



UNIVERSIDAD DE MÁLAGA

PROGRAMA DE DOCTORADO EN MATEMÁTICAS

FACULTAD DE CIENCIAS

DEPARTAMENTO DE ANÁLISIS MATEMÁTICO, ESTADÍSTICA
E INVESTIGACIÓN OPERATIVA Y MATEMÁTICA APLICADA

Multidimensional approximate Riemann solvers for hyperbolic systems

(Resolvedores de Riemann multidimensionales aproximados para sistemas hiperbólicos)

KLEITON ANDRÉ SCHNEIDER

TESIS DOCTORAL

DIRECTOR:

JOSÉ MARÍA GALLARDO MOLINA

UNIVERSIDAD DE MÁLAGA 2020


UNIVERSIDAD
DE MÁLAGA



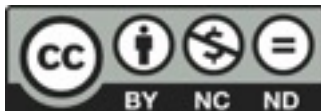


UNIVERSIDAD
DE MÁLAGA

AUTOR: Kleiton André Schneider

 <http://orcid.org/0000-0002-6892-8644>

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional:

<http://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>

Cualquier parte de esta obra se puede reproducir sin autorización pero con el reconocimiento y atribución de los autores.

No se puede hacer uso comercial de la obra y no se puede alterar, transformar o hacer obras derivadas.

Esta Tesis Doctoral está depositada en el Repositorio Institucional de la Universidad de Málaga (RIUMA): riuma.uma.es





DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

D. KLEITON ANDRÉ SCHNEIDER

Estudiante del programa de doctorado Matemáticas de la Universidad de Málaga, autor de la tesis, presentada para la obtención del título de doctor por la Universidad de Málaga, titulada:

Multidimensional approximate Riemann solvers for hyperbolic systems

Realizada bajo la tutorización de Carlos Parés Madroñal y la dirección de José María Gallardo Molina

DECLARO QUE:

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo.

Igualmente asumo, ante a la Universidad de Málaga y ante cualquier otra instancia, la responsabilidad que pudiera derivarse en caso de plagio de contenidos en la tesis presentada, conforme al ordenamiento jurídico vigente.

En Málaga, a 29 de abril de 2020

Fdo.: Kleiton André Schneider



UNIVERSIDAD
DE MÁLAGA

D. José María Gallardo Molina, Profesor Titular del Departamento de Análisis Matemático, Estadística e Investigación Operativa, y Matemática Aplicada de la Universidad de Málaga

CERTIFICA:

Que D. Kleiton André Schneider ha realizado en el Departamento de Análisis Matemático, Estadística e Investigación Operativa, y Matemática Aplicada de la Universidad de Málaga, bajo mi dirección, el trabajo de investigación correspondiente a su Tesis Doctoral, titulado:

**Multidimensional approximate Riemann solvers
for hyperbolic systems**

Revisado el presente trabajo, estimo que puede ser presentado al Tribunal que ha de juzgarlo. Y para que conste a efectos de lo establecido en el artículo octavo del Real Decreto 99/2011, autorizo la presentación de este trabajo en la Universidad de Málaga.

En Málaga, a 29 de abril de 2020



Dr. José María Gallardo Molina



UNIVERSIDAD
DE MÁLAGA

Agradecimientos

Es mi deseo expresar mi más sincera gratitud a todas las personas que de una u otra forma han contribuido, personal y profesionalmente, a la realización de esta memoria.

Mis más sinceros agradecimientos y profundo respeto a mi supervisor, Prof. José María Gallardo Molina, por el soporte ilimitado y continuo, por la paciencia y apoyo durante todo el doctorado. Muchas gracias por los consejos, el buen ambiente, la humildad y todo el conocimiento compartido.

Me gustaría también agradecer a Carlos Parés y Manuel J. Castro por el tiempo, paciencia y apoyo en todo el doctorado, los cuales también contribuyeron a los resultados obtenidos en esta tesis doctoral y a mi formación. Igualmente agradecer a todos los miembros del grupo EDANYA con los que he trabajado y compartido buenos momentos: Jorge Macías, José Manuel González Vida, Mari Luz Muñoz, Sergio Ortega, Tomás Morales de Luna, Carlos Sánchez Linares y Marc de la Asunción. Todos han contribuido a generar el mejor ambiente de trabajo. De igual manera, agradezco también a todos mis colegas, con los cuales he tenido el placer de convivir, aprender y pasar momentos especiales: Hugo Carrillo, Juan Carlos González, Cipriano Escalante, Ernesto Guerrero, Ernesto Pimentel, Irene Gómez Bueno, Federico Rodas y Emanuelle Macca.

Mis agradecimientos a Dinshaw Balsara, profesor de la Universidad de Notre Dame, Estados Unidos, con el cual realicé una estancia de investigación, y a Boniface Nkonga, profesor de la Universidad de Nice, Francia, los cuales contribuyeron a los resultados obtenidos en esta tesis doctoral y a mi formación.

Un agradecimiento especial a mis colegas de la Universidad Federal de Mato Grosso do Sul (UFMS), Brasil, por el apoyo, soporte y por la amistad. Agradezco también a dicha Universidad por el soporte financiero durante todo el periodo del doctorado.

Todo mi agradecimiento a mi familia, tanto para aquellos que estuvieron de lejos enviándome las mejores energías y vibraciones, en especial a mi mamá Gerta, como a aquellos que estuvieron cerca dándome soporte y motivación, vivenciando todos los altibajos de mi investigación, Patricia y mi amado hijo Arthur.

Gracias a todos.

Agradecimentos

Desejo expressar minha mais sincera gratidão a todas aquelas pessoas que de uma forma ou outra contribuíram, pessoal e profissionalmente, na realização desta tese de doutorado.

Meus mais sinceros agradecimentos e profundo respeito ao meu supervisor, Prof. José Maria Gallardo Molina, pelo suporte ilimitado e contínuo, pela paciência e apoio durante todo o doutorado. Muito obrigado pelos conselhos, o bom ambiente, a humildade e todo o conhecimento compartilhado.

Gostaria também de agradecer aos professores Carlos Parés e Manuel J. Castro, pelo tempo, paciência e apoio em todo o doutorado, os quais também contribuíram nos resultados obtidos nesta tese de doutorado e em minha formação. Igualmente agradecer a todos os membros do grupo EDANYA com os quais trabalhei e passei bons momentos: Jorge Macías, José Manuel González Vida, Mari Luz Muñoz, Sergio Ortega, Tomás Morales de Luna, Carlos Sánchez Linares e Marc de la Asunción. Todos contribuíram de forma a gerar o melhor ambiente de trabalho. Da mesma maneira, agradeço também a todos os meus colegas com os quais tive o prazer de conviver, aprender e passar momentos especiais: Hugo Carrillo, Juan Carlos González, Cipriano Escalante, Ernesto Guerrero, Ernesto Pimentel, Irene Gómez Bueno, Federico Rodas e Emanuelle Macca.

Meus agradecimentos a Dinshaw Balsara, professor da Universidade de Notre Dame, Estados Unidos, com o qual realizei uma estância de investigação, e Boniface Nkonga, professor da Universidade de Nice, França, os quais contribuíram nos resultados obtidos nesta tese de doutorado e em minha formação.

Um agradecimento especial aos meus colegas da Universidade Federal de Mato Grosso do Sul (UFMS), Brasil, pelo apoio, suporte e pela amizade. Agradeço também à UFMS pelo suporte financeiro durante todo o período do doutorado.

Todo o meu agradecimento à minha família, tanto para aqueles que estiveram de longe me enviando as melhores energias e vibrações, em especial à minha mãe Gerta, quanto aqueles que estiveram perto me dando suporte e motivação, vivenciando todos os altos e baixos da minha pesquisa, Patricia e meu filho amado Arthur.

Obrigado à todos.

Contents

List of figures	v
Introduction	1
Introduction (in Spanish)	8
1 Hyperbolic systems and finite volume schemes	14
1.1 Conservation laws	15
1.2 Finite volume approximation	18
1.3 Nonconservative systems	22
1.4 Path-conservative schemes	26
1.5 AVM-type solvers	30
1.6 High-order methods: general framework	36
1.7 MUSCL-Hancock procedure	40
2 Governing equations	41
2.1 Magnetohydrodynamic equations	42
2.2 Shallow water equations: one-layer approximation	47
2.3 Shallow water equations: two-layer approximation	48
2.4 Well-balancing	49
3 Multidimensional AVM-type solvers: the conservative case	52
3.1 Preliminaries and four-waves model	54
3.2 2D AVM-type solvers for conservation laws	60
3.3 Numerical results	65
3.3.1 First-order accuracy	66
3.3.2 Orszag-Tang vortex	70
3.3.3 The rotor problem	74
3.3.4 Two-dimensional Riemann problem	75
3.3.5 Spherical explosion	77
4 Multidimensional AVM-type solvers: the nonconservative case	80
4.1 General framework	81
4.2 Well-Balanced 2D HLL Riemann solver	87
4.2.1 Supersonic cases	92
4.3 Well-Balanced 2D AVM-type Riemann solvers	94
4.3.1 HLL 2D solver in AVM form	94
4.3.2 Multidimensional AVM solvers	97
4.3.3 Modified equation and linear stability	98
4.3.4 Second-order schemes	100
4.4 Numerical results	102
4.4.1 Second-order accuracy	102



4.4.2	<i>C</i> -property	103
4.4.3	Applications to the one-layer shallow water system	105
4.4.3.1	Circular dam-break	105
4.4.3.2	Non-linear breaking waves	108
4.4.4	Applications to the two-layer shallow water system	110
4.4.4.1	Internal circular dam-break	110
4.4.4.2	Evolution to a complex state	112
5	Efficient GPU implementation	116
5.1	Rewriting of the numerical schemes	117
5.1.1	Improving efficiency of AVM schemes	120
5.2	Parallelism sources	122
5.2.1	Vertex-based calculations	123
5.2.2	Edge-based calculations	126
5.2.3	Volume-based calculations	127
5.3	CUDA implementation	129
5.4	Experimental results	131
6	Conclusions and future work	135
6.1	Conclusions	135
6.2	Future work	137
A	PVM-Chebyshev and RVM-Newman coefficients	138
B	Full algorithm for the 2D HLL Riemann solver	140
C	Full algorithm for 2D AVM-type formulation	151
D	An efficient implementation of well-balanced PVM schemes	156
	Bibliography	158

List of Figures

A	Left: Sketch of the 2D Riemann solver based on projected 1D Riemann problems through the edges. Right: Sketch of the truly 2D approach considering also the vertex contributions from the 2D Riemann problems	4
1.1	Illustration of initial data for the Riemann problems (a) 1D Riemann problem: two constant states separated by a discontinuity at $x = 0$. (b) 2D Riemann problem: four constant states, with discontinuities along the edges of any two states and in the vertex $(x, y) = (0, 0)$	16
1.2	(a) Sketch of a SRS with s waves. (b) Skecth of HLL Riemann solver.	21
1.3	Top: Comparison of Chebyshev polynomials $\tau_8(x)$ and $\tau_8^\varepsilon(x)$; Chebyshev polynomials $\tau_{2p}^\varepsilon(x)$ for $p = 2, 3, 4$. Bottom: Comparison of Newman rational functions $R_4(x)$ and $R_4^\varepsilon(x)$; zoom near the origin of the Newman rational functions $R_r^\varepsilon(x)$ for $r = 4, 8, 12$	33
1.4	Top: Internal polynomials $p_n(x)$ for $n = 1, 2, 3, 4$, and Padé rational approximants $r_1^{[m/k]}(x)$ for $m = k = 1, 2, 3, 4$. Bottom: Padé approximants $r_n^{[1/1]}(x)$ for $n = 1, 2, 3, 4$, and zoom near the origin.	34
3.1	Stencil used to build the numerical flux $F_{i+1/2,j}$	55
3.2	Left: Local stencil. Right: Structure of the solution of the Riemann problem.	55
3.3	Local view (left) and global view (right) of the four-waves model.	56
3.4	Structure of the solution of the 2D Riemann problem at a vertex.	57
3.5	Assembling of one- and two-dimensional fluxes at an edge.	59
3.6	Test 3.3.1. Comparison of several 1D×1D-2D AVM methods on a 100×100 mesh. Diagonal cut along the main diagonal for the density variable.	66
3.7	Test 3.3.1. Solutions obtained with the Int- n 2D schemes, for $n = 1, 2, 3, 4, 5$. Diagonal cut along the main diagonal for the density variable.	67
3.8	Test 3.3.1. Diagonal cut along the main diagonal for the density variable on several meshes. Left: Int-3 2D. Right: Padé-[4/4] 2D.	67
3.9	Test 3.3.1. Error curves for several 1D×1D and 2D AVM methods.	68
3.10	Test 3.3.1 for MHD. Diagonal cut along the main diagonal for the B_\perp variable on several meshes. Left: Int-3 2D. Right: Padé-[4/4] 2D.	69
3.11	Test 3.3.2. Density (left) and pressure (right) computed at time $T = \pi$ on a 200×200 mesh. From top to bottom: HLL 2D, Int-3 2D and Padé-[4/4] 2D.	71



3.12	Test 3.3.2. Cuts along the main diagonal of the density at times $t = 1$, $t = 1.5$, $t = 2$ and $t = \pi$.	72
3.13	Test 3.3.2. Comparison of the divergence cleaning technique of Section 2.1 with the projection method using the Int-3 2D scheme. Cut along the main diagonal: density (left) and pressure (right) computed at time $T = \pi$ on a 100×100 mesh.	73
3.14	Test 3.3.3. Density ρ (top left), pressure P (top right), Mach number $ \mathbf{v} /a$ (bottom left) and magnetic pressure $ \mathbf{B} ^2/2$ (bottom right) computed at time $T = 0.295$. Results obtained with the Padé-[4/4] 2D scheme with 200×200 cells.	75
3.15	Test 3.3.4. Contours at time $T = 0.2$ obtained with the Padé-[4/4] 2D scheme on a 200×200 mesh. Left: B_x . Right: B_y .	76
3.16	Test 3.3.4. Cuts at $x = 0.93$ and $T = 0.2$. Solid line: reference solution. Dots: Padé-[4/4] 2D. Left: B_x . Right: v_x .	76
3.17	Test 3.3.5. Solution obtained with initial magnetic field $(B_x, B_y, B_z) = (0, 0, 0)$. Density ρ (top left), pressure P (top right), kinetic energy (bottom left) and density along $y = 0.5$ (bottom right) computed at time $T = 3$.	77
3.18	Test 3.3.5. Solution obtained with initial magnetic field $(B_x, B_y, B_z) = (0, 5/\sqrt{\pi}, 0)$. Density ρ (top left), pressure P (top right), magnetic pressure (bottom left) and kinetic energy (bottom right) computed at time $T = 3$.	78
3.19	Test 3.3.5. Solution obtained with initial magnetic field $(B_x, B_y, B_z) = (0, 50/\sqrt{\pi}, 0)$. Density ρ (top left), pressure P (top right), magnetic pressure (bottom left) and kinetic energy (bottom right) computed at time $T = 1.05$.	79
4.1	Structure of the solution of the 2D Riemann problem for the nonconservative system (5.1.1). Subsonic case.	82
4.2	Contributions in the numerical scheme (4.1.9).	85
4.3	Structure of the solution of the 2D Riemann problem for system (4.2.1). Subsonic case.	88
4.4	Structure of the solution of the 2D Riemann problem for system (4.2.1). Supersonic case in the y -direction.	92
4.5	Diagonal advection. Left: 1D \times 1D scheme. Center: 2D scheme. Right: exact solution.	99
4.6	Test 4.4.2. Cut along the main diagonal of the steady state solution.	105
4.7	Test 4.4.3.1. Diagonal cuts of the free surface at different times, computed with the 2D HLL Riemann solver on a 200×200 mesh.	106
4.8	Test 4.4.3.1. Contour plots of the free surface at different times, obtained with the 2D HLL Riemann solver on a 200×200 mesh.	107
4.9	Test 4.4.3.1. Cuts of the free surface along the x -axis and in the main diagonal direction, using an 80×80 coarse mesh (left top: HLL 1D \times 1D; right top: HLL 2D), and a 400×400 refined mesh (left down: HLL 1D \times 1D; right down: HLL 2D).	108
4.10	Test 4.4.3.2. Cuts of the free surface along the main diagonal at different times. Comparison between the 1D \times 1D and the 2D HLL Riemann solvers.	109
4.11	Test 4.4.3.2 Left: Potential vorticity contour plot at time $T = 2$. Right: L^1 and L^∞ norms until time $T = 25$.	110

4.12	Test 4.4.4.1. Cuts along the main diagonal at different times, computed on a 200×200 mesh grid.	111
4.13	Test 4.4.4.2. Cuts of the interface along the main diagonal at different times.	112
4.14	Test 4.4.4.2. Cuts of the interface along the x -axis at different times.	113
4.15	Test 4.4.4.2. Contour plots of the interface at times $t = 4, 5, 6, 7$, from top to bottom. Left: 2D HLL; center: 2D PVM Chebyshev; right: 2D RVM Newman.	115
5.1	(a) Global notation for the scheme (5.1.2)-(5.1.3). The twelve contributions to the cell C_{ij} are represented. (b) Local notation for a vertex.	118
5.2	Sketch of an arrangement of a 2D Grid of 7×7 blocks of 8×8 threads to a 50×50 meshgrid. The starting thread should be a multiple of the warp size. Figure adapted from [143]	122
5.3	(a) First-order CUDA solver. (b) Second-order CUDA solver.	124
5.4	Different vertex processing steps.	125
5.5	Storing in the vertex kernel and adding up in the volume kernel the vertex contributions to the accumulator in the control volume C_{ij}	126
5.6	Efficiency curves GPU time vs. L^1 -error. Comparison of the 2D AVM schemes with respect to their 1D \times 1D counterparts. Case $k = 1$	132
5.7	Efficiency curves GPU time vs. L^1 -error. Comparison of the 2D AVM schemes with respect to their 1D \times 1D counterparts. Case $k = 2$	133
B.1	Sketch of the jumps through the zero wave for the case $S_U < 0, S_L < 0 < S_R$	143
B.2	Sketch of the jumps through the zero wave for the case $S_L > 0, S_D < 0 < S_U$	143
B.3	Sketch of the jumps through the zero wave for the case $S_L > 0, S_D > 0$	144
B.4	Sketch of the jumps through the zero wave for the case $S_L > 0, S_U < 0$	145
B.5	Sketch of the jumps through the zero wave for the case $S_R < 0, S_D < 0 < S_U$	146
B.6	Sketch of the jumps through the zero wave for the case $S_R < 0, S_D > 0$	147
B.7	Sketch of the jumps through the zero wave for the case $S_R < 0, S_U < 0$	148



UNIVERSIDAD
DE MÁLAGA

Introduction

Nowadays there is an increasing need of powerful methods to simulate Physics and Engineering processes in practical situations. In particular, an important requirement is that the numerical methods have to be robust and computationally efficient. Usually, mathematical models focus on physical principles such as conservation laws in order to provide realistic approximations of the processes observed in nature. The Finite Volume (FV) approach has been extensively used in the framework of conservation laws, as it ensures the conservation of physically relevant quantities appearing in the equations (see, e.g., [126, 127, 131, 136, 190, 198, 202]). Among the advantages of this approach are: the flexibility to handle complex geometries (see, for instance, [34, 71, 81, 82, 133]), that often appear in practical applications; the possibility to perform local refinements without affecting the rest of the computational mesh; and the treatment of an increasing level of physical complexity, for instance when simulating turbulent flows.

However, many models cannot be written in conservation form, and the appearance of nonconservative terms increases the difficulty of designing accurate numerical schemes. A way to solve this difficulty in the framework of finite volume schemes was proposed by Parés ([160]), who introduced the so-called *path-conservative* schemes. Since then, this point of view has been followed by several practitioners: see, e.g., [52, 38, 56, 48, 43, 59, 41, 47, 45, 49, 50, 51, 40, 55, 57, 62, 77, 79, 78, 80, 85, 84, 86, 75, 83, 102, 101, 117, 149, 152, 178]. Besides the development of new schemes, the use of path-conservative schemes allowed to extend a number of well-known methods in the literature to the nonconservative case: Lax-Friedrichs ([125]), Rusanov ([171]), Lax-Wendroff ([129]), HLL ([115]), FORCE ([191]), MUSTA ([189, 193]), etc. Among them, one of the most important is Roe's method, whose definition is based on the concept of Roe linearization introduced in [197]. Nowadays, nonconservative systems and path-conservative schemes constitute a very active front of research with a wide range of applications: see, for instance, [11, 46, 55, 92, 91, 90, 96, 142, 174, 175].

In his groundbreaking work [108], Godunov proposed the use of the exact solution of Riemann problems in the numerical fluxes. However, the exact solution of a nonlinear Riemann problem may be difficult to compute, which in general makes a hard task to design numerical methods following Godunov's idea. In this sense, Roe ([168]) proposed a Godunov-type scheme based on locally approximate solutions of the Riemann problem at each interface, changing the original Riemann problem by a linearized one. The drawback in this case is that the construction of a Roe linearization needs the knowledge of the complete eigenstructure of the system. For many hyperbolic systems of practical interest, such as the multilayer shallow water equations, the spectral decomposition is unknown. Therefore, it is interesting to consider *incomplete* Riemann

solvers, in which only part of the spectral information of the system under study is used. In this direction, the Harten-Lax-van Leer (HLL) scheme ([115]) approximated the Riemann wave fan using only the maximal wave speeds of propagation, at the cost of increasing the numerical dissipation. Some variants as HLLC ([192, 24]) and HLLEM ([88, 89]) have less dissipation than the original HLL scheme, but they need to add some extra spectral information.

In [41], the authors introduced a new class of path-conservative incomplete Riemann solvers, the so-called PVM (Polynomial Viscosity Matrix) methods, in which the numerical viscosity matrix is computed as a polynomial evaluation of a Roe matrix, if available, or the Jacobian matrix of the flux at some average state, and only needs a bound of the maximal speed of propagation. Many classical schemes as Rusanov, HLL, FORCE, Roe, etc. can be seen as PVM methods: see [41] and the references therein. A natural extension using rational functions gave rise to the class of RVM (Rational Viscosity Matrix) methods in [49]; in what follows, we will use the term AVM (Approximate Viscosity Matrix) to refer to both PVM and RVM solvers. As it was pointed out in [104], an appropriate choice of the underlying functions for an AVM solver allows to control the amount of numerical diffusion of the resulting scheme, what makes these methods particularly efficient when confronted with complex problems. Other extensions of AVM solvers were proposed in [51], where Jacobian-free versions of some PVM solvers were considered, and [50], where AVM approximations to the classical Osher-Solomon method were proposed (see also [85]).

When solving multidimensional problems, it is a common practice to consider one-dimensional projected Riemann solvers on the edges of the cells of the computational domain. On the other hand, two-dimensional meshes consist of a collection of edges and vertices, so naturally the Riemann solver applied in this mesh should involve locally one-dimensional Riemann problems at the edges and two-dimensional Riemann problems at the vertices (see Figure A). Accordingly, many researchers consider that within the projected approach, one-dimensional solvers lose much of their efficiency, mainly due to the fact that they do not take into account features of the solution propagating transversally to the cells boundaries. Moreover, the directional biasing introduced by one-dimensional solvers produces a reduction of the permissible Courant number when simulating multidimensional flows. For these reasons, there has been many attempts in the literature to build genuinely multidimensional Riemann solvers (see [1, 2, 3, 32, 97, 98, 105, 134, 140, 199, 206] and the references therein; see also [169] for a more detailed account on multidimensional upwinding), although most of them are specifically designed for the Euler or magnetohydrodynamics (MHD) equations. More general multidimensional Riemann solvers have been proposed more recently, for instance, in [14, 15, 19, 16, 18, 21, 74, 104, 103, 178, 203].

In [206] Wendroff proposed a two-dimensional Riemann solver for the Euler equations which extended the one-dimensional HLL method, in which the two-dimensional interactions at cell corners were taken into account through the approximate solutions of two-dimensional Riemann problems. However, a drawback of Wendroff's solver is that it does not possess explicit expressions allowing its direct implementation; also, high-order extensions of the solver do not seem straightforward to build. Some years later, Balsara ([14]) modified Wendroff's formulation and proposed a two-dimensional HLL solver for the Euler and MHD equations on structured meshes, which included closed forms of the fluxes and allowed for an easier high-order extension. A more robust version of Balsara's solver was later proposed in [15], and an extension to

unstructured meshes was presented in [19]. Another interesting reformulation of Wendroff's approach has been recently proposed in [203]. In recent years, a new class of multidimensional Riemann solvers with self-similar internal structure has been recently introduced in [16, 18, 21].

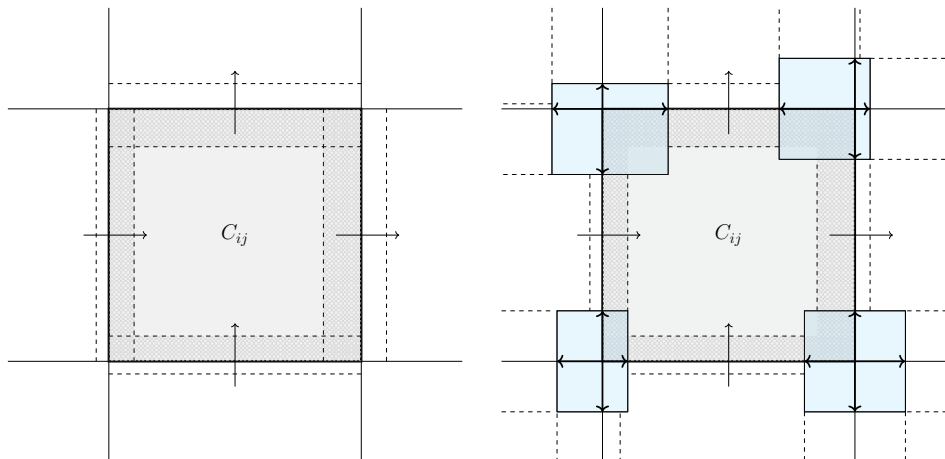


Figure A: Left: Sketch of the 2D Riemann solver based on projected 1D Riemann problems through the edges. Right: Sketch of the truly 2D approach considering also the vertex contributions from the 2D Riemann problems

Following the spirit of the previous works, the first part of this thesis focuses on a new class of genuinely two-dimensional incomplete Riemann solvers based on AVM-type solvers for general conservation laws introduced in [104]. These solvers can be viewed as a natural extension of Balsara's HLL method in [14], where the flux at edges is written as a suitable linear convex combination of one- and two-dimensional AVM fluxes. The two-dimensional contributions stem from the approximate solution of a two-dimensional Riemann problem at each vertex. The main idea is to reinterpret Balsara's two-dimensional HLL scheme as a two-dimensional PVM scheme in a suitable splitting way, making explicit the viscosity matrices in the coordinate directions. In other words, the two-dimensional HLL numerical fluxes are rewritten as a combination of one-dimensional PVM-HLL numerical fluxes. Once the one-dimensional HLL viscosity matrices have been identified, it is possible to build more precise two-dimensional AVM schemes.

The proposed solvers are applicable to general conservation laws, although in this thesis applications to MHD have been considered for the conservative case. There are several reasons for this choice. The first one is that the original work of Balsara ([14]) dealt with the MHD equations; thus, it seemed natural to analyze if our schemes also worked well within this setting. The second reason is related to the AVM philosophy: we would like to show that this kind of approach works well with very complex systems (as it was previously done with the one-dimensional PVM and RVM schemes in [41, 49, 50]). Finally, another reason was to introduce a novel divergence cleaning technique, based on the nonconservative writing of the MHD equations. As it is well-known, a difficulty that appears in the numerical resolution of MHD is the divergence-free constraint on the magnetic field, that has to be imposed in order to ensure the accuracy and stability of the numerical schemes. Several divergence cleaning

techniques have been proposed in the literature: see, for instance, [6, 22, 20, 30, 68, 69, 76, 93, 100, 99, 146, 147, 148, 163, 164, 194, 196, 211, 215]. The technique proposed in this work consists in using the MHD system augmented by the Godunov-Powell source terms within the path-conservative framework, with a subtle modification of the numerical fluxes accordingly to [149]. Numerical experiments show that our technique gives comparable results as the standard projection method ([30]) with regard to precision, but with a smaller computational cost.

The second part of this thesis focuses on the construction of a second-order extension of the genuinely two-dimensional AVM-type Riemann solvers to the case of general nonconservative hyperbolic systems. The proposed schemes are applied to the one-layer and two-layer shallow water systems including depth variations, in which the source term due to the bottom topography introduces an additional difficulty. An elegant way to overcome this difficulty consists in reformulating the problem in nonconservative form, within the framework of path-conservative schemes. A general strategy for constructing genuinely multidimensional Riemann solvers within this framework is presented. First of all, a simple two-dimensional nonconservative HLL scheme is considered, which takes into account the rectangular four-waves model proposed in [14], where only the maximal wave speeds in each direction are considered. The two-dimensional features at the vertices of the computational mesh are taken into account through the approximate solution of two-dimensional problems having as initial states the states at the four neighboring cells. These solutions roughly consist of a strong interaction region surrounded by intermediate states arising at the one-dimensional problems at edges. In the considered model, it is assumed that the strong interaction region do not interact directly with the four initial states, but only with the intermediate states. This simplification allows us to rewrite the two-dimensional HLL scheme in the PVM formulation, and thus it can be readily extended to a general two-dimensional AVM-type scheme. The differences of fluxes in the conservative formulation now are changed by fluctuations along the paths considered in the definition of the nonconservative products.

Many real-world problems demand a quick response to help in the decision making process, as it is the case of natural hazards (see, for instance, oceanographic applications in [87]). Modern *Graphics Processor Units* (GPUs) deliver hundreds of processors performing arithmetic operations in parallel at an affordable cost. The GPU designing company NVIDIA has made available the CUDA programming toolkit ([156]), along with a support to the C programming language, in order that the programmer does not need to deal with low level language complexities. Many CUDA solvers have been presented in the literature: see, for example, [7, 9, 10, 11, 8, 23, 33, 54, 92, 101, 124, 142, 143, 159, 174, 173, 208, 209]. Our two-dimensional numerical schemes present a high potential for data parallelization, which makes them well-suited to be implemented in a parallel code. We present here an efficient GPU implementation of the proposed two-dimensional schemes, which can be applied to general nonconservative hyperbolic systems.

Outline of the thesis

The outline of this dissertation is the following:

- In Chapter 1, the theoretical background required for the designing of the proposed two-dimensional solvers, for both the conservative and the nonconservative cases, is stated.

Concerning conservation laws and finite volume approximations, the main theoretical results are given. Nonconservative systems and path-conservative schemes are presented in a self-contained way. In particular, for the sake of completeness, a brief survey of AVM-type solvers is provided, with special attention to both PVM-solvers based on Chebyshev and internal polynomial approximations, and RVM-solvers based on Newman and Padé rational functions. These are the AVM-solvers that have been used in the numerical simulations presented in this thesis. Finally, a general framework to build high-order extensions for path-conservative schemes is also provided.

- In Chapter 2, first an overview about MHD equations is given. Furthermore, a novel technique to impose the divergence-free constraint is presented. Our method is based on the nonconservative formulation of the MHD equations, where the divergence of the magnetic field appears as a source term. This technique has been included into the two-dimensional AVM solvers by means of the path-conservative framework with a slightly modification on the numerical fluxes.

A brief review about the eigenstructure of the shallow water systems with bottom topography is also presented. The presence of source terms introduces a numerical difficulty related to the computation of steady or near steady solutions. The main results over *well-balanced* schemes are explained, including the modified identity technique.

- In Chapter 3, the four-waves model considered by Balsara in [14] is reviewed, including the detailed building of the two-dimensional HLL scheme for conservation laws. At each edge of the control volume, the numerical fluxes are assembled as a suitable linear convex combination of a one-dimensional contribution at the edge and two 2D contributions at the vertices. Next, we present the reinterpretation of the 2D HLL scheme as a 2D PVM scheme and the natural extension to a class of two-dimensional incomplete Riemann solvers relying on the AVM framework. The performances of the proposed numerical schemes are put to the test with several numerical experiments. An additional and important feature of our solvers is that they are theoretically stable up to a CFL number of unity.
- The form in which the two-dimensional AVM solvers have been defined in the previous chapter opens the door to extend them to the case of hyperbolic systems in non-conservative form, within the path-conservative ansatz. In particular, this allows to extend the two-dimensional AVM solvers to the case of hyperbolic systems with coupling and source terms. A possible way to do this is presented in Chapter 4, where the simplified four-waves model of the previous chapter is considered, together with a strategy to build general multidimensional Riemann solvers. The corresponding numerical schemes are shown to be consistent with the nonconservative system for smooth solutions. The extension to general AVM-type schemes is not straightforward, some algebra is needed to adjust the terms in a suitable way to make visible the viscosity matrices. Then, the 2D HLL scheme is rewritten in PVM form, splitting the vertex contributions in four contributions for each coordinate direction. The following step consists in substituting the underlying HLL polynomial by other suitable function which serves as basis of the corresponding AVM fluctuations. We prove the consistency and linear L^∞ -stability of the resultant first-order schemes. Second-order accuracy in space and time is obtained by means of a predictor-corrector MUSCL-

Hancock procedure. Applications to one-layer and two-layer shallow water equations have been considered. The schemes have been proven to be well-balanced for multilayer shallow water systems. An additional advantage of the proposed two-dimensional solvers is that they are stable up to a CFL number of unity, whereas the maximal CFL for schemes based on dimensional splitting is usually 0.5.

- In Chapter 5, a simple and efficient GPU algorithm is presented. The two-dimensional AVM schemes are rewritten in a suitable way in order to unify the implementations of the conservative numerical scheme presented in Chapter 3 and the nonconservative scheme presented in Chapter 4. Improvements on the implementations are proposed, in order to avoid matrix multiplications when calculating the viscosity matrices associated to polynomial/rational-based schemes. The main steps of the general CUDA algorithm are listed. Strategies to avoid memory conflicts in vertex calculations are proposed. Comparison of the genuinely 2D schemes and their $1D \times 1D$ counterparts in terms of efficiency have been performed through numerical experiments.
- The thesis ends with some concluding remarks and future research lines in Chapter 6.

Publications

This thesis is based on the following publications:

- J.M. Gallardo, K.A. Schneider and M.J. Castro. On a class of two-dimensional incomplete Riemann solvers. *Journal of Computational Physics*, 386 (2019), pp. 541-547. DOI: <https://doi.org/10.1016/j.jcp.2019.02.034>.
- J.M. Gallardo, K.A. Schneider and M.J. Castro. On a class of genuinely 2D incomplete Riemann solvers for hyperbolic systems. *Computational and Mathematical Methods*, (2019). DOI: <https://doi.org/10.1002/cmm4.1074>.
- K.A. Schneider, J.M. Gallardo, D.S. Balsara, B. Nkonga and C. Parés. Multidimensional approximate Riemann solvers for hyperbolic nonconservative systems. Applications to shallow water systems. *Journal of Computational Physics*. In preparation (2020).
- K.A. Schneider and J.M. Gallardo. Efficient GPU implementation for high-order multidimensional incomplete Riemann solvers for hyperbolic nonconservative systems. Applications to shallow water systems with topography and dry areas. In preparation (2020).

Introduction (in Spanish)

En la actualidad existe una creciente necesidad de desarrollar métodos numéricos potentes para la simulación de procesos de la Física y la Ingeniería en situaciones prácticas. En particular, un requerimiento importante es que los métodos numéricos sean robustos y computacionalmente eficientes. Normalmente, los modelos matemáticos se basan en principios físicos, tales como leyes de conservación, para proporcionar aproximaciones realistas a los procesos observados en la naturaleza. El método de volúmenes finitos (FV) ha sido ampliamente usado en el contexto de leyes de conservación, ya que asegura la conservación de cantidades físicamente relevantes que aparecen en las ecuaciones (véanse, e. g., [126, 127, 131, 136, 190, 198, 202]). Entre las ventajas de este método podemos destacar: la flexibilidad para tratar geometrías complejas (véanse, por ejemplo, [34, 71, 81, 82, 133]), que aparecen a menudo en las aplicaciones prácticas; la posibilidad de realizar refinamientos de malla locales sin afectar al resto del mallado computacional; y el tratamiento de un nivel elevado de complejidad física, por ejemplo, al simular flujos turbulentos.

Sin embargo, muchos modelos no pueden escribirse en forma conservativa, y la aparición de términos no conservativos incrementa la dificultad para diseñar esquemas numéricos precisos. Una manera de solventar esta dificultad en el contexto de los esquemas de volúmenes finitos fue propuesta por Parés ([160]), quien introdujo la noción de esquemas *camino-conservativos*. Desde entonces, numerosos investigadores han adoptado este punto de vista: véanse, e. g., [52, 38, 56, 48, 43, 59, 41, 47, 45, 49, 50, 51, 40, 55, 57, 62, 77, 79, 78, 80, 85, 84, 86, 75, 83, 102, 101, 117, 149, 152, 178]. Además del desarrollo de nuevos esquemas, el uso de esquemas camino-conservativos permite extender varios de los esquemas presentes en la literatura al caso no conservativo: Lax-Friedrichs ([125]), Rusanov ([171]), Lax-Wendroff ([129]), HLL ([115]), FORCE ([191]), MUSTA ([189, 193]) etc. Entre todos ellos, uno de los más importantes es el método de Roe, cuya definición se basa en el concepto de linealización de Roe introducido en [197]. Hoy en día, los sistemas no conservativos y los esquemas camino-conservativos constituyen un frente de investigación muy activo, con un amplio rango de aplicaciones: véanse, por ejemplo, [11, 46, 55, 92, 91, 90, 96, 142, 174, 175].

En su trabajo pionero [108], Godunov propuso el uso de la solución exacta de problemas de Riemann en los flujos numéricos. Sin embargo, la solución exacta de un problema de Riemann no lineal puede ser difícil de determinar, lo que en general hace que el diseño de métodos numéricos siguiendo las ideas de Godunov sea una dura tarea. En este sentido, Roe ([168]) propuso un esquema de tipo Godunov basado en soluciones locales aproximadas del problema de Riemann en cada interfaz, sustituyendo el problema de Riemann original por uno linealizado. La desventaja en este caso es que para la construcción de una linealización de Roe es necesario conocer la estructura espectral completa del sistema. Para muchos sistemas hiperbólicos de interés práctico,

tales como las ecuaciones de aguas someras multicapa, no se conoce la descomposición espectral. Por tanto, es interesante considerar resolvidores de Riemann *incompletos*, en los cuales solo se utiliza parte de la información espectral del sistema bajo estudio. En esta dirección, el esquema de Harten-Lax-van Leer (HLL) ([115]) aproxima la solución del problema de Riemann usando tan solo las velocidades de propagación maximales, con la contrapartida de incrementar la disipación numérica. Algunas variantes como HLLC ([192, 24]) and HLLEM ([88, 89]) tienen menos disipación que el esquema HLL original, pero necesitan de alguna información espectral adicional.

En [41] se introdujo una nueva clase de resolvidores de Riemann camino-conservativos incompletos, los denominados métodos PVM (Polynomial Viscosity Matrix), en los cuales la matriz de viscosidad numérica se calcula como una evaluación polinomial de la matriz de Roe, si es conocida, o del jacobiano del flujo en un estado intermedio, y solo necesita una cota de la máxima velocidad de propagación. Muchos esquemas clásicos como Rusanov, HLL, FORCE, Roe, etc., pueden interpretarse como métodos PVM: see [41]. Una extensión natural utilizando funciones racionales dio lugar a la clase de métodos RVM (Rational Viscosity Matrix) en [49]; a partir de ahora, usaremos el término AVM (Approximate Viscosity Matrix) para referirnos a ambos tipos de resolvidores, PVM y RVM. Como se indicó en [104], una elección apropiada de las funciones de base para un resolvidor AVM permite controlar la cantidad de difusión numérica del esquema asociado, lo que hace que estos métodos sean particularmente eficientes a la hora de resolver problemas complejos. Otras extensiones de los métodos AVM se propusieron en [51], donde se propusieron esquemas PVM libres de jacobiano, y [50], donde se construyeron aproximaciones del método clásico de Osher-Solomon (véase también [85]).

Cuando se resuelve un problema multidimensional, es usual considerar problemas de Riemann unidimensionales proyectados sobre las aristas de las celdas del dominio computacional. Por otro lado, los mallados bidimensionales consisten en una colección de aristas y vértices, por lo que de forma natural un resolvidor de Riemann debería involucrar problemas de Riemann unidimensionales en las aristas y problemas de Riemann bidimensionales en los vértices (véase la figura A). De esta forma, diversos investigadores consideran que en los métodos proyectados los resolvidores unidimensionales pierden mucha de su eficiencia, principalmente debido al hecho de que no toman en cuenta las características de la solución que se propagan transversalmente a las fronteras de las celdas. Además, el sesgo direccional que introducen los resolvidores unidimensionales produce una reducción del número de Courant permitido en la simulación de flujos multidimensionales. Por estos motivos, ha habido diversos intentos en la literatura para construir resolvidores de Riemann puramente multidimensionales (véanse [1, 2, 3, 32, 97, 98, 105, 134, 140, 199, 206]; véase también [169] para una discusión más detallada sobre descentrado multidimensional), aunque la mayoría de ellos están diseñados específicamente para las ecuaciones de Euler o de la magnetohidrodinámica (MHD). Recientemente, se han propuesto resolvidores de Riemann multidimensionales más generales, por ejemplo, en [14, 15, 19, 16, 18, 21, 74, 104, 103, 178, 203].

En [206], Wendroff propuso un resolvidor de Riemann bidimensional para las ecuaciones de Euler, que extendía el método HLL unidimensional y en el cual las interacciones bidimensionales en los vértices de las celdas se tenían en cuenta mediante las soluciones aproximadas de problemas de Riemann bidimensionales. Sin embargo, un inconveniente del resolvidor de Wendroff es que no dispone de expresiones explícitas que permitan su implementación directa; además, no parece

sencillo construir extensiones de alto orden. Algunos años después, Balsara ([14]) modificó la formulación de Wendroff y propuso un resolvidor HLL bidimensional sobre mallas estructuradas para las ecuaciones de Euler y de MHD, que incluía expresiones explícitas para los flujos y permitía una extensión sencilla a alto orden. Una versión más robusta del resolvidor de Balsara se propuso más adelante en [15], y una extensión a mallas no estructuradas se presentó en [19]. Recientemente, se ha propuesto otra interesante reformulación del método de Wendroff en [203]. Asimismo, una nueva clase de resolvidores de Riemann multidimensionales con estructura interna autosimilar se ha introducido recientemente en [16, 18, 21].

Siguiendo la senda de los trabajos previamente comentados, la primera parte de esta tesis se centra en una nueva clase de resolvidores de Riemann genuinamente bidimensionales, basados en métodos de tipo AVM, que pueden aplicarse a leyes de conservación generales: véase [104]. Estos métodos pueden verse como extensiones naturales del método HLL de Balsara en [14], donde el flujo en las aristas se escribe como una combinación lineal convexa adecuada de flujos AVM unidimensionales y bidimensionales. Las contribuciones bidimensionales se generan a partir de la solución aproximada de un problema de Riemann bidimensional en cada vértice. La idea principal se basa en reinterpretar el esquema HLL bidimensional de Balsara como un esquema PVM bidimensional, expresando de forma explícita la matrices de viscosidad en las direcciones principales. En otras palabras, los flujos HLL bidimensionales se reescriben como una combinación de flujos unidimensionales PVM-HLL. Una vez que las matrices de viscosidad HLL unidimensionales se han identificado, es posible definir esquema bidimensionales AVM más precisos.

Los esquemas propuestos pueden aplicarse a leyes de conservación generales, aunque en esta tesis hemos considerado, en el caso conservativo, aplicaciones a las ecuaciones de MHD. Existen varias razones para esta elección. La primera es que el trabajo original de Balsara ([14]) trataba con las ecuaciones de MHD; de esta forma, parece natural analizar si nuestros esquemas también funcionan bien en este contexto. El segundo motivo está relacionado con la filosofía AVM: queremos mostrar que este tipo de aproximaciones funciona bien para sistemas muy complicados (como ya se hizo anteriormente para los esquemas PVM y RVM unidimensionales en [41, 49, 50]). Finalmente, otra razón consiste en introducir una técnica novedosa para la imposición de la condición de divergencia nula, basada en la escritura no conservativa de las ecuaciones de MHD. Como es bien sabido, una dificultad que aparece en la resolución numérica de las ecuaciones de MHD es la condición de divergencia nula del campo magnético, que tiene que imponerse para asegurar la estabilidad y precisión de los esquemas numéricos. Se han propuesto en la literatura diversas técnicas para la imposición de la divergencia: véanse, por ejemplo, [6, 22, 20, 30, 68, 69, 76, 93, 100, 99, 146, 147, 148, 163, 164, 194, 196, 211, 215]. La técnica propuesta en nuestro trabajo consiste en utilizar el sistema MHD aumentado con los términos fuente de Godunov-Powell, dentro del marco de los esquemas camino-conservativos, con la modificación de los flujos numéricos propuesta en [149]. Los experimentos numéricos muestran que nuestra técnica proporciona resultados comparables a los del método de proyección ([30]) en lo que respecta a la precisión, pero con un menor coste computacional.

La segunda parte de esta tesis trata sobre la construcción de una extensión a segundo orden de los resolvidores de Riemann puramente bidimensionales de tipo AVM, al caso de sistemas hiperbólicos no conservativos generales. Los esquemas propuestos se han aplicado a las ecuaciones de aguas someras de una y dos capas incluyendo variaciones en el fondo, en las cuales

el término fuente debido a la topografía del fondo introduce una dificultad adicional. Una manera elegante de superar esta dificultad consiste en reformular el problema en forma no conservativa, dentro del marco de los esquemas camino-conservativos. Se ha presentado una estrategia general para la construcción de resolvedores de Riemann genuinamente bidimensionales en dicho marco. En primer lugar, se ha considerado un esquema bidimensional simple de tipo HLL para el caso no conservativo, que se basa en el modelo rectangular de cuatro ondas propuesto en [14], donde únicamente se consideran las velocidades maximales en cada dirección. Las características bidimensionales en los vértices de la malla computacional se tienen en cuenta mediante la solución aproximada de problemas de Riemann bidimensionales que tiene como estados iniciales los estados de las cuatro celdas vecinas. Estas soluciones consisten básicamente de una región de interacción fuerte, que se encuentra rodeada por los estados intermedios que surgen en los problemas unidimensionales considerados en las aristas. En el modelo considerado, se supone que la región de interacción fuerte no interactúa directamente con los cuatro estados iniciales, sino tan solo con los estados intermedios. Esta simplificación nos permite reescribir el esquema HLL bidimensional en formulación PVM, para de esta forma extenderlo naturalmente a un esquema bidimensional de tipo AVM general. Las diferencias de flujos que aparecían en la formulación conservativa se han cambiado ahora por fluctuaciones a lo largo de los caminos considerados en la definición de los productos no conservativos.

Muchos problemas del mundo real demandan una respuesta rápida para ayudar en el proceso de toma de decisiones, como es el caso de los desastres naturales (véase, por ejemplo, aplicaciones de tipo oceanográfico en [87]). Los modernos procesadores gráficos (GPU) contienen cientos de procesadores que realizan operaciones aritméticas en paralelo, a un coste asequible. La compañía de diseño de GPU NVIDIA proporciona la herramienta de programación CUDA ([156]), junto con soporte para el lenguaje de programación C, para que de esta forma el programador no tenga necesidad de tratar con complejidades de lenguaje de bajo nivel. Se han propuesto en la literatura numerosos códigos basados en CUDA: véanse, por ejemplo, [7, 9, 10, 11, 8, 23, 33, 54, 92, 101, 124, 142, 143, 159, 174, 173, 208, 209]. Los esquemas numéricos bidimensionales propuestos en esta tesis presentan un alto potencial de paralelización, el cual los hace muy adecuados para ser implementados en un código CUDA. Presentamos una implementación eficiente para GPU de los esquemas propuestos, que puede aplicarse a la resolución de sistemas hiperbólicos no conservativos generales.

Esquema de la memoria

El esquema de esta memoria es el siguiente:

- En el capítulo 1 se establecen los antecedentes necesarios para el diseño de los resolvedores bidimensionales propuestos, tanto para el caso conservativo como para el no conservativo. Se recopilan los resultados teóricos fundamentales sobre leyes de conservación y aproximaciones de volúmenes finitos. Asimismo, se presentan de forma autocontenida los resultados necesarios sobre sistemas no conservativos y esquemas camino-conservativos. También se hace un breve repaso sobre los esquemas de tipo AVM, prestando especial atención a los resolvedores PVM basados en aproximaciones internas y de Chebyshev, y a los resolvedores RVM basados en funciones racionales de tipo Newman y Padé. Estos son precisamente

los resolvedores AVM que se han usado en las simulaciones numéricas presentadas en esta tesis. Por último, se proporciona un marco general para la construcción de extensiones de alto orden de los esquemas camino-conservativos bidimensionales.

- En el capítulo 2, se hace en primer lugar un repaso sobre las ecuaciones de MHD. Además, se presenta una técnica novedosa para la imposición de la condición de divergencia nula. Nuestro método se basa en la formulación no conservativa de las ecuaciones de MHD, donde la divergencia del campo magnético aparece como un término fuente. Esta técnica se ha incluido en los resolvedores AVM bidimensionales mediante la teoría de los esquemas camino-conservativos, con una modificación adecuada de los flujos numéricos.

Se realiza asimismo un breve resumen sobre la estructura espectral de las ecuaciones de aguas someras con topografía del fondo. La presencia de términos fuente introduce una dificultad numérica adicional relacionada con el cálculo de las soluciones estacionarias o casi estacionarias. Los resultados principales sobre *bien equilibrado* también se han explicado, incluyendo la técnica de modificación de la identidad.

- El capítulo 3 comienza con un resumen del modelo de cuatro ondas considerado por Balsara in [14], incluyendo la construcción detallada del esquema HLL bidimensional para leyes de conservación. En cada arista del volumen de control, los flujos numéricos se construyen como una combinación lineal convexa adecuada de una contribución unidimensional en la arista, y dos contribuciones bidimensionales en los vértices. A continuación, realizamos la reinterpretación del esquema HLL 2D como un esquema PVM 2D, y la extensión natural a una clase de resolvedores de Riemann bidimensionales incompletos basados en la técnica AVM. El rendimiento de los esquemas numéricos propuestos se ha analizado mediante la realización de diversos experimentos numéricos. Una característica adicional importante de nuestros esquemas es que son estables, al menos teóricamente, hasta un número CFL de uno.
- La forma en que se han definido los esquemas AVM bidimensionales en el capítulo precedente abre la posibilidad de extenderlos al caso de sistemas hiperbólicos en forma no conservativa, usando para ello la teoría de esquemas camino-conservativos. En particular, esto permite extender los esquemas al caso de sistemas hiperbólicos con términos fuente y de acoplamiento. Una posible forma de hacer esto se presenta en el capítulo 4, basado en el modelo simplificado de cuatro ondas introducido en el capítulo anterior, junto con una estrategia para construir resolvedores de Riemann multidimensionales generales. Se muestra que los correspondientes esquemas numéricos son consistentes con el sistema no conservativo en el caso de soluciones regulares. La extensión a esquemas generales de tipo AVM no es inmediata, ya que es necesario realizar ajustar los términos de forma adecuada para hacer visibles las matrices de viscosidad. Una vez hecho esto, el esquema HLL 2D se reescribe en forma PVM, separando las contribuciones en los vértices en cuatro contribuciones para cada dirección coordenada. El siguiente paso consiste en sustituir el polinomio HLL subyacente por otra función adecuada, que sirva de base para construir las correspondientes fluctuaciones AVM. Demostramos la consistencia y la estabilidad lineal L^∞ del esquema de primer orden resultante. Asimismo, mediante un método predictor-corrector de tipo MUSCL-Hancock, se construye un esquema de segundo orden en tiempo

y en espacio. Para probar la eficiencia de los esquemas, se han considerado aplicaciones a las ecuaciones de aguas someras con una y dos capas. Los esquemas han demostrado ser bien equilibrados para soluciones de agua en reposo. Una ventaja adicional de los esquemas bidimensionales propuestos es que son estables para un número CFL de hasta la unidad, mientras que el CFL maximal para esquemas definidos dimensión por dimensión es usualmente 0.5.

- En el capítulo 5 se introduce un algoritmo simple y eficiente para GPU. Los esquemas AVM bidimensionales se reformulan de forma adecuada para unificar las implementaciones del método numérico para el caso conservativo considerado en el capítulo 3, y las del esquema para el caso no conservativo definido en el capítulo 4. Se proponen mejoras en las implementaciones, con el fin de evitar productos de matrices a la hora de calcular las matrices de viscosidad asociadas a los esquemas polinomiales y racionales. Se detallan asimismo los principales pasos en la definición del algoritmo CUDA general, y se proponen estrategias para evitar conflictos de memoria. Se comparan, por último, la eficiencia de los esquemas genuinamente 2D con la de sus contrapartidas 1D×1D a través de una serie de experimentos numéricos.
- La memoria concluye con algunos comentarios y una descripción de posibles líneas de trabajo futuras en el capítulo 6.

Publicaciones

Esta tesis se basa en las siguientes publicaciones:

- J.M. Gallardo, K.A. Schneider and M.J. Castro. On a class of two-dimensional incomplete Riemann solvers. *Journal of Computational Physics* 386, pp. 541-547 (2019). DOI: <https://doi.org/10.1016/j.jcp.2019.02.034>.
- J.M. Gallardo, K.A. Schneider and M.J. Castro. On a class of genuinely 2D incomplete Riemann solvers for hyperbolic systems. *Computational and Mathematical Methods* (2019). DOI: <https://doi.org/10.1002/cmm4.1074>.
- K.A. Schneider, J.M. Gallardo, D.S. Balsara, B. Nkonga and C. Parés. Multidimensional approximate Riemann solvers for hyperbolic nonconservative systems. Applications to shallow water systems. *Journal of Computational Physics*. In preparation (2020).
- K.A. Schneider and J.M. Gallardo. Efficient GPU implementation for high-order multidimensional incomplete Riemann solvers for hyperbolic nonconservative systems. Applications to shallow water systems with topography and dry areas. In preparation (2020).

Chapter 1

Hyperbolic systems and finite volume schemes

The purpose of this chapter is to provide a background about the theory of hyperbolic systems, in both the conservative and nonconservative cases, in order to state the notations and terminology that will be used in this memory. We describe a general methodology for developing high order schemes for hyperbolic systems of conservation laws, or, more generally, for hyperbolic systems with nonconservative products and/or source terms.

The aim of Sections 1.1 and 1.2 is to recall some of the main theoretical results about conservation laws and finite volume (FV) methods, respectively. These sections are primarily an overview; for further information about this topics, we refer to the classical books [107, 136, 190, 202, 214] and the references therein. In Section 1.3 we focus on nonconservative systems, in which the main difficulty, from both the numerical and the theoretical points of view, comes from the presence of nonconservative products, that do not make sense as distributions when the data present discontinuities. To formalize the notion of weak solutions for a nonconservative system there are several mathematical theories: Volpert [204], Colombeau [61], Dal Maso-LeFloch-Murat (DLM) [145]. We succinctly speak about the DLM theory, which conforms the basis of the so-called path-conservative schemes, introduced by Parés in [160]; this will be discussed in Section 1.4. We show that many well-known finite volume schemes for conservation laws can be extended for nonconservative systems under the framework of path-conservative schemes. In recent years, great progress have been made on path-conservative methods: see, for example, Parés ([160]), Castro et al. ([52, 54, 56, 42, 48, 43, 59, 41, 60, 50, 55]), Gallardo et al. ([102, 101, 103]), Dumbser et al. ([81, 82, 77, 79, 78, 80, 85, 84, 86, 75, 83]), Morales de Luna et al. ([149]), Muñoz et al. ([152]), among others. A review can be seen in [57]. Chapter 4 of this thesis, which is based on the paper [178], deals with nonconservative systems and path-conservative schemes for genuinely 2D Riemann solvers. In Section 1.5 we present a review on Approximate Viscosity Matrix (AVM) methods, which are a class of computationally efficient first-order path-conservative schemes, which will play a key role in this dissertation. This class of methods constitutes a family of incomplete Riemann solvers, in the sense that it is not necessary to compute the full eigenstructure of the system, but only a bound of the maximal speed of propagation. The numerical viscosity matrix is computed through polynomial or rational

evaluations of a Roe matrix, if available, or the Jacobian matrix of the flux evaluated at some average state. Among several AVM-type solvers, the results in this thesis are based on: a family of iterative internal polynomial approximations given in [51], and a class of rational Padé approximations introduced in [104] for MHD simulations (Chapter 3); a family of AVM methods based on Chebyshev polynomials as well as on Newman rational functions, both given in [49], for simulations in shallow water flows (Chapter 4). The high-order extension of the schemes is introduced in Sections 1.6 and 1.7.

1.1 Conservation laws

Formally, a general multidimensional system of conservation laws takes the form

$$\frac{d}{dt} \int_{\Omega} \mathbf{U}(\mathbf{x}, t) d\mathbf{x} + \sum_{j=1}^N \int_{\partial\Omega} \mathbf{F}_j(\mathbf{U}(\mathbf{x}, t)) \cdot \boldsymbol{\eta}_{x_j} d\gamma = 0, \quad (1.1.1)$$

where the mapping

$$\begin{aligned} \mathbf{U}: \Omega \times [0, \infty) &\longrightarrow \mathcal{O} \\ (x_1, \dots, x_N, t) &\longmapsto (U_1, \dots, U_m) \end{aligned}$$

is the set of conserved variables, considering that $\mathbf{U} = (U_1, \dots, U_m)$ is a m -component vector and $\Omega \subset \mathbb{R}^N$ is a N -dimensional open set. $\mathcal{O} \subset \mathbb{R}^m$ is called the *phase space*. We have that $\mathbf{x} = (x_1, \dots, x_N)$ and $\boldsymbol{\eta} = (\boldsymbol{\eta}_{\mathbf{x}}, \boldsymbol{\eta}_t)^T$ is an unitary vector pointing towards the boundary $\partial\Omega$. The flux functions for each j -component are defined as $\mathbf{F}_j: \mathcal{O} \rightarrow \mathbb{R}^m$, in such a way that $\mathcal{F} = (\mathbf{F}_1, \dots, \mathbf{F}_N) \subset \mathbb{R}^{N \times m}$ is the array of fluxes of the conserved quantities. In smooth regions, (1.1.1) can be rewritten in differential form as

$$\partial_t \mathbf{U} + \nabla \cdot \mathcal{F}(\mathbf{U}) = 0. \quad (1.1.2)$$

The integral form admit discontinuous solutions and a smoothness assumption is required in order to ensure that the differential form holds true. As it was pointed out in [190], this is a good reason, from the computational point of view, to consider finite volume approximations. We can interpret (1.1.1) as: the time variation on the average of \mathbf{U} on Ω is equal to the balance of mass inflow and outflow throughout the boundary $\partial\Omega$.

The Cauchy problem for (1.1.2) consists in finding a solution \mathbf{U} such that

$$\mathbf{U}(\mathbf{x}, 0) = \mathbf{U}_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (1.1.3)$$

where $\mathbf{U}_0: \Omega \rightarrow \mathcal{O}$ is a known function. To simplify the notation, let us rename the (x_1, x_2) coordinates as (x, y) . In particular, for $N = 1$ we can define the classical (one-dimensional) local Riemann problem taking

$$\mathbf{U}_0(x) = \begin{cases} U_L & \text{if } x < 0, \\ U_R & \text{if } x > 0, \end{cases}$$

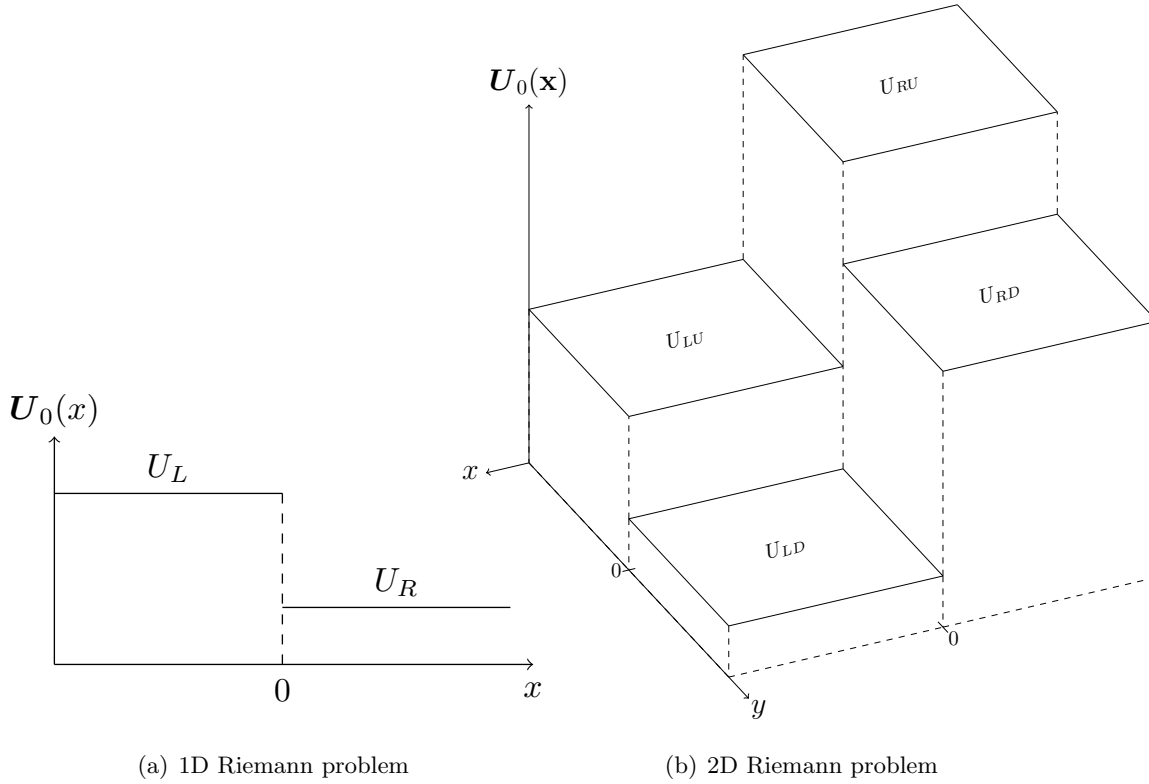


Figure 1.1: Illustration of initial data for the Riemann problems (a) 1D Riemann problem: two constant states separated by a discontinuity at $x = 0$. (b) 2D Riemann problem: four constant states, with discontinuities along the edges of any two states and in the vertex $(x, y) = (0, 0)$.

with the discontinuity situated in $x = 0$, where U_L (left) and U_R (right) are two constant values (see Figure 1.1(a)). For $N = 2$ we can define the two-dimensional local Riemann problem taking

$$U_0(\mathbf{x}) = \begin{cases} U_{LD} & \text{if } x < 0, y < 0, \\ U_{RD} & \text{if } x > 0, y < 0, \\ U_{LU} & \text{if } x < 0, y > 0, \\ U_{RU} & \text{if } x > 0, y > 0, \end{cases}$$

where we have discontinuities along the edges between any two neighboring states and in the vertex point $(x, y) = (0, 0)$. We assume here that U_{LD} (left down), U_{RD} (right down), U_{LU} (left up) and U_{RU} (right up) are constant values (see Figure 1.1(b)).

Let us consider the Jacobian matrix of system (1.1.2), given by

$$A(\mathbf{U}, \boldsymbol{\eta}) = \sum_{j=1}^N A_j(\mathbf{U})\eta_{x_j}. \tag{1.1.4}$$

The system (1.1.2) is called *hyperbolic* at the point (\mathbf{x}, t) if the matrix (1.1.4) has m real

eigenvalues

$$\lambda_1(\mathbf{U}, \boldsymbol{\eta}) \leq \dots \leq \lambda_m(\mathbf{U}, \boldsymbol{\eta}),$$

and a corresponding set of m linearly independent right eigenvectors $\{R_1(\mathbf{U}, \boldsymbol{\eta}), \dots, R_m(\mathbf{U}, \boldsymbol{\eta})\}$. The system is called *strictly hyperbolic* if the eigenvalues are all distinct. The matrix (1.1.4) is said *diagonalizable* if it can be expressed as

$$\mathbf{A}(\mathbf{U}, \boldsymbol{\eta}) = \mathcal{R}(\mathbf{U}, \boldsymbol{\eta})\boldsymbol{\Lambda}(\mathbf{U}, \boldsymbol{\eta})\mathcal{R}^{-1}(\mathbf{U}, \boldsymbol{\eta}),$$

where $\boldsymbol{\Lambda}$ is the diagonal matrix formed by the eigenvalues λ_j and $\mathcal{R} = (R_1 | \dots | R_m)$. The system (1.1.2) is said *diagonalizable* if the matrix (1.1.4) is diagonalizable. A λ_j -characteristic field is said to be *linearly degenerate* if

$$\nabla \lambda_j(\mathbf{U}, \boldsymbol{\eta}) \cdot R_j(\mathbf{U}, \boldsymbol{\eta}) = 0, \quad \forall \mathbf{U} \in \mathcal{O},$$

while it is *genuinely nonlinear* if

$$\nabla \lambda_j(\mathbf{U}, \boldsymbol{\eta}) \cdot R_j(\mathbf{U}, \boldsymbol{\eta}) \neq 0, \quad \forall \mathbf{U} \in \mathcal{O}.$$

The admissibility of discontinuous solutions for (1.1.2) makes it necessary to define the concept of *weak solutions* for the Cauchy problem (1.1.2)-(1.1.3).

Definition 1. [202] Let us assume that $\mathbf{U} \in L_{loc}^\infty(\Omega)^m$. A function $\mathbf{U} \in L_{loc}^\infty(\Omega \times [0, \infty))^m$ is called a *weak solution* of the Cauchy problem (1.1.2)-(1.1.3) if it verifies

$$\int_0^\infty \int_\Omega \left\{ \mathbf{U} \frac{\partial \phi}{\partial t} + \sum_{j=1}^N \mathbf{F}_j(\mathbf{U}) \frac{\partial \phi}{\partial x_j} \right\} d\mathbf{x} dt + \int_\Omega \mathbf{U}_0(\mathbf{x}) \cdot \phi(\mathbf{x}, 0) d\mathbf{x} = 0, \quad \forall \phi \in C_c^1(\Omega \times [0, \infty))^m. \quad (1.1.5)$$

In this way, any classical solution of (1.1.2)-(1.1.3) also satisfy (1.1.5). Conversely, any weak solution satisfies (1.1.5) in the distributional sense.

Theorem 1. [202] Let $\mathbf{U}: \Omega \times [0, \infty) \rightarrow \mathcal{O}$ be a piecewise C^1 function. Then, \mathbf{U} is a solution of (1.1.2) in the sense of distributions on $\Omega \times (0, \infty)$ if and only if the following two conditions hold:

- (i) \mathbf{U} is a classical solution of (1.1.2) in the domain where it is C^1 .
- (ii) \mathbf{U} verifies the jump condition

$$(\mathbf{U}^+ - \mathbf{U}^-)\eta_t + \sum_{j=1}^N (\mathbf{F}_j(\mathbf{U}^+) - \mathbf{F}_j(\mathbf{U}^-))\eta_{x_j} = 0 \quad (1.1.6)$$

along any surface of discontinuity, where $\boldsymbol{\eta} = (\eta_{x_1}, \dots, \eta_{x_N}, \eta_t)$ is the unit normal vector to the surface, pointing from Ω_+ to Ω_- , which are the two open subsets of Ω which are separated by the surface.

The expression (1.1.6) is known as the *Rankine-Hugoniot (RH) condition*. If $(\eta_{x_1}, \dots, \eta_{x_N}) \neq 0$, we can take $\boldsymbol{\eta} = (\mu, -s)$, where $s \in \mathbb{R}$ and $\mu = (\mu_1, \dots, \mu_N)$ is a unit vector in \mathbb{R}^N . So, (1.1.6) is equivalent to

$$s[\mathbf{U}] = \sum_{j=1}^N \mu_j [\mathbf{F}_j(\mathbf{U})]$$

where s is known as the *shock velocity* or velocity of propagation of the discontinuity. Here $[\mathbf{U}] = \mathbf{U}^+ - \mathbf{U}^-$ and, analogously, $[\mathbf{F}_j(\mathbf{U})] = \mathbf{F}_j(\mathbf{U}^+) - \mathbf{F}_j(\mathbf{U}^-)$.

In general, a weak solution of (1.1.2)-(1.1.3) is not necessarily unique, so we need to add some extra requirements (ideally from physical principles) to have uniqueness. Notice that some weak solutions can indeed be unphysical solutions, so *physically admissible* weak solutions, or *entropy solutions*, are identified as solutions that satisfy certain additional *entropy conditions*, related to certain *entropy functions* of the system. Let us consider an entropy pair $(\mathcal{H}, \mathcal{Q})$, i.e., a pair of regular functions $\mathcal{H}: \Omega \rightarrow \mathbb{R}$, \mathcal{H} convex, and $\mathcal{Q} = (\mathcal{Q}_1, \dots, \mathcal{Q}_N): \Omega \rightarrow \mathbb{R}^N$ such that

$$\nabla \mathcal{Q}_j(\mathbf{U}) = \nabla \mathcal{H}(\mathbf{U}) \cdot \mathbf{A}_j(\mathbf{U}), \quad \forall \mathbf{U} \in \Omega, \quad 1 \leq j \leq N$$

Definition 2. A weak solution of (1.1.2)-(1.1.3) is said to be an entropy solution if it verifies the entropy condition

$$\partial_t \mathcal{H}(\mathbf{U}) + \nabla \cdot \mathcal{Q}(\mathbf{U}) \leq 0$$

in the sense of distributions, for all entropy pair $(\mathcal{H}, \mathcal{Q})$.

1.2 Finite volume approximation

In what follows, let us consider a two-dimensional system of conservation laws ($N = 2$ in (1.1.2)) of the form

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) + \partial_y \mathbf{G}(\mathbf{U}) = 0, \quad (1.2.1)$$

where \mathbf{U} is defined on $\Omega \times [0, T]$, being $\Omega \subset \mathbb{R}^2$ a domain, and takes values on an open convex set $\mathcal{O} \subset \mathbb{R}^m$; $\mathcal{F} = (\mathbf{F}, \mathbf{G})$ is a regular function from \mathcal{O} to $\mathbb{R}^m \times \mathbb{R}^m$. We are interested in the numerical solution of (1.2.1) by means of finite volume methods on structured meshes.

The spatial domain is divided into rectangular cells $C_{ij} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$ of size $\Delta x \times \Delta y$ and center (x_i, y_j) , and the time step is denoted as $t^n = n\Delta t$. Integrating (1.2.1) on the control volume $C_{ij} \times [t^n, t^{n+1}]$ yields

$$\mathbf{U}_{ij}^{n+1} = \mathbf{U}_{ij}^n - \frac{\Delta t}{\Delta x} (\mathbf{F}_{i+1/2,j}^n - \mathbf{F}_{i-1/2,j}^n) - \frac{\Delta t}{\Delta y} (\mathbf{G}_{i,j+1/2}^n - \mathbf{G}_{i,j-1/2}^n), \quad (1.2.2)$$

where the value \mathbf{U}_{ij}^n is an approximation of the average value of \mathbf{U} over each cell C_{ij} at time t^n ,

$$\mathbf{U}_{ij}^n \approx \frac{1}{|C_{ij}|} \int_{C_{ij}} \mathbf{U}(x, y, t^n) dx dy, \quad (1.2.3)$$

and the numerical fluxes $\mathbf{F}_{i+1/2,j}^n$ and $\mathbf{G}_{i,j+1/2}^n$ are approximations of the mean value of the fluxes along the straight lines $x = x_{i+1/2}$ and $y = y_{j+1/2}$, respectively, in the time interval $[t^n, t^{n+1}]$:

$$\mathbf{F}_{i+1/2,j}^n \approx \frac{1}{\Delta y \Delta t} \int_{t^n}^{t^{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{F}(\mathbf{U}(x_{i+1/2}, y, t)) dy dt, \quad (1.2.4)$$

and

$$\mathbf{G}_{i,j+1/2}^n \approx \frac{1}{\Delta x \Delta t} \int_{t^n}^{t^{n+1}} \int_{x_{j-1/2}}^{x_{j+1/2}} \mathbf{G}(\mathbf{U}(x, y_{j+1/2}, t)) dx dt. \quad (1.2.5)$$

The next step is to define how to compute the numerical fluxes on the boundaries of the cell, which may depend on the neighboring cell averages around \mathbf{U}_{ij}^n . Usual definitions of the numerical fluxes are based on one-dimensional Riemann solvers in the coordinate directions, which lead to expressions of the form $\mathbf{F}_{i+1/2,j} = \mathcal{F}(\mathbf{U}_{ij}^n, \mathbf{U}_{i+1,j}^n)$ and $\mathbf{G}_{i,j+1/2}^n = \mathcal{G}(\mathbf{U}_{ij}^n, \mathbf{U}_{i,j+1}^n)$ for certain flux functions $\mathcal{F}(\mathbf{U}, \mathbf{V})$ and $\mathcal{G}(\mathbf{U}, \mathbf{V})$, for arbitrary states \mathbf{U}, \mathbf{V} . An essential requirement for a numerical flux is the consistency with the physical flux. In other words, if $\mathbf{U}(\mathbf{x}, t) \equiv \bar{\mathbf{U}}$ is constant in space, then \mathbf{U} will not change as time evolves, and the integrals (1.2.4) and (1.2.5) are reduced to $\mathbf{F}(\bar{\mathbf{U}})$ and $\mathbf{G}(\bar{\mathbf{U}})$, respectively, and \mathbf{U} in (1.2.2) remains constant. Therefore, the numerical fluxes are consistent if

$$\mathcal{F}(\bar{\mathbf{U}}, \bar{\mathbf{U}}) = \mathbf{F}(\bar{\mathbf{U}}), \quad \mathcal{G}(\bar{\mathbf{U}}, \bar{\mathbf{U}}) = \mathbf{G}(\bar{\mathbf{U}}), \quad \forall \bar{\mathbf{U}} \in \mathcal{O}.$$

Usually, some additional requirement like Lipschitz continuity must be satisfied by the numerical fluxes. An important convergence result is due a Lax and Wendroff [129], which states that if the approximate solution converges to some function $\mathbf{U}(\mathbf{x}, t)$ as the grid is refined, then this function will in fact be a weak solution of the conservation law. As it was commented in the previous section, the solution will not be necessarily unique and an additional entropy condition is needed in order to obtain a physically correct solution.

A necessary condition that must be satisfied by any finite volume method is the *CFL condition* [65], in order that the scheme be stable and convergent to the solution of the differential equation when the meshgrid is refined. However, the CFL condition is not always sufficient to guarantee stability. For instance, a simple one-directional unstable numerical flux can be constructed taking into account a simple average of states

$$\mathbf{F}_{i+1/2,j} = \mathcal{F}(\mathbf{U}_{ij}^n, \mathbf{U}_{i+1,j}^n) = \frac{1}{2}[\mathbf{F}(U_{ij}^n) + \mathbf{F}(U_{i+1,j}^n)]. \quad (1.2.6)$$

Taking the remaining fluxes in this way and substituting them in (1.2.2), the resulting method is generally unstable, no matter how small the timestep is taken for accomplishing the CFL condition. A truncation of order s , the order of our numerical method, in Taylor's expansions gives us the exact PDE satisfied by the numerical scheme: this so-called *modified equation* give us an indicator about the *numerical diffusion* of the scheme. For a first-order method, the resultant modified equation is a advection-diffusion PDE, and this analysis give us the amount of numerical diffusion of the method. Centered fluxes of the form (1.2.6) have a modified equation with negative numerical diffusion and the resultant scheme is unconditionally unstable. An additional upwinding or non-centered numerical diffusion, containing information about the directions of propagation, is needed in order to control the spurious oscillations arising in the finite volume scheme.

Assuming that the solution of (1.2.1) is a piecewise constant solution (1.2.3), we would have a Riemann problem at each of the cell boundaries. The solutions of Riemann problems are *self-similar*, which means that \mathbf{U} is a function of $\xi = \frac{x}{t}$ and $\psi = \frac{y}{t}$, referred to as *similarity variables*. We denote as $\mathcal{R}_x(\xi; \mathbf{U}, \mathbf{V})$ the self-similar solution of the one-dimensional Riemann problem in

the x -direction with initial data given by arbitrary states \mathbf{U} and \mathbf{V} ; similarly, $\mathcal{R}_y(\psi; \mathbf{U}, \mathbf{V})$ denotes the solution of the one-dimensional Riemann problem in the vertical direction.

Godunov scheme

Since the pioneering work of Godunov [108], a number of robust and accurate numerical schemes for hyperbolic conservation laws have been developed following his ideas. Godunov's approach consists in computing the numerical flux using the exact solution of the associated local Riemann problems. For instance,

$$\begin{aligned} \mathbf{F}_{i\pm 1/2,j}^n &\approx \frac{1}{\Delta y \Delta t} \int_{t^n}^{t^{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{F}(\tilde{\mathbf{U}}(x_{i\pm 1/2}, y, t)) dy dt \\ &= \frac{1}{\Delta y \Delta t} \int_{t^n}^{t^{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{F}(\mathcal{R}_x(0; \mathbf{U}_{ij}^n, \mathbf{U}_{i\pm 1,j}^n)) dy dt = \mathbf{F}(\mathcal{R}_x(0; \mathbf{U}_{ij}^n, \mathbf{U}_{i\pm 1,j}^n)) \end{aligned}$$

being $\tilde{\mathbf{U}}$ the exact solution of the associated Riemann problem, i.e.,

$$\tilde{\mathbf{U}}(\mathbf{x}, t) = \begin{cases} \mathcal{R}_x\left(\frac{x-x_{i-1/2}}{t}; \mathbf{U}_{ij}^n, \mathbf{U}_{i-1,j}^n\right) & \text{if } x \in [x_{i-1/2}, x_i), \\ \mathcal{R}_x\left(\frac{x-x_{i+1/2}}{t}; \mathbf{U}_{ij}^n, \mathbf{U}_{i+1,j}^n\right) & \text{if } x \in (x_i, x_{i+1/2}]. \end{cases}$$

The fluxes in the y -direction are constructed similarly. The method of Godunov provides an *exact Riemann solver*. A necessary CFL condition, in order to prevent the interaction of solutions from local Riemann problems, is given by

$$\Delta t \max_i (\lambda_\alpha^1(\mathbf{U}_{ij}^n, \mathbf{U}_{i+1,j}^n), \lambda_\alpha^N(\mathbf{U}_{ij}^n, \mathbf{U}_{i+1,j}^n)) \leq \frac{1}{2} \Delta \alpha,$$

where $\alpha = x, y$ represents the coordinate direction, while λ_α^1 and λ_α^N denote the maximal left- and right-going wave speeds associated to two neighbor states. The upwinding in Godunov's scheme is introduced through the solutions of the local Riemann problems.

Approximate Riemann solvers

When the exact solution of the local Riemann problem has not a simple form or it is computationally expensive, approximated solutions could be considered instead. This approach consists in changing the exact solution $\tilde{\mathbf{U}}$ by an approximation $\hat{\mathbf{U}}$, which is simpler to compute. The resulting flux reads as

$$\mathbf{F}_{i\pm 1/2,j}^n = \frac{1}{\Delta y \Delta t} \int_{t^n}^{t^{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{F}(\hat{\mathbf{U}}(x_{i\pm 1/2}, y, t)) dy dt,$$

and the resulting method is called an *approximate Riemann solver*. To fix ideas, suppose first that the y -direction is frozen. Let us assume that, for every pair of states \mathbf{U}_0 and \mathbf{U}_1 in \mathcal{O} , a finite number $s \geq 1$ of speeds

$$\sigma_0 = -\infty < \sigma_1 < \dots < \sigma_s < \sigma_{s+1} = +\infty \quad (1.2.7)$$

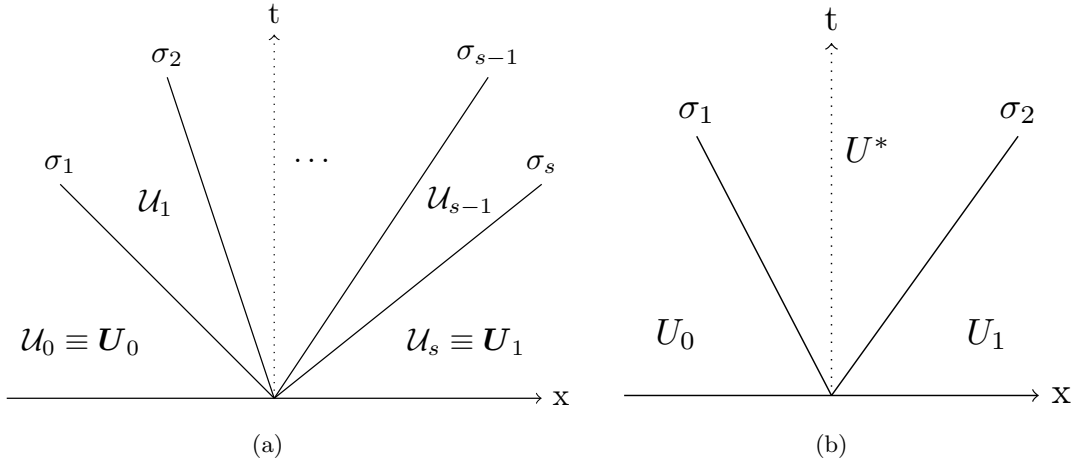


Figure 1.2: (a) Sketch of a SRS with s waves. (b) Sketch of HLL Riemann solver.

and $s - 1$ intermediate states

$$\mathcal{U}_0 = \mathbf{U}_0, \mathcal{U}_1, \dots, \mathcal{U}_{s-1}, \mathcal{U}_s = \mathbf{U}_1, \quad (1.2.8)$$

are chosen: see Figure 1.2(a). The function $\mathcal{R}_x(\xi; \mathbf{U}_0, \mathbf{U}_1) = \mathcal{U}_k$, for $\sigma_k < \xi < \sigma_{k+1}$, is said to be an approximate Riemann solver for the one-dimensional conservation law $\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = 0$ if it satisfies

$$\sum_{k=0}^{s-1} \sigma_{k+1} (\mathcal{U}_{k+1} - \mathcal{U}_k) = \mathbf{F}(\mathbf{U}_1) - \mathbf{F}(\mathbf{U}_0). \quad (1.2.9)$$

Formally, any approximate Riemann solver leads a numerical method of the form (1.2.2) where

$$\mathcal{F}(\mathbf{U}_0, \mathbf{U}_1) = \mathbf{F}(\mathbf{U}_1) + \int_0^\infty (\mathcal{R}_x(\xi; \mathbf{U}_0, \mathbf{U}_1) - \mathbf{U}_1) d\xi = \mathbf{F}(\mathbf{U}_0) - \int_{-\infty}^0 (\mathcal{R}_x(\xi; \mathbf{U}_0, \mathbf{U}_1) - \mathbf{U}_0) d\xi.$$

Similar definitions hold for the y -direction.

