université
PARIS-SACLAY

Improved numerical schemes for monotonic conservative scalar advection: tackling mesh imprinting and numerical wetting

Schémas numériques améliorés pour l'advection scalaire, monotone et conservative : réduction d'empreinte de maillage et de mouillage numérique

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 574, mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques appliquées
Unité de recherche : Université Paris-Saclay, CNRS, ENS Paris-Saclay, Centre Borelli, 91190, Gif-sur-Yvette, France
Référent : ENS Paris-Saclay

Thèse présentée et soutenue à Paris-Saclay, le 04/05/2021, par

Christina PAULIN

Composition du jury :

Prof. Frédéric Pascal ENS Paris-Saclay, Centre Borelli, France	Président
Prof. Philippe Helluy Université de Strasbourg, IRMA, France	Rapporteur
Prof. Knut-Andreas Lie SINTEF, Oslo, Norway	Rapporteur
Dr. Robin James Richard Williams AWE, Aldermaston, UK	Examinateur
Dr. William J. Rider Sandia, Albuquerque, NM, USA	Examinateur

Direction de la thèse :

Dr. Antoine Llor CEA, DAM/DIF	Directeur
Dr. Thibaud Vazquez-Gonzalez CEA, DAM/DIF	Co-encadrant

Thèse de doctorat

NNT:2021UPASM011

Wie alles sich zum Ganzen webt,
Eins in dem andern wirkt und lebt!
Wie Himmelskräfte auf und nieder steigen...

Johann Wolfgang Goethe

Remerciements

I would like to thank all members of the jury, for your detailed reading and for all of your interesting questions and remarks.

To everyone I met at conferences: whether we chatted at a coffee break, sat together at lunch, had diner afterwards, a Whiskey in Glasgow, a glass of wine in Trento, a Bosnian coffee (and lokum) in Sarajevo, not only did you help me with getting better at English, but you were there with your great kindness, advice, insight, cultural explanations and all sorts of discussions. Hopefully we will meet again!
À l'équipe de travail: Antoine Llor merci d'avoir proposé ce sujet de thèse. Thibaud, sans toi le rendu du manuscrit n'aurait pas été possible, Rémi, pour nos voyages en conférence, Renaud, tu m'a appris comment faire des bonnes présentations. Merci à Jean-Philippe, Thierry, Guillaume, Sébastien, William, Bertrand, Marie et Mathieu pour toutes vos remarques et suggestions. Et n'oublions pas les collègues de Bordeaux que j'ai eu l'occasion de rencontrer en conférence, Isabelle, Nicolas, Simon. Merci aussi à Jean-Philippe pour le stage d'avant thèse et notre première publication ensemble avec Renaud.

À l'équipe du centre Borelli (ex CMLA) merci pour l'acceuil charmant dès le premier jour du stage. À John, Alexandre et Lei.

Aux doctorants et anciens de Ter@tec : Nestor (tu connais toujours une solution, merci pour le partage de tes idées !), Théo, Hoby, Guillaume, Hugo et Hugo, Arthur, Sébastien, Simon, Florian, Anton, Van Man, Alexiane, Julie, Théo, Mathieu.
À Éloïse, une amie précieuse, merci pour tout ! À Adrien et Dani merci pour les défis sportifs. À Brigitte, le rayon de soleil. À Guillaume, c'était très motivant de faire le trajet ensemble.
Aux doctorants/postdocs que j'ai eu l'occasion de rencontrer (ou mieux connaître) dans ma dernière année de thèse: Mathilde, Albertine, Louise, Sandra, Cécile, Corentin, Éric, Jean-Cédric, Adrien, Baptiste, Bastien, Ronan, Olivier, Victor, Grégoire et Charles.
À mes autres amis: Jon j'ai beaucoup apprécié nos sorties cinéma et les balades,

Perrine, Ana, Claire, Estelle, Sébastien, Robin, Martin, Constant, Antoine, Guillaume, Antoine on peut toujours compter sur vous, pour des plans de loisir ou d'autres objectifs.

À ma famille en France, merci Christine, Frédéric et Amandine de m'avoir encouragé tout le long de mes études. A mini familie idr Schwyz, merci vöu mau dass dir immer a mi glaubt heit und Daniela mini Lieblingsschwöschter, i freue mi uf jede Bsuech vo dir.

A aui mini Fründe us dr Schwyz: e bsundere Dank ad Adilla, Stephan, Silkanny, Andrea, Thomas, a Ufmunterig hets im letschte Johr auso nid gfählt, und au e bsundere Dank ad Lilly, wo immer drfür sorgt dass mir ufem Laufende blibe.

Und zum Schluss no a mi Ma, dr Mattis, mi privat cuisinier, guet dass aber mängisch d schwyzer Chuchi au erlaubsch.

Contents

Acknowledgement **2**

Introduction **7**
0.1 Advection in hydrodynamics, aerodynamics and meteorology 7
0.2 Quasi-isotropic second-order advection for the GEEC formalism . . 9
0.2.1 Towards quasi-isotropic numerical diffusion 10
0.2.2 1D Slope-And-Bound (SAB): clipping procedure for 1D gradients 11
0.2.3 N-Dimensional-Slope-And-Bound (ND-SAB) 12

I Towards isotropic transport with co-meshes **14**

1 Towards isotropic transport with co-meshes **15**
1.1 Introduction 15
1.2 Generic form and properties of the discrete Eulerian transport operator 16
1.3 Co-mesh approach in 2D 17
1.3.1 Construction of the co-mesh 18
1.3.2 General method 18
1.3.3 Reducing anisotropy 22
1.4 Conclusion 23
1.5 Appendix 24
1.5.1 First-order development and consistency conditions 24
1.5.2 Modified equation applied to the first-order transport scheme with co-meshes 26
1.5.3 Anisotropy and CFL 28

II Doubly-monotonic constraint on interpolators: bridging second-order to singularity preservation to cancel "nu-

merical wetting" in transport schemes 30

2 Doubly-monotonic constraint on interpolators: bridging second-order to singularity preservation to cancel "numerical wetting" in transport schemes 31
2.1 Introduction . . . 31
2.1.1 Motivations: "numerical wetting" . . . 31
2.1.2 Numeric background: MUSCL-like scheme . . . 33
2.1.3 Present approach: from limiting to clipping of interpolants. . . 35
2.1.4 . . . and from simple to double monotonicity . . . 36
2.1.5 Mapping of interpolants and interpolators, 1D numerical tests 37
2.2 A convenient mapping for bounded monotonic functions . . . 38
2.2.1 Motivation: bounded monotonic functions as interpolants . . 38
2.2.2 L^2 bounds on bounded monotonic functions . . . 38
2.2.3 Elliptic coordinates mapping of $\mathcal{M}_{XY}(\overline{y})$ . . . 41
2.3 Slope-and-bounds interpolants . . . 42
2.3.1 Background: usual single-slope interpolants . . . 42
2.3.2 Slope-and-bounds interpolants . . . 44
2.4 Doubly-monotonic SAB interpolators . . . 46
2.4.1 Motivation: from interpolant to interpolator . . . 46
2.4.2 Doubly monotonic interpolators . . . 46
2.4.3 Doubly-monotonic second-order SAB interpolators . . . 47
2.4.4 Mapping of doubly-monotonic-constrained SAB interpolators 50
2.5 Numerical experiments . . . 50
2.5.1 General test conditions . . . 50
2.5.2 Square-wave tests: singularities, numerical wetting . . . 55
2.5.3 Sine-wave tests: erosion, steepening, terracing . . . 60
2.5.4 Zalesak-wave tests: sharp peaks, . . . 61
2.6 Conclusions . . . 63
2.7 Appendix . . . 64
2.8 Second-order MC slope on non-uniform mesh . . . 64
2.9 Front width of an error function profile . . . 64
2.10 Convergence study of MC and SB on a Gaussian pulse . . . 65

III Adapting Slope-And-Bound interpolation to multi-dimensional advection schemes to cancel "numerical wetting" 67

3 Adapting Slope-And-Bound interpolation to multidimensional advection schemes to cancel "numerical wetting" 68

3.1 Introduction . . . 68
3.2 Multidimensional Slope-And-Bound reconstruction algorithm . . . 72
3.2.1 Linear reconstruction of gradients with weighted least squares 74
3.2.2 Monotonicity analysis in multi-D to 1D reduction . . . 76
3.2.3 Nonlinear reconstruction by Slope-And-Bound limitation in the 1D reference frame . . . 80
3.2.4 Multidimensional upwind fluxes . . . 84
3.2.5 Algorithm . . . 86
3.3 Results . . . 87
3.3.1 Multidimensional Slope-And-Bound reconstruction algorithm 90
3.3.2 Combining Slope-And-Bound and co-mesh approach . . . 91
3.4 Conclusions . . . 97
3.5 Appendix . . . 99
3.5.1 Weighting factors deduced from gradients of a parabola . . . 99
3.5.2 Solving linear system (3.27) . . . 100

Conclusion **104**

List of Figures **127**

List of published works **128**

À l'intention du lecteur francophone **130**
Introduction . . . 130
Conclusion . . . 137

Introduction

0.1 Advection in hydrodynamics, aerodynamics and meteorology

Before the era of the first computers John von Neumann, Robert D. Richtmyer, Theodore von Kármán, and Lewis Fry Richardson shaped history as mathematical pioneers in the fields of nuclear engineering, aerospace engineering, and meteorology [1, 2, 3]. The beginning of the first computer simulations in the 1950s marks a milestone, and in the same decade Francis H. Harlow counts as the pioneer of Computational Fluid Dynamics (CFD)[4, 5, 6].

CFD codes distinguish Eulerian and Lagrangian formalisms. The Eulerian approach observes the fluid movement on a fixed grid, whereas in the Lagrangian approach the mesh moves with the fluid velocity. The theory of the motion of fluids we know today is based on the work of Leonhard Euler [7], George Gabriel Stokes [8], and Joseph-Louis Lagrange [9]. The following paragraphs discuss a selection of recent advances in the fields of hydrodynamics, aerodynamics and meteorology.

In nuclear engineering Lagrangian descriptions were originally used. Lagrangian hydrodynamics is still an important part of most solvers, for instance in Inertial Confinement Fusion (ICF) [10, 11]. CAVEAT [12, 13], GLACE [14], and EUCCLHYD [15] are famous codes of Godunov-type schemes, that simulate laser fusion numerically. However, the intrinsic entropy production of the acoustic Riemann solvers lead to too much numerical dissipation. Different approaches to reduce this dissipation are discussed in [16], and in [17, 18] entropy production is reduced in rarefaction region with the *isentropic flux*. A staggered grid Godunov-like approach is introduced in [19]. Further Riemann solvers are discussed in [20] Since the introduction to Arbitrary Lagrangian Eulerian (ALE) [21] formalisms, the advantages of both methods are combined in most hydrocodes [22]. Notably Lagrangian approaches are well-suited to maintaining material interfaces, whereas Eulerian approaches are no subject to critical tangling of the underlying meshes. An overview of advection in ALE Finite Elements Methods (FEM) and compared to Finite Volumes Methods (FVM) is given in [23]. The present work focuses

on advection problems in the context similar to FVM. Advection describes the transport of physical quantities, notably of fluids. It is an important element in the discretization of the Euler and Navier-Stokes equations. An advection step is encountered in every CFD solver with its own challenges. The most important work of the discretization of advection steps in numerical solvers is discribed as Flux Reconstruction (FR). Fundamental techniques of FR are recapped in [24]. The goal of most reconstructions is to improve the space discretization of advection equations to high order, where spurious numerical oscillations are prohibited with limitation strategies. Limiters are distinguished in two categories: *slope limiters* limit the reconstructed gradient functions, and *flux limiters* limit the resulting fluxes. The present work focuses on slope limitation, for which an exhaustive bibliography is given in the second and third part. Flux limiters are not less important, for instance, Flux Corrected Transport FCT [25, 26] was introduced in nuclear engineering. FCT solves the continuity equation while maintaining non-negativity, and the FCT algorithm separates the transport step in two stages: a "convective stage", where the numerical fluxes are defined and a "antidiffusive (or corrective) stage", where as its name says the antidiffusive counter parts are computed, which are then limited. FCT represents an elaborate approach, where an antidiffusive formula is used to maintain positivity and the fluxes are carefully limited without violating conservation. FCT has been studied extensively over the last few decades, and applications to aerodynamics, meteorolgy and oceanography have been developed as well [27].

In aerospace engineering, the development of CFD codes boomed in the 1970s with advances in the aircraft industry [28]. Advection schemes have always been key to obtain high-resolution schemes. For instance, the relationship between *control-volume* type schemes and *fluctuation splitting* schemes is introduced in [29], and nonconstant advection speed is discussed with a generalized von Neumann analysis in [30]. The most recent flux limitation strategies in aerodynamic solvers are Residual Distribution (RD) [31, 32, 33, 34, 35], Actif Flux (AF) [36, 37, 38, 39] and the Multidimensional Limiting Process (MLP) [40, 41]. Furthermore, MUSCL (Monotonic Upstream-centered Scheme for Conservation Laws) techniques, originally introduced in [42], are important for aerodynamics. More specific references on MUSCL and other limitation strategies are given in the second and third part of the present manuscript.

In meteorology global Numerical Weather Prediction (NWP) systems, such as the Integrated Forecasting System (IFS), have been developped extensively in the recent years. IFS uses Non-oscillatory Forward-in-Time (NFT) Eulerian advection [43], and an advection step with the Multidimensional Positive Definite Advection Transport Algorithm (MPDATA) [44, 45], which is discretized with exact second-order. Before MPDATA advection in meteorology was usually based on

Crowley-type schemes [46, 47], which are dissipative advection schemes. Further papers from the author of MPDATA and colleagues should be mentioned as well, for their interesting ideas on advection: [48] discusses transport algorithms, that combine Eulerian and Semi-Lagrangian approaches. In [49] interesting thoughts on the *advection-interpolation equivalence*, and a class of monotonic interpolation schemes inspired by Tremback's Flux Corrected Transport (FCT) version [50] is deduced, where, however, the exact conservation constraint has been neglected. More recently mesh adaptivity and remapping have been discussed in MPDATA with unstructured meshes [51, 52].

0.2 Quasi-isotropic second-order advection for the GEEC formalism

The present study is connected to the multi-fluid quasi-symplectic Geometry, Energy, and Entropy Compatible (GEEC) formalism [53]. GEEC is inspired by the Conservative Space- and Time-Staggered (CSTS) scheme [54], where the discrete Euler equations are derived by mimicking a discrete action integral. This formalism leads to both an indirect ALE approach [55, 56], where a Lagrangian step is followed by a remapping phase referred to as "Lagrange-plus-remap", and a direct ALE approach [57], where the advection fluxes are directly taken into account. The advection steps in the GEEC discretization of the Euler equations use the discrete upwind advection operator $\mathrm{D}_{\Delta t}$, which is defined through explicit MUSCL fluxes for advected fields a and b as

$$\begin{aligned}&\int_V \int_t (a \mathrm{D}_t b) \mathrm{d}\boldsymbol{x}\mathrm{d}t \\ &\rightsquigarrow a_c^{n+1} \mathrm{D}_{\Delta t} b_c^n = a_c^{n+1} \Big[V_c^{n+1} b_c^{n+1} - V_c^n b_c^n + \Delta t^{n+\frac{1}{2}} \sum_{d \in \mathcal{D}(c)} \Big(\mathring{V}_{cd} b_c^n - \mathring{V}_{dc} b_d^n \Big) \Big], \end{aligned} \tag{1}$$

with volume transfer rate $\mathring{V}_{cd}$ and cell volume V_c, where c is the cell index and cd designates the oriented face index between cells c and d, and n the time index with time step Δt. Furthermore, $\mathcal{D}(c)$ indicates the set of neighboring cells of cell c in the stencil, where a finite volume point of view is considered, which means that neighbors have a face in common with cell c (this is detailed in chapter 1 in the context of *first nearest neighbors*).

The goal of the present work is to transform this first-order advection operator (1) to a quasi-isotropic second-order formulation, that optimizes the reduction of numerical wetting with an adequate limiter strategy. The modifications of the advection step developped in the present work must be compatible with the discrete conservation equations for mass, momentum and total energy of the GEEC solver, which derive from a discrete *least action principle* [53, §4] and write

$$\mathrm{D}_{\Delta t}\Big([\alpha\rho]_c^{\varphi n}\Big) = 0, \tag{2a}$$

$$\mathrm{D}_{\Delta t}\Big([\alpha\rho]_c^{\varphi n-1} u_c^{\varphi n-\frac{1}{2}}\Big) = \mathrm{RHS}, \tag{2b}$$

$$\mathrm{D}_{\Delta t}\Big([\alpha\rho]_c^{\varphi n} e_c^{\varphi n}\Big) = \mathrm{RHS}^{\star}, \tag{2c}$$

where φ is the fluid index, α_c^{φ} the volumic fraction of fluid φ present in cell c (with $\sum_{\varphi} \alpha^{\varphi} = 1$), for density ρ, fluid velocity $\boldsymbol{u}$, and internal energy e. Any further detail about the designations of the fluid fields on the right hand side is not necessary at the moment, the interested reader is refered to [53], where the complete algorithm can be found. The right hand side (RHS) of the discrete momentum equation (2b) describes the pressure gradient in GEEC, whereas the right hand side (RHS*) of the discrete total energy equation (2c) is more complex.

Compatible with GEEC and tested in the Eulerian mono-fluid representation, the present work shows two possibilities to transform the discrete upwind advection operator (1): (i) the co-mesh strategy, which reduces the anisotropy of the numerical diffusion produced by the grid orientation effect, is discussed in chapter 1; (ii) the Slope-And-Bound (SAB) reconstruction, which is introduced in 1D in chapter 2, and extended to a multidimensional formulation called ND-SAB in chapter 3. These two methods are compatible with each other and tests on this combination are shown in chapter 3, where the ND-SAB algorithm has been applied to the co-meshes as well.

0.2.1 Towards quasi-isotropic numerical diffusion

Advection is the central ingredient of all numerical schemes for hyperbolic partial differential equations and in particular for hydrodynamics. Advection has thus been extensively studied in many of its features and for numerous specific applications. In more than one dimension, it is most commonly plagued by a major artifact: mesh imprinting (illustrated in Figure 1), which is a numerical effect caused by the orientation of the velocity field with respect to the mesh. Though mesh imprinting is generally inevitable, its *anisotropy* can be modulated and is thus amenable to significant reduction.

A new definition of stencils is introduced by taking into account *second nearest neighbors* (across cell corners) and the resulting strategy is called "co-mesh approach". The modified equation is used to study numerical dissipation and tune enlarged stencils in order to minimize the anisotropy in advection steps.

The work described in chapter 1 was simultanously studied in the context of incompressible multiphase flow in porous media by [58, 59]. Acting on the same neighborhood in a Cartesian grid with first and second nearest neighbors, a

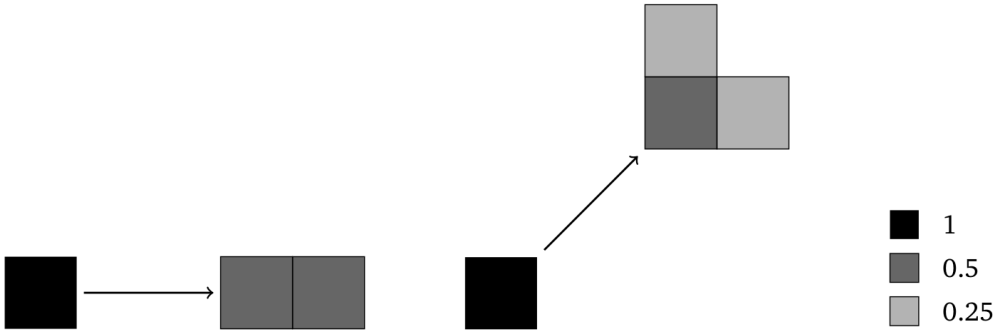


Figure 1: Mesh imprinting illustrated locally with 2D advection of one cell: the shape of the numerical solution is different for different velocity directions; left: velocity in direction of the x-axis; right: velocity in diagonal direction.

nine-point stencil is introduced for upwind fluxes and is analysed using Fourier analysis and a criteria is found to deduce an improved 9 point stencil, by means of the *least anisotropic behavior of the angular error*, as opposed to the present work, which studies the modified equation to derive a criterion in order to reduce anisotropy. Both strategies lead to new coefficients for optimal stencils including second nearest neighbors. [58, 59] were inspired by the work of [60] on the so-called Grid Orientation Effect (GOE). This effect was introduced in the 1980s in the framework of oil reservoir simulations [61, 62], and was recently studied in Godunov-type schemes [63, 64], and in the study of convergence of quadrilateral and triangular meshes [65, 66, 67].

Chapter 1 shows simulations on first-order advection on Cartesian grids, their co-meshes and the optimal linear combination of both. The modified equation has been studied in all of these cases and graphs illustrating the diffusive coefficients defined in some basis (by functions of the velocity directions) are shown next to the simulations. Furthermore, results of the co-mesh method on an arbitrarily deformed grid are shown. The co-mesh strategy succeeds in reducing mesh imprinting in all of these cases.

0.2.2 1D Slope-And-Bound (SAB): clipping procedure for 1D gradients

Monotonic interpolation and its avatars are major ingredients of many numerical schemes for solving partial differential equations (PDEs) under Total Variation Diminishing (TVD) or similar constraints. However, despite over forty years of extensive study of principles and applications, a key aspect of monotonic interpolant design can still appear somewhat empirical: how does a monotonic interpolator

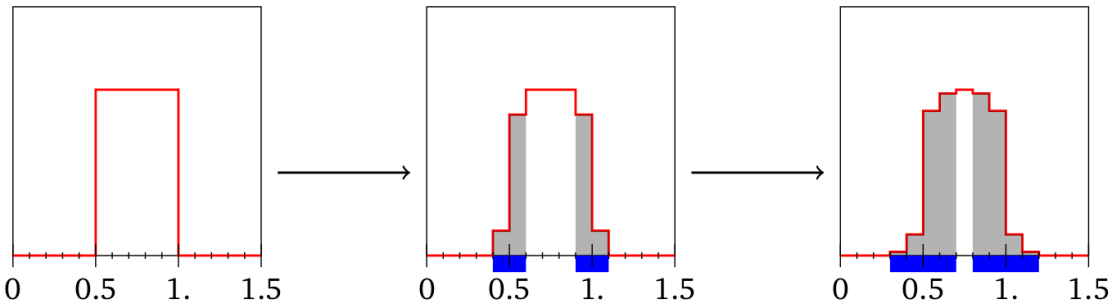


Figure 2: Illustration of "numerical wetting": increase of the support of numerical diffusion with the scheme's time step (cells of the support are highlighted in blue).

connect the limiting cases of smooth (differentiable) and singular (limited) functions in a consistent and possibly canonical way?

The present study aims at providing understanding in the basic but important case of per-cell monotonic one-dimensional scalar monotonic reconstruction and at applying it to second-order accurate advection. First, a general mapping of bounded monotonic functions in elliptic coordinates is built. Then, the usual "single-slope" second-order monotonic interpolants are continued into "slope-and-bound" monotonic interpolants. Finally, a critical constraint is introduced, the "double monotonicity," in order to build various slope-and-bound monotonic interpolators from this set of interpolants.

With these slope-and-bound interpolators, standard numerical tests show a complete cancellation of the "numerical wetting" (illustrated in Figure 2) that usual TVD advection schemes produce. When advecting scalar fields of compact support, this effect—not to be confused with usual numerical diffusion—is the low-level contamination that spreads linearly in time over all the regions of the computational domain where non-vanishing advection is present. To support this claim, chapter 2 compares the results of SAB limiters and compares them to their baseline slope-limiter functions on a square wave, a sine-wave and a multi-wave profile refered to as the Zalesak test case. Removal of numerical wetting is of particular importance in many industrial and academic applications, notably at "phase disappearance" episodes in multiphase flows or "wet–dry" transitions in shallow water flows. Improvement of the "numerical erosion" of extrema is also observed. The general principles exposed here can be extended to multi-dimensional settings, high-order schemes, and other PDEs.

0.2.3 N-Dimensional-Slope-And-Bound (ND-SAB)

In chapter 2, doubly monotonic [68] and Slope-And-Bound (SAB) interpolators are explored systematically in order to create very simple second-order limiters with 1D

upwind advection fluxes while attenuating numerical diffusion. A multidimensional extension called ND-SAB is introduced here as an alternative to classical MUSCL techniques: the multidimensional gradients are not directly limited, but reduced to a 1D representation, which is called the *1D reference frame*. This frame is used for limitation and for the computation of the total outflow of a cell. This total flux is then redistributed on the face fluxes.

The key element of the 1D SAB reconstruction is given with the so-called *second-order reference point*, which is the point given on the cell centroid of cell c that aligns the central values of the left and right neighboring cells. In uniform grids this second-order reference point simply corresponds to the mean of the neighboring values, whereas in non-uniform grids it is defined as a weighted mean, with weights given by the respective lengths of the neighboring cells. In the 1D reference frame of the present ND-SAB algorithm however, the sizes of the left and right neighbors are unknowns and the 1D SAB reconstruction cannot be applied directly. It is hence more difficult to define a fully multidimensional extension for the SAB strategy. In the present work, this is solved by introducing monotonicity thresholds in the 1D reference frame.

Furthermore, the ND-SAB method is compatible with the quasi-isotropic first-order advection introduced in [69]. Results are presented in chapter 3 to support this claim in 2D. Similar to the results obtained in chapter 2, numerical wetting is reduced to a compact support with the ND-SAB limitation strategy.

Part I

Towards isotropic transport with co-meshes

Chapter 1

Towards isotropic transport with co-meshes

1.1 Introduction

Various existing techniques for reducing the anisotropy of numerical transport resort to either of two strategies: (i) improve the order of accuracy of schemes, or (ii) make mesh and discretization stencil more isotropic. In the latter approach, one can mention Lagrange-remap schemes for hydrodynamics, where so-called corner fluxes appear [16, 21], face centered cubic (FCC) or body-centered cubic (BCC) lattices, often used for 3D wave propagation and linear magneto-hydrodynamics (MHD) [70, 71, 72], isotropic finite-differences, to correct lowest order error terms [73], interfacial flux splitting, to reduce mesh-locking effects for the heterogeneous, anisotropic diffusion problem [74], flux-corrected transport (FCT), which treats mesh-imprinting issues to achieve vorticity preservation [75], geometric correctors, to achieve consistency by constraining convergence to asymptotically regular meshes [66], etc.

Motivated by the development of hydro-codes for Inertial Confinement Fusion (ICF), [11], a novel multi-fluid multi-dimensional direct-ALE hydro-scheme approach was recently developed [57]. When deriving the scheme—designated as GEEC for Geometry, Energy and Entropy Compatible—a critical step appeared to be the definition of a proper discrete transport operator. In its present (first-order) form, its displays a significant anisotropic behavior that requires improvement for effective usage in applications.

The present work thus aims at studying and reducing the 2D anisotropy of the discrete first-order transport scheme. For this purpose, we privilege strategy (ii) above to improve isotropy before upgrading the transport operator to higher order. This is done with an enlarged first-order upwind stencil. Following strategy

(i) above would have introduced complexities in the quasi-symplectic design of the GEEC scheme due to corner fluxes without actually much improvement on anisotropy to second order.

This approach is inspired by the following quote from P. Roe: *"... respecting the correct propagation of information under all circumstances. This includes seeking modes of propagation that are isotropic when they should be."* [76].

1.2 Generic form and properties of the discrete Eulerian transport operator

The Eulerian transport operator for a field a under velocity field $\boldsymbol{u}$ writes

$$\mathrm{D}_t a = \partial_t a + \boldsymbol{\nabla} \cdot (a\boldsymbol{u}). \tag{1.1}$$

In the ALE context (Arbitrary Lagrangian Eulerian) $\boldsymbol{u}$ is the velocity in the reference frame, defined by the sum of relative-to-grid velocity $\boldsymbol{v}$ and grid velocity $\boldsymbol{w}$, $\boldsymbol{u} = \boldsymbol{v} + \boldsymbol{w}$. Remark that by definition $\boldsymbol{v} = 0$ represents Lagrangian transport by field $\boldsymbol{w}$, whereas $\boldsymbol{w} = 0$ represents Eulerian transport by field $\boldsymbol{v}$.

The generic first-order conservative discretization of the linear Eulerian transport operator (1.1) writes

$$\mathrm{D}_{\Delta t} a_c^n = V_c^{n+1} a_c^{n+1} - V_c^n a_c^n + \Delta t^n \sum_{d \in \mathcal{D}(c)} \left(a_c^n \mathring{V}_{cd}^n - a_d^n \mathring{V}_{dc}^n \right), \tag{1.2}$$

where the transported field a is defined at cell center x_c as its average value over (moving mesh) cell c of volume V_c—$\mathcal{D}(c)$ being the set of cell labels logically connected to cell c, as defined by the stencil. In order to preserve linearity with respect to velocity, the volume transfer rates $\mathring{V}_{cd}^n$ must be linear forms of relative-to-grid velocities $\boldsymbol{v}_q^n$—which are given at some nodes q related but not necessarily identical to the grid nodes—most generally represented by vectors $\boldsymbol{s}_{cdq}^n$

$$\mathring{V}_{cd}^n := \sum_{q \in \mathcal{Q}(c)} \boldsymbol{v}_q^n \cdot \boldsymbol{s}_{cdq}^n, \tag{1.3}$$

—$\mathcal{Q}(c)$ being the set of nodes q logically connected to cell c.

Elementary analysis of stability and consistency constrain the features of the transport operator (1.2) as follows: (i) for stability, transport must be upwinded with respect the velocity direction, i.e. $\boldsymbol{s}_{cdq}^n \cdot \boldsymbol{v}_q^n \geq 0$ in (1.3), and this makes $\mathring{V}_{cd}^n$ to be a piecewise linear function of the $\boldsymbol{v}_q^n$ or

$$\mathring{V}_{cd}^n := \sum_{q \in \mathcal{Q}(c)} \sigma_{cdq}^n \boldsymbol{v}_q^n \cdot \boldsymbol{s}_{cdq}^n, \qquad \text{where} \qquad \sigma_{cdq} := \mathrm{H}(\boldsymbol{v}_q \cdot \boldsymbol{s}_{cdq}), \tag{1.4}$$

H being the Heaviside function (H(0) = 1/2 is assumed); (ii) to avoid the DeBar artifact [77], velocities $\boldsymbol{v}_q$ must be collocated with transported field a_c, i.e. only one point $\boldsymbol{x}_q = \boldsymbol{x}_c$ is associated to any $\mathring{V}_{cd}^n$ in (1.3) and $\boldsymbol{s}_{cdq}$ reduces to $\boldsymbol{s}_{cd} := \boldsymbol{s}_{cdc}$

$$\mathring{V}_{cd}^n := \sigma_{cd}^n \boldsymbol{v}_c^n \cdot \boldsymbol{s}_{cd}^n; \tag{1.5}$$

(iii) consistency to first order with the continuous formulation (1.1) requires enforcing the following constraints (see Appendix 1.5.1)

$$\sum_{d \in \mathcal{D}(c)} (\sigma_{cd}^n \boldsymbol{s}_{cd}^n - \sigma_{dc}^n \boldsymbol{s}_{dc}^n) = \boldsymbol{0}, \tag{1.6a}$$

$$\sum_{d \in \mathcal{D}(c)} \sigma_{cd}^n \boldsymbol{s}_{cd}^n \otimes \delta \boldsymbol{x}_{cd}^n = V_c^n \, \mathbf{I}, \tag{1.6b}$$

where $\delta \boldsymbol{x}_{cd}^n := \boldsymbol{x}_d^n - \boldsymbol{x}_c^n$, and $\mathbf{I}$ is the identity matrix.

Condition (1.6a) is trivially ensured in a finite volumes setting where $\boldsymbol{s}_{cd}^n$ are the cell face vectors—normal to faces with magnitude given by face area—and if the the upwinding factors are consistent, that is if $\sigma_{cd}^n + \sigma_{dc}^n = 1$ for any couple cd. Under these conditions, (1.6a) reduces to the trivial identity

$$\sum_d \boldsymbol{s}_{cd}^n = 0. \tag{1.7}$$

Now, condition (1.6b) is far less trivial even in a finite element setting and is strongly dependent on the cell shapes and sizes. As visible from (1.19) in Appendix 1.5.1, the condition is fulfilled with a *uniform upwinding factor* and a *uniform Cartesian mesh* of squares or cubes in 2 or 3D. It is to be noted however, that conditions (1.6) are always *invariant* by both *affine transformations* and *convex linear combinations* of transport schemes.

The approach in the present work is to find the (possibly) best discretization to first order of the Eulerian transport operator within the framework defined by (1.2), (1.5), and (1.7), and complemented by (1.6b) whenever possible. It can be noticed that [57] used the same formalism on a structured (but non Cartesian and non uniform) mesh. This paper goes further by exploiting the freedom left in (1.6) to improve transport isotropy.

1.3 Co-mesh approach in 2D

Usual 2D Cartesian 5-point stencils of finite volume schemes only take into account first nearest neighbors (across cell faces). In the present work so-called *co-meshes* (as described in chapter 1.3.1) are introduced in order to deal with corner fluxes through second nearest neighbors (across cell corners).

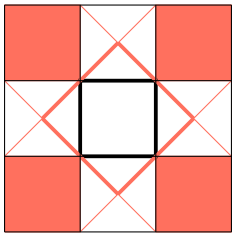


Figure 1.1: Cell with its first nearest neighbors (white) and its co-cell boundaries connected to second nearest neighbors (pink).

1.3.1 Construction of the co-mesh

The co-mesh represents a fictive grid that links second nearest neighbors (neighbors across cell corners in the initial mesh) through fictive cell faces (see Fig. 1.1). Notably, the co-mesh defines the vectors $\boldsymbol{s}_{cd}^{(2)}$ as its face normals, whereas $\boldsymbol{s}_{cd}^{(1)}$ are the face normals of the initial mesh. A cell of the co-mesh is called a *co-cell*. Each co-cell is built from the cell centers of the first nearest neighbors, where these cell centers act as the nodes of the co-mesh. This results in two co-meshes for a structured 2D grid, as shown in Fig 1.2. The main idea behind this construction is to build a mesh, on top of the initial one, which omits the numerical information of the first nearest neighbors. Fig. 1.3b illustrates the volume of the co-cells and how the omitted parts prevent the co-mesh from having "holes" in it, in order to result in a well-defined mesh. At the moment, the co-mesh strategy is applied only to quadrilateral (but not necessarily Cartesian) structured grids. It is not clear yet how this method will be adaptable to unstructured grids.

Let us remark at this point the importance of computing the cell volumes of the co-meshes $V_c^{(i)}$ exactly, in order to preserve conservation. Considering $V_c^{(2)} = 2V_c^{(1)}$ is of course true in the case of Cartesian meshes. However, this estimate is almost surely wrong in the case of more general meshes and violates conservation as illustrated in Fig. 1.6.

1.3.2 General method

The co-mesh approach consists in solving transport terms of a numerical scheme over an initial mesh and several related co-meshes (introduced in 1.3.1) and linearly combine the resulting schemes $\omega \times$ scheme $+ (1 - \omega) \times$ co-scheme with weight ω. This leads to a 9-point stencil on a fictive mesh as represented in Fig. 1.3.

The notation for the transport operator in (1.2) is not changed; only the neighborhood of cell c is redefined as

$$\mathcal{D}(c) = \mathcal{D}_1(c) \cup \mathcal{D}_2(c), \tag{1.8}$$

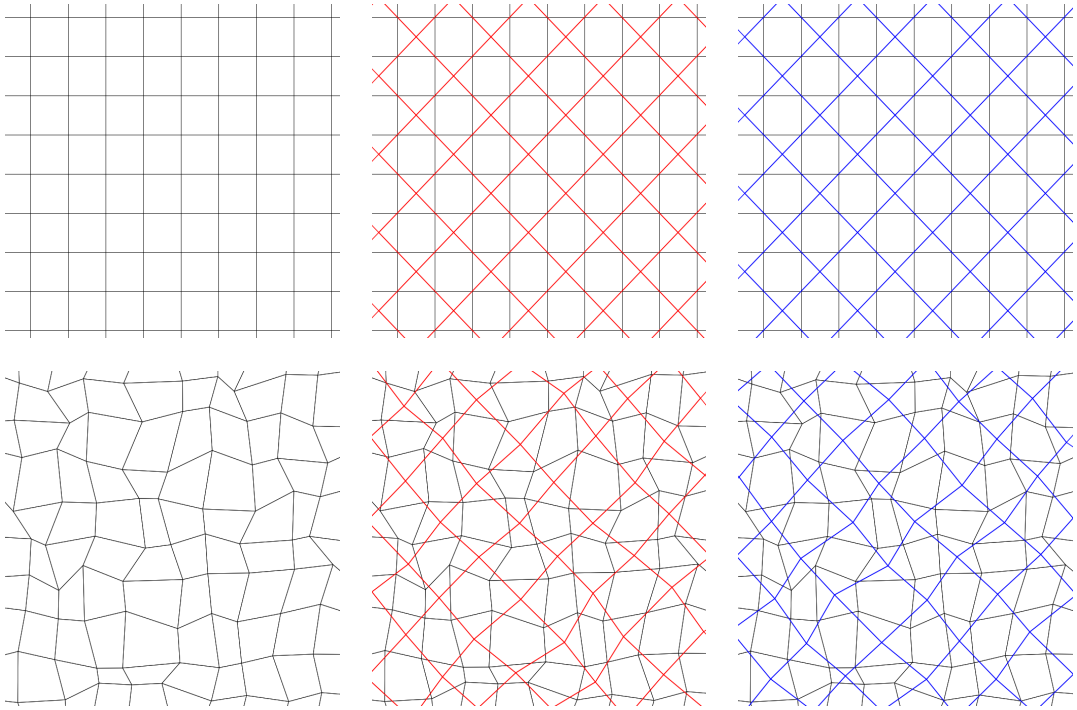
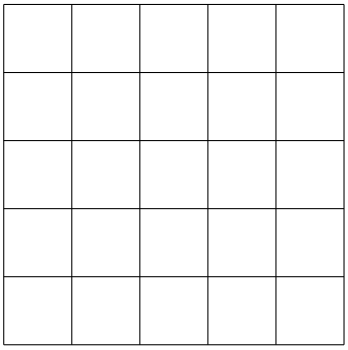
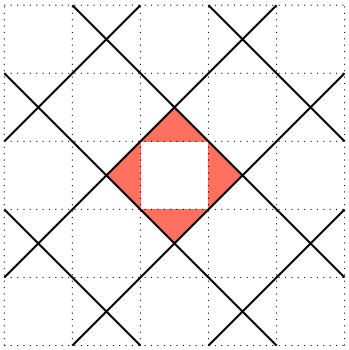


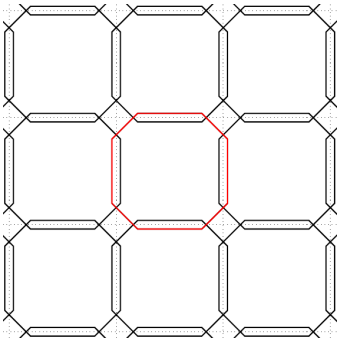
Figure 1.2: Meshes and the corresponding co-meshes. Cartesian grid (top) and randomly distorted quadrilateral grid (bottom).



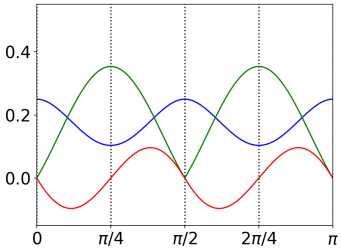
1.3a



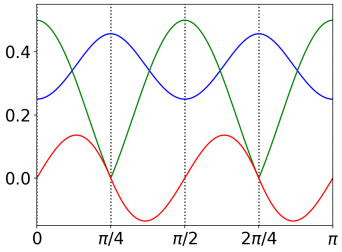
1.3b



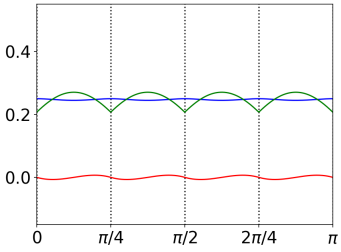
1.3c



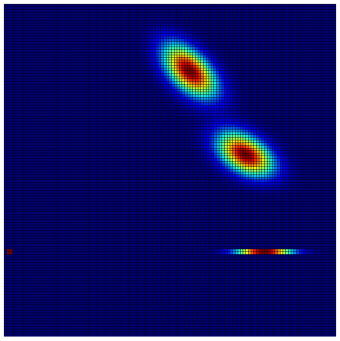
1.4a



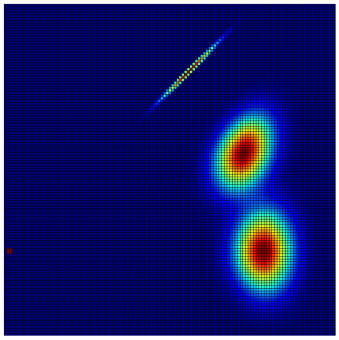
1.4b



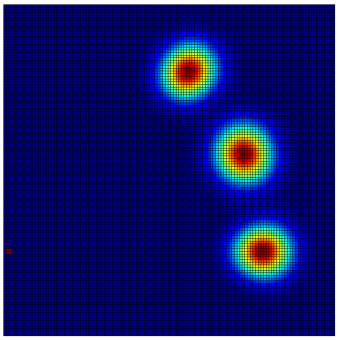
1.4c



1.5a



1.5b



1.5c

Results for three different schemes given by: $\omega = 1$ (a), $\omega = 0$ (b), and ω_{opt} (c):

Figure 1.3: Initial Cartesian grid (a) and *co-mesh* (b). Applying the co-mesh strategy is equivalent to applying the initial transport operator on a non-tailing but volume-preserving octagonal grid (c).

Figure 1.4: Graph of the dimensionless coefficients of matrix M in basis $\{\boldsymbol{e}, \boldsymbol{e}_\perp\} = \{\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}, \boldsymbol{e}_\perp\}$ as a function of transport direction θ: $\boldsymbol{e} \cdot \mathbf{M} \cdot \boldsymbol{e}$ (blue), $\boldsymbol{e}_\perp \cdot \mathbf{M} \cdot \boldsymbol{e}_\perp$ (green), and $\boldsymbol{e} \cdot \mathbf{M} \cdot \boldsymbol{e}_\perp$ (red), scaled by $h\|\boldsymbol{v}\|$.

Figure 1.5: Representation of numerical diffusion on the transport of a "delta" function (four cells at bottom left corner) along directions $\boldsymbol{v} = \|\boldsymbol{v}\|(\cos\theta, \sin\theta)$, for $\theta = 0$, $\pi/8$ and $\pi/4$; transport over a radius of $96h$ (where h is the spatial discretization step), on a 128×128 grid, in 192 iterations, with CFL $= 0.5$.

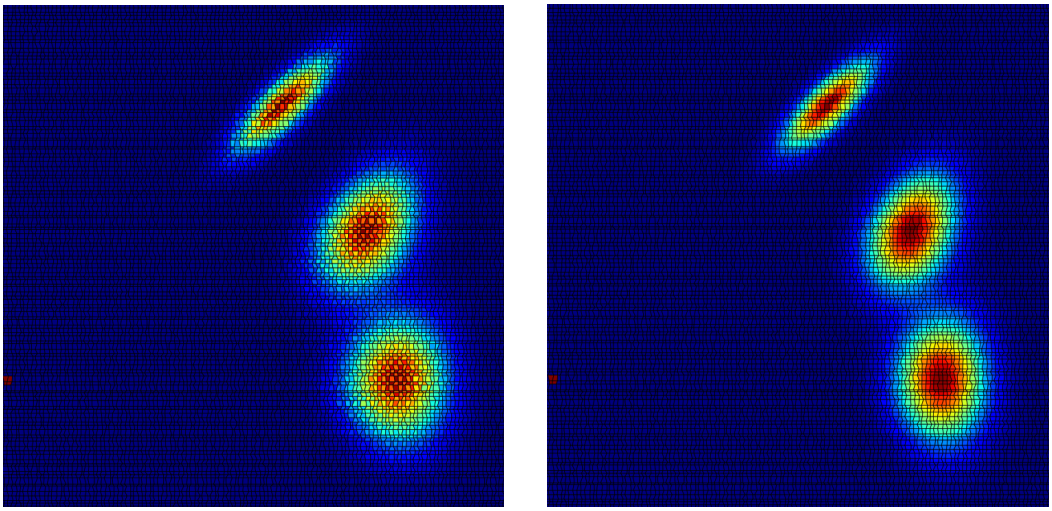


Figure 1.6: Solver applied to co-mesh ($\omega = 0$). Wrong estimation of cell-volume violates conservation and monotonicity (left), compared to the correct result (right). (For details on the simulation see Fig. 1.5.)

where $\mathcal{D}_1(c)$ and $\mathcal{D}_2(c)$ are the sets of respectively first and second nearest neighbors, and vectors $\boldsymbol{s}_{cd}$ are weighted with linear factors ω and $(1-\omega)$ as

$$\boldsymbol{s}_{cd} = \begin{cases} \omega \boldsymbol{s}_{cd}^{(1)} & \text{if} \quad d \in \mathcal{D}_1(c), \\ (1-\omega)\boldsymbol{s}_{cd}^{(2)} & \text{if} \quad d \in \mathcal{D}_2(c). \end{cases} \tag{1.9}$$

Hence, the co-mesh method applied to the first-order transport scheme $\mathrm{D}_{\Delta t} a_c^n = 0$ on an Eulerian grid (i.e. $V_c^{n+1} = V_c^n =: V_c = \omega V_c^{(1)} + (1-\omega) V_c^{(2)}$) writes

$$\frac{a_c^{n+1} - a_c^n}{\Delta t^n} + \frac{1}{V_c} \sum_{\substack{i=1,2 \\ d \in \mathcal{D}_i(c)}} \left(a_c^n \mathring{V}_{cd}^{(i),n} - a_d^n \mathring{V}_{dc}^{(i),n} \right) = 0, \tag{1.10}$$

where $\mathcal{D}_1(c)$ and $\mathcal{D}_2(c)$ are the set of first and second nearest neighbors respectively, and the superscripts (1) and (2) indicate initial and co-mesh. In other words, the geometry of the co-mesh defines the coefficients of the second nearest neighbors in the stencil.

