathcal{E}_K}|\sigma|\rho_\sigma^{n+1}u_{K,\sigma}^{n+1} = 0 \tag{4.2.5}$$

   - Energy balance:

$$\frac{|K|}{\delta t}(\rho_K^{n+1}e_K^{n+1} - \rho_K^n e_K^n) + \sum_{\sigma\in\mathcal{E}_K}|\sigma|e_\sigma^{n+1}\rho_\sigma^{n+1}u_{K,\sigma}^{n+1}$$
$$+ p_K^{n+1}\sum_{\sigma\in\mathcal{E}_K}|\sigma|F_{K,\sigma}^{n+1} = \tilde{q}_K - \sum_{\sigma\in\mathcal{E}_K}|\sigma|X_\sigma^n(1-X_\sigma^n)\mathcal{L}(p_\sigma^n)\rho_\sigma^n(u_R)_{K,\sigma}^n \tag{4.2.6}$$

   - Equation of state:

$$\left[p_K^{n+1} - (h_L(p_K^{n+1}) - e_K^{n+1})\rho_K^{n+1}\right]\left[\rho_L(h_L(p_K^{n+1})) - \rho_G(p_K^{n+1})\right]$$
$$- \rho_G(p_K^{n+1})\left[\rho_L(h_L(p_K^{n+1})) - \rho_K^{n+1}\right]\mathcal{L}(p_K^{n+1}) = 0 \tag{4.2.7}$$

The non-linear projection-correction step is solved using Newton's method with the non-linear system formed by the energy balance, the mass balance and the equation of state. Between each Newton iteration, the velocity is updated using the velocity update equation.

In practice the Jacobian is automatically generated to Fortran using a Maxima. At each Newton iteration, the linear system is solved using GMRES with ILU(0) preconditioner through the PETSc library.

4.2.2 Chisholm scalar slip

In the GENEPI code, the slip from the Chisholm model is calculated using a fixed point procedure, which is slow and lacks robustness. However, it is possible to use the more effective Newton's method to solve directly s. Let us recall the Chisholm model:

$$s = \min(s_1, s_2) \text{ with } \begin{cases} s_1 = [1 + (r-1)\, X_d]^{1/2} \\ s_2 = r^{1/4} \end{cases} \tag{4.2.8}$$

with $r = \rho_L/\rho_G$. Solving s amounts to determine X_d. Given that $r > 1$ and $X_d \in [0, 1]$, the function $s_1(X_d)$ is monotonic increasing. Therefore, we have:

$$X_d < \frac{\sqrt{r}-1}{r-1} \Rightarrow s(X_d) = s_1(X_d)$$
$$X_d > \frac{\sqrt{r}-1}{r-1} \Rightarrow s(X_d) = s_2$$

As a result, if $s_2 X/(1 + (s_2 - 1)X) \in [(\sqrt{r}-1)/(r-1), 1]$ then $X_d = s_2 X/(1 + (s_2 - 1)X)$. If not, we use Newton's method to determine X_d with the following non-linear function:

$$F(X_d) = X(X_d - 1)(1 + (r-1)X_d)^{1/2} + (1 - X)X_d$$

4.3 Validation tests

4.3.1 Problem setting

We consider the validation problem VE19 from GENEPI test suite presented in [109]. The domain is $\Omega = [x_e, x_s]$ is a simplified representation of the evaporator (see figure 4.1, right). The evaporator has a height of 10 m and a cross section of 1 m^2. The use of buffer zones $[0, x_e]$ at the inflow and $[x_s, L]$ at the outflow allows to deal more smoothly with the boundary conditions of Ω.

The secondary fluid enters the evaporator in liquid state i.e. with $X = 0$ and $h = h_L$. The mass flow rate is $G_e \equiv \rho_e u_e = 1 \times 10^4\,\mathrm{kg \cdot s^{-1} \cdot m^{-2}}$ and the enthalpy $h_e = 1.2 \times 10^5\,\mathrm{J \cdot kg^{-1}}$. This yields the following boundary conditions:

$$\begin{aligned} \rho|_{x_e} &= \rho_L(p_1)\rho_G(p_1)/(\rho_G(p_1) + (\rho_L(p_1) - \rho_G(p_1))X) \text{ with } X = (h_e - h_L(p_1))/\mathcal{L}(p_1) \\ u|_{x_e} &= G_e/\rho_e \\ e|_{x_e} &= h_e - p_1/\rho_e \\ (\partial_n p)|_{x_e} &= 0 \end{aligned}$$

with p_1 the pressure in the first cell above $x = 0$. At the outflow, the fluid exits with pressure $p_s = 9 \times 10^5$ Pa. Consequently the boundary conditions are defined as such:

$$\begin{aligned} (\partial_n \rho)|_{x_s} &= 0 \\ (\partial_n u)|_{x_s} &= 0 \\ (\partial_n e)|_{x_s} &= 0 \\ p|_{x_s} &= p_s \end{aligned}$$

Along the evaporator, i.e. for $x_e < x < x_s$, the fluid is heated with a constant flux $q = 4 \times 10^5\,\mathrm{W \cdot m^{-3}}$. The slip coefficient is either constant and taken to $s = 2$ or calculated with

the Chisholm model. At $t = 0$, the flow is initialized with a constant solution:

$$\begin{aligned} h^0 &= h_e \\ p^0 &= p_0 \\ X^0 &= (h_e - h_L(p_0))/\mathcal{L}(p_0) \\ \rho^0 &= \rho_L(p_0)\rho_G(p_0)/(\rho_G(p_0) + (\rho_L(p_0) - \rho_G(p_0))X^0) \\ u^0 &= G_e/\rho_e \\ e^0 &= h_e - p^0/\rho^0 \\ \rho^{-1} &= \rho^0 \end{aligned}$$

The initial state is located by the orange mark on the enthalpy-pressure diagram of figure 4.5. Note that the initial state of the benchmark is determined with CEA models of Freon R-114 whereas the diagram of figure 4.5 was produced by extracting the data from the original diagram of E.I. du Pont de Nemours, thereby introducing errors. This may explain why the orange mark is not located exactly on the saturated liquid curve.

4.3.2 Numerical results

The problem is discretized on a 550 cell grid and the timestep is chosen to $\delta t = 11/550 = 0.2$. The final pressure and enthalpy solutions obtained with our scheme and with GENEPI are shown on figure 4.2. Our results exhibit excellent agreement with the solution from GENEPI. While further validation tests would be required, we can expect our scheme to be able to handle every flow problem dealt by GENEPI.

In addition, the pressure-correction scheme is able to compute a transient. The transient for the constant slip and for the Chisholm model are respectively given in figures 4.3 and 4.4 (top). On the enthalpy-pressure diagrams, the transient consists of small variations around the initial point on the saturated liquid curve. The convergence to the steady state is being accelerated on some subsets of the parametric curve $(h(t), p(t))$. The steady state is clearly identified by the very small variations between successive states. The evolution of the thermodynamic variables on figures 4.3 and 4.4 (bottom) reveal a complex transient, though this does not bring instabilities to the convergence to the steady solution.

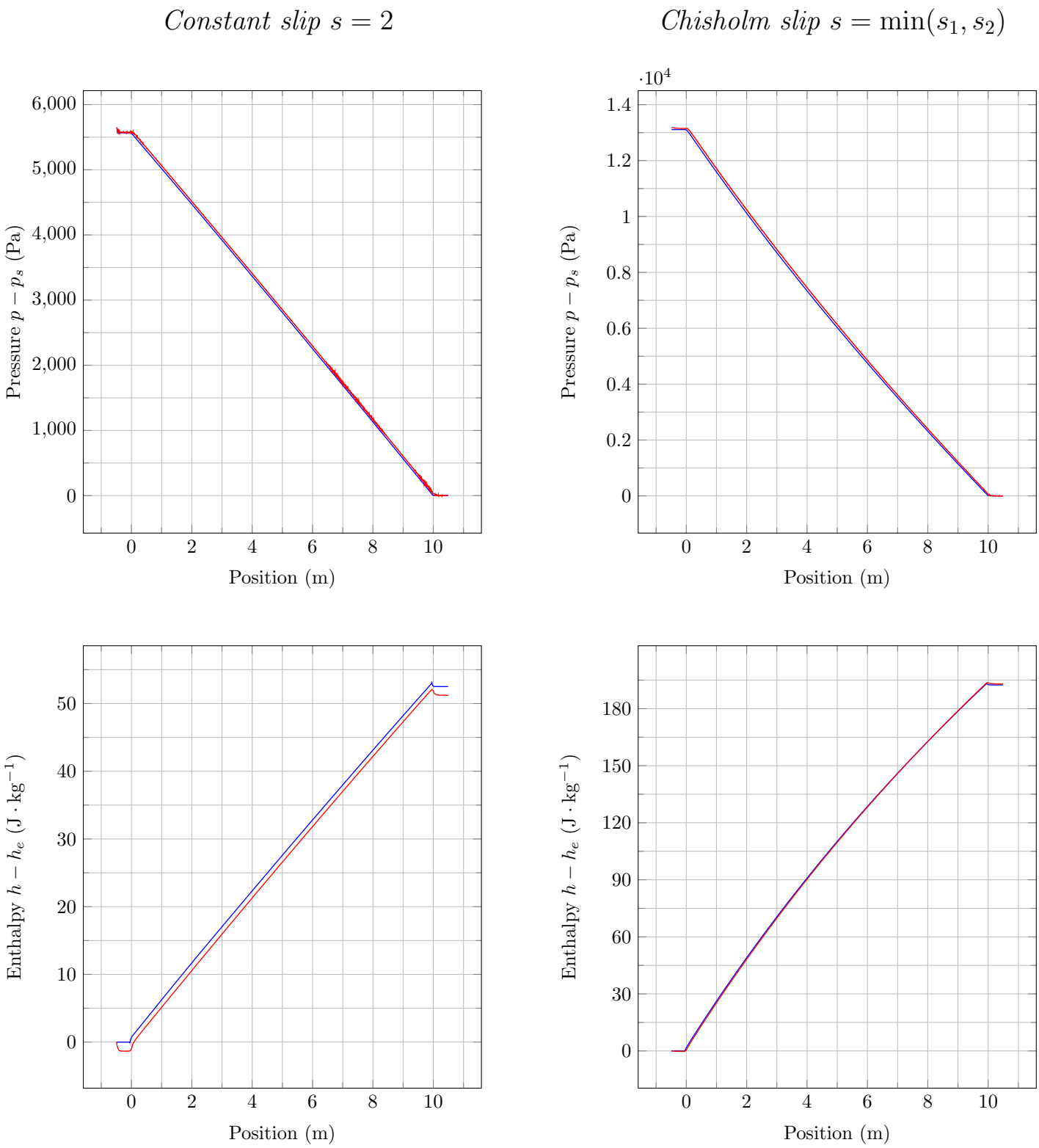


FIG. 4.2 – *Comparison of the steady state pressure and enthalpy solutions obtained with our projection algorithm (in blue) and GENEPI (in red).*

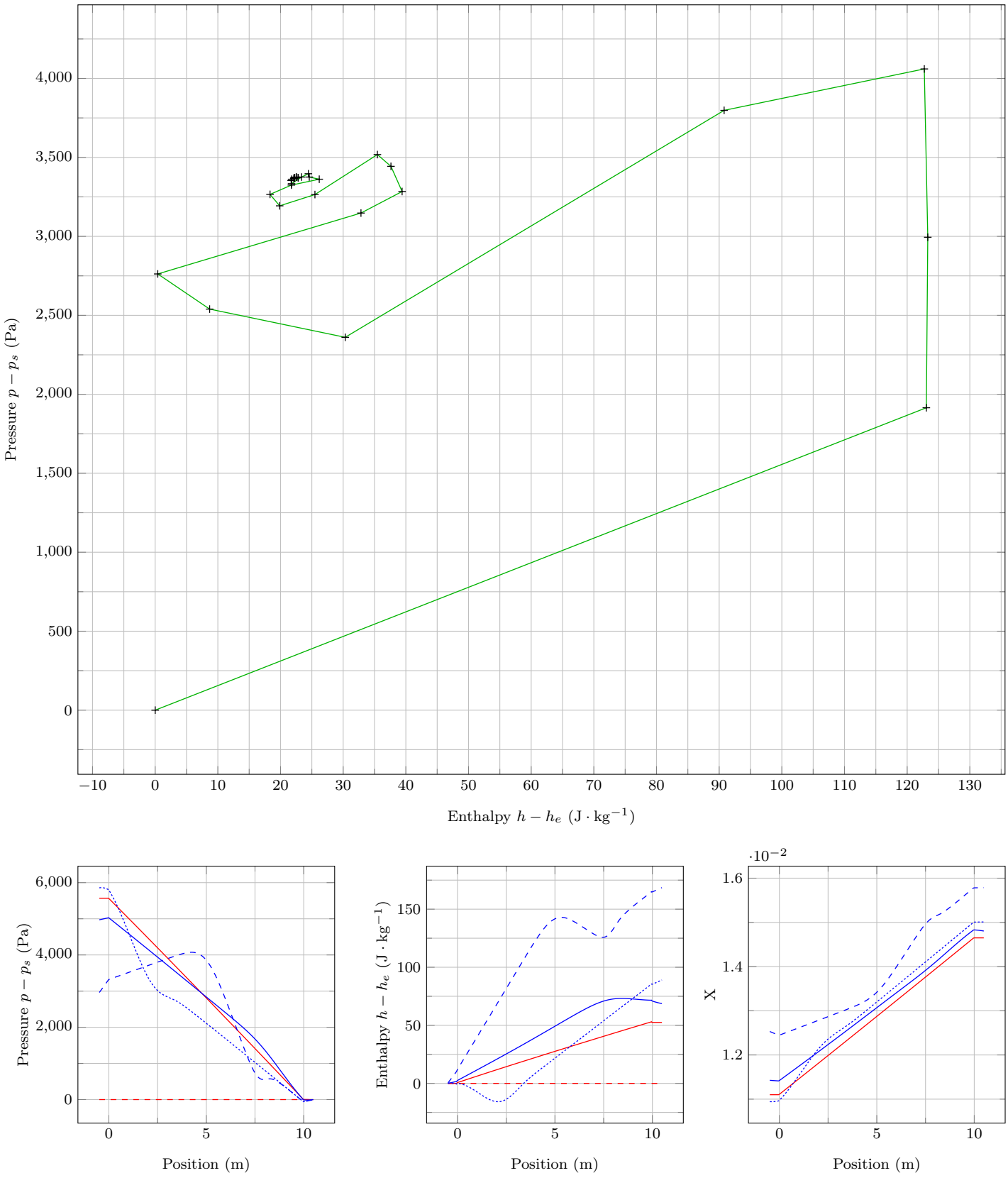


FIG. 4.3 – *Transient obtained with our projection algorithm for constant slip $s = 2$. On the enthalpy-pressure diagram, the black marks denote the flow state at the half height of the evaporator, taken at 31 evenly spaced timesteps from $t = 0$ to $t = 3000\delta t$. The bottom curve feature thermodynamic variables at initial (dashed curve) and final (solid curve) times in red and intermediate times (dashed for $t = 300\delta t$, dotted for $t = 600\delta t$ and solid for $t = 900\delta t$) in blue.*

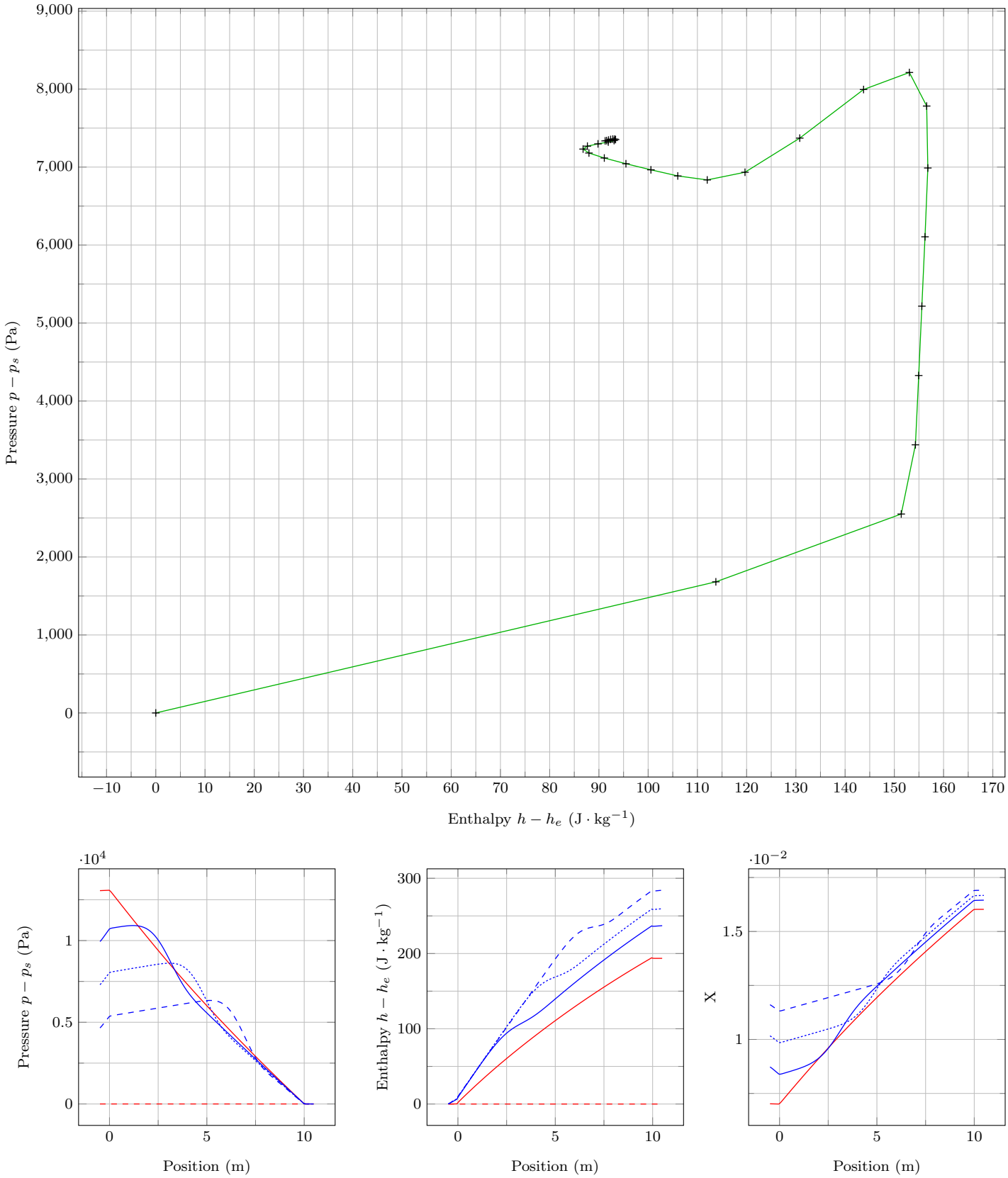


FIG. 4.4 – *Transient obtained with our projection algorithm for the Chisholm slip* $s = \min(s_1, s_2)$. *On the enthalpy-pressure diagram, the black marks denote the flow state at the half height of the evaporator, taken at 31 evenly spaced timesteps from* $t = 0$ *to* $t = 2000\delta t$. *The bottom curve feature thermodynamic variables at initial (dashed curve) and final (solid curve) times in red and intermediate times (dashed for* $t = 400\delta t$, *dotted for* $t = 600\delta t$ *and solid for* $t = 800\delta t$*) in blue.*

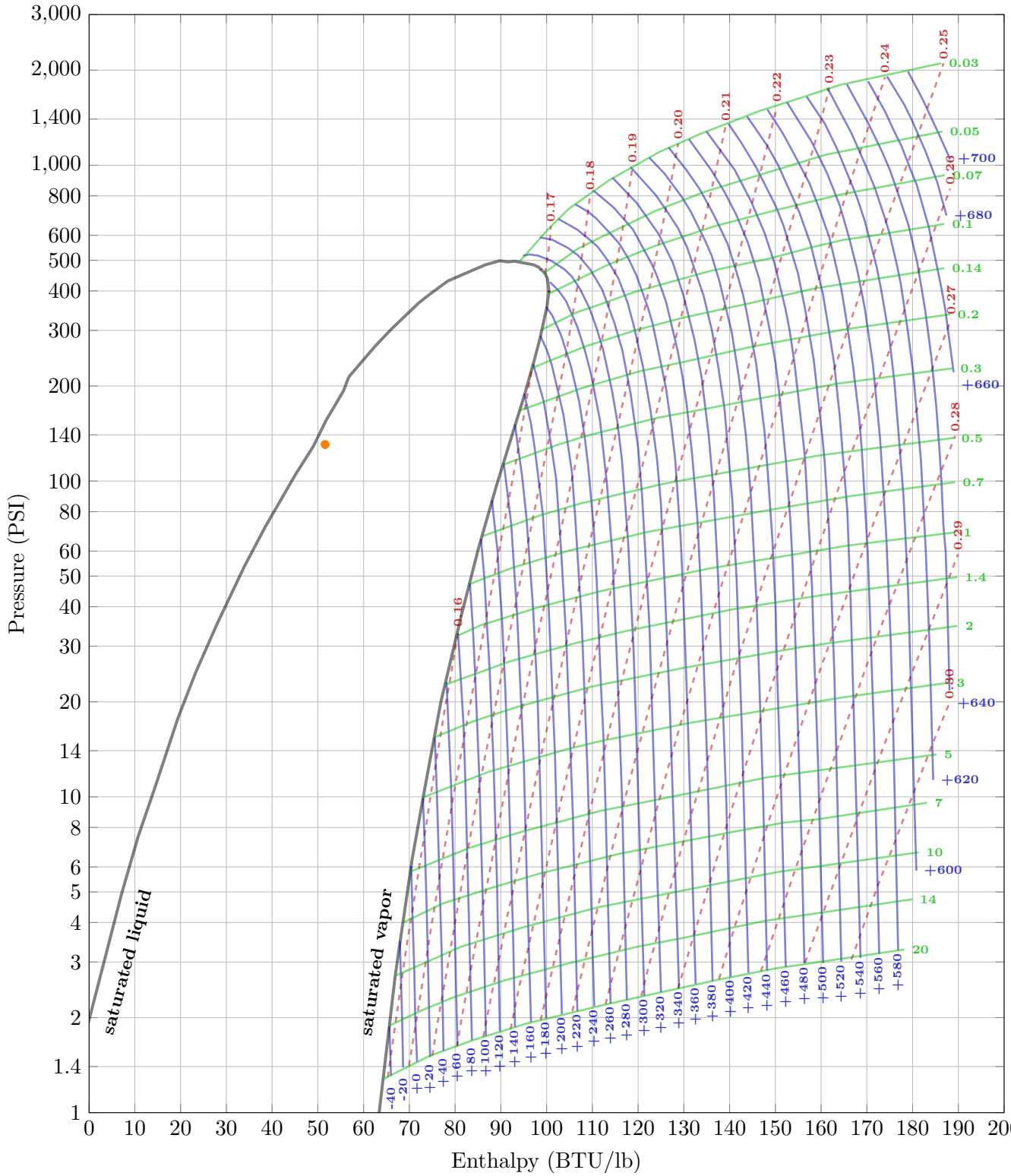


FIG. 4.5 – *Pressure-Enthalpy diagram for Freon R-114. Original data from E.I. du Pont de Nemours (1944), see for instance [84, p. 82]. The lines of constant entropy* (BTU/lb · °F) *are in red (15 values,* $S \in [0.16, 0.30]$*), the lines of constant volume* (ft^3/lb) *in green (18 values,* $V \in [0.03, 20]$*) and the lines of constant temperature* ($°F$) *in blue (38 values,* $T \in [-40, 700]$*). The orange mark locates the initial state in the verification benchmark.*

Part II

Adaptive Mesh Refinement

Chapter 5

ADAPTIVE GRIDS

*The purpose of adaptive mesh refinement is to adapt the space resolution of a numerical method by decreasing the mesh size (*h-refinement*) and possibly increasing the order of the scheme as well (*hp-refinement*) in regions where a higher spatial accuracy is desired. This leads to considerable savings in memory and computational effort, and it allows computations with a higher accuracy than the hardware limitations would permit. The scope of this Chapter is limited to h-refinement without time refinement, i.e. the problem is solved on the whole adaptive grid at every time step.*

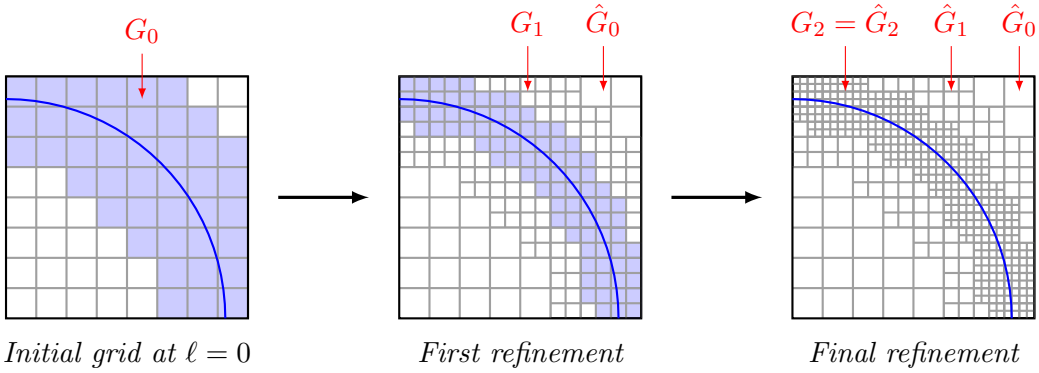


FIG. 5.1 – *Example of cell-by-cell refinement with respect to a level-set function.*

An example of adaptive grid is shown on figure 5.1. The computational domain is $\Omega = \Omega_0 = [0,1] \times [0,1]$. There are three *levels* $0 \leq \ell \leq 2$ matching regions of different refinement. The levels are indexed from the coarsest with a discretization step $h_0 = 1/8$ to the finest with $h_2 = 1/32$. The refinement factor $n_{\text{ref}} = 2$ is assumed to be constant, hence $h_\ell = h_{\ell+1} \cdot n_{\text{ref}}$. A grid with space step h_ℓ is denoted G_ℓ and its associated subdomain Ω_ℓ.

The refinement procedure is recursive. Starting from a *base grid* G_0 at level $\ell = 0$, a refinement criteria is evaluated cell-wise. This criteria can be arbitrary (eg. user-defined region) or automatic using a refinement indicator. The indicator may depend on an error estimation (eg. using Richardson extrapolation), on flow features (eg. shocks and contacts) or on geometric features (eg. level-set function, in this example). The set of cells tagged for refinement (in light blue on figure 5.1) defines the region of G_ℓ which will be replaced by a finer grid $G_{\ell+1}$. The new grid is the union of the previously generated grids at coarser levels $\{\hat{G}_k = G_k \setminus G_{k+1}; 0 \leq k \leq \ell\}$ and the newly generated grid $G_{\ell+1}$. The procedure is repeated on $G_{\ell+1}$ until the finest

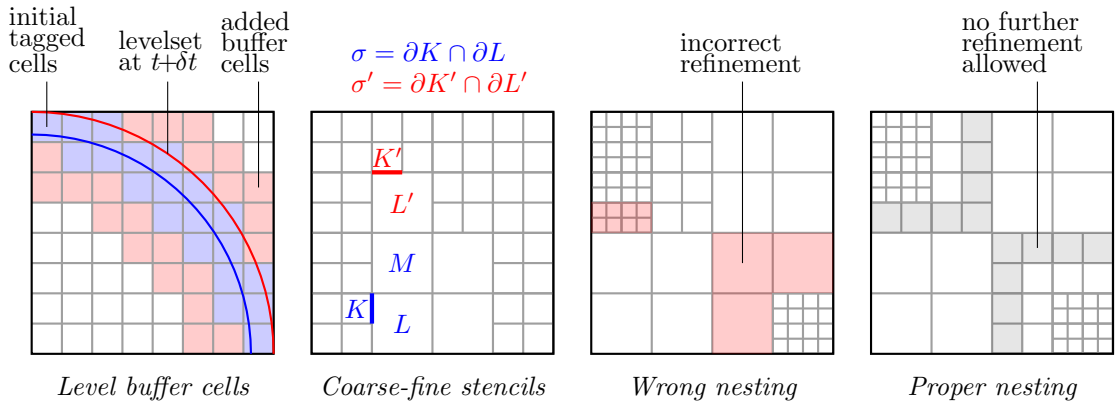


FIG. 5.2 – *Constrains on adaptive grids.*

refinement level $\ell_{\max} = 2$ is reached.

The grid G_ℓ is the *level grid* associated with level ℓ. The grid $\hat{G}_\ell$ is the *composite grid* associated with level ℓ, defined as $\hat{G}_\ell = G_\ell \setminus G_{\ell+1}$. The composite grid on the whole computational domain is denoted $\hat{G} = \cup_{\ell \geq 0} \hat{G}_\ell$.

While the adaptive grid should represent as closely as possible the refinement indicators, the resulting composite grid must satisfy several constrains (figure 5.2). Some are inferred by the refinement procedure, others have to be enforced separately:

- *Proper nesting*: only coarse-fine interfaces between two consecutive levels ℓ and $\ell + 1$ are allowed. We may require a layer of more than one cell between two levels ℓ and $\ell + 2$.
- *Level buffers*: when the refinement criterion is expected to change at the next time step $t + \delta t$, for instance a level-set (or a shock), further cells should be marked for refinement so that the level-set does not leave the finest grid when the next time step is reached.
- *Stencil constrains*: stencils at coarse-fine interfaces may not be defined. On figure 5.2, the stencil involves the two coarse and fine neighboring cell and an additional cell in the tangential direction to the face. For face σ, the stencil is $\{K, L, M\}$ but for face σ' we miss a coarse cell in the tangential direction. Therefore, depending the coarse-fine stencils, some tag patterns must be avoided.

In this work, we choose to focus on a family of adaptive mesh refinement methods called *Structured AMR* (SAMR) which will be dealt in depth in the next chapters. In this first chapter after presenting the advantages and drawbacks of the other important family of adaptive mesh refinement methods known as *cell-by-cell refinement*, we turn to the generation of SAMR grids and the practical issues which were faced in this work.

The most natural way to refine a level ℓ is cell-by-cell: each cell tagged for refinement would be replaced by n_{ref}^d cells from the finer level. This results in a single grid which can be described using a hierarchy of cells grouped by clusters of n_{ref}^d. This adaptive grid matches exactly the regions marked for refinement. On the other hand with Structured AMR, the adaptive grid is represented as the superposition of level grids, which are defined as unions of Cartesian grids. This yields a hierarchy of nested grids, which may match regions larger than those marked for refinement.

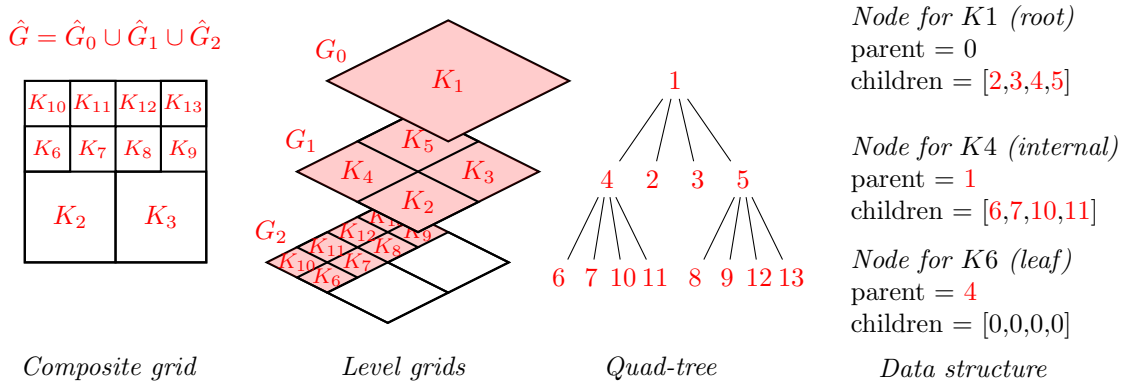


FIG. 5.3 – *Quadtree representation of a cell-by-cell adaptive grid.*

5.1 Single hierarchical grid

The first *cell-by-cell* AMR approaches — often referred to as *octree refinement* after their underlying data structures — were introduced in the 1991 paper of D. Young & al. [131] and in the PhD thesis of W. Coirier in 1994 [44]. While refinement and de-refinement are quite straightforward with these hierarchical grids, managing efficiently the connectivity in regard to the memory use and the asymptotic complexity is much more challenging. In addition usual numerical methods may need to be significantly modified to use such computational grids (eg. multigrid methods).

5.1.1 Representation

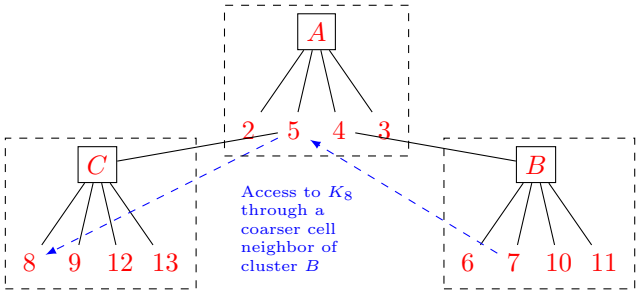
The grid is represented by a spatial hierarchical datastructure: a *binary tree* in 1D, a *quad-tree* in 2D and an *octree* in 3D. This datastructure arises naturally during the refinement process. The *root* of the tree is a single cell at the coarsest level. When refined, this cell yields 2^d finer cells which may be themselves refined as well. If a cell is further refined, it simply plays the role of an *internal node* in the tree. The *leafs* of the tree, which are not refined, are the cells effectively defining the computational grid.

This representation allows the refinement procedure to match exactly the cells tagged for refinement, which is generally not the case with the SAMR approach. Moreover the octree datastructure features the connectivity between a given cell and the coarser cell its belongs to (parent) and with the finer cells nested in it (children). However the overhead introduced by this connectivity information (pointers to parent and children) is very high compared to SAMR adaptive grids. Indeed, the connectivity is stored for every cell while in the SAMR approach, this information is stored for rectangular clusters of cells. In 2D, it amounts to at least 5 words of memory per cell (2^d for children pointers and 1 for parent pointer).

In addition, implementing numerical methods with cell-by-cell refinement would require access to further connectivity informations which are not inferred by the original octree datastructure:

1. For a given cell $K \in G_\ell$, loop over all its neighbors $\{L \in G_\ell; K|L \in \mathcal{E}_K\}$ at the same level

2. Find all the cells of any level which are contained in an arbitrary neighborhood



Graph of FTT implementation

Cluster A for $[K2, K3, K4, K5]$
parent = 0
children = [0,C,B,0]
neighbors = [0,0,0,0]

Cluster B for $[K6, K7, K10, K11]$
parent = 4
children = [0,0,0,0]
neighbors = [5,0,0,2]

Data structure

FIG. 5.4 – *FTT representation of the cell-by-cell adaptive grid introduced in figure 5.3.*

Realizing these operations efficiently is difficult. Indeed, the path for accessing every single neighbor of a cell may be as long as twice the depth of the tree (eg. the path from K_7 to its neighbor K_8 is $7 \to 4 \to 1 \to 3 \to 8$). One possible solution would be to store pre-computed connectivity information with respect to the neighbors in every cell. But this would increase the memory use to 9 words per cell ($2d$ additional words for neighbor pointers).

5.1.2 Transversal search

Transversal search for operation (1) can be performed efficiently and with a smaller overhead using *Fully Threaded Trees* (FTT) introduced by A. Khokhlov [88] instead of the original octree datastructure. Regarding operation (2), an efficient neighbor search within an arbitrary region can be performed using *Alternating Digital Tree* for searching over the octree [29].

Fully Threaded Tree

The FTT datastructure addresses several issues of the original octree representation [88]:

- Fast access to neighboring cells at the same level or at the coarser level
- Reduce the memory overhead of the tree data structure

In cell-by-cell adaptive grids a coarse cell is refined into 2^d finer cells, so it would make sense to store connectivity information between coarse cells and clusters of 2^d finer cells. In a FTT, cells are grouped by 2^d as shown on figure 5.4. This provides a direct access to neighbor information within the cluster. For accessing neighboring cells outside the cluster, either at the same level or at a coarser level, the pointers to the $2d$ neighboring coarser cells are stored. Likewise, each cells features a pointer to it child cluster. Thanks to this clustering, the overhead is lower than one word per cell in 2D (in fact $(1 + (2d + 1)/2^d)N$ words for N cells).

Alternating Digital Tree

Alternating Digital Tree datastructure, discussed in detail in [2] provide a more effective way to handle the connectivity of the adaptive grid.

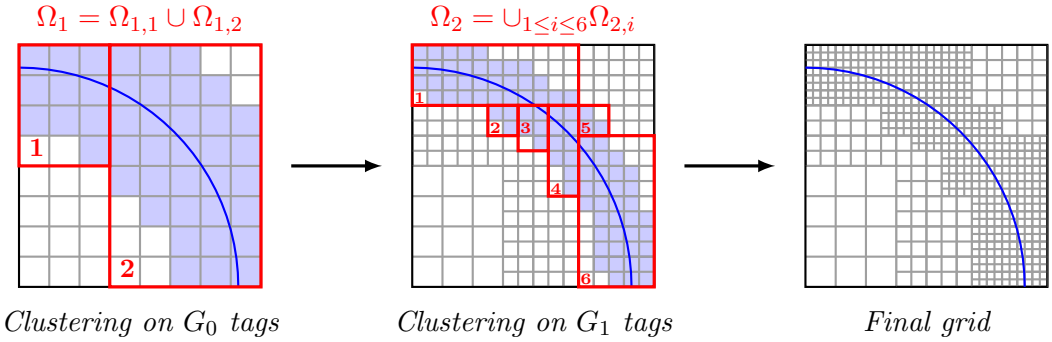


FIG. 5.5 – *Example of patch-based refinement with respect to a level-set function.*

5.2 Hierarchy of nested grids

Patch-based AMR also known as *Structured AMR* (SAMR) methods were introduced by M. Berger with J. Oliger [26] in 1984 and with P. Colella [25] in 1989. Each level grid G_ℓ is stored independently as an union of Cartesian grids $G_{\ell,i}$ called *patches*: $G_\ell = \cup_i G_{\ell,i}$. It makes easier the implementation of existing numerical methods for single grids as each level grids can be accessed independently. Unlike cell-by-cell AMR, level grids usually cover regions larger than those originally defined by the cells tagged for refinement. In that case the refinement is less accurate and the number of unknowns for each level is not optimal. Moreover as level grids are nested further unknowns (in overlapping regions between consecutive levels) which do not belong to the composite grid have to be taken into account.

5.2.1 Representation

Description

The refinement procedure is very different from cell-by-cell AMR. Given a set of cells marked for refinement on a level grid G_ℓ, a clustering algorithm generates a set of Cartesian grids (*patches*) $G_{\ell+1,i}$ so as to match as closely as possible the tagged cells.

Matching exactly the cells tagged for refinement — and obtaining thereby a composite grid identical as that of cell-by-cell refinement — comes at the expense of having to manage a large number of patches when solving a problem on this adaptive grid. On the other hand, allowing to match more loosely the tagged cells would yield fewer patches but more unknowns. On the example of figure 5.5, there are 40 additional cells on G_1 (24% overhead) and 112 additional cells on G_2 (30% overhead). A clustering without overhead would generate much more patches. To control the balance between the number of patches needed and the number of additional cells, a quantity called *efficiency* is introduced. It is defined as the ratio of the number of tagged cells over the total number of cells of a level grid. On figure 5.5, the respective efficiencies of level grid G_1 and G_2 are 77% and 78%.

The procedure is then repeated on the newly generated level grid $G_{\ell+1} = \cup_i G_{\ell+1,i}$.

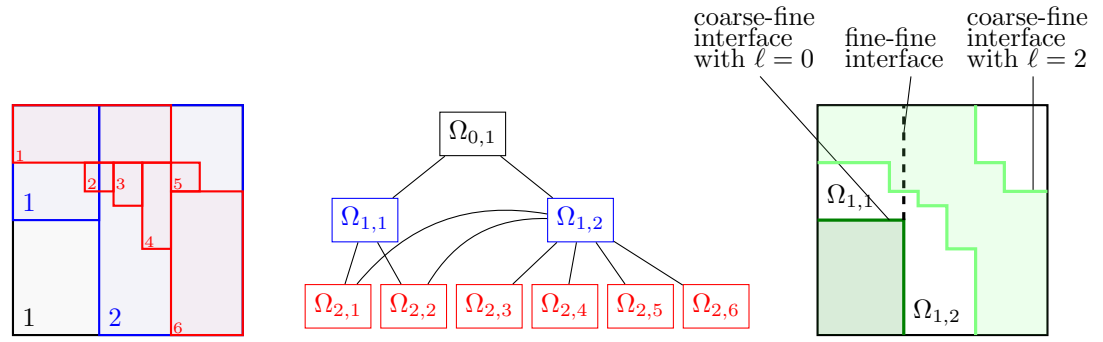


FIG. 5.6 – *Connectivity information for SAMR grids.*

Connectivity

In contrast to cell-by-cell refinement, the SAMR grid representation does not feature any connectivity information. When implementing numerical methods on SAMR adaptive grids, the following connectivity information should be available:

- *Nesting between patches*: for inter-level communication between two consecutive levels on overlapping regions of *level grids*. In the graph of figure 5.6, a node $\Omega_{\ell,i}$ is connected to both overlapping coarser patches $\{E \in (\Omega_{\ell-1,j})_j; E \cap \Omega_{\ell,i} \neq \{\emptyset\}\}$ and overlapping finer patches $\{E \in (\Omega_{\ell+1,j})_j; E \cap \Omega_{\ell,i} \neq \{\emptyset\}\}$.

- *Coarse-fine interfaces*: for coupling two consecutive levels at the interface between their respective *composite grids*. The boundary of a patch $\Omega_{\ell,i}$ may intersect a coarse-fine interface with the coarser level: $\{\partial\Omega_{\ell,i} \cap \partial\hat{\Omega}_{\ell-1}\}$. The faces of the cells of $G_{\ell,i}$ may intersect a coarse-fine interface with the finer level: $\{\Gamma \in \{\partial K, K \in G_{\ell,i}\}; \Gamma \cap \hat{\Omega}_{\ell+1} \neq \{\emptyset\}\}$.

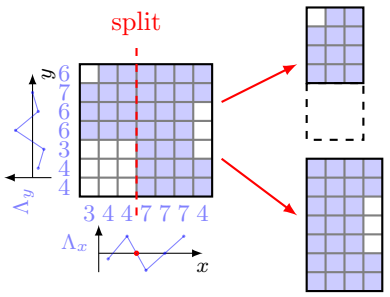
- *Fine-fine interfaces*: level grids are decomposed into patches hence the need for domain decomposition methods for coupling patches at fine-fine interfaces. The fine-fine interfaces at level ℓ are defined as $\{\cap_i \Omega_{\ell,i}\}$.

5.2.2 Grid generation

Clustering method

Most clustering algorithms for SAMR are derived from the original algorithm introduced by M. Berger and I. Rigoutsos in 1990 [24]. This is a recursive algorithm which uses ideas from artificial intelligence and medical imaging [53, 15] for feature detection. The objective is to split a given patch at a well chosen position such that the two newly generated patches match more closely the tagged cells. This can be achieved by isolating blocks of untagged cells and having them at the boundaries of the new patches, which can then be trimmed down.

These blocks are identified using *image signatures*. Let us denote T_{ij} the two dimensional tag field, set to one if cell K_{ij} is tagged and zero else. In 2D, the horizontal and vertical signatures



1. *Compute the signatures* S_x, S_y

$S_x(i) = \sum_j T_{ij}$

2. *Evaluate the Laplacian of signatures* Λ_x, Λ_y

$\Lambda_x(i) = S_x(i+1) - 2S_x(i) + S_x(i-1)$

3. *Find the steepest zero-crossing of* Λ

4. *Split and trim*

FIG. 5.7 – *Clustering by signatures.*

are defined as

$$S_x(i) = \sum_j T_{ij}$$
$$S_y(j) = \sum_i T_{ij}$$

Let us start with the patch shown on figure 5.7. This patch inherits the tagged cells of its level grid (in blue). In order to detect blocks of untagged cells from the signatures, M. Berger and I. Rigoutsos use a simplification of the Marr–Hildreth edge detection method [99]. The sharpest variation of the edge of the tag field shape is estimated at the *steepest zero-crossing* of the Laplacian of the signatures. This Laplacian is computed using the classical finite-difference formulae:

$$\Lambda_x(i) = S_x(i+1) - 2S_x(i) + S_x(i-1)$$
$$\Lambda_y(j) = S_y(j+1) - 2S_y(j) + S_y(j-1)$$

On figure 5.7 the steepest zero crossing of Λ is located along the horizontal axis between $i = 3$ and $i = 4$. The patch is then split at this position into two new patches. The new left patch is finally trimmed down by removing the 3×3 block of untagged cells. The algorithm is then applied to the left and right new patches until the desired efficiency criterion is met.

Suboptimal clusterings

Some specific tag fields as presented in [24] can lead to a non-optimal splittings. A first example of such tags is given in figure 5.8. When the standard Laplacian is used for edge detection, a non-optimal choice of partition is made. Out of the two choices for split, either between cells $(7, 1)$ and $(7, 2)$ or between cells $(1, 6)$ and $(1, 7)$, the second one is selected. The issue is that the signatures only provide an "external" view of the tag pattern. It is not possible using solely the signatures to predict that the split between cells $(1, 6)$ and $(1, 7)$ will come through the large 7×7 cell block. A possible workaround proposed by M. Berger and I. Rigoutsos in [24] is to

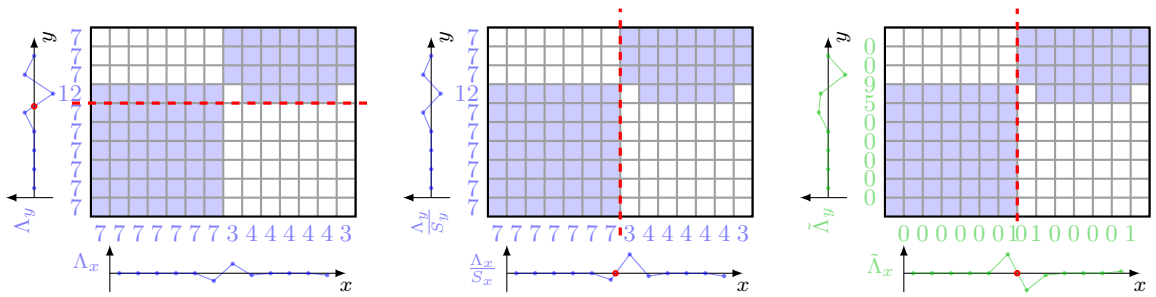


FIG. 5.8 – *Addressing the shortcomings of the clustering algorithm: irrelevant split.*

use a *scaled* Laplacian, defined as:

$$\Lambda_x(i) = \frac{S_x(i+1) - 2S_x(i) + S_x(i-1)}{S_x(i)}$$
$$\Lambda_y(j) = \frac{S_y(j+1) - 2S_y(j) + S_y(j-1)}{S_y(i)}$$

It balances the choices between the two space dimensions and leads to the correct choice for splitting the patch. This improvement works on this specific example but it is not a definite solution, as the indicator used for edge detection is still calculated upon the signatures. A second solution was proposed by the same authors: *"compute the sum of the absolute value of the gradient, and difference the results to get the second derivative"*. We the define the *gradient signatures as*:

$$G_x(i+1/2) = \sum_j |T_{i,j} - T_{i+1,j}|$$
$$G_y(j+1/2) = \sum_i |T_{i,j} - T_{i,j+1}|$$

The modified Laplacian is obtained from these quantities as:

$$\tilde{\Lambda}_x(i) = G_x(i+1/2) - G_x(i-1/2)$$
$$\tilde{\Lambda}_y(j) = G_y(j+1/2) - G_y(j-1/2)$$

Another problematic tag field is shown on figure 5.9. From "outside" the patch, relying solely on signature information, it is not possible to identify any edge. Nevertheless the ratio between the number of tagged cells and the number of total cells indicates that an edge does exist in this tag pattern. Using the modified Laplacian $\tilde{\Lambda}$, the internal variations of the shape can be detected. Another, simpler alternative proposed in [24] consists in splitting the patch in two equal grids, a process call *bisection*.

In practice, we will use the standard Laplacian with a bisection when necessary.

