Universidad Zaragoza
1542

Trabajo Fin de Máster con título

Desarrollo de esquemas de muy alto orden con aplicación a flujos geofísicos

presentado para la obtención del título en

Máster en modelización matemática, estadística y computación

Realizado por

Adrián Navas Montilla

y dirigido por

Dr. Javier Murillo Castarlenas

Ponente:

Dr. José Luis Gracia Lozano

Área de Mecánica de Fluidos

Departamento de Ciencia de Materiales y Fluidos

Escuela de Ingneiería y Arquitectura. Octubre, 2015. Zaragoza

AGRADECIMIENTOS

Quiero dedicar estas líneas para expresar mi agradecimiento a las personas que de un modo u otro me han ayudado y apoyado durante este año.

Me gustaría expresar un agradecimiento especial a Javier Murillo por su dedicación y seguimiento de mi trabajo durante ya más de un año, por estar ahí siempre que lo he necesitado y sobre todo por el apoyo y la motivación que siempre me ha ofrecido, así como por haber depositado en mi la confianza suficiente como para llevar a cabo estas labores de investigación.

También me gustaría agradecer a José Luis Gracia del Departamento de Matemática Aplicada de la Universidad de Zaragoza el haber aceptado la labor de ponencia de este Trabajo Fin de Máster, así como su interés y disponibilidad y a Pilar García por su confianza en mí y por haber hecho posible la presentacion de parte de este trabajo como una ponencia oral en la Universidad de Sheffield.

En Zaragoza, a 25 de Noviembre de 2015

Contents

For me, it is far better to grasp the Universe
as it really is than to persist in delusion,
however satisfying and reassuring.

C. SAGAN
The demon-haunted world, 1995.

Chapter 1

Introduction

Among the most important advances in science and technology during the 20^{th} century we find the ability to simulate complex physical systems and predict their spatial and temporal evolution. Such capabilities, as others, have been enhanced by a simultaneous development of computer science. In with respects to fluid mechanics, this progress has led to the appearance of a new discipline called *computational fluid dynamics* (CFD). This new discipline, propelled by aerospace industry, experimented a quick development over the past few decades and gave rise to the generation of simulation tools to be applied in a broad variety of fields, such as aerodynamics, weather science, geophysical science, space science, bioengineering, etc.

Many problems included within the scope of fluid mechanics are given by systems of partial differential equations, derived from the fundamental laws of physics. Analytical solutions for such systems cannot be found when dealing with complex geometries that appear in real problems of technological and scientific interest. However, it is possible to find approximate solutions provided by numerical resolution of the differential equations inside a computational domain, a discretization of the original domain. All those numerical tools and techniques fall within the scope of study of computational fluid dynamics.

Among the different types of systems of partial differential equations, only those with an hyperbolic nature will be considered in this work. Such systems are derived from conservation laws formulated for certain physical quantities such as mass, momentum or energy. Hyperbolic systems of conservation laws have been studied over the past few centuries, obtaining important results on the nature of their solutions. Among them, it is remarkable to mention the studies on the so-called Riemann Problem, a kind of initial value problem composed of a conservation equation together and a piecewise constant initial condition with a single discontinuity in the middle. Nowadays, numerical algorithms for the resolution of such problems, called *Riemann solvers*, are widespread and establish the basis of *Finite Volume Schemes*.

In this work, first order Godunov type Finite Volume Schemes and the corresponding Riemann solvers are first studied for the resolution of hyperbolic systems of conservation laws with source term, specially with those of geometric nature. The approach followed here is to use the Augmented solver presented in [1, 2]. Augmented solvers [3] are constructed to provide suitable explanations to the influence of the source terms in the numerical solution and the effect of the source terms in the stability region [4, 5]. They include an extra wave associated to the presence of the source terms in the approximate solution. In this family of Augmented solvers, the new wave provides two solutions at each side of the RP discontinuity. Based on the upwind discretization of the source terms in [6] and the Roe solver [7] defined for the homogeneous case, an Augmented solver in [1] was presented.

The preservation of high accuracy in both space and time when computing system of conservation laws with source terms has been a major step in the resolution of complex flows. The keystone for this important achievement is the Arbitrary Accuracy Derivatives Riemann Problem (ADER) approach for linear problems [8, 9] that allowed the construction of arbitrarily high-order accurate schemes for hyperbolic systems of conservation laws with source terms [10, 11, 12]. In contrast with Godunov's first order method [13], where the initial conditions of the Riemann Problem (RP) are piece-wise constant functions, in ADER schemes, the initial conditions are assumed to be smooth functions. This more general problem was termed Derivative Riemann Problem (DRP) where the initial conditions consist of polynomials of arbitrary degree.

Initial polynomial data for the DRP must be reconstructed by means of sophisticated conservative reconstruction procedures. Discontinuities may introduce spurious oscillations in the numerical solution and the choice of a proper reconstruction technique is decisive to avoid them. This issue was first addressed in the framework of finite differences, leading to the family of total-variation diminishing (TVD) schemes [14, 15, 16]. Later on, in the search of appropriate reconstruction techniques, the essentially non-oscillatory (ENO) method was proposed by Harten et al. [17]. Based on the definition of an smoothness indicator, the ENO method selects the departing information among different candidate stencils. Founded in the ENO approach, the WENO method was then developed by Liu et al. in [18], allowing a k-th order ENO reconstruction be transformed into an $(k+1)$-th order WENO reconstruction.

In this work, the WENO reconstruction method [18, 19, 20] and the sub-cell derivative WENO reconstruction procedure [21] is studied and implemented. In addition to this, a novel improvement for the traditional WENO reconstruction method is proposed. This enhanced procedure is termed WENO-PW method and addresses some convergence issues appearing in presence of critical points (points where derivatives vanish) when reconstructing smooth functions [22, 23]. Such problems in convergence have proved to be more noticeable when computing transport equations with stiff reactive terms, so that we test the performance of the WENO-PW reconstruction in combination with an ADER scheme for the resolution of the linear scalar equation with and without reactive term, in 1D and 2D. Convergence rate tests are also carried out and are presented in this text.

A novel ADER-type numerical scheme based on DRP Augmented solvers is also presented in this work. The proposed method will be called Augmented Roe ADER (AR-ADER) scheme. The performance of weak solutions for systems of equations involving discontinuous source terms is analyzed in the framework of flux-ADER numerical schemes. A novel DRP solver, that includes the presence of the source term at cell interfaces and solves the evolution equation of time derivatives, is presented. The AR-ADER scheme is presented for scalar non-linear equations first and it is next extended for systems of conservation laws. Numerical results are presented for the inviscid Burgers' equation with source term and for the Shallow Water Equations in 1D. In both cases, they evidence that the numerical scheme converges to the exact solution with the prescribed order of convergence. Moreover, when computing steady cases for the Shallow Water Equations, the numerical scheme provides the exact solution with independence of the grid size, since the discrete energy balance property is satisfied by the AR-ADER scheme. It is worth mentioning that the AR-ADER scheme has been recently published in [25].

The structure of this work is presented next. The first and second chapter are to serve as a theoretical framework for numerical schemes for systems of conservation laws, including the necessary definitions that establish the foundations for the development of Riemann solvers and numerical methods. In the third chapter, the Finite Volume Method is introduced, emphasizing its derivation from the integral form of the equations; Godunov's updating scheme is also presented and the Riemann Problem is defined. The fourth chapter is devoted to first order approximate Riemann solvers, recalling the Augmented solver presented in [1, 2] for scalar equations and systems of equations. In the fifth chapter, ADER numerical schemes are introduced and presented as the natural extension of Godunov's method; first, the DRP is introduced, then the WENO scheme is recalled and the WENO-PW is presented, after that, the ADER scheme for linear scalar equations is recalled and eventually the AR-ADER scheme is presented. In chapter sixth numerical tests are presented for the linear scalar equation in 1D and 2D, for Burgers' equation and for the Shallow Water Equations and finally, in chapter seventh, some conclusions of this work are summarized.

Chapter 2

Hyperbolic conservation laws in fluid mechanics

2.1 Introduction to conservation laws

A wide variety of physical events are described by systems of partial differential equations (PDEs) that correspond to conservation laws. In fluid mechanics, these conservation laws are commonly stated for mass, momentum and energy among others, and result naturally from the application of the fundamental laws of conservation of mass, Newton's second law and the law for the conservation of energy, respectively.

Let us consider a spatial domain $\Omega \subseteq \mathbb{R}^d$ where the fluid exists, with d the spatial dimension. Conservation laws normally state that the variation of the amount of a quantity inside a certain volume, called *control volume*, CV $\subseteq \Omega$, is due to the flux of that quantity across the surface delimiting the control volume, called *control surface*, CS $= \partial$CV, and to the contribution of a source, when present. For instance, let us consider the fixed control volume depicted in Figure 2.1 and a fluid with density $\rho = \rho(\mathbf{x}, t)$, where $\mathbf{x} \in \Omega \subseteq \mathbb{R}^3$ represents the spatial position in a Cartesian coordinate system and t the time. It is well known that due to the property of the conservation of mass, the variation of the mass, m, contained inside the CV can only be explained by a flux of mass, F_m, across the CS. If this flux, F_m, is defined as a leaving flux, then the following equation can be stated

$$\left.\frac{dm}{dt}\right|_{CV} = -F_m\,, \tag{2.1}$$

If defining

$$F_m = \iint_{CS} \rho\,\mathbf{v}\,\widehat{\mathbf{n}}\,dS\,, \tag{2.2}$$

then the variation of mass inside the control volume can be expressed as

$$\left.\frac{dm}{dt}\right|_{CV} = -\iint_{CS} \rho\,\mathbf{v}\,\widehat{\mathbf{n}}\,dS\,, \tag{2.3}$$

where the term on the right side of the equation stands for the net mass flow across the control surface as defined in (2.2), with $\mathbf{v} = \mathbf{v}(\mathbf{x}) \in \mathbb{R}^3$ the flow velocity and $\widehat{\mathbf{n}}$ the unitary vector normal to the control surface. The mass inside the control volume can be expressed as $m = \iiint_{CV} \rho dV$ and Equation (2.3) reads

$$\frac{d}{dt}\iiint_{CV} \rho dV = -\iint_{CS} \rho\,\mathbf{v}\,\widehat{\mathbf{n}}\,dS\,. \tag{2.4}$$

If considering the control volume not dependent on time and applying the divergence theorem to the surface integral of the flux, Equation (2.4) can be rewritten as

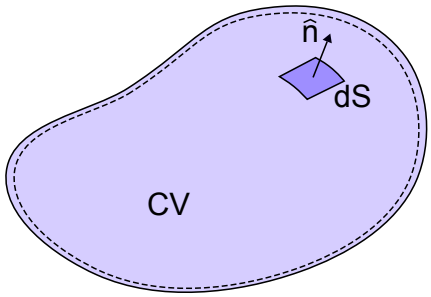

Figure 2.1: Fixed control volume (CV) containing a fluid of variable density $\rho(\mathbf{x}, t)$.

$$\iiint_{CV} \frac{\partial \rho}{\partial t} dV = - \iiint_{CV} \nabla \cdot (\rho \mathbf{v})\, dV \tag{2.5}$$

which leads to the equation for the conservation of mass in differential form

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0\,. \tag{2.6}$$

In this example, it is noticed that the variation of the conserved quantity, m, is only due to the mass flow entering and leaving the control volume, as outlined before. In other cases, it is possible that the variation of the conserved quantity is not only caused by the entering and leaving flow but also by the contribution of a certain *source term*. In more mathematical terms, the source term can be regarded as a function of the conserved quantities, spatial coordinates and time, that leads to a non-homogeneous PDE. In this work, conservation laws in presence of source terms will be studied.

2.2 Reynolds Transport Theorem and conservation laws

From Equation (2.4), it was noticed that the variation of mass inside a fixed volume was caused by the mass flux across its surface. Let us consider now that the surface is not still but moving at the same velocity than the flow, $\mathbf{v}_s = \mathbf{v}$. In this case, the integration volume is called *fluid volume* (V_f), also referred to as *closed system* since there is no mass flow across the boundaries. Equation (2.4) becomes

$$\left.\frac{dm}{dt}\right|_{V_f} = 0 \quad \Leftrightarrow \quad \frac{d}{dt} \iiint_{V_f} \rho dV = 0. \tag{2.7}$$

which states that the mass of the moving fluid parcel, that is, the fluid volume, is constant in time. Remark that some physical quantities such as mass, energy or momentum are conserved inside the fluid volume (closed system), as in this case, or equal to a certain source acting on the system, but this cannot be affirmed for an arbitrary CV, also referred to as *open system*. While fundamental physical laws have to be stated for the fluid volume, or system, it is worth mentioning that when facing a problem, integration inside a chosen CV is much simpler than using the fluid volume, since the CV can fit our geometry of interest.

It seems necessary to find a way to relate variations inside a CV to variations inside the fluid volume in order to state the conservation equations obeying certain physical laws in terms of variations inside the CV. For this purpose, the Reynolds Transport Theorem, hereafter RTT, was introduced, allowing to express the variation of a extensive quantity inside the fluid volume as the variation of this extensive quantity in a certain CV plus the flux of its associated intensive property across the CS. The utilization of this theorem supposes a great advantage since all calculations can be done over the selected CV, while at the same time the conservation of the physical quantity is stated inside the fluid volume by means of variations in the CV and flux through it.