1.3.3 Reducing anisotropy

The co-mesh strategy aims at reducing anisotropy. In order to find the most isotropic transport, we seek the value of ω leading to some minimal measure of anisotropy. Here, the modified equation [78] is used to study numerical dissipation of (1.10) and to determine ω. The modified equation is the equation that is actually solved to higher order by a first-order scheme of a given initial equation. It writes

$$(\partial_t a)_c^n + v_x \partial_x a_c^n + v_y \partial_y a_c^n = \left(M_{xx} (\partial_{xx}^2 a)_c^n + 2 M_{xy} (\partial_{xy}^2 a)_c^n + M_{yy} (\partial_{yy}^2 a)_c^n \right) \tag{1.11a}$$

$$=: \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix}^t \mathbf{M}(\boldsymbol{v}) \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} a_c^n. \tag{1.11b}$$

M_{xx}, M_{xy}, M_{yy} are the effective diffusive coefficients that characterize the numerical error, and depend on the magnitude and orientation of the velocity $\boldsymbol{v}$.

Consider (1.10) for constant transport direction $\boldsymbol{v} := \|\boldsymbol{v}\|(\cos\theta, \sin\theta)$, with $\theta \in [0, \pi/4]$. Then, the stencil is defined through cell $c = (i,j)$ and its donor cells $\mathcal{D}_1(c) = \{(i-1,j), (i,j-1)\}$ and $\mathcal{D}_2(c) = \{(i-1,j+1), (i-1,j-1)\}$. As detailed in Appendix 1.5.2, the diffusion matrix on the right hand side of (1.11a) has coefficients

$$M_{xx} = \frac{1}{2}\left(1 - \frac{v_x \Delta t}{\Delta x}\right) v_x \Delta x, \tag{1.12a}$$

$$M_{xy} = \frac{1}{2}\left((1-\omega) v_y - \sqrt{\frac{v_x \Delta t}{\Delta x} \frac{v_y \Delta t}{\Delta y}} \sqrt{v_x v_y}\right) \Delta x, \tag{1.12b}$$

$$M_{yy} = \frac{1}{2}\left((1-\omega) v_x + \left(\omega - \frac{v_y \Delta t}{\Delta y}\right) v_y\right) \Delta x. \tag{1.12c}$$

Consider matrix $\mathbf{M}(\boldsymbol{v})$ taken in the basis $\{\boldsymbol{e}, \boldsymbol{e}_\perp\} = \{\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}, \boldsymbol{e}_\perp\}$, with transport direction $\boldsymbol{v} = \|\boldsymbol{v}\|(\cos\theta, \sin\theta)$, and transverse direction $\boldsymbol{e}_\perp = (-\sin\theta, \cos\theta)$, which writes

$$\mathbf{M}_{\boldsymbol{v}} := \begin{pmatrix} M_\parallel & M_\times \\ M_\times & M_\perp \end{pmatrix} = \begin{pmatrix} \boldsymbol{e}^t\mathbf{M}\boldsymbol{e} & \boldsymbol{e}^t\mathbf{M}\boldsymbol{e}_\perp \\ \boldsymbol{e}_\perp^t\mathbf{M}\boldsymbol{e} & \boldsymbol{e}_\perp^t\mathbf{M}\boldsymbol{e}_\perp \end{pmatrix}. \tag{1.13}$$

Fig. 1.4 shows the coefficients of matrix $\mathbf{M}_{\boldsymbol{v}}$ over transport direction defined by θ. The imbalance between these coefficients reflects the transport anisotropy of the scheme. Transport would be isotropic, if the coefficients of $\mathbf{M}_{\boldsymbol{v}}$ would not change for different transport direction defined by angle θ. Thus, reducing transport anisotropy means reducing anisotropy of $\mathbf{M}_{\boldsymbol{v}}$. This is done through numerical optimization by minimizing some functional $g(\omega) = \|f_\omega(\theta)\|$, that describes transport anisotropy of $\mathbf{M}_{\boldsymbol{v}}$ depending on ω. Thus, minimizing over ω leads to an optimal value

$$\omega_{opt} := \arg\min_{\omega\in[0,1]} g(\omega) = \arg\min_{\omega\in[0,1]} \|f_\omega(\theta)\|. \tag{1.14}$$

$g(\omega)$	$\|M_\times\|_\infty$	$\|M_\times - \mu(M_\times)\|_2$	$\|M_\parallel - \mu(M_\parallel)\|_2$
ω_{opt}	0.58578643762690508	0.587514086875127	0.585863130802334

Table 1.1: Optimized values of ω for three different functionals g.

Table 1.1 shows the optimal ω computed for different minimization functional $g(\omega)$, where μ is the mean value over interval $(0, \pi/4)$. The results for ω_{opt} are very similar for these norms. However, minimizing coefficient $M_\times$ in the L^∞ norm seems to be the most ideal choice. From now on, the present work refers to the optimal value as $\omega_{opt} = \arg\min_{\omega\in[0,1]} \|M_\times\|_\infty = 0.58578643762690508$. Fig. 1.5c and 1.7 show the co-mesh strategy with ω_{opt} applied to the Cartesian grid and a randomly distorted quadrilateral grid.

1.4 Conclusion

The generic formulation for the discrete Eulerian transport operator has been introduced. A consistent version has been deduced on the co-mesh approach. The co-mesh strategy leads to improved isotropy for ω_{opt}, as visible by comparing Fig. 1.5a to Fig. 1.5c. The co-mesh approach has been introduced on usual 2D Cartesian, and distorted quadrilateral structured grids. Transport anisotropy is reduced on all of these general quadrilateral structured grids, where first-order consistency is guaranteed on 2D Cartesian, and uniformly distorted Cartesian grids (i.e. grids of identical parallelograms).

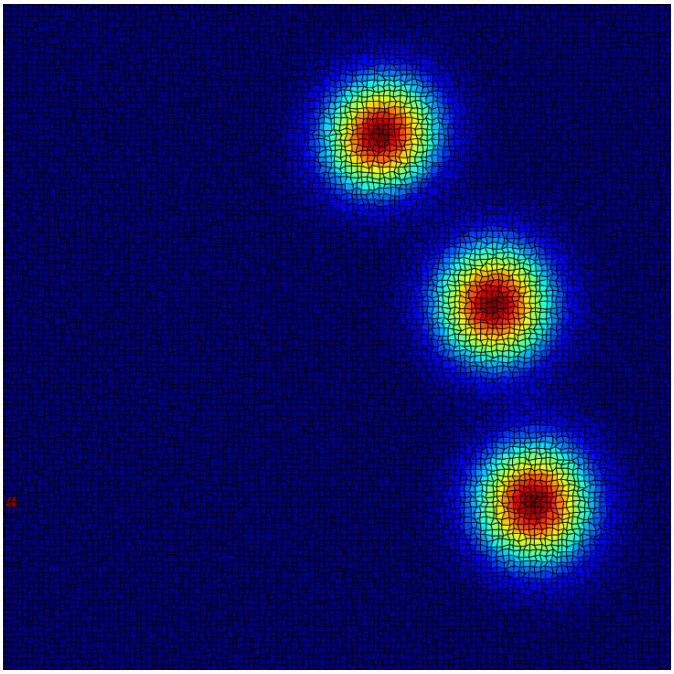


Figure 1.7: Co-mesh strategy applied to a randomly distorted quadrilateral grid with ω_{opt} (details on the simulation are provided in Fig. 1.5).

Applying the co-mesh strategy to 3D needs some further considerations. It is not obvious how this would work, especially because difficulties arise by introducing third nearest neighbors. However, it is immanent for a 3D extension that the meshes for first to third nearest neighbors are respectively built from hexahedra, rhombic dodecahedra and truncated octahedra in order to respect tessellation.

The co-mesh strategy has been tested on first-order transport on an Eulerian grid. It can be readily inserted in a GEEC approach, which defines a quasi-symplectic ALE scheme and requires a consistent fomulation of the transport operator for mass, momentum and internal energy equations. A second-order extension is also being investigated.

1.5 Appendix

1.5.1 First-order development and consistency conditions

This paragraph provides the derivation of consistency conditions (1.6) for the first-order discretization (1.2) of the transport operator (1.1). Some details on the special case of Cartesian meshes are also provided.

First-order Taylor expansions in time around t^n and space around center of

mass $\boldsymbol{x}_c^n$ of cell c give

$$V_c^{n+1} = V_c^n + \Delta t\ \partial_t V_c^n + \mathcal{O}(\Delta t^2), \tag{1.15a}$$
$$a_c^{n+1} = a_c^n + \Delta t\ (\partial_t a)_c^n + \mathcal{O}(\Delta t^2), \tag{1.15b}$$
$$a_d^n = a_c^n + \delta\boldsymbol{x}_{cd}^n \cdot (\boldsymbol{\nabla} a)_c^n + \mathcal{O}(||\delta\boldsymbol{x}||^2), \tag{1.15c}$$
$$\boldsymbol{v}_q = \boldsymbol{v}_c^n + \delta\boldsymbol{x}_{cq}^n \cdot (\boldsymbol{\nabla} \otimes \boldsymbol{v})_c^n + \mathcal{O}(||\delta\boldsymbol{x}||^2), \tag{1.15d}$$

where $\delta\boldsymbol{x}_{cd}^n := \boldsymbol{x}_d^n - \boldsymbol{x}_c^n$ and $\delta\boldsymbol{x}_{cq}^n := \boldsymbol{x}_q^n - \boldsymbol{x}_c^n$. Combining these expressions and with the definition of $\mathring{V}_{dc}^n$ in (1.3), the first-order expansion of the transport scheme (1.2) is

$$\begin{aligned}(\partial_t a)_c^n + \frac{\partial_t V_c^n}{V_c^n} a_c^n \\ + \frac{1}{V_c^n}\boldsymbol{v}_c^n \cdot \sum_{d,q}(\boldsymbol{s}_{cdq}^n - \boldsymbol{s}_{dcq}^n) a_c^n + \frac{1}{V_c^n}\sum_{d,q}\delta\boldsymbol{x}_{cq}^n \cdot (\boldsymbol{\nabla}\otimes\boldsymbol{v})_c^n \cdot (\boldsymbol{s}_{cdq}^n - \boldsymbol{s}_{dcq}^n) a_c^n \\ + \frac{1}{V_c^n}\sum_{d,q} -\boldsymbol{v}_c^n \cdot \boldsymbol{s}_{dcq}^n\ \delta\boldsymbol{x}_{cd}^n \cdot (\boldsymbol{\nabla} a)_c^n = \mathcal{O}(\Delta t, ||\delta\boldsymbol{x}||),\end{aligned} \tag{1.16}$$

where for simplicity the upwinding factors σ_{cdq} have been omitted (i.e. $\sigma_{cdq}\boldsymbol{s}_{dcq} \to \boldsymbol{s}_{dcq}$) and sums on d or q are now restricted by setting $\boldsymbol{s}_{cdq} = 0$ whenever $d \notin \mathcal{D}(c)$ or $q \notin \mathcal{Q}(c)$.

Now, the the Eulerian transport operator can be decomposed as $\mathrm{D}_t a = \partial_t a + a\boldsymbol{\nabla}\cdot\boldsymbol{w} + \boldsymbol{\nabla}\cdot(a\boldsymbol{v}) = \partial_t a + a\boldsymbol{\nabla}\cdot\boldsymbol{w} + a\boldsymbol{\nabla}\cdot\boldsymbol{v} + \boldsymbol{v}\cdot\boldsymbol{\nabla} a$, and thus term to term identification to first order with (1.16) yields

$$\frac{\partial_t V_c^n}{V_c^n} a_c^n = a_c^n (\boldsymbol{\nabla}\cdot\boldsymbol{w})_c^n, \tag{1.17a}$$
$$\frac{1}{V_c^n}\sum_{d,q}\boldsymbol{v}_c^n \cdot (\boldsymbol{s}_{cdq}^n - \boldsymbol{s}_{dcq}^n)\ a_c^n = 0, \tag{1.17b}$$
$$\frac{1}{V_c^n}\sum_{d,q}\delta\boldsymbol{x}_{cq}^n \cdot (\boldsymbol{\nabla}\otimes\boldsymbol{v})_c^n \cdot (\boldsymbol{s}_{cdq}^n - \boldsymbol{s}_{dcq}^n)\ a_c^n = a_c^n (\boldsymbol{\nabla}\cdot\boldsymbol{v})_c^n, \tag{1.17c}$$
$$\frac{1}{V_c^n}\sum_{d,q} -(\boldsymbol{v}_c^n \cdot \boldsymbol{s}_{dcq}^n)\ (\delta\boldsymbol{x}_{cd}^n \cdot (\boldsymbol{\nabla} a)_c^n) = \boldsymbol{v}_c^n \cdot (\boldsymbol{\nabla} a_c)^n. \tag{1.17d}$$

As these conditions must hold whatever the transported field a and the transport

velocity v—that is whatever a_c^n, $(\boldsymbol{\nabla} a)_c^n$, $\boldsymbol{v}_c^n$, and $(\boldsymbol{\nabla} \otimes \boldsymbol{v})_c^n$,—they simplify into

$$\partial_t V_c^n = V_c^n \, (\boldsymbol{\nabla} \cdot \boldsymbol{w})_c^n, \tag{1.18a}$$

$$\sum_{d,q} (\boldsymbol{s}_{cdq}^n - \boldsymbol{s}_{dcq}^n) = \mathbf{0}, \tag{1.18b}$$

$$\sum_{d,q} (\boldsymbol{s}_{cdq}^n - \boldsymbol{s}_{dcq}^n) \otimes \delta \boldsymbol{x}_{cq}^n = V_c^n \, \mathbf{I}, \tag{1.18c}$$

$$\sum_{d,q} -\boldsymbol{s}_{dcq}^n \otimes \delta \boldsymbol{x}_{cd}^n = V_c^n \, \mathbf{I}, \tag{1.18d}$$

where $\mathbf{I}$ is the identity matrix. The grid evolution always complies with (1.18a) (it is the so-called GCL condition) thus only the last three conditions need to be retained.

When further restricting the velocity discretization by setting the set of points $\boldsymbol{x}_q$ equal to the single point $\boldsymbol{x}_c$ in $\mathring{V}_{dc}$, conditions (1.18c) and (1.18d) become identical and, reintroducing the upwinding factors, the final two conditions provided in (1.6) are obtained.

In the case of a 2D Cartesian mesh with transport between adjacent cells, constraint (1.18d) is simply expanded along x and y coordinates and, with explicit upwinding factors σ, reduces to

$$\sum_{d \in \mathcal{D}(c)} \sigma_{cd} s_{cd,x} \delta x_{cd,x} = V_c, \tag{1.19a}$$

$$\sum_{d \in \mathcal{D}(c)} \sigma_{cd} s_{cd,y} \delta x_{cd,y} = V_c, \tag{1.19b}$$

$$\sum_{d \in \mathcal{D}(c)} \sigma_{cd} s_{cd,x} \delta x_{cd,y} = 0, \tag{1.19c}$$

$$\sum_{d \in \mathcal{D}(c)} \sigma_{cd} s_{cd,y} \delta x_{cd,x} = 0. \tag{1.19d}$$

It is readily observed that these conditions are fulfilled with a velocity of *uniform direction* and on a *uniform* mesh: only one donor cell appears in each sum and $V_c = s_{cd} \delta x_{cd}$ for any couple of neighboring cells cd.

1.5.2 Modified equation applied to the first-order transport scheme with co-meshes

The numerical diffusion coefficients M_{xx}, M_{xy} and M_{yy} of the modified equation (1.11a) can be calculated by the following recipe: first the second-order development of the scheme is computed and then time derivatives higher than the scheme's order and mixed time and space derivatives are eliminated. The

latter is a straight forward computation and can be implemented in any computer algebra system (CAS) performing symbolic computations, such as Mathematica or the Python library Sympy. However, this chapter reveals some details for the calculations on scheme (1.10).

In order to compute the second-order development, the second-order Taylor expansions in time and space are introduced.

$$a_c^{n+1} = a_c^n + \Delta t(\partial_t a)_c^n + \tfrac{1}{2}(\Delta t)^2(\partial_{tt}^2 a)_c^n + \mathcal{O}(\Delta t^3), \tag{1.20a}$$

$$a_d^n = a_c^n + \delta \boldsymbol{x}_{cd}^n \cdot (\boldsymbol{\nabla} a)_c^n + \tfrac{1}{2}\delta \boldsymbol{x}_{cd}^n \cdot (\boldsymbol{\nabla}^2 a)_c^n \cdot \delta \boldsymbol{x}_{cd}^n + \mathcal{O}(||\delta \boldsymbol{x}||^3), \tag{1.20b}$$

with $\delta \boldsymbol{x}_{cd}^n := \boldsymbol{x}_d^n - \boldsymbol{x}_c^n$, for $d \in \mathcal{D}_i(c),\ i = 1, 2$. Furthermore, $\delta \boldsymbol{x}_{dc}^n = \boldsymbol{x}_c^n - \boldsymbol{x}_d^n = -\delta \boldsymbol{x}_{cd}^n$. Considering a Cartesian grid and a constant velocity vector $\boldsymbol{v}_d = \boldsymbol{v}_c$, the space discretization can be simplified. Recall from chapter 1.2 that conservation of a field a is enforced over any volume V_c by constraint $\sum_{d\in\mathcal{D}(c)} \boldsymbol{s}_{cd}^n = 0$. It follows

$$\begin{aligned}\sum_{d\in\mathcal{D}(c)} \left(\mathring{V}_{cd}^n a_c^n - \mathring{V}_{dc}^n a_d^n\right) &\overset{(1.20b)}{=} \underbrace{\sum_{d\in\mathcal{D}(c)} \underbrace{(\sigma_{cd}^n + \sigma_{dc}^n)}_{=1} \boldsymbol{s}_{cd}^n \cdot \boldsymbol{v}_c^n a_c^n}_{=0} \\ &- \sum_{d\in\mathcal{D}(c)} \sigma_{dc}^n \boldsymbol{s}_{dc}^n \cdot \boldsymbol{v}_c^n \left(\delta \boldsymbol{x}_{cd}^n \cdot (\boldsymbol{\nabla} a)_c^n + \tfrac{1}{2}\delta \boldsymbol{x}_{cd}^n \cdot (\boldsymbol{\nabla}^2 a)_c^n \cdot \delta \boldsymbol{x}_{cd}^n\right). \end{aligned} \tag{1.21}$$

A simple computation shows that (1.18d) is valid on the Cartesian mesh and its co-mesh. Therefore, the following term on the right hand side of (1.21) can be simplified in this case and becomes

$$-\sum_{\substack{i=1,2\\ d\in\mathcal{D}_i(c)}} \sigma_{dc}^{(i)} (\boldsymbol{s}_{dc}^{(i)} \cdot \boldsymbol{v}_c)\ \delta \boldsymbol{x}_{cd}^n \cdot (\boldsymbol{\nabla} a)_c^n = \omega V_c^{(1)} \boldsymbol{v}_c \cdot (\boldsymbol{\nabla} a)_c^n + (1-\omega) V_c^{(2)} \boldsymbol{v}_c \cdot (\boldsymbol{\nabla} a)_c^n = V_c \boldsymbol{v}_c \cdot (\boldsymbol{\nabla} a)_c^n. \tag{1.22}$$

Then, the second-order development writes

$$(\partial_t a)_c^n + \boldsymbol{v}_c \cdot (\boldsymbol{\nabla} a)_c^n = -\tfrac{1}{2}\Delta t(\partial_{tt}^2 a)_c^n + \frac{1}{V_c} \sum_{\substack{i=1,2\\ d\in\mathcal{D}_i(c)}} \sigma_{dc}^{(i)} \boldsymbol{s}_{dc}^{(i)} \cdot \boldsymbol{v}_c \left(\tfrac{1}{2}\delta \boldsymbol{x}_{cd}^n \cdot (\boldsymbol{\nabla}^2 a)_c^n \cdot \delta \boldsymbol{x}_{cd}^n\right). \tag{1.23}$$

The modified equation is obtained by eliminating the second-order time derivative in (1.23). $(\partial_{tt}^2 a)_c^n$ is given by differentiation of (1.23) in time. In this expression the mixed time and space derivatives $(\partial_{tx}^2 a)_c^n$ and $(\partial_{ty}^2 a)_c^n$ appear, which can be eliminated by differentiation of (1.23) in each spatial direction. Remark that the computations in (1.19) are valid on the co-mesh of a Cartesian grid, which is used

in the following calculations.

$$(\partial_{tt}^2 a)_c^n = \frac{1}{V_c}\sum_{\substack{i=1,2\\ d\in\mathcal{D}_i(c)}} \sigma_{dc}^{(i)} \boldsymbol{s}_{dc}^{(i)} \cdot \boldsymbol{v}_c \Big(\delta x_{cd,x}(\partial_{tx}^2 a)_c^n + \delta x_{cd,y}(\partial_{ty}^2 a)_c^n\Big) \overset{(1.19)}{=} -v_x(\partial_{tx}^2 a)_c^n - v_y(\partial_{ty}^2 a)_c^n, \tag{1.24a}$$

$$(\partial_{tx}^2 a)_c^n = \frac{1}{V_c}\sum_{\substack{i=1,2\\ d\in\mathcal{D}_i(c)}} \sigma_{dc}^{(i)} \boldsymbol{s}_{dc}^{(i)} \cdot \boldsymbol{v}_c \Big(\delta x_{cd,x}(\partial_{xx}^2 a)_c^n + \delta x_{cd,y}(\partial_{xy}^2 a)_c^n\Big) \overset{(1.19)}{=} -v_x(\partial_{xx}^2 a)_c^n - v_y(\partial_{xy}^2 a)_c^n, \tag{1.24b}$$

$$(\partial_{ty}^2 a)_c^n = \frac{1}{V_c}\sum_{\substack{i=1,2\\ d\in\mathcal{D}_i(c)}} \sigma_{dc}^{(i)} \boldsymbol{s}_{dc}^{(i)} \cdot \boldsymbol{v}_c \Big(\delta x_{cd,x}(\partial_{xy}^2 a)_c^n + \delta x_{cd,y}(\partial_{yy}^2 a)_c^n\Big) \overset{(1.19)}{=} -v_x(\partial_{xy}^2 a)_c^n - v_y(\partial_{yy}^2 a)_c^n, \tag{1.24c}$$

and therefore

$$(\partial_{tt}^2 a)_c^n = v_x^2(\partial_{xx}^2 a)_c^n + 2v_x v_y(\partial_{xy}^2 a)_c^n + v_y^2(\partial_{yy}^2 a)_c^n. \tag{1.25}$$

Hence, the coefficients of the modified equation (1.11a) write

$$2M_{xx} = -\Delta t v_x^2 + \frac{1}{V_c}\sum_{\substack{i=1,2\\ d\in\mathcal{D}_i(c)}} \sigma_{dc}^{(i)} \boldsymbol{s}_{dc}^{(i)} \cdot \boldsymbol{v}_c \ \delta x_{cd,x}^2, \tag{1.26a}$$

$$2M_{xy} = -\Delta t v_x v_y + \frac{1}{V_c}\sum_{\substack{i=1,2\\ d\in\mathcal{D}_i(c)}} \sigma_{dc}^{(i)} \boldsymbol{s}_{dc}^{(i)} \cdot \boldsymbol{v}_c \ \delta x_{cd,x}\delta x_{cd,y}, \tag{1.26b}$$

$$2M_{yy} = -\Delta t v_y^2 + \frac{1}{V_c}\sum_{\substack{i=1,2\\ d\in\mathcal{D}_i(c)}} \sigma_{dc}^{(i)} \boldsymbol{s}_{dc}^{(i)} \cdot \boldsymbol{v}_c \ \delta x_{cd,y}^2. \tag{1.26c}$$

1.5.3 Anisotropy and CFL

Consider constant velocity vector $\boldsymbol{v}_d = \boldsymbol{v}_c$. In this case, there are eight possible sets of active donor cells. These sets are defined through the intervals $I_k = [\frac{k\pi}{4}, \frac{(k+1)\pi}{4}]$, for $k \in \mathbb{Z}/8\mathbb{Z}$. Choose for instance transport direction $\boldsymbol{v} = \|\boldsymbol{v}\|(\cos\theta, \sin\theta)$, with $\theta \in [0, \frac{\pi}{4}]$, then on the Cartesian mesh, where $\delta x_x = \delta x_y =: \Delta x$, the numerical diffusion matrix defined in (1.11b) writes

$$\mathbf{M}(\boldsymbol{v}) = \frac{1}{2}\Delta x \begin{pmatrix} v_x & (1-\omega)v_y \\ (1-\omega)v_y & (1-\omega)v_x + \omega v_y \end{pmatrix} + \mathbf{M}_{\text{CFL}}, \qquad \theta \in [0, \pi/4], \tag{1.27}$$

with

$$\mathbf{M}_{\mathrm{CFL}} = -\frac{\Delta t}{\Delta x}\begin{pmatrix} v_x^2 & v_x v_y \\ v_x v_y & v_y^2 \end{pmatrix} \tag{1.28}$$

Remark that the coefficients of $\mathbf{M}(\boldsymbol{v})$ change for the different sets of donor cells symmetrically. Fig. 1.4a to 1.4c illustrate the symmetries of $\mathbf{M}(\boldsymbol{v})$ in basis $\{\boldsymbol{e}, \boldsymbol{e}_\perp\}$ over I_0 to I_3.

Recall the representation of matrix $\mathbf{M}$ in basis $\{\boldsymbol{e}, \boldsymbol{e}_\perp\} = \{\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}, \boldsymbol{e}_\perp\}$, noted $\mathbf{M}_{\boldsymbol{v}}$, with transport direction $\boldsymbol{v} = \|\boldsymbol{v}\|(\cos\theta, \sin\theta)$, and transverse direction $\boldsymbol{e}_\perp = (-\sin\theta, \cos\theta)$.

$$\mathbf{M}_{\boldsymbol{v}} = \begin{pmatrix} M_\parallel & M_\times \\ M_\times & M_\perp \end{pmatrix} = \begin{pmatrix} \boldsymbol{e}\cdot\mathbf{M}\cdot\boldsymbol{e} & \boldsymbol{e}\cdot\mathbf{M}\cdot\boldsymbol{e}_\perp \\ \boldsymbol{e}\cdot\mathbf{M}\cdot\boldsymbol{e}_\perp & \boldsymbol{e}_\perp\cdot\mathbf{M}\cdot\boldsymbol{e}_\perp \end{pmatrix}. \tag{1.29}$$

The following calculations show that the representation of this matrix in basis $\{\boldsymbol{e}, \boldsymbol{e}_\perp\}$ does not depend on the CFL up to a linear term (and this only for coefficient $M_\parallel$), as the coefficients of $\mathbf{M}_{\boldsymbol{v},\mathrm{CFL}}$ are constant over θ.

$$M_{\parallel,\mathrm{CFL}} = \boldsymbol{e}\cdot\mathbf{M}_{\mathrm{CFL}}\cdot\boldsymbol{e} = -\frac{\Delta t}{\Delta x}\left(v_x^4 + 2v_x^2v_y^2 + v_y^4\right) = -\frac{\Delta t}{\Delta x}\Big(\underbrace{v_x^2 + v_y^2}_{=\cos^2\theta+\sin^2\theta=1}\Big)^2 = -\frac{\Delta t}{\Delta x}, \tag{1.30a}$$

$$M_{\times,\mathrm{CFL}} = \boldsymbol{e}\cdot\mathbf{M}_{\mathrm{CFL}}\cdot\boldsymbol{e}_\perp = -\frac{\Delta t}{\Delta x}\Big(\underbrace{v_x^3 v_{x_\perp} + v_x v_y (v_x v_{y_\perp} + v_{x_\perp} v_y) + v_y^3 v_{y_\perp}}_{=\sin\theta\cos\theta(-\cos^2\theta+\cos^2\theta-\sin^2\theta+\sin^2\theta)=0}\Big) = 0, \tag{1.30b}$$

$$\begin{aligned} M_{\perp,\mathrm{CFL}} = \boldsymbol{e}_\perp\cdot\mathbf{M}_{\mathrm{CFL}}\cdot\boldsymbol{e}_\perp &= -\frac{\Delta t}{\Delta x}\left(v_x^2v_{x_\perp}^2 + 2v_x v_{x_\perp} v_y v_{y_\perp} + v_y^2 v_{y_\perp}^2\right) \\ &= -\frac{\Delta t}{\Delta x}\Big(\underbrace{v_x v_y + v_{x_\perp} v_{y_\perp}}_{=-\cos\theta\sin\theta+\sin\theta\cos\theta=0}\Big)^2 = 0. \end{aligned} \tag{1.30c}$$

Part II

Doubly-monotonic constraint on interpolators: bridging second-order to singularity preservation to cancel "numerical wetting" in transport schemes

Chapter 2

Doubly-monotonic constraint on interpolators: bridging second-order to singularity preservation to cancel "numerical wetting" in transport schemes

2.1 Introduction

2.1.1 Motivations: "numerical wetting"

Transport operators are central elements of the PDEs found in many scientific fields, such as and foremost in physics: not only do they describe actual transport of conserved quantities but they also capture, after proper transformations, the propagation of invariants in hyperbolic equations. The design and understanding of numerical techniques for transport operators and equations are thus of prime importance and have drawn an enormous amount of effort over the last seventy years. Now, as early recognized, most of the difficulties in this far reaching endeavor already appear in the simplest system: the passive one-dimensional transport of a scalar field $y(t, x)$ by a velocity field $u(t, x)$

$$\partial_t y + \partial_x(uy) = \partial_t y + u\,\partial_x y + y\,\partial_x u = 0, \tag{2.1}$$

here written in conservative form. Because of its importance, this basic linear equation has attracted special attention, with ensuing publications in staggering numbers [79, and refs therein].

On even grids in dimension one, there is now a good understanding of the main distortions produced by solvers of (2.1): instabilities, smearing of discontinuities

and numerical diffusion, oscillations, and loss of conservativity, positivity or monotonicity. These can be kept under control by important scheme features which are now standard items of the numericists' toolbox: finite volumes approaches, higher-order discretizations, stability analysis, modified equation interpretations, flux upwinding, and TVD flux limitation [79, and refs therein]. However, another artifact, here designated as "numerical wetting," has been somewhat neglected so far, possibly for being less amenable to usual accuracy-focused topological analysis.

Numerical wetting designates the *low-level contamination that spreads linearly in time* over all the regions of the computational domain where non-vanishing transport is present. It is not to be confused with usual numerical diffusion which spreads profiles sub-linearly in time, as t^θ with $\theta \lesssim \frac{1}{2}$ [80]. Numerical wetting may appear of marginal importance—and indeed is in terms of numerical accuracy when compared to other more significant truncation errors—but it turns out to be especially irritating in practical situations involving fluid mixing and evanescent boundary conditions. Two such situations of interest in industrial and academic applications will be mentioned here although many others can also be found.

1. "Phase disappearance" in multiphase flows Multiphase or multi-fluid flows appear in numerous applications where they are modeled by coupled Euler like equations for their different components. When the flow conditions make one (or more) of the components disappear at a given time and position, numerical wetting forbids its full disappearance, i.e. its volume fraction cannot be made to cancel to round-off error under the sole action of the transport scheme. This can trigger singular or unstable behavior of the physical model or the numerical scheme: for instance, with highly contrasted compressible fluids such as water and air, pressure relaxation between fluids can become surprisingly stiff at vanishing volume fractions [81, 82, resp. eq. 6 & § 3.6].

2. "Wet–dry transition" in shallow water flows Shallow water calculations are generally restricted to positive depth regions, delimited by shores which, physically, can switch between wet and dry as waves propagate. This situation is captured by numerical schemes which either forbid full drying of shore cells—a thin numerical "film" of liquid is then always present—or provide for a special local treatment—with a numerical "shore-line" reconstruction. Both options require special attention as they can induce singular or fragile behavior. This is not surprising as the shallow water equations are formally identical with a degenerate form of the two-fluid flow equations.

More or less empirical and singular fixes of numerical wetting have been devised by almost all numericists involved in developing practical codes for these two examples. Understandably, these recipes have been seldom publicized but a few

examples can be found and are worth mentioning here. The basic "brute force" method is to artificially bring to zero all volume fractions falling below a (hopefully low) predefined threshold, bearing in mind that this may be detrimental to exact conservation over long times. Typical threshold values can be found at 10^{-3} for slugging in pipes [83, § 4.6 & refs therein] or 10^{-5} to 10^{-4} for boiling in nuclear reactors [84, § 7 & refs therein]. These levels can be considered as too high and unsatisfactory and have triggered the development of more subtle approaches. At the high end of sophistication, a surprising method was proposed whereby the evolution equations are extended to an artificial domain of *negative volume fraction* where all the closures are deliberately continued so as to mimic the behavior of the system at *zero volume fraction*. The same model equations and numerical scheme are thus applied in a uniform way throughout the computational domain. The technique appears to have been applied only to transport in porous media [85]. Similarly, the wet–dry transition in shallow water flows is amenable to widely different numerical treatments [86, § 1 and refs therein].

It must be stressed that the reduction or cancellation of numerical wetting cannot solve all the issues related to phase disappearance (or its formally equivalent avatars) such as thermodynamic consistency. However, it is a critical ingredient to tackle such singular situations. Furthermore, through symmetries and dualities, the discretization of transport can sometimes constrain or even fully define the discretization of all the evolution equations: an example was recently provided by a geometrically consistent discretization of the multi-fluid Euler equations [82]. Numerical wetting can thus impact a numerical scheme in many of its features beyond transport terms.

The cancellation of numerical wetting in practical applications was the primary motivation for the present investigation. Now, when analyzing this problem, the authors realized that a broader perspective had to be taken in order to provide an horizon of accessible solutions and to build actual optimal algorithms for wetting-free transport (here to second order). This work will thus be devoted mostly to the general doubly-monotonic interpolants introduced in subsection 2.1.4, whereas cancellation of numerical wetting will appear almost as a byproduct when applying the slope clipping introduced in subsection 2.1.3 to the numerical experiments of section 2.5.

2.1.2 Numeric background: Muscl-like scheme

General methods on numerical solvers of (2.1) are described and analyzed in many standard textbooks. Beyond their widely different and numerous interpretations—for instance as finite volumes or elements, with Riemann solvers, fluxes, swept fluxes, remap, averaging, etc.—they all involve a common critical feature of major interest for the present work: flux or slope limiting [79, § 6]. This stems from the much

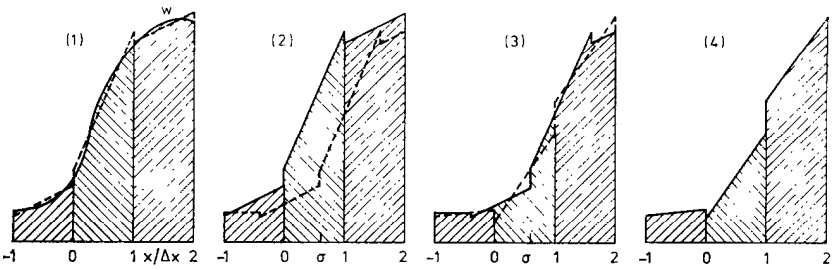


FIG. 2. The second-order upstream-centered scheme (in particular, scheme III). (1) approximating the initial-value distribution (solid line) in each slab by a linear distribution (broken line) with the same mesh integral. In this case the slopes are determined by least-squares fitting. (2) The approximate initial-value distribution before (solid) and after (broken) convection over a distance $\sigma\Delta x$. (3) Determining the new linear distributions (broken) in each mesh by least-squares fitting to the convected distribution (solid). (4) The initial values for the next time step.

Figure 2.1: Graphical illustration of MUSCL applied to scalar transport in 1D (reproduced from [92]).

celebrated theorem by Godunov [87] whereby all methods of order above one must be non-linear to achieve positivity or monotonicity. It has thus fueled the development of an enormous number of (non-linear) second- and higher-order limiters over the last half century, each with specific properties in connection with the underlying scheme and numerical setup as reviewed for instance in [88, 89, 90, 91].

The present work will elaborate its main concepts as extensions of the MUSCL limiter approach (Monotonic Upstream-centered Scheme for Conservation Laws) described in the original work of van Leer [92],[1] which was thoroughly commented by Rider [90]. When applied to (2.1) as illustrated in [92, figs 1 & 2], MUSCL is basically a "Reconstruct–Evolve–Average" scheme—REA, as designated in [79, § 4.10] and also known as "Reconstruct–Evolve–Project" (REP). Despite this restriction, the results obtained here will be applicable to practically all transport schemes. Notably, it was remarked that closely similar "incremental remap" schemes are equivalent to flux schemes [93, 94, 95, 96]. The subtle differences between them were recently reviewed [97].

The scheme retained here is actually an "Evolve–Reconstruct–Average" (ERA) or "Lagrange + Remap" (LR) as originally introduced in [98] and reproduced in Figure 2.1. It may be interpreted as performing a time splitting of the transport

[1]This reference discusses and combines many concepts, such as upwinding, REA, or TVD which were actually introduced in earlier works, but it provides clear understanding on the deep connections between them. The MUSCL acronym was introduced in an earlier publication by van Leer [92, ref. 8].

equation (2.1) according to

$$\partial_t y + y\,\partial_x u = 0, \tag{2.2a}$$
$$\partial_t y + u\,\partial_x y = 0, \tag{2.2b}$$

which displays an important property: *the remap sub-step does not actually perform time evolution and is independent of the Lagrangian sub-step.* Indeed, if all detailed cell-to-cell intersections are computed, the geometry and topology of the remapped mesh can be chosen at will, unconstrained by the initial mesh or the mesh distortions. This comes with significant advantages: (i) the order of accuracy of the LR step is given by the lowest of the Lagrangian and remap sub-step orders—no Strang or higher-order splitting is necessary;—(ii) analysis of dissipation and entropy is much simpler—only numerical diffusion appears during remap;—(iii) effects of non divergence-free, discontinuous, or oddly centered velocity fields [82, § 4.2 for instance] can be captured properly by the Lagrangian sub-step—yielding possibly non-monotonous-but-consistent positive evolution;—iv) remap is intrinsically upwind—thus allowing for centered Lagrangian sub-steps;—and v) second-order accuracy requires second-order reconstruction for the remap sub-step, but does not require any reconstruction for the Lagrangian sub-step if the discretization of u is space- and time-staggered with respect to that of y.

Following the above remarks, the present work will focus on the monotonicity of reconstruction approaches for the remap sub-step only. However, numerical tests will be carried out on the full LR transport scheme to first- or second-order accuracy with a simple reconstruction-free Lagrangian sub-step.

2.1.3 Present approach: from limiting to clipping of interpolants...

The simplest approach to second-order remap in Muscl relies on single-slope interpolants in each cell, defined so as to exactly connect the neighboring cells values whenever these are aligned. When cell values depart from alignment, the linear interpolant may violate the monotonicity condition and its slope must be limited. This limitation also applies on non-uniform meshes [92, § 4] as detailed in [99, § III], or to Courant-number-dependent fluxes. The behavior of the slope strength as a function of cell average values is usually represented on the so-called Sweby diagram [100, fig. 1]—or more conveniently, on its fully symmetric versions of Berger–Aftosmis–Muman [99, § II.C, fig. 3] or Waterson [101, § 2, fig. 2].

Similar limiting approaches were also designed for more complex interpolants in order to increase the order of accuracy, or to better capture extrema or strong gradient regions. Some selected examples are quadratic functions for third order schemes [92, § 4] or [102]; higher order polynomials [103, and refs therein]; piecewise

hyperbolic [104], rational [105], hyperbolic tangent [106], and double logarithm [107] functions for smoother monotonicity conditions; piecewise linear [108, § 4]; quadratic functions corrected at smooth extrema [109, 110, 111]. Most notable for the present study as discussed in section 2.2, is the method introduced by Després and Lagoutière [112, 113], based on a saturated step function and here designated as "downwind-limited upwind," which is closely related to interface reconstruction techniques and to the so-called "reservoir technique" [114].

Except in rare cases, many such schemes mingle the two independent substeps of accurate interpolation and monotonicity limitation. This can seriously impact the final limited interpolants by reducing the set of accessible functions and by requiring more complex calculations: the respective stencils for high-order interpolation and monotonicity are very different, especially in dimensions higher than one. Monotonicity is a scalar constraint defined by only two neighboring cells, irrespective of mesh topology.

This last remark is at the core of the present work: what is the general structure of monotonic interpolants, and how do they connect to a specific configuration were a selected high-order interpolant is exact? In the present work, these issues will be examined to second-order, with single-slope reconstructions *clipped to bounds* instead of *slope limited*. The set of new possibilities is then widened to the point that supplementary conditions need to be introduced in order to avoid inconsistent behaviors and generate practical schemes. This is the *double monotonicity condition* introduced below in subsection 2.1.4

2.1.4 . . . and from simple to double monotonicity

As already noted by van Leer [92, § 5 & fig. 6], cell interpolants away from extrema can be constrained by monotonicity in two different ways which will be designated here as weak and strong. Under *weak monotonicity* the interpolant is bounded by the neighboring left and right cell values: this ensures the positive, monotonic, TVD, or BV behavior of the scheme but can let appear inconsistent variations *within cells*: for instance, despite successive *increasing* cell values, the interpolant in a cell can be *decreasing* while still being bounded by neighboring cells (see Figure 2.2b). This was designated as "stegosaur bias" and is eliminated by a *strong monotonicity* constraint [92, eq. 66 & comment below]: the interpolant in a cell must not only be bounded but also display the same increasing or decreasing character as defined by the neighboring cells (see Figure 2.2c). The rationale for this strong constraint was connected to a detailed analysis of dispersion relationships of transport schemes [92, § 3]—a more refined property than mere TVD, which is seldom analyzed and often taken as "implicitly obvious."

However, in analyzing the behavior of a transport scheme over *many successive time steps*, a yet stronger monotonicity condition appears to be required: as

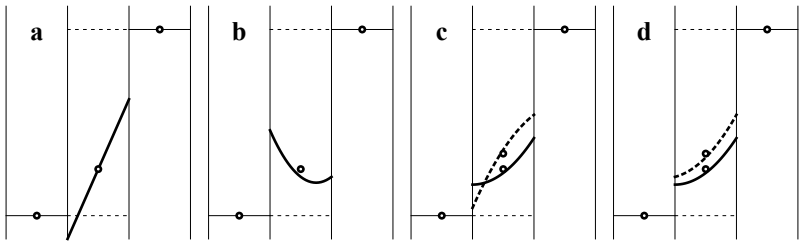


Figure 2.2: Representations of (a) non-monotonic, (b) weakly monotonic, (c) strongly (non-doubly) monotonic, and (d) doubly monotonic interpolants (solid lines) in a cell constrained by neighboring-cell values. Mean cell values $\overline{y}$ are represented by white dots. In (c) and (d) interpolants that would be generated for slightly higher values of $\overline{y}$ are also represented (dotted lines) so as to visualize double monotonicity.

illustrated in Figure 2.2c & d an interpolant function over a given cell must be monotonic with respect to the mean neighboring-cell values (*simple monotonicity*) but also be *everywhere increasing with respect to the mean cell value* (*double monotonicity*). As will be introduced in section 2.4, these two constraints are more conveniently discussed by distinguishing *interpolant* and *interpolator*: the former is an actual reconstructed function $y[x]$ over a cell with given mean value $\overline{y}$, whereas the latter is a reconstruction procedure which generates an interpolant $y[x] = z[\overline{y}, x]$ as a function of $\overline{y}$. Double monotonicity applies to interpolators $z[\overline{y}, x]$.

2.1.5 Mapping of interpolants and interpolators, 1D numerical tests

In order to provide guidance on the various acceptable strategies under double monotonicity, a new graphical representation of interpolants and interpolators is introduced in section 2.2 which vastly expands the Sweby diagram. It is based on elliptic coordinates with respect to poles defined by the order-one constant interpolant and the downwind-limited upwind interpolant.

Double monotonicity lets define some novel extensions of the classical Min-Mod, Monotonized Centered, Super-Bee, and Ultra-Bee interpolators. Basic 1D transport tests of square wave, sine wave, and Zalesak wave are provided in section 2.5 with the standard and newly defined interpolators. Specific metrics for the square-wave transport let monitor mean front widths and front support widths. Full suppression of numerical wetting will be observed with the new approaches, also with less "numerical erosion" (or better preservation of extrema).

2.2 A convenient mapping for bounded monotonic functions

2.2.1 Motivation: bounded monotonic functions as interpolants

Definition 1 (Set of bounded monotonic functions $\mathcal{M}_{XY}(\overline{y})$) *Let X and Y be two finite intervals of $\mathbb{R}$. $\mathcal{M}_{XY}(\overline{y})$ is the set of monotonic functions $y[x]$ from X towards Y with given mean*

$$\overline{y} = \int_X y[x]\,\mathrm{d}\,x \Big/ \int_X \mathrm{d}\,x. \tag{2.3}$$

Let $I = [0, 1]$. $\mathcal{M}^{\leq}_{II}(\overline{y})$ is the subset of increasing functions of $\mathcal{M}_{II}(\overline{y})$.

Functions in $\mathcal{M}_{XY}(\overline{y})$ represent, for a monotonic 1D discretization, all the monotonic interpolants over a given mesh cell X, to an interval Y whose bounds are given by values at the neighboring left and right cells. As already mentioned in subsection 2.1.4 and illustrated in Figure 2.2, monotonicity *within* a cell is sufficient but not necessary to ensure monotonicity *between* cell-average values and make a TVD Muscl scheme.