The easiest example comes for $s = 2$; in this case the approximate Riemann solver consists of two waves of speeds σ_1 and σ_2 linking three constant states, \mathbf{U}_0 , \mathbf{U}_1 and an intermediate one \mathbf{U}^* (see Figure 1.2(b)). Therefore, the integrals in the numerical fluxes can be easily computed, leading to the classical one-dimensional HLL scheme ([115]), where

$$\hat{\mathbf{U}} := \mathbf{U}^{HLL} = \begin{cases} \mathbf{U}_0 & \text{if } \xi \leq \sigma_1, \\ \mathbf{U}^* & \text{if } \sigma_1 \leq \xi \leq \sigma_2, \\ \mathbf{U}_1 & \text{if } \xi \geq \sigma_2, \end{cases} \quad (1.2.10)$$

$$\mathbf{U}^* = \frac{\sigma_2 \mathbf{U}_1 - \sigma_1 \mathbf{U}_0 + \mathbf{F}(\mathbf{U}_0) - \mathbf{F}(\mathbf{U}_1)}{\sigma_2 - \sigma_1} \quad (1.2.11)$$

and

$$\hat{\mathbf{F}} := \mathcal{F}^{HLL}(\mathbf{U}_0, \mathbf{U}_1) = \frac{\sigma_2^+ \mathbf{F}(\mathbf{U}_0) - \sigma_1^- \mathbf{F}(\mathbf{U}_1) + \sigma_1^- \sigma_2^+ (\mathbf{U}_1 - \mathbf{U}_0)}{\sigma_2^+ - \sigma_1^-}.$$

where $a^+ = \max(0, a)$ and $a^- = \min(0, a)$. In [168], Roe proposed a Godunov-type scheme in which the original local Riemann problem at each interface was changed by a linearized one. More precisely, a *Roe linearization* is obtained when a local linear problem is considered in (1.2.1), of the form

$$\widehat{U}_t + \widehat{A}_x \widehat{U}_x + \widehat{A}_y \widehat{U}_y = 0, \quad (1.2.12)$$

where $\widehat{A}_\zeta \equiv \widehat{A}_\zeta(\mathbf{U}, \mathbf{V})$, $\zeta = x, y$, are appropriate matrices depending on neighboring states \mathbf{U} and \mathbf{V} . It is assumed that each \widehat{A}_ζ is diagonalizable, so it can be decomposed as $\widehat{A}_\zeta = \widehat{R}_\zeta \widehat{\Lambda}_\zeta \widehat{R}_\zeta^{-1}$, where $\widehat{\Lambda}_\zeta$ is the diagonal matrix containing the eigenvalues of \widehat{A}_ζ , and the columns of \widehat{R}_ζ contains the associated eigenvectors. Since \widehat{U} is the exact solution of the locally linearized Riemann problem, the numerical fluxes are given by $\widehat{F}(\mathbf{U}) = \widehat{A}_x(\mathbf{U})$ and $\widehat{G}(\mathbf{U}) = \widehat{A}_y(\mathbf{U})$. In the x -direction we find that

$$\widehat{U} := \mathbf{U}^{Roe} = \mathbf{U}_0 + \sum_{\widehat{\lambda}_k \leq 0} \widehat{\alpha}_k \widehat{e}_k = \mathbf{U}_1 - \sum_{\widehat{\lambda}_k \geq 0} \widehat{\alpha}_k \widehat{e}_k$$

and

$$\widehat{F} := \mathbf{F}^{Roe}(\mathbf{U}_0, \mathbf{U}_1) = \mathbf{F}(\mathbf{U}_0) + \sum_{\widehat{\lambda}_k \leq 0} \widehat{\alpha}_k \widehat{\lambda}_k \widehat{e}_k = \mathbf{F}(\mathbf{U}_1) - \sum_{\widehat{\lambda}_k \geq 0} \widehat{\alpha}_k \widehat{\lambda}_k \widehat{e}_k,$$

where $\widehat{\lambda}_k$ and \widehat{e}_k are the eigenvalues and eigenvectors of \widehat{A}_x , and $\widehat{\alpha}_k$ are the coordinates of $\mathbf{U}_1 - \mathbf{U}_0$, i.e., the projections of the jumps that take place at the intermediate states expressed in the basis of eigenvectors. The numerical flux of Roe can be written as

$$\widehat{F} = \frac{1}{2}(\mathbf{F}(\mathbf{U}_0) + \mathbf{F}(\mathbf{U}_1)) - \frac{1}{2} \sum_{k=1}^m \widehat{\alpha}_k |\widehat{\lambda}_k| \widehat{e}_k = \frac{1}{2}(\mathbf{F}(\mathbf{U}_0) + \mathbf{F}(\mathbf{U}_1)) - \frac{1}{2} |\widehat{A}_x(\mathbf{U}_0, \mathbf{U}_1)| (\mathbf{U}_1 - \mathbf{U}_0),$$

and similarly for the y -direction. The method of Roe can be seen as a particular case of approximate Riemann solver with $s = m$ (see Figure 1.2(a)). A known drawback of Roe's method is that it can produce non-entropy weak solutions in the presence of sonic points. Therefore, some entropy-fix regularization is needed (for example, see [114]).

1.3 Nonconservative systems

Let us consider first-order quasi-linear PDE systems of the form

$$\partial_t \mathbf{U} + \nabla \cdot \mathcal{F}(\mathbf{U}) + \mathcal{B}(\mathbf{U}) \cdot \nabla \mathbf{U} = \mathcal{S}(\mathbf{U}) \cdot \nabla \sigma, \quad (1.3.1)$$

where the state $\mathbf{U}(\mathbf{x}, t)$ is defined on $\Omega \times [0, T]$, Ω being a domain of \mathbb{R}^2 , and takes values on an open convex subset $\mathcal{O} \subset \mathbb{R}^m$; $\mathcal{F} = (\mathbf{F}, \mathbf{G})$ and $\mathcal{S} = (\mathcal{S}_x, \mathcal{S}_y)$ are regular functions from \mathcal{O} to $\mathbb{R}^m \times \mathbb{R}^m$; $\mathcal{B} = (\mathcal{B}_x, \mathcal{B}_y)$ is a regular matrix-valued function from \mathcal{O} to $\mathcal{M}_m(\mathbb{R}) \times \mathcal{M}_m(\mathbb{R})$; finally, $\sigma(\mathbf{x})$ is a known function from Ω to \mathbb{R} . Observe that system (1.3.1) includes as particular cases systems of conservation laws ($\mathcal{B} = 0$ and $\mathcal{S} = 0$) and balance laws ($\mathcal{B} = 0$).

Notice that system (1.3.1) can be rewritten as a nonconservative system of the form

$$\partial_t \mathbf{W} + \mathcal{A}_x(\mathbf{W}) \partial_x \mathbf{W} + \mathcal{A}_y(\mathbf{W}) \partial_y \mathbf{W} = 0 \quad (1.3.2)$$

by adding the trivial equation $\partial_t \sigma = 0$ to (1.3.1) and defining

$$\mathbf{W} = \begin{pmatrix} \mathbf{U} \\ \sigma \end{pmatrix}$$

and

$$\mathcal{A}_\zeta(\mathbf{W}) = \left(\begin{array}{c|c} \mathbf{A}_\zeta(\mathbf{U}) & -\mathbf{S}_\zeta(\mathbf{U}) \\ \hline 0 & 0 \end{array} \right), \quad (1.3.3)$$

where $\mathbf{A}_\zeta(\mathbf{U}) = \mathbf{J}_\zeta(\mathbf{U}) + \mathbf{B}_\zeta(\mathbf{U})$, being $\mathbf{J}_\zeta(\mathbf{U})$ the Jacobian matrix of the flux function \mathcal{F} in the ζ -direction, $\zeta = x, y$. In what follows, it will be assumed that $\mathbf{W}(\mathbf{x}, t)$ takes values on a convex domain $\mathcal{D} \subset \mathbb{R}^M$, with $M = m + 1$, and the components \mathcal{A}_ζ are smooth and locally bounded matrix-valued functions from Ω to $\mathcal{M}_M(\mathbb{R})$. It will be also supposed that system (1.3.2) is *strictly hyperbolic* in the sense that, for each state $\mathbf{W} \in \Omega$, the matrix $\mathcal{A}_\zeta(\mathbf{W})$ has M real distinct eigenvalues. Observe that 0 is always an eigenvalue of $\mathcal{A}_\zeta(\mathbf{W})$.

As it was done for systems of conservation laws, after integration of (1.3.2) on the control volume $C_{ij} \times [t^n, t^{n+1}]$ we find

$$\begin{aligned} \int_{C_{ij}} \mathbf{W}(\mathbf{x}, t^{n+1}) d\mathbf{x} &= \int_{C_{ij}} \mathbf{W}(\mathbf{x}, t^n) d\mathbf{x} \\ &\quad - \int_{t^n}^{t^{n+1}} \int_{C_{ij}} (\mathcal{A}_x(\mathbf{W}(\mathbf{x}, t)) \partial_x \mathbf{W}(\mathbf{x}, t) + \mathcal{A}_y(\mathbf{W}(\mathbf{x}, t)) \partial_y \mathbf{W}(\mathbf{x}, t)) dx dt. \end{aligned} \quad (1.3.4)$$

The difficulty here comes from the fact that it is necessary to give a sense to the last integral, as it may be not well defined as a distribution when the solution contains discontinuities.

It is a known issue that system (1.3.2) is not properly defined within the framework of the theory of distributions as long as \mathbf{W} contains discontinuities. For that reason, as it is usual in this context, we will follow the theory of dal Maso, LeFloch and Murat [145] to give a sense to the nonconservative products as Borel measures. In particular, this definition relies on the choice of a family of Lipschitz-continuous functions $\Phi: [0, 1] \times \Omega \times \Omega \times S^1 \rightarrow \Omega$ linking states in the phase space, where $S^1 \subset \mathbb{R}^2$ denotes the unit sphere. For a complete account on this topic, we refer the reader to [43, 160] and the references therein. In particular, we will require that the family of paths Φ verifies the following natural property:

$$\int_0^1 \mathcal{A}_\zeta(\Phi_{0,2}) \frac{d\Phi_{0,2}}{ds} ds = \int_0^1 \mathcal{A}_\zeta(\Phi_{0,1}) \frac{d\Phi_{0,1}}{ds} ds + \int_0^1 \mathcal{A}_\zeta(\Phi_{1,2}) \frac{d\Phi_{1,2}}{ds} ds, \quad (1.3.5)$$

where

$$\Phi_{a,b} \equiv \Phi_{a,b}(s) = \Phi(s; \mathbf{W}_a, \mathbf{W}_b; \boldsymbol{\eta}) \quad (1.3.6)$$

denotes the path joining \mathbf{W}_a and \mathbf{W}_b , for arbitrary states $\mathbf{W}_a, \mathbf{W}_b \in \Omega$ and $\boldsymbol{\eta} = (\eta_x, \eta_y) \in S^1$, that is, $\Phi_{a,b}(0) = \mathbf{W}_a$ and $\Phi_{a,b}(1) = \mathbf{W}_b$. In addition, $\Phi(s; \mathbf{W}_a, \mathbf{W}_b; \boldsymbol{\eta}) = \Phi(1-s; \mathbf{W}_a, \mathbf{W}_b; -\boldsymbol{\eta})$. When no confuses arises, we can adopt the short notation for Φ in (1.3.6) and drop the dependence on $\boldsymbol{\eta}$. The most simple choice for Φ is given by the family of segments

$$\Phi(s) = \mathbf{W}_a + s(\mathbf{W}_b - \mathbf{W}_a), \quad (1.3.7)$$

which corresponds to the definition of nonconservative products proposed by Volpert in [204].

In fact, the family of paths allows us to give a sense to the last integral appearing in (1.3.4) for piecewise smooth functions \mathbf{W} . More precisely, given a bounded variation function $W: C_{ij} \rightarrow \mathbb{R}^M$, the Borel measure associated with the nonconservative product, for a fixed time t , is defined as

$$\begin{aligned} \langle [\mathcal{A}_x(\mathbf{W}(\cdot, t))\partial_x \mathbf{W}(\cdot, t) + \mathcal{A}_y(\mathbf{W}(\cdot, t))\partial_y \mathbf{W}(\cdot, t)]_{\Phi}, \varphi \rangle = \\ \int_{C_{ij}} (\mathcal{A}_x(\mathbf{W}(\mathbf{x}, t))\partial_x \mathbf{W}(\mathbf{x}, t) + \mathcal{A}_y(\mathbf{W}(\mathbf{x}, t))\partial_y \mathbf{W}(\mathbf{x}, t))\varphi(\mathbf{x})d\mathbf{x} + \\ \sum_{\ell} \left(\int_0^1 \mathcal{A}_x(\Phi(s; \mathbf{W}_{\ell}^-, \mathbf{W}_{\ell}^+)) \frac{d\Phi}{ds}(s; \mathbf{W}_{\ell}^-, \mathbf{W}_{\ell}^+) ds \right) \varphi(x_{\ell}(t)) \\ + \sum_m \left(\int_0^1 \mathcal{A}_y(\Phi(s; \mathbf{W}_m^-, \mathbf{W}_m^+)) \frac{d\Phi}{ds}(s; \mathbf{W}_m^-, \mathbf{W}_m^+) ds \right) \varphi(y_m(t)), \end{aligned} \quad (1.3.8)$$

where the first integral is the regular part, i.e., the derivatives that appears are the derivatives in the classical sense; \mathbf{W}_{ℓ}^{\pm} and \mathbf{W}_m^{\pm} represent, respectively, the limits of \mathbf{W} at the boundaries of the ℓ -th and m -th discontinuities (remember that the set of discontinuities of a bounded variation function is countable); and φ is a test function with compact support in C_{ij} ([152]). Observe that, in (1.3.8), the family of paths has been used to determine the weights of the Dirac measures placed at the discontinuity of \mathbf{W} . According to this definition, a weak solution can be defined as a function satisfying

$$\begin{aligned} \int_{C_{ij}} \mathbf{W}(\mathbf{x}, t^{n+1})d\mathbf{x} = \int_{C_{ij}} \mathbf{W}(\mathbf{x}, t^n)d\mathbf{x} \\ - \int_{t^n}^{t^{n+1}} \langle [\mathcal{A}_x(\mathbf{W}(\cdot, t))\partial_x \mathbf{W}(\cdot, t) + \mathcal{A}_y(\mathbf{W}(\cdot, t))\partial_y \mathbf{W}(\cdot, t)]_{\Phi}, 1_{C_{ij}} \rangle dt. \end{aligned}$$

An important concept is that of *Roe linearization* ([197]) associated to a family of paths Φ , which are functions $\mathcal{A}_{\zeta, \Phi}: \Omega \times \Omega \rightarrow \mathcal{M}_M(\mathbb{R})$, $\zeta = x, y$, verifying the following properties:

1. For each $\mathbf{W}_0, \mathbf{W}_1 \in \Omega$, $\mathcal{A}_{\zeta, \Phi}(\mathbf{W}_0, \mathbf{W}_1)$ has M distinct real eigenvalues.
2. $\mathcal{A}_{\zeta, \Phi}(\mathbf{W}, \mathbf{W}) = \mathcal{A}_{\zeta}(\mathbf{W})$, for every $\mathbf{W} \in \Omega$.
3. For any $\mathbf{W}_0, \mathbf{W}_1 \in \Omega$,

$$\mathcal{A}_{\zeta, \Phi}(\mathbf{W}_0, \mathbf{W}_1)(\mathbf{W}_1 - \mathbf{W}_0) = \int_0^1 \mathcal{A}_{\zeta}(\Phi_{0,1}) \frac{d\Phi_{0,1}}{ds} ds.$$

For a given family of paths, we use the notation $\Phi = (\Phi^U, \Phi^{\sigma})^t$ to distinguish its components, although when no confusion arises we will usually write Φ instead of Φ^U for the sake of clarity. Let us suppose that, given arbitrary states $\mathbf{W}_0 = (\mathbf{U}_0, \sigma_0)^t$ and $\mathbf{W}_1 = (\mathbf{U}_1, \sigma_1)^t$, it is possible to construct:

i. Matrices $\mathbf{J}_\zeta(\mathbf{U}_0, \mathbf{U}_1)$ such that

$$\mathbf{J}_x(\mathbf{U}_0, \mathbf{U}_1)(\mathbf{U}_1 - \mathbf{U}_0) = \mathbf{F}(\mathbf{U}_1) - \mathbf{F}(\mathbf{U}_0),$$

and

$$\mathbf{J}_y(\mathbf{U}_0, \mathbf{U}_1)(\mathbf{U}_1 - \mathbf{U}_0) = \mathbf{G}(\mathbf{U}_1) - \mathbf{G}(\mathbf{U}_0).$$

ii. Matrices $\mathbf{B}_{\zeta, \Phi}(\mathbf{U}_0, \mathbf{U}_1)$ satisfying

$$\mathbf{B}_{\zeta, \Phi}(\mathbf{U}_0, \mathbf{U}_1)(\mathbf{U}_1 - \mathbf{U}_0) = \int_0^1 \mathbf{B}_\zeta(\Phi_{0,1}) \frac{d\Phi_{0,1}}{ds} ds \quad (1.3.9)$$

for $\zeta = x, y$.

iii. Vectors $\mathbf{S}_{\zeta, \Phi}(\mathbf{U}_0, \mathbf{U}_1)$ verifying

$$\mathbf{S}_{\zeta, \Phi}(\mathbf{U}_0, \mathbf{U}_1)(\sigma_1 - \sigma_0) = \int_0^1 \mathbf{S}_\zeta(\Phi_{0,1}) \frac{d\Phi_{0,1}}{ds} ds \quad (1.3.10)$$

for $\zeta = x, y$.

Then, the matrices

$$\mathbf{A}_{\zeta, \Phi}(\mathbf{W}_0, \mathbf{W}_1) = \left(\begin{array}{c|c} \mathbf{A}_{\zeta, \Phi}(\mathbf{U}_0, \mathbf{U}_1) & -\mathbf{S}_{\zeta, \Phi}(\mathbf{U}_0, \mathbf{U}_1) \\ \hline 0 & 0 \end{array} \right), \quad \zeta = x, y, \quad (1.3.11)$$

where $\mathbf{A}_{\zeta, \Phi}(\mathbf{U}_0, \mathbf{U}_1) = \mathbf{J}_\zeta(\mathbf{U}_0, \mathbf{U}_1) + \mathbf{B}_{\zeta, \Phi}(\mathbf{U}_0, \mathbf{U}_1)$, constitute a Roe linearization for system (1.3.1) (see [43]). In the following we will drop the dependence on Φ and, when no confusion arises, also the dependence on the arguments.

Another fundamental property is the *generalized Rankine-Hugoniot condition*: across a discontinuity with speed v , we have that

$$\int_0^1 (\mathcal{A}_\zeta(\Phi(s; \mathbf{W}^-, \mathbf{W}^+)) \frac{d\Phi}{ds}(s; \mathbf{W}^-, \mathbf{W}^+) ds = \int_0^1 v \frac{d\Phi}{ds}(s; \mathbf{W}^-, \mathbf{W}^+) ds, \quad (1.3.12)$$

where \mathbf{W}^- and \mathbf{W}^+ are the left and right limits of the solution. In the particular case of a stationary discontinuity ($v = 0$), using (1.3.3) the Rankine-Hugoniot condition would read as

$$\int_0^1 \left(\mathbf{A}_\zeta(\Phi(s; \mathbf{U}^-, \mathbf{U}^+)) \frac{d\Phi^U}{ds}(s; \mathbf{U}^-, \mathbf{U}^+) - \mathbf{S}_\zeta(\Phi(s; \mathbf{U}^-, \mathbf{U}^+)) \frac{d\Phi^\sigma}{ds} \right) ds = 0.$$

Considering now a Roe linearization, we deduce that

$$\mathbf{A}_\zeta(\mathbf{U}^-, \mathbf{U}^+)(\mathbf{U}^+ - \mathbf{U}^-) = \mathbf{S}_\zeta(\mathbf{U}^-, \mathbf{U}^+)(\sigma^+ - \sigma^-).$$

Finally, assuming that the matrix $\mathbf{A}_\zeta(\mathbf{U}^-, \mathbf{U}^+)$ is invertible, we get the following form of the Rankine-Hugoniot condition:

$$\mathbf{U}^+ - \mathbf{U}^- = \mathbf{A}_\zeta(\mathbf{U}^-, \mathbf{U}^+)^{-1} \mathbf{S}_\zeta(\mathbf{U}^-, \mathbf{U}^+)(\sigma^+ - \sigma^-), \quad (1.3.13)$$

that will be thoroughly applied in Chapter 4. Similarly to theorem 1, a piecewise regular function \mathbf{W} is a weak solution of (1.3.2) if and only if the two following conditions are satisfied:

- (i) \mathbf{W} is a classical solution where it is smooth.
- (ii) At every point of discontinuity \mathbf{W} satisfies the jump condition (1.3.12).

Finally, as it happens for systems of conservation laws, in order to have uniqueness of solution an entropy condition has to be considered.

1.4 Path-conservative schemes

We are interested in the numerical solution of system (1.3.2) by means of finite volume methods on Cartesian meshes. Therefore, the spatial domain is divided into rectangular cells $C_{ij} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$ of size $\Delta x \times \Delta y$ and center (x_i, y_j) , and the time step is denoted as $t^n = n\Delta t$. As it is usual, \mathbf{W}_{ij}^n will denote the average of the solution at a given cell C_{ij} , at time t^n . In particular, we are interested in *path-conservative* schemes ([160]) of the form

$$\mathbf{W}_{ij}^{n+1} = \mathbf{W}_{ij}^n - \frac{\Delta t}{\Delta x} (\mathbf{D}_{i-1/2,j}^+ + \mathbf{D}_{i+1/2,j}^-) - \frac{\Delta t}{\Delta y} (\mathbf{D}_{i,j-1/2}^+ + \mathbf{D}_{i,j+1/2}^-), \quad (1.4.1)$$

where $\mathbf{D}_{i+1/2,j}^\pm = \mathbf{D}^\pm(\mathbf{W}_{ij}^n, \mathbf{W}_{i+1,j}^n)$, being \mathbf{D}^\pm continuous functions satisfying

$$\mathbf{D}^\pm(\mathbf{W}, \mathbf{W}) = 0, \quad \forall \mathbf{W} \in \Omega, \quad (1.4.2)$$

$$\mathbf{D}^-(\mathbf{W}_{ij}^n, \mathbf{W}_{i+1,j}^n) + \mathbf{D}^+(\mathbf{W}_{ij}^n, \mathbf{W}_{i+1,j}^n) = \int_0^1 \mathcal{A}_\zeta(\Phi_{ij,(i+1)j}) \frac{d}{ds} \Phi_{ij,(i+1)j} ds, \quad (1.4.3)$$

and

$$\mathbf{D}^-(\mathbf{W}_{ij}^n, \mathbf{W}_{i,j+1}^n) + \mathbf{D}^+(\mathbf{W}_{ij}^n, \mathbf{W}_{i,j+1}^n) = \int_0^1 \mathcal{A}_\zeta(\Phi_{ij,i(j+1)}) \frac{d}{ds} \Phi_{ij,i(j+1)} ds. \quad (1.4.4)$$

A remark about notation is in order. Eventually, we will only be interested in the component \mathbf{U} of the solution; for the sake of simplicity, this component will be also denoted as \mathbf{D} . Therefore, the corresponding numerical scheme would read as

$$\mathbf{U}_{ij}^{n+1} = \mathbf{U}_{ij}^n - \frac{\Delta t}{\Delta x} (\mathbf{D}_{i-1/2,j}^+ + \mathbf{D}_{i+1/2,j}^-) - \frac{\Delta t}{\Delta y} (\mathbf{D}_{i,j-1/2}^+ + \mathbf{D}_{i,j+1/2}^-).$$

Remark 1. *The definition of path-conservative scheme generalizes the notion of conservative method for a system of conservation laws, in the following sense: if $\mathcal{A}_x(\mathbf{W})$ is the Jacobian of the flux function \mathbf{F} , then (1.4.3) reduces to*

$$\mathbf{D}^-(\mathbf{W}_0, \mathbf{W}_1) + \mathbf{D}^+(\mathbf{W}_0, \mathbf{W}_1) = \mathbf{F}(\mathbf{W}_1) - \mathbf{F}(\mathbf{W}_0)$$

for every $\mathbf{W}_0, \mathbf{W}_1 \in \mathcal{O}$. Therefore, we can define

$$\mathbf{F}(\mathbf{W}_0, \mathbf{W}_1) = \mathbf{D}^-(\mathbf{W}_0, \mathbf{W}_1) + \mathbf{F}(\mathbf{W}_0) \quad (1.4.5)$$

or, equivalently,

$$\mathbf{F}(\mathbf{W}_0, \mathbf{W}_1) = \mathbf{F}(\mathbf{W}_1) - \mathbf{D}^+(\mathbf{W}_0, \mathbf{W}_1). \quad (1.4.6)$$

It follows from (1.4.2) that $\mathbf{F}(\mathbf{W}, \mathbf{W}) = \mathbf{F}(\mathbf{W})$ and the numerical flux is consistent with the physical flux. Furthermore, using (1.4.5)-(1.4.6) in (1.4.1), and following the same guidelines for the y -direction, it can be easily checked that the numerical method can be rewritten as the conservative method (1.2.2), with $\mathbf{F}_{i+1/2,j} = \mathbf{F}(\mathbf{W}_{ij}^n, \mathbf{W}_{i+1,j}^n)$ and $\mathbf{G}_{i,j+1/2} = \mathbf{G}(\mathbf{W}_{ij}^n, \mathbf{W}_{i,j+1}^n)$. Due to this, if a subsystem of (1.3.2) is a conservation law (as it happens with the mass equation in the one- and two-layer shallow water system), a path-conservative method will be conservative for this subsystem.

Choice of paths

As it was first pointed out in [56], path-conservative schemes may not converge to the physically relevant weak solution when the meshgrid is refined, as is the case of the Euler equations in primitive variables [5]. However, for several interesting applications as multi-layer shallow water systems with bottom topography ([52, 43, 124, 55]), Baer-Nunziato model ([13, 80, 84, 154]), non-hydrostatic model for dispersive waves ([92, 91, 90]), Saint Venant-Exner system ([42]), turbidity currents ([150, 45]), Ripa model ([175]), two-modes shallow-water system ([39]), Savage-Hutter models ([94]), Bingham shallow-water system ([95]), blood flow ([151]), shear shallow water flows ([64]), amongs others, path-conservative schemes have been successfully applied and their use seems to be justified. The goal of this paragraph is to shed some light on this particular.

The difficulty relies in that the generalized jump relations (1.3.12) for the nonconservative case depend not only on the states that are linked in phase space, but also on the path that connects them. We consider a system of the form (1.3.2), which is the vanishing diffusion limit of a parabolic regularization of the form

$$\partial_t \mathbf{W} + \mathcal{A}_x(\mathbf{W}) \partial_x \mathbf{W} + \mathcal{A}_y(\mathbf{W}) \partial_y \mathbf{W} = \varepsilon (\Theta_x \partial_{xx} \mathbf{W} + \Theta_y \partial_{yy} \mathbf{W}),$$

with an elliptic second-order viscous term, where Θ_ζ are positive definite matrices. In this case, the correct jump conditions should be consistent with the viscous profile, that are the traveling waves of the regularized system (see [130]). The jump conditions obtained for this system depend on the viscous profile (and consequently of the Θ_ζ matrices) and, after a reparametrization, the good choice for the path connecting neighboring states would be a viscous profile. The consequence of this observation is that the limits of the numerical solutions satisfy a jump condition which is related to the numerical viscosity of the method and not to the physically relevant one.

The path conservative framework ([160]) allows to extend to nonconservative systems many well-known conservative numerical methods in a consistent way. Moreover, they can be easily extended to: high-order accuracy using reconstruction operators ([52]); central-upwind schemes ([60]); discontinuous Galerkin methods ([86, 83]); and ADER methods ([78]). The experience shows that path-conservative schemes converge with the expected order of accuracy and are stable with the same CFL condition as their conservative counterpart. Next, these schemes have a importante advantage: they are automatically exactly well-balanced for water at rest solutions, which is a very important condition for shallow water flows. Also it is worth noting that the above mentioned failures happen for any numerical scheme in which the small scale effects are not controlled ([132]), regardless if it is path-conservative or not. Only viscosity-free

schemes as the random-choice method of Glimm ([106, 190]) or front-tracking methods are able to overcome this difficulty.

A promising direction to cover this drawback in the path-conservative framework are the so-called entropy-stable path-conservative schemes, developed in [47, 116], in which the schemes satisfy a discrete entropy inequality. A modified Roe-type path-conservative scheme has been proposed in [63], which has been able to control the problem for gas dynamic equations; see also [62]. For shallow water-type models and the nonconservative version of MHD equations, as well as many other systems found in practical situations, path-conservative schemes provide excellent results. Further investigations on this topic are the subject of future research.

The computation of viscous profiles may be a very difficult task and there are no guarantee of right convergence to the weak solution. Conversely, as it was demonstrated in [45], the corresponding jump conditions of the simple choice of segments paths (1.3.7) give a third-order approximation of the physically correct ones, which makes this methods computationally efficient ([7, 9, 11, 8, 54, 159]). In addition, these difficulties regarding convergence to the right weak solutions are not always present. For instance, in systems of balance laws with continuous σ there is no ambiguity in the definition of weak solutions, and all the numerical methods discussed in the next sections converge to them.

Godunov scheme

In the same manner that conservation laws (see Section 1.2), Godunov's method is based on the exact solution of the associated local Riemann problem (1.3.2). Once a family of paths has been chosen, it can be shown that the method can be written in the form (1.4.1) with

$$\begin{aligned} D(\mathbf{W}_{ij}^n, \mathbf{W}_{i+1,j}^n)^- &= \int_0^1 \mathcal{A}_x(\Phi(s; \mathbf{W}_{ij}^n, \mathbf{W}_0^-)) \frac{d\Phi}{ds}(s; \mathbf{W}_{ij}^n, \mathbf{W}_0^-) ds, \\ D(\mathbf{W}_{ij}^n, \mathbf{W}_{i+1,j}^n)^+ &= \int_0^1 \mathcal{A}_x(\Phi(s; \mathbf{W}_0^+, \mathbf{W}_{i+1,j}^n)) \frac{d\Phi}{ds}(s; \mathbf{W}_0^+, \mathbf{W}_{i+1,j}^n) ds, \end{aligned} \quad (1.4.7)$$

where

$$\mathbf{W}_0^\pm = \lim_{\xi \rightarrow 0^\pm} \mathcal{R}_x(\xi; \mathbf{W}_{ij}^n, \mathbf{W}_{i+1,j}^n) \equiv \mathbf{W}_{i+1/2,j}^{n,\pm},$$

with similar definitions for the y -direction.

Approximate Riemann solvers

Let us consider the following definition:

Definition 3. [57] *Let Φ be a family of paths in \mathcal{O} . Let us suppose that, for every pair of states \mathbf{U}_0 and \mathbf{U}_1 in \mathcal{O} , a finite number $s \geq 1$ of speeds*

$$\sigma_0 = -\infty < \sigma_1 < \dots < \sigma_s < \sigma_{s+1} = +\infty \quad (1.4.8)$$

and $s - 1$ intermediate states

$$\mathcal{W}_0 = \mathbf{W}_0, \mathcal{W}_1, \dots, \mathcal{W}_{s-1}, \mathcal{W}_s = \mathbf{W}_1, \quad (1.4.9)$$

are chosen. The function $\mathcal{R}: \mathbb{R} \times \mathcal{O} \times \mathcal{O} \rightarrow \mathcal{O}$ given by

$$\mathcal{R}(\xi; \mathbf{W}_0, \mathbf{W}_1) = \mathcal{W}_k \quad \text{if } \sigma_k < \xi < \sigma_{k+1}, \quad (1.4.10)$$

is said to be a Φ -approximate Riemann solver for the one-dimensional nonconservative system $\partial_t \mathbf{W} + \mathcal{A}_x(\mathbf{W}) \partial_x \mathbf{W} = 0$ it satisfies

$$\sum_{k=0}^{s-1} \sigma_{k+1} (\mathcal{W}_{k+1} - \mathcal{W}_k) = \int_0^1 \mathcal{A}_x(\Phi_{0,1}) \frac{d\Phi_{0,1}}{ds} ds. \quad (1.4.11)$$

A two-dimensional extension to 2D Riemann solvers based on projected one-dimensional Riemann solvers in the coordinate directions is straightforward. Any Φ -approximate Riemann solver for (1.3.2) leads to a numerical method which is path-conservative, where

$$\begin{aligned} D^-(\mathbf{W}_0, \mathbf{W}_1) &= - \int_{-\infty}^0 (\mathcal{R}_x(\xi; \mathbf{W}_0, \mathbf{W}_1) - \mathbf{W}_0) d\xi, \\ D^+(\mathbf{W}_0, \mathbf{W}_1) &= - \int_0^{\infty} (\mathcal{R}_x(\xi; \mathbf{W}_0, \mathbf{W}_1) - \mathbf{W}_1) d\xi, \end{aligned} \quad (1.4.12)$$

with similar definitions for the y -direction. Analogously to the conservative case, the easiest case of Φ -approximate Riemann solver comes when $s = 2$, where we have the HLL Riemann solver. The solution is very similar to the conservative case (see (1.2.10)-(1.2.11) and Figure 1.2(b)), where now in the star region it appears

$$\mathbf{W}^* = \frac{1}{\sigma_2 - \sigma_1} \left(\sigma_2 \mathbf{W}_1 - \sigma_1 \mathbf{W}_0 - \int_0^1 \mathcal{A}_x(\Phi_{0,1}) \partial_s \Phi_{0,1} \right).$$

The method of Roe is based on the Roe linearization discussed in Section 1.3. Once the linearization is chosen, the corresponding Roe scheme can be written in the form (1.4.1) with

$$\begin{aligned} D^-(\mathbf{W}_0, \mathbf{W}_1) &= \mathcal{A}_\zeta^-(\mathbf{W}_0, \mathbf{W}_1) (\mathbf{W}_1 - \mathbf{W}_0), \\ D^+(\mathbf{W}_0, \mathbf{W}_1) &= \mathcal{A}_\zeta^+(\mathbf{W}_0, \mathbf{W}_1) (\mathbf{W}_1 - \mathbf{W}_0), \end{aligned}$$

where

$$\mathcal{A}_\zeta^\pm(\mathbf{W}_0, \mathbf{W}_1) = R_\zeta(\mathbf{W}_0, \mathbf{W}_1) \Lambda_\zeta^\pm(\mathbf{W}_0, \mathbf{W}_1) R_\zeta^{-1}(\mathbf{W}_0, \mathbf{W}_1), \quad \zeta = x, y,$$

being $\Lambda_\zeta^\pm(\mathbf{W}_0, \mathbf{W}_1)$ the diagonal matrix in the ζ -direction whose coefficients are the positive/negative part of the eigenvalues of $\mathcal{A}_\zeta(\mathbf{W}_0, \mathbf{W}_1)$, $\lambda_\zeta(\mathbf{W}_0, \mathbf{W}_1)$, and $R_\zeta(\mathbf{W}_0, \mathbf{W}_1)$ is a $N \times N$ matrix whose columns are associated eigenvectors. The following identity holds:

$$\mathcal{A}_\zeta^\pm(\mathbf{W}_0, \mathbf{W}_1) = \frac{1}{2} (\mathcal{A}_\zeta(\mathbf{W}_0, \mathbf{W}_1) \pm |\mathcal{A}_\zeta(\mathbf{W}_0, \mathbf{W}_1)|), \quad (1.4.13)$$

where

$$|\mathcal{A}_\zeta(\mathbf{W}_0, \mathbf{W}_1)| = R_\zeta(\mathbf{W}_0, \mathbf{W}_1) |\Lambda_\zeta(\mathbf{W}_0, \mathbf{W}_1)| R_\zeta^{-1}(\mathbf{W}_0, \mathbf{W}_1).$$

Here $|\Lambda_\zeta(\mathbf{W}_0, \mathbf{W}_1)|$ is the diagonal matrix in the ζ -direction whose coefficients are the absolute value of the eigenvalues of $\mathcal{A}_\zeta(\mathbf{W}_0, \mathbf{W}_1)$. As it happens for conservation laws, an entropy-fix technique is needed in order to avoid non-entropy solutions in the vicinity of sonic points.

1.5 AVM-type solvers

For the sake of completeness, we give in this section a brief survey on AVM solvers, for which it will be enough to focus on one-dimensional systems. For clarity, we consider initially a system of conservation laws

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

whose numerical solution by means of a finite volume method is of the form

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x} (\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}),$$

with numerical flux given by

$$\mathbf{F}_{i+1/2} = \frac{\mathbf{F}(\mathbf{U}_i) + \mathbf{F}(\mathbf{U}_{i+1})}{2} - \frac{1}{2} \mathbf{Q}_{i+1/2} (\mathbf{U}_{i+1} - \mathbf{U}_i). \quad (1.5.1)$$

Here $\mathbf{Q}_{i+1/2}$ denotes the numerical *viscosity matrix*, which determines the numerical diffusion of the scheme. It is worth noticing that Roe's method ([168]) can be written in the form (1.5.1) by considering the viscosity matrix $\mathbf{Q}_{i+1/2} = |\mathbf{A}_{i+1/2}|$, being $\mathbf{A}_{i+1/2}$ a Roe matrix for the system. For nonconservative systems of the form (1.3.2), the idea is to replace (1.4.13) by

$$\hat{\mathbf{A}}_{\zeta}^{\pm}(\mathbf{W}_0, \mathbf{W}_1) = \frac{1}{2} (\mathbf{A}_{\zeta}(\mathbf{W}_0, \mathbf{W}_1) \pm \mathbf{Q}_{\zeta}(\mathbf{W}_0, \mathbf{W}_1)),$$

where

$$\mathbf{Q}_{\zeta}(\mathbf{W}_0, \mathbf{W}_1) = \left[\begin{array}{c|c} \mathbf{Q}_{\zeta}(\mathbf{W}_0, \mathbf{W}_1) & \mathbf{Q}_{\zeta}(\mathbf{W}_0, \mathbf{W}_1) \mathbf{A}_{\zeta}(\mathbf{W}_0, \mathbf{W}_1)^{-1} \mathbf{S}_{\zeta}(\mathbf{W}_0, \mathbf{W}_1) \\ \hline 0 & 0 \end{array} \right]. \quad (1.5.2)$$

For one-dimensional nonconservative systems of the form

$$\partial_t \mathbf{U} + \mathcal{A}(\mathbf{U}) \partial_x \mathbf{U} = 0,$$

whose numerical solution by means of a finite volume method is of the form

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x} (\mathbf{D}_{i+1/2}^- + \mathbf{D}_{i-1/2}^+),$$

the numerical fluctuations are given by

$$\begin{aligned} \mathbf{D}_{i+1/2}^{\pm}(\mathbf{W}_i, \mathbf{W}_{i+1}) &\equiv \mathbf{D}_{i+1/2}^{\pm}(\mathbf{U}_i, \sigma_i, \mathbf{U}_{i+1}, \sigma_{i+1}) = \\ &\frac{1}{2} (\mathbf{F}(\mathbf{U}_{i+1}) - \mathbf{F}(\mathbf{U}_i) + \mathcal{B}_{i+1/2}(\mathbf{U}_{i+1} - \mathbf{U}_i) - \mathbf{S}_{i+1/2}(\sigma_{i+1} - \sigma_i)) \\ &\pm \frac{1}{2} \mathbf{Q}_{i+1/2} (\mathbf{U}_{i+1} - \mathbf{U}_i - \mathbf{A}_{i+1/2}^{-1} \mathbf{S}_{i+1/2}(\sigma_{i+1} - \sigma_i)). \end{aligned} \quad (1.5.3)$$

The idea of PVM (Polynomial Viscosity Matrix) solvers introduced in [41] (see also the previous work by Degond et al. [70]) consists in approximating $|\mathbf{A}_{i+1/2}|$ by means of a suitable polynomial evaluation of $\mathbf{A}_{i+1/2}$. More precisely, let $p(x)$ be a polynomial approximation of $|x|$

in the interval $[-1, 1]$, and let $\lambda_{i+1/2, \max}$ be the eigenvalue of $\mathbf{A}_{i+1/2}$ with maximum modulus (or an upper bound of it). Then the viscosity matrix of the PVM method associated to $p(x)$ is given by

$$\mathbf{Q}_{i+1/2} = |\lambda_{i+1/2, \max}| p(|\lambda_{i+1/2, \max}|^{-1} \mathbf{A}_{i+1/2}). \quad (1.5.4)$$

Notice that the best $p(x)$ approaches $|x|$, the closer the behaviour of the associated PVM scheme will be to that of Roe's method, whose viscosity matrix is precisely $|\mathbf{A}_{i+1/2}|$. A fundamental issue is that the spectral decomposition of $\mathbf{A}_{i+1/2}$ is not needed in the construction of a PVM method, but only a bound on its spectral radius. This feature makes PVM methods greatly efficient and applicable to systems in which the eigenstructure is not known or it is difficult to obtain. In the cases in which a Roe matrix is not available or it is costly to compute, $\mathbf{A}_{i+1/2}$ can be taken as the Jacobian matrix of the system evaluated at some average state. Several well-known schemes in the literature belong to the family of PVM methods: Lax-Friedrichs, Rusanov, Lax-Wendroff, HLL, FORCE, GFORCE, MUSTA, Roe, etc. (see [41] and the references therein). The numerical scheme introduced in [70] and the Krylov-Riemann solver introduced in [195] can be viewed as particular cases of PVM schemes as well. We show some of these schemes written in PVM form in Table (1.1).

Table 1.1: Coefficients of some solvers written in PVM form.

Scheme	Polynomial	Coefficients
Lax-Friedrichs, Rusanov and Local Lax-Friedrichs	$P_0(x) = \alpha_0$	$\alpha_0 \in \{S_{LF}, S_{Rus}, S_{LF}^{mod}\}$ where $S_{LF} = \frac{\Delta x}{\Delta t}$, $S_{Rus} = \max_j \lambda_j^{i+1/2} $, $S_{LF}^{mod} = \delta \frac{\Delta x}{\Delta t}$
Lax-Wendroff	$P_2(x) = \alpha_2 x^2$	$\alpha_2 = \frac{\Delta t}{\Delta x}$
HLL	$P_1(x) = \alpha_0 + \alpha_1 x$	$\alpha_0 = \frac{\lambda_N^{i+1/2} \lambda_1^{i+1/2} - \lambda_1^{i+1/2} \lambda_N^{i+1/2} }{\lambda_N^{i+1/2} - \lambda_1^{i+1/2}}$ $\alpha_1 = \frac{ \lambda_N^{i+1/2} - \lambda_1^{i+1/2} }{\lambda_N^{i+1/2} - \lambda_1^{i+1/2}}$
FORCE	$P_2(x) = \alpha_0 + \alpha_2 x^2$	$\alpha_0 = \frac{\Delta x}{2\Delta t}$, $\alpha_2 = \frac{\Delta t}{2\Delta x}$

It is known that rational functions may provide much more precise approximations to $|x|$ than polynomials. The RVM methods introduced in [49] take advantage of this fact, following the PVM idea but using rational functions as basis instead of polynomials. In what follows we will encompass both kind of solvers, PVM and RVM, under the common term AVM (Approximate Viscosity Matrix) solvers. AVM solvers have also been used in [50] to construct efficient versions of the Osher-Solomon method, following the ideas in [85]. Recently, in [51] some types of AVM solvers have been implemented in Jacobian-free form; this is particularly interesting when solving complex systems as, for example, relativistic MHD.

Finally, we must remark that the stability of an AVM scheme strongly depends on the properties of the underlying function $f(x)$. In particular, it must verify the *stability condition*

$$|x| \leq f(x) \leq 1, \quad \forall x \in [-1, 1]. \quad (1.5.5)$$

With respect to the choice of solvers, in [49] the authors proposed a family of PVM methods based on Chebyshev polynomials, which provide optimal approximations to the absolute value function, and could also be implemented in Jacobian-free form. In the same paper, RVM solvers based on Newman rational functions [153] were found to be the more efficient choice: they provided similar results as Roe's method for complex problems in MHD and multilayer shallow water systems, but with a much smaller computational cost. Despite their efficiency, a drawback of Chebyshev and Newman AVM solvers is that they do not satisfy the stability condition (1.5.5) strictly, so a slight modification has to be performed which may perturb the computation of the external speeds of propagation. On the other hand, in [51] a new family of internal polynomial approximations to $|x|$ verifying (1.5.5) was proposed. Most recently, in [104] a class of rational Padé approximations was considered, based on approximations of the sign matrix function ([119]), which also satisfy (1.5.5). For the sake of completeness, we briefly describe them in what follows.

The following optimal uniform approximation to $|x|$ in $[-1, 1]$ holds:

$$|x| = \frac{2}{\pi} + \sum_{k=1}^{\infty} \frac{4}{\pi} \frac{(-1)^{k+1}}{(2k-1)(2k+1)} T_{2k}(x), \quad x \in [-1, 1], \quad (1.5.6)$$

where the Chebyshev polynomials of even degree $T_{2k}(x)$ are recursively defined as

$$T_0(x) = 1, \quad T_2(x) = 2x^2 - 1, \quad T_{2k}(x) = 2T_2(x)T_{2k-2}(x) - T_{2k-4}(x).$$

Then, we can consider the following approximations to $|x|$, obtained by truncation of (1.5.6):

$$\tau_{2p}(x) = \frac{2}{\pi} + \sum_{k=1}^p \frac{4}{\pi} \frac{(-1)^{k+1}}{(2k-1)(2k+1)} T_{2k}(x), \quad x \in [-1, 1].$$

Then, the viscosity matrix of a PVM method based on a Chebyshev approximation is defined as

$$\mathbf{Q}_{i+1/2} = |\lambda_{i+1/2, \max}| \tau_{2p}(|\lambda_{i+1/2, \max}|^{-1} \mathbf{A}_{i+1/2}) \approx |\mathbf{A}_{i+1/2}|.$$

Notice that $\tau_{2p}(x)$ do not satisfy the stability condition (1.5.5) strictly. This drawback can be avoided by using $\tau_{2p}^\varepsilon(x) = \tau_{2p}(x) + \varepsilon$ with

$$\varepsilon = \max\{|\tau_{2p}(x^*) - |x^*|| : \tau_{2p}'(x^*) = 1, x^* \in [0, 1]\}. \quad (1.5.7)$$

Figure 1.3 (top left) shows a comparison between the polynomials $\tau_8(x)$ and $\tau_8^\varepsilon(x)$. Also, Figure 1.3 (top right) shows the polynomials $\tau_{2p}^\varepsilon(x)$ for $p = 2, 3, 4$.

The Newman rational function ([153]) associated to a set of nodes $X = \{0 < x_1 < \dots < x_r \leq 1\}$ defined in $(0, 1]$, is given by

$$R_r(x) = x \frac{p(x) - p(-x)}{p(x) + p(-x)}, \quad (1.5.8)$$

where

$$p(x) = \prod_{k=1}^r (x + x_k), \quad r \geq 4.$$

The function (1.5.8) interpolates $|x|$ at the points $\{-x_r, \dots, -x_1, 0, x_1, \dots, x_r\}$ (see [49] for more details). Similarly to the Chebyshev polynomials, the stability condition (1.5.5) is not fully satisfied, so a modification $R_r^\varepsilon = R_r(x) + \varepsilon$ is required, with ε computed as in (1.5.7). Figure 1.3 (bottom left) shows a comparison between $R_4(x)$ and $R_4^\varepsilon(x)$. Also, Figure 1.3 (down right) shows a zoom in $[-0.1, 0.1]$ (outside of this interval the difference is not noticeable) of the Newman rational approximation for $r = 4, 8, 12$.

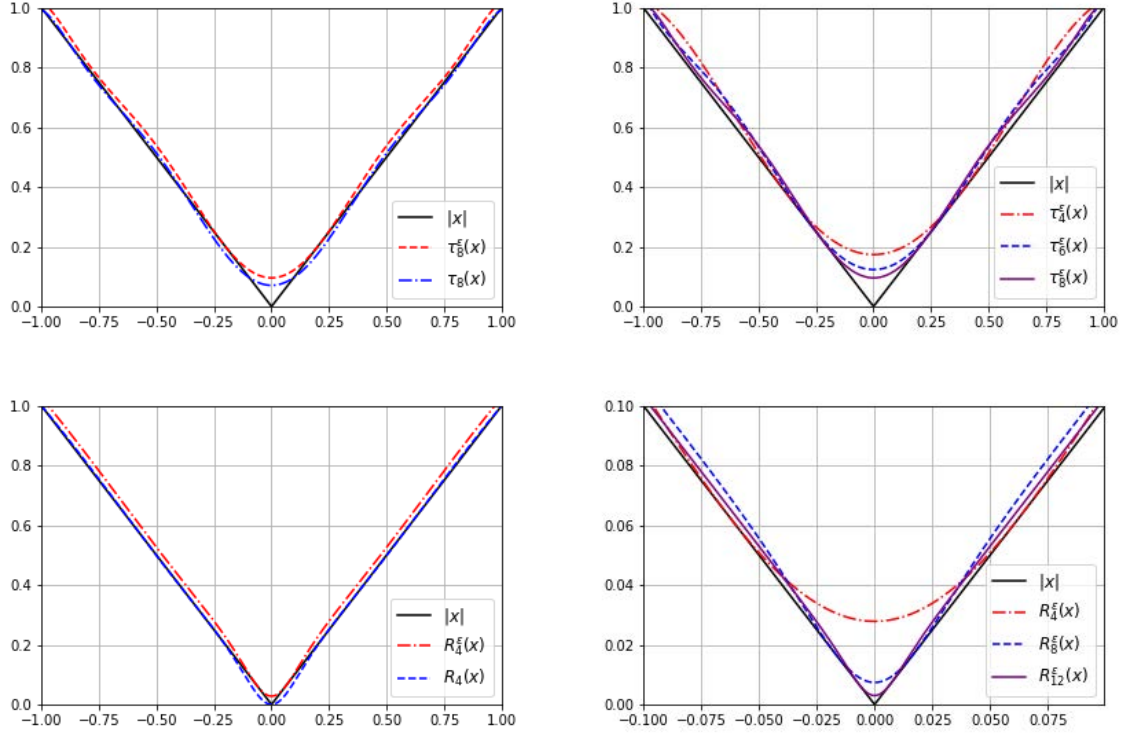


Figure 1.3: Top: Comparison of Chebyshev polynomials $\tau_8(x)$ and $\tau_8^\varepsilon(x)$; Chebyshev polynomials $\tau_{2p}^\varepsilon(x)$ for $p = 2, 3, 4$. Bottom: Comparison of Newman rational functions $R_4(x)$ and $R_4^\varepsilon(x)$; zoom near the origin of the Newman rational functions $R_r^\varepsilon(x)$ for $r = 4, 8, 12$.

In order to ease the implementation of the corresponding schemes, in Appendix A are shown the coefficients for some approximations based on Chebyshev polynomials and Newman rational functions.

The internal polynomial approximations are recursively defined as

$$p_0(x) \equiv 1,$$

$$p_{n+1}(x) = \frac{1}{2}(2p_n(x) - p_n(x)^2 + x^2), \quad n = 0, 1, 2, \dots$$

The following straightforward properties hold:

1. $p_n(x)$ is even and of degree 2^n .

2. $|x| < p_n(x) < 1$ for $x \in (-1, 1)$, and $p_n(\pm 1) = 1$.
3. $p'_n(1) = 1$ and $p'_n(-1) = -1$.
4. $\min_{-1 \leq x \leq 1} p_n(x) = p_n(0) > 0$.
5. The sequence $\{p_n(x)\}_{n \in \mathbb{N}}$ converges uniformly to $|x|$.

Regarding efficiency issues, we have found that a direct implementation of $p_n(x)$ is a better option than using its recursive form. In Table 1.2 are given the coefficients of the polynomials

$$p_n(x) = \alpha_0 x^{2n} + \alpha_1 x^{2n-1} + \cdots + \alpha_{n-1} x^2 + \alpha_n,$$

for $n = 1, 2, 3, 4$. Figure 1.4 (top left) shows the polynomials $p_n(x)$, for $n = 1, 2, 3, 4$.

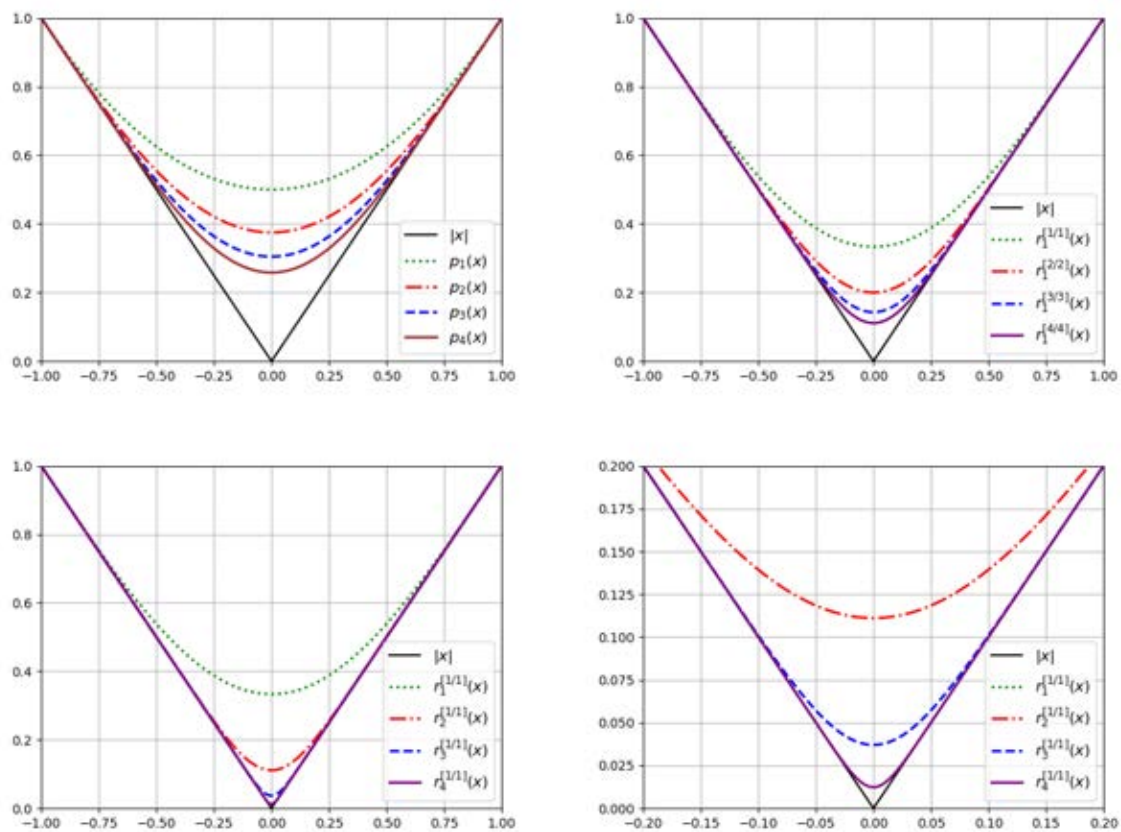


Figure 1.4: Top: Internal polynomials $p_n(x)$ for $n = 1, 2, 3, 4$, and Padé rational approximants $r_1^{[m/k]}(x)$ for $m = k = 1, 2, 3, 4$. Bottom: Padé approximants $r_n^{[1/1]}(x)$ for $n = 1, 2, 3, 4$, and zoom near the origin.

Table 1.2: Coefficients of the internal approximations $p_n(x)$, for $n = 1, 2, 3, 4$.

n	α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8
1	$\frac{1}{2}$	$\frac{1}{2}$							
2	$-\frac{1}{8}$	$\frac{3}{4}$	$\frac{3}{8}$						
3	$-\frac{1}{128}$	$\frac{3}{32}$	$-\frac{23}{64}$	$\frac{31}{32}$	$\frac{39}{128}$				
4	$-\frac{1}{32768}$	$\frac{3}{4096}$	$-\frac{59}{8192}$	$\frac{169}{4096}$	$-\frac{2635}{16384}$	$\frac{1693}{4096}$	$-\frac{5891}{8192}$	$\frac{4807}{4096}$	$\frac{8463}{32768}$

On the other hand, for given integers $k, m \geq 0$, we consider the polynomials

$$P_{km}(\xi) = \sum_{n=0}^k \frac{(\frac{1}{2})_n (\frac{1}{2} - m)_m (n - k - m)_m}{n! (-k - m)_m (n + \frac{1}{2} - m)_m} \xi^n$$

and

$$Q_{km}(\xi) = \sum_{n=0}^m \frac{(-m)_n (-\frac{1}{2} - k)_n}{n! (-k - m)_n} \xi^n,$$

where we have used the notation $(a)_n = a(a+1)\dots(a+n-1)$ and $(a)_0 = 1$. Then, we follow [119] to build a sequence of Padé approximations of the form

$$x_0 = 1, \\ x_{n+1} = x_n \frac{Q_{km}(1 - x^2/x_n^2)}{P_{km}(1 - x^2/x_n^2)}, \quad n = 0, 1, 2, \dots$$

which converges to $|x|$, for any given $x \in [-1, 1]$. Thus, the Padé approximants of order $[m/k]$ are defined recursively as

$$r_0^{[m/k]}(x) \equiv 1, \\ r_{n+1}^{[m/k]}(x) = r_n(x) \frac{Q_{km}(1 - x^2/r_n(x)^2)}{P_{km}(1 - x^2/r_n(x)^2)}, \quad n = 0, 1, 2, \dots$$

(the superscript $[m/k]$ will be dropped unless necessary). It is not difficult to prove that each $r_n(x)$ is even and verifies the properties 2-5 listed before for $p_n(x)$. Figure 1.4 (top right) shows the functions $r_1^{[m/k]}(x)$ for $m = k = 1, 2, 3, 4$.

For the ease of coding, we give in Table 1.3 the coefficients of the rational function $r_{n+1}(x)$ written in the form

$$r_{n+1}(x) = r_n(x) \frac{\alpha_0 r_n(x)^{2m} + \alpha_1 r_n(x)^{2(m-1)} x^2 + \dots + \alpha_{m-1} r_n(x)^2 x^{2(m-1)} + \alpha_m x^{2m}}{\beta_0 r_n(x)^{2k} + \beta_1 r_n(x)^{2(k-1)} x^2 + \dots + \beta_{k-1} r_n(x)^2 x^{2(k-1)} + \beta_k x^{2k}},$$

for $0 \leq k, m \leq 4$. For example, the family of Padé approximants of order $[1/1]$ would be given by

$$r_{n+1}(x) = r_n(x) \frac{r_n(x)^2 + 3x^2}{3r_n(x)^2 + x^2}, \quad n = 0, 1, 2, \dots$$

while those of order $[2/2]$ would be

$$r_{n+1}(x) = r_n(x) \frac{r_n(x)^4 + 10r_n(x)^2x^2 + 5x^4}{5r_n(x)^4 + 10r_n(x)^2x^2 + x^4}, \quad n = 0, 1, 2, \dots$$

Figure 1.4 (bottom) depicts the approximants of order $[1/1]$ for $n = 1, 2, 3, 4$, together with a zoom near the origin to check the precision of the approximations.

Table 1.3: Coefficients of the rational function $r_1(x)$, for $0 \leq k, m \leq 4$.

	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$m = 0$	$\frac{[1]}{[1]}$	$\frac{[-2]}{[-3,1]}$	$\frac{[8]}{[15,-10,3]}$	$\frac{[-16]}{[-35,35,-21,-5]}$	$\frac{[128]}{[315,-420,378,-180,35]}$
$m = 1$	$\frac{[1,1]}{[2]}$	$\frac{[1,3]}{[3,1]}$	$\frac{[4,20]}{[15,10,-1]}$	$\frac{[8,56]}{[35,35,-7,1]}$	$\frac{[64,576]}{[315,420,-126,36,-5]}$
$m = 2$	$\frac{[3,6,-1]}{[8]}$	$\frac{[1,6,1]}{[4,4]}$	$\frac{[1,10,5]}{[5,10,1]}$	$\frac{[6,84,70]}{[35,105,21,-1]}$	$\frac{[16,288,336]}{[105,420,126,-12,1]}$
$m = 3$	$\frac{[5,15,-5,1]}{[16]}$	$\frac{[5,45,15,-1]}{[24,40]}$	$\frac{[1,15,15,1]}{[6,20,6]}$	$\frac{[1,21,35,7]}{[7,35,21,1]}$	$\frac{[8,216,504,168]}{[63,420,378,36,-1]}$
$m = 4$	$\frac{[35,140,-70,28,-5]}{[128]}$	$\frac{[35,420,210,-28,3]}{[192,448]}$	$\frac{[7,140,210,28,-1]}{[48,224,112]}$	$\frac{[1,28,70,28,1]}{[8,56,56,8]}$	$\frac{[1,36,126,84,9]}{[9,84,126,36,1]}$

Finally, we would like to remark that an additional advantage of the considered approximations $\tau_{2p}^\varepsilon(x)$, $R_r^\varepsilon(x)$, $p_n(x)$ and $r_n(x)$ is that they provide an automatic entropy fix to handle sonic flow, as none of them cross the origin: see [51, Sect. 4].

Remark 2. After extensive numerical investigation, our conclusion is that for relatively simple problems it is sufficient to consider a simple AVM solver, as for example HLL or the one based in the internal polynomial $p_1(x)$. However, when complex structures appear, the choice of more precise AVM solvers is a determinant factor in terms of efficiency (see [49]). This is even true when AVM solvers are used as building blocks in the design of high-order schemes, as it was demonstrated in [50].

1.6 High-order methods: general framework

Following closely [57, 52, 43, 40], we describe the general framework to construct high-order finite volume schemes for (1.3.2) based on reconstruction of states. The following notation will be used for a given finite volume C_{ij} : $|C_{ij}|$ represents its area; \mathcal{N}_k is the set of indexes k such that C_k is a neighbor of C_{ij} ; $E_{ij,k}$ is the common edge of two neighboring cells C_{ij} and C_k and $|E_{ij,k}|$ is its length; $\boldsymbol{\eta}_{ij,k} = (\eta_{ij,k,x}, \eta_{ij,k,y})$ is the normal unit vector at the edge $E_{ij,k}$ pointing towards the cell C_k . Denoting by τ the mesh, $N_{\tau_x} \times N_{\tau_y}$ is the number of cells, where N_{τ_α} is the number of gridpoints in the α -direction, $\alpha = x, y$. We will assume that the reconstructions are calculated as follows: given a family $\{\mathbf{W}_{ij}\}_{i,j=1}^{N_{\tau_x}, N_{\tau_y}}$ of cell values, first an approximation function is constructed at every cell C_{ij} , based on the values at some cells close to C_{ij} :

$$P_{ij}^t(\mathbf{x}) \equiv P_{ij}^t(\mathbf{x}; \{\mathbf{W}_{i\pm l, j\pm m}(t)\}_{l,m \in \mathfrak{B}_{ij}}),$$

for some set of indexes \mathfrak{B}_{ij} (the stencil). If, for instance, the reconstruction only depends on the neighboring cells to C_{ij} , then $\mathfrak{B}_{ij} = \mathcal{N}_k \cup \{ij\}$. Moreover, for the 9-point stencil around the C_{ij} , we have $k \in \{i \pm 1, j \pm 1\}$. These approximation functions are usually calculated by means of an interpolation or approximation procedure. Once these functions have been constructed, the reconstructions at $\gamma \in E_{ij,k}$ are defined as follows:

$$\mathbf{W}_{ij,k}^-(\gamma, t) = \lim_{\mathbf{x} \rightarrow \gamma} P_{ij}^t(\mathbf{x}), \quad \mathbf{W}_{ij,k}^+(\gamma, t) = \lim_{\mathbf{x} \rightarrow \gamma} P_k^t(\mathbf{x}). \quad (1.6.1)$$

The reconstruction operator must satisfy the following properties:

(P1) It is conservative, i.e., the following equality holds for any cell C_{ij} :

$$\mathbf{W}_{ij} = \frac{1}{|C_{ij}|} \int_{C_{ij}} P_{ij}^t(\mathbf{x}) d\mathbf{x}. \quad (1.6.2)$$

(P2) If the operator is applied to the cell averages $\{\mathbf{W}_{ij}\}$ for some smooth function $\mathbf{W}(\mathbf{x})$, then

$$\mathbf{W}_{ij,k}^\pm(\gamma, t) = \mathbf{W}(\gamma, t) + \mathcal{O}(\Delta x^p), \quad \forall \gamma \in E_{ij,k},$$

and

$$\mathbf{W}_{ij,k}^+(\gamma, t) - \mathbf{W}_{ij,k}^-(\gamma, t) = \mathcal{O}(\Delta x^{p+1}), \quad \forall \gamma \in E_{ij,k},$$

where p denotes the order of the reconstruction operator.

(P3) It is of order q in the interior of the cells, i.e., if the operator is applied to a sequence $\{\mathbf{W}_{ij}\}$ for some smooth function $\mathbf{W}(\mathbf{x})$, then

$$P_{ij}^t(\mathbf{x}) = \mathbf{W}(\mathbf{x}, t) + \mathcal{O}(\Delta x^q), \quad \forall \mathbf{x} \in \text{int}(C_{ij}). \quad (1.6.3)$$

(P4) Under the assumption of the previous property, the gradient of P_{ij}^t provides an approximation of order m of the gradient of \mathbf{W} :

$$\nabla P_{ij}^t(\mathbf{x}) = \nabla \mathbf{W}(\mathbf{x}, t) + \mathcal{O}(\Delta x^m), \quad \forall \mathbf{x} \in \text{int}(C_{ij}). \quad (1.6.4)$$

Many examples of reconstruction operators satisfying (P1) and (P2) can be found in the literature: ENO ([113]), WENO ([182, 184]), CWENO ([66]), PHM ([144, 179]), among others (see also [77, 79, 81, 82, 101]). Once the first-order method and the spatial reconstruction operator have been chosen, in order to evolve in time several methods can be used. For instance, discretizing only in space leads to a system of ODE that can be solved using the TVD Runge-Kutta methods introduced in [111, 112, 183]. Another options are the ADER approach developed by Toro, Dumbser and Titarev ([188, 78]), and the Compact Approximate Taylor (CAT) methods recently introduced in [36, 37]. In this dissertation we will employ the predictor-corrector MUSCL-Hancock procedure ([200, 201]) to achieve second-order accuracy in space and time.

For clarity, let us consider initially the conservative equation (1.2.1). Let $\overline{\mathbf{W}}_{ij}(t)$ denote the cell average of a regular solution \mathbf{W} of (1.2.1) over the cell C_{ij} at time t . Integrating (1.2.1) over the cell C_{ij} , the following equation can be easily obtained for the cell averages:

$$\overline{\mathbf{W}}_{ij}'(t) = -\frac{1}{|C_{ij}|} \left(\sum_{k \in \mathcal{N}_k} \int_{E_{ij,k}} \mathbf{F}_{\boldsymbol{\eta}_{ij,k}}(\mathbf{W}(\gamma, t)) d\gamma \right), \quad (1.6.5)$$

where $\mathbf{F}_{\boldsymbol{\eta}}(\cdot) = \mathbf{F}(\cdot)\eta_x + \mathbf{G}(\cdot)\eta_y$. The first-order method and the reconstructions are now used to approach the values of the fluxes at the edges:

$$\mathbf{W}_{ij}'(t) = -\frac{1}{|C_{ij}|} \sum_{k \in \mathcal{N}_k} \int_{E_{ij,k}} \mathbf{F}(\mathbf{W}_{ij,k}^-(\gamma, t), \mathbf{W}_{ij,k}^+(\gamma, t), \boldsymbol{\eta}_{ij,k}) d\gamma, \quad (1.6.6)$$

being $\mathbf{W}_{ij}(t)$ the approximation to $\overline{\mathbf{W}}_{ij}(t)$ provided by the scheme and $\mathbf{W}_{ij,k}^\pm(\gamma, t)$ the reconstruction at $\gamma \in E_{ij,k}$ corresponding to the family $\{\mathbf{W}_{ij}(t) : i = 1, \dots, N_{\tau_x}, j = 1, \dots, N_{\tau_y}\}$. It can be shown that (1.6.6) is an approximation of order p of (1.6.5). In practice, the integral terms in (1.6.6) are approached by means of a numerical quadrature formula of order $\bar{r} \geq p$. Applying the divergence theorem, we can rewrite (1.6.6) as

$$\begin{aligned} \mathbf{W}_{ij}'(t) = & -\frac{1}{|C_{ij}|} \sum_{k \in \mathcal{N}_k} \int_{E_{ij,k}} \mathbf{D}_{ij}^-(\mathbf{W}_{ij,k}^-(\gamma, t), \mathbf{W}_{ij,k}^+(\gamma, t), \boldsymbol{\eta}_{ij,k}) d\gamma \\ & - \frac{1}{|C_{ij}|} \int_{C_{ij}} \left(\mathbf{J}_x(P_{ij}^t(\mathbf{x})) \frac{\partial P_{ij}^t}{\partial x}(\mathbf{x}) + \mathbf{J}_y(P_{ij}^t(\mathbf{x})) \frac{\partial P_{ij}^t}{\partial y}(\mathbf{x}) \right) d\mathbf{x}, \end{aligned} \quad (1.6.7)$$

where (see Remark 1)

$$\mathbf{D}_{ij}^-(\mathbf{W}_{ij,k}^-(\gamma, t), \mathbf{W}_{ij,k}^+(\gamma, t), \boldsymbol{\eta}_{ij,k}) = \mathbf{F}(\mathbf{W}_{ij,k}^-(\gamma, t), \mathbf{W}_{ij,k}^+(\gamma, t), \boldsymbol{\eta}_{ij,k}) - \mathbf{F}_{\boldsymbol{\eta}_{ij,k}}(\mathbf{W}_{ij,k}^-(\gamma, t)).$$

According to [160], by replacing \mathbf{J}_ζ by \mathbf{A}_ζ , $\zeta = x, y$, and \mathbf{D}_{ij}^- by the fluctuation of a path-conservative numerical method, the expression (1.6.7) can be extended for a nonconservative system as

$$\begin{aligned} \mathbf{W}_{ij}'(t) = & -\frac{1}{|C_{ij}|} \sum_{k \in \mathcal{N}_k} \int_{E_{ij,k}} \mathbf{D}_{\Phi}^-(\mathbf{W}_{ij,k}^-(\gamma, t), \mathbf{W}_{ij,k}^+(\gamma, t), \boldsymbol{\eta}_{ij,k}) d\gamma \\ & - \frac{1}{|C_{ij}|} \int_{C_{ij}} \left(\mathbf{A}_x(P_{ij}^t(\mathbf{x})) \frac{\partial P_{ij}^t}{\partial x}(\mathbf{x}) + \mathbf{A}_y(P_{ij}^t(\mathbf{x})) \frac{\partial P_{ij}^t}{\partial y}(\mathbf{x}) \right) d\mathbf{x}, \end{aligned} \quad (1.6.8)$$

where \mathbf{D}_{Φ}^\pm define a path-conservative scheme for system (1.3.2), in such a way that $\mathbf{D}_{\Phi}^-(\mathbf{W}_0, \mathbf{W}_1, \boldsymbol{\eta})$ satisfies (1.4.2), (1.4.3) and (1.4.4), as well as $\mathbf{D}_{\Phi}^+(\mathbf{W}_0, \mathbf{W}_1, \boldsymbol{\eta}) = \mathbf{D}_{\Phi}^-(\mathbf{W}_0, \mathbf{W}_1, -\boldsymbol{\eta})$. In (1.6.8), the last integral is used to approximate the regular part of the weak solution, while the fluctuations are used to split the Dirac measures corresponding to the discontinuities at the interfaces.

Theorem 2. [43] *Let us assume that \mathbf{A}_ζ are of class C^2 with bounded derivatives and $\mathbf{D}_{\Phi}^-(\cdot, \cdot, \boldsymbol{\eta}_{ij,k})$ is bounded for all i, j and k . Let us also suppose that the reconstruction*

operator satisfies the hypothesis (P1)-(P4). Then (1.6.8) is an approximation of order at least $\alpha = \min(p, q, m)$ to the system (1.3.2) in the following sense:

$$\begin{aligned} & \frac{1}{|C_{ij}|} \sum_{k \in \mathcal{N}_k} \int_{E_{ij,k}} \mathbf{D}_{\Phi}^{-}(\mathbf{W}_{ij,k}^{-}(\gamma, t), \mathbf{W}_{ij,k}^{+}(\gamma, t), \boldsymbol{\eta}_{ij,k}) d\gamma \\ & + \frac{1}{|C_{ij}|} \int_{C_{ij}} \left(\mathcal{A}_x(P_{ij}^t(\mathbf{x})) \frac{\partial P_{ij}^t}{\partial x}(\mathbf{x}) + \mathcal{A}_y(P_{ij}^t(\mathbf{x})) \frac{\partial P_{ij}^t}{\partial y}(\mathbf{x}) \right) d\mathbf{x} \\ & = \frac{1}{|C_{ij}|} \int_{C_{ij}} (\mathcal{A}_x(\mathbf{W}(\mathbf{x}, t)) \mathbf{W}_x(\mathbf{x}, t) + \mathcal{A}_y(\mathbf{W}(\mathbf{x}, t)) \mathbf{W}_y(\mathbf{x}, t)) d\mathbf{x} + \mathcal{O}(\Delta x^\alpha) \end{aligned}$$

for every solution \mathbf{W} smooth enough, being $\mathbf{W}_{ij,k}^{\pm}(\gamma, t)$ the associated reconstructions and P_{ij}^t the approximation functions corresponding to the family

$$\overline{\mathbf{W}}_{ij}(t) = \frac{1}{|C_{ij}|} \int_{C_{ij}} \mathbf{W}(\mathbf{x}, t) d\mathbf{x}.$$

Now, taking into account the relation between systems (1.3.1) and (1.3.2), it is possible to rewrite (1.6.8) as follows:

$$\begin{aligned} \mathbf{U}'_{ij}(t) &= - \frac{1}{|C_{ij}|} \sum_{k \in \mathcal{N}_k} \int_{E_{ij,k}} \mathbf{D}_{\Phi}^{-}(\mathbf{U}_{ij,k}^{-}(\gamma, t), \mathbf{U}_{ij,k}^{+}(\gamma, t), \sigma_{ij}^{-}(\gamma), \sigma_{ij}^{+}(\gamma), \boldsymbol{\eta}_{ij,k}) d\gamma \\ & - \frac{1}{|C_{ij}|} \sum_{k \in \mathcal{N}_k} \int_{E_{ij,k}} \mathbf{F}_{\boldsymbol{\eta}_{ij,k}}(\mathbf{U}_{ij,k}^{-}(\gamma, t)) d\gamma \\ & - \frac{1}{|C_{ij}|} \int_{C_{ij}} \left(\mathcal{B}_x(P_{ij}^{U,t}(\mathbf{x})) \frac{\partial P_{ij}^{U,t}}{\partial x}(\mathbf{x}) + \mathcal{B}_y(P_{ij}^{U,t}(\mathbf{x})) \frac{\partial P_{ij}^{U,t}}{\partial y}(\mathbf{x}) \right) d\mathbf{x} \\ & + \frac{1}{|C_{ij}|} \int_{C_{ij}} \left(\mathcal{S}_x(P_{ij}^{U,t}(\mathbf{x})) \frac{\partial P_{ij}^{\sigma}}{\partial x}(\mathbf{x}) + \mathcal{S}_y(P_{ij}^{U,t}(\mathbf{x})) \frac{\partial P_{ij}^{\sigma}}{\partial y}(\mathbf{x}) \right) d\mathbf{x}, \end{aligned} \tag{1.6.9}$$

where $P_{ij}^{U,t}$ is the reconstruction approximation function at time t of \mathbf{U}_{ij}^t at cell C_{ij} , defined using the stencil \mathfrak{B}_{ij} , and P_{ij}^{σ} is the reconstruction approximation function for σ . The functions $\mathbf{U}_{ij,k}^{\pm}(\gamma, t)$ are given by

$$\mathbf{U}_{ij,k}^{-}(\gamma, t) = \lim_{\mathbf{x} \rightarrow \gamma} P_{ij}^{U,t}(\mathbf{x}), \quad \mathbf{U}_{ij,k}^{+}(\gamma, t) = \lim_{\mathbf{x} \rightarrow \gamma} P_k^{U,t}(\mathbf{x}), \tag{1.6.10}$$

and $\sigma_{ij,k}^{\pm}(\gamma)$ are given by

$$\sigma_{ij,k}^{-}(\gamma) = \lim_{\mathbf{x} \rightarrow \gamma} P_{ij}^{\sigma}(\mathbf{x}), \quad \sigma_{ij,k}^{+}(\gamma) = \lim_{\mathbf{x} \rightarrow \gamma} P_k^{\sigma}(\mathbf{x}). \tag{1.6.11}$$

In practice, the integral terms in (1.6.9) must be approximated numerically using a high-order quadrature formula, whose order is related to the one of the reconstruction operator ([43]).

Remark 3. As it was pointed out in [52, 43, 40], in general we have $m \leq q \leq p$, and the order of the semi-discrete operator (1.6.9) would be p for conservation laws and m for nonconservative systems. Therefore, a loss of accuracy for nonconservative systems can be expected when reconstruction operators are applied. This fact was first noticed for WENO-like reconstructions in [52], where the accuracy order in the interior of the cells is smaller than the order at the boundaries, i.e., $q < p$. An interesting alternative to WENO reconstructions for which $q = p$ is given by the CWENO method ([66]). Nevertheless, in practice, the predicted order of accuracy is rather pessimistic: order q is often achieved (see [52, 43]).

1.7 MUSCL-Hancock procedure

In this section we provide general guidelines in order to construct the predictor-corrector MUSCL-Hancock scheme ([200, 201]) for system (1.3.1). The method is second-order accurate in space and time, so we can use a middle-point rule integration in order to evaluate the volume integrals in (1.6.9). The reconstruction operators $P_{ij}^{U,t}(\mathbf{x})$ and $P_{ij}^\sigma(\mathbf{x})$ are often given by a linear operator with some limiter function. In short, we can define the following steps:

- (i) Predictor step: Given reconstruction operators $P_{ij}^{U,t}(\mathbf{x})$ and $P_{ij}^\sigma(\mathbf{x})$, we estimate the solution at the center of the interfaces, $U_{ij,k}^\pm(\gamma, t)$ and $\sigma_{ij,k}^\pm(\gamma)$, in the same way as (1.6.10) and (1.6.11). By a Cauchy-Kowalewski procedure, we get

$$\begin{aligned} \partial_t U_{ij}(t) = & -\frac{1}{|C_{ij}|} \sum_{k \in \mathcal{N}_k} \int_{E_{ij,k}} \mathbf{F}_{\mathbf{n}_{ij,k}}(U_{ij,k}^-(\gamma, t)) d\gamma \\ & - \mathcal{B}_x(U_{ij}(t)) \frac{\Delta U_{ij}(t)}{\Delta x} - \mathcal{B}_y(U_{ij}(t)) \frac{\Delta U_{ij}(t)}{\Delta y} \\ & + \mathcal{S}_x(U_{ij}(t)) \frac{\Delta \sigma_{ij}}{\Delta x} + \mathcal{S}_y(U_{ij}(t)) \frac{\Delta \sigma_{ij}}{\Delta y}. \end{aligned} \quad (1.7.1)$$

- (ii) Corrector step: The center values $U_{ij}(t)$ and the boundary values $U_{ij,k}^\pm(\gamma, t)$ are now advanced at a half time using (1.7.1):

$$\tilde{U}_{ij}(t) = U_{ij}(t) + \frac{1}{2} \Delta t \partial_t U_{ij}(t), \quad \tilde{U}_{ij,k}^\pm(\gamma, t) = U_{ij,k}^\pm(\gamma, t) + \frac{1}{2} \Delta t \partial_t U_{ij}(t). \quad (1.7.2)$$

Finally, the full scheme reads as

$$\begin{aligned} U_{ij}^{n+1} = U_{ij}^n + \Delta t \left\{ & -\frac{1}{|C_{ij}|} \sum_{k \in \mathcal{N}_k} \int_{E_{ij,k}} \mathbf{D}_\Phi^-(\tilde{U}_{ij,k}^-(\gamma, t), \tilde{U}_{ij,k}^+(\gamma, t), \sigma_{ij,k}^-(\gamma), \sigma_{ij,k}^+(\gamma), \mathbf{n}_{ij,k}) d\gamma \right. \\ & -\frac{1}{|C_{ij}|} \sum_{k \in \mathcal{N}_k} \int_{E_{ij,k}} \mathbf{F}_{\mathbf{n}_{ij,k}}(\tilde{U}_{ij,k}^-(\gamma, t)) d\gamma \\ & - \mathcal{B}_x(\tilde{U}_{ij}(t)) \frac{\Delta U_{ij}(t)}{\Delta x} - \mathcal{B}_y(\tilde{U}_{ij}(t)) \frac{\Delta U_{ij}(t)}{\Delta y} \\ & \left. + \mathcal{S}_x(\tilde{U}_{ij}(t)) \frac{\Delta \sigma_{ij}}{\Delta x} + \mathcal{S}_y(\tilde{U}_{ij}(t)) \frac{\Delta \sigma_{ij}}{\Delta y} \right\}. \end{aligned}$$

Chapter 2

Governing equations

When the fluid under study is an electrically conductive plasma, in which a magnetic field is present, the governing equations can be described by the magnetohydrodynamic equations (MHD). If the magnetic field is zero, we recover the system of Euler equations in gas dynamics. Many astrophysical phenomena are influenced by the presence of magnetic fields and can be described by the MHD system ([17]). The aim of Section 2.1 is to describe the eigenstructure and mathematical properties of the MHD system. The divergence-free constraint of the magnetic field is automatically satisfied in 1D simulations where $\partial_y = \partial_z = 0$ and B_x is constant, while for multidimensional simulations $\nabla \cdot \mathbf{B} = 0$ is not ensured by standard numerical methods. The non-zero divergence can cause spurious numerical effects and numerical instability in practical simulations ([196]). A wide number of methods satisfying the divergence-free constraint in MHD have been proposed: see, e.g., [6, 22, 20, 30, 68, 69, 76, 93, 100, 99, 146, 147, 148, 163, 164, 194, 196, 211, 215]. Related to this, a novel technique for divergence cleaning based on Powell's ([163]) formulation of MHD equations is presented here. This technique has been included in the two-dimensional AVM solvers (Chapter 3), by modifying the numerical fluxes accordingly to [149]. Numerical experiments show that the obtained results are comparable to those computed with the well-known projection method ([30]), but with a smaller computational cost.