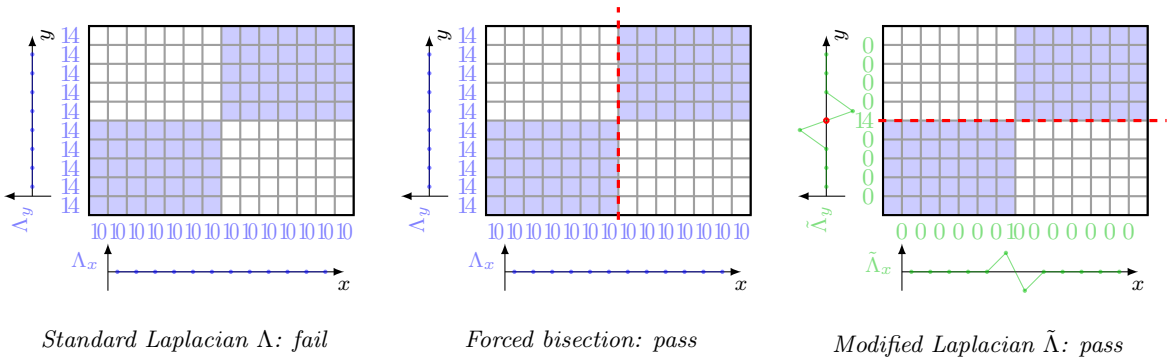


FIG. 5.9 – *Addressing the shortcomings of the clustering algorithm: edge not detected.*

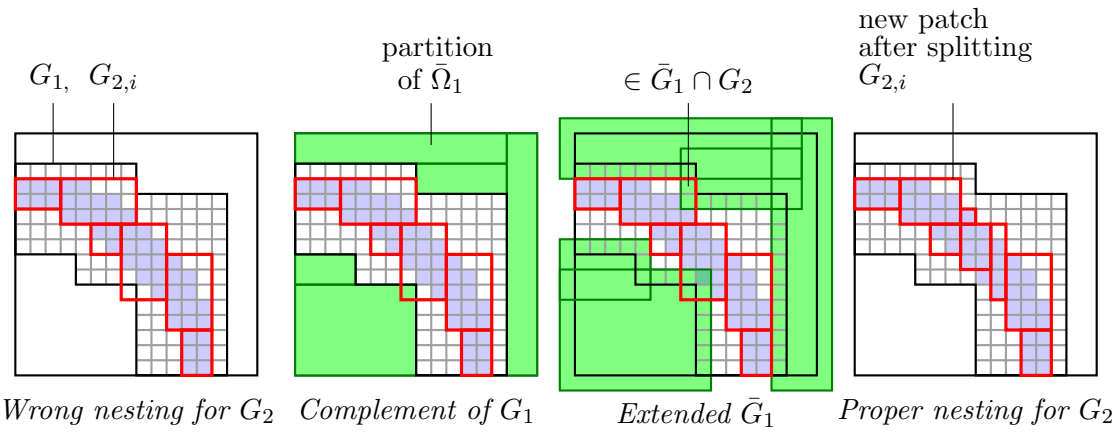


FIG. 5.10 – *Enforce nesting between levels.*

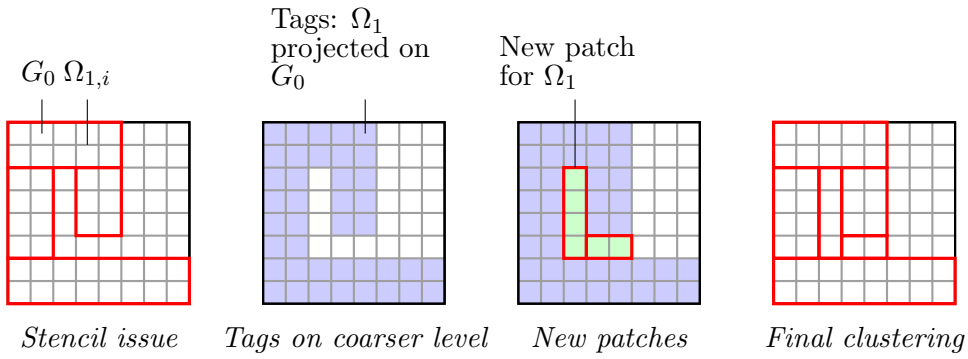


FIG. 5.11 – *Fix clustering incompatible with coarse-fine stencils.*

Incompatible clusterings

The generated composite grids must be compatible with the composite discretization stencils. For instance, when computing the edge value of a gradient using the composite discretization presented in next chapter, two coarse cells in the tangential directions are required. Therefore, in the first clustering attempt of figure 5.11, such stencil problems are encountered at cell $(3, 6)$. A possible workaround would be fix the tag field prior to the clustering. However, an *a priori* fix will never be a systematic solution for this problem: the clustering algorithm with an efficiency lower than one generates a different level shape — slightly larger — that the one defined by the tag field. The generated level has to be fixed *a posteriori*, using for example the following procedure:

1. Generate a tentative clustering using the tags defined on G_0.

2. Define a tag field on G_0 by projecting the tentative clustering on G_0.

3. Detect the untagged coarse cells responsible for the stencil definition issues. These cells are tagged into a new tag field.

4. Use the clustering algorithm with efficiency set to one to generate patches defined by the new tag field.

5. Append the newly generated patches to the tentative clustered level.

Another problem arising when clustered levels are generated with an efficiency lower than one is the proper nesting between levels, an example of which is depicted on figure 5.10. On this example, we require one buffer cell between consecutive levels and therefore two patches should be re-generated. We use an idea proposed by R. Deiterding in [51] which consists in using the complement of the tentative clustered level in order to identify the regions of it yielding to nesting problems:

1. Generate a tentative clustering using the tags defined on G_0.

2. Define the complement of the tentative level by using a modified clustering procedure with efficiency set to one, which would consider the "complement" of the original tag fields

3. Extend every patch of the complement level according to the width of the buffer zone required between two consecutive levels (one in the example of figure 5.10)

4. Look for the overlapping regions within each patch of the original clustered level and the complement level. This defines a new tag field on individual patches which matches in each individual patch the cells conflicting with the "proper nesting" condition.

5. On each problematic patch of the tentative clustering, the tagged cells define the region which must be "removed" from the patch. On each patch use the clustering algorithm with the new tag field, with efficiency set to one, to partition each of those patches.

6. Replace the problematic patches by their new partition.

Controlling patch shapes

The original clustering algorithm introduced by M. Berger and I. Rigoutsos features a control over the shape of the entire level grid, not on the patches taken individually. It may be of interest to have such control, so as to avoid very small patches or "highly flattened" patches. In addressing these two issues, three criteria are introduced, defined for a given patch, an associated tag field and a splitting position:

1. *rvmin*: the ratio between the volume of a newly generated patch and the parent patch
2. *vmin*: the volume of a newly generated patch, normalized by the space discretization step
3. *lmin*: the minimum of the lengths of a newly generated patch along each space dimension, normalized by the space discretization step

Prior to assessing the optimal splitting for a given patch, an algorithm pre-computes these three quantities for every possible split position. Then these criteria are used to discard potential splitting positions based on the zeros crossings of the Laplacian of the signatures. The algorithm for pre-computing the *vmin* quantity is given below:

```
    PreComputeVmin(G, tag)
 1: v1(:) ← 0
 2: for d ← 1 to dim(G) do
 3:     pmin(:) ← 1
 4:     pmax(:) ← 0
 5:     i ← sum(size(G, 1 : d − 1))
 6:     for j ← 1 to dim(G) do
 7:         i ← i + 1
 8:         [pmin, pmax] ← ShapeBounds(G, ref1, size1, d, j)
 9:         v1(i) ← j
10:         for k ← 1 to dim(G) do
11:             if k ≠ d then
12:                 v1(i) ← v1(i) ∗ (pmax(k) − pmin(k) + 1)
13: v2(:) ← 0
14: for d ← 1 to dim(G) do
15:     pmin(:) ← 1
16:     pmax(:) ← 0
17:     for j ← dim(G) to  step −1 do
18:         i ← sum(size(G), 1 : d − 1) + j − 1
19:         [pmin, pmax] ← ShapeBounds(G, ref1, size1, d, j)
20:         v2(i) ← size(G, d) − j + 1
21:         for k ← 1 to dim(G) do
22:             if k ≠ d then
23:                 v2(i) ← v2(i) ∗ (pmax(k) − pmin(k) + 1)
24: for i ← 1 to length(rv1) do
25:     vmin(i) ← min(v1(i), v2(i))
    return vmin

    ShapeBounds(G,j,d)
 1: ref1(:) ← localRef(G)
```

```
2: ref1(d) ← ref1(d) + j−1
3: size1(:) ← size(G)
4: size1(d) ← 1
5: for cell K in subregion (ref1,size1) of baseGrid(G) do
6:     if tag(K)≠0 then
7:         for i←1 to dim(G) do
8:             if i≠d then
9:                 if pmax(i) < pmin(i) then
10:                    pmax(i) ← index(K,i)
11:                    pmin(i) ← index(K,i)
12:                else
13:                    if index(K,i) > pmax(i) then
14:                        pmax(i) ← index(K,i)
15:                    if index(K,i) < pmin(i) then
16:                        pmin(i) ← index(K,i)
   return [pmin,pmax]
```

The function PreComputeVmin returns the array vmin(:) indexed with the set $\cup_{i=1}^{\dim(G)} J_i$, $\text{length}(J_i) = \text{size}(G, i)$. The local index $k \in J_d$ is referring to a splitting of the parent patch along the space dimension d between indices k and $k+1$. It matches the global index $j = \sum_{i=1}^{d-1} \text{length}(J_i) + k - 1$. The value vmin($j$) gives the minimal volume among the two patches that would be generated from this splitting, *including the trimming* of the latter. The function ShapeBounds returns the minimal and maximal indices in the parent patch which bound a new patch candidate along the selected space dimension. In practice, the selection of the splitting position is as follows:

1. Most relevant zero crossing of the Laplacian of the signatures meeting *vmin*, *rmin* and *rvmin* criteria

2. If (1) fails, most relevant zero crossing of the Laplacian of the signatures meeting *vmin*, *rmin* criteria

3. If (2) fails and if the signatures match the pathologic case illustrated in figure 5.9, perform a bisection

4. Else, no further split and the current patch is a final patch for the clustered level

5.2.3 Examples

Our version of the clustering algorithm is tested on the five original clustering problems presented in [24]. In each of those tests whose results are given in figures 5.12, 5.13 and 5.14 three level grids are generated: on the left (a) with rvol = 0 and eff < 1, in the middle (b) with rvol > 0 and eff < 1 and on the right (c) with eff = 1. The parameters for each of those are specified in table 5.1. The input parameters are "eff" (minimal efficiency required in any final patch), and ("rvmin","vmin","lmin") presented earlier. The output indicators are eff-loc (worst efficiency among all patches), "eff-lev" (global efficiency achieved on the level grid), "npatch" (the total number of patches).

In tests (a) and (b), the clustering algorithm manages to generates levels with a global efficiency greater than 0.8 while preserving a much lower number of patches compared to the

Test	eff	rvmin	vmin	lmin	eff-loc	eff-lev	npatch
1a	0.7	0	4	2	0.79	0.91	11
1b	0.75	3/4	4	2	0.75	0.84	9
1c	1	0	0	0	1	1	31
2a	0.7	0	4	2	0.81	0.90	11
2b	0.7	1/3	4	2	0.63	0.84	13
2c	1	0	0	0	1	1	34
3a	0.75	0	4	2	0.67	0.85	24
3b	0.75	1/3	4	2	0.44	0.80	20
3c	1	0	0	0	1	1	75
4a	0.7	0	4	2	0.63	0.84	7
4b	0.7	2/3	4	2	0.5	0.81	6
4c	1	0	0	0	1	1	28
5a	0.7	0	4	2	0.67	0.87	8
5b	0.7	3/4	4	2	0.67	0.82	8
5c	1	0	0	0	1	1	37

TAB. 5.1 – *Parameters and results for each clustering test.*

level matching the tags exactly in tests (c) (generated with the efficiency set to one). In the latter case, the number of patches is between 3 and 5 times greater. This yields to considerable savings in computation time for a reasonable overhead. In tests (a) The use of *vmin* and *lmin* criteria prevents the generation of patches of too small size so as to have a better balance between subdomain unknowns. This balance can be further improved by using the *rvmin* criteria as shown in tests (c), at the expense of a slightly greater overhead (inferior eff-lev values). The most balanced level grids are obtained in tests (1b). The effectiveness of the clustering algorithm is the worst for 45 degree straight edges in the tag shape, which is best seen on test (3). The only way to match with a good accuracy such shapes is to use very small or very thin patches, which can contradict the requirements set to the clustering algorithm.

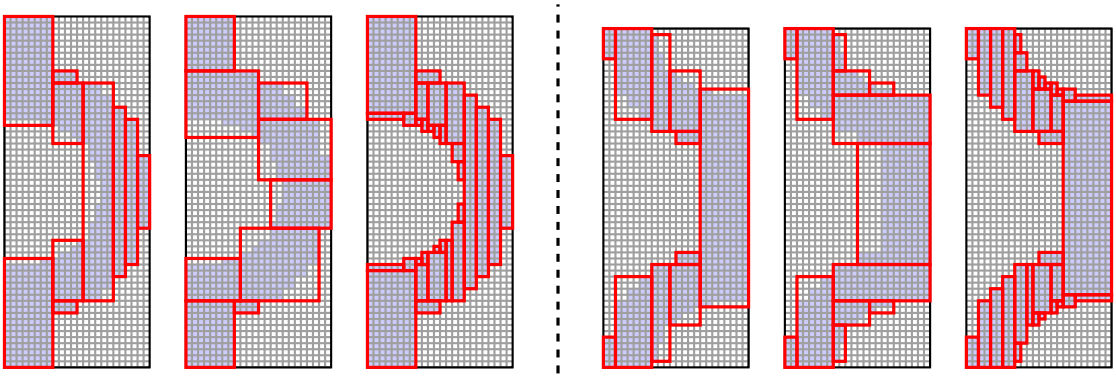


FIG. 5.12 – *Clustering tests (1) and (2) from [24].*

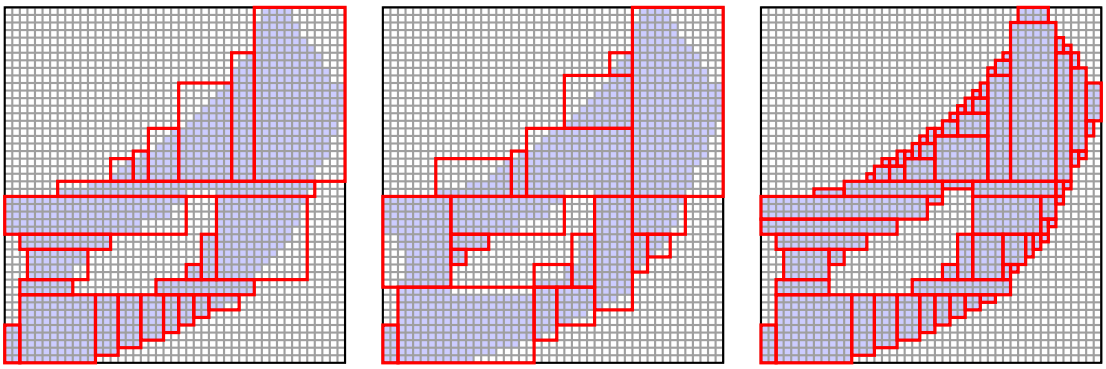


FIG. 5.13 – *Clustering test (3) from [24].*

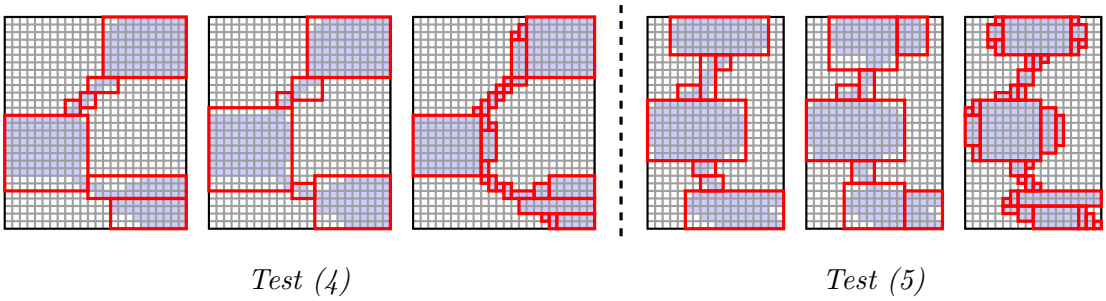


FIG. 5.14 – *Clustering tests (4) and (5) from [24].*

Chapter 6

SOLVING ON COMPOSITE GRIDS

In this chapter, we investigate the numerical solution of elliptic and hyperbolic PDE (in line with the projection scheme introduced in part 1) on level grids *and* composite grids. *This problem is two fold (see figure 6.1): first, as structured adaptive mesh refinements (SAMR) grids are generated as an union of composite grids, we have to deal with the discretization at the interface between two patches belonging to the same level, called* fine-fine interfaces. *Then for composite grids, at* coarse-fine interfaces *a specific discretization is also required. The AMR technique employs ghost-cells to cope with the interface problem. We show how these may be seen in the domain decomposition framework. We then give some insight on the multigrid solver that is implemented in the AMR code.*

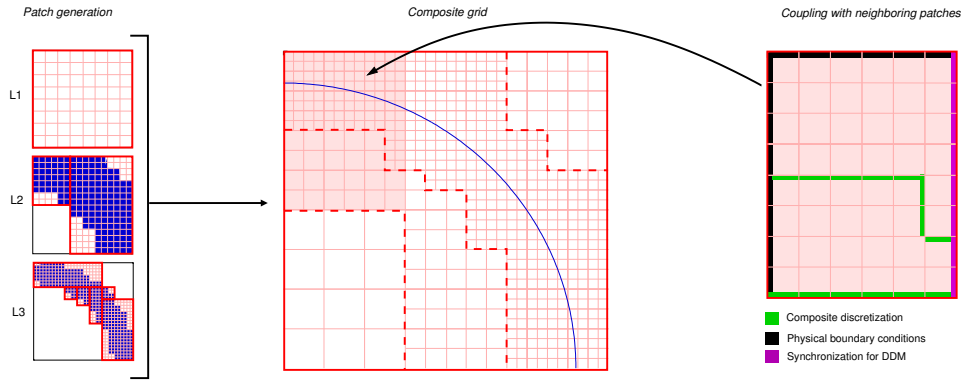
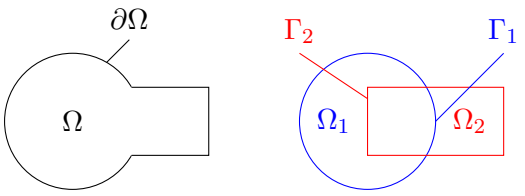


FIG. 6.1 – *Coarse-fine interfaces and fine-fine interfaces.*

The issue *fine-fine interfaces* is not related to the discretization scheme but to the numerical method for solving or relaxing (for multigrid) the discrete problem on a partitioned grid. A common technique for SAMR methods is to solve or relax individual patches and use ghost cells to synchronize them. After interpreting this ghost-cell technique in the broader context of domain decomposition methods, several numerical tests are performed to assess its efficiency and understand its behaviour either with solvers or relaxation methods.

As for *coarse-fine interfaces*, usual discretization schemes for computing a flux or an edge value have to be extended to the configuration of an edge separating grids at different levels. This specific discretization should involve both coarse cell values and fine cells values. At stake is to define a consistent discretization which has the required accuracy (first order for our pressure-correction scheme) and which does not add a significant computational expense nor



$\Omega = \Omega_1 \cup \Omega_2$
$\partial\Omega_i = (\partial\Omega_i \cap \partial\Omega) \cup \Gamma_i,\ i = 1, 2$

For $n = 0, 1, \ldots$ solve in sequence:

$$\begin{cases} \Delta u_1^{n+1} = f \text{ in } \Omega_1 \\ u_1^{n+1} = g \text{ on } \partial\Omega_1 \cap \partial\Omega \\ u_1^{n+1} = u_2^n \text{ on } \Gamma_1 \end{cases}$$

$$\begin{cases} \Delta u_2^{n+1} = f \text{ in } \Omega_2 \\ u_2^{n+1} = g \text{ on } \partial\Omega_2 \cap \partial\Omega \\ u_2^{n+1} = u_1^{n+1} \text{ on } \Gamma_2 \end{cases}$$

FIG. 6.2 – *Schwarz alternating algorithm on overlapping subdomains.*

introduces further complications with respect to the implementation.

6.1 Fine-fine interfaces

SAMR grids, whether level grids or composite grids, are partitioned as an union of Cartesian grids. We want to take advantage of existing numerical methods working on Cartesian grids for solving or relaxing on level grids. The most natural approach is to use domain decomposition methods: instead of solving a single problem on the level grid which may have a complex shape, we solve a set of coupled problems on the individual Cartesian grids making up the level grid.

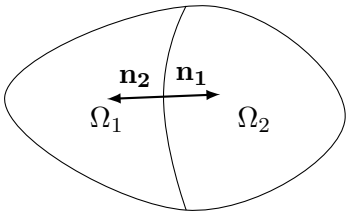
For the sake of simplicity, we will consider the Laplace equation although the following results apply to other elliptic PDE and to hyperbolic PDE as well.

$$\text{find } u \in \mathcal{C}^2(\Omega),\ \begin{cases} \Delta u = f \text{ in } \Omega \\ u = g \text{ on } \partial\Omega \end{cases} \tag{6.1.1}$$

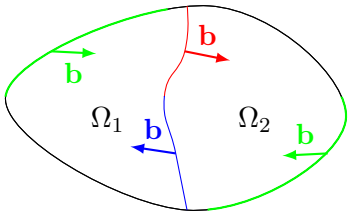
Domain decomposition methods were originally introduced by Schwarz [119] in 1870 as an analytic tool to prove the *Dirichlet principle* on complex domains [64], such as Ω in figure 6.2:

$$\begin{cases} u \in \left\{ w \in \mathcal{C}^2(\Omega),\ w|_{\partial\Omega} = g \right\} \\ \Delta u = f \text{ in } \Omega \end{cases} \iff \begin{cases} J(u) = \min \left\{ J(v),\ v \in \mathcal{C}^2(\Omega),\ v|_{\partial\Omega} = g \right\} \\ \forall v \in \mathcal{C}^2(\Omega),\ J(v) = \int_\Omega |\nabla v|^2 - fv \end{cases}$$

The Dirichlet principle was only proved on simple domains using Fourier analysis. On more complex domains it was still an open question whether the infimum of the Dirichlet integral is reached or not. Schwarz proposed a decomposition of complex domains into simpler domains in which the Dirichlet principle could be proved (see figure 6.2). The two domains overlap, and they are coupled through Dirichlet boundary conditions. The Schwarz alternating algorithm starts with an initial guess u_2^0, and the integration of the Dirichlet problem $\Delta u_1^{n+1} = f$ in Ω_1 with boundary conditions $u_1^{n+1}|_{\partial\Omega_1 \cap \partial\Omega} = g$ and $u_1^{n+1}|_{\Gamma_1} = u_2^n|_{\Gamma_1}$. It is then followed in Ω_2 by the integration of $\Delta u_2^{n+1} = f$ in Ω_2 with boundary conditions $u_2^{n+1}|_{\partial\Omega_2 \cap \partial\Omega} = g$ and $u_2^{n+1}|_{\Gamma_2} = u_1^{n+1}|_{\Gamma_2}$. By alternating between integrations in the two subdomains and using a maximum principle, Schwarz proved [119, 64] that the sequences $(u_1^n)_{n\in\mathbb{N}}$ and $(u_2^n)_{n\in\mathbb{N}}$ converge uniformly and their respective limits u_1 and u_2 and verify $u_1|_{\Gamma_i} = u_2|_{\Gamma_i}$ for $i = 1, 2$. It infers that u_1 and u_2 are restrictions to Ω_1 and Ω_2 of the same function u verifying the original problem on Ω. This algorithm is also referred to as *multiplicative Schwarz* (it can be interpreted as the composition of several operators). The larger the overlap region $\Omega_1 \cap \Omega_2$, the faster the convergence of the algorithm.



$\Omega = \Omega_1 \cup \Omega_2,\ \mathring{\Omega}_1 \cap \mathring{\Omega}_2 = \{\emptyset\},$
$\Gamma = \overline{\Omega}_1 \cap \overline{\Omega}_2,\ \mathbf{n} = \mathbf{n_1}$



Inflow boundaries: $\Gamma_{2,\text{in}}$, $\Gamma_{1,\text{in}}$, $\partial\Omega_{\text{in}}$

FIG. 6.3 – *Non-overlapping domain decomposition.*

Beyond its original use as an analytical tool, the Schwarz alternating algorithm is rather used nowadays as an iterative method for solving problems on overlapping decomposed domains. However overlapping Schwarz methods cannot be applied to our level grid decompositions, which are generated *without overlap* by the clustering algorithm presented in the last chapter (though an overlapping decomposition may be generated with another clustering algorithm). Therefore we will rather consider another class of domain decomposition methods which appeared along with the finite element method in the 60's, *substructuring domain decomposition methods.*

6.1.1 Non-overlapping domain decomposition

Multi-domain formulation

We now turn to *non-overlapping* domain decompositions. An example of non-overlapping decomposition is shown on figure 6.3. The domain Ω is partitioned into two subdomains Ω_1 and Ω_2 with boundary $\partial\Omega_i$ and normal vector $\mathbf{n_i}$. We denote $\Gamma = \overline{\Omega}_1 \cap \overline{\Omega}_2$ and $\mathbf{n} = \mathbf{n_1}$. We still consider the Laplace equation with Dirichlet boundary conditions:

$$\text{find } u \in \mathcal{H}^1(\Omega),\ \begin{cases} \Delta u = f \text{ in } \Omega \\ u = g \text{ on } \partial\Omega \end{cases} \tag{6.1.2}$$

The source term f is in $\mathcal{L}^2(\Omega)$ and the Dirichlet data g in $\mathcal{H}^{1/2}(\Omega)$. We would like to solve two problems (possibly coupled) on Ω_1 and Ω_2 such that their solutions match the restriction of the single-domain solution on their respective domains:

$$\text{for } i = 1, 2 :\ u_i = u|_{\Omega_i}$$

In order to establish the coupling between the two subdomains, the variational formulation of (6.1.2) is introduced. Given a test function $v \in \mathcal{H}^1_0(\Omega)$, we integrate by parts the single domain PDE. On the one hand, integrating by parts and noting that $v = 0$ on $\partial\Omega$, we have:

$$\begin{aligned} &\int_\Omega (\Delta u) v = \int_\Omega f v \\ \Leftrightarrow\ &\int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v \end{aligned}$$

On the other hand, splitting the integral on the two subdomains and integrating by parts, we get:

$$\begin{aligned}\int_\Omega (\Delta u)v &= \int_{\Omega_1} \nabla u_1 \cdot \nabla v + \int_{\Omega_2} \nabla u_2 \cdot \nabla v \\ &= \int_\Omega \nabla u \cdot \nabla v - \int_\Gamma (\nabla u_1) v \cdot \mathbf{n_1} - \int_\Gamma (\nabla u_2) v \cdot \mathbf{n_2}\end{aligned}$$

Therefore, we have

$$\int_\Gamma [(\nabla u_1 - \nabla u_2) \cdot \mathbf{n}]\, v = 0,$$

Therefore solving (6.1.2) is equivalent to solving two coupled problems on Ω_1 and Ω_2 with $u|_{\Omega_1} = u_1$ and $u|_{\Omega_2} = u_2$:

$$\begin{cases} \Delta u_1 = f & \text{in } \Omega_1 \\ u_1 = g & \text{on } \partial\Omega_1 \cap \partial\Omega \\ u_1 = u_2 & \text{on } \Gamma \qquad \text{(TC1)} \\ \partial_n u_1 = \partial_n u_2 & \text{on } \Gamma \qquad \text{(TC2)} \\ \Delta u_2 = f & \text{in } \Omega_2 \\ u_2 = g & \text{on } \partial\Omega_2 \cap \partial\Omega \end{cases} \tag{6.1.3}$$

The subdomains Ω_1 and Ω_2 are coupled through the *transmission conditions* (TC1) and (TC2). A multi-domain formulation can also be straightforwardly devised for a scalar conservation law. The single domain problem is stated as:

$$\text{find } u \in \mathcal{H}^1(\Omega),\ \begin{cases} au + \operatorname{div}(u\mathbf{b}) = f \text{ in } \Omega \\ u = g \text{ on } \partial\Omega_{\text{in}}, \end{cases} \tag{6.1.4}$$

where $\mathbf{b}$ denotes the advecting velocity and $\partial\Omega_{\text{in}}$ the inflow boundary, i.e. the set of points of $\partial\Omega$ where $\mathbf{bn} > 0$. We consider the decomposition given in figure 6.3 (right). Similarly the *inflow* part of $\Gamma = \partial\Omega_1 \cup \partial\Omega_2$ for Ω_1 is denoted $\Gamma_{1,\text{in}} = \Gamma_{2,\text{out}}$ and the *outflow* part $\Gamma_{1,\text{out}} = \Gamma_{2,\text{in}}$. Multiplying by a test function and integrating by parts as in the case of the pure diffusion yields the following transmission condition:

$$(\mathbf{b} \cdot \mathbf{n_1}) u_1 = -(\mathbf{b} \cdot \mathbf{n_2}) u_2 \quad \text{on } \Gamma_{1\text{in}} \cup \Gamma_{1\text{out}}$$

Hence the two-domain problem:

$$\begin{cases} au_1 + \operatorname{div}(u_1\mathbf{b}) = f & \text{in } \Omega_1 \\ u_1 = g & \text{on } \partial\Omega_1 \cap \partial\Omega_{\text{in}} \\ u_1 = u_2 & \text{on } \Gamma \qquad \text{(TC1)} \\ au_2 + \operatorname{div}(u_2\mathbf{b}) = f & \text{in } \Omega_2 \\ u_2 = g & \text{on } \partial\Omega_2 \cap \partial\Omega_{\text{in}} \end{cases} \tag{6.1.5}$$

Direct substructuring

Following [114], the above problems can reformulated as two Dirichlet problems coupled by an *interface equation* involving the *Steklov-Poincaré operator* [112, 113]. We start with the elliptic problem (6.1.2). Let λ be the trace of solution u to (6.1.2) on Γ. For $i = 1, 2$ we have:

$$\begin{cases} \Delta u_i = f & \text{in } \Omega_i \\ u_i = g & \text{on } \partial\Omega_i \cap \partial\Omega \\ u_i = \lambda & \text{on } \Gamma \end{cases} \tag{6.1.6}$$

The transmission condition (TC1) is given by $u_1|_\Gamma = u_2|_\Gamma = \lambda$. Enforcing condition (TC2) yields to a first order partial differential equation on λ. Let us first introduce two new operators after [114], $H_i : \mathcal{H}^{1/2}(\Gamma) \to \mathcal{H}^1(\Omega)$ and $Q_i : \mathcal{L}^2(\Omega) \to \mathcal{H}^1(\Omega)$:

$$H_i\mu = v \text{ with } \begin{cases} \Delta v = 0 \text{ in } \Omega_i \\ v = 0 \text{ on } \partial\Omega_i \cap \partial\Omega \\ v = \mu \text{ on } \Gamma \end{cases} \quad \text{and} \quad Q_i\rho = v \text{ with } \begin{cases} \Delta v = \rho \text{ in } \Omega_i \\ v = g \text{ on } \partial\Omega_i \cap \partial\Omega \\ v = 0 \text{ on } \Gamma \end{cases}$$

$Q_i\rho$ is the solution to the Poisson problem with data ρ, homogeneous Dirichlet condition on Γ and the original Dirichlet condition of the single domain problem (6.1.2) on $\partial\Omega_i \cap \Omega$. $H_i\mu$ is the extension (with a Poisson kernel in the present case) of the trace μ in Ω_i. As a result, any solution to the problems in Ω_1 and Ω_2 can be decomposed into the solution of two subproblems:

$$u_i = H_i(u_i|_\Gamma) + Q_i f \tag{6.1.7}$$

Using this decomposition, the transmission condition (TC2) can be reformulated as:

$$\partial_n H_1\lambda + \partial_n Q_1 f = \partial_n H_2\lambda + \partial_n Q_2 f$$

We introduce the *Steklov-Poincaré* operators S and S_i for $i = 1, 2$:

$$\begin{aligned} &S_i : \mu \in \mathcal{H}^{1/2}(\Gamma) \mapsto S_i\mu = (\partial_{n_i} H_i\mu)|_\Gamma \\ &S = S_1 + S_2 \end{aligned}$$

The Steklov-Poincaré operator is often referred to as *Dirichlet-to-Neumann map*. Indeed S_i maps a Dirichlet condition on Γ to the matching Neumann condition for the extension in Ω_i of the trace associated with the Dirichlet condition. The form of the Steklov-Poincaré operator depends on the PDE and it may not involve a normal derivative as in this example. An extensive analysis of Steklov-Poincaré operators in the context of domain decomposition methods can be found in references [112, 113, 3]. We eventually derive the *interface equation* from (TC2):

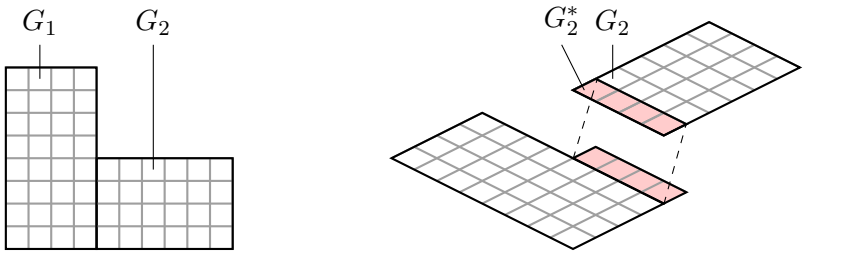
$$\boxed{S\lambda = \chi \quad \text{with} \quad \chi = \partial_n Q_2 f - \partial_n Q_1 f} \tag{6.1.8}$$

Upon solving the interface equation, problems (6.1.6) in Ω_1 and Ω_2 can be solved independently using the value of λ just calculated for the transmission condition (TC1). This is approach is called *direct substructuring*.

Substructuring domain decomposition methods were originally introduced by the structural engineering community [111] as a technique for solving problems with a larger number of degrees of freedom than the computer resources could handle with a single domain approach. The idea was to partition a structure into substructures, which would be individually represented as a single finite element [14]. These "super elements" would be coupled through artificial boundary conditions at their respective boundaries. A new problem would be defined using this super elements, equivalent to the interface problem (6.1.8) defined above. Upon solving this intermediate problem, the boundary unknowns of each super elements would be then determined. Thus the problems associated with each substructure could be solved independently.

For a scalar conservation law, the same arguments apply [113]. Let λ be the trace of solution u to (6.1.2) on Γ. For $i = 1, 2$ we have:

$$\begin{cases} au_i + \operatorname{div}(u_i\mathbf{b}) = f & \text{in } \Omega_i \\ u_i = g & \text{on } \partial\Omega_i \cap \partial\Omega \\ u_i = \lambda & \text{on } \Gamma \end{cases} \tag{6.1.9}$$



$G = G_1 \cup G_2$ *Ghost cell layers* G_1^*, G_2^* *for* G_1, G_2

FIG. 6.4 – *Simple two-domain decomposition with Cartesian grids.*

The transmission condition is given by $u_1|_\Gamma = u_2|_\Gamma = \lambda$. Let us introduce two operators $H_i : \mathcal{H}^{1/2}(\Gamma) \to \mathcal{H}^1(\Omega)$ and $Q_i : \mathcal{L}^2(\Omega) \to \mathcal{H}^1(\Omega)$:

$$H_i\mu = v \text{ with} \begin{cases} av + \operatorname{div}(v\mathbf{b}) = 0 \text{ in } \Omega_i \\ v = 0 \text{ on } \partial\Omega_i \cap \partial\Omega_{\text{in}} \\ v = \mu \text{ on } \Gamma_{i,\text{in}} \end{cases} ; \; Q_i\rho = v \text{ with} \begin{cases} av + \operatorname{div}(v\mathbf{b}) = \rho \text{ in } \Omega_i \\ v = g \text{ on } \partial\Omega_i \cap \partial\Omega_{\text{in}} \\ v = 0 \text{ on } \Gamma_{i,\text{in}} \end{cases}$$

The solution to the problem in Ω_1 or Ω_2 can be decomposed into the solution of two subproblems:

$$u_i = H_i(u_i|_\Gamma) + Q_i f \tag{6.1.10}$$

Using this decomposition, the transmission condition can be reformulated as:

$$(\mathbf{b}\cdot\mathbf{n})(H_1\lambda + Q_1 f) = (\mathbf{b}\cdot\mathbf{n})(H_2\lambda + Q_2 f) \quad \text{on } \Gamma$$

The *Steklov-Poincaré* operators S and S_i read for $i = 1, 2$:

$$\begin{aligned} &S_i : \mu \in \mathcal{H}^{1/2}(\Gamma) \mapsto S_i\mu = ((\mathbf{v}\cdot\mathbf{n_i})H_i\mu)|_\Gamma \\ &S = S_1 + S_2 \end{aligned}$$

We finally obtain the interface equation:

$$\boxed{S\lambda = \chi \quad \text{with} \quad \chi = (\mathbf{b}\cdot n)Q_2 f - (\mathbf{b}\cdot n)Q_1 f} \tag{6.1.11}$$

Algebraic formulation

The decomposition exposed at the continuous level also stands at the algebraic level. Let us consider the cell-centered finite-volume discretization of (6.1.3) on Cartesian grids (see figure 6.4). Despite all unknowns are defined as cell-averages, we introduce edge-average unknowns at the internal interface Γ for the purpose of exposing the method. The full discrete cell-centered scheme is recovered by substituting the discrete transmission conditions into the discretized PDE of each subdomain. The discrete problems in each subdomain read for all $K \in G_i$, $i = 1, 2$:

$$\sum_{\sigma\in\mathcal{E}_K} |\sigma| \frac{u_{i\sigma} - u_{iK}}{h/2} = |K| f_{iK} \quad \text{with} \begin{cases} u_{i\sigma} = (u_{iK} + u_{iL})/2 & \text{if } \sigma = K|L \in \mathcal{E}_{\text{int}} \\ u_{i\sigma} = g_{i\sigma} & \text{if } \sigma \in \mathcal{E}_{\text{ext}} \end{cases} \tag{6.1.12}$$

and the discrete transmission conditions (TC1) and (TC2) for $\sigma \in \mathcal{E}_\Gamma$, $i = 1, 2$ and $j \neq i$:

$$u_{i\sigma} = u_{j_\sigma} \equiv u_\sigma \tag{6.1.13a}$$

$$\frac{u_{i\sigma} - u_{iK}}{h/2} = -\frac{u_{j_\sigma} - u_{jK}}{h/2} \tag{6.1.13b}$$

For i=1,2 we denote $U_i = (u_{iK})_{K\in G_i}$ and $U_\Gamma = (u_\sigma)_{\sigma\in\mathcal{E}_\Gamma}$. F_i is a vector of the same size as U_i. The linear system arising from the discrete systems (6.1.12) and (6.1.13) read:

$$\begin{pmatrix} A_{11} & 0 & A_{1\Gamma} \\ 0 & A_{22} & A_{2\Gamma} \\ A_{\Gamma 1} & A_{\Gamma 2} & A_{\Gamma\Gamma,1} + A_{\Gamma\Gamma,2} \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \\ U_\Gamma \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \\ 0 \end{pmatrix} \tag{6.1.14}$$

The first two block rows match respectively the discrete PDE in Ω_1 and Ω_2. In the flux balance, a flux $(\partial_n u_i)_\sigma$ contributes to A_{ii} if $\sigma \in \mathcal{E}_{\text{int}}$, to both A_{ii} and $A_{i\Gamma}$ if $\sigma \in \mathcal{E}_\Gamma$ and to A_{ii} and F_i (for the Dirichlet boundary data) if $\sigma \in \mathcal{E}_{\text{ext}}$. The last block row matches the transmission condition (6.1.13b): $\{(\partial_{n_i} u_i)_\sigma, \sigma \in \mathcal{E}_\Gamma\} = A_{\Gamma i}U_i + A_{\Gamma\Gamma,i}U_\Gamma$. We can now define the algebraic equivalent operators to H_i and Q_i:

$$\begin{cases} H_i\Lambda = V \text{ with } A_{ii}V + A_{i\Gamma}\Lambda = 0 \\ Q_iF_i = V \text{ with } A_{ii}U_i = F_i \end{cases} \Rightarrow \begin{cases} H_i\Lambda = -A_{ii}^{-1}A_{i\Gamma}\Lambda \\ Q_iF_i = A_{ii}^{-1}F_i \end{cases}$$

Note that here F_i embeds both the source term f_{iK} and the Dirichlet boundary data $g_{i\sigma}$. Likewise the *discrete Steklov-Poincaré operator*, which is formally the *Schur complement*, is defined as:

$$\begin{aligned} S_i\Lambda &= (\partial_{n_i} H_i\lambda)_{\sigma\in\mathcal{E}_\Gamma} \\ &= A_{\Gamma i}(H_i\Lambda) + A_{\Gamma\Gamma,i}\Lambda \\ &= (A_{\Gamma\Gamma,i} - A_{\Gamma i}A_{ii}^{-1}A_{i\Gamma})\Lambda \end{aligned}$$

The algebraic equivalent of χ is simply $X = -A_{\Gamma 2}A_{22}^{-1}F_2 - A_{\Gamma 1}A_{11}^{-1}F_1$ as homogeneous Dirichlet conditions on Γ (or $U_\Gamma = 0$) are used to define operator Q_i. Thus following the same steps as in the continuous formulation yields the algebraic form of the interface equation:

$$\boxed{S\Lambda = X} \tag{6.1.15}$$

For a scalar conservation law, we choose the following discretization with an upwinding of advected quantities with respect to $\mathbf{b}$:

$$a|K|u_{iK} + \sum_{\sigma\in\mathcal{E}_K} |\sigma| u_{i\sigma}(\mathbf{b}_\sigma \cdot \mathbf{n}_{K,\sigma}) = |K| f_{iK} \tag{6.1.16}$$

$$\text{with } \begin{cases} u_{i\sigma} = u_{iK} \text{ if } \sigma = K|L \in \mathcal{E}_{\text{int}} \text{ and } \mathbf{b}_\sigma \cdot \mathbf{n}_{K,\sigma} < 0 \\ u_{i\sigma} = u_{iL} \text{ if } \sigma = K|L \in \mathcal{E}_{\text{int}} \text{ and } \mathbf{b}_\sigma \cdot \mathbf{n}_{K,\sigma} \geq 0 \\ u_{i\sigma} = g_{i\sigma} \text{ if } \sigma \in \mathcal{E}_{\text{ext}} \cap \partial\Omega_{\text{in}} \end{cases} \tag{6.1.17}$$

and the discrete transmission condition on $\sigma = K|K' \in \mathcal{E}_{\Gamma_{i,\text{in}}}$ with $K \in G_i$ and $K' \in G_j$, $i = 1, 2$ and $j \neq i$:

$$(\mathbf{b}_\sigma \cdot \mathbf{n}_{K',\sigma}) u_{i\sigma} = (\mathbf{b}_\sigma \cdot \mathbf{n}_{K,\sigma}) u_{j\,K'} \tag{6.1.18}$$

The algebraic system is the same as (6.1.14). For matrices $A_{\Gamma i}$ and $A_{\Gamma\Gamma,i}$ the rows matching $\Gamma_{i,\text{out}}$ are filled with zeros. At the algebraic level the definition of operators H_i and Q_i is the same, and yields to the same interface equation (6.1.15).

6.1.2 Iterative substructuring algorithms

While original substructuring methods relied on a direct resolution of the interface equation, recent methods use an iterative resolution of the interface equation (*iterative substructuring*). Two classical iterative algorithms are reviewed for elliptic PDE.

Dirichlet-Neumann algorithm

In the *Dirichlet-Neumann* algorithm [28, 14, 114] for two subdomains, starting with an initial guess u_1^0 in Ω_1, u_2^0 in Ω_2, and $\lambda^0 = u_1^0|_\Gamma$ on Γ the following problems are solved in sequence for $n = 0, 1, \ldots$

$$\begin{cases} \Delta u_1^{n+1} = f & \text{in } \Omega_1 \\ u_1^{n+1} = g & \text{on } \partial\Omega_1 \cap \partial\Omega \\ u_1^{n+1} = \lambda^n & \text{on } \Gamma \end{cases} \qquad \begin{cases} \Delta u_2^{n+1} = f & \text{in } \Omega_2 \\ u_2^{n+1} = g & \text{on } \partial\Omega_2 \cap \partial\Omega \\ \partial_n u_2^{n+1} = \partial_n u_1^{n+1} & \text{on } \Gamma \end{cases} \tag{6.1.19}$$

with next the update of the trace λ:

$$\lambda^{n+1} = (1-\theta)\lambda^n + \theta u_2^{n+1}|_\Gamma \tag{6.1.20}$$

with $\theta \in]0,1[$ a relaxation parameter. When convergence is reached, both transmission conditions (TC1) and (TC2) are satisfied. This algorithm can be interpreted as an interactive resolution of the interface equation (6.1.8). Using decomposition (6.1.7) the Neumann condition on Γ for the problem on Ω_2 yields:

$$\begin{array}{lll} & \partial_n u_2^{n+1} = \partial_n u_1^{n+1} & \text{on } \Gamma \\ \Leftrightarrow & \partial_n (u_2^{n+1} - Q_2 f) = \partial_n (u_1^{n+1} - Q_2 f) & \text{on } \Gamma \\ \Leftrightarrow & \partial_{n_2} H_2 (u_2^{n+1}|_\Gamma) = \partial_n (Q_2 f - Q_1 f - H_1 \lambda^n) & \text{on } \Gamma \\ \Leftrightarrow & S_2 (u_2^{n+1}|_\Gamma) = -S_1 \lambda^n + \partial_n (Q_2 f - Q_1 f) & \text{on } \Gamma \end{array}$$

Then the iteration (6.1.20) can be reformulated as:

$$\begin{aligned} \lambda^{n+1} &= \lambda^n + \theta (u_2^{n+1}|_\Gamma - \lambda^n) \\ &= \lambda^n + \theta (S_2^{-1}(S_1 \lambda^n - \chi) - \lambda^n) \\ &= \lambda^n + \theta S_2^{-1} (S\lambda^n - \chi) \end{aligned}$$

Therefore the Dirichlet-Neumann algorithm is equivalent to solving the interface equation using a Richardson iterative procedure with preconditioner S_2^{-1} and relaxation parameter θ. An alternative formulation of the Dirichlet-Neumann algorithm consists in updating the trace λ^n according to an iterative method preconditioned with S_2^{-1}, for instance preconditioned conjugate gradient or GMRES which are scalable unlike Richardson iterations.

Using the same notations as in the previous section, we denote $U_{\Gamma,i}$ the trace of the solution vector U_i on Γ. The Dirichlet-Neumann algorithm reads at the algebraic level, starting with initial guess Λ^0, for $n = 0, 1, \ldots$

Solve first in Ω_1 :

$$A_{11} U_1^{n+1} = F_1 - A_{1\Gamma} \Lambda^n$$

Then solve in Ω_2 :

$$\begin{cases} A_{22} U_2^{n+1} + A_{2\Gamma} U_\Gamma^{n+1} = F_2 \\ A_{\Gamma 2} U_2^{n+1} + A_{\Gamma\Gamma,2} U_\Gamma^{n+1} = -(A_{\Gamma 1} U_1^{n+1} + A_{\Gamma\Gamma,1} \Lambda^n) \end{cases}$$

followed by the update of trace Λ^n:

$$\Lambda^{n+1} = (1-\theta)\Lambda^n + \theta U_\Gamma^{n+1}$$

As exposed at the continuous level, this algorithm is equivalent to solving the discrete interface equation using a Richardson iterative scheme:

$$\begin{aligned}
& A_{\Gamma 2}U_2^{n+1} + A_{\Gamma\Gamma,2}U_{\Gamma,2}^{n+1} = -(A_{\Gamma 2}U_2^{n+1} + A_{\Gamma\Gamma,2}\Lambda^n) \\
\Leftrightarrow\ & A_{\Gamma 2}(U_2^{n+1} - A_{\Gamma 2}A_{22}^{-1}F_2) + A_{\Gamma\Gamma,2}U_{\Gamma,2}^{n+1} = -(A_{\Gamma 1}U_1^{n+1} + A_{\Gamma 2}A_{22}^{-1}F_2 + A_{\Gamma\Gamma,1}\Lambda^n) \\
\Leftrightarrow\ & -A_{\Gamma 2}A_{22}^{-1}A_{2\Gamma}U_{\Gamma,2}^{n+1} + A_{\Gamma\Gamma,2}U_{\Gamma,2}^{n+1} = -(A_{\Gamma 1}(A_{11}^{-1}F_1 - A_{11}^{-1}A_{1\Gamma}\Lambda^n) + A_{\Gamma 2}A_{22}^{-1}F_2 + A_{\Gamma\Gamma,1}\Lambda^n) \\
\Leftrightarrow\ & S_2U_{\Gamma,2}^{n+1} = -S_1\Lambda^n - A_{\Gamma 2}A_{22}^{-1}F_2 - A_{\Gamma 1}A_{11}^{-1}F_1
\end{aligned}$$

hence:

$$\Lambda^{n+1} = \Lambda^n + \theta S_2^{-1}(S\Lambda^n - X)$$

Neumann-Neumann algorithm

In the *Neumann-Neumann* algorithm [30, 14, 114] for two subdomains, starting with an initial guess u_1^0 in Ω_1, u_2^0 in Ω_2, and $\lambda^0 = u_1^0|_\Gamma$ on Γ the following problems are solved in sequence for $n = 0, 1, \ldots$

$$\begin{cases} \Delta u_i^{n+1} = f & \text{in } \Omega_i \\ u_i^{n+1} = g & \text{on } \partial\Omega_i \cap \partial\Omega \\ u_i^{n+1} = \lambda^n & \text{on } \Gamma \end{cases} \qquad \begin{cases} \Delta \psi_i^{n+1} = 0 & \text{in } \Omega_i \\ \psi_i^{n+1} = 0 & \text{on } \partial\Omega_i \cap \partial\Omega \\ \partial_n\psi_i^{n+1} = \partial_n u_1^{n+1} - \partial_n u_2^{n+1} & \text{on } \Gamma \end{cases} \tag{6.1.21}$$

with next the update of the trace λ:

$$\lambda^{n+1} = \lambda^n - (\sigma_1\psi_1^{n+1}|_\Gamma - \sigma_2\psi_2^{n+1}|_\Gamma) \tag{6.1.22}$$

where $\sigma_i \in]0,1[$ for $i = 1, 2$ are relaxation parameters. As for the Dirichlet-Neumann algorithm, transmission conditions (TC1) and (TC2) are verified at convergence. The Neumann condition on Ω_i for the second problem of (6.1.21) yields:

$$\begin{aligned}
\psi_i^{n+1}|_\Gamma &= S_i^{-1}(\partial_n u_1^{n+1} - \partial_n u_2^{n+1}) \\
&= S_i^{-1}(\partial_n H_1\lambda^n - \partial_n H_2\lambda^n - \chi) \\
&= S_i^{-1}(S\lambda^n - \chi)
\end{aligned}$$

The iteration (6.1.22) can be reformulated as:

$$\lambda^{n+1} = \lambda^n - (\sigma_1 S_1^{-1} + \sigma_2 S_2^{-1})(S\lambda^n - \chi)$$

Again, we have a Richardson iterative procedure preconditioned by $(\sigma_1 S_1^{-1} + \sigma_2 S_2^{-1})$ which could be replaced by a more efficient iterative scheme (eg. Krylov subspace methods).