As most of the properties explored in the following are invariant through x-axis shifting, scaling, and inversion, it will be sufficient to simply consider set $\mathcal{M}^{\leq}_{II}(\overline{y})$. With further symmetrization of $y[x]$ into $1 - y[-x]$, the mean value can be constrained to $\overline{y} \in [0, \frac{1}{2}]$ without loss of generality. Incidentally, $\mathcal{M}_{XY}(\overline{y})$ and $\mathcal{M}^{\leq}_{II}(\overline{y})$ are convex.

2.2.2 L^2 bounds on bounded monotonic functions

Definition 2 (Edge and middle functions in $\mathcal{M}_{XY}(\overline{y})$) *To any couple of real numbers $(\alpha, \beta) \in I^2$ is associated a function $y_{\alpha\beta}[x]$ of $\mathcal{M}^{\leq}_{II}(\overline{y})$ designated as an "edge function" and defined as*

$$y_{\alpha\beta}[x] = \begin{cases} 0 & \text{for} \quad 0 \leq x < \alpha(1-\overline{y}), \\ \dfrac{(1-\beta)\overline{y}}{(1-\alpha)-(\beta-\alpha)\overline{y}} & \text{for} \quad \alpha(1-\overline{y}) \leq x < 1-\beta\overline{y}, \\ 1 & \text{for} \quad 1-\beta\overline{y} \leq x \leq 1. \end{cases} \tag{2.4}$$

The "middle function" of $\mathcal{M}^{\leq}_{II}(\overline{y})$ is defined as $m = (y_{11} + y_{00})/2$. The definition extends to $\mathcal{M}_{XY}(\overline{y})$ through shifting, scaling, and symmetries.

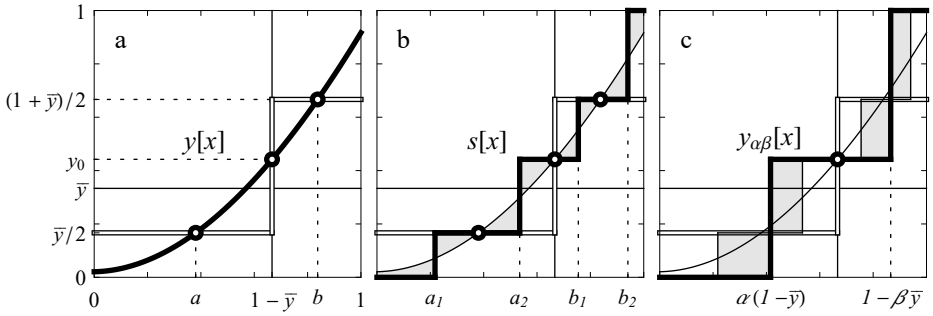


Figure 2.3: Graphical illustration in support of the Proof of Theorem 1: (a) function $y[x] \in \mathcal{M}_{II}^{\leq}(\overline{y})$ (thick line) and its intersection points (circled) with $m[x]$ (double line); (b) step function $s[x]$ (thick line) with same intersection points and means on each interval I_i; (c) "edge" step function $y_{\alpha\beta}[x]$ (thick line) of same left and right means as $s[x]$ (and $y[x]$), and with $y_{\alpha\beta}[1-\overline{y}] = y[1-\overline{y}] = y_0$.

Direct integration confirms that $y_{\alpha\beta}$ and $m \in \mathcal{M}_{II}^{\leq}(\overline{y})$. The step function $y_{\alpha\beta}$ becomes a constant function $\overline{y}$ for $\alpha = \beta = 0$ but notations y_{00} and $\overline{y}$ will be preserved to distinguish function and number. These definitions are important in the present context as $y_{\alpha\beta}[x]$ is an intermediate between the two extremes $y_{00}[x]$ and $y_{11}[x]$, also known as respectively "first order" and "downwind-limited upwind" interpolants [112, 113] in the context of MUSCL like numerical schemes for transport. The set of $y_{\alpha\beta}$ functions is also a convex generator system for $\mathcal{M}_{II}^{\leq}(\overline{y})$ in the sense of distributions (it is not a free system, however). Incidentally, functions of y_{00} and y_{11} are related through inversion as $y_{00}^{-1}[x] = y_{11}[x]$.

Lemma 1 (L^2 radius of edge functions) *For any given $\overline{y}$, all edge functions of $\mathcal{M}_{II}^{\leq}(\overline{y})$ are at the same L^2-distance of its middle function, i.e. $\|y_{\alpha\beta} - m\|_2^2 = \overline{y}(1-\overline{y})/4$. The lemma extends to $\mathcal{M}_{XY}(\overline{y})$ through shifting, scaling, and symmetries.*

Proof This results from direct integration of (2.4). ■

A stronger result is then obtained:

Theorem 1 (L^2 bounds on $\mathcal{M}_{XY}(\overline{y})$) *For any $y \in \mathcal{M}_{II}^{\leq}(\overline{y})$*

$$\|y - m\|_2^2 \leq \overline{y}(1-\overline{y})/4, \tag{2.5}$$

where m is the middle function of $\mathcal{M}_{II}^{\leq}(\overline{y})$, equality being reached if and only if there exists $(\alpha, \beta) \in I^2$ such that $y = y_{\alpha\beta} \in \mathcal{M}_{II}^{\leq}(\overline{y})$. The theorem extends to $\mathcal{M}_{XY}(\overline{y})$ through shifting, scaling, and symmetries.

Proof A function $y \in \mathcal{M}_{II}^{\leq}(\overline{y})$ being given, let ordinate y_0 and abscissas a and b defined as (see Figure 2.3a)

$$y_0 = y[1-\overline{y}], \tag{2.6a}$$

$$y[a] = m[a] = \overline{y}/2 \qquad \text{or} \quad a = \begin{cases} 0 & \text{if} \quad \forall x \in [0, 1-\overline{y}],\ y[x] > m[a], \\ 1-\overline{y} & \text{if} \quad \forall x \in [0, 1-\overline{y}],\ y[x] < m[a], \end{cases} \tag{2.6b}$$

$$y[b] = m[b] = (1+\overline{y})/2 \quad \text{or} \quad b = \begin{cases} 1-\overline{y} & \text{if} \quad \forall x \in [1-\overline{y}, 1],\ y[x] > m[b], \\ 1 & \text{if} \quad \forall x \in [1-\overline{y}, 1],\ y[x] < m[b]. \end{cases} \tag{2.6c}$$

Let the step function

$$s[x] = \begin{cases} 0 & \text{for} \quad 0 \leq x \leq a_1, \\ \overline{y}/2 & \text{for} \quad a_1 < x \leq a_2, \\ y_0 & \text{for} \quad a_2 < x < b_1, \\ (1+\overline{y})/2 & \text{for} \quad b_1 \leq x < b_2, \\ 1 & \text{for} \quad b_2 \leq x \leq 1, \end{cases} \tag{2.7}$$

defined by points a_1, a_2, b_1, and b_2 such that s and y have identical means over the four successive intervals $I_1 = [0, a]$, $I_2 = [a, 1-\overline{y}]$, $I_3 = [1-\overline{y}, b]$, and $I_4 = [b, 1-\overline{y}]$, i.e. (see Figure 2.3b)

$$\text{for } i = 1 \text{ to } 4 \quad \int_{I_i} \Big(y[x] - s[x]\Big)\, \mathrm{d}\, x = 0,$$

$$\text{thus} \quad \int_0^1 s[x]\, \mathrm{d}\, x = \int_0^1 y[x]\, \mathrm{d}\, x = \overline{y}, \quad \text{and} \quad s \in \mathcal{M}_{II}^{\leq}(\overline{y}). \tag{2.8}$$

This is always possible from the definition of a and b and as y is an increasing function, and thus $0 \leq a_1 \leq a \leq a_2 \leq b_1 \leq b \leq b_2 \leq 1$. Over each of the four intervals I_i, $y - m$ is of constant sign and bounded as $\Big|y[x] - m[x]\Big| \leq \max_{I_i} \Big|s[x] - m[x]\Big|$. The L^2 integrals can thus be bounded as

$$\begin{aligned} \int_{I_i} \Big(y[x] - m[x]\Big)^2\, \mathrm{d}\, x &\leq \max_{I_i} \Big|s[x] - m[x]\Big| \int_{I_i} \Big|y[x] - m[x]\Big|\, \mathrm{d}\, x \\ &\leq \max_{I_i} \Big|s[x] - m[x]\Big| \int_{I_i} \Big|s[x] - m[x]\Big|\, \mathrm{d}\, x = \int_{I_i} \Big(s[x] - m[x]\Big)^2\, \mathrm{d}\, x. \end{aligned} \tag{2.9}$$

Thus $\|y - m\|_2^2 \leq \|s - m\|_2^2$.

Now, let an edge function $y_{\alpha\beta} \in \mathcal{M}_{II}^{\leq}(\overline{y})$ be defined by its middle value $y_{\alpha\beta}[1 - \overline{y}] = y_0$ and by α such that (see Figure 2.3c)

$$\int_{a_1}^{a_2} \Big(s[x] - y_{\alpha\beta}[x]\Big)\, \mathrm{d}\, x = 0. \tag{2.10}$$

α exists such that $\alpha(1-\overline{y}) \in [a_1, a_2]$ because $0 \leq s[x] = \overline{y}/2 \leq y_0$. $1-\beta\overline{y} \in [b_1, b_2]$ is thus obtained from y_0 and α according to (2.4). As $s[x] = m[x]$ over $[a_1, a_2] \cup [b_1, b_2]$ and s and $y_{\alpha\beta}$ coincide over the complementary domain $[0, a_1] \cup [a_2, b_1] \cup [b_2, 1]$, it is found $\|s - m\|_2^2 \leq \|y_{\alpha\beta} - m\|_2^2$.

Therefore, according to Lemma 1, $\|y - m\|_2^2 \leq \|y_{\alpha\beta} - m\|_2^2 = \overline{y}(1-\overline{y})/4$. The proof also holds for discontinuous y using the left and right limits around the various points defined above. ■

2.2.3 Elliptic coordinates mapping of $\mathcal{M}_{XY}(\overline{y})$

Corollary 1 (Elliptic coordinates mapping of $\mathcal{M}_{XY}(\overline{y})$) *For any $y \in \mathcal{M}_{II}^{\leq}(\overline{y})$,*

$$\left(\frac{\|y - y_{00}\|_2}{\|y_{11} - y_{00}\|_2}\right)^2 + \left(\frac{\|y - y_{11}\|_2}{\|y_{11} - y_{00}\|_2}\right)^2 \leq 1, \tag{2.11}$$

equality being reached if and only if there exists $(\alpha, \beta) \in I^2$ such that $y = y_{\alpha\beta}$. The corollary extends to $\mathcal{M}_{XY}(\overline{y})$ through shifting, scaling, and symmetries.

Proof This results from Lemma 1 and Theorem 1 after elementary algebra, bearing in mind that y_{00}, m, and y_{11} are aligned and form a diameter. ■

(2.11) naturally leads to considering the elliptic coordinates in $\mathcal{M}_{XY}(\overline{y})$ with respect to poles y_{00} and y_{11}: as represented in Figure 2.4, any $y \in \mathcal{M}_{XY}(\overline{y})$ can be mapped (injection) to a point of the half disk of radius $\frac{1}{2}$ bounded by the edge functions $y_{\alpha\beta}$—the mapping is one-to-one only along the y_{00}–y_{11} axis. The half disk is both the 2D axi-symmetric projection of $\mathcal{M}_{XY}(\overline{y})$ for each $\overline{y}$ and a polar projection of the $\mathcal{M}_{XY}(\overline{y})$ for all $\overline{y}$ (which are subsets of parallel hyperplanes of functions). For reference, the classical transformation from elliptic coordinates (d_+, d_-) with poles $(\pm\frac{1}{2}, 0)$ to Cartesian coordinates (x, y) is given by

$$(x - \tfrac{1}{2})^2 = (1 + d_+^2 - d_-^2)^2/4, \tag{2.12a}$$

$$y^2 = d_+^2 d_-^2 - (1 - d_+^2 - d_-^2)^2/4. \tag{2.12b}$$

The elliptic coordinates mapping provides guidance for designing and evaluating general interpolators: it is not restricted to specific profiles such as linear interpolants as mapped on a Sweby diagram [100, fig. 1] and applies whatever the stencil width and order of the scheme. It has some limitations however, as it is for instance not one-to-one, not linear, and worse, not convex: in general, the map of a convex subset of $\mathcal{M}_{II}^{\leq}(\overline{y})$ may not be convex as a segment connecting two functions in $\mathcal{M}_{II}^{\leq}(\overline{y})$ maps as a concave arc of hyperbola in the elliptic coordinates.

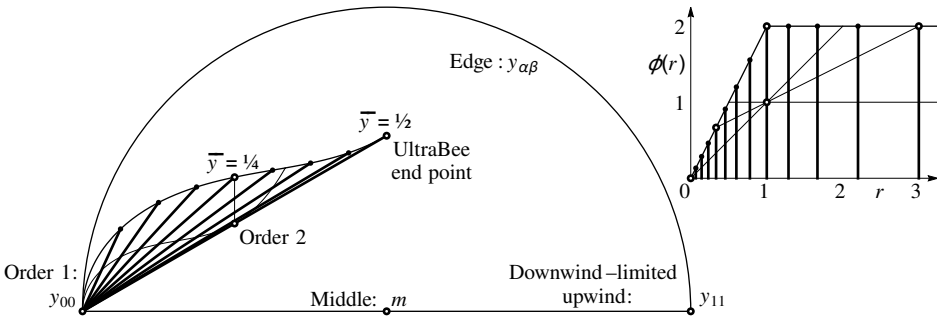


Figure 2.4: 2D map of the set of bounded monotonic functions $\mathcal{M}^{<}_{II}(\overline{y})$ in the elliptic coordinates from poles y_{00} and y_{11}: all y are within the half circle centered on $m = (y_{00} + y_{11})/2$ bounded by $y_{\alpha\beta}$ as stated in Corollary 1. The thick line segments correspond to single-slope interpolants (see subsection 2.3.2) of fixed $\overline{y}$ values ranging from 0 to $\frac{1}{2}$ in steps of $\frac{1}{16}$. The 2nd order point is the interpolant that would result from any second-order scheme over a uniform mesh. The insert is the usual Sweby diagram [100, fig. 1] of these single-slope interpolants which map two-to-one in the half circle (as represented by the corresponding lines and end points).

2.3 Slope-and-bounds interpolants

2.3.1 Background: usual single-slope interpolants

Among all possible interpolants, affine functions obviously play a central role: they are the algebraically simplest to approximate smooth functions to second order, and for given $\overline{y}$, depend on only one parameter, the slope g. They have thus been extensively studied and are at the core of all second-order schemes with numerous possible closures [88, 89, 90, 91, for reviews]. The present work will only consider the four most basic slope definitions usually designated as Min-Mod (MM), Monotonized Centered (MC, also known as Barth–Jespersen), Super-Bee (SB), and Ultra-Bee (UB, also known as Upper-Bound). With the usual shifting, scaling, and symmetrizing on $\mathcal{M}^{<}_{II}(\overline{y})$ but on a non-uniform grid, these are defined

as[2] (remember that $0 < \overline{y} \leq \frac{1}{2}$)

$$g^{\mathrm{MM}}[\overline{y}] = 2\min\left\{\frac{1-\overline{y}}{1+h_1}, \frac{\overline{y}}{1+h_0}\right\} \overset{h_{0,1}=1}{=} \overline{y}, \tag{2.13a}$$

$$g^{\mathrm{MC}}[\overline{y}] = \frac{\frac{1+2h_0}{1+h_1}(1-\overline{y}) + \frac{1+2h_1}{1+h_0}\overline{y}}{1+h_1+h_0} \overset{h_{0,1}=1}{=} \frac{1}{2}, \tag{2.13b}$$

$$g^{\mathrm{SB}}[\overline{y}] = 2\max\left\{\frac{1-\overline{y}}{1+h_1}, \frac{\overline{y}}{1+h_0}\right\} \overset{h_{0,1}=1}{=} 1-\overline{y}, \tag{2.13c}$$

$$g^{\mathrm{UB}}[\overline{y}] = 2\min\Big\{(1-\overline{y}), \overline{y}\Big\} = 2\,\overline{y}. \tag{2.13d}$$

where the left and right cells widths, h_0 and h_1, are scaled by the current cell width, and the expression for the MC slope is justified in Appendix 2.8. All the slopes collapse to the exact second-order interpolant for aligned cell values, at

$$g^{\mathrm{XX}}[\overline{y}] = g^{\mathrm{O2}} = 2/(2+h_1+h_0), \quad \text{for} \quad \overline{y} = \overline{y}^{\mathrm{O2}} = (1+h_0)/(2+h_1+h_0). \tag{2.14}$$

The definitions in (2.13) however, *do not yield I-bounded interpolants* in general and must thus be either limited, corrected, or clipped: following the remarks in subsection 2.1.3, this is how accurate interpolation and monotonicity limitation are here separated.

The traditional approach for limitation of slopes as those in (2.13) basically amounts to the substitution

$$g^{\mathrm{XX\ Lim.}} = \min\Big\{g^{\mathrm{XX}}, g^{\mathrm{UB}}\Big\}. \tag{2.15}$$

As expected the MM slope is unchanged by this limitation step. Now, monotonicity can be achieved more generally by modifying the shape of affine interpolants when required. In principle, higher-degree polynomials—such as parabolic interpolants which are the next most algebraically simple and are usually invoked for third-order accuracy [92, 102]—could be a possibility but they almost always display non-monotonic behavior for some values of $\overline{y}$. Rational [105] or transcendental [106] functions could also be considered.

We here select the simplest non-algebraic extension, whereby single-slope interpolants are merely "clipped" to bounds, thus producing slope-and-bound and slope-and-two-bounds interpolants. These interpolants, as defined below in subsection 2.3.2, are still described by a single parameter (the slope g) and connect to each other in a continuous way.

[2] In these formulae, the threshold value of $\overline{y}$ contained in the min and max functions may differ from $\frac{1}{2}$ for non-uniform grids. As previously noticed [99], Sweby-like diagrams can be centered on either of the monotonicity or second order thresholds. Monotonicity is clearly privileged here.

2.3.2 Slope-and-bounds interpolants

Definition 3 (Slope-and-bounds interpolants) *Single-Slope (SS), Slope-And-Bound (SAB), and Slope-And-Two-Bounds (SAB2) interpolants are defined by the respective functions $y \in \mathcal{M}_{II}^{\leq}(\overline{y})$ which depend on a single parameter $g \in \mathbb{R}^+$ (remember that $0 < \overline{y} \leq \frac{1}{2}$)*

$$y[x] = \begin{cases} \overline{y} + g \times (x - \frac{1}{2}), & \text{for} \quad 0 \leq g < 2\overline{y}, \\ \max\left\{0, \sqrt{2\overline{y}g} + g \times (x-1)\right\}, & \text{for} \quad 2\overline{y} \leq g < 1/(2\overline{y}), \\ \min\left\{1, \max\left\{0, \frac{1}{2} + g \times (x - 1 + \overline{y})\right\}\right\}, & \text{for} \quad 1/(2\overline{y}) \leq g < \infty. \end{cases} \tag{2.16}$$

The definition extends to $\mathcal{M}_{XY}(\overline{y})$ through shifting, scaling, and symmetries.

The slope-and-bounds functions do not aim at being "close" to the $y_{\alpha\beta}$ edge functions, less at being "dense" in $\mathcal{M}_{II}^{\leq}(\overline{y})$, but they continuously connect y_{00} and y_{11} for respective slopes $g = 0$ and $g \to \infty$. Connection to the saturated step function y_{11}—the downwind-limited upwind of Després and Lagoutière [112, 113]—is an especially interesting feature that few interpolants display. Incidentally, the inverses of SS, SAB, and SAB2 functions are themselves SAB2, SAB, and SS functions respectively—thus extending the connection $y_{00}^{-1}[x] = y_{11}[x]$ already noted in subsection 2.2.2.

Proposition 1 (Elliptic coordinates of slope-and-bounds interpolants) *Single-Slope (SS), Slope-And-Bound (SAB), and Slope-And-two-Bounds (SAB2) interpolants have squared elliptic coordinates from poles y_{00} and y_{11} given respectively by (remember that $0 < \overline{y} \leq \frac{1}{2}$)*

$$\|y - y_{00}\|_2^2 = \begin{cases} g^2/12, & \text{for} \quad 0 \leq g < 2\overline{y}, \\ \overline{y}\left(\frac{2}{3}\sqrt{2\overline{y}g} - \overline{y}\right), & \text{for} \quad 2\overline{y} \leq g < 1/(2\overline{y}), \\ \overline{y}(1-\overline{y}) - 1/(6g), & \text{for} \quad 1/(2\overline{y}) \leq g, \end{cases} \tag{2.17a}$$

$$\|y - y_{11}\|_2^2 = \begin{cases} \overline{y}(1-\overline{y})(1-g) + g^2/12, & \text{for} \quad 0 \leq g < 2\overline{y}, \\ \overline{y}\left(1 - \frac{4}{3}\sqrt{2\overline{y}g} + \overline{y}g\right), & \text{for} \quad 2\overline{y} \leq g < 1/(2\overline{y}), \\ 1/(12g), & \text{for} \quad 1/(2\overline{y}) \leq g. \end{cases} \tag{2.17b}$$

The proposition extends to $\mathcal{M}_{XY}(\overline{y})$ through shifting, scaling, and symmetries.

Proof This results from direct integration of (2.16). ■

Applying (2.12) to (2.17), the set of SS functions for any given value of $\overline{y}$ is found to map on the half disk as a straight segment connected to the y_{00} point as visible

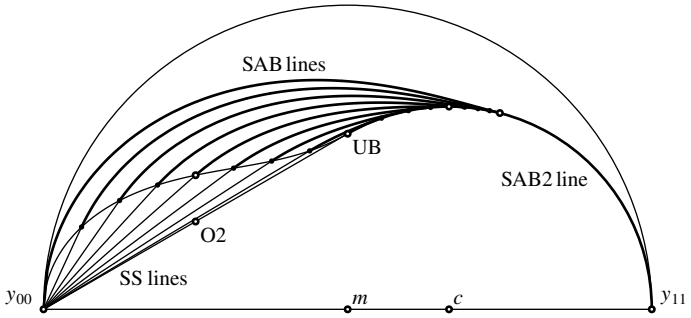


Figure 2.5: 2D map (elliptic coordinates from poles y_{00} and y_{11}) of slope-and-bound and slope-and-two-bounds interpolants (SAB zone, thick lines) for fixed $\overline{y}$ values from 0 to $\frac{1}{2}$ in steps of $\frac{1}{16}$: these extend the corresponding single-slope interpolants of same $\overline{y}$ values (thin lines, see also Figure 2.4) and make a full connection from y_{00} to y_{11} running close to the $y_{\alpha\beta}$ edge.

on Figure 2.4. The segments for various $\overline{y}$ can be put in one-to-one correspondence with vertical lines on the Sweby diagram [100, fig. 1b] (see insert on Figure 2.4) which is traditionally used for mapping SS interpolants and interpolators over coordinates $(r, \phi) = (1/\overline{y} - 1, g/\overline{y})$. The Sweby diagram however, is unsuitable to map any other type of interpolant such as SAB or SAB2. Two canonical points along the $\overline{y} = \frac{1}{2}$ line are marked in both representations of Figure 2.4: the "second-order" (on uniform grid) and "Ultra-Bee" points for respectively $g = \frac{1}{2}$ and 1. For non-uniform grids, the "second-order" point may not coincide with the middle of the first-order to Ultra-Bee segment.

Inspection of Figure 2.4 reveals that single-slope interpolants only sample a small leaf-shaped portion on the map of interpolants, skewed towards the first-order point y_{00} on which they collapse as $\overline{y} \to 0$. This reduced sampling of the space of accessible interpolants is a major theoretical motivation for introducing the slope-and-bounds interpolants of Definition 3.

Applying again (2.12) to (2.17), the set of SAB and SAB2 functions, for any given value of $\overline{y}$, are found to map on the half disk as lines extending the SS segments and connecting to the y_{11} point as visible on Figure 2.5. These lines run typically halfway between the middle function m and the edge functions $y_{\alpha\beta}$. For small values of $\overline{y}$ a very significant expansion of accessible interpolants is thus achieved. However, the SAB2 interpolants will appear unsuitable for practical second-order applications for being incompatible with the doubly monotonic constraint to be discussed in section 2.4.

Incidentally, the reduced elliptic coordinates for SAB2 interpolants deduced from (2.17) depend on a single parameter, the combination $\overline{y}(1 - \overline{y})g$: as visible in Figure 2.5, this collapses the SAB2 lines for any $\overline{y}$ into what can be found to be a

portion of circle of radius $\frac{1}{3}$ centered on $c = \frac{1}{3}y_{00} + \frac{2}{3}y_{11}$.

2.4 Doubly-monotonic SAB interpolators

2.4.1 Motivation: from interpolant to interpolator

The present discussion has been focused so far on some general properties of interpolants, i.e. bounded monotonic functions $y \in \mathcal{M}_{XY}(\overline{y})$. Though useful for general understanding, it does not provide clues on actual prescriptions to build an *interpolator* $z[\overline{y}, x]$, which yields an *interpolant* $y[x] = z[\overline{y}, x]$ for any given $\overline{y}$ value. Only for special values of $\overline{y}$ is it possible to prescribe $y[x]$ in a univocal way: for instance, second-order interpolants must be exact for $\overline{y} = \overline{y}^{\mathrm{O2}}$ as given in (2.14).

The prescription of interpolator formulae, often believed to proceed from well established principles, is actually based on perceived self-evidences or trial-and-error strategies. A common example is the improperly-called "second-order region" in the Sweby diagram (see inset of Figure 2.8) which is in fact posited by constraining the interpolated slope to be a convex combination of left and right slopes [100, § 3 & fig. 1]—as given for instance by the MM and SB in (2.13). Yet, stable second-order TVD interpolators extending beyond this region have actually been considered and successfully used [115].

Here, with the present positivity or monotonicity driven approach, it appears more consistent to constrain interpolators z to be *doubly monotonic*.

2.4.2 Doubly monotonic interpolators

Definition 4 (Doubly monotonic interpolators) *A doubly increasing interpolator is a function z from I^2 to I which is increasing for each of its variables the other being fixed, and verifies $\int z[\overline{y}, x]\,\mathrm{d}\,x = \overline{y}$. Therefore $\forall \overline{y} \in I$, $y[x] = z[\overline{y}, x] \in \mathcal{M}^{\leq}_{II}(\overline{y})$ is an interpolant. The definition extends to $\mathcal{M}_{XY}(\overline{y})$ through shifting, scaling, and symmetries: these extended interpolators remain increasing in the first variable $\overline{y}$ but will be designated as "doubly monotonic."*

Remark 1 *The usual SS limited interpolants such as defined in (2.13, 2.15) are obviously doubly monotonic.*

For prescribing interpolator formulae, this definition may appear somewhat loose, but with complementary constraints it provides an accessible optimal "horizon:" useful interpolators are generally defined by following this horizon. Two complementary constraints of importance will be explored in subsection 2.4.3 as illustrated in Figure 2.6: (i) exact linear profile when $\overline{y}$ is at the second-order aligned value (2.14) and (ii) slope-and-bound profiles in Definition 3. These conditions will

allow for the preservation of compact support—with the ensuing cancellation of numerical wetting.

In principle the doubly monotonic condition with supplementary constraints can be applied to any shape of interpolator but it can lead to very convoluted conditions on z as a function of $\overline{y}$—this is already visible on quadratic functions. As will appear in subsection 2.4.3 however, application of this condition to slope-and-bound interpolators is straightforward as it reduces to conditions at the edges of the slope. In all the following, only doubly-monotonic second-order SAB interpolators will be considered.

2.4.3 Doubly-monotonic second-order SAB interpolators

The principle of double monotonicity is illustrated in figure Figure 2.6 for various SS interpolators on a uniform grid: at $\overline{y} = \frac{1}{2}$ the slope exactly aligns with the neighboring cell values (here assuming uniform mesh) but evolves along different strategies as $\overline{y}$ decreases. Usual approaches, which focus on regularity and adjust the slope to the lowest (MM), mean (MC), or highest (SB) values built from the neighboring cell values (see Figure 2.6a–c), are compatible with double monotonicity but do not take full advantage of it. The lowest and highest slope values are actually constrained by the intersection points of the second-order line with the cell boundaries: this defines two novel interpolators designated here as Min-Mod$^-$ and Super-Bee$^+$

$$g^{\mathrm{MM}-}[\overline{y}] = 2\Big(\overline{y} - \frac{h_0}{2+h_0+h_1}\Big) \overset{h_{0,1}=1}{=} 2\overline{y} - 1/2, \tag{2.18a}$$

$$g^{\mathrm{SB}+}[\overline{y}] = 2\Big(\frac{2+h_0}{2+h_0+h_1} - \overline{y}\Big) \overset{h_{0,1}=1}{=} 3/2 - 2\overline{y}. \tag{2.18b}$$

respectively represented in Figure 2.6f and d.

As $\overline{y}$ keeps diminishing, the steeper interpolators MC, SB, and SB+ eventually reach the boundary values from the neighboring cells. In a pure SS strategy, these interpolators have then no other choice than reducing the slope value all the way to zero at $\overline{y} = 0$ as illustrated in Figure 2.6b–d—this is the usual monotonicity limit of the Sweby diagram. In contrast, the SAB strategy opens many more options of which only two canonical choices will be examined here, designated as "SAB-Balanced" and "SAB-Maximal." As illustrated in respective Figure 2.6g–i and Figure 2.6l–n, the former preserves the *slope* of the interpolant as if at the boundary threshold, and the latter preserves the *intersection point* of the interpolant with the neighboring cell as if at the boundary threshold. Elementary calculations of these conditions let define in a unified way the SS and SAB-extended interpolators as in Definition 3, where g are a functions of $\overline{y}$ given by (remember that $0 < \overline{y} \leq \frac{1}{2}$)

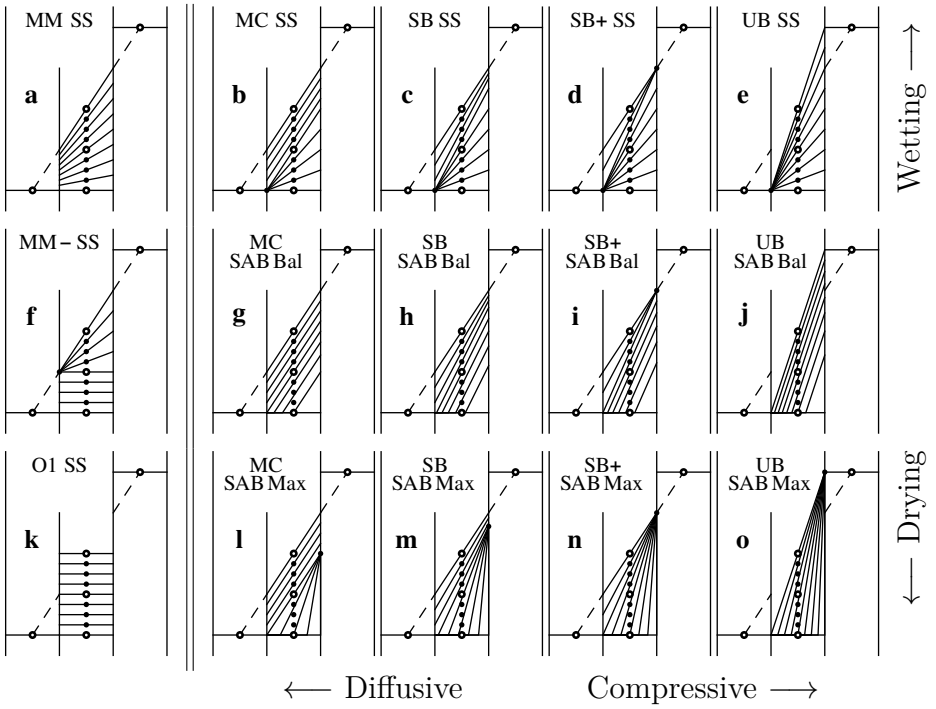


Figure 2.6: Graphical representations of various doubly-monotonic Single-Slope (SS) and Slope-And-Bound (SAB) interpolators on a uniform grid: (a) to (e) usual wetting (SS) interpolators of (2.13) Min-Mod (MM), Monotonized-Centered (MC), Super-Bee (SB), Super-Bee$^+$ (SB+, see description in subsection 2.4.3), and Ultra-Bee (UB); (f) Min-Mod$^-$ interpolator (MM−, see description in subsection 2.4.3); (g) to (j) Mildly wetting Slope-And-Bound (SAB) extensions of the MC to UB interpolators in (b) to (e); (k) Order-one most diffusive interpolator (for reference); (l) to (o) Least wetting Slope-And-Bound (SAB) extensions of the MC to UB interpolators in (b) to (e). All profiles are plotted for $\overline{y}$ values from 0 to $\frac{1}{2}$ in steps of $\frac{1}{16}$. The dashed line marks the exact second-order interpolation at $\overline{y} = \frac{1}{2}$.

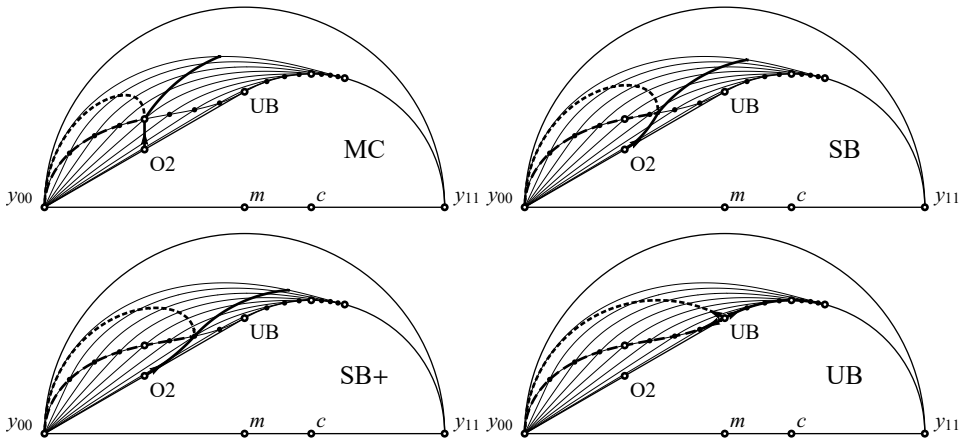


Figure 2.7: 2D maps (elliptic coordinates from poles y_{00} and y_{11}) of the four doubly-monotonic interpolators MC, SB, SB+, and UB (see Figure 2.6) in their usual Single-Slope versions (dashed lines) and Slope-And-Bound extensions, Balanced (dotted lines) and Maximal (solid lines).

$$\begin{cases} \text{if} \quad \overline{y} \geq \overline{y}^{\text{XX}}, \quad \text{then} \quad g^{\text{XX Bal}}[\overline{y}] = g^{\text{XX Max}}[\overline{y}] = g^{\text{XX}}[\overline{y}], \\ \qquad\qquad\qquad\quad\ \ \text{from (2.13, 2.18)}, \\ \text{if} \quad \overline{y} < \overline{y}^{\text{XX}}, \quad \text{then} \quad \begin{cases} g^{\text{XX}}[\overline{y}] & = 2\overline{y}, \\ g^{\text{XX Bal}}[\overline{y}] & = 2\overline{y}^{\text{XX}}, \\ g^{\text{XX Max}}[\overline{y}] & = 2\Big(\overline{y}^{\text{XX}}\Big)^2/\overline{y}, \end{cases} \end{cases} \tag{2.19}$$

with thresholds

$$\overline{y}^{\text{MC}} = \frac{1}{2+h_0+h_1} \overset{h_{0,1}=1}{=} \frac{1}{4}, \tag{2.20a}$$

$$\overline{y}^{\text{SB}} = \frac{1}{2+h_1} \overset{h_{0,1}=1}{=} \frac{1}{3}, \tag{2.20b}$$

$$\overline{y}^{\text{SB+}} = \frac{1+h_0/2}{2+h_0+h_1} \overset{h_{0,1}=1}{=} \frac{3}{8}, \tag{2.20c}$$

$$\overline{y}^{\text{UB}} = \frac{1}{2}. \tag{2.20d}$$

The corresponding trajectories of the interpolators in the elliptic coordinates mapping are represented in Figure 2.7. As expected, the SS and SAB-Max follow opposite paths as $\overline{y} \to 0$, the former reaching the order-one limit at y_{00} and the latter ending close to the UB point. More surprising are the trajectories of the SAB-Bal version which seem to depart from the SS trajectories but eventually reach the y_{00} point.

2.4.4 Mapping of doubly-monotonic-constrained SAB interpolators

The impact of the doubly-monotonic constraint on SAB interpolators can be appreciated through its mapping in elliptic coordinates—and on the Sweby diagram for their SS parts.

Let a doubly-monotonic-constrained SAB interpolator be given by Definition 3, where g is some function of $\overline{y}$, $g[\overline{y}]$. Because an SAB interpolant $y[x] = z[\overline{y}, x]$ is a continuous piecewise linear function of x, double monotonicity for any x is obtained if and only if $\frac{\partial}{\partial \overline{y}} z[\overline{y}, x] \geq 0$ is ensured at all knot points x. According to Definition 3, this translates into conditions on $g[\overline{y}]$

$$0 \leq 2 - g'[\overline{y}] \quad \text{and} \quad 0 \leq 2 + g'[\overline{y}] \quad \text{if} \quad g[\overline{y}] \leq 2\overline{y}, \tag{2.21a}$$

$$0 \leq \Big(\overline{y}/g[\overline{y}]\Big)' \quad \text{and} \quad 0 \leq \Big(g[\overline{y}]/\overline{y}\Big)' \quad \text{if} \quad g[\overline{y}] > 2\overline{y}. \tag{2.21b}$$

Starting from any given $\overline{y}_i$ value, these differential inequalities provide the lower and upper bounding functions

$$g_\pm[\overline{y}_i, \overline{y}] = g_{\pm i} \pm 2(\overline{y} - \overline{y}_i) \quad \text{if} \quad g_\pm[\overline{y}_i, \overline{y}] \leq 2\overline{y}, \tag{2.22a}$$

$$g_\pm[\overline{y}_i, \overline{y}] = g_{\pm i} \times (\overline{y}/\overline{y}_i)^{\pm 1} \quad \text{if} \quad g_\pm[\overline{y}_i, \overline{y}] > 2\overline{y}. \tag{2.22b}$$

which define local cones within which $g[\overline{y}]$ is always constrained.

Lengthy but straightforward calculations provide the elliptic coordinates of $g_\pm[\overline{y}_i, \overline{y}]$ functions (3.55) represented as line bundles in Figure 2.8. The first condition (2.22a) applies to SS interpolators and can also be mapped on the Sweby diagram as bundles of straight lines shown in insert to Figure 2.8. Functions $g_\pm[\overline{y}^{\mathrm{O2}}, \overline{y}]$ limit the area of doubly-monotonic second-order-accessible interpolants, as represented by the shaded region in Figure 2.8 for $\overline{y}^{\mathrm{O2}} = \frac{1}{2}$ (uniform mesh). From their definitions, it thus appears that $g^{\mathrm{MM}-}[\overline{y}] = g_-[\overline{y}^{\mathrm{O2}}, \overline{y}]$ and $g^{\mathrm{SB}+}[\overline{y}] = g_+[\overline{y}^{\mathrm{O2}}, \overline{y}]$. The so-called "second-order region" traditionally defined by the MM and SB interpolators in the Sweby diagram [100, § 3 & fig. 1] (see inset of Figure 2.8) actually appears more restrictive than the doubly-monotonic second-order-accessible zone. However, this latter zone appears to be restricted to the left hand side of the diagram, and it remains an open question to determine the second-order-accessible domain of interpolants if the SAB shape is not imposed.

2.5 Numerical experiments

2.5.1 General test conditions

The three tests consisted in the transport of three standard profiles under a constant and uniform velocity field $u = 1$ over a 1D periodic domain $[0, 1]$ over an

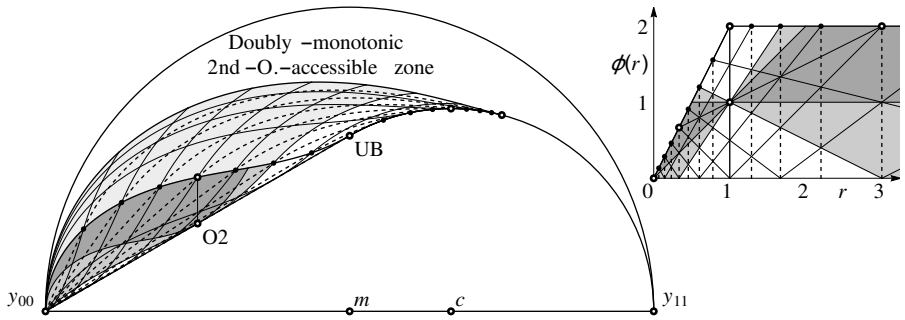


Figure 2.8: 2D map (elliptic coordinates from poles y_{00} and y_{11}) of evenly-sampled limiting cones of double monotonicity (thin lines) defined by slopes $g_\pm[\overline{y}_i, \overline{y}]$ (3.55) which constrain SAB interpolators: when varying $\overline{y}$, the trajectory of a doubly monotonic interpolator must, at any point, stay within the local limiting cone. Dotted lines are SAB interpolants of varying slopes and gien $\overline{y}$ values from 0 to $\frac{1}{2}$ in steps of $\frac{1}{16}$). The dark, medium and light gray areas map the interpolants accessible from the second-order point (for uniform mesh) as given respectively by Sweby [100, fig. 1b], by doubly-monotonic Single-Slope interpolators (SS), and by doubly-monotonic Slope-And-Bound interpolators (SAB) in (2.16). The corresponding curves and regions for the SS interpolators are also mapped on the usual Sweby diagram [100, fig. 1] in insert.

integer number of cycles. These elementary but revealing profiles were selected to benchmark the SAB interpolators over the three main features of interest here: (i) a square wave for step singularities, (ii) a single sine wave for smooth functions, and (iii) the Zalesak combined wave for extrema (as defined in [116, § 8.1]).

Each test was carried out for all the 15 interpolators of Figure 2.6 (defined by (2.13, 2.16, 2.18, 2.19, 2.20)) on a 100 cells mesh, under three numerical settings: two on uniform mesh at Courant numbers $\frac{1}{2}$ and $\frac{1}{10}$, and one with random non-uniform cell widths spanning a 3:1 range and with a maximal Courant number of $\frac{1}{2}$—similar random-mesh tests have been carried out [99, § III.D, fig. 6 & App. B]. This accounted for 45 runs on each profile but depending on relevance, all results are not necessarily reported—tests with the MM, MM− and O1 interpolators were carried out as references, their behavior being already well known.

Data processing was test dependent and is described in the following subsections. No convergence tests were carried out as they fall out of the scope of the present study and their interpretation is highly convoluted for non-linear and singularity-dependent schemes.

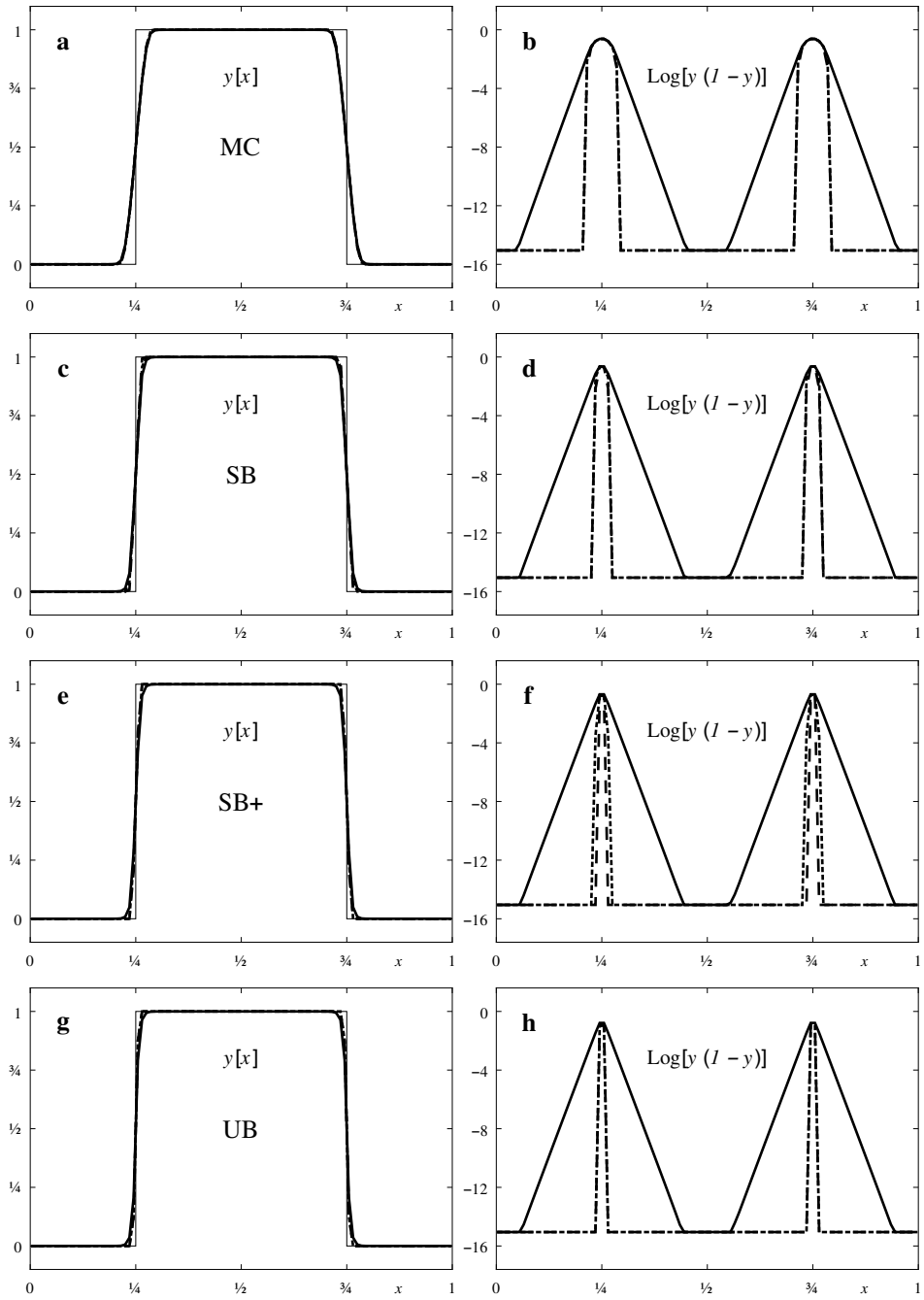


Figure 2.9: Square-wave profiles after uniform cyclic transport over one period of 100 cells at Courant number $\frac{1}{2}$: rows (a–b), (c–d), (e–f), and (g–h) computed with respective "Monotonised Centered," "Super-Bee," "Super-Bee$^+$," and "Ultra-Bee" interpolators (MC, SB, SB+, and UB in corresponding columns of Figure 2.6), each limited as "Single-Slope," "Balanced Slope-and-Bound," and "Maximal Slope-and-Bound" respectively in solid, dotted, and dashed rows (SS, SAB-Bal, and SAB-Max in corresponding rows of Figure 2.6). Initial profile is represented by thin lines in column (a–g). Profile differences between SS, SAB-Bal, and SAB-Max versions appear marginal on standard profiles $y[x]$ in column (a–g) but are revealed in logarithmic scale $\log\big(y[x](1-y[x])\big)$ in column (b–h). Significant numerical wetting is present at similar strengths with all the single-slope interpolators, but it is absent with all the slope-and-bound interpolators.