Let $\mathbf{P} \in \mathbb{R}^d$ be any extensive property of the fluid (energy, mass, momentum...) with $d \in \mathbb{Z}^+$ and let $\mathbf{p} = d\mathbf{P}/dV$ be the intensive value of $\mathbf{P}$ per unit volume. Then, let us define a control volume, CV(t),

bounded by its control surface, CS(t), and a fluid volume, $V_f(t)$, bounded by, $S_f(t)$, all depending upon time and let

$$CV(0) = V_f(0)\,, \tag{2.8}$$

that is, the control volume and fluid volume coincide at $t = 0$. The total amount of $\mathbf{P}$ inside each volume is calculated as

$$\mathbf{P}_{CV}(t) = \iiint_{CV(t)} \mathbf{p} dV \qquad \mathbf{P}_{V_f}(t) = \iiint_{V_f(t)} \mathbf{p} dV \tag{2.9}$$

at time t. From conditions (2.8) and (2.9), it is straightforward to notice that the total amount of $\mathbf{P}$ in both volumes is the same at $t = 0$

$$\mathbf{P}_{CV}(0) = \mathbf{P}_{V_f}(0)\,. \tag{2.10}$$

On the other hand, the conservation of $\mathbf{P}$ inside the control volume can be expressed, as in Equation (2.4) without sources, as follows

$$\frac{d}{dt}\iiint_{CV} \mathbf{p} dV = -\iint_{CS} \mathbf{p}(\mathbf{v} - \mathbf{v}_s)\cdot\hat{\mathbf{n}} dS\,. \tag{2.11}$$

where $\mathbf{v}$ is the velocity of S_f, that is, the velocity of the fluid and $\mathbf{v}_s$ is the velocity of the CS that has to be accounted for since it is now moving. Using definition in (2.9) and the definition of derivative, Equation (2.11) can be expressed as

$$\lim_{\Delta t\to 0} \frac{\mathbf{P}_{CV}(\Delta t) - \mathbf{P}_{CV}(0)}{\Delta t} = -\iint_{CS} \mathbf{p}(\mathbf{v} - \mathbf{v}_s)\cdot\hat{\mathbf{n}} dS\,. \tag{2.12}$$

Considering that $\Delta t \to 0$, we can express $\mathbf{P}$ at $t = \Delta t$ as

$$\mathbf{P}_{CV}(\Delta t) = \mathbf{P}_{CV}(0) - \Delta t \iint_{CS} \mathbf{p}(\mathbf{v} - \mathbf{v}_s)\cdot\hat{\mathbf{n}} dS \tag{2.13}$$

The same is done for the quantity inside V_f after a time Δt, but in this case no outflow is present since the fluid volume follows the flow and therefore

$$\mathbf{P}_{V_f}(\Delta t) = \mathbf{P}_{V_f}(0) \tag{2.14}$$

Combination of Equations (2.10) and (2.14) allows to express (2.13) as

$$\mathbf{P}_{V_f}(\Delta t) = \mathbf{P}_{CV}(\Delta t) + \Delta t \iint_{CS} \mathbf{p}(\mathbf{v} - \mathbf{v}_s)\cdot\hat{\mathbf{n}} dS \tag{2.15}$$

Subtracting $\mathbf{P}_{V_f}(0)$ and $\mathbf{P}_{CV}(0)$ (which are the same) on the left and right sides of (2.15), respectively, and dividing by Δt, it yields

$$\frac{\mathbf{P}_{V_f}(\Delta t) - \mathbf{P}_{V_f}(0)}{\Delta t} = \frac{\mathbf{P}_{CV}(\Delta t) - \mathbf{P}_{CV}(0)}{\Delta t} + \iint_{CS} \mathbf{p}(\mathbf{v} - \mathbf{v}_s)\cdot\hat{\mathbf{n}} dS \tag{2.16}$$

Finally, the definition of derivative can be used again for (2.16) since $\Delta t \to 0$, leading to

$$\frac{d}{dt}\mathbf{P}_{V_f}(t) = \frac{d}{dt}\iiint_{CV} \mathbf{p} dV + \iint_{CS} \mathbf{p}(\mathbf{v} - \mathbf{v}_s)\cdot\hat{\mathbf{n}} dS \tag{2.17}$$

which represents the RTT. The term on the left hand side of the equation stands for the total variation of quantity $\mathbf{P}$ inside the fluid volume, V_f, that must be nil when the quantity is conserved (e.g. mass conservation) or, on the other hand, equal to a certain source (e.g. conservation of linear or angular

momentum, conservation of energy...). When existing, the sources will be considered acting on the control volume, since the system coincides with the control volume at the first moment.

It is remarkable to show that the RTT inside the fluid volume is given by Leibniz's rule for differentiation under the integral sign, that reads

$$\frac{d}{dt}\mathbf{P}_{V_f}(t) = \iiint_{V_f(t)} \frac{\partial \mathbf{p}}{\partial t} dV + \iint_{S_f(t)} \mathbf{p}(\mathbf{v}\cdot\hat{\mathbf{n}}) dS \tag{2.18}$$

For a better understanding of the RTT and the different frames of reference that can be used when analyzing a fluid flow, let us consider two different cases: in the first case, the observer is assumed to follow the fluid parcel as it moves along the streamlines, whereas in the second case, the observer is considered to be steady. It is worth mentioning that, generally, the quantity $\mathbf{p}$ can be defined as a property of the flow that depends upon the spatial position, $\mathbf{x}$, and time, t.

a) The first case corresponds to the so called *Lagrangian specification of the flow field* and considers that the property only depends on time, since the observer follows the fluid parcel as it moves along the streamlines. As outlined before, we know that $\mathbf{p} = \mathbf{p}(\mathbf{x}, t)$ but since in Lagrangian mechanics the position of a particle can be calculated as $\mathbf{x} = \mathbf{x}(\mathbf{x_0}, t)$, the quantity can be written just as a function of t and the initial point $\mathbf{x_0} = \mathbf{x}(0)$ as

$$\mathbf{p} = \mathbf{p}_l(\mathbf{x_0}, t) \tag{2.19}$$

where the subscript l stands for Lagrangian. In this case, the spatial coordinate at time t, $\mathbf{x}$, is considered a dependent variable that can be expressed in terms of $\mathbf{x_0}$ and t as

$$\mathbf{x}(t) = \mathbf{x_0} + \int_0^t \mathbf{v}(\mathbf{x}(\tau), \tau) d\tau \tag{2.20}$$

b) In the second case, the observer is considered to be still, with $\mathbf{v} = 0$. Unlike in the previous case, here magnitude $\mathbf{p}$ is parametrized as $\mathbf{p} = \mathbf{p}_e(\mathbf{x}, t)$ and can be regarded as a scalar or vector field. This approach is called *Eulerian specification of the flow field*. The relation between Lagrangian and Eulerian specifications of the flow field is given by

$$\mathbf{x} = \mathbf{x}\left(\mathbf{x_0}, t\right) \quad \longleftrightarrow \quad \mathbf{x_0} = \mathbf{x_0}\left(\mathbf{x}, t\right) \tag{2.21}$$

and analogously a relationship between derivatives can be found by applying the chain rule when differentiating $\mathbf{p}_e(\mathbf{x}, t)$ with respect to time, yielding

$$\begin{aligned} \frac{\partial}{\partial t}\mathbf{p}_l(\mathbf{x_0}, t) \equiv \underbrace{\frac{D}{Dt}\mathbf{p}_e(\mathbf{x}, t)}_{\text{Lagrangian der.}} &= \underbrace{\frac{\partial}{\partial t}\mathbf{p}_e(\mathbf{x}, t)}_{\text{local time der.}} + \underbrace{\frac{\partial x_1}{\partial t}\frac{\partial}{\partial x_1}\mathbf{p}_e(\mathbf{x}, t) + ... + \frac{\partial x_d}{\partial t}\frac{\partial}{\partial x_d}\mathbf{p}(\mathbf{x}, t)}_{\text{spatial advection}} = \\ &= \frac{\partial}{\partial t}\mathbf{p}_e(\mathbf{x}, t) + \mathbf{v}\cdot\nabla\mathbf{p}_e(\mathbf{x}, t) \end{aligned} \tag{2.22}$$

The previous expression is called material or susbtantial derivative of $\mathbf{p}$ and leads to the definition of operator

Definition 1. *(Material derivative operator). Operator*

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{v}\cdot\nabla\,, \tag{2.23}$$

with ∇ the Del operator with respect to the spatial coordinates and $\mathbf{v}$ the velocity field, allows to calculate the total variation of a certain quantity as its variation in time plus its variation produced by its advection under the velocity field.

The Eulerian specification of the flow field is the most common approach in fluid mechanics.

When addressing the resolution of a problem in fluid mechanics, two options are possible. The first choice would be to use the integral formulation in order to find approximate solutions at certain points or interfaces of the problem. The second choice, which this work is devoted to, would be to obtain the equivalent differential formulation from (2.17) or (2.18) and solve the PDEs inside the spatial domain by means of computational methods. The general procedure for the derivation of the differential form departing from the RTT formulation is next presented.

The Gauss-Strogadsky theorem can be applied to Equation (2.18) and considering the volume of infinitesimal size with $\mathbf{p}$ uniform inside it, the differential form of (2.18) is obtained

$$\frac{d\mathbf{p}}{dt} = \frac{\partial}{\partial t}\mathbf{p} + \nabla\left(\mathbf{p}\mathbf{v}\right) \tag{2.24}$$

Using (2.23), it can be rewritten in terms of the material derivative of $\mathbf{p}$ as

$$\frac{d\mathbf{p}}{dt} = \frac{D}{Dt}\mathbf{p} + (\mathbf{p}\cdot\nabla)\mathbf{v} \tag{2.25}$$

For instance, if considering the example of mass conservation equation in (2.6), it is noticeable that it can be rewritten in the form of (2.25), yielding

$$\frac{D\rho}{Dt} + \rho\nabla\mathbf{v} = 0 \tag{2.26}$$

2.3 Conservation laws: general formulation and hyperbolicity

When applying the RTT to a specific problem, the derivation of its equivalent differential form is straightforward, as it was done in (2.5)-(2.6) and more generally in (2.24), obtaining a set of PDEs that could be analytically or numerically solved. Conservation laws described in the previous section can be expressed in their divergence form as

$$\frac{\partial\mathbf{U}}{\partial t} + \nabla\cdot\mathbf{E}(\mathbf{U}) = \mathbf{S} \tag{2.27}$$

where $\mathbf{U} = \mathbf{U}(\mathbf{x}, t) \in \mathbb{R}^n$ is the vector of conserved variables with $\mathbf{x} \in \Omega \subseteq \mathbb{R}^d$, ∇ is the Del operator with respect to the spatial coordinates, $\mathbf{E}(\mathbf{U}) : \mathbb{R}^n \longrightarrow \mathbb{R}^{n\times d}$ is the matrix of fluxes, a nonlinear mapping of the conserved variables given by the physical flux and $\mathbf{S} \in \mathbb{R}^n$ is the vector of sources, yet to be defined. Normally, this vector of sources is of the form $\mathbf{S} = \mathbf{S}(\mathbf{U}, \mathbf{x}, t)$.

System in (2.27) can also be expressed as

$$\frac{\partial\mathbf{U}}{\partial t} + \sum_{j=1}^{d}\frac{\partial\mathbf{E}_j(\mathbf{U})}{\partial x_j} = \mathbf{S} \tag{2.28}$$

where $\mathbf{E}_j(\mathbf{U})$ represents the flux in the i-th spatial direction. It is possible to apply the *chain rule* to derivatives in (2.28) yielding

$$\frac{\partial\mathbf{U}}{\partial t} + \sum_{j=1}^{d}\mathbf{J}_j(\mathbf{U})\frac{\partial\mathbf{U}}{\partial x_j} = \mathbf{S} \tag{2.29}$$

with $\mathbf{J}_j(\mathbf{U})$ the Jacobian matrix of $\mathbf{E}_j(\mathbf{U})$, defined as

$$\mathbf{J}_j(\mathbf{U}) = \frac{\partial\mathbf{E}_j(\mathbf{U})}{\partial\mathbf{U}} \tag{2.30}$$

Definition 2. *(Hyperbolic system). The system in (2.27) is said to be hyperbolic if the matrix* $\mathcal{J}(\mathbf{k}) \in \mathbb{R}^{n\times n}$ *defined as*

$$\mathcal{J}(\mathbf{k}) = \sum_{j=1}^{d} k_j \mathbf{J}_j(\mathbf{U}) \,, \tag{2.31}$$

is diagonalizable with real eigenvalues for all $\mathbf{k} \in \mathbb{R}^d$ *and for all* $\mathbf{U} \in C$ *with* $C \subseteq \mathbb{R}^n$ *the subset of physically relevant values of* $\mathbf{U}$*. If the* n *eigenvalues are distinct, then the system is said to be strictly hyperbolic [30].*

Definition 3. *(Eliptic and parabolic systems). The system in (2.27) is said to be eliptic if none of the eigenvectors of* $\mathcal{J}(\mathbf{k}) \in \mathbb{R}^{n\times n}$ *is real. It is said to be parabolic if all eigenvectors are real and identical.*

2.3.1 Integral form of conservation laws

For the derivation of the integral form of (2.27), it is sufficient to integrate the equation in the domain $\mathcal{Q} = \Omega \times [0, \Delta t]$, with $\Omega \subseteq \mathbb{R}^d$ and $\mathbf{x} \in \Omega$, as

$$\int_0^{\Delta t} \int_\Omega \left(\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{E}(\mathbf{U}) \right) d\Omega dt = \int_0^{\Delta t} \int_\Omega \mathbf{S} d\Omega dt \tag{2.32}$$

and applying Gauss-Ostrogradsky theorem, the following expression results

$$\int_\Omega \mathbf{U}(\mathbf{x}, \Delta t) d\Omega = \int_\Omega \mathbf{U}(\mathbf{x}, 0) d\Omega - \int_0^{\Delta t} \int_{\partial\Omega} \mathbf{E}(\mathbf{U}(\mathbf{x}, t)) \hat{\mathbf{n}} d\Gamma dt + \int_0^{\Delta t} \int_\Omega \mathbf{S}(\mathbf{U}(\mathbf{x}, t), \mathbf{x}, t) d\Omega dt \tag{2.33}$$

that represents that the integral of the conserved quantities at $t = \Delta t$ is equal to the integral of the conserved quantities at $t = 0$ minus the integral in time of the total leaving fluxes across the surface $\partial\Omega$, plus the contribution of the source terms. This result is of great importance when finding weak solutions for the equations and constructing finite volume numerical schemes and it will be explained in the following chapter.