Using the same notations as in the previous section, the Neumann-Neumann algorithm reads at the algebraic level, starting with initial guess Λ^0, for $n = 0, 1, \ldots$

First solve for U_i, $i = 1, 2$:

$$A_{ii}U_i^{n+1} + A_{i\Gamma}\Lambda^n = F_i$$

Then solve for Ψ_i, $i = 1, 2$:

$$\begin{cases} A_{ii}\Psi_i^{n+1} + A_{i\Gamma}\Psi_{\Gamma,i} = 0 \\ A_{\Gamma i}\Psi_i^{n+1} + A_{\Gamma\Gamma,i}\Psi_{\Gamma,i} = A_{\Gamma 1}U_1^{n+1} + A_{\Gamma 2}U_2^{n+1} + A_{\Gamma\Gamma}\Lambda^n \end{cases}$$

with the iteration on Λ^n:

$$\Lambda^{n+1} = \Lambda^n - (\sigma_1\Psi_1^{n+1}|_\Gamma - \sigma_2\Psi_2^{n+1}|_\Gamma)$$

Similarly the iteration can we rewritten as:

$$\Lambda^{n+1} = \Lambda^n - (\sigma_1 S_1^{-1} + \sigma_2 S_2^{-1})(S\Lambda^n - X)$$

6.1.3 Ghost-cell equivalent decomposition

In SAMR methods, a simple and common practice for solving (possibly inexact, eg. relaxation) a discrete problem on a given level partitioned into Cartesian grids is to use ghost cells. Let us start with the simple elliptic PDE dealt with previously.

Elliptic problems

The finite-volume discretization of the two-domain Poisson problem (6.1.2) using ghost cells reads:

$$\sum_{\sigma\in\mathcal{E}_K}|\sigma|\frac{u_{i\sigma}-u_{iK}}{h/2}=|K|f_{iK}\quad\text{with}\quad\begin{cases}u_{i\sigma}=(u_{iK}+u_{iL})/2 & \text{if } \sigma=K|L\in\mathcal{E}_{\text{int}}\\ u_{i\sigma}=g_{i\sigma} & \text{if } \sigma\in\mathcal{E}_{\text{ext}}\\ u_{i\sigma}=(u_{iK}+u^*_{i\,L})/2 & \text{if } \sigma=K|L\in\mathcal{E}_\Gamma\end{cases}\tag{6.1.23}$$

For an edge $\sigma=K|L\in\mathcal{E}_\Gamma$ with $K\in G_i$, cell L belongs to the ghost-cell layers G_i^* of grid G_i. u_i^* denotes a cell-average belonging to ghost cell layer G_i^*, hence not a discrete unknown for Ω_i. Within an iterative algorithm we would solve the discrete problem on Ω_1, then update ghost cells of Ω_2 using overlapping interior cells of Ω_1, then solve the problem on Ω_2 and update ghost cells of Ω_1. This algorithm can be equivalently described with interface transmission conditions by introducing interface unknowns:

First solve in Ω_1

$$\sum_{\sigma\in\mathcal{E}_K}|\sigma|\frac{u_{1\sigma}^{n+1}-u_{1K}^{n+1}}{h/2}=|K|f_{1K}$$

with:

$$\begin{cases}u_{1\sigma}^{n+1}=\frac{u_{1K}^{n+1}+u_{1L}^{n+1}}{2} & \text{if } \sigma=K|L\in\mathcal{E}_{\text{int}}\\ u_{1\sigma}^{n+1}=g_{1\sigma} & \text{if } \sigma\in\mathcal{E}_{\text{ext}}\end{cases}$$

using transmission conditions on $\mathcal{E}_\Gamma$:

$$\begin{cases}u_{2\sigma}=u_{1\sigma}^{n+1}\\ \frac{u_{1\sigma}^{n+1}-u_{1K}^{n+1}}{h/2}=-\frac{u_{2\sigma}-u_{1L}^{*n+1}}{h/2}\end{cases}$$

Update G_2 *ghost cells* $K\in G_2^*$

$L\in G_1\,/\,L\cap K\neq\{\emptyset\}:\ u_{2K}^{*n+1}=u_{1L}^{n+1}$

Then solve in Ω_2

$$\sum_{\sigma\in\mathcal{E}_K}|\sigma|\frac{u_{2\sigma}^{n+1}-u_{2K}^{n+1}}{h/2}=|K|f_{2K}$$

with:

$$\begin{cases}u_{2\sigma}^{n+1}=\frac{u_{2K}^{n+1}+u_{2L}^{n+1}}{2} & \text{if } \sigma=K|L\in\mathcal{E}_{\text{int}}\\ u_{2\sigma}^{n+1}=g_{2\sigma} & \text{if } \sigma\in\mathcal{E}_{\text{ext}}\end{cases}$$

using transmission conditions on $\mathcal{E}_\Gamma$:

$$\begin{cases}u_{1\sigma}=u_{2\sigma}^{n+1}\\ \frac{u_{2\sigma}^{n+1}-u_{2K}^{n+1}}{h/2}=-\frac{u_{1\sigma}-u_{2L}^{*n+1}}{h/2}\end{cases}$$

Update G_1 *ghost cells* $K\in G_1^*$

$L\in G_2\,/\,L\cap K\neq\{\emptyset\}:\ u_{1K}^{*n+2}=u_{2L}^{n+1}$

It yields the following equivalent algorithm at the algebraic level:

First solve in Ω_1 for U_1 :

$$\begin{cases}A_{11}U_1^{n+1}+A_{1\Gamma}U_\Gamma^{n+1/2}=F_1\\ A_{\Gamma1}U_1^{n+1}+A_{\Gamma2}\boxed{U_2^n}+A_{\Gamma\Gamma}U_\Gamma^{n+1/2}=0\end{cases}$$

Then solve in Ω_2 for U_2 :

$$\begin{cases}A_{22}U_2^{n+1}+A_{2\Gamma}U_\Gamma^{n+1}=F_2\\ A_{\Gamma1}\boxed{U_1^{n+1}}+A_{\Gamma2}U_2^{n+1}+A_{\Gamma\Gamma}U_\Gamma^{n+1}=0\end{cases}$$

with $A_{\Gamma\Gamma}=A_{\Gamma\Gamma,1}+A_{\Gamma\Gamma,2}$. The two vector variables U_2^n and U_1^{n+1} are the counterpart of the ghost-cell update steps of the discrete algorithm. Starting from the algebraic transmission condition for the second substep (in Ω_2):

$$\begin{aligned}&A_{\Gamma2}U_2^{n+1}+A_{\Gamma1}U_1^{n+1}+A_{\Gamma\Gamma}U_\Gamma^{n+1}=0\\ \Leftrightarrow\ &A_{\Gamma2}A_{22}^{-1}(F_2-A_{2\Gamma}U_\Gamma^{n+1})+A_{\Gamma1}A_{11}^{-1}(F_1-A_{1\Gamma}U_\Gamma^{n+1/2})+(A_{\Gamma\Gamma,1}+A_{\Gamma\Gamma,2})U_\Gamma^{n+1}=0\\ \Leftrightarrow\ &A_{\Gamma\Gamma,1}(U_\Gamma^{n+1}-U_\Gamma^{n+1/2})+S_1U_\Gamma^{n+1/2}+S_2U_\Gamma^{n+1}-X=0\end{aligned}$$

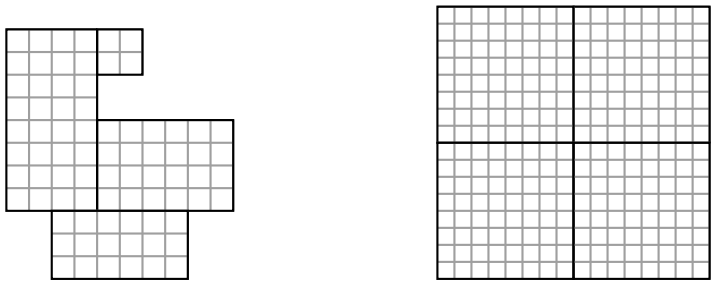


FIG. 6.5 – *More general domain decomposition with Cartesian grids.*

A similar equation can be obtained for the first substep (in Ω_1) as well:

$$A_{\Gamma\Gamma,2}(U_\Gamma^{n+1/2} - U_\Gamma^n) + S_1 U_\Gamma^{n+1/2} + S_2 U_\Gamma^n - X = 0$$

As a result this algorithm is equivalent to solving the discrete interface equation (6.1.15) with an iterative procedure with two semi-implicit substeps. Let $\Lambda = U_\Gamma$; then we have:

$$\begin{aligned} &(1) \quad A_{\Gamma\Gamma,2}(\Lambda^{n+1/2} - \Lambda^n) + S_1 \Lambda^{n+1/2} + S_2 \Lambda^n - X = 0 \\ &(2) \quad A_{\Gamma\Gamma,1}(\Lambda^{n+1} - \Lambda^{n+1/2}) + S_1 \Lambda^{n+1/2} + S_2 \Lambda^{n+1} - X = 0 \end{aligned}$$

We can identify in the above scheme the original Peaceman–Rachford *alternating direction implicit method* [108] with relaxation parameters $A_{\Gamma\Gamma,1}$ and $A_{\Gamma\Gamma,2}$.

However, the above configuration is not representative of the domain decomposition generated by the clustering algorithm applied to level grids. Indeed at a fixed level, we should rather assume a number $N > 2$ of subdomains with an arbitrary connectivity between each. Therefore it is important to define the underlying iterative scheme on the interface equation for this more general case as well. A straightforward calculation gives the interface equation:

$$S\Lambda = X$$

The right hand side is defined as the block vector $X = (X_\Gamma)_{\Gamma \in \{\Omega_i \cap \Omega_j, i \neq j\}}$ with

$$X_\Gamma = -A_{\Gamma i} A_{ii}^{-1} F_i - A_{\Gamma j} A_{jj}^{-1} F_j, \ \Gamma = \Omega_i \cap \Omega_j$$

and the Schur complement as the block matrix $S = (S_{\Gamma\Gamma'})_{\Gamma,\Gamma' \in \{\Omega_i \cap \Omega_j, i \neq j\}}$ with

$$\begin{array}{ll} S_{\Gamma\Gamma'} = S_{\Gamma\Gamma,i} + S_{\Gamma\Gamma,j} \text{ with } S_{\Gamma\Gamma,i} = A_{\Gamma\Gamma,i} - A_{\Gamma i} A_{ii}^{-1} A_{i\Gamma} & \text{if } \Gamma' = \Gamma = \Omega_i \cap \Omega_j, \ i \neq j \\ S_{\Gamma\Gamma'} = -A_{\Gamma i} A_{ii}^{-1} A_{k\Gamma'} & \text{if } \Gamma = \Omega_i \cap \Omega_j, \ \Gamma' = \Omega_i \cap \Omega_k, \ i \neq j \neq k \\ S_{\Gamma\Gamma'} = 0 & \text{else} \end{array}$$

Using the same iterative algorithm with ghost-cells for enforcing transmission conditions, the update of interface unknown U_Γ with $\Gamma = \Omega_i \cap \Omega_j$, resulting from the resolution of the subproblem in domain Ω_i takes the general form:

$$\begin{aligned} A_{\Gamma\Gamma,j}(U_\Gamma^{\phi_{i,\Gamma}} - U_\Gamma^{\phi_{i,\Gamma}-1/2}) + S_{\Gamma\Gamma,i} U_\Gamma^{\phi_{i,\Gamma}} + S_{\Gamma\Gamma,j} U_\Gamma^{\phi_{j,\Gamma}} \\ + \sum_{\substack{\Gamma' \in \{\Omega_i \cap \Omega_k\} \\ k \in [\![1,N]\!],\ k \neq i}} S_{\Gamma\Gamma'} U_{\Gamma'}^{\phi_{i,\Gamma'}} + \sum_{\substack{\Gamma' \in \{\Omega_j \cap \Omega_k\} \\ k \in [\![1,N]\!],\ k \neq j}} S_{\Gamma\Gamma'} U_{\Gamma'}^{\phi_{j,\Gamma'}} = X_\Gamma \end{aligned}$$

The values of the superscript $\phi_{i,\Gamma}$ depend on the sequence in which the problems of each subdomains are solved. In the above local interface equation, $\phi_{i,\Gamma} = \phi_{j,\Gamma} + 1/2$. For each $\Gamma' \in (\Omega_i \cap \Omega_k)_{1 \le k \le N}$, $\phi_{i,\Gamma} \ge \phi_{i,\Gamma'}$; for $\Gamma' \in (\Omega_j \cap \Omega_k)_{1 \le k \le N}$, $\phi_{i,\Gamma} \ge \phi_{j,\Gamma'}$. The local equations verified by the interfaces of a subdomain are coupled together which makes it more difficult to analyse the equivalent iterative algorithm solving the discrete interface equation.

As an example, let us consider a 2D rectangular domain partitioned into four subdomains of equal size, indexed clockwise from Ω_1 to Ω_4. The interface between Ω_i and Ω_j is denoted $\Gamma_{ij} = \Gamma_{ji}$. The same algorithm as with two subdomains is used. The subproblems are solved sequentially starting with Ω_1 then Ω_2, Ω_3 and Ω_4. This procedure would be equivalent to solving in sequence the linear systems:

$$\begin{pmatrix} A_{\Gamma_{41}\Gamma_{41},4} & 0 \\ 0 & A_{\Gamma_{12}\Gamma_{12},2} \end{pmatrix} \left[\begin{pmatrix} U^{n+1/2}_{\Gamma_{41}} \\ U^{n+1/2}_{\Gamma_{12}} \end{pmatrix} - \begin{pmatrix} U^{n}_{\Gamma_{41}} \\ U^{n}_{\Gamma_{12}} \end{pmatrix} \right] + \begin{pmatrix} S_{\Gamma_{41}\Gamma_{41},1} & S_{\Gamma_{41}\Gamma_{12}} \\ S_{\Gamma_{12}\Gamma_{41}} & S_{\Gamma_{12}\Gamma_{12},1} \end{pmatrix} \begin{pmatrix} U^{n+1/2}_{\Gamma_{41}} \\ U^{n+1/2}_{\Gamma_{12}} \end{pmatrix}$$
$$+ \begin{pmatrix} S_{\Gamma_{41}\Gamma_{41},4} & 0 \\ 0 & S_{\Gamma_{12}\Gamma_{12},2} \end{pmatrix} \begin{pmatrix} U^{n}_{\Gamma_{41}} \\ U^{n}_{\Gamma_{12}} \end{pmatrix} + \begin{pmatrix} S_{\Gamma_{41}\Gamma_{34}} U^{n}_{\Gamma_{34}} \\ S_{\Gamma_{12}\Gamma_{23}} U^{n-1/2}_{\Gamma_{23}} \end{pmatrix} - \begin{pmatrix} X_{\Gamma_{41}} \\ X_{\Gamma_{12}} \end{pmatrix} = 0$$

$$\begin{pmatrix} A_{\Gamma_{12}\Gamma_{12},1} & 0 \\ 0 & A_{\Gamma_{23}\Gamma_{23},3} \end{pmatrix} \left[\begin{pmatrix} U^{n+1}_{\Gamma_{12}} \\ U^{n+1/2}_{\Gamma_{23}} \end{pmatrix} - \begin{pmatrix} U^{n+1/2}_{\Gamma_{12}} \\ U^{n}_{\Gamma_{23}} \end{pmatrix} \right] + \begin{pmatrix} S_{\Gamma_{12}\Gamma_{12},2} & S_{\Gamma_{12}\Gamma_{23}} \\ S_{\Gamma_{23}\Gamma_{12}} & S_{\Gamma_{23}\Gamma_{23},2} \end{pmatrix} \begin{pmatrix} U^{n+1}_{\Gamma_{12}} \\ U^{n+1/2}_{\Gamma_{23}} \end{pmatrix}$$
$$+ \begin{pmatrix} S_{\Gamma_{12}\Gamma_{12},1} & 0 \\ 0 & S_{\Gamma_{23}\Gamma_{23},3} \end{pmatrix} \begin{pmatrix} U^{n+1/2}_{\Gamma_{12}} \\ U^{n}_{\Gamma_{23}} \end{pmatrix} + \begin{pmatrix} S_{\Gamma_{12}\Gamma_{41}} U^{n+1/2}_{\Gamma_{41}} \\ S_{\Gamma_{23}\Gamma_{34}} U^{n-1/2}_{\Gamma_{34}} \end{pmatrix} - \begin{pmatrix} X_{\Gamma_{12}} \\ X_{\Gamma_{23}} \end{pmatrix} = 0$$

$$\begin{pmatrix} A_{\Gamma_{23}\Gamma_{23},2} & 0 \\ 0 & A_{\Gamma_{34}\Gamma_{34},4} \end{pmatrix} \left[\begin{pmatrix} U^{n+1}_{\Gamma_{23}} \\ U^{n+1/2}_{\Gamma_{34}} \end{pmatrix} - \begin{pmatrix} U^{n+1/2}_{\Gamma_{23}} \\ U^{n}_{\Gamma_{34}} \end{pmatrix} \right] + \begin{pmatrix} S_{\Gamma_{23}\Gamma_{23},3} & S_{\Gamma_{23}\Gamma_{34}} \\ S_{\Gamma_{34}\Gamma_{23}} & S_{\Gamma_{23}\Gamma_{23},3} \end{pmatrix} \begin{pmatrix} U^{n+1}_{\Gamma_{23}} \\ U^{n+1/2}_{\Gamma_{34}} \end{pmatrix}$$
$$+ \begin{pmatrix} S_{\Gamma_{23}\Gamma_{23},2} & 0 \\ 0 & S_{\Gamma_{34}\Gamma_{34},4} \end{pmatrix} \begin{pmatrix} U^{n+1/2}_{\Gamma_{23}} \\ U^{n}_{\Gamma_{34}} \end{pmatrix} + \begin{pmatrix} S_{\Gamma_{23}\Gamma_{12}} U^{n+1}_{\Gamma_{12}} \\ S_{\Gamma_{34}\Gamma_{41}} U^{n}_{\Gamma_{41}} \end{pmatrix} - \begin{pmatrix} X_{\Gamma_{23}} \\ X_{\Gamma_{34}} \end{pmatrix} = 0$$

$$\begin{pmatrix} A_{\Gamma_{41}\Gamma_{41},1} & 0 \\ 0 & A_{\Gamma_{34}\Gamma_{34},3} \end{pmatrix} \left[\begin{pmatrix} U^{n+1}_{\Gamma_{41}} \\ U^{n+1}_{\Gamma_{34}} \end{pmatrix} - \begin{pmatrix} U^{n+1/2}_{\Gamma_{41}} \\ U^{n+1/2}_{\Gamma_{34}} \end{pmatrix} \right] + \begin{pmatrix} S_{\Gamma_{41}\Gamma_{41},4} & S_{\Gamma_{41}\Gamma_{34}} \\ S_{\Gamma_{34}\Gamma_{41}} & S_{\Gamma_{34}\Gamma_{34},4} \end{pmatrix} \begin{pmatrix} U^{n+1}_{\Gamma_{41}} \\ U^{n+1}_{\Gamma_{34}} \end{pmatrix}$$
$$+ \begin{pmatrix} S_{\Gamma_{41}\Gamma_{41},1} & 0 \\ 0 & S_{\Gamma_{34}\Gamma_{34},3} \end{pmatrix} \begin{pmatrix} U^{n+1/2}_{\Gamma_{41}} \\ U^{n+1/2}_{\Gamma_{34}} \end{pmatrix} + \begin{pmatrix} S_{\Gamma_{41}\Gamma_{12}} U^{n+1/2}_{\Gamma_{12}} \\ S_{\Gamma_{34}\Gamma_{23}} U^{n+1}_{\Gamma_{23}} \end{pmatrix} - \begin{pmatrix} X_{\Gamma_{41}} \\ X_{\Gamma_{34}} \end{pmatrix} = 0$$

This is clearly not the ADI iterative method though there are some similarities: in each linear system the 2×2 block matrices may be seen as a splitting of the matching block rows of the Schur complement matrix S.

Hyperbolic problems

The same results apply for a scalar conservation law, except that ghost-cell synchronization is only carried out at the inflow part of the interface between two subdomains.

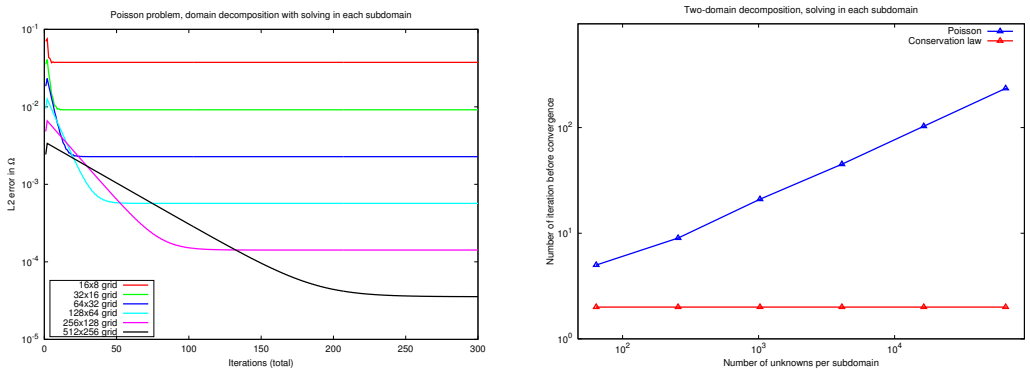


FIG. 6.6 – *Solving subproblems with a two domain decomposition.*

6.1.4 Numerical tests

First the ghost-cell domain decomposition algorithm is tested using a full resolution of each subproblem, which is the usual way to proceed for solving decomposed problems. Then having in mind the use of multigrid solvers we turn to a domain decomposition algorithm where a few relaxation steps are performed in each subdomain instead of a full resolution.

Subdomain resolution

The efficiency of the ghost-cell domain decomposition is assessed on an elliptic problem and on a hyperbolic problem. The elliptic problem under consideration is a Poisson problem on $\Omega = [0, L_x] \times [0, L_y]$:

$$\begin{cases} \Delta u = f & \text{in } \Omega \\ u = g & \text{on } \partial\Omega \end{cases}$$

with right hand side $f(x, y) = -8\pi^2 \sin(2\pi x) \sin(2\pi y)$ and with Dirichlet boundary data g the trace on $\partial\Omega$ of the exact solution $u_{\text{ref}}(x, y) = \sin(2\pi x) \sin(2\pi y)$. The finite volume discretization is the same as (6.1.12).

The hyperbolic problem is a simple scalar conservation law on $\Omega = [0, L_x] \times [0, L_y]$:

$$\begin{cases} \partial_t u + \operatorname{div}(u\mathbf{b}) = 0 & \text{in } \Omega \\ u = g & \text{on } \partial\Omega_{\text{in}} \end{cases}$$

with advecting velocity $\mathbf{b} = [2, 1]^t$. The initial data is $u^0 = 1 + \exp[-(x^2 + y^2)/(1/20)]$. The exact solution is given by the method of characteristics by $u_{\text{ref}}(x, y) = 1 + \exp[-((x - b_x t)^2 + (y - b_y t)^2)/(1/20)]$. The inflow boundary data on $\partial\Omega_{\text{in}}$ is set to the trace of u_{ref}. Using a simple time implicit Euler scheme with time step δt, the cell-centered finite-volume discretization of the scalar conservation law is given by (6.1.16) with $a = 1/\delta t$ and $f = u^0/\delta t$. The timestep δt is set to $1/128$.

In the first test, we take $L_x = 2L_y = 2$. The domain Ω is decomposed into two subdomains Ω_1 and Ω_2 of equal size. The objective is to evaluate the number of iterations of the domain decomposition algorithm to reach convergence when the number of unknowns in each subdomain increases. The respective grids of the two subdomains, G_1 and G_2, are set to 8×8, 16×16, 32×32, 64×64, 128×128 and 256×256 cells. The solution is initialized by zero. At convergence, the solution to the two-domain problem matches the solution of the single domain problem. For the

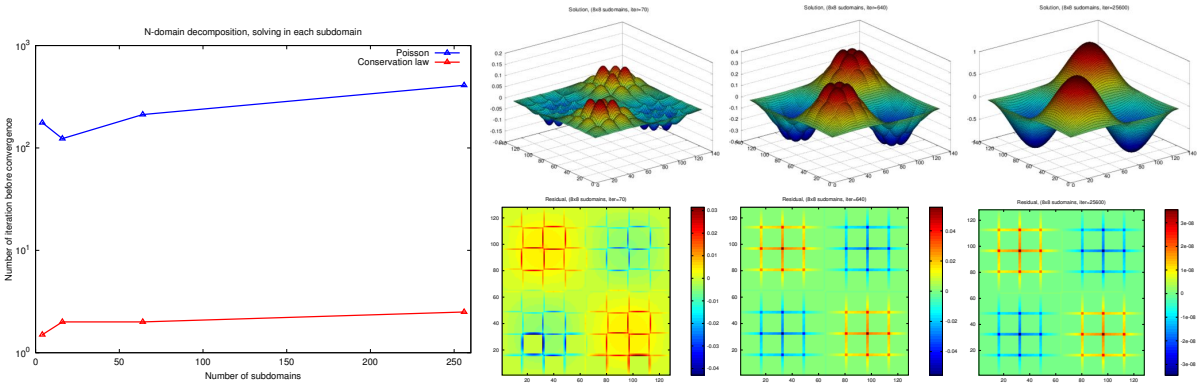


FIG. 6.7 – *Solving subproblems with more than two subdomains decompositions.*

elliptic problem the energy norm of the consistency error decreases as $o(h^2)$, which is consistent with the second order spatial discretization. Likewise a first order convergence is observed with the conservation law, discretized with a first order scheme. When solving the Poisson problem the number of iterations of the domain decomposition algorithm increases linearly with N^α, denoting by N the number of unknowns and $\alpha > 0$ (see figure 6.6). It clearly shows that this algorithm is not scalable. A very different behaviour is observed with the scalar conservation law: the method always converges always in one iteration. Indeed the first subdomain solved (Ω_1) has an *outflow* boundary Γ with its neighbor Ω_2. As a result using a first order upwind scheme the discrete PDE in Ω_1 is decoupled from Ω_2. This would not be the case had we started with Ω_2 instead of Ω_1.

In the second test, we set $L_x = L_y = 1$. The domain Ω is partitioned into $n_d = 4^k$ subdomains of equal size, with $1 \leq k \leq 4$. The aim is to observe the convergence of the domain decomposition algorithm when the number of unknowns of the problem in Ω is fixed while it is decomposed into an increasing number of subdomains. The subdomains are solved in sequence and the order in which they are solved should affect the convergence. Only one order will be dealt with, the order in which the subdomains have been created by the grid generation algorithm. The solution is initialized by zero in each subdomain. The hyperbolic problem always needs very few domain decomposition cycles. The number of cycles for the elliptic problem is fairly stable around 10^2 (see figure 6.7). The evolution of the solution in each subdomain and that of the *local residuals* can provide insightful information with respect to the behaviour of the domain decomposition algorithm. On figure 6.7 right, the solution for a 8×8 partition with the Poisson problem (first row) and the initial local residuals (second row) are shown after respectively 1 cycle, 10 cycles and 400 cycles. The initial local residual is useful in giving information on how well the transmission conditions are verified. After 1 cycle and 10 cycles the order of magnitude of the residual is still 10^{-1}: the transmissions conditions (TC1) and (TC2) are very loosely satisfied. After 400 cycles, the order of magnitude of the residual is 10^{-8}: the transmission conditions are much better verified and the solution on $\cup_i \Omega_i$ is identical (up to machine precision) to the single domain solution on Ω. Furthermore, it may be noticed that the transmission conditions are almost immediately verified where the solution is almost constant whereas they show a large error where the steepest gradients of the solution are located. The highest error on the transmission conditions is situated at the intersection between subdomain interfaces — which would be the so-called *crosspoints* in finite element

domain decompositions.

Subdomain relaxation

As a prelude to multigrid methods which will be discussed in the next chapter, we investigate the use of *smoothers* instead of solvers in the ghost-cell domain decomposition algorithm. The same discrete problems are considered. The linear system associated with a subdomain Ω_i reads:

$$A_i U_i = b_i \text{ with } \begin{cases} A_i = A_{ii} - \displaystyle\sum_{\Gamma=\Omega_i\cup\Omega_j, j\neq i} A_{i\Gamma} A_{\Gamma\Gamma}^{-1} A_{\Gamma i} \\ b_i = F_i + \displaystyle\sum_{\Gamma=\Omega_i\cup\Omega_j, j\neq i} A_{i\Gamma} A_{\Gamma\Gamma}^{-1} A_{\Gamma j} U_j \end{cases}$$

Let us point out that further interpretations of smoothing on multiple domain exist: *block relaxation*, *local smoothing*, *line relaxation*, *grid partitioning* approach, *chaotic relaxation*. We prefer to view relaxation on partitioned domain as a relaxation on individual problems coupled with relevant transmission conditions, hence the choice of the domain decomposition framework.

In domain decomposition methods, it can be somewhat questionable to solve to machine precision subproblems while they are not provided with the "correct" transmission conditions (in the sense the latter are "approximate"). The same concern stands when relaxing on subdomains: is it possible to define an optimal number of relaxation iterations per subdomain? The aim of the numerical tests will be to assess the impact of the partitioning and of number of local relaxation iterations on the overall efficiency of the smoother.

Smoothers are relaxation methods which feature a *smoothing property* [33], i.e. to be very efficient at damping the oscillatory (high frequency) components of the error while having a small impact on the smooth (low frequency) ones. Note that in this section the *error* refers to the error with respect to the discrete solution of the PDE ($e = A^{-1}b - U$), not the consistency error. Such iterative methods are usually very inefficient for solving linear systems because of their slow convergence on the low frequency components of the error. In contrast the high frequency components of the error are typically damped after a few iterations, and this is what makes them a key component of multigrid. Common smoothers include the *Jacobi* method and the *Gauss-Seidel* method with different choices of relaxation patterns (eg. Red-Black ordering). We will only consider the Gauss-Seidel method with lexicographical ordering.

We define the splitting $A = A_\ell + A_d + A_u$ with A_d the diagonal part of A, A_ℓ its lower triangular part and A_u its upper triangular part. The iteration scheme of lexicographical Gauss-Seidel reads:

$$MU^{k+1} = NU^k + b \quad \Leftrightarrow \quad U^{k+1} = M^{-1}NU^k + M^{-1}b$$

with $M = A_d - A_\ell$ and $N = A_u$; or put in more general form:

$$U^{k+\nu} = R^\nu U^k + R^{\nu-1} b$$

with $R = M^{-1}N$.

Now comes into question how efficient smoothers are when a domain is partitioned into many subdomains. Also, how the number of relaxation steps in each subdomain impacts the reduction of the high frequency components of the global error.

In the following tests, we will monitor the power spectrum of the global error. The base grid G is a square Cartesian grid with $N \times N$ cells. Assuming each cell is indexed by the pair

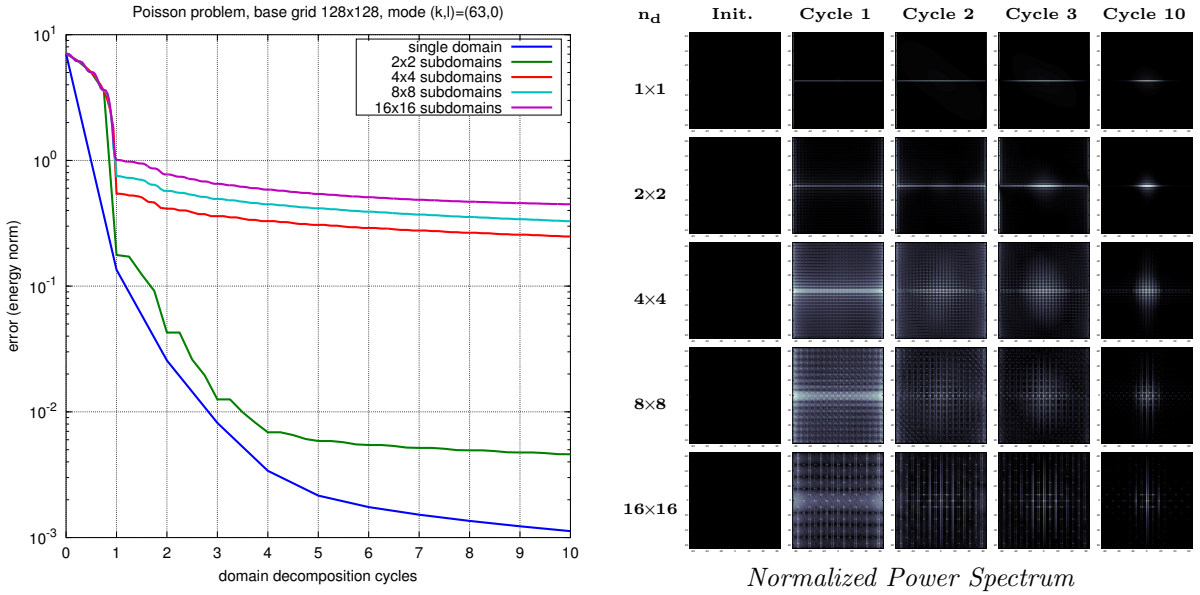


FIG. 6.8 – *Poisson problem: base grid* 128×128 *cells, single relaxation step per subdomain. Initialization: solution to the discrete PDE perturbed with the mode* $w^{(63)}$*. Left: evolution of the error in energy norm with domain decomposition cycles. Right: normalized power spectrum after several cycles. The power spectrum ranges from zero (black) to its highest value (white). The horizontal frequencies range from mode* $k = -64$ *to* $k = 64$ *and the vertical frequencies from* $l = -64$ *to* $l = 64$*.*

$(i, j) \in [\![1, N]\!]^2$, the 2D Fourier transform of the unknown u_{ij} reads:

$$U_{kl} = \frac{1}{N^2} \sum_{1 \leq i,j \leq N} u_{ij} e^{-J2\pi(ki+lj)/N}$$

with $J = \sqrt{-1}$. The domain decomposition algorithm will be initialized with the solution to the discrete PDE perturbed with a mode $w^{(k)}$ along the horizontal dimension, defined on the base grid as:

$$\forall i, j \in [\![1, N]\!]^2, \; w_{ij}^{(k)} = 10 \sin\left(i \frac{2\pi k}{N}\right)$$

with $-N/2 \leq k \leq N/2$. Its Fourier transform results in two Diracs centered at frequencies $\pm k/N$.

In the first test, we used as initial value a perturbation to the discrete solution with the mode $w^{(63)}$ on a 128×128 base grid partitioned into 2×2, 4×4, 8×8 and 16×16 subdomains. On this grid, the mode $w^{(63)}$ belongs to the highest frequencies. The number of relaxation steps is set to $\nu = 1$ and 10 domain decomposition cycles are performed. With the Poisson problem, the reduction of the energy norm of the error is made up of two stages. First, a fast decrease by several orders of magnitude: 4 orders in 10 cycles for the single domain problem, 3 orders in 4 cycles for the 2×2 partition and only 1 order in 1 cycle for the other partitions. Then, a slow decrease of the energy norm is observed until convergence (up to thousands of cycles). During the first stage, the high frequency components of the error are damped but at the same time some low frequency components are created. During the second stage, the total energy of the error does not change much, but a transfer of energy from the high frequency components

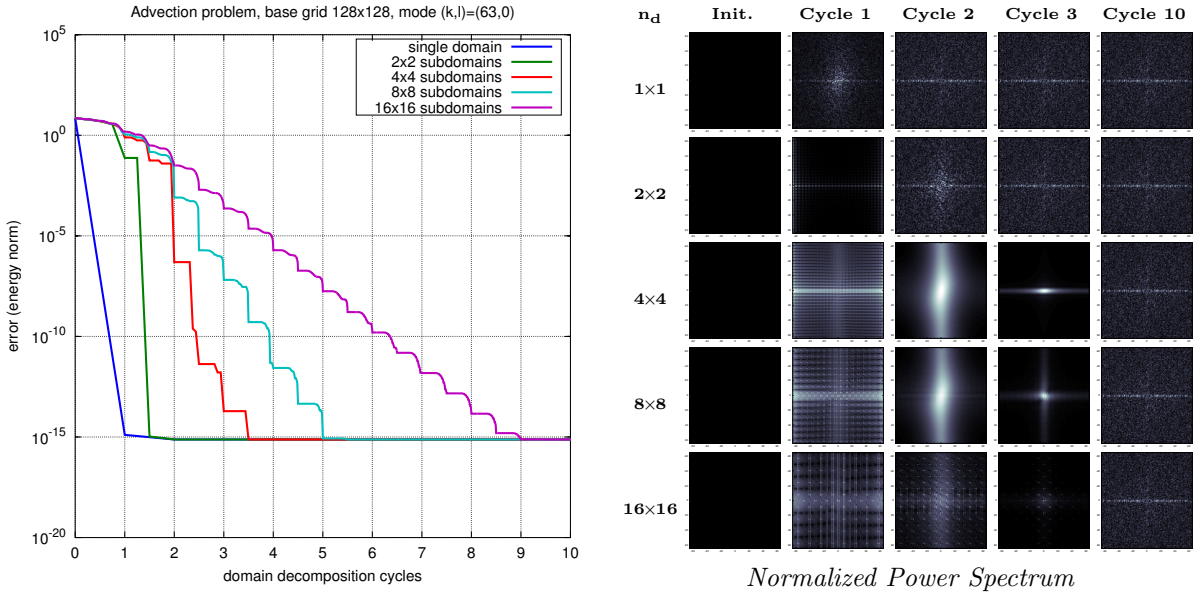


FIG. 6.9 – *Advection problem: base grid* 128×128 *cells, single relaxation step per subdomain. Initialization: solution to the discrete PDE perturbed with the mode* $w^{(63)}$*. Left: evolution of the error in energy norm with domain decomposition cycles. Right: normalized power spectrum after several cycles. The power spectrum ranges from zero (black) to its highest value (white). The horizontal frequencies range from mode* $k = -64$ *to* $k = 64$ *and the vertical frequencies from* $l = -64$ *to* $l = 64$*.*

to the low frequency ones occurs. The influence of the domain partitioning is very noticeable: after the first iteration, further high frequency modes are created. With 16 subdomains and more, these additional modes spread over the whole spectrum of the error. In the second stage for $n_d > 16$, the energy norm is almost steady ; nonetheless the next domain decomposition cycles do play an important role in the reduction of the oscillatory modes with what seems to be a transfer of energy from the high frequencies to the low frequencies.

As for the advection problem, the energy norm of the error decreases in less than 10 cycles to the floating point error. At the same time the high frequency components disappear, new components are created in the low frequency range of the error.

In the second test, we deal with the influence of the number of relaxation steps $\nu \in \{1, 2, 3, 5, 10\}$ carried out in each subdomain, i.e. the *inner* iterations of the algorithm in contrast to the *outer* iterations corresponding to the domain decomposition cycles. The same grid and the same initialization are used, with a fixed partition of 16×16 subdomains. Up to two cycles of domain decomposition, the algorithm is less efficient at reducing the energy norm of the error when using an increasing number of inner iterations. After three cycles and more, this trend is reversed. Regarding the spectrum of the error, it appears that more inner iterations help to concentrate the error in the low frequency components though the gain is not significant.

As a conclusion of these numerical tests, when smoothing on a partitioned grid, very few inner iterations (1 or 2) are necessary. The number of outer iterations should be greater or equal to 2 to allow enough synchronizations between subdomain solutions.

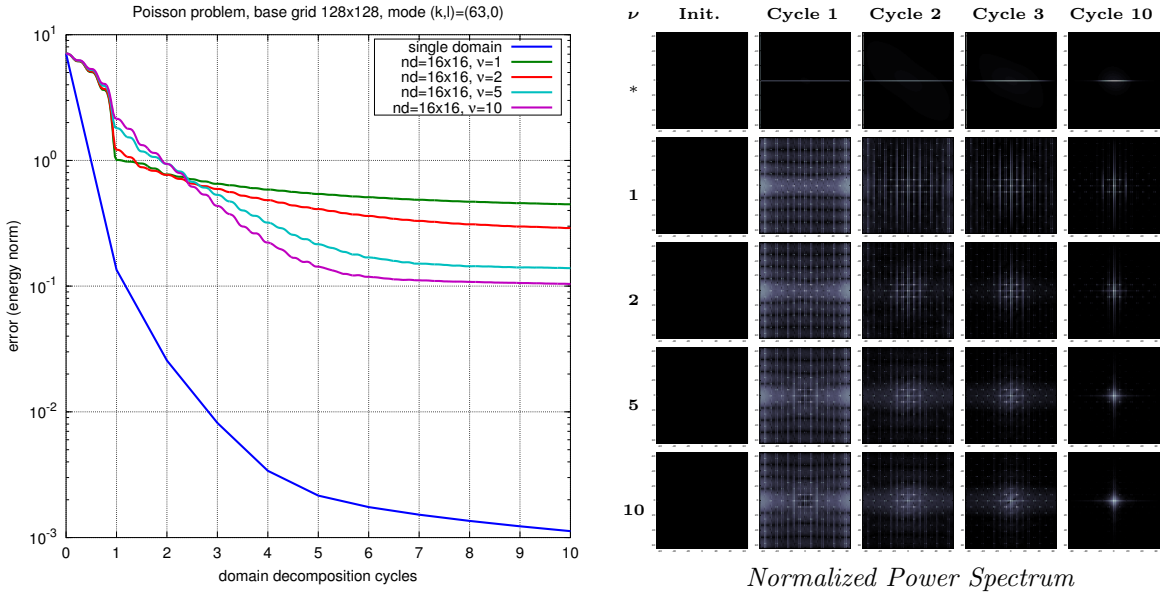


FIG. 6.10 – *Poisson problem: base grid* 128×128 *cells, partition into* 16×16 *subdomains. Initialization: solution to the discrete PDE perturbed with the mode* $(k, l) = (63, 0)$. *Left: evolution of the error in energy norm with domain decomposition cycles for several choices of relaxation steps* ν. *Right: normalized power spectrum after several cycles. The power spectrum ranges from zero (black) to its highest value (white). The horizontal frequencies range from mode* $k = -64$ *to* $k = 64$ *and the vertical frequencies from* $l = -64$ *to* $l = 64$.

6.2 Coarse-fine interfaces

6.2.1 Domain decomposition with non-matching grids

In the previous section, we presented non-overlapping domain decomposition methods for smoothing a level grid partitioned into patches, i.e. a domain decomposition method involving the same discretization in each subdomain. Coarse-fine interface can also could also be dealt with domain decomposition methods, more precisely with the Mortar method. The latter is intended to couple subdomains featuring different space discretizations. This technique was studied in a side work to this thesis [118] in collaboration with A. Samake, S. Bertoluzza, M. Pennacchio and C. Prud'Homme. We never actually used it for AMR and we simply give here the basis of the its formulation, as it is a relevant technique for coupling coarse grids and fine grids. The notations introduced hereby hold for this section only.

The mortar finite element method was initially introduced by C. Bernardi, Y. Maday and A. Patera in [27] as a method for solving domain decomposition problems with different finite element discretizations in each subdomain. In the original mortar formulation, the transmission conditions between subdomains were defined through a modification of the original functional spaces of each subdomain. Later another formulation introduced by F. Ben Belgacem and Y. Maday [20], which will be considered here, uses Lagrange multipliers defined on a common trace space at subdomain interfaces to impose the weak transmission conditions. For a given interface between two subdomains, the Lagrange multiplier is chosen to belong arbitrarily to the trace space of one of the two subdomains.

Let us consider the simple Poisson problem

$$\begin{cases} -\Delta u = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{cases}$$

with $\Omega \subset \mathbb{R}^n$, $u \in H_0^1(\Omega)$ and $f \in L^2(\Omega)$. The weak formulation of the problem reads:

$$\underbrace{\int_\Omega \nabla u \cdot \nabla v}_{a(u,v)} = \underbrace{\int_\Omega fv}_{f(v)} \quad \forall v \in H_0^1(\Omega)$$

which is equivalent to find $u \in H_0^1(\Omega)$ by solving the minimization problem:

$$J(u) = \min_{v \in H_0^1(\Omega)} \left\{ \frac{1}{2} \int_\Omega |\nabla v|^2 - \int_\Omega fv \right\} = \min_{v \in H_0^1(\Omega)} J(v)$$

Now domain Ω is decomposed into two subdomains Ω_1 and Ω_2 i.e. $\Omega = \Omega_1 \cup \Omega_2$ separated by the interface $\Gamma = \Omega_1 \cap \Omega_2$. The restriction of the solution u to subdomain Ω_i is denoted $u_i = u|_{\Omega_i}$. The minimization problem can now be reformulated as:

$$J(u_1) + J(u_2) = \min_{v_1, v_2 \in E} \{J(v_1) + J(v_2)\}$$

assuming the following definition for functional space E:

$$E = \left\{ (u_1, u_2) \in H^1(\Omega_1) \times H^1(\Omega_2) \ ; \ u_1 = u_2 \text{ on } \Gamma \text{ and } u_i = 0 \text{ on } \partial\Omega_i \setminus \Gamma \right\}$$

This is equivalent [5, Chapter 10] to search $u_1, u_2 \in H^1(\Omega_1) \times H^1(\Omega_2)$ and the Lagrange multiplier $\lambda \in H^{1/2}(\Gamma)$ such that:

$$\begin{cases} \mathcal{L}(u_1, u_2, \lambda) = J(u_1) + J(u_2) + \int_\Gamma \lambda(u_1 - u_2) \\ \mathcal{L}(v_1, v_2, \lambda) < \mathcal{L}(u_1, u_2, \lambda) < \mathcal{L}(u_1, u_2, \mu) \quad \forall v_1, v_2 \in H^1(\Omega_1) \times H^1(\Omega_2), \ \forall \mu \in H^{1/2}(\Gamma) \end{cases}$$

The last term on the right hand side stands for the weak form of the transmission condition (presently the continuity of the unknown u) at interface Γ. The above problem can be approximated by the following saddle point problem:

$$\begin{cases} a_1(u_1, v_1) + b(v_1, \lambda) = f_1(v_1) & \forall v_1 \in H(\Omega_1) \\ a_2(u_2, v_2) + b(v_2, \lambda) = f_2(v_2) & \forall v_2 \in H(\Omega_2) \\ b(u_1, \mu) - b(u_2, \mu) = 0 & \forall \mu \in H^{1/2}(\Gamma) \end{cases}$$

with the bilinear forms $a_i(u, v) = \int_{\Omega_i} uv$ and $b(u, q) = \int_\Gamma uq$. When discretizing this system of equations with the finite element method, the Lagrange multiplier must be chosen to belong to the trace space of the finite element space of Ω_1 or Ω_2, as we assume different discretizations in each subdomain. For instance to relate the present example to AMR, the meshes of both subdomains could be Cartesian grids, with the mesh of Ω_2 twice as much finer as the mesh of Ω_1.