Water flows in which the water depth is much smaller than the horizontal length scale can be modelled by the multi-layer shallow water equations. These equations are derived by vertical integration of the incompressible Navier-Stokes equations, with the assumption of hydrostatic pressure between layers. While the one-layer system is hyperbolic, the multi-layer system is conditionally hyperbolic: the eigenvalues can become complex because of the coupling terms between layers (see [44] for a numerical treatment for the loss of hyperbolicity in the two-layer case). In Sections 2.2 and 2.3 are recalled, respectively, the one- and two-layer equations with bottom topography, that will be used in the numerical experiments in Chapter 4.

Contact discontinuities related to the discontinuous topography are normally not well captured with conventional numerical methods. The *well-balanced* property is strongly related to the capacity of a method to approach well enough contact discontinuities. For Roe-type methods, this feature is closely related with the choice of the integral paths ([161]). In this sense, well-balanced numerical methods play an important role when the solutions of shallow water systems involve perturbations of a steady state, whose amplitude is of greater order

than the truncation error of the method. As it was pointed out in [137], this is the case of a tsunami wave propagation at the ocean: its initial amplitude is small (related to the depth of the ocean), while presenting a long wavelength, and typically increases when the depth of the water decreases. It is not ever possible to refine the mesh to the point where the truncation error of the method is lower than the wave amplitude (in this sense, Adaptively Mesh Refinement (AMR) algorithms are often used in these situations; see, for instance, [28, 26, 25, 27, 137]). Another difficulty is related with the numerical computation of steady or nearly steady state solutions. Standard numerical methods may fail in these situations. In the shallow water framework, if the method is able to capture water at rest solutions exactly, it is said to satisfy the *C-property*, which was introduced by Bermúdez and Vázquez-Cendón in [29]. The design of well-balanced methods is nowadays an important front of research: see, for instance, [12, 38, 48, 58, 57, 72, 73, 74, 102, 165, 135, 141, 155, 162, 166, 210]. In Section 2.4 we give the necessary background about well-balanced methods in order to be used within our 2D AVM schemes.

2.1 Magnetohydrodynamic equations

In this section we focus on the ideal Newtonian MHD equations, which characterizes a conducting fluid in the presence of a magnetic field. This system can be viewed as a coupling of the hydrodynamic Euler equations with the Maxwell equations of electrodynamics. We will use the following notation: ρ is density, $\mathbf{v} = (v_x, v_y, v_z)^t$ is the velocity field, $\mathbf{B} = (B_x, B_y, B_z)^t$ is the magnetic field, E is the total energy and P is the hydrodynamic pressure. Then, the equations of MHD read as

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0, \\ \partial_t (\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \mathbf{v}^t + (P + \frac{1}{2} \mathbf{B}^2) \mathbf{I} - \mathbf{B} \mathbf{B}^t) = 0, \\ \partial_t \mathbf{B} + \nabla \cdot (\mathbf{B} \mathbf{v}^t - \mathbf{v} \mathbf{B}^t) = 0, \\ \partial_t E + \nabla \cdot ((E + P + \frac{1}{2} \mathbf{B}^2) \mathbf{v} - \mathbf{B} (\mathbf{v} \cdot \mathbf{B})) = 0, \end{cases} \quad (2.1.1)$$

together with the divergence-free condition

$$\nabla \cdot \mathbf{B} = 0. \quad (2.1.2)$$

To close the system, an ideal equation of state $P = (\gamma - 1)\rho\varepsilon$ is considered, where γ is the adiabatic constant and ε is the specific internal energy, which is related to the total energy by $E = \frac{1}{2}\rho\mathbf{v}^2 + \frac{1}{2}\mathbf{B}^2 + \rho\varepsilon$. The total pressure is defined as $P^* = P + \frac{1}{2}\mathbf{B}^2$, where $\frac{1}{2}\mathbf{B}^2$ represents the magnetic pressure. Notice that if $\mathbf{B} = 0$ we recover the conservative system of Euler equations.

On the other hand, define $\mathbf{b} = \mathbf{B}/\sqrt{\rho}$ and the acoustic sound speed as $a = \sqrt{\gamma P/\rho}$. The set of eigenvalues and eigenvectors for system (2.1.1) is known: a detailed description of its spectral structure can be found in [31, 181]. The Jacobian of system (2.1.1) is diagonalizable; a full set of eigenvalues in the α -direction, for $\alpha = x, y$, is given by

$$\lambda_\alpha^1 \leq \lambda_\alpha^2 \leq \lambda_\alpha^3 \leq \lambda_\alpha^4 \leq \lambda_\alpha^5 \leq \lambda_\alpha^6 \leq \lambda_\alpha^7,$$

where

$$\lambda_\alpha^{1,7} = v_\alpha \mp c_{\alpha,f}, \quad \lambda_\alpha^{2,6} = v_\alpha \mp c_{\alpha,a}, \quad \lambda_\alpha^{3,5} = v_\alpha \mp c_{\alpha,s}, \quad \lambda_\alpha^4 = v_\alpha, \quad (2.1.3)$$

with

$$c_{\alpha,a}^2 = \frac{B_\alpha^2}{\rho}, \quad c_{\alpha,f/s}^2 = \frac{1}{2}(a^2 + \mathbf{b}^2 \pm \sqrt{(a^2 + \mathbf{b}^2)^2 - 4a^2b_\alpha^2}).$$

Brio and Wu ([31]) remarked that the ideal MHD system (2.1.1) is nonconvex, which means that the characteristic fields can be not genuinely nonlinear or linearly degenerate. Thus, the solution of a Riemann problem can develop compound waves, for example a shock followed by a rarefaction, what poses additional numerical difficulties.

We remark that in the derivation of (2.1.1) from physical principles, the condition (2.1.2) is thoroughly used. However, it is also possible to derive a MHD model without imposing the divergence constraint in the process, as proposed in [163] (see also [100, 164]). This leads to the *nonconservative form* of the MHD equations:

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0, \\ \partial_t (\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \mathbf{v}^t + (P + \frac{1}{2} \mathbf{B}^2) \mathbf{I} - \mathbf{B} \mathbf{B}^t) = -(\nabla \cdot \mathbf{B}) \mathbf{B}, \\ \partial_t \mathbf{B} + \nabla \cdot (\mathbf{B} \mathbf{v}^t - \mathbf{v} \mathbf{B}^t) = -(\nabla \cdot \mathbf{B}) \mathbf{v}, \\ \partial_t E + \nabla \cdot ((E + P + \frac{1}{2} \mathbf{B}^2) \mathbf{v} - \mathbf{B}(\mathbf{v} \cdot \mathbf{B})) = -(\nabla \cdot \mathbf{B})(\mathbf{v} \cdot \mathbf{B}). \end{cases} \quad (2.1.4)$$

Notice that if condition (2.1.2) is imposed, then (2.1.4) reduces to (2.1.1). The spectral structure of problem (2.1.4) was fully analyzed in [163]; the eigenvalue corresponding to the 8-th wave is equal to v_α ; in particular, the maximal speeds of propagation are the same as in the conservative case.

When designing numerical methods for solving (2.1.1), it is fundamental to handle the divergence constraint (2.1.2) in a proper way. In particular, in the presence of shocks, standard numerical methods can produce large divergence errors which may lead to negative densities or pressures. Several methods have been proposed in the literature to impose the divergence-free condition numerically: see [196] for an in-depth review on this topic. A technique of particular interest to handle the divergence constraint was proposed by Powell in [163], which is based in the eight-waves model (2.1.4). Notice that taking divergence of the magnetic field equation in (2.1.4) gives

$$\partial_t (\nabla \cdot \mathbf{B}) + \nabla \cdot ((\nabla \cdot \mathbf{B}) \mathbf{v}) = 0,$$

so any possible non-zero divergence produced by the numerical scheme is advected by the velocity field and eventually leaves the domain through the boundaries. Thus, the nonconservative form (2.1.4) of the MHD system includes a mechanism to handle the divergence constraint.

From now on we will concentrate on the two-dimensional case. Considering the vector of conserved variables $\mathbf{U} = (\rho, \rho \mathbf{v}, \mathbf{B}, E)^t$, system (2.1.1) can be written in conservative form as

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) + \partial_y \mathbf{G}(\mathbf{U}) = 0,$$

where the physical fluxes are given by

$$\mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho v_x \\ \rho v_x^2 + P^* - B_x^2 \\ \rho v_x v_y - B_x B_y \\ \rho v_x v_z - B_x B_z \\ 0 \\ v_x B_y - v_y B_x \\ v_x B_z - v_z B_x \\ v_x(E + P^*) - B_x(\mathbf{v} \cdot \mathbf{B}) \end{pmatrix}, \quad \mathbf{G}(\mathbf{U}) = \begin{pmatrix} \rho v_y \\ \rho v_x v_y - B_x B_y \\ \rho v_y^2 + P^* - B_y^2 \\ \rho v_y v_z - B_y B_z \\ v_y B_x - v_x B_y \\ 0 \\ v_y B_z - v_z B_y \\ v_y(E + P^*) - B_y(\mathbf{v} \cdot \mathbf{B}) \end{pmatrix}.$$

As commented before, some kind of divergence cleaning mechanism has to be added to the numerical scheme. For doing this we consider the nonconservative system (2.1.4), that can be expressed as

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) + \partial_y \mathbf{G}(\mathbf{U}) = -(\nabla \cdot \mathbf{B}) \mathbf{S}(\mathbf{U}), \quad (2.1.5)$$

where the source term is given by

$$\mathbf{S}(\mathbf{U}) = (0, \mathbf{B}, \mathbf{v}, \mathbf{v} \cdot \mathbf{B})^t.$$

Alternatively, (2.1.5) can be written in the form

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) + \partial_y \mathbf{G}(\mathbf{U}) = -\mathcal{B}_x(\mathbf{U}) \partial_x \mathbf{U} - \mathcal{B}_y(\mathbf{U}) \partial_y \mathbf{U}, \quad (2.1.6)$$

where $\mathcal{B}_x(\mathbf{U})$ and $\mathcal{B}_y(\mathbf{U})$ are the 8×8 matrices

$$\mathcal{B}_x(\mathbf{U}) = (\mathbf{0} | \mathbf{0} | \mathbf{0} | \mathbf{0} | \mathbf{S}(\mathbf{U}) | \mathbf{0} | \mathbf{0} | \mathbf{0}), \quad \mathcal{B}_y(\mathbf{U}) = (\mathbf{0} | \mathbf{0} | \mathbf{0} | \mathbf{0} | \mathbf{0} | \mathbf{S}(\mathbf{U}) | \mathbf{0} | \mathbf{0}),$$

with $\mathbf{0}$ the zero column vector in \mathbb{R}^8 . At this point, we would like to make a remark that will be useful later in Chapter 3: following [149], both the HLL states and the numerical AVM fluxes must be properly modified in order to take into account the extra terms $\mathcal{B}_x(\mathbf{U})$ and $\mathcal{B}_y(\mathbf{U})$ appearing in (2.1.6). In particular, the HLL states should be modified as

$$\tilde{\mathbf{U}}_x^{\text{HLL}}(\mathbf{U}_0, \mathbf{U}_1, S_0, S_1) = \frac{S_1 \mathbf{U}_1 - S_0 \mathbf{U}_0 + \mathbf{F}(\mathbf{U}_0) - \mathbf{F}(\mathbf{U}_1) - \mathcal{B}_x(\tilde{\mathbf{U}})(\mathbf{U}_1 - \mathbf{U}_0)}{S_1 - S_0}$$

and

$$\tilde{\mathbf{U}}_y^{\text{HLL}}(\mathbf{U}_0, \mathbf{U}_1, S_0, S_1) = \frac{S_1 \mathbf{U}_1 - S_0 \mathbf{U}_0 + \mathbf{G}(\mathbf{U}_0) - \mathbf{G}(\mathbf{U}_1) - \mathcal{B}_y(\tilde{\mathbf{U}})(\mathbf{U}_1 - \mathbf{U}_0)}{S_1 - S_0},$$

being $\tilde{\mathbf{U}}$ some average state. Accordingly, the numerical fluxes now should read as

$$\tilde{\mathbf{F}}^{\text{AVM}}(\mathbf{U}_0, \mathbf{U}_1) = \frac{\mathbf{F}(\mathbf{U}_0) + \mathbf{F}(\mathbf{U}_1)}{2} - \frac{1}{2}(\mathbf{Q}_x - \mathcal{B}_x(\tilde{\mathbf{U}}))(\mathbf{U}_1 - \mathbf{U}_0)$$

and

$$\tilde{\mathbf{G}}^{\text{AVM}}(\mathbf{U}_0, \mathbf{U}_1) = \frac{\mathbf{G}(\mathbf{U}_0) + \mathbf{G}(\mathbf{U}_1)}{2} - \frac{1}{2}(\mathbf{Q}_y - \mathcal{B}_y(\tilde{\mathbf{U}}))(\mathbf{U}_1 - \mathbf{U}_0).$$

Remark 4. For MHD, the vectors $\mathcal{B}_\zeta(\tilde{\mathbf{U}})(\mathbf{U}_1 - \mathbf{U}_0)$ are simply

$$\mathcal{B}_x(\tilde{\mathbf{U}}) = ((B_x)_1 - (B_x)_0)\mathcal{S}(\tilde{\mathbf{U}}), \quad \mathcal{B}_y(\tilde{\mathbf{U}}) = ((B_y)_1 - (B_y)_0)\mathcal{S}(\tilde{\mathbf{U}}).$$

Notice that the proposed schemes are designed for solving the eight-waves model (2.1.4), so they automatically include a divergence cleaning procedure. Thus, if both the initial and boundary conditions satisfy the divergence constraint, it is expected that the scheme will keep it up to the accuracy of the truncation error. Notice also that the proposed nonconservative divergence cleaning technique could be applied, in a similar way, to any arbitrary numerical method.

The well-known projection method ([30]) is based on the Hodge decomposition of the magnetic field

$$\mathbf{B}^n = \text{curl } \Phi + \nabla \Psi, \quad (2.1.7)$$

where \mathbf{B}^n may be not divergence-free at each time step. Taking the divergence in (2.1.7), we see that Ψ satisfies a Poisson equation

$$-\nabla \Psi = \nabla \cdot \mathbf{B}^n, \quad (2.1.8)$$

which needs to be resolved by an iterative method for elliptic equations. The corrected field

$$\mathbf{B}^* = \mathbf{B}^n - \nabla \Psi \quad (2.1.9)$$

will be then divergence-free. Tóth ([196]) states that the resultant projected field should be fairly close to the minimal solution and the numerical error is not increased in smooth parts of the solution by the projection. However, this method can be very expensive computationally, as it requires the solution of a collection of elliptic problems at each time step, with a proper set of boundary conditions. Many other different approaches have been proposed in order to solve correctly and efficiently the constraint (2.1.2): divergence cleaning ([68]), potential-based genuinely multidimensional (GMD) methods ([148]), special divergence operators/staggering ([22, 172, 194]), and constrained transport method ([93, 170]), among others.

In this dissertation, we will consider two versions of the two-dimensional AVM methods introduced in Chapter 3: the first one, for solving the conservative MHD equations (2.1.1) using the projection method for divergence cleaning, and the second one for solving the nonconservative MHD system (2.1.4) as explained above. We will see that both kind of methods produce similar results, although the second type has a smaller computational cost, as expected.

It must be noted that for the nonconservative case (2.1.4), the viscosity matrix (1.5.4) is evaluated using $\mathbf{A}_\zeta(\mathbf{U}) = \mathbf{J}_\zeta(\mathbf{U}) + \mathcal{B}_\zeta(\mathbf{U})$, i.e., the Jacobian matrix $\mathbf{J}_\zeta(\mathbf{U})$ plus the coupling matrix $\mathcal{B}_\zeta(\mathbf{U})$ (see (1.3.11) and (2.1.6)). For the sake of completeness, we give the form of the

Jacobian matrix of (2.1.4):

$$\mathbf{J}_\zeta(\mathbf{U}) = \begin{bmatrix} 0 & \eta_x & \eta_y & 0 & 0 & 0 & 0 & 0 \\ \delta_{21} & \delta_{22} & \delta_{23} & \delta_{24} & \delta_{25} & \delta_{26} & \delta_{27} & \delta_{28} \\ \delta_{31} & \delta_{32} & \delta_{33} & \delta_{34} & \delta_{35} & \delta_{36} & \delta_{37} & \delta_{38} \\ -v_x v_z \eta_x - v_y v_z \eta_y & v_z \eta_x & v_z \eta_y & v_x \eta_x + v_y \eta_y & -B_z \eta_x & -B_z \eta_y & -B_x \eta_x - B_y \eta_y & 0 \\ \frac{B_y v_x - B_x v_y \eta_y}{\rho} & \frac{-B_y \eta_y}{\rho} & \frac{B_x \eta_y}{\rho} & 0 & v_y \eta_y & -v_x \eta_y & 0 & 0 \\ \frac{-B_y v_x + B_x v_y \eta_x}{\rho} & \frac{B_y \eta_x}{\rho} & \frac{-B_x \eta_x}{\rho} & 0 & -v_y \eta_x & v_x \eta_x & 0 & 0 \\ \frac{-B_z v_x + B_x v_z \eta_x}{\rho} & \frac{B_z \eta_x}{\rho} & \frac{B_z \eta_y}{\rho} & \frac{-B_x \eta_x - B_y \eta_y}{\rho} & -v_z \eta_x & -v_z \eta_y & v_x \eta_x + v_y \eta_y & 0 \\ + \frac{-B_z v_y + B_y v_z \eta_y}{\rho} & & & & & & & \\ \delta_{81} & \delta_{82} & \delta_{83} & \delta_{84} & \delta_{85} & \delta_{86} & \delta_{87} & \delta_{88} \end{bmatrix} \quad (2.1.10)$$

where the specific total enthalpy is defined as

$$H^* = \frac{E + P^*}{\rho},$$

and

$$\begin{aligned} \delta_{21} &= (-v_x^2 + a^2 + (\gamma - 1)(\mathbf{v}^2 + \mathbf{b}^2 - H^*))\eta_x - v_x v_y \eta_y, \\ \delta_{22} &= (2v_x - (\gamma - 1)v_x)\eta_x + v_y \eta_y, \quad \delta_{23} = -(\gamma - 1)v_y \eta_x + v_x \eta_y, \\ \delta_{24} &= -(\gamma - 1)v_z \eta_x, \quad \delta_{25} = -\gamma B_x \eta_x - B_y \eta_y, \\ \delta_{26} &= (2 - \gamma)B_y \eta_x - B_x \eta_y, \quad \delta_{27} = (2 - \gamma)B_z \eta_x, \quad \delta_{28} = (\gamma - 1)\eta_x, \\ \delta_{31} &= -v_x v_y \eta_x + (-v_y^2 + a^2 + (\gamma - 1)(\mathbf{v}^2 + \mathbf{b}^2 - H^*))\eta_y, \\ \delta_{32} &= v_y \eta_x - (\gamma - 1)v_x \eta_y, \quad \delta_{33} = v_x \eta_x + (2v_y - (\gamma - 1)v_y)\eta_y, \\ \delta_{34} &= -(\gamma - 1)v_z \eta_y, \quad \delta_{35} = -B_y \eta_x - (2 - \gamma)B_x \eta_y, \\ \delta_{36} &= -B_x \eta_x - \gamma B_y \eta_y, \quad \delta_{37} = (2 - \gamma)B_z \eta_y, \quad \delta_{38} = (\gamma - 1)\eta_y, \\ \delta_{81} &= \left(-v_x H^* + v_x(\delta_{21} + v_x^2) + B_x \frac{(\mathbf{v} \cdot \mathbf{B})}{\rho}\right)\eta_x + \left(-v_y H^* + v_y(\delta_{31} + v_y^2) + B_y \frac{(\mathbf{v} \cdot \mathbf{B})}{\rho}\right)\eta_y, \\ \delta_{82} &= \left(H^* + v_x(\delta_{22} - 2v_x) - \frac{B_x^2}{\rho}\right)\eta_x + \left(v_y \delta_{32} - \frac{B_x B_y}{\rho}\right)\eta_y, \\ \delta_{83} &= \left(v_x \delta_{23} - \frac{B_x B_y}{\rho}\right)\eta_x + \left(H^* + v_y(\delta_{33} - 2v_y) - \frac{B_y^2}{\rho}\right)\eta_y, \\ \delta_{84} &= \left(v_x \delta_{24} - \frac{B_x B_z}{\rho}\right)\eta_x + \left(v_y \delta_{34} - \frac{B_y B_z}{\rho}\right)\eta_y, \\ \delta_{85} &= ((1 - \gamma)B_x v_x - \mathbf{v} \cdot \mathbf{B})\eta_x + (v_y \delta_{25} - B_y v_x)\eta_y, \\ \delta_{86} &= (v_x \delta_{26} - B_x v_y)\eta_x + ((1 - \gamma)B_y v_y - (\mathbf{v} \cdot \mathbf{B}))\eta_y, \\ \delta_{87} &= (v_x \delta_{27} - B_x v_z)\eta_x + (v_y \delta_{37} - B_y v_z)\eta_y, \\ \delta_{88} &= v_x \gamma \eta_x + v_y \gamma \eta_y. \end{aligned}$$

A Roe linearization for the 8-wave system (2.1.4) is obtained by taking ([35])

$$\widehat{\mathbf{U}}(\mathbf{U}_0, \mathbf{U}_1) = \mathbf{U}(\underline{\rho}, \underline{v_x}, \underline{v_y}, \underline{v_z}, \underline{B_x}, \underline{B_y}, \underline{B_z}, \underline{H^*}),$$

where the Roe averages depend of the quantities

$$\bar{\xi} \equiv \bar{\xi}(\xi_0, \xi_1) = \frac{\sqrt{\rho_0}\xi_0 + \sqrt{\rho_1}\xi_1}{\sqrt{\rho_0} + \sqrt{\rho_1}}, \quad \underline{\xi} \equiv \underline{\xi}(\xi_0, \xi_1) = \frac{\sqrt{\rho_1}\xi_0 + \sqrt{\rho_0}\xi_1}{\sqrt{\rho_0} + \sqrt{\rho_1}}. \quad (2.1.11)$$

2.2 Shallow water equations: one-layer approximation

The PDE system governing the flow of a shallow layer of an homogeneous inviscid fluid occupying a domain $\Omega \subset \mathbb{R}^2$ can be written as

$$\begin{cases} \partial_t h + \nabla \cdot (h\mathbf{v}) = 0, \\ \partial_t (h\mathbf{v}) + \nabla \cdot (h\mathbf{v}\mathbf{v}^t + \frac{1}{2}gh^2\mathbf{I}) = gh\nabla H, \end{cases} \quad (2.2.1)$$

or in the quasi-linear form (1.3.1) with

$$\mathbf{U} = \begin{pmatrix} h \\ q_x \\ q_y \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} q_x \\ \frac{q_x^2}{h} + \frac{g}{2}h^2 \\ \frac{q_x q_y}{h} \end{pmatrix}, \quad \mathbf{G}(\mathbf{U}) = \begin{pmatrix} q_y \\ \frac{q_x q_y}{h} \\ \frac{q_y^2}{h} + \frac{g}{2}h^2 \end{pmatrix},$$

$$\mathbf{B}_x(\mathbf{U}) = 0, \quad \mathbf{B}_y(\mathbf{U}) = 0, \quad \mathbf{S}_x(\mathbf{U}) = \begin{pmatrix} 0 \\ gh \\ 0 \end{pmatrix}, \quad \mathbf{S}_y(\mathbf{U}) = \begin{pmatrix} 0 \\ 0 \\ gh \end{pmatrix},$$

where: $\sigma(\mathbf{x}) \equiv H(\mathbf{x})$ is the depth function, measured from a fixed level of reference; g is the gravity constant; $h(\mathbf{x}, t)$ is the thickness of the water layer; and $q_\zeta(\mathbf{x}, t)$, $\zeta = x, y$, are the discharges in the coordinate directions, which are related to the vertical averaged velocities by $q_\zeta(\mathbf{x}, t) = h(\mathbf{x}, t)v_\zeta(\mathbf{x}, t)$. The non-zero eigenvalues of the system are given by $v_x \pm c$ and $v_y \pm c$, where $c = \sqrt{gh}$.

In the particular case in which the paths Φ are chosen as segments, we can build a Roe linerization of the form (1.3.11) with ([161])

$$\mathbf{A}_x(\mathbf{U}_0, \mathbf{U}_1) = \begin{pmatrix} 0 & 1 & 0 \\ -\hat{v}_x^2 + \hat{c}^2 & 2\hat{v}_x & 0 \\ -\hat{v}_x\hat{v}_y & \hat{v}_y & \hat{v}_x \end{pmatrix}, \quad \mathbf{S}_1(\mathbf{U}_0, \mathbf{U}_1) = \begin{pmatrix} 0 \\ \hat{c}^2 \\ 0 \end{pmatrix}, \quad (2.2.2)$$

$$\mathbf{A}_y(\mathbf{U}_0, \mathbf{U}_1) = \begin{pmatrix} 0 & 0 & 1 \\ -\hat{v}_x\hat{v}_y & \hat{v}_y & \hat{v}_x \\ -\hat{v}_y^2 + \hat{c}^2 & 0 & 2\hat{v}_y \end{pmatrix}, \quad \mathbf{S}_y(\mathbf{U}_0, \mathbf{U}_1) = \begin{pmatrix} 0 \\ 0 \\ \hat{c}^2 \end{pmatrix}, \quad (2.2.3)$$

where

$$\hat{v}_\zeta = \frac{\sqrt{h_0}v_{\zeta,0} + \sqrt{h_1}v_{\zeta,1}}{\sqrt{h_0} + \sqrt{h_1}}, \quad \hat{c} = \sqrt{g\hat{h}}, \quad \hat{h} = \frac{h_0 + h_1}{2}.$$

2.3 Shallow water equations: two-layer approximation

The equations modelling a two-dimensional flow of two superposed immiscible layers of shallow water fluids can be written as

$$\begin{cases} \partial_t h_1 + \nabla \cdot (h_1 \mathbf{v}_1) = 0, \\ \partial_t (h_1 \mathbf{v}_1) + \nabla \cdot (h_1 \mathbf{v}_1 \mathbf{v}_1^t + \frac{1}{2} g h_1^2 \mathbf{I}) = -g h_1 \nabla h_2 + g h_1 \nabla H, \\ \partial_t h_2 + \nabla \cdot (h_2 \mathbf{v}_2) = 0, \\ \partial_t (h_2 \mathbf{v}_2) + \nabla \cdot (h_2 \mathbf{v}_2 \mathbf{v}_2^t + \frac{1}{2} g h_2^2 \mathbf{I}) = -\rho g h_2 \nabla h_1 + g h_2 \nabla H, \end{cases} \quad (2.3.1)$$

or under the form (1.3.1) taking

$$\mathbf{U} = (h_1, q_{1,x}, q_{1,y}, h_2, q_{2,x}, q_{2,y})^t,$$

$$\mathbf{F}(\mathbf{U}) = \begin{pmatrix} q_{1,x} \\ \frac{q_{1,x}^2}{h_1} + \frac{g}{2} h_1^2 \\ \frac{q_{1,x} q_{1,y}}{h_1} \\ q_{2,x} \\ \frac{q_{2,x}^2}{h_2} + \frac{g}{2} h_2^2 \\ \frac{q_{2,x} q_{2,y}}{h_2} \end{pmatrix}, \quad \mathbf{G}(\mathbf{U}) = \begin{pmatrix} q_{1,y} \\ \frac{q_{1,x} q_{1,y}}{h_1} \\ \frac{q_{1,y}^2}{h_1} + \frac{g}{2} h_1^2 \\ q_{2,y} \\ \frac{q_{2,x} q_{2,y}}{h_2} \\ \frac{q_{2,y}^2}{h_2} + \frac{g}{2} h_2^2 \end{pmatrix},$$

$$\mathbf{B}_x(\mathbf{U}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -g h_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\rho g h_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{S}_x(\mathbf{U}) = \begin{pmatrix} 0 \\ g h_1 \\ 0 \\ 0 \\ g h_2 \\ 0 \end{pmatrix},$$

$$\mathbf{B}_y(\mathbf{U}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -g h_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\rho g h_2 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{S}_y(\mathbf{U}) = \begin{pmatrix} 0 \\ 0 \\ g h_1 \\ 0 \\ 0 \\ g h_2 \end{pmatrix},$$

where index 1 and 2 refer, respectively, to the upper and lower layer of fluid. It is assumed that the two layers have constant densities $\rho_1 < \rho_2$, and their ratio is denoted as $\rho = \rho_1/\rho_2$. The rest of the notation is analogous to that in Section 2.2. An additional numerical difficulty arises when ρ is close to unity, as the internal eigenvalues are close to zero; this situation arises in many practical situations (see, e.g., [53]).

For the two-layer system, there are no simple analytical expressions for the eigenstructure of the system. However, the following approximation to the maximal speeds of propagation is available when $v_{1,\zeta} \approx v_{2,\zeta}$ and $\rho \approx 1$ ([176]):

$$|\lambda_\zeta^{\max}| \approx |\bar{v}_\zeta| + \bar{c}, \quad \zeta = x, y,$$

where

$$\bar{v}_\zeta = \frac{q_{1,\zeta} + q_{2,\zeta}}{h_1 + h_2}, \quad \bar{c} = \sqrt{g(h_1 + h_2)}.$$

Finally, it is also possible to construct a Roe linearization for the two-layer shallow water system, in a similar way as shown in Section 2.2: see [43, Sect. 7.2] for details.

2.4 Well-balancing

It is a known issue that, in the presence of source terms, standard numerical methods may fail when computing steady or nearly steady state solutions. A way to overcome this problem is to consider *well-balanced* methods, which are able to preserve a chosen family of stationary solutions (see, e.g., [48]). In this section, we resume the main results about the well-balanced property for the scheme (1.6.9).

Definition 4. Consider a semi-discrete method to approximate (1.3.1)

$$\begin{cases} \mathbf{U}'_{ij}(t) = -\frac{1}{|C_{ij}|} \mathcal{H}(\mathbf{U}_k(t), k \in \mathfrak{B}_{ij}), \\ \mathbf{U}(0) = \mathbf{U}_0, \end{cases} \quad (2.4.1)$$

where $\mathbf{U}(t) = \{\mathbf{U}_{ij}(t)\}_{i,j=1}^{N_{\tau_x}, N_{\tau_y}}$ represents the vector of approximations to the averaged values of the exact solution; $\mathbf{U}_0 = \{\mathbf{U}_{ij}(0)\}$ is the vector of averages of the initial condition; and \mathfrak{B}_{ij} are the stencils. Given a smooth stationary solution \mathbf{U} of the system, the numerical scheme is said to be exactly well-balanced for \mathbf{U} if the vector of cell averages is a critical point of (2.4.1), i.e.,

$$\mathcal{H}(\mathbf{U}_k(t), k \in \mathfrak{B}_{ij}) = 0. \quad (2.4.2)$$

Let us also introduce now the concept of well-balanced reconstruction operator:

Definition 5. Given a smooth stationary solution of (1.3.1), a reconstruction operator is said to be well-balanced for $\mathbf{U}(\mathbf{x})$ if the approximation functions $P_{ij}^t(\mathbf{x})$ associated to the averaged values of \mathbf{U} are also stationary solutions of system (1.3.1).

The following results can be proven [40]:

Theorem 3. Let \mathbf{U} be a stationary solution of (1.3.1) and let us assume that the family of paths $\Phi(s; \mathbf{W}_0, \mathbf{W}_1, \boldsymbol{\eta}) = (\Phi_U(s; \mathbf{W}_0, \mathbf{W}_1, \boldsymbol{\eta}), \Phi_\sigma(s; \mathbf{W}_0, \mathbf{W}_1, \boldsymbol{\eta}))^t$ connecting two states $\mathbf{W}_0 = (\mathbf{U}(x_0), \sigma(x_0))^t$ and $\mathbf{W}_1 = (\mathbf{U}(x_1), \sigma(x_1))^t$ with $x_0 < x_1$ is a reparametrization of $x \in [x_0, x_1] \mapsto \mathbf{U}(x)$, then the first-order AVM scheme is exactly well-balanced for \mathbf{U} .

Theorem 4. *Let \mathbf{U} be a stationary solution of (1.3.1). Let us suppose that the first-order AVM path-conservative scheme and the reconstruction operator chosen are exactly well-balanced for \mathbf{U} . Then the numerical scheme (1.6.9) is also exactly well-balanced for \mathbf{U} .*

Remark 5. *Note that if the stationary solution is smooth, then $\mathbf{U}_{ij}^- = \mathbf{U}_{ij}^+$ and $\mathbf{D}_\Phi^\pm = 0$. Therefore, the well-balanced property of the high-order method only depends on the well-balanced property of the reconstruction operator.*

Notice that standard reconstruction operators are not expected, in general, to be well-balanced. In [48], a general procedure to modify any standard reconstruction operator in order to be well-balanced for every stationary solution was proposed. Let us also remark that quadrature formulae also play an important role in preserving the well-balanced properties of the scheme. The previous results have been established assuming that the integrals are exactly well-balanced. Otherwise, the strategy developed in [58] can be used.

In our case, we are interested in the so-called *C-property*, which consists in preserving water at rest solutions:

$$q_x = q_y = 0, \quad h - H = \text{constant} \quad (2.4.3)$$

for the one-layer shallow water system, and

$$q_{1,x} = q_{2,x} = q_{1,y} = q_{2,y} = 0, \quad h_1 = \text{constant}, \quad h_2 - H = \text{constant}$$

for the two-layer system. To accomplish this, we follow the *modified identity* technique introduced in [59]. For the sake of completeness, we briefly describe here the procedure for the one-layer shallow water equations, being the adaptation to the bilayer system straightforward.

Notice that the fluctuations in the x -direction given by (1.5.3) are not well-defined when $q_x = q_y = 0$, as the matrix \mathbf{A}_x becomes singular. However, for regular \mathbf{A}_x it is easy to check that $\mathbf{A}_x^{-1}\mathbf{S}_x = (1, 0, 0)^t \equiv \widehat{\mathbf{A}_x^{-1}\mathbf{S}_x}$. Thus, we consider the modified fluctuations

$$\mathbf{D}_x^\pm = \frac{1}{2}(\mathbf{F}_1 - \mathbf{F}_0 - \mathbf{S}_x(H_1 - H_0)) \pm \frac{1}{2}\mathbf{Q}_x(\mathbf{U}_1 - \mathbf{U}_0 - \widehat{\mathbf{A}_x^{-1}\mathbf{S}_x}(H_1 - H_0)),$$

which can be written as

$$\mathbf{D}_x^\pm = \frac{1}{2}(\mathbf{F}_1 - \mathbf{F}_0 - \mathbf{S}_x(H_1 - H_0)) \pm \frac{1}{2}\mathbf{Q}_x \begin{pmatrix} \eta_1 - \eta_0 \\ q_{x,1} - q_{x,0} \\ q_{y,1} - q_{y,0} \end{pmatrix},$$

and similarly in the y -direction. Therefore, for a stationary solution of the form (2.4.3), we would have

$$\mathbf{D}_x^\pm = \frac{1}{2} \begin{pmatrix} 0 \\ \frac{g}{2}(h_1^2 - h_0^2) - g\hat{h}(H_1 - H_0) \\ 0 \end{pmatrix} \pm \mathbf{Q}_x \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

As a consequence, the scheme preserves the stationary solution.

Remark 6. *In the particular case of a PVM method, where $\mathbf{Q} = \sum_{i=0}^r \alpha_i \mathbf{A}^i$, we have that*

$$\mathbf{Q}\mathbf{A}^{-1}\mathbf{S} = \alpha_0\mathbf{A}^{-1}\mathbf{S} + \sum_{i=1}^r \alpha_i \mathbf{A}^{i-1}\mathbf{S},$$

so the proposed modification only needs to be applied in the α_0 term.

Regarding high-order schemes, in order to have a reconstruction operator which is well-balanced for water at rest solutions the following strategy may be followed for the one-layer shallow water equations: given a set $(h_{ij}, q_{x,ij}, q_{y,ij}, H_{ij})$ of cell values, consider the new set $(h_{ij}, q_{x,ij}, q_{y,ij}, \eta_{ij})$, with $\eta_{ij} = h_{ij} - H_{ij}$, and apply the reconstruction operator to obtain polynomials

$$P_{ij,h}, \quad P_{ij,q_x}, \quad P_{ij,q_y}, \quad P_{ij,\eta}$$

then, define

$$P_{ij,H} = P_{ij,h} - P_{ij,\eta}.$$

This reconstruction is exactly well-balanced for stationary solutions corresponding to water at rest if the operator is exact for constant functions (see [43, Prop. 17]). The same idea can be applied for the two-layer case: given a set $(h_{1,ij}, q_{x1,ij}, q_{y1,ij}, h_{2,ij}, q_{x2,ij}, q_{y2,ij}, H_{ij})$, we apply the reconstruction operator to the new set $(h_{1,ij}, q_{x1,ij}, q_{y1,ij}, h_{2,ij}, q_{x2,ij}, q_{y2,ij}, \eta_{2,ij})$, with $\eta_{2,ij} = h_{2,ij} - H_{ij}$, to obtain

$$P_{ij,h_1}, \quad P_{ij,q_{x1}}, \quad P_{ij,q_{y1}}, \quad P_{ij,h_2}, \quad P_{ij,q_{x2}}, \quad P_{ij,q_{y2}}, \quad P_{ij,\eta_2};$$

then, define

$$P_{ij,H} = P_{ij,h_2} - P_{ij,\eta_2}.$$

(see [43, Prop. 19]).

Chapter 3

Multidimensional AVM-type solvers: the conservative case

Since the pioneering work of Godunov ([108]), Riemann solvers have been an important ingredient in the design of robust and accurate numerical methods for hyperbolic conservation laws. Usually, the exact solution of a Riemann problem contains many complex features which makes it unsuitable in practice, even more in multidimensional problems ([180, 213, 214]). For this reason, a number of incomplete Riemann solvers have been devised in the literature (Lax-Friedrichs, Rusanov, HLL, FORCE, etc; see [41] and the references therein), which take into account only some of the waves appearing in the Riemann fan.

In Section 1.5, we described a general framework for the so-called AVM methods, which constitute a general class of incomplete Riemann solvers defined in terms of viscosity matrices based on the polynomial/rational evaluation of a given Roe matrix, if available, or the Jacobian of the flux at some average state. This class of Roe-type Riemann solvers was introduced for the first time in [41] for polynomial approximations (PVM), and later extended in [49] to rational functions (RVM), which provide a much more precise representation of internal waves. In [49] it was also shown that the choice of a precise first-order solver is important when designing high-order methods in terms of computational efficiency, at least for solutions involving complex structures. In particular, the choice of appropriate underlying functions for an AVM solver allows to control the amount of numerical diffusion of the resulting scheme.

When applied to multidimensional problems, most of the numerical methods found in the literature are based on a dimensional splitting, which does not take into account multidimensional features. Furthermore, as it has been pointed out in the literature, this type of approximation can produce errors in maintaining the isotropy of solutions and reduce the CFL number in multidimensional simulations. Related to this, many authors have developed several techniques which introduce multidimensional features in the numerical schemes: see, e.g., [14, 15, 19, 16, 18, 21, 74, 104, 178, 203]. It is worth mentioning that there are alternative classes of schemes that are Riemann-solver free, although they are outside the scope of this thesis: see, for instance, residual distributions schemes in [4, 67, 167], or Godunov-type Riemann-problem-solver-free central schemes in [121, 123, 120, 60] and the references therein.

In [206], Wendroff proposed an multidimensional HLL Riemann solver for the Euler

equations, where features at corners were taken into account. However, the extension of Wendroff's method to high-order and to general hyperbolic systems is not easy. In [14, 15], Balsara introduced a genuinely two-dimensional Riemann solver based on the HLL and HLLC methods, which provided closed-form expressions of the fluxes and allowed an easy extension to high-order. Another extension of Wendroff's scheme has been proposed in [203]. In recent years, a new class of multidimensional Riemann solvers with self-similar internal structure has been introduced in [19, 21]. Also, extensions to unstructured meshes have been considered in [16, 18]. Recently, a new class of genuinely two-dimensional incomplete Riemann solvers based on AVM-type solvers was introduced in [104], which constitutes the core of the present chapter.

These solvers can be viewed as general AVM extensions of Balsara's HLL solver [15]. To achieve this, first we propose a reinterpretation of Balsara's solver in the particular case in which a simple four-wave model is considered for the two-dimensional corrections at cell vertices. These corrections can be viewed as suitable combinations of one-dimensional numerical fluxes of HLL type. Once the role of the underlying one-dimensional HLL solver is clearly identified, it can be changed by an appropriate one-dimensional AVM numerical flux. This leads to two-dimensional contributions of AVM type at the cell corners, which take into account the transversal features of the flow. It is noteworthy that the proposed solvers are applicable to general hyperbolic systems of conservation laws; in particular, they are not restricted to the Euler or MHD equations. An additional feature of these multidimensional AVM solvers is that they are theoretically stable up to a CFL number of unity.

Several families of basis functions for AVM methods were considered in [41, 49]; in particular, good results were obtained with Chebyshev polynomials and Newman rational functions. In this chapter we will consider PVM solvers based on the internal polynomials previously studied in [51], together with new families of RVM solvers based on the Padé approximations proposed in [104], which enjoy good stability properties (see Section 1.5 for details). An additional advantage of these solvers is that no entropy fix is needed in the presence of sonic points.

We will focus on the comparison of the proposed methods in the first-order case, as done previously for one-dimensional RVM methods in [49]. As commented before, the choice of accurate first-order solvers is important when analyzing complex scenarios. On the other hand, the use of our solvers as building blocks for high-order methods will be considered in Chapter 4.

In order to compare with other methods in the literature, applications to MHD equations will be considered, using the novel technique for divergence cleaning based on Powell's ([163]) formulation of MHD equations presented in Section 2.1. This technique has been included in our two-dimensional AVM solvers, modifying the numerical fluxes accordingly to [149]. Numerical experiments show that the results of our methods are comparable to those obtained with the well-known projection method [30], but with a smaller computational cost.

As it was remarked in [110], the inclusion of source terms in Balsara's solver seems to be a nontrivial matter. A way to handle source or coupling terms is to reformulate the system in nonconservative form, and then to apply the theory of path-conservative numerical schemes. Taking into account the form of our multidimensional AVM methods, it is possible to extend them to the nonconservative case following the guidelines in [41, 49]. This extension will be analyzed in detail in Chapter 4, where we will construct high-order multidimensional AVM schemes that will be applied, in particular, for solving multilayer shallow water systems including depth variations.

This chapter is organized as follows. In Section 3.1, Balsara's multidimensional HLL solver is recalled. The core of the chapter is Section 3.2, where multidimensional AVM solvers are introduced. Applications to MHD, including the proposed divergence cleaning technique, and numerical experiments are then presented in Section 3.3.

3.1 Preliminaries and four-waves model

The purpose of this section is to give a brief review of Balsara's HLL multidimensional solver. Although all the details can be found in [14, 15] (see also [203]), we give here the information that will be needed later to construct our schemes in Section 3.2. We focus here in a two-dimensional conservation law of the form (1.2.1), i.e.,

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) + \partial_y \mathbf{G}(\mathbf{U}) = 0, \quad (3.1.1)$$

where \mathbf{U} is defined on $\Omega \times [0, T]$, being $\Omega \subseteq \mathbb{R}^2$ a domain, and takes values on an open convex set $\mathcal{O} \subset \mathbb{R}^m$; $\mathcal{F} = (\mathbf{F}, \mathbf{G})$ is a regular function from \mathcal{O} to $\mathbb{R}^m \times \mathbb{R}^m$. In addition, we are interested in finite volume approximations given by

$$\mathbf{U}_{ij}^{n+1} = \mathbf{U}_{ij}^n - \frac{\Delta t}{\Delta x} (\mathbf{F}_{i+1/2,j}^n - \mathbf{F}_{i-1/2,j}^n) - \frac{\Delta t}{\Delta y} (\mathbf{G}_{i,j+1/2}^n - \mathbf{G}_{i,j-1/2}^n), \quad (3.1.2)$$

accordingly to (1.2.2) (see Section 1.2).

To define the numerical flux at a given edge, Balsara's solver considers a linear convex combination of a one-dimensional HLL flux and two multidimensional contributions at vertices. To be more precise, let us consider for a given cell C_{ij} the six-cell stencil depicted in Figure 3.1. The flux in the x -direction through the right edge can be expressed as

$$\mathbf{F}_{i+1/2,j} = \alpha \mathbf{F}_{i+1/2,j+1/2}^* + \beta \mathbf{F}_{i+1/2,j}^* + \gamma \mathbf{F}_{i+1/2,j-1/2}^*, \quad (3.1.3)$$

where $\mathbf{F}_{i+1/2,j}^*$ is the usual one-dimensional HLL flux between the cells C_{ij} and $C_{i+1,j}$, and $\mathbf{F}_{i+1/2,j\pm 1/2}^*$ represent the two-dimensional contributions at the vertices $(x_{i+1/2}, y_{j\pm 1/2})$; proper choices of the coefficients α , β and γ will be given later. Similar expressions can be stated for the y -flux $\mathbf{G}_{i,j+1/2}$.

To define the two-dimensional flux $\mathbf{F}_{i+1/2,j+1/2}^*$, Balsara considered a two-dimensional Riemann problem with initial condition given by the constant states at each of the four cells which have $(x_{i+1/2}, y_{j+1/2})$ as a common vertex. To simplify the notation, let us consider the local stencil depicted in Figure 3.2 (left); for example, for $O \equiv (x_{i+1/2}, y_{j+1/2})$ the cell values would be given by $\mathbf{U}_{LD} = \mathbf{U}_{ij}$, $\mathbf{U}_{RD} = \mathbf{U}_{i+1,j}$, $\mathbf{U}_{LU} = \mathbf{U}_{i,j+1}$ and $\mathbf{U}_{RU} = \mathbf{U}_{i+1,j+1}$. The initial condition for the local Riemann problem is then

$$\mathbf{U}(x, y, t^0) = \begin{cases} \mathbf{U}_{LD} & \text{if } x < 0, y < 0, \\ \mathbf{U}_{RD} & \text{if } x > 0, y < 0, \\ \mathbf{U}_{LU} & \text{if } x < 0, y > 0, \\ \mathbf{U}_{RU} & \text{if } x > 0, y > 0. \end{cases} \quad (3.1.4)$$

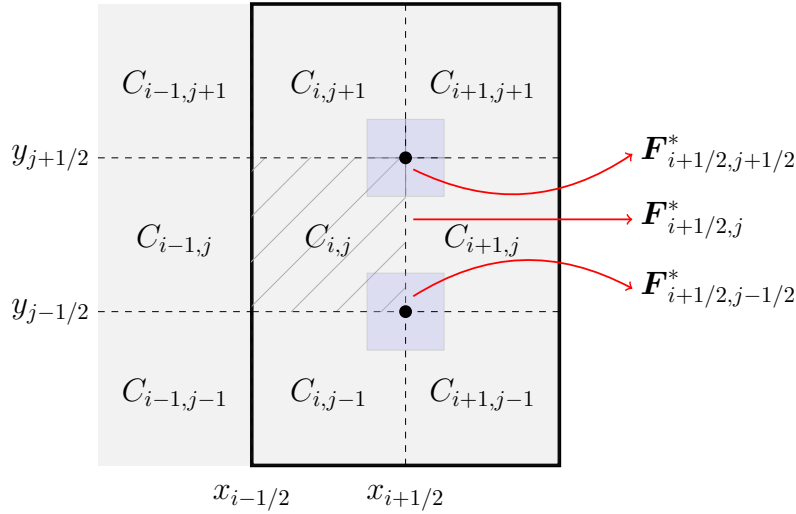


Figure 3.1: Stencil used to build the numerical flux $F_{i+1/2,j}$.

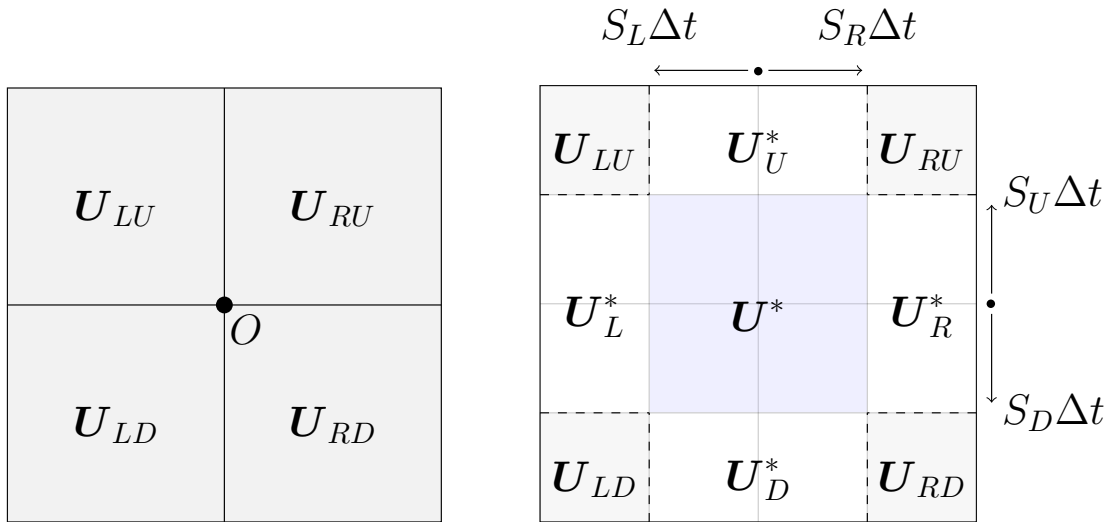


Figure 3.2: Left: Local stencil. Right: Structure of the solution of the Riemann problem.

It is known that problem (3.1.1)-(3.1.4) has a self-similar solution, roughly consisting of four directional one-dimensional Riemann problems arising at common edges, along with a region of strong interaction where complex structures appear; see, for example, [128, 180, 213] for more detailed descriptions. In the design of numerical methods, however, it is impractical to resolve all the fine structures arising in the strong interaction region. Thus, this region is averaged into a single constant state, in a similar way to the intermediate state arising in the one-dimensional HLL solver. In Balsara’s formulation, the region of strong interaction is assumed

to be rectangular, although other options are possible (see, e. g., [203]).

To build a two-dimensional Riemann solver of HLL type, first we need to identify the maximal speeds in the coordinate directions. Following the notation in [15], let S_L^U and S_R^U denote the fastest left and right speeds at the edge between \mathbf{U}_{LU} and \mathbf{U}_{RU} (see Figure 3.2, right). An usual choice for S_R^U and S_L^U are given by

$$\begin{aligned} S_R^U &= \max(\lambda_x^N(\mathbf{U}_{RU}), \bar{\lambda}_x^N(\mathbf{U}_{LU}, \mathbf{U}_{RU})), \\ S_L^U &= \min(\lambda_x^1(\mathbf{U}_{LU}), \bar{\lambda}_x^1(\mathbf{U}_{LU}, \mathbf{U}_{RU})), \end{aligned} \quad (3.1.5)$$

where $\lambda_x^1(\mathbf{U}_{LU})$ and $\lambda_x^N(\mathbf{U}_{RU})$ denote the maximal left- and right-going wave speeds associated to the states \mathbf{U}_{LU} and \mathbf{U}_{RU} , respectively, while $\bar{\lambda}_x^1(\mathbf{U}_{LU}, \mathbf{U}_{RU})$ and $\bar{\lambda}_x^N(\mathbf{U}_{LU}, \mathbf{U}_{RU})$ are the maximal left- and right-going speeds arising from a linearized Riemann solver between the states \mathbf{U}_{LU} and \mathbf{U}_{RU} . Similar definitions apply for S_L^D and S_R^D . Speeds in the y -direction, S_D^L , S_D^R , S_U^L and S_U^R , are defined in an analogous way. Finally, the strong interaction region is assumed to be bounded by the maximal left and right wave speeds given by

$$S_L = \min(S_L^D, S_L^U), \quad S_R = \max(S_R^D, S_R^U), \quad (3.1.6)$$

and the maximal downward and upward speeds

$$S_D = \min(S_D^L, S_D^R), \quad S_U = \max(S_U^L, S_U^R). \quad (3.1.7)$$

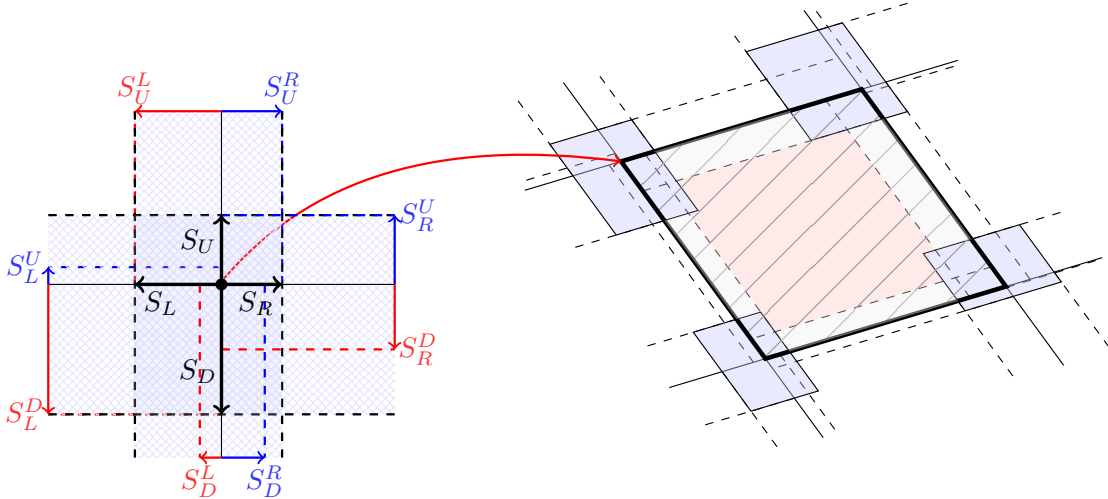


Figure 3.3: Local view (left) and global view (right) of the four-waves model.

A simplifying assumption can be made at this point, which consists in considering only the four speeds S_L , S_R , S_D and S_U in the wave structure of the solution. That is, we assume that $S_\alpha = S_\alpha^D = S_\alpha^U$ for $\alpha = L, R$, and $S_\beta = S_\beta^L = S_\beta^R$ for $\beta = D, U$. This leads to compact expressions for the strongly interacting state and its associated numerical fluxes. See Figure 3.3

for both local and global views of the four-waves model. On the right side, we have a arbitrary cell where are depicted all the waves arising at the edges and vertices. On the left side, we have a representation of the strong interaction region determined by the maximal wave speeds for each direction. As later in Section 3.2 we will be interested on a four-wave model, this assumption will be made in what follows.

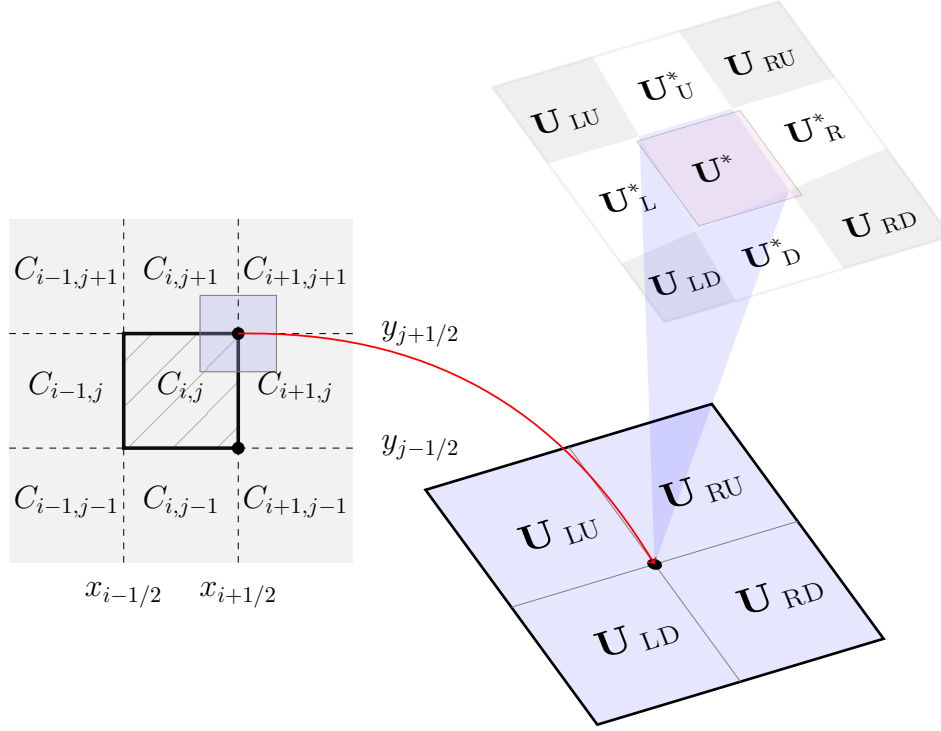


Figure 3.4: Structure of the solution of the 2D Riemann problem at a vertex.

Assume now that we are in the subsonic case in both directions, that is, $S_L < 0 < S_R$ and $S_D < 0 < S_U$. Figure 3.2 (right) shows the proposed wave structure at time $t = \Delta t$. The one-dimensional Riemann problems arising at the edges give rise to four HLL states; thus, for $\alpha = D, U$, the states $U_{L\alpha}$ and $U_{R\alpha}$ give

$$U_{\alpha}^* = \frac{S_R U_{R\alpha} - S_L U_{L\alpha} + F_{L\alpha} - F_{R\alpha}}{S_R - S_L}, \quad (3.1.8)$$

while for $\beta = L, R$, the states $U_{\beta D}$ and $U_{\beta U}$ give

$$U_{\beta}^* = \frac{S_U U_{\beta U} - S_D U_{\beta D} + G_{\beta D} - G_{\beta U}}{S_U - S_D}, \quad (3.1.9)$$

where $F_{\beta\alpha}$ and $G_{\beta\alpha}$ denote, respectively, the fluxes $F(U_{\beta\alpha})$ and $G(U_{\beta\alpha})$. The corresponding HLL fluxes are then given by

$$F_{\alpha}^* = \frac{S_R F_{L\alpha} - S_L F_{R\alpha} + S_L S_R (U_{R\alpha} - U_{L\alpha})}{S_R - S_L}, \quad \alpha = D, U,$$

and

$$\mathbf{G}_\beta^* = \frac{S_U \mathbf{G}_{\beta D} - S_D \mathbf{G}_{\beta U} + S_D S_U (\mathbf{U}_{\beta U} - \mathbf{U}_{\beta D})}{S_U - S_D}, \quad \beta = L, R.$$

Now, integration of (3.1.1) on the control volume $(S_L \Delta t, S_R \Delta t) \times (S_D \Delta t, S_U \Delta t) \times (0, \Delta t)$ leads to an explicit expression for the constant resolved state \mathbf{U}^* , namely

$$\begin{aligned} \mathbf{U}^* = & \frac{S_R S_U \mathbf{U}_{RU} + S_L S_D \mathbf{U}_{LD} - S_R S_D \mathbf{U}_{RD} - S_L S_U \mathbf{U}_{LU}}{(S_R - S_L)(S_U - S_D)} \\ & - \frac{(\mathbf{F}_{RU} - \mathbf{F}_{LU}) S_U - (\mathbf{F}_{RD} - \mathbf{F}_{LD}) S_D}{2(S_R - S_L)(S_U - S_D)} \\ & - \frac{(\mathbf{G}_{RU} - \mathbf{G}_{RD}) S_R - (\mathbf{G}_{LU} - \mathbf{G}_{LD}) S_L}{2(S_R - S_L)(S_U - S_D)} \\ & - \frac{\mathbf{F}_R^* - \mathbf{F}_L^*}{2(S_R - S_L)} - \frac{\mathbf{G}_U^* - \mathbf{G}_D^*}{2(S_U - S_D)}. \end{aligned}$$

Figure 3.4 shows the considered structure of the solution of the two-dimensional Riemann problem at a vertex. Notice that all the terms appearing in the definition of \mathbf{U}^* are known, except the *transverse fluxes* appearing in the last two terms, which have to be defined taking into account the corresponding intermediate state and the normal flux. For example, \mathbf{F}_R^* is constructed using the state \mathbf{U}_R^* and the flux \mathbf{G}_R^* previously defined; in particular, notice that $\mathbf{F}_R^* \neq \mathbf{F}(\mathbf{U}_R^*)$ in general. Balsara offers an explicit solution in [15] for the Euler equations, although similar constructions may be done for other systems: see Section 3.3 for the case of MHD equations.

To compute the resolved flux \mathbf{F}^* , equation (3.1.1) is integrated on the subvolume $(0, S_R \Delta t) \times (S_D \Delta t, S_U \Delta t) \times (0, \Delta t)$, yielding

$$\begin{aligned} \mathbf{F}^* = & \frac{S_U}{S_U - S_D} \mathbf{F}_U^* - \frac{S_D}{S_U - S_D} \mathbf{F}_D^* \\ & - \frac{S_L S_R}{(S_R - S_L)(S_U - S_D)} (\mathbf{G}_{RU} - \mathbf{G}_{LU} + \mathbf{G}_{LD} - \mathbf{G}_{RD}) \\ & - \frac{S_L S_U (\mathbf{F}_{RU} - \mathbf{F}_R^*) - S_R S_U (\mathbf{F}_{LU} - \mathbf{F}_L^*)}{(S_R - S_L)(S_U - S_D)} \\ & - \frac{S_R S_D (\mathbf{F}_{LD} - \mathbf{F}_L^*) - S_L S_D (\mathbf{F}_{RD} - \mathbf{F}_R^*)}{(S_R - S_L)(S_U - S_D)}. \end{aligned} \tag{3.1.10}$$

Notice again that, in general, $\mathbf{F}^* \neq \mathbf{F}(\mathbf{U}^*)$. Similarly, integration of (3.1.1) on $(S_L \Delta t, S_R \Delta t) \times (0, S_U \Delta t) \times (0, \Delta t)$ leads to

$$\begin{aligned} \mathbf{G}^* = & \frac{S_R}{S_R - S_L} \mathbf{G}_R^* - \frac{S_L}{S_R - S_L} \mathbf{G}_L^* \\ & - \frac{S_D S_U}{(S_R - S_L)(S_U - S_D)} (\mathbf{F}_{RU} - \mathbf{F}_{LU} + \mathbf{F}_{LD} - \mathbf{F}_{RD}) \\ & - \frac{S_R S_D (\mathbf{G}_{RU} - \mathbf{G}_U^*) - S_L S_D (\mathbf{G}_{LU} - \mathbf{G}_U^*)}{(S_R - S_L)(S_U - S_D)} \\ & - \frac{S_L S_U (\mathbf{G}_{LD} - \mathbf{G}_D^*) - S_R S_U (\mathbf{G}_{RD} - \mathbf{G}_D^*)}{(S_R - S_L)(S_U - S_D)}. \end{aligned} \tag{3.1.11}$$

As stated before, the fluxes (3.1.10) and (3.1.11) are used when the flow is subsonic in both spatial directions. When the flow is supersonic in some of the spatial directions, the corresponding expressions for \mathbf{F}^* and \mathbf{G}^* may be defined as the corresponding upwinded fluxes; the details can be found in [15, Sect. 2.1].

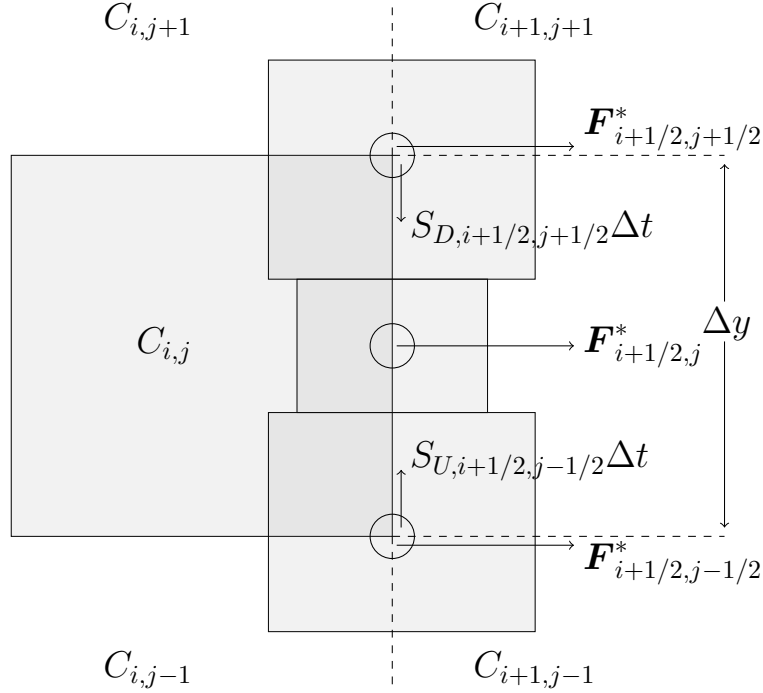


Figure 3.5: Assembling of one- and two-dimensional fluxes at an edge.

Finally, the assembling (3.1.3) is obtained from a time-space average of the normal flux through the edge (see Figure 3.5). This leads to the coefficients

$$\alpha = -S_{D,i+1/2,j+1/2} \frac{\Delta t}{2\Delta y}, \quad \gamma = S_{U,i+1/2,j-1/2} \frac{\Delta t}{2\Delta y}$$

and

$$\beta = 1 - (S_{U,i+1/2,j-1/2} - S_{D,i+1/2,j+1/2}) \frac{\Delta t}{2\Delta y},$$

with the obvious notations. Strictly speaking, these values are valid only for the subsonic case in both directions. However, supersonic cases can be accommodated by simply substituting $S_{U,R}$ by $S_{U,R}^+ = \max(S_{U,R}, 0)$ and $S_{D,L}$ by $S_{D,L}^- = \min(S_{D,L}, 0)$. Another option is to use Simpson's rule to approximate the integral, which leads to the simpler choice

$$\alpha = \frac{1}{6}, \quad \beta = \frac{2}{3}, \quad \gamma = \frac{1}{6}.$$

We have chosen this approach in the numerical experiments, although no significative differences have been found when the first option is used. More accurate approaches are possible: see, e.g., [203, Sect. 3.3.2].

With respect to the CFL condition needed to ensure stability, the multidimensional corrections allow for a maximum CFL number of unity: see [14] for a discussion on this point. In particular, the condition which ensures that the waves emanating from the corners do not touch each other is given by

$$\Delta t \leq \min \left(\frac{\Delta x}{S_{R,i-1/2,j+1/2} - S_{L,i+1/2,j+1/2}}, \frac{\Delta x}{S_{R,i-1/2,j-1/2} - S_{L,i+1/2,j-1/2}}, \frac{\Delta y}{S_{U,i+1/2,j-1/2} - S_{D,i+1/2,j+1/2}}, \frac{\Delta y}{S_{U,i-1/2,j-1/2} - S_{D,i-1/2,j+1/2}} \right).$$

Nevertheless, this condition could be relaxed, considering that the numerical flux at the edge is assembled as a linear convex combination of the one-dimensional contribution at the edge and the two-dimensional contributions at the vertices. Therefore, in order to allow for the maximal CFL number, we impose

$$\Delta t \leq \min \left(\frac{2\Delta x}{S_{R,i-1/2,j+1/2} - S_{L,i+1/2,j+1/2}}, \frac{2\Delta x}{S_{R,i-1/2,j-1/2} - S_{L,i+1/2,j-1/2}}, \frac{2\Delta y}{S_{U,i+1/2,j-1/2} - S_{D,i+1/2,j+1/2}}, \frac{2\Delta y}{S_{U,i-1/2,j-1/2} - S_{D,i-1/2,j+1/2}} \right). \quad (3.1.12)$$

Despite this, we use a standard central zone CFL condition in our experiments in Section 3.3. In Section 4.3.3, after analyzing the modified equation of the scheme (4.1.9), we find exactly the condition (3.1.12) for the 2D linear transport equation, and we prove the linear L^∞ -stability of the scheme.

3.2 2D AVM-type solvers for conservation laws

The purpose of this section is to extend Balsara's multidimensional solver by changing, in an appropriate way, the underlying HLL flux by a more precise AVM solver. The key point to achieve this is the reinterpretation of Balsara's solver in such a way that the two-dimensional corrections are expressed as suitable combinations of one-dimensional numerical fluxes of HLL type, which in turn can be changed by another AVM flux. We will use the same notations as in Section 3.1 for the speeds, states and fluxes. In particular, we are interested in a simple four-wave model, so only the speeds S_L , S_R , S_D and S_U will be taken into account.

To begin with, consider the local stencil shown in Figure 3.2, and let us see how the two-dimensional resolved flux \mathbf{F}^* is defined. Let \mathbf{U}_L^* and \mathbf{U}_R^* be the HLL states given by (3.1.9), and let \mathbf{F}_L^* and \mathbf{F}_R^* be the transverse fluxes considered in Section 3.1. Notice that the explicit form of these transverse fluxes is problem-dependent: see Section 3.3 for the case of the MHD equations. Next, consider the AVM-type flux (1.5.1) in the x -direction given by

$$\mathbf{F}^* = \frac{\mathbf{F}_L^* + \mathbf{F}_R^*}{2} - \frac{1}{2} \mathbf{Q}_x^*(\mathbf{U}_R^* - \mathbf{U}_L^*), \quad (3.2.1)$$

with $\mathbf{Q}_x^* = |\lambda_{\max}^x| f(|\lambda_{\max}^x|^{-1} \mathbf{A}^*)$, where $f(x)$ is a function verifying the stability condition (1.5.5), \mathbf{A}^* is a Roe-like matrix such that

$$\mathbf{A}^*(\mathbf{U}_R^* - \mathbf{U}_L^*) = \mathbf{F}_R^* - \mathbf{F}_L^*, \quad (3.2.2)$$

and λ_{\max}^x is a bound on its spectral radius.

Remark 7. As stated in Section 3.1, in general $\mathbf{F}_\alpha^* \neq \mathbf{F}(\mathbf{U}_\alpha^*)$, $\alpha = L, R$, so \mathbf{A}^* is not exactly a Roe matrix. Its form will depend on how the transverse fluxes are built. In the cases in which such a matrix cannot be constructed, \mathbf{A}^* can be taken as the Jacobian of the flux \mathbf{F} evaluated at some average state, so (3.2.2) holds only approximately.

Remark 8. Notice that the vertical contributions in the fluxes (3.2.1) are included through the transverse fluxes $\mathbf{F}_R^* \equiv \mathbf{F}_R^*(\mathbf{U}_R^*, \mathbf{G}_R^*)$ and $\mathbf{F}_L^* \equiv \mathbf{F}_L^*(\mathbf{U}_L^*, \mathbf{G}_L^*)$, as commented in Section 3.1.

It was shown in [41] that the HLL method can be viewed as an AVM solver based on the first-order polynomial

$$f(x) = \alpha_0 + \alpha_1 x, \quad (3.2.3)$$

with

$$\alpha_0 = \frac{S_R |S_L| - S_L |S_R|}{S_R - S_L}, \quad \alpha_1 = \frac{|S_R| - |S_L|}{S_R - S_L}.$$

Thus, substituting in (3.2.1) and using (3.2.2), we get the following expression:

$$\begin{aligned} \mathbf{F}^* = & \frac{S_U}{S_U - S_D} \left(\frac{\mathbf{F}_L^* + \mathbf{F}_R^*}{2} - \frac{1}{2} (\alpha_0 (\mathbf{U}_{RU} - \mathbf{U}_{LU}) + \alpha_1 (\mathbf{F}_R^* - \mathbf{F}_L^*)) \right) \\ & - \frac{S_D}{S_U - S_D} \left(\frac{\mathbf{F}_L^* + \mathbf{F}_R^*}{2} - \frac{1}{2} (\alpha_0 (\mathbf{U}_{RD} - \mathbf{U}_{LD}) + \alpha_1 (\mathbf{F}_R^* - \mathbf{F}_L^*)) \right) \\ & + \frac{\alpha_0}{2(S_U - S_D)} (\mathbf{G}_{RU} - \mathbf{G}_{LU} + \mathbf{G}_{LD} - \mathbf{G}_{RD}), \end{aligned} \quad (3.2.4)$$

where the explicit form (3.1.9) for the HLL states has been applied. Notice that the first two lines of (3.2.1) can be interpreted as two one-dimensional HLL fluxes associated, respectively, to the upper and lower parts of the stencil, while the last line accounts for the cross derivative of the flux \mathbf{G} .

Assuming now that we are in the subsonic case in both directions, so

$$\alpha_0 = -2 \frac{S_L S_R}{S_R - S_L}, \quad \alpha_1 = \frac{S_R + S_L}{S_R - S_L},$$

substitution in (3.2.4) leads to exactly the same formula (3.1.10) for Balsara's resolved flux. This means that the two-dimensional resolved flux (3.1.10) can be reinterpreted as an AVM-type flux of the form (3.2.1). Analogous conclusions are obtained when the flux is supersonic in one or both directions.

Similar computations can be done to write the resolved flux (3.1.11) in the y -direction as

$$\mathbf{G}^* = \frac{\mathbf{G}_D^* + \mathbf{G}_U^*}{2} - \frac{1}{2} \mathbf{Q}_y^*(\mathbf{U}_U^* - \mathbf{U}_D^*), \quad (3.2.5)$$

where $\mathbf{Q}_y^* = |\lambda_{\max}^y|g(|\lambda_{\max}^y|^{-1}\mathbf{B}^*)$; here $g(x)$ is a function verifying the stability condition (1.5.5), \mathbf{B}^* is a Roe-like matrix verifying $\mathbf{B}^*(\mathbf{U}_U^* - \mathbf{U}_D^*) = \mathbf{G}_U^* - \mathbf{G}_D^*$, and λ_{\max}^y is a bound on its spectral radius.

Once the two-dimensional fluxes (3.1.10) and (3.1.11) have been expressed as the AVM-type fluxes (3.2.1) and (3.2.5), it is possible to extend these to build more precise two-dimensional solvers. This is achieved by changing the underlying HLL polynomial (3.2.3) by another suitable function satisfying (1.5.5), thus obtaining an AVM-type two-dimensional flux. It is important to remark that to build an AVM solver, only the maximum wave speeds S_α are needed; despite simplicity, this is another reason why we consider a four-wave model. In particular, we always assume that the wave structure of the solution is that shown in Figure 3.2 (right). On the other hand, notice that in (3.2.1)-(3.2.5) the resolved states \mathbf{U}_α^* are precisely the HLL states (3.1.8)-(3.1.9). We will conserve these resolved states even though the basis function $f(x)$ used to build the fluxes (3.2.1)-(3.2.5) is changed. A possible way to introduce more structure in the resolved states, in the case of a PVM solver, could be through the equivalence of PVM solvers and simple Riemann solvers stated in [149].

An important consequence of the above considerations is that our two-dimensional AVM solvers could be adapted to define path-conservative schemes for nonconservative systems, in the spirit of [160]. In particular, this allows to extend the two-dimensional AVM solvers to the case of hyperbolic systems with coupling and source terms. This is the subject of Chapter 4.

For the sake of completeness, we end this section by summarizing the construction of the horizontal numerical flux $\mathbf{F}_{i+1/2,j}$ and the vertical numerical flux $\mathbf{G}_{i,j+1/2}$ for a 2D AVM method.

First of all, for generic cell values \mathbf{U}_0 and \mathbf{U}_1 , and maximal speeds of propagation S_0 and S_1 , define the HLL states

$$\mathbf{U}_x^{\text{HLL}}(\mathbf{U}_0, \mathbf{U}_1, S_0, S_1) = \frac{S_1\mathbf{U}_1 - S_0\mathbf{U}_0 + \mathbf{F}(\mathbf{U}_0) - \mathbf{F}(\mathbf{U}_1)}{S_1 - S_0}$$

and

$$\mathbf{U}_y^{\text{HLL}}(\mathbf{U}_0, \mathbf{U}_1, S_0, S_1) = \frac{S_1\mathbf{U}_1 - S_0\mathbf{U}_0 + \mathbf{G}(\mathbf{U}_0) - \mathbf{G}(\mathbf{U}_1)}{S_1 - S_0}.$$

Given a function $f(x)$ satisfying (1.5.5), define the one-dimensional AVM fluxes

$$\mathbf{F}^{\text{AVM}}(\mathbf{U}_0, \mathbf{U}_1) = \frac{\mathbf{F}(\mathbf{U}_0) + \mathbf{F}(\mathbf{U}_1)}{2} - \frac{1}{2}\mathbf{Q}_x(\mathbf{U}_1 - \mathbf{U}_0)$$

and

$$\mathbf{G}^{\text{AVM}}(\mathbf{U}_0, \mathbf{U}_1) = \frac{\mathbf{G}(\mathbf{U}_0) + \mathbf{G}(\mathbf{U}_1)}{2} - \frac{1}{2}\mathbf{Q}_y(\mathbf{U}_1 - \mathbf{U}_0),$$

where the viscosity matrices \mathbf{Q}_x and \mathbf{Q}_y are defined as in (1.5.4), in terms of suitable Roe or Jacobian matrices \mathbf{A}_x and \mathbf{A}_y , respectively.

Let us focus first on the horizontal flux $\mathbf{F}_{i+1/2,j}$. Consider the six-cell stencil around the vertical edge depicted in Figure 3.1. Then:

- The 1D flux through the edge is defined by

$$\mathbf{F}_{i+1/2,j}^* = \mathbf{F}^{\text{AVM}}(\mathbf{U}_{ij}, \mathbf{U}_{i+1,j}).$$

- To build the 2D flux $\mathbf{F}_{i+1/2,j+1/2}^*$ at the vertex $(x_{i+1/2}, y_{j+1/2})$, first compute the four local speeds of propagation S_L, S_R, S_U and S_D (we omit the indexes for clarity) as explained in Section 3.1.

Subsonic case in both directions: $S_L < 0 < S_R$ and $S_D < 0 < S_U$. Define the states

$$\begin{aligned} \mathbf{U}_L^* &= \mathbf{U}_y^{\text{HLL}}(\mathbf{U}_{ij}, \mathbf{U}_{i,j+1}, S_D, S_U), \\ \mathbf{U}_R^* &= \mathbf{U}_y^{\text{HLL}}(\mathbf{U}_{i+1,j}, \mathbf{U}_{i+1,j+1}, S_D, S_U), \end{aligned} \quad (3.2.6)$$

and the 1D AVM fluxes

$$\begin{aligned} \mathbf{G}_L^* &= \mathbf{G}^{\text{AVM}}(\mathbf{U}_{ij}, \mathbf{U}_{i,j+1}), \\ \mathbf{G}_R^* &= \mathbf{G}^{\text{AVM}}(\mathbf{U}_{i+1,j}, \mathbf{U}_{i+1,j+1}). \end{aligned} \quad (3.2.7)$$

Then build the transverse flux \mathbf{F}_L^* in terms of \mathbf{U}_L^* and \mathbf{G}_L^* , and the transverse flux \mathbf{F}_R^* in terms of \mathbf{U}_R^* and \mathbf{G}_R^* . Finally, the 2D flux is given by

$$\mathbf{F}_{i+1/2,j+1/2}^* = \frac{\mathbf{F}_L^* + \mathbf{F}_R^*}{2} - \frac{1}{2} \mathbf{Q}_x^*(\mathbf{U}_R^* - \mathbf{U}_L^*),$$

where \mathbf{Q}_x^* is defined as explained in Section 1.5 in terms of a Roe-like matrix \mathbf{A}_x^* satisfying (3.2.2); if such a matrix is not available, we simply take the Jacobian of the physical flux \mathbf{F} evaluated at some average state.