2.4 Conservation laws in 1D

This work focuses on nonlinear systems of conservation laws in 1D that can be expressed as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \mathbf{S} \,. \tag{2.34}$$

where $\mathbf{U} = \mathbf{U}(x, t) \in \mathbb{R}^n$ is the vector of conserved variables with $x \in \Omega \subseteq \mathbb{R}$, $\mathbf{F}(\mathbf{U}) : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is the vector of fluxes and $\mathbf{S} \in \mathbb{R}^n$ the vector of sources.

It is possible to define a Jacobian matrix for the flux $\mathbf{F}(\mathbf{U})$ as

$$\mathbf{J}(\mathbf{U}) = \frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}} \tag{2.35}$$

that provides sufficient information for the hyperbolicity of (2.34) according to Definition 2. Making use of the chain rule, system in (2.34) is rewritten as

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{J}(\mathbf{U}) \frac{\partial \mathbf{U}}{\partial x} = \mathbf{S} \,. \tag{2.36}$$

In the case when $\mathbf{F} = \mathbf{F}(\mathbf{U}, x)$, the previous approach must be rewritten as

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{J}(\mathbf{U}) \frac{\partial \mathbf{U}}{\partial x} + \frac{\delta \mathbf{F}(\mathbf{U}, x)}{\delta x} = \mathbf{S} \,. \tag{2.37}$$

Assuming that the system is hyperbolic with $N_\lambda = n$ real eigenvalues

$$\lambda^1(\mathbf{U}) \leq \lambda^2(\mathbf{U}) \leq ... \leq \lambda^{N_\lambda}(\mathbf{U}) \tag{2.38}$$

and N_λ linearly independent eigenvectors

$$\mathbf{e}^1(\mathbf{U}),\, \mathbf{e}^2(\mathbf{U}),\, ...\,,\, \mathbf{e}^{N_\lambda}(\mathbf{U}) \tag{2.39}$$

it is possible define two matrices $\mathbf{P}(\mathbf{U}) = (\mathbf{e}^1(\mathbf{U}), \mathbf{e}^2(\mathbf{U}), ..., \mathbf{e}^{N_\lambda}(\mathbf{U}))$ and $\mathbf{P}^{-1}(\mathbf{U})$ with the property that they diagonalize the Jacobian $\mathbf{J}$ as

$$\mathbf{J}(\mathbf{U}) = \mathbf{P}(\mathbf{U})\mathbf{\Lambda}(\mathbf{U})\mathbf{P}^{-1}(\mathbf{U}) \tag{2.40}$$

with $\mathbf{\Lambda}(\mathbf{U}) = \text{diag}\left(\lambda^1(\mathbf{U}), ..., \lambda^{N_\lambda}(\mathbf{U})\right)$ a diagonal matrix composed by the eigenvalues of the Jacobian.

Each eigenvalue $\lambda^m(\mathbf{U})$, or eigenvector $\mathbf{e}^m(\mathbf{U})$ equivalently, for $m = 1, ..., N_\lambda$ defines a *characteristic field* associated to it. The properties of the characteristic fields will provide useful information about the solution. Two types of characteristic fields are identified and defined next, according to [29].

Definition 4. *(Linearly degenerate field). A λ^m-characteristic field is said to be linearly degenerate when*

$$\nabla_u \lambda^m(\mathbf{U}) \cdot \mathbf{e}^m(\mathbf{U}) = 0\,, \quad \forall \mathbf{U} \in C\,, \tag{2.41}$$

with $C \subseteq \mathbb{R}^n$ and where ∇_u stands for the gradient with respect to the components of vector $\mathbf{U}$.

Definition 5. *(Genuinely nonlinear field). A λ^m-characteristic field is said to be genuinely nonlinear when*

$$\nabla_u \lambda^m(\mathbf{U}) \cdot \mathbf{e}^m(\mathbf{U}) \neq 0\,, \quad \forall \mathbf{U} \in C\,, \tag{2.42}$$

with $C \subseteq \mathbb{R}^n$ and where ∇_u stands for the gradient with respect to the components of vector $\mathbf{U}$.

2.4.1 Linear conservation laws in 1D

When the Jacobian matrix in (2.35) does not depend either upon $\mathbf{U}$ or x, it will be constant and the system in (2.34) is said to be *linear*. In this case, the flux function can be expressed as

$$\mathbf{F}(\mathbf{U}) = \mathbf{J}\mathbf{U} \tag{2.43}$$

leading to the following linear system of conservation laws

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{J}\frac{\partial \mathbf{U}}{\partial x} = \mathbf{S} \tag{2.44}$$

where $\mathbf{J}$ is a matrix of constant coefficients.

Considering that the linear problem presented in (2.44) is hyperbolic, the diagonalization of the Jacobian matrix can be expressed as

$$\mathbf{J} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1} \tag{2.45}$$

where $\mathbf{P} = (\mathbf{e}^1, \mathbf{e}^2, ..., \mathbf{e}^{N_\lambda})$ and $\mathbf{\Lambda} = \text{diag}(\lambda^1, \lambda^2, ..., \lambda^{N_\lambda})$ are constant matrices composed of the eigenvectors of $\mathbf{J}$

$$\mathbf{e}^1,\, \mathbf{e}^2,\, ...\,,\, \mathbf{e}^{N_\lambda} \tag{2.46}$$

and the eigenvalues of $\mathbf{J}$

$$\lambda^1 \leq \lambda^2 \leq ... \leq \lambda^{N_\lambda} \tag{2.47}$$

respectively. In the case when the system in (2.44) is strictly hyperbolic, eigenvalues in (2.47) are all distinct.

Now, it is possible to define a new set of variables, denoted by $\mathbf{W} = (w^1, w^2, ..., w^{N_\lambda})$ and called *characteristic variables*, by means of the transformation

$$\mathbf{W} = \mathbf{P}^{-1}\mathbf{U} \tag{2.48}$$

that represent the projection of the conserved variables onto the Jacobian's eigenvectors basis. Considering that $\mathbf{P}$ is constant, the following relations are stated

$$\frac{\partial \mathbf{W}}{\partial t} = \mathbf{P}^{-1}\frac{\partial \mathbf{U}}{\partial t} \qquad \frac{\partial \mathbf{W}}{\partial x} = \mathbf{P}^{-1}\frac{\partial \mathbf{U}}{\partial x} \tag{2.49}$$

Equivalently, a new set of variables, $\mathbf{B} = (\beta^1, \beta^2, ..., \beta^{N_\lambda})$, is defined for the source term as

$$\mathbf{B} = \mathbf{P}^{-1}\mathbf{S} \tag{2.50}$$

ensuring the same relations presented for the derivatives of $\mathbf{W}$ and $\mathbf{U}$ in (2.49). From (2.45) and (2.49), it is possible to rewrite the initial system in (2.44) as a decoupled system of PDEs as

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{\Lambda}\frac{\partial \mathbf{W}}{\partial x} = \mathbf{B}\,. \tag{2.51}$$

that corresponds to the expression of the original system of PDEs on the Jacobian's eigenvectors basis. System in (2.51) is composed of a set of independent linear scalar advection equations with source term, called *characteristic equations* and given by

$$\frac{\partial w^m}{\partial t} + \lambda^m \frac{\partial w^m}{\partial x} = \beta^m \quad \text{for } m = 1, ..., N_\lambda \tag{2.52}$$

where λ^m is the eigenvalue associated to the m-th wave and represents its propagation velocity, called *characteristic speed*.

Along the characteristic lines, depicted in Figure 2.2, the characteristic variables remain constant when the contribution of the source term is nil since

$$\frac{D}{Dt}(w^m) = 0 \quad \text{along } x = x_0 + \lambda^m t\,, \text{ for } m = 1, ..., N_\lambda \tag{2.53}$$

where $\frac{D}{Dt}$ represents the material derivative operator, defined in (2.23).

The solution for the original system in (2.44), $w^1(x,t), w^2(x,t), ..., w^m(x,t)$, can be obtained as a function of the solutions provided by the decoupled equations in (2.52). Regarding the previous results, the wave nature of the solution is noticed: the characteristic information will travel across the domain at different wave speeds given by λ^1, λ^2, ..., λ^m and the solution for the primitive variables will be obtained as a linear combination of the N_λ waves.

The initial condition for the decoupled system in Equation (2.52) is given by the projection of the initial condition $\mathring{\mathbf{U}} = \mathbf{U}(x, 0)$ onto the Jacobian's eigenvectors basis, as

$$\mathring{\mathbf{W}} = \mathbf{P}^{-1}\mathring{\mathbf{U}}\,. \tag{2.54}$$

At a given point (x, t), it is possible to express the vector of primitive variables $\mathbf{U}(x, t)$ as a linear combination of the Jacobian's eigenvectors using the relation $\mathbf{U} = \mathbf{P}\mathbf{W}$, as

$$\mathbf{U}(x,t) = \sum_{m=1}^{N_\lambda} w^m(x,t)\mathbf{e}^m\,, \tag{2.55}$$

where the scalar values $w^m(x, t)$ are the characteristic variables at the sought point and represent the strength of each wave.

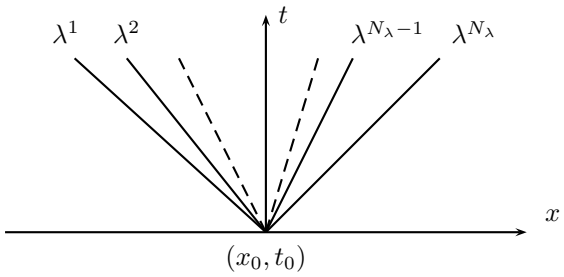

Figure 2.2: Characteristic lines passing through the point (x_0, t_0).

When considering that $\mathbf{S} = 0$, characteristic equations in (2.52) are reduced to linear scalar transport equations. Therefore, the initial values for the characteristic variables $\mathring{w}^m(x, 0)$ are simply advected at their corresponding wave speeds

$$w^m(x,t) = \mathring{w}^m(x - \lambda^m t) \quad \text{for } m = 1, ..., N_\lambda \,, \tag{2.56}$$

with no change in shape. Then, the solution can be expressed as the superposition of the N_λ waves that have been advected independently, as

$$\mathbf{U}(x,t) = \sum_{m=1}^{N_\lambda} \mathring{w}^m(x - \lambda^m t)\mathbf{e}^m \,. \tag{2.57}$$

It is worth saying that numerical methods for the resolution of hyperbolic systems developed in this work are based on linear approximate solutions, being the previous results the foundations for such algorithms.

Chapter 3

Finite volume numerical schemes for hyperbolic conservation laws

3.1 Introduction to Finite Volume schemes

When considering realistic problems modelled by hyperbolic conservation laws, the systems of equations are generally nonlinear. Moreover, initial conditions and source terms are normally complex enough to make impossible the utilization of analytical methods for the resolution of the system of equations. The most common approach to compute the solution is the discretization of the computational domain in volume cells where equations can be integrated leading to an algebraic system of equations instead of having the original PDEs. Inside each cell, the conserved quantities are integrated as well, leading to a finite set of cell averaged values that provides the approximate solution of the original system of PDEs inside the computational domain. This approach is the so-called *finite volume method*.

Let us consider the system of conservation laws in Equation (2.27) for d spatial dimensions to compose the following Initial Boundary Value Problem (IVBP)

$$\begin{cases} \text{PDEs:} & \dfrac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{E}(\mathbf{U}) = \mathbf{S} \\ \text{IC:} & \mathbf{U}(\mathbf{x},0) = \mathring{\mathbf{U}}(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega \\ \text{BC:} & \mathbf{U}(\mathbf{x},t) = \mathbf{U}_{\partial\Omega}(\mathbf{x},t) \quad \forall \mathbf{x} \in \partial\Omega \end{cases} \tag{3.1}$$

defined inside the domain $\Omega \times [0,T]$ with $\Omega \subseteq \mathbb{R}^d$ and $T \in \mathbb{R}^+$. As outlined before, the spatial domain is discretized in N volume cells, defined as $\Omega_i \subset \Omega$, such that $\Omega = \bigcup_{i=1}^{N} \Omega_i$. The volume contained in each of these cells is computed as

$$\vartheta_i = \int_{\Omega_i} d\Omega_i \qquad i = 1, ..., N \tag{3.2}$$

Inside each cell at time $t^n = n\Delta t$, the conserved quantities are defined as cell averages as

$$\mathbf{U}_i^n = \frac{1}{\vartheta_i} \int_{\Omega_i} \mathbf{U}(\mathbf{x}, t^n) d\Omega_i \qquad i = 1, ..., N\,. \tag{3.3}$$

provided the initial condition $\mathbf{U}(\mathbf{x},0) = \mathring{\mathbf{U}}(\mathbf{x})$.

Conservation law in (3.1) is integrated inside each cell Ω_i following (2.33) and using definition in (3.3), leading to

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{1}{\vartheta_i} \left(\int_0^{\Delta t} \int_{\partial\Omega_i} \mathbf{E}(\mathbf{U}(\mathbf{x},t)) \hat{\mathbf{n}} d\Gamma_i dt + \int_0^{\Delta t} \int_{\Omega_i} \mathbf{S}(\mathbf{U}(\mathbf{x},t), \mathbf{x}, t) d\Omega_i dt \right). \tag{3.4}$$

From (3.4), it is noticed that the cell average at t^{n+1}, denoted by $\mathbf{U}_i^{n+1}$, can be computed explicitly from the cell average at t^n plus a suitable approximation of the integral of the fluxes across $\partial\Omega_i$ and the contribution of the source term.