An example of a four subdomain partition with non-matching grids is given in figure 6.11. A possible algebraic formulation of the Mortar finite element reads:

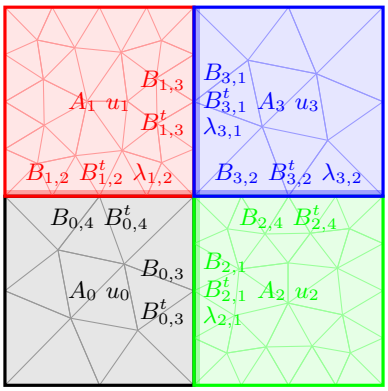


FIG. 6.11 – *Example of Finite Element Mortar domain decomposition.*

$$\left(\begin{array}{cccc|cccc} A_0 & 0 & \dots & 0 & B^t_{0,3} & B^t_{0,4} & 0 & 0 \\ 0 & A_1 & \ddots & \vdots & 0 & B^t_{1,2} & B^t_{1,3} & 0 \\ \vdots & \ddots & A_3 & 0 & 0 & 0 & B^t_{3,1} & B^t_{3,2} \\ 0 & \dots & 0 & A_2 & B^t_{2,1} & 0 & 0 & B^t_{2,4} \\ \hline B_{0,3} & 0 & 0 & B_{2,1} & 0 & \dots & \dots & 0 \\ B_{0,4} & B_{1,2} & 0 & 0 & \vdots & \ddots & & \vdots \\ 0 & B_{1,3} & B_{3,1} & 0 & \vdots & & \ddots & \vdots \\ 0 & 0 & B_{3,2} & B_{2,4} & 0 & \dots & \dots & 0 \end{array}\right) \left(\begin{array}{c} u_0 \\ u_1 \\ u_3 \\ u_2 \\ \hline \lambda_{2,1} \\ \lambda_{1,2} \\ \lambda_{3,1} \\ \lambda_{3,2} \end{array}\right) = \left(\begin{array}{c} F_0 \\ F_1 \\ F_3 \\ F_2 \\ \hline 0 \\ \vdots \\ \vdots \\ 0 \end{array}\right)$$

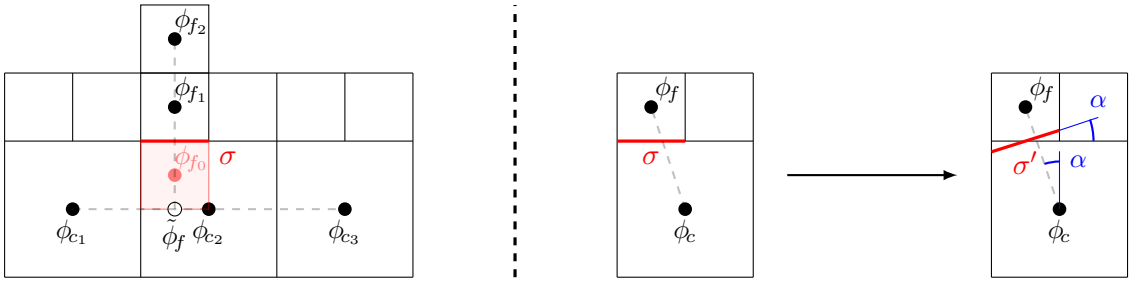
The interfaces of a subdomain Ω_i are denoted by Γ_i, j with $j = 1$ for the left interface, $j = 2$ for the lower interface, $j = 3$ for the right interface and $j = 4$ for the upper interface. The jump matrices $B_{i,j}$ and the discrete Lagrange multipliers $\Lambda_{i,j}$ use the same notation. For instance $\Lambda_{0,4}$ belongs to the discrete trace space associated to Ω_0 on its upper interface $\Gamma_{0,4}$. In this example, Ω_3 hosts all the Lagrangian multipliers at its interfaces while Ω_1 and Ω_2 host respectively the Lagrange multipliers of their lower and left interfaces.

The main issue with Mortar discretizations is to determine an appropriate preconditioning to solve effectively the algebraic saddle-point problem arising from the discretization.

6.2.2 Composite discretizations using interpolation

The issue of the discretization at coarse-fine interfaces amounts to define a scheme for computing edge values of a variable (eg. centered or upwind) or of its gradient on the fine side and on the coarse side of the interface. A common approach consists in defining a ghost-cell at the fine level using an interpolation scheme involving neighboring cells. This allows the definition of the fine edge value with a stencil involving only fine grid cells. The coarse edge values would be then deduced by averaging the fine edge values.

We first review a few techniques for computing such edge values at the fine level for the example of an edge normal gradient. Then we give the actual discretization used to computing centered or upwind values and edge gradients as well in our AMR implementation.



(a) *Ghost cell interpolation* (b) *Non-orthogonality* (c) *Measure correction*

FIG. 6.12 – *Different choices of composite stencils.*

Common approaches

When developing a second order Multigrid-SAMR method for the Poisson problem in 2D [100], D.F. Martin and K.L. Cartwright used the following procedure to compute the fine edge normal gradient. Let us consider the configuration of figure 6.12. First a quadratic interpolation with coarse 1D stencil $\{\phi_{c_1}, \phi_{c_2}, \phi_{c_3}\}$ is used to determine intermediate value $\tilde{\phi}_f$, i.e. in the present configuration:

$$\tilde{\phi}_f = \frac{5}{32}\phi_{c_1} + \frac{15}{16}\phi_{c_2} - \frac{3}{32}\phi_{c_3}$$

Note that $\tilde{\phi}_f$ is aligned with ϕ_{f_1} and ϕ_{f_2}. Quadratic interpolation is used again with the fine 1D stencil $\{\phi_{f_1}, \phi_{f_2}, \tilde{\phi}_f\}$ to compute "ghost cell" value ϕ_{f_0}:

$$\begin{aligned}\phi_{f_0} &= \frac{2}{5}\phi_{f_2} - \phi_{f_1} + \frac{8}{5}\tilde{\phi}_f \\ &= \frac{2}{5}\phi_{f_2} - \phi_{f_1} + \frac{1}{4}\phi_{c_1} + \frac{3}{2}\phi_{c_2} - \frac{3}{20}\phi_{c_3}\end{aligned}$$

Finally, the normal edge gradient at face σ is calculated as:

$$(\nabla_n \phi)_{f_1,\sigma} = \frac{\phi_{f_0} - \phi_{f_1}}{h_f}$$

with h_f the space step of the fine level. Therefore it yields a 2D stencil, coupling coarse cells and fine cells. The second order accuracy of this scheme has been verified on the Poisson problem in [100]. According to the authors quadratic interpolations are compulsory for reaching second order accuracy. This coarse-fine discretization was also used for cell-by-cell refinement in [110].

Linear interpolation is also often used [50, 97] in place of quadratic interpolation for computing intermediate values. An even simpler choice was proposed in [36, 97], where only one coarse cell and one fine cell define the stencil of the normal gradient at σ on the fine level side:

$$(\nabla_n \phi)_{f_1,\sigma} = \frac{\phi_{c_2} - \phi_{f_1}}{|x_{c_2} - x_{f_1}|}$$

The finite volume scheme was proved to converge in [36] provided the number of non conforming interfaces is not too large; indeed, the numerical flux is not consistent on non conforming interfaces. (Note however, such a two point scheme for the flux has the advantage of preserving

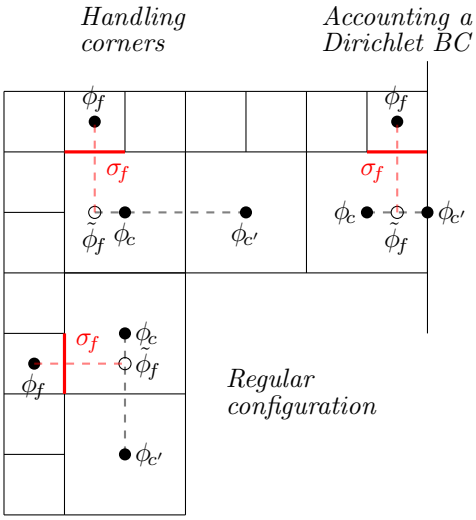


FIG. 6.13 – *Composite stencils actually used in our AMR implementation.*

the physical bounds [56]). The theoretical convergence result was confirmed by numerical experiments [36, 97]. One could also improve the computation of the numerical diffusive flux on non conforming interfaces [54] by shifting the fine faces by an angle α as shown on figure 6.12 (right), and hence obtaining proper orthogonality. This scheme can be very simply implemented by dividing the measure of the fine face by $\cos\alpha$.

High order discretizations at coarse-fines interfaces were also investigated by some authors, see for instance [133].

Actual implementation

In our implementation of the SAMR method, we choose to use linear interpolation to define the intermediate value $\tilde{\phi}_f$ depicted on figure 6.13. Three specific configurations are identified for the definition of the linear interpolation stencil $\{\phi_c, \phi_{c'}\}$. In the regular configuration, the nearest coarse values are chosen. In the case of a corner, $\tilde{\phi}_f$ lies outside the stencil, hence an expected degradation of the accuracy of the scheme. Finally near a physical boundary a fictitious coarse value $\phi_{c'}$ is defined depending the boundary condition. In practice, level grid faces are tagged whether they are purely internal, on an interface with a coarser level, on an interface with a finer level or on a physical boundary.

In any case the intermediate value $\tilde{\phi}_f$ is expressed as:

$$\begin{aligned} \tilde{\phi}_f &= \alpha\phi_c + (1-\alpha)\phi'_c \\ \alpha &= |\tilde{x}_f - x'_c|/|x_c - x'_c| \end{aligned}$$

The fine grid normal gradient at face σ is then determined by:

$$(\nabla_n \phi)_{f,\sigma} = \frac{\tilde{\phi}_f - \phi_f}{|\tilde{x}_f - x_f|}$$

Likewise, a centered fine edge value is calculated as:

$$\begin{aligned}\phi_{\sigma_f} &= \alpha\tilde{\phi}_f + (1-\alpha)\phi_f \\ \alpha &= |x_{\sigma_f} - \tilde{x}_f|/|x_f - \tilde{x}_f|\end{aligned}$$

and for an upwind fine edge value:

$$\begin{aligned}\phi_{\sigma_f} &= \alpha\tilde{\phi}_f + (1-\alpha)\phi_f \\ \alpha &= 1 \text{ if } u_{\sigma_f}\cdot n_{\sigma_f} < 0, \text{else } 0\end{aligned}$$

For the consistency of the discretization, coarse level edge value are deduced from the average of the fine level edge values:

$$|\sigma_c|\phi_{\sigma_c} = \sum_{\sigma_f\subset\sigma_c} |\sigma_f|\phi_{\sigma_f}$$

In our implementation, this procedure is generalized to SAMR grids with any refinement factor. Several simple tests are used to verify the implementation of coarse-fine discretization, for instance the Poisson problem, the advection problem, and finally the full projection scheme given in part I with smooth source terms.

6.3 Multigrid methods

Multigrid methods are very efficient for dealing with diffusion and advection equations, which need to be solved at each iteration of the prediction-correction algorithm. We give here an idea of the method when designed for uniform grids, and then turn to a particular choice of implementation in the case of composite grids, which was adapted from [8, 100].

6.3.1 Multigrid on uniform grids

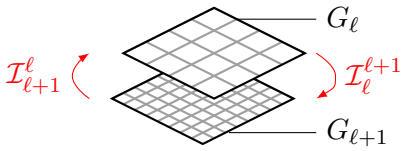


FIG. 6.14 – *Prolongation and restriction operators on two* level grids.

Multigrid methods are best understood with elliptic problems, because of the spectral properties of elliptic operators though all the following results can be applied to more complex operators (eg. advection equation, advection-diffusion equation).

Consequently we will consider in this section the discrete Poisson problem (6.1.1) on a square domain Ω with a hierarchy of Cartesian grids G_ℓ, $\ell \geq 1$ after the notations introduced in the first chapter. For the record two grids G_ℓ and $G_{\ell+1}$ are related by a constant refinement factor $n_{\text{ref}} = h_{\ell+1}/h_\ell$. Each vector and matrix associated with grid G_ℓ bears the subscript "(ℓ)". For instance, the algebraic problem on G_ℓ states:

$$A_{(\ell)}u_{(\ell)} = f_{(\ell)}$$

Two intergrid operators are defined: the *prolongation* operator $\mathcal{I}_\ell^{\ell+1}$ and the *restriction* operator $\mathcal{I}_{\ell+1}^{\ell}$ which are intended to approximate a discrete variable respectively from G_ℓ to $G_{\ell+1}$ and from $G_{\ell+1}$ to G_ℓ. We will define the prolongation operator as a bilinear interpolation and the restriction operator as a weighted average, though other choices exist.

Historical account

Multigrid methods are iterative methods for solving linear and non-linear problems which take advantage of the representation of a problem at different space scales. Three ideas are at the basis of multigrid: the use of *smoothers*, the *correction scheme* and the use of a *good initial guess*.

Some of these ideas can be traced back to the beginning of the XXth century. Back then, relaxation methods were very popular among the engineering community to solve linear systems. Despite their simplicity, they were plagued with a very slow convergence rate and they were not scalable. One way to improve the convergence of relaxation methods was to find a good initial guess. In the 1920's, R. Southwell introduced the idea of using a much coarser auxiliary grid to compute an approximation to the discrete solution defined on the original grid. Once interpolated back on the original grid, this approximation would make a very good initial guess. This was formalized in his 1935 paper [121] and later in [122]. The idea could be further extended using a set of even coarser grids in order to speed up the computation of initial guess. It leads to the following algorithm, which starts at the coarsest grid $\ell = 1$ with an initial guess for $u_{(1)}$:

SouthwellRelax(ℓ,$u_{(\ell)}$)
1: $u_{(\ell)} \leftarrow \mathcal{S}(A_{(\ell)}, f_{(\ell)}; u_{(\ell)}, \nu)$ with $\nu > 0$ and $|f_{(\ell)} - A_{(\ell)}u_{(\ell)}| < \varepsilon$
2: **if** $\ell < \ell_{\max}$ **then**
3: call SouthwellRelax($\ell + 1$,$\mathcal{I}_\ell^{\ell+1}u_{(\ell)}$)

It was already a great improvement over the usual practice of relaxation methods. However this algorithm, which could be classified as *one way multigrid* [52], did not take full advantage of the coarser problems. It is only in the 1960's that the very first multigrid algorithm was introduced, with the works of R.P. Fedorenko in the case of the Laplace equation (see [62] for numerical experiments and [63] for an analysis of the algorithm). As outlined in the previous chapter, smoothers eliminate the oscillatory components of the error in a few iterations but take a lot of time to converge on the smooth components. Fedorenko proposed to solve the error equation to eliminate the smooth components of the error, but on a much coarser grid in order to make this step less computationally expensive: "the computer operating time necessary for this can be reckoned insignificant."[62]. The error would then be interpolated back on the original grid to correct the solution. This cycle is repeated again until convergence. The method proposed by Fedorenko was using only two levels and it started with $\ell = 2$ and $r_\ell = f_{(2)}$:

FedorenkoMG(ℓ,$r_{(\ell)}$)
1: **repeat**
2: $u_{(\ell)} \leftarrow \mathcal{S}(A_{(\ell)}, r_{(\ell)}; u^{(\ell)}, \nu)$
3: $r_{(\ell)} \leftarrow f_{(\ell)} - A_{(\ell)}u_{(\ell)}$
4: **if** $\ell > 1$ **then**
5: $u_{(\ell)} \leftarrow u_{(\ell)} + \mathcal{I}_{\ell-1}^{\ell}$FedorenkoMG($\ell - 1, \mathcal{I}_\ell^{\ell-1}r_{(\ell)}$)
6: **if** $\ell < \ell_{\max}$ **then return** $u_{(\ell)}$

7: **until** $|r_{(\ell)}| < \varepsilon$

While this algorithm is pretty close to the current form of multigrid algorithms, Fedorenko did not investigate the effect of intergrid operators nor the correction scheme on the Fourier components of the error. The ideas introduced by Southwell and Fedorenko were the starting point of the fundamental work of A. Brandt [31] in the 1970's which laid the basis of today's research on multigrid methods. A standard multigrid iteration reads [52], starting with $\ell = \ell_{\max}$ and $r_\ell = f_{(\ell_{\max})}$:

StandardMG(ℓ,$r_{(\ell)}$)
1: **if** $\ell = 1$ **then**
2: $\quad u_{(\ell)} = A_{(\ell)}^{-1} r_{(\ell)}$
3: **else**
4: $\quad$ **for** $i = 1 \dots \mu_{(\ell)}$ **do**
5: $\quad\quad u_{(\ell)} \leftarrow \mathcal{S}(A_{(\ell)}, r_{(\ell)}; u^{(\ell)}, \nu_1)$
6: $\quad\quad r_{(\ell)} \leftarrow f_{(\ell)} - A_{(\ell)} u_{(\ell)}$
7: $\quad\quad u_{(\ell)} \leftarrow u_{(\ell)} + \mathcal{I}_{\ell-1}^{\ell}$StandardMG($\ell - 1$,$\mathcal{I}_{\ell}^{\ell-1} r_{(\ell)}$)
8: $\quad\quad u_{(\ell)} \leftarrow \mathcal{S}(A_{(\ell)}, r_{(\ell)}; u^{(\ell)}, \nu_2)$
return $u_{(\ell)}$

Different variations of the algorithm are obtained with different choices for μ.

Interpretation

Let us write the simple two level multigrid cycle with no post-smoothing:

TwoLevelMG($u_{(2)}$)
1: $u_{(2)} \leftarrow \mathcal{S}(A_{(2)}, r_{(2)}; u_{(2)}, \nu)$
2: $r_{(2)} \leftarrow f_{(2)} - A_{(2)} u_{(2)}$
3: $u_{(1)} \leftarrow A_{(1)}^{-1} \mathcal{I}_2^1 r_{(2)}$
4: $u_{(2)} \leftarrow u_{(2)} + \mathcal{I}_1^2 u_{(1)}$

At the fine level $\ell = 2$, ν smoothing iterations are performed on the original linear system $A_{(2)}u(2) = f_{(2)}$, starting with the provided initialization for $u_{(2)}$. Then the residual $r_{(2)}$ is transfered to the coarse level $\ell = 1$ with the restriction operator $\mathcal{I}_2^1$, providing the right hand side to the coarse level error equation. Upon solving the error equation, the error $u_{(1)}$ is transfered to the fine level with the prolongation operator $\mathcal{I}_1^2$ to correct the solution $u_{(1)}$.

The ν smoothing iterations eliminate the high frequency components of the error to the solution $u_{(2)}$. On the other hand, the direct resolution at the coarsest level (line 3) ensures that the remaining low frequency components of the error are completely removed.

To conclude, the definite features of multigrid methods are the nested nature of the smoothing of the different components of the error and the coarse-grid correction procedure. Multigrid methods are scalable and SAMR grids provide a natural framework for these methods.

6.3.2 Multigrid on adaptive grids

Multigrid-AMR algorithms

Multigrid methods extend intuitively to SAMR grids. The residual at a given level is not defined solely with respect to the finest level. Rather the residual is composite in that the regions of $\cup_{\ell_1 \le \ell \le \ell_2} \Omega_\ell$ are somehow associated to a multigrid process between the corresponding

We want to solve $Ax = b$ by smoothing the error, which satisfies $Ae = r$ with $r = b - Ax$. Each V-Cycle starts with **relax**(ℓ_{finest}).

relax(ℓ):

Smooth composite error
$e = 0$
smooth($A_{\text{comp(c)}}$,r,e)
$\tilde{x} = x$
$x = x + e$

Prepare coarser residual
$r_c|_{\Omega_c \setminus \Omega} = b_c - A_{\text{comp(cf)}} x_c$
$r_c|_{\Omega_c \cap \Omega} = \mathbf{rest}(\Omega_c \cap \Omega_f, r - A_{\text{comp(c)}} e)$

Relax on coarser level
call **relax**($\ell - 1$)

Correction from coarser level
$\varepsilon = \mathbf{intp}(\Omega_c \cap \Omega, e_c)$
$e = e + \varepsilon$
$x = \tilde{x} + e$

Remarks:

- Coarsest level: direct resolution
- Finest level defines its own residual

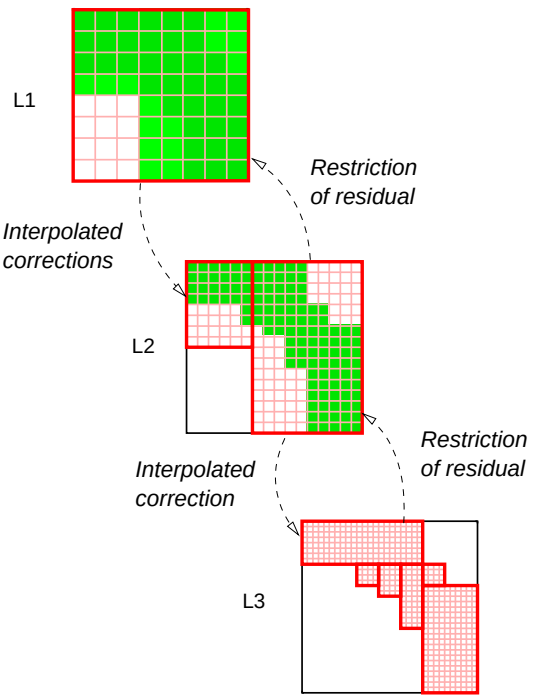


FIG. 6.15 – *Example of multigrid-AMR algorithm with a two levels of refinement.*

overlapping regions of level grids for levels $\ell \in [\ell_1, \ell_2]$. A SAMR multigrid V-Cycle can be viewed as several classical multigrid V-Cycle coupled at coarse-fine interfaces through the definition of the residual of the original problem. A example of such procedure is illustrated on figure 6.15. In practice, depending on the type of discretization, on the use of time refinement or not, several complications may arise.

The possibility of extending multigrid methods to locally refined grids was first pointed by A. Brandt [31, pp. 359–360]. Regarding the present class of SAMR methods, earliest works include [6] where a multigrid-SAMR extension of a "particle-particle particle-mesh" method for finite difference discretizations was introduced. This served as a basis of the multigrid-AMR solver proposed later by D.F. Martin and K.L. Cartwright [100] for solving elliptic problems on finite-volume discretizations. Our multigrid-AMR solver — presented hereafter — is actually a simplification of the original algorithm of the latter authors.

Multigrid-AMR algorithms based on the original aforementioned methods were involved for solving complex problems including, but not limited to the incompressible Euler equations [101], high order elliptic problems [16], the variable density Navier-Stokes equations with time refinement [7] and two-phase flows in porous media [107].

Implemented algorithm

We now describe the Multigrid-AMR algorithm as it was implemented in our numerical code, given under a simplified form hereafter. The "RelaxLevel" function is a recursive function which performs a complete V-Cycle for solving a linear system, itself obtained from the discretization

of a hyperbolic or elliptic partial differential equation on the composite grid:

$$Au = b$$

Within the grid hierarchy, each patch $\Omega_{\ell,i}$ at level ℓ, $\ell \leq \ell_{\max}$ features a local version of this linear system:

$$A_{\ell,i} u_{\ell,i} = b_{\ell,i}$$

The linear system sys=$\{A_{\ell,i}, b_{\ell,i}\}$ is assembled by function "asbOpr(sys,var,compwc,compcf)". The latter always take into account the physical boundary conditions attached to variable "var" which are set by function "setPhysGC(var)". The discretization at interfaces with the finer level (located inside the patch) is taken into account when "compwf" flag is activated. Likewise interfaces with the coarser level (located at the boundaries of the patchs) are accounted for if "compwc" is activated. In practice the linear system "sys" is never fully reassembled, only relevant rows of the linear system are redefined.

The "RelaxLevel" function is intended to be called first from the finest level, i.e. RelaxLevel($\ell_{\max}$, n_{dd}, n_{sm}). At the finest level, the algorithm first computes the residual of the current level $r_{\ell,i}$ (lines 9 to 14). This residual is defined with respect to the composite problem, i.e. both interfaces with the coarser and the finer level are considered. Prior to the calculation of the residual, the ghost-cells of all level variables $u_{\ell,i}$ are synchronized with "synAll" function (line 10).

The error equation of the current level is then relaxed (lines 17 to 26). The outer loop controls the number of iterations of the ghost-cell domain decomposition technique, set to n_{dd}. The inner loop over each patch of the level performs n_{sm} local relaxations with a Gauss-Seidel smoother. The linear system associated with the error is coupled with the coarser level only and is provided with homogeneous Dirichlet conditions. Once the error is relaxed on $\Omega_{\ell,i}$, the ghost-cells of each neighboring patches are updated with "synFromSrc" function (line 26). At the end of the level relaxation procedure the main unknown is updated (lines 28 to 30).

After that, the residual of the coarser level is calculated. First, the residual of the original problem over the non-overlapping region is defined (lines 31 to 33). Therefore, the linear system line 35 is assembled for the composite problem, i.e. by taking into account both the interface with coarser and finer levels with respect to level $\ell - 1$. This amounts to start a new multigrid cycle on the subset of the coarser level which does not overlap with the current level. In practice, this coarser residual is defined everywhere on the coarser level. Indeed, on overlapping regions between the current level and the coarser level, the coarser residual will feature the residual of the error equation of the current level (lines 34 to 38). This correspond to a continuation the multigrid cycle started at the current level or at a finer level if applicable.

Once the coarser residual is defined, the same operations are executed recursively up to the coarsest level. At the coarsest level ($\ell = 0$), the error equation is solved directly up to machine precision (lines 1 to 7). The V-Cycle then continues down to the finest level again, by importing the correction calculated by the coarser levels to improve the solution (lines 40 to 49).

RelaxLevel($\ell, n_{\text{dd}}, n_{\text{sm}}$)
1: **if** $\ell = 0$ **then**
2: asbSys(sys=$\{A_\ell, b_\ell\}$, var=e_ℓ)
3: $b_\ell \leftarrow r_\ell$
4: solve(sys=$\{A_\ell, b_\ell\}$, var=e_ℓ)
5: $u_\ell \leftarrow u_\ell + e_\ell$
6: setPhysGC(var=u_ℓ)
7: setPhysGC(var=e_ℓ)

else
  if $\ell = \ell_{\max}$ **then**
    synAll(var=$(u_{\ell,i})_{G_{\ell,i}\subset G_\ell}$)
    for $G_{\ell,i} \subset G_\ell$ **do**
      asbOpr(sys=$\{A_{\ell,i}, b_{\ell,i}\}$, var=$u_{\ell,i}$, compwc=$T$, compwf=$T$)
      $r_{\ell,i} \leftarrow s_{\ell,i} - (A_{\ell,i}u_{\ell,i} - b_{\ell,i})$
      $e_{\ell,i} \leftarrow 0$
  $\hat{u}_{\ell,i} \leftarrow u_{\ell,i} \quad \forall G_{\ell,i} \subset G_\ell$
  $e_{\ell-1,i} \leftarrow 0 \quad \forall G_{\ell-1,i} \subset G_{\ell-1}$
  repeat n_{dd} times
    for $G_{\ell,i} \subset G_\ell$ **do**
      asbOpr(sys=$\{A_{\ell,i}, b_{\ell,i}\}$, var=$e_{\ell,i}$, compwc=$T$, compwf=$F$)
      $r'_{\ell,i} \leftarrow r_{\ell,i} - (A_{\ell,i}e_{\ell,i} - b_{\ell,i})$
      $b_{\ell,i} \leftarrow r'_{\ell,i}$
      $e'_{\ell,i} \leftarrow 0$
      repeat n_{sm} times
        smooth(sys=$\{A_{\ell,i}, b_{\ell,i}\}$, var=$e'_{\ell,i}$)
      $e_{\ell,i} \leftarrow e_{\ell,i} + e'_{\ell,i}$
      synFromSrc(var=$e_{\ell,i}$)
  setPhysGC(var=$(e_{\ell,i})_{G_{\ell,i}\subset G_\ell}$)
  $u_{\ell,i} \leftarrow u_{\ell,i} + e_{\ell,i} \quad \forall G_{\ell,i} \subset G_\ell$
  synAll(var=$(u_{\ell,i})_{G_{\ell,i}\subset G_\ell}$)
  setPhysGC(var=$(u_{\ell,i})_{G_{\ell,i}\subset G_\ell}$)
  for $G_{\ell-1,i} \subset G_{\ell-1}$ **do**
    asbOpr(sys=$\{A_{\ell-1,i}, b_{\ell-1,i}\}$, var=$u_{\ell-1,i}$, compwc=$T$, compwf=$T$)
    $r_{\ell-1,i} \leftarrow s_{\ell-1,i} - (A_{\ell-1,i}u_{\ell-1,i} - b_{\ell-1,i})$
  for $G_{\ell,i} \subset G_\ell$ **do**
    asbOpr(sys=$\{A_{\ell,i}, b_{\ell,i}\}$, var=$e_{\ell,i}$, compwc=$T$, compwf=$F$)
    $r'_{\ell,i} \leftarrow r_{\ell,i} - (A_{\ell,i}e_{\ell,i} - b_{\ell,i})$
    for $G_{\ell-1,j} \subset G_{\ell-1}$ s.t. $G_{\ell-1,j} \cap G_{\ell,i} \neq \{\emptyset\}$ **do**
      $r_{\ell-1,j}|_{G_{\ell,i}\cap G_{\ell-1,j}} \leftarrow$ rest(var=$r'_{\ell,i}$, set=$G_{\ell,i} \cap G_{\ell-1,j}$)
  RelaxLevel($\ell - 1, n_{\text{dd}}, n_{\text{sm}}$)
  for $G_{\ell,i} \subset G_\ell$ **do**
    $e'_{\ell,i} \leftarrow 0$
    for $G_{\ell-1,j} \subset G_{\ell-1}$ s.t. $G_{\ell-1,j} \cap G_{\ell,i} \neq \{\emptyset\}$ **do**
      $e'_{\ell,i}|_{G_{\ell,i}\cap G_{\ell-1,j}} \leftarrow e'_{\ell,i}|_{G_{\ell,i}\cap G_{\ell-1,j}} +$ intp(var=$e_{\ell-1,j}$, set=$G_{\ell,i} \cap G_{\ell-1,j}$)
    $e_{\ell,i} = e_{\ell,i} + e'_{\ell,i}$
    $u_{\ell,i} = \hat{u}_{\ell,i} + e_{\ell,i}$
  synAll(var=$(e_{\ell,i})_{G_{\ell,i}\subset G_\ell}$)
  setPhysGC(var=$(e_{\ell,i})_{G_{\ell,i}\subset G_\ell}$)
  synAll(var=$(u_{\ell,i})_{G_{\ell,i}\subset G_\ell}$)
  setPhysGC(var=$(u_{\ell,i})_{G_{\ell,i}\subset G_\ell}$)

Compressible Euler equations

The pressure-correction algorithm presented in part I straightforwardly to AMR if no time refinement is involved. A simplified form of the algorithm implemented in our code is presented below.

The prediction step is performed by solving the equivalent composite problem to (4.2.3) with the multigrid AMR algorithm formerly introduced (lines 5 and 6). The non-linear projection-correction step is carried out with a fixed point procedure. Though less robust than Newton's method, it was found to converge with no more than 5 iterations in our numerical tests. Within these inner iterations, the density and the energy balance are solve on the AMR hierarchy (lines 22 to 23).

Once the resolution of the problem is completed for the current timestep, the adaptive grid is regenerated according to the new state of flow variables (line 25). The newly calculated solution is transfered to the new hierarchy (line 26) either through a direct copy of cell values (eg. if an old patch and new patch overlap) or by using a bilinear interpolation of coarser cell values from the old AMR hierarchy. The linear system of each patch are also reinitialized (line 27). The latter remain unchanged during a complete timestep at the exception of the rows affected by the coarse-fine discretization.

SolveEuler
1: initAMR()
2: initGrid()
3: initOpr()
4: **while** $t < t_{\text{final}}$ **do**
5: solvePredUx()
6: solvePredUy()
7: setBC(var=$\tilde{u}_x$, physbc=T, compwc=T, compwf=T)
8: setBC(var=$\tilde{u}_y$, physbc=T, compwc=T, compwf=T)
9: setBC(var=ρ, physbc=T, compwc=T, compwf=T)
10: $R \leftarrow \ldots$ according to (1.3.8)
11: $p^* \leftarrow \ldots$ according to (1.2.14d)
12: synAll(var=p^*)
13: $i \leftarrow 0$
14: **while** ($i = 0$ or $|\delta p|_{L_\infty} > \varepsilon$) or ($i < 50$) **do**
15: $i \leftarrow i + 1$
16: $u_x^* \leftarrow \ldots$ according to (4.2.4)
17: $u_y^* \leftarrow \ldots$ according to (4.2.4)
18: synAll(var=u_x^*)
19: synAll(var=u_y^*)
20: setBC(var=u_x^*, physbc=T, compwc=T, compwf=T)
21: setBC(var=u_y^*, physbc=T, compwc=T, compwf=T)
22: solveDensity()
23: solveInternalEnergy()
24: $p^* \leftarrow \ldots$ according to (1.2.14d)
25: regrid()
26: reinitAMR()
27: initOpr()

Chapter 7

APPLICATION TO COMPRESSIBLE FLOWS

In this chapter, three numerical tests are carried out in order to assess our adaptive pressure-correction scheme on shock hydrodynamic problems. The first section is dedicated to a 2D Riemann problem already tested in part I on an uniform grid. The second section addresses a more complex problem with Mach reflections. Mach reflection were originally discovered by Ernst Mach in the late 1870's. A large body of research on this phenomena was initiated during Manhattan Project [68] and it constitutes nowadays a well-documented and challenging benchmark. The numerical calculations performed in this chapter will ultimately give an appreciation on the suitability of our adaptive pressure-correction scheme for the weakly and highly compressible flow problems in nuclear reactors.

The numerical method presented in the three previous chapters was implemented in our numerical code MNFD, specially created for the purpose of this work. All the numerical results were post-processed with VisIt visualization software [41], using in our code the HDF5 format with a hierarchy compatible with the import format for Chombo AMR code [1] in VisIt.

7.1 2D Riemann problem

7.1.1 Problem setting

This preliminary AMR test deals with the 2D Riemann problem 12 presented in part I. Three levels of refinement are used, reaching a space resolution of $h_3 = 1/400$ at the finest level with a refinement factor $n_{\text{ref}} = 2$. The refinement criteria is based on the variations of the density. A cell K belonging to the level grid G_ℓ is flagged for refinement if the following condition is satisfied:

$$\max_{L \in \mathcal{N}(K)} |\rho_K - \rho_L| > h_\ell \cdot \varepsilon \tag{7.1.1}$$

with $\varepsilon = 1$. The adaptive grid generation algorithm presented in chapter 5 is used with the "standard Laplacian" for edge detection. The efficiency is set to $emin = 0.6$ and the shape of the patches is controlled by parameters $rvmin = 0.3$ and $vmin = lmin = 0$.

The compressible Euler equations are solved with the Multigrid-AMR method presented in Chapter 6. The stopping criterion for the multigrid solver is 10^{-10} on the absolute L^∞ norm of the composite residual. In each multigrid solve, the smoothing procedure on level grids consists in three outer iterations over all the patches, with for each patch two smoothing steps (Gauss-Seidel) with synchronization of ghost-cell values. The iterations of the fixed point procedure for the non-linear projection-correction step end when the absolute L^∞ norm of the pressure

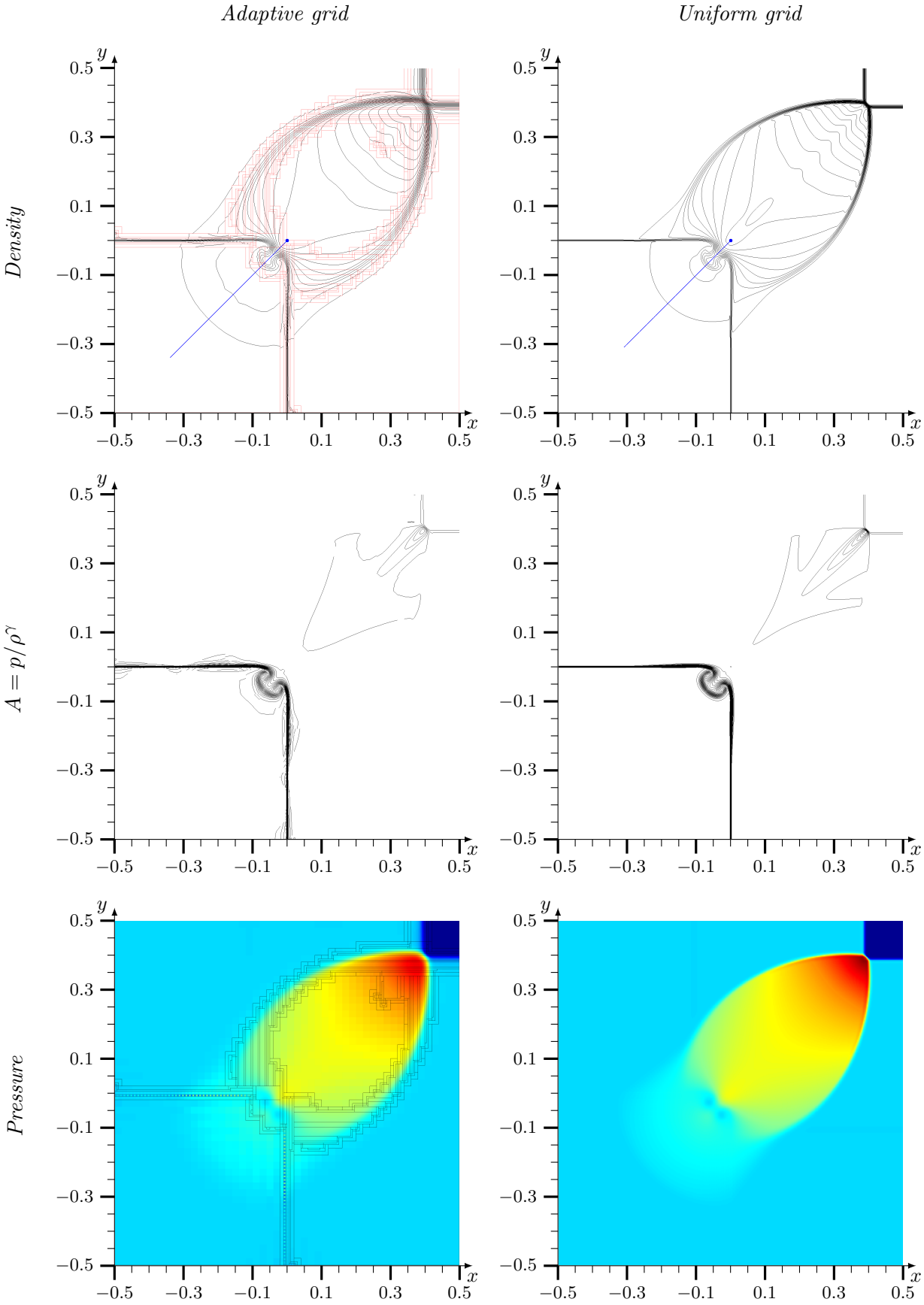


FIG. 7.1 – *Top to bottom: isopycnics (30 values) with a streamline leaving from* $(0, 0)$*, iso-values of* $A = p/\rho^\gamma$ *(30 values) and pressure field at* $t = 0.25$ *with the patches of the three levels of refinement.*

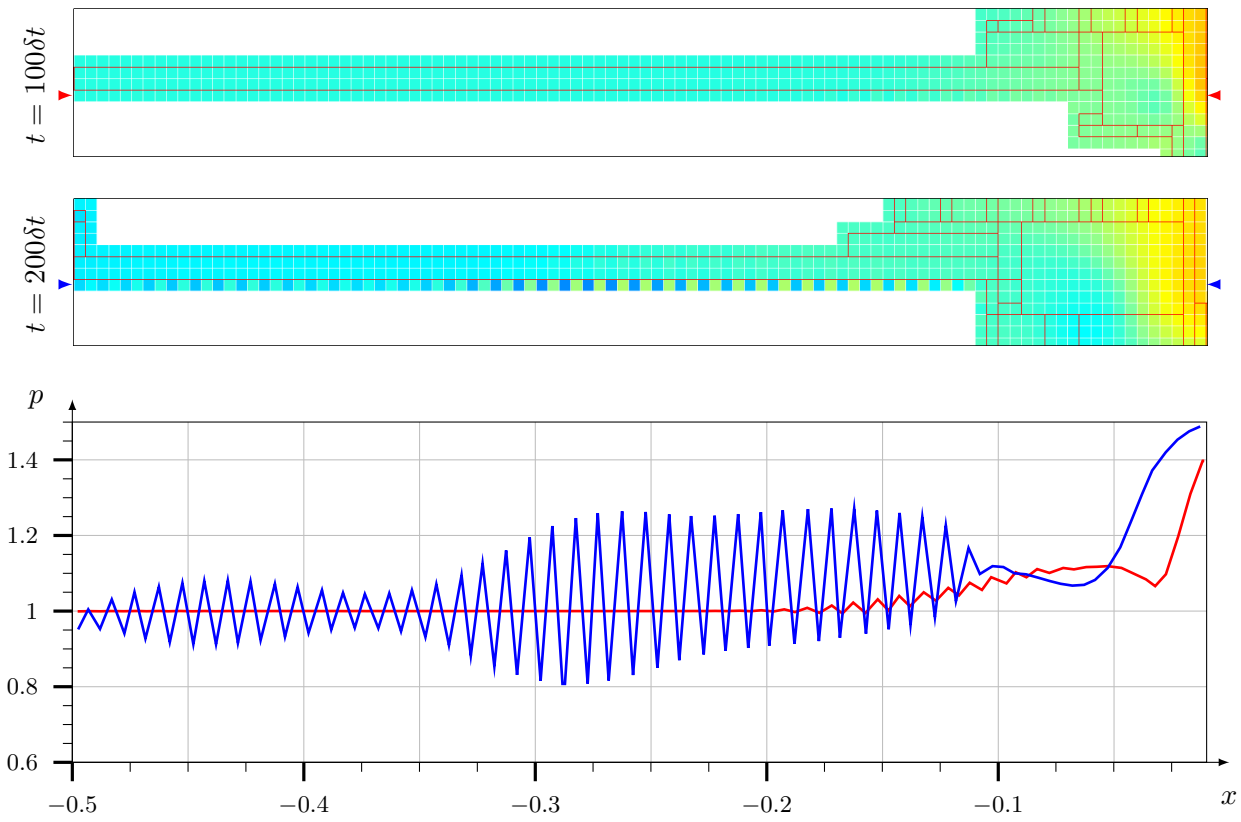


FIG. 7.2 – *Top: close up view of the pressure field belonging to the level grid of the second level of refinement. The red lines indicate the boundaries of the patches belonging to the third level of refinement. Bottom: pressure profile at* $t = 100\delta t$ *(in red) and* $t = 200\delta t$ *(in blue) along the line delimited by the two arrows in the pressure fields. For the record the timestep is* $\delta t = 1.25 \times 10^{-3}$.

increment with respect to the fixed point iterations is below 10^{-12}. Finally the outer iterations in time have a timestep tied to the finest grid step size as $\delta t = h_3/2 = 1.25 \times 10^{-3}$.

7.1.2 Numerical results

The results of the AMR simulation are compared on figure 7.1 to the numerical results obtained in part I with the same problem. The latter were calculated on an uniform grid of space step $h = 1/400$. The adaptive solution features the two static contact waves and the two straight shocks moving to the top right corner of the domain. Two bow shocks grow between the contact waves and the straight shocks. A jet is produced at the center of the domain towards the lower left corner. The isopycnics confirm the symmetry of the solution. The three levels of refinement are also symmetric, and they are concentrated at shocks, contacts and around the jet. Despite the increase of resolution (factor 6 between the finest level and the base grid) shocks are poorly calculated, which is quite unexpected for the two straight shocks present at initialization. The jet grows as expected, as evidenced by the trajectory of the streamline leaving from the center of the domain (blue line on figure 7.1).

However the main issue in this test is not the lack of resolution at shocks but the presence of spurious pressure oscillations which grow in amplitude with time. This numerical noise is

located along the two contact waves and is visible on the contours of $A = p/\rho^\gamma$. Figure 7.1 shows the pressure field on the second level of refinement ($h_2 = 1/200$). The boundaries of this second level are either physical boundaries or coarse-fine interfaces with the first level of refinement. The patches of the third level of refinement help to locate the coarse-fine interfaces with the latter. The spurious pressure modes are located solely on the composite grid of level 2, which has only one cell in height.

A comparison of the pressure profiles on the composite grid at $t = 100\delta t$ and $t = 200\delta t$ shows that the numerical disturbance grows in amplitude and is propagated along the contact wave to the domain boundary. It seems to originate from a "triple corner", i.e. a location at $x = -0.11$ where three corners from different levels are separated by only one cell. We believe the specific coarse-fine discretization at this corner is responsible for the generation of this numerical noise. Such spurious oscillations were never observed on further AMR tests of the same problem using a single level of refinement. This issue could be addressed by imposing a buffer of at least two cells between successive levels of refinement.

7.2 Double Mach Reflection

7.2.1 Introduction

In this section we present numerical simulations of a *Double Mach Reflection* problem (DMR) using the all-Mach solver and the SAMR method respectively introduced in parts 1 and 2. The problem under consideration is a shock diffraction with a concave corner. Depending the Mach number M_S of the shock, the heat capacity ratio γ of the fluid and the angle θ of the corner different complex structures unfold. In our case this will be a *Double Mach Reflection of type "DMR+" with an attached shock*, resulting from the interaction of a straight shock at $M_S = 10$ of an ideal diatomic gas with a corner inclined by $\theta = 30°$.

In 1989, M.J. Berger and P. Colella presented the first adaptive mesh refinement algorithm applied to the DMR problem [25]. The compressible Euler equations were solved using a second-order Godunov-type scheme [45]. The adaptive mesh refinement method was based on the patch-based AMR algorithm introduced by M.J. Berger and J. Oliger in [26]. The second order accuracy in space and a refinement factor of 4 on two levels allowed the authors to perform numerical simulations of the DMR problem at a resolution never reached before. A triple Mach stem configuration was observed for the first time for the DMR problem with low γ.

Since then DMR has become a standard benchmark for assessing numerical codes for compressible flows [130] and in particular AMR codes. Indeed the physics of DMR are rather well understood nowadays [21, 78, 93] both qualitatively (eg. shock configurations) and quantitatively (eg. shock polars) and have been largely studied both through numerical simulations [130, 69, 46] and physical experiments [22, 23, 46]. Moreover for the former the most complex flow structures are only solved beyond a certain level of accuracy, which makes this benchmark especially relevant for testing adaptive resolution methods.

In the original work of M.J. Berger and P. Colella [25] the numerical method was second order in space and the finest level had a space step of $h_2 = 1/320$. In contrast our all-Mach pressure correction scheme, presented in the first part of this thesis, is only first order in space and in our simulations, the space step at the finest level is $h_2 = 1/1600$. Our objective is to first evaluate the relevance of our numerical method with respect to the physics of DMR, then to determine how close to [25] — where a five times coarser grid is used — the results of our first order method can be.

In a first section, we recall some background knowledge on shock wave diffraction which will be at the basis of the analysis of our numerical results. Then a comparison of the simulation of the DMR problem on adaptive grid and uniform grid is presented. Finally a refined simulation of the DMR problem with adaptive refinement on the two Mach reflections is analyzed.

7.2.2 Shock diffraction

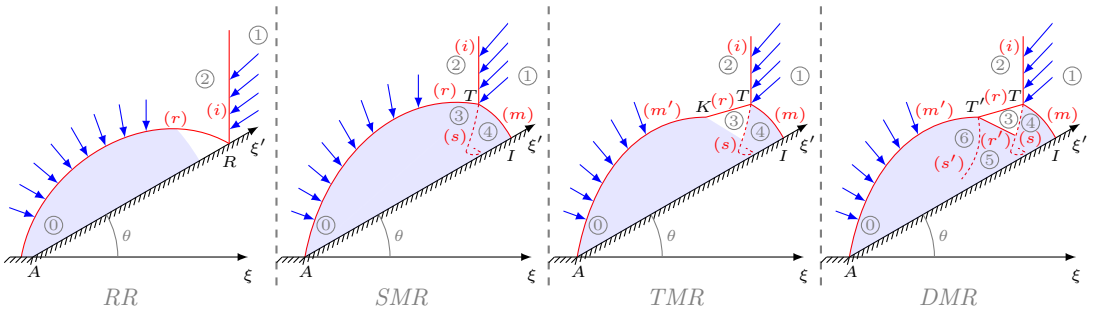


FIG. 7.3 – *Shock systems for a RR, SMR, TMR and DMR. Solid line is a shock wave, dashed lined is a contact wave. The subsonic regions in the self-similar frame of reference are in light blue. The pseudo-streamlines of states* (1) *and* (2) *are in blue ; note that the former converge to point A.*

Introduction

When a straight shock of Mach number M_S with uniform induced flow $(\mathbf{u}_2, p_2, \rho_2)$ moving through still air (p_1, ρ_1) encounters a concave corner with angle θ, the *shock wave diffraction* at the apex creates a complex self-similar structure which grows along the corner wall. Shock diffractions can be of four types [95], all represented on figure 7.3 : *Regular Reflection* (RR), *Single Mach Reflection* (SMR), *Transitional-Mach Reflection* (TMR) and *Double Mach Reflection* (DMR). G. Ben-Dor and I.I. Glass proposed in [22, 23] an empirical delimitation of the domains of these four shock diffraction in the (M_S, θ) plane, reproduced on figure 7.6 for a diatomic gas. A large body of studies is dedicated to the transition criteria between these different shock diffractions, see for instance [21] and references therein. Two specific frames of reference will be used, in addition to the laboratory frame of reference: the self-similar frame, more adapted to the study of the physics of shock diffraction globally, and the frame of reference attached to a triple point, more relevant for analyzing locally the corresponding oblique shock configurations.