Supersonic case in some or both directions: The flux $\mathbf{F}_{i+1/2,j+1/2}^*$ is defined as the corresponding upwinded flux, exactly as in [15, Sect. 2.1].

- The 2D flux $\mathbf{F}_{i+1/2,j-1/2}^*$ at the lower vertex is defined similarly.
- Finally, the 1D and 2D fluxes are assembled as

$$\mathbf{F}_{i+1/2,j} = \frac{1}{6} \mathbf{F}_{i+1/2,j+1/2}^* + \frac{2}{3} \mathbf{F}_{i+1/2,j}^* + \frac{1}{6} \mathbf{F}_{i+1/2,j-1/2}^*, \quad (3.2.8)$$

where, for simplicity, Simpson's rule has been considered (see the end of Section 3.1).

In short, the two-dimensional flux \mathbf{F}^* can be interpreted as follows. First, from the right states \mathbf{U}_{RD} and \mathbf{U}_{RU} in Figure 3.2, we build the associated state \mathbf{U}_R^* and the AVM vertical flux \mathbf{G}_R^* ; from these we define the transverse flux \mathbf{F}_R^* , which may be viewed as a flux in the x -direction involving the states \mathbf{U}_{RD} and \mathbf{U}_{RU} and the contributions \mathbf{G}_{RD} and \mathbf{G}_{RU} in the y -direction. Next, the flux \mathbf{F}_L^* is similarly defined from the left states \mathbf{U}_{LD} and \mathbf{U}_{LU} . Once $\mathbf{U}_L^*, \mathbf{U}_R^*, \mathbf{F}_L^*$ and \mathbf{F}_R^* have been constructed, we use them as building blocks to define \mathbf{F}^* as an AVM-type flux: $\mathbf{F}^* \equiv \mathbf{F}^{\text{AVM}}(\mathbf{U}_L^*, \mathbf{U}_R^*)$, where the fluxes $\mathbf{F}(\mathbf{U}_L^*)$ and $\mathbf{F}(\mathbf{U}_R^*)$ are substituted by \mathbf{F}_L^* and \mathbf{F}_R^* respectively. The AVM solver used is determined by the choice of the viscosity matrix \mathbf{Q}_x^* .

Similarly, the vertical flux $\mathbf{G}_{i,j+1/2}^*$ is constructed as follows. Consider the six-cell stencil around the horizontal edge between cells C_{ij} and $C_{i,j+1}$ (see Figure 3.1). Then:

- The 1D flux through the edge is simply given by

$$\mathbf{G}_{i,j+1/2}^* = \mathbf{G}^{\text{AVM}}(\mathbf{U}_{ij}, \mathbf{U}_{i,j+1}).$$

- To define the 2D flux $\mathbf{G}_{i+1/2,j+1/2}^*$ at the vertex $(x_{i+1/2}, y_{j+1/2})$, first compute the four local speeds of propagation S_L, S_R, S_U and S_D (Section 3.1).

Subsonic case in both directions: Consider the states

$$\begin{aligned} \mathbf{U}_D^* &= \mathbf{U}_x^{\text{HLL}}(\mathbf{U}_{ij}, \mathbf{U}_{i+1,j}, S_L, S_R), \\ \mathbf{U}_U^* &= \mathbf{U}_x^{\text{HLL}}(\mathbf{U}_{i,j+1}, \mathbf{U}_{i+1,j+1}, S_L, S_R), \end{aligned}$$

and the 1D AVM fluxes

$$\begin{aligned} \mathbf{F}_D^* &= \mathbf{F}^{\text{AVM}}(\mathbf{U}_{ij}, \mathbf{U}_{i+1,j}), \\ \mathbf{F}_U^* &= \mathbf{F}^{\text{AVM}}(\mathbf{U}_{i,j+1}, \mathbf{U}_{i+1,j+1}). \end{aligned}$$

Next, define the transverse flux \mathbf{G}_D^* in terms of \mathbf{U}_D^* and \mathbf{F}_D^* , and the transverse flux \mathbf{G}_U^* in terms of \mathbf{U}_U^* and \mathbf{F}_U^* . The 2D flux is then given by

$$\mathbf{G}_{i+1/2,j+1/2}^* = \frac{\mathbf{G}_D^* + \mathbf{G}_U^*}{2} - \frac{1}{2} \mathbf{Q}_y^*(\mathbf{U}_U^* - \mathbf{U}_D^*),$$

where \mathbf{Q}_y^* is a functional evaluation of the matrix \mathbf{A}_y^* (see Section 1.5).

Supersonic case in some or both directions: The flux $\mathbf{G}_{i+1/2,j+1/2}^*$ is defined as the corresponding upwinded flux (see [15, Sec. 2.1]).

- The 2D flux $\mathbf{G}_{i-1/2,j+1/2}^*$ at the left vertex is defined in an analogous way.
- Finally, the 1D and 2D fluxes are assembled as

$$\mathbf{G}_{i,j+1/2} = \frac{1}{6} \mathbf{G}_{i+1/2,j+1/2}^* + \frac{2}{3} \mathbf{G}_{i,j+1/2}^* + \frac{1}{6} \mathbf{G}_{i-1/2,j+1/2}^*.$$

Notice that the two-dimensional fluxes at vertices have been constructed as combinations of several one-dimensional fluxes. Thus, for practical implementation only the definition of a one-dimensional AVM flux is really needed, what makes the computer coding simple and clear.

Finally, the choice of the time step is given by the CFL condition

$$\Delta t = 2\delta \cdot \min_{ij} \Delta t_{ij}, \quad \delta \in (0, 1], \quad (3.2.9)$$

where each local time step Δt_{ij} is defined as

$$\Delta t_{ij} = \left(\frac{\lambda_{ij}^x}{\Delta x} + \frac{\lambda_{ij}^y}{\Delta y} \right)^{-1},$$

where λ_{ij}^α denotes the maximal speed of propagation in the cell C_{ij} in the α -direction, $\alpha = x, y$, taking into account both the speeds at the edges and those arising in the two-dimensional Riemann problems at vertices. As it was remarked in Section 3.1 (see also [14]), the two-dimensional contributions allow the use of a maximal CFL number δ of unity. This represents an additional advantage with respect to standard methods in which a one-dimensional solver is applied dimension by dimension, for which the maximum CFL number is 0.5.

3.3 Numerical results

The performances of the proposed multidimensional AVM schemes are analyzed in this section. To this end, we have chosen a number of tests that constitute standard references in MHD, in order that our schemes could be confronted with other methods in the literature.

An important issue in the construction of the schemes is the proper choice of the speeds of propagation. For this we have followed the guidelines stated in Section 3.1, with the expressions (2.1.3) for the maximal wave speeds. In particular, for the intermediate speeds $\bar{\lambda}_\alpha(\mathbf{U}_0, \mathbf{U}_1)$ appearing in (3.1.5) we have simply chosen the speeds $\lambda_\alpha(\tilde{\mathbf{U}})$ at the intermediate state $\tilde{\mathbf{U}} = \frac{\mathbf{U}_0 + \mathbf{U}_1}{2}$. Another possible but more expensive choice is to consider $\tilde{\mathbf{U}}$ as the Roe state associated to the states \mathbf{U}_0 and \mathbf{U}_1 . In our computational experiments, both choices have led to similar results. Once the maximal speeds of propagation have been computed, the time step is determined from the CFL condition (3.2.9). In particular, we remark that for the 2D methods the theoretical admissible CFL number is 1, while for the 1D×1D methods is 0.5.

The transverse fluxes (see Remark 8) can be defined using values from the associated intermediate state and the normal fluxes. Thus, denoting the components of the states (3.2.6) as $\mathbf{U}_\alpha^* = (u_{\alpha,i}^*)_{i=1}^8$ and the fluxes (3.2.7) as $\mathbf{G}_\alpha^* = (g_{\alpha,i}^*)_{i=1}^8$, for $\alpha = L, R$, the transverse fluxes \mathbf{F}_α^* are defined as

$$\mathbf{F}_\alpha^* = \begin{pmatrix} u_{\alpha,2}^* \\ g_{\alpha,3}^* + \frac{(u_{\alpha,2}^*)^2 - (u_{\alpha,3}^*)^2}{u_{\alpha,1}^*} + (u_{\alpha,6}^*)^2 - (u_{\alpha,5}^*)^2 \\ g_{\alpha,2}^* \\ \frac{u_{\alpha,2}^* u_{\alpha,4}^*}{u_{\alpha,1}^*} - u_{\alpha,5}^* u_{\alpha,7}^* \\ 0 \\ \frac{u_{\alpha,2}^* u_{\alpha,6}^* - u_{\alpha,3}^* u_{\alpha,5}^*}{u_{\alpha,1}^*} \\ \frac{u_{\alpha,2}^* u_{\alpha,7}^* - u_{\alpha,4}^* u_{\alpha,5}^*}{u_{\alpha,1}^*} \\ \frac{u_{\alpha,2}^*}{u_{\alpha,1}^*} (u_{\alpha,8}^* + g_{\alpha,3}^* - \frac{(u_{\alpha,3}^*)^2}{u_{\alpha,1}^*} + u_{\alpha,6}^{*2}) - \frac{u_{\alpha,5}^*}{u_{\alpha,1}^*} (u_{\alpha,2}^* u_{\alpha,5}^* + u_{\alpha,3}^* u_{\alpha,6}^* + u_{\alpha,4}^* u_{\alpha,7}^*) \end{pmatrix},$$

taking into account the form of the fluxes \mathbf{F} and \mathbf{G} . The transverse fluxes \mathbf{G}_β^* , for $\beta = D, U$, can be constructed in an analogous way from \mathbf{U}_β^* and \mathbf{F}_β^* .

The definition of a two-dimensional AVM flux for the nonconservative system (2.1.6) follows exactly the same guidelines explained at the end of Section 3.2, substituting $\mathbf{U}_x^{\text{HLL}}$, $\mathbf{U}_y^{\text{HLL}}$, \mathbf{F}^{AVM} and \mathbf{G}^{AVM} by $\tilde{\mathbf{U}}_x^{\text{HLL}}$, $\tilde{\mathbf{U}}_y^{\text{HLL}}$, $\tilde{\mathbf{F}}^{\text{AVM}}$ and $\tilde{\mathbf{G}}^{\text{AVM}}$ respectively.

We will denote as Int- n and Padé- $[m/k]$ the AVM methods based on the internal polynomial $p_n(x)$ and the Padé approximation $r_1^{[m/k]}(x)$, respectively. For each AVM method, 2D denotes its multidimensional version, where the numerical flux at an edge is given by (3.2.8); on the

other hand, $1D \times 1D$ refers to the corresponding one-dimensional solver applied dimension by dimension, which is equivalent to taking $\mathbf{F}_{i+1/2,j} = \mathbf{F}_{i+1/2,j}^*$ in (3.2.8).

Finally, unless otherwise stated, the divergence cleaning technique introduced in Section 2.1 will be used in all the numerical experiments.

3.3.1 First-order accuracy

To study the accuracy of the proposed schemes, we first consider a test from [139] for the ideal Euler equations with exact smooth solution given by

$$\rho(x, y, t) = 1 + 0.2 \sin(\pi(x + y - t/2)), \quad v_x = 1, \quad v_y = -1/2, \quad P = 1,$$

and adiabatic constant $\gamma = 1.4$. The computational domain is $[-1, 1] \times [-1, 1]$, with periodic boundary conditions in both directions. The schemes have been run until a final time $T = 4$, which corresponds to a full period of the wave.

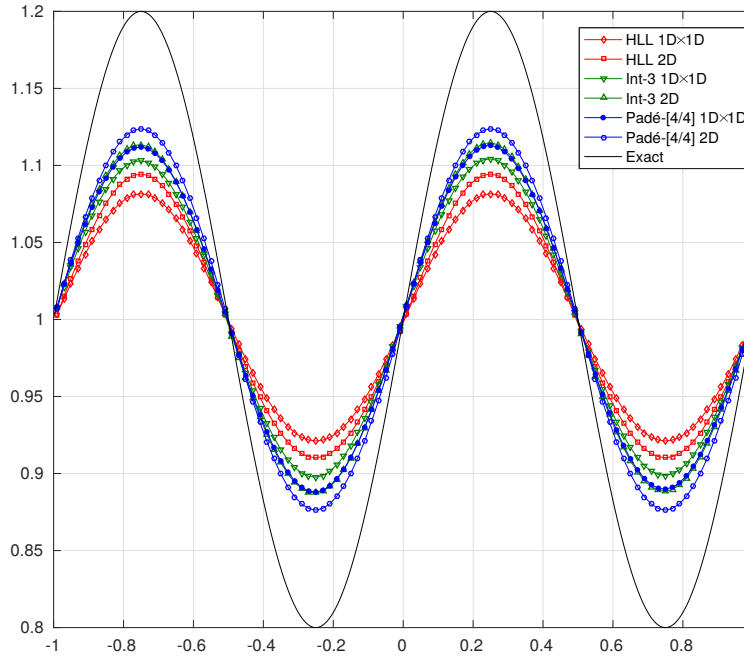


Figure 3.6: Test 3.3.1. Comparison of several $1D \times 1D$ - $2D$ AVM methods on a 100×100 mesh. Diagonal cut along the main diagonal for the density variable.

Figure 3.6 shows a cut along the main diagonal of the final densities obtained on a 100×100 mesh with the $1D \times 1D$ and $2D$ versions of HLL, Int-3 and Padé-[4/4] schemes. Notice that the Int-3 method is based on a polynomial of eighth degree, while the Padé-[4/4] method is based on a rational function of degree [8/8]. All the computations have been done using a common CFL number of 0.5, although the $2D$ methods perform well with CFL numbers up to 0.8. As

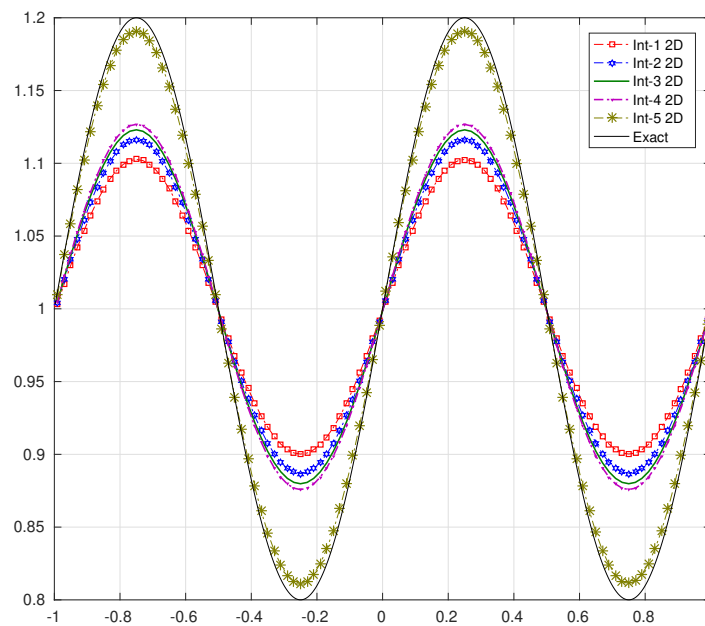


Figure 3.7: Test 3.3.1. Solutions obtained with the Int- n 2D schemes, for $n = 1, 2, 3, 4, 5$. Diagonal cut along the main diagonal for the density variable.

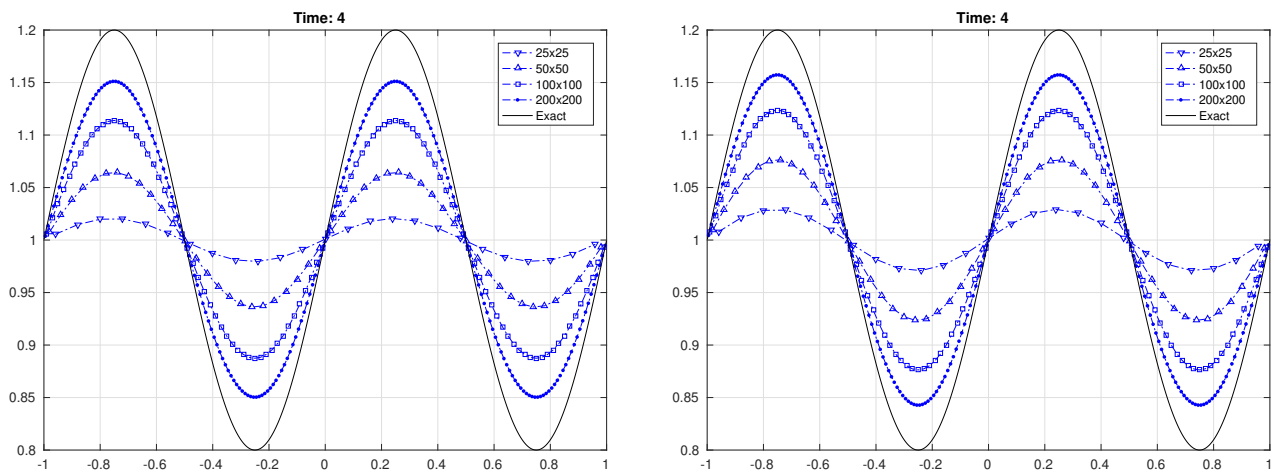


Figure 3.8: Test 3.3.1. Diagonal cut along the main diagonal for the density variable on several meshes. Left: Int-3 2D. Right: Padé-[4/4] 2D.

it can be observed, for each scheme the two-dimensional contributions provide a more precise computation of the solution than their one-dimensional counterparts.

Table 3.1: Test 3.3.1. Comparison of the L^1 -errors obtained with several 1D \times 1D and 2D AVM schemes.

Scheme	Mesh size	L^1 -error	Scheme	Mesh size	L^1 -error
HLL 1D \times 1D	25 \times 25	4.9299e-01	HLL 2D	25 \times 25	4.8377e-01
	50 \times 50	4.2625e-01		50 \times 50	4.0087e-01
	100 \times 100	3.0486e-01		100 \times 100	2.7504e-01
	200 \times 200	1.8684e-01		200 \times 200	1.6402e-01
Int-3 1D \times 1D	25 \times 25	4.6978e-01	Int-3 2D	25 \times 25	4.5315e-01
	50 \times 50	3.7274e-01		50 \times 50	3.4422e-01
	100 \times 100	2.4677e-01		100 \times 100	2.2038e-01
	200 \times 200	1.4393e-01		200 \times 200	1.2603e-01
Padé-[4/4] 1D \times 1D	25 \times 25	4.5418e-01	Padé-[4/4] 2D	25 \times 25	4.3194e-01
	50 \times 50	3.4670e-01		50 \times 50	3.1343e-01
	100 \times 100	2.2308e-01		100 \times 100	1.9403e-01
	200 \times 200	1.2791e-01		200 \times 200	1.0871e-01

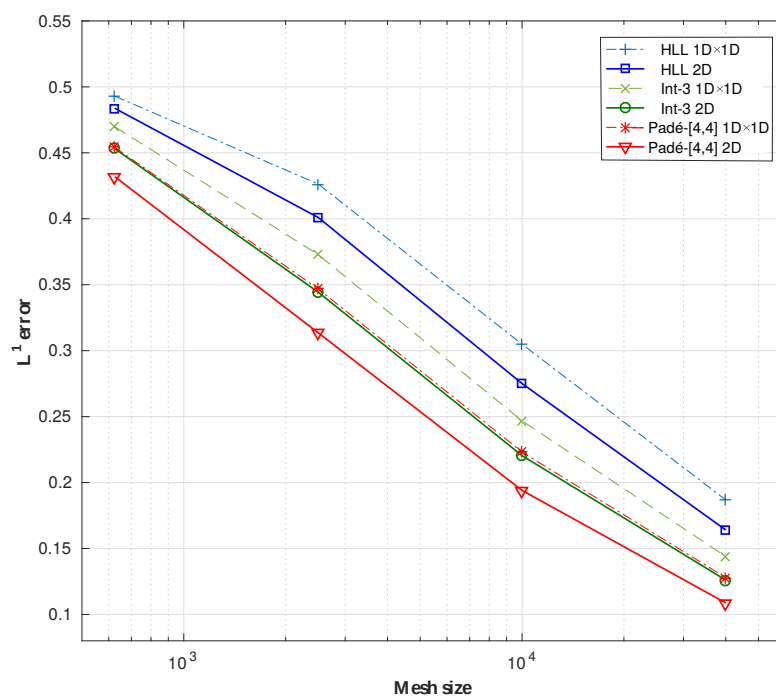
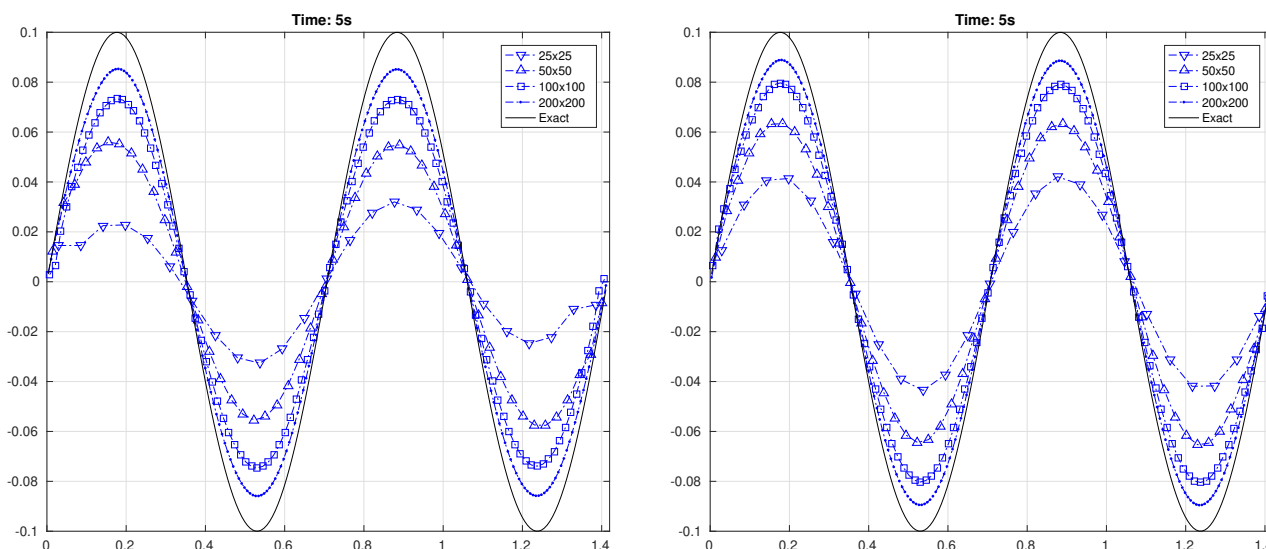
**Figure 3.9:** Test 3.3.1. Error curves for several 1D \times 1D and 2D AVM methods.

Table 3.2: Test 3.3.1 for MHD. Comparison of L^1 -errors for the components of the magnetic field.

Scheme	Mesh size	$\ B_{\parallel}\ _1$	$\ B_{\perp}\ _1$	$\ B_z\ _1$
HLL 2D	25×25	3.0669e-02	1.1424e-01	1.2152e-01
	50×50	2.4932e-02	9.3082e-02	1.0301e-01
	100×100	1.7197e-02	6.4162e-02	7.3016e-02
	200×200	1.1149e-02	4.1526e-02	4.7752e-02
Int-3 2D	25×25	2.3482e-02	8.3083e-02	7.4624e-02
	50×50	1.5832e-02	5.5190e-02	4.7937e-02
	100×100	9.3169e-03	3.2080e-02	2.7637e-02
	200×200	4.7735e-03	1.7490e-02	1.5038e-02
Padé-[4/4] 2D	25×25	1.8935e-02	7.0178e-02	7.2807e-02
	50×50	1.2102e-02	4.4361e-02	4.6467e-02
	100×100	6.8124e-03	2.4736e-02	2.6259e-02
	200×200	3.6967e-03	1.3130e-02	1.4008e-02

**Figure 3.10:** Test 3.3.1 for MHD. Diagonal cut along the main diagonal for the B_{\perp} variable on several meshes. Left: Int-3 2D. Right: Padé-[4/4] 2D.

To check the behavior of the solutions in terms of the basis function chosen for an AVM 2D method, we have represented in Figure 3.7 the results obtained with the Int- n 2D schemes, for $n = 1, 2, 3, 4, 5$. It is clear that the precision of the solutions increases as the degree of the basis polynomial $p_n(x)$ increases. A similar remark is also valid for the Padé- $[m/k]$ 2D schemes.

On the other hand, to check the convergence to the exact solution as the mesh is refined, in

Figure 3.8 are represented the solutions obtained with different meshes of size $N \times N$, for $N = 25, 50, 100, 200$. Table 3.1 shows the L^1 -errors in the density variable, while the corresponding error curves are depicted in Figure 3.9. Notice that the Int-3 2D scheme provides very similar result as the Padé-[4/4] 1D×1D method.

In order to verify the accuracy of our schemes for MHD flows, we consider now the rotated circularly polarized Alfvén wave test proposed in [196], for which an analytical solution is available. The computational domain is given by $[0, 1/\cos(\alpha)] \times [0, 1/\sin(\alpha)]$ for a given angle α , with periodic boundary conditions. As initial conditions we take $\rho = 1$, $v_{\parallel} = 0$, $p = 0.1$, $B_{\parallel} = 1$, $v_{\perp} = 0.1 \sin[2\pi(x \cos \alpha + y \sin \alpha)] = B_{\perp}$ and $v_z = 0.1 \cos[2\pi(x \cos \alpha + y \sin \alpha)] = B_z$, with adiabatic constant $\gamma = 5/3$. The simulations have been run until a final time $T = 5$ (corresponding to five full periods), using a CFL number of 0.8. Table 3.2 shows the L^1 -errors corresponding to the components of the magnetic field for an angle $\alpha = 45^\circ$, which are in good agreement with those presented in [196]. Finally, to check the convergence to the exact solution as the mesh is refined, Figure 3.10 shows the solutions obtained with different meshes of size $N \times N$, for $N = 25, 50, 100, 200$.

3.3.2 Orszag-Tang vortex

The Orszag-Tang vortex [158] is a well-known model of complex MHD flow containing many significant features. Departing from an initial smooth state, the system develops a series of complex interactions between the different shock waves generated as the system evolves in the transition to turbulence.

Specifically, we consider the initial conditions proposed in [100]:

$$\begin{aligned} \rho(x, y, 0) &= \gamma^2, & v_x(x, y, 0) &= -\sin(y), & v_y(x, y, 0) &= \sin(x), \\ B_x(x, y, 0) &= -\sin(y), & B_y(x, y, 0) &= \sin(2x), & P(x, y, 0) &= \gamma, \end{aligned}$$

with $\gamma = 5/3$. The computational domain is given by $[0, 2\pi] \times [0, 2\pi]$, with periodic boundary conditions in both directions.

We use this problem to test the robustness of several AVM 2D schemes equipped with the divergence cleaning technique introduced in Section 2.1. As it is well-known (see, e.g., [100]), negative pressures may appear if the divergence is not properly controlled. In order to compare, we have considered the HLL 2D, Int-3 2D and Padé-[4/4] 2D methods, with a common CFL number of 0.8.

Figure 3.11 shows the densities and pressures obtained at time $T = \pi$. Although there is no accepted reference solution for this test, our results are in good agreement with those found in the literature: see, e.g., [14, 100, 138]. As it can be observed, the Int-3 2D and Padé-[4/4] 2D schemes provide a sharper resolution of the solution than Balsara's HLL 2D method. This can also be seen in Figure 3.12, where cuts along the main diagonal at different times are shown. As time evolves and the solution becomes more complex, the Padé-[4/4] scheme provides significantly the best approximations, followed by Int-3 2D.

On the other hand, we have compared the technique for imposing the divergence constraint defined in Section 2.1 with the well-known projection method introduced in [30]. Figure 3.13 shows cuts along the main diagonal of the solutions obtained with the Int-3 2D scheme at time $T = \pi$ with both divergence cleaning techniques. As it can be observed, the results are

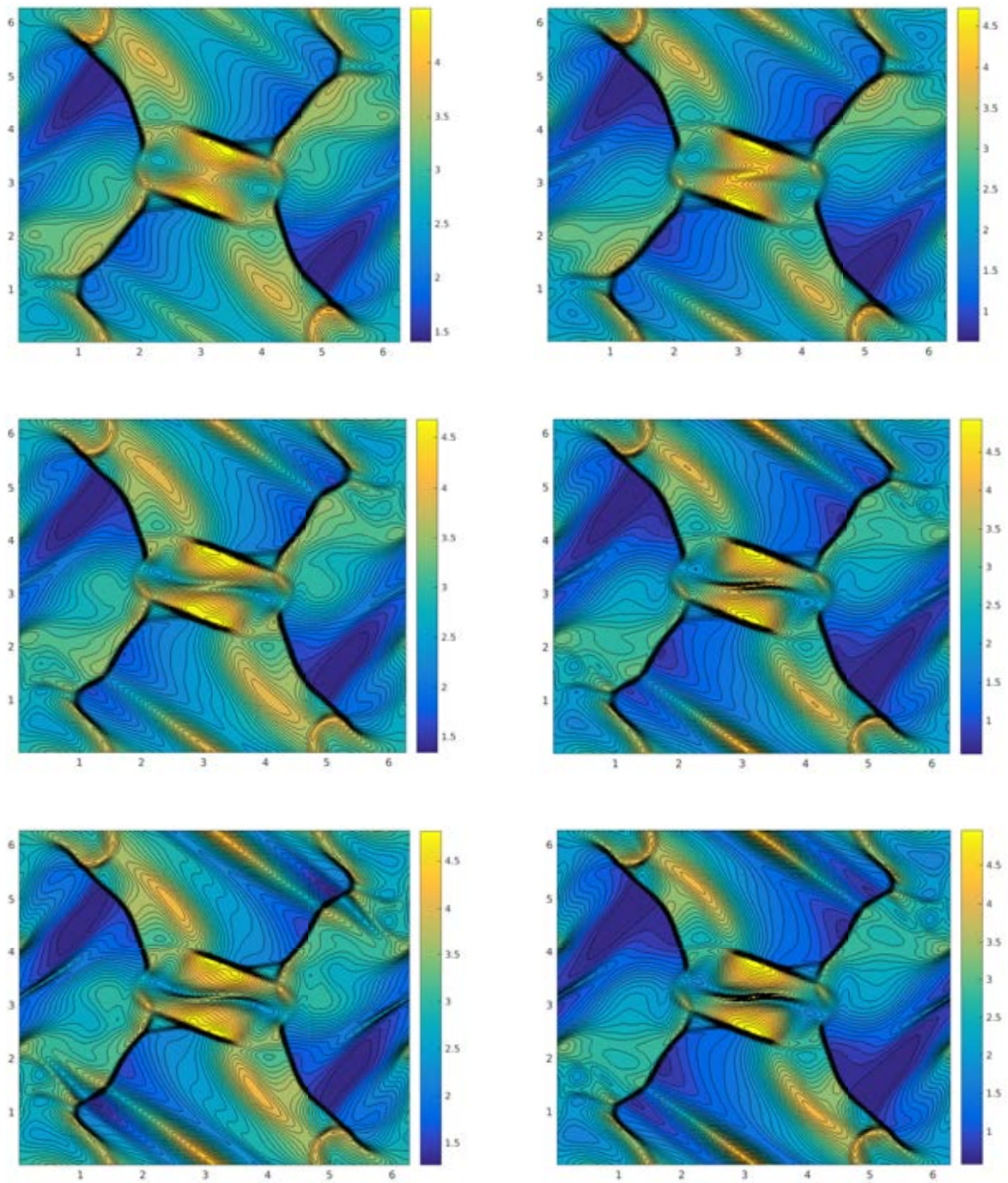


Figure 3.11: Test 3.3.2. Density (left) and pressure (right) computed at time $T = \pi$ on a 200×200 mesh. From top to bottom: HLL 2D, Int-3 2D and Padé-[4/4] 2D.

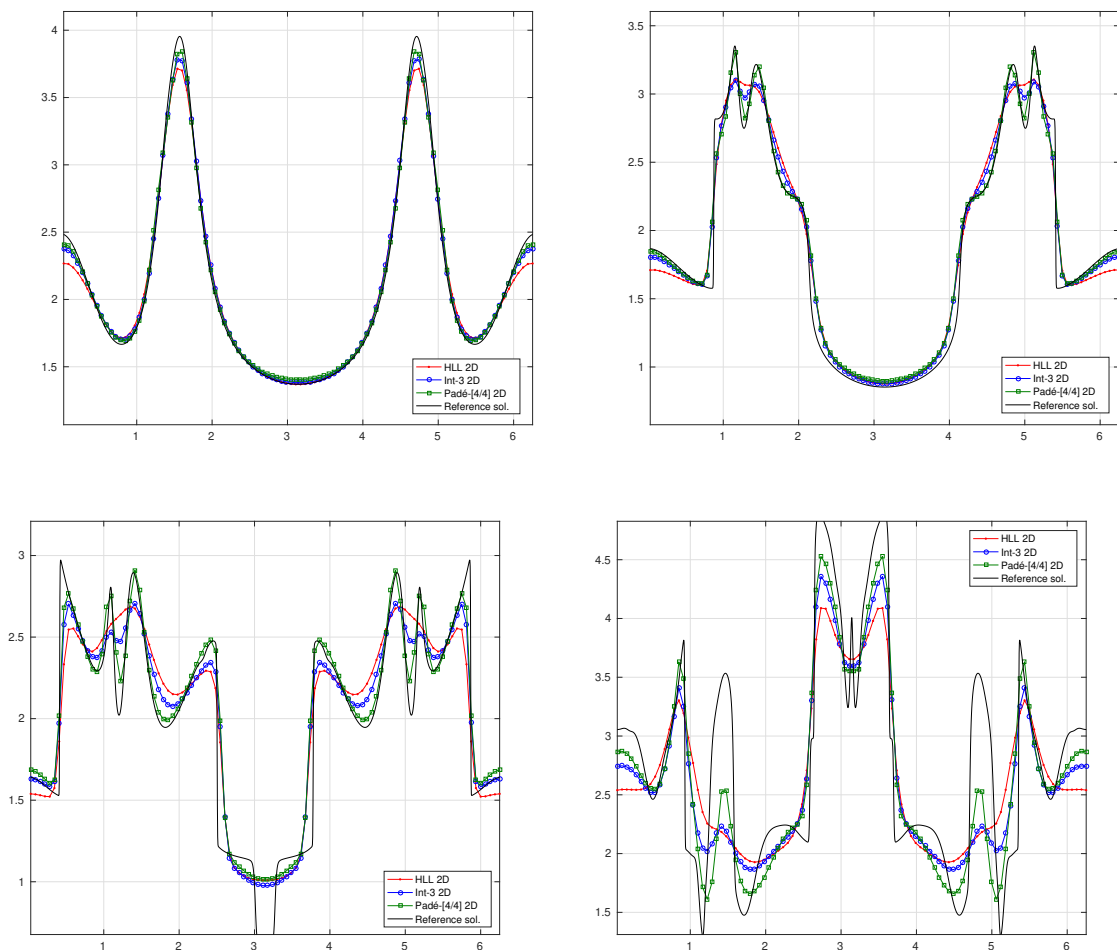


Figure 3.12: Test 3.3.2. Cuts along the main diagonal of the density at times $t = 1$, $t = 1.5$, $t = 2$ and $t = \pi$.

comparable. Also, Table 3.3 shows the relative CPU times obtained when using both divergence cleaning methods. The results are again comparable, although our method seems to be faster.

As it was remarked in [100], the numerical solutions for this test may have non-zero divergence, possibly due to the fact that the periodic boundary conditions reintroduce any non-zero divergence that exits through the domain boundary. However, these divergence errors decrease as the mesh is refined, which may be a reason for the stability of the schemes. To test the accuracy of a scheme it is usual to compute the maximum pressure, as no analytical solution is available. Table 3.4 shows the maximum pressure and the L^1 -norm of the divergence of the magnetic field at the final time $T = \pi$. In particular, it can be observed that in fact the divergence of the magnetic field decreases when the mesh is refined. The obtained results are in good agreement with those presented in [100, Table 2] for their schemes based on the

Table 3.3: Test 3.3.2. Relative CPU times. Comparison between the divergence cleaning technique in Section 2.1 (left) and the projection method (right) on several meshes.

Mesh size	Scheme	CPU	CPU (projection)
100×100	HLL 2D	1.00	1.27
	Int-3 2D	5.02	5.29
	Padé-[4/4] 2D	9.40	9.91
200×200	HLL 2D	1.00	1.74
	Int-3 2D	5.13	5.62
	Padé-[4/4] 2D	9.78	10.50
300×300	HLL 2D	1.00	1.53
	Int-3 2D	5.72	5.94
	Padé-[4/4] 2D	10.37	10.70
400×400	HLL 2D	1.00	1.35
	Int-3 2D	5.22	5.40
	Padé-[4/4] 2D	9.86	10.32
500×500	HLL 2D	1.00	1.21
	Int-3 2D	5.16	5.27
	Padé-[4/4] 2D	9.84	10.14

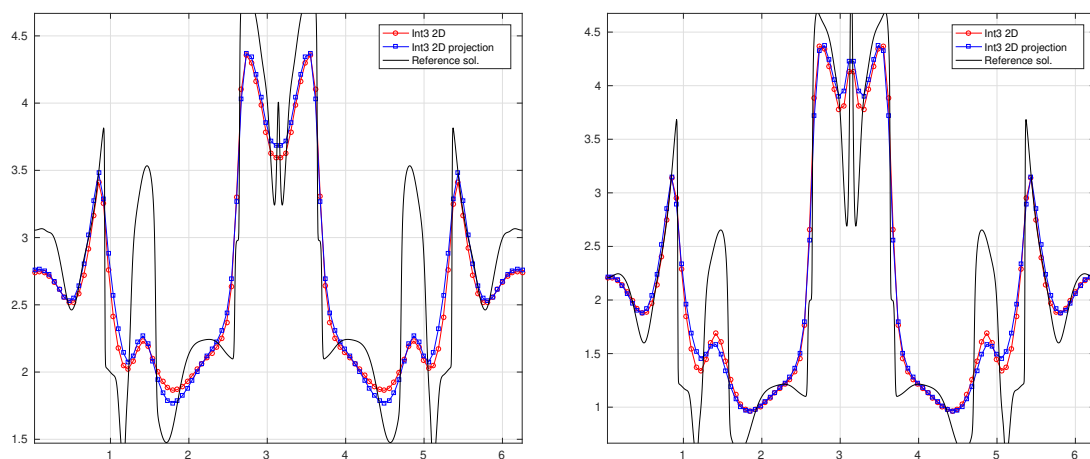


Figure 3.13: Test 3.3.2. Comparison of the divergence cleaning technique of Section 2.1 with the projection method using the Int-3 2D scheme. Cut along the main diagonal: density (left) and pressure (right) computed at time $T = \pi$ on a 100×100 mesh.

nonconservative form of the MHD equations.

Table 3.4: Test 3.3.2. Maximum pressure (top) and L^1 norm of the divergence of the magnetic field (bottom) at time $T = \pi$ for the Orszag-Tang vortex problem.

Mesh size	HLL 2D	Int-3 2D	Padé-[4/4] 2D
25×25	3.381	3.405	3.421
50×50	3.720	3.974	4.250
100×100	4.346	4.681	4.848
200×200	4.833	5.051	5.084
400×400	5.144	5.283	5.400
800×800	5.382	5.780	5.872
25×25	1.324	3.456	6.592
50×50	1.749	3.579	6.691
100×100	1.821	3.172	5.985
200×200	1.690	2.766	5.192
400×400	1.538	2.416	4.261
800×800	1.399	1.902	3.277

3.3.3 The rotor problem

The rotor problem, initially proposed in [22], is considered in this section. The initial condition consists of a dense rotating disk at the center of the domain, which is connected by means of a taper function with the ambient fluid at rest. The rotor is not in equilibrium, as the centrifugal forces are not balanced. Eventually, the rotating dense fluid will be confined into an oblate shape due to the action of the magnetic field.

Given $r_0 = 0.1$, $r_1 = 0.115$, $f = (r_1 - r)/(r_1 - r_0)$ and $r = [(x - 0.5)^2 + (y - 0.5)^2]^{1/2}$, the initial conditions are given by

$$(\rho, v_x, v_y) = \begin{cases} (10, -(y - 0.5)/r_0, (x - 0.5)/r_0) & \text{if } r < r_0, \\ (1 + 9f, -(y - 0.5)f/r, (x - 0.5)f/r) & \text{if } r_0 < r < r_1, \\ (1, 0, 0) & \text{if } r > r_1, \end{cases}$$

with $B_x = 2.5/\sqrt{4\pi}$, $B_y = 0$ and $P = 0.5$. We take $\gamma = 5/3$. The computational domain is $[0, 1] \times [0, 1]$ and transmissive boundary conditions are considered.

Figure 3.14 shows the solution obtained with the Padé-[4/4] 2D scheme at time $T = 0.295$ on a 200×200 mesh with CFL= 0.8. The results are in good agreement with those presented in [22, 50, 138, 196]. In particular, it is worth noticing that the results produced with the first-order Padé-[4/4] 2D scheme have a similar quality as those obtained in [50] with a one-dimensional third-order approximate Chebyshev-DOT scheme on the same mesh.

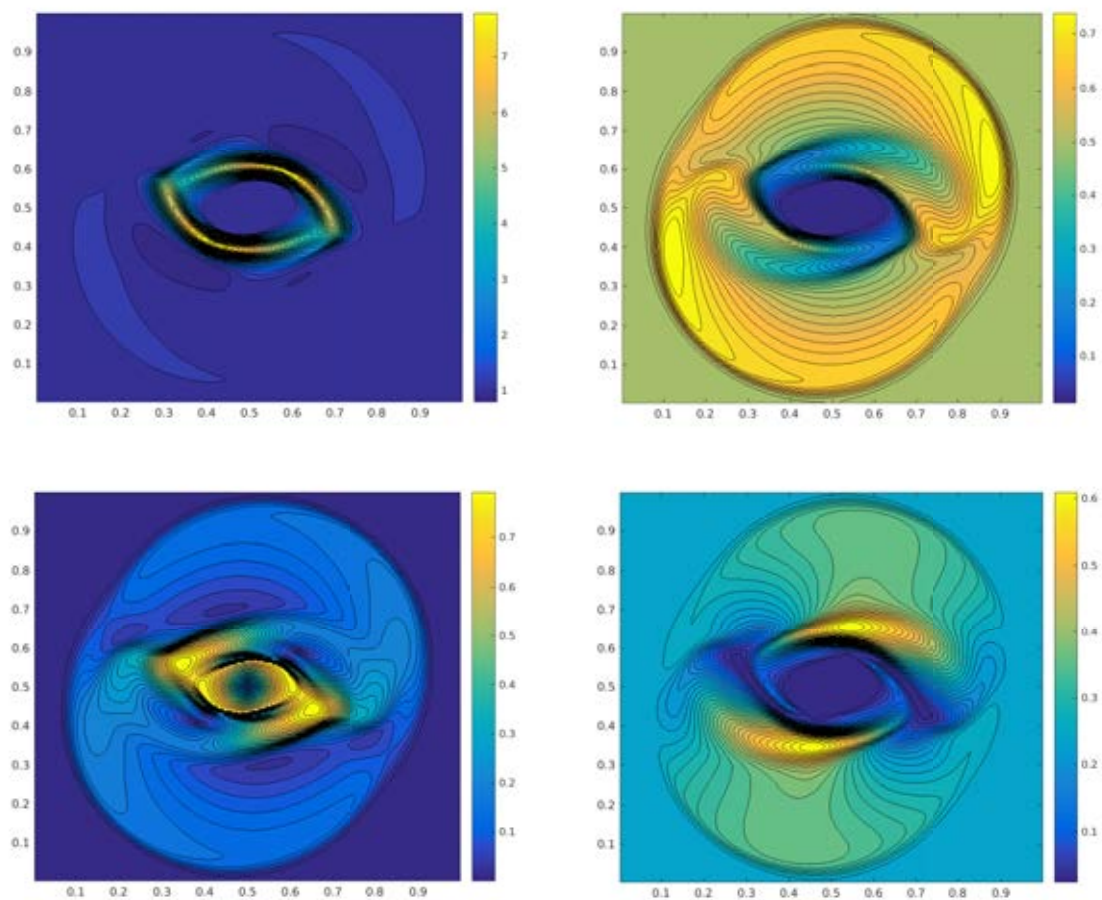


Figure 3.14: Test 3.3.3. Density ρ (top left), pressure P (top right), Mach number $|\mathbf{v}|/a$ (bottom left) and magnetic pressure $|\mathbf{B}|^2/2$ (bottom right) computed at time $T = 0.295$. Results obtained with the Padé-[4/4] 2D scheme with 200×200 cells.

3.3.4 Two-dimensional Riemann problem

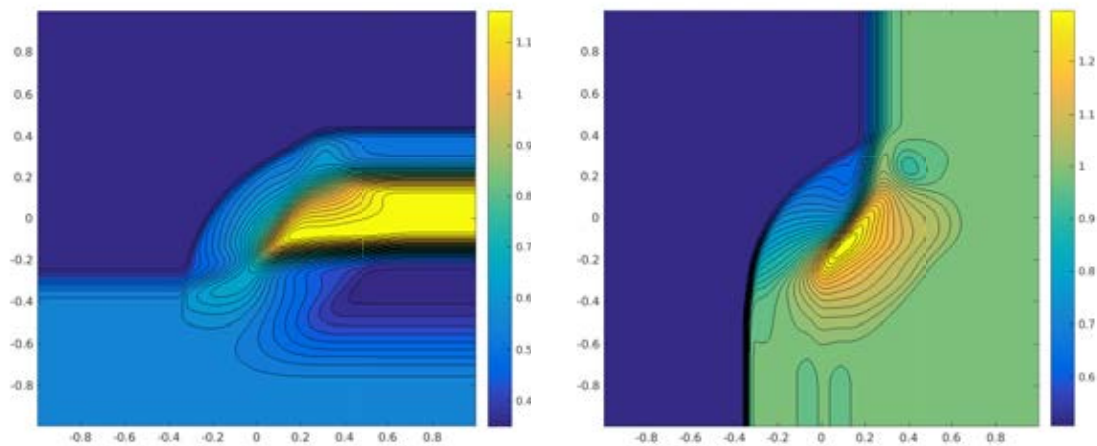
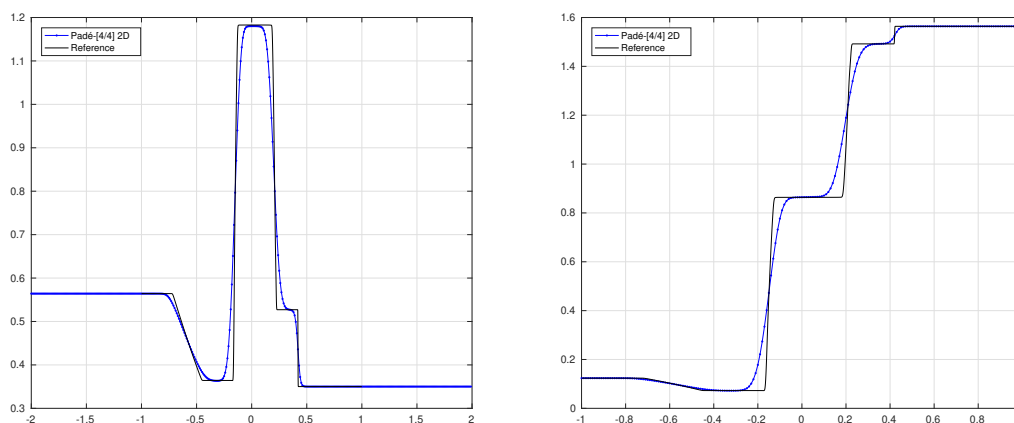
In this section we consider a two-dimensional Riemann problem originally proposed in [68]. The initial condition, given in Table 3.5, is chosen so that the solutions of three of the four one-dimensional Riemann problems are simple waves. Denoting the quadrants by Roman numbers as in Table 3.5, there is a rarefaction wave for problem I \leftrightarrow II and shocks for II \leftrightarrow III and III \leftrightarrow IV.

The problem has been solved in the computational domain $[-1, 1] \times [-1, 1]$ using a 200×200 mesh until a final time $T = 0.2$, with $\gamma = 5/3$. Following [138], we have considered Neumann boundary conditions.

Figure 3.15 shows the contours of B_x and B_y computed with the Padé-[4/4] 2D scheme using a CFL of 0.8; for this test, the Int-3 2D scheme gives similar results. It was reported in [68] that some schemes present problems to keep B_y constant across the shock in II \leftrightarrow III, and they can also produce strong distortions in B_x and B_y behind the rarefaction wave in I \leftrightarrow II. As it can be

Table 3.5: Test 3.3.4. Initial data for the 2d Riemann problem.

Quadrant	ρ	ρv_x	ρv_y	ρv_z	B_x	B_y	B_z	E
I: $x > 0, y > 0$	0.9308	1.4557	-0.4633	0.0575	0.3501	0.9830	0.3050	5.0838
II: $x < 0, y > 0$	1.0304	1.5774	-1.0455	-0.1016	0.3501	0.5078	0.1576	5.7813
III: $x < 0, y < 0$	1.0000	1.7500	-1.0000	0.0000	0.5642	0.5078	0.2539	6.0000
IV: $x > 0, y < 0$	1.8887	0.2334	-1.7422	0.0733	0.5642	0.9830	0.4915	12.999

**Figure 3.15:** Test 3.3.4. Contours at time $T = 0.2$ obtained with the Padé-[4/4] 2D scheme on a 200×200 mesh. Left: B_x . Right: B_y .**Figure 3.16:** Test 3.3.4. Cuts at $x = 0.93$ and $T = 0.2$. Solid line: reference solution. Dots: Padé-[4/4] 2D. Left: B_x . Right: v_x .

observed in Figure 3.15, none of those pathologies appear in the computed solutions, that can be directly compared with those presented in [68, 138].

Finally, Figure 3.16 shows a comparison between the solution of the two-dimensional Riemann problem obtained with the Padé-[4/4] 2D scheme and a one-dimensional reference solution. This serves to compare the quality of the solution of the I \leftrightarrow IV Riemann problem at $x = 0.93$, in a similar way as it was done in [68, 138].

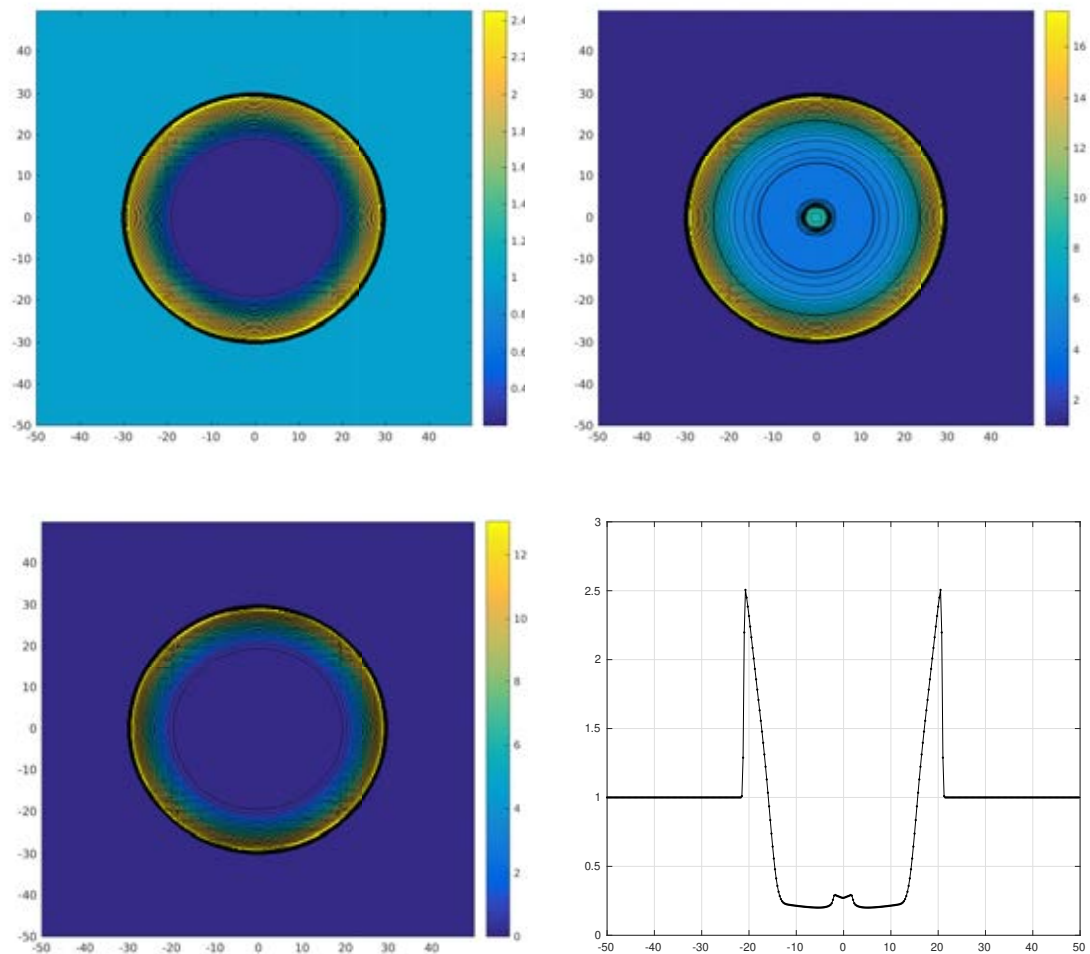


Figure 3.17: Test 3.3.5. Solution obtained with initial magnetic field $(B_x, B_y, B_z) = (0, 0, 0)$. Density ρ (top left), pressure P (top right), kinetic energy (bottom left) and density along $y = 0.5$ (bottom right) computed at time $T = 3$.

3.3.5 Spherical explosion

This test concerns the evolution of an explosion produced by an overpressured spherical region located at the center of the domain. It was originally proposed in [212], although we have

considered here the data proposed in [186]. Specifically, the initial pressure is $P = 100$ inside a circle of radius $r = 10$ and $P = 1$ outside. The density is $\rho = 1$ on the whole domain and the velocity field is set to zero. As in [186], three different values of the initial magnetic field are considered, with a common value of the adiabatic constant $\gamma = 2$. The computations have been performed with the Padé-[4/4] 2D scheme on the computational domain $[-50, 50] \times [-50, 50]$, using a 400×400 uniform mesh and CFL=0.8.

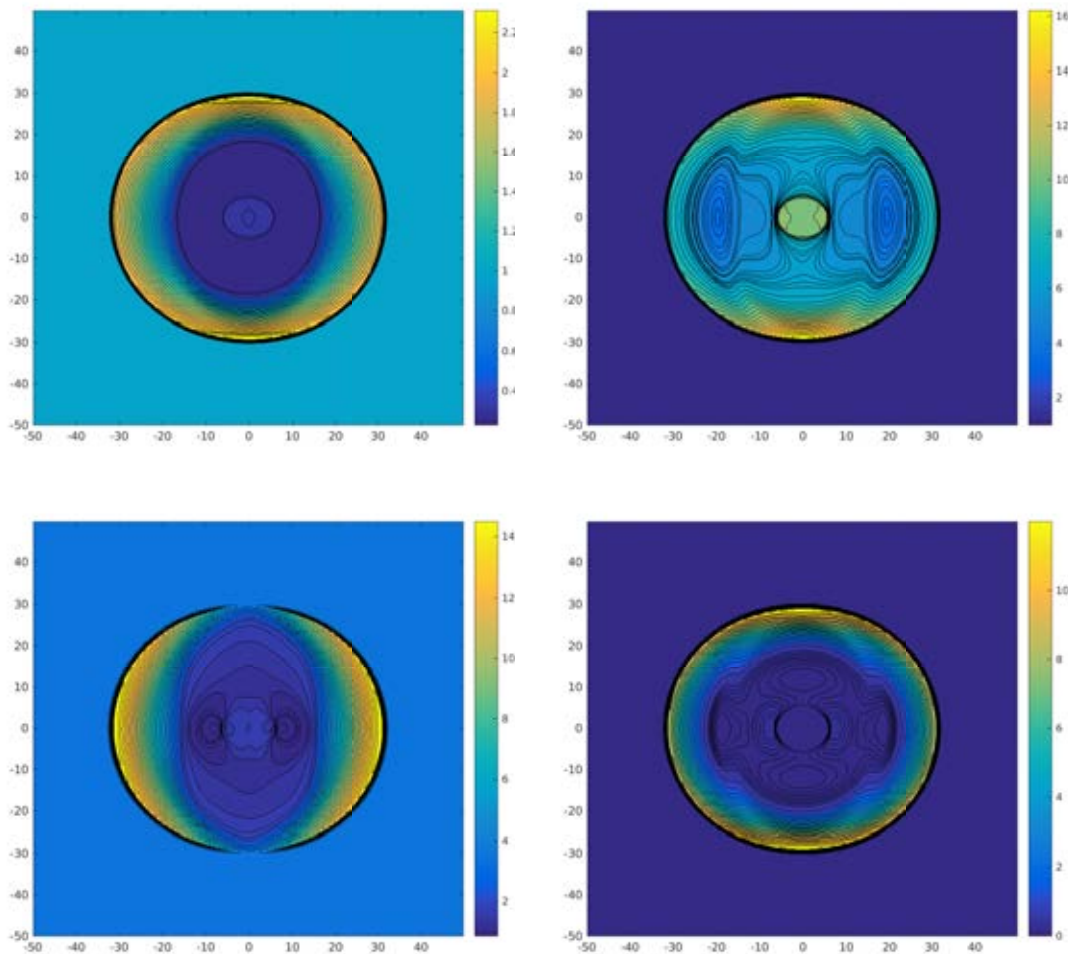


Figure 3.18: Test 3.3.5. Solution obtained with initial magnetic field $(B_x, B_y, B_z) = (0, 5/\sqrt{\pi}, 0)$. Density ρ (top left), pressure P (top right), magnetic pressure (bottom left) and kinetic energy (bottom right) computed at time $T = 3$.

Taking the initial magnetic field as $(B_x, B_y, B_z) = (0, 0, 0)$ the solution propagates symmetrically in the radial direction, as expected for the Euler equations. This behavior can be seen in Figure 3.17, which shows the results of the simulation at time $T = 3$. If the initial magnetic field is taken as $(B_x, B_y, B_z) = (0, 5/\sqrt{\pi}, 0)$, the increasing of the strength of the magnetic field produces a slightly elongated shock in the y -direction: see Figure 3.18.

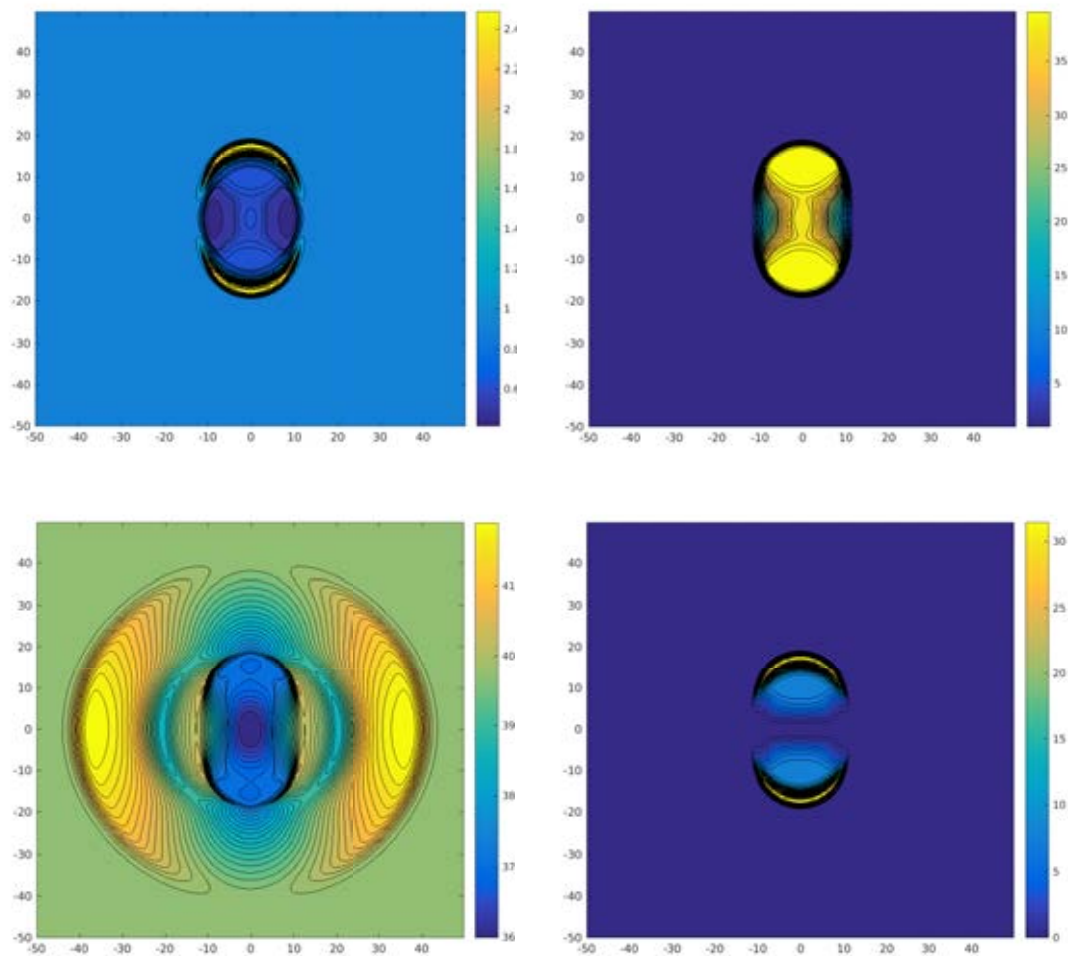


Figure 3.19: Test 3.3.5. Solution obtained with initial magnetic field $(B_x, B_y, B_z) = (0, 50/\sqrt{\pi}, 0)$. Density ρ (top left), pressure P (top right), magnetic pressure (bottom left) and kinetic energy (bottom right) computed at time $T = 1.05$.

For an even stronger initial magnetic field $(B_x, B_y, B_z) = (0, 50/\sqrt{\pi}, 0)$, the solution becomes highly anisotropic, with essentially no displacement of fluid in the orthogonal direction to the magnetic field. Figure 3.19 shows the results obtained at time $T = 1.05$, which is the time considered in [186] for which the perturbation has not yet reached the boundary.

In all the three cases so far considered, our results are in good agreement with those presented in [186, 212].

Chapter 4

Multidimensional AVM-type solvers: the nonconservative case

The extension of the multidimensional Riemann solvers seen in Chapter 3 from the conservative to the nonconservative case is not straightforward. However, the form in which the two-dimensional AVM solvers have been defined opens the door to extend them to the case of hyperbolic systems in nonconservative form, within the framework of path-conservative schemes (see Section 1.4). In particular, this allows to extend the two-dimensional AVM solvers to the case of hyperbolic systems with coupling and source terms. A possible way to do this is presented in this chapter, whose results have been presented in [178].

In this Chapter, we use the path-conservative formalism in order to derive a general strategy to construct two-dimensional Riemann solvers for nonconservative hyperbolic systems on structured meshes. In particular, we define second-order genuinely two-dimensional AVM-type Riemann solvers for general hyperbolic systems. Our schemes are based on the rectangular four-waves model proposed in [14] (see Section 3.1), where the differences of fluxes in the conservative formulation are replaced by fluctuations along the paths considered in the definition of the nonconservative products. Initially, we consider the extension of a 2D HLL scheme, which is then rewritten as a 2D scheme in PVM form than can be readily extended to a general 2D AVM-type scheme. We prove the consistency and linear L^∞ -stability for the resultant first-order schemes. Second-order accuracy in space and time is obtained by means of a predictor-corrector MUSCL-Hancock procedure. Applications to the one-layer and two-layer shallow water equations have been considered. The schemes have been proved to be well-balanced for water at rest solutions. An additional advantage of the proposed two-dimensional solvers is that they are stable up to a CFL number of unity, whereas the maximal CFL for schemes based on dimensional splitting is usually 0.5.

Section 4.1 is devoted to building genuinely multidimensional Riemann solvers on structured meshes for general hyperbolic systems. As a particular case, we derive a simple multidimensional HLL solver in Section 4.2. Then, in Section 4.3 we introduce AVM-type Riemann solvers for two-dimensional nonconservative systems. In particular, a study of linear stability is presented in Section 4.3.3 and a second-order extension in Section 4.3.4. A number of numerical experiments for shallow water systems are presented in Section 4.4 in order to verify the accuracy, well-balancedness and performances of the proposed numerical schemes.

4.1 General framework

In this section we derive a general strategy to construct genuinely multidimensional Riemann solvers on structured meshes. To this end, we will follow the ideas in [14, 15] to introduce multidimensional effects through the solution of two-dimensional Riemann problems at vertices.

To make the exposition clearer, we first focus on a purely conservative system:

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) + \partial_y \mathbf{G}(\mathbf{U}) = 0. \quad (4.1.1)$$

where \mathbf{U} , \mathbf{F} and \mathbf{G} are defined in (1.2.1). The initial condition for the two-dimensional Riemann problem at a given point consists of four constant states denoted as \mathbf{U}_{RU} , \mathbf{U}_{LU} , \mathbf{U}_{LD} and \mathbf{U}_{RD} : see Figure 3.2 (left). It is known that this problem has a self-similar solution consisting of four directional one-dimensional Riemann problems arising at common edges, along with a region of strong interaction where complex structures appear ([128]). However, when designing numerical methods it is impractical to consider all the detailed structure arising in the strong interaction region, so this is usually averaged into a single constant state (see, e.g., [14, 15, 203, 104]). Here, we will follow the approach in [15], in which the strong interaction region is assumed to be rectangular and limited by the maximal speeds of propagation.

Following the four-waves model presented in Section 3.1, we assume that the strong interaction region arising around the vertex is bounded by the maximal horizontal wave speeds (3.1.6) and the maximal vertical speeds (3.1.7) (see also Figure 3.3). For the sake of clarity we will denote $\mathbf{F}_{\nu\mu} = \mathbf{F}(\mathbf{U}_{\nu\mu})$ and $\mathbf{G}_{\nu\mu} = \mathbf{G}(\mathbf{U}_{\nu\mu})$, for $\nu \in \{L, R\}$, $\mu \in \{D, U\}$.

Assume first that we are in the subsonic case in both directions, that is, $S_L < 0 < S_R$ and $S_D < 0 < S_U$. In this case, the wave structure of the solution of the two-dimensional Riemann problem is assumed to be of the form represented in Figure 4.1, which consists of a central strongly interacting state surrounded by the solutions of the one-dimensional Riemann problems arising at edges.

Consider now the similarity variables $\xi = x/t$ and $\psi = y/t$, and denote as $\mathbf{R}_x(\xi; \mathbf{U}, \mathbf{V})$ the self-similar solution of the one-dimensional Riemann problem in the x -direction with initial data given by arbitrary states \mathbf{U} and \mathbf{V} ; similarly, $\mathbf{R}_y(\psi; \mathbf{U}, \mathbf{V})$ denotes the solution of the one-dimensional Riemann problem in the vertical direction. Finally, the solution of the two-dimensional Riemann problem in similarity variables is denoted as $\mathbf{U}(x, y, t) \equiv \mathbf{R}(\xi, \psi)$.

Integration of (4.1.1) over a space-time domain $\mathcal{Q} \times [0, T]$ gives

$$\int_{\mathcal{Q}} (\mathbf{U}(x, y, T) - \mathbf{U}_0) dx dy = - \int_0^T \int_{\partial \mathcal{Q}} \mathbf{F} \cdot \mathbf{n} d\gamma dt,$$

where $\mathbf{U}_0 \equiv \mathbf{U}_0(x, y, 0)$ is the initial condition of the Riemann problem. In particular, taking $\mathcal{Q} = [S_L T, S_R T] \times [S_D T, S_U T]$ and denoting $\partial \mathcal{Q} = \cup_{i=1}^4 \Gamma_i$ (see Figure 4.1), we get

$$\begin{aligned} & \int_{S_L T}^{S_R T} \int_{S_D T}^{S_U T} (\mathbf{U}(x, y, t) - \mathbf{U}_0) dx dy = \\ & - \int_0^T \left\{ - \int_{\Gamma_1} \mathbf{G}(\mathbf{U}(x, S_D T, t)) dx + \int_{\Gamma_2} \mathbf{F}(\mathbf{U}(S_R T, y, t)) dy \right. \\ & \left. + \int_{\Gamma_3} \mathbf{G}(\mathbf{U}(x, S_U T, t)) dx - \int_{\Gamma_4} \mathbf{F}(\mathbf{U}(S_L T, y, t)) dy \right\} dt. \end{aligned}$$

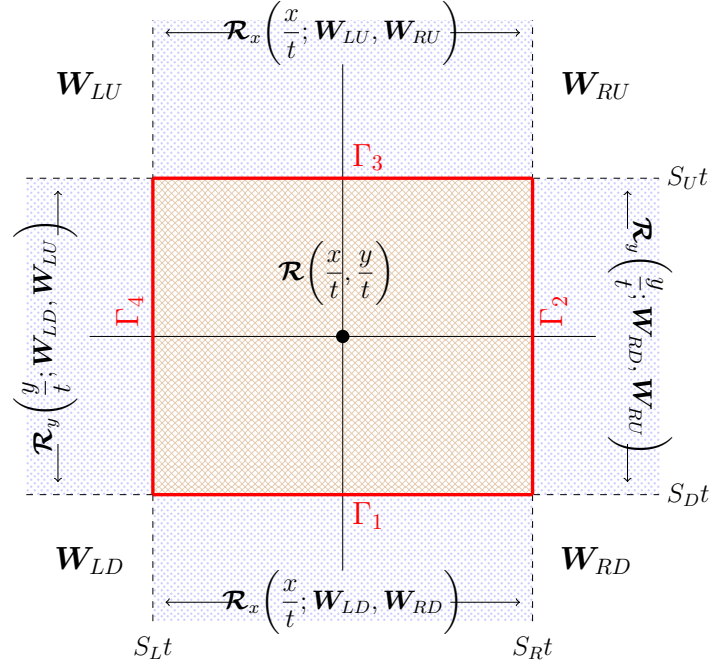


Figure 4.1: Structure of the solution of the 2D Riemann problem for the nonconservative system (5.1.1). Subsonic case.

Taking into account the solution of the 1D Riemann problem between \mathbf{U}_{LD} and \mathbf{U}_{RD} , we have

$$\begin{aligned} \int_0^T \int_{\Gamma_1} \mathbf{G}(\mathbf{U}(x, S_D T, t)) dx dt &= \int_0^T \left\{ \int_{S_L T}^{S_L t} \mathbf{G}_{LD} dx \right. \\ &\quad \left. + \int_{S_L t}^{S_R t} \mathbf{G}(\mathcal{R}_x\left(\frac{x}{t}; \mathbf{U}_{LD}, \mathbf{U}_{RD}\right)) dx + \int_{S_R t}^{S_R T} \mathbf{G}_{RD} dx \right\} dt \\ &= \frac{T^2}{2} \left(-S_L \mathbf{G}_{LD} + \int_{S_L}^{S_R} \mathbf{G}(\mathcal{R}_x(\xi; \mathbf{U}_{LD}, \mathbf{U}_{RD})) d\xi + S_R \mathbf{G}_{RD} \right), \end{aligned}$$

and similarly for the integrals over the remaining edges. Finally, choosing $T = 1$, we deduce the following *consistency condition* in similarity variables:

$$\begin{aligned} \int_{S_L}^{S_R} \int_{S_D}^{S_U} (\mathbf{R}(\xi, \psi) - \mathbf{U}_0(\xi, \psi)) d\xi d\psi &= \frac{S_D(\mathbf{F}_{RD} - \mathbf{F}_{LD}) - S_U(\mathbf{F}_{RU} - \mathbf{F}_{LU})}{2} \\ &\quad - \frac{1}{2} \int_{S_D}^{S_U} (\mathbf{F}(\mathcal{R}_y(\psi; \mathbf{U}_{RD}, \mathbf{U}_{RU})) - \mathbf{F}(\mathcal{R}_y(\psi; \mathbf{U}_{LD}, \mathbf{U}_{LU}))) d\psi \\ &\quad + \frac{S_L(\mathbf{G}_{LU} - \mathbf{G}_{LD}) - S_R(\mathbf{G}_{RU} - \mathbf{G}_{RD})}{2} \\ &\quad - \frac{1}{2} \int_{S_L}^{S_R} (\mathbf{G}(\mathcal{R}_x(\xi; \mathbf{U}_{LU}, \mathbf{U}_{RU})) - \mathbf{G}(\mathcal{R}_x(\xi; \mathbf{U}_{LD}, \mathbf{U}_{RD}))) d\xi \end{aligned} \quad (4.1.2)$$

where $U_0(\xi, \psi) = U(\xi, \psi, 0)$.

Remark 9. In [15] Balsara considered an HLL-type 2D solver for (4.1.1), in which the 1D solver R_x is the usual HLL solver defined by the HLL state U_α^* and the HLL flux F_α^* ($\alpha = L, R$), and similarly for R_y . Moreover, the strongly interaction area was assumed to be constant: $R(\xi, \psi) \equiv U^*$. In this case, condition (4.1.2) allows us to explicitly determine the resolved state U^* , which in turn coincides with the one obtained in [15] (see also Section 3.1).

Let us now focus on the nonconservative case

$$\partial_t \mathbf{W} + \mathcal{A}_x(\mathbf{W}) \partial_x \mathbf{W} + \mathcal{A}_y(\mathbf{W}) \partial_y \mathbf{W} = 0. \quad (4.1.3)$$

where \mathbf{W} , \mathcal{A}_x and \mathcal{A}_y are defined in (1.3.2). Following [160], condition (4.1.2) can be extended in a natural way by expressing the differences of fluxes as integrals along the paths Φ considered in the definition of the nonconservative products. For example, writing

$$\mathbf{F}_{RD} - \mathbf{F}_{LD} = \int_0^1 \frac{d\mathbf{F}}{d\mathbf{U}}(\Phi_{LD,RD}) \frac{d}{ds} \Phi_{LD,RD} ds,$$

(recall the notation $\Phi_{a,b}$ from (1.3.6)) the Jacobian $\frac{\partial \mathbf{F}}{\partial \mathbf{U}}$ could be substituted by $\mathcal{A}_x(\Phi_{LD,RD})$, and similarly for the remaining terms in (4.1.2). This procedure leads to the following consistency condition for the nonconservative system (5.1.1):

$$\begin{aligned} & \int_{S_L}^{S_R} \int_{S_D}^{S_U} (\mathcal{R}(\xi, \psi) - \mathbf{W}_0(\xi, \psi)) d\xi d\psi = \\ & \frac{S_D}{2} \int_0^1 \mathcal{A}_x(\Phi_{LD,RD}) \frac{d}{ds} \Phi_{LD,RD} ds - \frac{1}{2} \int_{S_D}^{S_U} \int_0^1 \mathcal{A}_x(\Phi_{L\psi,R\psi}) \frac{d}{ds} \Phi_{L\psi,R\psi} ds \\ & - \frac{S_U}{2} \int_0^1 \mathcal{A}_x(\Phi_{LU,RU}) \frac{d}{ds} \Phi_{LU,RU} ds \\ & + \frac{S_L}{2} \int_0^1 \mathcal{A}_y(\Phi_{LD,LU}) \frac{d}{ds} \Phi_{LD,LU} ds - \frac{1}{2} \int_{S_L}^{S_R} \int_0^1 \mathcal{A}_y(\Phi_{\xi D, \xi U}) \frac{d}{ds} \Phi_{\xi D, \xi U} ds \\ & - \frac{S_R}{2} \int_0^1 \mathcal{A}_y(\Phi_{RD,RU}) \frac{d}{ds} \Phi_{RD,RU} ds, \end{aligned} \quad (4.1.4)$$

where

$$\Phi_{L\psi,R\psi} = \Phi(s; \mathcal{R}_y(\psi; \mathbf{W}_{LD}, \mathbf{W}_{LU}), \mathcal{R}_y(\psi; \mathbf{W}_{RD}, \mathbf{W}_{RU})) \quad (4.1.5)$$

and

$$\Phi_{\xi D, \xi U} = \Phi(s; \mathcal{R}_x(\xi; \mathbf{W}_{LD}, \mathbf{W}_{RD}), \mathcal{R}_x(\xi; \mathbf{W}_{LU}, \mathbf{W}_{RU})). \quad (4.1.6)$$

Note that when the self-similar solution only depends on U we use the notation \mathbf{R} , while for $\mathbf{W} = (U, \sigma)^T$ we use the notation \mathcal{R} . Notice that the right hand side in (4.1.4) can be written as

$$- \int_{S_L}^{S_R} \int_{S_D}^{S_U} (\mathcal{R}(\xi, \psi) - \mathbf{W}_0(\xi, \psi)) d\xi d\psi = \mathcal{D}^{RU} + \mathcal{D}^{LU} + \mathcal{D}^{RD} + \mathcal{D}^{LD}, \quad (4.1.7)$$

being the four contributions at the vertex given by

$$\begin{aligned}
\mathcal{D}^{RU} &= - \int_0^{S_R} \int_0^{S_U} (\mathcal{R}(\xi, \psi) - \mathbf{W}_{RU}) d\xi d\psi, \\
\mathcal{D}^{LU} &= - \int_{S_L}^0 \int_0^{S_U} (\mathcal{R}(\xi, \psi) - \mathbf{W}_{LU}) d\xi d\psi, \\
\mathcal{D}^{RD} &= - \int_0^{S_R} \int_{S_D}^0 (\mathcal{R}(\xi, \psi) - \mathbf{W}_{RD}) d\xi d\psi, \\
\mathcal{D}^{LD} &= - \int_{S_L}^0 \int_{S_D}^0 (\mathcal{R}(\xi, \psi) - \mathbf{W}_{LD}) d\xi d\psi.
\end{aligned} \tag{4.1.8}$$

At a given cell C_{ij} , the contributions at the four edges are defined by using a one-dimensional path-conservative scheme (see Section 1.4). Specifically, we consider fluctuations of the form $\mathcal{D}_{i-1/2,j}^+$ and $\mathcal{D}_{i+1/2,j}^-$ at vertical edges, and of the form $\mathcal{D}_{i,j-1/2}^+$ and $\mathcal{D}_{i,j+1/2}^-$ at horizontal ones. Moreover, the two-dimensional contributions at vertices are given by (4.1.8), which in this case read as

$$\begin{aligned}
\mathcal{D}_{i-1/2,j-1/2}^{RU} &= - \int_0^{S_{i-1/2,j-1/2}^R} \int_0^{S_{i-1/2,j-1/2}^U} (\mathcal{R}(\xi, \psi) - \mathbf{W}_{ij}) d\xi d\psi, \\
\mathcal{D}_{i-1/2,j+1/2}^{RD} &= - \int_0^{S_{i-1/2,j+1/2}^R} \int_{S_{i-1/2,j+1/2}^D}^0 (\mathcal{R}(\xi, \psi) - \mathbf{W}_{ij}) d\xi d\psi, \\
\mathcal{D}_{i+1/2,j-1/2}^{LU} &= - \int_{S_{i+1/2,j-1/2}^L}^0 \int_0^{S_{i+1/2,j-1/2}^U} (\mathcal{R}(\xi, \psi) - \mathbf{W}_{ij}) d\xi d\psi, \\
\mathcal{D}_{i+1/2,j+1/2}^{LD} &= - \int_{S_{i+1/2,j+1/2}^L}^0 \int_{S_{i+1/2,j+1/2}^D}^0 (\mathcal{R}(\xi, \psi) - \mathbf{W}_{ij}) d\xi d\psi.
\end{aligned}$$

where $S_{i+1/2,j+1/2}^\alpha$, $\alpha = L, R, D, U$, denotes the corresponding maximal speed of propagation given by (3.1.6) and (3.1.7) at the vertex $(x_{i+1/2}, y_{j+1/2})$. Finally, the numerical scheme in semidiscrete form is defined as (see Figure 4.2):

$$\begin{aligned}
\frac{d\mathbf{W}_{ij}}{dt} &= - \frac{1}{\Delta x \Delta y} \left[\beta_L \mathcal{D}_{i-1/2,j}^+ + \beta_R \mathcal{D}_{i+1/2,j}^- + \beta_D \mathcal{D}_{i,j-1/2}^+ + \beta_U \mathcal{D}_{i,j+1/2}^- \right. \\
&\quad \left. + \Delta t (\mathcal{D}_{i-1/2,j-1/2}^{RU} + \mathcal{D}_{i+1/2,j-1/2}^{LU} + \mathcal{D}_{i-1/2,j+1/2}^{RD} + \mathcal{D}_{i+1/2,j+1/2}^{LD}) \right],
\end{aligned} \tag{4.1.9}$$

where

$$\begin{cases} \beta_L = \Delta y + S_{i-1/2,j+1/2}^D \Delta t - S_{i-1/2,j-1/2}^U \Delta t, \\ \beta_R = \Delta y + S_{i+1/2,j+1/2}^D \Delta t - S_{i+1/2,j-1/2}^U \Delta t, \\ \beta_D = \Delta x + S_{i+1/2,j-1/2}^L \Delta t - S_{i-1/2,j-1/2}^R \Delta t, \\ \beta_U = \Delta x + S_{i+1/2,j+1/2}^L \Delta t - S_{i-1/2,j+1/2}^R \Delta t. \end{cases} \tag{4.1.10}$$

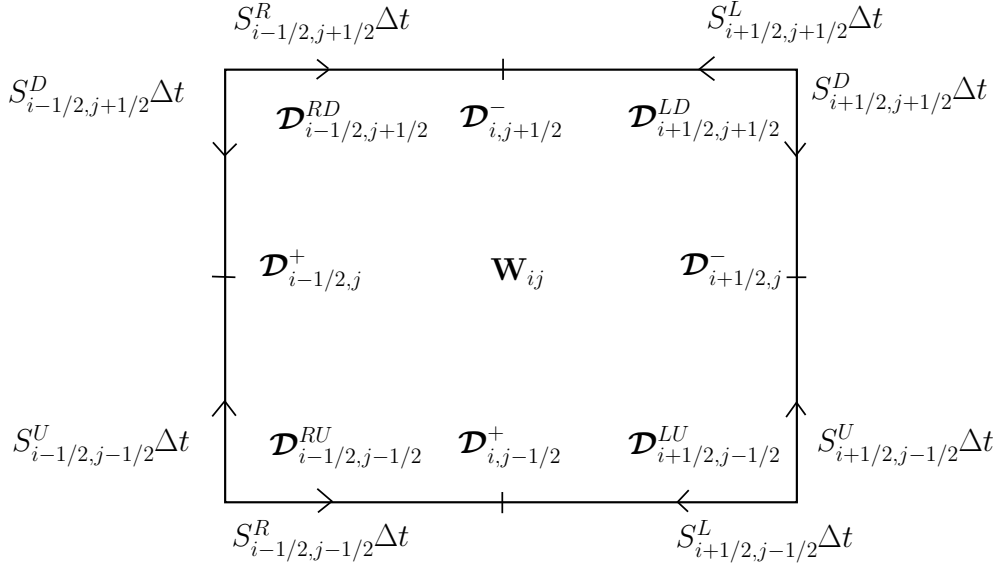


Figure 4.2: Contributions in the numerical scheme (4.1.9).