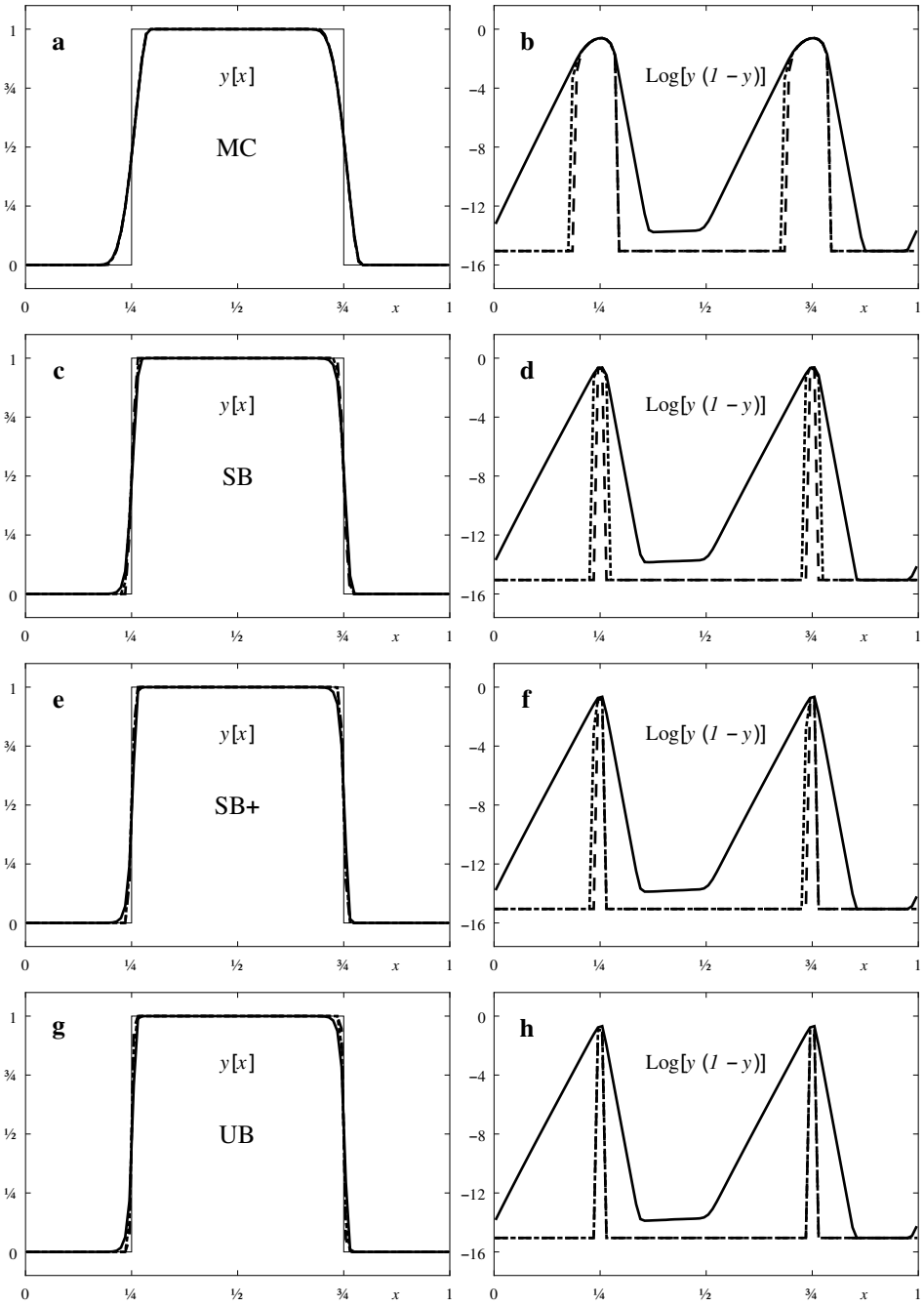


Figure 2.10: Same as Figure 2.9 with Courant number $\frac{1}{10}$: profile asymmetry is now present and numerical diffusion is now higher for the MC interpolators.

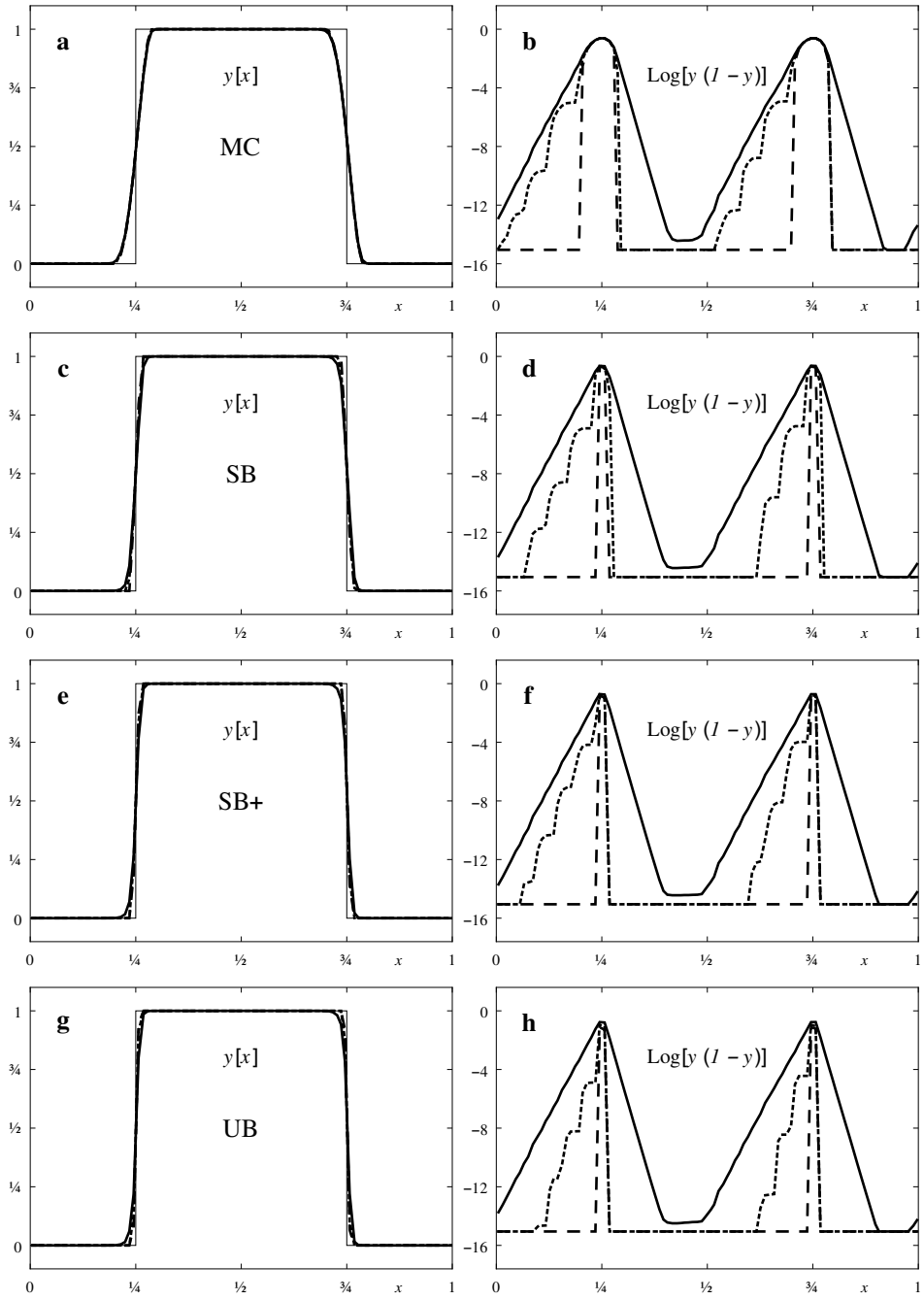


Figure 2.11: Same as Figure 2.9 on a random mesh (1:3 spread of mesh width, Courant number $\frac{1}{2}$ with respect to smallest cell): some profile asymmetry is present as in Figure 2.10 but now, the balanced slope-and-bound interpolators (dotted lines) reduce but do not fully cancel the numerical wetting effect. A form of terracing also appear in the wet zones.

2.5.2 Square-wave tests: singularities, numerical wetting

The initial condition is defined over $[0, 1]$ as

$$y[x] = \begin{cases} 1 & \text{if} \quad \frac{1}{4} \leq x < \frac{3}{4}, \\ 0 & \text{otherwise,} \end{cases} \tag{2.23}$$

and the final time is $t = 1$ (one full cycle). Final $y[x]$ profiles are shown for all the interpolators and the three numerical settings in columns (a–g) of Figures 2.9, 2.10, 2.11. No visually significant differences can be observed between the SS, SAB-Bal, and SAB-Max versions in each of the MC, SB, SB+ and UB families. Front spreading is of course reduced when going from MC to UB.

A more relevant representation for observing numerical wetting is provided by the logarithmic error with respect to the exact solution $\log_{10}\Big(y[x](1-y[x])\Big)$. In order to screen overly noisy and irrelevant small values at the round-off level, the quantity is clipped as

$$\log_{10}\Big(\mathrm{H}\Big[\, y(1-y) - \varepsilon \,\Big]\Big(y(1-y) - \varepsilon\Big) + \varepsilon\Big), \tag{2.24}$$

where H is the Heaviside function and ε is a threshold typically set to the inverse dynamic range of calculations—here $\varepsilon = 2^{-50}$ for double-precision floating-point numbers. Significant differences now appear between the SS, SAB-Bal, and SAB-Max versions as visible in columns (b–h) of Figures 2.9, 2.10, 2.11:

- all the SS interpolators experience spreading of the front support, or numerical wetting, with identical exponential tails of $y[x](1-y[x])$ and some asymmetry between leading and trailing edges when the Courant number departs from $\frac{1}{2}$;
- all the SAB-Max interpolators preserve a compact support of the front, possibly with some asymmetry when the Courant number departs from $\frac{1}{2}$;
- all the SAB-Bal interpolators preserve a compact support of the front for *uniform mesh*, but do experience a partial numerical wetting for non-*uniform mesh* with high asymmetry and some form of "terracing."

Inspection of Figure 2.6 reveals that interpolators can be defined to continuously connect the SS and SAB-Max extremes: some interpolators in between should thus behave in intermediate ways between wetting and non-wetting. The SAB-Bal interpolators appear to display a form of mixed behavior.

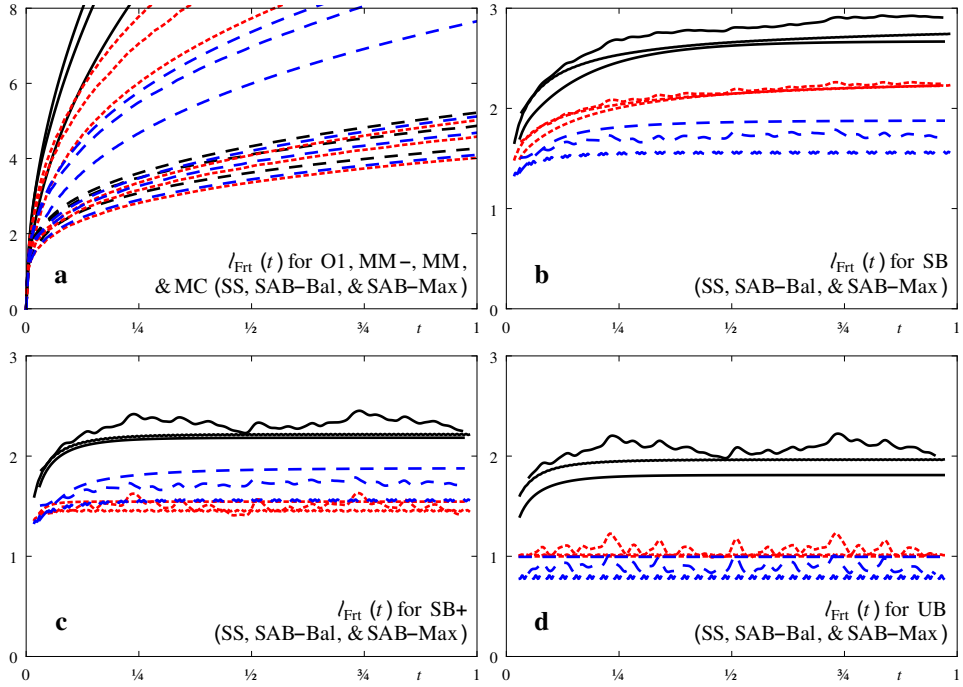


Figure 2.12: Time evolution of mean front widths $\ell_{\mathrm{Frt}}(t)$ (2.25a) in the simulations of Figures 2.9, 2.10, 2.11: (a) diffusive interpolators O1, MM−, and MM (from top towards bottom in respectively solid black, red dotted, and blue dashed lines), and MC (nine lowest lines), (b) SB interpolators, (c) SB+ interpolators, (d) UB interpolators; SS, SAB-BAL, and SAB-Max versions (respectively solid black, red dotted, and blue dashed lines) in each of the MC, SB, SB+, and UB are displayed in (a) to (d); for each interpolator, the three simulation conditions (Courant numbers $\frac{1}{2}$ and $\frac{1}{10}$, and non-uniform mesh) are shown with identical line types. For legibility, time filtering was applied on some of the data sets.

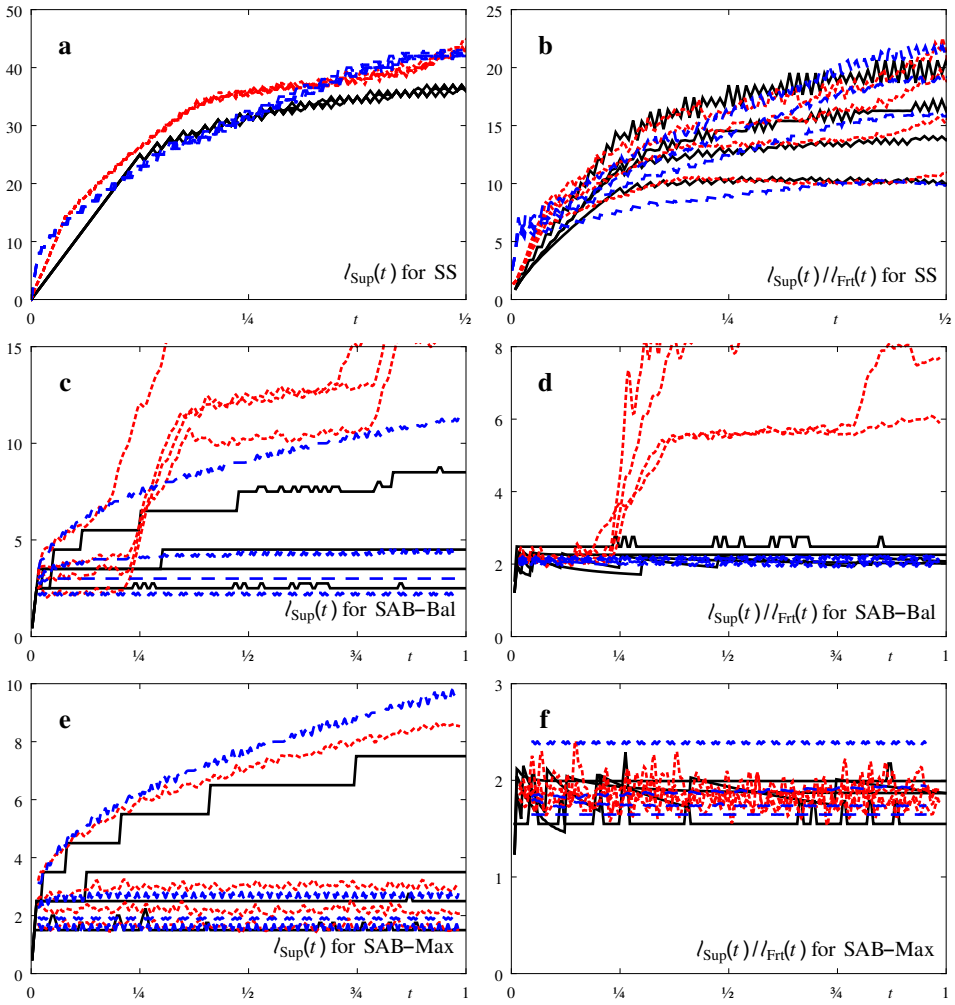


Figure 2.13: Time evolution of front support widths $\ell_{\mathrm{Sup}}(t)$ (2.25b) in mean cell units and in ℓ_{Frt} units in respective columns (a–e) and (b–f) from the simulations of Figures 2.9, 2.10, 2.11: (a–b) usual SS interpolators (notice expanded time scale), (c–d) balanced SAB interpolators, and (e–f) maximal SAB interpolators; Courant numbers of $\frac{1}{2}$ (solid black lines), $\frac{1}{10}$ (blue dashed lines), and $\frac{1}{2}$ on non-uniform mesh (red dotted lines); in each color and line type of each frame, interpolators are MC, SB, SB+, and UB in that order from top to bottom. For legibility, time filtering was applied on some of the data sets.

Quantification of these observations is provided by the mean front width and the front support width, defined here in numbers of mean cell widths

$$\ell_{\rm Frt} = \frac{\sqrt{2}\pi}{2} n_c \Big(\sum_c y_c(1-y_c) h_c \Big) \Big/ \Big(\sum_c h_c \Big), \tag{2.25a}$$

$$\ell_{\rm Sup} = \frac{1}{2} n_c \Big(\sum_c \mathrm{H}\big[y_c(1-y_c) - \varepsilon \big] h_c \Big) \Big/ (\sum_c h_c), \tag{2.25b}$$

where c and h_c are cell labels and cell widths, and n_c is the number of cells (here 100). A factor $1/2$ is present in both (2.25a, 2.25b) to take into account the presence of two fronts over the summation domain and factor $\sqrt{2}\pi$ in (2.25a) is justified in Appendix 2.9.

Time evolution of mean front widths $\ell_{\rm Frt}(t)$ for all the simulations are represented in Figure 2.12. For order-one, MM (standard and MM−), and MC interpolators display respective t^2, t^3, and t^4 late-time growths, partly visible in Figure 2.12a. The t^2 are t^3 diffusion are well known from previous studies of equivalent equations [117, 80], but the t^4 diffusion appears to be approximate and induced by the closeness of a super-convergence effect: when neglecting non-linearities, MC interpolators reduce to Fromm's scheme which is known to be fourth-order super-convergent for Courant number $\frac{1}{2}$ [117, 80, resp. tab. 1 & eq. 20]. In contrast, all the compressive SB, SB+, and UB interpolators in respective Figure 2.12b to d, quickly reach constant but diverse front widths between one and three cells. Numerical conditions (Courant number and mesh regularity) appear to have little impact on $\ell_{\rm Frt}(\infty)$, but in each family the SS, SAB-Bal, and SAB-Max versions produce progressively smaller values by up to 50%.

Time evolution of front support widths $\ell_{\rm Sup}(t)$ for all the simulations are represented in Figure 2.12. Noise and discrete step artifacts are necessarily present due to the inherently thresholded definition of $\ell_{\rm Sup}$ but can be reduced by time filtering. As visible in Figure 2.12a, all the usual SS interpolators produce an initial close-to-linear rapid growth of $\ell_{\rm Sup}$, followed by a progressive slowdown due to the saturation of the computation domain and the round-off truncation at low $y(1-y)$ values. In contrast, all the SAB-Max interpolators produce modest growths of $\ell_{\rm Sup}(t)$ which appear to be constrained by the underlying growths of the mean front widths $\ell_{\rm Frt}$: this is confirmed by the evolution of the $\ell_{\rm Sup}/\ell_{\rm Frt}$ ratio in Figure 2.12f, which is practically constant at about 2 in all cases. For all the standard SS interpolators, the $\ell_{\rm Sup}/\ell_{\rm Frt}$ ratio grows rapidly and can reach (non-universal) values of about 15, see Figure 2.12b. As analyzed above for $\ell_{\rm Frt}$, the SAB-Bal interpolators behave for $\ell_{\rm Sup}$ as the SAB-Max interpolators, except on the non-uniform grid where a SS-like behavior is observed, see Figures 2.12c and d.

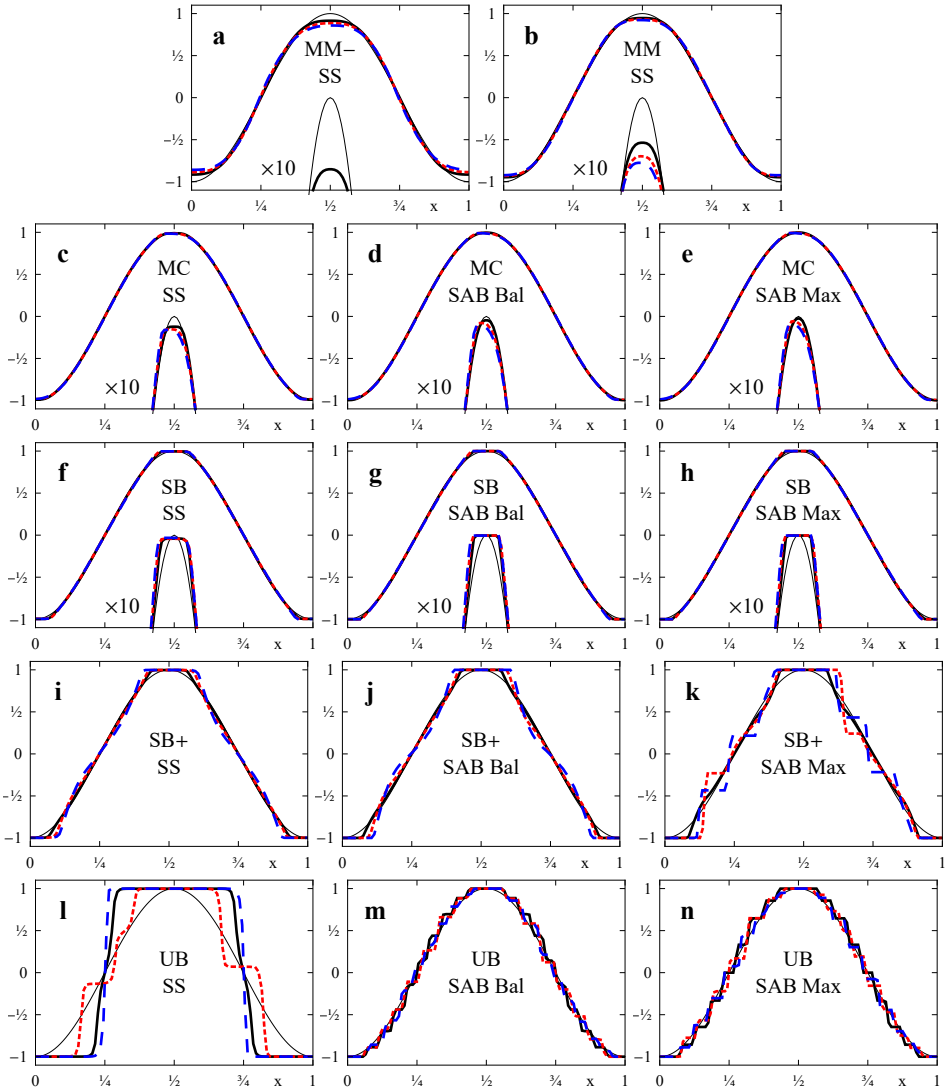


Figure 2.14: Sine-wave profiles after uniform cyclic transport over four periods of 100 cells computed with the interpolators of Figure 2.6 (except the order-one O1) for three numerical conditions in each frame: Courant numbers $\frac{1}{2}$, $\frac{1}{10}$, and $\frac{1}{2}$ on non-uniform mesh (in respectively black solid, red dotted, and blue dashed lines). Initial profile is represented by thin lines and $\times 10$ vertically expanded views of the maximum are provided for interpolators MM, MC, and SB.

2.5.3 Sine-wave tests: erosion, steepening, terracing

The initial condition is defined over $[0, 1]$ as

$$y[x] = -\cos[2\pi x], \tag{2.26}$$

and the final time is $t = 4$ (four cycles). Final $y[x]$ profiles are shown for all the interpolators and the three numerical settings in Figure 2.14. Significant differences are revealed when going from diffusive (MM) to compressive interpolators and from wetting (SS) to non-wetting (SAB-Max):

- Over-compressive interpolators SB+ and UB in Figures 2.14i and l produce "terracing" or "staircasing" of smooth monotonic regions as first observed in [118, fig. 9]. The behavior of the SB+ interpolator Figure 2.14i is in agreement with a previous conjecture [115, § 4 Con. 4.1], whereby slope limiters above the SB limit in the Seby diagram (designated as over-compressive) produce terracing. SAB extensions of SB+ and UB in Figures 2.14i to n appear to worsen this effect.

- Strongly over-compressive interpolators, as here the SAB extensions of UB in Figures 2.14m and n, produce "micro-terracing," an effect previously observed with the most compressive interpolator, the "downwind-limited upwind" y_{11} from [112, 113, resp. figs. 3 & 6].

- Compressive interpolators SB in Figures 2.14f to h, do not produce terracing but steepen slopes and cancel second derivatives almost everywhere, leaving piecewise linear functions but preserving extremal values. SAB extensions do not appear to have impact on this effect.

- Diffusive interpolators MM− and MM in Figures 2.14a and b, produce significant erosion of extrema while keeping smooth but distorted profiles.

- Balanced interpolants MC provide the best trade-off between the erosion and steepening artifacts. The SAB extensions reduce erosion even further.

- Remarkably, all these properties appear to be independent of the numerical conditions of Courant number and mesh uniformity.

Convergence tests have not been attempted: all the interpolators except UB are formally second-order accurate, but accuracy (or lack of it) at late times is dominated by non-linearities and singularities where second-order is necessarily lost. Instead, a simple quantitative measure of erosion is here provided by the reduction of the maximum amplitude of the wave as a function of time. The result in Figure 2.15 for the MC interpolators only show significant improvements from the SAB corrections in all conditions.

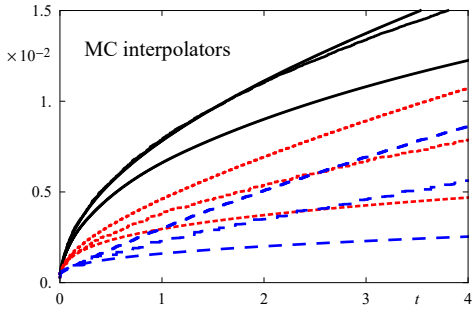


Figure 2.15: Time evolution of the amplitude reduction in the sine-wave tests of Figures 2.14c to e with the MC interpolators in its usual SS, SAB-Bal, and SAB-Max extensions (respectively black solid, red dotted, and blue dashed lines). From top to bottom in each series of same line type numerical conditions are successively Courant number $\frac{1}{10}$, and Courant number $\frac{1}{2}$ with uniform and non-uniform meshes.

2.5.4 Zalesak-wave tests: sharp peaks,

This test was adapted from from [119] in [116, § 8.1]. Following this last reference, the initial condition is defined over $[-1, 1]$ as

$$y[x] = \begin{cases} \exp\left[-\log 2\Big((x + \frac{7}{10})/0.03\Big)^2\right] & \text{if} \quad x \in [-0.8, -0.6], \\ 1 & \text{if} \quad x \in [-0.4, -0.2], \\ 1 - 10\left|x - \frac{1}{10}\right| & \text{if} \quad x \in [0.0, 0.2], \\ \sqrt{1 - 100\,(x - \frac{1}{2})^2} & \text{if} \quad x \in [0.4, 0.6], \\ 0 & \text{otherwise.} \end{cases} \tag{2.27}$$

and the final times are $t = 8$ and 20 (four and ten cycles). Simple sampling of this function is sufficient for second-order investigations (corrections to generate cell values are added for higher order). Other studies may consider different final times [120, 121, 89] in order to enhance departures from the exact solution. Intermediate and final $y[x]$ profiles are shown in Figure 2.16 for the MC interpolator, in the three SS and SAB versions and for Courant numbers $\frac{1}{2}$ and $\frac{1}{10}$ on uniform mesh. Limiters other than MC have not been considered as the induce severe distortions at late times as expected from the sine-wave test results in Figure 2.5.3.

The results in Figure 2.16 appear very similar to those previously reported [89, fig. 5 for instance]: numerical diffusion is more detrimental at Courant number $\frac{1}{10}$ than $\frac{1}{2}$. The impact of SAB extensions appears marginal—cancellation of numerical wetting is not shown—except at maxima were it improves peak amplitudes by up to 10% when numerical diffusion has not yet mixed the peaks (see Figure 2.16c). Amplitudes are less properly captured when peaks are defined by fewer mesh cells.

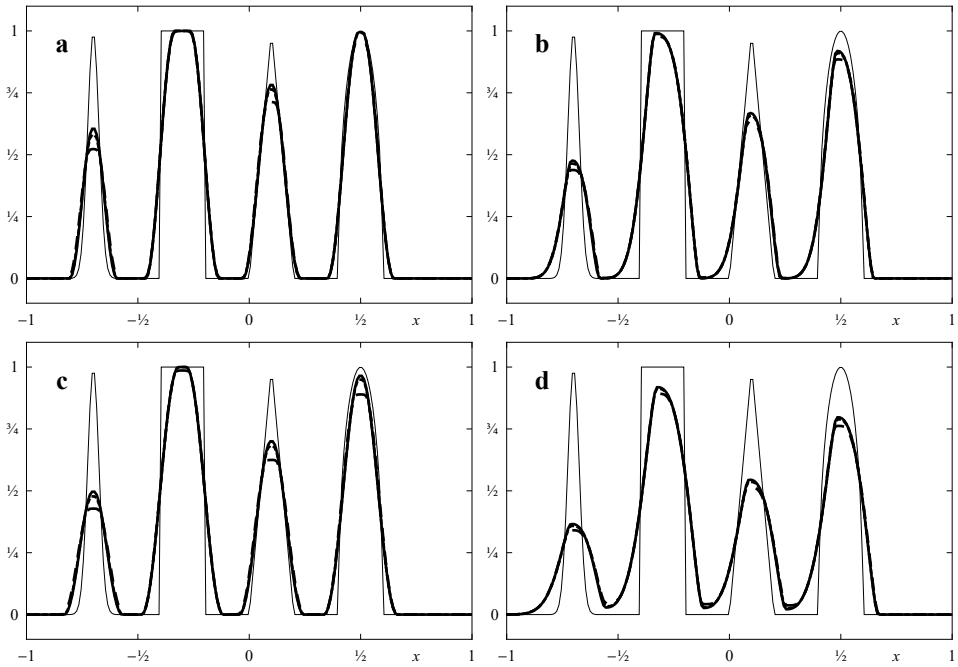


Figure 2.16: Zalesak wave profiles after uniform cyclic transport over four (a–b) and ten (c–d) periods of 200 cells computed with the MC interpolator in its standard SS and modified SAB-Bal and SAB-Max versions (respectively dashed, dotted and solid thick lines) at Courant numbers $\frac{1}{2}$ and $\frac{1}{10}$ (respectively columns (a–c) and (b–d)). Thin line: initial condition.

2.6 Conclusions

This work has provided some insight on the following aspects of monotonic conservative interpolation in the context of MUSCL-like solvers for 1D hyperbolic equations:

- A general topological property of monotonic 1D interpolants provides a new convenient mapping in elliptical coordinates with respect to the two extreme interpolants (see Figures 2.4, 2.5).
- A new constraint, double monotonicity, appears natural for crafting new interpolator formulae.
- According to double monotonicity, the usual second-order interpolators Min-Mod and Super-Bee can be redesigned into more "extreme" versions Min-Mod$^-$ and Super-Bee$^+$ (see Figure 2.6).
- The simple extension of Single-Slope (SS) interpolants into Slope-And-Bounds (SAB) interpolants appears as most natural to comply with monotonicity conditions while widely expanding the accessible points in the elliptical coordinates mapping.
- According to double monotonicity, the Min-Mod, Monotonized-Centered, Super-Bee, and Ultra-Bee interpolators can be extended into SAB versions (see Figure 2.6).
- As confirmed numerically, compact support preservation—here designated as "numerical non-wetting" behavior—is obtained by using SAB extensions of any interpolator MC, SB, SB+, and UB (see Figures 2.9, 2.10, 2.11), with otherwise limited changes in overall behavior.
- Well-known numerical "terracing" and "micro-terracing" effects appear on the SAB extensions of only the "over-compressive" interpolators SB+ and UB (see Figure 2.14).
- Numerical erosion of extrema are also significantly reduced for the SAB-extended MC interpolators compared to the original SS MC (see Figure 2.15).

The new SAB-extended interpolators can bring a simple and efficient option to tackle issues of phase-disappearance in multiphase flows and of wet–dry transition in shallow water flows. Also related although not specifically explored here, interface capture—instead of interface tracking, with smooth interfaces spread over typically up to three cells—could also benefit from this approach.

Some fundamental questions are now open. What is the most general second-order accessible zone under double monotonicity (not constrained by SAB interpolation as in Figure 2.8)? What interpolator features control the wetting–non-wetting transition, as observed between the SAB-Bal and SAB-Max? Though understanding of the terracing phenomenon is still elusive [115, § 4 Con. 4.1], why is it not much affected by the SAB extension of usual SS interpolators?

Higher-dimensional extensions of the non-wetting SAB interpolator approach should be feasible and are being actively pursued.

2.7 Appendix

2.8 Second-order MC slope on non-uniform mesh

A Monotonic Centered (MC) slope definition (2.13b) is always constrained to match the exact second-order value (2.14). However, there is not a unique definition when $\overline{y}$ departs from $\overline{y}^{\mathrm{O2}}$.

The expression retained here is defined as a "best quadratic approximation:" $g^{\mathrm{MC}}[\overline{y}]$ is the mean middle-cell averaged slope of the quadratic polynomial which has average values 0, $\overline{y}$, and 1 on the three cells. Introducing an unknown quadratic polynomial $q[x]$, this definition amounts to

$$g^{\mathrm{MC}}[\overline{y}] = \int_0^1 q'[x]\,\mathrm{d}\,x, \tag{2.28a}$$

$$\overline{y} = \int_0^1 q[x]\,\mathrm{d}\,x, \qquad 0 = \int_{-h_0}^0 q[x]\,\mathrm{d}\,x, \qquad 1 = \int_1^{1+h_1} q[x]\,\mathrm{d}\,x/h_1. \tag{2.28b}$$

Elementary calculations let eliminate $q[x]$ form these expressions and eventually yield (2.13b).

2.9 Front width of an error function profile

An estimate of front width through an integral expression like (2.25a) must be scaled to actually provide a consistent value. This value can be adjusted so as to be exact for some predefined front shape: linear, cubic, sine, hyperbolic tangent, etc. Here the error function shape was retained because it is the exact late-time profile for usual diffusion as produced by order-one interpolators and also because it turns out to be a very close approximation of diffusive order-two front shapes. This was previously observed with the MM interpolator whose analytical front shape can be obtained explicitly [80, § 5.2 & fig. 4]. An rigorous analysis of front shapes for

other second-order interpolators such as MC, though feasible in principle, would be complex and is unavailable at this time.

For an error function profile $y[x] = \frac{1}{2}(1 + \mathrm{erf}[x])$, the front width is estimated as the inverse maximum gradient $\frac{1}{2}\nabla \mathrm{erf}[0]$ at the center of the profile. The $\sqrt{2\pi}$ factor in (2.25a) therefore derives from the identity

$$\tfrac{1}{2}\nabla \mathrm{erf}[0] \int_{-\infty}^{\infty} \tfrac{1}{4}\Big(1 + \mathrm{erf}[x]\Big)\Big(1 - \mathrm{erf}[x]\Big)\,\mathrm{d}\,x = \frac{1}{\sqrt{2\pi}}, \tag{2.29}$$

which is obtained from elementary calculus.

2.10 Convergence study of MC and SB on a Gaussian pulse

Limiter	N	L_1 error	Order	L_2 error	Order	L_∞ error	Order
MC	256	2.31×10^{-4}		5.63×10^{-5}		7.84×10^{-3}	
	512	4.34×10^{-5}		1.02×10^{-5}		2.61×10^{-3}	
	1024	8.29×10^{-6}		1.82×10^{-6}		8.62×10^{-4}	
	2048	1.52×10^{-6}		3.20×10^{-7}		2.83×10^{-4}	
	4096	2.83×10^{-7}		5.54×10^{-8}		9.20×10^{-5}	
	8192	5.21×10^{-8}		9.56×10^{-9}		2.98×10^{-5}	
	16384	9.52×10^{-9}	1.63	1.63×10^{-9}	1.64	9.57×10^{-6}	1.07
MC SAB	256	5.84×10^{-5}		1.02×10^{-5}		1.84×10^{-3}	
	512	9.23×10^{-6}		1.42×10^{-6}		5.10×10^{-4}	
	1024	1.49×10^{-6}		2.20×10^{-7}		1.48×10^{-4}	
	2048	2.39×10^{-7}		3.55×10^{-8}		4.40×10^{-5}	
	4096	3.83×10^{-8}		5.77×10^{-9}		1.31×10^{-5}	
	8192	6.20×10^{-9}		9.38×10^{-10}		3.94×10^{-6}	
	16384	1.02×10^{-9}	1.88	1.53×10^{-10}	2.08	1.18×10^{-6}	1.33
SB	256	1.62×10^{-3}		2.04×10^{-4}		1.73×10^{-2}	
	512	4.64×10^{-4}		4.86×10^{-5}		7.85×10^{-3}	
	1024	1.27×10^{-4}		1.09×10^{-5}		3.53×10^{-3}	
	2048	3.31×10^{-5}		2.38×10^{-6}		1.40×10^{-3}	
	4096	8.45×10^{-6}		5.21×10^{-7}		5.76×10^{-4}	
	8192	2.14×10^{-6}		1.14×10^{-7}		2.34×10^{-4}	
	16384	5.36×10^{-7}	1.15	2.52×10^{-8}	1.4	9.29×10^{-5}	0.68
SB SAB	256	1.64×10^{-3}		2.09×10^{-4}		1.95×10^{-2}	
	512	4.64×10^{-4}		4.87×10^{-5}		8.68×10^{-3}	
	1024	1.27×10^{-4}		1.08×10^{-5}		3.42×10^{-3}	
	2048	3.30×10^{-5}		2.38×10^{-6}		1.49×10^{-3}	
	4096	8.44×10^{-6}		5.20×10^{-7}		5.93×10^{-4}	
	8192	2.13×10^{-6}		1.14×10^{-7}		2.40×10^{-4}	
	16384	5.36×10^{-7}	1.16	2.52×10^{-8}	1.43	9.52×10^{-5}	0.77

Table 2.1: Error analysis and convergence study of the Monotonized-Central and Super-Bee Slope-And-Bound interpolators (MC SAB, SB SAB) compared to their original slope-limiter variants (MC, SB). The computations were performed on the Gaussian pulse $\exp[-100(x-0.5)^2]$ over one period on 100 cells with a CFL number of 0.5.

Part III

Adapting Slope-And-Bound interpolation to multidimensional advection schemes to cancel "numerical wetting"

Chapter 3

Adapting Slope-And-Bound interpolation to multidimensional advection schemes to cancel "numerical wetting"

3.1 Introduction

In the context of multidimensional advection, limiters are needed to avoid spurious oscillations [122, 42, 123, 28]. The main objective of limiters such as ultrabee, superbee and Barth-Jespersen is to steepen gradients in order to catch discontinuities exactly. However, these limiters are too compressive and transform smooth waves into square waves. [68] introduces Slope-And-Bound (SAB) interpolators computed from a given gradient reconstruction and compared to their classical limiter functions (minmod, superbee, monotonized central, ultrabee). The SAB approach was inspired by the downwind-limited upwind flux [124], which preserves functions with a compact support, with a similar behavior as the ultrabee limiter: square waves and "delta" functions are spread over no more than two or three cells respectively. However, so-called *staircasing* is observed for sinusoidal functions. This drawback has been analysed in [68] with a categorisation of 1D second-order interpolator functions, and their influence on *numerical wetting* is discussed, which is defined as the "low-level contamination that spreads linearly in time over all the regions of the computational domain where non-vanishing transport is present." Two applications are mentioned: (i) *phase disappearance* in multiphase or multi-fluid flows, where numerical wetting is present in fluid mixing; (ii) *wet-dry transition* in shallow water flows, where boundary conditions are tricky to handle because of the disappearance of the fluid in cells on the boundary. The main founding of [68]

can be resumed in two results: (i) the steep gradients that were introduced with the SAB interpolators succeed in reducing the numerical wetting in the stencil; (ii) the effect of staircasing is prohibited in the case of monotonized central gradient reconstructions, as opposed to SAB interpolation based on superbee gradients, where this numerical phenomenon is still present.

In the present work the focus lies on slope limitation strategies. Note that slope limiters and flux limiters are equivalent [125]. The maximum principle plays an important role in the developement of slope limiters. Multidimensional *maximum principle regions* have been introduced in [126]. Admissible reconstruction, general maximum principle and positivity preserving properties were discussed in [127, 128]. These principles are discussed in the context of the Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL), which was originally introduced in [42]. Slope limitation is a 1D concept, where gradients are somehow expressed by 1D slopes. Strategies like MUSCL Alternating Direction (AD) [129] circumvent this problem by projecting the multidimensional fluxes via directional splitting. The basic ideas of the MUSCL methods are stated in [130]. An extensive stability analysis is given in [131]. Further studies on that topic include the discussion of monotonic fluxes in the context of discrete maximum principles [132] (see also [133]), limiting with gradient components [134], and moment limiter [135, 136]. A comparison of the major troubles in recent work on flux and slope limiters and a comparison of both strategies is given in [137]. Flux splitting [138] and more general introductions to upwind schemes [139, 140, 29, 76] must be mentioned as well.

SAB interpolation counts in the category of slope limitation. The key idea of SAB interpolation is to limit a chosen gradient reconstruction with clipping via bounds defined by the cell data of neighbors in the stencil. [137] points out that limiting criteria in slope limiter strategies are generally insufficient to ensure the (local) Discrete Maximum Principle (DMP) property, which results in overshoots and undershoots. This topic is discussed for Discontinuous Galerkin (DG) methods in [135]. Inspirations on strategies to ensure the DMP property with an optimal correction factor can be found in [137] and references therein. Furthermore, remapping techniques [141, 142, 143] and multilevel methods [144, 145, 146, 147] are discussed in this context.

Before detailing the present work, some important references on the general topic of Flux Reconstruction (FR) are indicated. FR strategies are used to improve the space discretization of advection equations to high order. Fundamental techniques of FR are recaped in [24], limiter strategies in [125], famous high-order methods on unstructured grids in [148], conical unstructured meshes in [149]. While slope limiters have already been mentioned above, the most famous flux limitation methods are: Flux Corrected Transport (FCT) [25, 26, 27], Actif Flux (AF)

[36, 37, 38, 39] and Residual Distribution (RD) [31, 32, 150, 33, 34]. Recent advances compare and combine these methods [37, 38, 35].

The most popular high-order techniques are (i) refinement strategies of meshes; and (ii) high-order polynomial reconstructions for fluxes [151, 152, 153]. The first is not regarded in the present work, nevertheless two examples should be mentioned anyway: Adaptive Mesh Refinement (AMR) [145] and anisotropic mesh adaptation [154, 155]. The second is discussed with some further examples: In this context one must mention the Multi-dimensional Optimal Order Detection (MOOD) [156, 157], where different types of interpolators are tested simultaniously and flaws on the polynomial reconstruction are detected automatically. Furthermore, Discontinuous Galerkin (DG) methods are widely used nowadays [158, 159, 160, 161], numerous adaptations are possible, such as a posteriori subcell limiting techniques [162], or Discontinuous Galerkin Spectral Element Methods (DGSEM) [163]. [30] discusses a stability analysis in nonconstant coefficient advection for DG schemes in aerodynamics applications. Essentially Non-Oscillatory (ENO) [164] and Weighted Essentially Non-Oscillatory (WENO) [165] are one of the most famous high-order reconstruction and interpolation strategies.