3.2 Godunov's method in 1D

When considering the particular case of one spatial dimension, the IVBP in (2.34) becomes

$$\begin{cases} \text{PDEs:} & \dfrac{\partial \mathbf{U}}{\partial t} + \dfrac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \mathbf{S} \\ \text{IC:} & \mathbf{U}(x,0) = \mathring{\mathbf{U}}(x) \\ \text{BC:} & \mathbf{U}(a,t) = \mathbf{U}_a(t) \quad \mathbf{U}(b,t) = \mathbf{U}_b(t) \end{cases} \tag{3.5}$$

defined inside the domain $[a,b]\times[0,T]$, with $\mathring{\mathbf{U}}(x)$ the initial condition and $\mathbf{U}_a(t)$ and $\mathbf{U}_b(t)$ the left and right boundary conditions. In this case, the computational grid is composed by N cells

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < ... < x_{N-\frac{1}{2}} < x_{N+\frac{1}{2}} = b \tag{3.6}$$

as shown in Figure 3.1, with cells defined as

$$\Omega_i = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] \qquad i = 1, ..., N \tag{3.7}$$

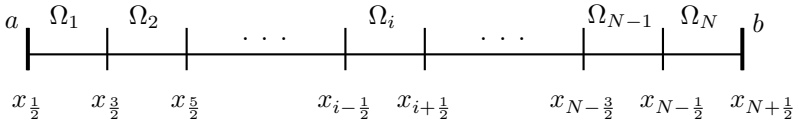

Figure 3.1: Mesh discretization

Cell sizes are derived from (3.2) and defined as

$$\Delta x_i = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} dx = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} \qquad i = 1, ..., N \tag{3.8}$$

Inside each cell, the conserved quantities are defined as cell averages as

$$\mathbf{U}_i^n = \frac{1}{\Delta x_i}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{U}(x,t^n)dx \qquad i = 1, ..., N\,. \tag{3.9}$$

at time t^n and the integration of (2.34) yields

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{1}{\Delta x_i}\left(\int_{t^n}^{t^{n+1}} \mathbf{F}(\mathbf{U}(x_{i+\frac{1}{2}},t))dt - \int_{t^n}^{t^{n+1}} \mathbf{F}(\mathbf{U}(x_{i-\frac{1}{2}},t))dt\right) + \int_{t^n}^{t^{n+1}} \frac{1}{\Delta x_i}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{S}(\mathbf{U}(x,t),x,t)dx\,dt \tag{3.10}$$

with $t^{n+1} = t^n + \Delta t$. If considering a explicit suitable approximation of the integral in time of the physical fluxes at cell boundaries, it is possible to define the numerical fluxes

$$\mathbf{F}^-_{i+\frac{1}{2}} \approx \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \mathbf{F}(\mathbf{U}(x_{i+\frac{1}{2}},t))dt \qquad \mathbf{F}^+_{i-\frac{1}{2}} \approx \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \mathbf{F}(\mathbf{U}(x_{i-\frac{1}{2}},t))dt \tag{3.11}$$

and express them in terms of the two contiguous averages at $t = t^n$ as $\mathbf{F}^-_{i+\frac{1}{2}} = \mathbf{F}^-_{i+\frac{1}{2}}(\mathbf{U}^n_i, \mathbf{U}^n_{i+1})$ and $\mathbf{F}^+_{i-\frac{1}{2}} = \mathbf{F}^+_{i-\frac{1}{2}}(\mathbf{U}^n_{i-1}, \mathbf{U}^n_i)$. Equivalently, the source term

$$\bar{\mathbf{S}}_i \approx \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{S}(\mathbf{U}(x,t),x,t)dx\,dt \tag{3.12}$$

can be expressed as $\bar{\mathbf{S}}_i = \bar{\mathbf{S}}_i(\mathbf{U}^n_i, x_i, t^n)$, making possible to rewrite (3.10) as the following explicit updating formula

$$\mathbf{U}^{n+1}_i = \mathbf{U}^n_i - \frac{\Delta t}{\Delta x_i}\left(\mathbf{F}^-_{i+\frac{1}{2}} - \mathbf{F}^+_{i-\frac{1}{2}}\right) + \frac{\Delta t}{\Delta x_i}\bar{\mathbf{S}}_i \tag{3.13}$$

that represents the Godunov's numerical scheme.

Remark that depending on the nature of the source term, a centered integration of the source term as used in (3.13) may not be adequate to preserve an exact balance between fluxes and sources and to achieve the equilibrium. Otherwise, it may be necessary to account for the value of its jump across the interface by means of including it in the approximate fluxes obtained in the resolution of the DRP [1, 25]. This is the case of the so-called *geometric source terms*, which are of the form

$$\mathbf{S}(\mathbf{U},x,t) = \mathbf{S}_s(\mathbf{U},x,t)\left(\frac{\partial}{\partial x}\mathbf{S}_g(\mathbf{U},x,t)\right) \tag{3.14}$$

with $\mathbf{S}_s(\mathbf{U},x,t)$ and $\mathbf{S}_g(\mathbf{U},x,t)$ two different functions with dependence upon $\mathbf{U}$, x and t, where $\mathbf{S}_g$ is the geometric part and may present discontinuities along the spatial domain. In this case, the integral of the source at the interface will be expressed as $\bar{\mathbf{S}}_{i+1/2} = \bar{\mathbf{S}}_{i+1/2}(\mathbf{U}^n_i, \mathbf{U}^n_{i+1}, x_i, x_{i+1}, t^n)$ and included in the numerical fluxes, leading to a modified updating formula

$$\mathbf{U}^{n+1}_i = \mathbf{U}^n_i - \frac{\Delta t}{\Delta x_i}\left(\mathbf{F}^-_{i+\frac{1}{2}} - \mathbf{F}^+_{i-\frac{1}{2}}\right) \tag{3.15}$$

where $\mathbf{F}^-_{i+\frac{1}{2}} = \mathbf{F}^-_{i+\frac{1}{2}}(\mathbf{U}^n_i, \mathbf{U}^n_{i+1}, \bar{\mathbf{S}}_{i+1/2})$ and $\mathbf{F}^+_{i-\frac{1}{2}} = \mathbf{F}^+_{i-\frac{1}{2}}(\mathbf{U}^n_{i-1}, \mathbf{U}^n_i, \bar{\mathbf{S}}_{i-1/2})$.

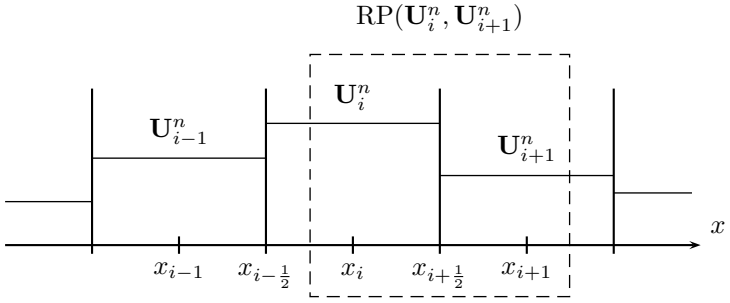

Figure 3.2: Neighbouring region of cell Ω_i and representation of piecewise defined data, showing RP at $x_{i+\frac{1}{2}}$ that will be referred to as RP($\mathbf{U}^n_i, \mathbf{U}^n_{i+1}$).

3.3 The Riemann Problem

At each interface, numerical fluxes in (3.11) can be computed by locally solving a initial value problem (IVP) composed of the system of PDEs and a initial condition given by the piecewise constant data at both sides of the interface, as depicted in Figure 3.2. For instance, the problem to be solved at cell interface $x_{i+\frac{1}{2}}$ is defined as

$$\begin{cases} \text{PDEs:} \quad \dfrac{\partial \mathbf{U}}{\partial t} + \dfrac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \mathbf{S} \\ \\ \text{IC:} \quad \mathbf{U}(x,t^n) = \begin{cases} \mathbf{U}_i^n & x < x_{i+\frac{1}{2}} \\ \mathbf{U}_{i+1}^n & x > x_{i+\frac{1}{2}} \end{cases} \end{cases} \tag{3.16}$$

inside the domain $[x_{i+1/2} - \frac{\Delta x}{2}, x_{i+1/2} + \frac{\Delta x}{2}] \times [t^n, t^n + \Delta t]$. Problem in (3.16) is called Riemann Problem, hereafter RP. At interface $x_{i+\frac{1}{2}}$, it will be referred to as RP($\mathbf{U}_i^n, \mathbf{U}_{i+1}^n$). For the sake of clarity, spatial and temporal variables will be redefined setting the reference for the spatial coordinate at $x_{i+\frac{1}{2}}$ to $x = 0$ and for the time t^n to $t = 0$, leading to

$$\begin{cases} \dfrac{\partial \mathbf{U}}{\partial t} + \dfrac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \mathbf{S} \\ \\ \mathbf{U}(x,0) = \begin{cases} \mathbf{U}_i & x < 0 \\ \mathbf{U}_{i+1} & x > 0 \end{cases} \end{cases} \tag{3.17}$$

inside the domain $[-\frac{\Delta x}{2}, \frac{\Delta x}{2}] \times [0, \Delta t]$. The similarity solution is denoted by $\mathbf{U}(x/t)$ and composed of $N_\lambda + 1$ constant states separated by N_λ waves [29]. When the system in (3.17) is linear, the solution can be easily constructed as a superposition of N_λ waves that advect the initial condition independently with the only possible change in shape due to the presence of the source term. When the system in (3.17) is non-linear, the waves may lead to shocks, rarefaction waves or contact waves and the solution is normally more complex.

For each λ^m wave defining a characteristic field, three different situations are possible. The left and right states of solution at each side of the discontinuity carried by λ^m wave are denoted by $\mathbf{U}_L^*$ and $\mathbf{U}_R^*$. Definitions 4 and 5 are used to determine the nature of each wave [29] as:

- **Shock wave:** If λ^m defines a *genuinely non-linear field* and the RH and entropy conditions apply

$$\mathbf{F}(\mathbf{U}_L^*) - \mathbf{F}(\mathbf{U}_R^*) = \mathcal{S}^m \left(\mathbf{U}_L^* - \mathbf{U}_R^*\right) \tag{3.18}$$

$$\lambda^m(\mathbf{U}_L^*) > \mathcal{S}^m > \lambda^m(\mathbf{U}_R^*) \tag{3.19}$$

  then left and right states $\mathbf{U}_L^*$ and $\mathbf{U}_R^*$ will be connected by a single jump discontinuity wave of speed $\mathcal{S}^m$ called shock wave.

- **Contact wave:** If λ^m defines a *linearly degenerate field* and the following conditions apply:
  - RH condition and parallel characteristic condition:

$$\mathbf{F}(\mathbf{U}_L^*) - \mathbf{F}(\mathbf{U}_R^*) = \mathcal{S}^m \left(\mathbf{U}_L^* - \mathbf{U}_R^*\right) \tag{3.20}$$

$$\lambda^m(\mathbf{U}_L^*) = \mathcal{S}^m = \lambda^m(\mathbf{U}_R^*) \tag{3.21}$$

  - Conservation of the Riemann Invariants across the wave if the contribution of the source term is nil.

  then left and right states $\mathbf{U}_L^*$ and $\mathbf{U}_R^*$ will be connected by a single jump discontinuity wave of speed $\mathcal{S}^m$ called contact wave.

- **Rarefaction wave:** If λ^m defines a *genuinely non-linear field* and the following conditions apply:
  - Divergence of characteristic

$$\lambda^m(\mathbf{U}_L^*) < \mathcal{S}^m < \lambda^m(\mathbf{U}_R^*) \tag{3.22}$$

  - Conservation of the Riemann Invariants across the wave if the contribution of the source term is nil.

then left and right states $\mathbf{U}_L^*$ and $\mathbf{U}_R^*$ will be connected by a smooth transition called rarefaction wave.

For this particular case of a RP, it can also be useful to derive its integral form. Integrating (3.17) over the control volume $[-x_L, x_R] \times [0, \Delta t]$

$$\int_{-x_L}^{x_R} \int_0^{\Delta t} \left(\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} - \mathbf{S} \right) dxdt = 0 \tag{3.23}$$

the following expression for the integral volume of $\mathbf{U}(x, \Delta t)$ is obtained

$$\int_{-x_L}^{x_R} \mathbf{U}(x, \Delta t)\, dx = x_R \mathbf{U}_{i+1} + x_L \mathbf{U}_i - (\delta \mathbf{F} - \bar{\mathbf{S}})_{i+\frac{1}{2}} \Delta t \tag{3.24}$$

with $\delta(\cdot)_{i+\frac{1}{2}} = (\cdot)_{i+1} - (\cdot)_i$, $\mathbf{F}_{i+1} = \mathbf{F}(\mathbf{U}_{i+1})$ and $\mathbf{F}_i = \mathbf{F}(\mathbf{U}_i)$ and the source term integrated as

$$\int_{-x_L}^{x_R} \int_0^{\Delta t} \mathbf{S}(\mathbf{U}_i, \mathbf{U}_{i+1}, t = 0)\, dxdt = \Delta t \bar{\mathbf{S}}_{i+\frac{1}{2}}\,. \tag{3.25}$$

3.4 Riemann solvers

In the previous section, it was outlined that the method of Godunov and its high order extensions requires to solve the RPs at the interfaces. The algorithm that provides the numerical solution for a certain RP is widely known in the literature as *Riemann solver*. When including the source term in the solution of the RP, as outlined above, the solver is referred to as *augmented (Riemann) solver*.

There are two kinds of Riemann solvers: *exact Riemann solvers* where the exact solution to the RP is normally found by iterative procedures and *approximate Riemann solvers* that provide an approximation of the intercell numerical fluxes. This approximation can be done in two ways, either constructing directly an approximation of the fluxes at the interface or finding an approximation to the state at the interface and evaluating then the physical flux. Approximate Riemann solvers are based on linearized solutions for the PDEs obtained when finding the value of the conserved quantities and fluxes at both sides of the interface using the integral form of the problem (weak solution). In the following chapters, both first order approximate Riemann solvers and high order approximate Riemann solvers are presented.

Chapter 4

First order approximate Riemann solvers

In the previous chapter, the necessity of computing the solution for the conserved quantities or fluxes at the interface was evidenced and the possibility of using approximate Riemann solvers was mentioned. Here, first order approximate solvers for scalar equations and systems of equations with source terms are briefly recalled, following [1]. These solutions are approximate and based on linearized weak solutions of the equations. The detailed procedure for the derivation of the solution for the scalar case and for a system of equations is presented in Appendix A, following [1].