We have the following result.

Proposition 1. *Let us suppose that $\mathcal{A}_\zeta(\mathbf{W})$, for $\zeta = x, y$, are C^1 matrices with bounded derivatives. Let us consider regular finite volume meshes with constant size of edges. The scheme (4.1.9) is consistent with the nonconservative system (5.1.1) for smooth solutions.*

Proof. Let \mathbf{W} be a smooth solution of (5.1.1) and define $\mathbf{W}_{ij}^n = \mathbf{W}(x_i, y_j, t^n)$ (the index n will be dropped unless necessary). Given $\mathbf{x} = (x, y)$, t , we want to estimate the consistency error

$$\begin{aligned} \mathbf{E}(\mathbf{x}, t) = & \frac{\mathbf{W}_{ij}^{n+1} - \mathbf{W}_{ij}^n}{\Delta t} + \frac{1}{\Delta x \Delta y} \left[\beta_L \mathcal{D}_{i-1/2, j}^+ + \beta_R \mathcal{D}_{i+1/2, j}^- + \beta_D \mathcal{D}_{i, j-1/2}^+ + \beta_U \mathcal{D}_{i, j+1/2}^- \right. \\ & \left. + \Delta t (\mathcal{D}_{i-1/2, j-1/2}^{RU} + \mathcal{D}_{i+1/2, j-1/2}^{LU} + \mathcal{D}_{i-1/2, j+1/2}^{RD} + \mathcal{D}_{i+1/2, j+1/2}^{LD}) \right], \end{aligned}$$

where

$$\mathbf{W}_{ij}^n = \mathbf{W}(x + i\Delta x, y + j\Delta y, t + n\Delta t), \quad i, j = -1, 0, 1, \quad n = 0, 1.$$

First we have

$$\begin{aligned} \mathbf{E}(\mathbf{x}, t) = & \frac{\partial \mathbf{W}}{\partial t}(\mathbf{x}, t) + \frac{1}{\Delta x \Delta y} \left[\hat{\beta}_L \hat{\mathcal{D}}_{i-1/2, j}^+ + \hat{\beta}_R \hat{\mathcal{D}}_{i+1/2, j}^- + \hat{\beta}_D \hat{\mathcal{D}}_{i, j-1/2}^+ + \hat{\beta}_U \hat{\mathcal{D}}_{i, j+1/2}^- \right. \\ & \left. + \Delta t (\hat{\mathcal{D}}_{i-1/2, j-1/2}^{RU} + \hat{\mathcal{D}}_{i+1/2, j-1/2}^{LU} + \hat{\mathcal{D}}_{i-1/2, j+1/2}^{RD} + \hat{\mathcal{D}}_{i+1/2, j+1/2}^{LD}) \right] \\ & + \mathcal{O}(\Delta \mathbf{x}, \Delta t), \end{aligned}$$

where

$$\begin{aligned}\widehat{\mathcal{D}}_{i\pm 1/2,j}^{\pm} &= \mathcal{D}^{\pm}(\mathbf{W}(x - \Delta x/2, y, t), \mathbf{W}(x + \Delta x/2, y, t)), \\ \widehat{\mathcal{D}}_{i,j\pm 1/2}^{\pm} &= \mathcal{D}^{\pm}(\mathbf{W}(x, y - \Delta y/2, t), \mathbf{W}(x, y + \Delta y/2, t)),\end{aligned}$$

the $\widehat{\mathcal{D}}^{\nu\mu}$, $\nu = L, R$, $\mu = D, U$, fluctuations are computed using the states

$$\begin{aligned}\mathbf{W}_{LD} &= \mathbf{W}(x - \Delta x/2, y - \Delta y/2, t), & \mathbf{W}_{LU} &= \mathbf{W}(x - \Delta x/2, y + \Delta y/2, t), \\ \mathbf{W}_{RD} &= \mathbf{W}(x + \Delta x/2, y - \Delta y/2, t), & \mathbf{W}_{RU} &= \mathbf{W}(x + \Delta x/2, y + \Delta y/2, t),\end{aligned}$$

and the wave speeds are taken to be

$$\begin{aligned}S_R &= \max(|\lambda_x^1(\mathbf{W}(\mathbf{x}, t))|, |\lambda_x^N(\mathbf{W}(\mathbf{x}, t))|), & S_L &= -S_R, \\ S_U &= \max(|\lambda_y^1(\mathbf{W}(\mathbf{x}, t))|, |\lambda_y^N(\mathbf{W}(\mathbf{x}, t))|), & S_D &= -S_U.\end{aligned}$$

Using (4.1.4), (1.4.3), (1.4.4), we obtain

$$\begin{aligned}\mathbf{E}(\mathbf{x}, t) &= \frac{\partial \mathbf{W}}{\partial t}(\mathbf{x}, t) + \frac{1}{\Delta x \Delta y} \left[(\Delta y - 2S_U \Delta t) \int_0^1 \mathcal{A}_x \left(\Phi_{(i-1/2)j, (i+1/2)j} \right) \frac{d}{ds} \Phi_{(i-1/2)j, (i+1/2)j} ds \right. \\ &+ (\Delta x - 2S_R \Delta t) \int_0^1 \mathcal{A}_y \left(\Phi_{i(j-1/2), i(j+1/2)} \right) \frac{d}{ds} \Phi_{i(j-1/2), i(j+1/2)} ds \\ &+ \Delta t \frac{S_U}{2} \int_0^1 \mathcal{A}_x \left(\Phi_{(i-1/2)(j-1/2), (i+1/2)(j+1/2)} \right) \frac{d}{ds} \Phi_{(i-1/2)(j-1/2), (i+1/2)(j+1/2)} ds \\ &+ \Delta t \frac{1}{2} \int_{-S_U}^{S_U} \int_0^1 \mathcal{A}_x \left(\Phi_{(i-1/2)\psi, (i+1/2)\psi} \right) \frac{d}{ds} \Phi_{(i-1/2)\psi, (i+1/2)\psi} ds d\psi \\ &+ \Delta t \frac{S_U}{2} \int_0^1 \mathcal{A}_x \left(\Phi_{(i-1/2)(j+1/2), (i+1/2)(j+1/2)} \right) \frac{d}{ds} \Phi_{(i-1/2)(j+1/2), (i+1/2)(j+1/2)} ds \\ &+ \Delta t \frac{S_R}{2} \int_0^1 \mathcal{A}_y \left(\Phi_{(i-1/2)(j-1/2), (i-1/2)(j+1/2)} \right) \frac{d}{ds} \Phi_{(i-1/2)(j-1/2), (i-1/2)(j+1/2)} ds \\ &+ \Delta t \frac{1}{2} \int_{-S_R}^{S_R} \int_0^1 \mathcal{A}_y \left(\Phi_{\xi(j-1/2), \xi(j+1/2)} \right) \frac{d}{ds} \Phi_{\xi(j-1/2), \xi(j+1/2)} ds d\xi \\ &+ \Delta t \frac{S_R}{2} \int_0^1 \mathcal{A}_y \left(\Phi_{(i+1/2)(j-1/2), (i+1/2)(j+1/2)} \right) \frac{d}{ds} \Phi_{(i+1/2)(j-1/2), (i+1/2)(j+1/2)} ds \left. \right] \\ &+ \mathcal{O}(\Delta \mathbf{x}, \Delta t).\end{aligned}$$

Changing now the extremes of the paths (what add new errors of order $\mathcal{O}(\Delta \mathbf{x})$) we obtain

$$\begin{aligned}
\mathbf{E}(\mathbf{x}, t) = & \frac{\partial \mathbf{W}}{\partial t}(\mathbf{x}, t) + \frac{1}{\Delta x \Delta y} \left[(\Delta y - 2S_U \Delta t) \int_0^1 \mathcal{A}_x \left(\Phi_{(i-1/2)j, (i+1/2)j} \right) \frac{d}{ds} \Phi_{(i-1/2)j, (i+1/2)j} ds \right. \\
& + (\Delta x - 2S_R \Delta t) \int_0^1 \mathcal{A}_y \left(\Phi_{i(j-1/2), i(j+1/2)} \right) \frac{d}{ds} \Phi_{i(j-1/2), i(j+1/2)} ds \\
& + \Delta t \frac{S_U}{2} \int_0^1 \mathcal{A}_x \left(\Phi_{(i-1/2)j, (i+1/2)j} \right) \frac{d}{ds} \Phi_{(i-1/2)j, (i+1/2)j} ds \\
& + \Delta t \frac{1}{2} \int_{-S_U}^{S_U} \int_0^1 \mathcal{A}_x \left(\Phi_{(i-1/2)j, (i+1/2)j} \right) \frac{d}{ds} \Phi_{(i-1/2)j, (i+1/2)j} ds d\psi \\
& + \Delta t \frac{S_U}{2} \int_0^1 \mathcal{A}_x \left(\Phi_{(i-1/2)j, (i+1/2)j} \right) \frac{d}{ds} \Phi_{(i-1/2)j, (i+1/2)j} ds \\
& + \Delta t \frac{S_R}{2} \int_0^1 \mathcal{A}_y \left(\Phi_{i(j-1/2), i(j+1/2)} \right) \frac{d}{ds} \Phi_{i(j-1/2), i(j+1/2)} ds \\
& + \Delta t \frac{1}{2} \int_{-S_R}^{S_R} \int_0^1 \mathcal{A}_y \left(\Phi_{i(j-1/2), i(j+1/2)} \right) \frac{d}{ds} \Phi_{i(j-1/2), i(j+1/2)} ds d\xi \\
& \left. + \Delta t \frac{S_R}{2} \int_0^1 \mathcal{A}_y \left(\Phi_{i(j-1/2), i(j+1/2)} \right) \frac{d}{ds} \Phi_{i(j-1/2), i(j+1/2)} ds \right] + \mathcal{O}(\Delta \mathbf{x}, \Delta t),
\end{aligned}$$

what leads to

$$\begin{aligned}
\mathbf{E}(\mathbf{x}, t) = & \frac{\partial \mathbf{W}}{\partial t}(\mathbf{x}, t) + \frac{1}{\Delta x} \int_0^1 \mathcal{A}_x \left(\Phi_{(i-1/2)j, (i+1/2)j} \right) \frac{d}{ds} \Phi_{(i-1/2)j, (i+1/2)j} ds \\
& + \frac{1}{\Delta y} \int_0^1 \mathcal{A}_y \left(\Phi_{i(j-1/2), i(j+1/2)} \right) \frac{d}{ds} \Phi_{i(j-1/2), i(j+1/2)} ds + \mathcal{O}(\Delta \mathbf{x}, \Delta t) \\
= & \frac{\partial \mathbf{W}}{\partial t}(\mathbf{x}, t) + \mathcal{A}_x(\mathbf{W}(\mathbf{x}, t)) \mathbf{W}_x(\mathbf{x}, t) + \mathcal{A}_y(\mathbf{W}(\mathbf{x}, t)) \mathbf{W}_y(\mathbf{x}, t) + \mathcal{O}(\Delta \mathbf{x}, \Delta t) \\
= & \mathcal{O}(\Delta \mathbf{x}, \Delta t),
\end{aligned}$$

as we wanted to prove. □

Remark 10. *The scheme (4.1.9) is conservative when it is applied to a conservative system.*

4.2 Well-Balanced 2D HLL Riemann solver

The results presented in Section 4.1 were valid for a general nonconservative system of the form (5.1.1) for the subsonic case in both directions. The supersonic cases will be discussed in Section 4.2.1. In this section we return to the original system

$$\partial_t \mathbf{U} + \nabla \cdot \mathcal{F}(\mathbf{U}) + \mathcal{B}(\mathbf{U}) \cdot \nabla \mathbf{U} = \mathcal{S}(\mathbf{U}) \cdot \nabla \sigma \quad (4.2.1)$$

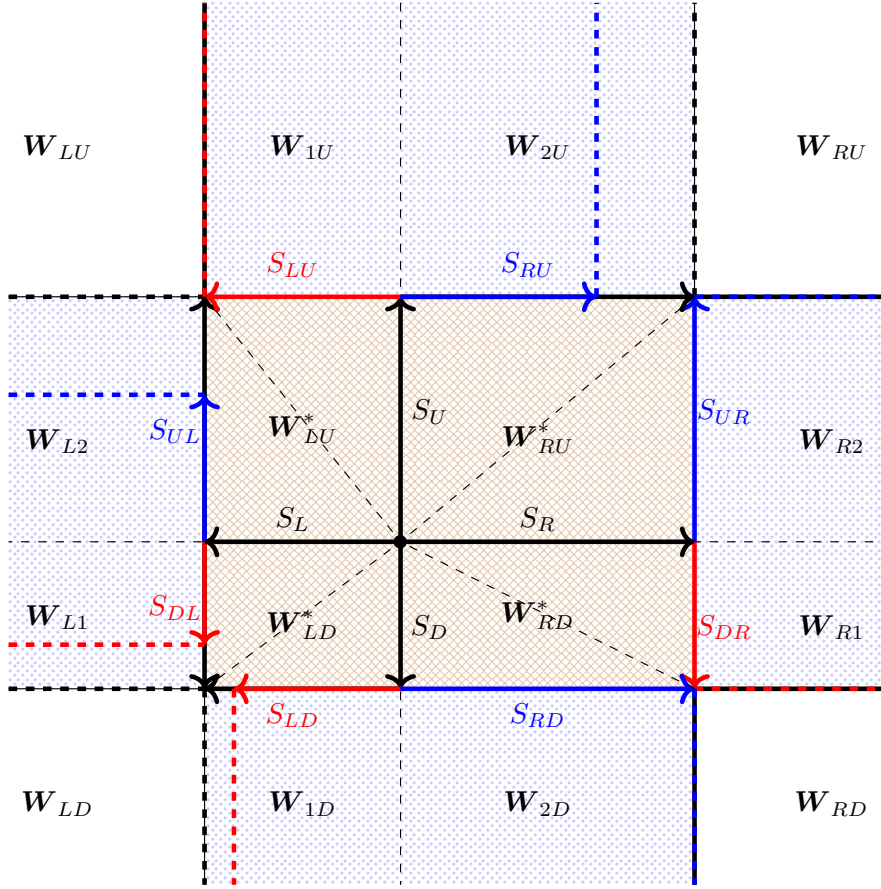


Figure 4.3: Structure of the solution of the 2D Riemann problem for system (4.2.1). Subsonic case.

where $U, \mathcal{F}, \mathcal{B}, \mathcal{S}, \sigma$ are defined in (1.3.1), for which we will derive a simple HLL-type Riemann solver. Later in Section 4.3 we will see how to build more precise solvers of AVM type.

The first step is to consider an approximation to the structure of the solution of the 2D Riemann problem. The following notation will be useful:

$$\Delta \mathcal{F}_\mu = F_{R\mu} - F_{L\mu} + \mathcal{B}_{x,\Phi}(U_{R\mu} - U_{L\mu}) - \mathcal{S}_{x,\Phi}(\sigma_{R\mu} - \sigma_{L\mu}), \quad \mu = 1, 2, D, U,$$

and

$$\Delta \mathcal{G}_\nu = G_{\nu U} - G_{\nu D} + \mathcal{B}_{y,\Phi}(U_{\nu U} - U_{\nu D}) - \mathcal{S}_{y,\Phi}(\sigma_{\nu U} - \sigma_{\nu D}), \quad \nu = 1, 2, L, R,$$

where $\mathcal{B}_{\zeta,\Phi}$ and $\mathcal{S}_{\zeta,\Phi}$ are evaluated in the corresponding states and depend on the chosen family of paths Φ (see Section 1.4). When no confusion arises, we can drop the dependence on Φ . In our case, the proposed solution has the structure sketched in Figure 4.3. In particular,

$$W_{\nu\mu}^* = \begin{pmatrix} U_{\nu\mu}^* \\ \sigma_{\nu\mu} \end{pmatrix}, \quad \nu = L, R, \quad \mu = D, U,$$

being $\mathbf{U}_{\nu\mu}^*$ states to be determined. The remaining HLL states $\mathbf{W}_{\alpha i}$ ($\alpha \in \{L, R, D, U\}$, $i = 1, 2$) come from the corresponding one-dimensional Riemann problems arising at edges. For example, for $\mu = D, U$, we would have $\mathbf{W}_{\mu 1} = (\mathbf{U}_{\mu 1}, \sigma_{L\mu})^t$ and $\mathbf{W}_{\mu 2} = (\mathbf{U}_{\mu 2}, \sigma_{R\mu})^t$, where

$$\mathbf{U}_{\mu 1} = \frac{S_R \mathbf{U}_{R\mu} - S_L \mathbf{U}_{L\mu} - \Delta \mathcal{F}_\mu}{S_R - S_L} - \frac{S_R}{S_R - S_L} \mathbf{A}_x^{-1} \mathbf{S}_x (\sigma_{R\mu} - \sigma_{L\mu}) \quad (4.2.2)$$

and

$$\mathbf{U}_{\mu 2} = \frac{S_R \mathbf{U}_{R\mu} - S_L \mathbf{U}_{L\mu} - \Delta \mathcal{F}_\mu}{S_R - S_L} - \frac{S_L}{S_R - S_L} \mathbf{A}_x^{-1} \mathbf{S}_x (\sigma_{R\mu} - \sigma_{L\mu}), \quad (4.2.3)$$

and similarly for the remaining states: see [149] for details.

In this context, the consistency condition (4.1.4) adopts the following form:

$$\begin{aligned} & \int_{S_L}^{S_R} \int_{S_D}^{S_U} (\mathcal{R}(\xi, \psi) - \mathbf{W}_0(\xi, \psi)) d\xi d\psi = \\ & \frac{S_D}{2} \int_0^1 \mathcal{A}_x(\Phi_{LD, RD}) \frac{d}{ds} \Phi_{LD, RD} ds + \frac{S_D}{2} \int_0^1 \mathcal{A}_x(\Phi_{L1, R1}) \frac{d}{ds} \Phi_{L1, R1} ds \\ & - \frac{S_U}{2} \int_0^1 \mathcal{A}_x(\Phi_{L2, R2}) \frac{d}{ds} \Phi_{L2, R2} ds - \frac{S_U}{2} \int_0^1 \mathcal{A}_x(\Phi_{LU, RU}) \frac{d}{ds} \Phi_{LU, RU} ds \\ & + \frac{S_L}{2} \int_0^1 \mathcal{A}_y(\Phi_{LD, LU}) \frac{d}{ds} \Phi_{LD, LU} ds + \frac{S_L}{2} \int_0^1 \mathcal{A}_y(\Phi_{1D, 1U}) \frac{d}{ds} \Phi_{1D, 1U} ds \\ & - \frac{S_R}{2} \int_0^1 \mathcal{A}_y(\Phi_{2D, 2U}) \frac{d}{ds} \Phi_{2D, 2U} ds - \frac{S_R}{2} \int_0^1 \mathcal{A}_y(\Phi_{RD, RU}) \frac{d}{ds} \Phi_{RD, RU} ds, \end{aligned} \quad (4.2.4)$$

where property (1.3.5) has been applied. The self-similar solution $\mathcal{R}(\xi, \psi)$ is supposed to be constant in the domain $\mathcal{Q}_\xi = [S_L, S_R] \times [S_D, S_U]$, i.e., $\mathcal{R}(\xi, \psi) = \mathbf{W}^*$. We have also assumed that \mathcal{Q}_ξ is limited by the maximal waves speeds for each direction, in such a way that there are no boundary parts between the original states and the strong interaction region (see Figure 4.3). Furthermore, the approximated structure of the solution for the 2D Riemann problem is taken such that \mathbf{W}^* is decomposed in a set of four non overlapping subregions, i.e. $\mathbf{W}^* = \sum \mathbf{W}_{AB}^*$, where AB corresponds to LD, LU, RD, RU , with \mathbf{W}_{AB}^* constant states, not necessarily the same. Then, taking into account (1.3.3) and (1.3.11), we deduce the following condition:

$$\begin{aligned} & S_L S_D (\mathbf{U}_{LD}^* - \mathbf{U}_{LD}) - S_L S_U (\mathbf{U}_{LU}^* - \mathbf{U}_{LU}) \\ & - S_R S_D (\mathbf{U}_{RD}^* - \mathbf{U}_{RD}) + S_R S_U (\mathbf{U}_{RU}^* - \mathbf{U}_{RU}) = \mathcal{C}, \end{aligned} \quad (4.2.5)$$

where the term \mathcal{C} is given by

$$\mathcal{C} = \frac{S_D}{2} (\Delta \mathcal{F}_D + \Delta \mathcal{F}_1) - \frac{S_U}{2} (\Delta \mathcal{F}_2 + \Delta \mathcal{F}_U) + \frac{S_L}{2} (\Delta \mathcal{G}_L + \Delta \mathcal{G}_1) - \frac{S_R}{2} (\Delta \mathcal{G}_2 + \Delta \mathcal{G}_R). \quad (4.2.6)$$

On the other hand, in order to have a complete nonlinear system, we use the jump conditions through the internal boundaries of \mathcal{Q}_ξ . The Rankine-Hugoniot conditions (1.3.13) through the

zero waves can be expressed as

$$\begin{cases} \mathbf{U}_{LU}^* - \mathbf{U}_{LD}^* = \mathbf{A}_y(\mathbf{U}_{LD}^*, \mathbf{U}_{LU}^*)^{-1} \mathbf{S}_y(\mathbf{U}_{LD}^*, \mathbf{U}_{LU}^*)(\sigma_{LU} - \sigma_{LD}) & \equiv \alpha, \\ \mathbf{U}_{RU}^* - \mathbf{U}_{RD}^* = \mathbf{A}_y(\mathbf{U}_{RD}^*, \mathbf{U}_{RU}^*)^{-1} \mathbf{S}_y(\mathbf{U}_{RD}^*, \mathbf{U}_{RU}^*)(\sigma_{RU} - \sigma_{RD}) & \equiv \beta, \\ \mathbf{U}_{RD}^* - \mathbf{U}_{LD}^* = \mathbf{A}_x(\mathbf{U}_{LD}^*, \mathbf{U}_{RD}^*)^{-1} \mathbf{S}_x(\mathbf{U}_{LD}^*, \mathbf{U}_{RD}^*)(\sigma_{RD} - \sigma_{LD}) & \equiv \gamma, \\ \mathbf{U}_{RU}^* - \mathbf{U}_{LU}^* = \mathbf{A}_x(\mathbf{U}_{LU}^*, \mathbf{U}_{RU}^*)^{-1} \mathbf{S}_x(\mathbf{U}_{LU}^*, \mathbf{U}_{RU}^*)(\sigma_{RU} - \sigma_{LU}) & \equiv \delta. \end{cases} \quad (4.2.7)$$

As $\alpha - \beta - \gamma + \delta = 0$, we can consider (4.2.5) and three of the equations in (4.2.7) to determine the four star states. We thus obtain

$$\begin{aligned} \mathbf{U}_{RU}^* = \frac{1}{(S_R - S_L)(S_U - S_D)} & (S_L S_D \mathbf{U}_{LD} - S_L S_U \mathbf{U}_{LU} - S_R S_D \mathbf{U}_{RD} \\ & + S_R S_U \mathbf{U}_{RU} + S_L S_D \alpha - S_L (S_U - S_D) \delta - S_R S_D \beta + \mathcal{C}) \end{aligned} \quad (4.2.8)$$

and

$$\mathbf{U}_{RD}^* = \mathbf{U}_{RU}^* - \beta, \quad \mathbf{U}_{LU}^* = \mathbf{U}_{RU}^* - \delta, \quad \mathbf{U}_{LD}^* = \mathbf{U}_{RU}^* - \delta - \alpha. \quad (4.2.9)$$

A remark about the computation of \mathbf{U}_{RU}^* is in order. Notice that \mathbf{U}_{RU}^* depends on the term \mathcal{C} defined by (4.2.6), which in turn rely on terms of the form (1.3.9)-(1.3.10). To be consistent with the structure of the approximate solution (see Figure 4.3), the path should also involve the intermediate states. In fact, integration along the boundaries of the subregions of \mathcal{Q}_ξ results in coupling terms of the form (1.3.9), with dependence on the star states \mathbf{W}_{AB}^* . This dependence is also clear in the jump conditions (4.2.7). For example, using (1.3.5) we would have

$$\begin{aligned} \int_0^1 \mathcal{B}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2} ds &= \int_0^1 \mathcal{B}_x(\Phi_{L2,LU^*}) \frac{d}{ds} \Phi_{L2,LU^*} ds \\ &+ \int_0^1 \mathcal{B}_x(\Phi_{LU^*,RU^*}) \frac{d}{ds} \Phi_{LU^*,RU^*} ds + \int_0^1 \mathcal{B}_x(\Phi_{RU^*,R2}) \frac{d}{ds} \Phi_{RU^*,R2} ds, \end{aligned}$$

so, using (1.3.9),

$$\begin{aligned} \mathcal{B}_{x,\Phi}(U_{L2}, U_{R2})(U_{R2} - U_{L2}) &= \mathcal{B}_{x,\Phi}(U_{L2}, U_{LU}^*)(U_{LU}^* - U_{L2}) \\ &+ \mathcal{B}_{x,\Phi}(U_{LU}^*, U_{RU}^*)(U_{RU}^* - U_{LU}^*) + \mathcal{B}_{x,\Phi}(U_{RU}^*, U_{R2})(U_{R2} - U_{RU}^*), \end{aligned} \quad (4.2.10)$$

and similarly for the other terms appearing in (4.2.6). In the case in which a linearization $\mathcal{B}_{x,\Phi}$ is not available, the path integrals in (1.3.9) should be numerically approximated using an appropriate quadrature rule. An analogous situation has already been considered in [75, Sect. 2.1] for the one-dimensional case, and we have followed here a similar approach. Therefore, (4.2.8) would define \mathbf{U}_{RU}^* in an implicit way. Therefore, the solution of the system formed by (4.2.4) and three of the jump relations through the zero wave (4.2.7) requires an iterative procedure. First, the initial value can be set to

$$\begin{aligned} (\mathbf{U}_{RU}^*)^0 &= \frac{1}{(S_R - S_L)(S_U - S_D)} (S_L S_D \mathbf{U}_{LD} - S_L S_U \mathbf{U}_{LU} - S_R S_D \mathbf{U}_{RD} \\ &+ S_R S_U \mathbf{U}_{RU} + S_L S_D \alpha^0 - S_L (S_U - S_D) \delta^0 - S_R S_D \beta^0 + \mathcal{C}^0) \end{aligned}$$

where the \mathcal{C}^0 term is given by (4.2.6) with

$$\Delta \mathcal{F}_\mu = \mathbf{F}_{R\mu} - \mathbf{F}_{L\mu} + \mathbf{B}_{x,\Phi^0}(\mathbf{U}_{R\mu} - \mathbf{U}_{L\mu}) - \mathbf{S}_{x,\Phi^0}(\sigma_{R\mu} - \sigma_{L\mu}), \quad \mu = 1, 2, D, U, \quad (4.2.11)$$

and

$$\Delta \mathcal{G}_\nu = \mathbf{G}_{\nu U} - \mathbf{G}_{\nu D} + \mathbf{B}_{y,\Phi^0}(\mathbf{U}_{\nu U} - \mathbf{U}_{\nu D}) - \mathbf{S}_{y,\Phi^0}(\sigma_{\nu U} - \sigma_{\nu D}), \quad \nu = 1, 2, L, R, \quad (4.2.12)$$

Here, Φ^0 is the path joining $\mathbf{U}_{L\mu}$ and $\mathbf{U}_{R\mu}$ (or $\mathbf{U}_{\nu D}$ and $\mathbf{U}_{\nu U}$) without passing through the intermediate star states. Also, after evaluating $(\mathbf{U}_{RU}^*)^0$, we define

$$\begin{aligned} (\mathbf{U}_{RD}^*)^0 &= (\mathbf{U}_{RU}^*)^0 - \beta^0, \\ (\mathbf{U}_{LU}^*)^0 &= (\mathbf{U}_{RU}^*)^0 - \delta^0, \\ (\mathbf{U}_{LD}^*)^0 &= (\mathbf{U}_{RU}^*)^0 - \delta^0 - \alpha^0. \end{aligned}$$

The initial guess for $\alpha, \beta, \gamma, \delta$ can be taken as

$$\begin{cases} \mathbf{A}_y(\mathbf{U}_{LD}, \mathbf{U}_{LU})^{-1} \mathbf{S}_y(\mathbf{U}_{LD}, \mathbf{U}_{LU})(\sigma_{LU} - \sigma_{LD}) & \equiv \alpha^0, \\ \mathbf{A}_y(\mathbf{U}_{RD}, \mathbf{U}_{RU})^{-1} \mathbf{S}_y(\mathbf{U}_{RD}, \mathbf{U}_{RU})(\sigma_{RU} - \sigma_{RD}) & \equiv \beta^0, \\ \mathbf{A}_x(\mathbf{U}_{LD}, \mathbf{U}_{RD})^{-1} \mathbf{S}_x(\mathbf{U}_{LD}, \mathbf{U}_{RD})(\sigma_{RD} - \sigma_{LD}) & \equiv \gamma^0, \\ \mathbf{A}_x(\mathbf{U}_{LU}, \mathbf{U}_{RU})^{-1} \mathbf{S}_x(\mathbf{U}_{LU}, \mathbf{U}_{RU})(\sigma_{RU} - \sigma_{LU}) & \equiv \delta^0. \end{cases}$$

Finally, we can compute, for instance,

$$\begin{aligned} (\mathbf{U}_{RU}^*)^{m+1} &= \frac{1}{(S_R - S_L)(S_U - S_D)} (S_L S_D \mathbf{U}_{LD} - S_L S_U \mathbf{U}_{LU} - S_R S_D \mathbf{U}_{RD} \\ &\quad + S_R S_U \mathbf{U}_{RU} + S_L S_D \alpha^{m+1} - S_L (S_U - S_D) \delta^{m+1} - S_R S_D \beta^{m+1} + \mathcal{C}^{m+1}) \end{aligned}$$

where \mathcal{C}^{m+1} is still given by (4.2.6), but with the integral path in (4.2.11) and (4.2.12) passing through the intermediate states (see (4.2.10)). We consider now

$$\begin{cases} \mathbf{A}_y((\mathbf{U}_{LD}^*)^m, (\mathbf{U}_{LU}^*)^m)^{-1} \mathbf{S}_y((\mathbf{U}_{LD}^*)^m, (\mathbf{U}_{LU}^*)^m)(\sigma_{LU} - \sigma_{LD}) & \equiv \alpha^{m+1}, \\ \mathbf{A}_y((\mathbf{U}_{RD}^*)^m, (\mathbf{U}_{RU}^*)^m)^{-1} \mathbf{S}_y((\mathbf{U}_{RD}^*)^m, (\mathbf{U}_{RU}^*)^m)(\sigma_{RU} - \sigma_{RD}) & \equiv \beta^{m+1}, \\ \mathbf{A}_x((\mathbf{U}_{LD}^*)^m, (\mathbf{U}_{RD}^*)^m)^{-1} \mathbf{S}_x((\mathbf{U}_{LD}^*)^m, (\mathbf{U}_{RD}^*)^m)(\sigma_{RD} - \sigma_{LD}) & \equiv \gamma^{m+1}, \\ \mathbf{A}_x((\mathbf{U}_{LU}^*)^m, (\mathbf{U}_{RU}^*)^m)^{-1} \mathbf{S}_x((\mathbf{U}_{LU}^*)^m, (\mathbf{U}_{RU}^*)^m)(\sigma_{RU} - \sigma_{LU}) & \equiv \delta^{m+1}. \end{cases}$$

Obviously, after computing $(\mathbf{U}_{RU}^*)^{m+1}$, we define

$$\begin{aligned} (\mathbf{U}_{RD}^*)^{m+1} &= (\mathbf{U}_{RU}^*)^{m+1} - \beta^{m+1}, \\ (\mathbf{U}_{LU}^*)^{m+1} &= (\mathbf{U}_{RU}^*)^{m+1} - \delta^{m+1}, \\ (\mathbf{U}_{LD}^*)^{m+1} &= (\mathbf{U}_{RU}^*)^{m+1} - \delta^{m+1} - \alpha^{m+1}. \end{aligned}$$

Finally, notice that there is no need to consider a path Φ_σ involving an intermediate state in the definition of the source terms (1.3.10), as for the variable σ we have only a 1-wave problem. Therefore, it suffices to consider a simple choice (e.g., a segment) for the path-integral connecting different values of σ .

This completely determines the approximate solution $\mathbf{R}(\xi, \psi)$ of the 2D Riemann problem, which in turn allows to compute the contributions (4.1.8) at vertices. Therefore, the scheme (4.1.9) can be fully defined. Notice that the one-dimensional contributions \mathbf{D}^\pm are simply those given by the usual HLL solver.

4.2.1 Supersonic cases

Up to now, we have focused in the case in which the flow is subsonic in both directions. When the flow is supersonic in some direction, several possibilities could be considered (see, e.g., [15, 203, 206]). The simplest one consists in substituting the speeds S_L, S_R, S_D and S_U by $S_L^-, S_R^+, S_D^-, S_U^+$, respectively, where $a^- = \min(a, 0)$ and $a^+ = \max(a, 0)$; this forces the solution to be always in the subsonic case. A more accurate approach is to analyze the position of the interaction area in each case, similarly as it was done in [203]. To this end, we proceed as in Section 4.1 with $\mathcal{Q} = [S_L^- T, S_R^+ T] \times [S_D^- T, S_U^+ T]$ to derive the corresponding consistency condition.

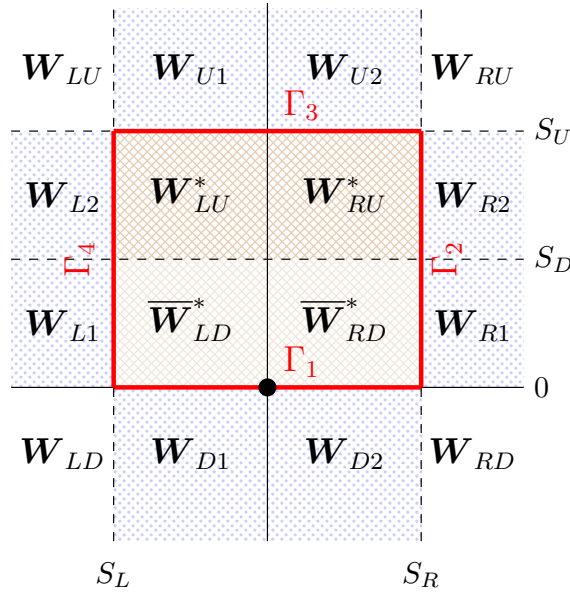


Figure 4.4: Structure of the solution of the 2D Riemann problem for system (4.2.1). Supersonic case in the y -direction.

For example, Figure 4.4 shows the structure of the solution when the flow is subsonic in the x -direction and supersonic in the y -direction, i.e., $S_L < 0 < S_R$ and $S_D > 0$ (for clarity, a bar has been added to those states related to supersonic cases). In this case, the contributions at the edges Γ_1 and Γ_3 would be again

$$\begin{aligned} \frac{S_L}{2} \int_0^1 \mathcal{A}_y(\Phi_{LD,LU}) \frac{d}{ds} \Phi_{LD,LU} ds - \frac{1}{2} \int_{S_L}^{S_R} \int_0^1 \mathcal{A}_y(\Phi_{\xi D, \xi U}) \frac{d}{ds} \Phi_{\xi D, \xi U} ds d\xi \\ - \frac{S_R}{2} \int_0^1 \mathcal{A}_y(\Phi_{RD,RU}) \frac{d}{ds} \Phi_{RD,RU} ds, \end{aligned}$$

following the notation in (4.1.5) and (4.1.6), while the contributions at the edges Γ_2 and Γ_4

would be given by

$$-\frac{S_D}{2} \int_0^1 \mathcal{A}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1} ds - \frac{S_U}{2} \int_0^1 \mathcal{A}_x(\Phi_{LU,RU}) \frac{d}{ds} \Phi_{LU,RU} ds \\ - \frac{1}{2} \int_{S_D}^{S_U} \int_0^1 \mathcal{A}_x(\Phi(s; \mathcal{R}_2(\psi; \mathbf{W}_{L1}, \mathbf{W}_{LU}), \mathcal{R}_2(\psi; \mathbf{W}_{R1}, \mathbf{W}_{RU})) \Phi'(s) ds d\psi.$$

This allows to write the corresponding term \mathcal{C} in this case. Notice that the states $\mathbf{U}_{\nu 1}$, $\nu = L, R$, are now defined as

$$\mathbf{U}_{\nu 1} = \mathbf{U}_{\nu D} + \mathbf{A}_y^{-1} \mathbf{S}_y (\sigma_{\nu U} - \sigma_{\nu D})$$

instead of (4.2.2), while the states $\mathbf{U}_{\nu 2}$ are still given by (4.2.3). On the other hand, the approximate solution \mathbf{R} is given by

$$\mathbf{R}(\xi, \psi) = \begin{cases} \mathbf{U}_{LU}^* & \text{in } [S_L, 0] \times [S_D, S_U], \\ \mathbf{U}_{RU}^* & \text{in } [0, S_R] \times [S_D, S_U], \\ \bar{\mathbf{U}}_{LD}^* & \text{in } [S_L, 0] \times [0, S_D], \\ \bar{\mathbf{U}}_{RD}^* & \text{in } [0, S_R] \times [0, S_D]. \end{cases}$$

Integrating $\mathcal{R}(\xi, \psi) - \mathbf{W}_0(\xi, \psi)$ on $[S_L, S_R] \times [0, S_U]$ we get

$$-S_L(S_U - S_D)\mathbf{U}_{LU}^* + S_R(S_U - S_D)\mathbf{U}_{RU}^* \\ - S_L S_D \bar{\mathbf{U}}_{LD}^* + S_R S_D \bar{\mathbf{U}}_{RD}^* + S_L S_U \mathbf{U}_{LU} - S_R S_U \mathbf{U}_{RU} = \mathcal{C}. \quad (4.2.13)$$

Moreover, the Rankine-Hugoniot conditions (1.3.13) read now as

$$\begin{cases} \mathbf{U}_{RU}^* - \mathbf{U}_{LU}^* = \mathbf{A}_x(\mathbf{U}_{LU}^*, \mathbf{U}_{RU}^*)^{-1} \mathbf{S}_x(\mathbf{U}_{LU}^*, \mathbf{U}_{RU}^*) (\sigma_{RU} - \sigma_{LU}) & \equiv \alpha, \\ \bar{\mathbf{U}}_{RD}^* - \bar{\mathbf{U}}_{LD}^* = \mathbf{A}_x(\bar{\mathbf{U}}_{LD}^*, \bar{\mathbf{U}}_{RD}^*)^{-1} \mathbf{S}_x(\bar{\mathbf{U}}_{LD}^*, \bar{\mathbf{U}}_{RD}^*) (\sigma_{RU} - \sigma_{LU}) & \equiv \beta, \\ \bar{\mathbf{U}}_{LD}^* - \mathbf{U}_{1D} = \mathbf{A}_y(\mathbf{U}_{1D}, \bar{\mathbf{U}}_{LD}^*)^{-1} \mathbf{S}_y(\mathbf{U}_{1D}, \bar{\mathbf{U}}_{LD}^*) (\sigma_{LU} - \sigma_{LD}) & \equiv \gamma, \\ \bar{\mathbf{U}}_{RD}^* - \mathbf{U}_{2D} = \mathbf{A}_y(\mathbf{U}_{2D}, \bar{\mathbf{U}}_{RD}^*)^{-1} \mathbf{S}_y(\mathbf{U}_{2D}, \bar{\mathbf{U}}_{RD}^*) (\sigma_{RU} - \sigma_{RD}) & \equiv \delta. \end{cases} \quad (4.2.14)$$

In this case is not true that $\alpha - \beta - \gamma + \delta = 0$ and the system (4.2.13)-(4.2.14) is overdetermined. However, $\beta = \delta - \gamma$ if we use the modify identity technique for the well-balancedness (see Section (2.4)), and that system have unique solution. Finally, we get

$$\mathbf{U}_{LU}^* = \frac{S_L S_D \bar{\mathbf{U}}_{LD}^* - S_R S_D \bar{\mathbf{U}}_{RD}^* - S_L S_U \mathbf{U}_{LU} + S_R S_U \mathbf{U}_{RU} + \mathcal{C}}{(S_R - S_L)(S_U - S_D)} - \frac{S_R}{S_R - S_L} \delta, \quad (4.2.15)$$

and

$$\mathbf{U}_{RU}^* = \mathbf{U}_{LU}^* + \delta, \quad \bar{\mathbf{U}}_{LD}^* = \mathbf{U}_{1D} + \alpha, \quad \bar{\mathbf{U}}_{RD}^* = \mathbf{U}_{2D} + \beta. \quad (4.2.16)$$

Again, in order to evaluate the star states an iterative procedure is required, similarly to the subsonic case. Once the star states have been computed, the contributions at the vertex read now as

$$\mathbf{D}^{LD} = 0, \\ \mathbf{D}^{LU} = S_L S_D (\bar{\mathbf{U}}_{LD}^* - \mathbf{U}_{LU}) + S_L (S_U - S_D) (\mathbf{U}_{LU}^* - \mathbf{U}_{LU}), \\ \mathbf{D}^{RD} = 0, \\ \mathbf{D}^{RU} = -S_R S_D (\bar{\mathbf{U}}_{RD}^* - \mathbf{U}_{RU}) - S_R (S_U - S_D) (\mathbf{U}_{RU}^* - \mathbf{U}_{RU}).$$

The remaining supersonic cases can be treated in a totally analogous way. For the sake of completeness, we present in Appendix B the complete solution for (4.2.1) with all the subsonic and supersonic cases.

4.3 Well-Balanced 2D AVM-type Riemann solvers

4.3.1 HLL 2D solver in AVM form

The purpose of this section is to reinterpret the simple HLL 2D solver introduced in Section 4.2 in a form that resembles an AVM solver. In particular, the 2D vertex contributions (4.1.8) will be rewritten as combinations of 1D AVM-type Riemann solvers.

Assume first that we are in the subsonic case, and let us analyze one of the contributions in (4.1.8), say D^{LD} :

$$D^{LD} = - \int_{S_L}^0 \int_{S_D}^0 (\mathbf{U}_{LD}^* - \mathbf{U}_{LD}) d\xi d\psi = -S_L S_D (\mathbf{U}_{RU}^* - \delta - \alpha - \mathbf{U}_{LD}).$$

Substituting the value of \mathbf{U}_{RU}^* found in Section 4.2, we have that

$$\begin{aligned} D^{LD} &= S_L S_D \mathbf{U}_{LD} + S_L S_D \delta + S_L S_D \alpha \\ &\quad - \frac{S_L S_D}{(S_R - S_L)(S_U - S_D)} \left\{ S_L S_D \mathbf{U}_{LD} - S_L S_U \mathbf{U}_{LU} - S_R S_D \mathbf{U}_{RD} \right. \\ &\quad \quad \quad + S_R S_U \mathbf{U}_{RU} + S_L S_D (\alpha + \delta) - S_L S_U \delta - S_R S_D \beta \\ &\quad \quad \quad + \frac{S_D}{2} \Delta \mathcal{F}_D + \frac{S_D}{2} \Delta \mathcal{F}_1 - \frac{S_U}{2} \Delta \mathcal{F}_2 - \frac{S_U}{2} \Delta \mathcal{F}_U \\ &\quad \quad \quad \left. + \frac{S_L}{2} \Delta \mathcal{G}_L + \frac{S_L}{2} \Delta \mathcal{G}_1 - \frac{S_R}{2} \Delta \mathcal{G}_2 - \frac{S_R}{2} \Delta \mathcal{G}_R \right\}. \end{aligned}$$

After some algebra, the above expression can be written as

$$\begin{aligned} D^{LD} &= -\frac{S_L}{2} \left(\frac{S_D S_U}{S_U - S_D} (\mathbf{U}_{1U} - \mathbf{U}_{1D} - \alpha) - \frac{S_D}{S_U - S_D} \Delta \mathcal{G}_1 \right) \\ &\quad - \frac{S_D}{2} \left(\frac{S_L S_R}{S_R - S_L} (\mathbf{U}_{R1} - \mathbf{U}_{L1} - \gamma) - \frac{S_L}{S_R - S_L} \Delta \mathcal{F}_1 \right) \\ &\quad - \frac{S_L S_D}{2} \mathbf{U}_{1D} - \frac{S_L S_D}{2} \mathbf{U}_{L1} + S_L S_D \mathbf{U}_{LD} \\ &\quad - \frac{S_L S_D}{(S_U - S_D)(S_R - S_L)} \left(\frac{S_U}{2} (\Delta \mathcal{F}_1 - \Delta \mathcal{F}_2) + \frac{S_R}{2} (\Delta \mathcal{G}_1 - \Delta \mathcal{G}_2) \right). \end{aligned}$$

Define now the contributions in the x -direction:

$$D_x^{\mu, \pm} = \frac{1}{2} \Delta \mathcal{F}_\mu \pm \frac{1}{2} \mathbf{Q}_x^{\text{HLL}} (\mathbf{U}_{R\mu} - \mathbf{U}_{L\mu} - \mathbf{A}_x^{-1} \mathbf{S}_x (\sigma_{R\mu} - \sigma_{L\mu})), \quad (4.3.1)$$

for $\mu = 1, 2, D, U$, being $\mathbf{Q}_x^{\text{HLL}} = \alpha_0 \mathbf{I} + \alpha_1 \mathbf{A}_x$ the HLL diffusion matrix, where $\alpha_0 = \frac{S_R |S_L| - S_L |S_R|}{S_R - S_L} = -\frac{2S_L S_R}{S_R - S_L}$ and $\alpha_1 = \frac{|S_R| - |S_L|}{S_R - S_L} = \frac{S_R + S_L}{S_R - S_L}$. Similarly, the contributions in the



y -direction are given by

$$\mathbf{D}_y^{\nu,\pm} = \frac{1}{2}\Delta\mathcal{G}_\nu \pm \frac{1}{2}\mathbf{Q}_y^{\text{HLL}}(\mathbf{U}_{U\nu} - \mathbf{U}_{D\nu} - \mathbf{A}_y^{-1}\mathbf{S}_y(\sigma_{U\nu} - \sigma_{D\nu})), \quad (4.3.2)$$

for $\nu = 1, 2, L, R$, with $\mathbf{Q}_y^{\text{HLL}} = \beta_0\mathbf{I} + \beta_1\mathbf{A}_y$, $\beta_0 = \frac{S_U|S_D| - S_D|S_U|}{S_U - S_D} = -\frac{2S_DS_U}{S_U - S_D}$ and $\beta_1 = \frac{|S_U| - |S_D|}{S_U - S_D} = \frac{S_U + S_D}{S_U - S_D}$. Using this notation, we can write

$$\begin{aligned} \mathbf{D}^{LD} = & -\frac{S_L}{2}\mathbf{D}_y^{1,-} - \frac{S_D}{2}\mathbf{D}_x^{1,-} - \frac{S_LS_D}{2}(\mathbf{U}_{L1} + \mathbf{U}_{1D} - 2\mathbf{U}_{LD}) \\ & - \frac{S_LS_D}{(S_U - S_D)(S_R - S_L)} \left\{ \frac{S_U}{2}(\Delta\mathcal{F}_1 - \Delta\mathcal{F}_2) + \frac{S_R}{2}(\Delta\mathcal{G}_1 - \Delta\mathcal{G}_2) \right\}. \end{aligned}$$

Now, on the one hand,

$$-\frac{S_LS_D}{2}(\mathbf{U}_{L1} + \mathbf{U}_{1D} - 2\mathbf{U}_{LD}) = -\frac{S_D}{2}\mathbf{D}_x^{D,-} - \frac{S_L}{2}\mathbf{D}_y^{L,-},$$

and, on the other hand,

$$\begin{aligned} & -\frac{S_LS_D}{(S_U - S_D)(S_R - S_L)} \left\{ \frac{S_U}{2}(\Delta\mathcal{F}_1 - \Delta\mathcal{F}_2) + \frac{S_R}{2}(\Delta\mathcal{G}_1 - \Delta\mathcal{G}_2) \right\} = \\ & \quad \frac{S_DS_U}{2(S_U - S_D)}(\mathbf{D}_x^{1,-} - \mathbf{D}_x^{2,-}) \\ & \quad - \frac{S_DS_U}{2(S_U - S_D)} \frac{S_LS_R}{S_R - S_L}(\mathbf{U}_{R1} - \mathbf{U}_{L1} - \gamma - (\mathbf{U}_{R2} - \mathbf{U}_{L2} - \delta)) \\ & \quad + \frac{S_LS_R}{2(S_R - S_L)}(\mathbf{D}_y^{1,-} - \mathbf{D}_y^{2,-}) \\ & \quad - \frac{S_LS_R}{2(S_R - S_L)} \frac{S_DS_U}{S_U - S_D}(\mathbf{U}_{1U} - \mathbf{U}_{1D} - \alpha - (\mathbf{U}_{2U} - \mathbf{U}_{2D} - \beta)). \end{aligned}$$

Notice that

$$\mathbf{U}_{R1} - \mathbf{U}_{L1} - \gamma - (\mathbf{U}_{R2} - \mathbf{U}_{L2} - \delta) = -\beta + \alpha - \gamma + \delta = 0$$

and, similarly,

$$\mathbf{U}_{1U} - \mathbf{U}_{1D} - \alpha - (\mathbf{U}_{2U} - \mathbf{U}_{2D} - \beta) = 0.$$

Therefore, we finally deduce the following expression for \mathbf{D}^{LD} :

$$\begin{aligned} \mathbf{D}^{LD} = & \frac{S_L}{2} \left(\frac{S_L}{S_R - S_L}\mathbf{D}_y^{1,-} - \frac{S_R}{S_R - S_L}\mathbf{D}_y^{2,-} - \mathbf{D}_y^{L,-} \right) \\ & + \frac{S_D}{2} \left(\frac{S_D}{S_U - S_D}\mathbf{D}_x^{1,-} - \frac{S_U}{S_U - S_D}\mathbf{D}_x^{2,-} - \mathbf{D}_x^{D,-} \right). \end{aligned} \quad (4.3.3)$$

Reasoning in a totally analogous way, we obtain

$$\begin{aligned} \mathbf{D}^{LU} = & \frac{S_L}{2} \left(\frac{S_L}{S_R - S_L}\mathbf{D}_y^{1,+} - \frac{S_R}{S_R - S_L}\mathbf{D}_y^{2,+} - \mathbf{D}_y^{L,+} \right) \\ & - \frac{S_U}{2} \left(\frac{S_D}{S_U - S_D}\mathbf{D}_x^{1,-} - \frac{S_U}{S_U - S_D}\mathbf{D}_x^{2,-} - \mathbf{D}_x^{U,-} \right), \end{aligned} \quad (4.3.4)$$

$$\begin{aligned} \mathbf{D}^{RD} = & -\frac{S_R}{2} \left(\frac{S_L}{S_R - S_L} \mathbf{D}_y^{1,-} - \frac{S_R}{S_R - S_L} \mathbf{D}_y^{2,-} - \mathbf{D}_y^{R,-} \right) \\ & + \frac{S_D}{2} \left(\frac{S_D}{S_U - S_D} \mathbf{D}_x^{1,+} - \frac{S_U}{S_U - S_D} \mathbf{D}_x^{2,+} - \mathbf{D}_x^{D,+} \right), \end{aligned} \quad (4.3.5)$$

and

$$\begin{aligned} \mathbf{D}^{RU} = & -\frac{S_R}{2} \left(\frac{S_L}{S_R - S_L} \mathbf{D}_y^{1,+} - \frac{S_R}{S_R - S_L} \mathbf{D}_y^{2,+} - \mathbf{D}_y^{R,+} \right) \\ & - \frac{S_U}{2} \left(\frac{S_D}{S_U - S_D} \mathbf{D}_x^{1,+} - \frac{S_U}{S_U - S_D} \mathbf{D}_x^{2,+} - \mathbf{D}_x^{U,+} \right). \end{aligned} \quad (4.3.6)$$

Similar computations can be performed when the flow is supersonic in some direction. For example, is $S_L < 0 < S_R$ and $S_D > 0$, we would have

$$\begin{aligned} \mathbf{D}^{LU} = & \frac{S_U}{2} \mathbf{D}_x^{U,-} + \frac{S_U - S_D}{2} \mathbf{D}_x^{2,-} + \frac{S_D}{2} \mathbf{D}_x^{1,-} \\ & - \frac{S_L}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{LD,LU}) \frac{d}{ds} \Phi_{LD,LU}^U - \mathbf{S}_y(\Phi_{LD,LU}) \frac{d}{ds} \Phi_{LD,LU}^\sigma \right) ds \\ & + \frac{S_L}{2(S_R - S_L)} \left(S_L \int_0^1 \left(\mathbf{A}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^U - \mathbf{S}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^\sigma \right) ds \right. \\ & \left. - S_R \int_0^1 \left(\mathbf{A}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^U - \mathbf{S}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^\sigma \right) ds \right) \end{aligned}$$

and

$$\begin{aligned} \mathbf{D}^{RU} = & \frac{S_U}{2} \mathbf{D}_x^{U,+} + \frac{S_U - S_D}{2} \mathbf{D}_x^{2,+} + \frac{S_D}{2} \mathbf{D}_x^{1,+} \\ & + \frac{S_R}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{RD,RU}) \frac{d}{ds} \Phi_{RD,RU}^U - \mathbf{S}_y(\Phi_{RD,RU}) \frac{d}{ds} \Phi_{RD,RU}^\sigma \right) ds \\ & - \frac{S_R}{2(S_R - S_L)} \left(S_L \int_0^1 \left(\mathbf{A}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^U - \mathbf{S}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^\sigma \right) ds \right. \\ & \left. - S_R \int_0^1 \left(\mathbf{A}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^U - \mathbf{S}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^\sigma \right) ds \right), \end{aligned}$$

while $\mathbf{D}^{LD} = \mathbf{D}^{RD} = 0$.

A complete solution considering all the subsonic and supersonic cases is given in Appendix D. As commented in Section 4.2, in order to be consistent with the structure of the approximate solution (see Figure 4.3), the family of paths Φ must pass through the intermediate star states of the 2D Riemann problem, which means that the star states have, in general, a non-linear dependence on the coupling terms $\mathcal{B}_\zeta \equiv \mathcal{B}_{\zeta,\Phi}$. So, in order to estimate the contribution of the coupling terms in $\mathbf{D}_x^{\mu,\pm}$ and $\mathbf{D}_y^{\nu,\pm}$, we need to compute the star states through an implicit procedure, as shown in Section 4.2. Luckily, a few number of iterations is enough for convergence.

4.3.2 Multidimensional AVM solvers

For the 2D HLL solver introduced in Section 4.2, the results stated in the previous section allow to express the right hand side of the numerical scheme (4.1.9) as a linear combination of one-dimensional fluctuations. All these fluctuations can be written, for arbitrary states $\mathbf{W}_0 = (\mathbf{U}_0, \sigma_0)^t$ and $\mathbf{W}_1 = (\mathbf{U}_1, \sigma_1)^t$, under the common form

$$\begin{aligned} \mathbf{D}_x^\pm &= \frac{1}{2}(\mathbf{F}_1 - \mathbf{F}_0 + \mathbf{B}_x(\mathbf{U}_1 - \mathbf{U}_0) - \mathbf{S}_x(\sigma_1 - \sigma_0)) \\ &\quad \pm \frac{1}{2}\mathbf{Q}_x^{\text{HLL}}(\mathbf{U}_1 - \mathbf{U}_0 - \mathbf{A}_x^{-1}\mathbf{S}_x(\sigma_1 - \sigma_0)) \end{aligned} \quad (4.3.7)$$

in the x -direction, and

$$\begin{aligned} \mathbf{D}_y^\pm &= \frac{1}{2}(\mathbf{G}_1 - \mathbf{G}_0 + \mathbf{B}_y(\mathbf{U}_1 - \mathbf{U}_0) - \mathbf{S}_y(\sigma_1 - \sigma_0)) \\ &\quad \pm \frac{1}{2}\mathbf{Q}_y^{\text{HLL}}(\mathbf{U}_1 - \mathbf{U}_0 - \mathbf{A}_y^{-1}\mathbf{S}_y(\sigma_1 - \sigma_0)) \end{aligned} \quad (4.3.8)$$

in the y -direction. Remember that \mathbf{A}_ζ , \mathbf{B}_ζ , \mathbf{S}_ζ depend on \mathbf{U}_0 and \mathbf{U}_1 , and also on the family of paths Φ . On the other hand, the viscosity matrix $\mathbf{Q}_x^{\text{HLL}} = \alpha_0\mathbf{I} + \alpha_1\mathbf{A}_x$ can be expressed as $\mathbf{Q}_x = p(\mathbf{A}_x)$, where $p(x) = \alpha_0 + \alpha_1x$ and, similarly, $\mathbf{Q}_y^{\text{HLL}} = q(\mathbf{A}_y)$, with $q(x) = \beta_0 + \beta_1x$. Therefore, the numerical fluctuations (4.3.7)-(4.3.8) can be viewed as particular cases of AVM solvers based on the first-order polynomials $p(x)$ and $q(x)$, respectively. These considerations allow to define general two-dimensional AVM solvers by substituting the viscosity matrices $\mathbf{Q}_x^{\text{HLL}}$ and $\mathbf{Q}_y^{\text{HLL}}$ by proper functional approximations of the form $\mathbf{Q}_x = f(\mathbf{A}_x)$ and $\mathbf{Q}_y = g(\mathbf{A}_y)$, respectively.

For the sake of clarity, the detailed construction of a 2D AVM solver is described in what follows:

- Consider functional approximations $f(x)$ and $g(x)$ to the absolute value function, verifying the stability condition (1.5.5). In the numerical tests, we have taken $f \equiv g$.
- Use $f(x)$ and \mathbf{A}_x to define the viscosity matrix $\mathbf{Q}_x^{\text{AVM}}$ as in (1.5.4), and similarly with $g(x)$ and \mathbf{A}_y to define $\mathbf{Q}_y^{\text{AVM}}$.
- Compute the 1D contributions given by (4.3.1) and (4.3.2) with $\mathbf{Q}_\zeta^{\text{AVM}}$ instead of $\mathbf{Q}_\zeta^{\text{HLL}}$, for $\zeta = x, y$. Use them to define the 2D vertex contributions (4.3.3)-(4.3.6).
- Similarly, define the corresponding 1D AVM fluctuations \mathbf{D}^\pm at edges, using (4.3.7) and (4.3.8) with $\mathbf{Q}_\zeta^{\text{AVM}}$ instead of $\mathbf{Q}_\zeta^{\text{HLL}}$, for $\zeta = x, y$.
- Finally, use the 1D and 2D contributions previously computed to assemble the numerical scheme (4.1.9).

Only the subsonic case has been considered here, as the adaptation to include supersonic cases is straightforward following the guidelines in Section 4.2.1 (See also Appendix D).

4.3.3 Modified equation and linear stability

In this section we perform a linear stability analysis of the genuinely two-dimensional Riemann solver (4.1.9), for which we focus on the linear advection equation

$$w_t + aw_x + bw_y = 0.$$

Assume that $S_R = |a|$, $S_L = -|a|$, $S_U = |b|$ and $S_D = -|b|$, so we are always in the subsonic case. The shrink parameters for the one-dimensional contributions at each edge are given by $\beta_L = \beta_R = \Delta y - 2|b|\Delta t$ and $\beta_D = \beta_U = \Delta x - 2|a|\Delta t$. Departing from the splitting writing in Section 4.3.1, after some straightforward calculations it is possible to write the scheme (4.1.9) as

$$\begin{aligned} w_{ij}^{n+1} = & w_{ij}^n - \frac{\Delta t}{\Delta x} \left[\left(1 - 2|b|\frac{\Delta t}{\Delta y} \right) (D_{i-1/2,j}^+ + D_{i+1/2,j}^-) \right. \\ & + |b|\frac{\Delta t}{2\Delta y} (D_x^{D,-} + D_x^{*, -})_{i+1/2,j+1/2} + |b|\frac{\Delta t}{2\Delta y} (D_x^{U,-} + D_x^{*, -})_{i+1/2,j-1/2} \\ & + |b|\frac{\Delta t}{2\Delta y} (D_x^{D,+} + D_x^{*, +})_{i-1/2,j+1/2} + |b|\frac{\Delta t}{2\Delta y} (D_x^{U,+} + D_x^{*, +})_{i-1/2,j-1/2} \left. \right] \\ & - \frac{\Delta t}{\Delta y} \left[\left(1 - 2|a|\frac{\Delta t}{\Delta x} \right) (D_{i,j-1/2}^+ + D_{i,j+1/2}^-) \right. \\ & + |a|\frac{\Delta t}{2\Delta x} (D_y^{L,-} + D_y^{*, -})_{i+1/2,j+1/2} + |a|\frac{\Delta t}{2\Delta x} (D_y^{L,+} + D_y^{*, +})_{i+1/2,j-1/2} \\ & + |a|\frac{\Delta t}{2\Delta x} (D_y^{R,-} + D_y^{*, -})_{i-1/2,j+1/2} + |a|\frac{\Delta t}{2\Delta x} (D_y^{R,+} + D_y^{*, +})_{i-1/2,j-1/2} \left. \right], \end{aligned}$$

where we have used * instead of 1 or 2 to identify the intermediate one-dimensional HLL states. It is easy to check that

$$D_x^\pm = \frac{a \pm |a|}{2}(w_1 - w_0), \quad D_y^\pm = \frac{b \pm |b|}{2}(w_1 - w_0)$$

for arbitrary states w_0 and w_1 , so

$$\begin{aligned} \frac{w_{ij}^{n+1} - w_{ij}^n}{\Delta t} = & -\frac{1}{2\Delta x} \left(1 - 2|b|\frac{\Delta t}{\Delta y} \right) (a(w_{i+1,j} - w_{i-1,j}) - |a|(w_{i+1,j} - 2w_{ij} + w_{i-1,j})) \\ & -\frac{1}{2\Delta y} \left(1 - 2|a|\frac{\Delta t}{\Delta x} \right) (b(w_{i,j+1} - w_{i,j-1}) - |b|(w_{i,j+1} - 2w_{ij} + w_{i,j-1})) \\ & + \frac{(a - |a|)(b - |b|)\Delta t}{4\Delta x\Delta y} w_{i+1,j+1} - \frac{(a - |a|)(b + |b|)\Delta t}{4\Delta x\Delta y} w_{i+1,j-1} \\ & - \frac{(a + |a|)(b - |b|)\Delta t}{4\Delta x\Delta y} w_{i-1,j+1} + \frac{(a + |a|)(b + |b|)\Delta t}{4\Delta x\Delta y} w_{i-1,j-1} \\ & - \frac{|a|(b - |b|)\Delta t}{2\Delta x\Delta y} w_{i,j+1} - \frac{|b|(a - |a|)\Delta t}{2\Delta x\Delta y} w_{i+1,j} \\ & + \frac{|a|(b + |b|)\Delta t}{2\Delta x\Delta y} w_{i,j-1} + \frac{|b|(a + |a|)\Delta t}{2\Delta x\Delta y} w_{i-1,j} - 3\frac{|a||b|\Delta t}{\Delta x\Delta y} w_{ij}. \end{aligned}$$

Therefore, the corresponding modified equation would be

$$w_t + aw_x + bw_y = \frac{|a|\Delta x}{2}w_{xx} + \frac{|b|\Delta y}{2}w_{yy} + \frac{\Delta t}{4}\tilde{w}_{xy},$$

where the anti-diffusion term \tilde{w}_{xy} is given by

$$\begin{aligned}\tilde{w}_{xy} = & (a - |a|)(b - |b|)(w_{xy})_{i+1/2,j+1/2} + (a - |a|)(b + |b|)(w_{xy})_{i+1/2,j-1/2} \\ & + (a + |a|)(b - |b|)(w_{xy})_{i-1/2,j+1/2} + (a + |a|)(b + |b|)(w_{xy})_{i-1/2,j-1/2} \\ & + \mathcal{O}(\Delta x^2, \Delta y^2).\end{aligned}$$

The effect of this term is shown in Figure 4.5, where a diagonal advected Gaussian profile is shown. As it can be observed, the anti-diffusion term allows to preserve the symmetry of the initial data.

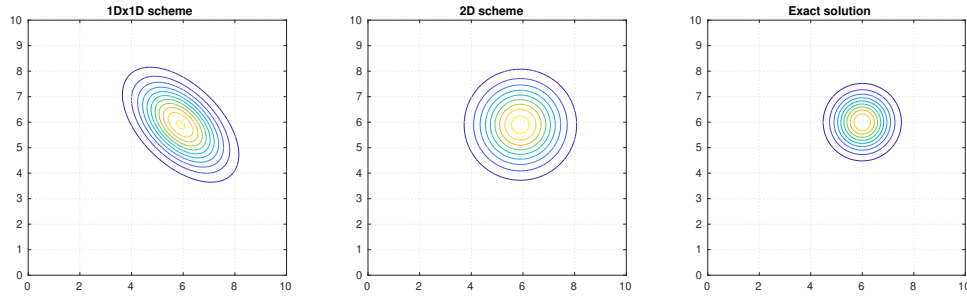


Figure 4.5: Diagonal advection. Left: 1D×1D scheme. Center: 2D scheme. Right: exact solution.

On the other hand, observe that the coefficients of the vertex contributions $w_{i\pm 1,j\pm 1}$ are clearly positive. The coefficients of $w_{i,j\pm 1}$ are positive under the condition $\Delta t \leq \frac{\Delta x}{|a|}$, and the coefficients of $w_{i\pm 1,j}$ are positive if $\Delta t \leq \frac{\Delta y}{|b|}$. Finally, the coefficients of w_{ij} are positive if

$$1 - \Delta t \left(\frac{|b|}{\Delta y} + \frac{|a|}{\Delta x} \right) + \Delta t^2 \frac{|a||b|}{\Delta x \Delta y} \geq 0,$$

which is accomplished if

$$\Delta t \leq \min \left(\frac{\Delta x}{|a|}, \frac{\Delta y}{|b|} \right). \quad (4.3.9)$$

A similar analysis can be performed for supersonic cases. As a result, we have that

$$\max_{ij} \|w_{ij}^{n+1}\|_{\infty} \leq \max_{ij} \|w_{ij}^n\|_{\infty},$$

which implies the following result.

Proposition 2. *The scheme (4.1.9) is linearly L^{∞} -stable under the stability condition (4.3.9).*

The choice of the time step for nonlinear problems is given by the CFL condition

$$\Delta t = \delta \min_{ij} \Delta t_{ij}, \quad \delta \in (0, 1], \quad (4.3.10)$$

with each local time step Δt_{ij} being defined as

$$\Delta t_{ij} = \frac{\min(\Delta x, \Delta y)}{\sqrt{(\lambda_{ij}^x)^2 + (\lambda_{ij}^y)^2}}, \quad (4.3.11)$$

where λ_{ij}^α denotes the maximal speed of propagation at cell C_{ij} in the α -direction, taking into account both the speeds at the edges and those arising in the two-dimensional Riemann problems at vertices. In practice, the two-dimensional contributions allow the use of a maximal CFL number δ of unity, whereas for the equivalent 1D \times 1D schemes the maximum CFL is 0.5.

4.3.4 Second-order schemes

In this section we describe a second-order extension of the scheme (4.1.9), both in space and time, using a MUSCL-Hancock procedure following the guidelines in [75]. Spatial reconstructions are performed by means of piecewise bilinear operators of the form

$$P_{ij}^t(\mathbf{x}) = \alpha_0 + \alpha_1(x - x_i) + \alpha_2(y - y_j) + \alpha_3(x - x_i)(y - y_j), \quad (4.3.12)$$

combined with a minmod limiter, similarly as it was done in [74]. The predictor step approximates the solution at a half-time level: the reconstructed states are given by

$$\begin{aligned} \widetilde{\mathbf{W}}_{i\pm 1/2, j}^{n+1/2, \mp} &= \mathbf{W}_{ij}^n \pm \frac{1}{2} \Delta_x \mathbf{W}_{ij}^n + \frac{1}{2} \Delta t \partial_t \mathbf{W}_{ij}^n, \\ \widetilde{\mathbf{W}}_{i, j\pm 1/2}^{n+1/2, \mp} &= \mathbf{W}_{ij}^n \pm \frac{1}{2} \Delta_y \mathbf{W}_{ij}^n + \frac{1}{2} \Delta t \partial_t \mathbf{W}_{ij}^n \end{aligned}$$

for the midpoint of edges, and

$$\widetilde{\mathbf{W}}_{i\pm 1/2, j\pm 1/2}^{n+1/2, \mp, \mp} = \mathbf{W}_{ij}^n \pm \frac{1}{2} \Delta_x \mathbf{W}_{ij}^n \pm \frac{1}{2} \Delta_y \mathbf{W}_{ij}^n + \left(\pm \frac{1}{2}\right) \left(\pm \frac{1}{2}\right) \Delta_{xy} \mathbf{W}_{ij}^n + \frac{1}{2} \Delta t \partial_t \mathbf{W}_{ij}^n$$

for the vertices. The time derivative is obtained by means of a Cauchy-Kowalevski procedure:

$$\partial_t \mathbf{W}_{ij}^n = -\mathcal{A}_x(\mathbf{W}_{ij}^n) \Delta_x \mathbf{W}_{ij}^n - \mathcal{A}_y(\mathbf{W}_{ij}^n) \Delta_y \mathbf{W}_{ij}^n,$$

and the spatial derivatives are computed as

$$\begin{aligned} \Delta_x \mathbf{W}_{ij}^n &= \text{minmod}(\theta \delta_x \mathbf{W}_{ij}^n, \theta \delta_x \mathbf{W}_{i+1, j}^n, \mu_x \delta_x \mathbf{W}_{i+1, j}^n), \\ \Delta_y \mathbf{W}_{ij}^n &= \text{minmod}(\theta \delta_y \mathbf{W}_{ij}^n, \theta \delta_y \mathbf{W}_{i, j+1}^n, \mu_y \delta_y \mathbf{W}_{i, j+1}^n), \\ \Delta_{xy} \mathbf{W}_{ij}^n &= \text{minmod}\left(\frac{1}{2} \theta \delta_x \delta_y (\mathbf{W}_{i+1, j+1}^n + \mathbf{W}_{ij}^n), \frac{1}{2} \theta \delta_x \delta_y (\mathbf{W}_{i+1, j}^n + \mathbf{W}_{i, j+1}^n), \mu_x \mu_y \delta_x \delta_y \mathbf{W}_{i+1, j+1}^n\right), \end{aligned}$$

with $\theta \in [1, 2]$, $\mu_x \mathbf{W}_{ij}^n := \frac{1}{2}(\mathbf{W}_{ij}^n + \mathbf{W}_{i-1,j}^n)$ and $\delta_x \mathbf{W}_{ij}^n := \mathbf{W}_{ij}^n - \mathbf{W}_{i-1,j}^n$, and similarly for δ_y and μ_y . Finally, after a second half-time step, and dropping the σ component, the corrector step is given by

$$\begin{aligned} \mathbf{U}_{ij}^{n+1} = & \mathbf{U}_{ij}^n - \frac{\Delta t}{\Delta x \Delta y} \left[\tilde{\beta}_L \tilde{\mathbf{D}}_{i-1/2,j}^+ + \tilde{\beta}_R \tilde{\mathbf{D}}_{i+1/2,j}^- + \tilde{\beta}_D \tilde{\mathbf{D}}_{i,j-1/2}^+ + \tilde{\beta}_U \tilde{\mathbf{D}}_{i,j+1/2}^- \right. \\ & + \Delta t (\tilde{\mathbf{D}}_{i-1/2,j-1/2}^{RU} + \tilde{\mathbf{D}}_{i+1/2,j-1/2}^{LU} + \tilde{\mathbf{D}}_{i-1/2,j+1/2}^{RD} + \tilde{\mathbf{D}}_{i+1/2,j+1/2}^{LD}) \\ & + \Delta_y (\tilde{\mathbf{F}}_{i+1/2,j} - \tilde{\mathbf{F}}_{i-1/2,j}) + \Delta_x (\tilde{\mathbf{G}}_{i,j+1/2} - \tilde{\mathbf{G}}_{i,j-1/2}) \\ & + \Delta_y \mathcal{B}_x(\tilde{\mathbf{U}}_{ij}^{n+1/2}) \Delta_x \tilde{\mathbf{U}}_{ij}^n + \Delta_x \mathcal{B}_y(\tilde{\mathbf{U}}_{ij}^{n+1/2}) \Delta_y \tilde{\mathbf{U}}_{ij}^n \\ & \left. - \Delta_y \mathcal{S}_x(\tilde{\mathbf{U}}_{ij}^{n+1/2}) \Delta_x \tilde{\sigma}_{ij}^n - \Delta_x \mathcal{S}_y(\tilde{\mathbf{U}}_{ij}^{n+1/2}) \Delta_y \tilde{\sigma}_{ij}^n \right], \end{aligned} \quad (4.3.13)$$

where the coefficients $\tilde{\beta}$ are computed as in (4.1.10) using reconstructed states; $\tilde{\mathbf{U}}_{ij}^{n+1/2}$ are the reconstructed values at the cell center computed in the predictor step,

$$\tilde{\mathbf{U}}_{ij}^{n+1/2} = \mathbf{U}_{ij}^n + \frac{1}{2} \Delta t \partial_t \mathbf{U}_{ij}^n; \quad (4.3.14)$$

the fluctuations are given by

$$\begin{aligned} \tilde{\mathbf{D}}_{i\pm 1/2,j}^\pm &= \mathbf{D}_{\text{AVM}}^\pm(\tilde{\mathbf{U}}_{i\pm 1/2,j}^{n+1/2,-}, \tilde{\mathbf{U}}_{i\pm 1/2,j}^{n+1/2,+}), \\ \tilde{\mathbf{D}}_{i,j\pm 1/2}^\pm &= \mathbf{D}_{\text{AVM}}^\pm(\tilde{\mathbf{U}}_{i,j\pm 1/2}^{n+1/2,-}, \tilde{\mathbf{U}}_{i,j\pm 1/2}^{n+1/2,+}), \\ \tilde{\mathbf{D}}_V^{AB} &= \mathbf{D}_{\text{AVM}}^{AB}(\tilde{\mathbf{U}}_V^{n+1/2,+,+}, \tilde{\mathbf{U}}_V^{n+1/2,+,-}, \tilde{\mathbf{U}}_V^{n+1/2,-,+}, \tilde{\mathbf{U}}_V^{n+1/2,-,-}), \end{aligned} \quad (4.3.15)$$

for $AB \in \{LD, LU, RD, RU\}$ and $V = (x_{i\pm 1/2}, y_{j\pm 1/2})$, where \mathbf{D}_{AVM} denotes the fluctuations associated to a given AVM solver; finally, $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{G}}$ are the physical fluxes evaluated at the reconstructed states.

In the original works of Balsara ([14, 15]), the 2D Riemann solver was applied both in the predictor and corrector steps with the consequence of rising the maximal CFL number. Later, Vides et al. ([203]) noticed that this is only necessary for certain problems, so in general it suffices to apply a 1D \times 1D method in the predictor step. In the present work we have adopted the latter technique, so we always use the 1D \times 1D Riemann solver in the predictor step, combined with the polynomial reconstruction (4.3.12) which includes cross derivatives only in the corrector step. This allows to use a maximal CFL number of unity in the numerical experiments, reducing at the same time the total computational cost. In addition, it has also been found in the simulations that isotropy is preserved.

As commented previously, another important issue is that the class of 2D AVM-type solvers only need a bound of the maximal wave speeds in order to compute the viscosity matrices of the scheme. This means that we could reduce always to the subsonic case, just taking S_R as the absolute maximal wave speed and $S_L = -S_R$, and similarly in the y -direction. This choice can help to reduce the computational cost of the scheme; in practice, no significative differences have been found with respect to 2D AVM solvers taking into account supersonic cases.

4.4 Numerical results

In order to test the performances of the proposed multidimensional AVM-type schemes, we present a number of numerical experiments in this section. On the one hand, we test the differences between the 2D solvers and their projected 1D×1D counterparts and, on the other hand, we compare the simple 2D HLL scheme with more precise 2D AVM-type schemes.

As it has already been commented for the conservative case in Section 3.3, the maximal and minimal wave speeds at each direction are taken from the expressions (3.1.6) and (3.1.7). In particular, for the intermediate speeds $\bar{\lambda}_\alpha(\mathbf{U}_0, \mathbf{U}_1)$ appearing in (3.1.5) we have simply chosen the speeds $\lambda_\alpha(\tilde{\mathbf{U}})$ at the intermediate state $\tilde{\mathbf{U}} = \frac{\mathbf{U}_0 + \mathbf{U}_1}{2}$. Another possible choice, more expensive, is to consider $\tilde{\mathbf{U}}$ as the Roe state associated to the states \mathbf{U}_0 and \mathbf{U}_1 . No noticeable differences have been found in our experiments between both choices. The time step is computed by the CFL condition (4.3.10). The accuracy test 4.4.1 and the C -property test 4.4.2 have been run with CFL 1.0, while for the remainder tests a CFL number of 0.9 has been considered.

We will denote simply by *Chebyshev* the PVM solver based on an eighth-degree Chebyshev polynomial, and by *Newman* the RVM solver based on a Newman rational function of degree 8/8. For each AVM method, its two-dimensional version is denoted as 2D, whereas 1D×1D is the projected one-dimensional version.

All the presented results have been computed with the second-order predictor-corrector method, which can be shortly summarized as:

- Predictor step: Using spatial reconstructed states at each edge of the cell into the 1D×1D Riemann solver, by a Cauchy-Kowalevski procedure compute the predicted half-time evaluation of the solution.
- Corrector step: Using the spatial reconstruction operator (4.3.12) and the half-time solution previously calculated, compute the four states that go into the 2D Riemann solver. In this step we also evaluate the volume integrals appearing in the scheme in Section 4.3.4.

4.4.1 Second-order accuracy

In order to test the accuracy of the proposed schemes for the one-layer and two-layer 2D shallow water systems, let us first build exact steady state solutions. This can be achieved by imposing rotational symmetry ($\frac{\partial}{\partial \theta} = 0$) in cylindrical coordinates r - θ ; this procedure is described in [78] for the two-layer case with flat bottom. The resultant ODEs read, respectively, as

$$gh \frac{\partial}{\partial r}(h - H) = \frac{(v_1^\theta)^2 h}{r}, \quad (4.4.1)$$

and

$$\begin{cases} gh_1 \frac{\partial}{\partial r}(h_1 + h_2 - H) = \frac{(v_1^\theta)^2 h_1}{r}, \\ gh_2 \frac{\partial}{\partial r}(h_2 + \rho h_1 - H) = \frac{(v_2^\theta)^2 h_2}{r}, \end{cases} \quad (4.4.2)$$

where v_i^θ denote the velocities in the angular direction θ . Now, we need to define velocities verifying the compatibility conditions (4.4.1) and (4.4.2). Following [78], we can easily check

that

$$v_i^\theta = rv_{i0}e^{s_i(1-r^2)}$$

satisfies $v_i^\theta \rightarrow 0$ for $r \rightarrow \infty$, and consequently the compatibility conditions. In order to solve (4.4.1) and (4.4.2) we still need to set appropriate boundary conditions for the conserved height variables. We can choose $h(0) = h_0$ for the one-layer case and $h_i(0) = h_{i0}$ for the two-layer case. With these data, we can solve the nonlinear ODEs (4.4.1) and (4.4.2), obtaining

$$\varphi := h(r) = \frac{v_{10}^2 e^{2s_1}}{4gs_1} (1 - e^{-2s_1 r^2}) + h_0 + H(r) - H(0),$$

and

$$\begin{aligned} \varphi_1 := h_1(r) &= \frac{v_{10}^2 s_2 e^{2s_1} (1 - e^{-2s_1 r^2}) - v_{20}^2 s_1 e^{2s_2} (1 - e^{-2s_2 r^2})}{4gs_1 s_2 (1 - \rho)} + h_{10}, \\ \varphi_2 := h_2(r) &= \frac{v_{20}^2 s_1 e^{2s_2} (1 - e^{-2s_2 r^2}) - \rho v_{10}^2 s_2 e^{2s_1} (1 - e^{-2s_1 r^2})}{4gs_1 s_2 (1 - \rho)} + h_{20} + H(r) - H(0). \end{aligned}$$

The computational domain has been taken as $[-5, 5] \times [-5, 5]$ with periodic boundary conditions. We set $g = 10$, density ratio $\rho = 0.9$, and take

$$H(r) = 2 - 0.2e^{-r^2}.$$

The other parameters are chosen as $s_1 = 2$, $v_{10} = 1$ and $h_0 = 1.5$ for the one-layer case, and $s_1 = 0.5$, $s_2 = 1$, $v_{10} = 0.75$, $v_{20} = 0.1$, and $h_{10} = h_{20} = 1$ for the two-layer case. The steady state solutions are finally given by

$$\mathbf{W} = (\varphi, -v_1^\theta \sin(\theta), v_1^\theta \cos(\theta), -H(r)),$$

and

$$\mathbf{W} = (\varphi_1, -v_1^\theta \sin(\theta), v_1^\theta \cos(\theta), \varphi_2, -v_2^\theta \sin(\theta), v_2^\theta \cos(\theta), -H(r)),$$

for the one-layer and two-layer 2D shallow water systems, respectively. The results of the numerical convergence study are summarized in Tables 4.1 and 4.2, where the maximal CFL number of unity has been used. The solutions were computed until a final time $T = 2$ with different meshes of size $N \times N$, for $N = 25, 50, 100, 200, 400, 800$, and no limiters were applied. As expected, although the 2D PVM based on Chebyshev polynomials and the 2D RVM based on Newman rational functions are slightly better than two-dimensional HLL scheme, the solvers produce similar results, mainly due to the smoothness of the particular solutions considered. Moreover, it can be observed that all the solvers present second-order accuracy in space and time.