The goal of the present work is to introduce a multidimensional extension of the Slope-And-Bound (SAB) limitation, that does not use directional splitting, but fully mimicks a multidimensional gradient before applying a 1D limiting procedure. In this context the 1D reference frame is introduced, which defines the framework requirement for 1D SAB limitation. Although this method differs from usual flux reconstruction strategies, the main problem — common to most slope limiter methods — still remains: the resulting reconstructed flux must be retransformed to the multidimensional setup. The present work mimicks the monotonized central gradient in the multidimensional setup with a weighted least-squares gradient reconstruction. Reconstruction of least-squares gradients can be found in [166] and references therein. Furthermore, a method of higher-order least-squares reconstructions is presented in [167]. In order to give a proof of concept, the formalism is restricted to second order in this paper. Higher order extensions could for instance use the Piecewiese Parabolic Method (PPM) [168]. In the context of the GEEC scheme [53], which is a quasi-symplectic, multi-fluid direct-ALE approach, the advection operator must be as simple as possible and must at the same time correct numerical wetting, which is a major artifact of hydrocodes. This work extends the GEEC advection operator to second order and is compatible with the co-mesh approach [69], which corrects the grid orientation effect.

The core part of this work is the ND-SAB algorithm which is introduced step by step in chapter 3.2 and summarized in chapter 3.2.5. Each section focuses on a particular technique: 3.2.1 states a weighted least-squares approximation

that fits best the stencil; 3.2.2 introduces the 1D reference frame that mimicks the multidimensional gradient with a 1D slope and aggregates the upwind and downwind cell data; 3.2.3 applies the 1D slope-and-bound (SAB) limitation strategy [68] to the 1D reference frame and the total upwind flux of each cell is computed as the integral of the resulting SAB interpolator function over a certain flux domain; 3.2.4 redistributes the total flux to the face fluxes of the multidimensional setting; Several redistribution weighting factors are introduced for which results are shown in chapter 3.3. Chapter 3.2.5 summarizes the algorithm combining these techniques. Furthermore, the present second-order reconstruction strategy is compatible with the co-mesh approach [69], which corrects the grid orientation effect. Results are shown to support this claim for both the 2D SAB algorithm and the co-mesh approach in chapter 3.3.

Notation

a	advected field,
$a(\boldsymbol{x})$	interpolator function of the advected field,
$\boldsymbol{u}$	velocity in the reference frame (ALE context: $\boldsymbol{u} = \boldsymbol{v} + \boldsymbol{w}$),
$\boldsymbol{v}$	relative-to-grid velocity,
$\boldsymbol{w}$	grid velocity,
cd	oriented index from cell c to cell d — also designates face between cells c and d,
ϕ, ϕ_{cd}	total (outgoing) flux, and face flux,
$\mathcal{V}_{cd}$	flux support,
$\hat{\boldsymbol{x}}_{cd}$	centroid of flux support,
$\hat{a}_{cd}$	central value on $\hat{\boldsymbol{x}}_{cd}$,
$\mathring{V}_{cd}$	volume transfer rate,
σ_{cd}	upwind coefficient,
$\boldsymbol{s}_{cd}$	outward normal vector,
V_c	cell volume,
$\alpha(\xi)$	1D reduction of the interpolator function $a(\boldsymbol{x})$,
	$\alpha(\xi) = \alpha_1 \xi + \alpha_0$ (with slope α_1 and its affine shift α_0),
$\Delta\xi_c$	flux support in 1D reduction,
$-$, $+$	cell indices in 1D reduction (of donor cell c_- and receiver cell c_+),
α_-, α_+	1D reduction of neighboring fields,
a_c, α_c	central values (in multi-D and 1D) on cell centroid,
$a_c^\pm$, $\alpha_c^\pm$	monotonicity thresholds (in multi-D and 1D),
$\alpha^{\star,\pm}$	Slope-And-Bound limitation coefficient,
$\mathbb{H}(x)$	Heaviside function ($\mathbb{H}(0) = 1/2$ is assumed), $\mathbb{H}(x) = \frac{1}{2}(1 + \mathrm{sign}(x))$,
$\mathbb{S}_{[a,b]}(x)$	clipping function, $\mathbb{S}_{[a,b]}(x) = \max\{a, \min\{x, b\}\}$.

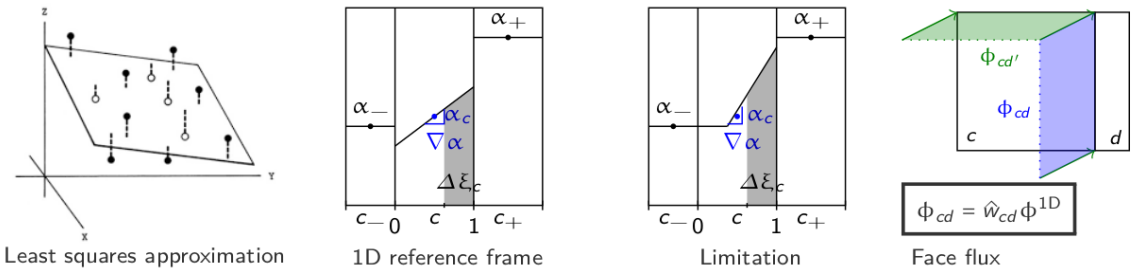


Figure 3.1: Illustration of ND-SAB algorithm in four steps.

3.2 Multidimensional Slope-And-Bound reconstruction algorithm

The present work introduces a multidimensional extension of the Slope-And-Bound (SAB) limitation strategy named ND-SAB, as an alternative to classical MUSCL techniques for advection problems. The ND-SAB algorithm (illustrated in Figure 3.1) goes as follows in four steps: (i) a gradient reconstruction technique is chosen (the present work considers a 2D least squares approximation, section 3.2.1), (ii) the multidimensional gradients are reduced to a 1D representation called the *1D reference frame* (section 3.2.2), (iii) limitation takes place in this *1D reference frame*, which then defines the total outflow of the cell called *1D flux* (section 3.2.3), (iv) this total flux of cell c is then redistributed to the face fluxes of the upwind neighboring cells (section 3.2.4).

Mathematically, the advection equation for a conserved quantity — expressed here by the scalar field a — writes

$$\partial_t a + \boldsymbol{\nabla} \cdot (a\boldsymbol{u}) = 0. \tag{3.1}$$

In this continuity equation, $\boldsymbol{u}$ is the velocity field and $\boldsymbol{\nabla}\cdot$ the divergence operator. [69] introduces the generic discrete explicit upwind Eulerian advection operator in a first-order description. Using a similar notation, the basic MUSCL scheme that solves equation (3.1) writes

$$V_c^{n+1} a_c^{n+1} - V_c^n a_c^n + \sum_d \left(\phi_{cd}^n(\Delta t^n) - \phi_{dc}^n(\Delta t^n)\right) = 0, \tag{3.2}$$

where $\phi_{cd}(\Delta t)$ represents the flux advected in time Δt from cell c towards cell d, V_c is the volume of cell c and $\mathcal{D}(c)$ the set of neighboring cells defined by the scheme's stencil. Defining a high-order expression of the advection flux means giving a high-order description of the following integral

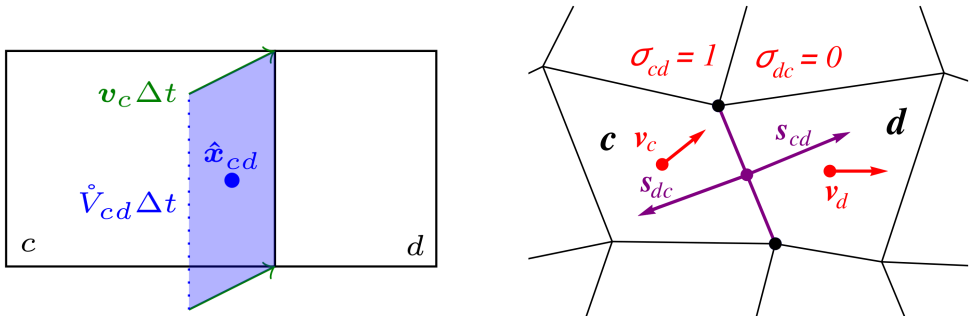


Figure 3.2: Illustrations of geometrical variables: centroid $\hat{\boldsymbol{x}}_{cd}$ of flux support $\mathcal{V}_{cd} = \mathring{V}_{cd}\Delta t$ (left), outward normal vector $\boldsymbol{s}_{cd}$ and upwind coefficient σ_{cd} (right).

$$\phi_{cd}^n(\Delta t^n) = \int_{\mathcal{V}_{cd}^n} a_c^n(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}, \tag{3.3}$$

with $\mathcal{V}_{cd}^n$ the flux support and $a_c^n(\boldsymbol{x})$ the reconstructed field function on cell c at time index n. Let (3.3) be the GEEC upwind flux [169, 53], then the flux support writes

$$\mathcal{V}_{cd}^n = \mathring{V}_{cd}^n \Delta t^n, \tag{3.4a}$$
$$\mathring{V}_{cd}^n := \sigma_{cd}^n \boldsymbol{s}_{cd}^n \cdot \boldsymbol{v}_c^n, \tag{3.4b}$$
$$\sigma_{cd}^n := \mathbb{H}(\boldsymbol{s}_{cd}^n \cdot \boldsymbol{v}_c^n), \tag{3.4c}$$

where $\mathring{V}_{cd}$ is geometrically defined by the upwind coefficient σ_{cd}, with the Heaviside function $\mathbb{H}$, the outward normal vector $\boldsymbol{s}_{cd}$ and the relative-to-grid velocity $\boldsymbol{v}_c$ (see illustration on Figure 3.2). As described in the GEEC formalism, this choice for the flux support is suitable for direct-ALE multi-fluid flow. The Arbitrary Lagrangian Eulerian (ALE) framework introduces the relative-to-grid velocity $\boldsymbol{v}$ and grid velocity $\boldsymbol{w}$, with $\boldsymbol{u} = \boldsymbol{v} + \boldsymbol{w}$.

Consider a classical second-order reconstruction strategy. The reconstructed field is expressed as a linear function given by the first-order Taylor expansion at point $\boldsymbol{x}_c^n$

$$a_c^n(\boldsymbol{x}) = a_c^n + (\boldsymbol{\nabla} a)_c^n \cdot (\boldsymbol{x} - \boldsymbol{x}_c^n). \tag{3.5}$$

For traditional slope reconstructions the integral (3.3) can be simplified with (3.5) and the total flux of cell c can be written as a function of Δt^n.

$$\phi_c^n(\Delta t^n) = \sum_d \phi_{cd}^n(\Delta t^n) = \sum_d \hat{a}_{cd}^n \mathring{V}_{cd}^n \Delta t^n, \tag{3.6a}$$
$$\phi_{cd}^n = \hat{a}_{cd}^n \mathring{V}_{cd}^n \Delta t^n, \tag{3.6b}$$

where $\hat{a}_{cd}^n = a_c^n(\hat{\boldsymbol{x}}_{cd}^n)$ is the reconstructed value on the centroid of the flux domain $\mathring{V}_{cd}^n \Delta t^n$. The coordinates of the centroid $\hat{\boldsymbol{x}}_{cd}^n$ can be computed through the nodes of the polygonal flux domain. In the case of a constant velocity $\boldsymbol{v}^n$ the flux domain is a parallelepiped (as illustrated in Figure 3.2) and $\hat{\boldsymbol{x}}_{cd}^n$ can be deduced directly from $\boldsymbol{x}_{cd}$ — being the center of face cd — as follows

$$\hat{\boldsymbol{x}}_{cd}^n = \boldsymbol{x}_{cd}^n - \frac{\boldsymbol{v}^n \Delta t^n}{2}. \tag{3.7}$$

At this point, so-called *non-limited slopes* are introduced with (3.5), for which the discrete advection equation (3.1) is not TVD. In order to be TVD, limitation must be added. Standard limitation strategies on multidimensional gradients $(\boldsymbol{\nabla} a)_c$ are discussed in [125, 130] and references therein. In the present work the flux integral (3.3) is computed more explicitly as the integral of a SAB interpolator over the flux domain (explanations are given in chapter 3.2.3), *i.e.* the resulting flux is not necessarily of the form (3.6b).

To facilitate writing, the time index n is omitted from now on: consider all field variables given at time t^n. Let it be emphasized that the notation stays generic this way and amenable to any time-discretization (such as the leap-frog formulation in GEEC).

3.2.1 Linear reconstruction of gradients with weighted least squares

Linear reconstructions in multiple dimensions are commonly computed by a weighted least squares (LS) approximation [166]. The linear approximation (3.5) that best fits the neighboring cells is given by

$$(\boldsymbol{\nabla} a)_c = \arg\min_{\boldsymbol{\nabla} a} \sum_d w_{cd} \left((\boldsymbol{\nabla} a)_c \cdot \delta \boldsymbol{x}_{cd} + a_c - a_d \right)^2, \tag{3.8}$$

where w_{cd} is the weighting factor of the least squares problem, which is explained in this chapter below for the 1D and 2D cases.

Solving the extremal problem (3.8) leads to

$$(\boldsymbol{\nabla} a)_c = M_c^{-1} \cdot \sum_d w_{cd} \delta \boldsymbol{x}_{cd} (a_d - a_c), \tag{3.9a}$$

$$M_c = \sum_d w_{cd} \delta \boldsymbol{x}_{cd} \otimes \delta \boldsymbol{x}_{cd}. \tag{3.9b}$$

Gradients on a non-uniform grid are usually computed as functions defined by linear factors, which depend on the cell dimensions of the considered cells, see for instance [125]. The present approach aims to mimick the second-order gradient reconstructions via second-order polynomials and defines the weighting factors w_{cd} of the present least-squares gradients $(\boldsymbol{\nabla} a)_c$.

Second-order accurate weighting factor in 1D

The main idea here is to deduce the weighting factor of a second-order accurate gradient reconstruction as linear factors that define the gradient of a second-order polynomial. Computations following this strategy are given in Appendix 3.5.1 and the following 1D weights are obtained

$$w_{\pm} = \frac{h_{\mp} + h_c/2}{h_{\pm} + h_{\mp} + h_c}, \tag{3.10}$$

where, the indices $\pm$ designate the right and left neighbors in the stencil, h is the 1D volume. Furthermore, the 1D gradient g writes

$$g = w_- g_+ + w_+ g_-. \tag{3.11}$$

In the present work (3.11) is seen as an extension of the monotonized central gradient [42] to non-uniform meshes.

Second-order accurate weighting factor in multi-D

The multidimensional weighting factor is deduced by mimicking the 1D gradient (3.11). With $\delta\boldsymbol{x}_{c\pm} = \pm\frac{1}{2}(h_{\pm} + h_c)$ gradient $(\boldsymbol{\nabla} a)_c$ (3.9a) projected to 1D writes

$$(\boldsymbol{\nabla} a)_c\Big|_{\text{1D}} = M_c^{-1}\left(\frac{1}{2}w_{c+}(h_+ + h_c)(a_+ - a_c) - \frac{1}{2}w_{c-}(h_- + h_c)(a_- - a_c)\right), \tag{3.12a}$$

$$M_c = w_{c+}\frac{(h_+ + h_c)^2}{4} + w_{c-}\frac{(h_- + h_c)^2}{4}. \tag{3.12b}$$

Identifying (3.12a) with (3.11) leads to

$$w_{c\pm} = \frac{h_{\mp} + \frac{h_c}{2}}{\frac{1}{4}(h_{\pm} + h_c)^2}. \tag{3.13}$$

The multidimensional weighting factor w_{cd} that corresponds to $w_{c\pm}$ in 1D is not unique.

$$w_{cd} = \frac{\sum\limits_{d' \neq d} V_{d'} + \dfrac{V_c}{\#\mathcal{D}(c)}}{\delta\boldsymbol{x}_{cd}^2}, \qquad d \in \mathcal{D}(c). \tag{3.14}$$

This choice here uses the number of neighbors $\#\mathcal{D}(c)$ in the stencil around cell c as a factor in order to include unstructured grids with an arbitrary number of neighboring cells.

Comparing (3.10) and (3.13) we observe that both weights (which are deduced from a parabolic function), somehow include the geometry of the other cells in the stencil.

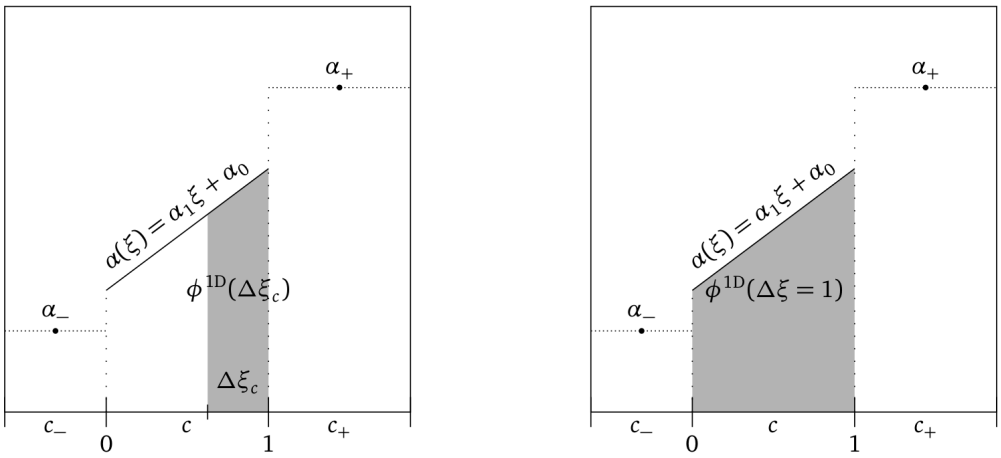


Figure 3.3: Illustration of fluxes $\phi^{1D}(\Delta\xi_c)$ and $\phi^{1D}(\Delta\xi = 1)$ in the 1D reference frame over cell c.

3.2.2 Monotonicity analysis in multi-D to 1D reduction

It is a delicate problem to define rules to limit fluxes and gradients in a multidimensional setup. The goal of this chapter is to carefully construct a 1D representation of the numerical flux, which is linked to its multidimensional data and to define the monotonicity thresholds therein, which assure the framework requirement for Slope-And-Bound (SAB) interpolators described in the next chapter.

This chapter is divided in two paragraphs. First, the 1D reference frame is constructed, second, the monotonicity thresholds are defined therein.

Construction of 1D reference frame

The 1D reference frame mimicks the multidimensional data and in a 1D representation: It defines the redimensioning of the linear approximation $a_c(\boldsymbol{x})$ on cell c to a 1D analogon on interval $[0, 1]$.

$$\alpha(\xi) = \alpha_1\xi + \alpha_0, \tag{3.15}$$

with slope coefficient α_1 and affine shift α_0.

The analogon of (3.6a) in the 1D reference frame is expressed as a function of the flux support $\Delta\xi = \xi_1 - \xi_0$ and writes

$$\phi_c^{1D}(\Delta\xi) = \int_{\xi_0}^{\xi_1} \alpha(\xi)\ \mathrm{d}\xi = \left[\frac{1}{2}\alpha_1\xi^2 + \alpha_0\xi\right]_{\xi_0}^{\xi_1}. \tag{3.16}$$

Let $(\Delta\xi)_c$ be the flux support at time lapse Δt on $[0,1]$, which is resized by the volume size V_c and writes

$$(\Delta\xi)_c = \frac{\Delta t \sum_d \mathring{V}_{cd}}{V_c}. \tag{3.17}$$

Furthermore, consider a certain time step Δt_1 describing the time lapse for which the volume of the total flux domain coincides with the volume V_c, *i.e.* for which $\Delta t \sum_d \mathring{V}_{cd} = V_c$. In other words

$$\Delta t_1 = V_c / \sum_d \mathring{V}_{cd}, \tag{3.18}$$

describes the time lapse that is needed for emptying cell c.

Now, the next part of the construction of the 1D reference frame is to determine the coefficients α_0 and α_1 of (3.15): They can be deduced by identifying the multidimensional second-order upwind flux (3.6a) and its reduction in the 1D reference frame (3.16) at the two specific time lapses Δt and Δt_1, which define the effectively advected upwind flux, and the maximal possible flux. This is described with the linear system

$$\phi_c\Big(\Delta t\Big) = \phi_c^{\mathrm{1D}}(\Delta\xi_c), \qquad \text{with } \xi_0 = 1 - \Delta\xi_c,\ \xi_1 = 1, \tag{3.19a}$$

$$\phi_c\Big(\Delta t_1\Big) = \phi_c^{\mathrm{1D}}\Big(\Delta\xi = 1\Big), \qquad \text{with } \xi_0 = 0,\ \xi_1 = 1, \tag{3.19b}$$

Figure 3.3 illustrates the 1D reference frame with the effective upwind flux $\phi_c^{\mathrm{1D}}(\Delta\xi_c)$ and the maximal flux $\phi_c^{\mathrm{1D}}(\Delta\xi = 1)$.

Solving the linear system (3.19) leads to

$$\alpha_0 = \phi_c(\Delta t_1) - \frac{1}{2}\alpha_1, \tag{3.20a}$$

$$\alpha_1 = \frac{2(\phi_c(\Delta t) - \phi_c(\Delta t_1)\Delta\xi_c)}{\Delta\xi_c(1 - \Delta\xi_c)}. \tag{3.20b}$$

With (3.15) the central value writes

$$\alpha_c = \alpha\left(\xi = \frac{1}{2}\right) = \alpha_0 + \frac{1}{2}\alpha_1. \tag{3.21}$$

Furthermore, the field data $\alpha_\pm$ of the neighboring cells $c_\pm$ in the 1D reference frame is computed as averaged values aggregating donor cells and receiver cells

$$\alpha_+ = \sum_d \frac{\mathring{V}_{cd}}{\sum_d \mathring{V}_{cd}}\alpha_d, \tag{3.22a}$$

$$\alpha_- = \sum_d \frac{\mathring{V}_{dc}}{\sum_d \mathring{V}_{dc}}\alpha_d, \tag{3.22b}$$

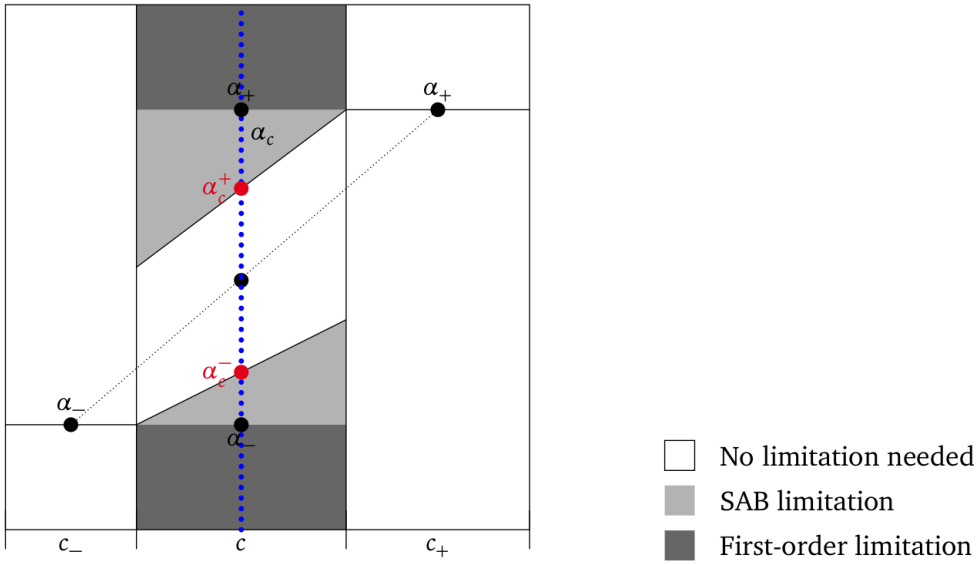


Figure 3.4: Zones beyond the monotonicity thresholds $\alpha_c^{\pm}$ (red) and the TVD thresholds $\alpha_{\pm}$ indicating where limitation is needed for different central values (blue).

where index + designates the aggregation of upwind neighboring cells (illustrated in Figure 3.3 as the *right* neighbor) and − designates the aggregation of downwind neighboring cells (illustrated in Figure 3.3 as the *left* neighbor). Thus, the upwind flux goes from cell c to cell c_+.

Monotonicity thresholds in 1D reference frame

The monotonicity thresholds define the limit at which reconstructed gradients must be limited in order to obtain a monotonic scheme. The difficulty in the 1D reference frame is the lack of knowledge about the respective sizes $h_{\pm}$ of the left and right neighboring cells, which are needed to compute traditional limitation strategies [125] and the SAB interpolators [68]. The present work uses a strategy that mimicks the monotonicity thresholds in the 1D reference frame. Computations are given in this chapter.

Figure 3.4 illustrates the monotonicity and TVD thresholds for a given linear function in the 1D reference frame. Beyond these thresholds, limitation is needed to prevent spurious oscillations, which is where the classical limiter function act. The simplest way to do that is to truncate the gradient to zero, which is usually done beyond the TVD threshold, whereas between the monotonicity and the TVD threshold non-zero gradients are desired. In the present work, this is called the

SAB zone, and defines the zone where SAB interpolators are defined (see next section 3.2.3).

Let $a_c^\pm$ be the monotonicity thresholds in the multidimensional representation (with $\boldsymbol{x}_c$ the centroid) and $\alpha_c^\pm$ the monotonicity thresholds in the 1D reference frame. The linear functions analogon to (3.5) and (3.15) on the monotonicity thresholds write

$$a^\pm(\boldsymbol{x}) = a_c^\pm + (\boldsymbol{\nabla} a)_c^\pm \cdot (\boldsymbol{x} - \boldsymbol{x}_c), \tag{3.23a}$$

$$\alpha^\pm(\xi) = \alpha_1^\pm \xi + \alpha_0^\pm. \tag{3.23b}$$

Recall that the reference frame is defined on the intervall $[0, 1]$. Thus, the 1D monotonicity thresholds $\alpha_c^\pm$ are defined at $\xi = \frac{1}{2}$ and only require $\alpha_1^\pm$, $\alpha_0^\pm$.

$$\alpha_c^\pm = \alpha^\pm\left(\xi = \frac{1}{2}\right) = \frac{1}{2}\alpha_1^\pm + \alpha_0^\pm. \tag{3.24}$$

Assume that $\alpha_1^\pm$, $\alpha_0^\pm$ are defined by equations (3.19), which describe two linear systems — on the lower and upper monotonicity thresholds. The threshold fluxes write (analogously to (3.6a) and (3.16))

$$\phi_c^\pm(\Delta t) = \sum_d \left(a_c^\pm + (\boldsymbol{\nabla} a)_c^\pm \cdot (\hat{\boldsymbol{x}}_{cd}(\Delta t) - \boldsymbol{x}_c)\right) \mathring{V}_{cd} \Delta t, \tag{3.25a}$$

$$\phi_c^{1\mathrm{D},\pm}(\Delta\xi) = \int_{\xi_0}^{\xi_1} \alpha^\pm(\xi)\, \mathrm{d}\xi = \left[\frac{1}{2}\alpha_1^\pm \xi^2 + \alpha_0^\pm \xi\right]_{\xi_0}^{\xi_1}. \tag{3.25b}$$

However, the field variables $a_c^\pm, (\boldsymbol{\nabla} a)_c^\pm$ on the monotonicity thresholds are unknown. This means some further information is needed to approximate the associated gradients: the 1D monotonicity threshold slopes are delimited by the left and right values $\alpha_\pm$ given in (3.22). This is illustrated in figure 3.4 and the affine shifts $\alpha_0^\pm$ (defined at $\xi = 0$) can mathematically be written as

$$\alpha_0^\pm = \alpha_\pm - \frac{1}{2}\alpha_1^\pm (1 \mp 1). \tag{3.26}$$

Hence, $\alpha_0^\pm$ are given in 3.26 and $\alpha_1^\pm$ are deduced from the two linear systems $(\pm)$:

$$(\boldsymbol{\nabla} a)_c^\pm = \arg\min_{\boldsymbol{\nabla} a} \max_\lambda \sum_d w_{cd} \left((\boldsymbol{\nabla} a)_c^\pm \cdot \delta\boldsymbol{x}_{cd} + a_c^\pm - a_d\right)^2 + \lambda \left(V_c (\boldsymbol{\nabla} a)_c^\pm \cdot \boldsymbol{X}\right), \tag{3.27a}$$

$$\phi_c^\pm\big(\Delta t) = \phi_c^{1\mathrm{D},\pm}(\Delta\xi_c), \qquad \text{with } \xi_0 = 1 - \Delta\xi_c,\ \xi_1 = 1, \tag{3.27b}$$

$$\phi_c^\pm\big(\Delta t_1) = \phi_c^{1\mathrm{D},\pm}\big(\Delta\xi = 1\big), \qquad \text{with } \xi_0 = 0,\ \xi_1 = 1, \tag{3.27c}$$

$$\alpha_0^\pm = \alpha_\pm - \frac{1}{2}\alpha_1^\pm (1 \mp 1). \tag{3.27d}$$

The formula for the resulting slope coefficients $\alpha_1^{\pm}$ are too long to state here. For completeness, the solutions of these linear systems are given in Appendix 3.5.2.

3.2.3 Nonlinear reconstruction by Slope-And-Bound limitation in the 1D reference frame

This chapter is divided in three paragraphs: (i) the generic second-order SAB interpolator is introduced; (ii) the SAB limitation strategy [68] is explained in the 1D reference frame; (iii) the flux integration of a generic second-order SAB interpolator is given.

In [68], a categorisation of second-order interpolator functions is stated, and the most accurate interpolator is deduced by some rule connected to the *second-order reference point*: the *second-order reference point* is the central value (the value given on the cell centroid) of cell c that connects the mean values of the left and right neighboring cells in alignement. Thus, it depends linearly on the respective sizes of the neighboring cells. In the case of a uniform grid, the level of the second-order reference point is exactly in the middle between the levels of left and right central values. However, in the present ND-SAB algorithm, where the SAB limitation takes place in the 1D reference frame, the sizes of of the neighboring cells are unknown and thus, the 1D reference point is unknown in the 1D reference frame.

Second-order Slope-And-Bound interpolator in the 1D reference frame

The generic second-order SAB interpolator consists in three linear parts: a left bound, which is a constant function (at α_- over an interval of length η_-), a middle part, which is defined by a gradient reconstruction through a first-order polynomial (with slope α_1 and its affine shift α_0), and a right bound, which is again a constant function (at α_+ over an interval of length η_+). As illustrated in Figure 3.6 the SAB interpolator is defined locally on cell c resized on the interval $[0, 1]$ by the 6-tuple $(\alpha_1, \alpha_0, \eta_\pm, \alpha_\pm)$ and writes

$$\alpha(\xi) = \begin{cases} \alpha_-, & \text{if } \xi \in [0, \eta_-], \\ \alpha_1 \xi + \alpha_0, & \text{if } \xi \in [\eta_-, 1 - \eta_+], \\ \alpha_+, & \text{if } \xi \in [1 - \eta_+, 1]. \end{cases} \tag{3.28}$$

Remark that Single-Slope interpolators can be interpreted as SAB interpolators where both the left and the right bound are of size zero — which is why (3.28) is refered to as the "generic" SAB interpolator.

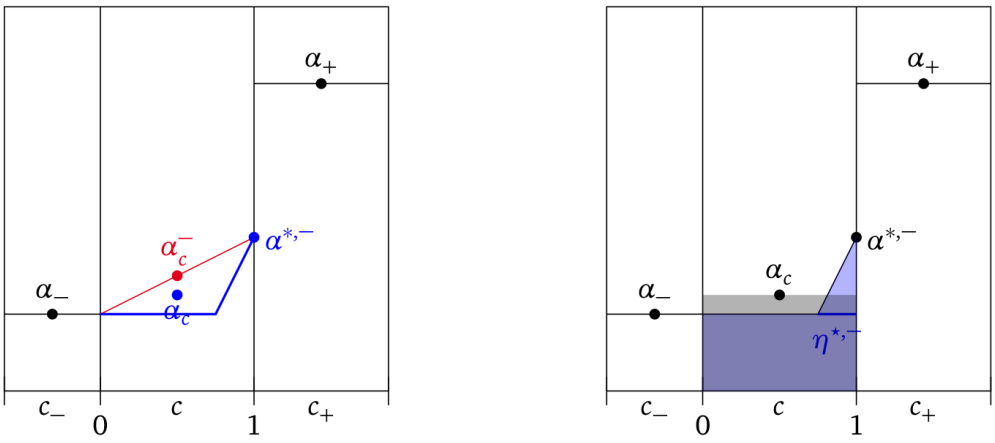


Figure 3.5: Left: SAB interpolator (blue) with reference point $\alpha^{*,-}$ and monotonicity threshold slope (red); right: slope support $\eta^{\star,-}$ of the SAB interpolator, and illustration of the preserved integral (blue) given by (3.31).

Slope-And-Bound limitation strategy in the 1D reference frame

This paragraph explains the SAB limitation strategy in the 1D reference frame. Before detailing the construction of the SAB interpolator functions, the link of the present work with the doubly-monotonic constraint of article [68] must be emphasized: The slope reconstruction strategy introduced above in chapter 3.2.2 suggests limitation with SAB interpolators verifying the doubly-monotonic constraint [68, §4.3] (defined in the present work in chapter 2.4), which excludes the *slope-and-two-bounds* (SAB2) interpolators, *i.e.* the previously introduced generic SAB interpolator reduces to a particular form of a slope with at most one bound, which is the subject of this chapter. Indeed, the SAB2 interpolators are not second-order accurate as they do not align the *second-order reference point* [68, §3.1, (3.2)] (stated in the present work in equation (2.14)) with the central values of the left and right cells [68].

In particular, gradients reconstructed with a least squares approximation (3.8) verify the doubly-monotonic constraint. Thus, the SAB strategy is an adequate choice to limit linear approximation $\alpha(\xi)$ (3.15) in the 1D reference frame.

The previously computed 1D reference slopes $\alpha(\xi)$ need limitation if they are beyond the threshold slopes (with the threshold zones illustrated in figure 3.4). The limitation criterion is stated via the central value α_c (3.21) of the reference slope and the central values $\alpha_c^\pm$ (3.24) of the threshold slopes: Limitation is needed if

$$\alpha_c \notin [\min\{\alpha_c^\pm\}, \max\{\alpha_c^\pm\}], \qquad \text{(limitation constraint)} \qquad (3.29)$$

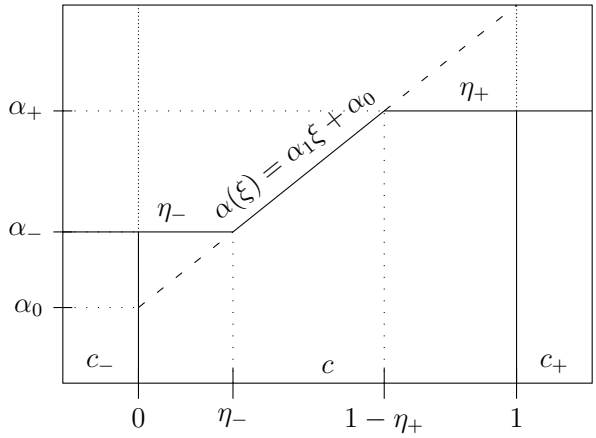


Figure 3.6: Illustration of the generic second-order Slope-And-Bound interpolator defined by the 6-tuple $(\alpha_1, \alpha_0, \eta_\pm, \alpha_\pm)$ on a cell c resized to $[0, 1]$.

whereas TVD truncation to first-order is added beyond the TVD thresholds, *i.e.* if

$$\alpha_c \notin [\min\{\alpha_\pm\}, \max\{\alpha_\pm\}]. \qquad \text{(TVD limitation constraint)} \tag{3.30}$$

The SAB strategy limits the slopes beyond the monotonicity thresholds $\alpha_c^\pm$ given in (3.24) by interpolation via a reference point called $\alpha^{\star,\pm}$ (defined below and illustrated in figure 3.5). Under preservation of the mean value of the cell quantity

$$\alpha_c = \frac{1}{V_c}\int_{V_c} \alpha(\xi) d\eta. \tag{3.31}$$

This defines a unique SAB interpolator of the following form: (i) the generic form reduces to two linear parts, where either the left or the right bound is of size zero (*i.e.* $\eta_- = 0$ or $\eta_+ = 0$); (ii) the bounds are defined by $\alpha_\pm$ given in (3.22); (iii) the middle part is the slope part and reaches the corresponding reference point $\alpha^{\star,+}$ or $\alpha^{\star,-}$ on the unbounded side. The interpolation reference points $\alpha^{\star,\pm}$ are computed via the threshold slopes $\alpha_c^\pm(\xi)$ (3.23b) with

$$\alpha^{\star,\pm} = \alpha_\pm \mp \alpha_1^\pm. \tag{3.32a}$$

And the slope support $\eta^{\star,\pm}$ is determined by preservation of the volume under the interpolator defined by (3.31) and illustrated in figure 3.5.

$$\eta^{\star,\pm} = \frac{2(\alpha_c - \alpha_\mp)}{\alpha^{\star,\pm} - \alpha_\mp}. \tag{3.33}$$

Then, the two 4-tuples $(\alpha_1^+, \alpha_0^+, \eta_\pm^+)$ and $(\alpha_1^-, \alpha_0^-, \eta_\pm^-)$ can be deduced with

$$\eta_-^+ = 0, \tag{3.34a}$$
$$\eta_+^+ = \mathbb{S}_{[0,1]}(1 - \eta^{\star,+}), \tag{3.34b}$$
$$\alpha_1^+ = \frac{\alpha^{\star,+} - \alpha_-}{\eta^{\star,+}}, \qquad \text{(if } \eta^{\star,+} \neq 0, \text{ and 0 otherwise)}, \tag{3.34c}$$
$$\alpha_0^+ = \alpha^{\star,+}. \tag{3.34d}$$

$$\eta_-^- = \mathbb{S}_{[0,1]}(1 - \eta^{\star,-}), \tag{3.35a}$$
$$\eta_+^- = 0, \tag{3.35b}$$
$$\alpha_1^- = \frac{\alpha_+ - \alpha^{\star,-}}{\eta^{\star,-}}, \qquad \text{(if } \eta^{\star,-} \neq 0, \text{ and 0 otherwise)}, \tag{3.35c}$$
$$\alpha_0^- = \alpha^{\star,-} - \alpha_1^-. \tag{3.35d}$$

For improved performances of the implemented algorithm, the previous computations can be combined with two boolean coefficients $\sigma^{\pm}$ that indicate which threshold governs the interpolator function (*i.e.* $(-)$ governs if $\alpha_c \in [\min\{\alpha_-, \alpha_c^-\}, \max\{\alpha_-, \alpha_c^-\}]$ and $(+)$ governs if $\alpha_c \in [\min\{\alpha_+, \alpha_c^+\}, \max\{\alpha_+, \alpha_c^+\}]$), with $\alpha_\pm$ given in (3.22).

$$\sigma^{\pm} = ((\alpha_c^{\pm} - \alpha_c)(\alpha_c - \alpha_{\pm}) > 0). \tag{3.36}$$

Recall that limitation is needed as described by limitation constraint (3.29). Let σ^0 be the boolean where no limitation is needed.

$$\sigma^0 = ((\alpha_c^+ - \alpha_c)(\alpha_c - \alpha_c^-) \geq 0). \tag{3.37}$$

Furthermore, the standard first-order truncation is added for the TVD constraint (3.30) with

$$\sigma^{\text{TVD}} = ((\alpha_+ - \alpha_c)(\alpha_- - \alpha_c) \geq 0). \tag{3.38}$$

Flux integration of a generic second-order Slope-And-Bound interpolator in 1D

The SAB limitation strategy is a novel linear approximation introduced in [68] as an alternative to standard limitation strategies of type "limiter function times slope" or "limiter function times flux". Thus, for this novel type of linear approximations the simplification (3.6b) is not valid anymore. This paragraph explains how the flux integral (3.3) is computed for a generic second-order Slope-And-Bound interpolator (3.28).

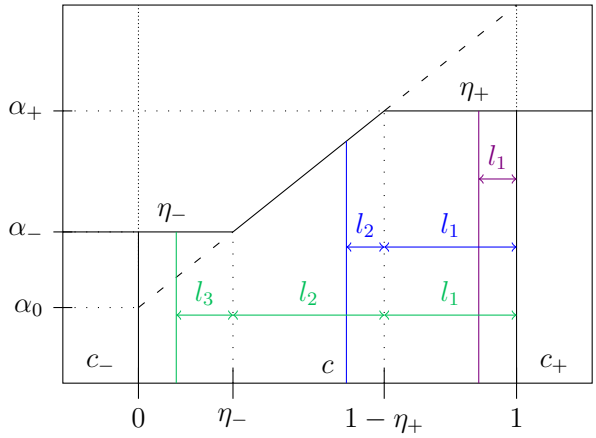


Figure 3.7: Flux support parameters l_1, l_2 and l_3 of flux integration with the generic second-order Slope-And-Bound interpolator. Different sizes of flux supports are indicated in color (violet, blue, green).

As illustrated in figure 3.7, the flux parts on the three SAB parts are integrated one by one. Consider the upwind flux going form the reference cell c to the right. Let the SAB parts be numerated from the right to the left: $1 \rightarrow$ right bound, $2 \rightarrow$ middle part, $3 \rightarrow$ left bound. The corresponding lengths l_1, l_2, and l_3 of each part are deduced from the 1D flux domain δ_ϕ as follows

$$l_1 = \min\{\delta_\phi, \eta_+\}, \tag{3.39a}$$
$$l_2 = \min\{\delta_\phi - l_1, 1 - \eta_+ - \eta_-\}, \tag{3.39b}$$
$$l_3 = \max\{0, \min\{\delta_\phi - l_1 - l_2, \eta_-\}\}, \tag{3.39c}$$
$$\delta_\phi = l_1 + l_2 + l_3. \tag{3.39d}$$

Usually the 1D flux domain is defined by $\delta_\phi = v\Delta t$, whereas in the 1D reduction reference frame it is defined by (3.17) with $\delta_\phi = (\Delta\xi)_c$.

Then, the total flux $\phi_c^{1\mathrm{D}}(\Delta t)$ is the sum of the three integral parts given by those lengths

$$\phi_c^{1\mathrm{D}}(\Delta t) = l_1\alpha_+ + l_2\left(\alpha_1(1 - l_1 - l_2/2) + \alpha_0\right) + l_3\alpha_-, \tag{3.40}$$

where $\alpha_1(1 - l_1 - l_2/2) + \alpha_0$ is the mean value of the middle part.

3.2.4 Multidimensional upwind fluxes

The total flux amount $\phi_c^{1\mathrm{D}}(\Delta t)$ for each cell c, was computed in the previous chapter with the SAB philosophy, which introduces steep gradients for limited slopes. The

difficulty in the multidimensional setup is to define a rule to separate the total flux amount ϕ_c^{1D} and redistribute the corresponding parts to the upwind face fluxes ϕ_{cd} of cell c.

Weighted redistribution

In this section a redistribution strategy is presented, which uses positive convex weighting factors ω_{cd} for the redistribution of $\phi_c^{\mathrm{1D}}(\Delta t)$:

$$\phi_{cd}(\Delta t) = \omega_{cd}\phi_c^{\mathrm{1D}}(\Delta t), \tag{3.41a}$$

$$\sum_d \omega_{cd} = 1. \tag{3.41b}$$

There are three intuitive choices of weighting factors:

$$\omega_{cd} = \frac{\mathring{V}_{cd}}{\sum_d \mathring{V}_{cd}}, \tag{3.42a}$$

$$\hat{\omega}_{cd} = \frac{\mathring{V}_{cd} a_c(\hat{\boldsymbol{x}}_{cd}(\Delta t))}{\sum_d \mathring{V}_{cd} a_c(\hat{\boldsymbol{x}}_{cd}(\Delta t))} = \frac{\mathring{V}_{cd}\hat{a}_{cd}}{\sum_d \mathring{V}_{cd}\hat{a}_{cd}}, \tag{3.42b}$$

$$\tilde{\omega}_{cd} = \begin{cases} \omega_{cd} & \text{if correction is needed,} \\ \hat{\omega}_{cd} & \text{else.} \end{cases} \tag{3.42c}$$

The first choice (3.42a) uses $\mathring{V}_{cd}$ and is thus purely geometric and was already introduced in the computation of $\alpha_\pm$ (3.22) in the 1D reference frame. The second choice (3.42b) uses a preference for the gradient direction, where the values $\hat{a}_{cd} = a_c^n(\hat{\boldsymbol{x}}_{cd})$ are linearly reconstructed with gradient $(\boldsymbol{\nabla} a)_c$ (3.9a) and in order to avoid negative values in the weighting factors, negative values are truncated to 0. And the third choice (3.42c) is a linear combination of the first two choices, under some constraint regarding corrections for optimal flux redistribution. The present work suggests a correction on the inflow fluxes, which is discussed as follows:

By construction the previously introduced *total flux amount* $\phi_c^{\mathrm{1D}}(\Delta t)$ corresponds exactly to the total outflow flux of cell c

$$\phi_c(\Delta t) = \sum_d \phi_{cd}(\Delta t) = \phi_c^{\mathrm{1D}}(\Delta t) \underbrace{\sum_d \omega_{cd}}_{=1} = \phi_c^{\mathrm{1D}}(\Delta t). \tag{3.43}$$

Regarding the MUSCL scheme (3.2) with an Eulerian description (*i.e.* a constant grid $V_c := V_c^n = V_c^{n+1}$), a_c^{n+1}, the outflow flux ϕ_c^{out} and the inflow flux ϕ_c^{in} write

$$a_c^{n+1} = a_c^n - \frac{\phi_c^{\text{out}}}{V_c} + \frac{\phi_c^{\text{in}}}{V_c}, \tag{3.44a}$$

$$\phi_c^{\text{out}} = \sum_d \phi_{cd}^n(\Delta t^n) \overset{(3.43)}{=} \phi_c^{\text{1D}}(\Delta t), \tag{3.44b}$$

$$\phi_c^{\text{in}} = \sum_d \phi_{dc}^n(\Delta t^n) = \sum_d \omega_{dc}\phi_d^{\text{1D}}(\Delta t). \tag{3.44c}$$

In order to prevent overshoot and undershoot, the discrete maximum principle (DMP) must be verified, which states

$$a_{\min}^n \leq a_c^{n+1} \leq a_{\max}^n. \tag{3.45}$$

The present work obts to correct the fluxes whenever (3.45) is violated by adapting the redistribution weighting factor (3.42c). The DMP (3.45) with the regarded MUSCL scheme (3.44) writes

$$a_{\min}^n \leq a_c^n - \frac{\phi_c^{\text{1D}}(\Delta t)}{V_c} + \frac{\sum_d \hat{\omega}_{dc}\phi_d^{\text{1D}}(\Delta t)}{V_c} \leq a_{\max}^n. \tag{3.46}$$

This expression shows that the redistribution weights only act on the inflow flux ϕ_c^{in}. (Recall that $\phi_{dc} = -\phi_{cd}$.) Therefore, verifying (3.46) means that each cell must receive a correct amount of inflow. In the following this is called the inflow problem. The simplest way to solve this inflow problem is presented here with redistribution weight (3.42c), which applied to the solver as follows: (i) the face fluxes are computed with weight (3.42b); (ii) the validation of the DMP (3.46) is checked; (iii) wherever the DMP is invalid, the inflow fluxes need correction. Hence, ω_{cd} is corrected in the downwind neighboring cells with (3.42a).