As outlined before, the presence of geometric source terms is a major issue in the construction of numerical schemes since ensuring convergence to the physical solution is not a trivial task and may depend on the discretization of the source term. In [1], it was proposed to include the source term in the solution of the RP as an extra wave of velocity $\mathcal{S} = 0$, generating two different states at each side of the discontinuity that are connected by this stationary wave.

4.1 First order augmented solver for scalar equations

The scalar version of RP in (3.17) is considered in this section

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial f(u)}{\partial x} = s \\ u(x,0) = \begin{cases} u_i & x < 0 \\ u_{i+1} & x > 0 \end{cases} \end{cases} \tag{4.1}$$

where $u \in \mathbb{R}$ is the conserved variable, $s \in \mathbb{R}$ the source term and $f(u) : \mathbb{R} \to \mathbb{R}$ the physical flux, which is a nonlinear function of the conserved variable.

To derive a weak solution for (4.1), an approximate problem that uses a linear flux, $\hat{f}(\hat{u}) = \tilde{\lambda}_{i+1/2}\hat{u}$, is solved instead. Such problem is presented in Equation (A.5) and can be regarded as the linear approach to (4.1). The so-called *consistency condition*, which corresponds to enforce the equality between conserved variables for the original and approximate problems in their integral form, is used to derive an expression for the wave speed of the approximate problem. The relation presented in Equation (A.8) between the wave speed, jump of fluxes, and jump of conserved variables is obtained from the consistency condition and can be used in combination with the Rankine-Hugoniot (RH) condition to obtain the solution in the x-t plane.

The solution at the left and right sides of the interface, denoted by u_i^- and u_{i+1}^+ respectively, reads

$$u_i^- = \begin{cases} u_i & \text{if} \quad \widetilde{\lambda}_{i+\frac{1}{2}} > 0 \\ u_i + (\theta\delta u)_{i+\frac{1}{2}} & \text{if} \quad \widetilde{\lambda}_{i+\frac{1}{2}} < 0 \end{cases}$$
$$u_{i+1}^+ = \begin{cases} u_{i+1} - (\theta\delta u)_{i+\frac{1}{2}} & \text{if} \quad \widetilde{\lambda}_{i+\frac{1}{2}} > 0 \\ u_{i+1} & \text{if} \quad \widetilde{\lambda}_{i+\frac{1}{2}} < 0 \end{cases} \tag{4.2}$$

where $\delta u_{i+\frac{1}{2}}$ represents the jump of the conserved variable across the interface and $\theta_{i+\frac{1}{2}}$ accounts for the contribution of the source term, as defined in Equation (A.17).

Analogously, the fluxes at both sides of the interface are given by

$$f_i^- = \begin{cases} f_i & \text{if} \quad \widetilde{\lambda}_{i+\frac{1}{2}} > 0 \\ f_{i+1} - \bar{s}_{i+\frac{1}{2}} & \text{if} \quad \widetilde{\lambda}_{i+\frac{1}{2}} < 0 \end{cases}$$
$$f_{i+1}^+ = \begin{cases} f_i + \bar{s}_{i+\frac{1}{2}} & \text{if} \quad \widetilde{\lambda}_{i+\frac{1}{2}} > 0 \\ f_{i+1} & \text{if} \quad \widetilde{\lambda}_{i+\frac{1}{2}} < 0 \end{cases} \tag{4.3}$$

4.2 First order augmented solver for systems of N_λ waves

When moving to systems of an arbitrary number of waves, N_λ, different approaches can be taken to provide the numerical solution, such as the Roe solver [7] or the HLL(C)(S) solver [46]. In this work we will focus on the augmented version of the Roe solver, called *ARoe solver* [1], which includes an extra wave that accounts for the contribution of the source term.

Let us consider again RP in (3.17). As in the scalar case, an approximate linear problem, given by (A.19), is solved instead. This approximated problem has a flux

$$\hat{\mathbf{F}}(\hat{\mathbf{U}}) = \tilde{\mathbf{J}}_{i+1/2}\hat{\mathbf{U}}\,, \tag{4.4}$$

where $\hat{\mathbf{F}}$ and $\hat{\mathbf{U}}$ are the approximate fluxes and conserved quantities respectively and $\tilde{\mathbf{J}}_{i+1/2}$ is a constant coefficient matrix that, most of the time, can be regarded as the approximation of the Jacobian matrix of the physical flux at $x_{i+1/2}$. Relation

$$\delta\mathbf{F}_{i+1/2} = \widetilde{\mathbf{J}}_{i+1/2}\delta\mathbf{U}_{i+1/2} \tag{4.5}$$

results from the application of consistency condition to enforce equality between original and approximate solutions in the weak form.

Since the problem is considered hyperbolic in the whole computational domain, matrix $\tilde{\mathbf{J}}_{i+1/2}$ will be diagonalizable with real eigenvalues at every point. As a result, it is possible to project the problem onto the eigenvector basis of the matrix, leading to a decoupled set of scalar PDEs where each characteristic variable is advected at a particular speed, as shown in Section 2.4. Results obtained for the scalar case in Section 4.1 are used to find the numerical solution for each of such equations; these solutions are called characteristic solutions. Then, the characteristic solutions are combined to compose the solution in the original vector basis, as done in (2.57). This step is depicted in Figure A.3, showing that the solution consists of N_λ inner states separated by a stationary shock wave of celerity $\mathcal{S} = 0$ at $x_{i+1/2}$.

The solution on the left and right sides of the interface, $\mathbf{U}_i^-$ and $\mathbf{U}_{i+1}^+$ respectively, is given by

$$\begin{cases} \mathbf{U}_i^- = \mathbf{U}_i + \displaystyle\sum_{m_1=1}^{I} (\theta\alpha\widetilde{\mathbf{e}})_{i+\frac{1}{2}}^{m_1} \\ \mathbf{U}_{i+1}^+ = \mathbf{U}_{i+1} - \displaystyle\sum_{m_1=I+1}^{N_\lambda} (\theta\alpha\widetilde{\mathbf{e}})_{i+\frac{1}{2}}^{m_1} \end{cases} \tag{4.6}$$

where $\left\{\lambda^1_{i+1/2},...,\lambda^I_{i+1/2}\right\}$ is the set of left-moving (negative) waves (eigenvalues) and $\left\{\lambda^{I+1}_{i+1/2},...,\lambda^{N_\lambda}_{i+1/2}\right\}$ the right-moving (positive) waves (eigenvalues), $\widetilde{\mathbf{e}}^{m_1}_{i+\frac{1}{2}}$ the eigenvector associated to eigenvalue $\lambda^{m_1}_{i+1/2}$, $\alpha^{m_1}_{i+1/2}$ is the jump of the m_1 characteristic variable across the interface and $\theta^{m_1}_{i+1/2}$ is defined in (A.31). Analogously, the fluxes at both sides of the interface are given by

$$\begin{cases} \mathbf{F}^-_i = \mathbf{F}_i + \displaystyle\sum_{m_1=1}^{I} (\lambda\theta\alpha\widetilde{\mathbf{e}})^{m_1}_{i+\frac{1}{2}} \\ \mathbf{F}^+_{i+1} = \mathbf{F}_{i+1} - \displaystyle\sum_{m_1=I+1}^{N_\lambda} (\lambda\theta\alpha\widetilde{\mathbf{e}})^{m_1}_{i+\frac{1}{2}} \end{cases} \tag{4.7}$$

Chapter 5

High order Riemann solvers and ADER schemes

In this chapter, the theoretical framework for the construction of high order Goudunov-type numerical schemes and its corresponding approximate Riemann solvers is provided. The methodology followed here is to construct an ADER scheme, based on a high order extension of Godunov's method by means of a Taylor power series expansion in time. A generalization of the Riemann Problem for the ADER scheme is required. Such problem is called Derivative Riemann Problem (DRP) and will be presented first. When solving the DRP, spatial reconstruction of the conserved variables is required and therefore in the second section a non-oscillatory reconstruction procedure will be presented. The third section thoroughly outlines the construction of ADER schemes and the two different approaches to the problem. Then, in fourth and fifth sections an ADER-type numerical scheme, called TT-ADER scheme, is constructed for the resolution of the linear scalar transport equation, for 1D and 2D. Finally, in the sixth section a novel ADER scheme, named AR-ADER, and its corresponding approximate Riemann solver are presented.

5.1 Introduction: The Derivative Riemann Problem

In Section 3.3 the Riemann Problem was described as an IVP whose initial condition is given by two piecewise constant states. This classic RP may be regarded as a first order approach to a general Cauchy problem with a discontinuity at $x = 0$. A higher order approach to the Cauchy problem is given by the DRP, that is a IVP defined by a system of N_λ EDPs and a initial condition consisting of piecewise polynomial data (with K nontrivial derivatives) separated by a single discontinuity at $x = 0$ as

$$\begin{cases} \dfrac{\partial \mathbf{U}}{\partial t} + \dfrac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \mathbf{S} \\ \mathbf{U}(x,0) = \begin{cases} \mathbf{U}_i(x) & x < 0 \\ \mathbf{U}_{i+1}(x) & x > 0 \end{cases} \end{cases} \tag{5.1}$$

where the initial states $\mathbf{U}_i(x)$ and $\mathbf{U}_{i+1}(x)$ are smooth functions of distance x that can be defined using suitable reconstruction procedures at the initial time. Recall that x stands for the local spatial coordinate, centered at $x_{i+1/2}$. DRP in (5.1) is depicted in Figure 5.1 for the case when $N_\lambda = 2$.
For DRP in (5.1), it is possible to define the following values for vector $\mathbf{U}$ at the interface

$$\mathbf{U}^{(0)}_{i_R} = \lim_{x\to 0^-} \mathbf{U}_i(x) \qquad \mathbf{U}^{(0)}_{(i+1)_L} = \lim_{x\to 0^+} \mathbf{U}_{i+1}(x) \tag{5.2}$$

and for its derivatives

$$\mathbf{U}^{(k)}_{i_R} = \lim_{x\to 0^-} \frac{\partial^k}{\partial x^k}\mathbf{U}_i(x) \qquad \mathbf{U}^{(k)}_{(i+1)_L} = \lim_{x\to 0^+} \frac{\partial^k}{\partial x^k}\mathbf{U}_{i+1}(x) \tag{5.3}$$

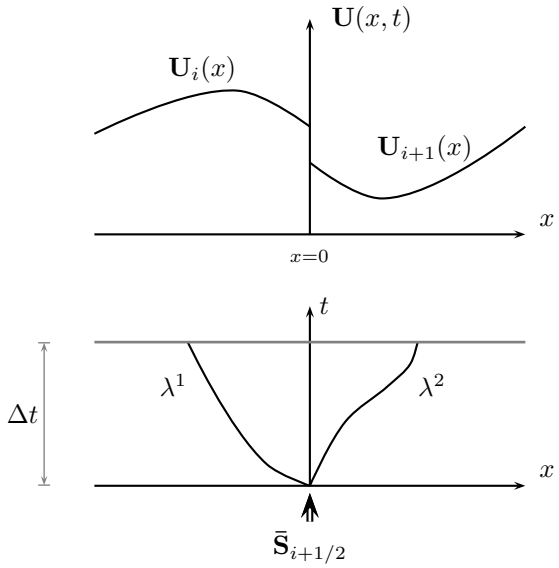

Figure 5.1: Graphical representation of the DRP_K showing the piecewise smooth states (upper figure) and wave velocities that depend upon time (lower figure).

at the initial time, with $k = 1, ..., K$.

Analogously, it is possible to define the following values for the physical fluxes $\mathbf{F}(\mathbf{U})$ at the interface

$$\mathbf{F}^{(0)}_{i_R} = \lim_{x\to 0^-} \mathbf{F}(\mathbf{U}_i(x)) \qquad \mathbf{F}^{(0)}_{(i+1)_L} = \lim_{x\to 0^+} \mathbf{F}(\mathbf{U}_{i+1}(x)) \tag{5.4}$$

and for their spatial derivatives

$$\mathbf{F}^{(k)}_{i_R} = \lim_{x\to 0^-} \frac{\partial^k}{\partial x^k}\mathbf{F}(\mathbf{U}_i(x)) \qquad \mathbf{F}^{(k)}_{(i+1)_L} = \lim_{x\to 0^+} \frac{\partial^k}{\partial x^k}\mathbf{F}(\mathbf{U}_{i+1}(x)) \tag{5.5}$$

at the initial time, with $k = 1, ..., K$.

The spatial reconstruction of the source term $\mathbf{S}(\mathbf{U}, x, t)$ will be denoted in the same way

$$\mathbf{S}^{(0)}_{i_R} = \lim_{x\to 0^-} \mathbf{S}(\mathbf{U}_i(x), x, 0) \qquad \mathbf{S}^{(0)}_{(i+1)_L} = \lim_{x\to 0^+} \mathbf{S}(\mathbf{U}_{i+1}(x), x, 0) \tag{5.6}$$

and also its derivatives

$$\mathbf{S}^{(k)}_{i_R} = \lim_{x\to 0^-} \frac{\partial^k}{\partial x^k}\mathbf{S}(\mathbf{U}_i(x), x, 0) \qquad \mathbf{S}^{(k)}_{(i+1)_L} = \lim_{x\to 0^+} \frac{\partial^k}{\partial x^k}\mathbf{S}(\mathbf{U}_{i+1}(x), x, 0) \tag{5.7}$$

at the initial time, with $k = 1, ..., K$.

The value of u at the center of each cell will be denoted as $\mathbf{U}^0_i = \mathbf{U}_i(x_i)$. Subscripts L and R are defined with reference to the cell center, as depicted in Figure 5.2. We will denote by DRP_K the Cauchy problem presented in (5.1), with K continuous non-trivial spatial derivatives.