4.4.2 C-property

This test was designed in [43] to verify numerically the C -property in shallow water systems. The bathymetry function $H(\mathbf{x})$ is taken as a smooth function with a random perturbation (see Figure 4.6). The initial condition for the one-layer case is given by $h(\mathbf{x}, 0) = H(\mathbf{x})$ and $\mathbf{q} = 0$,

Scheme	N	$\ h\ _1$		$\ q_x\ _1$		$\ q_y\ _1$	
		Error	Order	Error	Order	Error	Order
HLL 2D	25	1.39e-04	–	1.62e-03	–	1.62e-03	–
	50	7.42e-05	0.90	7.16e-04	1.18	7.16e-04	1.18
	100	1.74e-05	2.09	1.42e-04	2.33	1.42e-04	2.33
	200	2.61e-06	2.74	2.32e-05	2.62	2.32e-05	2.62
	400	3.77e-07	2.79	3.31e-06	2.81	3.31e-06	2.81
	800	6.53e-08	2.53	5.23e-07	2.66	5.23e-07	2.66
Chebyshev 2D	25	9.52e-05	–	1.04e-03	–	1.04e-03	–
	50	3.72e-05	1.36	3.21e-04	1.70	3.21e-04	1.70
	100	7.39e-06	2.33	5.59e-05	2.52	5.59e-05	2.52
	200	1.21e-06	2.61	8.99e-06	2.64	8.99e-06	2.64
	400	2.16e-07	2.48	1.55e-06	2.54	1.55e-06	2.54
	800	4.80e-08	2.17	3.27e-07	2.24	3.27e-07	2.24
Newman 2D	25	9.38e-05	–	1.02e-03	–	1.02e-03	–
	50	3.52e-05	1.41	3.04e-04	1.74	3.04e-04	1.74
	100	7.03e-06	2.32	5.32e-05	2.51	5.32e-05	2.51
	200	1.16e-06	2.59	8.63e-06	2.62	8.63e-06	2.62
	400	2.11e-07	2.46	1.50e-06	2.52	1.50e-06	2.52
	800	4.74e-08	2.15	3.22e-07	2.23	3.22e-07	2.23

Table 4.1: Test 4.4.1. L^1 -errors and orders of accuracy obtained with the 2D AVM schemes for the conserved variables of the one-layer shallow water system.

Scheme	N	$\ h_1\ _1$		$\ q_{x,1}\ _1$		$\ q_{y,1}\ _1$		$\ h_2\ _2$		$\ q_{x,2}\ _2$		$\ q_{y,2}\ _2$	
		Error	Order	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order
HLL 2D	25	1.63e-04	–	4.47e-04	–	4.47e-04	–	1.51e-04	–	7.32e-05	–	7.32e-05	–
	50	3.09e-05	2.40	8.81e-05	2.34	8.81e-05	2.34	2.82e-05	2.42	1.97e-05	1.90	1.97e-05	1.90
	100	5.17e-06	2.58	1.41e-05	2.64	1.41e-05	2.64	4.75e-06	2.57	3.31e-06	2.57	3.31e-06	2.57
	200	7.54e-07	2.78	2.00e-06	2.82	2.00e-06	2.82	6.86e-07	2.79	4.60e-07	2.85	4.60e-07	2.85
	400	1.17e-07	2.69	3.03e-07	2.72	3.03e-07	2.72	1.05e-07	2.71	6.42e-08	2.84	6.42e-08	2.84
	800	2.41e-08	2.28	5.99e-08	2.34	5.99e-08	2.34	2.06e-08	2.35	1.07e-08	2.58	1.07e-08	2.58
Chebyshev 2D	25	4.93e-05	–	1.31e-04	–	1.31e-04	–	4.50e-05	–	3.90e-05	–	3.90e-05	–
	50	9.43e-06	2.39	2.33e-05	2.49	2.33e-05	2.49	8.54e-06	2.40	6.59e-06	2.56	6.59e-06	2.56
	100	1.70e-06	2.47	3.69e-06	2.66	3.69e-06	2.66	1.52e-06	2.49	9.53e-07	2.79	9.53e-07	2.79
	200	3.33e-07	2.36	7.02e-07	2.39	7.02e-07	2.39	2.91e-07	2.39	1.48e-07	2.69	1.48e-07	2.69
	400	7.67e-08	2.12	1.68e-07	2.06	1.68e-07	2.06	6.49e-08	2.16	2.93e-08	2.33	2.93e-08	2.33
	800	2.06e-08	1.89	4.23e-08	1.99	4.23e-08	1.99	1.74e-08	1.90	7.00e-09	2.07	7.00e-09	2.07
Newman 2D	25	4.40e-05	–	1.17e-04	–	1.17e-04	–	4.02e-05	–	3.73e-05	–	3.73e-05	–
	50	8.37e-06	2.39	2.03e-05	2.53	2.03e-05	2.53	7.59e-06	2.40	6.00e-06	2.64	6.00e-06	2.64
	100	1.55e-06	2.44	3.23e-06	2.65	3.23e-06	2.65	1.38e-06	2.46	9.06e-07	2.73	9.06e-07	2.73
	200	3.22e-07	2.27	6.53e-07	2.31	6.53e-07	2.31	2.76e-07	2.32	1.48e-07	2.62	1.48e-07	2.62
	400	7.62e-08	2.08	1.63e-07	2.00	1.63e-07	2.00	6.40e-08	2.11	3.02e-08	2.29	3.02e-08	2.29
	800	2.05e-08	1.89	4.11e-08	1.99	4.11e-08	1.99	1.73e-08	1.89	6.89e-09	2.13	6.89e-09	2.13

Table 4.2: Test 4.4.1. L^1 -errors and orders of accuracy obtained with the 2D AVM schemes for the conserved variables of the 2D two-layer shallow water system.

Scheme	$\ h\ _1$	$\ q_x\ _1$	$\ q_y\ _1$
HLL 2D	2.46e-20	1.11e-19	1.51e-19
Chebyshev 2D	2.46e-20	1.11e-19	1.51e-19
Newman 2D	2.46e-20	1.11e-19	1.51e-19

Table 4.3: Test 4.4.2. Verification of C -property for the one-layer SWE.

Scheme	$\ h_1\ _1$	$\ q_{x,1}\ _1$	$\ q_{y,1}\ _1$	$\ h_2\ _2$	$\ q_{x,2}\ _2$	$\ q_{y,2}\ _2$
HLL 2D	9.18e-20	5.09e-20	1.85e-19	9.36e-20	9.77e-19	8.62e-19
Chebyshev 2D	1.11e-19	9.38e-20	1.06e-19	1.04e-19	6.61e-19	9.86e-19
Newman 2D	1.29e-19	2.01e-19	2.06e-19	1.32e-19	1.36e-18	1.35e-18

Table 4.4: Test 4.4.2. Verification of C -property for the two-layer SWE.

while for two-layer case is taken as $h_1(\mathbf{x}, 0) = 0.4$, $h_2(\mathbf{x}, 0) = H(\mathbf{x})$ and $\mathbf{q}_1 = \mathbf{q}_2 = 0$. We set a 100×100 mesh grid over a square domain $[0, 1] \times [0, 1]$ with periodic boundary conditions. The ratio of densities is set to $\rho = 0.998$, $g = 9.81$ and the CFL number is 1. Tables 4.3 and 4.4 show the L^1 -error at time $T = 5$. As expected, all the numerical schemes preserve the water at rest solution up to machine accuracy.

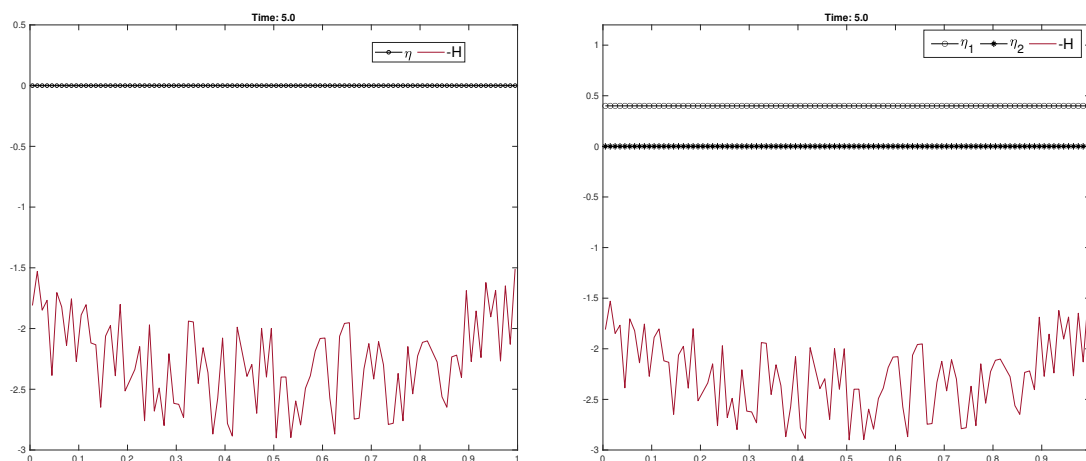


Figure 4.6: Test 4.4.2. Cut along the main diagonal of the steady state solution.

4.4.3 Applications to the one-layer shallow water system

4.4.3.1 Circular dam-break

This test consists of a circular dam-break in a square domain $[-5, 5] \times [-5, 5]$ with wall boundary conditions. The depth function is given by

$$H(\mathbf{x}) = 4 - 1.5e^{-r^2},$$

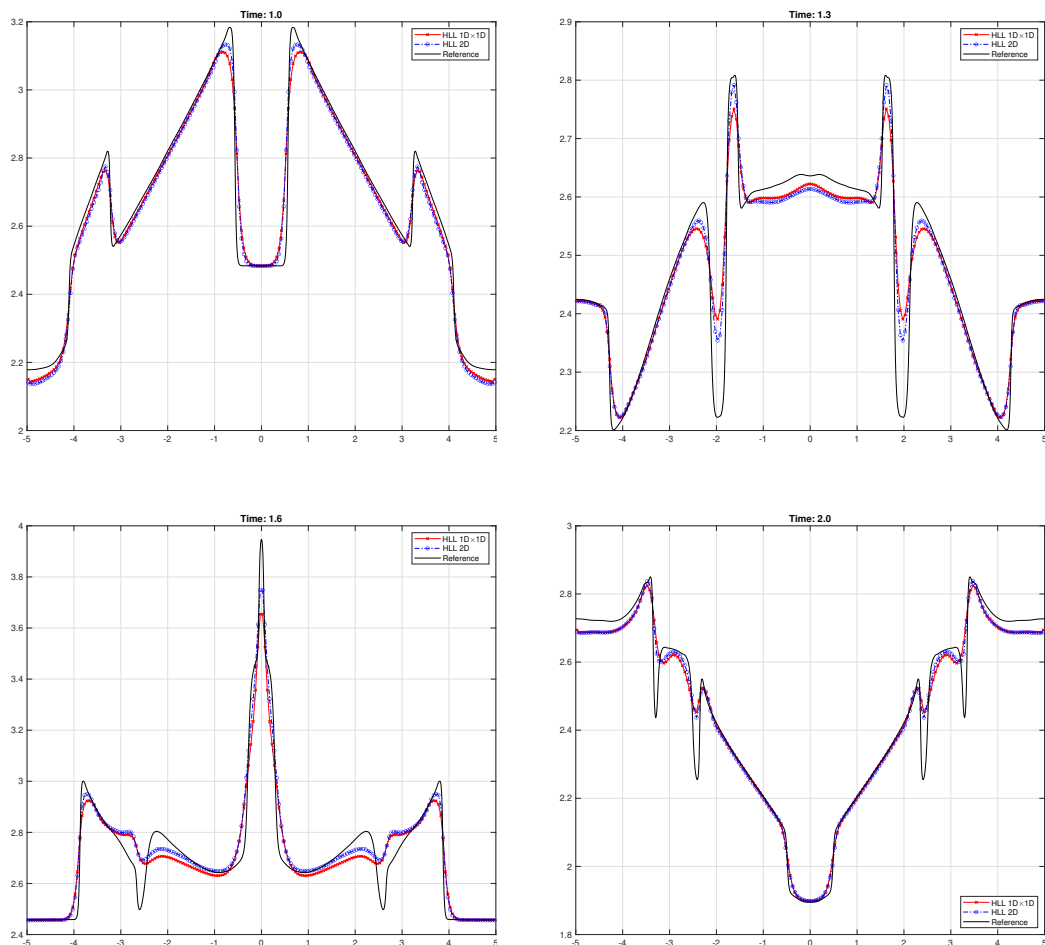


Figure 4.7: Test 4.4.3.1. Diagonal cuts of the free surface at different times, computed with the 2D HLL Riemann solver on a 200×200 mesh.

where $r^2 = x^2 + y^2$, and for the initial condition we take $\mathbf{q} = 0$ and $h(\mathbf{x}, 0) = H(\mathbf{x}) + f(\mathbf{x})$, with

$$f(\mathbf{x}) = \begin{cases} 5 & \text{if } r^2 < 1, \\ 2.5 & \text{otherwise.} \end{cases}$$

We run the test with $\text{CFL}=0.9$ until the final time $T = 2$, and the spatial domain is discretized with 200×200 grid points. Figure 4.7 shows a comparison between the $1\text{D} \times 1\text{D}$ and the genuinely 2D versions of the HLL Riemann solver: it can be clearly seen the improvement of including vertex contributions in the Riemann solver. In Figure 4.8 we plot the contour lines of the free surface: the results are in good agreement with those presented in [159].

We should remark that, in this problem, the PVM and RVM Riemann solvers do not present significant improvements compared with HLL. Even though the 2D one-layer SWE is a 3-waves model and the middle waves are ignored by HLL, we only found tiny differences in the

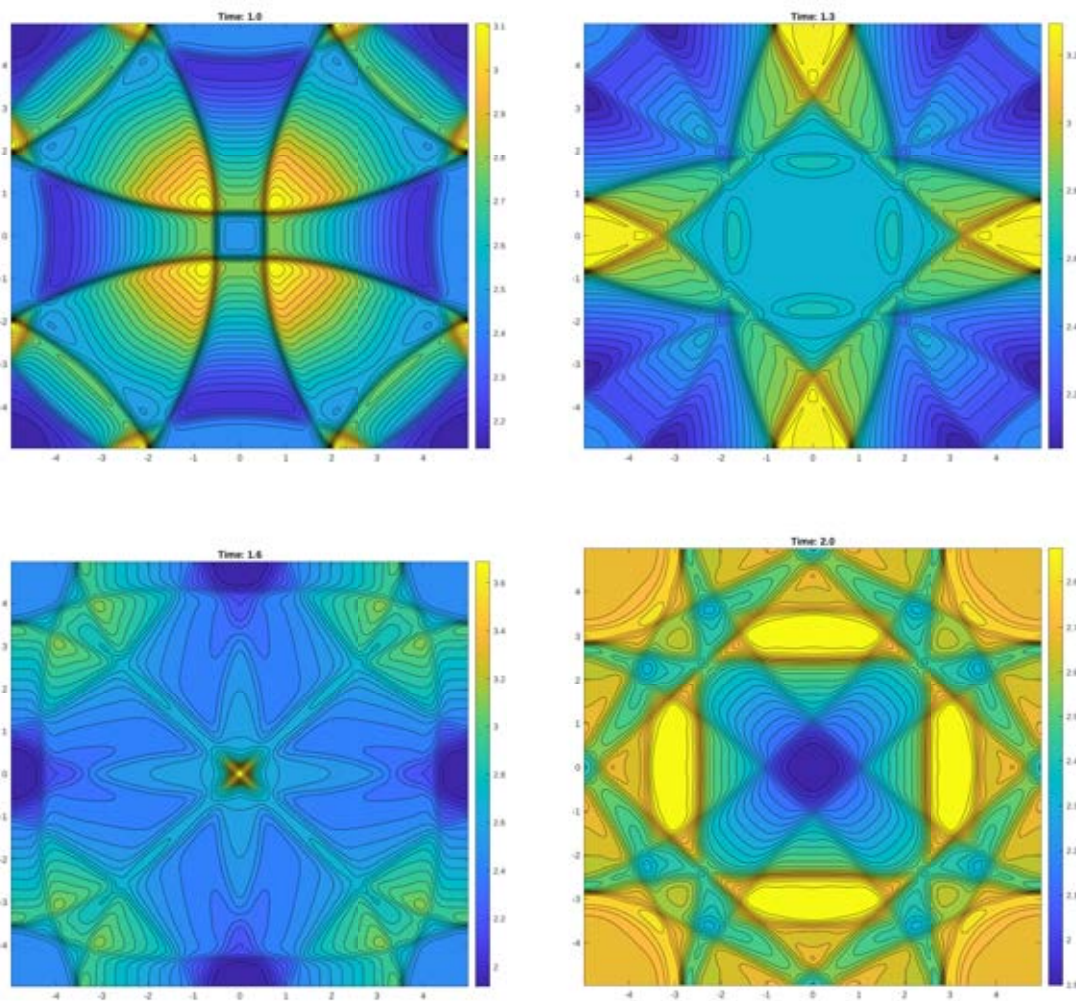


Figure 4.8: Test 4.4.3.1. Contour plots of the free surface at different times, obtained with the 2D HLL Riemann solver on a 200×200 mesh.

test case presented here. On the other hand, we do have found significant differences between the $1D \times 1D$ and the 2D Riemann solvers.

We have also done a slightly change in the test, considering an extended domain $[-12, 12] \times [-12, 12]$ with free boundary conditions, and a final time $T = 1$. A coarse mesh of 80×80 grid points and a refined mesh of 400×400 grid points have been considered. In Figure 4.9 we can see that the 2D Riemann solver preserves the isotropy of the solution, whereas the $1D \times 1D$ solver suffers a loss of isotropy on the upper and lower extremes of the wave propagation for both the coarse and the refined meshes.

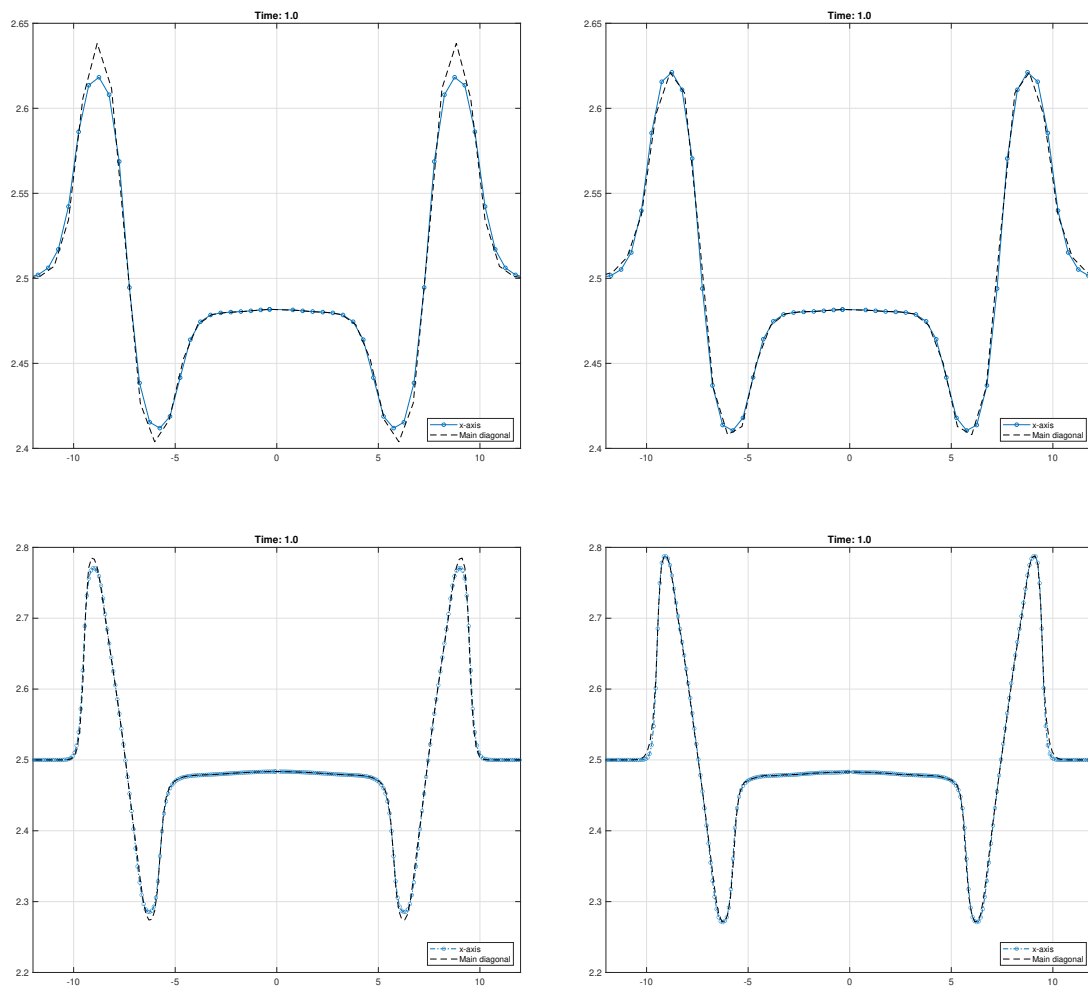


Figure 4.9: Test 4.4.3.1. Cuts of the free surface along the x -axis and in the main diagonal direction, using an 80×80 coarse mesh (left top: HLL 1D \times 1D; right top: HLL 2D), and a 400×400 refined mesh (left down: HLL 1D \times 1D; right down: HLL 2D).

4.4.3.2 Non-linear breaking waves

This test, originally presented in [187], simulates rapid spatial and temporal deformations of the free surface. The sharper gradients formed by the nonlinear waves allow to see significant differences when we consider two-dimensional vertex contributions in the numerical scheme. A

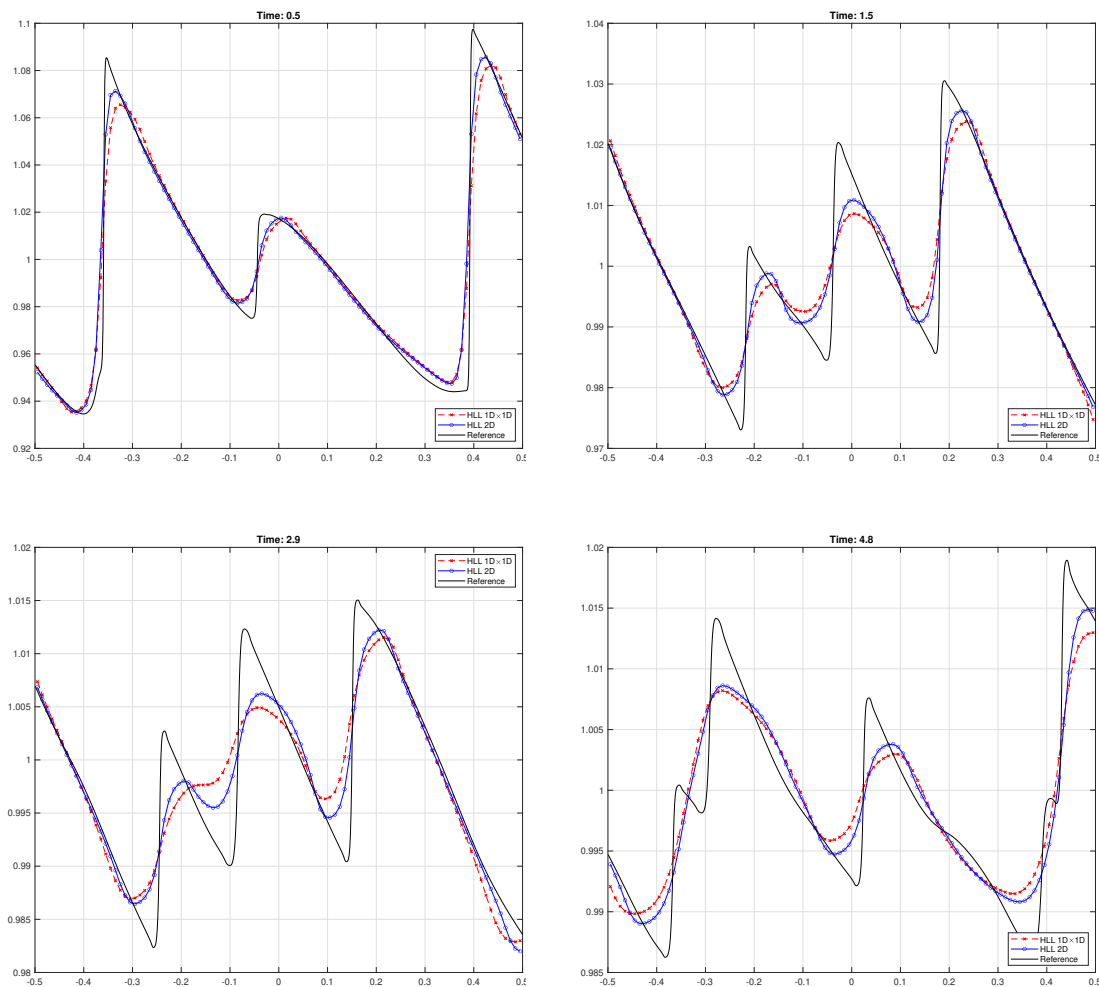


Figure 4.10: Test 4.4.3.2. Cuts of the free surface along the main diagonal at different times. Comparison between the 1D \times 1D and the 2D HLL Riemann solvers.

flat bottom is considered and the initial condition is given by

$$\begin{aligned}
 h(\mathbf{x}) &= 1 + A \sin(l y) \sin(k x), \\
 u_x(\mathbf{x}) &= -\frac{k A g}{\omega} \sin(l y) \sin(k x), \\
 u_y(\mathbf{x}) &= \frac{l A g}{\omega} \cos(l y) \cos(k x),
 \end{aligned}$$

where the parameters are taken as $k = 2\pi m$, $l = (2n + 1)\pi$, $\omega^2 = g(k^2 + l^2)$, $A = 0.1$, $g = 1$, $m = 2$ and $n = 0$. A 100×100 meshgrid over a square domain $[-0.5, 0.5] \times [-0.5, 0.5]$ has been taken, with wall boundary conditions at $y = \pm 0.5$ and periodic boundary conditions in the x -direction.

Figure 4.10 shows a cut along the main diagonal comparing the 1D \times 1D and 2D HLL schemes with 200×200 grid points at different times. Near the sharper gradients, it is visible that the two-dimensional contributions increase the accuracy of the numerical scheme.

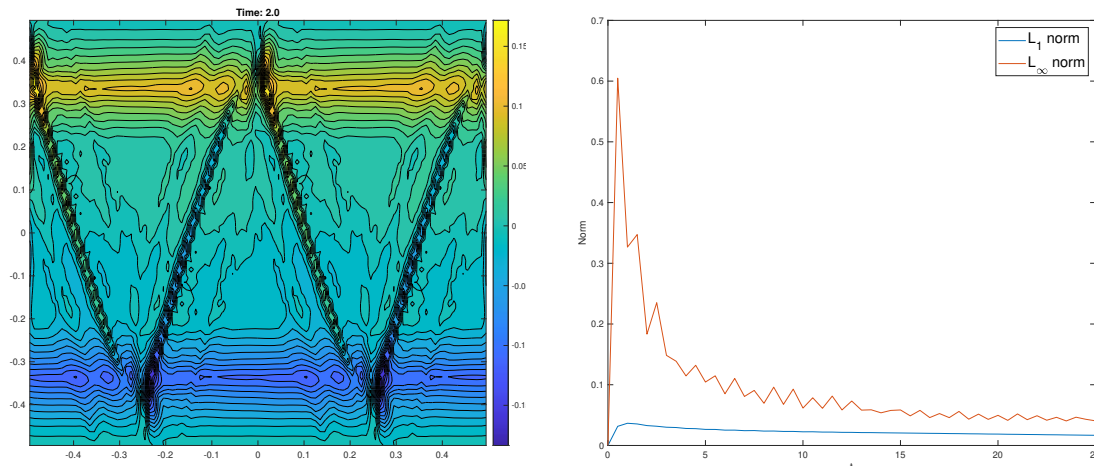


Figure 4.11: Test 4.4.3.2 Left: Potential vorticity contour plot at time $T = 2$. Right: L^1 and L^∞ norms until time $T = 25$.

A conserved quantity that the numerical scheme should preserve is the potential vorticity (PV), defined as $\Pi = (\nabla \times \mathbf{u})/h$. In Figure 4.11 (left) we show the contour plot of the potential vorticity at time $T = 2$, which is in good agreement with the one presented in [207]. We also present, in Figure 4.11 (right), the L^1 and L^∞ norms of Π : it can be seen that, after the higher amplitudes around $T \approx 0.5$, the L^∞ -norm decreases and remains almost constant as time progresses. The L^1 -norm shows that the numerical scheme conserves the potential vorticity.

4.4.4 Applications to the two-layer shallow water system

4.4.4.1 Internal circular dam-break

Let us consider an internal circular dam-break problem over a flat bottom $H(\mathbf{x}) = 2$, initially proposed in [43]. The initial condition is given by

$$h_1(\mathbf{x}, 0) = \begin{cases} 1.8 & \text{if } x^2 + y^2 > 4, \\ 0.2 & \text{otherwise,} \end{cases}$$

$h_2(\mathbf{x}, 0) = 2 - h_1(\mathbf{x}, 0)$ and $u_x = u_y = 0$, on the computational domain $[-5, 5] \times [-5, 5]$ discretized with a 200×200 mesh. The CFL number is set to 0.9 and $g = 9.81$.

The ratio of densities is taken as $\rho = 0.998$ in order to set closer to zero the internal eigenvalues of the two-layer system (see Section 2.3). As expected, this choice makes more evident the differences between RVM and PVM Riemann solvers, because the rational approximations of $|x|$ are much closer to zero than the polynomial approximations of the same

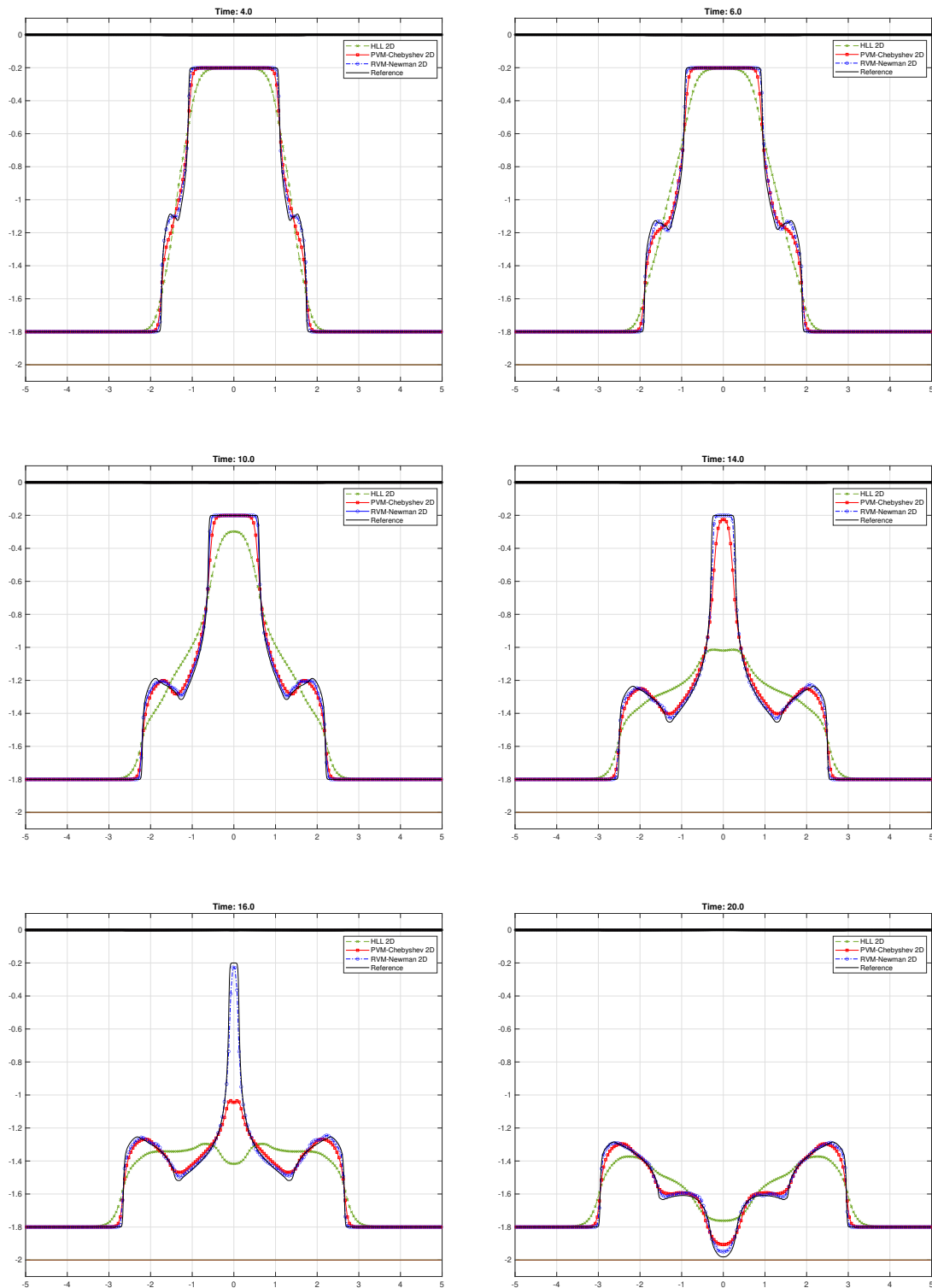


Figure 4.12: Test 4.4.4.1. Cuts along the main diagonal at different times, computed on a 200×200 mesh grid.



order. In Figure 4.12 there are shown cuts along the main diagonal of the free surface, the interface and the bottom topography, comparing the two-dimensional versions of HLL, PVM-Chebyshev and RVM-Newman schemes, for different computational times. It is worth noticing the significant differences when approximating the internal slow waves.

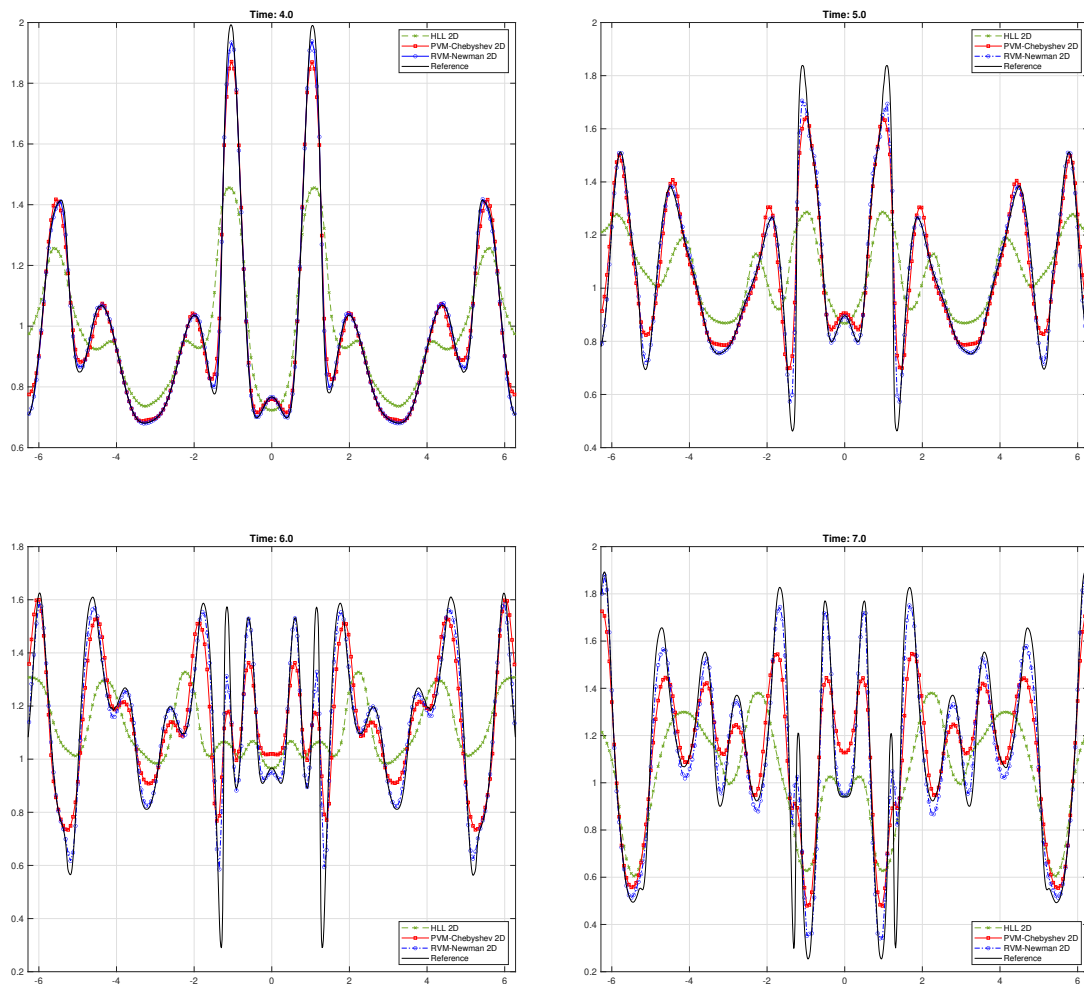


Figure 4.13: Test 4.4.4.2. Cuts of the interface along the main diagonal at different times.

4.4.4.2 Evolution to a complex state

In order to test the robustness of the 2D AVM Riemann solvers, we propose here a new test showing the transition to a complex state. Initially, it is assumed that the free surface and the interface are constants, $\eta_1 = 5$ and $\eta_2 = 1$. The velocity fields are defined as

$$v_{x1} = -0.9A \sin(y), \quad v_{y1} = 0.9B \sin(x), \quad v_{x2} = -A \sin(y), \quad v_{y2} = B \sin(x),$$

in such a way that there are not big differences in the velocity gradient between the two layers. We take the parameters $A = 1$ and $B = 0.25$. In addition, the bottom topography is given by

$$H(x, y) = 2 - 0.5e^{-2(x^2+y^2)}.$$

We impose wall boundary conditions in the y -direction and periodic ones in the x -direction. The spatial domain is taken as $[-2\pi, 2\pi] \times [-2\pi, 2\pi]$. Finally, we take $g = 9.81$ and the ratio of densities $\rho = 0.998$.

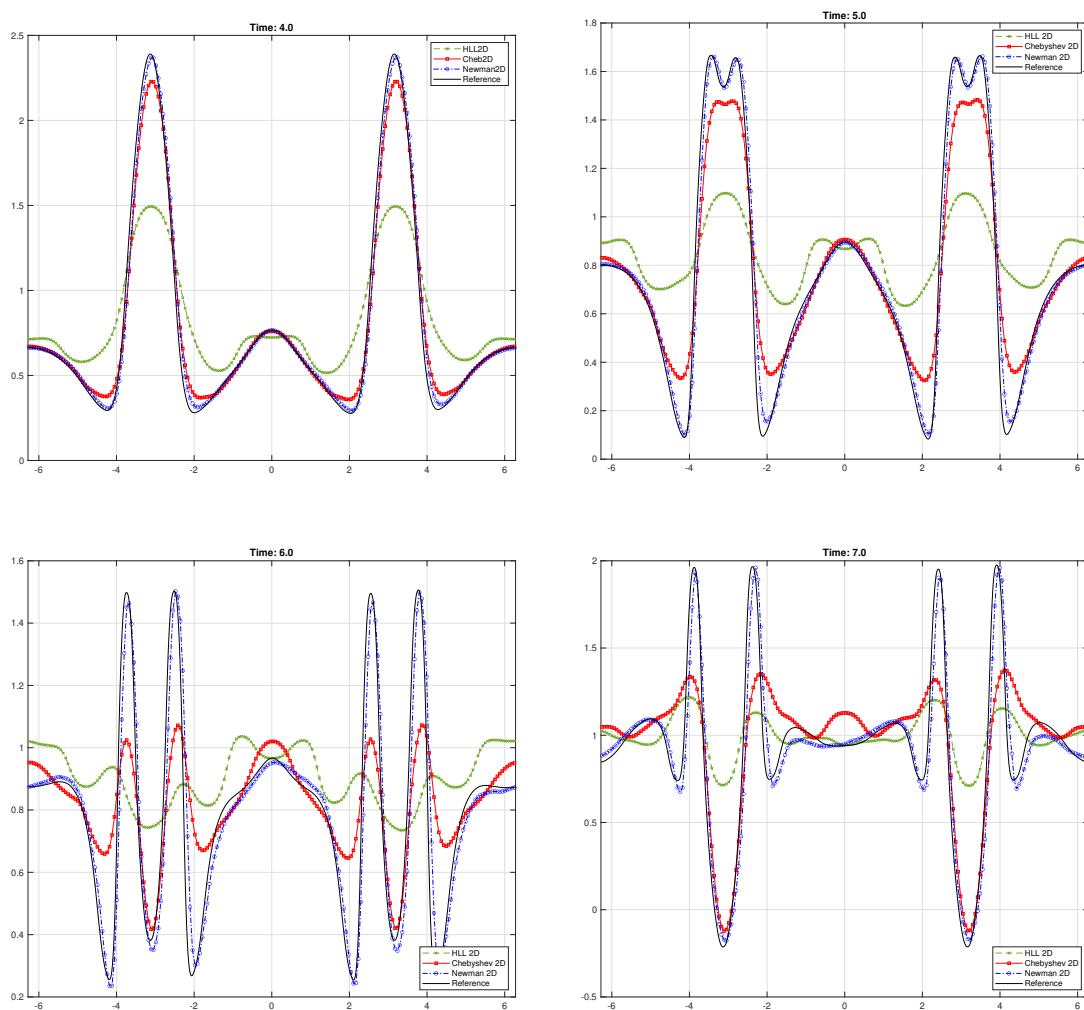


Figure 4.14: Test 4.4.4.2. Cuts of the interface along the x -axis at different times.

In Figures 4.13 and 4.14 we show cuts along the main diagonal and the x -axis, respectively, comparing the 2D HLL scheme, the 2D PVM Chebyshev and the 2D RVM Newman schemes. The solutions have been computed on a 200×200 mesh with CFL number 0.9. It is clear from the figures that the AVM solvers provide much better results than the HLL solver and, also,

that the rational-based solver gives better approximations than the polynomial-based scheme. This can also be seen in Figure 4.15, where contour plots of the interface at different times are compared. As a conclusion, this test clearly shows that the proposed 2D AVM solvers are robust, running stable and accurately for CFL up to unity in complex problems and for a ratio of densities close to 1.

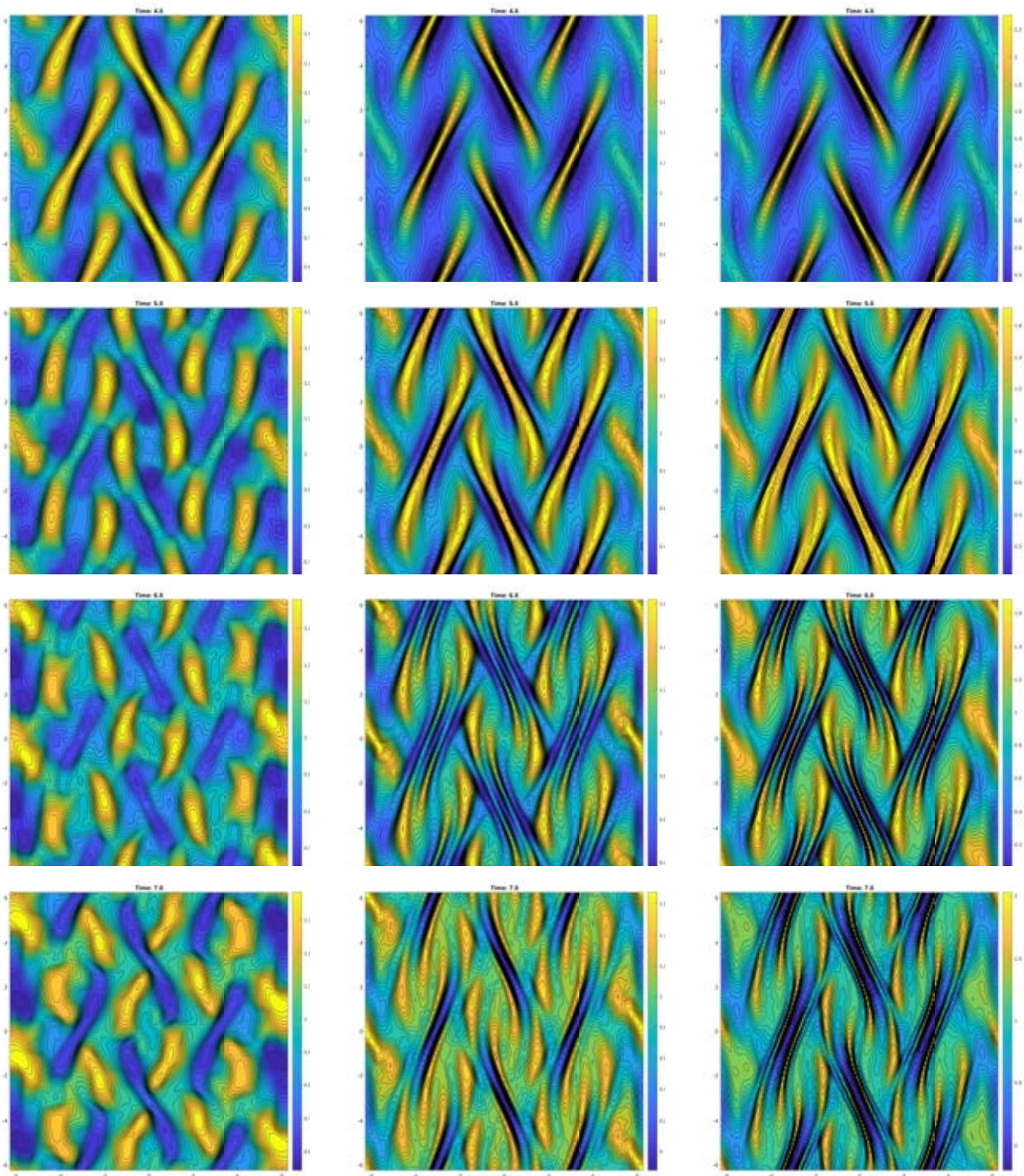


Figure 4.15: Test 4.4.4.2. Contour plots of the interface at times $t = 4, 5, 6, 7$, from top to bottom. Left: 2D HLL; center: 2D PVM Chebyshev; right: 2D RVM Newman.

Chapter 5

Efficient GPU implementation

This chapter deals with the development of an efficient implementation based on Graphics Processing Units (GPUs) architectures for the genuinely two-dimensional Riemann solvers presented in this thesis, whose results are based on a recent and ongoing work [177]. GPU codes can provide dramatic speedups over their counterpart sequential codes ([9, 10, 54]). Parallelism appears to be a key ingredient to increase performance, which is important in realistic applications where big computational domains and long time simulations are necessary. See, for instance, the webpage of the EDANYA research group ([87]) and the publications therein, where real-life problems as avalanches, tsunami propagation, ocean currents, hazard assessment, among others, are considered.

Architecturally, GPUs are a kind of vectorial processors, i.e., they are able to perform the same task over multiple portions of data simultaneously. The GPU computation consists in the combined use of CPU/GPU in a co-processing system, in which data with high parallelism degree are executed in GPU and the rest of the code is executed in CPU. This arrangement gives support to the High Performance Computation (HPC) at a very affordable price, due to the mass production for the entertainment market. An interesting overview about GPU programming is given in [33].

Since 2006, the company NVIDIA has made available the CUDA programming toolkit ([156]), which allows the use of the C language, offering a set of additional functions in order to access and manage data processing on GPU, avoiding the programmer having to understand the internal details of the GPU. Efficient GPU implementations, based on CUDA programming, of finite volume solvers with applications to shallow water systems have been presented, for instance, in [7, 9, 10, 11, 8, 54, 92, 101, 124, 159]. Specific CUDA solvers for one-layer and two-layer systems, using two-dimensional Roe-type solvers based on projected one-dimensional splitting, are described, respectively, in [10] and [9]. CUDA solvers for MHD equations can be found, for instance, in [23, 208, 209].

In this chapter we revisit the genuinely 2D AVM-type schemes presented in this thesis and propose an efficient implementation of the two-dimensional numerical algorithm on GPU architectures. Our numerical schemes present a high degree of potential data parallelism, as well as the systems of shallow water and MHD flows, which suggests the design of parallel codes in order to solve and analyze these systems in reasonable execution times.

5.1 Rewriting of the numerical schemes

In this section we present a brief description of the first- and second-order two-dimensional AVM schemes derived in (4.1.9) and (4.3.13), respectively, in order to make them easier to implement. This reinterpretation allows also to unify the implementations of the two-dimensional scheme for conservation laws introduced in Chapter 3 and the nonconservative path-conservative schemes discussed in Chapter 4. Let us now focus on the general case

$$\partial_t \mathbf{W} + \mathcal{A}_x(\mathbf{W})\partial_x \mathbf{W} + \mathcal{A}_y(\mathbf{W})\partial_y \mathbf{W} = 0, \quad (5.1.1)$$

where \mathbf{W} , \mathcal{A}_x and \mathcal{A}_y are defined in (1.3.2). Considering the splitting AVM formulation given in Sections 4.3.1 and 4.3.2, we can share the vertex contributions in a suited combination of fluctuations in the coordinate directions. Therefore, in an analogous fashion as the two-dimensional scheme for conservation laws in Chapter 3, we can rewrite the numerical fluctuations at a given edge of the first-order scheme (4.1.9) as a linear convex combination of a one-dimensional contribution at the edge and two 2D contributions at vertices. Both the first- and second-order schemes can be put in the form

$$\mathbf{U}_{ij}^{n+1} = \mathbf{U}_{ij}^n + \mathbf{M}_{ij}^n, \quad (5.1.2)$$

where we have dropped the $\boldsymbol{\sigma}$ component of $\mathbf{W} = (\mathbf{U}, \boldsymbol{\sigma})^t$. The vector \mathbf{M}_{ij}^n represents the fluctuations of all the edge and vertex contributions for a cell C_{ij} at time t^n , and it has the same dimension as \mathbf{U}_{ij}^n . For each \mathbf{M}_{ij}^n , we have twelve contributions (see Figure 5.1 for clarity), four from the Riemann problems arising at the edges and eight from the 2D Riemann problems at the vertices. Mimicking the numerical fluxes assembling in (3.1.3), the fluctuations for the first-order scheme (4.1.9) can be written as

$$\begin{aligned} \mathbf{M}_{ij}^n = & -\frac{\Delta t}{\Delta x} \left\{ \alpha_{x,i-1/2,j+1/2}^R \mathbf{D}_{x,i-1/2,j+1/2}^{RD} + \beta_{x,i-1/2}^R \mathbf{D}_{i-1/2,j}^+ + \gamma_{x,i-1/2,j-1/2}^R \mathbf{D}_{x,i-1/2,j-1/2}^{RU} \right\} \\ & -\frac{\Delta t}{\Delta x} \left\{ \alpha_{x,i+1/2,j+1/2}^L \mathbf{D}_{x,i+1/2,j+1/2}^{LD} + \beta_{x,i+1/2}^L \mathbf{D}_{i+1/2,j}^- + \gamma_{x,i+1/2,j-1/2}^L \mathbf{D}_{x,i+1/2,j-1/2}^{LU} \right\} \\ & -\frac{\Delta t}{\Delta y} \left\{ \alpha_{y,i+1/2,j-1/2}^U \mathbf{D}_{y,i+1/2,j-1/2}^{LU} + \beta_{y,j-1/2}^U \mathbf{D}_{i,j-1/2}^+ + \gamma_{y,i-1/2,j-1/2}^U \mathbf{D}_{y,i-1/2,j-1/2}^{RU} \right\} \\ & -\frac{\Delta t}{\Delta y} \left\{ \alpha_{y,i+1/2,j+1/2}^D \mathbf{D}_{y,i+1/2,j+1/2}^{LD} + \beta_{y,j+1/2}^D \mathbf{D}_{i,j+1/2}^- + \gamma_{y,i-1/2,j+1/2}^D \mathbf{D}_{y,i-1/2,j+1/2}^{RD} \right\}, \end{aligned} \quad (5.1.3)$$

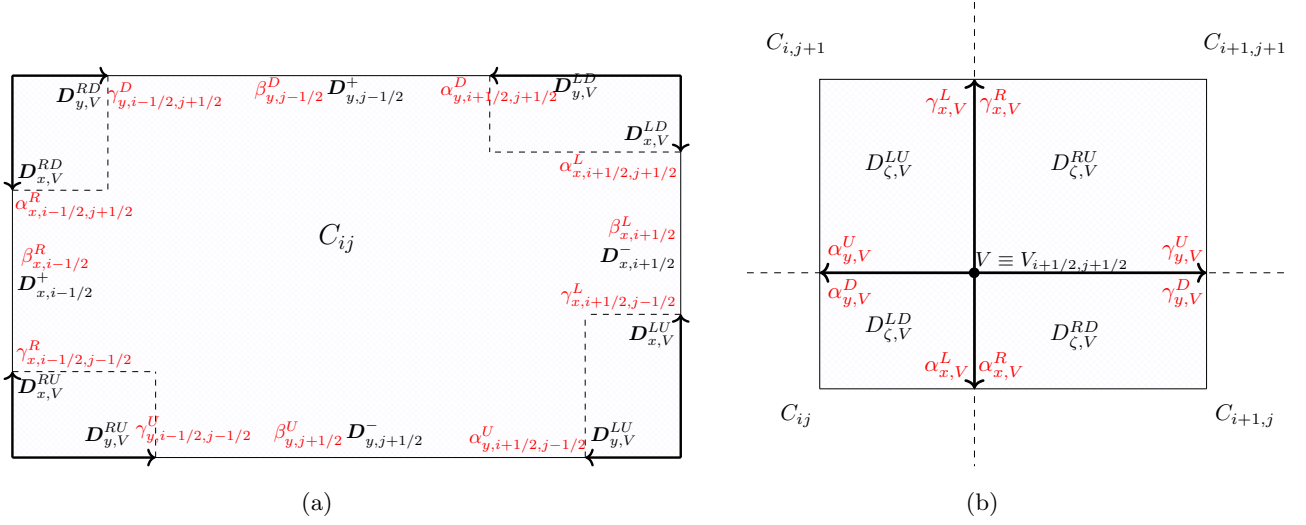


Figure 5.1: (a) Global notation for the scheme (5.1.2)-(5.1.3). The twelve contributions to the cell C_{ij} are represented. (b) Local notation for a vertex.

where the vertex fluctuations are given by

$$\begin{aligned}
 D_x^{LD} &= -\frac{S_D}{S_U - S_D} D_x^{1,-} + \frac{S_U}{S_U - S_D} D_x^{2,-}, \\
 D_y^{LD} &= -\frac{S_L}{S_R - S_L} D_y^{1,-} + \frac{S_R}{S_R - S_L} D_y^{2,-}, \\
 D_x^{LU} &= -\frac{S_D}{S_U - S_D} D_x^{1,+} + \frac{S_U}{S_U - S_D} D_x^{2,+}, \\
 D_y^{LU} &= -\frac{S_L}{S_R - S_L} D_y^{1,+} + \frac{S_R}{S_R - S_L} D_y^{2,+}, \\
 D_x^{RD} &= -\frac{S_D}{S_U - S_D} D_x^{1,+} + \frac{S_U}{S_U - S_D} D_x^{2,+}, \\
 D_y^{RD} &= -\frac{S_L}{S_R - S_L} D_y^{1,-} + \frac{S_R}{S_R - S_L} D_y^{2,-}, \\
 D_x^{RU} &= -\frac{S_D}{S_U - S_D} D_x^{1,+} + \frac{S_U}{S_U - S_D} D_x^{2,+}, \\
 D_y^{RU} &= -\frac{S_L}{S_R - S_L} D_y^{1,+} + \frac{S_R}{S_R - S_L} D_y^{2,+},
 \end{aligned} \tag{5.1.4}$$

and

$$\begin{aligned}
\beta_x^R &= 1 + (S_{i-1/2,j+1/2}^D - S_{i-1/2,j-1/2}^U) \frac{\Delta t}{2\Delta y}, & \beta_x^L &= 1 + (S_{i+1/2,j+1/2}^D - S_{i+1/2,j-1/2}^U) \frac{\Delta t}{2\Delta y}, \\
\beta_x^U &= 1 + (S_{i+1/2,j-1/2}^L - S_{i-1/2,j-1/2}^R) \frac{\Delta t}{2\Delta x}, & \beta_x^D &= 1 + (S_{i+1/2,j+1/2}^L - S_{i-1/2,j+1/2}^R) \frac{\Delta t}{2\Delta x}, \\
\alpha_x^R &= -S_{i-1/2,j+1/2}^D \frac{\Delta t}{2\Delta y}, & \alpha_x^L &= -S_{i+1/2,j+1/2}^D \frac{\Delta t}{2\Delta y}, \\
\alpha_y^U &= -S_{i+1/2,j-1/2}^L \frac{\Delta t}{2\Delta x}, & \alpha_y^D &= -S_{i+1/2,j+1/2}^L \frac{\Delta t}{2\Delta x}, \\
\gamma_x^R &= S_{i-1/2,j-1/2}^U \frac{\Delta t}{2\Delta y}, & \gamma_x^L &= S_{i+1/2,j-1/2}^U \frac{\Delta t}{2\Delta y}, & \gamma_y^U &= S_{i-1/2,j-1/2}^R \frac{\Delta t}{2\Delta x}, & \gamma_y^D &= S_{i-1/2,j+1/2}^R \frac{\Delta t}{2\Delta x},
\end{aligned} \tag{5.1.5}$$

with $\mathbf{D}_x^{\mu,\pm}$ and $\mathbf{D}_y^{\nu,\pm}$ defined by (4.3.1) and (4.3.2), respectively. Figure 5.1 helps to clarify the notation. In a similar way, the fluctuations for the second-order scheme read as

$$\begin{aligned}
\mathbf{M}_{ij}^n &= -\frac{\Delta t}{\Delta x} \left\{ \tilde{\alpha}_{x,i-1/2,j+1/2}^R \tilde{\mathbf{D}}_{x,i-1/2,j+1/2}^{RD} + \tilde{\beta}_{x,i-1/2}^R \tilde{\mathbf{D}}_{i-1/2,j}^+ + \tilde{\gamma}_{x,i-1/2,j-1/2}^R \tilde{\mathbf{D}}_{x,i-1/2,j-1/2}^{RU} \right\} \\
&\quad -\frac{\Delta t}{\Delta x} \left\{ \tilde{\alpha}_{x,i+1/2,j+1/2}^L \tilde{\mathbf{D}}_{x,i+1/2,j+1/2}^{LD} + \tilde{\beta}_{x,i+1/2}^L \tilde{\mathbf{D}}_{i+1/2,j}^- + \tilde{\gamma}_{x,i+1/2,j-1/2}^L \tilde{\mathbf{D}}_{x,i+1/2,j-1/2}^{LU} \right\} \\
&\quad -\frac{\Delta t}{\Delta y} \left\{ \tilde{\alpha}_{y,i+1/2,j-1/2}^U \tilde{\mathbf{D}}_{y,i+1/2,j-1/2}^{LU} + \tilde{\beta}_{y,j-1/2}^U \tilde{\mathbf{D}}_{i,j-1/2}^+ + \tilde{\gamma}_{y,i-1/2,j-1/2}^U \tilde{\mathbf{D}}_{y,i-1/2,j-1/2}^{RU} \right\} \\
&\quad -\frac{\Delta t}{\Delta y} \left\{ \tilde{\alpha}_{y,i+1/2,j+1/2}^D \tilde{\mathbf{D}}_{y,i+1/2,j+1/2}^{LD} + \tilde{\beta}_{y,j+1/2}^D \tilde{\mathbf{D}}_{i,j+1/2}^- + \tilde{\gamma}_{y,i-1/2,j+1/2}^D \tilde{\mathbf{D}}_{y,i-1/2,j+1/2}^{RD} \right\} \\
&\quad -\frac{\Delta t}{\Delta x} \left\{ \tilde{\mathbf{F}}_{i+1/2,j} - \tilde{\mathbf{F}}_{i-1/2,j} + \mathcal{B}_x(\tilde{\mathbf{U}}_{ij}^{n+1/2}) \Delta_x \tilde{\mathbf{U}}_{ij}^n - \mathcal{S}_x(\tilde{\mathbf{U}}_{ij}^{n+1/2}) \Delta_x \tilde{\sigma}_{ij}^n \right\} \\
&\quad -\frac{\Delta t}{\Delta y} \left\{ \tilde{\mathbf{G}}_{i,j+1/2} - \tilde{\mathbf{G}}_{i,j-1/2} + \mathcal{B}_y(\tilde{\mathbf{U}}_{ij}^{n+1/2}) \Delta_y \tilde{\mathbf{U}}_{ij}^n - \mathcal{S}_y(\tilde{\mathbf{U}}_{ij}^{n+1/2}) \Delta_y \tilde{\sigma}_{ij}^n \right\},
\end{aligned} \tag{5.1.6}$$

where the coefficients $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}$ are computed as in (5.1.5), but with vertex speeds calculated by using time-space reconstructed states; $\tilde{\mathbf{U}}_{ij}^{n+1/2}$ are the reconstructed values in time at the center zone computed as in (4.3.14); $\tilde{\mathbf{D}}^\pm$ are the 1D fluctuations on the cell boundaries evaluated at the time-space reconstructed states; and the vertex fluctuations are calculated as in (4.3.15), changing $\tilde{\mathbf{D}}_{\zeta,V}^{AB}$ by the vertex fluctuations $\tilde{\mathbf{D}}_{\zeta,V}^{AB}$ in (5.1.4), for $\zeta = x, y$ and $AB = LD, LU, RD, RU$; finally, $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{G}}$ are the physical fluxes evaluated at the reconstructed states in space and time.

5.1.1 Improving efficiency of AVM schemes

The idea behind of AVM schemes (Section 1.5) is to use a functional evaluation of a Roe matrix of the system, $f(\mathbf{A}_\zeta) \approx |\mathbf{A}_\zeta|$, where $\mathbf{A}_\zeta = \mathbf{A}_x \eta_x + \mathbf{A}_y \eta_y$, if available, or more generally the Jacobian matrix evaluated in a given averaged state, in order to compute the numerical viscosity matrix of the numerical scheme. For the PVM and RVM cases, this involves the computation of power matrices. Particularly, for the proposed 2D AVM schemes, this leads to several matrix multiplications, which can reduce the computational efficiency of the schemes. Considering the splitting formulation of the 2D scheme presented in Sections 4.3.1 and 4.3.2, for each vertex we need to apply the one-dimensional Riemann solver eight times, four in each direction. So, while for the 1D×1D AVM solver, for a $N_x \times N_y$ meshgrid, we need to compute $(N_x + 1)N_y + N_x(N_y + 1)$ times the one-dimensional Riemann solver, for the 2D versions we have additional $4(N_x + 1)(N_y + 1)$ applications for that Riemann solver in the vertex contributions, which represents many additional matrix multiplications.

An idea to circumvent this problem is to make use of the Cayley-Hamilton theorem (see, for instance, [118]), that allows to powerfully simplify matrix multiplications. As the Roe matrix of the system (or the approximated Jacobian matrix) is a square matrix, calculating the characteristic polynomial $p(\lambda) = \det(\lambda \mathcal{I} - \mathbf{A}_\zeta)$, where \mathcal{I} represents the identity matrix, the theorem ensures that $p(\mathbf{A}_\zeta) = \mathbf{0}$. So, at least from certain degree m , the order of the characteristic polynomial, we can switch the matrix power \mathbf{A}_ζ^m by a sum of matrices, i.e.,

$$\mathbf{A}_\zeta^m = -c_{m-1} \mathbf{A}_\zeta^{m-1} - \dots - c_1 \mathbf{A}_\zeta - c_0 \mathcal{I}, \quad (5.1.7)$$

where the coefficients c_i , $i = 0, 1, \dots, m-1$, can be determined by a Newton's recurrence relation (see [205]). For example, consider the Roe linearization (2.2.2)-(2.2.3) for the one-layer shallow water system (2.2.1). The characteristic polynomial in the x -direction (for the y -direction the procedure is completely analogous) is given by

$$p(\lambda) = \lambda^3 - 3\hat{v}_x \lambda^2 - (-3\hat{v}_x^2 + g\hat{h})\lambda + \hat{v}_x(\hat{v}_x^2 - g\hat{h}\hat{v}_x).$$

Solving $p(\mathbf{A}_x) = 0$, we find that

$$\mathbf{A}_x^3 = 3\hat{v}_x \mathbf{A}_x^2 + (-3\hat{v}_x^2 + g\hat{h})\mathbf{A}_x + \hat{v}_x(\hat{v}_x^2 - g\hat{h}\hat{v}_x)\mathcal{I}$$

or, more generally, that

$$\mathbf{A}_x^m = 3\hat{v}_x \mathbf{A}_x^{m-1} + (-3\hat{v}_x^2 + g\hat{h})\mathbf{A}_x^{m-2} + \hat{v}_x(\hat{v}_x^2 - g\hat{h}\hat{v}_x)\mathbf{A}_x^{m-3}, \quad m \geq 3. \quad (5.1.8)$$

Furthermore, formula (5.1.8) ensures that we can express all the m th power matrices \mathbf{A}_x^m , for $m \geq 3$, as a linear combination of \mathbf{A}_x^2 , \mathbf{A}_x and \mathcal{I} , i.e., all the power matrices of degree m can be written as linear combination of the power matrices $\mathbf{A}_\zeta^{m-1}, \dots, \mathbf{A}_\zeta, \mathcal{I}$. In this way, all the m th power matrices can be expressed as a sum of $m(m-1)/2$ matrices.

Additionally, in some cases it is possible to calculate explicitly the power matrices of order less than m , further reducing the computational cost. This is the case, for instance, of the Roe matrix for the one-layer shallow water system.

For the two-layer shallow water system, a Roe linearization in the x -direction of the form (1.3.11) for the two-dimensional AVM scheme is given by

$$\mathbf{A}_x(\mathbf{U}_0, \mathbf{U}_1) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\hat{v}_{1x}^2 + \hat{c}_1^2 & 2\hat{v}_{1x} & 0 & \hat{c}_1^2 & 0 & 0 \\ -\hat{v}_{1x}\hat{v}_{1y} & \hat{v}_{1y} & \hat{v}_{1x} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \rho\hat{c}_2^2 & 0 & 0 & -\hat{v}_{2x}^2 + \hat{c}_2^2 & 2\hat{v}_{2x} & \\ 0 & 0 & 0 & -\hat{v}_{2x}\hat{v}_{2y} & \hat{v}_{2y} & \hat{v}_{2x} \end{pmatrix},$$

where

$$\hat{v}_{i\zeta} = \frac{\sqrt{h_{i0}}v_{i\zeta,0} + \sqrt{h_{i1}}v_{i\zeta,1}}{\sqrt{h_{i0}} + \sqrt{h_{i1}}}, \quad \hat{c}_i = \sqrt{gh_i}, \quad \hat{h}_i = \frac{h_{i0} + h_{i1}}{2}.$$

The characteristic polynomial for this matrix can be read as

$$p(\lambda) = -(\hat{v}_{1x} - \lambda)(\hat{v}_{2x} - \lambda)(-\lambda^4 + a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0),$$

where

$$a_3 = 2(\hat{v}_{1x} + \hat{v}_{2x}), \quad a_2 = \alpha_1 + \alpha_2 - 4\hat{v}_{1x}\hat{v}_{2x}, \quad a_1 = -2(\alpha_1\hat{v}_{2x} + \alpha_2\hat{v}_{1x}), \quad a_0 = -\alpha_1\alpha_2 + (g\hat{h}_1)(\rho g\hat{h}_2),$$

defining $\alpha_1 = -\hat{v}_{1x}^2 + gh_1$ and $\alpha_2 = -\hat{v}_{2x}^2 + gh_2$. Solving $p(\mathbf{A}_x) = 0$, we find that

$$\mathbf{A}_x^6 = -c_5\mathbf{A}_x^5 - c_4\mathbf{A}_x^4 - c_3\mathbf{A}_x^3 - c_2\mathbf{A}_x^2 - c_1\mathbf{A}_x - c_0\mathcal{I}$$

or, more generally, that

$$\mathbf{A}_x^m = -c_5\mathbf{A}_x^{m-1} - c_4\mathbf{A}_x^{m-2} - c_3\mathbf{A}_x^{m-3} - c_2\mathbf{A}_x^{m-4} - c_1\mathbf{A}_x^{m-5} - c_0\mathbf{A}_x^{m-6}, \quad m \geq 6, \quad (5.1.9)$$

where, using the short notation $\hat{v}_{1x} + \hat{v}_{2x} = \bar{v}_x$,

$$\begin{aligned} c_5 &= a_3 + \bar{v}_x, & c_4 &= \hat{v}_{1x}\hat{v}_{2x} + a_3\bar{v}_x - a_2, & c_3 &= -\hat{v}_{1x}\hat{v}_{2x}a_3 + \bar{v}_xa_2 - a_1, \\ c_2 &= -\hat{v}_{1x}\hat{v}_{2x}a_2 + \bar{v}_xa_1 - a_0, & c_1 &= -\hat{v}_{1x}\hat{v}_{2x}a_1 + \bar{v}_xa_0, & c_0 &= \det\mathbf{A}_x = -\hat{v}_{1x}\hat{v}_{2x}a_0. \end{aligned} \quad (5.1.10)$$

In the case of RVM schemes, one matrix inversion is required. The functional viscosity matrix is approximated by $f(\mathbf{A}) = p(\mathbf{A})q(\mathbf{A})^{-1} \approx |\mathbf{A}|$, for certain polynomial functions p and q . An interesting tool for square matrices is a blockwise matrix inversion ([109]) by using the analytic inversion formula

$$\begin{pmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{M}_{11}^{-1} + \mathbf{M}_{11}^{-1}\mathbf{M}_{12}\mathbf{S}^{-1}\mathbf{M}_{21}\mathbf{M}_{11}^{-1} & -\mathbf{M}_{11}^{-1}\mathbf{M}_{12}\mathbf{S}^{-1} \\ -\mathbf{S}^{-1}\mathbf{M}_{21}\mathbf{M}_{11}^{-1} & \mathbf{S}^{-1} \end{pmatrix}, \quad (5.1.11)$$

where \mathbf{M}_{ij} are square sub-block matrices. The only matrices requiring inversion are \mathbf{M}_{11} and the so-called *Schur complement* of \mathbf{M}_{11} , defined as $\mathbf{S} = \mathbf{M}_{22} - \mathbf{M}_{21}\mathbf{M}_{11}^{-1}\mathbf{M}_{12}$. This formula is particularly interesting when \mathbf{M}_{11} and \mathbf{S} are small matrices, as it can reduce the computational cost of the code.

5.2 Parallelism sources

In this section we describe the main steps to build the first- and second-order algorithms for the two-dimensional AVM numerical schemes explained in the previous section, as well their main sources of parallelism. We follow the guidelines of [7, 9, 10] in order to build our CUDA solver. We present here a small explanation about the features of a GPU CUDA code. For a complete explanation, we refer to the recent publication [143].

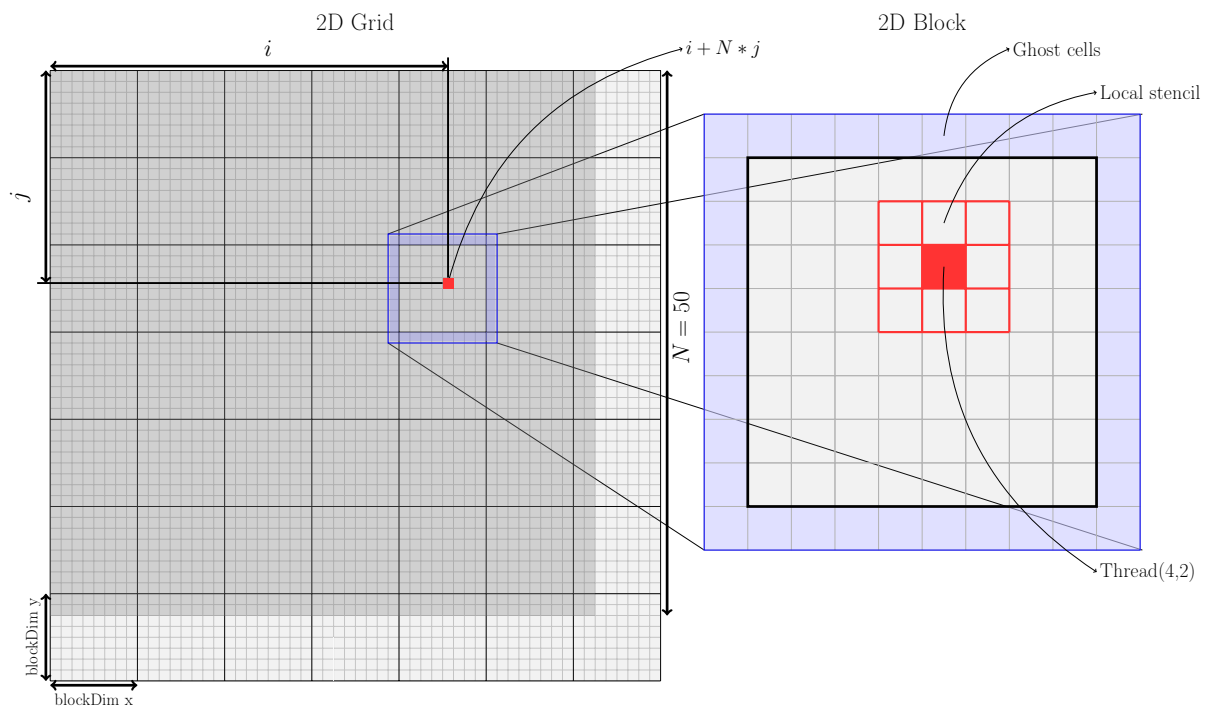


Figure 5.2: Sketch of an arrangement of a 2D Grid of 7×7 blocks of 8×8 threads to a 50×50 meshgrid. The starting thread should be a multiple of the warp size. Figure adapted from [143]

Each component of the finite volume algorithm can be implemented as a CUDA *kernel*, which is a function executed on the GPU ([156]). In the CUDA program, when a kernel is executed, this function is performed by several processing units at the same time, creating distinct running *threads*. Internally, each thread has its unique identifier in order to select which data it will process. CUDA groups threads into a grid of *blocks*. The blocks can be 1D, 2D, or 3D, meaning that the threads will have identifiers with 1, 2 or 3 indexes, respectively. This enables to identify the physical coordinates with each thread and block. Each thread runs in a single multiprocessor. The threads are staggered for execution in groups called *warps*. Each warp is a group of 32 threads that run physically in parallel on a multiprocessor ([143]). For this reason, it is always convenient that the number of threads defined by kernel be a multiple of 32. Figure 5.2 shows a possible arrangement of a 2D grid to a 50×50 physical meshgrid: each gridpoint is perfectly identified by a block identifier (`blockId`) and a thread identifier (`threadId`).

For example, the coordinate i can be defined as $i=\text{threadId.x}+\text{blockDim.x}*\text{blockId.x}$, and similarly $j=\text{threadId.y}+\text{blockDim.y}*\text{blockId.y}$.

For memory access there are various types of memory, with different sizes and access speeds. *Registers* are the fastest access memory and are uniquely allocated to each thread. Within a block, *shared memory* is shared by all the threads of a block. The *global memory* can be accessed by all the threads of a grid and also by the CPU. It is the slowest memory and it have greater storage capacity. For detailed information, we refer the reader to [143, 33].

The CUDA code typically performs the following steps: it reads the input data with the appropriate setting of initial and boundary conditions; GPU memory space is reserved; it transfers CPU memory input data to the global GPU memory; it runs the kernels (after calculating the initial time step, the process is repeated until the final time); it transfers the processed data from GPU memory to CPU memory; it displays the results and performs the required completion.

In Figure 5.3 we represent using a simple diagram the main calculation phases and mark with a circle the steps where we have a high degree of data parallelism. One accumulator array of $\text{NVAR}*\text{NV}$ `double` elements is used to store in the global memory the contributions given by the edges and vertex calculations, where NVAR is the number of conserved variables of the system and NV is the total number of volumes of the discretization. We can identify the accumulator as the term M_{ij}^n in (5.1.2). The main stages of the first- and second-order algorithms are summarized in what follows.

5.2.1 Vertex-based calculations

In the kernel design, there are some features to be taken into account. A GPU has a lot of processors and each one can execute multiple blocks concurrently. An important care to be taken when programming a parallel code is to avoid memory conflicts when many threads try to write to the same address. If many threads, whether from the same warp or different warps, access the same address, the final result may not reflect the true result. If each thread write different values over the same GPU memory location, it is possible that only one is read and so broadcasted to all threads ([143]). For vertex-based calculations, each control volume C_{ij} is accessed four times for four distinct vertex, which means that can exist memory conflicts at the time of adding the contributions of each cell around the vertex to the accumulator array in the global memory.

We consider three ways to avoid this problem. The first idea is to separate the vertex kernel in four non-intersecting kernels each one working on a subset of the vertex meshgrid, in a way such that the four states surrounding the vertex do not intersect the other four states surrounding other vertex. This subdivision can be seen schematically in Figure 5.4. A consequence of non-overlapping cells is that the vertices for a single kernel do not require synchronization when it comes to compute the contributions for a specific control volume.

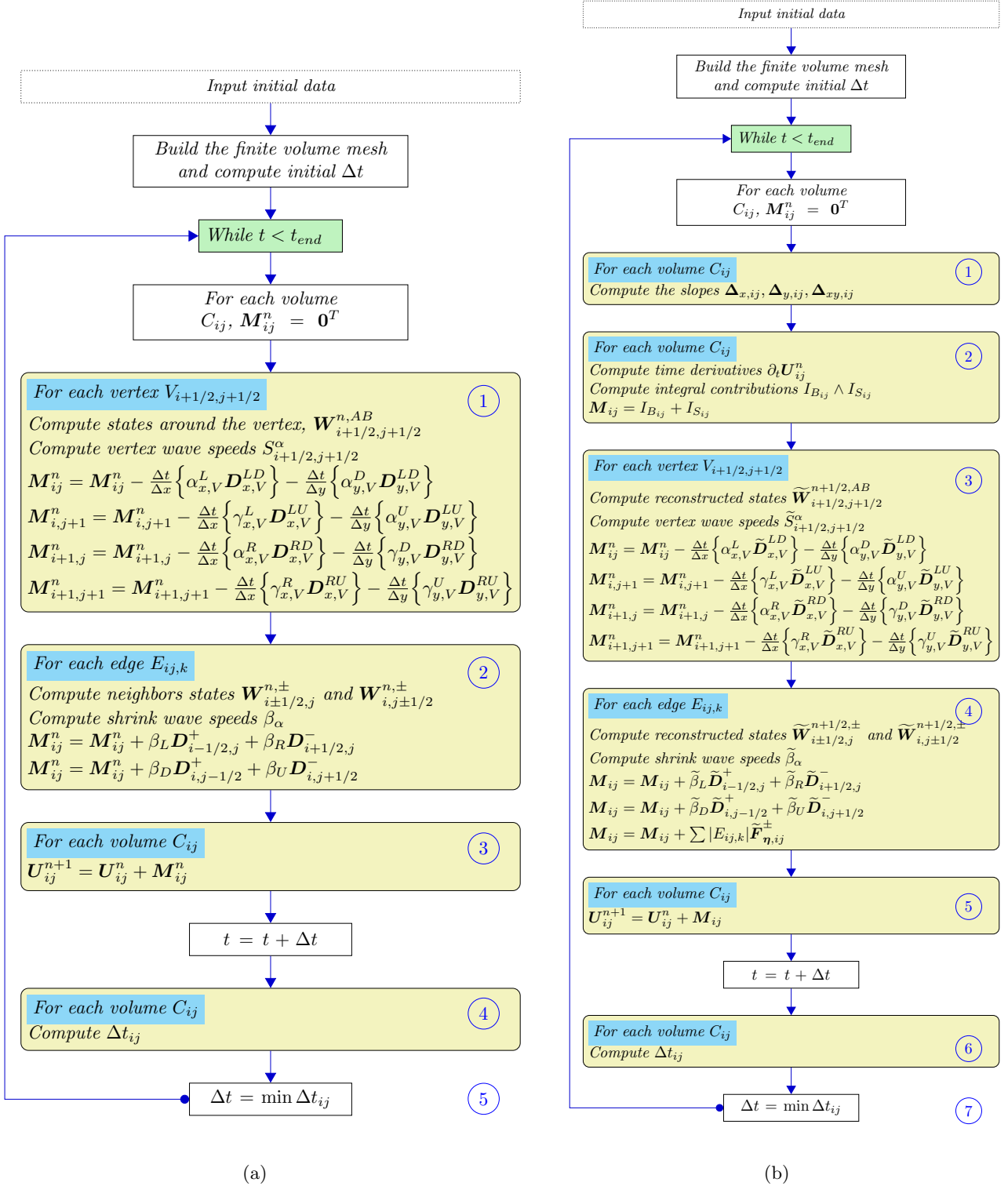


Figure 5.3: (a) First-order CUDA solver. (b) Second-order CUDA solver.