The self-similar frame is defined through the change of variables $(x, y, t) \to (\xi, \eta, \zeta)$ with $\xi = x/t$, $\eta = y/t$ and $\zeta = t$. In this frame of reference instead of $\mathbf{u}$ we use the *pseudo-velocity* $\bar{\mathbf{u}} = \mathbf{u} - [x/t, y/t]^t$. As a result in a constant state, the uniform velocity field $\mathbf{u}$ in the laboratory frame of reference will be radial in the self-similar frame of reference, with center the point (u_x, u_y). The corresponding *pseudo-Mach number* is denoted $\bar{M} = |\bar{\mathbf{u}}|/a$, with $a = \sqrt{\gamma p/\rho}$ the speed of sound. Solving the compressible Euler equations (1.1.2) in the laboratory frame is

equivalent to solving the following stationary hyperbolic system in the self-similar frame [120]:

$$\widetilde{\mathrm{div}}(\rho\bar{\mathbf{u}}) = -2\rho \qquad \text{in } \Omega \qquad (7.2.1a)$$

$$\widetilde{\mathbf{div}}(\rho\bar{\mathbf{u}}\otimes\bar{\mathbf{u}}) + \widetilde{\nabla}p = -3\rho\bar{\mathbf{u}} \qquad \text{in } \Omega \qquad (7.2.1b)$$

$$\widetilde{\mathrm{div}}\left(\left(\frac{1}{2}\rho|\bar{\mathbf{u}}|^2 + \rho e + p\right)\bar{\mathbf{u}}\right) = -2(\rho|\bar{\mathbf{u}}|^2 + \rho e + p) \qquad \text{in } \Omega \qquad (7.2.1c)$$

$$\rho \geq 0, e \geq 0, p = (\gamma-1)\rho e \qquad (7.2.1d)$$

with $\widetilde{\mathrm{div}}$ and $\widetilde{\nabla}$ the divergence and the gradient operators defined in the (ξ,η) coordinate system. This system of equations is identical to the steady compressible Euler equations with the addition of source terms (-2ρ), $(-3\rho\bar{\mathbf{u}})$ and $(-2(\rho|\bar{\mathbf{u}}|^2+\rho e+p))$ respectively in the mass balance, in the momentum balance and in the total energy balance. An extensive analysis of the self-similar Euler equations in the context of shock diffraction is found in references [120, 83].

Deflection vs Reflection

As depicted on figure 7.4, a shock diffraction develops from the interaction between of two phenomena [93, 95, 78] : *flow deflection* and *shock reflection*. Upstream at the wedge apex A, a bow shock (b) deflects the flow along the corner wall. The bow shock can be either attached or detached at point A. It generates compressive (resp. expansive) waves towards state (3) when $p_0 < p_3$ (resp. $p_0 > p_3$). Downstream a shock reflection occurs with incident shock (i). The most simple reflection is the *Regular Reflection* (RR). As shown on figure 7.4 (right) the deflection of the streamline through the reflected shock of a RR ensures that the flow still verifies the wall boundary condition. In addition to the classical [92] limitation on the incidence angle α, such reflection can only subsist without the influence of the pressure perturbations from the apex, i.e. the flow is supersonic in region (0) in the self-similar frame (see figure 7.3).

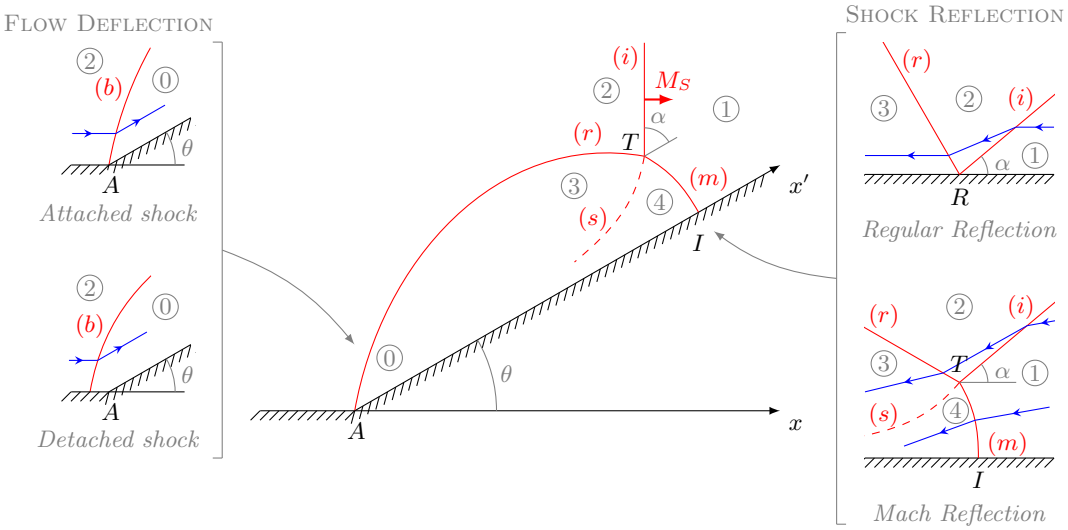


FIG. 7.4 – *Single Mach Reflection (center) ; flow deflections (left, laboratory frame of reference) and shock reflections (right, local frame of reference). Solid line is a shock wave, dashed lined is a contact wave.*

As suggested in [79] if the flow in region (0) is subsonic in the self-similar frame, a length scale — that could be for example the distance $[AR]$ — is communicated to the RR which turn to a more complex shock system. The reflection point R is detached and a third shock called *Mach stem* (m) links the reflection point to the wall at point I. The reflection now takes places at *triple point* T, located at the intersection between the three shocks. Another consequence of the pseudo-subsonic speed of the flow is the continuous curvature along the bow shock and the reflected shock at T, the latter being straight (see figure 7.4, right) without the influence of the compression waves [95]. This complex shock configuration is called *Mach reflection*, after Ernst Mach who discovered it in 1878 (see [89] for a historical account). A SMR is a shock diffraction featuring only a Mach reflection.

Note that the Mach stem has a finite length in contrast to (i) and (r), which is directly related to the length scale communicated to it [75]. Unsurprisingly (m) is perpendicular to the wall so that the streamlines close to the wall verify the boundary condition. The incoming flow from state (1) reaches the region left to T either through the two shocks (i) and (r) or through shock (m). The two flows from these two paths have different velocities, hence the appearance of a contact discontinuity — named *slipstream* or *vortex sheet*. The slipstream (s) splits the region left to T into state (3) and state (4).

Towards TMR and DMR

When the flow in state (3) becomes supersonic in the self-similar frame, the pressure perturbations coming from the apex through the subsonic channel of region (0) stop at the sonic arc delimiting states (0) and (3) in figure 7.3, middle right (see [95, 93]). This arc intersects with the reflected shock (r) at point K. Therefore (r) stays straight up to point K. Left to K the shock is curved under the influence of pressure perturbations in the subsonic channel. When the latter are of compressive type this shock diffraction configuration is called *Transitional-Mach Reflection*[1] (TMR).

If the compression waves converge to form a shock wave — denoted (r') in figure 7.3 — a second Mach reflection arises, defined by triple point T', incident shock wave (r), reflected shock wave (r'), Mach stem (m') and slipstream (s'). This shock diffraction pattern is named *Double Mach reflection*[2] (DMR). It was discovered in 1951 by D.R. White [129]. When the angle χ associated to T is lower than the angle χ associated to T', we have a DMR+ diffraction [21] (see figure 7.5). When the first slipstream reaches the wall it is being "rolled" towards the Mach stem, because of the compression waves. A jet is created in region (4), which has an impact on the curvature of the Mach stem. An extensive study of the wall-jetting effect is found in [75].

Modelling DMR

For analysing the interaction between the flow deflection at A and the shock reflection at I it is very convenient to use the self-similar frame of reference. In this self-similar frame of reference regions (0), (5) and (5) are subsonic while region (3) is supersonic. As explained previously, shock (r') is created by the compression waves coming from the apex. In self-similar frame these pressure signals are not coming from A but from $A_3 = A + \mathbf{u}_3$. In this stationary problem their farthest reach is determined by a_3 (the speed of sound in state (3)) i.e. by the arc of center A_3

[1] TMR is also referred to as *Complex Mach Reflection* (CMR) and *Transitional-Irregular Reflection* (TIR)
[2] DMR is also known as *Double Irregular Reflection* (DIR)

and radius a_3. Using a simple geometrical construction introduced by H. Li and G. Ben-Dor for TMR, it is possible to determine $\mathbf{u}_3$ and therefore predict the position of this arc.

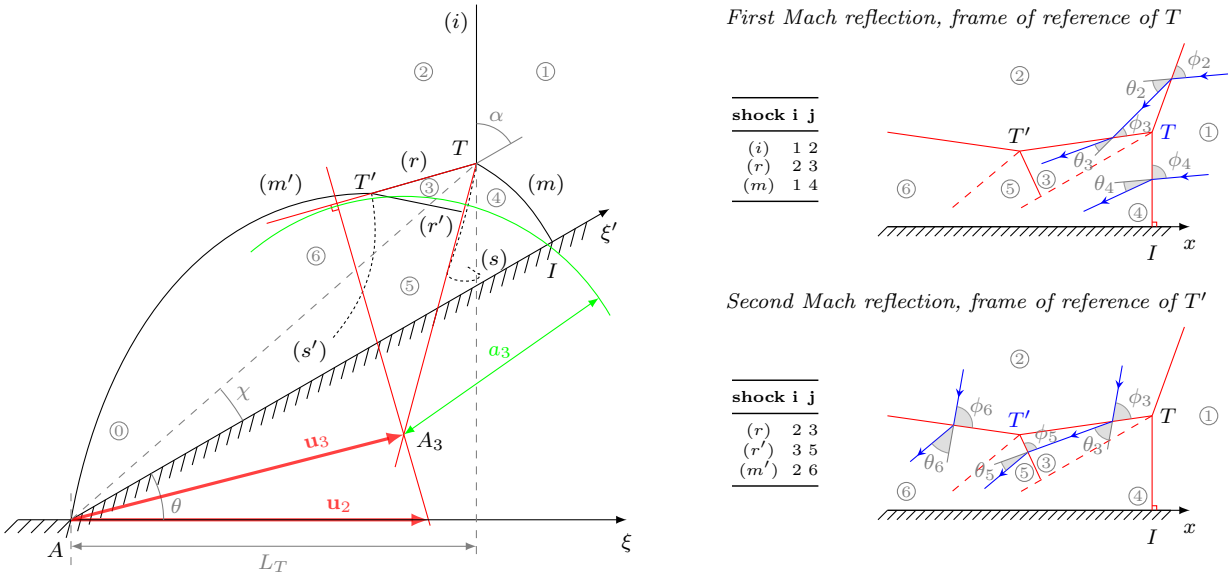


FIG. 7.5 – *Modelling DMR. Left: geometric construction (in red) for locating the leading pressure disturbance (in green) in the self-similar frame of reference. Right: notations for oblique shock (in red) refractions in the two triple shock systems.*

Let us focus on the refraction at shock (r) (figure 7.5) between states (2) and (3). According to classical oblique shock theory [92] incident velocity $\mathbf{u}_2$ and refracted velocity $\mathbf{u}_3$ at shock (r) have the same tangential component. As a result the vector $\mathbf{u}_3 - \mathbf{u}_2$ is orthogonal to (r). On figure 7.5, point A_3 will necessary be on the line perpendicular to (r) passing through A_2. Now let us switch to the self-similar frame: in the radial velocity field $\bar{\mathbf{u}}_3$ in state (3), which originates in A_3, we know that the streamline passing through T will be tangent to the slipstream (s). Thus A_3 will be at the intersection between the line coincident to the slipstream (s) and the line perpendicular to (r) passing through A_2.

Now we turn to the local Mach reflection at the two triple points. The flow being self-similar, triple points T and T' move with a constant radial direction from A with respective angles χ and χ' defined on figure 7.5. The distances between A and the projections of T and T' on the x-axis are denoted L_T and $L_{T'}$. Point T is tied to the incident shock (i) hence:

$$L_T = |\mathbf{u}_s|\Delta t$$

or $L_T = |\mathbf{u}_s|$ in the self-similar frame. Regarding the trajectory of the second triple point, the *Law-Glass assumption* states that point T' travels with the speed $\mathbf{u}_2$ of the incident shock induced flow:

$$L_{T'} = |\mathbf{u}_2|\Delta t$$

A physical justification of this assumption is found in [93]. A major concern over this approach is that it does not take into account the compression waves generated by the deflection [95], which is the very phenomenon causing the curvature reversal at T'. Another approach was proposed by H. Li and G. Ben-Dor in 1995 [95]. It consists in applying the general triple shock theory

to the two Mach reflections of DMR. This requires a stationary shock system, hence the use of the frame of reference attached to a triple point. This yields a complex non-linear system, the resolution of which gives access to all three thermodynamic states, oblique shock angles and the trajectory angle of the triple point. Instead, we will rather verify individually classical results for the contact discontinuity and the three oblique shocks of each Mach reflection. For the contact discontinuity, the *slipstream matching condition* from the triple shock theory states [21] in the frame of reference of the triple point:

$$\begin{cases} p_i = p_j \\ \theta_j = \theta_k - \theta_i \end{cases} \tag{7.2.2}$$

referring to the notations of figure 7.5, with θ the deflection angle. For the Mach reflection at T, we have $i = 3$, $j = 4$ and $k = 2$ and at triple point T', $i = 6$, $j = 5$ and $k = 3$. The second equation of condition (7.2.2) ensures that the streamlines deflected by the Mach stem and on the other hand by the incident shock and the reflected shock are parallel. As for the three shocks of each Mach reflection, classical oblique shock theory [92] gives the deflection angle θ_j and the thermodynamic state $(p_j, \rho_j, \hat{M}_j)$ behind the shock wave in the frame of reference of the triple point:

$$p_j = p_i \left(1 + \frac{2\gamma}{\gamma+1}(\hat{M}_i^2 \sin^2\phi_j - 1)\right) \tag{7.2.3}$$

$$\rho_j = \rho_i \left(1 + \frac{2 + (\hat{M}_i^2 \sin^2\phi_j - 1)}{(\gamma-1)\hat{M}_i^2 \sin^2\phi_j + 2}\right) \tag{7.2.4}$$

$$\tan\theta_j = \left[\tan\phi_j \left(\frac{(\gamma+1)\hat{M}_i^2}{2(\hat{M}_i^2 \sin^2\phi_j - 1) - 1} - 1\right)\right]^{-1} \tag{7.2.5}$$

$$\hat{M}_j = \left(\frac{2 + (\gamma-1)\hat{M}_i^2}{2\gamma\hat{M}_i^2 \sin^2\phi_j - (\gamma-1)} + \frac{2\hat{M}_i^2 \cos^2\phi_j}{2 + (\gamma-1)\hat{M}_i^2 \sin^2\phi_j}\right)^{1/2} \tag{7.2.6}$$

referring to the notations of figure 7.5, with ϕ_j the incident angle, θ_j the deflection angle and $\hat{M}_i$ the Mach number in state (i) with respect to the triple point. The values of (i, j) for each shock are given in the tables of figure 7.5.

7.2.3 Problem setting

Physical problem

The DMR configuration to be solved is defined in [130]. We recall here the geometry, the physical problem and the implementation of the boundary conditions as given by the authors. The computational domain $\Omega \times [0, T]$ with $\Omega = [0, 4] \times [0, 1]$ and $T = 0.25$ (see figure 7.6). The angle of the corner is $\theta = \pi/6$. In order to avoid the complications of the wedge geometry, the wedge wall is taken horizontal and the incident shock is turned by an angle θ. The half-line along the x-axis of figure 7.6 starting at x_0 is the x'-axis of the wall in figure 7.4.

A shock with angle $\alpha = \pi/2 - \theta$ and Mach number $M_S = 10$ moves through still air ($\gamma = 1.4$) with density $\rho_1 = 1.4$ and pressure $p_1 = 1$ hence a shock speed of $|\mathbf{u}_S| = 10$. The post-shock state is determined by the Rankine-Hugoniot jump conditions. The shock reaches the apex

$A = (x_0, 0)$ of the wedge at $t = 0$. The initial conditions are therefore defined as:

$$\begin{array}{ll} \textit{Pre-shock state} & \textit{Post-shock state} \\ x > x_0 + y\tan\theta & x < x_0 + y\tan\theta \\ u_{x1} = 0 & u_{x2} = 8.25\cos(\theta) \\ u_{y_1} = 0 & u_{y_2} = -8.25\sin(\theta) \\ \rho_1 \ = 1.4 & \rho_2 \ = 8 \\ p_1 \ = 1 & p_2 \ = 116.5 \end{array} \tag{7.2.7}$$

The left boundary has inflow boundary conditions. The lower boundary for $x < x_0$ and the right boundary have outflow conditions. Regarding the upper boundary condition, an artificial boundary condition is set to follow the shock as it moves to the right of the domain. The intersection point between the shock and the upper boundary moves at speed $|\mathbf{u}_S|/\cos(\theta)$ and is located at $x_S(t) = x_0 + \tan(\theta) + |\mathbf{u}_S|/\cos(\theta)\, t$. This results in the following set of boundary conditions:

$$\begin{array}{l|l} x = 0 \text{ and } y \in [0,1] & x \in [0, x_0] \text{ and } y = 0 \\ \quad u_x = 8.25\cos(\theta) & \quad p = 116.5 \\ \quad u_y = -8.25\sin(\theta) & \\ \quad \rho \ = 1.4 & x \in [0, x_S(t)] \text{ and } y = 1 \\ & \quad u_x = 8.25\cos(\theta) \\ x = 4 \text{ and } y \in [0,1] & \quad u_y = -8.25\sin(\theta) \\ \quad p = 1.4 & \quad \rho \ = 1.4 \\ & \quad p \ = 116.5 \\ x \in [x_0, 4] \text{ and } y = 0 & \\ \quad \nabla_n(u_x) = 0 & x \in [x_S(t), 4] \text{ and } y = 1 \\ \quad\quad u_y = 0 & \quad u_x = 0 \\ & \quad u_y = 0 \end{array} \tag{7.2.8}$$

Numerical method

Our objective is to have an increased spatial resolution for shock waves, contacts waves and for the jet as well. The first AMR test — presented in section 7.2.4 — features two levels of refinement both determined with criteria (7.2.9). The refinement factor is $n_{\text{ref}} = 2$ and the finest level step size is $h_2 = 1/200$. Instead of focusing of the whole domain we may rather assign all the computational effort to increase the resolution at the two Mach reflections. In the second AMR test featured in section 7.2.5 two levels of refinement are used but the first level is defined by the arbitrary window of refinement (7.2.10) encompassing the two triple points. The second level of refinement is generated using criteria (7.2.9). The refinement factor is $n_{\text{ref}} = 4$ and the finest level step size is $h_2 = 1/1600$.

The first refinement criteria is based on the variations of the density. A cell K belonging to the level grid G_ℓ is flagged for refinement if the following condition is satisfied:

$$\max_{L \in \mathcal{N}(K)} |\rho_K - \rho_L| > \varepsilon \cdot h_\ell \tag{7.2.9}$$

with $\varepsilon = 1$ in the test of section 7.2.4 and $\varepsilon = 50$ in the test of section 7.2.5. The second refinement criterion defines an arbitrary window of refinement focused on the two Mach reflections.

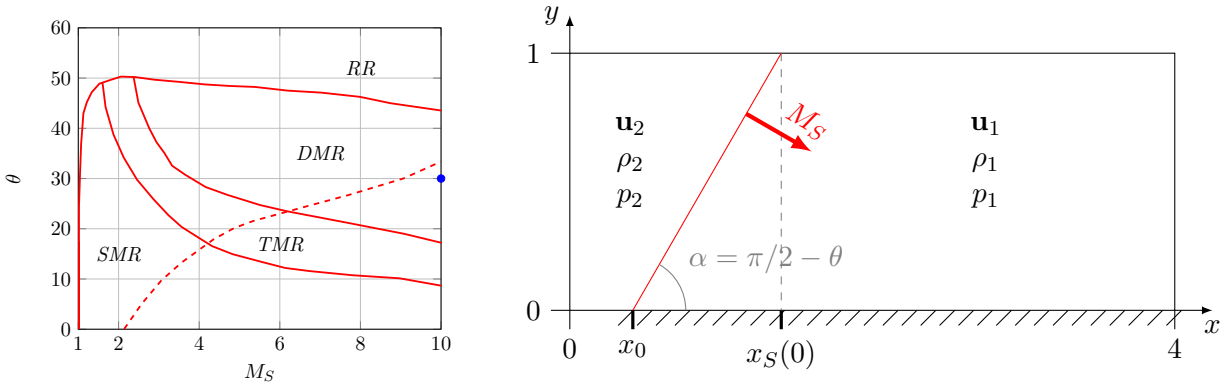


FIG. 7.6 – *Left: shock diffraction domains for a diatomic gas, reproduced from [22]. Solid curves are the boundaries of different shock diffraction domains. Below the dashed curve the bow shock is attached, above it detached. The blue mark matches the parameters of the benchmark. Right: domain for the DMR problem to be solved under such parameters.*

This window is moving with triple point T. The latter speed is $|\mathbf{u}_T| = |\mathbf{u}_S|/\cos(\theta + \chi)$ and its space coordinates are:

$$\begin{aligned} x_T(t) &= x_0 + |\mathbf{u}_T|\cos(\chi)t \\ y_T(t) &= |\mathbf{u}_T|\sin(\chi)t \end{aligned}$$

An empirical determination of χ yields $x_T(t) = x_0 + 12.8t$ and $y_T(t) = 2t$. A cell K belonging to the level grid G_ℓ is inside the window of refinement at time t if the subsequent condition are verified:

$$\begin{cases} x_K > x_T(t) - \max(a, (x_T(t) - x_0)/4) \\ y_K > 0 \\ x_K < x_T(t) + a \\ y_K < y_T(t) + a \end{cases} \tag{7.2.10}$$

with $a = 0.15$. In the original work [25], a window of refinement was also used for the first level but it constantly covered the height of the domain so that the incident shock would not cross any coarse-fine interface in contrast to our refinement strategy. The impact of this choice will be discussed in section 7.2.5.

The adaptive grid generation algorithm presented in chapter 5 is used with the "standard Laplacian" for edge detection in the clustering algorithm. In the test of section 7.2.4 the efficiency is set to $emin = 1$. In the test of section 7.2.5 the efficiency is set to $emin = 0.5$. It was chosen not to add any restriction on the shape of the patches, i.e. $rvmin = vmin = lmin = 0$.

Once the composite grid has been generated, the compressible Euler equations are solved with the Multigrid-AMR method presented in Chapter 6. The stopping criterion for the multigrid solver is 10^{-10} on the absolute L^∞ norm of the composite residual. In each multigrid resolution, the smoothing on each level consists in three outer iterations over all the patches, with for each patch two smoothing steps (Gauss-Seidel) with synchronization of ghost-cell values. The iterations of the fixed point procedure for the non-linear projection-correction step end when the absolute L^∞ norm of pressure increment with respect to the fixed point iterations is below 10^{-12}. Finally the outer iterations in time have a timestep tied to the finest grid step size as $\delta t = h_2/40 = 1.25 \times 10^{-4}$ in the test of section 7.2.4 and as $\delta t = h_2/20 = 3.125 \times 10^{-5}$ in the test of section 7.2.5.

7.2.4 Adaptive and uniform grid solutions

The aim of this first test is to compare two numerical simulations of the DMR problem, one with adaptive mesh refinement — as described in the foregoing section — and one with an uniform grid using the space step $h_2 = 1/200$ of the finest AMR level. The uniform grid solution will stand as a reference to assess the accuracy of the adaptive solution.

Self-similar solution

The evolution of the solution in the self-similar frame of reference is given in figure 7.7. In both solutions the isopycnics allow to identify clearly the incident shock (i), the reflected shock (r) connected to the attached bow shock (m') through a curvature reversal and the Mach stem (m). Triple points can be therefore located accurately and they appear to be perfectly aligned in the self-similar frame at different timesteps (see green lines in figure 7.7). This confirms the self-similarity property of the numerical solution with both adaptive and uniform grids.

Regarding the flow inside the diffraction pattern, the base grid in the adaptive solution (with space step $h_0 = 1/50$) is unfortunately too coarse for our first order pressure-correction scheme: the variations of the density are too loose to trigger the aforementioned refinement criterion. Thus only the uniform grid solution features the reflected shock (r') and the slipstream (s).

The comparable sharpness of shock waves (i), (r), (m') and (m) in both adaptive and uniform grid solutions is misleading. Indeed the adaptive solution features a small though not negligible error in the computation of these shocks. In the uniform grid solution, triple points are located at $T = (12.75, 2.04)$ and $T' = (10.37, 1.98)$ while in the adaptive grid solution we have $T = (12.69, 1.94)$ and $T' = (10.53, 1.94)$. We believe that this is difference is caused by the difference of resolution in region $(0) \cup (3) \cup (4) \cup (5) \cup (6)$, i.e. $h_2 = 1/200$ for the uniform grid solution and $h_0 = 1/50$ for the adaptive grid solution.

Local refinement

The evolution of the number of unknowns for the adaptive grid calculation is compared to the number of unknowns for the uniform grid case. The ratio between the two is given in the last column:

Time	*Level 0*	*Level 1*	*Level 2*	*Composite*	*Total*	*Ratio*
0	10000	1028	2408	1.3×10^4	1.3×10^4	0.08
$250\delta t$	10000	1632	5072	1.5×10^4	1.7×10^4	0.10
$500\delta t$	10000	2160	6776	1.7×10^4	1.9×10^4	0.12
$750\delta t$	10000	2684	8344	1.8×10^4	2.1×10^4	0.13
$1000\delta t$	10000	3104	9704	2.0×10^4	2.3×10^4	0.14
$1250\delta t$	10000	3596	11240	2.1×10^4	2.5×10^4	0.16
$1500\delta t$	10000	4120	12896	2.3×10^4	2.7×10^4	0.17
$1750\delta t$	10000	4620	14456	2.4×10^4	2.9×10^4	0.18
$2000\delta t$	10000	5160	16184	2.6×10^4	3.1×10^4	0.20

TAB. 7.1 – *Evolution of the number of unknowns in adaptive and uniform grid calculations.*

At $t = 2000\delta t = 0.25$, the number of unknowns with adaptive grids is five times smaller than the number of unknowns with the uniform grid. The number of unknowns of finest level being significantly larger than that of the first level of refinement, the composite grid size does

not differ much from the total number of unknowns. The evolution of the number of patches is given hereafter:

Time	*Level 0*	*Level 1*	*Level 2*
0	1	66	66
$250\delta t$	1	61	95
$500\delta t$	1	68	105
$750\delta t$	1	86	121
$1000\delta t$	1	75	122
$1250\delta t$	1	93	128
$1500\delta t$	1	103	145
$1750\delta t$	1	117	157
$2000\delta t$	1	125	161

TAB. 7.2 – *Evolution of the number of patches in adaptive grid calculation.*

The refinement being concentrated on shocks, the clustering algorithm with efficiency set to $emin = 1$ generates a large number of patches, which increases the computational cost of the adaptive grid calculation.

Conclusion

This first numerical test emphasizes the need to start with a minimal resolution at the coarsest level so that essential flow features can be detected by the refinement procedure. Moreover, the refinement factor should be large enough to yield a sufficient increase of resolution between two levels, which is an issue given the amount of diffusion generated by our first order pressure correction scheme. In consequence in the next test the base grid will have a step size $h_0 = 1/200$ and the refinement factor will be $n_{\text{ref}} = 4$.

7.2.5 Local refinement on the Mach reflections

Non-steady solution

Figure 7.9 gives an overview of the solution at different timesteps up to $t = 0.25$. The first triple shock system and the curvature reversal are clearly seen on the isopycnics, thereby identifying the two triple points. The green lines start at points T and T' of the solution at $t = 8000\delta t$ and are absolutely aligned with the triple points of the solution back to $t = 4000\delta t$. At earlier timesteps, a small discrepancy is observed. Evidence of this error is more easily seen by locating the intersection of the two green lines, which should cross at point A at $t = 0$. While for $t > 4000\delta t$ the solution seems perfectly self-similar it is expected to feature a slight error introduced at earlier timesteps. This should have an impact for instance on the position of point I.

Despite this small source of error the numerical solution we will be considered self-similar hereafter. The numerical results will be analyzed using the solution at $t = 8000\delta t = 0.25$ in the self similar frame of reference.

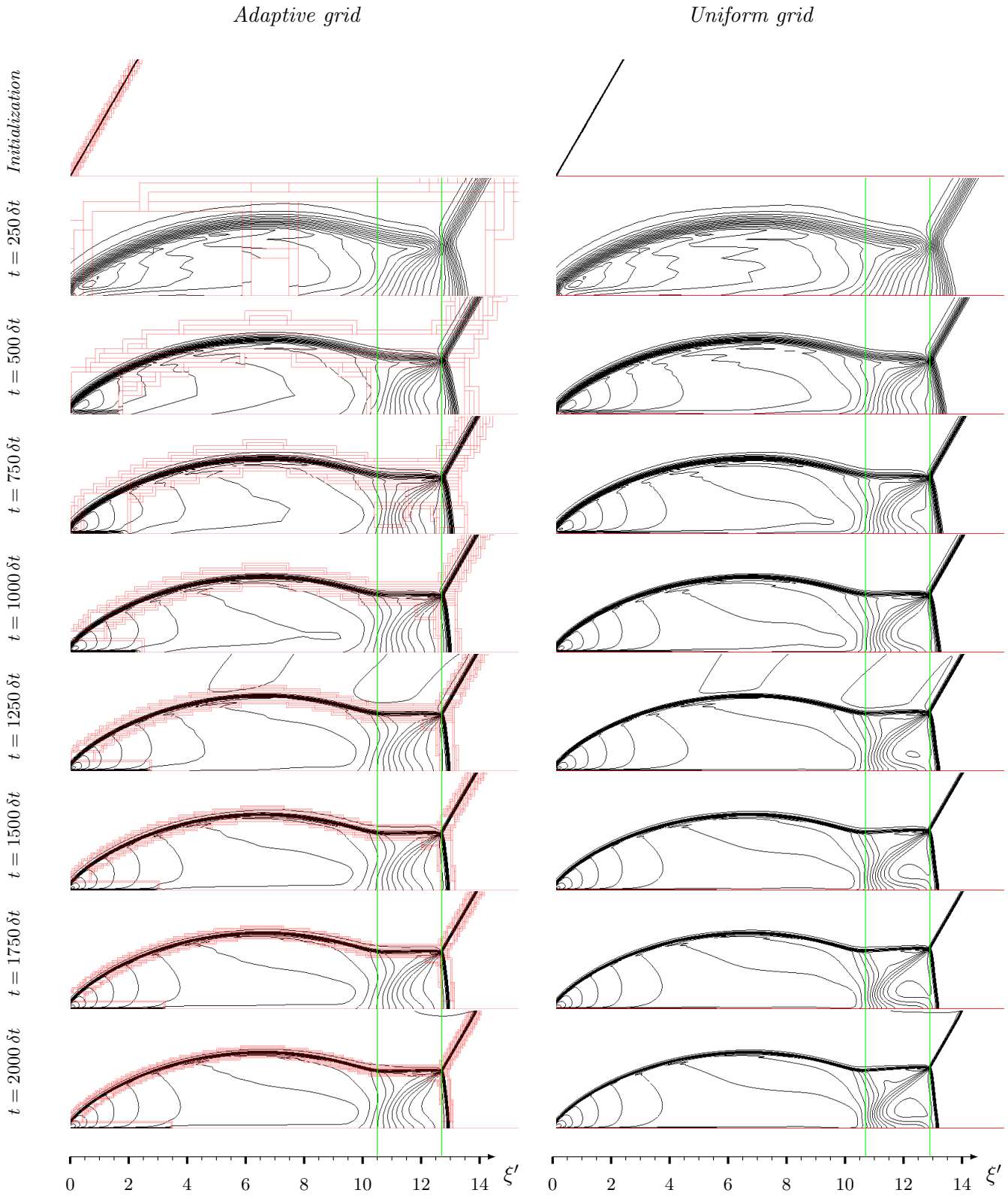


FIG. 7.7 – *Evolution of the solution in the self-similar frame of reference between* $t = 3.125 \times 10^{-2}$ *and* $t = 0.25$. *For the record the timestep is* $\delta t = 1.25 \times 10^{-4}$. *Left: isopycnics (30 values, in black) of the solution with AMR (finest level:* $h_2 = 1/200$*) ; AMR patches for the two levels of refinement are in red. Right: isopycnics (30 values) of the solution with uniform grid (*$h = 1/200$*). Note that the horizontal axis starts at* x_0. *The green lines follow the position of the two triple points.*

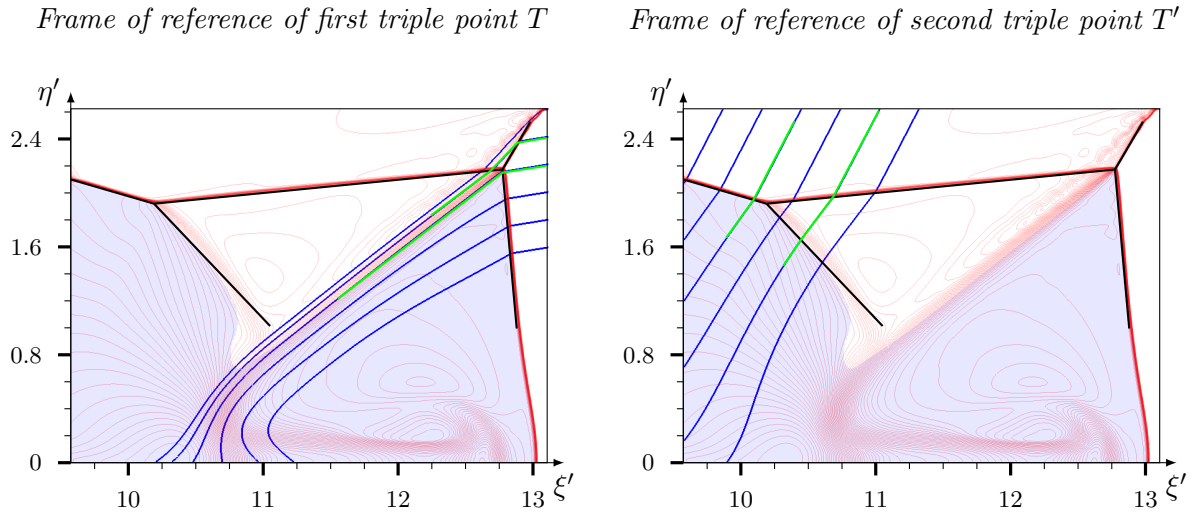


FIG. 7.8 – *Isopycnics (in red, 100 contours, $\rho \in [1.4, 17.6]$), pseudo-subsonic region (in light blue) and velocity streamlines (in blue) respectively in the frame of reference of triple points T (left) and T' (right). The line segments approximating shocks (in black) and streamlines (in green) are used for measuring oblique shock deflection angles.*

Self-similar solution

The isopycnics and the isobars of the solution (figure 7.10) identify very clearly the discontinuities of the density and of the pressure. Regarding the first triple shock system, the incident shock wave (i), the reflected shock wave (r) and the Mach stem (m) are very sharp whereas the contact discontinuity (s) features a lot of diffusion. The Mach stem hits the wall perpendicularly at point I of coordinates $(13.05, 0)$. The triple point T is found at coordinates $(12.80, 2.19)$. It makes an angle $\chi = 9.24°$ with the ξ'-axis, hence an expected speed $|\mathbf{u}_T| = 12.91$ which is consistent with the position of T in the self-similar frame.

As for the second triple shock system, the incident shock wave (r) and the second Mach stem (m') are also very sharp. However we observe a lot of diffusion at the reflected shock (r'). The latter features a small deviation at it reaches the first slipstream (s), whereas it should have a continuous curvature. This issue will be addressed later on. The second slipstream (s') is not noticeable at all. It is known to be very difficult to observe experimentally [22] and to compute accurately [25] due to the very slight variations of density across it. Nevertheless some authors have succeeded to observe it using local mesh refinement [75]. The curvature reversal is apparent and allows to identify the second triple point T' at $(10.22, 1.94)$. The latter trajectory angle is found to $\chi' = 10.12°$, greater than χ and thereby confirming that we have a DMR"+" shock diffraction. The expected speed of T' with the Law-Glass assumption is $|\mathbf{u}_{T'}| = 10.79$ which is coherent with the measured position of T' in the self-similar frame.

The pseudo-velocity streamlines on figure 7.10 provide a good understanding of the flow in the different regions of the solution. In pseudo-supersonic regions (i.e. with respect to the pseudo-Mach $\bar{M}$) (2), (1) and (3) the density and the pressure are constant and the streamlines are straight lines converging to three different sink points. The streamlines passing near triple point T either through shocks (i), (r) or through (m) are parallel to the first slipstream. They reach the wall at stagnation point $P_2 = (10.32, 0)$. At P_2, part of the flow (actually from state (3)

is directed towards sink point $P_1 = (6.09, 0)$ in region $(0) \cup (5) \cup (6)$. We believe the second slipstream to be parallel to the streamlines leaving second triple point T' towards P_1. Along the wall between A and P_2 the density has almost reached its minimal value at P_1. Its maximum is located at point A, which is in line with the experimental observations of G. Ben-Dor and I.I. Glass in [22] for an attached bow shock.

A closer view of the two Mach reflections is provided in figure 7.12. At stagnation point P_2 the streamlines belonging to state (4) form a jet along the wall in state (4) as observed by P. Woodward and P. Colella in [130] for this very DMR configuration. The jet rolls up behind point $P_3 = (12.75, 0)$ to form a vortex, which is correlated to the slight peak in the density profile along the wall between P_2 and I. This does not impact the general trend of the density along this portion of the wall which increases from I to P_2 as noticed experimentally in [22]. This indicates a compression of the flow along the wall. The jet has a strong influence on the Mach stem, moving its foot forward. As a result the Mach stem features an inflexion point as pointed out in [75]. The foot of the Mach stem is clearly perpendicular to the wall, which is in accordance with the description of the Mach reflection given in the introductory section.

We use after [130] the iso-values of $A = p/\rho^\gamma$ (figure 7.10) to identify the numerical noise to the solution. Three regions of disturbance stand out. First at the intersection between the incident shock and the window of refinement, a severe disturbance is generated and propagates along the incident shock towards the reflected shock and state (3). This issue was pointed out by the authors of [25], who found the amplitude of the disturbances to be proportional to the strength of the incident shock. Let us notice that in the previous test, where no shock was crossing any coarse-fine interface, no such numerical noise was observed. Another issue, listed in [130], is caused by the initial shock width being spread over a few cell in contrast to the boundary condition at the upper wall which is "exact". Finally an artificial boundary layer spreading along the wall from the apex A adds further numerical error. The latter two disturbance were found to be impactless while the first one has a strong influence on the boundaries pseudo-supersonic region (3).

Deflection vs reflection

As explained in the introductory section, shock diffraction results from the interaction between flow deflection and shock reflection through the propagation of pressure disturbances. Using the geometric construction presented in the foregoing part of this chapter, we will try to estimate the origin of the pressure disturbance and their leading front. On figure 7.11, two half-lines starting at triple point T extend the reflected shock (r) and the first slipstream (s). The constant velocity vector $\mathbf{u}_2$ is represented using its theoretical value hence a length of $|\mathbf{u}_2| = 8.25$ in the self-similar frame of reference. The line perpendicular to the extension of shock wave (r) passing through the tip of $\mathbf{u}_2$ intersects the extension of the slipstream at point A_3. Vector $\overrightarrow{AA_3}$ matches the velocity vector $\mathbf{u}_3$ of state (3). The norm of the velocity vector $\mathbf{u}_3$ obtained with the geometric construction is $|\mathbf{u}_3| = 7.3$, which is in line with the measured value $|\mathbf{u}_3| = 7.4$ in our numerical results. Pressure disturbances in the self-similar frame of reference originate at point A_3 (which is stationary) and move towards triple point T with the speed of sound $a_3 = \sqrt{\gamma p_3/\rho_3} = 4.95$. The resulting arc (in green) indicates the farthest theoretical reach of the pressure signals leaving from A_3. Remarkably the arc matches very closely pseudo-sonic line next to the slipstream. This somehow confirms that the window which arbitrarily separates the flow defection from the shock reflection does not have a significant impact on the propagation of pressure disturbances from A_3.

Triple shocks systems

We now assess the deflection angles of the two triple shock systems. The streamlines in the frame of reference of T and T' are shown on figure 7.8. Referring to the notations of figure 7.5, classical oblique shock theory yields the following predictions:

FoR	(i,j)	ϕ_j	M_i	p_i	ρ_i	θ_j^*	θ_j	$\hat{M}_j^*$	$\hat{M}_j$	p_j^*	p_j	ρ_j^*	ρ_j
T	$(1,2)$	50.0	12.97	1	1.4	38.2	36.6	1.90	1.82	115	117	8.0	8.0
T	$(2,3)$	40.7	1.82	117	8.0	7.6	9.4	1.56	1.50	172	192	10.6	11.3
T	$(1,4)$	86.3	12.97	1	1.4	16.8	28.6	0.41	0.45	195	196	8.2	8.2
T'	$(2,3)$	56.5	1.44	117	8.0	8.1	9.0	1.13	1.08	176	192	10.7	11.3
T'	$(3,5)$	80.4	1.08	192	11.3	1.0	2.6	0.96	0.83	222	279	12.5	14.7
T'	$(2,6)$	111.5	1.44	117	8.0	10.5	8.3	0.90	0.70	225	284	12.7	14.8

TAB. 7.3 – *Predicted and measured deflection angles and states at oblique shocks.*

The two first columns indicate the frame of reference (FoR) and the oblique shock ((i,j) indices, see figure 7.5). The four next columns give the data used to predict the deflection angles and the state before the shock using equations (7.2.3). The next columns feature the predicted value using oblique shock theory (with a star superscript) and the value obtained with our numerical simulation.

For the first triple shock system, the predicted deflection angles are in quite good agreement with measured angles except for the Mach stem ($i = 1$ and $j = 4$). However unlike the predicted values, the deflection angles obtained from the numerical simulation do verify the *slipstream matching conditions*:

$$\theta_3 - \theta_2 = 27.2$$
$$\theta_4 = 28.6$$

which is consistent with the two selected streamlines being parallel in figure 7.8. The error in the prediction of the deflection angle $\theta_4^\star$ is probably to blame to the assumption of a straight shock whereas (m) is curved. The Mach number, the pressure and the density predicted are in rather good accordance with the numerical values.

As for the second triple shock system, the predictions are more difficult to carry out because of the important variations of density and of pressure in states (5) and (6). The predicted values and the numerical values of the deflection angle and of the state before the shocks coarsely match.

Local refinement

The numerical solution obtained in the window of refinement is represented on figure 7.13 (left), scaled to the self-similar frame of reference. The green lines follow the two triple points and confirm the self-similarity of the numerical solution.

The refinement criterion is seen to match very accurately the regions of high density gradient thanks to the fine space step of the first level of refinement ($h_1 = 1/800$) and to the refinement factor $n_{\text{ref}} = 4$. The second level of refinement covers the jet with its vortex and shock waves (i), (r), (m'), (m), (r') and (s). However the latter two waves feature a high amount of diffusion, which results in larger patches and a higher computational cost.

A more serious concern is the propagation of the numerical disturbance along the incident shock (figure 7.13, right). This noise originates to the intersection of the incident shock with the coarse-fine interface of the window of refinement. Its influence is considerable at earlier timesteps. At $t = 2000\delta t$ the disturbance crosses reflected shock (r) into state (3) up to stationary point P_2. Clearly this moves the sonic line separating state (3) and state (5) to the wall. Hopefully the amplitude of this numerical noise appears to decrease at later timesteps. The sonic line recovers its shape though at $t = 8000\delta t$ it still features a slight deviations towards P_2.

The evolution of the number of unknowns as the diffraction pattern grows is given in the table below. Compared to a numerical simulation with an uniform grid of space step equal to h_2, i.e. with 1.024×10^7 unknowns, the gains in computational effort with AMR are evident. Thanks to the refinement factor $n_{\text{ref}} = 4$, the size of the problem at the final timestep is as little as 5% of that of the uniform grid problem.

Time	*Level 0*	*Level 1*	*Level 2*	*Composite*	*Total*	*Ratio*
0	40000	8192	7424	5.5×10^4	5.6×10^4	0.005
$1000\delta t$	40000	11264	65040	1.1×10^5	1.2×10^5	0.011
$2000\delta t$	40000	16576	107664	1.6×10^5	1.6×10^5	0.016
$3000\delta t$	40000	26320	166368	2.2×10^5	2.3×10^5	0.023
$4000\delta t$	40000	37392	181616	2.5×10^5	2.6×10^5	0.025
$5000\delta t$	40000	50384	226208	3.0×10^5	3.2×10^5	0.031
$6000\delta t$	40000	65296	262592	3.5×10^5	3.7×10^5	0.036
$7000\delta t$	40000	83520	301760	4.0×10^5	4.3×10^5	0.042
$8000\delta t$	40000	102432	351504	4.7×10^5	4.9×10^5	0.048

TAB. 7.4 – *Evolution of the number of unknowns compare to the case of uniform grid.*

In our numerical experiments, the inter patch operations with our current implementation were found to be quite expensive and to impede the scalability with respect to the number of patches (hence a soaring computational cost as the self-similar solution grows). This motivated the choice of a low efficiency in the clustering algorithm with $emin = 0.5$. As a result the number of patches increases at a much lower pace and stays below 100.

It should be emphasized that use of the procedure for fixing incompatible clusterings presented in chapter 5 was found to be compulsory. During the present calculation, as many as 30539 new patches were generated by this procedure over the 8000 timesteps.

Time	*Level 0*	*Level 1*	*Level 2*
0	1	1	23
$1000\delta t$	1	1	9
$2000\delta t$	1	1	13
$3000\delta t$	1	1	20
$4000\delta t$	1	1	45
$5000\delta t$	1	1	50
$6000\delta t$	1	1	57
$7000\delta t$	1	1	69
$8000\delta t$	1	1	90

TAB. 7.5 – *Evolution of the number of patches.*

Finally the scalability of the Multigrid-AMR solver proved to be essential. The average number of iterations was always between 3 and 5.

Conclusion

These numerical results are very insightful, as they demonstrate the capability to accurately compute the complex shock systems of shock diffraction with a first order all-Mach pressure correction scheme using a Multigrid-AMR method. In contrast the very first AMR simulation of this problem was performed with an explicit AMR method based on a Godunov-type solver [25].

Ongoing work include a second order extension in a bid to recover the same accuracy as the results from [25]. More broadly the numerical accuracy was found to play a critical role in the ability of the AMR method to produce physically relevant results. An increased resolution in some region of the flow is pointless if essential flow features are missed because of the refinement criterion.

Finally, we should insist on the fact that the arbitrary window of refinement did not seem to disturb the interaction between flow deflection and the shock reflection, which is quite remarkable.