High-order numerical methods of the ADER type require the solution at the interface position $x_{i+1/2}$ as a function of time t, allowing to compute the numerical fluxes and construct a numerical scheme of $K+1$-th order of accuracy in both space and time. Following [10, 11, 12] the solution will contain a leading term, provided by the DRP_0, equivalent to the classical piecewise constant data Riemann problem, associated with the first order Godunov scheme [13] and higher-order terms, associated with the k different RPs for the derivatives. It is worth saying that the Derivative Riemann Problem DRP_K can be decomposed in $K + 1$ RPs where conventional Riemann Solvers are of application.

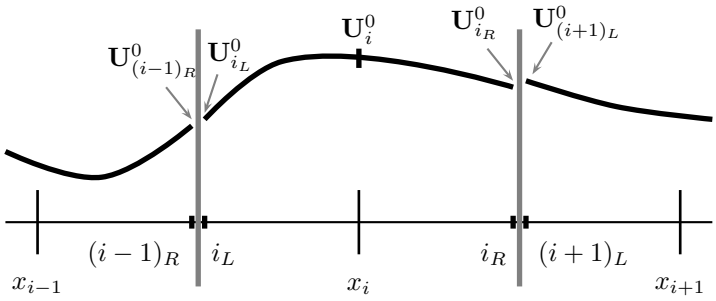

Figure 5.2: Mesh discretization

5.2 High order non-oscillatory reconstruction: traditional WENO and WENO-PW methods

The necessity of reconstruction procedures of very high order [44, 11, 12] for the conserved variables, fluxes and source terms was evidenced in the previous section. Discontinuities may introduce spurious oscillations in the numerical solution [45] and the choice of a proper reconstruction technique is decisive to avoid them. In this work, we decide to use the so-called weighted essentially non-oscillatory (WENO) reconstruction method, originally introduced in [18]. The WENO method uses a dynamic set of stencils where lower order polynomials are first constructed. Then, these lower order polynomials are combined either to create a higher order polynomial in smooth regions (optimal reconstruction) or an off-center reconstruction able to capture discontinuities in non-smooth regions. The definition of a smoothness indicator allows to distinguish between those two cases.

In this section, the traditional WENO reconstruction method, also referred to as WENO-JS, is briefly explained first, presenting then a novel modification of this method that addresses some convergence issues. More details concerning the WENO reconstruction procedure, sub-cell WENO derivative reconstruction procedure and their extension to 2D can be found in Appendices B, C, D and E.

The departing data for the WENO reconstruction procedure are the cell average values of a function $u(x)$ defined in a computational grid composed of N cells, with cells and cell sizes defined by

$$\Omega_i = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] \qquad \Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} \equiv \text{constant} \tag{5.8}$$

and cell averages of $u(x)$ are defined in the following way

$$\bar{u}_i = \frac{1}{\Delta x_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(\xi)\, d\xi, \quad i = 1, 2, ..., N \tag{5.9}$$

To construct a WENO reconstruction of degree $(2k-1)$ on the cell $\Omega_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ for the function $u(x)$, k different stencils linked to k cells are needed. These stencils are given by $S_r(i) = \{\Omega_{i-r}, ..., \Omega_{i+k-r-1}\}$ $(r = 0, ..., k-1)$, where r represents the number of cells on the left hand side of Ω_i. These stencils are used to generate a bigger stencil $\mathcal{T}(i) = \cup_{r=0}^{k-1} S_r(i) = \{\Omega_{i-k+1}, ..., \Omega_{i+k-1}\}$. The general procedure of the WENO reconstruction is summarized below:

a) *Definition of the optimal weights*

Following [20], there is a unique polynomial $p_r(x)$ defined in each stencil S_r, which is a k-th order approximation of the function $u(x)$ on the stencil $S_r(i)$ if this function is smooth inside it. The expression of $p_r(x)$ is expressed as a linear combination of the cell averages in the stencil. At $x_{i+\frac{1}{2}}$, the approximation of $u(x_{i+\frac{1}{2}})$ is given by

$$u_{i+\frac{1}{2}}^{(r)} = p_r(x_{i+\frac{1}{2}}) = \sum_{j=0}^{k-1} c_{rj}^{(k)} \bar{u}_{i-r+j} = u\left(x_{i+\frac{1}{2}}\right) + O\left(\Delta x^k\right) \tag{5.10}$$

where $c_{rj}^{(k)}$ are coefficients derived from the Lagrange interpolation formula. The same procedure can be used to obtain a polynomial $q(x)$, which is a $(2k-1)$-th order approximation of the function $u(x)$ on the big stencil $\mathcal{T}(i)$. At $x_{i+\frac{1}{2}}$, this approximation of $u(x_{i+\frac{1}{2}})$ is given by

$$u_{i+\frac{1}{2}} = q(x_{i+\frac{1}{2}}) = \sum_{j=1}^{2k-1} c_{k-1,j}^{(2k-1)} \bar{u}_{i-k+j} = u\left(x_{i+\frac{1}{2}}\right) + O\left(\Delta x^{2k-1}\right) \tag{5.11}$$

Note that in (5.11), the value of r is fixed, with $r = k-1$, as the big stencil $\mathcal{T}(i)$ is symmetric. The $(2k-1)$-th order approximation in (5.11) can also be expressed as a linear convex combination of the k-th order reconstructions provided by (5.10) as

$$u_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} \gamma_r u_{i+\frac{1}{2}}^{(r)} = u\left(x_{i+\frac{1}{2}}\right) + O\left(\Delta x^{2k-1}\right) \tag{5.12}$$

where γ_r are the optimal weights that can be easily computed relating $c_{k-1,j}^{(2k-1)}$ and $c_{rj}^{(k)}$ [20]. In the linear combination in (5.12) the optimal weights are calculated algebraically.

b) *Definition of the smoothness indicator: smoothness indicator for WENO-JS*

The so called smoothness indicator, β_r, which measure the smoothness of the initial data, is able to detect the presence of discontinuities. In the case of the traditional WENO reconstruction, called WENO-JS and proposed by Jiang and Shu [20], this indicator reads

$$\beta_r = \sum_{l=1}^{k-1} \int_{x_{i+\frac{1}{2}}}^{x_{i-\frac{1}{2}}} \Delta x^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial x^l}\right)^2 dx, \qquad r = 0, ..., k-1 \tag{5.13}$$

and represents the variations in u inside the small stencils.

c) *Definition of the WENO-JS weights*

Departing from the optimal weights, it is possible to define the WENO-JS weights, denoted by ω_r^{JS}, that satisfy

$$\sum_{r=0}^{k-1} \omega_r^{JS} = 1, \quad \omega_r^{JS} \geq 0 \tag{5.14}$$

They generate a convex combination of the low order reconstructions to compute the final approximation. First the α_r^{JS} coefficients are formulated and then normalized leading to the WENO ω_r^{JS} weights

$$\alpha_r^{JS} = \frac{\gamma_r}{(\beta_r + \epsilon)^2} \qquad \omega_r^{JS} = \frac{\alpha_r}{\sum_{l=0}^{k-1} \alpha_l}, \qquad r = 0, ..., k-1 \tag{5.15}$$

with ϵ a properly defined small parameter. The final WENO approximation of $u(x)$ at $x_{i+\frac{1}{2}}$ is given by

$$u_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} \omega_r^{JS} u_{i+\frac{1}{2}}^{(r)} = u\left(x_{i+\frac{1}{2}}\right) + O\left(\Delta x^{2k-1}\right) \tag{5.16}$$

5.2.1 The WENO-PW reconstruction

A simple and robust improvement of the classical WENO-JS reconstruction method is presented in this work and applied to ADER type numerical schemes. The proposed modification is based on a simple and effective correction of the power exponent used to define the WENO-JS non-oscillatory weights and introduces a global smoothness indicator that is used either to strengthen the essentially non-oscillatory property when discontinuities are present or to compute a better approximation to the optimal weights in smooth regions, specially at critical points (points where derivatives of the reconstructed variable vanish). This global smoothness indicator is constructed using the original smoothness indicators defined in the WENO-JS reconstruction method. This method will be referred to as WENO-PW, where PW stands for the power exponent in the computation of the weights. The WENO-PW method ensures the required accuracy for the optimal weights and numerical results evidence that they are accurately recovered even in presence of critical points. This novel approach presents a robust and simple improvement that does not require an extra computational effort as other methods do.

The global smoothness indicator presented here and denoted by ξ, is defined as follows

$$\xi = \chi^b \,, \qquad \chi = \left(\frac{|\beta_0 - \beta_{k-1}|}{\beta_0 + \beta_{k-1} + \epsilon} \right) \tag{5.17}$$

where ϵ is a small constant to avoid division by zero, selected in this work as 10^{-m}, with m the number of digits of precision of the machine. Parameter b is a positive constant that enhances the ratio inside the parenthesis. The global smoothness indicator is defined to ensure that the ratio inside the parenthesis is always less than unity and greater or equal zero. The α_r coefficients in (5.15) are reformulated using parameter ξ as a power exponent, and then normalized leading to the WENO-PW weights, ω_r^{PW}

$$\alpha_r^{PW} = \frac{\gamma_r}{(\beta_r + \epsilon)^{p\xi}} \qquad \omega_r^{PW} = \frac{\alpha_r^{PW}}{\sum_{l=0}^{k-1} \alpha_l^{PW}}, \qquad r = 0, ..., k-1 \tag{5.18}$$

where parameter p is a positive integer and ϵ is set as in (5.17). Therefore suitable values of b and p are required.

Depending on the relative values of the different β_r smoothness indicators, one can observe that

- when the function is smooth in $\mathcal{T}(i)$, then $\beta_0 \approx \beta_{k-1}$ making ξ tend to 0^+ and α_r^{PW} coefficients become closer to the optimal weights.
- when the function has a discontinuity in $\mathcal{T}(i)$, then $\beta_0 \ll \beta_{k-1}$ or $\beta_0 \gg \beta_{k-1}$ making ξ tend to 1^-, and the WENO-JS strategy is recovered avoiding oscillatory reconstructions.
- when the function is symmetric in $\mathcal{T}(i)$ with respect to x_i, then $\beta_0 \approx \beta_{k-1}$ making ξ tend to 0^+, recovering the optimal weights.

Other existing improved WENO procedures, such as the WENO-M, WENO-Z and WENO-MZ are detailed in Appendix B.3.

5.3 Fundamentals of ADER-type numerical schemes

Following the Godunov method, the expression for the updating scheme is constructed as

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}[\mathbf{F}_{i+1/2}^- - \mathbf{F}_{i-1/2}^+] + \frac{\Delta t}{\Delta x}[\bar{\mathbf{S}}_{i_R, i_L}] \tag{5.19}$$

with the numerical fluxes $\mathbf{F}_{i+1/2}^-$ and $\mathbf{F}_{i-1/2}^+$ defined as time-integral averages of the fluxes at the interfaces

$$\mathbf{F}_{i+1/2}^- = \frac{1}{\Delta t} \int_0^{\Delta t} \mathbf{F}_{i_R}^-(\tau)\, d\tau \qquad \mathbf{F}_{i-1/2}^+ = \frac{1}{\Delta t} \int_0^{\Delta t} \mathbf{F}_{i_L}^+(\tau)\, d\tau \tag{5.20}$$

and $\bar{\mathbf{S}}_{i_R,i_L}$ a suitable approximation of the spatial integral of the source term inside the cell. It is worth mentioning that when constructing high order ADER schemes, the integral of the source term inside the cell is always required in the updating formula since in the resolution of the DRP, geometric source terms are accounted for by integrating them in a region of differential width across the boundaries.

Two different approaches can be used to compute left and right intercell numerical fluxes $\mathbf{F}^-_{i_R}$ and $\mathbf{F}^+_{i_L}$ in (5.20). The first approach is called *state-expansion ADER* and proposes to obtain the solution for conserved variables at the interface by solving the DRP_K with a suitable solver and to evaluate the physical fluxes using this solution. The second option is to use the *flux-expansion ADER* approach, where fluxes $\mathbf{F}^-_{i_R}$ and $\mathbf{F}^+_{i_L}$ are constructed as a truncated power series expansion in time and the components of the expansion are functions of the approximate fluxes defined for each RP associated to the DRP_K.

a) **State-expansion ADER approach.** The numerical fluxes in (5.20) are evaluated using the time-dependent solutions of the DRP_K, $\mathbf{U}^-_{i_R}(\tau)$ and $\mathbf{U}^+_{(i+1)_L}(\tau)$, as

$$\mathbf{F}^-_{i+1/2} = \frac{1}{\Delta t}\int_0^{\Delta t} \mathbf{F}(\mathbf{U}^-_{i_R}(\tau))d\tau \qquad \mathbf{F}^+_{i+1/2} = \frac{1}{\Delta t}\int_0^{\Delta t} \mathbf{F}(\mathbf{U}^+_{(i+1)_L}(\tau))d\tau \tag{5.21}$$

b) **Flux-expansion ADER approach.** When adopting the flux-expansion ADER approach, we seek a truncated Taylor time expansion of the fluxes at the interfaces as

$$\mathbf{F}^-_{i_R}(\tau) = \mathbf{F}^{-,(0)}_{i_R} + \sum_{k=1}^{K} \mathbf{F}^{-,(k)}_{i_R}\frac{\tau^k}{k!} \qquad \mathbf{F}^+_{(i+1)_L}(\tau) = \mathbf{F}^{+,(0)}_{(i+1)_L} + \sum_{k=1}^{K} \mathbf{F}^{+,(k)}_{(i+1)_L}\frac{\tau^k}{k!} \tag{5.22}$$

that, after integration, leads to the following expression of the numerical fluxes

$$\mathbf{F}^-_{i+1/2} = \mathbf{F}^{-,0}_{i_R} + \sum_{k=1}^{K} \mathbf{F}^{-,(k)}_{i_R}\frac{\Delta t^k}{(k+1)!} \qquad \mathbf{F}^+_{i+1/2} = \mathbf{F}^{+,0}_{(i+1)_L} + \sum_{k=1}^{K} \mathbf{F}^{+,(k)}_{(i+1)_L}\frac{\Delta t^k}{(k+1)!} \tag{5.23}$$

where $\mathbf{F}^{-,0}_{i_R}$ and $\mathbf{F}^{+,0}_{(i+1)_L}$ represent the leading terms, obtained as a result of the resolution of the DRP_0 and

$$\mathbf{F}^{-,(k)}_{i_R} = \left[\frac{\partial^k}{\partial t^k}\mathbf{F}^-_{i_R}(\tau)\right]_{t=0} \qquad \mathbf{F}^{+,(k)}_{(i+1)_L} = \left[\frac{\partial^k}{\partial t^k}\mathbf{F}^+_{(i+1)_L}(\tau)\right]_{t=0} \tag{5.24}$$

represent the high order terms of the numerical fluxes. Fluxes in (5.24) will be computed by solving the K RPs corresponding to the evolution equations of the K first spatial or temporal derivatives of $\mathbf{U}$.