The attentive reader might remark that this correction is not an ideal choice, as it does not exactly verify the DMP property on general grids and with arbirary velocities. As a matter of fact, on a Cartesian grid and with a constant velocity vector, the weights (3.42a) are constants and verify $\sum_d \omega_{dc} = 1$, which is not the case with weight (3.42b), where the sum of the inflow weight coefficients can exceed 1. Hence, the hybrid redistribution weight (3.42c) separates the total flux more evenly than (3.42b) does. Results of all three choices of (3.42) are shown in section 3.3.

3.2.5 Algorithm

The ND-SAB algorithm is summarized step-by-step: (i) reconstruction of a multidimensional gradient (ii) reduction of the multidimensional gradient to the 1D

reference frame, and an aggregation of upwind (+) and downwind (−) data from the stencil defines the bounds of the SAB strategy, (iii) SAB limitation and computation of the total upwind flux of cell c (in the 1D reference frame), iv) redistribution of the total flux to the face fluxes in the multidimensional setup.

$(\boldsymbol{\nabla} a)_c$	α_1, α_0	$\alpha_1^{\pm}$, $\alpha_0^{\pm}$	$\alpha_{\pm}$	$(\alpha_1, \alpha_0, \eta_{\pm})^{\text{limited}}$	$\phi_c^{\text{1D}}(\Delta t)$	$\phi_{cd}(\Delta t)$
(3.9a)						
	(3.20)					
		(3.27)				
			(3.22)			
				(3.34), (3.35)		
					(3.40)	
						(3.42)
3.2.1	3.2.2			3.2.3		3.2.4

Table 3.1: ND-SAB: the multidimensional SAB reconstruction algorithm.

There are two main techniques crucial to this algorithm: approximation of linear data (through weighted least squares) and the dimensional reduction to the 1D reference frame, which are used twice: The first time, these techniques are used in the computation of gradient $(\boldsymbol{\nabla} a)_c$ and the 1D reduction slope coefficients α_1, α_0, which can be computed in two seperated steps. Whereas the second time, the computation of the monotonicity thresholds at $a_c^{\pm}$ with $(\boldsymbol{\nabla} a)_c^{\pm}$ and the 1D reduction slope coefficients $\alpha_1^{\pm}, \alpha_0^{\pm}$ are given by one inseperable linear system of equations (3.27) (in other words: here $a_c^{\pm}$ represents an additional unknown compared to the first system of equations where a_c is known — i.e. one more equation is needed to solve the linear system, which is why (3.26) has been added to the linear system (3.27)).

3.3 Results

In this section the advection problem of a circle on a uniform Cartesian grid is shown for constant velocities. The velocity vectors are defined by angle θ as

$$\boldsymbol{v}(\theta) = ||\boldsymbol{v}||(\cos(\theta), \sin(\theta)). \tag{3.47}$$

Each figure shows the results of three simulations superposed in one figure, where the three directions $\theta \in \{0, \pi/8, \pi/4\}$ are chosen. Remark that choosing other multiples of $\pi/8$ for θ give the same results (up to symmetry).

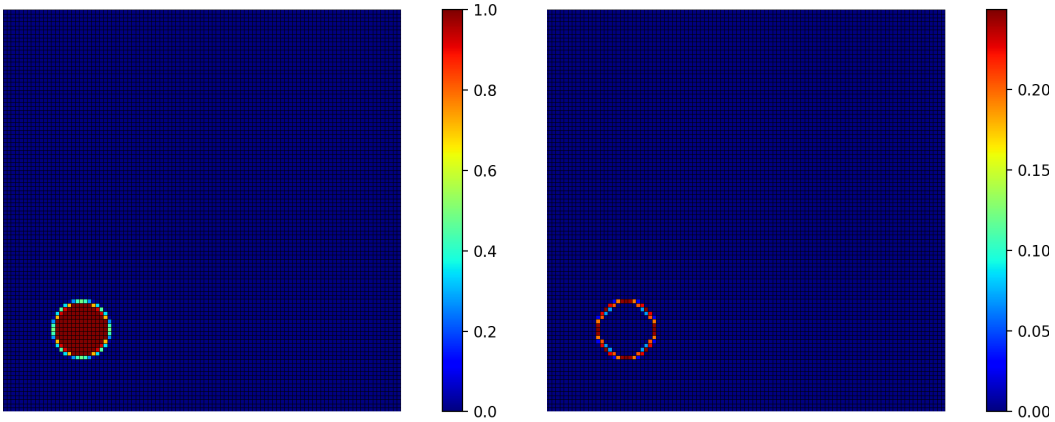


Figure 3.8: Initialisation of circle with radius $r = 0.7$ and density $\rho = 1$ on a Cartesian grid on $[0, 1]^2$ with 100×100 cells. Density plots of ρ (left) and $\rho(1 - \rho)$ (right).

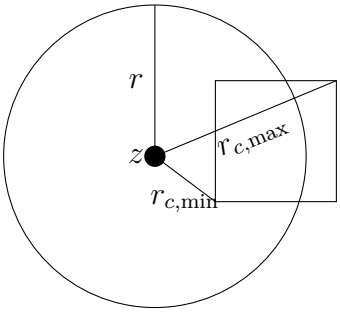


Figure 3.9: Radius parameters of a cell c on a circle with radius r and center z.

The simulations consider mono-fluid advection of the density field ρ, with CFL= 0.5 and final time $T = 0.5$. This is described by the MUSCL scheme (3.2) with $a_c = \rho_c$ and the second-order flux computed with the ND-SAB algorithm presented in this work. Given a circle at center $z(0.205, 0.205)$ with radius 0.7 on a Cartesian grid on $[0, 1]^2$ with 100×100 cells. The density field in each cell c is initialized at 1 inside the circle; 0 outside the circle; and with a density value $\rho \in [0, 1]$ on the sphere. The latter is approximately computed with a simple formula considering the minimal and maximal distances of a given cell c to the center of the circle z via the nodes $n \in \mathcal{N}(c)$.

$$\rho_c = \mathbb{S}_{[0,1]} \left(\frac{r - r_{c,\min}}{r_{c,\max} - r_{c,\min}} \right), \tag{3.48a}$$

$$r_{c,\min} = \min_{n \in \mathcal{N}(c)} ||n - z||_2, \tag{3.48b}$$

$$r_{c,\max} = \max_{n \in \mathcal{N}(c)} ||n - z||_2. \tag{3.48c}$$

Remark that the clipping function $\mathbb{S}_{[0,1]}$ assures the initialization for cells not containing the sphere of the circle. Figure 3.9 illustrates a cell on the sphere, *i.e.* the case $r_{\min} < r < r_{\max}$.

Figure 3.8 shows the density plots ρ and $\rho(1 - \rho)$ of the initial condition: The plot on the left (ρ) is the originally scaled solution, whereas the plot on the right ($\rho(1 - \rho)$) illustrates whatever values are not either zero or one. This second plot applied to the solution data of the present scheme illustrates best the support of the numerical diffusion and is thus systematically added to all the figures. Furthermore, plots in log scale of the latter (*i.e.* plots of $\log(|\rho(1 - \rho)|)$) are added to illustrate the numerical wetting in the results (over the support of the numerical diffusion respectively).

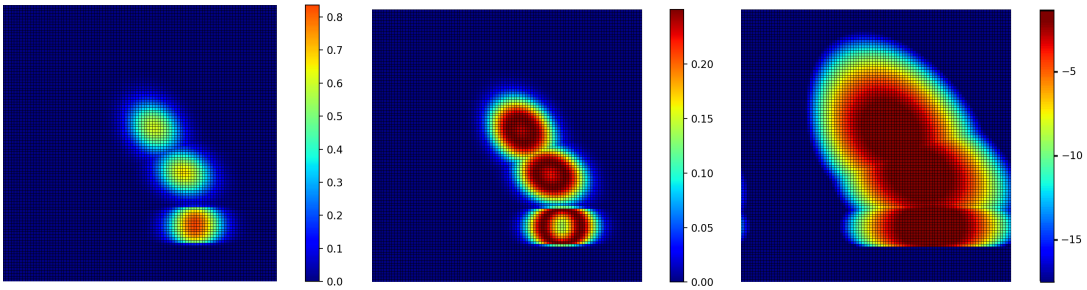


Figure 3.10: First-order baseline. Cartesian grid on $[0,1]^2$ with 100×100 cells, CFL$= 0.5$ and final time $T = 0.5$. Density plots of ρ (left), $\rho(1-\rho)$ (middle), and $\log(|\rho(1-\rho)|)$ (right).

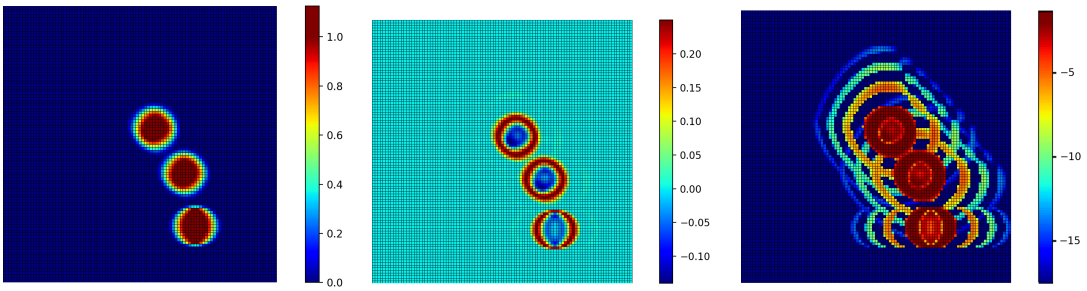


Figure 3.11: Non-limited second-order baseline with least squares gradient reconstruction. Cartesian grid on $[0,1]^2$ with 100×100 cells, CFL$= 0.5$ and final time $T = 0.5$. Density plots of ρ (left), $\rho(1-\rho)$ (middle), and $\log(|\rho(1-\rho)|)$ (right).

3.3.1 Multidimensional Slope-And-Bound reconstruction algorithm

Results of the present ND-SAB algorithm with the weighted redistribution introduced in section 3.2.4 are shown for the three weighting factors given in (3.42). Before showing these figures, baseline schemes are introduced with (3.2) for first-order and second-order non-limited fluxes:

Figure 3.10 shows the results for the first-order flux given by

$$\phi_{cd}^{order1} = a_c \mathring{V}_{cd} \Delta t, \tag{3.49a}$$

$$\phi_{dc}^{order1} = a_d \mathring{V}_{dc} \Delta t. \tag{3.49b}$$

Figure 3.11 shows the non-limited second-order flux given by (3.6b). Recall,

$$\phi_{cd}^{non-limited} = \hat{a}_{cd} \mathring{V}_{cd} \Delta t = a_c(\hat{\boldsymbol{x}}_{cd}) \mathring{V}_{cd} \Delta t, \tag{3.50a}$$

$$\phi_{dc}^{non-limited} = \hat{a}_{dc} \mathring{V}_{dc} \Delta t = a_d(\hat{\boldsymbol{x}}_{dc}) \mathring{V}_{dc} \Delta t. \tag{3.50b}$$

The results obtained with this non-limited second-order reconstruction significantly diminish the first-order numerical diffusion. However, the non-limited flux leads to strong overshooting. Regarding figure 3.11, overshoots are observed in the left plot where the density variable ρ clearly exceeds 1 in the center of the circle, which is greater than the maximal value of the initial condition and is thus violating monotonicity. In the logarithmic plot on the right the numerical wetting is observed as oscillations.

The three weighting factors (3.42) have been tested on a Cartesian grid with a constant velocity field. (3.42a) only depends on the grid orientation with respect to the velocity direction, i.e. the ω_{cd} are constant for every cell c, where d indicates the *south*, *east*, *north* and *west* neighbors. Results are shown in figures 3.12. In these simulations with constant velocities on a Cartesian grid, no overshoot or undershoot have been observed, however a lot of numerical diffusion (similar to first-order reconstructions) exists perpendicular to the velocity direction of the third simulation ($\theta = \pi/4$). (3.42b) reduces this numerical diffusion, however it introduces strong overshoot and undershoot, as can be observed in figures 3.13. (3.42c) is used to correct these overshoot and undershoot locally with the first choice, wherever correction is desirable. This is illustrated in figures 3.14. Furthermore, and most importantly the plots on the right in figures 3.13 and 3.14 show that numerical wetting has been reduced significantly.

3.3.2 Combining Slope-And-Bound and co-mesh approach

Despite the reduction of the numerical diffusion with the second-order strategy presented in this work, mesh imprinting is still present and can be observed in all the previous figures 3.10-3.14. Mesh imprinting has been improved with the co-mesh approach in [69], where results are shown for the baseline first-order MUSCL scheme (3.49). The co-mesh approach [69] is compatible with the present ND-SAB algorithm. Figures 3.15 show the results of the ND-SAB algorithm with flux redistribution factor (3.42c) and combined with the co-mesh strategy introduced in chapter one.

The co-mesh approach corresponds to a linear combination of solving the advection step with "$\omega\times$ mesh + $(1-\omega)\times$ co-meshes": two co-meshes define each a covering of the initial mesh and the scheme reads

$$0 = \frac{1}{V_c}\Big(V_c^{n+1}a_c^{n+1} - V_c^n a_c^n + \omega \sum_d \left(\phi_{cd}^{(1),n} - \phi_{dc}^{(1),n}\right) + \frac{1-\omega}{2}\sum_d \left(\phi_{cd}^{(2),n} - \phi_{dc}^{(2),n}\right) + \frac{1-\omega}{2}\sum_d \left(\phi_{cd}^{(3),n} - \phi_{dc}^{(3),n}\right)\Big), \tag{3.51}$$

where $V_c = \omega V_c^{(1)} + \frac{1-\omega}{2}V_c^{(2)} + \frac{1-\omega}{2}V_c^{(3)}$ on an Eulerian grid ($V_c := V_c^{n+1} = V_c^n$), and

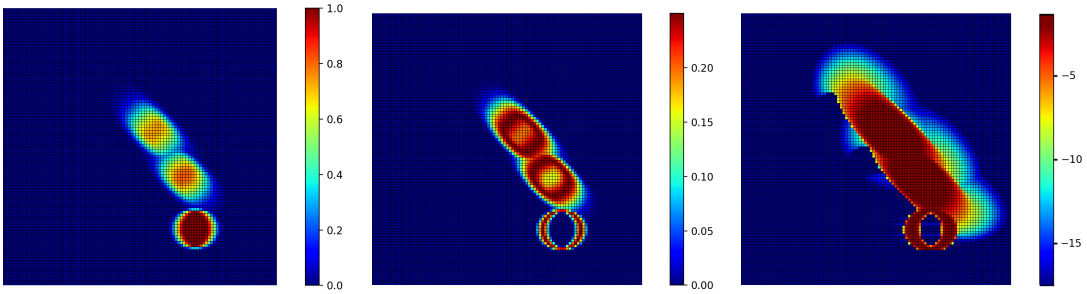


Figure 3.12: Second-order reconstruction with ND-SAB algorithm and flux redistribution weight (3.42a). Cartesian grid on $[0,1]^2$ with 100×100 cells, CFL$= 0.5$ and final time $T = 0.5$. Density plots of ρ (left), $\rho(1-\rho)$ (center), and $\log(|\rho(1-\rho)|)$ (right).

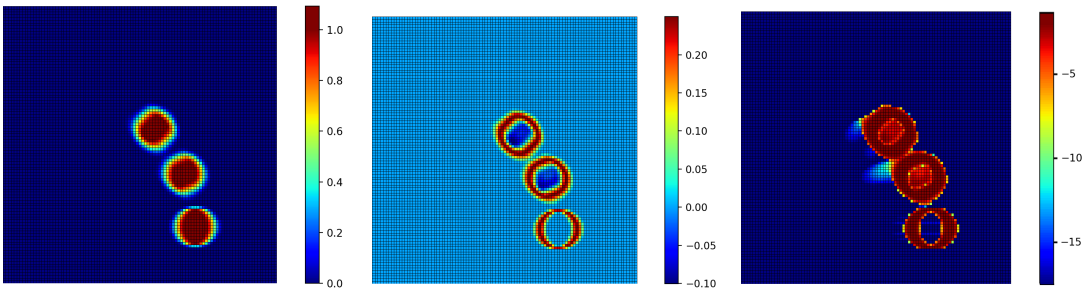


Figure 3.13: Second-order reconstruction with ND-SAB algorithm and flux redistribution weight (3.42b). Cartesian grid on $[0,1]^2$ with 100×100 cells, CFL$= 0.5$ and final time $T = 0.5$. Density plots of ρ (left), $\rho(1-\rho)$ (center), and $\log(|\rho(1-\rho)|)$ (right).

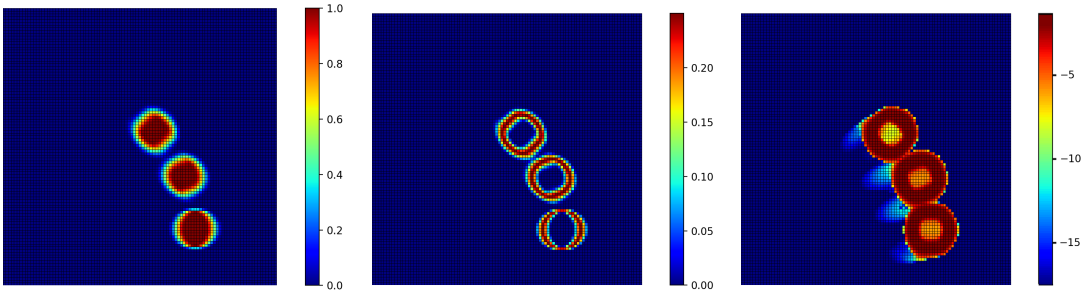


Figure 3.14: Second-order reconstruction with ND-SAB algorithm and flux redistribution weight (3.42c). Cartesian grid on $[0,1]^2$ with 100×100 cells, CFL$= 0.5$ and final time $T = 0.5$. Density plots of ρ (left), $\rho(1-\rho)$ (center), and $\log(|\rho(1-\rho)|)$ (right).

$\phi_{cd}^{(1),n} = \phi_{cd}^{n}(\Delta t^n)$ the flux of the initial mesh, (2) and (3) indicate the corresponding fluxes on the co-meshes, which are both build as instructed in [69] via nodes defined on the cell-centers of the initial Cartesian mesh. Furthermore, [69] introduces an optimal linear factor $\omega = \omega_{opt} = 0.585\underline{8}$, which was deduced through analysis of the modified equation of the advection scheme.

Mesh imprinting is significantly reduced with the co-mesh approach in figures 3.15 compared to figures 3.14. This result is once again illustrated on the initial Cartesian mesh, the co-mesh and the optimal combination of both meshes, with density plots in Figures 3.16 and the plots showing the support of the numerical diffusion in Figures 3.17.

Let this result be discussed in detail: Figures 3.18 and 3.19 show the results of the novel multidimensional ND-SAB algorithm together with the co-mesh apporach on the initial mesh (bottom left), its co-mesh (bottom right) and the optimal linear combination of the resolution on initial mesh and co-meshes (top right): the corresponding density plots are shown in Figure 3.16, profiles over slices centered in the center of the circles are shown in direction of the velocity vectors $\boldsymbol{v}(\theta) = ||\boldsymbol{v}||(\cos(\theta), \sin(\theta))$ given by $\theta \in \{0, \pi/8, \pi/4\}$ (in Figure 3.18), and perpendicular to the velocity directions (in Figure 3.19).

The velocity vectors (3.47) of the present simulations are given by $\theta \in \{0, \pi/8, \pi/4\}$. Up to symmetry any other choice of a multiple of $\pi/8$ for θ leads to the same results, which represent three categories: (i) $\theta \in \{k\pi/2 | k \in \mathbb{Z}\}$ aligns the velocity vectors with the grid; (ii) $\theta \in \{k\pi/4 | k \in \mathbb{Z}\}$ aligns the velocity vectors with the diagonal of the grid; (iii) $\theta \in \{k\pi/8 | k \in \mathbb{Z}\}$ represents some intermediate direction without alignement of the grid.

These three categories are distinguished in Figures 3.18 and 3.19 with three different colors: (i) in red (dashed); (ii) in green (dotted); (iii) in blue (dashed); There is not much variation observed on the plots over slices following the velocity direction (Figures 3.18). On the figures perpendicular to the velocity directions

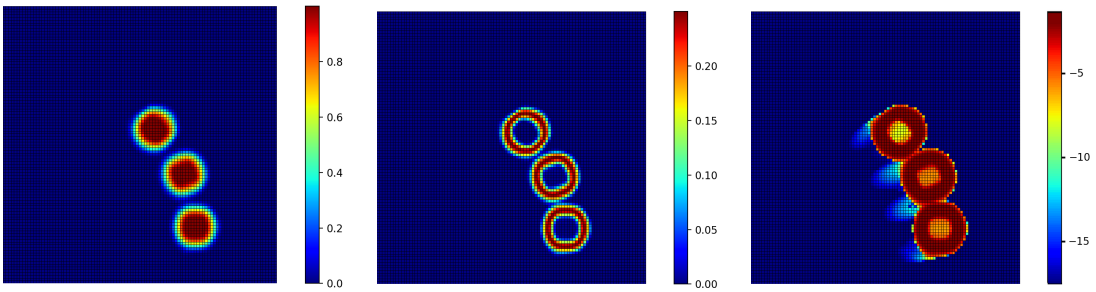


Figure 3.15: Second-order reconstruction with ND-SAB algorithm and co-mesh. Cartesian grid on $[0, 1]^2$ with 100×100 cells, CFL= 0.5 and final time $T = 0.5$. Density plots of ρ (left), $\rho(1 - \rho)$ (center), and $\log(|\rho(1 - \rho)|)$ (right).

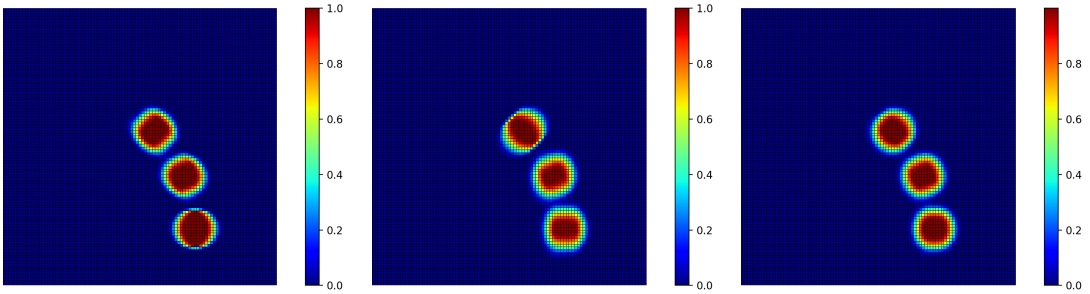


Figure 3.16: Density plot ρ of ND-SAB second-order reconstruction with co-mesh strategy: solutions are shown on the initial grid (left), the co-meshes (center), and an optimal linear combination of initial grid and its co-meshes (right).

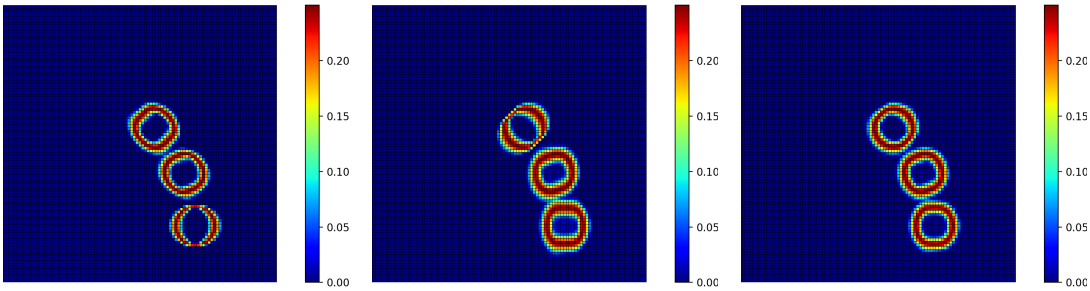


Figure 3.17: Density plot $\rho(1-\rho)$ (illustrating support of numerical diffusion) of ND-SAB second-order reconstruction with co-mesh strategy: solutions are shown on the initial grid (left), the co-meshes (center), and an optimal linear combination of initial grid and its co-meshes (right).

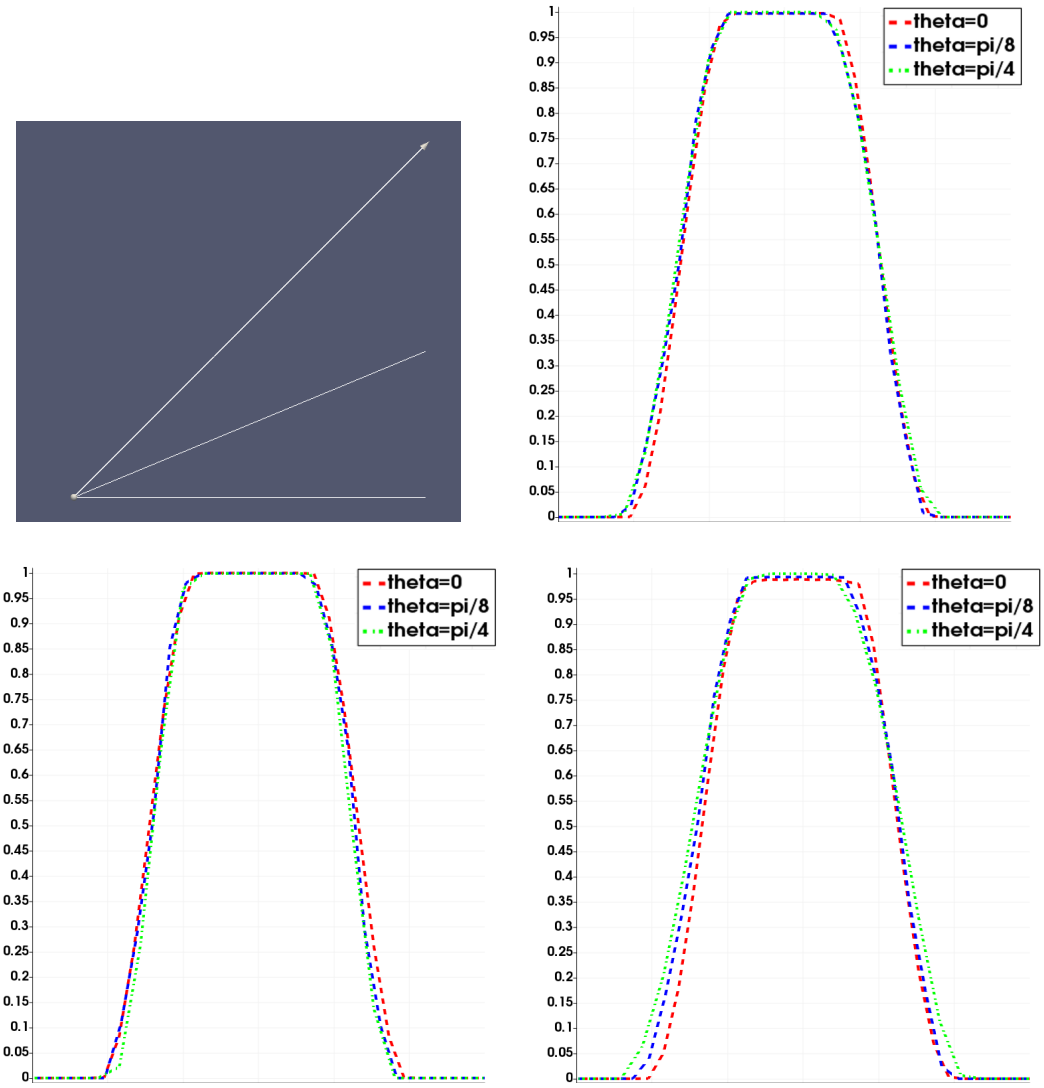


Figure 3.18: Slices in velocity directions: solutions are shown on the initial grid (bottom left), the co-mesh (bottom right) and for an optimal linear combination of initial grid and its co-mesh (top right).

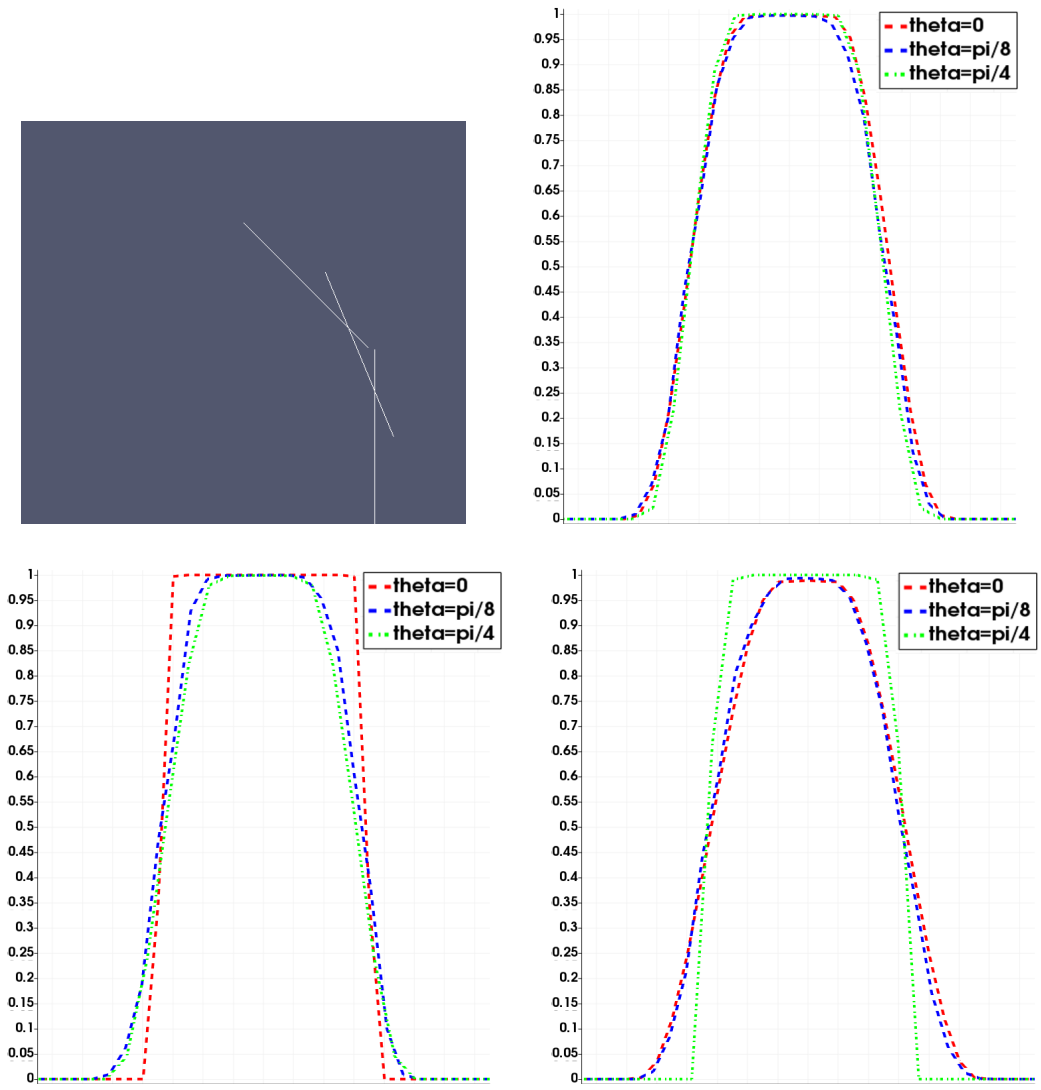


Figure 3.19: Slices transversal to the velocity directions: solutions are shown on the initial grid (bottom left), the co-mesh (bottom right) and for an optimal linear combination of initial grid and its co-mesh (top right).

(Figures 3.19) on the other hand, anisotropic behavior is observed on the initial mesh (bottom left): no diffusion is marked on the red plot (velocity direction $\theta = 0$); as well as on the co-mesh (bottom right): no diffusion is marked on the green plot (velocity direction $\theta = \pi/4$); this is explained by the grid orientation effect, where cells have been canceled out of the stencil. The plots with the co-mesh strategy (top right) indicate no directional difference, which signifies a quasi-isotropic behavior.

3.4 Conclusions

The goal of the present work is to reduce numerical wetting, which is a major artifact of the advection step in hydrocodes. Advection problems are often solved with a multi-fluid discretization and encounter deformed meshes. To solve this problem, a second-order, explicit, upwind advection operator using SAB limitation is introduced, compatible with the GEEC scheme [53], which is a quasi-symplectic, multi-fluid direct-ALE formalism and a MUSCL formulation is used for the advection steps.

The present ND-SAB algorithm is introduced here as an alternative to classical MUSCL techniques. Furthermore, it is compatible with the co-mesh approach [69], which corrects the grid orientation effect. Results are presented for both the 2D SAB algorithm and the co-mesh approach: numerical wetting and the grid orientation effect have been reduced.

The present ND-SAB algorithm is limited to uniform Cartesian grids and constant velocity fields. Extensions to arbitrary meshes and non-constant velocities are work in progress. The current description of the advection scheme in such conditions does not guarantee the validity of the DMP property. Indeed, overshooting has been observed in experiences on arbitrarily deformed Cartesian grids and non-constant, but divergence free velocity fields. In order to be compatible with the advection steps of the multi-fluid, ALE GEEC solver, the SAB reconstruction advection step should be free of overshooting for any deformation of the Cartesian grid and any divergence free velocity field.

As identified in the DMP of the present algorithm (3.46), corrections added to the flux redistribution step are acting on the inflow fluxes of the neighboring cells in the stencil for a given total outflow flux amount. Attemps to search for an "*exact correction*" of the inflow problem have been tackled by tracking the cells where *overshoot/undershoot* occur and correcting the fluxes responsible. The naive version given by the hybrid redistribution weighting factor (3.42) shows some interesting results, however, for more general cases of deformed meshes and arbitrary velocities, further corrections are needed. The most intuitive idea is to correct the total inflow (of the cell with the overflow problem) in a manner that it corresponds exactly to the maximal possible flux amount that could *flow in*. Preferences on *where*

the maximal flux amount should come from must be discussed, and a criterion to deduce a "*privileged flux direction*" must be defined, and the rest of the flux is redistributed to the other upwind faces. Different criteria could be considered: (i) the velocity direction, (ii) the gradient direction, (iii) maximum values of the previously discussed weighting factors (3.42), (iv) some combination of what is mentioned here.

Other adaptations for the present algorithm should be investigated as well, while preserving the SAB philosophy: (i) choose a gradient reconstruction; (ii) define bounds with the cell data of the neighbors in the stencil (upwind neighbors are aggregated to the right and downwind neighbors to the left); (iii) clip the gradients, such that the resulting interpolator function fits between the two bounds and the mean value of the cell is preserved. In 1D the SAB strategy only depends on the definition of the gradient reconstruction, and the computation of the outflow flux is straightforward. [68] shows results for different gradients inspired by minmod, superbee and monotonized central reconstructions. However, the multidimensional extension (summarized in table 3.1) is more complex, where choosing a gradient reconstruction does not directly define the rules for computing an outflow flux. As mentioned above, adaptation of the fourth step has been explored in the present work, and further tests are work in progress. Regarding the other steps of the algorithm, the influence of choosing a different gradient reconstruction in the first step should be investigated as well. Furthermore, the reconstruction of the bounds should be discussed as well. The present formulas (3.22) were introduced here as some intuitive choice, where the levels of the bounds are constructed by weighting the cell data of the neighboring cells with the respective sizes of the flux supports. Instead of using weighting functions, min and max functions could be considered for instance.

In conclusion, the SAB limitation strategy is an interesting tool to reduce numerical wetting in advection problems. We plan to investigate the effect of the present SAB algorithm in the advection step of the multi-fluid GEEC solver, where discretizations of the Euler equations (discrete mass, momentum, and energy conservation equations) are given with MUSCL advection fluxes. In this context the advection scheme must be adapted to a general ALE approach with moving grids. Furthermore, the 1D SAB limitation is examined in a 2D MUSCL ADI code (MUSCL: Monotonic Upstream-centered Scheme for Conservation Laws [42], AD: Alternating Direction [129]), where directional limitations are fully 1D and thus better suited to compare traditional limiter functions such as superbee with the SAB superbee version. Last but not least, extensions to higher-order reconstructions are debated as well. A third-order technique could be established with parabolic reconstructions and bounds.

3.5 Appendix

3.5.1 Weighting factors deduced from gradients of a parabola

This section shows the computations that justify the weighting factors (3.10). The main idea is to deduce the seeked coefficients as weights that define the gradient of a reconstructed polynomial that is accurate at the scheme's order. Thus, for a second-order accurate gradient a locally parabolic function p is considered.

Let p be centered over the center of cell c, *i.e.* at $x_c = 0$, and g be its gradient at this point. Then, a general parabola with coefficients a and c writes

$$p(x) = ax^2 + gx + c. \tag{3.52}$$

Assume that the integrals of p over cell c and its left and right neighbors $c_\pm$ equals their corresponding mean values times the cell volume. With the 1D cell volumes h this translates into

$$\int_{V_c} p(x)dx = a_c h_c, \tag{3.53a}$$
$$\int_{V_{c-}} p(x)dx = a_{c-} h_-, \tag{3.53b}$$
$$\int_{V_{c+}} p(x)dx = a_{c+} h_+. \tag{3.53c}$$

Integrating (3.53) leads to the mean values of each cell

$$a_c = c + \frac{1}{12} h_c^2 a \tag{3.54a}$$
$$a_\pm = c \pm \frac{1}{2}(h_c + h_\pm)g + \frac{1}{12}(3h_c^2 + 6h_c h_\pm + 4h_\pm^2)a, \tag{3.54b}$$

where a_c is the value at $x_c = 0$, a_{c-} at $x_{c-} = -\frac{1}{2}(h_c + h_{c-})$ and a_{c+} at $x_{c+} = \frac{1}{2}(h_c + h_{c+})$.

The traditional right and left gradients [125] given by mean values (3.54b) write

$$g_\pm = \frac{\pm 2(a_\pm - a_c)}{(h_c + h_{c\pm})} \overset{(3.54b)}{=} g \pm \frac{1}{3}(h_c + 2h_\pm)a. \tag{3.55}$$

Now, we are interested in deducing weighting factors $w_\pm$, such that the seeked slope g is of the form

$$g = w_+ g_+ + w_- g_- \overset{(3.55)}{=} (w_+ + w_-)g + \frac{a}{3}(w_+(h_c + 2h_+) - w_-(h_c + 2h_-)). \tag{3.56}$$

Thus, the following linear system must be solved.

$$1 = w_+ + w_-, \tag{3.57a}$$
$$0 = w_+(h_c + 2h_+) - w_-(h_c + 2h_-). \tag{3.57b}$$

Hence,

$$w_\pm = \frac{h_\mp + h_c/2}{h_\pm + h_\mp + h_c}. \tag{3.58}$$

3.5.2 Solving linear system (3.27)

This section shows how the linear systems (3.27) with unknowns $\alpha_0^\pm, \alpha_1^\pm, a_c^\pm, (\boldsymbol{\nabla} a)_c^\pm$ is explicitely written in 2D. For the sake of completeness, the expressions for $\alpha_1^\pm$ and $\alpha_0^\pm$ are stated — which are obtained with SymPy (the python computer algebra library).

Let $(\boldsymbol{\nabla} a)_c^\pm = \begin{pmatrix} a_x^\pm \\ a_y^\pm \end{pmatrix}$, $\boldsymbol{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$, and δx, δy be the components of $\delta \boldsymbol{x}_{cd} = \boldsymbol{x}_d - \boldsymbol{x}_c = \begin{pmatrix} x_d - x_c \\ y_d - y_c \end{pmatrix} =: \begin{pmatrix} \delta x \\ \delta y \end{pmatrix}$.

$$\alpha_1^\pm - \frac{1}{\Delta\xi_c(1-\Delta\xi_c)} \left(\left(a_x^\pm v_x + a_y^\pm v_y\right)(\Delta t_1 - \Delta t)\Delta t \sum_d \mathring{V}_{cd} \right) = 0, \tag{3.59a}$$

$$\frac{1}{2}\alpha_1^\pm + \alpha_0^\pm - \sum_d \mathring{V}_{cd}\Delta t_1 a_c^\pm - \sum_d (\hat{x}_{cd}(\Delta t_1) - x_c)\mathring{V}_{cd}\Delta t_1 a_x^\pm - \sum_d (\hat{y}_{cd}(\Delta t_1) - y_c)\mathring{V}_{cd}\Delta t_1 a_y^\pm = 0, \tag{3.59b}$$

$$\frac{1}{2}(1 \mp 1)\alpha_1^\pm + \alpha_0^\pm = \alpha_\pm, \tag{3.59c}$$

$$\sum_d w_{cd}(x_d - x_c)a_c^\pm + \sum_d w_{cd}\delta x^2 a_x^\pm + \sum_d w_{cd}\delta x \delta y a_y^\pm = \sum_d w_{cd}(x_d - x_c), \tag{3.59d}$$

$$\sum_d w_{cd}(y_d - y_c)a_c^\pm + \sum_d w_{cd}\delta x \delta y a_x^\pm + \sum_d w_{cd}\delta y^2 a_y^\pm = \sum_d x_{cd}(y_d - y_c). \tag{3.59e}$$

Which is expressed in terms of $AX = b$ with

$$A^{\pm} = \begin{pmatrix} 1 & 0 & 0 & A_{14} & A_{14} \\ \frac{1}{2} & 1 & A_{23} & A_{24} & A_{25} \\ \frac{1}{2}(1 \mp 1) & 1 & 0 & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & A_{45} \\ 0 & 0 & A_{53} & A_{45} & A_{55} \end{pmatrix} \tag{3.60}$$

$$= \begin{pmatrix} 1 & 0 & 0 & -\frac{v_x(\Delta t_1 - \Delta t)\Delta t \sum_d \mathring{V}_{cd}}{\Delta\xi_c(1-\Delta\xi_c)} & -\frac{v_y(\Delta t_1 - \Delta t)\Delta t \sum_d \mathring{V}_{cd}}{\Delta\xi_c(1-\Delta\xi_c)} \\ \frac{1}{2} & 1 & -\sum_d \mathring{V}_{cd}\Delta t_1 & -\sum_d(\hat{x}_{cd}(\Delta t_1) - x_c)\mathring{V}_{cd}\Delta t_1 & -\sum_d(\hat{y}_{cd}(\Delta t_1) - y_c)\mathring{V}_{cd}\Delta t_1 \\ \frac{1}{2}(1 \mp 1) & 1 & 0 & 0 & 0 \\ 0 & 0 & \sum_d w_{cd}(x_d - x_c) & \sum_d w_{cd}\delta x^2 & \sum_d w_{cd}\delta x\delta y \\ 0 & 0 & \sum_d w_{cd}(y_d - y_c) & \sum_d w_{cd}\delta x\delta y & \sum_d w_{cd}\delta y^2 \end{pmatrix},$$

$$X^{\pm} = (\alpha_1^{\pm}, \alpha_0^{\pm}, a_c^{\pm}, a_x^{\pm}, a_y^{\pm})^t, \tag{3.61}$$

$$b^{\pm} = (0, 0, b_3, b_4, b_5)^t = (0, 0, \alpha_{\pm}, \sum_d w_{cd}(x_d - x_c), \sum_d x_{cd}(y_d - y_c))^t. \tag{3.62}$$

Solving these systems with SymPy by inverting the 5×5 matrices $A^{\pm}$ finally leads to $\alpha_1^{\pm}$, $\alpha_0^{\pm}$, which are given in the corresponding SymPy output in figures 3.20 and 3.21.

```
---
X1 =
0.5*(-A14*A23*A45*b5 + A14*A23*A55*b4 + A14*A25*A43*b5 - A14*A25*A53*b4 +
A14*A43*A55*b3 - A14*A45*A53*b3 + A15*A23*A44*b5 - A15*A23*A45*b4 - A15*A24*A43*b5
+ A15*A24*A53*b4 - A15*A43*A45*b3 + A15*A44*A53*b3)/(0.25*A14*A43*A55 -
0.25*A14*A45*A53 - 0.25*A15*A43*A45 + 0.25*A15*A44*A53 - 0.5*A23*A44*A55 +
0.5*A23*A45**2 + 0.5*A24*A43*A55 - 0.5*A24*A45*A53 - 0.5*A25*A43*A45 +
0.5*A25*A44*A53)
---
X2 =
(0.5*A14*A23*A45*b5 - 0.5*A14*A23*A55*b4 - 0.5*A14*A25*A43*b5 + 0.5*A14*A25*A53*b4
- 0.25*A14*A43*A55*b3 + 0.25*A14*A45*A53*b3 - 0.5*A15*A23*A44*b5 +
0.5*A15*A23*A45*b4 + 0.5*A15*A24*A43*b5 - 0.5*A15*A24*A53*b4 + 0.25*A15*A43*A45*b3
- 0.25*A15*A44*A53*b3 - 0.5*A23*A44*A55*b3 + 0.5*A23*A45**2*b3 +
0.5*A24*A43*A55*b3 - 0.5*A24*A45*A53*b3 - 0.5*A25*A43*A45*b3 + 0.5*A25*A44*A53*b3)/
(0.25*A14*A43*A55 - 0.25*A14*A45*A53 - 0.25*A15*A43*A45 + 0.25*A15*A44*A53 -
0.5*A23*A44*A55 + 0.5*A23*A45**2 + 0.5*A24*A43*A55 - 0.5*A24*A45*A53 -
0.5*A25*A43*A45 + 0.5*A25*A44*A53)
---
```

Figure 3.20: Sympy output for linear system (+), where $X1 = \alpha_1^+$ and $X2 = \alpha_0^+$.

```
---
X1 =
(0.5*A14*A23*A45*b5 - 0.5*A14*A23*A55*b4 - 0.5*A14*A25*A43*b5 + 0.5*A14*A25*A53*b4
- 0.5*A14*A43*A55*b3 + 0.5*A14*A45*A53*b3 - 0.5*A15*A23*A44*b5 +
0.5*A15*A23*A45*b4 + 0.5*A15*A24*A43*b5 - 0.5*A15*A24*A53*b4 + 0.5*A15*A43*A45*b3
- 0.5*A15*A44*A53*b3)/(0.25*A14*A43*A55 - 0.25*A14*A45*A53 - 0.25*A15*A43*A45 +
0.25*A15*A44*A53 + 0.5*A23*A44*A55 - 0.5*A23*A45**2 - 0.5*A24*A43*A55 +
0.5*A24*A45*A53 + 0.5*A25*A43*A45 - 0.5*A25*A44*A53)
---
X2 =
b3
---
```

Figure 3.21: Sympy output for linear system (–), where $X1 = \alpha_1^-$ and $X2 = \alpha_0^-$.