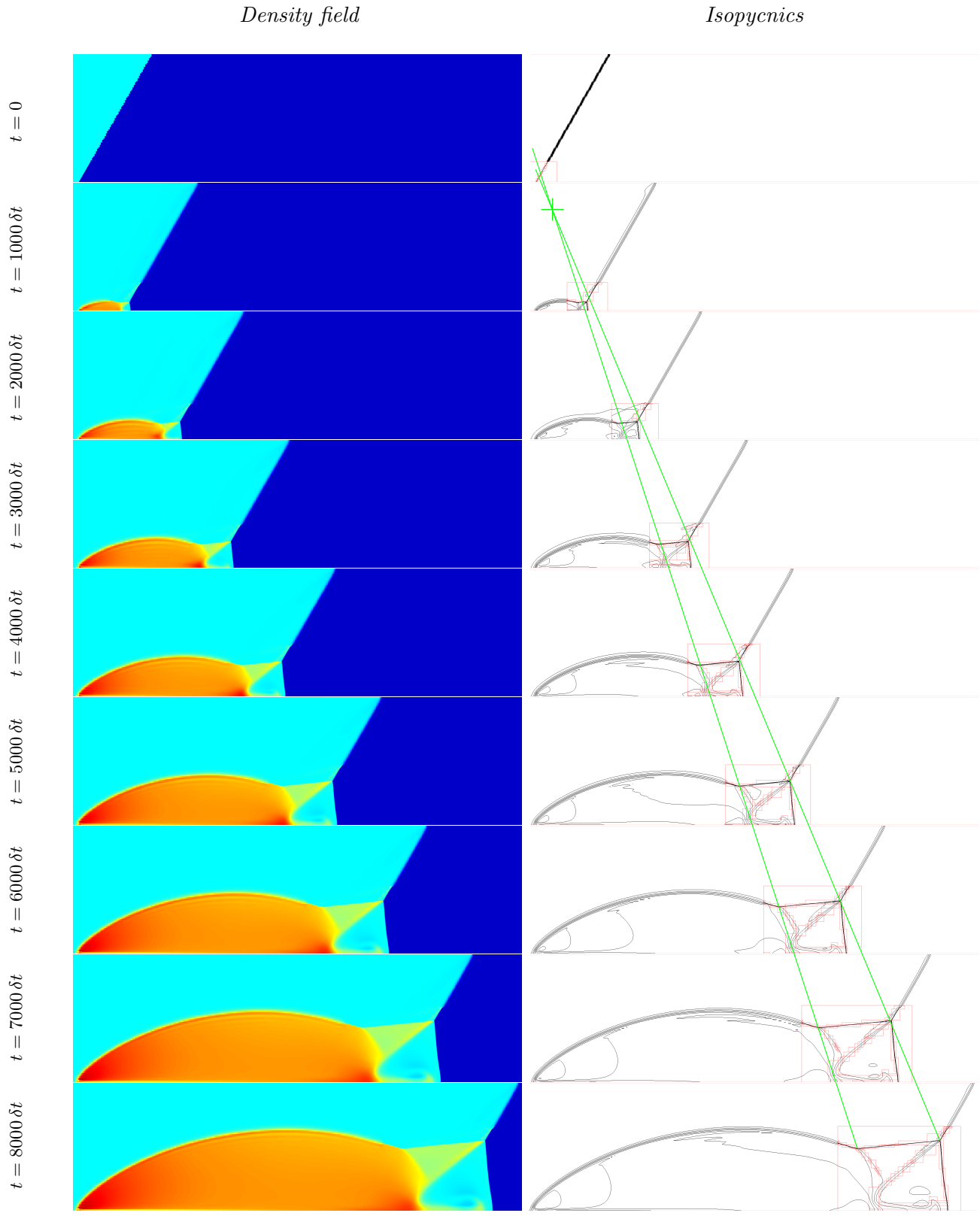


FIG. 7.9 – *Growth of the self-similar solution between $t = 0$ and $t = 0.25$. For the record the timestep is $\delta t = 3.125 \times 10^{-5}$. Left column: density field. Right column: isopycnics (in black, 12 contours) with patches (in red) for the two levels of refinement. Note that the horizontal axis has been trimmed down. The green lines follow the trajectory of the two triple points and they intersect at the green cross.*

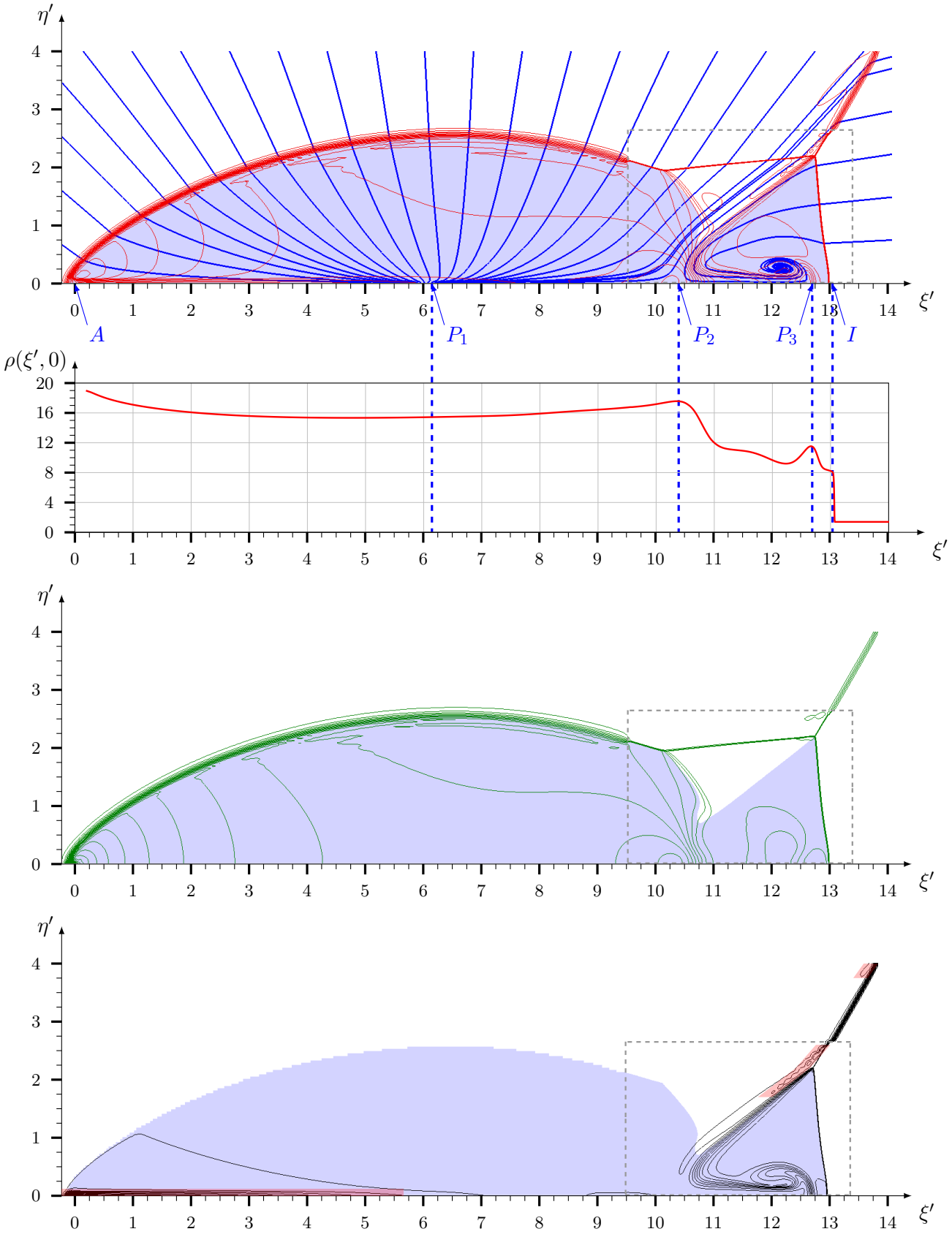


FIG. 7.10 – *Top: isopycnics (in red, 30 contours, $\rho \in [1.4, 21.3]$), pseudo-subsonic region (in light blue) and pseudo-velocity streamlines (in blue). The dashed gray line delimits the first level of refinement. Middle top: density profile along the wedge wall. Middle bottom: isobars (in green, 30 contours, $p \in [1, 624]$). Bottom: contours of $A = p/\rho^\gamma$ (in black) with problematic regions overlayed in red.*

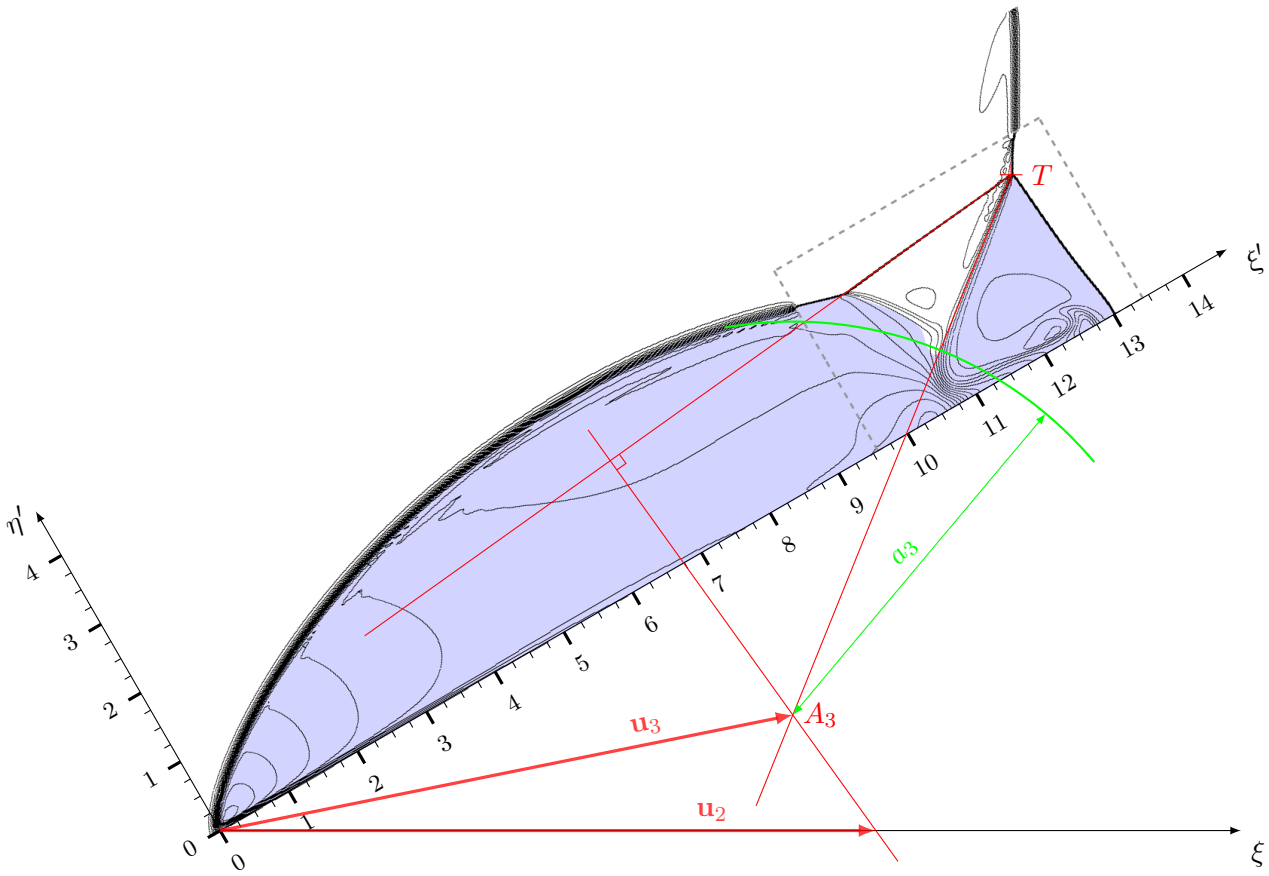


FIG. 7.11 – *Construction (in red) of the sonic line (in green) w.r.t. state* (3)*, which gives an insight on the location of the reflected shock* (r') *of the second Mach reflection. The pseudo-subsonic region is in light blue and the isopycnics are in black.*

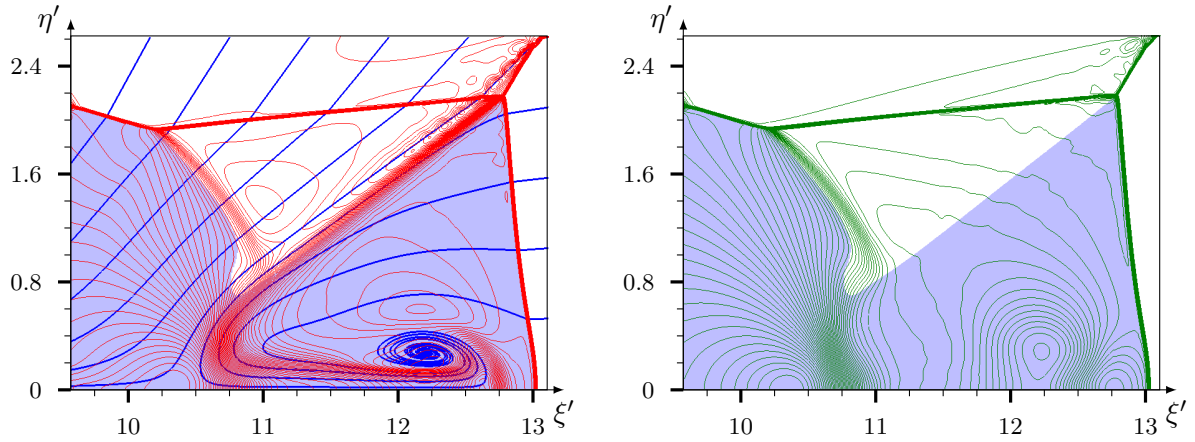


FIG. 7.12 – *Close up view of the window of refinement. Left: isopycnics (in red, 100 contours,* $\rho \in [1.4, 17.6]$*), pseudo-subsonic region (in light blue) and pseudo-velocity streamlines (in blue). Right: isobars (in green, 100 contours,* $p \in [1, 351]$*).*

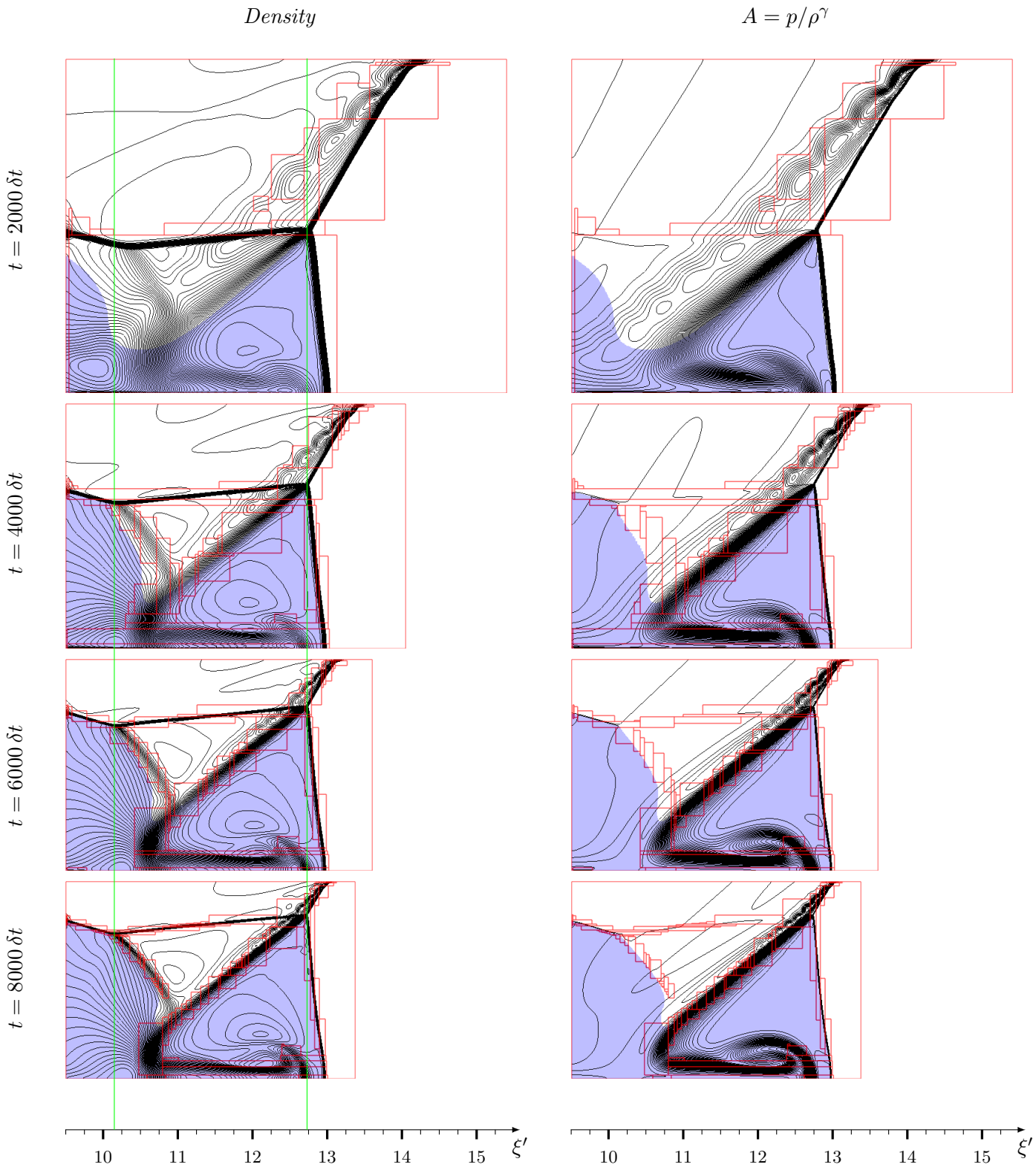


FIG. 7.13 – *Comparison of the Mach reflections in the window of refinement at different times, represented here in the self-similar frame of reference. The first column features the isopycnics (100 contours, in black) and the right column the iso-values of $A = p/\rho^\gamma$ (100 contours, in black). The pseudo-subsonic region is in light blue and the patches of the two levels of refinement are in red. The green lines follow the two triple points.*

Part III

Fluid-porous interface problem

Chapter 8

MODELLING AT DIFFERENT SCALES

In this chapter, we review the porous modelling in the GENEPI code at CEA. Then we recall the classical approaches for modelling cross filtration at the mesoscopic scale and at the macroscopic scale in the linear viscous regime.

8.1 Porous model in GENEPI

8.1.1 Governing equations

We recall here the two-phase flow model of GENEPI first presented in Chapter 4. For each phase K of the secondary loop fluid (either liquid or vapour) the local conserved equations read [104]:

$$\partial_t(\rho_K) + \operatorname{div}(\rho_K \mathbf{u}_K) = 0 \tag{8.1.1}$$

$$\partial_t(\rho_K \mathbf{u}_K) + \operatorname{div}(\rho_K \mathbf{u}_K \otimes \mathbf{u}_K) = -\operatorname{div}(\tau(\mathbf{u}_K)) + \rho_K F_K \tag{8.1.2}$$

$$\begin{aligned}\partial_t \left(e_K + \frac{|\mathbf{u}_K|^2}{2} \right) + \operatorname{div} \left(\rho_K \left(e_K + \frac{|\mathbf{u}_K|^2}{2} \right) \mathbf{u}_K \right) = &-\operatorname{div}(q_K) + \operatorname{div}(\tau(\mathbf{u}_K) \cdot \mathbf{u}_K) \\ &+\rho_K \mathbf{F}_K \cdot \mathbf{u}_K\end{aligned} \tag{8.1.3}$$

where $\tau(\mathbf{u})$ the viscous stress tensor and $\mathbf{F}_K$ an external force (eg. gravity). For the primary fluid the local enthalpy balance reads:

$$\partial_t(\rho_P h_P) + \operatorname{div}(\rho_P u_P h_P) = -\operatorname{div}(q_P) + \operatorname{div}(\tau(\mathbf{u}_P) \cdot \mathbf{u}_P) \tag{8.1.4}$$

with h_P denoting the enthalpy. The subscript P associates a variable with the primary loop fluid. This set of equations is closed by the equation of state of the secondary fluid. This local set of equations is then volume-averaged twice over a representative elementary volume (REV) (see figure 8.1). The first averaging, over the two phases, yields an homogeneous two-phase flow model. These equations are then averaged over the free-fluid regions and the obstacles within the REV, introducing the "artificial" porosity β. The final averaged equations (see [104] for a step-by-step derivation) give a balance of the system at the *mesoscopic scale*:

- Primary fluid, *energy balance*:

$$\rho_P C_P \partial_t T_P + \rho_P C_P \mathbf{u}_P \cdot \nabla T_P - \operatorname{div}(C_P \chi_T \nabla T_P) = -\frac{\gamma_0 h_{\text{eq}}}{\beta_{P_0}} (T_P - T_W)$$

- Secondary fluid, *mass balance*:

$$\beta\partial_t\rho + \operatorname{div}(\beta\rho\mathbf{u}) = 0$$

- Secondary fluid, *momentum balance*:

$$\beta\rho\partial_t\mathbf{u} + \beta\rho\mathbf{u}\cdot\nabla\,\mathbf{u} - \operatorname{div}(\beta 2\mu_T(\nabla\mathbf{u} + \nabla^t\mathbf{u})) + \beta\,\nabla\,p = \beta\rho\mathbf{g} - \beta\mathbf{\Lambda}\cdot\rho\mathbf{u} - \operatorname{div}(\beta X(1-X)\rho\mathbf{u}_R\otimes\mathbf{u}_R)$$

- Secondary fluid, *energy balance*:

$$\beta\rho\partial_t h + \beta\rho\mathbf{u}\cdot\nabla\,h - \operatorname{div}(\beta\chi_T\,\nabla\,h) = \tau\gamma_0 h_{\text{eq}}(T_P - T_W) - \operatorname{div}(\beta X(1-X)\rho\mathcal{L}\mathbf{u}_R) + Dp/Dt$$

- Secondary fluid, *equation of state*: tabulated values for water, fitted with high order polynomials.

Here are denoted by T the temperature, C the specific heat capacity, χ_T the turbulent thermal conductivity, h_{eq} the equivalent exchange coefficient, β_{P_0} the bundle primary porosity, γ_0 the heating surface density between the primary loop and the secondary loop, T_W the temperature at the wall of the U-tubes, μ_T the turbulent viscosity, $\mathbf{g}$ the gravitational constant, $\mathbf{\Lambda}$ a friction tensor and $\mathbf{u}_R$ a drift velocity. The latter is associated with the static quality X defined in chapter 4. In the above equations, ρ, T, $\mathbf{u}$, p and h are all mixture variables.

8.1.2 Discussion

The above mesoscopic porous model is referred to as the *continuous porosity* approach. The porosity and the flow variables present a continuous variation between the porous medium representing the evaporator and the free-fluid, as outlined on figure 8.1 (center). The relevance of this approach strongly depends on the space resolution of the numerical method in the region of transition between the porous medium and the free-fluid. In practice for industrial simulations, the accuracy in space in this region is not sufficient and an incoherent behaviour is observed at the fluid-porous interface in the upper part of the evaporator. The flow rate is higher than expected, which yields to the use of cumbersome workarounds.

Lately, an alternative modelling with a *discontinuous porosity* (figure 8.1, right) was considered. This model is defined at the macroscopic scale so proper transmission conditions must be established. The third part of this thesis aimed at contributing to the derivation of such conditions for realistic industrial flows, starting with the incompressible single-phase Navier-Stokes equations.

8.2 Modelling cross-flow filtration

In what follows, we consider a viscous incompressible pressure-driven flow over a porous bed. The flow is assumed to be *near parallel* [86]: the pressure gradient has the same order of magnitude in the plain fluid region and in the porous medium. The dynamic viscosity of the fluid is denoted by μ and the porous medium is characterized by its porosity $\phi \in [0,1]$ and its permeability tensor K. The physical domain extends over $\Omega = [0,L]\times[-H,H]$ as shown on

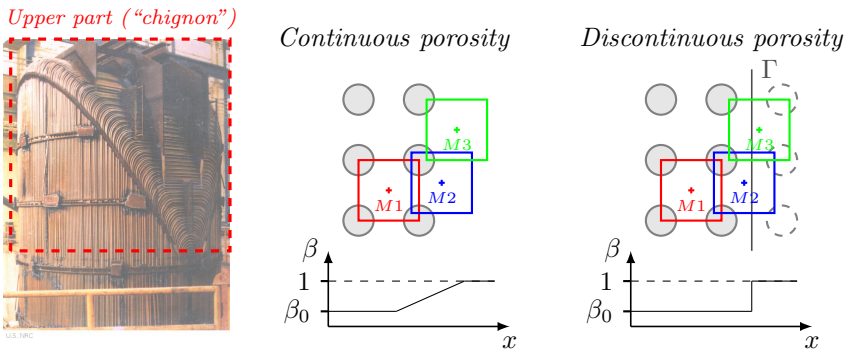


FIG. 8.1 – *Two modelling approaches for GENEPI: continuous porosity and discontinuous porosity.*

figure 8.3. Three subdomains are defined: the *"free-fluid region"* Ω_f, the *"transition region"* Ω_t and the *"plain porous region"* Ω_p. They are separated by two interfaces $\Gamma_f = \Omega_f \cap \Omega_t$ and $\Gamma_p = \Omega_p \cap \Omega_t$.

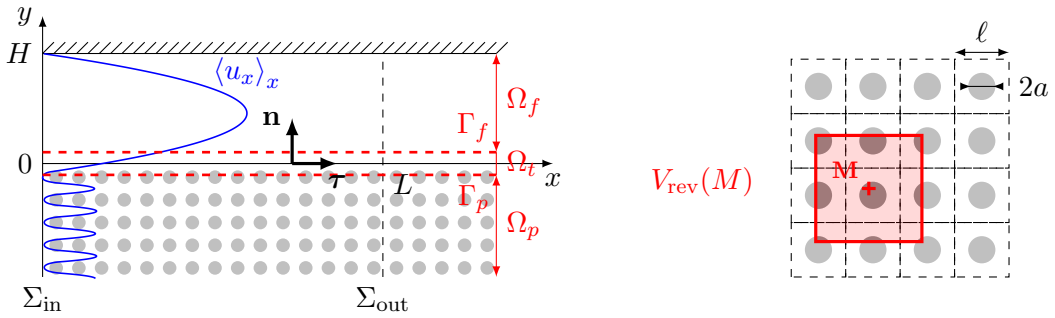
Here, we consider a porous medium formed by an array of solid inclusions, the pores being occupied by the fluid. The two subdomains Ω_f and Ω_p are homogeneous, in the sense that the former contains only plain fluid and the latter only plain porous medium. The free-fluid and the porous medium both overlap with the Ω_t region. The surface tangent to the inclusions of the porous medium is strictly above Γ_p and strictly below Γ_f.

For a variable ψ, the volume average over a representative elementary volume V_{rev} is defined as:

$$\langle \psi \rangle = \frac{1}{V_{\text{rev}}} \int_{V_{\text{rev}}} \psi(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

Likewise the average along x:

$$\langle \psi \rangle_x = \frac{1}{L} \int_0^L \psi(x,y)\,\mathrm{d}x$$



Poiseuille flow over a porous bed *Representative Elementary Volume*

FIG. 8.2 – *Fluid-porous problem under consideration.*

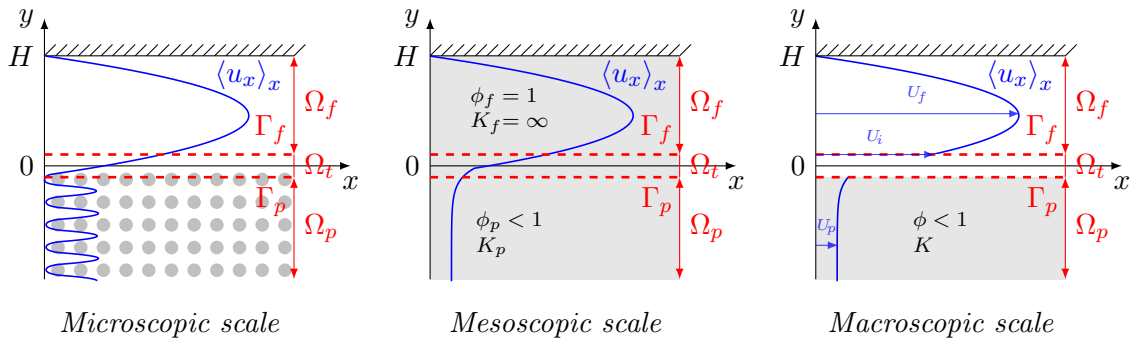


FIG. 8.3 – *Filtration with a near parallel flow, viewed from three different space scales.*

8.2.1 Filtration phenomena

The pressure gradient imposed between Σ_{in} and Σ_{out} generates in the free-fluid region Ω_f a Poiseuille flow with a zero velocity at the upper wall and a non-zero slip velocity on Γ_f caused by the filtration into the porous medium (figure 8.2). The flow in Ω_t and in Ω_p can be described from three different space scales [38]: microscopic, mesoscopic and macroscopic.

At the *microscopic scale* (figure 8.3, left), the flow is governed in the entire domain $\Omega = \Omega_f \cup \Omega_t \cup \Omega_p$ by the incompressible Navier-Stokes equations (or possibly the Stokes equations in the linear viscous regime) using the local variables (u, p). As opposed to the simple Poiseuille flow in Ω_f, the flow between the inclusions in Ω_t and Ω_p is complex and depends on the structure of the porous matrix.

At the *mesoscopic scale* (figure 8.3, center), the flow is described in the whole domain $\Omega_f \cup \Omega_t \cup \Omega_p$ by volume averaged variables $(\langle u \rangle, \langle p \rangle)$. The domain can be seen as a single porous medium characterized by a permeability varying continuously from a constant value K_p in Ω_p to $K_f = \infty$ in Ω_f with transition in Ω_t. At this space scale, a boundary-layer of thickness $\delta = O(\sqrt{K})$ makes the transition between the flows in Ω_f and Ω_p. The flow in Ω is governed by upscaled conserved equations, for instance by the Darcy-Brinkman equation at low Reynolds number. Note that in Ω_f volume averaged variables identify with local flow variables.

Finally at the *macroscopic scale* (figure 8.3, right), only Ω_f and Ω_p are considered. An "abstract" interface Γ separates the two domains: $\Gamma \cap \Omega_f = \Gamma_f$ and $\Gamma \cap \Omega_p = \Gamma_p$. Carefully chosen transmission conditions on Γ make the coupling between Ω_f and Ω_p. These conditions embed the physics of the transition region Ω_t which is now reduced to the interface Γ. Across Γ, volume-averaged flow variables can be discontinuous. If Ω_t has a thickness $\delta = O(\sqrt{K})$, then Γ may embed the boundary-layer. Other definitions of Γ allow a boundary layer inside Ω_p. In Ω_f, the flow is governed by the local (Navier-)Stokes equations while in Ω_p which is viewed from an upper scale, the Darcy law is used (or alternatively the Darcy-Brinkman law for high porosities or also the Darcy-Forchheimer law at high Reynolds numbers). This issue of the transmission condition is two-fold: first the condition has to be physically relevant, second the coupled problem has to be well-posed.

8.2.2 Fluid-porous models

Mesoscopic scale modeling

As exposed in the previous section, at the mesoscopic scale the flow can be modeled by a unique set of equations in the whole domain $\Omega = \Omega_f \cup \Omega_t \cup \Omega_p$. The main advantage of this approach is that no further modeling is required to compute the flow in Ω in contrast to macroscopic models which depend on undetermined parameters. The downside is that the thin boundary layer between the fluid flow and the porous medium must be accurately computed. As a consequence mesoscopic models are more adapted to porous media which feature a smooth transition of the permeability to the free-fluid.

An example of mesoscopic model is the *generalized Brinkman equation* [9] which embeds in a single system of equations the Stokes flow in the fluid region and the Darcy-Brinkman flow in the porous region:

$$\begin{cases} -\operatorname{div}(\tilde{\mu}\,\nabla\,\mathbf{u}) + \nabla\, p + \dfrac{\mu}{K}\mathbf{u} = 0 & \text{in } \Omega \\ \operatorname{div}\mathbf{u} = 0 & \text{in } \Omega \end{cases} \tag{8.2.1}$$

where $\tilde{\mu} = \mu/\phi$ is the *effective viscosity*, μ the dynamic viscosity of the fluid, ϕ the porosity and K the permeability tensor defined as:

$$K = \begin{cases} K_p & \text{in } \Omega_{p'} = \Omega_p \cup \Omega_t \\ \dfrac{1}{\varepsilon} & \text{in } \Omega_f \end{cases}$$

with $\varepsilon \to 0$. A similar approach was presented by E. Arquis and J.-P. Caltagirone [13] for a flow governed by the Navier-Stokes equations in the fluid region. The following generalized momentum equation was used:

$$\rho\partial_t\mathbf{u} + \rho\mathbf{u}\cdot\nabla\,\mathbf{u} = \mu\Delta\mathbf{u} - \left(\frac{\mu}{K}\mathbf{u} + \nabla\, p + \rho\mathbf{g}\right) \tag{8.2.2}$$

A *transition region of variable permeability* (in the present case Ω_t) where the generalized momentum equation is loosely equivalent to the Brinkman equation makes the transition between two regions of constant permeability, respectively the fluid region Ω_f (infinite permeability so that (8.2.2) tends to the Navier-Stokes momentum equation) and the porous region Ω_p (where the Darcy term prevails in (8.2.2)). As for the generalized Brinkman equation (8.2.1) despite being defined at the mesoscopic scale, it can be interpreted as a specific macroscopic model. Indeed as proved by P. Angot in [9] the limit model of (8.2.1) is a two domain problem coupling Stokes/Brinkman with continuity of the velocity and of the normal stress at the interface Γ_f:

$$\begin{cases} -\mu\Delta\mathbf{u} + \nabla\, p = 0 & \text{in } \Omega_f \\ \tilde{\mu}\Delta\mathbf{u} + \nabla\, p + \dfrac{\mu}{K}\mathbf{u} = 0 & \text{in } \Omega_p \cup \Omega_t \\ \operatorname{div}\mathbf{u} = 0 & \text{in } \Omega_f \cup \Omega_p \cup \Omega_t \\ \mathbf{u}|_{\Omega_f} = \mathbf{u}|_{\Omega_p \cup \Omega_t} & \text{on } \Gamma_f \\ (-p\mathbf{n} + \mu\,\nabla\,\mathbf{u}\cdot\mathbf{n})|_{\Omega_f} = (-p\mathbf{n} + \tilde{\mu}\,\nabla\,\mathbf{u}\cdot\mathbf{n})|_{\Omega_p \cup \Omega_t} & \text{on } \Gamma_f \end{cases} \tag{8.2.3}$$

The equivalence between the *single domain* problem (8.2.1) and the *two-domain* problem (8.2.3) has been confirmed with numerical experiments by B. Goyeau & al. in [70]. This so called "two-domain" formulation leads us to classical two-domain interface conditions: two domains are considered, Ω_f and Ω_p which are respectively governed by a different set of equations and coupled on the interface Γ with proper jump conditions. Further discussion on the equivalence between the single domain and the two domain approach is found in reference [82].

Ochoa-Tapia–Whitaker model

In 1995, J.A. Ochoa-Tapia and S. Whitaker used the homogenization technique to derive from the local Stokes equations an interface condition for coupling the Stokes/Brinkman problem [105, 106]:

$$\begin{cases} \tilde{\mu}\partial_n \langle u_\tau \rangle |_{\Omega_p} - \mu \partial_n u_\tau|_{\Omega_f} = \dfrac{\mu\beta}{\sqrt{K}} \langle u_\tau \rangle |_{\Omega_p} & \text{on } \Gamma \\ \langle u_\tau \rangle |_{\Omega_p} = u_\tau|_{\Omega_f} & \text{on } \Gamma \end{cases} \tag{8.2.4}$$

with $\langle u_\tau \rangle |_{\Omega_p}$ the tangential component w.r.t Γ of the volume-average velocity in the porous medium and β the parameter of the law. It was later re-derived by M. Chandesris and D. Jamet using the method of matched asymptotic expansions [37]. The determination of β is still an open problem: indeed in the derivation of OTW law, β depends on *surface excess* quantities for which a closure problem is yet to be determined.

This condition yields a small boundary layer in the porous medium Ω_p. As a result, the Ochoa-Tapia–Whitaker (OTW) model may not be strictly considered as a macroscopic interface model. The continuity of the tangential velocity and therefore of the convection term at the interface Γ is attractive for modeling heat transfer at moderate Reynolds number. However the relevance of the OTW condition depends on that of the Brinkman model, which seems limited to porous media with high porosities [103] (typically $\phi \geq 0.8$).

Despite the OTW law allows a jump of the normal derivative of the tangential velocity, it is closely tied to the mesoscopic model (8.2.1). Indeed in 2003, B. Goyeau & al. [70] established an explicit expression of β as a function of the permeability, the effective viscosity, the thickness of the transition region Ω_t and the average fluid velocity in the porous medium, solution of (8.2.1). A generalized methodology for determining such semi-analytical expression of β is exposed in [127].

Beavers-Joseph model

A properly macroscopic interface model was introduced by G.S. Beavers and D.D. Joseph in 1967 upon physical experiments for coupling the Stokes/Darcy problem [17]:

$$\mu\partial_n u_\tau|_{\Omega_f} = \frac{\mu\alpha}{\sqrt{K}}(u_\tau|_{\Omega_f} - U_p) \text{ on } \Gamma \tag{8.2.5}$$

$$\tag{8.2.6}$$

where U_p is the Darcy velocity in the porous medium and α the slip coefficient, only parameter of the law. The latter is non-dimensional and according to Beavers and Joseph [17], it should depend only on the structure of the porous medium (in particular close to the interface) and not on the properties of the flow. This law is widely accepted for moderate Reynolds numbers at low porosities and high porosities as well. For the case of a small Darcy velocity w.r.t the slip velocity, P.G. Saffman [116] derived a simplified version of (8.2.5) in 1971 :

$$\mu\partial_n u_\tau|_{\Omega_f} = \frac{\mu\alpha}{\sqrt{K}} u_\tau|_{\Omega_f} \text{ on } \Gamma \tag{8.2.7}$$

$$\tag{8.2.8}$$

The Beavers–Joseph–Saffman law was later rigorously derived using the theory of homogenization by W. Jäger and A. Mikelić [80].

The issue of the well-posedness of the coupled Stokes/Brinkman problem with condition (8.2.4) was investigated in [10] and that of the Stokes/Darcy problem with condition (8.2.5) in [35, 10].

Chapter 9

CONVECTIVE REGIME

In this chapter, we focus on an extension of the Beavers-Joseph interface condition for convective flows. First we recall past studies on the dependence of Beavers-Joseph slip coefficient with increasing advection. Then a non-linear interface condition derived from a kinetic energy balance is proposed and evaluated against direct numerical simulations for different flow regimes and porous mediums.

9.1 Previous work

9.1.1 Experimental observations

Notations refereed to in this section are defined on figure 9.1. The interface Γ is parallel to the horizontal axis. The average of the velocity component tangent to the interface Γ along direction x is denoted $\langle u_x \rangle_x$ and the Darcy velocity U_p.

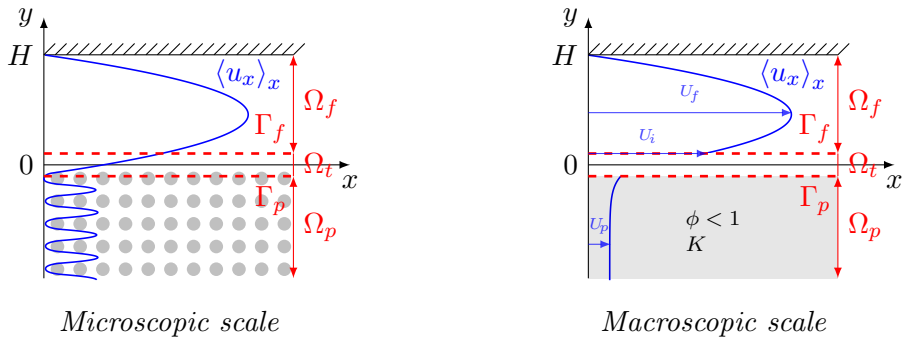


FIG. 9.1 – *Pressure-driven flow over a porous medium ; microscopic scale (left) and macroscopic scale (right).*

Few studies have challenged the validity of the Beavers-Joseph law for high speed flows and no definite conclusion can be drawn yet from these prior works. In 1974, G.S. Beavers, E.M. Sparrow and B.A. Masha performed several physical experiments [18] involving an air flow over a sample of Foemetal for increasing Reynolds numbers in the same configuration as the original experiment of G.S. Beavers and D.D. Joseph [17]. Their measured Reynolds number Re_f is based on the height and on the average velocity in the duct section. Their experiments were performed with Re_f up to 1000 and the corresponding values of the slip coefficient α did

not seem to show a dependence of α with Re_f. The dispersion within the values of α deduced from their measurements was lower than 5%.

M. Sahraoui and M. Kaviany investigated this issue in their 1991 paper [117] using direct numerical simulations also in the same configuration as [17]. The incompressible Navier-Stokes equations were solved in a 2D channel over an array of cylinders. The authors were interested in the dependence of α with respect to a Reynolds number Re_l defined by the length of the porous periodic cell and the Darcy velocity. The slip coefficient α was found to increase with the local Reynolds number Re_l for $0.1 < Re_l < 10$.

Direct numerical simulations were also performed in 2011 by Q. Liu and A. Prosperetti [96] in a configuration similar to [117] but with a porous medium instead of the upper solid wall. The incompressible Navier-Stokes equations were solved in a 3D domain with spheric inclusions. The Reynolds number of interest Re_i was determined by the sphere diameter and by the local fluid velocity at the interface. In the range of local Reynolds number tested $0.5 < Re_i < 16$ the authors observe a linear increase of α with Re_i.

9.1.2 Interface models

A first extension of the Beavers-Joseph law was proposed by Q. Liu and A. Prosperetti with a variable slip coefficient [96]:

$$\mu\partial_y \langle u_x\rangle_x = \frac{\mu\alpha(Re_i)}{\sqrt{K}}(\langle u_x\rangle_x - U_p)$$
$$\text{with } \alpha(Re_i) = \alpha^0(1+\theta Re_i)$$

with $Re_i = 2aU_i/\mu$, a the spheres' radius and $\langle u_x\rangle_x$ the horizontal fluid velocity at the interface Γ. The parameters of this law are $\alpha^0, \theta \in \mathbb{R}$. This interface condition seems to fit the authors' numerical results within the range of local Reynolds number $0.5 < Re_i < 16$. While this condition provides a first insight into the modeling of an extension to Beavers-Joseph law in the convective regime, its validity has yet to be assessed on a wider range of Reynolds number.

On the other hand in 2013, A. Marciniak-Czochra and A. Mikelic introduced an extension of Beavers-Joseph-Saffman law [98]. In addition to the Saffman hypothesis $\langle u_x\rangle_x \gg U_p$ the authors assume a pressure-driven flow only in the fluid channel. Their interface condition is derived rigorously using a specific technique for establishing interface conditions exposed in detail in [81]. A first order interface condition is obtained using the theory of matched asymptotic expansions [132]. Higher order "corrective" terms to the interface condition are obtained by considering a boundary layer problem at the fluid-porous interface. The non-linear interface condition takes the following form:

$$\langle u_x\rangle_x = C(\varepsilon,\eta)\big(\partial_y \langle u_x\rangle_x\big) - \varepsilon^{3/2+\eta}\left\langle \beta\left(\frac{x}{\varepsilon}\right)|_\Gamma\right\rangle \big(\partial_y \langle u_x\rangle_x\big)^2 + O(\varepsilon^{3/2+19\eta/12})$$

where ε is the characteristic pore size, η a parameter which depends on the Reynolds number and on the order of magnitude of the fracture width and $\beta(x/\varepsilon)$ a function which arises from the boundary layer problem at the interface. This interface condition can be though as the Beavers-Joseph-Saffman law with a high order correction introduced by the second term at the right hand side. To our best knowledge this is the only interface condition derived rigorously from local equations using the theory of homogenization.

9.2 Proposed interface condition

When convection becomes significant, the dynamic pressure is not negligible and is expected to grow accordingly. Therefore, a jump of velocity at the arbitrary interface Γ would result in jump of the kinetic energy at the interface. We propose an interface condition where the normal derivative of the tangential velocity is proportional to the jump of kinetic energy at interface Γ:

$$\mu\partial_y \langle u_x\rangle_x = \frac{\mu\alpha_{\text{kin}}}{2\left|\langle u_x\rangle_x\right|\sqrt{K}}\left(\left|\langle u_x\rangle_x\right|^2 - |U_p|^2\right) \tag{9.2.1}$$

This law relies only on a single parameter α_{kin}, which we expect to depend only on the geometry of the porous medium, not on the properties of the flow. This interface condition is thought as an extension of the Beavers-Joseph law. Indeed the jump of kinetic energy can be reformulated as:

$$\mu\partial_y \langle u_x\rangle_x = \frac{\mu\alpha_{\text{nl}}}{\sqrt{K}}\left(\langle u_x\rangle_x - U_p\right) \quad \text{with } \alpha_{\text{nl}} = \frac{\alpha_{\text{kin}}}{2}\left(1 + \frac{U_p}{\langle u_x\rangle_x}\right) \tag{K1}$$

Remark 9.2.1. Another motivation for introducing the kinetic energy comes from recent works on the well-posedness of the Stokes problem with non-linear outflow conditions [11].

Our numerical results have led us to propose a variation of (K1) with a second parameter $\gamma(\phi)$ where the slip coefficient α_{nl} is a convex function of $1 + U_p/\langle u_x\rangle_x$. Unlike (K1), it is not possible to link the jump condition to a local kinetic energy balance anymore except for $\gamma = 1$:

$$\boxed{\mu\partial_y \langle u_x\rangle_x = \frac{\mu\alpha_{\text{nl}}}{\sqrt{K}}\left(\langle u_x\rangle_x - U_p\right) \quad \text{with } \alpha_{\text{nl}} = \frac{\alpha_{\text{kin}}}{2}\left(1 + \frac{U_p}{\langle u_x\rangle_x}\right)^{\gamma(\phi)}} \tag{K2}$$

Remark 9.2.2. For consistency with the viscous regime we set $\lim_{Re\to 0}\alpha_{nl} = \alpha_{BJ}$ with α_{BJ} the original slip coefficient of Beavers-Joseph law.

9.3 Problem setting

In order to assess the validity of the proposed interface condition (K2), several numerical simulations are performed with different porous medium. The incompressible Navier-Stokes equations are solved in the fluid and the periodic porous region.

9.3.1 Continuous problem

The steady-state Navier-Stokes equations with an averaged pressure gradient $\langle\nabla p\rangle$ imposed are solved in non-dimensional form in $\Omega = [0, L]\times[-H, H]$ with $H = 2$ and $L = 2H/5$ (see figure 9.1). The problem is L-periodic, the fluid channel is located in the $y > 0$ region and the porous medium in the $y \leq 0$ region. The latter is made of a periodic array of elementary porous cells $[0, L]\times[0, L]$ with either square inclusions of edge length a or cylinder inclusions of diameter a. The unknowns are the velocity field $\mathbf{u}$ and the pressure perturbation $\tilde{p}$. The continuous problem reads:

$$\begin{cases} \operatorname{div}(\mathbf{u}\otimes\mathbf{u}) - \dfrac{1}{Re}\Delta\mathbf{u} + \nabla\tilde{p} + \langle\nabla p\rangle = 0 & \text{in } \Omega \\ \nabla\cdot\mathbf{u} = 0 & \text{in } \Omega \\ \mathbf{u}|_{\Sigma_{\text{in}}} = \mathbf{u}|_{\Sigma_{\text{out}}},\ \tilde{p}|_{\Sigma_{\text{in}}} = \tilde{p}|_{\Sigma_{\text{out}}} & \text{on } \partial\Omega \\ \partial_n u_x|_{\Sigma_{\text{inf}}} = 0,\ u_y|_{\Sigma_{\text{inf}}} = 0,\ \mathbf{u}|_{\Sigma_{\text{wall}}} = 0 & \end{cases}$$

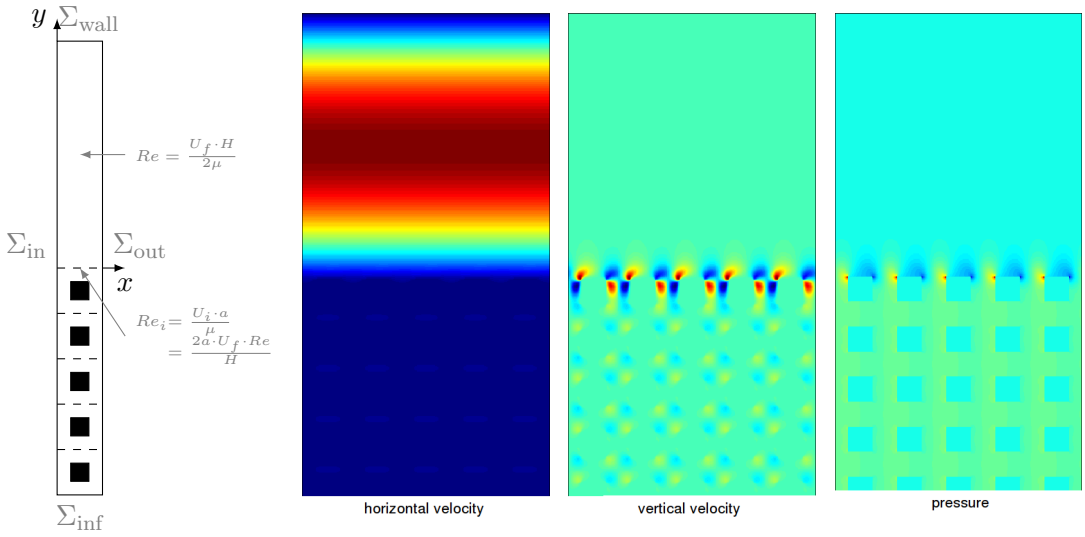


FIG. 9.2 – *Direct numerical simulation of a pressure-driven near-parallel flow.*

with $\langle \nabla p \rangle = -1 \cdot \mathbf{e}_x$. The convection in the fluid channel is controlled by the global Reynolds number Re:

$$Re = \frac{U_f \cdot H}{2\mu}$$

where U_f is the average fluid velocity over the channel section and $H/2$ the half-height of the channel. A local Reynolds number Re_i is also defined in order to have an insight on the amount of advection in the region between the fluid and the porous medium:

$$Re_i = \frac{U_i \cdot a}{\mu}$$

where $U_i \equiv \langle u_x \rangle_x$ is the fluid velocity at the interface Γ and a the characteristic size of the inclusion. The two Reynolds numbers are related as:

$$Re_i = \frac{2a \cdot U_f \cdot Re}{H}$$

9.3.2 Numerical methods

We will consider cell-centered finite-volume discretizations on a uniform $N \times 10N$ Cartesian grid with $N \in \{64, 128\}$. Two numerical methods were used to compute the steady-state Navier-Stokes equations: a projection method where a transient must be solved in order to reach the steady state and Newton's method which allows a direct resolution of the steady-state. The porous inclusions are taken into account using a first order immersed boundary method.