As done for the fluxes, the source term inside cell Ω_i can also be approximated by a truncated Taylor power series expansion in time

$$\mathbf{S}_i(x,\tau) = \mathbf{S}_i(x,0) + \sum_{k=1}^{K}\left[\frac{\partial^k \mathbf{S}_i}{\partial t^k}\right]_{x,t=0}\frac{\tau^k}{k!} \tag{5.25}$$

leading to the following expression for its integral inside the cell

$$\bar{\mathbf{S}}_{i_R,i_L} = \bar{\mathbf{S}}^{(0)}_{i_R,i_L} + \sum_{k=1}^{K}\bar{\mathbf{S}}^{(k)}_{i_R,i_L} \tag{5.26}$$

with

$$\begin{aligned}
\bar{\mathbf{S}}^{(0)}_{i_R,i_L} &= \frac{1}{\Delta t}\int_0^{\Delta t}\int_{x_{i_L}}^{x_{i_R}} \mathbf{S}_i(x,0)\; dx\; dt \\
\bar{\mathbf{S}}^{(k)}_{i_R,i_L} &= \frac{1}{\Delta t}\int_0^{\Delta t}\int_{x_{i_L}}^{x_{i_R}} \left[\frac{\partial^k \mathbf{S}_i}{\partial t^k}\right]_{x,t=0}\frac{t^k}{k!}\; dx\; dt
\end{aligned} \tag{5.27}$$

that will be integrated by means of approximated quadrature rules.

As outlined in previous chapters, depending on the nature of the source term, it may be necessary to account for the value of its jump across the interface by means of including it in the approximate fluxes obtained in the resolution of the DRP [1, 25]. That is the case of geometric source terms of the type of (3.14). In this case, the integral of the source and its derivatives at the interface will be denoted by

$$\begin{aligned} \bar{\mathbf{S}}^{(0)}_{i+1/2} &= \frac{1}{\Delta t}\int_0^{\Delta t}\int_{x^-_{i+1/2}}^{x^+_{i+1/2}} \mathbf{S}(x,0)\; dx\; dt \\ \bar{\mathbf{S}}^{(k)}_{i+1/2} &= \frac{1}{\Delta t}\int_0^{\Delta t}\int_{x^-_{i+1/2}}^{x^+_{i+1/2}} \left[\frac{\partial^k \mathbf{S}_i}{\partial t^k}\right]_{x,t=0} dx\; dt \end{aligned} \tag{5.28}$$

that will be integrated using suitable approximations.

5.3.1 Cauchy-Kowalevski Theorem

When dealing with EDPs of the type of (5.1), relations between temporal and spatial derivatives of $\mathbf{U}$ are provided by the Cauchy-Kowalevski Theorem. Here, it is used to derive analytic expressions for time derivatives of $\mathbf{F}$ and $\mathbf{U}$ departing from the information provided by the spatial reconstruction method. It allows to express time derivatives of the physical fluxes at $t = 0$ as functions $\mathbf{R}^{(k)}$ of spatial derivatives of $\mathbf{U}$ and $\mathbf{S}$

$$\partial_t^{(k)}\mathbf{F} = \mathbf{R}^{(k)}(\partial_x^{(k)}\mathbf{U}, \partial_x^{(k-1)}\mathbf{U}, ..., \mathbf{U}, \partial_x^{(k)}\mathbf{S}, \partial_x^{(k-1)}\mathbf{S}, ..., \mathbf{S}) \tag{5.29}$$

Spatial derivatives defined at the cell interface $i + 1/2$ in (5.3) are calculated using the sub-cell WENO reconstruction method [21]. They allow to compute the values of $\mathbf{R}^{(k)}$ at each side of the discontinuity in the DRP$_K$

$$\begin{aligned} \mathbf{R}^{(k)}_{i_R} &= \lim_{x\to 0^-} \mathbf{R}^{(k)} \approx \mathbf{R}^{(k)}(\mathbf{U}^{(k)}, \mathbf{U}^{(k-1)}, ..., \mathbf{U}^0, \mathbf{S}^{(k)}, \mathbf{S}^{(k-1)}, ..., \mathbf{S}^0)_{i_R} \\ \mathbf{R}^{(k)}_{(i+1)_L} &= \lim_{x\to 0^+} \mathbf{R}^{(k)} \approx \mathbf{R}^{(k)}(\mathbf{U}^{(k)}, \mathbf{U}^{(k-1)}, ..., \mathbf{U}^0, \mathbf{S}^{(k)}, \mathbf{S}^{(k-1)}, ..., \mathbf{S}^0)_{(i+1)_L} \end{aligned} \tag{5.30}$$

Analogously, it is possible to construct temporal derivatives of $\mathbf{U}$ at $t = 0$ as functions $\mathbf{D}^{(k)}$ of spatial derivatives of $\mathbf{U}$ and $\mathbf{S}$

$$\partial_t^{(k)}\mathbf{U} = \mathbf{D}^{(k)}(\partial_x^{(k)}\mathbf{U}, \partial_x^{(k-1)}\mathbf{U}, ..., \mathbf{U}, \partial_x^{(k)}\mathbf{S}, \partial_x^{(k-1)}\mathbf{S}, ..., \mathbf{S}) \tag{5.31}$$

allowing to compute the values of $\mathbf{D}^{(k)}$ at each side of the discontinuity in the DRP$_K$

$$\begin{aligned} \mathbf{D}^{(k)}_{i_R} &= \lim_{x\to 0^-} \mathbf{D}^{(k)} \approx \mathbf{D}^{(k)}(\mathbf{U}^{(k)}, \mathbf{U}^{(k-1)}, ..., \mathbf{U}^0, \mathbf{S}^{(k)}, \mathbf{S}^{(k-1)}, ..., \mathbf{S}^0)_{i_R} \\ \mathbf{D}^{(k)}_{(i+1)_L} &= \lim_{x\to 0^+} \mathbf{D}^{(k)} \approx \mathbf{D}^{(k)}(\mathbf{U}^{(k)}, \mathbf{U}^{(k-1)}, ..., \mathbf{U}^0, \mathbf{S}^{(k)}, \mathbf{S}^{(k-1)}, ..., \mathbf{S}^0)_{(i+1)_L} \end{aligned} \tag{5.32}$$

Temporal derivatives of the source term, $\mathbf{S}$, at $t = 0$ can also be obtained using the Cauchy-Kowalevski procedure as functions $\mathbf{Q}^{(k)}$ of spatial derivatives of $\mathbf{U}$ and $\mathbf{S}$

$$\partial_t^{(k)}\mathbf{S} = \mathbf{Q}^{(k)}(\partial_x^{(k)}\mathbf{U}, \partial_x^{(k-1)}\mathbf{U}, ..., \mathbf{U}, \partial_x^{(k)}\mathbf{S}, \partial_x^{(k-1)}\mathbf{S}, ..., \mathbf{S}) \tag{5.33}$$

5.3.2 Evolution equation for derivatives

DRP in (5.1) provides the evolution equation for variable $\mathbf{U}$. Evolution equations for spatial or temporal derivatives of $\mathbf{U}$, denoted by $\partial_x^{(k)}\mathbf{U}$ and $\partial_t^{(k)}\mathbf{U}$ respectively, are straightforward obtained when substituting the conserved variable by its spatial or temporal derivative as

$$\frac{\partial}{\partial t}\left(\partial_x^{(k)}\mathbf{U}\right)+\frac{\partial}{\partial x}\left(\partial_x^{(k)}\mathbf{F}(\mathbf{U})\right)=\partial_x^{(k)}\mathbf{S}\qquad k=1,...,K \tag{5.34}$$

and

$$\frac{\partial}{\partial t}\left(\partial_t^{(k)}\mathbf{U}\right)+\frac{\partial}{\partial x}\left(\partial_t^{(k)}\mathbf{F}(\mathbf{U})\right)=\partial_t^{(k)}\mathbf{S}\qquad k=1,...,K \tag{5.35}$$

respectively. Algebraic manipulations of (5.34) yields

$$\frac{\partial}{\partial t}\left(\partial_x^{(k)}\mathbf{U}\right)+\mathbf{J}(\mathbf{U})\frac{\partial}{\partial x}\left(\partial_x^{(k)}\mathbf{U}\right)=\boldsymbol{\Upsilon}^{(k)} \tag{5.36}$$

where $\boldsymbol{\Upsilon}^{(k)}=\boldsymbol{\Upsilon}^{(k)}(\partial_x^{(k)}\mathbf{U},\partial_x^{(k-1)}\mathbf{U},...,\mathbf{U},\partial_x^{(k)}\mathbf{S},\partial_x^{(k-1)}\mathbf{S},...,\mathbf{S})$ is a function of spatial derivatives of the fluxes and sources, that can be expressed as

$$\boldsymbol{\Upsilon}^{(k)}=-\frac{\partial^k}{\partial x^k}\left(\mathbf{J}(\mathbf{U})\frac{\partial\mathbf{U}}{\partial x}\right)+\mathbf{J}(\mathbf{U})\frac{\partial}{\partial x}\left(\frac{\partial^k\mathbf{U}}{\partial x^k}\right)+\frac{\partial^k\mathbf{S}}{\partial x^k} \tag{5.37}$$

Analogously, evolution equations for time derivatives of $\mathbf{U}$ are expressed as

$$\frac{\partial}{\partial t}\left(\partial_t^{(k)}\mathbf{U}\right)+\mathbf{J}(\mathbf{U})\frac{\partial}{\partial x}\left(\partial_t^{(k)}\mathbf{U}\right)=\boldsymbol{\Psi}^{(k)} \tag{5.38}$$

where $\boldsymbol{\Psi}^{(k)}=\boldsymbol{\Psi}^{(k)}(\partial_t^{(k)}\mathbf{U},\partial_t^{(k-1)}\mathbf{U},...,\mathbf{U},\partial_t^{(k)}\mathbf{S},\partial_t^{(k-1)}\mathbf{S},...,\mathbf{S})$ is again a function of temporal derivatives of the fluxes and sources.

Different approaches can be done to find the solution for temporal derivatives in (5.36) or (5.38). In this work, the Jacobian will be considered as a constant coefficient matrix evaluated at $t=0$, that means, spatial and temporal derivatives will be evolved using constant wave speeds corresponding to the eigenvalues of the Jacobian at the initial time. This leads to the following simplification of the evolution equations for derivatives

$$\frac{\partial}{\partial t}\left(\partial_x^{(k)}\mathbf{U}\right)+\mathbf{J}(\mathbf{U}^{(0)})\frac{\partial}{\partial x}\left(\partial_x^{(k)}\mathbf{U}\right)=\partial_x^{(k)}\mathbf{S} \tag{5.39}$$

$$\frac{\partial}{\partial t}\left(\partial_t^{(k)}\mathbf{U}\right)+\mathbf{J}(\mathbf{U}^{(0)})\frac{\partial}{\partial x}\left(\partial_t^{(k)}\mathbf{U}\right)=\partial_t^{(k)}\mathbf{S} \tag{5.40}$$

noticing that derivatives of $\mathbf{U}$ and variable $\mathbf{U}$ are evolved using the same law.

5.4 ADER scheme for linear scalar PDEs

In order to illustrate the fundamentals of ADER-type numerical schemes, the ADER scheme proposed in [24], referred to as TT-ADER scheme or TT-ADER solver, is considered in this chapter for the resolution of the linear scalar advection-reaction equation

$$\frac{\partial u}{\partial t}+\lambda\frac{\partial u}{\partial x}=\zeta u \tag{5.41}$$

where λ is a constant propagation speed and ζ represents the strength of the reactive term. It is worth mentioning that the resolution of the same problem when the propagation speeds depends upon the spatial coordinate, that is $\lambda=\lambda(x)$, is equivalent.

In (5.41), the source term is expressed as

$$s(x,t)=\zeta u(x,t) \tag{5.42}$$

and the integral value of the source term in space and time inside a cell Ω_i is given by

$$\bar{s}_i = \frac{1}{\Delta t}\int_0^{\Delta t}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \zeta u(x,t)dxdt \tag{5.43}$$

The value of the function u is unknown at an arbitrary time $t = \tau$, however, this value can be reconstructed by means of a power-series time expansion in the following way

$$u_i(x,\tau) = u_i(x,0) + \sum_{k=1}^{K}\left[\frac{\partial^k}{\partial t^k}u_i(x,0)\right]\frac{\tau^k}{k!} \tag{5.44}$$

where $K+1$ is equal to the desired order of accuracy.