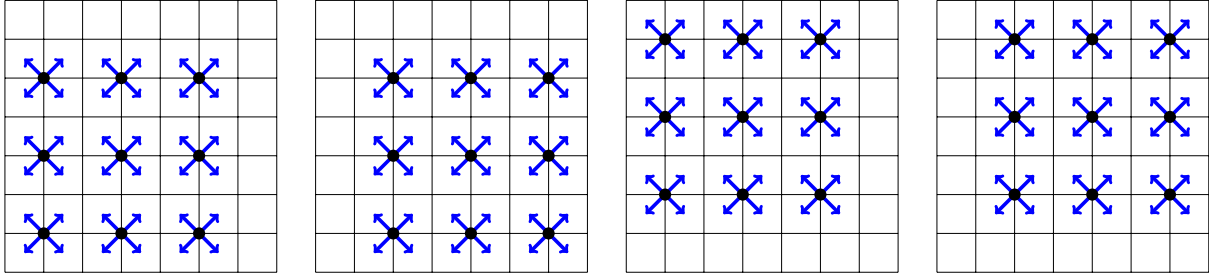


Figure 5.4: Different vertex processing steps.

For each vertex $V \equiv V_{i+1/2,j+1/2}$, we compute

$$\begin{aligned}
 \mathbf{M}_{ij}^n &= -\frac{\Delta t}{\Delta x} \left\{ \alpha_{x,V}^L \mathbf{D}_{x,V}^{LD} \right\} - \frac{\Delta t}{\Delta y} \left\{ \alpha_{y,V}^D \mathbf{D}_{y,V}^{LD} \right\}, \\
 \mathbf{M}_{i,j+1}^n &= -\frac{\Delta t}{\Delta x} \left\{ \gamma_{x,V}^L \mathbf{D}_{x,V}^{LU} \right\} - \frac{\Delta t}{\Delta y} \left\{ \alpha_{y,V}^U \mathbf{D}_{y,V}^{LU} \right\}, \\
 \mathbf{M}_{i+1,j}^n &= -\frac{\Delta t}{\Delta x} \left\{ \alpha_{x,V}^R \mathbf{D}_{x,V}^{RD} \right\} - \frac{\Delta t}{\Delta y} \left\{ \gamma_{y,V}^D \mathbf{D}_{y,V}^{RD} \right\}, \\
 \mathbf{M}_{i+1,j+1}^n &= -\frac{\Delta t}{\Delta x} \left\{ \gamma_{x,V}^R \mathbf{D}_{x,V}^{RU} \right\} - \frac{\Delta t}{\Delta y} \left\{ \gamma_{y,V}^U \mathbf{D}_{y,V}^{RU} \right\},
 \end{aligned} \tag{5.2.1}$$

which represent the contributions to the calculation of the new states around the vertex V , namely $\mathbf{U}_{LD} = \mathbf{U}_{ij}$, $\mathbf{U}_{RD} = \mathbf{U}_{i+1,j}$, $\mathbf{U}_{LU} = \mathbf{U}_{i,j+1}$ and $\mathbf{U}_{RU} = \mathbf{U}_{i+1,j+1}$. Each thread computes the contribution of every vertex of four adjacent states and adds up to the partial sums \mathbf{M}_{AB}^n associated with each state \mathbf{U}_{AB} around the vertex, being $AB = LD, LU, RD, RU$.

The second idea to avoid that different threads access and modify the same data is to use *atomic functions* of CUDA ([157]). Atomic operations ensure correct updates for shared variables in the global memory without conflicts. Thus, there is no interference of concurrent threads to read-modify-write operations, i.e., other thread cannot interfere in the current operation at the same address until the atomic operation is completed. For instance, an atomic add instruction reads a variable from the memory, modify that variable adding up some quantity to it, and write it back to the memory, all without interference of other threads. Although `AtomicAdd()` instructions in CUDA only supports data type of `int` or `float`, it is possible to overload atomic sums for `double` data type using `AtomicCAS()` (compare and swap) instructions. `AtomicAdd()` instructions only works for CUDA 6.0 or higher.

The third idea is not to write over the accumulator array in the vertex kernel. Instead, we store the four vertex contributions around a given vertex V in an array \mathbf{S} of size $4 \cdot \text{NVAR} \cdot \text{NV}$, and later in the volume kernel we add these contributions to the accumulator array. If the volume kernel have not memory conflicts, for a given cell C_{ij} , we identify the neighboring vertices around the cell, access the vertex contributions stored in the \mathbf{S} arrays and add the respective portion of each vertex to the control volume. Figure 5.5 shows a depiction of this procedure.

Thereby, in all the proposed ideas, a single accumulator is enough to store the contributions of each vertex and an appreciable reduction of memory access is achieved. In our experiments,

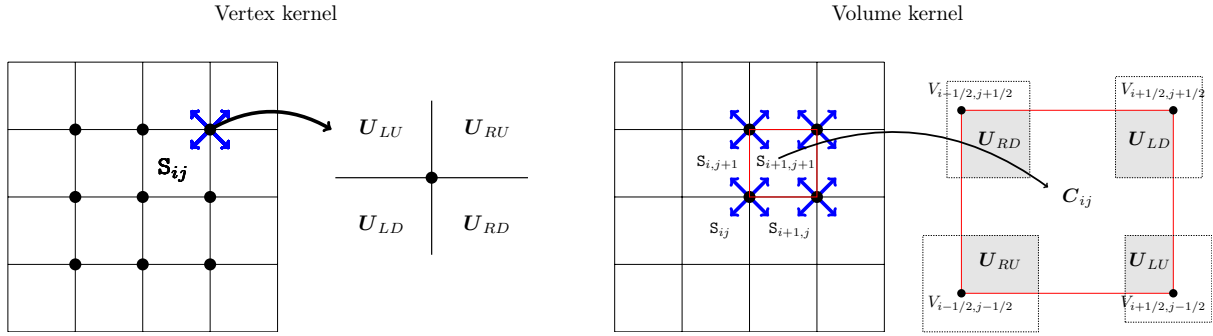


Figure 5.5: Storing in the vertex kernel and adding up in the volume kernel the vertex contributions to the accumulator in the control volume C_{ij} .

for small meshes the three techniques provided similar performances, while as the mesh is refined the vertex kernel using atomic functions presents a slight improvement in terms of computational cost. Further investigations on this topic are the subject of future research.

5.2.2 Edge-based calculations

Taking into account the notation given in Section 1.6, we denote as $E_{ij,k}$, with $k \in \{i \pm 1, j \pm 1\}$, the edge communicating two adjacent cells, C_{ij} and C_k . The edges processing follows the same approach as the one applied in [9, 10]. In order to avoid memory conflicts when adding the contributions to adjacent cells communicated through an edge $E_{ij,k}$ to the respective volumes, the edges are split in four non-intersecting sets: horizontal and vertical edges, with odd and even numbering. Within this approach, it is not possible that the same memory address can be accessed at the same time for different threads. For vertical edges $\eta_y = 0$, while for horizontal edges $\eta_x = 0$, where $\boldsymbol{\eta} = (\eta_x, \eta_y)$ is the normal unit vector at the edge $E_{ij,k}$ pointing towards the cell C_k . Thus, all the operations where this terms appears can be avoided using the four kernels, thus enhancing the efficiency. Also, as each edge $E_{ij,k}$ only needs the data of the neighboring control volumes C_{ij} and C_k , there is a higher spatial locality. Note that this works only for structured meshes. Further details can be found in [9, 10] and [7].

The contributions between the volumes C_{ij} and $C_{i+1,j}$ are computed by

$$\mathbf{M}_{ij}^n = -\frac{\Delta t}{\Delta x} \beta_{x,i+1/2}^L \mathbf{D}_{i+1/2,j}^-, \quad \mathbf{M}_{i+1,j}^n = -\frac{\Delta t}{\Delta x} \beta_{x,i+1/2}^R \mathbf{D}_{i+1/2,j}^+, \quad (5.2.2)$$

which represents the contributions to the calculation of the new states adjacent to the edge, namely \mathbf{U}_{ij} and $\mathbf{U}_{i+1,j}$. These contributions are added up to the respective partial sums \mathbf{M}_{ij} and $\mathbf{M}_{i+1,j}$ associated with the adjacents states.

Also, in high-order path-conservative schemes, an additional difference of fluxes appears from the integration of the Jacobian matrix of the system (recall the Section 1.6). We add these contributions to the respective accumulator array also in the edge kernels, taking advantage on the data access.

5.2.3 Volume-based calculations

Each thread represents a volume and it has not overlapping problems with respect to read-modify-write procedures in the calculations. The new state \mathbf{U}_{ij}^{n+1} is calculated over each volume C_{ij} from the n th state and the data stored on the accumulators in the previous steps (see formula (5.1.2)).

For second-order schemes, we have two additional steps: one kernel to compute the slopes for the second-order reconstruction operator (4.3.12); and another to calculate the predictor states at time $n + 1/2$ used in the MUSCL-Hancock procedure and the volume integrals corresponding to the regular part of the path-conservative schemes (see Section 4.3.4). The first one computes and stores the slopes in the coordinate directions, as well as the cross derivatives for each volume C_{ij} using a compact 9-stencil; also, limiters are applied to avoid spurious oscillations. In our experiments we adopt the generalized minmod limiter ([185]). Three `double` arrays of size `NVAR*NV` are used to store the slopes. The other kernel update \mathbf{W}_{ij}^n in a half-time, $\mathbf{W}_{ij}^{n+1/2}$, for each volume, by a Cauchy-Kowalewski procedure and stores it in a `double` array of size `NVAR*NV`. Furthermore, the regular integral contributions (1.3.9) and (1.3.10) are added to the accumulators of the respective volume. The full \mathbf{M}_{ij}^n is obtained adding all the edges, vertex and integrals contributions and updates the state \mathbf{U}_{ij}^n in the volume kernel.

Volume-based calculations are also used to compute the time step. Each thread represents a volume and computes the local Δt_{ij} corresponding to the volume V_{ij} . Later, the minimum is obtained by applying a reduction parallel algorithm of CUDA.

Table 5.1 shows the mean values of the percentages of the execution times for the main GPU kernels that are depicted in the diagram 5.3 for a second-order scheme using HLL, PVM-Chebyshev and RVM-Newman schemes, respectively. The PVM-Chebyshev method is based on a polynomial of eighth degree, while the RVM-Newman is based on a rational function of degree 8/8. We take into account the two-dimensional schemes based on the projected one-dimensional Riemann solvers, denoted as 1D×1D, and the genuinely two-dimensional AVM schemes considering the three strategies for the vertex kernel previously discussed in order to avoid memory conflicts: atomic operations, adding the vertex contributions in the volume kernel, and four non-overlapping vertex kernels. Clearly, for the genuinely 2D schemes, the vertex processing step consumes the major part of the execution time. As expected, PVM-Chebyshev and RVM-Newman schemes spent more time than the HLL scheme in the vertex processing, and particularly RVM-schemes spent much more time since at each step it is necessary to compute inverse matrices. As it was previously commented, in our experiments the vertex processing using atomic instructions provided a slight improvement with respect to the other two strategies. Table 5.1 was built using experiments for the two-layer shallow water equations, since for this system we have both source and coupling terms. When considering the one-layer shallow water system, which does not contain coupling terms, we have a reduction of about 5% in the vertex processing steps.

Table 5.1: Mean values of the percentages of the execution times for the main GPU kernels for the second-order HLL, PVM-Chebyshev and RVM-Newman schemes, respectively. For genuinely 2D schemes, the three strategies considered in Section 5.2.1 have been considered.

Computing step	% Execution time			
	HLL 1D×1D	HLL 2D with AtomicAdd() instructions	HLL 2D with adding vertex contr. in volume kernel	HLL 2D with four vertex kernels
Process vertex	-	44.50	37.12	47.92
Process vertical edges	20.36	17.66	17.26	16.03
Process horizontal edges	22.15	14.77	14.36	13.05
Compute predicted states and integrals contributions	26.20	7.68	9.33	8.41
Compute U_{ij}^{n+1}	13.32	5.75	11.69	5.38
Compute slopes	11.58	6.85	7.25	6.40
Compute Δt_{ij}	1.69	0.78	0.80	0.73
Get minimum Δt	0.26	0.07	0.08	0.07
Other minor instructions	4.44	1.94	2.11	2.01

Computing step	% Execution time			
	PVM- Chebyshev 1D×1D	PVM-Chebyshev 2D with AtomicAdd() instructions	PVM-Chebyshev 2D with adding vertex contr. in volume kernel	PVM-Chebyshev 2D with four vertex kernels
Process vertex	-	45.04	41.15	48.57
Process vertical edges	30.44	18.01	18.48	17.69
Process horizontal edges	30.50	15.69	17.24	16.32
Compute predicted states and integrals contributions	15.59	7.07	6.89	6.36
Compute U_{ij}^{n+1}	10.32	5.29	8.65	4.08
Compute slopes	7.70	6.32	5.33	4.86
Compute Δt_{ij}	1.98	0.72	0.59	0.55
Get minimum Δt	0.26	0.07	0.06	0.05
Other minor instructions	3.21	1.79	1.61	1.52

Computing step	% Execution time			
	RVM- Newman 1D×1D	RVM-Newman 2D with AtomicAdd() instructions	RVM-Newman 2D with adding vertex contr. in volume kernel	RVM-Newman 2D with four vertex kernels
Process vertex	-	52.04	52.03	53.08
Process vertical edges	33.36	17.12	16.44	17.13
Process horizontal edges	38.17	15.75	15.42	16.00
Compute predicted states and integrals contributions	12.07	5.01	4.79	5.02
Compute U_{ij}^{n+1}	6.85	3.75	6.03	3.24
Compute slopes	6.09	4.49	3.72	3.85
Compute Δt_{ij}	0.98	0.51	0.41	0.44
Get minimum Δt	0.07	0.05	0.04	0.04
Other minor instructions	2.41	1.28	1.12	1.20

5.3 CUDA implementation

In this section we put in order the main details of the parallel implementation in CUDA of the second-order 2D AVM schemes. We closely follow the general approach applied in [9, 10], with substantial modifications regarding vertex contributions and the MUSCL-Hancock procedure. The general steps of the parallel algorithm are depicted in Figure 5.3. When executing the parallel code, each step performed by a CUDA kernel is a sequential code with a large number of threads operating simultaneously, with its own set of instructions. Each kernel instruction will be shortly described in what follows. We have considered the vertex kernel approach by using atomic instructions of CUDA. The adaptation to the other strategies is straightforward.

1. *Build data structure and initialize Δt*

From the initial data read in CPU, the data structure that will be used in GPU calculations is built. To store the initial data, an array of `double` data type of $NVAR*NV$ elements is used, i.e., we store the initial state of physical variables \mathbf{U}_0 for each control volume. In fact, the meshgrid is enlarged in order to fill in with ghost cells, since boundary conditions are applied before each iteration. Later, the initial step is calculated and the main loop starts, until the final time is reached. At the beginning of each iteration, the fluctuations accumulator \mathbf{M}_{ij}^n is set to zero and the meshgrid is fulfilled with the boundary conditions. For shallow water systems, the bathymetry \mathbf{H}_{ij} is fixed for each volume in such a way that \mathbf{M}_{ij}^n can be taken as an array of $(NVAR-1)*NV$ elements.

2. *Reconstruction of states*

In this step, the slopes $\Delta_{x,ij}$, $\Delta_{y,ij}$ and the cross derivatives $\Delta_{xy,ij}$ are calculated and stored, separately, in arrays of $NVAR*NV$ `double` elements in the global memory. Slope limiters are used to control spurious oscillations. For shallow water systems, we reconstruct the free surface (one-layer case) or the interface (two-layer case), in order to ensure well-balancing (see Section 2.4). Next, each mesh of local derivatives is extended with ghost cells, according to the boundary conditions. This is done because the reconstructed states around a given vertex are used to compute the vertex contributions in the vertex kernel.

3. *Predictor step*

For each volume, spatial reconstructions in the coordinate directions are performed by using the previously computed slope arrays, since it is enough to apply a $1D \times 1D$ method in the predictor step (recall Section 4.3.4). Following the MUSCL-Hancock method, we compute the time derivatives $\partial_t \mathbf{W}_{ij}^n$ through a Cauchy-Kowalweski procedure and store $\Delta t \partial_t \mathbf{W}_{ij}^n$ in a `double` array of $NVAR*NV$ elements.

At the same time, the numerical approximation of the volume integrals of type (1.3.9) and (1.3.10) are performed in this step. The reconstructed values at time $\mathbf{W}_{ij}^{n+1/2}$ at the quadrature points are computed, and using a second-order quadrature rule we estimate the approximate volume integrals (see the expressions in (5.1.6)) and add them to the accumulator array \mathbf{M}_{ij}^n , for each volume.

In the same way as the spatial reconstructions, the meshgrid of time derivatives is extended with ghost cells in order to use that values in the vertex kernel.

3. *Vertex processing*

For each vertex $V_{i+1/2,j+1/2}$, reconstructions in space and time are performed for each cell around the vertex, namely $\tilde{U}_{ij}^{n+1/2}$, $\tilde{U}_{i+1,j}^{n+1/2}$, $\tilde{U}_{i,j+1}^{n+1/2}$ and $\tilde{U}_{i+1,j+1}^{n+1/2}$. For shallow water systems, also the topography is reconstructed, following the well-balancing approach given in Section 2.4. Secondly, the wave speeds arising in the strong interaction region, namely $\tilde{S}_{i+1/2,j+1/2}^{LD}$, $\tilde{S}_{i+1/2,j+1/2}^{RD}$, $\tilde{S}_{i+1/2,j+1/2}^{LU}$ and $\tilde{S}_{i+1/2,j+1/2}^{RU}$, are computed using the reconstructed states. Next, the star states related to the four one-dimensional Riemann problems arising in the coordinate directions are computed. Once the twelve states around the strong interaction region have been defined, we can calculate the four star states inside the strong interaction region, namely $U_V^{LD,*}$, $U_V^{RD,*}$, $U_V^{LU,*}$ and $U_V^{RU,*}$, following the iterative procedure given in Section 4.3.1, when nonconservative products are present. Now, we are able to calculate the eight fluctuations, four for each coordinate direction, given by equations 5.1.4, changing $D_{\zeta,V}^{AB}$ by $\tilde{D}_{\zeta,V}^{AB}$. These contributions are added to the accumulator array related to the respective control volume (see (5.2.1)), by using atomic instructions of CUDA.

Remark 11. *For the two-layer shallow water system (2.3.1), we can follow the same idea as in [64] for the shear shallow water system, and compute exactly the star states in the strong interaction region. In the nonconservative product B_ζ , $\zeta = x, y$, it only appears terms depending on the water heights h_1 and h_2 , which conform a conservative subsystem of (2.3.1). Therefore, we can compute explicitly $h_1^{AB,*}$ and $h_2^{AB,*}$ by using (4.2.8) and (4.2.9) without the coupling and source terms. Next, the nonconservative subsystem given by the moments equations can be determined by using again (4.2.8) and (4.2.9) with the coupling terms completely defined with the water height star states previously constructed. The topography H only changes through the zero-wave and it is directly computed by the reconstructed topography around the vertex. This avoids the iterative procedure and increases the efficiency of the code.*

4. *Edge processing*

The edge processing follows the same guidelines as in [9, 10]. The edges are divided into four non-intersecting kernels: vertical and horizontal edges, with even and odd numbering. For each kernel in the x -direction, the reconstructed states in space and time of two adjacent cells $U_{i\pm 1,j}^{n+1/2,-}$ and $U_{i\pm 1,j}^{n+1/2,+}$ are computed, as well as the topography $H_{i\pm 1,j}^-$ and $H_{i\pm 1,j}^+$ in shallow water systems, using the appropriate stencil values. The shrink parameters $\tilde{\beta}_L$ and $\tilde{\beta}_R$, that determine the proportion of the one-dimensional contributions with respect to the edge, are estimated using the vertex wave speeds computed beforehand in the vertex kernel. Following this, the contributions are independently computed and added to the respective accumulator array. The kernels in the y -direction follow the same steps.

5. *Compute the new state U_{ij}^{n+1}*

For each volume, the $(n+1)$ -th state U_{ij}^{n+1} is finally computed from the previous state U_{ij}^n and the accumulator array M_{ij}^n .

6. Compute the local Δt_{ij} for each volume

For each volume, the local Δt_{ij} is computed in the central zone by (4.3.11).

7. Get the minimum Δt

By a reduction algorithm of CUDA, the minimum of the local Δt_{ij} previously computed is obtained. This minimum is the next time step that will be used in the time loop of the algorithm.

Remark 12. To ensure positivity of the numerical schemes, first we consider the desingularization formula ([122])

$$v_{\zeta} = \frac{\sqrt{2}hq_{\zeta}}{\sqrt{h^4 + \max(h^4, \varepsilon)}}, \quad \zeta = x, y, \quad (5.3.1)$$

where usually $\varepsilon = \max\{\Delta x^4, \Delta y^4\}$ (see [173] for a different choice). Next, the discharge q_{ζ} is redefined by $q_{\zeta} := v_{\zeta}h$. Following [122], the reconstruction of the water height is modified to ensure the positivity of the reconstructed heights.

5.4 Experimental results

The CUDA code for the genuinely two-dimensional AVM schemes is tested in this section. A comparison in terms of efficiency is done against the 1D×1D counterpart schemes. The numerical computations have been performed at the Laboratory of Numerical Methods of the University of Málaga, which has an HPC cluster with 8 computing nodes, each server with dual processors Intel(R) Xeon(R) CPU 2.10 GHz, equipped with two supercomputing GPUs GeForce GTX Titan Black with 2880 processing kernels @889 MHz at each node, with KEPLER architecture from NVIDIA. In total, the power processement is about 870 GFLOPS ([87]). NVIDIA CUDA Compiler 10.2 has been used. All the computations have been realized using double precision, in order to avoid numerical instabilities.

We will focus in a numerical experiment concerning the two-layer shallow water system, where coupling and source terms appear. As test problem, we consider the evolution to a complex state introduced in Section 4.4.4.2, due to the fact that the differences between PVM and RVM schemes are more evident for complex problems ([49, 50]). In order to compare the 2D AVM schemes with their 1D×1D counterparts, jointly with the original problem we consider also a slight modification in the initial conditions, giving a higher frequency to the velocity fields, i.e., considering short wavelenghts in the propagation of the solution, in order to make faster the change in the gradients and thus to increase the difficulty of the test. Namely, the velocity fields will be given by

$$v_{x1} = -0.9A \sin(ky), \quad v_{y1} = 0.9B \sin(kx), \quad v_{x2} = -A \sin(ky), \quad v_{y2} = -A \sin(kx). \quad (5.4.1)$$

where for $k = 1$ we recover the original initial condition. The test was run for $k = 1$ and $k = 2$ until the final time $T = 0.5$. We use the best configuration for each scheme, with 95% of the maximal theoretical CFL number. The PVM-Chebyshev Riemann solver is based on a polynomial of eighth degree, while the RVM-Newman Riemann solver is defined using a rational function of 8/8 degree.

Figures 5.6 and 5.7 show the efficiency curves for $k = 1$ and $k = 2$, respectively. These curves display, in logarithmic scale, the GPU simulation times versus the L^1 -norm with respect to a reference solution computed with a very fine mesh, as we do not have the exact solution for this problem, for increasing refined meshes. As it can be seen, the 2D RVM solvers have a clear advantage over the 2D PVM schemes and indeed over the 2D HLL scheme, mainly in terms of the quality of the solution. This fact was also confirmed for the $1D \times 1D$ RVM schemes in [50], mainly for complex internal wave patterns.

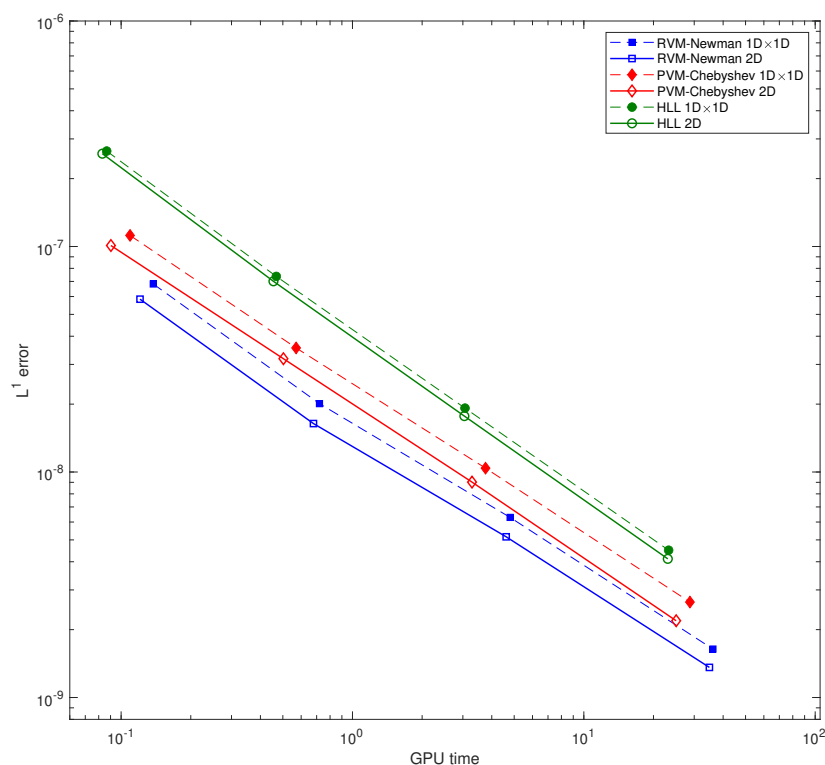


Figure 5.6: Efficiency curves GPU time vs. L^1 -error. Comparison of the 2D AVM schemes with respect to their $1D \times 1D$ counterparts. Case $k = 1$.

The genuinely two-dimensional codes for RVM-Newman, PVM-Chebyshev and HLL are more efficient with respect to the $1D \times 1D$ versions in our experiments. In Table 5.2 we compare the rate of the GPU computational time, where we measure how much faster or slower is the 2D CUDA code with respect to the $1D \times 1D$ CUDA code, and the rate of the L^1 -error with respect to a reference solution, where we measure how much more accurate is the 2D solution with respect to the $1D \times 1D$ solution. If

$$\text{rate} \frac{t^{2D}}{t^{1D \times 1D}} < \text{rate} \frac{e^{1D \times 1D}}{e^{2D}},$$

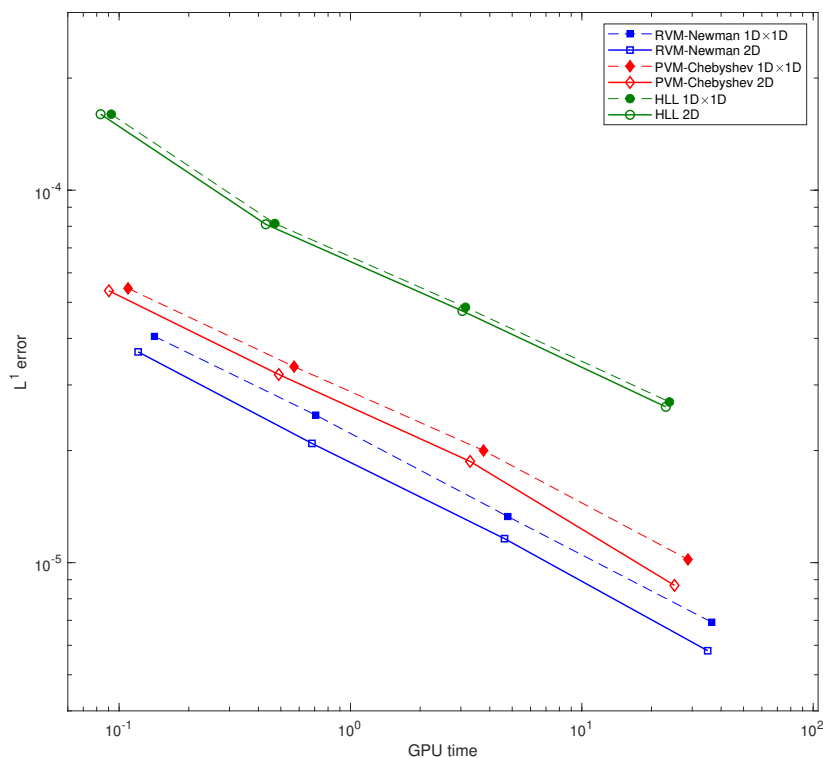


Figure 5.7: Efficiency curves GPU time vs. L^1 -error. Comparison of the 2D AVM schemes with respect to their 1D \times 1D counterparts. Case $k = 2$.

is accomplished, we could say that the 2D CUDA solver is more efficient than the 1D \times 1D counterpart. As it can be seen, 2D RVM-Newman, 2D PVM-Chebyshev and 2D HLL solvers provide faster and more accurate solutions than their 1D \times 1D versions as the mesh is refined. Although our CUDA kernels are of *memory-bound* type, where the kernels spend more time accessing memory than in calculations, the short writing (5.1.2)-(5.1.6) instead of (4.3.13) with fluctuations (4.3.3)-(4.3.6) is crucial for this behavior, mainly for the RVM-Newman 2D, due to the matrix inverse calculations. Even if the number of time steps is halved by the 2D code (the permissible CFL number is twice than for the 1D \times 1D code), the matrix inversions reduce the computational efficiency as the mesh is refined. For each vertex computation, using (5.1.2)-(5.1.6) we reduce from eight to four additional viscosity matrix computations with respect to the 1D \times 1D code, and as the mesh is refined this makes the difference since the time spent in this task accumulates. Table 5.1 shows that the time spent in the vertex processing is around 52% for the RVM-Newman code using atomic instructions in the vertex kernel or using the other strategies commented in Section 5.2.1. If we use (4.3.13) with (4.3.3)-(4.3.6) the time spent in this task would increase to 67% for the RVM-Newman code. Even with the positive results, we

would like to further deepen this topic in future investigations.

Table 5.2: Efficiency rates comparison between the genuinely 2D and the 1D×1D AVM CUDA codes for different meshes of sizes $N \times N$. The upper part of the table is related to the numerical experiment with $k = 1$ in (5.4.1), while the lower part is related to $k = 2$. Both tests were run until the final time $T = 0.5$, with a 95% of the maximum theoretical CFL number for each scheme.

N	HLL		PVM-Chebyshev		RVM-Newman	
	rate $\frac{t^{2D}}{t^{1D \times 1D}}$	rate $\frac{e^{1D \times 1D}}{e^{2D}}$	rate $\frac{t^{2D}}{t^{1D \times 1D}}$	rate $\frac{e^{1D \times 1D}}{e^{2D}}$	rate $\frac{t^{2D}}{t^{1D \times 1D}}$	rate $\frac{e^{1D \times 1D}}{e^{2D}}$
100	0.957712	1.02713	0.826552	1.10891	0.876436	1.16781
200	0.971985	1.05136	0.881478	1.11635	0.941926	1.22561
400	0.993731	1.08475	0.871478	1.15299	0.961445	1.21899
800	0.991841	1.09223	0.871884	1.21005	0.965497	1.20588
100	0.899876	1.00000	0.826332	1.01490	0.850749	1.10054
200	0.914167	1.00494	0.858403	1.05000	0.962792	1.19139
400	0.969587	1.02321	0.875040	1.06952	0.966767	1.14655
800	0.964692	1.03053	0.874789	1.17376	0.960590	1.19310

Chapter 6

Conclusions and future work

The primary aim of this research is the development of a general class of genuinely incomplete multidimensional Riemann solvers for general hyperbolic systems. The numerical framework is based on a general formulation which takes into account robustness, easy coding, computational efficiency and flexibility in order to be applied for general hyperbolic systems. We present a summary of the main results obtained in this thesis, as well as future research lines.

6.1 Conclusions

In the first part of this dissertation, we deal with the building of genuinely multidimensional incomplete Riemann solvers for conservation laws. Following Balsara's work, a four-waves model is considered which introduces a single resolved state in the so-called strong interaction region. The vertex region is bounded by the maximal wave speeds for each coordinate direction. Nevertheless, it is known that the HLL scheme introduces excessive numerical diffusion in the solution. Departing from a reinterpretation of Balsara's two-dimensional HLL scheme, we have extended it by substituting the underlying one-dimensional HLL solver by an arbitrary AVM-type numerical flux. This allows, in particular, to control the numerical diffusion of the scheme by using suitable approximations to the absolute value function as basis of the AVM flux. The numerical flux at an edge is built as a linear convex combination of a one-dimensional AVM flux and two multidimensional AVM corrections at the corners, which take into account transversal features of the flow. As Balsara's multidimensional HLL scheme, our solver includes approximate and closed-form expressions for the fluxes in the region of strong interaction. In principle, the two-dimensional AVM solvers could be extended to three-dimensional problems, although this point is not addressed in this thesis.

We have focused on applications to the MHD equations, in order to make comparisons with other existing methods in the literature and to show that this approach works well for complex systems. An accuracy analysis for multidimensional Euler and MHD problems shows that the proposed 2D AVM schemes have better accuracy than their projected one-dimensional counterparts. Additionally, we have proposed a new technique for imposing the divergence-free condition on the magnetic field for MHD simulations, based on the nonconservative writing of MHD equations. The nonconservative form allows to impose the divergence-free

condition automatically in the framework of path-conservative schemes. This technique has been incorporated in our multidimensional Riemann solvers, giving comparable results as the standard projection method. The performances of the proposed schemes have been illustrated by several numerical results obtained from test problems including genuinely two-dimensional effects, especially in the subsonic case in both directions with strong discontinuities.

Next, a possible extension of the multidimensional AVM-type solvers to general hyperbolic systems is obtained by considering a four-waves model in the vertices of the cell. We design a general strategy for defining two-dimensional Riemann solvers. In particular, we have introduced a simple version of the 2D HLL scheme and, departing from its rewriting as a 2D PVM scheme, we have established a new class of genuinely two-dimensional AVM-type Riemann solvers. We have also supplied an alternative splitting writing of the fluctuations at each face as a linear convex combination of one- and two-dimensional contributions, likewise to the conservative case. The second-order extension is constructed by using a predictor-corrector MUSCL-Hancock procedure, where the predictor-step is applied with the projected one-dimensional Riemann solvers on the edges of the cell, and in the corrector step we consider a reconstruction operator which takes into account cross derivatives. It has been shown through a modified equation analysis that our schemes are L^∞ -linearly stable for CFL number up to unity.

The proposed multidimensional AVM methods can be applied to arbitrary hyperbolic systems, both in conservative and nonconservative form. We focus on applications to one-layer and two-layers shallow water systems with bottom topography. The presence of bottom topography and nonconservative coupling terms represent an additional difficulty, which is solved by reformulating the problem within the path-conservative framework. A number of numerical experiments have been run in order to test the accuracy, well-balancedness and performances of the proposed schemes.

Finally, the 2D AVM schemes have been implemented on GPU architectures. GPU implementations using the CUDA programming toolkit were carried out, testing the efficiency of our genuinely 2D AVM schemes against the two-dimensional Riemann solvers based on projected one-dimensional Riemann solvers.

Different implementations of the two-dimensional AVM schemes on structured meshes have been performed. An unified rewriting of the first- and second-order 2D AVM schemes for both conservative and nonconservative has been proposed. Optimization strategies for the efficient parallelization of the schemes have been considered, especially for the kernel of vertex contributions. To avoid memory conflicts of read-modify-write operations, three strategies have been designed: splitting up the kernel in four non-intersecting kernels, using atomic instructions of CUDA, and adding the vertex contributions later in the volume kernel. Similar results have been found for coarse meshes, while a single kernel using atomic instructions presented better results for fine meshes.

Improving techniques to avoid matrix multiplications in PVM/RVM schemes have been also considered. Two-dimensional RVM-Newman, PVM-Chebyshev and HLL schemes have shown to be more efficient than their $1D \times 1D$ counterparts. Both proved to be more accurate, using double precision arithmetic, and faster with respect to GPU computational time in comparison with the projected $1D \times 1D$ counterparts. The proposed CUDA solvers allow to obtain a fast and accurate method, with a simple and flexible algorithm that can be adapted for arbitrary hyperbolic systems.

6.2 Future work

An extension of the first- and second-order 2D AVM-type schemes presented in this memory to very high-order schemes by means of the ADER procedure ([78]) or an approximate Taylor method ([36]) seem to be possible, using our solver as a building block. The presented 2D schemes can be applied to general hyperbolic systems, in such a way that applications to different hyperbolic systems, with inclusion of different source terms, can be performed. Moreover, inclusion of wet/dry situations in shallow water simulations should be included in our already well-balanced methods.

Our solvers have been designed for structured meshes within the framework of path-conservative schemes. A natural extension to unstructured meshes can be considered following [19, 52]. Also, in [58], shallow water equations in spherical coordinates have been efficiently solved by means of path-conservative schemes, and an adaptation to consider vertex contributions seems to be worthwhile.

Further improvements regarding GPU implementations will be carried out. Particularly, since our code is memory-bound, enhancements to memory access in the kernel of the vertices will be analyzed. High degree of spatial locality, reorganization of the data into self-contained patterns and reduction of memory latencies are issues that we will be investigated. These features will be taken into account in the designing of new multidimensional solvers.

Appendix A

PVM-Chebyshev and RVM-Newman coefficients

The PVM scheme based on Chebyshev polynomials and the RVM scheme based on Newman rational functions have been applied in the experiments presented in Chapters 4 and 5. To help in the implementation, the explicit form of the functions $\tau_{2p}^\varepsilon(x)$ and $R_r^\varepsilon(x)$ are detailed in this appendix.

Table A.1: Coefficients of the polynomial function $\tau_{2p}^\varepsilon(x)$, for $p = 2, 3, 4, 5$.

$\tau_4^\varepsilon(x)$	$\tau_6^\varepsilon(x)$
$\varepsilon = 0.046247442822796479754767575100032$	$\varepsilon = 0.032374415885002219312182891997173$
$a_0 = 0.12732395447351627648657768077101$	$a_0 = 0.090945681766797340347555486265004$
$a_2 = 1.5278874536821953178389321692521$	$a_2 = 2.1826963624031361683413316703601$
$a_4 = -0.6790610905254201412617476307787$	$a_4 = -2.4252181804479290759348129670668$
	$a_6 = 1.1641047266150059564487102241921$
$\tau_8^\varepsilon(x)$	$\tau_{10}^\varepsilon(x)$
$\varepsilon = 0.024964785249174241687787479115755$	$\varepsilon = 0.020335772797933675236913796189363$
$a_0 = 0.070735530263064598048098711539448$	$a_0 = 0.057874524760689216584808036714094$
$a_2 = 2.8294212105225839219239484615779$	$a_2 = 3.4724714856413529950884822028456$
$a_4 = -5.6588424210451678438478969231558$	$a_4 = -10.803244621995320429164166853298$
$a_6 = 6.3379035115705879851096445539345$	$a_6 = 20.742229674231015223995200358331$
$a_8 = -2.5868993924777910143304671648712$	$a_8 = -19.048986435518279287342530941325$
	$a_{10} = 6.5848348172161953092048255105813$

As commented in [49, Appendix], the different choices of the set of interpolation nodes for Newman rational functions do not entail noticeable differences. Therefore, the original definition

of the nodes has been considered here ([153]). For the sake of completeness, in Table A.1 are given the coefficients of the function $\tau_{2p}^\varepsilon(x)$ defined in Section 1.5 for $p = 2, 3, 4, 5$, while in Table A.2 are shown the coefficients of the rational function $R_r^\varepsilon(x)$ defined in Section 1.5 for $r = 4, 8, 12$.

Table A.2: Coefficients of the rational functions $R_r^\varepsilon(x)$, for $r = 4, 8, 12$.

$R_4^\varepsilon(x)$	
$\varepsilon = 0.027822696713222914606955742417911$	
$a_2 = 0.49033751037680525897615105585372$	$b_0 = 0.049787068367863942979342415650062$
$a_4 = 2.1975402610325055741326037759167$	$b_2 = 1.6380907030414468901294124161203$
$R_8^\varepsilon(x)$	
$\varepsilon = 0.0073705383650891072632840183732724$	
$a_2 = 0.0018842014579903037656603840393621$	$b_0 = 0.000050200029851686123319301219847672$
$a_4 = 0.22393339969828904285363062126558$	$b_2 = 0.028396779593646475165309199312845$
$a_6 = 2.6603751323278896683651454746098$	$b_4 = 1.0092054053131152259629651657934$
$a_8 = 3.1593617359609195449592217044025$	$b_6 = 4.0079020845084751726920645179912$
$R_{12}^\varepsilon(x)$	
$\varepsilon = 0.0029990160769122371705919882347516$	
$a_2 = 0.00000049159393999766806046665951661945$	$b_0 = 0.0000000053159245815805308986785370108644$
$a_4 = 0.00042183197118407599197574301796079$	$b_2 = 0.000019261453699132621490963079004224$
$a_6 = 0.051057717051541852586659107695069$	$b_4 = 0.00574011953476405690681722811493$
$a_8 = 1.2221699193416871403891138452786$	$b_6 = 0.30414447621500876915267669568926$
$a_{10} = 5.7856224139413543609980179910168$	$b_8 = 3.2889807590223027020888602118755$
$a_{12} = 3.8632917934540477728566520487052$	$b_{10} = 6.323679545812055958140103204936$

Appendix B

Full algorithm for the 2D HLL Riemann solver

In the generalized 2D consistency condition (4.1.4), the intermediate states have been constructed for the subsonic case in both directions. In order to consider also the supersonic cases, the integration limits in the faces Γ_i change (see Figure 4.1), as seen in Section 4.2.1. In particular, these changes play an important role in the computation of the term \mathcal{C} in (4.2.6).

To accomodate all the cases in a compact expression and facilitate the computation of face contributions, we can write the contributions of the faces Γ_1 and Γ_3 for the 2D HLL Riemann solver as

$$\begin{aligned} & \frac{S_L^-}{2} \int_0^1 \mathcal{A}_y(\Phi_{LD,LU}) \frac{d}{ds} \Phi_{LD,LU} ds - \frac{|S_L| + S_R^-}{2} \int_0^1 \mathcal{A}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U} ds \\ & + \frac{S_L^+ - |S_R|}{2} \int_0^1 \mathcal{A}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U} ds - \frac{S_R^+}{2} \int_0^1 \mathcal{A}_y(\Phi_{RD,RU}) \frac{d}{ds} \Phi_{RD,RU} ds. \end{aligned}$$

For the contributions of the Γ_2 and Γ_4 faces the idea is the same. To distinguish between subsonic and supersonic cases, we use a bar over the star state when we are in some supersonic case. Thus, the solutions of the 2D Riemann problem are given by nine cases:

- If $S_L < 0 < S_R$ and $S_D < 0 < S_U$,

$$\mathbf{R}(\xi, \psi) = \begin{cases} \mathbf{U}_{LD}^* & \text{in } [S_L, 0] \times [S_D, 0], \\ \mathbf{U}_{LU}^* & \text{in } [S_L, 0] \times [0, S_U], \\ \mathbf{U}_{RD}^* & \text{in } [0, S_R] \times [S_D, 0], \\ \mathbf{U}_{RU}^* & \text{in } [0, S_R] \times [0, S_U]. \end{cases} \quad (\text{B.0.1})$$

- If $S_L < 0 < S_R$ and $S_D > 0$,

$$\mathbf{R}(\xi, \psi) = \begin{cases} \mathbf{U}_{LU}^* & \text{in } [S_L, 0] \times [S_D, S_U], \\ \mathbf{U}_{RU}^* & \text{in } [0, S_R] \times [S_D, S_U], \\ \bar{\mathbf{U}}_{LD}^* & \text{in } [S_L, 0] \times [0, S_D], \\ \bar{\mathbf{U}}_{RD}^* & \text{in } [0, S_R] \times [0, S_D]. \end{cases} \quad (\text{B.0.2})$$

- If $S_L < 0 < S_R$ and $S_U < 0$,

$$\mathbf{R}(\xi, \psi) = \begin{cases} \mathbf{U}_{LD}^* & \text{in } [S_L, 0] \times [S_D, S_U], \\ \mathbf{U}_{RD}^* & \text{in } [0, S_R] \times [S_D, S_U], \\ \overline{\mathbf{U}}_{LU}^* & \text{in } [S_L, 0] \times [S_U, 0], \\ \overline{\mathbf{U}}_{RU}^* & \text{in } [0, S_R] \times [S_U, 0]. \end{cases} \quad (\text{B.0.3})$$

- If $S_L > 0$ and $S_D < 0 < S_U$,

$$\mathbf{R}(\xi, \psi) = \begin{cases} \mathbf{U}_{RU}^* & \text{in } [S_L, S_R] \times [0, S_U], \\ \mathbf{U}_{RD}^* & \text{in } [S_L, S_R] \times [S_D, 0], \\ \overline{\mathbf{U}}_{LU}^* & \text{in } [0, S_L] \times [0, S_U], \\ \overline{\mathbf{U}}_{LD}^* & \text{in } [0, S_L] \times [S_D, 0]. \end{cases} \quad (\text{B.0.4})$$

- If $S_L > 0$ and $S_D > 0$,

$$\mathbf{R}(\xi, \psi) = \begin{cases} \mathbf{U}_{RU}^* & \text{in } [S_L, S_R] \times [S_D, S_U], \\ \overline{\mathbf{U}}_{LU}^* & \text{in } [0, S_L] \times [S_D, S_U], \\ \overline{\mathbf{U}}_{RD}^* & \text{in } [S_L, S_R] \times [0, S_D], \\ \overline{\mathbf{U}}_{LD}^* & \text{in } [0, S_L] \times [0, S_D]. \end{cases} \quad (\text{B.0.5})$$

- If $S_L > 0$ and $S_U < 0$,

$$\mathbf{R}(\xi, \psi) = \begin{cases} \mathbf{U}_{RD}^* & \text{in } [S_L, S_R] \times [S_D, S_U], \\ \overline{\mathbf{U}}_{LD}^* & \text{in } [0, S_L] \times [S_D, S_U], \\ \overline{\mathbf{U}}_{RU}^* & \text{in } [S_L, S_R] \times [S_U, 0], \\ \overline{\mathbf{U}}_{LU}^* & \text{in } [0, S_L] \times [S_U, 0]. \end{cases} \quad (\text{B.0.6})$$

- If $S_R < 0$ and $S_D < 0 < S_U$,

$$\mathbf{R}(\xi, \psi) = \begin{cases} \mathbf{U}_{LU}^* & \text{in } [S_L, S_R] \times [0, S_U], \\ \mathbf{U}_{LD}^* & \text{in } [S_L, S_R] \times [S_D, 0], \\ \overline{\mathbf{U}}_{RU}^* & \text{in } [S_R, 0] \times [0, S_U], \\ \overline{\mathbf{U}}_{RD}^* & \text{in } [S_R, 0] \times [S_D, 0]. \end{cases} \quad (\text{B.0.7})$$

- If $S_R < 0$ and $S_D > 0$,

$$\mathbf{R}(\xi, \psi) = \begin{cases} \mathbf{U}_{LU}^* & \text{in } [S_L, S_R] \times [S_D, S_U], \\ \overline{\mathbf{U}}_{RU}^* & \text{in } [S_R, 0] \times [S_D, S_U], \\ \overline{\mathbf{U}}_{LD}^* & \text{in } [S_L, S_R] \times [0, S_D], \\ \overline{\mathbf{U}}_{RD}^* & \text{in } [S_R, 0] \times [0, S_D]. \end{cases} \quad (\text{B.0.8})$$

- If $S_R < 0$ and $S_U < 0$,

$$\mathbf{R}(\xi, \psi) = \begin{cases} \mathbf{U}_{LD}^* & \text{in } [S_L, S_R] \times [S_D, S_U], \\ \overline{\mathbf{U}}_{RD}^* & \text{in } [S_R, 0] \times [S_D, S_U], \\ \overline{\mathbf{U}}_{LU}^* & \text{in } [S_L, S_R] \times [S_U, 0], \\ \overline{\mathbf{U}}_{RU}^* & \text{in } [S_R, 0] \times [S_U, 0]. \end{cases} \quad (\text{B.0.9})$$

From the contributions for the faces Γ_i we can recalculate the term \mathcal{C} in (4.2.6) for each case. Knowing the solution of the 2D Riemann problem in all the cases (B.0.1)-(B.0.9), we can compute the consistency condition for each case, and finally compute the corresponding intermediate states. The subsonic case in both directions has already been seen in Section 4.2. The case where the flow is subsonic in the x -direction and $S_D > 0$ in the y -direction was analyzed in Section 4.2.1. It is important to recall that the HLL intermediate one-dimensional states also change when we are in some supersonic case [149]. The subsonic case in the x -direction was recalled in (4.2.2)-(4.2.3). For the supersonic cases, if $0 < S_L < S_R$ the state $\mathbf{U}_{\mu 1}$ changes to

$$\mathbf{U}_{\mu 1} = \mathbf{U}_{L\mu} + \mathbf{A}_x^{-1} \mathbf{S}_x(\sigma_{R\mu} - \sigma_{L\mu}),$$

while $\mathbf{U}_{\mu 2}$ is given by (4.2.3). On the other hand, if $S_L < S_R < 0$ the state $\mathbf{U}_{\mu 2}$ changes to

$$\mathbf{U}_{\mu 2} = \mathbf{U}_{R\mu} - \mathbf{A}_x^{-1} \mathbf{S}_x(\sigma_{R\mu} - \sigma_{L\mu}),$$

while $\mathbf{U}_{\mu 1}$ is defined by (4.2.2). Following the procedure described in Section 4.2.1, we summarize the remaining cases:

- Case $S_U < 0$, $S_L < 0 < S_R$. Integrating $\mathcal{R}(\xi, \psi) - \mathbf{W}_0(\xi, \psi)$ on $[0, S_L] \times [S_D, S_U]$ we get

$$\begin{aligned} -S_L(S_U - S_D)\mathbf{U}_{LD}^* + S_R(S_U - S_D)\mathbf{U}_{RD}^* + S_L S_U \overline{\mathbf{U}}_{LU}^* \\ - S_R S_U \overline{\mathbf{U}}_{RU}^* - S_L S_D \mathbf{U}_{LD} + S_R S_D \mathbf{U}_{RD} = \mathcal{C}. \end{aligned}$$

By the Rankine-Hugoniot conditions (1.3.13) (see Figure B.1),

$$\begin{cases} \mathbf{U}_{RD}^* - \mathbf{U}_{LD}^* = \mathbf{A}_x(\mathbf{U}_{LD}^*, \mathbf{U}_{RD}^*)^{-1} \mathbf{S}_x(\mathbf{U}_{LD}^*, \mathbf{U}_{RD}^*)(\sigma_{RD} - \sigma_{LD}) & \equiv \alpha \\ \overline{\mathbf{U}}_{RU}^* - \overline{\mathbf{U}}_{LU}^* = \mathbf{A}_x(\overline{\mathbf{U}}_{LU}^*, \overline{\mathbf{U}}_{RU}^*)^{-1} \mathbf{S}_x(\overline{\mathbf{U}}_{LU}^*, \overline{\mathbf{U}}_{RU}^*)(\sigma_{RD} - \sigma_{LD}) & \equiv \beta \\ \mathbf{U}_{1U} - \overline{\mathbf{U}}_{LU}^* = \mathbf{A}_y(\overline{\mathbf{U}}_{LU}^*, \mathbf{U}_{1U})^{-1} \mathbf{S}_y(\overline{\mathbf{U}}_{LU}^*, \mathbf{U}_{1U})(\sigma_{LU} - \sigma_{LD}) & \equiv \gamma \\ \mathbf{U}_{2U} - \overline{\mathbf{U}}_{RU}^* = \mathbf{A}_y(\overline{\mathbf{U}}_{RU}^*, \mathbf{U}_{2U})^{-1} \mathbf{S}_y(\overline{\mathbf{U}}_{RU}^*, \mathbf{U}_{2U})(\sigma_{RU} - \sigma_{RD}) & \equiv \delta \end{cases} \quad (\text{B.0.10})$$

Therefore,

$$\begin{cases} \overline{\mathbf{U}}_{LU}^* = \mathbf{U}_{1U} - \alpha, \\ \overline{\mathbf{U}}_{RU}^* = \mathbf{U}_{2U} - \beta, \\ \mathbf{U}_{LD}^* = \frac{-S_L S_U \overline{\mathbf{U}}_{LU}^* + S_R S_U \overline{\mathbf{U}}_{RU}^* + S_L S_D \mathbf{U}_{LD} - S_R S_D \mathbf{U}_{RD} + \mathcal{C}}{(S_R - S_L)(S_U - S_D)} - \frac{S_R}{S_R - S_L} \gamma, \\ \mathbf{U}_{RD}^* = \mathbf{U}_{LD}^* + \gamma. \end{cases} \quad (\text{B.0.11})$$

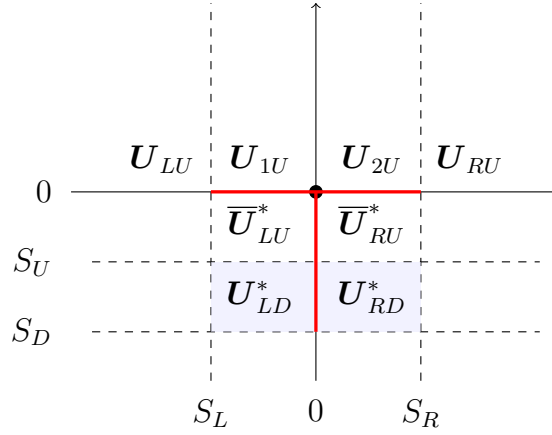


Figure B.1: Sketch of the jumps through the zero wave for the case $S_U < 0, S_L < 0 < S_R$.

- Case $S_L > 0, S_D < 0 < S_U$. Integrating $\mathcal{R}(\xi, \psi) - \mathbf{W}_0(\xi, \psi)$ on $[S_L, S_R] \times [0, S_D]$ we get

$$\begin{aligned}
 & -(S_R - S_L)S_D \mathbf{U}_{RD}^* + (S_R - S_L)S_U \mathbf{U}_{RU}^* - S_L S_D \bar{\mathbf{U}}_{LD}^* \\
 & + S_L S_U \bar{\mathbf{U}}_{LU}^* + S_R S_D \mathbf{U}_{RD} - S_R S_U \mathbf{U}_{RU} = \mathcal{C}.
 \end{aligned} \tag{B.0.12}$$

By the Rankine-Hugoniot conditions (1.3.13) (see Figure B.2),

$$\begin{cases}
 \mathbf{U}_{RU}^* - \mathbf{U}_{RD}^* = \mathbf{A}_y(\mathbf{U}_{RD}^*, \mathbf{U}_{RU}^*)^{-1} \mathbf{S}_y(\mathbf{U}_{RD}^*, \mathbf{U}_{RU}^*)(\sigma_{RU} - \sigma_{RD}) & \equiv \alpha \\
 \bar{\mathbf{U}}_{LU}^* - \bar{\mathbf{U}}_{LD}^* = \mathbf{A}_y(\bar{\mathbf{U}}_{LD}^*, \bar{\mathbf{U}}_{LU}^*)^{-1} \mathbf{S}_y(\bar{\mathbf{U}}_{LD}^*, \bar{\mathbf{U}}_{LU}^*)(\sigma_{RU} - \sigma_{RD}) & \equiv \beta \\
 \bar{\mathbf{U}}_{LU}^* - \mathbf{U}_{L2} = \mathbf{A}_x(\mathbf{U}_{L2}, \bar{\mathbf{U}}_{LU}^*)^{-1} \mathbf{S}_x(\mathbf{U}_{L2}, \bar{\mathbf{U}}_{LU}^*)(\sigma_{RU} - \sigma_{LU}) & \equiv \gamma \\
 \bar{\mathbf{U}}_{LD}^* - \mathbf{U}_{L1} = \mathbf{A}_x(\mathbf{U}_{L1}, \bar{\mathbf{U}}_{LD}^*)^{-1} \mathbf{S}_x(\mathbf{U}_{L1}, \bar{\mathbf{U}}_{LD}^*)(\sigma_{RD} - \sigma_{LD}) & \equiv \delta
 \end{cases} \tag{B.0.13}$$

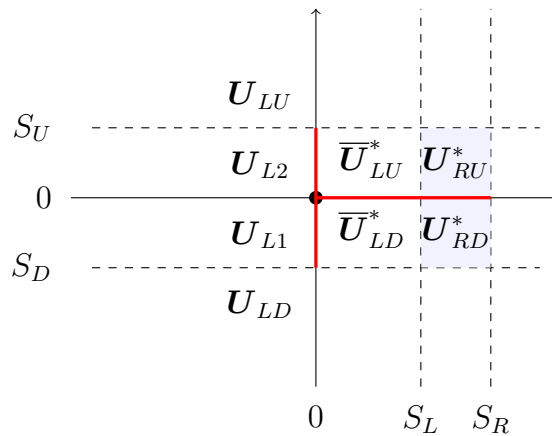


Figure B.2: Sketch of the jumps through the zero wave for the case $S_L > 0, S_D < 0 < S_U$.

Therefore,

$$\begin{cases} \bar{U}_{LU}^* = U_{L2} + \delta, \\ \bar{U}_{LD}^* = U_{L1} + \gamma, \\ U_{RD}^* = \frac{S_L S_D \bar{U}_{LD}^* - S_L S_U \bar{U}_{LU}^* - S_R S_D U_{RD} + S_R S_U U_{RU} + \mathcal{C}}{(S_R - S_L)(S_U - S_D)} - \frac{S_U}{S_U - S_D} \beta, \\ U_{RU}^* = U_{RD} + \beta. \end{cases} \quad (\text{B.0.14})$$

- Case $S_L > 0, S_D > 0$. Integrating $\mathcal{R}(\xi, \psi) - \mathbf{W}_0(\xi, \psi)$ on $[0, S_L] \times [0, S_D]$ we get

$$\begin{aligned} S_L S_D \bar{U}_{LD}^* + S_L (S_U - S_D) \bar{U}_{LU}^* + (S_R - S_L) S_D \bar{U}_{RD}^* \\ + (S_R - S_L)(S_U - S_D) U_{RU}^* - S_R S_U U_{RU} = \mathcal{C}. \end{aligned} \quad (\text{B.0.15})$$

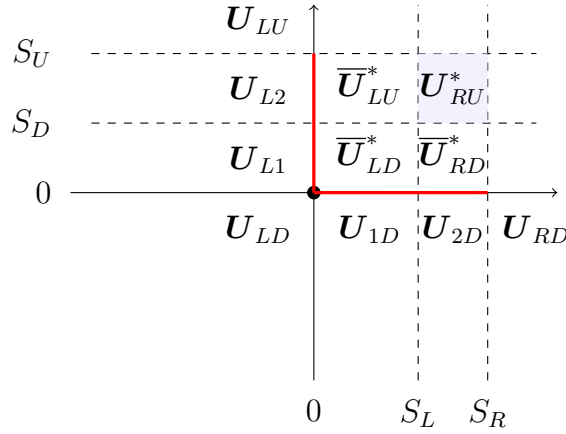


Figure B.3: Sketch of the jumps through the zero wave for the case $S_L > 0, S_D > 0$.

By the Rankine-Hugoniot conditions (1.3.13) (see Figure B.3),

$$\begin{cases} \bar{U}_{LU}^* - U_{L2} = \mathbf{A}_x(U_{L2}, \bar{U}_{LU}^*)^{-1} \mathbf{S}_x(U_{L2}, \bar{U}_{LU}^*)(\sigma_{RU} - \sigma_{LU}) & \equiv \alpha \\ \bar{U}_{LD}^* - U_{L1} = \mathbf{A}_x(U_{L1}, \bar{U}_{LD}^*)^{-1} \mathbf{S}_x(U_{L1}, \bar{U}_{LD}^*)(\sigma_{RU} - \sigma_{LU}) & \equiv \beta \\ \bar{U}_{LD}^* - U_{1D} = \mathbf{A}_y(U_{1D}, \bar{U}_{LD}^*)^{-1} \mathbf{S}_y(U_{1D}, \bar{U}_{LD}^*)(\sigma_{RU} - \sigma_{RD}) & \equiv \gamma \\ \bar{U}_{RD}^* - U_{2D} = \mathbf{A}_y(U_{2D}, \bar{U}_{RD}^*)^{-1} \mathbf{S}_y(U_{2D}, \bar{U}_{RD}^*)(\sigma_{RU} - \sigma_{RD}) & \equiv \delta \end{cases} \quad (\text{B.0.16})$$

Therefore,

$$\begin{cases} \bar{U}_{LU}^* = U_{L2} + \delta, \\ \bar{U}_{RD}^* = U_{2D} + \beta, \\ \bar{U}_{LD}^* = U_{L1} + \delta, \\ U_{RU}^* = \frac{S_R S_U U_{RU} - S_L (S_U - S_D) \bar{U}_{LU}^* - S_D (S_R - S_L) \bar{U}_{RD}^* - S_L S_D \bar{U}_{LD}^* + \mathcal{C}}{(S_R - S_L)(S_U - S_D)}. \end{cases} \quad (\text{B.0.17})$$

- Case $S_L > 0, S_U < 0$. Integrating $\mathcal{R}(\xi, \psi) - \mathbf{W}_0(\xi, \psi)$ on $[0, S_L] \times [S_D, S_U]$ we get

$$\begin{aligned} S_L(S_U - S_D)\bar{\mathbf{U}}_{LD}^* - S_L S_U \bar{\mathbf{U}}_{LU}^* - (S_R - S_L)S_U \bar{\mathbf{U}}_{RU}^* \\ + (S_R - S_L)(S_U - S_D)\mathbf{U}_{RD}^* + S_R S_D \mathbf{U}_{RD} = \mathcal{C}. \end{aligned} \quad (\text{B.0.18})$$

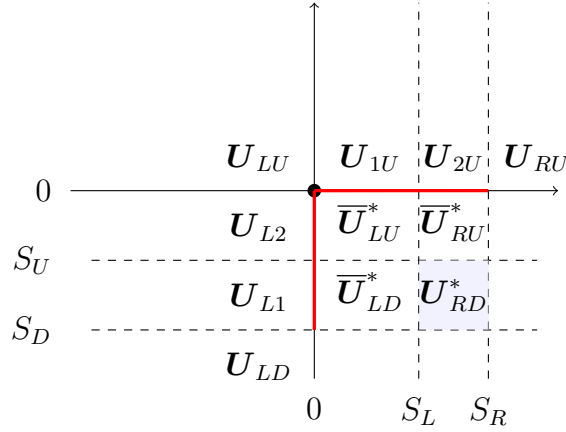


Figure B.4: Sketch of the jumps through the zero wave for the case $S_L > 0, S_U < 0$.

By the Rankine-Hugoniot conditions (1.3.13) (see Figure B.4),

$$\begin{cases} \bar{\mathbf{U}}_{LU}^* - \mathbf{U}_{L2} = \mathbf{A}_x(\mathbf{U}_{L2}, \bar{\mathbf{U}}_{LU}^*)^{-1} \mathbf{S}_x(\mathbf{U}_{L2}, \bar{\mathbf{U}}_{LU}^*)(\sigma_{RD} - \sigma_{LD}) & \equiv \alpha \\ \bar{\mathbf{U}}_{LD}^* - \mathbf{U}_{L1} = \mathbf{A}_x(\mathbf{U}_{L1}, \bar{\mathbf{U}}_{LD}^*)^{-1} \mathbf{S}_x(\mathbf{U}_{L1}, \bar{\mathbf{U}}_{LD}^*)(\sigma_{RD} - \sigma_{LD}) & \equiv \beta \\ \mathbf{U}_{2U} - \bar{\mathbf{U}}_{RU}^* = \mathbf{A}_y(\bar{\mathbf{U}}_{RU}^*, \mathbf{U}_{2U})^{-1} \mathbf{S}_y(\bar{\mathbf{U}}_{RU}^*, \mathbf{U}_{2U})(\sigma_{RU} - \sigma_{RD}) & \equiv \gamma \\ \mathbf{U}_{1U} - \bar{\mathbf{U}}_{LU}^* = \mathbf{A}_y(\bar{\mathbf{U}}_{LU}^*, \mathbf{U}_{1U})^{-1} \mathbf{S}_y(\bar{\mathbf{U}}_{LU}^*, \mathbf{U}_{1U})(\sigma_{RU} - \sigma_{RD}) & \equiv \delta \end{cases} \quad (\text{B.0.19})$$

Therefore,

$$\begin{cases} \bar{\mathbf{U}}_{RU}^* = \mathbf{U}_{2U} - \beta, \\ \bar{\mathbf{U}}_{LU}^* = \mathbf{U}_{L2} + \gamma, \\ \bar{\mathbf{U}}_{LD}^* = \mathbf{U}_{L1} + \gamma, \\ \mathbf{U}_{RD}^* = \frac{-S_R S_D \mathbf{U}_{RD} + S_L S_U \bar{\mathbf{U}}_{LU}^* + S_U (S_R - S_L) \bar{\mathbf{U}}_{RU}^* - S_L (S_U - S_D) \bar{\mathbf{U}}_{LD}^* + \mathcal{C}}{(S_R - S_L)(S_U - S_D)}. \end{cases} \quad (\text{B.0.20})$$

- Case $S_R < 0, S_D < 0 < S_U$. Integrating $\mathcal{R}(\xi, \psi) - \mathbf{W}_0(\xi, \psi)$ on $[0, S_L] \times [S_D, S_U]$ we get

$$\begin{aligned} -(S_R - S_L)S_D \mathbf{U}_{LD}^* + (S_R - S_L)S_U \mathbf{U}_{LU}^* + S_R S_D \bar{\mathbf{U}}_{RD}^* \\ - S_R S_U \bar{\mathbf{U}}_{RU}^* - S_L S_D \mathbf{U}_{LD} + S_L S_U \mathbf{U}_{LU} = \mathcal{C}. \end{aligned} \quad (\text{B.0.21})$$

By the Rankine-Hugoniot conditions (1.3.13) (see Figure B.5),

$$\begin{cases} \mathbf{U}_{LU}^* - \mathbf{U}_{LD}^* = \mathbf{A}_2(\mathbf{U}_{LD}^*, \mathbf{U}_{LU}^*)^{-1} \mathbf{S}_2(\mathbf{U}_{LD}^*, \mathbf{U}_{LU}^*)(\sigma_{LU} - \sigma_{LD}) & \equiv \alpha \\ \bar{\mathbf{U}}_{RU}^* - \bar{\mathbf{U}}_{RD}^* = \mathbf{A}_2(\bar{\mathbf{U}}_{RD}^*, \bar{\mathbf{U}}_{RU}^*)^{-1} \mathbf{S}_2(\bar{\mathbf{U}}_{RD}^*, \bar{\mathbf{U}}_{RU}^*)(\sigma_{LU} - \sigma_{LD}) & \equiv \beta \\ \mathbf{U}_{R2} - \bar{\mathbf{U}}_{RU}^* = \mathbf{A}_1(\bar{\mathbf{U}}_{RU}^*, \mathbf{U}_{R2})^{-1} \mathbf{S}_1(\bar{\mathbf{U}}_{RU}^*, \mathbf{U}_{R2})(\sigma_{RU} - \sigma_{LU}) & \equiv \gamma \\ \mathbf{U}_{R1} - \bar{\mathbf{U}}_{RD}^* = \mathbf{A}_1(\bar{\mathbf{U}}_{RD}^*, \mathbf{U}_{R1})^{-1} \mathbf{S}_1(\bar{\mathbf{U}}_{RD}^*, \mathbf{U}_{R1})(\sigma_{RD} - \sigma_{LD}) & \equiv \delta \end{cases} \quad (\text{B.0.22})$$

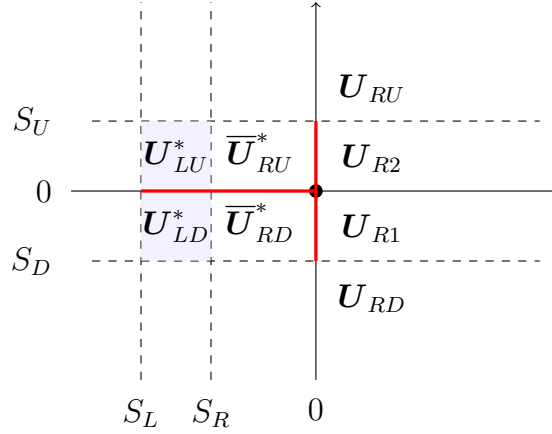


Figure B.5: Sketch of the jumps through the zero wave for the case $S_R < 0$, $S_D < 0 < S_U$.

Therefore,

$$\begin{cases} \bar{\mathbf{U}}_{RU}^* = \mathbf{U}_{R2} - \delta, \\ \bar{\mathbf{U}}_{RD}^* = \mathbf{U}_{R1} - \gamma, \\ \mathbf{U}_{LD}^* = \frac{S_R S_U \bar{\mathbf{U}}_{RU}^* - S_R S_D \bar{\mathbf{U}}_{RD}^* + S_L S_D \mathbf{U}_{LD} - S_L S_U \mathbf{U}_{LU} + \mathcal{C}}{(S_R - S_L)(S_U - S_D)} - \frac{S_U}{S_U - S_D} \alpha, \\ \mathbf{U}_{LU}^* = \mathbf{U}_{LD}^* + \alpha. \end{cases} \quad (\text{B.0.23})$$

- Case $S_R < 0$, $S_D > 0$. Integrating $\mathcal{R}(\xi, \psi) - \mathbf{W}_0(\xi, \psi)$ on $[S_L, S_R] \times [S_D, 0]$ we get

$$\begin{aligned} (S_R - S_L) S_D \bar{\mathbf{U}}_{LD}^* + (S_R - S_L)(S_U - S_D) \mathbf{U}_{LU}^* - S_R S_D \bar{\mathbf{U}}_{RD}^* \\ - S_R(S_U - S_D) \bar{\mathbf{U}}_{RU}^* + S_L S_U \mathbf{U}_{LU} = \mathcal{C}. \end{aligned} \quad (\text{B.0.24})$$

By the Rankine-Hugoniot conditions (1.3.13) (see Figure B.6),

$$\begin{cases} \mathbf{U}_{R2} - \bar{\mathbf{U}}_{RU}^* = \mathbf{A}_x(\bar{\mathbf{U}}_{RU}^*, \mathbf{U}_{R2})^{-1} \mathbf{S}_x(\bar{\mathbf{U}}_{RU}^*, \mathbf{U}_{R2})(\sigma_{RU} - \sigma_{LU}) & \equiv \alpha \\ \mathbf{U}_{R1} - \bar{\mathbf{U}}_{RD}^* = \mathbf{A}_x(\bar{\mathbf{U}}_{RD}^*, \mathbf{U}_{R1})^{-1} \mathbf{S}_x(\bar{\mathbf{U}}_{RD}^*, \mathbf{U}_{R1})(\sigma_{RU} - \sigma_{LU}) & \equiv \beta \\ \bar{\mathbf{U}}_{RD}^* - \mathbf{U}_{2D} = \mathbf{A}_y(\mathbf{U}_{2D}, \bar{\mathbf{U}}_{RD}^*)^{-1} \mathbf{S}_y(\mathbf{U}_{2D}, \bar{\mathbf{U}}_{RD}^*)(\sigma_{LU} - \sigma_{LD}) & \equiv \gamma \\ \bar{\mathbf{U}}_{LD}^* - \mathbf{U}_{1D} = \mathbf{A}_y(\mathbf{U}_{1D}, \bar{\mathbf{U}}_{LD}^*)^{-1} \mathbf{S}_y(\mathbf{U}_{1D}, \bar{\mathbf{U}}_{LD}^*)(\sigma_{LU} - \sigma_{LD}) & \equiv \delta \end{cases} \quad (\text{B.0.25})$$

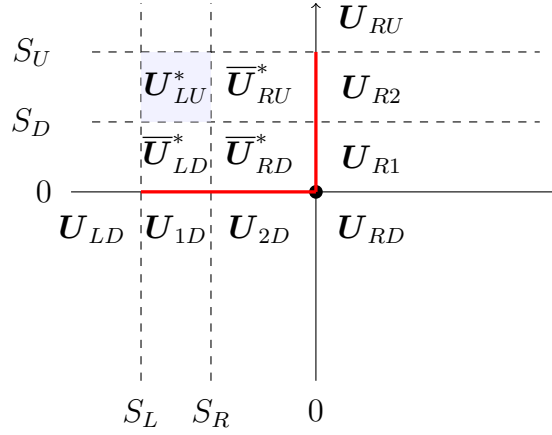


Figure B.6: Sketch of the jumps through the zero wave for the case $S_R < 0, S_D > 0$.

Therefore,

$$\begin{cases} \bar{U}_{RU}^* = U_{R2} - \delta, \\ \bar{U}_{RD}^* = U_{2D} + \alpha, \\ \bar{U}_{LD}^* = U_{1D} + \alpha, \\ U_{LU}^* = \frac{S_R S_D \bar{U}_{RD}^* + S_R (S_U - S_D) \bar{U}_{RU}^* - S_D (S_R - S_L) \bar{U}_{LD}^* - S_L S_U U_{LU} + \mathcal{C}}{(S_R - S_L)(S_U - S_D)}. \end{cases} \quad (\text{B.0.26})$$

- Case $S_R < 0, S_U < 0$. Integrating $\mathcal{R}(\xi, \psi) - \mathbf{W}_0(\xi, \psi)$ on $[S_L, S_R] \times [0, S_U]$ we get

$$\begin{aligned} (S_R - S_L)(S_U - S_D)U_{LD}^* - S_R(S_U - S_D)\bar{U}_{RD}^* - (S_R - S_L)S_U\bar{U}_{LU}^* \\ + S_R S_U \bar{U}_{RU}^* - S_L S_D U_{LD} = \mathcal{C}. \end{aligned} \quad (\text{B.0.27})$$

By the Rankine-Hugoniot conditions (1.3.13) (see Figure B.7),

$$\begin{cases} U_{1U} - \bar{U}_{LU}^* = \mathbf{A}_y(\bar{U}_{LU}^*, U_{1U})^{-1} \mathbf{S}_y(\bar{U}_{LU}^*, U_{1U})(\sigma_{LU} - \sigma_{LD}) \equiv \alpha \\ U_{2U} - \bar{U}_{RU}^* = \mathbf{A}_y(\bar{U}_{RU}^*, U_{2U})^{-1} \mathbf{S}_y(\bar{U}_{RU}^*, U_{2U})(\sigma_{LU} - \sigma_{LD}) \equiv \beta \\ U_{R2} - \bar{U}_{RU}^* = \mathbf{A}_x(\bar{U}_{RU}^*, U_{R2})^{-1} \mathbf{S}_x(\bar{U}_{RU}^*, U_{R2})(\sigma_{RD} - \sigma_{LD}) \equiv \gamma \\ U_{R1} - \bar{U}_{RD}^* = \mathbf{A}_x(\bar{U}_{RD}^*, U_{R1})^{-1} \mathbf{S}_x(\bar{U}_{RD}^*, U_{R1})(\sigma_{RD} - \sigma_{LD}) \equiv \delta \end{cases} \quad (\text{B.0.28})$$

Therefore,

$$\begin{cases} \bar{U}_{LU}^* = U_{1U} - \alpha, \\ \bar{U}_{RU}^* = U_{R2} - \gamma, \\ \bar{U}_{RD}^* = U_{R1} - \delta, \\ U_{LD}^* = \frac{S_R(S_U - S_D)\bar{U}_{RD}^* + (S_R - S_L)S_U\bar{U}_{LU}^* - S_R S_U \bar{U}_{RU}^* + S_L S_D \bar{U}_{LD} + \mathcal{C}}{(S_R - S_L)(S_U - S_D)}. \end{cases} \quad (\text{B.0.29})$$

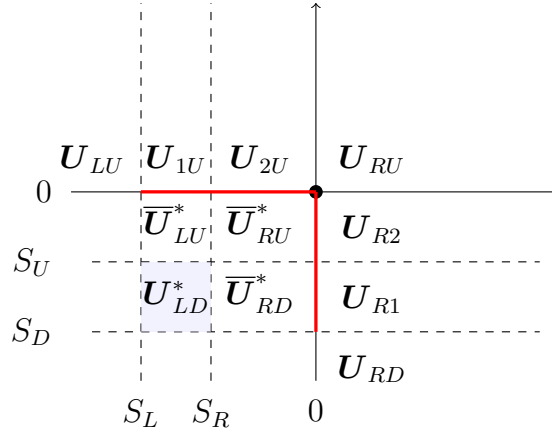


Figure B.7: Sketch of the jumps through the zero wave for the case $S_R < 0, S_U < 0$.