Projection method

For the only purpose of presenting the classical projection method used here, Dirichlet boundary conditions are assumed for the velocity on the boundary $\partial\Omega$. We recall briefly the formulation of incremental pressure-correction schemes [43, 123, 124, 72] and how to address the issues

arising from the cell-centered finite-volume discretization. The aim is to solve the unsteady incompressible Navier-Stokes equations, which read in semi-discrete form:

$$\begin{cases} \dfrac{1}{\delta t}(\mathbf{u}^{n+1}-\mathbf{u}^n)+\operatorname{div}(\mathbf{u}^{n+1}\otimes\mathbf{u}^n)-\dfrac{1}{Re}\Delta\mathbf{u}^{n+1}+\nabla\tilde{p}^{n+1}+\langle\nabla p\rangle=0 & \text{in } \Omega \\ \operatorname{div}\mathbf{u}^{n+1}=0 & \text{in } \Omega \\ \mathbf{u}^{n+1}=\mathbf{g} & \text{on } \partial\Omega \end{cases} \tag{9.3.1}$$

The velocity and the pressure gradient are centered. The discretization is second order in space and first order in time. The above system is solved in a decoupled fashion. First a velocity prediction $\mathbf{u}^{n+1/2}$ is calculated by solving a momentum balance without accounting the incompressibility constrain:

$$\begin{cases} \dfrac{1}{\delta t}(\mathbf{u}^{n+1/2}-\mathbf{u}^n)+\operatorname{div}(\mathbf{u}^{n+1/2}\otimes\mathbf{u}^n)-\dfrac{1}{Re}\Delta\mathbf{u}^{n+1/2}+\nabla\tilde{p}^{n}+\langle\nabla p\rangle=0 & \text{in } \Omega \\ \mathbf{u}^{n+1/2}=\mathbf{g} & \text{on } \partial\Omega \end{cases} \tag{9.3.2}$$

The Helmholtz-Hodge decomposition [91] yields the following decomposition of the predicted velocity $\mathbf{u}^{n+1/2}$ into a solenoidal field and an irrotational field:

$$\mathbf{u}^{n+1/2}=\mathbf{u}^{n+1}+\delta t\,\nabla\phi \tag{9.3.3}$$

with $\phi=\tilde{p}^{n+1}-\tilde{p}^n$ for consistency with the original problem (9.3.1). Finding ϕ allows to correct the predicted velocity $\mathbf{u}^{n+1/2}$ so as to verify the incompressibility constrain $\operatorname{div}\mathbf{u}^{n+1}=0$. Using this decomposition a Poisson problem is derived on the pressure increment ϕ:

$$\Delta\phi=\frac{1}{\delta t}\operatorname{div}\mathbf{u}^{n+1/2}\text{ in }\Omega \tag{9.3.4}$$

with homogeneous Neumann boundary conditions for ϕ on $\partial\Omega$ in order to enforce $\mathbf{u}^{n+1}\cdot\mathbf{n}=\mathbf{u}^{n+1/2}\cdot\mathbf{n}$ on $\partial\Omega$. This "non-natural" boundary condition introduces an error in the form the classical artificial boundary layer on the pressure. However, the most critical accuracy issue comes rather from the cell-centered discretization itself. It is well-known that the discrete problem (9.3.1) is not well-posed [60]. The kernel of the cell-centered finite-volume pressure gradient features constant pressures and checkerboard modes. Though system (9.3.1) cannot be solved, it is still possible to obtain a solution using a projection method, which introduces a decoupling between the velocity $\mathbf{u}$ and the pressure $\tilde{p}$.

The constant pressure issue is addressed by imposing the value of the pressure in one cell. The apparition of checkerboard pressure modes is not systematic, and it can be treated for instance using Rhie&Chow stabilization technique [115]. Let us first sum up the steps of the projection method introduced here:

1. Predict velocity $\mathbf{u}^{n+1}$ using (9.3.2)
2. Compute pressure increment $\phi=\tilde{p}^{n+1}-\tilde{p}^n$ using (9.3.4)
3. Correct the velocity using (9.3.3)

We introduced the Rhie&Chow correction using the approach of S. Faure, J. Laminie and R. Temam in [61]. A discrete quantity w is indexed as w_{ij} on the Cartesian grid with $1\leq i\leq N$ and $1\leq j\leq 10N$. The pressure contribution to the predicted horizontal velocity is of the form:

$$\frac{1}{a_{ij}}\frac{h}{2}\left(\tilde{p}^n_{i+1,j}-\tilde{p}^n_{i-1,j}\right)$$

with $a_{ij} = \frac{h^2}{\delta t} + \frac{4}{Re} + \frac{1}{2}(\text{div}\,\mathbf{u}^n)_{ij}$. The contribution of this horizontal pressure gradient to the divergence of the predicted velocity $\mathbf{u}^{n+1/2}$ in the right hand side of 9.3.4 exhibits a decoupling between neighboring pressure unknowns:

$$-\frac{h}{4}\left[\frac{1}{a_{i-1,j}}\tilde{p}_{i-2,j} - \left(\frac{1}{a_{i-1,j}} + \frac{1}{a_{i+1,j}}\right)\tilde{p}_{ij} + \frac{1}{a_{i+1,j}}\tilde{p}_{i+2,j}\right]$$

Instead, a stabilization term is introduced in the right hand side of 9.3.4 so that the contribution of the horizontal pressure gradient takes the following form:

$$-\frac{h}{4}\left[\frac{1}{a_{ij}}\tilde{p}_{i-1,j} - \left(\frac{1}{a_{i-1,j}} + \frac{1}{a_{i+1,j}}\right)\tilde{p}_{ij} + \frac{1}{a_{ij}}\tilde{p}_{i+1,j}\right]$$

The complete correction term for the Rhie&Chow stabilization of the full pressure gradient from the momentum equation 9.3.2 is then deduced:

$$\begin{aligned}
R_{ij} &= h(w_{i+1/2,j} - w_{i-1/2,j} + w_{i,j+1/2} - w_{i,j-1/2}) \\
w_{i+1/2,j} &= \frac{h}{4}\left[\left(\frac{1}{a_{i+1,j}}\tilde{p}_{i+2,j} - \frac{2}{a_{ij}}\tilde{p}_{i+1,j} + \frac{1}{a_{i+1,j}}\tilde{p}_{ij}\right) - \left(\frac{1}{a_{ij}}\tilde{p}_{i+1,j} - \frac{2}{a_{i+1,j}}\tilde{p}_{ij} + \frac{1}{a_{ij}}\tilde{p}_{i-1,j}\right)\right] \\
w_{i,j+1/2} &= \frac{h}{4}\left[\left(\frac{1}{a_{i,j+1}}\tilde{p}_{i,j+2} - \frac{2}{a_{ij}}\tilde{p}_{i,j+1} + \frac{1}{a_{i,j+1}}\tilde{p}_{ij}\right) - \left(\frac{1}{a_{ij}}\tilde{p}_{i,j+1} - \frac{2}{a_{i,j+1}}\tilde{p}_{ij} + \frac{1}{a_{ij}}\tilde{p}_{i,j-1}\right)\right]
\end{aligned}$$

In some respect this correction mimics the MAC scheme in that it attempts to build the classical centered pressure gradient of the MAC scheme. This helps to damp the oscillations but on the other hand it adds further diffusion to the pressure.

Convergence tests for the target configuration (square or cylindrical obstacle) and with appropriate boundary conditions (periodic, Dirichlet, Neumann) were performed for the projection method and are given in figure 9.3.

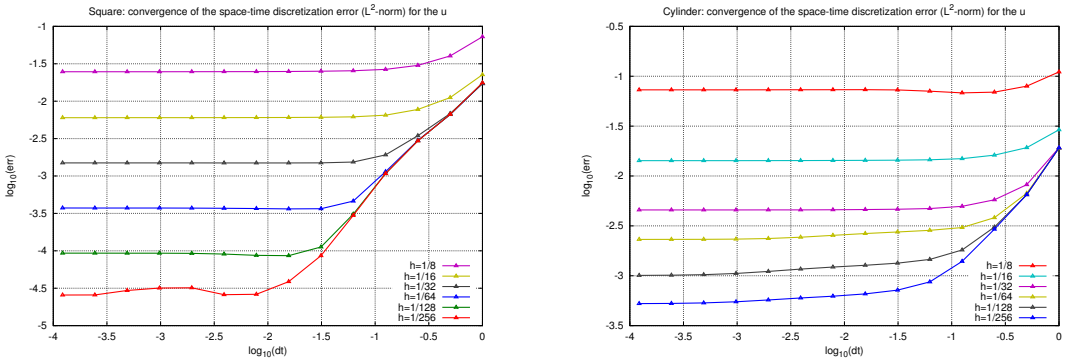


FIG. 9.3 – *Convergence test for the projection method: L^2 space-time error for the velocity with a square obstacle (left) and a cylindrical obstacle (right).*

Newton's method

The advantage of using a projection method to solve the steady state of the discrete problem (9.3.1) lies in its simplicity, and in the limited influence of checkerboard pressure modes due to the loose coupling between the pressure and the velocity. On the other hand it is much slower than a direct resolution of the steady-state Navier-Stokes equations as the entire transient

must be solved. Moreover, the relevance of the steady-state solution depends on the steady-state criteria set to stop the pseudo-time iterations.

As a consequence we also considered solving directly the steady-state Navier-Stokes equations using Newton's method i.e. solve the non-linear problem $\mathcal{F}(X) = 0$ with:

$$\mathcal{F}(X) = \begin{cases} \displaystyle\sum_{\sigma\in\mathcal{E}_K} |\sigma|\mathbf{u}_\sigma u_{K,\sigma} - \frac{1}{Re}\sum_{\sigma\in\mathcal{E}_K}|\sigma|\frac{\mathbf{u}_\sigma - \mathbf{u}_K}{d_{K,\sigma}} + \sum_{\sigma\in\mathcal{E}_K}|\sigma|\tilde{p}_\sigma\mathbf{n}_{K,\sigma} + |K|\,\langle\nabla p\rangle \\ \displaystyle\sum_{\sigma\in\mathcal{E}_K}|\sigma|u_{K,\sigma} \end{cases}$$

and the block unknown $X = [(\mathbf{u}_K)_{K\in\mathcal{T}}\ (\tilde{p}_K)_{K\in\mathcal{T}}]$. The edge velocity is centered and the pressure gradient as well. It is well known that this problem is not well-posed [60]. We address this issue by adding a stabilization term, inspired by the Brezzi-Pitkäranta technique for the finite element method [32, 59]:

$$\mathcal{F}(X) = \begin{cases} \displaystyle\sum_{\sigma\in\mathcal{E}_K} |\sigma|\mathbf{u}_\sigma u_{K,\sigma} - \frac{1}{Re}\sum_{\sigma\in\mathcal{E}_K}|\sigma|\frac{\mathbf{u}_\sigma - \mathbf{u}_K}{d_{K,\sigma}} + \sum_{\sigma\in\mathcal{E}_K}|\sigma|\tilde{p}_\sigma\mathbf{n}_{K,\sigma} + |K|\,\langle\nabla p\rangle \\ \displaystyle\sum_{\sigma\in\mathcal{E}_K}|\sigma|u_{K,\sigma}\boxed{-\lambda\sum_{\sigma\in\mathcal{E}_K}|\sigma|(h_K+h_L)(\tilde{p}_L-\tilde{p}_K)} \end{cases}$$

with λ a stabilization parameter, which we set equal to:

$$\lambda = h_\mathcal{T}/10 \tag{9.3.5}$$

If λ is too high, the stabilization introduces too much diffusion in the solution. On the other hand the smaller λ, the worst the conditioning of the Jacobian matrix $J(X)$ of functional $\mathcal{F}$. Nevertheless we defined λ according to (9.3.5) and avoided the issue of dealing with an ill-conditioned Jacobian by computing its LU decomposition with a direct solver for iterating on $J(X^k)\times(X^{k+1}-X^k) = -\mathcal{F}(X^k)$.

Immersed obstacles

Porous inclusions are taken into account using a first order embedded boundary method. The boundary condition imposed on a piece-wise approximation of the original boundary are reported on the respective mesh faces which form the staircase boundary (see figure 9.4). The geometry of the square inclusions is chosen so as to always match the faces of the Cartesian grid hence no further error introduced. As for cylinder inclusions, a $o(h)$ error is introduced due to the staircase approximation.

9.4 Numerical Results

9.4.1 Overview

Previous works [117, 96] have suggested that the slip coefficient of the Beavers-Joseph law increases with advection. Our objective is to verify whether the parameters of interface condition (K2) — α_{nl} and γ — depend only on the geometry of the porous matrix or on the fluid flow as well.

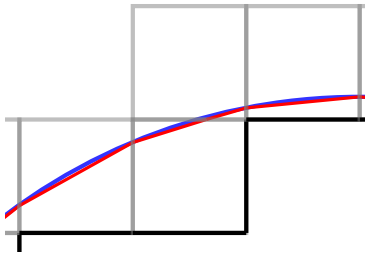


FIG. 9.4 – *Piecewise (red line) and staircase (black line) approximation of the obstacle boundary (blue line).*

Several numerical simulations are performed with a Reynolds number Re ranging from 10 to 100 by 10 increment. Two types of porous inclusions are tested matching a porosity between $\phi = 0.2$ and $\phi = 0.99$. The shape of the inclusions is defined by characteristic size a — diameter for the cylinders and edge length for squares. These configurations are listed in table 9.1 in the case of a *porous unit cell*.

In configurations S1 to S5 a pressure-correction scheme was used whereas in configurations C1 to C7 were handled with Newton's method. For the pressure-correction scheme, the time-step was $\delta t = 0.1$ and the stopping criteria:

$$\max\left(\frac{|u_x^{n+1} - u_x^n|_{L^\infty}}{|u_x^n|_{L^\infty}}, \left|\alpha_{\text{nl}}^{n+1} - \alpha_{\text{nl}}^n\right|\right) < 10^{-4}\delta t$$

with α_{nl}^n an estimation of the slip coefficient. For a 64×640 Cartesian grid, between 5000 and 10000 iterations were needed to match the steady-state criteria. As for Newton's method, the stopping criteria for Newton's iterations was:

$$\left|X^{k+1} - X^k\right|_{L^\infty} < 10^{-9}$$

About 3 to 7 iterations were needed to converge. The use of a direct solver, though expensive with respect to the memory, allowed to perform every Newton iteration at a constant cost.

Name	**Shape**	**a**	ϕ	$\mathbf{K_{xx}}$
S1	square	0.875	0.23	2.1×10^{-4}
S2	square	0.75	0.44	1.6×10^{-3}
S3	square	0.5	0.75	1.4×10^{-2}
S4	square	0.25	0.94	6.0×10^{-2}
S5	square	0.125	0.98	1.2×10^{-1}
C1	cylinder	0.85	0.43	1.1×10^{-3}
C2	cylinder	0.75	0.56	3.6×10^{-3}
C3	cylinder	0.65	0.67	8.4×10^{-3}
C4	cylinder	0.55	0.76	1.6×10^{-2}
C5	cylinder	0.45	0.84	3.0×10^{-2}
C6	cylinder	0.35	0.90	4.7×10^{-2}
C7	cylinder	0.25	0.95	7.2×10^{-2}

TAB. 9.1 – *Configurations of the different periodic porous medium tested.*

Remark 9.4.1. In order to compute the permeability of each porous medium tested, the Stokes problem is solved in a porous unit cell $\Omega = [0,1] \times [0,1]$ with periodic conditions on the entire boundary $\partial\Omega$. The dynamic viscosity is set to $\mu = 1$ and an average pressure gradient $\langle \nabla p \rangle$ is imposed either in the horizontal or in the vertical direction.

$$\begin{cases} -\mu \Delta \mathbf{u} + \nabla \tilde{p} + \langle \nabla p \rangle = \frac{1}{\varepsilon} \mathbf{u} \text{ in } \Omega \\ \operatorname{div} \mathbf{u} = 0 \text{ in } \Omega \\ \mathbf{u}, \tilde{p} \text{ periodic on } \partial\Omega \end{cases}$$

The obstacle is taken into account using volume penalization: inside the inclusion the parameter ε tends to 0 and outside it tends to $+\infty$. Then the local velocities are upscaled using Darcy's law. The symmetry of the porous cell yields to $K_{xx} = K_{yy}$ and $K_{xy} = 0$.

$$\langle \mathbf{u} \rangle = -\frac{1}{\mu} \begin{pmatrix} K_{xx} & K_{xy} \\ K_{yx} & K_{yy} \end{pmatrix} \underbrace{\langle \nabla p \rangle}_{=-1 \cdot \mathbf{e}_x} \Rightarrow K_{xx} = \langle \mathbf{u} \rangle \cdot \mathbf{e}_x$$

9.4.2 Interface condition parameters

Figure 9.5 (left) shows the velocity profile $\langle u_x \rangle_x (y)$ in the first porous cell below the fluid channel. The inclusion in this porous cell is here a square of edge length 0.05 centered at $y = -0.05$. The cross marks match the center of the finite volumes cells of the Cartesian grid. Figure 9.5 (right) gives a closer view on boundary layer between $y = -0.04$ and $y = -0.015$.

The interface Γ is defined as the plane above the surface tangent to the inclusions to a distance h (position (a) on figure 9.5). Alternative choices such as (b) were too far from the obstacle to capture the physics modeled by our interface condition. The choice of interface position for fluid-porous interface conditions is still subject to discussion, see for instance [37], [70] and references therein.

Assuming the choice of interface position described above, the averaged velocity $\langle u_x \rangle_x$ increases almost linearly with y. The choice of the method for computing this normal derivative

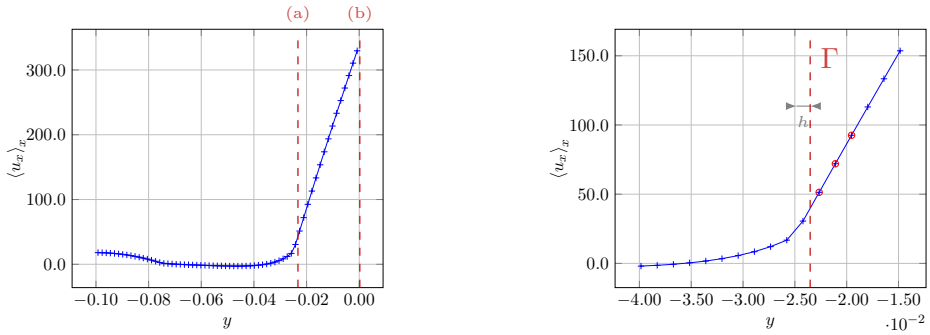


FIG. 9.5 – *Choice of interface position and stencil for normal derivative.*

does not have a significant impact on the verification of the law. We choose a second order finite-difference formulae. The three-point stencil is located above the interface (markers circled in red in figure 9.5, right).

$$(\partial_y \langle u_x \rangle_x)_{j+1/2} = \frac{1}{h}\left[-(\langle u_x \rangle_x)_{j+2} + 3(\langle u_x \rangle_x)_{j+1} - 2(\langle u_x \rangle_x)_j\right]$$

The evolution of the ratio $\partial_y \langle u_x \rangle_x / (\langle u_x \rangle_x - U_p)$ with $(1 + U_p/\langle u_x \rangle_x)^\gamma$ (which increases with the Reynolds number Re) if linear would confirm the law and the slope would be $\alpha_{\text{kin}}/(2\sqrt{K})$.

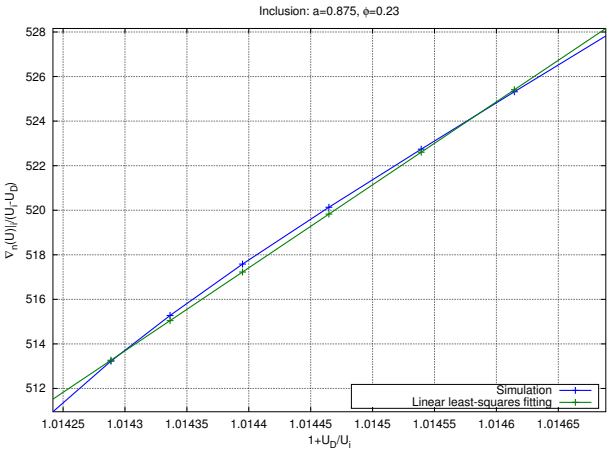
9.4.3 Validity for a channel flow

Moderate porosity

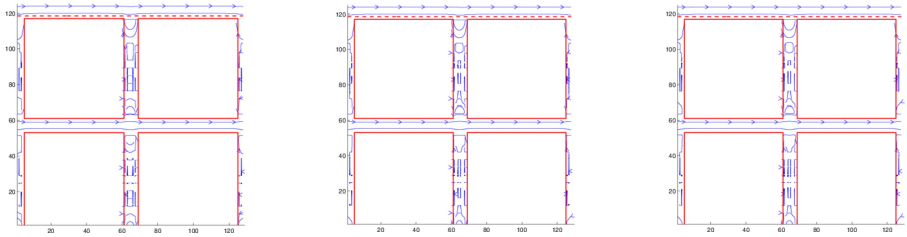
The porosity deemed as "moderate" when lower than 0.75, i.e. for configurations S1, S2, S3, C1, C2, C3 and C4. It seems we always have $\gamma(\phi) = 1$, which is interesting as the interface condition is effectively the kinetic energy jump at the interface (9.2.1). As for the non-linear slip coefficient, it is found to be independent from the flow regime. For instance with configuration C1 the local Reynolds number Re_i varies between 4.4×10^{-3} and 32 but we always have $\alpha_{\text{nl}}/(2\sqrt{K}) = 6.4 \times 10^3$. It should be noted that the interface condition seem to match the numerical results for flows compatible with the Saffman hypothesis (eg. with configuration C1, $U_p/\langle u_x \rangle_x = 8 \times 10^{-3}$) and for flow where the Darcy velocity is close to the fluid velocity at the interface (eg. with configuration C4, $U_p/\langle u_x \rangle_x = 1.2 \times 10^{-1}$).

High porosity

Configurations S4, S5, C5, C6 and C7 are associated with high porosities ($\phi > 0.75$) and Darcy velocities closer to the fluid velocity at the interface, with a ratio $0.18 < U_p/\langle u_x \rangle_x < 0.67$. This time it is necessary to determine parameter $\gamma(phi) > 1$. Note that we don't have a link to a kinetic energy balance anymore. The non-linear slip coefficient, is found to be independent from the flow regime. Regarding parameter γ, numerical results for cylinders and for squares as well indicate that the higher the porosity (and the higher the ratio $U_p/\langle u_x \rangle_x$), the lower γ. It is not clear whether γ depends only on the geometry (more specifically on ϕ) or also on the fluid flow. Moreover, the transition from $\gamma = 1$ to high values of γ is not obvious from current results.



Re	Re_i	$\langle u_x \rangle_x$	U_p
10	7.0e-2	2.0e-2	2.9e-4
20	2.8e-1	4.0e-2	5.7e-4
30	6.3e-1	6.0e-2	8.6e-4
40	1.1	7.9e-2	1.1e-3
50	1.7	9.9e-2	1.4e-3
60	2.5	1.2e-1	1.7e-3
70	3.3	1.4e-1	2.0e-3
80	4.3	1.6e-1	2.3e-3



Re=10 *Re=50* *Re=80*

FIG. 9.6 – *Validity of the interface condition, configuration S1*

9.4.4 Validity for a thin film flow

We now turn to the so-called thin-film configuration. This time the height of the fluid channel is smaller with respect to the porous cells. The fluid domain extends to $[0, L] \times [0, H]$ and the porous domain to $[0, L] \times [-5H/2, 0]$. The computational domain is still a Cartesian grid, this time of size $N \times 7N$. Only cylinder inclusions are considered here, and periodic porous mediums of configurations $T1$ to $T7$ are respectively the same as configurations $C1$ to $C7$. In such configurations, the Saffman hypothesis is generally not satisfied: we have $0.1 < U_p / \langle u_x \rangle_x < 1$ for configurations T2 to T7.

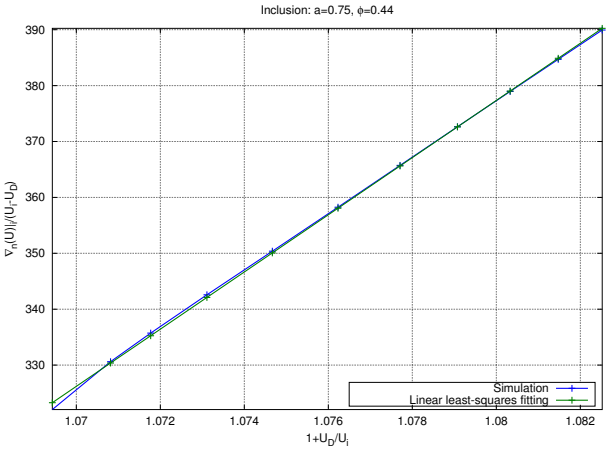
The linearity sought after in the fittings of figures 9.18 to 9.22 is much more difficult to obtain, even for optimal values of $\gamma(\phi)$. For the highest porosity at $\phi = 0.95$, the ratio $\partial_y \langle u_x \rangle_x / (\langle u_x \rangle_x - U_p)$ does not even increases monotonically with $(1 + U_p / \langle u_x \rangle_x)^{\gamma}(\phi)$.

Conclusion

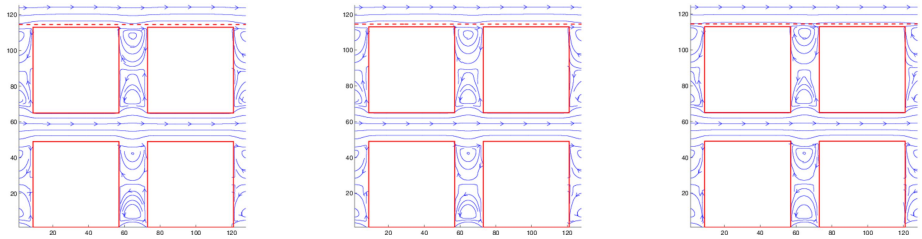
The parameters of the proposed model for the channel fluid flow are summed up in table 9.2. We recall the interface condition proposed for the non-linear regime:

$$\boxed{\mu \partial_y \langle u_x \rangle_x = \frac{\mu \alpha_{\text{nl}}}{\sqrt{K}} \left(\langle u_x \rangle_x - U_p \right) \quad \text{with } \alpha_{\text{nl}} = \frac{\alpha_{\text{kin}}}{2} \left(1 + \frac{U_p}{\langle u_x \rangle_x} \right)^{\gamma(\phi)}}$$

Following the presentation of numerical results for several flow ranges and porous matrices we draw the following conclusions:



Re	Re_i	$\langle u_x \rangle_x$	U_p
1	1.0e-3	3.4e-3	2.4e-4
10	1.0e-1	3.3e-2	2.4e-3
20	3.9e-1	6.6e-2	4.7e-3
30	8.7e-1	9.7e-2	7.1e-3
40	1.5	1.3e-1	9.4e-3
50	2.3	1.5e-1	1.2e-2
60	3.3	1.8e-1	1.4e-2
70	4.4	2.1e-1	1.6e-2
80	5.6	2.3e-1	1.9e-2
90	7.0	2.6e-1	2.1e-2
100	8.6	2.9e-1	2.4e-2

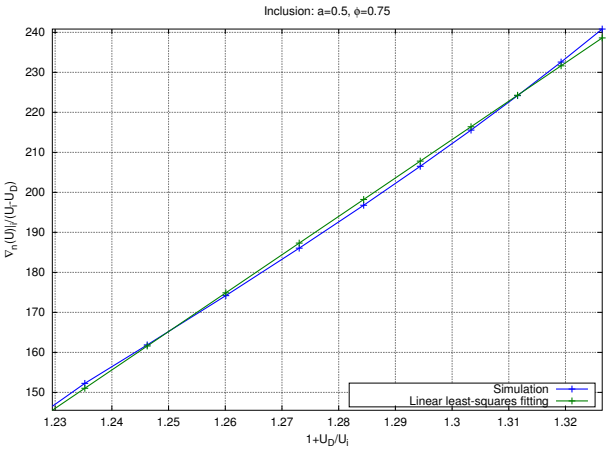


Re=10 *Re=50* *Re=100*

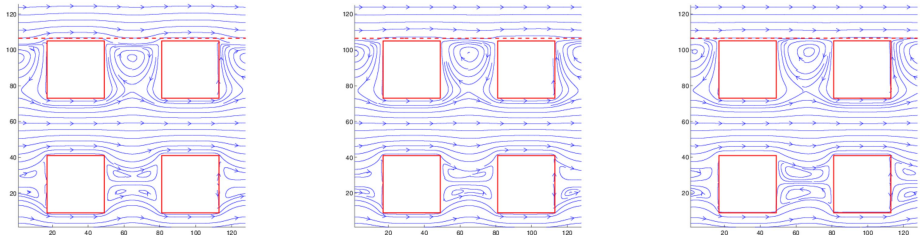
FIG. 9.7 – *Validity of the interface condition, configuration S2*

1. Parameter α_{nl} only depends on the geometry
2. Parameter $\gamma(\phi)$ is constant for moderate porosities: $\gamma(\phi) = 1$ if $\phi < 0.75$.
3. For high porosities, $\gamma(\phi)$ is of order 10^1 and seems to decrease when $\phi \to 1$. For such values of ϕ it may be relevant to consider a non-linear extension of Ochoa-Tapia–Whitaker law.
4. At the limit $\phi \to 1$ we tend to a fluid-fluid configuration where our modeling does not make sense anymore.
5. When $\gamma(\phi) = 1$ the interface condition is equivalent to a jump of the kinetic energy
6. For thin film configurations the interface condition is more loosely verified but still stands for $\phi \leq 0.9$.

To our best knowledge, this interface condition is the more accurate modelling proposed so far for extending Beavers-Joseph law in the convective regime. Furthermore it has a strong physical interpretation as it is derived from a kinetic energy balance at the interface.

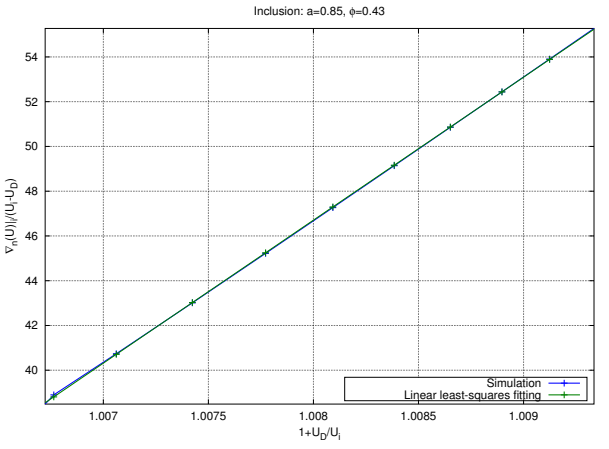


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	1.9e-3	9.3e-3	2.1e-3
10	1.8e-1	9.0e-2	2.1e-2
20	6.9e-1	1.7e-1	4.2e-2
30	1.5	2.4e-1	6.3e-2
40	2.5	3.1e-1	8.4e-2
50	3.7	3.7e-1	1.1e-1
60	5.1	4.3e-1	1.3e-1
70	6.8	4.8e-1	1.5e-1
80	8.6	5.4e-1	1.7e-1
90	11	5.9e-1	1.9e-1
100	13	6.4e-1	2.1e-1

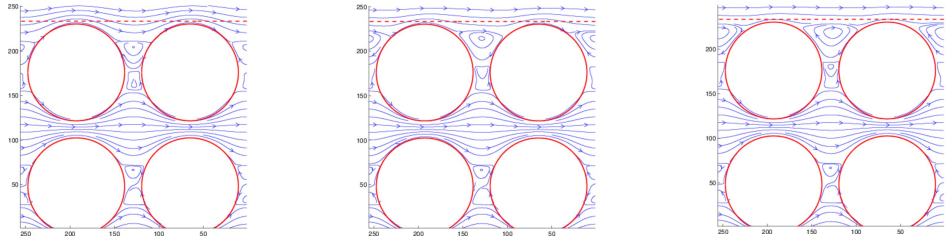


Re=10 *Re=50* *Re=100*

FIG. 9.8 – *Validity of the interface condition, configuration S3*

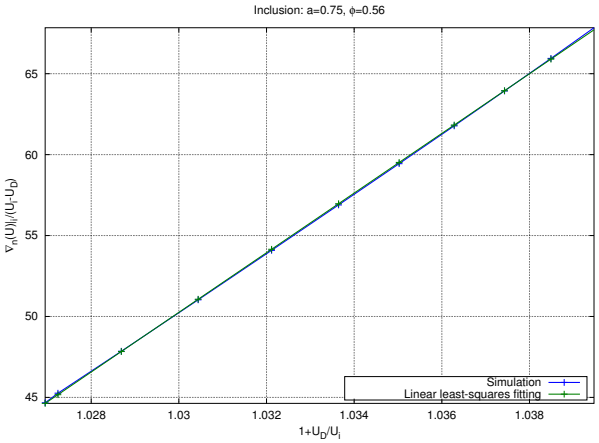


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	4.4e-3	2.6e-2	1.8e-4
10	4.4e-1	2.6e-1	1.8e-3
20	1.7	5.0e-1	3.5e-3
30	3.6	7.1e-1	5.3e-3
40	6.1	9.0e-1	7.0e-3
50	9.2	1.1	8.7e-3
60	1.3e+1	1.3	1.0e-2
70	1.7e+1	1.4	1.2e-2
80	2.1e+1	1.6	1.4e-2
90	2.6e+1	1.7	1.6e-2
100	3.2e+1	1.9	1.7e-2

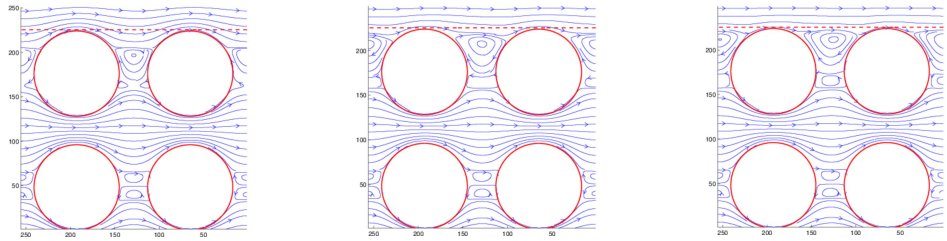


Re=10 *Re=50* *Re=100*

FIG. 9.9 – *Validity of the interface condition, configuration C1*

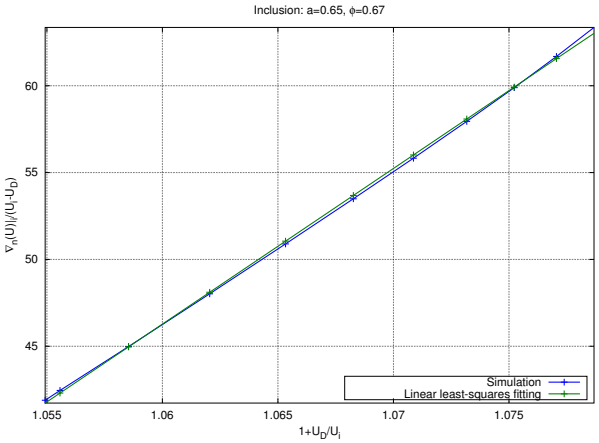


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	3.5e-3	2.3e-2	6.3e-4
10	3.5e-1	2.3e-1	6.3e-3
20	1.3	4.4e-1	1.3e-2
30	2.8	6.2e-1	1.9e-2
40	4.7	7.8e-1	2.5e-2
50	7.0	9.3e-1	3.1e-2
60	9.7	1.1	3.8e-2
70	1.3e+1	1.2	4.4e-2
80	1.6e+1	1.3	5.0e-2
90	2.0e+1	1.5	5.6e-2
100	2.4e+1	1.6	6.3e-2

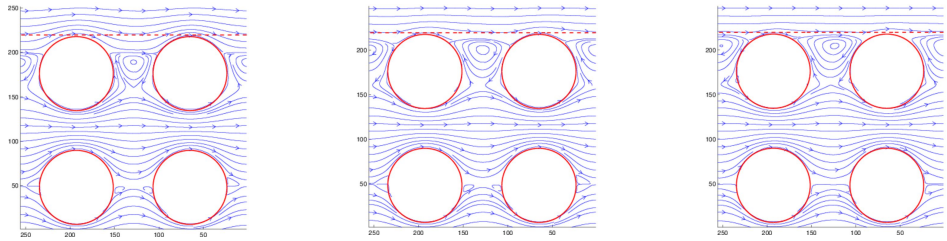


Re=10 *Re=50* *Re=100*

FIG. 9.10 – *Validity of the interface condition, configuration C2*

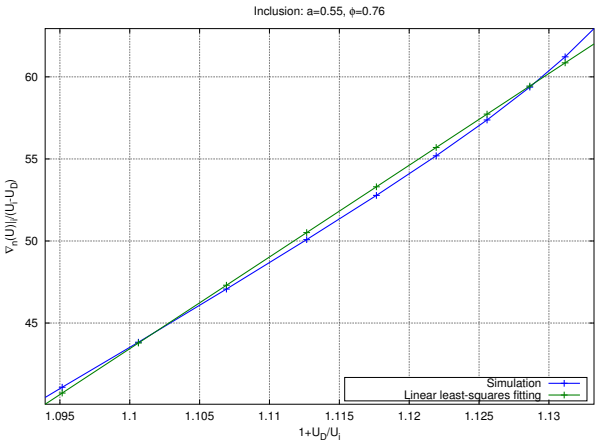


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	3.4e-3	2.6e-2	1.4e-3
10	3.3e-1	2.5e-1	1.4e-2
20	1.3	4.8e-1	2.8e-2
30	2.7	6.8e-1	4.3e-2
40	4.5	8.7e-1	5.7e-2
50	6.7	1.0	7.1e-2
60	9.3	1.2	8.5e-2
70	1.2e+1	1.4	9.9e-2
80	1.6e+1	1.5	1.1e-1
90	1.9e+1	1.6	1.3e-1
100	2.3e+1	1.8	1.4e-1

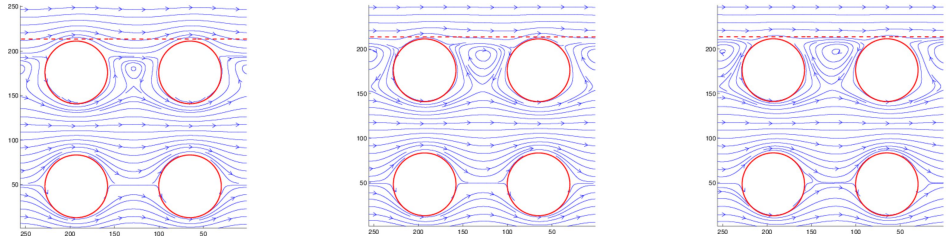


Re=10 *Re=50* *Re=100*

FIG. 9.11 – *Validity of the interface condition, configuration C3*

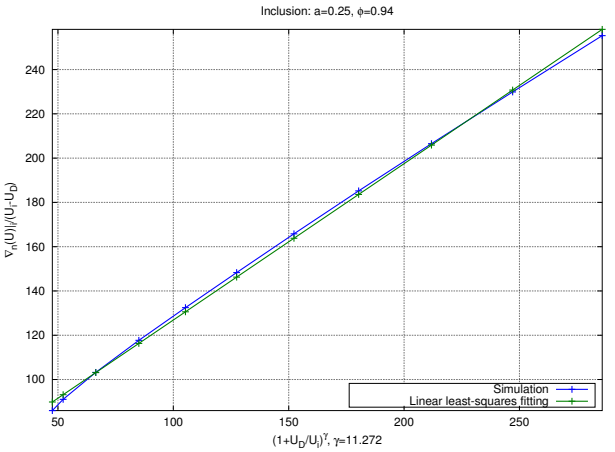


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	3.1e-3	2.8e-2	2.6e-3
10	3.1e-1	2.8e-1	2.6e-2
20	1.2	5.2e-1	5.3e-2
30	2.4	7.4e-1	7.9e-2
40	4.1	9.4e-1	1.1e-1
50	6.2	1.1	1.3e-1
60	8.5	1.3	1.6e-1
70	1.1e+1	1.5	1.8e-1
80	1.4e+1	1.6	2.1e-1
90	1.8e+1	1.8	2.3e-1
100	2.1e+1	1.9	2.6e-1

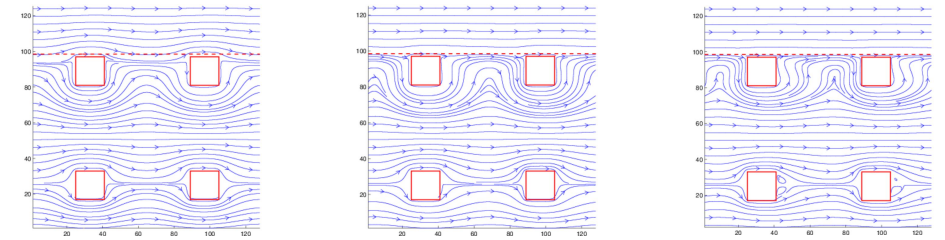


Re=10 *Re=50* *Re=100*

FIG. 9.12 – *Validity of the interface condition, configuration C4*

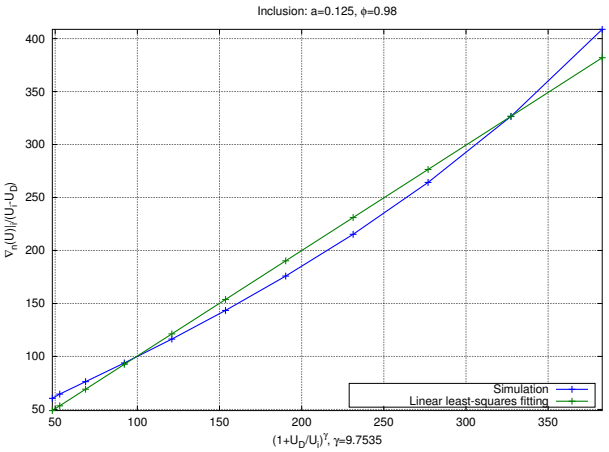


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	2.1e-3	2.1e-2	8.6e-3
10	2.0e-1	2.0e-1	8.5e-2
20	7.5e-1	3.8e-1	1.7e-1
30	1.6	5.2e-1	2.5e-1
40	2.6	6.5e-1	3.3e-1
50	3.9	7.7e-1	4.1e-1
60	5.2	8.7e-1	4.9e-1
70	6.7	9.6e-1	5.6e-1
80	8.3	1.0	6.3e-1
90	10	1.1	7.0e-1
100	12	1.2	7.7e-1

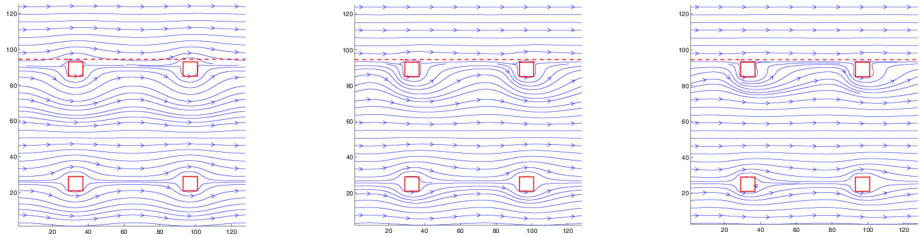


Re=10 *Re=50* *Re=100*

FIG. 9.13 – *Validity of the interface condition, configuration S4*

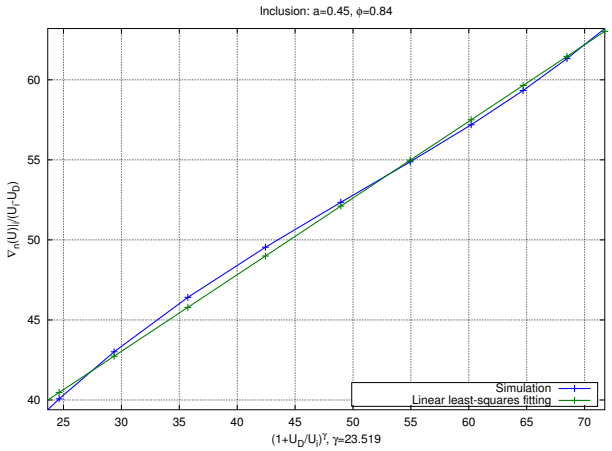


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	1.7e-3	3.5e-2	1.7e-2
10	1.7e-1	3.3e-1	1.7e-1
20	6.1e-1	6.1e-1	3.3e-1
30	1.2	8.3e-1	4.9e-1
40	2.0	1.0	6.4e-1
50	2.9	1.1	7.7e-1
60	3.8	1.3	9.0e-1
70	4.8	1.4	1.0
80	5.9	1.5	1.1
90	7.0	1.6	1.3
100	8.2	1.6	1.4

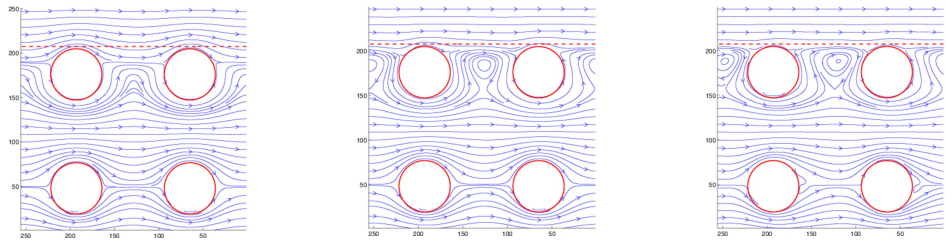


Re=10 *Re=50* *Re=100*

FIG. 9.14 – *Validity of the interface condition, configuration S5*

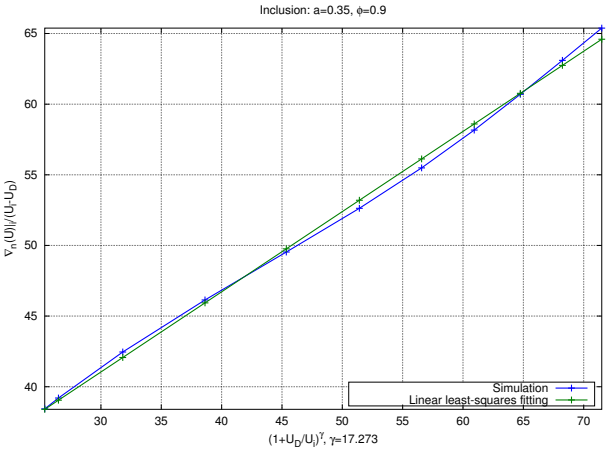


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	2.8e-3	3.1e-2	4.4e-3
10	2.7e-1	3.0e-1	4.4e-2
20	1.0	5.7e-1	8.9e-2
30	2.2	8.1e-1	1.3e-1
40	3.7	1.0	1.8e-1
50	5.5	1.2	2.2e-1
60	7.6	1.4	2.6e-1
70	1.0e+1	1.6	3.0e-1
80	1.3e+1	1.8	3.4e-1
90	1.6e+1	1.9	3.8e-1
100	1.9e+1	2.1	4.2e-1