Conclusion

Conclusion

A novel discretization for the explicit, upwind advection operator (1) of the GEEC solver has been introduced. More precisely the GEEC formalism defines a direct-ALE scheme for an arbitrary number of fluids, and thus acts on arbitrarily deformed Cartesian grids. The direct-ALE formulation of GEEC provides a second-order Lagrangian description, whereas the Eulerian description is defined by the advection operator (1), which is only at first-order. The present work adds two techniques to improve the formulation in (1): (i) the co-mesh strategy [69], which reduces mesh imprinting; (ii) the ND-SAB reconstruction algorithm, which defines a straight forward second-order reconstruction technique and represents an alternative strategy to classical directional splitting methods. The main objective of SAB is to reduce numerical wetting in the advection step of a hydrodynamic solver.

The present ND-SAB reconstruction algorithm was tested on uniform Cartesian grids and constant velocity fields. Extensions to the general case of arbitrary meshes and non-constant velocities are work in progress. The current description of the ND-SAB algorithm does not guarantee the validity of the (local) Discrete Maximum Principle (DMP) property in the general case. Indeed, overshooting has been observed in experiments on arbitrarily deformed Cartesian grids and non-constant, but divergence free velocity fields. Let it be emphasized that the hereby introduced advection operator is described for an Eulerian formalism, which is fully compatible with the multi-fluid ALE GEEC solver, and acts on the advection step in GEEC. It is not clear yet whether the issues of overshooting in the present ND-SAB algorithm will be worked out in GEEC by the ALE description or if additional flux corrections are needed. The main difference between working on Eulerian grids and the ALE formulation of the upwind advection operator (1), concerns the flux support, which in this case, is given by arbitrarily deformed quadrilaterals, due to node velocities. In a first approach, this only changes the computations of the centroids on which estimated values are calculated.

As discussed in chapter three, a strategy that corrects the DMP property on Eulerian grids should be considered. Furthermore, it is more likely to guarantee a good behavior of GEEC, once the DMP property is satisfied. Especially as the present ND-SAB reconstruction algorithm is amenable to changes, while preserving

the SAB philosophy on: (i) choosing a different gradient reconstruction; (ii) adapting the computations of the *bounds* for the SAB clipping in the 1D reference frame (upwind neighbors are aggregated to the right and downwind neighbors to the left); (iii) correcting redistribution weights of the total flux amount that is redistributed to the face fluxes (in the last step of the algorithm).

In conclusion, the ND-SAB limitation strategy is an interesting tool to reduce numerical wetting in advection problems, while preserving compact supports, and it is compatible with the co-mesh approach, which reduces mesh imprinting. The main advantage of these techniques is that the physical fields are kept collocated and thus introduce no further complexity to the GEEC solver. Let it be emphasized that both the co-mesh and the ND-SAB limitation strategy are not limited to the GEEC solver, but are designed for more general use of explicit upwind advection operators.

Bibliography

[1] F. G. Shuman, "History of numerical weather prediction at the national meteorological center," *Weather and Forecasting*, vol. 4, no. 3, pp. 286–296, 1989.

[2] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis, "Cfd vision 2030 study: a path to revolutionary computational aerosciences," 2014.

[3] R. Kimura, "Numerical weather prediction," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 90, no. 12-15, pp. 1403–1414, 2002.

[4] F. H. Harlow, M. Evans, and R. D. Richtmyer, *A machine calculation method for hydrodynamic problems*. Los Alamos Scientific Laboratory of the University of California, 1955.

[5] F. H. Harlow, "Hydrodynamic problems involving large fluid distortions," *Journal of the ACM (JACM)*, vol. 4, no. 2, pp. 137–142, 1957.

[6] F. H. Harlow, "Fluid dynamics in group t-3 los alamos national laboratory:(la-ur-03-3852)," *Journal of Computational Physics*, vol. 195, no. 2, pp. 414–433, 2004.

[7] L. Euler, "Principes généraux du mouvement des fluides," *Mémoires de l'Académie des Sciences de Berlin*, pp. 274–315, 1757.

[8] G. G. Stokes, "On the steady motion of incompressible fluids," *Transactions of the Cambridge Philosophical Society*, vol. 7, p. 439, 1848.

[9] J. L. de Lagrange, *Mécanique analytique*, vol. 1. Mallet-Bachelier, 1853.

[10] P.-H. Maire, "Contribution to the numerical modeling of inertial confinement fusion," tech. rep., Bordeaux-1 Univ., 2011.

[11] B. Zohuri, *Inertial confinement fusion driven thermonuclear energy*. Springer, 2017.

[12] F. L. Addessio, J. R. Baumgardner, J. K. Dukowicz, N. L. Johnson, B. A. Kashiwa, R. M. Rauenzahn, and C. Zemach, "Caveat: A computer code for fluid dynamics problems with large distortion and internal slip," *Unknow*, 1992.

[13] S. Atzeni, "The physical basis for numerical fluid simulations in laser fusion," *Plasma physics and controlled fusion*, vol. 29, no. 11, p. 1535, 1987.

[14] B. Després and C. Mazeran, "Lagrangian gas dynamics in two dimensions and lagrangian systems," *Archive for Rational Mechanics and Analysis*, vol. 178, no. 3, pp. 327–372, 2005.

[15] P.-H. Maire, R. Abgrall, J. Breil, and J. Ovadia, "A cell-centered lagrangian scheme for two-dimensional compressible flow problems," *SIAM Journal on Scientific Computing*, vol. 29, no. 4, pp. 1781–1824, 2007.

[16] D. E. Burton, N. R. Morgan, T. C. Carney, and M. A. Kenamond, "Reduction of dissipation in lagrange cell-centered hydrodynamics (cch) through corner gradient reconstruction (cgr)," *Journal of Computational Physics*, vol. 299, pp. 229–280, 2015.

[17] C. Paulin, J.-P. Braeunig, and R. Motte, "Isentropic correction for collocated lagrange-remap scheme," *Computers & Mathematics with Applications*, vol. 78, no. 2, pp. 623–642, 2019.

[18] J.-P. Braeunig, "Reducing the entropy production in a collocated lagrange–remap scheme," *Journal of Computational Physics*, vol. 314, pp. 127–144, 2016.

[19] N. R. Morgan, K. N. Lipnikov, D. E. Burton, and M. A. Kenamond, "A lagrangian staggered grid godunov-like approach for hydrodynamics," *Journal of Computational Physics*, vol. 259, pp. 568–597, 2014.

[20] E. F. Toro, *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.

[21] C. W. Hirt, A. A. Amsden, and J. Cook, "An arbitrary lagrangian-eulerian computing method for all flow speeds," *Journal of computational physics*, vol. 14, no. 3, pp. 227–253, 1974.

[22] A. J. Barlow, P.-H. Maire, W. J. Rider, R. N. Rieben, and M. J. Shashkov, "Arbitrary lagrangian–eulerian methods for modeling high-speed compressible multimaterial flows," *Journal of Computational Physics*, vol. 322, pp. 603–665, 2016.

[23] D. Fressmann and P. Wriggers, "Advection approaches for single-and multi-material arbitrary lagrangian–eulerian finite element procedures," *Computational Mechanics*, vol. 39, no. 2, pp. 153–190, 2007.

[24] F. Witherden, P. Vincent, and A. Jameson, "High-order flux reconstruction schemes," in *Handbook of numerical analysis*, vol. 17, pp. 227–263, Elsevier, 2016.

[25] J. P. Boris and D. L. Book, "Flux-corrected transport. i. SHASTA, a fluid transport algorithm that works," *Journal of computational physics*, vol. 11, no. 1, pp. 38–69, 1973.

[26] S. T. Zalesak, "Fully multidimensional flux-corrected transport algorithms for fluids," *Journal of computational physics*, vol. 31, no. 3, pp. 335–362, 1979.

[27] D. Kuzmin, R. Löhner, and S. Turek, *Flux-corrected transport: principles, algorithms, and applications*. Springer, 2012.

[28] P. Roe, "My way: A computational autobiography," *Communications on Applied Mathematics and Computation*, vol. 2, no. 3, pp. 321–340, 2020.

[29] D. Sidilkover and P. Roe, "Unification of some advection schemes in two dimensions.," tech. rep., INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING HAMPTON VA, 1995.

[30] J. Manzanero, G. Rubio, E. Ferrer, and E. Valero, "Dispersion-dissipation analysis for advection problems with nonconstant coefficients: Applications to discontinuous galerkin formulations," *SIAM Journal on Scientific Computing*, vol. 40, no. 2, pp. A747–A768, 2018.

[31] H. J. Paillere, "Multidimensional upwind residual distribution schemes for the euler and navier-stokes equations on unstructured grids," 1995.

[32] H. Deconinck, K. Sermeus, and R. Abgrall, "Status of multidimensional upwind residual distribution schemes and applications in aeronautics," in *Fluids 2000 Conference and Exhibit*, p. 2328, 2000.

[33] H. Deconinck and M. Ricchiuto, "Residual distribution schemes: foundations and analysis," *Encyclopedia of computational mechanics*, 2004.

[34] R. Abgrall, "Residual distribution schemes: current status and future trends," *Computers & Fluids*, vol. 35, no. 7, pp. 641–669, 2006.

[35] R. Abgrall, "A combination of residual distribution and the active flux formulations or a new class of schemes that can combine several writings of the same hyperbolic problem: application to the 1d euler equations," *arXiv preprint arXiv:2011.12572*, 2020.

[36] T. A. Eymann and P. L. Roe, "Multidimensional active flux schemes," in *21st AIAA computational fluid dynamics conference*, p. 2940, 2013.

[37] P. L. Roe, T. Lung, and J. Maeng, "New approaches to limiting," in *22nd AIAA Computational Fluid Dynamics Conference*, p. 2913, 2015.

[38] P. L. Roe, J. Maeng, and D. Fan, "Comparing active flux and discontinuous galerkin methods for compressble flow," in *2018 AIAA Aerospace Sciences Meeting*, p. 0836, 2018.

[39] W. Barsukow, J. Hohm, C. Klingenberg, and P. L. Roe, "The active flux scheme on cartesian grids and its low mach number limit," *Journal of Scientific Computing*, vol. 81, no. 1, pp. 594–622, 2019.

[40] K. H. Kim and C. Kim, "Accurate, efficient and monotonic numerical methods for multi-dimensional compressible flows: Part ii: Multi-dimensional limiting process," *Journal of computational physics*, vol. 208, no. 2, pp. 570–615, 2005.

[41] J. S. Park, S.-H. Yoon, and C. Kim, "Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids," *Journal of Computational Physics*, vol. 229, no. 3, pp. 788–812, 2010.

[42] B. Van Leer, "Towards the ultimate conservative difference scheme. iv. a new approach to numerical convection," *Journal of computational physics*, vol. 23, no. 3, pp. 276–299, 1977.

[43] C. Kühnlein, W. Deconinck, R. Klein, S. Malardel, Z. P. Piotrowski, P. K. Smolarkiewicz, J. Szmelter, and N. P. Wedi, "Fvm 1.0: a nonhydrostatic finite-volume dynamical core for the ifs," *Geoscientific model development*, vol. 12, pp. 651–676, 2019.

[44] P. K. Smolarkiewicz, "A fully multidimensional positive definite advection transport algorithm with small implicit diffusion," *Journal of Computational Physics*, vol. 54, no. 2, pp. 325–362, 1984.

[45] P. K. Smolarkiewicz, "Multidimensional positive definite advection transport algorithm: an overview," *International journal for numerical methods in fluids*, vol. 50, no. 10, pp. 1123–1144, 2006.

[46] W. Crowley, "Second-order numerical advection," *Journal of Computational Physics*, vol. 1, no. 4, pp. 471–484, 1967.

[47] P. K. Smolarkiewicz, "On forward-in-time differencing for fluids," *Monthly weather review*, vol. 119, no. 10, pp. 2505–2510, 1991.

[48] P. K. Smolarkiewicz and P. J. Rasch, "Monotone advection on the sphere: An eulerian versus semi-lagrangian approach," *Journal of the Atmospheric Sciences*, vol. 48, no. 6, pp. 793–810, 1991.

[49] P. K. Smolarkiewicz and G. A. Grell, "A class of monotone interpolation schemes," *Journal of Computational Physics*, vol. 101, no. 2, pp. 431–440, 1992.

[50] C. J. Tremback, J. Powell, W. R. Cotton, and R. A. Pielke, "The forward–in-time upstream advection scheme: Extension to higher orders," *Monthly Weather Review*, vol. 115, no. 2, pp. 540–555, 1987.

[51] J. Szmelter and P. K. Smolarkiewicz, "Mpdata error estimator for mesh adaptivity," *International journal for numerical methods in fluids*, vol. 50, no. 10, pp. 1269–1293, 2006.

[52] J. Szmelter and M. Gillard, "A multidimensional positive definite remapping algorithm for unstructured meshes," *Computers & Fluids*, vol. 200, p. 104454, 2020.

[53] T. Vazquez-Gonzalez, A. Llor, and C. Fochesato, "A mimetic numerical scheme for multi-fluid flows with thermodynamic and geometric compatibility on an arbitrarily moving grid," *International Journal of Multiphase Flow*, vol. 132, p. 103324, 2020.

[54] A. Llor, A. Claisse, and C. Fochesato, "Energy preservation and entropy in lagrangian space-and time-staggered hydrodynamic schemes," *Journal of Computational Physics*, vol. 309, pp. 324–349, 2016.

[55] A. Marboeuf, *Schémas ALE multi-matériaux totalement conservatifs pour l'hydrodynamique*. PhD thesis, Université Paris-Saclay (ComUE), 2018.

[56] A. Marboeuf, A. Claisse, and P. Le Tallec, "Conservative and entropy controlled remap for multi-material ale simulations with space-staggered schemes," *Journal of Computational Physics*, vol. 390, pp. 66–92, 2019.

[57] T. Vazquez-Gonzalez, *Conservative and mimetic numerical schemes for compressible multiphase flows simulation*. Theses, Université Paris-Saclay, June 2016.

[58] K. Laurent, *Étude de nouveaux schémas numériques pour la simulation des écoulements à rapport de mobilités défavorable dans un contexte EOR*. PhD thesis, Université Paris-Saclay, 2019.

[59] K. Laurent, É. Flauraud, C. Preux, Q. H. Tran, and C. Berthon, "Design of coupled finite volume schemes minimizing the grid orientation effect in reservoir simulation," *Journal of Computational Physics*, vol. 425, p. 109923, 2021.

[60] R. Eymard, C. Guichard, and R. Masson, "Grid orientation effect in coupled finite volume schemes," *IMA Journal of Numerical Analysis*, vol. 33, no. 2, pp. 582–608, 2013.

[61] G. R. Shubin and J. B. Bell, "An analysis of the grid orientation effect in numerical simulation of miscible displacement," *Computer methods in applied mechanics and engineering*, vol. 47, no. 1-2, pp. 47–71, 1984.

[62] F. S. V. Hurtado, C. R. Maliska, and A. F. C. da Silva, "On the factors influencing the grid orientation effect in reservoir simulation," in *19th International Congress of Mechanical Engineering*, pp. 5–9, ABCM Brasilia, 2007.

[63] K.-A. Lie and S. Noelle, "An improved quadrature rule for the flux-computation in staggered central difference schemes in multidimensions," *Journal of scientific computing*, vol. 18, no. 1, pp. 69–81, 2003.

[64] S. Dellacherie, P. Omnes, and F. Rieper, "The influence of cell geometry on the godunov scheme applied to the linear wave equation," *Journal of Computational Physics*, vol. 229, no. 14, pp. 5315–5338, 2010.

[65] M. Berndt, K. Lipnikov, M. Shashkov, M. F. Wheeler, and I. Yotov, "Superconvergence of the velocity in mimetic finite difference methods on quadrilaterals," *SIAM Journal on Numerical Analysis*, vol. 43, no. 4, pp. 1728–1749, 2005.

[66] D. Bouche, J.-M. Ghidaglia, and F. Pascal, "Error estimate and the geometric corrector for the upwind finite volume method applied to the linear advection equation," *SIAM Journal on Numerical Analysis*, vol. 43, no. 2, pp. 578–603, 2005.

[67] F. Pascal, "On supra-convergence of the finite volume method for the linear advection problem," in *ESAIM: proceedings*, vol. 18, pp. 38–47, EDP Sciences, 2007.

[68] C. Paulin, A. Llor, T. Vazquez-Gonzalez, J.-P. Perlat, and É. Heulhard de Montigny, "Doubly-monotone constraint on interpolators: bridging second-order to singularity preservation to cancel "numerical wetting" in transport," In preparation.

[69] C. Paulin, É. Heulhard de Montigny, and A. Llor, "Towards isotropic transport with co-meshes," *Coupled Systems Mechanics*, vol. 9, no. 1, pp. 63–75, 2020.

[70] M. E. Potter, M. Lamoureux, and M. Nauta, "An FDTD scheme on a face-centered-cubic (FCC) grid for the solution of the wave equation," *Journal of computational physics*, vol. 230, no. 15, pp. 6169–6183, 2011.

[71] M. Salmasi and M. Potter, "Discrete exterior calculus approach for discretizing Maxwell's equations on face-centered cubic grids for FDTD," *Journal of Computational Physics*, vol. 364, pp. 298–313, 2018.

[72] B. Hamilton and S. Bilbao, "On finite difference schemes for the 3-d wave equation using non-cartesian grids," in *Proceedings of Stockholm Musical Acoustics Conference/Sound and Music Computing Conference*, pp. 592–599, 2013.

[73] A. Kumar, "Isotropic finite-differences," *Journal of Computational Physics*, vol. 201, no. 1, pp. 109–118, 2004.

[74] K. M. Terekhov, B. T. Mallison, and H. A. Tchelepi, "Cell-centered nonlinear finite-volume methods for the heterogeneous anisotropic diffusion problem," *Journal of Computational Physics*, vol. 330, pp. 245–267, 2017.

[75] T. B. Lung and P. L. Roe, "Toward a reduction of mesh imprinting," *International Journal for Numerical Methods in Fluids*, vol. 76, no. 7, pp. 450–470, 2014.

[76] P. Roe, "Multidimensional upwinding," in *Handbook of Numerical Analysis*, vol. 18, pp. 53–80, Elsevier, 2017.

[77] R. DeBar, "Fundamentals of KRAKEN code," Tech. Rep. Report UCIR-17366, Lawrence Livermore National Laboratory, 1974.

[78] R. Warming and B. Hyett, "The modified equation approach to the stability and accuracy analysis of finite-difference methods," *Journal of computational physics*, vol. 14, no. 2, pp. 159–179, 1974.

[79] R. Leveque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge, 2002.

[80] J. W. Banks, T. Aslam, and W. J. Rider, "On sub-linear convergence for linearly degenerate waves in capturing schemes," *J. Comput. Phys.*, vol. 227, no. 14, pp. 6985–7002, 2008.

[81] T. Vazquez-Gonzalez, A. Llor, and C. Fochesato, "Ransom test results from various two-fluid schemes: is enforcing hyperbolicity a thermodynamically consistent option?," *Int. J. Multiph. Flow*, vol. 81, pp. 104–112, 2016.

[82] T. Vazquez-Gonzalez, A. Llor, and C. Fochesato, "A mimetic numerical scheme for multi-fluid flows with thermodynamic and geometric compatibility on an arbitrarily moving grid," *Int. J. Multiph. Flow*, vol. 132, p. 103324, 2020.

[83] B. I. Krasnopolsky and A. A. Lukyanov, "A conservative fully implicit algorithm for predicting slug flows," *J. Comput. Phys.*, vol. 355, pp. 597–619, 2018.

[84] R. Nourgaliev and M. Christon, "Solution algorithms for multi-fluid-flow averaged equations," Tech. Rep. Report INL/EXT-12-27187, Idaho National Laboratory, 2012.

[85] M. Panfilov and M. Rasoulzadeh, "Reconsidering remap methods," *Transp. Porous Media*, vol. 83, no. 1, pp. 73–98, 2010.

[86] Y. Xing and C. W. Shu, "A survey of high order schemes for the shallow water equations," *J. Math. Study*, vol. 47, no. 3, pp. 221–249, 2014.

[87] S. K. Godunov, "Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics," *Math. Sb.*, vol. 47(89), no. 3, pp. 271–306, 1959. Translated by I. Bohachevsky.

[88] N. P. Waterson and H. Deconinck, "Design principles for bounded higher-order convection schemes – a unified approach," *J. Comput. Phys.*, vol. 224, no. 1, pp. 182–207, 2007.

[89] F. Kemm, "A comparative study of TVD-limiters – well-known limiters and an introduction of new ones," *Internat. J. Numer. Methods Fluids*, vol. 67, no. 4, pp. 404–440, 2011.

[90] W. J. Rider, "Reconsidering remap methods," *Internat. J. Numer. Methods Fluids*, vol. 76, no. 9, pp. 587–610, 2014.

[91] D. Zhang, C. Jiang, D. Liang, and L. Cheng, "A review on TVD schemes and a refined flux-limiter for steady-state calculations," *J. Comput. Phys.*, vol. 302, pp. 114–154, 2015.

[92] B. van Leer, "Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection," *J. Comput. Phys.*, vol. 23, no. 3, pp. 276–299, 1977.

[93] J. K. Dukowicz and J. R. Baumgardner, "Incremental remapping as a transport/advection algorithm," *J. Comput. Phys.*, vol. 160, no. 1, pp. 318–335, 2000.

[94] P. H. Lauritzen, C. Erath, and R. Mittal, "On simplifying 'incremental remap'-based transport schemes," *J. Comput. Phys.*, vol. 230, no. 22, pp. 7957–7963, 2011.

[95] P. A. Ullrich, P. H. Lauritzen, and C. Jablonowski, "Some considerations for high-order 'incremental remap'-based transport schemes: edges, reconstructions, and area integration," *Internat. J. Numer. Methods Fluids*, vol. 71, no. 9, pp. 1131–1151, 2013.

[96] M. Klima, M. Kucharik, and M. Shashkov, "Local error analysis and comparison of the swept- and intersection-based remapping methods," *Commun. Comput. Phys.*, vol. 21, no. 2, pp. 526–558, 2017.

[97] D. Kuzmin, "A new perspective on flux and slope limiting in discontinuous Galerkin methods for hyperbolic conservation laws," *Comput. Methods Appl. Mech. Engrg.*, vol. 373, no. 22, p. 113569, 2021.

[98] W. F. Noh, "CEL: a time-dependent, two-space-dimensional, coupled Eulerian-Lagrange code," Tech. Rep. Report UCRL-7463, U. California, Lawrence Radiation Laboratory, 1963.

[99] M. Berger, M. Aftosmis, and S. Muman, "Analysis of slope limiters on irregular grids," in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005.

[100] P. K. Sweby, "High resolution schemes using flux limiters for hyperbolic conservation laws," *SIAM J. Numer. Anal.*, vol. 21, no. 5, pp. 995–1011, 1984.

[101] N. P. Waterson, "A symmetric formulation for flux-limited convection schemes," *Internat. J. Numer. Methods Fluids*, vol. 56, no. 8, pp. 1575–1581, 2008.

[102] P. Colella and P. R. Woodward, "The Piecewise Parabolic Method (PPM) for gas-dynamical simulations," *J. Comput. Phys.*, vol. 54, no. 15, pp. 174–201, 1984.

[103] S. Del Pino and H. Jourdren, "Arbitrary high-order schemes for the linear advection and wave equations: application to hydrodynamics and aeroacoustics," *C. R. Math.*, vol. 342, pp. 441–446, 2006.

[104] A. Marquina, "Local piecewise hyperbolic reconstruction of numerical fluxes for nonlinear scalar conservation laws," *SIAM J. Sci. Comput.*, vol. 15, no. 4, pp. 892–915, 1994.

[105] F. Xiao, T. Yabe, and T. Ito, "Constructing oscillation preventing scheme for advection equation by rational function," *Comput. Phys. Commun.*, vol. 93, no. 1, pp. 1–12, 1996.

[106] F. Xiao, Y. Honma, and T. Kono, "A simple algebraic interface capturing scheme using hyperbolic tangent function," *Internat. J. Numer. Methods Fluids*, vol. 48, no. 3, pp. 1023–1040, 2005.

[107] R. Artebrant and H. Schroll, "Limiter-free third order logarithmic reconstruction," *SIAM J. Sci. Comput.*, vol. 28, no. 1, pp. 359–381, 2006.

[108] T. Buffard and S. Clain, "Monoslope and multislope MUSCL methods for unstructured meshes," *J. Comput. Phys.*, vol. 229, no. 10, pp. 3745–3776, 2010.

[109] P. Colella and M. D. Sekora, "A limiter for PPM that preserves accuracy at smooth extrema," *J. Comput. Phys.*, vol. 227, no. 15, pp. 7069–7076, 2008.

[110] M. D. Sekora and P. Colella, "Extremum-preserving limiters for MUSCL and PPM," 2009.

[111] A. J. Kriel, "Error analysis of flux limiter schemes at extrema," *J. Comput. Phys.*, vol. 328, pp. 371–386, 2017.

[112] B. Després and F. Lagoutière, "Un schéma non linéaire anti-dissipatif pour l'équation d'advection linéaire," *C. R. Acad. Sci. Ser. I: Math.*, vol. 328, no. 10, pp. 939–943, 1999.

[113] B. Després and F. Lagoutière, "Contact discontinuity capturing schemes for linear advection and compressible gas dynamics," *J. Sci. Comput.*, vol. 16, no. 10, pp. 939–943, 2001.

[114] F. Alouges, F. De Vuyst, G. Le Coq, and E. Lorin, "The reservoir technique: a way to make Godunov-type schemes zero or very low diffuse. Application to Colella-–Glaz solver," *Eur. J. Mech. B Fluids*, vol. 27, no. 6, pp. 643–664, 2008.

[115] K.-A. Lie and S. Noelle, "On the artificial compression method for second-order nonoscillatory central difference schemes for systems of conservation laws," *SIAM J. Sci. Comput.*, vol. 24, no. 4, pp. 1157–1174, 2003.

[116] G. Jiang and C. Shu, "Efficient implementation of weighted ENO schemes," *J. Comput. Phys.*, vol. 126, no. 1, pp. 202–228, 1996.

[117] D. Bouche, G. Bonnaud, and D. Ramos, "Comparison of numerical schemes for solving the advection equation," *Appl. Math. Lett.*, vol. 16, no. 2, pp. 147–154, 2003.

[118] D. Book, J. Boris, and K. Hain, "Flux-corrected transport II: generalizations of the method," *J. Comput. Phys.*, vol. 18, no. 3, pp. 248–283, 1975.

[119] S. Zalesak, "A preliminary comparison of modem shock-capturing schemes: linear advection," in *Advances in Computer Methods for Partial Differential Equations* (R. Vichnevetsky and R. Stepleman, eds.), Handbook of Numerical Analysis, pp. –1, IMACS, New Brunswick, 2016.

[120] B. Leonard, "The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection," *Comput. Methods Appl. Mech. Engrg.*, vol. 88, no. 1, pp. 17–74, 1991.

[121] A. Suresh and H. T. Huynh, "Accurate monotonicity-preserving schemes with Runge–Kutta time stepping," *J. Comput. Phys.*, vol. 136, no. 1, pp. 83–99, 1997.

[122] A. Harten, "High resolution schemes for hyperbolic conservation laws," *Journal of computational physics*, vol. 135, no. 2, pp. 260–278, 1997.

[123] P. K. Sweby, "High resolution schemes using flux limiters for hyperbolic conservation laws," *SIAM journal on numerical analysis*, vol. 21, no. 5, pp. 995–1011, 1984.

[124] B. Després and F. Lagoutière, "Un schéma non linéaire anti-dissipatif pour l'équation d'advection linéaire," *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, vol. 328, no. 10, pp. 939–943, 1999.

[125] M. Berger, M. Aftosmis, and S. Muman, "Analysis of slope limiters on irregular grids," in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, p. 490, 2005.

[126] M. Hubbard, "Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids," *Journal of Computational Physics*, vol. 155, no. 1, pp. 54–74, 1999.

[127] S. Clain, "Finite volume L^∞-stability for hyperbolic scalar problems," *preprint HAL available at http://hal. archives-ouvertes. fr/hal-00467650/fr*, 2010.

[128] T. Buffard and S. Clain, "Monoslope and multislope MUSCL methods for unstructured meshes," *Journal of Computational Physics*, vol. 229, no. 10, pp. 3745–3776, 2010.

[129] D. W. Peaceman and H. H. Rachford, Jr, "The numerical solution of parabolic and elliptic differential equations," *Journal of the Society for industrial and Applied Mathematics*, vol. 3, no. 1, pp. 28–41, 1955.

[130] J. L. F. Azevedo, L. F. Figueira da Silva, and D. Strauss, "Order of accuracy study of unstructured grid finite volume upwind schemes," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 32, no. 1, pp. 78–93, 2010.

[131] F. Haider, J.-P. Croisille, and B. Courbet, "Stability analysis of the cell centered finite-volume muscl method on unstructured grids," *Numerische Mathematik*, vol. 113, no. 4, pp. 555–600, 2009.

[132] T. Barth, R. Herbin, and M. Ohlberger, "Finite volume methods: foundation and analysis," *Encyclopedia of Computational Mechanics Second Edition*, pp. 1–60, 2018.

[133] R. Eymard, T. Gallouët, and R. Herbin, "Finite volume methods," *Handbook of numerical analysis*, vol. 7, pp. 713–1018, 2000.

[134] S. May and M. Berger, "Two-dimensional slope limiters for finite volume schemes on non-coordinate-aligned meshes," *SIAM Journal on Scientific Computing*, vol. 35, no. 5, pp. A2163–A2187, 2013.

[135] A. Giuliani and L. Krivodonova, "A moment limiter for the discontinuous galerkin method on unstructured triangular meshes," *SIAM Journal on Scientific Computing*, vol. 41, no. 1, pp. A508–A537, 2019.

[136] A. Giuliani and L. Krivodonova, "A moment limiter for the discontinuous galerkin method on unstructured tetrahedral meshes," *Journal of Computational Physics*, vol. 404, p. 109106, 2020.

[137] D. Kuzmin, "A new perspective on flux and slope limiting in discontinuous galerkin methods for hyperbolic conservation laws," *Computer Methods in Applied Mechanics and Engineering*, vol. 373, p. 113569, 2021.

[138] H. Deconinck, P. L. Roe, and R. Struijs, "A multidimensional generalization of roe's flux difference splitter for the euler equations," *Computers & fluids*, vol. 22, no. 2-3, pp. 215–222, 1993.

[139] R. J. LeVeque, "Wave propagation algorithms for multidimensional hyperbolic systems," *Journal of computational physics*, vol. 131, no. 2, pp. 327–353, 1997.

[140] J. O. Langseth and R. J. LeVeque, "A wave propagation method for three-dimensional hyperbolic conservation laws," *Journal of Computational Physics*, vol. 165, no. 1, pp. 126–166, 2000.

[141] P. Bochev and M. Shashkov, "Constrained interpolation (remap) of divergence-free fields," *Computer methods in applied mechanics and engineering*, vol. 194, no. 2-5, pp. 511–530, 2005.

[142] M. Kucharik, M. Shashkov, and B. Wendroff, "An efficient linearity-and-bound-preserving remapping method," *Journal of computational physics*, vol. 188, no. 2, pp. 462–471, 2003.

[143] L. Margolin and M. Shashkov, "Second-order sign-preserving conservative interpolation (remapping) on general grids," *Journal of Computational Physics*, vol. 184, no. 1, pp. 266–298, 2003.

[144] A. Cervone, S. Manservisi, and R. Scardovelli, "An optimal constrained approach for divergence-free velocity interpolation and multilevel vof method," *Computers & fluids*, vol. 47, no. 1, pp. 101–114, 2011.

[145] A. Giuliani and L. Krivodonova, "Adaptive mesh refinement on graphics processing units for applications in gas dynamics," *Journal of Computational Physics*, vol. 381, pp. 67–90, 2019.

[146] C. Leroy, G. Oger, D. Le Touzé, and B. Alessandrini, "2d validation of a high-order adaptive cartesian-grid finite-volume characteristic-flux model with embedded boundaries," *International Journal of Mathematical and Computational Sciences*, vol. 5, no. 10, pp. 1622–1628, 2011.

[147] P. Bigay, G. Oger, P.-M. Guilcher, and D. Le Touzé, "A weakly-compressible cartesian grid approach for hydrodynamic flows," *Computer Physics Communications*, vol. 220, pp. 31–43, 2017.

[148] H. Huynh, Z. J. Wang, and P. E. Vincent, "High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids," *Computers & fluids*, vol. 98, pp. 209–220, 2014.

[149] A. Bernard-Champmartin, P. Hoch, and N. Seguin, *Stabilité locale et montée en ordre pour la reconstruction de quantités volumes finis sur maillages coniques non-structurés en dimension 2*. PhD thesis, CEA, CEA/DAM/DIF, Bruyères-le-Châtel, France; Univ-Rennes1; Université Paris 6, 2020.

[150] A. Csık, M. Ricchiuto, and H. Deconinck, "A conservative formulation of the multidimensional upwind residual distribution schemes for general nonlinear conservation laws," *Journal of computational physics*, vol. 179, no. 1, pp. 286–312, 2002.

[151] M. Dumbser and C.-D. Munz, "Building blocks for arbitrary high order discontinuous galerkin schemes," *Journal of Scientific Computing*, vol. 27, no. 1, pp. 215–230, 2006.

[152] V. A. Titarev and E. F. Toro, "Ader: Arbitrary high order godunov approach," *Journal of Scientific Computing*, vol. 17, no. 1, pp. 609–618, 2002.

[153] S. Del Pino and H. Jourdren, "Arbitrary high-order schemes for the linear advection and wave equations: application to hydrodynamics and aeroacoustics," *Comptes Rendus Mathematique*, vol. 342, no. 6, pp. 441–446, 2006.

[154] F. Alauzet and A. Loseille, "A decade of progress on anisotropic mesh adaptation for computational fluid dynamics," *Computer-Aided Design*, vol. 72, pp. 13–39, 2016.

[155] P.-J. Frey and F. Alauzet, "Anisotropic mesh adaptation for cfd computations," *Computer methods in applied mechanics and engineering*, vol. 194, no. 48-49, pp. 5068–5082, 2005.

[156] S. Clain, S. Diot, and R. Loubère, "A high-order finite volume method for systems of conservation laws—multi-dimensional optimal order detection (mood)," *Journal of computational Physics*, vol. 230, no. 10, pp. 4028–4050, 2011.

[157] M. Kucharik and R. Loubère, "High-accurate and robust conservative remapping combining polynomial and hyperbolic tangent reconstructions," *Computers & Fluids*, p. 104614, 2020.

[158] B. Cockburn, G. E. Karniadakis, and C.-W. Shu, "The development of discontinuous galerkin methods," in *Discontinuous Galerkin Methods*, pp. 3–50, Springer, 2000.

[159] J. Manzanero, C. Redondo, G. Rubio, E. Ferrer, E. Valero, S. Gómez-Álvarez, and Á. Rivero-Jiménez, "A high-order discontinuous galerkin solver for multiphase flows," in *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018*, pp. 313–323, Springer, Cham, 2020.

[160] X. Liu, N. R. Morgan, and D. E. Burton, "A lagrangian discontinuous galerkin hydrodynamic method," *Computers & Fluids*, vol. 163, pp. 68–85, 2018.

[161] N. R. Morgan, X. Liu, and D. E. Burton, "A lagrangian discontinuous galerkin hydrodynamic method for higher-order elements," in *2018 AIAA Aerospace Sciences Meeting*, p. 1092, 2018.

[162] M. Dumbser, O. Zanotti, R. Loubère, and S. Diot, "A posteriori subcell limiting of the discontinuous galerkin finite element method for hyperbolic conservation laws," *Journal of Computational Physics*, vol. 278, pp. 47–75, 2014.

[163] A. M. Rueda-Ramírez, E. Ferrer, D. A. Kopriva, G. Rubio, and E. Valero, "A statically condensed discontinuous galerkin spectral element method on gauss-lobatto nodes for the compressible navier-stokes equations," *Journal of Computational Physics*, p. 109953, 2020.

[164] T. Sonar, *Mehrdimensionale ENO-Verfahren: Zur Konstruktion nichtoszillatorischer Methoden für hyberbolische Erhaltungsgleichungen*. Springer-Verlag, 2013.

[165] C.-W. Shu, "Essentially non-oscillatory and weighted essentially non-oscillatory schemes," *Acta Numerica*, vol. 29, pp. 701–762, 2020.

[166] W. Rider and D. Kothe, "Constrained minimization for monotonic reconstruction," in *13th Computational Fluid Dynamics Conference*, p. 2036, 1997.

[167] F. Haider, B. Courbet, and J.-P. Croisille, "A high-order interpolation for the finite volume method: The coupled least squares reconstruction," *Computers & Fluids*, vol. 176, pp. 20–39, 2018.

[168] P. Colella and P. R. Woodward, "The piecewise parabolic method (ppm) for gas-dynamical simulations," *Journal of computational physics*, vol. 54, no. 1, pp. 174–201, 1984.

[169] T. Vazquez-Gonzalez, A. Llor, and C. Fochesato, "A novel GEEC (Geometry, Energy, and Entropy Compatible) procedure applied to a staggered direct-ALE scheme for hydrodynamics," *European Journal of Mechanics-B/Fluids*, vol. 65, pp. 494–514, 2017.

List of Figures

1 Mesh imprinting illustrated locally with 2D advection of one cell: the shape of the numerical solution is different for different velocity directions; left: velocity in direction of the x-axis; right: velocity in diagonal direction. 11

2 Illustration of "numerical wetting": increase of the support of numerical diffusion with the scheme's time step (cells of the support are highlighted in blue). 12

1.1 Cell with its first nearest neighbors (white) and its co-cell boundaries connected to second nearest neighbors (pink). 18

1.2 Meshes and the corresponding co-meshes. Cartesian grid (top) and randomly distorted quadrilateral grid (bottom). 19

1.3 Initial Cartesian grid (a) and *co-mesh* (b). Applying the co-mesh strategy is equivalent to applying the initial transport operator on a non-tailing but volume-preserving octagonal grid (c). 20

1.4 Graph of the dimensionless coefficients of matrix M in basis $\{\boldsymbol{e}, \boldsymbol{e}_\perp\} = \{\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}, \boldsymbol{e}_\perp\}$ as a function of transport direction θ: $\boldsymbol{e} \cdot \mathbf{M} \cdot \boldsymbol{e}$ (blue), $\boldsymbol{e}_\perp \cdot \mathbf{M} \cdot \boldsymbol{e}_\perp$ (green), and $\boldsymbol{e} \cdot \mathbf{M} \cdot \boldsymbol{e}_\perp$ (red), scaled by $h\|\boldsymbol{v}\|$. 20

1.5 Representation of numerical diffusion on the transport of a "delta" function (four cells at bottom left corner) along directions $\boldsymbol{v} = \|\boldsymbol{v}\|(\cos\theta, \sin\theta)$, for $\theta = 0$, $\pi/8$ and $\pi/4$; transport over a radius of $96h$ (where h is the spatial discretization step), on a 128×128 grid, in 192 iterations, with CFL $= 0.5$. 20

1.6 Solver applied to co-mesh ($\omega = 0$). Wrong estimation of cell-volume violates conservation and monotonicity (left), compared to the correct result (right). (For details on the simulation see Fig. 1.5.) 21

1.7 Co-mesh strategy applied to a randomly distorted quadrilateral grid with ω_{opt} (details on the simulation are provided in Fig. 1.5). 24

2.1 Graphical illustration of MUSCL applied to scalar transport in 1D (reproduced from [92]). 34

2.2 Representations of (a) non-monotonic, (b) weakly monotonic, (c) strongly (non-doubly) monotonic, and (d) doubly monotonic interpolants (solid lines) in a cell constrained by neighboring-cell values. Mean cell values $\overline{y}$ are represented by white dots. In (c) and (d) interpolants that would be generated for slightly higher values of $\overline{y}$ are also represented (dotted lines) so as to visualize double monotonicity. 37
2.3 Graphical illustration in support of the Proof of Theorem 1: (a) function $y[x] \in \mathcal{M}_{II}^{\leq}(\overline{y})$ (thick line) and its intersection points (circled) with $m[x]$ (double line); (b) step function $s[x]$ (thick line) with same intersection points and means on each interval I_i; (c) "edge" step function $y_{\alpha\beta}[x]$ (thick line) of same left and right means as $s[x]$ (and $y[x]$), and with $y_{\alpha\beta}[1-\overline{y}] = y[1-\overline{y}] = y_0$. 39
2.4 2D map of the set of bounded monotonic functions $\mathcal{M}_{II}^{\leq}(\overline{y})$ in the elliptic coordinates from poles y_{00} and y_{11}: all y are within the half circle centered on $m = (y_{00} + y_{11})/2$ bounded by $y_{\alpha\beta}$ as stated in Corollary 1. The thick line segments correspond to single-slope interpolants (see subsection 2.3.2) of fixed $\overline{y}$ values ranging from 0 to $\frac{1}{2}$ in steps of $\frac{1}{16}$. The 2nd order point is the interpolant that would result from any second-order scheme over a uniform mesh. The insert is the usual Sweby diagram [100, fig. 1] of these single-slope interpolants which map two-to-one in the half circle (as represented by the corresponding lines and end points). 42
2.5 2D map (elliptic coordinates from poles y_{00} and y_{11}) of slope-and-bound and slope-and-two-bounds interpolants (SAB zone, thick lines) for fixed $\overline{y}$ values from 0 to $\frac{1}{2}$ in steps of $\frac{1}{16}$: these extend the corresponding single-slope interpolants of same $\overline{y}$ values (thin lines, see also Figure 2.4) and make a full connection from y_{00} to y_{11} running close to the $y_{\alpha\beta}$ edge. 45
2.6 Graphical representations of various doubly-monotonic Single-Slope (SS) and Slope-And-Bound (SAB) interpolators on a uniform grid: (a) to (e) usual wetting (SS) interpolators of (2.13) Min-Mod (MM), Monotonized-Centered (MC), Super-Bee (SB), Super-Bee$^+$ (SB+, see description in subsection 2.4.3), and Ultra-Bee (UB); (f) Min-Mod$^-$ interpolator (MM−, see description in subsection 2.4.3); (g) to (j) Mildly wetting Slope-And-Bound (SAB) extensions of the MC to UB interpolators in (b) to (e); (k) Order-one most diffusive interpolator (for reference); (l) to (o) Least wetting Slope-And-Bound (SAB) extensions of the MC to UB interpolators in (b) to (e). All profiles are plotted for $\overline{y}$ values from 0 to $\frac{1}{2}$ in steps of $\frac{1}{16}$. The dashed line marks the exact second-order interpolation at $\overline{y} = \frac{1}{2}$. 48

2.7 2D maps (elliptic coordinates from poles y_{00} and y_{11}) of the four doubly-monotonic interpolators MC, SB, SB+, and UB (see Figure 2.6) in their usual Single-Slope versions (dashed lines) and Slope-And-Bound extensions, Balanced (dotted lines) and Maximal (solid lines). 49
2.8 2D map (elliptic coordinates from poles y_{00} and y_{11}) of evenly-sampled limiting cones of double monotonicity (thin lines) defined by slopes $g_\pm[\overline{y}_i, \overline{y}]$ (3.55) which constrain SAB interpolators: when varying $\overline{y}$, the trajectory of a doubly monotonic interpolator must, at any point, stay within the local limiting cone. Dotted lines are SAB interpolants of varying slopes and gien $\overline{y}$ values from 0 to $\frac{1}{2}$ in steps of $\frac{1}{16}$). The dark, medium and light gray areas map the interpolants accessible from the second-order point (for uniform mesh) as given respectively by Sweby [100, fig. 1b], by doubly-monotonic Single-Slope interpolators (SS), and by doubly-monotonic Slope-And-Bound interpolators (SAB) in (2.16). The corresponding curves and regions for the SS interpolators are also mapped on the usual Sweby diagram [100, fig. 1] in insert. 51
2.9 Square-wave profiles after uniform cyclic transport over one period of 100 cells at Courant number $\frac{1}{2}$: rows (a–b), (c–d), (e–f), and (g–h) computed with respective "Monotonised Centered," "Super-Bee," "Super-Bee$^+$," and "Ultra-Bee" interpolators (MC, SB, SB+, and UB in corresponding columns of Figure 2.6), each limited as "Single-Slope," "Balanced Slope-and-Bound," and "Maximal Slope-and-Bound" respectively in solid, dotted, and dashed rows (SS, SAB-Bal, and SAB-Max in corresponding rows of Figure 2.6). Initial profile is represented by thin lines in column (a–g). Profile differences between SS, SAB-Bal, and SAB-Max versions appear marginal on standard profiles $y[x]$ in column (a–g) but are revealed in logarithmic scale $\log\Big(y[x](1-y[x])\Big)$ in column (b–h). Significant numerical wetting is present at similar strengths with all the single-slope interpolators, but it is absent with all the slope-and-bound interpolators. 52
2.10 Same as Figure 2.9 with Courant number $\frac{1}{10}$: profile asymmetry is now present and numerical diffusion is now higher for the MC interpolators. 53