Once the intermediate states have been computed, we need to define the contributions at the vertex for each case. According to (4.1.8), the contributions for each case are given by

- If $S_L < 0 < S_R, S_D < 0 < S_U$:

$$\begin{aligned} \mathbf{D}^{LD} &= -S_L S_D (\mathbf{U}_{LD}^* - \mathbf{U}_{LD}), \\ \mathbf{D}^{LU} &= S_L S_U (\mathbf{U}_{LU}^* - \mathbf{U}_{LU}), \\ \mathbf{D}^{RD} &= S_R S_D (\mathbf{U}_{RD}^* - \mathbf{U}_{RD}), \\ \mathbf{D}^{RU} &= -S_R S_U (\mathbf{U}_{RU}^* - \mathbf{U}_{RU}). \end{aligned}$$

- If $S_L < 0 < S_R, S_D > 0$:

$$\begin{aligned} \mathbf{D}^{LD} &= \mathbf{D}^{RD} = 0, \\ \mathbf{D}^{LU} &= S_L S_D (\bar{\mathbf{U}}_{LD}^* - \mathbf{U}_{LU}) + S_L (S_U - S_D) (\mathbf{U}_{LU}^* - \mathbf{U}_{LU}), \\ \mathbf{D}^{RU} &= -S_R S_D (\bar{\mathbf{U}}_{RD}^* - \mathbf{U}_{RU}) - S_R (S_U - S_D) (\mathbf{U}_{RU}^* - \mathbf{U}_{RU}). \end{aligned}$$

- If $S_L < 0 < S_R, S_U < 0$:

$$\begin{aligned} \mathbf{D}^{LU} &= \mathbf{D}^{RU} = 0, \\ \mathbf{D}^{LD} &= S_L (S_U - S_D) (\mathbf{U}_{LD}^* - \mathbf{U}_{LD}) - S_L S_U (\bar{\mathbf{U}}_{LU}^* - \mathbf{U}_{LD}), \\ \mathbf{D}^{RD} &= -S_R (S_U - S_D) (\mathbf{U}_{RD}^* - \mathbf{U}_{RD}) + S_R S_U (\bar{\mathbf{U}}_{RU}^* - \mathbf{U}_{RD}). \end{aligned}$$

- If $S_L > 0, S_D < 0 < S_U$:

$$\begin{aligned} \mathbf{D}^{LD} &= \mathbf{D}^{LU} = 0, \\ \mathbf{D}^{RD} &= S_L S_D (\bar{\mathbf{U}}_{LD}^* - \mathbf{U}_{RD}) + (S_R - S_L) S_D (\mathbf{U}_{RD}^* - \mathbf{U}_{RD}), \\ \mathbf{D}^{RU} &= -S_L S_U (\bar{\mathbf{U}}_{LU}^* - \mathbf{U}_{RU}) - (S_R - S_L) S_U (\mathbf{U}_{RU}^* - \mathbf{U}_{RU}). \end{aligned}$$

Algorithm 1 Computation of the intermediate HLL states in all the possible cases.

```

1: subsonicx ← (SL < 0) * (SR > 0)
2: subsonicy ← (SD < 0) * (SU > 0)
3: if subsonicx then
4:   if subsonicy then
5:     UAB* ← (4.2.8)-(4.2.9)
6:   else
7:     if SD ≥ 0 then
8:       UAB* ← (4.2.15)-(4.2.16)
9:     else if SU ≤ 0 then
10:      UAB* ← (B.0.11)
11:     end if
12:   end if
13: else
14:   if SL ≥ 0 then
15:     if subsonicy then
16:       UAB* ← (B.0.14)
17:     else
18:       if SD ≥ 0 then
19:         UAB* ← (B.0.17)
20:       else if SU ≤ 0 then
21:         UAB* ← (B.0.20)
22:       end if
23:     end if
24:   else if SR ≤ 0 then
25:     if subsonicy then
26:       UAB* ← (B.0.23)
27:     else
28:       if SD ≥ 0 then
29:         UAB* ← (B.0.26)
30:       else if SU ≤ 0 then
31:         UAB* ← (B.0.29)
32:       end if
33:     end if
34:   end if
35: end if

```

- If $S_L > 0, S_D > 0$:

$$\begin{aligned} \mathbf{D}^{LD} = \mathbf{D}^{LU} = \mathbf{D}^{RD} &= 0, \\ \mathbf{D}^{RU} &= -S_L S_D (\bar{\mathbf{U}}_{LD}^* - \mathbf{U}_{RU}) - S_L (S_U - S_D) (\bar{\mathbf{U}}_{LU}^* - \mathbf{U}_{RU}) \\ &\quad - (S_R - S_L) S_D (\bar{\mathbf{U}}_{RD}^* - \mathbf{U}_{RU}) - (S_R - S_L) (S_U - S_D) (\mathbf{U}_{RU}^* - \mathbf{U}_{RU}). \end{aligned}$$

- If $S_L > 0, S_U < 0$:

$$\begin{aligned} \mathbf{D}^{LD} = \mathbf{D}^{LU} = \mathbf{D}^{RU} &= 0, \\ \mathbf{D}^{RD} &= -S_L (S_U - S_D) (\bar{\mathbf{U}}_{LD}^* - \mathbf{U}_{RD}) + S_L S_U (\bar{\mathbf{U}}_{LU}^* - \mathbf{U}_{RD}) \\ &\quad - (S_R - S_L) (S_U - S_D) (\mathbf{U}_{RD}^* - \mathbf{U}_{RD}) + (S_R - S_L) S_U (\bar{\mathbf{U}}_{RU}^* - \mathbf{U}_{RD}). \end{aligned}$$

- If $S_R < 0, S_D < 0 < S_U$:

$$\begin{aligned} \mathbf{D}^{RD} = \mathbf{D}^{RU} &= 0, \\ \mathbf{D}^{LD} &= -S_R S_D (\bar{\mathbf{U}}_{RD}^* - \mathbf{U}_{LD}) + (S_R - S_L) S_D (\mathbf{U}_{LD}^* - \mathbf{U}_{LD}), \\ \mathbf{D}^{LU} &= S_R S_U (\bar{\mathbf{U}}_{RU}^* - \mathbf{U}_{LU}) - (S_R - S_L) S_U (\mathbf{U}_{LU}^* - \mathbf{U}_{LU}). \end{aligned}$$

- If $S_R < 0, S_D > 0$:

$$\begin{aligned} \mathbf{D}^{LD} = \mathbf{D}^{RD} = \mathbf{D}^{RU} &= 0, \\ \mathbf{D}^{LU} &= -(S_R - S_L) S_D (\bar{\mathbf{U}}_{LD}^* - \mathbf{U}_{LU}) - (S_R - S_L) (S_U - S_D) (\mathbf{U}_{LU}^* - \mathbf{U}_{LU}) \\ &\quad + S_R S_D (\bar{\mathbf{U}}_{RD}^* - \mathbf{U}_{LU}) + S_R (S_U - S_D) (\bar{\mathbf{U}}_{RU}^* - \mathbf{U}_{LU}). \end{aligned}$$

- If $S_R < 0, S_U < 0$:

$$\begin{aligned} \mathbf{D}^{LU} = \mathbf{D}^{RD} = \mathbf{D}^{RU} &= 0, \\ \mathbf{D}^{LD} &= -(S_R - S_L) (S_U - S_D) (\mathbf{U}_{LD}^* - \mathbf{U}_{LD}) + (S_R - S_L) S_U (\bar{\mathbf{U}}_{LU}^* - \mathbf{U}_{LD}) \\ &\quad + S_R (S_U - S_D) (\bar{\mathbf{U}}_{RD}^* - \mathbf{U}_{LD}) - S_R S_U (\bar{\mathbf{U}}_{RU}^* - \mathbf{U}_{LD}). \end{aligned}$$

The algorithm for the computation of the contributions \mathcal{D}^{AB} for each case follows the same structure of Algorithm 1, changing \mathbf{U}_{AB}^* by \mathcal{D}^{AB} in the respective cases. Recall that when the self-similar solution only depends on \mathbf{U} , we write $\mathcal{R}(\xi, \psi)$; when it depends on $\mathbf{W} = (\mathbf{U}, \sigma)^T$, we write $\mathcal{R}(\xi, \psi)$. In the same way, \mathcal{D}^{AB} indicates dependence only on \mathbf{U} , while \mathcal{D}^{AB} indicates dependence on \mathbf{W} .

Appendix C

Full algorithm for 2D AVM-type formulation

In Section 4.3.1, the two-dimensional version of the HLL scheme in Section 4.2 was rewritten in the PVM framework. In Section 4.3.2 the PVM-HLL two-dimensional scheme was extended to more precise two-dimensional AVM-type Riemann solvers. Here, for the sake of completeness, we present the full algorithm for 2D AVM-type solvers, considering all the subsonic and supersonic cases. To do this, we need to consider the intermediate states and the \mathbf{D}^{AB} contributions defined on Appendix B. Following the same procedure as in Section 4.3.2, we can find the following expressions:

- Subsonic case in both directions, i.e., $S_L < 0 < S_R$ and $S_D < 0 < S_U$:

$$\begin{aligned} \mathbf{D}^{LD} &= \frac{S_L}{2} \left(\frac{S_L}{S_R - S_L} \mathbf{D}_y^{1,-} - \frac{S_R}{S_R - S_L} \mathbf{D}_y^{2,-} - \mathbf{D}_y^{L,-} \right) \\ &\quad + \frac{S_D}{2} \left(\frac{S_D}{S_U - S_D} \mathbf{D}_x^{1,-} - \frac{S_U}{S_U - S_D} \mathbf{D}_x^{2,-} - \mathbf{D}_x^{D,-} \right), \\ \mathbf{D}^{LU} &= \frac{S_L}{2} \left(\frac{S_L}{S_R - S_L} \mathbf{D}_y^{1,+} - \frac{S_R}{S_R - S_L} \mathbf{D}_y^{2,+} - \mathbf{D}_y^{L,+} \right) \\ &\quad - \frac{S_U}{2} \left(\frac{S_D}{S_U - S_D} \mathbf{D}_x^{1,-} - \frac{S_U}{S_U - S_D} \mathbf{D}_x^{2,-} - \mathbf{D}_x^{U,-} \right), \\ \mathbf{D}^{RD} &= -\frac{S_R}{2} \left(\frac{S_L}{S_R - S_L} \mathbf{D}_y^{1,-} - \frac{S_R}{S_R - S_L} \mathbf{D}_y^{2,-} - \mathbf{D}_y^{R,-} \right) \\ &\quad + \frac{S_D}{2} \left(\frac{S_D}{S_U - S_D} \mathbf{D}_x^{1,+} - \frac{S_U}{S_U - S_D} \mathbf{D}_x^{2,+} - \mathbf{D}_x^{D,+} \right), \\ \mathbf{D}^{RU} &= -\frac{S_R}{2} \left(\frac{S_L}{S_R - S_L} \mathbf{D}_y^{1,+} - \frac{S_R}{S_R - S_L} \mathbf{D}_y^{2,+} - \mathbf{D}_y^{R,+} \right) \\ &\quad - \frac{S_U}{2} \left(\frac{S_D}{S_U - S_D} \mathbf{D}_x^{1,+} - \frac{S_U}{S_U - S_D} \mathbf{D}_x^{2,+} - \mathbf{D}_x^{U,+} \right). \end{aligned} \tag{C.0.1}$$

- Case $S_L < 0 < S_R$ and $S_D > 0$:

$$\begin{aligned}
\mathbf{D}^{LD} &= \mathbf{D}^{RD} = 0, \\
\mathbf{D}^{LU} &= \frac{S_U}{2} \mathbf{D}_x^{U,-} + \frac{(S_U - S_D)}{2} \mathbf{D}_x^{2,-} + \frac{S_D}{2} \mathbf{D}_x^{1,-} \\
&\quad - \frac{S_L}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{LD,LU}) \frac{d}{ds} \Phi_{LD,LU}^U - \mathbf{S}_y(\Phi_{LD,LU}) \frac{d}{ds} \Phi_{LD,LU}^\sigma \right) ds \\
&\quad + \frac{S_L}{2(S_R - S_L)} \left(S_L \int_0^1 \left(\mathbf{A}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^U - \mathbf{S}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^\sigma \right) ds \right. \\
&\quad \left. - S_R \int_0^1 \left(\mathbf{A}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^U - \mathbf{S}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^\sigma \right) ds \right), \\
\mathbf{D}^{RU} &= \frac{S_U}{2} \mathbf{D}_x^{U,+} + \frac{(S_U - S_D)}{2} \mathbf{D}_x^{2,+} + \frac{S_D}{2} \mathbf{D}_x^{1,+} \\
&\quad + \frac{S_R}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{RD,RU}) \frac{d}{ds} \Phi_{RD,RU}^U - \mathbf{S}_y(\Phi_{RD,RU}) \frac{d}{ds} \Phi_{RD,RU}^\sigma \right) ds \\
&\quad - \frac{S_R}{2(S_R - S_L)} \left(S_L \int_0^1 \left(\mathbf{A}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^U - \mathbf{S}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^\sigma \right) ds \right. \\
&\quad \left. - S_R \int_0^1 \left(\mathbf{A}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^U - \mathbf{S}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^\sigma \right) ds \right).
\end{aligned} \tag{C.0.2}$$

- Case $S_L < 0 < S_R$ and $S_U > 0$:

$$\begin{aligned}
\mathbf{D}^{LU} &= \mathbf{D}^{RU} = 0, \\
\mathbf{D}^{LD} &= -\frac{S_D}{2} \mathbf{D}_x^{D,-} + \frac{(S_U - S_D)}{2} \mathbf{D}_x^{1,-} - \frac{S_U}{2} \mathbf{D}_x^{2,-} \\
&\quad - \frac{S_L}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{LD,LU}) \frac{d}{ds} \Phi_{LD,LU}^U - \mathbf{S}_y(\Phi_{LD,LU}) \frac{d}{ds} \Phi_{LD,LU}^\sigma \right) ds \\
&\quad + \frac{S_L}{2(S_R - S_L)} \left(S_L \int_0^1 \left(\mathbf{A}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^U - \mathbf{S}_2(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^\sigma \right) ds \right. \\
&\quad \left. - S_R \int_0^1 \left(\mathbf{A}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^U - \mathbf{S}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^\sigma \right) ds \right), \\
\mathbf{D}^{RD} &= -\frac{S_D}{2} \mathbf{D}_x^{D,+} + \frac{(S_U - S_D)}{2} \mathbf{D}_x^{1,+} - \frac{S_U}{2} \mathbf{D}_x^{2,+} \\
&\quad + \frac{S_R}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{RD,RU}) \frac{d}{ds} \Phi_{RD,RU}^U - \mathbf{S}_y(\Phi_{RD,RU}) \frac{d}{ds} \Phi_{RD,RU}^\sigma \right) ds \\
&\quad - \frac{S_R}{2(S_R - S_L)} \left(S_L \int_0^1 \left(\mathbf{A}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^U - \mathbf{S}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^\sigma \right) ds \right. \\
&\quad \left. - S_R \int_0^1 \left(\mathbf{A}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^U - \mathbf{S}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^\sigma \right) ds \right).
\end{aligned} \tag{C.0.3}$$

- Case $S_D < 0 < S_U$ and $S_L > 0$:

$$\begin{aligned}
\mathbf{D}^{LD} &= \mathbf{D}^{LU} = 0, \\
\mathbf{D}^{RD} &= \frac{S_R}{2} \mathbf{D}_y^{R,-} + \frac{(S_R - S_L)}{2} \mathbf{D}_y^{2,-} + \frac{S_L}{2} \mathbf{D}_y^{1,-} \\
&\quad - \frac{S_D}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{LD,RD}) \frac{d}{ds} \Phi_{LD,RD}^U - \mathbf{S}_x(\Phi_{LD,RD}) \frac{d}{ds} \Phi_{LD,RD}^\sigma \right) ds \\
&\quad + \frac{S_D}{2(S_U - S_D)} \left(S_D \int_0^1 \left(\mathbf{A}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^U - \mathbf{S}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^\sigma \right) ds \right. \\
&\quad \left. - S_U \int_0^1 \left(\mathbf{A}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^U - \mathbf{S}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^\sigma \right) ds \right), \\
\mathbf{D}^{RU} &= \frac{S_R}{2} \mathbf{D}_y^{R,+} + \frac{(S_R - S_L)}{2} \mathbf{D}_y^{2,+} + \frac{S_L}{2} \mathbf{D}_y^{1,+} \\
&\quad + \frac{S_U}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{LU,RU}) \frac{d}{ds} \Phi_{LU,RU}^U - \mathbf{S}_x(\Phi_{LU,RU}) \frac{d}{ds} \Phi_{LU,RU}^\sigma \right) ds \\
&\quad - \frac{S_U}{2(S_U - S_D)} \left(S_D \int_0^1 \left(\mathbf{A}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^U - \mathbf{S}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^\sigma \right) ds \right. \\
&\quad \left. - S_U \int_0^1 \left(\mathbf{A}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^U - \mathbf{S}_1(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^\sigma \right) ds \right).
\end{aligned} \tag{C.0.4}$$

- Case $S_D < 0 < S_U$ and $S_R < 0$:

$$\begin{aligned}
\mathbf{D}^{RD} &= \mathbf{D}^{RU} = 0, \\
\mathbf{D}^{LD} &= -\frac{S_L}{2} \mathbf{D}_y^{L,-} + \frac{(S_R - S_L)}{2} \mathbf{D}_y^{1,-} - \frac{S_R}{2} \mathbf{D}_y^{2,-} \\
&\quad - \frac{S_D}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{LD,RD}) \frac{d}{ds} \Phi_{LD,RD}^U - \mathbf{S}_x(\Phi_{LD,RD}) \frac{d}{ds} \Phi_{LD,RD}^\sigma \right) ds \\
&\quad + \frac{S_D}{2(S_U - S_D)} \left(S_D \int_0^1 \left(\mathbf{A}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^U - \mathbf{S}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^\sigma \right) ds \right. \\
&\quad \left. - S_U \int_0^1 \left(\mathbf{A}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^U - \mathbf{S}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^\sigma \right) ds \right), \\
\mathbf{D}^{LU} &= -\frac{S_L}{2} \mathbf{D}_y^{L,+} + \frac{(S_R - S_L)}{2} \mathbf{D}_y^{1,+} - \frac{S_R}{2} \mathbf{D}_y^{2,+} \\
&\quad + \frac{S_U}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{LU,RU}) \frac{d}{ds} \Phi_{LU,RU}^U - \mathbf{S}_x(\Phi_{LU,RU}) \frac{d}{ds} \Phi_{LU,RU}^\sigma \right) ds \\
&\quad - \frac{S_U}{2(S_U - S_D)} \left(S_D \int_0^1 \left(\mathbf{A}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^U - \mathbf{S}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^\sigma \right) ds \right. \\
&\quad \left. - S_U \int_0^1 \left(\mathbf{A}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^U - \mathbf{S}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^\sigma \right) ds \right).
\end{aligned} \tag{C.0.5}$$

- Case $S_L > 0$ and $S_D > 0$:

$$\begin{aligned}
\mathbf{D}^{LD} &= \mathbf{D}^{RD} = \mathbf{D}^{LU} = 0, \\
\mathbf{D}^{RU} &= \frac{S_L}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^U - \mathbf{S}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^\sigma \right) ds \\
&\quad + \frac{S_R - S_L}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^U - \mathbf{S}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^\sigma \right) ds \\
&\quad + \frac{S_R}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{RD,RU}) \frac{d}{ds} \Phi_{RD,RU}^U - \mathbf{S}_y(\Phi_{RD,RU}) \frac{d}{ds} \Phi_{RD,RU}^\sigma \right) ds \quad (\text{C.0.6}) \\
&\quad + \frac{S_D}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^U - \mathbf{S}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^\sigma \right) ds \\
&\quad + \frac{S_U - S_D}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^U - \mathbf{S}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^\sigma \right) ds \\
&\quad + \frac{S_U}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{LU,RU}) \frac{d}{ds} \Phi_{LU,RU}^U - \mathbf{S}_x(\Phi_{LU,RU}) \frac{d}{ds} \Phi_{LU,RU}^\sigma \right) ds.
\end{aligned}$$

- Case $S_L > 0$ and $S_U < 0$:

$$\begin{aligned}
\mathbf{D}^{LD} &= \mathbf{D}^{LU} = \mathbf{D}^{RU} = 0, \\
\mathbf{D}^{RD} &= \frac{S_L}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^U - \mathbf{S}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^\sigma \right) ds \\
&\quad + \frac{S_R - S_L}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^U - \mathbf{S}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^\sigma \right) ds \\
&\quad + \frac{S_R}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{RD,RU}) \frac{d}{ds} \Phi_{RD,RU}^U - \mathbf{S}_y(\Phi_{RD,RU}) \frac{d}{ds} \Phi_{RD,RU}^\sigma \right) ds \quad (\text{C.0.7}) \\
&\quad - \frac{S_D}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{LD,RD}) \frac{d}{ds} \Phi_{LD,RD}^U - \mathbf{S}_x(\Phi_{LD,RD}) \frac{d}{ds} \Phi_{LD,RD}^\sigma \right) ds \\
&\quad + \frac{S_U - S_D}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^U - \mathbf{S}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^\sigma \right) ds \\
&\quad - \frac{S_U}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^U - \mathbf{S}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^\sigma \right) ds.
\end{aligned}$$

- Case $S_R < 0$ and $S_D > 0$:

$$\begin{aligned}
\mathbf{D}^{LD} &= \mathbf{D}^{RD} = \mathbf{D}^{RU} = 0, \\
\mathbf{D}^{LU} &= -\frac{S_L}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{LD,LU}) \frac{d}{ds} \Phi_{LD,LU}^U - \mathbf{S}_y(\Phi_{LD,LU}) \frac{d}{ds} \Phi_{LD,LU}^\sigma \right) ds \\
&\quad + \frac{S_R - S_L}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^U - \mathbf{S}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^\sigma \right) ds \\
&\quad - \frac{S_R}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^U - \mathbf{S}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^\sigma \right) ds \\
&\quad + \frac{S_D}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^U - \mathbf{S}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^\sigma \right) ds \\
&\quad + \frac{S_U - S_D}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^U - \mathbf{S}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^\sigma \right) ds \\
&\quad + \frac{S_U}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{LU,RU}) \frac{d}{ds} \Phi_{LU,RU}^U - \mathbf{S}_x(\Phi_{LU,RU}) \frac{d}{ds} \Phi_{LU,RU}^\sigma \right) ds.
\end{aligned} \tag{C.0.8}$$

- Case $S_R < 0$ and $S_U < 0$:

$$\begin{aligned}
\mathbf{D}^{LU} &= \mathbf{D}^{RD} = \mathbf{D}^{RU} = 0, \\
\mathbf{D}^{LD} &= -\frac{S_L}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{LD,LU}) \frac{d}{ds} \Phi_{LD,LU}^U - \mathbf{S}_y(\Phi_{LD,LU}) \frac{d}{ds} \Phi_{LD,LU}^\sigma \right) ds \\
&\quad + \frac{S_R - S_L}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^U - \mathbf{S}_y(\Phi_{1D,1U}) \frac{d}{ds} \Phi_{1D,1U}^\sigma \right) ds \\
&\quad - \frac{S_R}{2} \int_0^1 \left(\mathbf{A}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^U - \mathbf{S}_y(\Phi_{2D,2U}) \frac{d}{ds} \Phi_{2D,2U}^\sigma \right) ds \\
&\quad - \frac{S_D}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{LD,RD}) \frac{d}{ds} \Phi_{LD,RD}^U - \mathbf{S}_x(\Phi_{LD,RD}) \frac{d}{ds} \Phi_{LD,RD}^\sigma \right) ds \\
&\quad + \frac{S_U - S_D}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^U - \mathbf{S}_x(\Phi_{L1,R1}) \frac{d}{ds} \Phi_{L1,R1}^\sigma \right) ds \\
&\quad - \frac{S_U}{2} \int_0^1 \left(\mathbf{A}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^U - \mathbf{S}_x(\Phi_{L2,R2}) \frac{d}{ds} \Phi_{L2,R2}^\sigma \right) ds.
\end{aligned} \tag{C.0.9}$$

It is important to remind that, in order to be consistent with (4.1.7), we must have

$$-\sum_{\substack{A \in \{L,R\} \\ B \in \{D,U\}}} \mathbf{D}^{AB} = \mathcal{C} \tag{C.0.10}$$

where \mathbf{D}^{AB} is given by (C.0.1)-(C.0.9), for a given vertex region \mathcal{V} . Adding up the four contributions in each case we have (C.0.10) satisfied.

Appendix D

An efficient implementation of well-balanced PVM schemes

In [50, Appendix A] it was showed a short way to compute an approximate Osher Solomon-Chebyshev scheme. We reproduce here the algorithm in the case of well-balanced PVM schemes by using the modified identity technique (see Section 2.4). Consider the numerical fluctuation (4.3.7) written as

$$\mathbf{D}_\zeta^\pm = \frac{1}{2}(\Delta\mathbf{F} + \mathbf{B}_\zeta(\Delta\mathbf{U}) - \mathbf{S}_\zeta(\Delta H)) \pm \frac{1}{2}\mathbf{Q}_\zeta(\Delta\mathbf{U} - \mathbf{A}_\zeta^{-1}\mathbf{S}_\zeta(\Delta H)), \quad (\text{D.0.1})$$

where $\Delta(\cdot) = (\cdot)_1 - (\cdot)_0$. The dependence of \mathbf{B}_ζ and \mathbf{S}_ζ on the family of paths Φ was dropped for clarity. The product of the viscosity matrix (1.5.4) for a PVM scheme with $\Delta\mathbf{U} - \mathbf{A}_\zeta^{-1}\mathbf{S}_\zeta(\Delta H)$ can be written as

$$\begin{aligned} \Theta_\zeta &:= |\lambda_{\zeta,\max}|p(|\lambda_{\zeta,\max}|^{-1}\mathbf{A}_\zeta)(\Delta\mathbf{U} - \mathbf{A}_\zeta^{-1}\mathbf{S}_\zeta(\Delta H)) \\ &= |\lambda_{\zeta,\max}|\alpha_0(\Delta\mathbf{U} - \mathbf{A}_\zeta^{-1}\mathbf{S}_\zeta(\Delta H)) + \sum_{i=1}^r \frac{\alpha_i}{|\lambda_{\zeta,\max}|^{i-1}}\mathbf{A}_\zeta^{i-1}(\Delta\mathbf{F} + \mathbf{B}_\zeta(\Delta\mathbf{U}) - \mathbf{S}_\zeta(\Delta H)), \end{aligned}$$

where p is some polynomial function of degree r . Recall that $\mathbf{A}_\zeta = \mathbf{J}_\zeta + \mathbf{B}_\zeta$, where \mathbf{J}_ζ is the Jacobian matrix with respect to $\zeta = x, y$. By the modified identity procedure, we can define, respectively, for the one-layer and two-layer shallow water systems,

$$\mathcal{I}_{\text{mod}} = \Delta\mathbf{U} - \widehat{\mathbf{A}_\zeta^{-1}\mathbf{S}_\zeta}(\Delta H) = \begin{pmatrix} \Delta\eta \\ \Delta q_x \\ \Delta q_y \end{pmatrix}$$

or

$$\mathcal{I}_{\text{mod}} = \Delta\mathbf{U} - \widehat{\mathbf{A}_\zeta^{-1}\mathbf{S}_\zeta}(\Delta H) = \begin{pmatrix} \Delta h_1 \\ \Delta q_{x1} \\ \Delta q_{y1} \\ \Delta \eta_2 \\ \Delta q_{x2} \\ \Delta q_{y2} \end{pmatrix}.$$

Once $\phi := \Delta \mathbf{F} + \mathcal{B}_\zeta(\Delta \mathbf{U}) - \mathcal{S}_\zeta(\Delta H)$ has been computed, we can calculate recursively the diffusion term of (D.0.1) as follows:

- $\Theta_0 = |\lambda_{\zeta, \max}|^{-1} \mathbf{A}_\zeta * \phi,$
- $\Theta_1 = |\lambda_{\zeta, \max}|^{-1} \mathbf{A}_\zeta * \Theta_0,$
- $\Theta_r = |\lambda_{\zeta, \max}|^{-1} \mathbf{A}_\zeta * \Theta_{r-1}, r \geq 2,$

and, finally

$$\Theta_\zeta = |\lambda_{\zeta, \max}| \alpha_0 \mathcal{I}_{\text{mod}} + \alpha_1 \phi + \sum_{i=2}^r \alpha_i \Theta_i.$$

In this way, we compute efficiently the diffusion term, avoiding matrix multiplications. Jacobian-free implementations are also possible following the guidelines in [50].



UNIVERSIDAD
DE MÁLAGA

Bibliography

- [1] R. Abgrall. *A genuinely multidimensional Riemann solver*. [Research Report] RR-1859. INRIA, 1993, 20 pp. URL: <https://hal.inria.fr/inria-00074814>.
- [2] R. Abgrall. Approximation du problème Riemann vraiment multidimensionnel des équations d'Euler par une méthode de type Roe, I: La linéarisation. *C. R. Acad. Sci. Paris Sér. I Math.* 319(5) (1994), 499–504.
- [3] R. Abgrall. Approximation du problème Riemann vraiment multidimensionnel des équations d'Euler par une méthode de type Roe, II: Solution du problème de Riemann approché. *C. R. Acad. Sci. Paris Sér. I Math.* 319(6) (1994), 625–629.
- [4] R. Abgrall. *A review of Residual distribution schemes for hyperbolic and parabolic problems: the july 2010 state of the art*. [Research Report] INRIA. 2011, 39 pp. URL: <https://hal.inria.fr/inria-00526162v2>.
- [5] R. Abgrall and S. Karni. A comment on the computation of non-conservative products. *Journal of Computational Physics* 229 (2010), 2759–2763. DOI: <https://doi.org/10.1016/j.jcp.2009.12.015>.
- [6] R. Artebrandt and M. Torrilhon. Increasing the accuracy in locally divergence-preserving finite volume schemes for MHD. *J. Comput. Phys.* 227 (2008), 3405–3427. DOI: <https://doi.org/10.1016/j.jcp.2007.12.003>.
- [7] M. de la Asunción. Desarrollo de software numérico de simulación de flujos geofísicos basado en volúmenes finitos usando hardware gráfico. PhD Thesis. Universidad de Granada, 2012. URL: <https://hera.ugr.es/tesisugr/21605099.pdf>.
- [8] M. de la Asunción, M.J. Castro, J.M. Mantas, and S. Ortega. Numerical simulation of tsunamis generated by landslides on multiple GPUs. *Advances in Engineering Software* 99 (2016), 59–72. DOI: <https://doi.org/10.1016/j.advengsoft.2016.05.005>.
- [9] M. de la Asunción, J.M. Mantas, and M.J. Castro. Programming CUDA-based GPUs to simulate two-layer shallow water flows. In: *Lectures Notes in Compute Science*. Ed. by P. D'Ambra, M. Guarracino, and D. Talia. Vol. 6272. Euro-Par 2010 – Parallel Processing. Springer, Berlin, Heidelberg, 2010. DOI: https://doi.org/10.1007/978-3-642-15291-7_32.
- [10] M. de la Asunción, J.M. Mantas, and M.J. Castro. Simulation of one-layer shallow water systems on multicore and CUDA architectures. *The Journal of Supercomputing* 58(2) (2011), 206–214. DOI: <https://doi.org/10.1007/s11227-010-0406-2>.



- [11] M. de la Asunción, J.M. Mantas, M.J. Castro, and E.D. Fernández-Nieto. An MPI-CUDA implementation of an improved Roe method for two-layer shallow water systems. *Journal of Parallel and Distributed Computing, Special Issue on Accelerators for High-Performance Computing* 72.9 (2012), 1065–1072. DOI: <https://doi.org/10.1016/j.jpdc.2011.07.012>.
- [12] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, and B. Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM J. Sci. Comput.* 25 (2004), 2050–2065. DOI: <https://doi.org/10.1137/S1064827503431090>.
- [13] M.R. Baer and J.W. Nunziato. A two-phase mixture theory for the deflagration-to-detonation transition (ddt) in reactive granular materials. *Int. J. Multiphase Flow* 12(6) (1986), 861–889. DOI: [https://doi.org/10.1016/0301-9322\(86\)90033-9](https://doi.org/10.1016/0301-9322(86)90033-9).
- [14] D.S. Balsara. Multidimensional HLLC Riemann solver: application to Euler and magnetohydrodynamic flows. *J. Comput. Phys.* 229 (2010), 1970–1993. DOI: <https://doi.org/10.1016/j.jcp.2009.11.018>.
- [15] D.S. Balsara. A two-dimensional HLLC Riemann solver for conservation laws: application to Euler and magnetohydrodynamic flows. *J. Comput. Phys.* 231 (2012), 7476–7503. DOI: <https://doi.org/10.1016/j.jcp.2011.12.025>.
- [16] D.S. Balsara. Multidimensional Riemann problem with self-similar internal structure - Part I - Application to hyperbolic conservation laws on structured meshes. *J. Comput. Phys.* 277 (2014), 163–200. DOI: <https://doi.org/10.1016/j.jcp.2014.07.053>.
- [17] D.S. Balsara. Higher-order accurate space-time schemes for computational astrophysics - Part I: finite volume methods. *Living Rev. Comput. Astrophys.* 3:2 (2017). 138 pp. DOI: <https://doi.org/10.1007/s41115-017-0002-8>.
- [18] D.S. Balsara and M. Dumbser. Multidimensional Riemann problem with self-similar internal structure - Part II - Application to hyperbolic conservation laws on unstructured meshes. *J. Comput. Phys.* 287 (2015), 269–292. DOI: <https://doi.org/10.1016/j.jcp.2014.11.004>.
- [19] D.S. Balsara, M. Dumbser, and R. Abgrall. Multidimensional HLLC Riemann solver for unstructured meshes - With application to Euler and MHD flows. *J. Comput. Phys.* 261 (2014), 172–208. DOI: <https://doi.org/10.1016/j.jcp.2013.12.029>.
- [20] D.S. Balsara and J. Kim. A comparison between divergence-cleaning and staggered-mesh formulations for numerical magnetohydrodynamics. *Astrophys. J.* 602 (2004), 1079–1090. DOI: <https://doi.org/10.1086/381051>.
- [21] D.S. Balsara and B. Nkonga. Multidimensional Riemann problem with self-similar internal structure - Part III - A multidimensional analogue of the HLLC Riemann solver for conservative hyperbolic systems. *J. Comput. Phys.* 346 (2017), 25–48. DOI: <https://doi.org/10.1016/j.jcp.2017.05.038>.
- [22] D.S. Balsara and D.S. Spicer. A staggered mesh algorithm using high order Godunov fluxes to ensure solenoidal magnetic fields in magnetohydrodynamic simulations. *J. Comput. Phys.* 149 (1999), 270–292. DOI: <https://doi.org/10.1006/jcph.1998.6153>.

- [23] C.M. Bard and J.C. Dorelli. A simple GPU-accelerated two-dimensional MUSCL-Hancock solver for ideal magnetohydrodynamics. *J. Comput. Phys.* 259 (2014), 444–460. DOI: <https://doi.org/10.1016/j.jcp.2013.12.006>.
- [24] P. Batten, W. Clarke, C. Lambert, and D.M. Causon. On the choice of wavespeeds for the HLLC Riemann solver. *SIAM J. Sci. Comput. Phys.* 18 (1997), 1553–1570. DOI: <https://doi.org/10.1137/S1064827593260140>.
- [25] M.J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.* 82 (1989), 64–84. DOI: [https://doi.org/10.1016/0021-9991\(89\)90035-1](https://doi.org/10.1016/0021-9991(89)90035-1).
- [26] M.J. Berger and A. Jameson. Automatic adaptive grid refinement for the Euler equations. *AIAA Journal* 23 (1985), 561–568. DOI: <https://doi.org/10.2514/3.8951>.
- [27] M.J. Berger and R. leVeque. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM J. Num. Anal.* 35 (1998), 2298–2316. DOI: <https://doi.org/10.1137/S0036142997315974>.
- [28] M.J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.* 53 (1984), 484–512. DOI: [https://doi.org/10.1016/0021-9991\(84\)90073-1](https://doi.org/10.1016/0021-9991(84)90073-1).
- [29] A. Bermúdez and M.E. Vázquez. Upwind methods for hyperbolic conservation laws with source terms. *Computers & Fluids* 23.8 (1994), 1049–1071. DOI: [https://doi.org/10.1016/0045-7930\(94\)90004-3](https://doi.org/10.1016/0045-7930(94)90004-3).
- [30] J.U. Brackbill and J.C. Barnes. The effect of nonzero $\nabla \cdot B$ on the numerical solution of the magnetohydrodynamic equations. *J. Comput. Phys.* 35 (1980), 426–430. DOI: [https://doi.org/10.1016/0021-9991\(80\)90079-0](https://doi.org/10.1016/0021-9991(80)90079-0).
- [31] M. Brio and C.C. Wu. An upwind differencing scheme for the equations of ideal magnetohydrodynamics. *J. Comput. Phys.* 75 (1988), 400–422. DOI: [https://doi.org/10.1016/0021-9991\(88\)90120-9](https://doi.org/10.1016/0021-9991(88)90120-9).
- [32] M. Brio, A.R. Zakharian, and G.M. Webb. Two dimensional Riemann solver for Euler equations of gas dynamics. *J. Comput. Phys.* 167 (2001), 177–195. DOI: <https://doi.org/10.1006/jcph.2000.6666>.
- [33] A.R. Brodtkorb, T.R. Hagen, and M.L. Sætra. Graphics processing unit (GPU) programming strategies and trends in GPU computing. *J. Parallel Distrib. Comput.* 73(1) (2013), 4–13. DOI: <https://doi.org/10.1016/j.jpdc.2012.04.003>.
- [34] D. Calhoun and R.J. LeVeque. A Cartesian grid finite-volume method for the advection-diffusion equation in irregular geometries. *J. Comput. Phys.* 157 (2000), 143–180. DOI: <https://doi.org/10.1006/jcph.1999.6369>.
- [35] P. Cargo and G. Gallice. Roe matrices for ideal MHD and systematic construction of Roe matrices for systems of conservation laws. *J. Comput. Phys.* 136 (1997), 446–466. DOI: <https://doi.org/10.1006/jcph.1997.5773>.

- [36] H. Carrillo and C. Parés. Compact Approximate Taylor methods for systems of conservation laws. *J. Sci. Comput.* 80 (2019), 1832–1866. DOI: <https://doi.org/10.1007/s10915-019-01005-1>.
- [37] H. Carrillo, C. Parés, and D. Zorío. Lax Wendroff approximate Taylor methods with fast and optimized weighted essentially non-oscillatory reconstructions. *To appear* (2020). URL: <https://arxiv.org/abs/2002.08426>.
- [38] M.J. Castro, T. Chacón, E.D. Fernández-Nieto, and C. Parés. On well-balanced finite volume methods for non-conservative non-homogeneous hyperbolic systems. *SIAM J. Sci. Comput.* 29 (2007), 1093–1126. DOI: <https://doi.org/10.1137/040607642>.
- [39] M.J. Castro, Y. Cheng, A. Chertock, and A. Kurganov. Solving two-mode shallow water equations using finite volume methods. *Commun. Comput. Phys.* 15(5) (2014), 1323–1354. DOI: <https://doi.org/10.4208/cicp.180513.230514a>.
- [40] M.J. Castro, M. de la Asunción, E.D. Fernández-Nieto, J.M. Gallardo, J.M. González-Vida, J. Macías, T. Morales, S. Ortega, and C. Parés. A review on high order well-balanced path-conservative finite volume schemes for geophysical flows. *Proc. Int. Congr. of Math. - ICM 2018* 3 (2018), 3499–3522. URL: <https://eta.impa.br/dl/119.pdf>.
- [41] M.J. Castro and E.D. Fernández-Nieto. A class of computationally fast first order finite volume solvers: PVM methods. *SIAM J. Sci. Comput.* 34 (2012), A2173–A2196. DOI: <https://doi.org/10.1137/100795280>.
- [42] M.J. Castro, E.D. Fernández-Nieto, and A.M. Ferreiro. Sediment transport models in shallow water equations and numerical approach by high order finite volume methods. *Comput. & Fluids* 37(3) (2008), 299–316. DOI: <https://doi.org/10.1016/j.compfluid.2007.07.017>.
- [43] M.J. Castro, E.D. Fernández-Nieto, A.M. Ferreiro, J.A. García-Rodríguez, and C. Parés. High order extensions of Roe schemes for two dimensional nonconservative hyperbolic systems. *J. Sci. Comput.* 39 (2009), 67–114. DOI: <https://doi.org/10.1007/s10915-008-9250-4>.
- [44] M.J. Castro, E.D. Fernández-Nieto, J.M. González-Vida, and C. Parés. Numerical treatment of the loss of hyperbolicity of the two-layer shallow-water system. *J. Sci. Comput.* 48 (2011), 16–40. DOI: <https://doi.org/10.1007/s10915-010-9427-5>.
- [45] M.J. Castro, E.D. Fernández-Nieto, T. Morales de Luna, G. Narbona-Reina, and C. Parés. A HLLC scheme for nonconservative hyperbolic systems. Application to turbidity currents with sediment transport. *ESAIM: Math. Model. Numer. Anal.* 47 (2013), 1–32. DOI: <https://doi.org/10.1051/m2an/2012017>.
- [46] M.J. Castro, A.M. Ferreiro Ferreiro, J.A. García-Rodríguez, J.M. González-Vida, J. Macías, C. Parés, and M.E. Vázquez-Cendón. The numerical treatment of wet/dry fronts in shallow flows: application to one-layer and two-layer systems. *Math. Comput. Model.* 42 (2005), 419–439. DOI: <https://doi.org/10.1016/j.mcm.2004.01.016>.
- [47] M.J. Castro, U.S. Fjordholm, S. Mishra, and C. Parés. Entropy conservative and entropy stable schemes for nonconservative hyperbolic systems. *SIAM J. Numer. Anal.* 51 (2013), 1371–1391. DOI: <https://doi.org/10.1137/110845379>.



- [48] M.J. Castro, J.M. Gallardo, J.A. López-García, and C. Parés. Well-balanced high order extensions of Godunov's method for semilinear balance laws. *SIAM J. Numer. Anal.* 46 (2009), 1012–1039. DOI: <https://doi.org/10.1137/060674879>.
- [49] M.J. Castro, J.M. Gallardo, and A. Marquina. A class of incomplete Riemann solvers based on uniform rational approximations to the absolute value function. *J. Sci. Comput.* 60 (2014), 363–389. DOI: <https://doi.org/10.1007/s10915-013-9800-2>.
- [50] M.J. Castro, J.M. Gallardo, and A. Marquina. Approximate Osher-Solomon schemes for hyperbolic systems. *Appl. Math. Comput.* 272 (2016), 347–368. DOI: <https://doi.org/10.1016/j.amc.2015.06.104>.
- [51] M.J. Castro, J.M. Gallardo, and A. Marquina. Jacobian-free approximate solvers for hyperbolic systems: Applications to relativistic magnetohydrodynamics. *Comput. Phys. Commun.* 219 (2017), 108–120. DOI: <https://doi.org/10.1016/j.cpc.2017.05.013>.
- [52] M.J. Castro, J.M. Gallardo, and C. Parés. High order finite volume schemes based on reconstruction of states for solving hyperbolic systems with nonconservative products. Applications to shallow water systems. *Math. Comput.* 75 (2006), 1103–1134. DOI: [10.1090/S0025-5718-06-01851-5](https://doi.org/10.1090/S0025-5718-06-01851-5).
- [53] M.J. Castro, J.A. García-Rodríguez, J.M. González-Vida, C. Parés, and M.E. Vázquez-Cendón. Numerical simulation of two-layer shallow water flows through channels with irregular geometry. *J. Comput. Phys.* 195 (2004), 202–235. DOI: <https://doi.org/10.1016/j.jcp.2003.08.035>.
- [54] M.J. Castro, J.A. García, J.M. González, and C. Parés. A parallel 2D finite volume scheme for solving systems of balance laws with nonconservative products: Application to shallow flows. *Comput. Methods in Appl. Mech. Eng.* 195.19–22 (2006), 2788–2815. DOI: <https://doi.org/10.1016/j.cma.2005.07.007>.
- [55] M.J. Castro, A. Kurganov, and T. Morales de Luna. Path-conservative Central-upwind schemes for nonconservative hyperbolic systems. *ESAIM: Math. Model. Numer. Anal.* 53 (2019), 959–985. DOI: <https://doi.org/10.1051/m2an/2018077>.
- [56] M.J. Castro, P.G. LeFloch, M.L. Muñoz-Ruiz, and C. Parés. Why many theories of shock waves are necessary: convergence error in formally path-consistent schemes. *J. Comput. Phys.* 227 (2008), 8107–8129. DOI: <https://doi.org/10.1016/j.jcp.2008.05.012>.
- [57] M.J. Castro, T. Morales de Luna, and C. Parés. Chapter 6–Well–Balanced Schemes and Path–Conservative Numerical Methods. In: *Handbook of Numerical Methods for Hyperbolic Problems*. Ed. by Rémi Abgrall and Chi–Wang Shu. Vol. 18. Handbook of Numerical Analysis. Elsevier, 2017, 131–175. DOI: <https://doi.org/10.1016/bs.hna.2016.10.002>. URL: <http://www.sciencedirect.com/science/article/pii/S1570865916300333>.
- [58] M.J. Castro, S. Ortega, and C. Parés. Well-balanced methods for the shallow water equations in spherical coordinates. *Comput. & Fluids* 157 (2017), 196–207. DOI: <https://doi.org/10.1016/j.compfluid.2017.08.035>.

- [59] M.J. Castro, A. Pardo, C. Parés, and E.F. Toro. On some fast well-balanced first order solvers for nonconservative systems. *Math. Comput.* 79 (2010), 1427–1472. DOI: [10.1090/S0025-5718-09-02317-5](https://doi.org/10.1090/S0025-5718-09-02317-5).
- [60] M.J. Castro, C. Parés, G. Puppo, and G. Russo. Central schemes for nonconservative hyperbolic systems. *SIAM J. Sci. Comput.* 34 (2012), 523–558. DOI: <https://doi.org/10.1137/110828873>.
- [61] J.J. Cauret, J.F. Colombeau, and A.Y. Le Roux. Discontinuous generalized solutions of nonlinear nonconservative hyperbolic equations. *J. of Math. Analysis and Appl.* 139(2) (1989), 552–573. DOI: [https://doi.org/10.1016/0022-247X\(89\)90129-7](https://doi.org/10.1016/0022-247X(89)90129-7).
- [62] C. Chalons. Path-conservative in-cell discontinuous reconstruction schemes for nonconservative hyperbolic systems. *To appear in Commun. Math. Sci.* (2020). hal-02263335. URL: <https://hal.archives-ouvertes.fr/hal-02263335/document>.
- [63] C. Chalons and F. Coquel. A new comment on the computation of non-conservative products using Roe-type path conservative schemes. *J. Comput. Phys.* 335 (2017), 592–604. DOI: <https://doi.org/10.1016/j.jcp.2017.01.016>.
- [64] P. Chandrashekar, B. Nkonga, A.K. Meena, and A. Bhole. A path conservative finite volume method for a shear shallow water model. *J. Comput. Phys.* (2019). To appear. URL: <https://hal.inria.fr/hal-02294026v2>.
- [65] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen differenzgleichungen der mathematischen physiks. *Mathematische Annalen* 100(1) (1928), 32–74. DOI: <https://doi.org/10.1007/BF01448839>.
- [66] I. Cravero and M. Semplice. On the accuracy of WENO and CWENO reconstructions of third order on nonuniform meshes. *J. Sci. Comput.* 67(3) (2016), 1219–1246. DOI: <https://doi.org/10.1007/s10915-015-0123-3>.
- [67] H. Deconinck and M. Richiuto. Residual distribution schemes: Foundations and analysis. In: *Encyclopedia of Comput. Mech.* Wiley Online Library. 2007. DOI: <https://doi.org/10.1002/0470091355.ecm054>.
- [68] A. Dedner, F. Kemm, D. Kröner, C.D. Munz, T. Schnitzer, and M. Wesenberg. Hyperbolic divergence cleaning for the MHD equations. *J. Comput. Phys.* 175 (2002), 645–673. DOI: <https://doi.org/10.1006/jcph.2001.6961>.
- [69] A. Dedner, C. Rhode, and M. Wesenberg. A new approach to divergence cleaning in magnetohydrodynamic simulations. In: *Hyperbolic Problems: Theory, Numerics, Applications*. Ed. by T.Y. Hou and E. Tadmor. Berlin; Heidelberg: Springer-Verlag, 2003, 509–518. DOI: https://doi.org/10.1007/978-3-642-55711-8_47.
- [70] P. Degond, P.F. Peyrard, G. Russo, and Ph. Villedieu. Polynomial upwind schemes for hyperbolic systems. *C. R. Acad. Sci. Paris Sér. I, t. 328* (1999), 479–483. DOI: [https://doi.org/10.1016/S0764-4442\(99\)80194-3](https://doi.org/10.1016/S0764-4442(99)80194-3).
- [71] I.A. Demirdžić. A finite volume method for computation of fluid flow in complex geometries. PhD Thesis. Imperial College London (University of London), 1982. URL: <http://hdl.handle.net/10044/1/7597>.



- [72] V. Desveaux, M. Zenk, C. Berthon, and C. Klingenberg. A well-balanced scheme to capture non-explicit steady states in the Euler equations with gravity. *Intern. J. Num. Methods in Fluids* 81(2) (2016), 104–127. DOI: <https://doi.org/10.1002/flid.4177>.
- [73] V. Desveaux, M. Zenk, C. Berthon, and C. Klingenberg. Well-balanced schemes to capture non-explicit steady states: Ripa model. *Math. of Comput.* 85(300) (2016), 1571–1602. DOI: <https://doi.org/10.1090/mcom/3069>.
- [74] M. Dudzinski and M. Lukáčová-Medvid'ová. Well-balanced bicharacteristic-based scheme for multilayer shallow water flows including wet/dry fronts. *J. Comput. Phys.* 235 (2013), 82–113. DOI: <https://doi.org/10.1016/j.jcp.2012.10.037>.
- [75] M. Dumbser and D.S. Balsara. A new efficient formulation of the HLLEM Riemann solver for general conservative and non-conservative hyperbolic systems. *J. Comput. Phys.* 304 (2016), 275–319. DOI: <https://doi.org/10.1016/j.jcp.2015.10.014>.
- [76] M. Dumbser, D.S. Balsara, M. Tavelli, and F. Fambri. A divergence-free semi-implicit finite volume scheme for ideal, viscous, and resistive magnetohydrodynamics. *Int. J. Numer. Meth. Fluids* 89 (2019), 16–42. DOI: <https://doi.org/10.1002/flid.4681>.
- [77] M. Dumbser, D.S. Balsara, E.F. Toro, and C.D. Munz. A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured schemes. *J. Comput. Phys.* 227(18) (2008), 8209–8253. DOI: <https://doi.org/10.1016/j.jcp.2008.05.025>.
- [78] M. Dumbser, M.J. Castro, C. Parés, and E.F. Toro. ADER schemes on structured meshes for nonconservative hyperbolic systems: Applications to geophysical flows. *Computer & Fluids* 38 (2009), 1731–1748. DOI: <https://doi.org/10.1016/j.compfluid.2009.03.008>.
- [79] M. Dumbser, C. Enaux, and E.F. Toro. Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws. *J. Comput. Phys.* 227(8) (2008), 3971–4001. DOI: <https://doi.org/10.1016/j.jcp.2007.12.005>.
- [80] M. Dumbser, A. Hidalgo, M.J. Castro, C. Parés, and E.F. Toro. FORCE schemes on unstructured meshes II: non-conservative hyperbolic systems. *Comput. Methods Appl. Mech. Eng.* 199 (2010), 625–647. DOI: <https://doi.org/10.1016/j.cma.2009.10.016>.
- [81] M. Dumbser and M. Käser. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *J. Comput. Phys.* 221(2) (2007), 693–723. DOI: <https://doi.org/10.1016/j.jcp.2006.06.043>.
- [82] M. Dumbser, M. Käser, V.A. Titarev, and E.F. Toro. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *J. Comput. Phys.* 226(1) (2007), 204–243. DOI: <https://doi.org/10.1016/j.jcp.2007.04.004>.
- [83] M. Dumbser and R. Loubère. A simple robust and accurate a posteriori sub-cell finite volume limiter for the discontinuous Galerkin method on unstructured meshes. *J. Comput. Phys.* 319 (2016), 163–199. DOI: <https://doi.org/10.1016/j.jcp.2016.05.002>.

- [84] M. Dumbser and E.F. Toro. A simple extension of the Osher Riemann solver to non-conservative hyperbolic systems. *J. Sci. Comput.* 48 (2011), 70–88. DOI: <https://doi.org/10.1007/s10915-010-9400-3>.
- [85] M. Dumbser and E.F. Toro. On universal Osher-type schemes for general nonlinear hyperbolic conservation laws. *Commun. Comput. Phys.* 10 (2011), 635–671. DOI: <https://doi.org/10.4208/cicp.170610.021210a>.
- [86] M. Dumbser, O. Zanotti, R. Loubère, and S. Diot. A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *J. Comput. Phys.* 278 (2014), 47–75. DOI: <https://doi.org/10.1016/j.jcp.2014.08.009>.
- [87] *EDANYA web group*. <https://edanya.uma.es/hysea>.
- [88] B. Einfeldt. On Godunov-type methods for gas dynamics. *SIAM J. Numer. Anal.* 25 (1988), 294–318. DOI: <https://doi.org/10.1137/0725021>.
- [89] B. Einfeldt, P.L. Roe, C.D. Munz, and B. Sjogreen. On Godunov-type methods near low densities. *J. Comput. Phys.* 92 (1991), 273–295. DOI: [https://doi.org/10.1016/0021-9991\(91\)90211-3](https://doi.org/10.1016/0021-9991(91)90211-3).
- [90] C. Escalante, M. Dumbser, and M.J. Castro. An efficient hyperbolic relaxation system for dispersive non-hydrostatic water waves and its solution with high order discontinuous Galerkin schemes. *J. Comput. Phys.* 394 (2019), 385–416. DOI: <https://doi.org/10.1016/j.jcp.2019.05.035>.
- [91] C. Escalante, E.D. Fernández-Nieto, T. Morales, and M.J. Castro. An efficient two-layer non-hydrostatic approach for dispersive water waves. *J. Sci. Comput.* 79 (2019), 273–320. DOI: <https://doi.org/10.1007/s10915-018-0849-9>.
- [92] C. Escalante, T. Morales de Luna, and M.J. Castro. Non-hydrostatic pressure flows: GPU implementation using finite volume and finite difference scheme. *Appl. Math. Comput.* 338 (2018), 631–659. DOI: <https://doi.org/10.1016/j.amc.2018.06.035>.
- [93] C. Evans and J.F. Hawley. Simulation of magnetohydrodynamics flow: a constrained transport method. *Astrophys. J.* 332 (1988), 659–677. DOI: <https://doi.org/10.1086/166684>.
- [94] E.D. Fernández-Nieto, F. Bouchut, D. Bresch, M.J. Castro, and A. Mangeney. A new Savage-Hutter type model for submarine avalanches and generated tsunamis. *J. Comput. Phys.* 227(16) (2008), 7720–7754. DOI: <https://doi.org/10.1016/j.jcp.2008.04.039>.
- [95] E.D. Fernández-Nieto, J.M. Gallardo, and P. Vigneaux. Efficient numerical schemes for viscoplastic avalanches: Part 1: The 1D case. *J. Comput. Phys.* 264(1) (2014), 55–90. DOI: <https://doi.org/10.1016/j.jcp.2014.01.026>.
- [96] A.M. Ferreiro-Ferreiro, J.A. García-Rodríguez, J.G. López-Salas, C. Escalante, and M.J. Castro. Global optimization for data assimilation in landslide tsunami models. *J. Comput. Phys.* 403.109069 (2020). DOI: <https://doi.org/10.1016/j.jcp.2019.109069>.
- [97] M. Fey. Multidimensional upwinding. Part I. The method of transport for solving the Euler equations. *J. Comput. Phys.* 143 (1998), 159–180. DOI: <https://doi.org/10.1006/jcph.1998.5958>.



- [98] M. Fey. Multidimensional upwinding. Part II. Decomposition of the Euler equations into advection equations. *J. Comput. Phys.* 143 (1998), 181–199. DOI: <https://doi.org/10.1006/jcph.1998.5959>.
- [99] F.G. Fuchs, K. Karlsen, S. Mishra, and N.H. Risebro. Stable upwind schemes for the magnetic induction equation. *ESAIM: M2AN* 43(5) (2009), 825–852. DOI: <https://doi.org/10.1051/m2an/2009006>.
- [100] F.G. Fuchs, S. Mishra, and N.H. Risebro. Splitting based finite volume schemes for ideal MHD equations. *J. Comput. Phys.* 228 (2009), 641–660. DOI: <https://doi.org/10.1016/j.jcp.2008.09.027>.
- [101] J.M. Gallardo, S. Ortega, M. de la Asunción, and J.M. Mantas. Two-dimensional third-order polynomial reconstructions. Solving nonconservative hyperbolic systems using GPUs. *J. Sci. Comput.* 48 (2011), 141–163. DOI: <https://doi.org/10.1007/s10915-011-9470-x>.
- [102] J.M. Gallardo, C. Parés, and M.J. Castro. On a well-balanced high-order finite volume scheme for shallow water equations with topography and dry areas. *J. Comput. Phys.* 227 (2007), 574–601. DOI: <https://doi.org/10.1090/psapm/067.2/2605254>.
- [103] J.M. Gallardo, K.A. Schneider, and M.J. Castro. On a class of genuinely 2D incomplete Riemann solvers for hyperbolic systems. *Comp. and Math. Methods* (2019), 1–21. DOI: <https://doi.org/10.1002/cmm4.1074>.
- [104] J.M. Gallardo, K.A. Schneider, and M.J. Castro. On a class of two-dimensional incomplete Riemann solvers. *J. Comput. Phys.* 386 (2019), 541–547. DOI: <https://doi.org/10.1016/j.jcp.2019.02.034>.
- [105] H. Gilquing, J. Laurens, and C. Rosier. Multidimensional Riemann problems for linear hyperbolic systems: Part II. *Notes Numer. Fluid Mech.* 43 (1993), 284–290. DOI: <https://doi.org/10.1051/m2an/1996300505271>.
- [106] J. Glimm. Solution in the large for nonlinear hyperbolic systems of equations. *Commun. Pure Appl. Math.* 18 (1965), 697–715. DOI: <https://doi.org/10.1002/cpa.3160180408>.
- [107] E. Godlewski and P.A. Raviart. *Numerical approximation of hyperbolic systems of conservation laws*. Springer Science & Business Media, 2013. DOI: <https://doi.org/10.1007/978-1-4612-0713-9>.
- [108] S.K. Godunov. Finite difference methods for the computation of discontinuous solutions of the equations of fluid dynamics. *Math. USSR Sbornik* 47 (1959), 271–306. URL: <https://hal.archives-ouvertes.fr/hal-01620642>.
- [109] G.H. Golub and C. F. Van Loan. *Matrix Computations (3rd Ed.)* Johns Hopkins University Press, 1996.
- [110] L. Gosse. A two-dimensional version of the Godunov scheme for scalar balance laws. *SIAM J. Numer. Anal.* 52 (2014), 626–652. DOI: <https://doi.org/10.1137/130925906>.
- [111] S. Gottlieb and C.-W. Shu. Total variation diminishing Runge-Kutta schemes. *Math. Comp.* 67(221) (1998), 73–85. DOI: <https://doi.org/10.1090/S0025-5718-98-00913-2>.

- [112] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review* 43(1) (2001), 89–112. DOI: <https://doi.org/10.1137/S003614450036757X>.
- [113] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy. Uniformly high-order accurate essentially nonoscillatory schemes. III. *J. Comput. Phys.* 712 (1987), 231–303. DOI: https://doi.org/10.1007/978-3-642-60543-7_12.
- [114] A. Harten and J.M. Hyman. Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.* 50(2) (1983), 235–269. DOI: [https://doi.org/10.1016/0021-9991\(83\)90066-9](https://doi.org/10.1016/0021-9991(83)90066-9).
- [115] A. Harten, P. Lax, and B. van Leer. On upstream differencing and Godunov type schemes for hyperbolic conservation laws. *SIAM Rev.* 25(1) (1983), 35–61. DOI: <https://doi.org/10.1137/1025002>.
- [116] A. Hildebrand, S. Mishra, and C. Parés. Entropy-stable space-time DG schemes for non-conservative hyperbolic systems. *ESAIM: Math. Model. Numer. Anal.* 52(3) (2018), 995–1022. DOI: <https://doi.org/10.1051/m2an/2017056>.
- [117] T.Y. Hou and P.G. LeFloch. Why nonconservative schemes converge to wrong solutions. Error analysis. *Math. Comp.* 62 (1994), 497–530. DOI: <https://doi.org/10.2307/2153520>.
- [118] T. Kaczorek. *Polynomial and Rational matrices. Applications in Dynamical Systems Theory.* Communications and Control Engineering. Springer-Verlag London, 2007. DOI: <https://doi.org/10.1007/978-1-84628-605-6>.
- [119] C. Keney and A.J. Laub. Rational iterative methods for the matrix sign function. *SIAM J. Matrix Anal. Appl.* 12 (1991), 273–291. DOI: <https://doi.org/10.1137/0612020>.
- [120] A. Kurganov. Finite-volume schemes for shallow-water equations. *Acta Numerica* (2018), 289–351. DOI: <https://doi.org/10.1017/S0962492918000028>.
- [121] A. Kurganov, S. Noelle, and G. Petrova. Semi-discrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations. *SIAM J. Sci. Comput.* 23 (2000), 707–740. DOI: <https://doi.org/10.1137/S1064827500373413>.
- [122] A. Kurganov and G. Petrova. A second-order well-balanced positivity preserving central-upwind scheme for the Saint-Venant system. *Commun. Math. Sci.* 5(1) (2007), 133–160. DOI: <https://doi.org/10.4310/CMS.2007.v5.n1.a6>.
- [123] A. Kurganov and E. Tadmor. Solution of two-dimensional Riemann problems for gas dynamic without Riemann problem solvers. *Numer. Meth. Partial Diff. Equations* 18(5) (2002), 584–608. DOI: <https://doi.org/10.1002/num.10025>.
- [124] M. Lastra, M.J. Castro Díaz, C. Ureña, and M. de la Asunción. Efficient multilayer shallow-water simulation system based on GPUs. *Math. and Computers in Simul.* 148 (2018), 48–65. DOI: <https://doi.org/10.1016/j.matcom.2017.11.008>.
- [125] P. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Comm. Pure Appl. Math.* 7 (1954), 159–193. DOI: <https://doi.org/10.1002/cpa.3160070112>.

- [126] P. Lax. Hyperbolic systems of conservation laws. II. *Comm. Pure Appl. Math.* 10 (1957), 537–566. DOI: <https://doi.org/10.1002/cpa.3160100406>.
- [127] P. Lax. *Hyperbolic systems of conservation laws and the mathematical theory of shock waves*. CBMS Monograph, Society for Industrial and Applied Mathematics, 1973. DOI: <https://doi.org/10.1137/1.9781611970562>.
- [128] P. Lax and X. Liu. Solution of two-dimensional Riemann problems of gas dynamics by positive schemes. *SIAM J. Sci. Comput.* 19 (1998), 319–340. DOI: <https://doi.org/10.1137/S1064827595291819>.
- [129] P. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure Appl. Math.* 13(2) (1960), 217–237. DOI: <https://doi.org/10.1002/cpa.3160130205>.
- [130] P.G. LeFloch. Shock waves for nonlinear hyperbolic systems in nonconservative form. Institute for Math. and its Appl., Minneapolis, Preprint 503. 1989. URL: <http://hdl.handle.net/11299/5107>.
- [131] P.G. LeFloch. *Hyperbolic systems of conservation laws: the theory of classical and nonclassical shock waves*. Lecture Notes in Mathematics, ETH Zurich, Birkhäuser, 2002. DOI: <https://doi.org/10.1007/978-3-0348-8150-0>.
- [132] P.G. LeFloch and S. Mishra. Numerical methods with controlled dissipation for small-scale dependent shocks. *Acta Numer.* 23 (2014), 743–816. DOI: <https://doi.org/10.1017/S0962492914000099>.
- [133] R.J. LeVeque. High resolution Finite Volume methods on arbitrary grids via wave propagation. *J. Comput. Phys.* 78 (1988), 36–63. DOI: https://doi.org/10.1007/978-3-642-60543-7_20.
- [134] R.J. LeVeque. Wave propagation algorithms for multidimensional hyperbolic systems. *J. Comput. Phys.* 131 (1997), 327–353. DOI: <https://doi.org/10.1006/jcph.1996.5603>.
- [135] R.J. LeVeque. Balancing source terms and flux gradients in high-resolution Godunov methods: the quasi-steady wave-propagation algorithm. *J. Comput. Phys.* 146(1) (1998), 346–365. DOI: <https://doi.org/doi:10.1006/jcph.1998.6058>.
- [136] R.J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge University Press, 2002. DOI: <https://doi.org/10.1017/CB09780511791253>.
- [137] R.J. LeVeque, D.L. George, and M.J. Berger. Tsunami modelling with adaptively refined finite volume methods. *J. Comput. Phys.* 20 (2011), 211–289. DOI: <https://doi.org/10.1017/S0962492911000043>.
- [138] F. Li and C.-W. Shu. Locally divergence-free discontinuous Galerkin methods for MHD equations. *J. Sci. Comput.* 22–23 (2005), 413–442. DOI: <https://doi.org/10.1007/s10915-004-4146-4>.
- [139] R. Liska and B. Wendroff. Comparison of several difference schemes on 1d and 2d test problems for the Euler equations. *SIAM J. Sci. Comput.* 25 (2003), 995–1017. DOI: <https://doi.org/10.1137/S1064827502402120>.

- [140] M. Lukáčová-Medvid'ová, K.W. Morton, and G. Warnecke. Finite volume evolution Galerkin methods for Euler equations of gas dynamics. *Int. J. Numer. Methods in Fluids* 40 (3–4) (2002), 425–434. DOI: <https://doi.org/10.1002/flid.297>.
- [141] M. Lukáčová-Medvid'ová, S. Noelle, and M. Kraft. Well-balanced finite volume evolution Galerkin methods for the shallow water equations. *J. Comput. Phys.* 22(1) (2007), 122–147. DOI: <https://doi.org/10.1016/j.jcp.2006.06.015>.
- [142] J. Macías, M.J. Castro, S. Ortega, C. Escalante, and J.M. González-Vida. Performance benchmarking of Tsunami-HySEA model for NTHMP's inundation mapping activities. *Pue Appl. Geophys.* 174 (2017), 3147–3183. DOI: <https://doi.org/10.1007/s00024-017-1583-1>.
- [143] J.M. Mantas, M. de la Asunción, and M.J. Castro. An introduction to GPU computing for numerical simulation. In: *Numerical Simulations in Physics and Engineering*. Ed. by I. Higuera, T. Roldán, and J. Torrens. Vol. 9. SEMA SIMAI Springer Series. Springer, Cham, 2016. DOI: https://doi.org/10.1007/978-3-319-32146-2_5.
- [144] A. Marquina. Local piecewise hyperbolic reconstruction of numerical fluxes for nonlinear scalar conservation laws. *SIAM J. Sci. Comput.* 15(4) (1994), 892–915. DOI: <https://doi.org/10.1137/0915054>.
- [145] G. dal Maso, P.G. LeFloch, and F. Murat. Definition and weak stability of nonconservative products. *J. Math. Pures Appl.* 74 (1995), 483–548.
- [146] A. Mignone and P. Tzeferacos. A second-order unsplit Godunov scheme for cell-centered MHD: the CTU-GLM scheme. *J. Comput. Phys.* 229 (2010), 2117–2138. DOI: <https://doi.org/10.1016/j.jcp.2009.11.026>.
- [147] A. Mignone, P. Tzeferacos, and G. Bodo. High-order conservative finite difference GLM-MHD schemes for cell-centered MHD. *J. Comput. Phys.* 229 (2010), 5896–5920. DOI: <https://doi.org/10.1016/j.jcp.2010.04.013>.
- [148] S. Mishra and E. Tadmor. Constraint preserving schemes using potential-based fluxes. III. Genuinely multi-dimensional schemes for MHD equations. *ESAIM: Math. Model. Numer. Anal.* 46 (2012), 661–680. DOI: <https://doi.org/10.1051/m2an/20110509>.
- [149] T. Morales de Luna, M.J. Castro, and C. Parés. Relation between PVM schemes and simple Riemann solvers. *Numer. Math. Part. D. E.* 30 (2014), 1315–1341. DOI: <https://doi.org/10.1002/num.21871>.
- [150] T. Morales de Luna, M.J. Castro, C. Parés, and E.D. Fernández-Nieto. On a shallow water model for the simulation of turbidity currents. *Commun. Comput. Phys.* 6(4) (2009), 848–882. DOI: <https://doi.org/10.4208/cicp.2009.v6.p848>.
- [151] L.O. Müller, C. Parés, and E.F. Toro. Well-balanced high-order numerical schemes for one-dimensional blood flow in Vessels with varying mechanical properties. *J. Comput. Phys.* 242 (2013), 53–85. DOI: <https://doi.org/10.1016/j.jcp.2013.01.050>.
- [152] M.L. Muñoz and C. Parés. Godunov methods for nonconservative hyperbolic systems. *Math. Model. Numer. Anal.* 41 (2007), 169–185. DOI: <https://doi.org/10.1051/m2an:2007011>.

- [153] D.J. Newman. Rational approximation to $|x|$. *Mich. Math. J.* 11 (1964), 11–14. DOI: <https://doi.org/10.1307/mmj/1028999029>.
- [154] N.T. Nguyen and M. Dumbser. A path-conservative finite volume scheme for compressible multi-phase flows with surface tension. *Appl. Math. Comput.* 271 (2015), 959–978. DOI: <https://doi.org/10.1016/j.amc.2015.09.026>.
- [155] S. Noelle, N. Pankratz, G. Puppo, and J.R. Natvig. Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows. *J. Comput. Phys.* 213(2) (2006), 474–499. DOI: <https://doi.org/10.1016/j.jcp.2005.08.019>.
- [156] NVIDIA. *CUDA home page*. http://www.nvidia.com/object/cuda_home_new.html.
- [157] NVIDIA. *CUDA toolkit documentation*. <https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>.
- [158] S.A. Orszag and C.M. Tang. Small scale structure of two-dimensional magnetohydrodynamic turbulence. *J. Fluid Mech.* 90 (1979), 129–143. DOI: <https://doi.org/10.1017/S002211207900210X>.
- [159] S. Ortega. Esquemas de volúmenes finitos de alto orden: implementación en GPUs y aplicación a la simulación de flujos geofísicos. PhD Thesis. Universidad de Málaga, 2016. URL: <http://hdl.handle.net/10630/11946>.
- [160] C. Parés. Numerical methods for nonconservative hyperbolic systems: a theoretical framework. *SIAM J. Num. Anal.* 44 (2006), 300–321. DOI: <https://doi.org/10.1137/050628052>.
- [161] C. Parés and M.J. Castro. On the well-balance property of Roe’s method for nonconservative hyperbolic systems. Applications to shallow-water systems. *ESAIM: M2AN* 38 (2004), 821–852. DOI: <https://doi.org/10.1051/m2an:2004041>.
- [162] C. Parés and C. Parés-Pulido. Well-balanced high-order finite difference methods for systems of balance laws. *To appear* (2020). URL: <https://arxiv.org/abs/2001.10074>.
- [163] K.G. Powell. *An approximate Riemann solver for magnetohydrodynamics (that works in more than one space dimension)*. Tech. rep. 94-24. ICASE, Hampton, VA, 1994. URL: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19940028527.pdf>.
- [164] K.G. Powell, P.L. Roe, T.J. Linde, T.I. Gombosi, and D.L. De Zeeuw. A solution adaptive upwind scheme for ideal MHD. *J. Comput. Phys.* 154 (1999), 284–309. DOI: <https://doi.org/10.1006/jcph.1999.6299>.
- [165] R. Käppeli R. and S. Mishra. Well-balanced schemes for gravitationally stratified media. In: *Numerical Modeling of Space Plasma Flows ASTRONUM-2014*. Ed. by N.V. Pogorelov, E. Audit, and G.P. Zank. Vol. 498. Astronomical Society of the Pacific Conference Series. 2015, 210–215. URL: <http://hdl.handle.net/20.500.11850/96676>.
- [166] S. Rhebergen, O. Bokhove, and J.J.W. van der Vegt. Discontinuous Galerkin finite element methods for hyperbolic nonconservative partial differential equations. *J. Comput. Phys.* 227(3) (2008), 1887–1922. DOI: <https://doi.org/10.1016/j.jcp.2007.10.007>.

- [167] M. Ricchiuto. Contributions to the development of residual discretizations for hyperbolic conservation laws with application to shallow water flows. PhD Thesis. Université Sciences et Technologies - Bordeaux I, 2011. URL: <https://tel.archives-ouvertes.fr/tel-00651688v1>.
- [168] P.L. Roe. Approximate Riemann solvers, parameters vectors, and difference schemes. *J. Comput. Phys.* 43 (1981), 357–372. DOI: [https://doi.org/10.1016/0021-9991\(81\)90128-5](https://doi.org/10.1016/0021-9991(81)90128-5).
- [169] P.L. Roe. Multidimensional upwinding. In: *Handbook of Numerical Analysis, vol. 18*. Ed. by Elsevier. Amsterdam, 2017, 53–80. DOI: <https://doi.org/10.1016/bs.hna.2016.10.009>.
- [170] J.A. Rossmann. An unstaggered, high-resolution constrained transport method for magnetohydrodynamic flows. *SIAM J. Sci. Comput.* 28(5) (2006), 1766–1797. DOI: <https://doi.org/10.1137/050627022>.
- [171] V.V. Rusanov. Calculation of interaction of non-steady shock waves with obstacles. *USSR Comput. Mathematics and Math. Phys.* 1(2) (1962), 304–320. DOI: [https://doi.org/10.1016/0041-5553\(62\)90062-9](https://doi.org/10.1016/0041-5553(62)90062-9).
- [172] D. Ryu, F. Miniati, T.W. Jones, and A. Frank. A divergence-free upwind code for multidimensional magnetohydrodynamic flows. *The Astrophys. J.* 509 (1998), 244–255. DOI: <https://doi.org/10.1086/306481>.
- [173] M.L. Sætra. Shallow waters simulations on graphics hardware. PhD Thesis. University of Oslo, 2014. URL: <http://urn.nb.no/URN:NBN:no-45020>.
- [174] C. Sánchez-Linares, M. de la Asunción, M.J. Castro, S. Mishra, and J. Šukys. Multi-level Monte Carlo finite volume method for shallow water equations uncertain parameters applied to landslides-generated tsunamis. *Appl. Math. Model.* 39 (23-24) (2015), 7211–7226. DOI: <https://doi.org/10.1016/j.apm.2015.03.011>.
- [175] C. Sánchez-Linares, T. Morales de Luna, and M.J. Castro. A HLLC scheme for Ripa model. *Appl. Math. Comput.* 272, Part 2 (2016), 369–384. DOI: <https://doi.org/10.1016/j.amc.2015.05.137>.
- [176] J.B. Schijf and J.C. Schonfeld. Theoretical considerations on the motion of salt and fresh water. In: *Proceedings of the Minnesota International Hydraulics Convention, Joint meeting International Association Hydraulic Research and Hydraulic Division American Society Civil Engineer.* 1953, 321–333. URL: <http://resolver.tudelft.nl/uuid:5d1c2eb0-d51c-4b3c-ad77-a77513941c6c>.
- [177] K.A. Schneider and J.M. Gallardo. Efficient GPU implementation for high-order multidimensional incomplete Riemann solvers for hyperbolic nonconservative systems. Applications to shallow water systems with topography and dry areas. In preparation (2020).
- [178] K.A. Schneider, J.M. Gallardo, D.S. Balsara, B. Nkonga, and C. Parés. Multidimensional approximate Riemann solvers for hyperbolic nonconservative systems. Applications to shallow water systems. *J. Comput. Phys.* In preparation (2020).

- [179] A. Schroll and F. Svensson. A bi-hyperbolic finite volume method on quadrilateral meshes. *J. Sci. Comput.* 26(2) (2006), 237–260. DOI: <https://doi.org/10.1007/s10915-004-4927-9>.
- [180] C.W. Schulz-Rinne, J.P. Collins, and H.M. Glaz. Numerical solution of the Riemann problem for two-dimensional gas dynamics. *SIAM J. Sci. Comput.* 14 (1993), 1394–1414. DOI: <https://doi.org/10.1137/0914082>.
- [181] S. Serna. A characteristic-based nonconvex entropy-fix upwind scheme for the ideal magnetohydrodynamic equations. *J. Comput. Phys.* 228 (2009), 4232–4247. DOI: <https://doi.org/10.1016/j.jcp.2009.03.001>.
- [182] C.-W. Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In: *Advanced numerical approximation of nonlinear hyperbolic equations (Cetrao,1997)*, vol. 1697. *Lecture Notes in Math.* Ed. by Springer. Berlin, 1998, 325–432. DOI: <https://doi.org/10.1007/bfb0096355>.
- [183] C.-W. Shu and S. Osher. Efficient implementation of essentially nonoscillatory shock-capturing schemes. *J. Comput. Phys.* 77(2) (1988), 439–471. DOI: [https://doi.org/10.1016/0021-9991\(88\)90177-5](https://doi.org/10.1016/0021-9991(88)90177-5).
- [184] C.-W. Shu and S. Osher. Efficient implementation of essentially nonoscillatory shock-capturing schemes. II. *J. Comput. Phys.* 83(1) (1989), 32–78. DOI: [https://doi.org/10.1016/0021-9991\(89\)90222-2](https://doi.org/10.1016/0021-9991(89)90222-2).
- [185] P.K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.* 21 (1984), 995–1011. URL: <https://www.jstor.org/stable/2156939>.
- [186] H.-Z. Tang and K. Xu. A high-order gas-kinetic method for multidimensional ideal magnetohydrodynamics. *J. Comput. Phys.* 165 (2000), 69–88. DOI: <https://doi.org/10.1006/jcph.2000.6597>.
- [187] P. Tassi, O. Bokhove, and C. Vionnet. Space discontinuous Galerkin method for shallow water flows - kinetic and HLLC flux, and potential vorticity generation. *Adv. in Water Resources* 30 (2007), 998–1015. DOI: <https://doi.org/10.1016/j.advwatres.2006.09.003>.
- [188] V.A. Titarev and E.F. Toro. ADER schemes for three-dimensional non-linear hyperbolic systems. *J. Comput. Phys.* 204(2) (2005), 715–736. DOI: <https://doi.org/10.1016/j.jcp.2004.10.028>.
- [189] E.F. Toro. MUSTA: A multi-stage numerical flux. *Appl. Num. Math.* 56(10-11) (2006), 1464–1479. DOI: <https://doi.org/10.1016/j.apnum.2006.03.022>.
- [190] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. third ed. Springer, 2009. DOI: <https://doi.org/10.1007/b79761>.
- [191] E.F. Toro and S.J. Billet. Centered TVD schemes for hyperbolic conservation laws. *IMA J. Numer. Anal.* 20 (2000), 44–79. DOI: <https://doi.org/10.1093/imanum/20.1.47>.
- [192] E.F. Toro, M. Spruce, and W. Spears. Restoration of the contact surface in HLL-Riemann solver. *Shock Waves* 4 (1994), 25–34. DOI: <https://doi.org/10.1007/BF01414629>.

- [193] E.F. Toro and V.A. Titarev. MUSTA fluxes for systems of conservation laws. *J. Comput. Phys.* 216(2) (2006), 403–429. DOI: <https://doi.org/10.1016/j.jcp.2005.12.012>.
- [194] M. Torrilhon. Locally divergence-preserving upwind finite volume schemes for magneto-hydrodynamic equations. *SIAM J. Sci. Comput.* 26(4) (2005), 1166–1191. DOI: <https://doi.org/10.1137/S1064827503426401>.
- [195] M. Torrilhon. Krylov-Riemann solver for large hyperbolic systems of conservation laws. *SIAM J. Sci. Comput.* 34(4) (2012), A2072–A2091. DOI: <https://doi.org/10.1137/110840832>.
- [196] G. Tóth. The $\nabla \cdot B = 0$ constraint in shock-capturing magnetohydrodynamics codes. *J. Comput. Phys.* 161 (2000), 605–652. DOI: <https://doi.org/10.1006/jcph.2000.6519>.
- [197] I. Tóth. A weak formulation of Roe’s approximate Riemann solver. *J. Comput. Phys.* 102 (1992), 360–373. DOI: [https://doi.org/10.1016/0021-9991\(92\)90378-C](https://doi.org/10.1016/0021-9991(92)90378-C).
- [198] J.A. Trangenstein. *Numerical solution of hyperbolic partial differential equations*. Cambridge, 2009.
- [199] P. Váchal, R. Liska, and B. Wendroff. Fully two dimensional HLLEC Riemann solver. In: *Proceedings of Czech-Japanese Seminar in Applied Mathematics*. Prague, 2004, 195–206. URL: <http://geraldine.fjfi.cvut.cz>.
- [200] B. Van Leer. Towards the ultimate conservative difference scheme. V. A second order sequel to Godunov’s method. *Computational Physics* 32 (1979), 101–136. DOI: [https://doi.org/10.1016/0021-9991\(79\)90145-1](https://doi.org/10.1016/0021-9991(79)90145-1).
- [201] B. Van Leer. On the relation between the upwind-differencing schemes of Godunov, Engquist-Osher and Roe. *SIAM J. Sci. Statist. Comput.* 5(1) (1984), 1–20. DOI: <https://doi.org/10.1137/0905001>.
- [202] M.E. Vázquez-Cendón. *Solving hyperbolic equations with finite volume methods*. Springer, 2015. DOI: <https://doi.org/10.1007/978-3-319-14784-0>.
- [203] J. Vides, B. Nkonga, and E. Audit. A simple two-dimensional extension of the HLL Riemann solver for hyperbolic systems of conservation laws. *J. Comput. Phys.* 280 (2015), 643–675. DOI: <https://doi.org/10.1016/j.jcp.2014.10.013>.
- [204] A. I. Volpert. Spaces BV and quasilinear equations. *Mat. Sb. (N.S.)* 73 (115) (1967), 255–302. DOI: <https://doi.org/10.1070/SM1967v002n02ABEH002340>.
- [205] H. Wayland. Expansion of determinantal equations into polynomial form. *Quarterly Appl. Math.* 2(4) (1945), 277–306. URL: <https://www.jstor.org/stable/43633470>.
- [206] B. Wendroff. A two-dimensional HLLE Riemann solver and associated Godunov type difference scheme for gas dynamics. *Comput. Math. Appl.* 38 (1999), 175–185. DOI: [https://doi.org/10.1016/S0898-1221\(99\)00296-5](https://doi.org/10.1016/S0898-1221(99)00296-5).
- [207] N. Wintermeyer, A.R. Winters, G.J. Gassner, and D.A. Kopriva. An entropy stable nodal discontinuous Galerkin method for the two dimensional shallow water equations on unstructured curvilinear meshes with discontinuity bathymetry. *J. Comput. Phys.* 340 (2017), 200–242. DOI: <https://doi.org/10.1016/j.jcp.2017.03.036>.



- [208] H.-C. Wong, U.-H. Wong, X. Feng, and Z. Tang. Efficient magnetohydrodynamic simulations on graphics processing units with CUDA. *Computer Phys. Commun.* 182 (2011), 2132–2160. DOI: <https://doi.org/10.1016/j.cpc.2011.05.011>.
- [209] U.-H. Wong, H.-C. Wong, and Y. Ma. Global magnetohydrodynamic simulations on multiple GPUs. *Computer Phys. Commun.* 185 (2014), 144–152. DOI: <https://doi.org/10.1016/J.CPC.2013.08.027>.
- [210] Y. Xing and C.-W. Shu. High order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms. *J. Comput. Phys.* 214(2) (2006), 567–598. DOI: <https://doi.org/10.1016/j.jcp.2005.10.005>.
- [211] M. Yagi, K. Seki, and Y. Matsumoto. Development of a magnetohydrodynamic simulation code satisfying the solenoidal magnetic field condition. *Computer Phys. Commun.* 180 (2009), 1550–1557. DOI: <https://doi.org/10.1016/j.cpc.2009.04.010>.
- [212] A.L. Zachary, A. Malagoli, and P. Colella. A higher-order Godunov method for multi-dimensional magnetohydrodynamics. *SIAM J. Sci. Comput.* 15 (1994), 263–284. DOI: <https://doi.org/10.1137/0915019>.
- [213] T. Zhang and Y. Zheng. Conjecture on the structure of solutions of the Riemann problem for two-dimensional gas dynamics systems. *SIAM J. Math. Anal.* 21 (1990), 593–630. DOI: <https://doi.org/10.1137/0521032>.
- [214] Y. Zheng. *Systems of Conservation Laws: Two-dimensional Riemann problems*. Vol. 38. Progress in Nonlinear Differential Equations and their Applications. Birkhäuser, 2001. DOI: <https://doi.org/10.1007/978-1-4612-0141-0>.
- [215] U. Ziegler. A central-constrained transport scheme for ideal magnetohydrodynamics. *J. Comput. Phys.* 196 (2004), 393–416. DOI: <https://doi.org/10.1016/j.cpc.2003.11.003>.