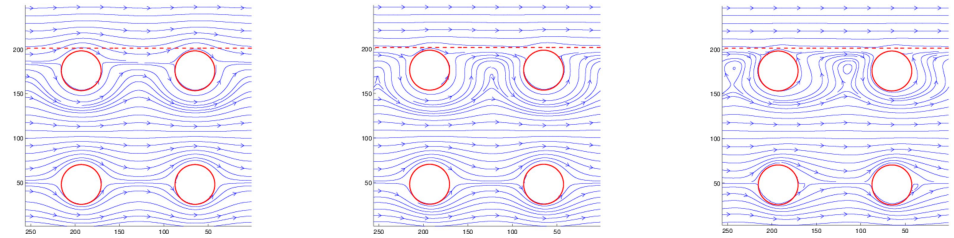


Re=10 *Re=50* *Re=100*

FIG. 9.15 – *Validity of the interface condition, configuration C5*

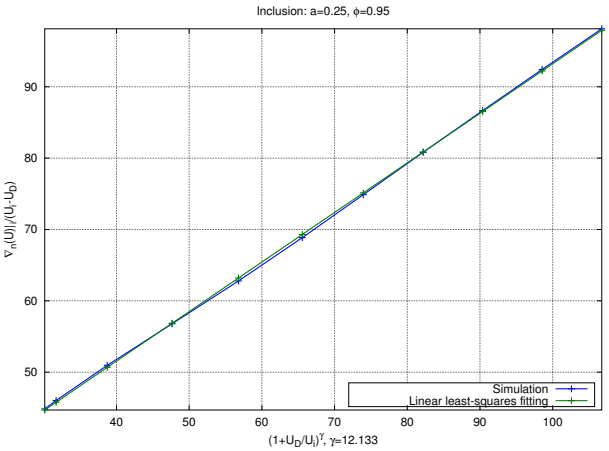


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	2.4e-3	3.4e-2	7.1e-3
10	2.4e-1	3.4e-1	7.1e-2
20	8.9e-1	6.4e-1	1.4e-1
30	1.9	9.0e-1	2.1e-1
40	3.2	1.1	2.8e-1
50	4.7	1.4	3.5e-1
60	6.5	1.6	4.1e-1
70	8.6	1.7	4.7e-1
80	1.1e+1	1.9	5.3e-1
90	1.3e+1	2.1	5.8e-1
100	1.6e+1	2.3	6.4e-1

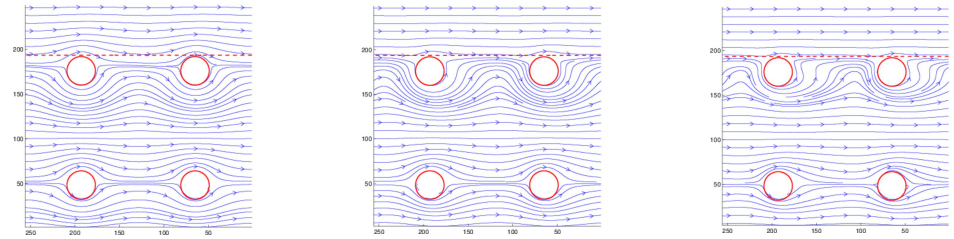


Re=10 *Re=50* *Re=100*

FIG. 9.16 – *Validity of the interface condition, configuration C6*

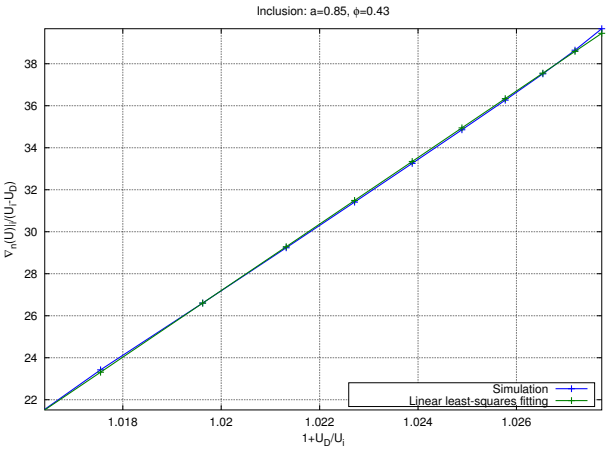


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	1.7e-3	3.5e-2	1.1e-2
10	1.7e-1	3.4e-1	1.1e-1
20	6.4e-1	6.4e-1	2.3e-1
30	1.3	9.0e-1	3.4e-1
40	2.2	1.1	4.4e-1
50	3.3	1.3	5.4e-1
60	4.5	1.5	6.3e-1
70	5.8	1.7	7.2e-1
80	7.2	1.8	8.1e-1
90	8.7	1.9	8.9e-1
100	1.0e+1	2.1	9.7e-1

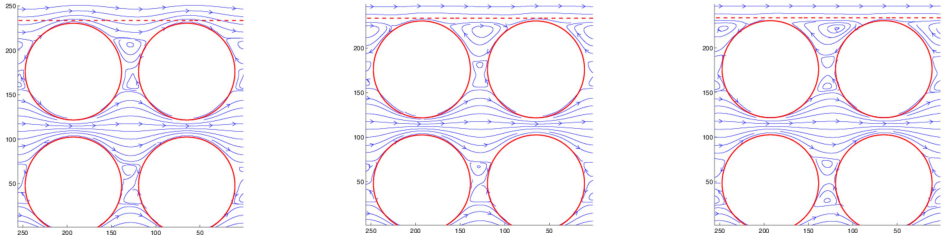


Re=10 *Re=50* *Re=100*

FIG. 9.17 – *Validity of the interface condition, configuration C7*

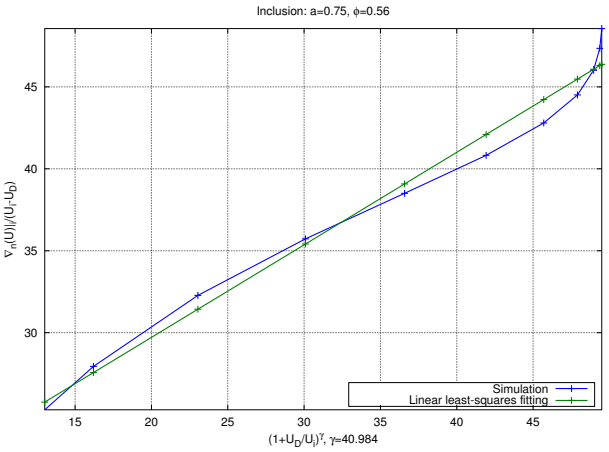


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	2.9e-2	6.7e-2	1.1e-3
10	2.7	6.2e-1	1.1e-2
20	9.5	1.1	2.2e-2
30	2.0e+1	1.5	3.3e-2
40	3.3e+1	1.9	4.4e-2
50	4.9e+1	2.3	5.5e-2
60	6.7e+1	2.6	6.5e-2
70	8.8e+1	3.0	7.6e-2
80	1.1e+2	3.3	8.7e-2
90	1.4e+2	3.6	9.7e-2
100	1.6e+2	3.9	1.1e-1

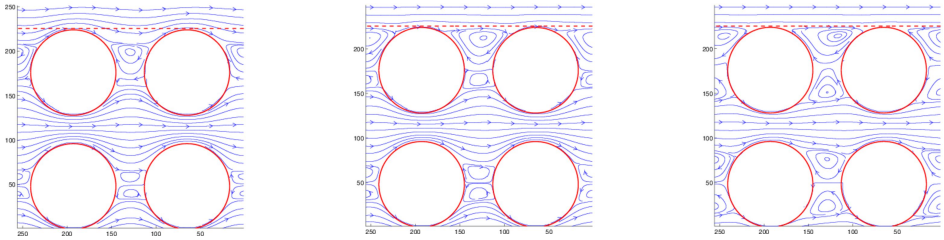


Re=10 *Re=50* *Re=100*

FIG. 9.18 – *Validity of the interface condition, configuration T1*

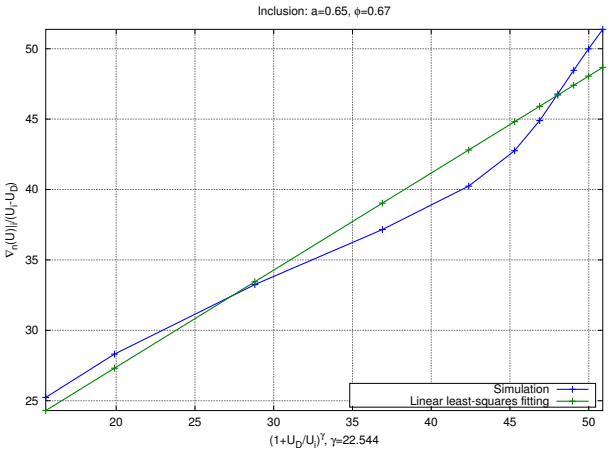


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	2.3e-2	6.2e-2	4.0e-3
10	2.1	5.7e-1	4.0e-2
20	7.6	1.0	8.0e-2
30	1.6e+1	1.4	1.2e-1
40	2.6e+1	1.7	1.6e-1
50	3.9e+1	2.1	2.0e-1
60	5.3e+1	2.4	2.3e-1
70	7.0e+1	2.7	2.6e-1
80	8.9e+1	3.0	2.9e-1
90	1.1e+2	3.2	3.2e-1
100	1.3e+2	3.5	3.5e-1

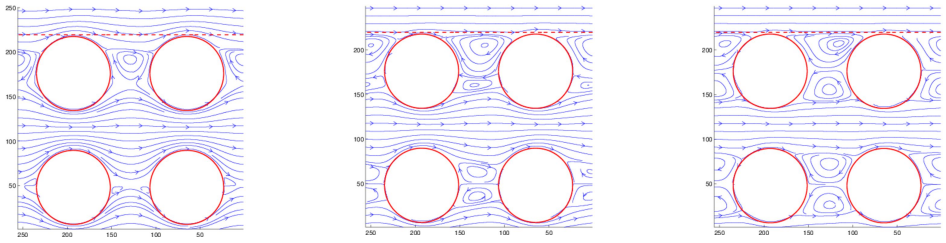


Re=10 *Re=50* *Re=100*

FIG. 9.19 – *Validity of the interface condition, configuration T2*

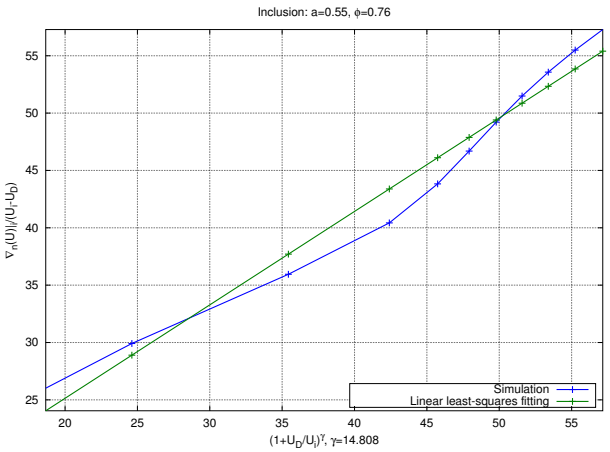


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	2.2e-2	6.9e-2	8.9e-3
10	2.0	6.2e-1	8.9e-2
20	7.1	1.1	1.8e-1
30	1.5e+1	1.5	2.6e-1
40	2.4e+1	1.9	3.4e-1
50	3.6e+1	2.2	4.1e-1
60	5.0e+1	2.6	4.8e-1
70	6.5e+1	2.9	5.4e-1
80	8.3e+1	3.2	6.0e-1
90	1.0e+2	3.5	6.6e-1
100	1.2e+2	3.7	7.1e-1

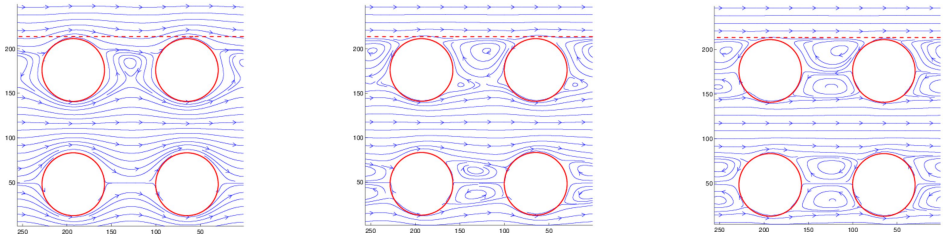


Re=10 *Re=50* *Re=100*

FIG. 9.20 – *Validity of the interface condition, configuration T3*

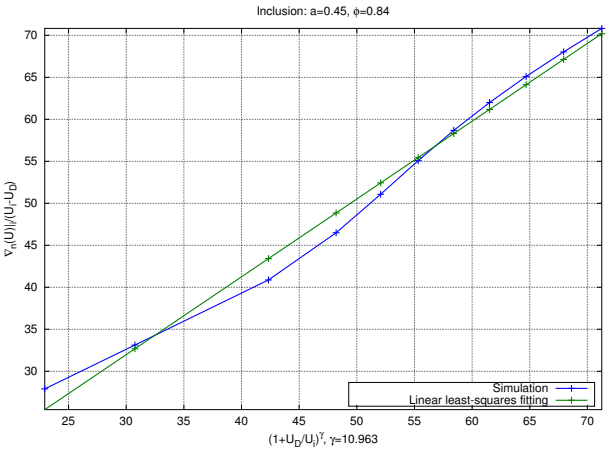


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	2.1e-2	7.6e-2	1.7e-2
10	1.9	6.8e-1	1.6e-1
20	6.6	1.2	3.3e-1
30	1.4e+1	1.6	4.7e-1
40	2.2e+1	2.0	6.0e-1
50	3.3e+1	2.4	7.2e-1
60	4.6e+1	2.8	8.4e-1
70	6.0e+1	3.1	9.5e-1
80	7.6e+1	3.4	1.1
90	9.3e+1	3.7	1.2
100	1.1e+2	4.0	1.3

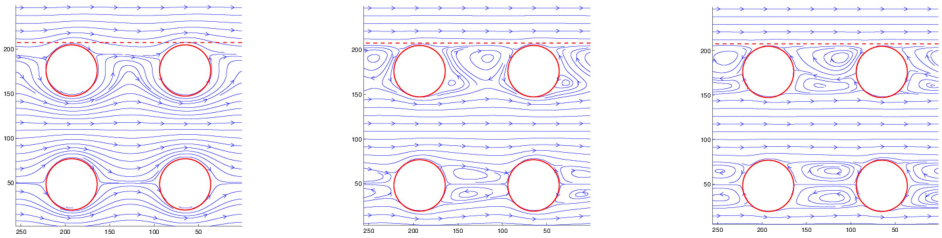


Re=10 *Re=50* *Re=100*

FIG. 9.21 – *Validity of the interface condition, configuration T4*

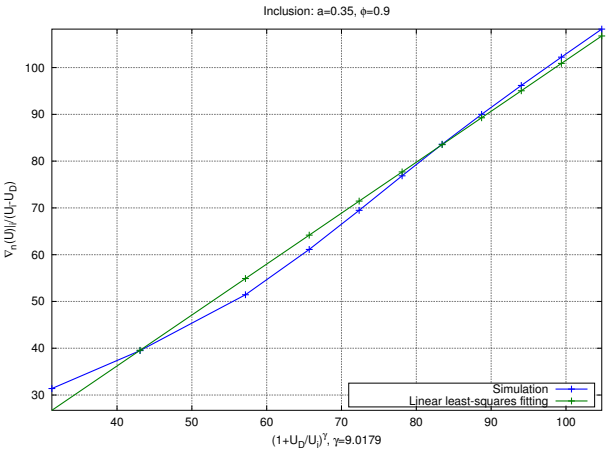


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	1.9e-2	8.4e-2	2.8e-2
10	1.7	7.5e-1	2.8e-1
20	5.9	1.3	5.4e-1
30	1.2e+1	1.8	7.6e-1
40	2.0e+1	2.2	9.7e-1
50	3.0e+1	2.6	1.2
60	4.1e+1	3.0	1.4
70	5.3e+1	3.4	1.5
80	6.7e+1	3.7	1.7
90	8.2e+1	4.0	1.9
100	9.8e+1	4.3	2.1

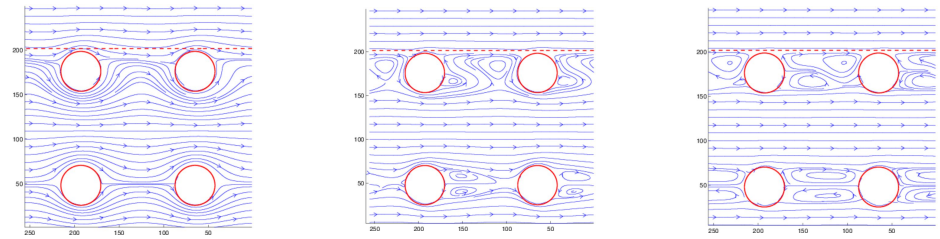


Re=10 *Re=50* *Re=100*

FIG. 9.22 – *Validity of the interface condition, configuration T5*

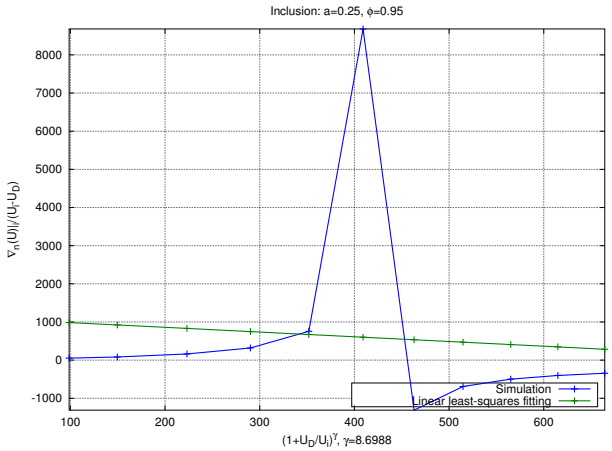


Re	Re_i	$\langle u_x \rangle_x$	U_p
1	1.7e-2	9.5e-2	4.4e-2
10	1.5	8.4e-1	4.4e-1
20	5.1	1.5	8.2e-1
30	1.0e+1	2.0	1.2
40	1.7e+1	2.4	1.5
50	2.5e+1	2.8	1.8
60	3.4e+1	3.2	2.0
70	4.4e+1	3.6	2.3
80	5.6e+1	4.0	2.6
90	6.8e+1	4.3	2.9
100	8.2e+1	4.7	3.2

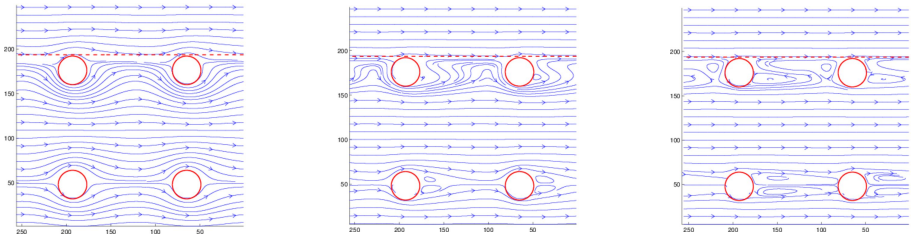


Re=10 *Re=50* *Re=100*

FIG. 9.23 – *Validity of the interface condition, configuration T6*



Re	Re_i	$\langle u_x \rangle_x$	U_p
1	1.3e-2	1.0e-1	7.2e-2
10	1.1	9.0e-1	7.0e-1
20	3.7	1.5	1.3
30	7.3	2.0	1.8
40	1.2e+1	2.4	2.3
50	1.7e+1	2.7	2.7
60	2.3e+1	3.1	3.2
70	3.0e+1	3.4	3.6
80	3.8e+1	3.8	4.1
90	4.6e+1	4.1	4.5
100	5.5e+1	4.4	4.9



Re=10 *Re=50* *Re=100*

FIG. 9.24 – *Validity of the interface condition, configuration T7*

ϕ	$\mathbf{u_D/u_i}$	**Conf.**	$\boldsymbol{\alpha_{nl}/(2\sqrt{K})}$	γ
0.23	1.4×10^{-2}	S1	3.7×10^{4}	1
0.44	7.6×10^{-2}	S2	5.1×10^{3}	1
0.75	2.8×10^{-1}	S3	9.6×10^{2}	1
0.43	8.0×10^{-3}	C1	6.4×10^{3}	1
0.56	3.3×10^{-2}	C2	1.8×10^{3}	1
0.67	6.7×10^{-2}	C3	9.0×10^{2}	1
0.76	1.2×10^{-1}	C4	5.6×10^{2}	1
0.94	5.3×10^{-1}	S4	7.1×10^{-1}	11.3
0.98	6.7×10^{-1}	S5	1.0	9.8
0.84	1.8×10^{-1}	C5	4.8×10^{-1}	23.5
0.90	2.5×10^{-1}	C6	5.7×10^{-1}	17.2
0.95	4.0×10^{-1}	C7	6.9×10^{-1}	12.1

TAB. 9.2 – *Numerical parameters of the non-linear interface condition.*

CONCLUSION

Results

In the course of this thesis, a cell-centered pressure correction scheme and its extension to a multigrid-AMR solver is presented. A numerical code was developed from scratch so as to test extensively this numerical scheme. Finally a fluid porous interface model for convective flow regimes is proposed and assessed with direct simulations.

In the first part, an original cell-centered pressure-correction scheme for solving the compressible Navier-Stokes and Euler equations is presented. This scheme is intended to deal with all Mach flows. The well-posedness and the convergence of the scheme are proved for the compressible Euler equations.

The scheme is then tested in highly compressible regimes with strong shocks and compared with another pressure-correction scheme, "SLK". Excellent agreement is found with the literature for both schemes. Shocks are sharply calculated and possible numerical oscillations at shocks are addressed with adaptive artificial diffusion. Our scheme is found more expensive than SLK due to the non-linear projection step but it has the distinctive advantage of using an internal energy balance instead of a total energy balance.

These properties have straightforward extension to two-phase flow models. The scheme was thus modified to deal with a simplified model of the compressible two-phase flow model of GENEPI. A comparison was carried out with GENEPI, which yield the conclusion that on selected tests our numerical scheme is able to compute the exact steady state as with GENEPI. Moreover, our scheme is also able to compute the full transient regime, which is not the case with GENEPI. In the context of the ongoing upgrade of GENEPI, this makes our scheme a serious asset for dealing directly with the models which are used at CEA both for low Mach flows and highly compressible flows.

An advantage of the cell-centered discretization is that it provides a natural framework for developing adaptive methods. The pressure-correction scheme introduced in the first part is extended to a multigrid-AMR solver. First a thorough study of grid generation algorithms is carried out and several improvement are proposed over the original clustering algorithm: specific criteria for controlling the shape of the patches and more importantly a technique for avoiding at all times adaptive grids incompatible with selected composite discretizations.

Then several aspects of the design of the multigrid-AMR extension of the projection scheme are discussed. At fine-fine interfaces, the classical ghost-cell synchronization procedure is interpreted with domain decomposition concepts. The efficiency of relaxation methods on levels grids is investigated, providing a better understanding of the impact of grid partitioning. Moreover this gives an insight on the relevant parameters to be set for level relaxation of the targeted multigrid solver. Then a multigrid-AMR algorithm is presented, and integrated to there reso-

lution of the compressible Euler equations with our projection scheme.

Finally extensive numerical tests are performed to assess the multigrid-AMR projection scheme. The projection scheme naturally handles low Mach flows, hence a particular focus on highly compressible flows in these tests. The 2D Riemann problem reveals the presence of spurious pressure modes when corners between successive levels are too close. Next a standard benchmark for the Double Mach Reflection problem is studied. The numerical results are analyzed quantitatively and qualitatively using up-to-date knowledge of the DMR problem. Despite issues of numerical noise and low accuracy (first order), the adaptive projection scheme performs very well and the most relevant features of DMR are verified.

The last part of this work is devoted to fluid-porous interface modelling at the macroscopic scale. The aim would be to ultimately derive an interface condition suitable with the complex compressible two-phase flows of GENEPI. However even the extension of classical interface conditions to Navier-Stokes regime is yet an open problem. In this context we propose an interface condition derived from a kinetic energy balance intended for coupling the incompressible Navier-Stokes equations with the Forchheimer equation. Direct simulations give excellent agreement with our model except for very high porosities.

Perspectives

A second order formulation of the projection algorithm using a modified MUSCL scheme is ongoing. This would yield more interesting results with adaptive mesh refinement as it has a dramatic impact on the increase of resolution as further levels of refinement are added or as the refinement factor is increased.

As for the multigrid-AMR method, the resolution of the non-linear projection-correction step with Newton-Multigrid or with the full approximation scheme would make the scheme much more robust. Regarding fine-fine interfaces for level relaxation, it would be interesting to compare the smoothing efficiency of the ghost cell synchronization with advanced substructuring methods such as the Robin-Robin method. The algorithm of resolution would save significant computational efforts if time refinement was supported.

The transient regime for two phase flows obtained with our projection scheme needs further investigation through new numerical tests. More realistic models including viscous and non-viscous effects would allow to use the scheme on more realistic qualification benchmarks. In addition the case of highly compressible two-phase flows is yet to be addressed.

Lastly, the fluid-porous interface condition introduced in part III is a first step towards the mathematical analysis of the coupled Navier-Stokes/Forchheimer problem.

BIBLIOGRAPHY

[1] M. Adams, P. Colella, D. T. Graves, J. N. Johnson, N. D. Keen, T. J. Ligocki, D. F. Martin, P. McCorquodale, D. Modiano, P. Schwartz, T. Sternberg, and B. V. Straalen, *Chombo Software Package for AMR Applications – Design Document*, Tech. Rep. LBNL-6616E, Lawrence Berkeley National Laboratory, 2012.

[2] M. J. Aftosmis, *Solution adaptive cartesian grid methods for aerodynamic flows with complex geometries*, VKI Lecture Series, 2 (1997).

[3] V. I. Agoshkov, *Poincaré-Steklov's operators and domain decomposition methods in finite dimensional spaces*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM Philadelphia, 1988, pp. 73–112.

[4] T. Alazard, *A minicourse on the low Mach number limit*, Discrete and Continuous Dynamical Systems-Series S, 1 (2008), pp. 365–404.

[5] G. Allaire, *Analyse numérique et optimisation*, Editions Ecole Polytechnique, 2005.

[6] A. Almgren, T. Buttke, and P. Colella, *A Fast Adaptive Vortex Method in Three Dimensions*, Journal of Computational Physics, 113 (1994), pp. 117–200.

[7] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, *A Conservative Adaptive Projection Method for the Variable Density Incompressible Navier-Stokes Equations*, Journal of Computational Physics, 142 (1998), pp. 1–46.

[8] A. S. Almgren, T. Buttke, and P. Colella, *A fast adaptive vortex method in three dimensions*, J. Comput. Phys., 113 (1994), pp. 177–200.

[9] P. Angot, *Analysis of Singular Perturbations on the Brinkman Problem for Fictitious Domain Models of Viscous Flows*, Mathematical Methods in the Applied Sciences, 22 (1999), pp. 1395–1412.

[10] ——, *On the well-posed coupling between free fluid and porous viscous flows*, Applied Mathematics Letters, 24 (2011), pp. 803–810.

[11] ——, *On the unsteady Stokes problem with a nonlinear open artificial boundary condition modelling a singular load*, (in preparation), (2013).

[12] F. Archambeau, J.-M. Hérard, and J. Laviéville, *Comparative study of pressure-correction and Godunov-type schemes on unsteady compressible cases*, Computers & Fluids, 38 (2009), pp. 1495–1509.

[13] E. Arquis and J.-P. Caltagirone, *Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide-milieu poreux application la convection naturelle*, C. R. Acad. Sc. Paris – Série II, 299 (1984), pp. 1–4.

[14] B. Smith and P. Bjorstad and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.

[15] D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice-Hall, 1982.

[16] M. Barad and P. Colella, *A fourth-order accurate local refinement method for Poisson's equation*, Journal of Computational Physics, 209 (2005), pp. 1–18.

[17] G. S. Beavers and D. D. Joseph, *Boundary conditions at a naturally permeable wall*, Journal of Fluid Mechanics, 30 (1967), pp. 197–207.

[18] G. S. Beavers, E. M. Sparrow, and B. A. Masha, *Boundary Condition at a Porous Surface Which Bounds a Fluid Flow*, AIChE Journal, 20 (1974), pp. 596–597.

[19] M. Belliard and S. Casaburi, *Mémento des modélisations physiques et numériques pour la simulation numérique d'un échangeur de type REP*, Tech. Rep. 96/022, Commissariat à l'Energie Atomique, CEA/DEN/DEC/SECA/LTEA, 1997.

[20] F. Ben Belgacem and Y. Maday, *The mortar element method for three dimensional finite elements*, M2AN, 31 (1997), pp. 289–302.

[21] G. Ben-Dor, *Shock Wave Reflection Phenomena*, Springer, 2007.

[22] G. Ben-Dor and I. I. Glass, *Domains and boundaries of non-stationary oblique shock-wave reflexions – 1. Diatomic gas*, Journal of Fluid Mechanics, 92 (1979), pp. 459–496.

[23] ——, *Domains and boundaries of non-stationary oblique shock-wave reflexions – 2. Monoatomic gas*, Journal of Fluid Mechanics, 96 (1980), pp. 735–756.

[24] M. Berger and I. Rigoutsos, *An algorithm for point clustering and grid generation*, IEEE Transactions on Systems, Man and Cybernetics, 21 (1991), pp. 1278–1286.

[25] M. J. Berger and P. Colella, *Local adaptive mesh refinement for shock hydrodynamics*, Journal of Computational Physics, 82 (1989), pp. 64–84.

[26] M. J. Berger and J. Oliger, *Adaptive mesh refinement for hyperbolic partial differential equations*, Journal of Computational Physics, 53 (1984), pp. 484–512.

[27] C. Bernardi, Y. Maday, and A. Patera, *Domain decomposition by the mortar element method*, in Asymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters, H. Kaper, M. Garbey, and G. Pieper, eds., vol. 384 of NATO ASI Series, Springer Netherlands, 1993, pp. 269–286.

[28] P. E. Bjørstad and O. B. Widlund, *Iterative methods for the solution of elliptic problems on regions partitioned into substructures*, SIAM Journal on Numerical Analysis, 23 (1986), pp. 1097–1120.

[29] J. Bonet and J. Peraire, *An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems*, International Journal for Numerical Methods in Engineering, 31 (1991), pp. 1–17.

[30] J. Bourgat, R. Glowinski, P. L. Tallec, and M. Vidrascu, *Variational formulation and algorithm for trace operator in domain decomposition calculations*, in Second International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, 1988, pp. 3–16.

[31] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Mathematics of Computation, 31 (1977), pp. 333–390.

[32] F. Brezzi and J. Pitkäranta, *On the stabilization of finite element approximations of the stokes equations*, in Efficient Solutions of Elliptic Systems, W. Hackbusch, ed., vol. 10 of Notes on Numerical Fluid Mechanics, Vieweg+Teubner Verlag, 1984, pp. 11–19.

[33] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial: Second Edition*, SIAM, 2000.

[34] A. Cadiou, L. Le Penven, and M. Buffat, *Asymptotic and numerical analysis of an inviscid bounded vortex flow at low Mach number*, Journal of Computational Physics, 227 (2008), pp. 8268–8280.

[35] Y. Cao, M. Gunzburger, F. Hua, and X. Wang, *Coupled Stokes-Darcy model with Beavers-Joseph interface boundary condition*, Communications in Mathematical Science, 8 (2010), pp. 1–25.

[36] R. Cautrès, R. Herbin, and F. Hubert, *The Lions domain decomposition algorithm on non-matching cell-centred finite volume meshes*, IMA J. Numer. Anal., 24 (2004), pp. 465–490.

[37] M. Chandesris and D. Jamet, *Boundary conditions at a planar fluid–porous interface for a Poiseuille flow*, International Journal of Heat and Mass Transfer, 49 (2006), pp. 2137–2150.

[38] ——, *Boundary conditions at a fluid–porous interface: An a priori estimation of the stress jump coefficients*, International Journal of Heat and Mass Transfer, 50 (2007), pp. 3422–3436.

[39] J. Cheissoux, J. Haquet, M. Grandotto, M. Bernard, and E. de Langre, *Spécifications physiques du code GENEPI (Générateurs de vapeur des REP)*, Tech. Rep. 86/756, Commissariat à l'Energie Atomique, CEA/DEN/DRE/STRE/LGV, 1986.

[40] J. L. Cheissoux, T. Delorme, and P. Obry, *Tables polynomiales des propriétés thermophysiques des matériaux et des fluides dans le code Genepi (version 1)*, Tech. Rep. 90/1005, Commissariat à l'Energie Atomique, CEA/DEN/DTE/STRE/LGV, 1991.

[41] H. Childs, E. S. Brugger, K. S. Bonnell, J. S. Meredith, M. Miller, B. J. Whitlock, and N. Max, *A contract-based system for large data visualization*, in Proceedings of IEEE Visualization 2005, 2005, pp. 190–198.

[42] A. J. Chorin, *A numerical method for solving incompressible viscous flow problems*, Journal of Computational Physics, 2 (1967), pp. 12–26.

[43] ——, *Numerical Solution of the Navier-Stokes Equations*, Mathematics of Computation, 22 (1968), pp. 745–762.

[44] W. L. Coirier, *Simulation of steady viscous flow on an adaptively refined cartesian grid*, PhD thesis, University of Michigan, 1994.

[45] P. Colella, *Multidimensional upwind for hyperbolic conservation laws*, Journal of Computational Physics, 87 (1990), pp. 171–200.

[46] P. Colella and L. F. Henderson, *The von Neumann paradox for the diffraction of weak shock waves*, Journal of Fluid Mechanics, 213 (1990), pp. 71–94.

[47] P. Colella and K. Pao, *A Projection Method for Low Speed Flows*, Journal of Computational Physics, 149 (1998), pp. 245–269.

[48] P. Degond and M. Tang, *All speed scheme for the low Mach number limit of the Isentropic Euler equation*, Communications in Computational Physics, 10 (2011), pp. 1–31.

[49] K. Deimling, *Nonlinear Functional Analysis*, Springer-Verlag, 1985.

[50] R. Deiterding, *Block-structured Adaptive Mesh Refinement Methods for Conservation Laws – Using the SAMR approach for elliptic and parabolic problems.* Multi-resolution Summer School, Fréjus, 2010.

[51] ——, *Block-structured adaptive mesh refinement — theory, implementation and application*, ESAIM Proc., 34 (2011), pp. 97–151.

[52] C. C. Douglas, *A review of numerous parallel multigrid methods*, SIAM News, 25 (1992), pp. 14–15.

[53] S. J. Dwyer, R. W. McLaren, and C. A. Harlow, *Computer-Aided Diagnosis of Breast Cancer from Thermography*, in Pattern Recognition and Artificial Intelligence, C. H. Chen, ed., Academic Press, New York, 1976, pp. 233–247.

[54] R. Eymard. Personal communication.

[55] R. Eymard and T. G. ;, *H-convergence and numerical schemes for elliptic equations*, SIAM Journal on Numerical Analysis, 41 (2000), pp. 539–562.

[56] R. Eymard, T. Gallouët, and R. Herbin, *The finite volume method*, North Holland, 2000.

[57] R. Eymard, T. Gallouët, and R. Herbin, *Convergence analysis of a colocated finite volume scheme for the incompressible Navier-Stokes equations on general 2 or 3D meshes* , SIAM Journal on Numerical Analysis, 30 (2007), pp. 1–36.

[58] ——, *Discretisation of heterogeneous and anisotropic diffusion problems on general non-conforming meshes. SUSHI: a scheme using stabilisation and hybrid interfaces*, SIAM Journal on Numerical Analysis, 30 (2010), pp. 1009–1043.

[59] R. Eymard, R. Herbin, and J.-C. Latché, *Convergence analysis of a colocated finite volume scheme for the incompressible Navier-Stokes equations on general 2D or 3D meshes*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 1–36.

[60] R. Eymard, R. Herbin, J.-C. Latché, and B. Piar, *Convergence analysis of a locally stabilized collocated finite volume scheme for incompressible flows*, M2AN, 43 (2009), pp. 889–927.

[61] S. Faure, J. Laminie, and R. Temam, *Colocated finite volume schemes for fluid flows*, Communications in Computational Physics, 4 (2008), pp. 1–25.

[62] R. P. Fedorenko, *A relaxation method for solving elliptic difference equations*, USSR Computational Mathematics and Mathematical Physics, 1 (1962), pp. 1092–1096.

[63] ——, *The speed of convergence of one iterative process*, USSR Computational Mathematics and Mathematical Physics, 4 (1964), pp. 227–235.

[64] M. Gander, *Schwarz methods over the course of time*, Electronic Transactions on Numerical Analysis, 31 (2008), pp. 228–255.

[65] L. Gastaldo, R. Herbin, and J.-C. Latché, *A discretization of the phase mass balance in fractional step algorithms for the drift-flux model*, IMA Journal of Numerical Analysis, 31 (2011), pp. 116–146.

[66] ——, *A discretization of the phase mass balance in fractional step algorithms for the drift-flux model*, IMA Journal of Numerical Analysis, 31 (2011), pp. 116–146.

[67] L. Gastaldo, R. Herbin, J.-C. Latché, and N. Therme, *A consistency result for explicit staggered schemes for the Euler equations*, (submitted), (2014).

[68] S. Glasstone and P. J. Dolan, *The effects of nuclear weapons*, U.S. Department of Defense; U.S. Department of Energy, 1977, pp. 86–126.

[69] H. M. Glaz, P. Colella, I. I. Glass, and R. L. Deschambault, *A numerical study of oblique shock-wave reflections with experimental comparisons*, Proceedings of the Royal Society of London A – Mathematical and Physical Sciences, 398 (1985), pp. 117–140.

[70] B. Goyeau, D. Lhuillier, D. Gobin, and M. G. Velarde, *Momentum transport at a fluid–porous interface*, International Journal of Heat and Mass Transfer, 46 (2003), pp. 4071–4081.

[71] D. Grapsas, W. Kheriji, R. Herbin, and J.-C. Latché., *An unconditionally stable finite-element-finite volume pressure correction scheme for the compressible Navier-Stokes equations*, submitted, (2014).

[72] J. L. Guermond, P. Minev, and J. Shen, *An overview of projection methods for incompressible flows*, Computer Methods in Applied Mechanics and Engineering, 195 (2006), pp. 6011–6045.

[73] H. Guillard and C. Viozat, *On the behaviour of upwind schemes in the low Mach number limit*, Computers & Fluids, 28 (1999), pp. 63–86.

[74] F. H. Harlow and A. A. Amsden, *Numerical calculation of almost incompressible flow*, Journal of Computational Physics, 3 (1968), pp. 80–93.

[75] L. F. Henderson, E. I. Vasilev, G. Ben-Dor, and T. Elperin, *The wall-jetting effect in Mach reflection: theoretical consideration and numerical investigation*, Journal of Fluid Mechanics, 479 (2003), pp. 259–286.

[76] R. Herbin, W. Kheriji, and J.-C. Latché, *On some implicit and semi-implicit staggered schemes for the shallow water and euler equations*, ESAIM: Mathematical Modelling and Numerical Analysis, 48 (2014), pp. 1807–1857.

[77] R. Herbin, W. Kheriji, and J.-C. Latché, *On some implicit and semi-implicit staggered schemes for the shallow water and Euler equations*, M2AN, 48 (2014), pp. 1807–1857.

[78] H. Hornung, *Regular and Mach reflection of shock waves*, Annual Review of Fluid Mechanics, 18 (1986), pp. 33–58.

[79] H. G. Hornung, H. Oertel, and R. J. Sandeman, *Transition to Mach reflexion of shock waves in steady and pseudosteady flow with and without relaxation*, Journal of Fluid Mechanics, 90 (1979), pp. 541–560.

[80] W. Jäger and A. Mikelić, *On the interface boundary conditions by Beavers, Joseph and Saffman*, SIAM Journal on Applied Mathematics, 60 (2000), pp. 1111–1127.

[81] ——, *Modeling Effective Interface Laws for Transport Phenomena Between an Unconfined Fluid and a Porous Medium Using Homogenization*, Transport in Porous Media, 78 (2009), pp. 489–508.

[82] D. Jamet, M. Chandesris, and B. Goyeau, *On the Equivalence of the Discontinuous One- and Two-Domain Approaches for the Modeling of Transport Phenomena at a Fluid/Porous Interface* , Transport in Porous Media, 78 (2009), pp. 403–418.

[83] D. M. Jones, P. M. E. Martin, and C. K. Thornhill, *A Note on the Pseudo-Stationary Flow behind a Strong Shock Diffracted or Reflected at a Corner*, Proceedings of the Royal Society of London A, 209 (1951), pp. 238–248.

[84] M. Karasabun, *An experimental apparatus to study nucleate pool boiling of R-114 and oil mixtures*, PhD thesis, U.S. Navy Naval Postgraduate School, Monterey, California, 1984.

[85] K. C. Karki and S. V. Patankar, *Pressure based calculation procedure for viscous flows at all speeds in arbitrary configurations*, AIAA Journal, 27 (1989), pp. 1167–1174.

[86] M. Kaviany, *Principles of Heat Transfer in Porous Media*, Mechanical Engineering Series, Springer, 1995.

[87] W. Kheriji, R. Herbin, and J.-C. Latché, *Pressure correction staggered schemes for barotropic one-phase and two-phase flows*, Comput. & Fluids, 88 (2013), pp. 524–542.

[88] A. M. Khokhlov, *Fully threaded tree algorithms for adaptive refinement fluid dynamics simulations*, Journal of Computational Physics, 143 (1998), pp. 519–543.

[89] P. Krehl and M. van der Geest, *The discovery of the Mach reflection effect and its demonstration in an auditorium*, Shock Waves, 1 (1991), pp. 3–15.

[90] A. Kurganov and Y. Liu, *New adaptive artificial viscosity method for hyperbolic systems of conservation laws*, Journal of Computational Physics, 231 (2012), pp. 8114–8132.

[91] O. A. Ladyzhenskaya, *The mathematical theory of viscous incompressible flow*, Gordon and Breach, second ed., 1963, pp. 23–31.

[92] L. D. Landau and E. M. Lifshitz, *Course of Theoretical Physics, Volume 6: Fluid Mechanics*, Pergamon Press, 1959.

[93] C. K. Law, *Diffraction of strong shock waves by a sharp compressive corner*, Tech. Rep. AFOSR 70-0767 TR, University of Toronto/IAS, 1970.

[94] P. D. Lax and X.-D. Liu, *Solution of two-dimensional Riemann problems of gas dynamics by positive schemes*, SIAM Journal on Scientific Computing, 19 (1998), pp. 319–340.

[95] H. Li and G. Ben-Dor, *Reconsideration of pseudo-steady shock wave reflections and the transition criteria between them*, Shock Waves, 5 (1995), pp. 59–73.

[96] Q. Liu and A. Prosperetti, *Pressure-driven flow in a channel with porous walls*, Journal of Fluid Mechanics, 679 (2011), pp. 77–100.

[97] F. Losasso, R. Fedkiw, and S. Osher, *Spatially adaptive techniques for level set methods and incompressible flow*, Computers & Fluids, 35 (2006), pp. 995–1010.

[98] A. Marciniak-Czochra and A. Mikelić, *A nonlinear effective slip interface law for transport phenomena between a fracture flow and a porous medium*, Discrete and Continuous Dynamical Systems – Series S, 7 (2014), pp. 1065–1077.

[99] D. Marr and E. Hildren, *Theory of edge detection*, Proceedings of the Royal Society of London, Series B, Biological Sciences, 207 (1980), pp. 187–217.

[100] D. Martin and K. Cartwright, *Solving poisson's equation using adaptive mesh refinement*, Tech. Rep. UCB/ERL M96/66, EECS Department, University of California, Berkeley, 1996.

[101] D. F. Martin and P. Colella, *A Cell-Centered Adaptive Projection Method for the Incompressible Euler Equations*, Journal of Computational Physics, 163 (2000), pp. 271–312.

[102] P. Mathon, F. Archambeau, and J.-M. Hérard, *Implantation d'un algorithme compressible dans Code_Saturne*, Tech. Rep. HI-83/03/016/A, EDF R&D / MF2E, 2004.

[103] D. A. Nield, *The limitations of the Brinkman-Forchheimer equation in modeling flow in a saturated porous medium and at an interface*, International Journal of Heat and Fluid Flow, 12 (1991), pp. 269–272.

[104] P. Obry and J.-L. Cheissoux, *Etablissement des équations homogénéisées de conservation (système à trois et quatre équations) du logiciel GENEPI (système TRIO)*, Tech. Rep. 90/1014, Commissariat à l'Energie Atomique, CEA/DEN/DTE/STRE/LGV, 1990.

[105] J. A. Ochoa-Tapia and S. Whitaker, *Momentum transfer at the boundary between a porous medium and a homogeneous fluid – I. Theoretical development*, International Journal of Heat and Mass Transfer, 38 (1995), pp. 2635–2646.

[106] ——, *Momentum transfer at the boundary between a porous medium and a homogeneous fluid – II. Comparison with experiment*, International Journal of Heat and Mass Transfer, 38 (1995), pp. 2647–2655.

[107] G. S. H. Pau, J. B. Bell, A. S. Almgren, K. M. Fagnan, and M. J. Lijewski, *An Adaptive Mesh Refinement Algorithm for Compressible Two-Phase Flow In Porous Media*, Computational Geosciences, 16 (2012), pp. 577–592.

[108] D. W. Peaceman and H. H. Rachford, *The numerical solution of parabolic and elliptic differential equations*, Journal of the Society for Industrial & Applied Mathematics, 3 (1955), pp. 28–41.

[109] R. Plessier, *Comparaison des résultats du logiciel Genepi version 1 à des solutions analytiques*, Tech. Rep. 91/006, Commissariat à l'Energie Atomique, CEA/DEN/DER/SCC/LTDE, 1991.

[110] S. Popinet, *Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries*, Journal of Computational Physics, 190 (2003), pp. 572–600.

[111] J. S. Przemieniecki, *Theory of Matrix Structural Analysis*, Dover Civil and Mechanical Engineering, Dover, 1985.

[112] A. Quarteroni and A. Valli, *Theory and Application of Steklov-Poincaré Operators for Boundary-Value Problems*, in Applied and Industrial Mathematics, R. Spigler, ed., vol. 56 of Mathematics and Its Applications, Springer Netherlands, 1991, pp. 179–203.

[113] ——, *Theory and application of Steklov-Poincaré operators for boundary-value problems. The heterogenous operator case*, in Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, 1991, pp. 58–81.

[114] ——, *Domain Decomposition Methods for Partial Differential Equations*, Clarendon Press, 1999.

[115] C. M. Rhie and W. L. Chow, *Numerical study of the turbulent flow past an airfoil with trailing edge separation*, AIAA, 21 (1983), pp. 1525–1532.

[116] P. G. Saffman, *On the boundary condition at the interface of a porous medium*, Studies Applied Mathematics, 1 (1971), pp. 93–101.

[117] M. Sahraoui and M. Kaviany, *Slip and no-slip velocity boundary conditions at interface of porous, plain media*, International Journal of Heat and Mass Transfer, 35 (1992), pp. 927–943.

[118] A. Samake, S. Bertoluzza, M. Pennacchio, C. Prud'homme, and C. Zaza, *A Parallel Implementation of the Mortar Element Method in 2D and 3D*, ESAIM: Proc., 43 (2013), pp. 213–224.

[119] H. A. Schwarz, *Über einen Grenzübergang durch alternierendes Verfahren*, Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich, 15 (1870), pp. 272–286.

[120] D. Serre, *Chapter 2: Shock reflection in gas dynamics*, vol. 4 of Handbook of Mathematical Fluid Dynamics, North-Holland, 2007, pp. 39–122.

[121] R. V. Southwell, *Stress-calculation in frameworks by the method of" systematic relaxation of constraints". i and ii*, Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences, 151 (1935), pp. 56–95.

[122] ——, *Relaxation Methods in Theoretical Physics*, Clarendon Press, Oxford, 1946.

[123] R. Temam, *Sur l'approximation de la solution des quations de Navier-Stokes par la mthode des pas fractionnaires I*, Archive for Rational Mechanics and Analysis, 32 (1969), pp. 135–153.

[124] ——, *Sur l'approximation de la solution des quations de Navier-Stokes par la mthode des pas fractionnaires II*, Archive for Rational Mechanics and Analysis, 33 (1969), pp. 377–385.

[125] N. Therme and C. Zaza, *Comparison of cell-centered and staggered pressure-correction schemes for all-mach flows*, in Finite Volumes for Complex Applications VII-Elliptic, Parabolic and Hyperbolic Problems, vol. 78 of Springer Proceedings in Mathematics & Statistics, Springer International Publishing, 2014, pp. 975–983.

[126] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, Springer-Verlag, 1999.

[127] F. J. Valdés-Parada, J. Alvarez-Ramírez, B. Goyeau, and J. A. Ochoa-Tapia, *Computation of Jump Coefficients for Momentum Transfer Between a Porous Medium and a Fluid Using a Closed Generalized Transfer Equation*, Transport in Porous Media, 78 (2009), pp. 439–457.

[128] G. Volpe, *Performance of compressible flow codes at low Mach numbers*, AIAA Journal, 31 (1993), pp. 49–56.

[129] D. R. White, *An experimental survey of the Mach reflection of shock waves*, Tech. Rep. II-10, Princeton University Department of Physics, 1951.

[130] P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, Journal of Computational Physics, 54 (1984), pp. 115–173.

[131] D. P. Young, R. G. Melvin, M. B. Bieterman, F. T. Johnson, and S. S. Samant, *A locally refined rectangular grid finite element method: application to computational fluid dynamics and computational physics*, Journal of Computational Physics, 92 (1991), pp. 1–66.

[132] R. K. Zeytounian, *Asymptotic Modelling of Fluid Flow Phenomena*, Springer, 2002.

[133] Q. Zhang, *High-order, multidimensional, and conservative coarse-fine interpolation for adaptive mesh refinement*, Computer Methods in Applied Mechanics and Engineering, 200 (2011), pp. 3159–3168.