2.11 Same as Figure 2.9 on a random mesh (1:3 spread of mesh width, Courant number $\frac{1}{2}$ with respect to smallest cell): some profile asymmetry is present as in Figure 2.10 but now, the balanced slope-and-bound interpolators (dotted lines) reduce but do not fully cancel the numerical wetting effect. A form of terracing also appear in the wet zones. 54
2.12 Time evolution of mean front widths $\ell_{\text{Frt}}(t)$ (2.25a) in the simulations of Figures 2.9, 2.10, 2.11: (a) diffusive interpolators O1, MM−, and MM (from top towards bottom in respectively solid black, red dotted, and blue dashed lines), and MC (nine lowest lines), (b) SB interpolators, (c) SB+ interpolators, (d) UB interpolators; SS, SAB-BAL, and SAB-Max versions (respectively solid black, red dotted, and blue dashed lines) in each of the MC, SB, SB+, and UB are displayed in (a) to (d); for each interpolator, the three simulation conditions (Courant numbers $\frac{1}{2}$ and $\frac{1}{10}$, and non-uniform mesh) are shown with identical line types. For legibility, time filtering was applied on some of the data sets. 56
2.13 Time evolution of front support widths $\ell_{\text{Sup}}(t)$ (2.25b) in mean cell units and in ℓ_{Frt} units in respective columns (a–e) and (b–f) from the simulations of Figures 2.9, 2.10, 2.11: (a–b) usual SS interpolators (notice expanded time scale), (c–d) balanced SAB interpolators, and (e–f) maximal SAB interpolators; Courant numbers of $\frac{1}{2}$ (solid black lines), $\frac{1}{10}$ (blue dashed lines), and $\frac{1}{2}$ on non-uniform mesh (red dotted lines); in each color and line type of each frame, interpolators are MC, SB, SB+, and UB in that order from top to bottom. For legibility, time filtering was applied on some of the data sets. 57
2.14 Sine-wave profiles after uniform cyclic transport over four periods of 100 cells computed with the interpolators of Figure 2.6 (except the order-one O1) for three numerical conditions in each frame: Courant numbers $\frac{1}{2}$, $\frac{1}{10}$, and $\frac{1}{2}$ on non-uniform mesh (in respectively black solid, red dotted, and blue dashed lines). Initial profile is represented by thin lines and $\times 10$ vertically expanded views of the maximum are provided for interpolators MM, MC, and SB. 59
2.15 Time evolution of the amplitude reduction in the sine-wave tests of Figures 2.14c to e with the MC interpolators in its usual SS, SAB-Bal, and SAB-Max extensions (respectively black solid, red dotted, and blue dashed lines). From top to bottom in each series of same line type numerical conditions are successively Courant number $\frac{1}{10}$, and Courant number $\frac{1}{2}$ with uniform and non-uniform meshes. . . . 61

2.16 Zalesak wave profiles after uniform cyclic transport over four (a–b) and ten (c–d) periods of 200 cells computed with the MC interpolator in its standard SS and modified SAB-Bal and SAB-Max versions (respectively dashed, dotted and solid thick lines) at Courant numbers $\frac{1}{2}$ and $\frac{1}{10}$ (respectively columns (a–c) and (b–d)). Thin line: initial condition. 62

3.1 Illustration of ND-SAB algorithm in four steps. 72
3.2 Illustrations of geometrical variables: centroid $\hat{\boldsymbol{x}}_{cd}$ of flux support $\mathcal{V}_{cd} = \mathring{V}_{cd}\Delta t$ (left), outward normal vector $\boldsymbol{s}_{cd}$ and upwind coefficient σ_{cd} (right). 73
3.3 Illustration of fluxes $\phi^{\mathrm{1D}}(\Delta\xi_c)$ and $\phi^{\mathrm{1D}}(\Delta\xi = 1)$ in the 1D reference frame over cell c. 76
3.4 Zones beyond the monotonicity thresholds $\alpha_c^{\pm}$ (red) and the TVD thresholds $\alpha_{\pm}$ indicating where limitation is needed for different central values (blue). 78
3.5 Left: SAB interpolator (blue) with reference point $\alpha^{*,-}$ and monotonicity threshold slope (red); right: slope support $\eta^{\star,-}$ of the SAB interpolator, and illustration of the preserved integral (blue) given by (3.31). 81
3.6 Illustration of the generic second-order Slope-And-Bound interpolator defined by the 6-tuple $(\alpha_1, \alpha_0, \eta_{\pm}, \alpha_{\pm})$ on a cell c resized to $[0, 1]$. 82
3.7 Flux support parameters l_1, l_2 and l_3 of flux integration with the generic second-order Slope-And-Bound interpolator. Different sizes of flux supports are indicated in color (violet, blue, green). 84
3.8 Initialisation of circle with radius $r = 0.7$ and density $\rho = 1$ on a Cartesian grid on $[0, 1]^2$ with 100×100 cells. Density plots of ρ (left) and $\rho(1 - \rho)$ (right). 88
3.9 Radius parameters of a cell c on a circle with radius r and center z. 89
3.10 First-order baseline. Cartesian grid on $[0, 1]^2$ with 100×100 cells, CFL= 0.5 and final time $T = 0.5$. Density plots of ρ (left), $\rho(1 - \rho)$ (middle), and $\log(|\rho(1 - \rho)|)$ (right). 90
3.11 Non-limited second-order baseline with least squares gradient reconstruction. Cartesian grid on $[0, 1]^2$ with 100×100 cells, CFL= 0.5 and final time $T = 0.5$. Density plots of ρ (left), $\rho(1 - \rho)$ (middle), and $\log(|\rho(1 - \rho)|)$ (right). 90
3.12 Second-order reconstruction with ND-SAB algorithm and flux redistribution weight (3.42a). Cartesian grid on $[0, 1]^2$ with 100×100 cells, CFL= 0.5 and final time $T = 0.5$. Density plots of ρ (left), $\rho(1 - \rho)$ (center), and $\log(|\rho(1 - \rho)|)$ (right). 92

3.13 Second-order reconstruction with ND-SAB algorithm and flux redistribution weight (3.42b). Cartesian grid on $[0,1]^2$ with 100×100 cells, CFL= 0.5 and final time $T = 0.5$. Density plots of ρ (left), $\rho(1-\rho)$ (center), and $\log(|\rho(1-\rho)|)$ (right). 92
3.14 Second-order reconstruction with ND-SAB algorithm and flux redistribution weight (3.42c). Cartesian grid on $[0,1]^2$ with 100×100 cells, CFL= 0.5 and final time $T = 0.5$. Density plots of ρ (left), $\rho(1-\rho)$ (center), and $\log(|\rho(1-\rho)|)$ (right). 92
3.15 Second-order reconstruction with ND-SAB algorithm and co-mesh. Cartesian grid on $[0,1]^2$ with 100×100 cells, CFL= 0.5 and final time $T = 0.5$. Density plots of ρ (left), $\rho(1-\rho)$ (center), and $\log(|\rho(1-\rho)|)$ (right). 93
3.16 Density plot ρ of ND-SAB second-order reconstruction with co-mesh strategy: solutions are shown on the initial grid (left), the co-meshes (center), and an optimal linear combination of initial grid and its co-meshes (right). 94
3.17 Density plot $\rho(1-\rho)$ (illustrating support of numerical diffusion) of ND-SAB second-order reconstruction with co-mesh strategy: solutions are shown on the initial grid (left), the co-meshes (center), and an optimal linear combination of initial grid and its co-meshes (right). 94
3.18 Slices in velocity directions: solutions are shown on the initial grid (bottom left), the co-mesh (bottom right) and for an optimal linear combination of initial grid and its co-mesh (top right). 95
3.19 Slices transversal to the velocity directions: solutions are shown on the initial grid (bottom left), the co-mesh (bottom right) and for an optimal linear combination of initial grid and its co-mesh (top right). 96
3.20 Sympy output for linear system (+), where $X1 = \alpha_1^+$ and $X2 = \alpha_0^+$. 102
3.21 Sympy output for linear system (–), where $X1 = \alpha_1^-$ and $X2 = \alpha_0^-$. 102
4.1 Illustration de l'empreinte du maillage appliqué localement à une maille: les formes de la solution numérique sont différentes selon la direction de la vitesse; gauche: vitesse en direction de l'axe x; droite: vitesse en direction diagonale. 134
4.2 Illustration du "mouillage numérique": augmentation du support de la diffusion numérique avec le pas de temps du schema (les mailles du support sont accentuées en bleu). 135
4.3 Simulations avec une advection d'ordre un : sur un maillage cartésien sur $[0,1]^2$ avec 100×100 mailles, CFL= 0.5 et temps final $T = 0.5$. Profils de densité ρ (gauche), $\rho(1-\rho)$ (milieux), et $\log(|\rho(1-\rho)|)$ (droite). 137

4.4 Méthode des co-mesh dans une advection d'ordre un : sur un maillage cartésien sur $[0,1]^2$ avec 100×100 mailles, CFL= 0.5 et temps final $T = 0.5$. Profils de densité ρ (gauche), $\rho(1-\rho)$ (milieux), et $\log(|\rho(1-\rho)|)$ (droite). 137
4.5 Simulations avec l'algorithme ND-SAB : sur un maillage cartésien sur $[0,1]^2$ avec 100×100 mailles, CFL= 0.5 et temps final $T = 0.5$. Profils de densité ρ (gauche), $\rho(1-\rho)$ (milieux), et $\log(|\rho(1-\rho)|)$ (droite). 138
4.6 Simulations avec l'algorithme ND-SAB combiné avec la stratégie des co-mesh : sur un maillage cartésien sur $[0,1]^2$ avec 100×100 mailles, CFL= 0.5 et temps final $T = 0.5$. Profils de densité ρ (gauche), $\rho(1-\rho)$ (milieux), et $\log(|\rho(1-\rho)|)$ (droite). 138

List of published works

Papers:

- 2021 Article (prép.), *Adapting Slope-And-Bound interpolation to multidimensional advection schemes to cancel "numerical wetting"*, C. Paulin, T. Vazquez-Gonzalez, J.-P. Perlat, A. Llor.
- 2021 Article (prép.), *Doubly-monotone constraint on interpolators: bridging second-order to singularity preservation to cancel "numerical wetting" in transport schemes*, C. Paulin, A. Llor, T. Vazquez-Gonzalez, J.-P. Perlat, É. Heulhard de Montigny.
- 2020 C. Paulin, É. Heulhard de Montigny, A. Llor, *Towards isotropic transport with co-meshes*, Coupled Systems Mechanics.

Talks:

- *Slope-And-Bound Limiter: an Extension to 2D*, ECCOMAS Congress 2020, Paris (France).
- *Towards isotropic transport with co-meshes*, ECCOMAS MSF 2019, Sarajevo (Bosnia-Herzegovina).

Posters:

- *Co-Meshes: Towards Isotropic Transport*, GAMNI 2020, Paris (France).
- *Towards isotropic transport with co-meshes*, MultiMat 2019, Trento (Italy).

À l'intention du lecteur francophone

Introduction

Advection en hydrodynamique, aérodynamique and météorologie

Avant l'ère des premiers ordinateurs, John von Neumann, Robert D. Richtmyer, Theodore von Kármán et Lewis Fry Richardson ont façonné l'histoire en tant que pionniers dans les domaines de l'ingénierie nucléaire, de l'ingénierie aérospatiale et de la météorologie [1, 2, 3]. Le début des premières simulations numériques dans les années 1950 marque un premier jalon, et dans la même décennie Francis H.Harlow est le père de la Mécanique des Fluides Numérique (MFN ou CFD pour *Computational Fluid Dynamics* en Anglais) [4, 5, 6].

Les codes CFD distinguent les formalismes eulériens et lagrangiens. L'approche eulérienne observe le mouvement du fluide sur un maillage fixe, alors que dans l'approche lagrangienne, le maillage se déplace avec la vitesse du fluide. La théorie du mouvement des fluides que nous connaissons aujourd'hui est basée sur les travaux de Leonhard Euler [7], George Gabriel Stokes [8] et Joseph-Louis Lagrange [9]. Les paragraphes suivants présentent une sélection des progrès récents dans les domaines de l'hydrodynamique, l'aérodynamique et de la météorologie.

C'est dans le domaine du génie nucléaire, que les formalismes lagrangiens ont été introduits. L'hydrodynamique lagrangienne est toujours une partie importante dans la plupart des solveurs, par exemple pour la Fusion par Confinement Inertiel (FCI) [10, 11]. CAVEAT [12, 13], GLACE [14] et EUCCLHYD [15] sont des codes célèbres de schémas de type Godunov, qui simulent numériquement la fusion laser. Cependant, la production d'entropie est une propriété intrinsèque des solveurs acoustiques de Riemann et conduit à une dissipation numérique trop grande. Différentes approches sont discutées afin de réduire cette dissipation dans [16], et dans [17, 18] le *flux isentropique* est introduit pour réduire la production d'entropie pour les ondes de détente, aussi appelées ondes de raréfaction. Une approche de type Godunov à maillage décalée est introduite dans [19]. D'autres solveurs de Riemann sont discutés dans [20]. Depuis l'introduction du formalisme Arbitraire Lagrangien Eulérien ALE [21], les avantages des deux méthodes sont combinés dans des nombreux codes hydrodynamiques [22]. Notamment les approches lagrangiennes sont bien adaptées au maintien des interfaces matérielles, alors que les approches eulériennes ne font pas l'objet de distorsions critiques de maillages. Un aperçu de l'advection dans les Méthodes des Éléments Finis (MEF) et comparé aux Méthodes des Volumes Finis (MVF) est donné dans [23]. Le présent travail se concentre sur les problèmes d'advection dans un contexte de volumes finis. L'advection décrit le transport de quantités physiques, notamment de fluides. C'est un élément important dans la discrétisation des équations d'Euler et de Navier-Stokes. Une

étape d'advection est rencontrée dans chaque solveur CFD avec ses propres défis. Le travail le plus important de discrétisation des étapes d'advection dans les schémas numériques est celui de la reconstruction du flux (FR pour *Flux Reconstruction* en anglais). Les techniques fondamentales de FR sont rappelées dans [24]. L'objectif de la plupart des reconstructions est d'améliorer la discrétisation spatiale des équations d'advection à un ordre élevé, où les oscillations numériques parasites sont évitées avec des stratégies de limitation. Les limiteurs se divisent en deux catégories: les *limiteurs de pente* limitent les fonctions de gradient reconstruites, tandis que les *limiteurs de flux* limitent les flux résultants. Le présent travail se concentre sur la limitation de pente, pour laquelle une bibliographie exhaustive est donnée dans la deuxième et la troisième partie de ce manuscrit. Les limiteurs de flux ne sont pas moins importants : par exemple, une correction du flux est proposée par la méthode FCT (pour *Flux Corrected Transport* en anglais) [25, 26], qui a été introduite dans l'ingénierie nucléaire. FCT résout l'équation de continuité tout en maintenant la non-négativité, et l'algorithme FCT sépare l'étape du transport en deux étapes: une "étape convective", où les flux numériques sont définis et une "étape antidiffusive (ou corrective)", où, comme son nom l'indique, les contre-parties antidiffusives sont calculées puis limitées. FCT représente une approche élaborée, où une formule antidiffusive est utilisée pour maintenir la positivité et les flux sont soigneusement limités sans violer la conservation. Le FCT a été largement étudié au cours des dernières décennies, et des applications à l'aérodynamique, à la météorologie et à l'océanographie ont également été développées [27].

Dans l'ingénierie aérospatiale, le développement des codes CFD s'est intensifié dans les années 1970 avec les progrès de l'industrie aéronautique [28]. Afin d'obtenir des schémas à haute résolution, les améliorations dans les étapes d'advection des schémas sont cruciales. Par exemple, la relation entre les schémas de type *volume de contrôle* et les schémas *fluctuation splitting* est introduite dans [29], et la vitesse d'advection non constante est discutée avec une analyse de von Neumann généralisée dans [30]. Les stratégies de limitation de flux les plus récentes dans les solveurs aérodynamiques dénombrent les méthodes RD (pour *Residual Distribution*) [31, 32, 33, 34, 35], AF (pour *Actif Flux*) [36, 37, 38, 39], et MLP (pour *Multidimensional Limiting Process*) [40, 41]. De plus, les techniques de type MUSCL (pour *Monotonic Upstream-centered Scheme for Conservation Laws*), introduites à l'origine dans [42], sont importantes pour l'aérodynamique. Des références plus spécifiques sur MUSCL et d'autres stratégies de limitation sont données dans la deuxième et la troisième partie du présent manuscrit.

En météorologie, les systèmes mondiaux de Prévision Numérique du Temps (PNT ou NWP pour *Numerical Weather Prediction* en anglais), tels que le système de prévision intégré (IFS pour *Integrated Forecasting System*), ont été largement développés ces dernières années. IFS utilise une advection eulérienne avec le

schéma NFT (pour *Non-Oscillatory Forward-in-Time*) [43], et une étape d'advection avec l'algorithme MPDATA (pour *Multidimensional Positive Definite Advection Transport Algorithm*) [44, 45] avec un ordre deux exact. Avant MPDATA, l'advection en météorologie était généralement basée sur des schémas de type Crowley [46, 47], qui sont des schémas d'advection dissipatifs. D'autres articles de l'auteur de MPDATA et de ses collègues devraient également être mentionnés, pour leurs idées intéressantes sur l'advection: [48] discute des algorithmes de transport, qui combinent des approches eulériennes et semi-lagrangiennes. Dans [49], des réflexions intéressantes sur *l'équivalence de l'advection-interpolation*, et une classe de schémas d'interpolation monotone inspirés de la version FCT de Tremback [50] est déduite, où, cependant, une contrainte de conservation exacte a été négligée. Plus récemment, des méthodes d'adaptabilité de maillages et de projections ont été discutés dans MPDATA avec des maillages non structurés [51, 52].

Advection quasi-isotropique à l'ordre deux dans le formalisme GEEC

La présente étude est liée au formalisme multi-fluide quasi-symplectique avec Compatibilité Géométrique, Énergétique et Entropique (GEEC) [53]. GEEC est inspiré du schéma *décalé en temps et en espace conservative* (CSTS pour *Conservative Space- and Time-Staggered* en anglais) [54], où les équations d'Euler discrètes sont dérivées en utilisant un mimétisme d'une intégrale d'action discrète. Ce formalisme conduit à la fois à une approche ALE indirecte [55, 56], où une étape lagrangienne est suivie d'une phase de projection (*remapping*) appelée "Lagrange-plus-projection" (ou encore "Lagrange-plus-remap"), et à une approche ALE directe [57], où les flux d'advection sont directement pris en compte dans les équations d'évolution. Les étapes d'advection dans la discrétisation GEEC des équations d'Euler utilisent l'opérateur d'advection amont (*upwind*) discret $\mathrm{D}_{\Delta t}$, qui est défini par des flux de type MUSCL explicites pour les champs advectés a et b avec

$$\begin{aligned}&\int_V \int_t (a \mathrm{D}_t b) \mathrm{d}\boldsymbol{x} \mathrm{d}t \\ &\rightsquigarrow a_c^{n+1} \mathrm{D}_{\Delta t} b_c^n = a_c^{n+1} \Big[V_c^{n+1} b_c^{n+1} - V_c^n b_c^n + \Delta t^{n+\frac{1}{2}} \sum_{d \in \mathcal{D}(c)} \left(\mathring{V}_{cd} b_c^n - \mathring{V}_{dc} b_d^n \right) \Big], \quad (4.1)\end{aligned}$$

avec le taux de transfer de volume $\mathring{V}_{cd}$ et le volume de la maille V_c, où c est l'index de la maille et cd designe l'index de la face orientée entre les maille c et d, et n est l'index du temps avec le pas de temps Δt. $\mathcal{D}(c)$ designe l'ensemble des mailles voisines à la maille c dans le stencil, qui est défini dans le cadre des volumes finis comme mailles aillants une face en commun avec la maille c (ce qui est détaillé dans le chapitre 1 dans le contexte de *premier plus proche voisin*).

L'objective de ces travaux est de transformer cet opérateur d'advection d'ordre un (4.1) en une formulation quasi-isotrope à l'ordre deux, qui optimise la réduction du mouillage numérique avec une stratégie de limiteur compatible. Les modifications de l'étape d'advection développées dans ces travaux doivent être compatibles avec les équations de conservation discrètes pour la masse, la quantité de mouvement et l'énergie totale du solveur GEEC, qui dérivent d'un *principe de moindre action* discrète [53, §4]:

$$\mathrm{D}_{\Delta t}\Big([\alpha\rho]_c^{\varphi n}\Big) = 0, \tag{4.2a}$$

$$\mathrm{D}_{\Delta t}\Big([\alpha\rho]_c^{\varphi n-1} u_c^{\varphi n-\frac{1}{2}}\Big) = \mathrm{RHS}, \tag{4.2b}$$

$$\mathrm{D}_{\Delta t}\Big([\alpha\rho]_c^{\varphi n} e_c^{\varphi n}\Big) = \mathrm{RHS}^\star, \tag{4.2c}$$

où φ est l'indice du fluide, α_c^φ la fraction volumique du fluide φ présente dans la maille c (avec $\sum_\varphi \alpha^\varphi = 1$). Ces équations décrivent l'advection pour la densité ρ, la vitesse du fluide $\boldsymbol{u}$ et l'énergie interne e. Les parties droites des équations — (RHS) pour la quantité de mouvement (4.2b) et (RHS*) pour l'énergie totale (4.2c) — sont pour le moment ignorés, le lecteur intéressé est renvoyé à [53], où l'algorithme complet peut être trouvé.

Compatible avec GEEC et testé dans la représentation eulérienne mono-fluide, ces travaux montrent deux possibilités pour transformer l'opérateur d'advection amont discret (4.1): (i) la stratégie de *co-mesh*, qui réduit l'anisotropie de la diffusion numérique produit par l'effet d'orientation du maillage, est discuté dans le chapitre 1; (ii) la reconstruction *Slope-And-Bound* (SAB), qui est introduite dans le chapitre 2 en 1D, et étendue dans le chapitre 3 à une formulation multidimensionnelle appelée ND-SAB. Ces deux méthodes sont compatibles l'une avec l'autre et les tests sur cette combinaison sont présentés dans le chapitre 3, où l'algorithme ND-SAB a également été appliqué aux co-mesh.

Vers une diffusion numérique quasi-isotrope

L'advection est l'ingrédient central de tous les schémas numériques pour les équations aux dérivées partielles hyperboliques et en particulier pour l'hydrodynamique. L'advection a ainsi été largement étudié dans ses fonctionnalités et pour de nombreuses applications spécifiques. En dimension supérieure à un, il est le plus souvent hanté par un artefact majeur: l'empreinte de maillage (ou *mesh imprinting* en anglais, voir Figure 4.1), qui est un effet numérique causé par l'orientation du champ de vitesse par rapport au maillage. Bien que l'empreinte de maillage soit généralement inévitable, son *anisotropie* peut être modulée et se prête donc à une réduction significative.

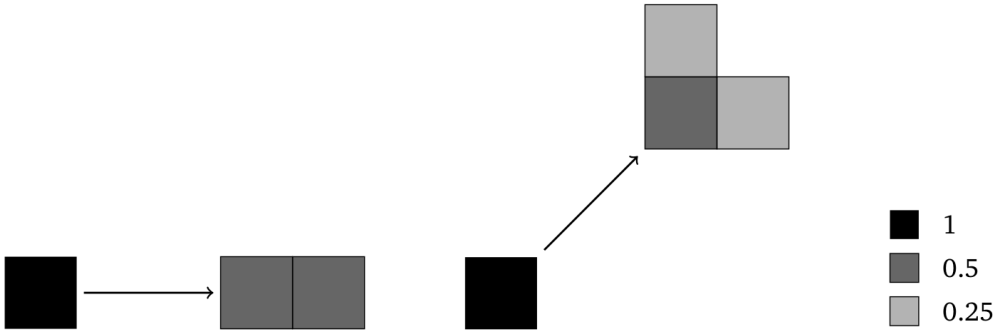


Figure 4.1: Illustration de l'empreinte du maillage appliqué localement à une maille: les formes de la solution numérique sont différentes selon la direction de la vitesse; gauche: vitesse en direction de l'axe x; droite: vitesse en direction diagonale.

Une nouvelle définition des stencils est introduite en prenant en compte les *deuxièmes voisins les plus proches* (à travers les coins des mailles) et la stratégie résultante est appelée "approche co-mesh". L'équation modifiée est utilisée pour étudier la dissipation numérique et régler les stencils agrandis afin de minimiser l'anisotropie de l'advection.

Les travaux décrits dans le chapitre 1 ont été simultanément étudiés dans le contexte d'écoulement multiphasique incompressible en milieu poreux par [58, 59]. Agissant sur un même voisinage dans un maillage cartésien avec des premier et second plus proches voisins, un stencil à neuf points est introduit pour les flux amont. Celui-ci est analysé à l'aide de l'analyse de Fourier et un critère est trouvé pour déduire un *comportement le moins anisotrope de l'erreur angulaire*, par opposition aux travaux sur le co-mesh, qui étudie l'équation modifiée pour dériver un critère afin de réduire l'anisotropie. Les deux stratégies conduisent à de nouveaux coefficients qui permettent d'ajouter les seconds plus proches voisins afin de trouver des stencils optimaux vis-à-vis du comportement anisotrope. [58, 59] ont été inspirés par [60] sur l'effet d'orientation du maillage (GOE pour *Grid Orientation Effect* en anglais). Cet effet a été introduit dans les années 1980 dans le cadre de la prospection pétrolière [61, 62], et a été récemment étudié dans des schémas de type Godunov [63, 64], ainsi que dans des études de convergence des maillages quadrilatéraux et triangulaires [65, 66, 67].

Le chapitre 1 montre des simulations sur l'advection d'ordre un sur des maillages cartésiens, leurs co-mesh et la combinaison linéaire optimale des deux. L'équation modifiée a été étudiée dans tous ces cas et des graphiques illustrant les coefficients diffusifs définis dans une certaine base (en fonction de la direction de la vitesse) sont présentés avec les résultats des simulations. De plus, les résultats de la méthode co-mesh sur un maillage arbitrairement déformé sont présentés. La stratégie co-mesh réussit à réduire l'empreinte de maillage dans tous ces cas.

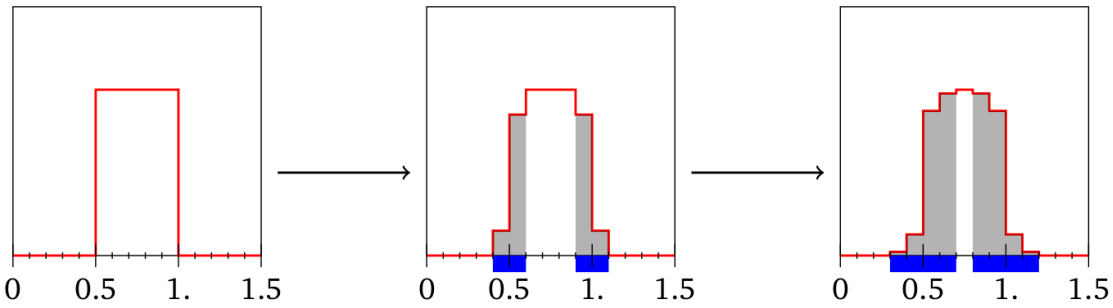


Figure 4.2: Illustration du “mouillage numérique”: augmentation du support de la diffusion numérique avec le pas de temps du schema (les mailles du support sont accentuées en bleu).

Pente-Borne 1D: une procédure de “clipping” pour les gradients en 1D

L’interpolation monotone et ses avatars sont des ingrédients majeurs de nombreux schémas numériques pour résoudre des équations aux dérivées partielles (EDP) sous contrainte TVD (*Total Variation Diminishing*) ou similaires. Cependant, malgré plus de quarante ans d’étude approfondie des principes et des applications, un aspect clé de la conception d’interpolateurs monotones peut encore paraître quelque peu empirique: comment un interpolateur monotone relie-t-il les cas limites de fonctions lisses (différentiables) et singulières (limitées) de manière canonique?

La présente étude vise à fournir une compréhension dans le cas de base de la reconstruction monotone scalaire en 1D, soit monotone par maille, et à l’appliquer à l’advection à l’ordre deux. Tout d’abord, une cartographie générale des fonctions monotones bornées en coordonnées elliptiques est construite. Ensuite, les interpolateurs classiques monotones à l’ordre deux “à une seule pente” (*single-slope* en anglais) sont transformé en interpolateurs monotones “pente-borne” (*Slope-And-Bound* SAB en anglais). Enfin, une contrainte critique est introduite, la “double monotonicité”, pour construire divers interpolateurs monotones de type pente-borne à partir de cet ensemble d’interpolateurs.

Avec ces interpolateurs pente-borne, les tests numériques standard montrent une annulation complète du “mouillage numérique” (ou *numerical wetting* en anglais, voir Figure 4.2) que produisent les schémas d’advection TVD habituels. Lors de l’étape d’advection d’un champ scalaire à support compact, cet effet - à ne pas confondre avec la diffusion numérique habituelle - est la contamination de bas niveau qui se propage linéairement dans le temps sur toutes les régions du domaine de calcul où l’advection “non-disparaissant” (*non-vanishing* en anglais) reste présente. Pour appuyer cette affirmation, le chapitre 2 compare les résultats

des limiteurs SAB et les compare à leurs fonctions de limiteur de pente de base sur une onde carrée, une onde sinusoïdale et un profil multi-onde appelé le cas test de Zalesak. La suppression du mouillage numérique est d'une importance particulière dans de nombreuses applications industrielles et académiques, notamment lors d'épisodes de "disparition de phase" dans les écoulements multiphasiques ou de transitions "humide-sec" (*wet-dry* en anglais) dans les écoulements en eau peu profonde, décrits par les équations de Barré de Saint-Venant. Une amélioration de "l'érosion numérique" des extrema est également observée. Les principes généraux exposés ici peuvent être étendus aux paramètres multidimensionnels, aux schémas d'ordre élevé et à d'autres EDP.

Pente-borne n-dimensionnelle

Dans le chapitre 2, les interpolateurs doublement monotones [68] et pente-borne (SAB pour *Slope-And-Bound*) sont explorés systématiquement afin de créer des limiteurs d'ordre deux très simples avec des flux d'advection 1D tout en atténuant la diffusion numérique. Une extension multidimensionnelle appelée ND-SAB (pour *N-Dimensional Slope-And-Bound*) est présentée ici comme alternative aux techniques MUSCL classiques: les gradients multidimensionnels ne sont pas directement limités, mais réduits à une représentation 1D, que l'on appelle le référentiel 1D (ou *1D reference frame*). Ce référentiel est utilisée pour la limitation et pour le calcul du flux total d'une maille. Ce flux total est ensuite redistribué sur les flux aux faces.

L'élément clé de la reconstruction 1D SAB est donné avec le *point de référence d'ordre deux*, qui est le point donné au centre de gravité de la maille c qui aligne les valeurs des centre de gravité de ses voisins de gauche et de droite. Dans les mailles uniformes, ce point de référence d'ordre deux correspond simplement à la moyenne des valeurs voisines, tandis que dans les maillages non uniformes, il est défini comme une moyenne pondérée, avec des poids donnés par les longueurs respectives des mailles voisines. Cependant, dans ce référentiel 1D du présent algorithme ND-SAB, les tailles des mailles voisines à gauche et à droite sont inconnues et la reconstruction 1D SAB ne peut pas être appliquée directement. Il est donc plus difficile de définir une extension totalement multidimensionnelle de la stratégie SAB. Dans ces travaux, ceci est résolu en introduisant des seuils de monotonicité dans ce référentiel 1D.

De plus, la méthode ND-SAB est compatible avec l'advection quasi-isotrope d'ordre un en utilisant la stratégie des co-mesh, introduite dans [69]. Les résultats sont présentés dans le chapitre 3 et rappelé ici dans les figures 4.3 à 4.6: trois simulations d'advection d'un cercle sont superposées dans chaque image. Les simulations sont définies par des vecteurs de vitesse constant $\boldsymbol{v}(\theta) = ||\boldsymbol{v}||(\cos(\theta), \sin(\theta))$, pour les trois directions choisies $\theta \in \{0, \pi/8, \pi/4\}$. De la même façon que pour les résultats obtenus dans le chapitre 2, le mouillage numérique est réduit à un

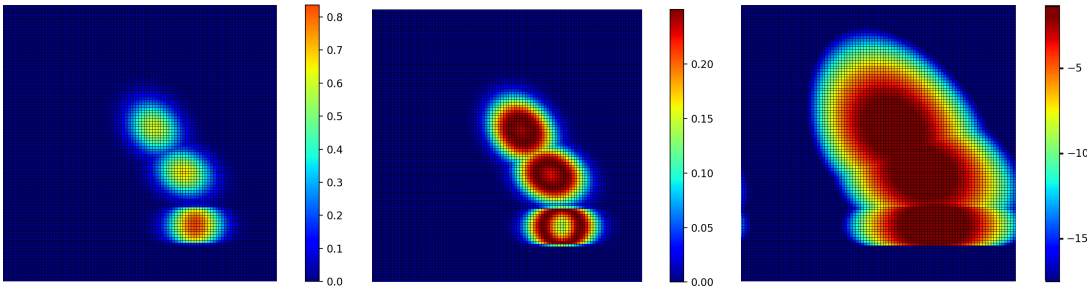


Figure 4.3: Simulations avec une advection d'ordre un : sur un maillage cartésien sur $[0,1]^2$ avec 100×100 mailles, CFL$= 0.5$ et temps final $T = 0.5$. Profils de densité ρ (gauche), $\rho(1-\rho)$ (milieux), et $\log(|\rho(1-\rho)|)$ (droite).

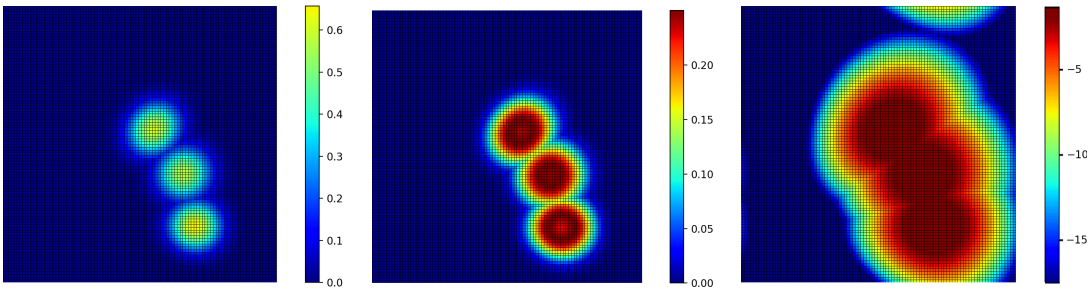


Figure 4.4: Méthode des co-mesh dans une advection d'ordre un : sur un maillage cartésien sur $[0,1]^2$ avec 100×100 mailles, CFL$= 0.5$ et temps final $T = 0.5$. Profils de densité ρ (gauche), $\rho(1-\rho)$ (milieux), et $\log(|\rho(1-\rho)|)$ (droite).

support compact, ce qui peut être observé avec les représentations logarithmiques des résultats dans ces figures 4.5 et 4.6 de droite.

Conclusion

Une nouvelle discrétisation pour l'opérateur d'advection amont explicite (4.1) du solveur GEEC a été introduite. Plus précisément le formalisme GEEC définit un schéma ALE direct pour un nombre arbitraire de fluides, et agit ainsi sur des maillages cartésiennes déformées arbitrairement. La formulation ALE directe de GEEC fournit une description lagrangienne à l'ordre deux, alors que la description eulérienne est définie dans l'opérateur d'advection (4.1), qui n'est qu'à l'ordre un. Ces travaux ajoutent deux techniques pour améliorer la formulation de (4.1): (i) la stratégie co-mesh [69], qui réduit l'empreinte du maillage; (ii) l'algorithme de reconstruction ND-SAB, qui définit une technique de reconstruction simple à l'ordre deux et représente une stratégie alternative aux méthodes classiques de "splitting"

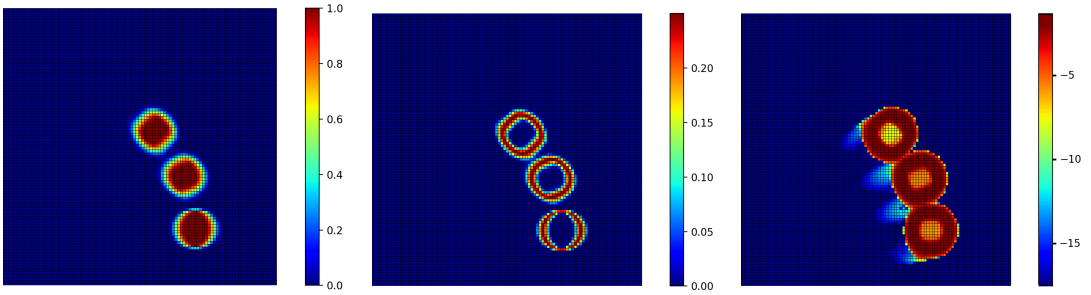


Figure 4.5: Simulations avec l'algorithme ND-SAB : sur un maillage cartésien sur $[0,1]^2$ avec 100×100 mailles, CFL$= 0.5$ et temps final $T = 0.5$. Profils de densité ρ (gauche), $\rho(1-\rho)$ (milieux), et $\log(|\rho(1-\rho)|)$ (droite).

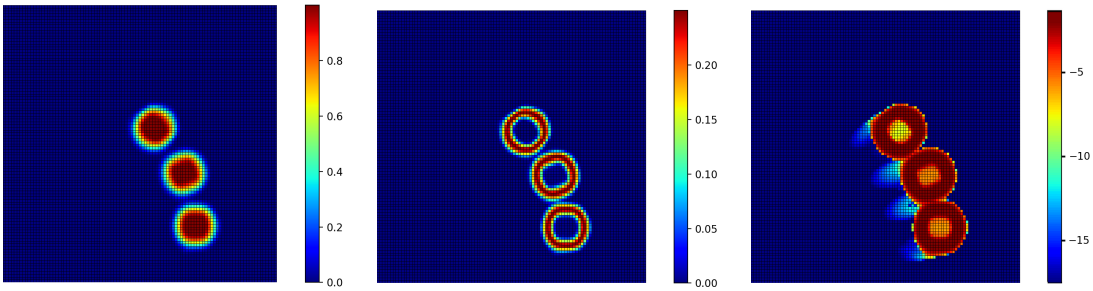


Figure 4.6: Simulations avec l'algorithme ND-SAB combiné avec la stratégie des co-mesh : sur un maillage cartésien sur $[0,1]^2$ avec 100×100 mailles, CFL$= 0.5$ et temps final $T = 0.5$. Profils de densité ρ (gauche), $\rho(1-\rho)$ (milieux), et $\log(|\rho(1-\rho)|)$ (droite).

directionnel. L'objectif principal de ND-SAB est de réduire le mouillage numérique dans l'étape d'advection d'un solveur hydrodynamique.

Le présent algorithme de reconstruction ND-SAB a été testé sur des maillages cartésiens uniformes et des champs de vitesse constante. Des extensions au cas général des maillages arbitraires et des vitesses non constantes sont en cours. La description actuelle de l'algorithme ND-SAB ne garantit pas la validité de la propriété du principe du maximum discret (DMP pour *Discrete Maximum Principle*, éventuellement local) dans le cas général. En effet, des "overshoots" ont été observé dans les expériences sur des maillages cartésiens arbitrairement déformés et avec des champs de vitesse non constant (à divergence nulle). Soulignons que l'opérateur d'advection introduit ici est décrit pour un formalisme eulérien, qui est entièrement compatible avec la description multi-fluide en représentation ALE-directe du solveur GEEC, et agit sur l'étape d'advection dans GEEC. Il n'est pas encore clair si les problèmes d'"overshoots" dans sa forme actuel de l'algorithme ND-SAB seront résolus dans GEEC par sa description ALE ou si des corrections supplémentaires de flux seront nécessaires. La principale différence entre travailler sur des maillages eulériens et la formulation ALE de l'opérateur d'advection amont (4.1), concerne le support du flux, qui dans ce cas, est donné par des quadrilatères arbitrairement déformés, en raison des vitesses aux nœuds. Dans une première approche, cela ne modifie que les calculs des centres de gravité sur lesquels les valeurs estimées sont calculées.

Comme discuté dans le chapitre 3, une stratégie qui corrige la propriété DMP sur les maillages eulériennes devrait être envisagée. En outre, il est plus susceptible de garantir un bon comportement dans GEEC, une fois que la propriété DMP est satisfaite. D'autant que le présent algorithme de reconstruction multidimensionnelle ND-SAB est flexible à des changements, tout en préservant la philosophie SAB sur: (i) le choix de la reconstruction de différents gradients; (ii) une adaptation des calculs de *bornes* dans le référentiel 1D (où les voisins amont sont agrégés à droite et les voisins aval à gauche); (iii) corriger les poids de redistribution de la quantité totale du flux qui est redistribuée aux flux aux faces (dans la dernière étape de l'algorithme).

En conclusion, la stratégie ND-SAB est un outil intéressant pour réduire le mouillage numérique dans les problèmes d'advection en préservant un support compacte, et il est naturellement compatible avec l'approche co-mesh, qui réduit l'empreinte de maillage. Le principal avantage de ces techniques est que les grandeurs physiques sont maintenues colocalisées et n'introduisent donc aucune complexité supplémentaire pour le solveur GEEC. Soulignons que le co-mesh et la stratégie de limitation ND-SAB ne sont pas limités au solveur GEEC, mais sont conçus pour une utilisation plus générale des opérateurs d'advection amont explicites.

Titre: Schémas numériques améliorés pour l'advection scalaire, monotone et conservative : réduction d'empreinte de maillage et de mouillage numérique

Mots clés: Advection, monotonicité, conservation, ordre deux, empreinte de maillage, mouillage numérique.

Résumé: Le CEA met en œuvre des méthodes numériques pour ses calculs hydrodynamique sous des contraintes particulièrement sévères : advections et déformations sur de grandes échelles, évolutions isentropiques et sous chocs forts, équations d'état complexes et mélanges de matériaux, couplages à d'autres physiques multiples et complexes, maillages importants requérant des ordinateurs massivement parallèles, etc.

Dans ce contexte le schéma GEEC (pour Geometry, Energy, and Entropy Compatible) a été développé dans de récents travaux au CEA. Ce schéma est d'ordre deux en temps et en espace, hormis l'advection relative à la grille qui n'est que d'ordre un. Le travail présenté ajoute deux techniques mutuellement compatibles afin d'améliorer la formulation de l'opérateur de transport avec son objectif principal qui consiste à réduire l'effet d'"empreinte de maillage" ainsi que le "mouillage numérique": (i) la stratégie de "co-mesh" corrige le comportement anisotrope de l'advection empreint sur les solutions numériques et causé par l'orientation du champ de vitesse relativement au maillage, phénomène aussi connu sous le nom d'"effet d'orientation du maillage"; (ii) l'algorithme de reconstruction ND-SAB (pour n-dimensional Slope-And-Bound ou encore en français pente-borne n-dimensionnelle) définit une technique de reconstruction à l'ordre deux directe, et peut être vu comme une alternative à des stratégies classiques de type "splitting directionnel". La technique ND-SAB est inspirée de la stratégie de limitation SAB en 1D, qui introduit une méthode de "clipping" pour la reconstruction de pentes avec le but de supprimer le mouillage numérique.

Les deux techniques sont validées individuellement et combinées sur des maillages cartésiens avec des simulations à des vitesses constantes. Une prochaine étape sera d'intégrer dans GEEC cette nouvelle forme de l'opérateur de transport d'ordre deux, sachant qu'elle y est déjà entièrement compatible dans sa formulation multi-fluide et ALE-directe.

Title: Improved numerical schemes for monotonic conservative scalar advection: tackling mesh imprinting and numerical wetting

Keywords: Advection, monotonicity, conservation, second order, mesh imprinting, numerical wetting.

Abstract: The French atomic energy commission (CEA, France) deploys specific numerical methods to carry out its hydrodynamics simulations under especially severe constraints : advection and deformation over long distances, isentropic evolutions and strong shocks, complex equations of state and material mixtures, coupling with other various and complex physics, large meshes requiring massively parallel computers, etc.

In this context the numerical scheme GEEC (Geometry, Energy, and Entropy Compatible) was recently developed at CEA. GEEC is generally second-order in space and time, except for the advection operator relative to the mesh which is only first-order. The present work adds two techniques compatible with each other to improve the formulation of the advection operator with the main objective of reducing "mesh-imprinting" and "numerical wetting": (i) the "co-mesh" strategy corrects the anisotropic behavior of advection imprinted on the numerical solution and caused by the orientation of the grid with respect to the velocity field, which is also known as "grid orientation effect"; (ii) the n-dimensional Slope-And-Bound (ND-SAB) reconstruction algorithm defines a straight forward second-order reconstruction technique, and can be seen as an alternative strategy to classical directional splitting methods. The ND-SAB technique is inspired by the 1D SAB limitation strategy, which introduces a "clipping" method for slope reconstructions with the goal of suppressing numerical wetting.

Both techniques have been validated individually and combined on uniform Cartesian grids with simulations for constant velocities. Future works will include testing this novel form of a second-order advection operator fully compatible with GEEC in its multi-fluid direct-ALE formalism.