In order to compute the evolution in time of u using (5.44), time derivatives of the function are needed. The Cauchy-Kowalewski method is applied to express time derivatives as functions of space derivatives. A general expression for the k-th derivative of u can be derived from the PDE

$$\frac{\partial^k u}{\partial t^k} = \sum_{l=0}^{k}(-\lambda)^l\zeta^{k-l}\frac{k!}{l!(k-l)!}\frac{\partial^l u}{\partial x^l} \tag{5.45}$$

Inserting 5.45 in 5.44, the latter becomes

$$u_i(x,\tau) = u_i(x,0) + \sum_{k=1}^{K}\left[\sum_{l=0}^{k}(-\lambda)^l\zeta^{k-l}\frac{k!}{l!(k-l)!}\frac{\partial^l u_i}{\partial x^l}(x,0)\right]\frac{\tau^k}{k!} \tag{5.46}$$

where $\frac{\partial^l u_i}{\partial x^l}(x,0)$ are the piece-wise reconstruction of derivatives carried out by a suitable derivative reconstruction procedure, such as the WENO sub-cell derivative reconstruction method. For the sake of simplicity, the following function is defined

$$\psi(k,l) = (-\lambda)^l\zeta^{k-l}\frac{k!}{l!(k-l)!} \tag{5.47}$$

and can be used to rewrite (5.46) in a more compact form

$$u_i(x,\tau) = u_i(x,0) + \sum_{k=1}^{K}\left[\sum_{l=0}^{k}\psi(k,l)\frac{\partial^l u_i}{\partial x^l}(x,0)\right]\frac{\tau^k}{k!} \tag{5.48}$$

Numerical scheme

To construct high-order numerical methods of the ADER type it is sufficient to find the solution for the fluxes at the interface position $x = 0$, as a function of time τ. In the scalar linear case, functions $u_i^- = u(x = 0^-, t = \tau)$ and $u_{i+1}^+ = u(x = 0^+, t = \tau)$ will provide sufficient information to compute the numerical fluxes to construct a numerical scheme of $K+1$-th order of accuracy in both space and time. The DRP

$$\begin{cases} \dfrac{\partial u}{\partial t} + \lambda\dfrac{\partial u}{\partial x} = \zeta u \\ u(x,0) = \begin{cases} u_i(x) & x < 0 \\ u_{i+1}(x) & x > 0 \end{cases} \end{cases} \tag{5.49}$$

is solved at each cell interface using linear solutions, leading to the following explicit conservative formula

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}[f_{i+1/2}^- - f_{i-1/2}^+] + \bar{s}_i\Delta t \tag{5.50}$$

with the numerical fluxes $f_{i+1/2}^{\pm}$ defined as a time-integral averages

$$f_{i+1/2}^- = \frac{1}{\Delta t}\int_0^{\Delta t} f_{i_R}^- d\tau \qquad f_{i+1/2}^+ = \frac{1}{\Delta t}\int_0^{\Delta t} f_{(i+1)_L}^+ d\tau \tag{5.51}$$

Step I: The leading terms.

To compute the leading terms, weak solutions of the classical Riemann Problem are provided using the integral form of (5.41), neglecting the source term. The following RP is defined

$$\partial_t u + \partial_x f = 0 \qquad u(x,0) = \begin{cases} u^{(0)}_{i_R} & \text{if} \quad x < 0 \\ u^{(0)}_{(i+1)_L} & \text{if} \quad x > 0 \end{cases} \tag{5.52}$$

with

$$u^{(0)}_{i_R} = \lim_{x \to 0^-} u_i(x) \qquad u^{(0)}_{(i+1)_L} = \lim_{x \to 0^+} u_{i+1}(x) \tag{5.53}$$

Using previous definitions, the approximate solutions for the DRP_0, $u^{-,(0)}_{i_R}$ and $u^{+,(0)}_{(i+1)_L}$ are given by

$$\begin{aligned} u^{-,(0)}_{i_R} &= \begin{cases} u^{(0)}_{i_R} & \text{if} \quad \widetilde{\lambda}_{i+1/2} > 0 \\ u^{(0)}_{i_R} + \delta u^{(0)}_{i+1/2} & \text{if} \quad \widetilde{\lambda}_{i+1/2} < 0 \end{cases} \\ u^{+,(0)}_{(i+1)_L} &= \begin{cases} u^{(0)}_{(i+1)_L} - \delta u^{(0)}_{i+1/2} & \text{if} \quad \widetilde{\lambda}_{i+1/2} > 0 \\ u^{(0)}_{(i+1)_L} & \text{if} \quad \widetilde{\lambda}_{i+1/2} < 0 \end{cases} \end{aligned} \tag{5.54}$$

where $\widetilde{\lambda}_{i+1/2}$ stands for the approximate wave celerity at $x_{i+1/2}$ obtained when imposing the consistency condition using the integral form of the (5.52). If solving a linear flux of the type $f(u) = \lambda(x)u$, then $\widetilde{\lambda}_{i+1/2} = \lambda(x_{i+1/2})$. In what follows, the wave speed $\lambda(x)$ is considered to be known in the spatial domain and will be denoted by $\lambda_{i+1/2} = \lambda(x_{i+1/2})$. It is worth mentioning that when solving a linear RP, the approximate solver provides the exact solution, as in this case.

The values of the approximate fluxes for the DRP_0, $f^{-,(0)}_{i_R}$ and $f^{+,(0)}_{(i+1)_L}$ are given by

$$f^{-,(0)}_{i_R} = f^{(0)}_{i_R} + (\lambda^- \delta u^{(0)})_{i+1/2}\,, \qquad f^{+,(0)}_{(i+1)_L} = f^{(0)}_{(i+1)_L} - (\lambda^+ \delta u^{(0)})_{i+1/2} \tag{5.55}$$

with

$$f^{(0)}_{i_R} = \lim_{x \to 0^-} f(u_i(x))\,, \qquad f^{(0)}_{(i+1)_L} = \lim_{x \to 0^+} f(u_{i+1}(x)) \tag{5.56}$$

and

$$\lambda^{\pm}_{i+1/2} = \frac{1}{2}\left(\lambda \pm |\lambda|\right)_{i+1/2} \tag{5.57}$$

Step II : Higher order terms. There are three sub-steps here:

(1) Time derivatives in terms of spatial derivatives: Application of the Cauchy–Kowalewski procedure to equation in (5.41) gives

$$\partial^{(k)}_t u = \sum_{l=0}^{k} \psi(k,l) \partial^{(l)}_x u \qquad k = 1, \ldots, K \tag{5.58}$$

$$\partial^{(k)}_t f = \lambda \sum_{l=0}^{k} \psi(k,l) \partial^{(l)}_x u \qquad k = 1, \ldots, K \tag{5.59}$$

(2) Evolution equations for spatial derivatives: Now the problem is reduced to solving for the spatial derivatives at the interface, posing the following equation

$$\partial_t(\partial^{(k)}_x u) + \lambda \partial_x(\partial^{(k)}_x u) = \zeta \partial^{(k)}_x u \qquad k = 1, \ldots, K \tag{5.60}$$

according to (5.36).

(3) Riemann problems for spatial derivatives: One solves a simplified Riemann problem of that in (5.61) by assuming that the source term can be neglected.

$$\partial_t(\partial_x^{(k)}u) + \lambda\partial_x(\partial_x^{(k)}u) = 0 \qquad \partial_x^{(k)}u(x,0) = \begin{cases} u_{i_R}^{(k)} & \text{if} \quad x < 0 \\ u_{(i+1)_L}^{(k)} & \text{if} \quad x > 0 \end{cases} \tag{5.61}$$

with

$$u_{i_R}^{(k)} = \lim_{x\to 0^-} u_i^{(k)}(x) \qquad u_{(i+1)_L}^{(k)} = \lim_{x\to 0^+} u_{i+1}^{(k)}(x) \tag{5.62}$$

By solving these Riemann Problems for space derivatives, the following solution is obtained

$$\begin{aligned} u_{i_R}^{-,(k)} &= \begin{cases} u_{i_R}^{(k)} & \text{if} \quad \lambda_{i+1/2} > 0 \\ u_{i_R}^{(k)} + \delta u_{i+1/2}^{(k)} & \text{if} \quad \lambda_{i+1/2} < 0 \end{cases} \\ u_{(i+1)_L}^{+,(k)} &= \begin{cases} u_{(i+1)_L}^{(k)} - \delta u_{i+1/2}^{(k)} & \text{if} \quad \lambda_{i+1/2} > 0 \\ u_{(i+1)_L}^{(k)} & \text{if} \quad \lambda_{i+1/2} < 0 \end{cases} \end{aligned} \tag{5.63}$$

The values of the approximate fluxes, $f_{i_R}^{-,(k)}$ and $f_{(i+1)_L}^{+,(k)}$ are given by

$$f_{i_R}^{-,(k)} = f_{i_R}^{(k)} + (\lambda^-\delta u^{(k)})_{i+1/2}\,, \qquad f_{(i+1)_L}^{+,(0)} = f_{(i+1)_L}^{(k)} - (\lambda^+\delta u^{(k)})_{i+1/2} \tag{5.64}$$

with

$$f_{i_R}^{(k)} = \lim_{x\to 0^-} \frac{\partial^k}{\partial t^k} f(u_i(x))\,, \qquad f_{(i+1)_L}^{(k)} = \lim_{x\to 0^+} \frac{\partial^k}{\partial t^k} f(u_{i+1}(x)) \tag{5.65}$$

For this particular case, derivatives of the fluxes in (5.65) can be straightforward derived assuming

$$\partial_t f^{(k)} = \lambda_{i+1/2}\partial_t u^{(k)}\,, \quad k = 1,\ldots,K \tag{5.66}$$

leading to

$$f_{i_R}^{(k)} = \lambda_{i+1/2} u_{i_R}^{(k)}\,, \qquad f_{(i+1)_L}^{(k)} = \lambda_{i+1/2} u_{(i+1)_L}^{(k)} \tag{5.67}$$

Step (III): The solution of the DRP$_K$. The solution is computed as a power series expansion by replacing time derivatives by spatial derivatives using the results of the Cauchy–Kowalewski procedure in (5.58)

$$\begin{aligned} u_{i_R}^{-}(\tau) &= u_{i_R}^{-,(0)} + \sum_{k=1}^{K}\sum_{l=0}^{k} \psi(k,l)\left[u_{i_R}^{-,(l)}\right]\frac{\tau^k}{k!}\,, \\ u_{(i+1)_L}^{+}(\tau) &= u_{(i+1)_L}^{+,(0)} + \sum_{k=1}^{K}\sum_{l=0}^{k} \psi(k,l)\left[u_{(i+1)_L}^{+,(l)}\right]\frac{\tau^k}{k!}\,. \end{aligned} \tag{5.68}$$

Two different approaches can be taken now: the first one is the so-called state expansion ADER, which computes the numerical fluxes in (5.51) as the evaluation of the physical fluxes using the solution of the DRP$_K$ in (5.68); the second approach is called flux-expansion ADER and is based in the construction of the the numerical fluxes in (5.51) as a truncated power series expansion in time, using the Cauchy–Kowalewski procedure. Due to the linear nature of the problem, both options lead to the same expression of the numerical fluxes

$$f^{-}_{i+1/2} = \left[f^{(0)}_{i_R} + (\lambda^{-}\delta u^{(0)})_{i+1/2}\right] + \sum_{k=1}^{K}\sum_{l=0}^{k}\psi(k,l)\left[\lambda_{i+1/2}u^{(l)}_{i_R} + (\lambda^{-}\delta u^{(l)})_{i+1/2}\right]\frac{\Delta t^k}{(k+1)!}$$
$$f^{+}_{i+1/2} = \left[f^{(0)}_{(i+1)_L} - (\lambda^{+}\delta u^{(0)})_{i+1/2}\right] + \sum_{k=1}^{K}\sum_{l=0}^{k}\psi(k,l)\left[\lambda_{i+1/2}u^{(l)}_{(i+1)_L} - (\lambda^{+}\delta u^{(l)})_{i+1/2}\right]\frac{\Delta t^k}{(k+1)!} \tag{5.69}$$

Step (IV): Integration of the source term. The derivation of the exact expression for the definite integral of the source term inside the cell is straightforward departing from (5.48). Inserting (5.48) in (5.43), we obtain

$$\bar{s}_i = \frac{1}{\Delta t}\int_0^{\Delta t}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\zeta\left[u_i(x,0) + \sum_{k=1}^{K}\left(\sum_{l=0}^{k}\psi(k,l)\frac{\partial^l u_i}{\partial x^l}(x,0)\right)\frac{\tau^k}{k!}\right]dxd\tau \tag{5.70}$$

Then, the expression for the integral can be obtained as a function of spatial derivatives of u as

$$\begin{aligned}\bar{s}_i = \frac{\zeta}{\Delta t}&\left[\bar{u}_i\Delta t\Delta x + \sum_{k=2}^{K}\left(\sum_{l=2}^{k}\psi(k,l)\left[\frac{\partial^{l-1}u_i}{\partial x^{l-1}}(x,0)\right]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\right)\frac{\Delta t^{k+1}}{(k+1)!}\right] + \\ +\frac{\zeta}{\Delta t}&\left[\sum_{k=1}^{K}\psi(k,0)\bar{u}_i\Delta x\frac{\Delta t^{k+1}}{(k+1)!} + \sum_{k=1}^{K}\psi(k,1)\left(u_i(x_{i+\frac{1}{2}}) - u_i(x_{i-\frac{1}{2}})\right)\frac{\Delta t^{k+1}}{(k+1)!}\right]\end{aligned} \tag{5.71}$$

It is worth recalling that $K+1$ is equal to the desired order of accuracy.

Step (V): Update the solution. Once numerical fluxes and source term have been calculated, they can be used in Equation (5.50) to update the solution in time. The time step (here and in previous steps) must be chosen taking into account the Courant–Friedrichs–Lewy (CFL) condition

$$\Delta t = CFL\cdot\min\left(\frac{\Delta x}{\lambda_{i+1/2}}\right) \tag{5.72}$$

where $CFL < 1$ in 1D and $CFL < 0.5$ in 2D.

5.5 Extension of the ADER scheme to 2 spatial dimensions

Let us consider now the homogeneous problem

$$\frac{\partial u}{\partial t} + \frac{\partial f_1}{\partial x} + \frac{\partial f_2}{\partial y} = 0\,, \qquad \forall x, y \in \Omega \subseteq \mathbb{R}^2 \tag{5.73}$$

where $f_1 = f_1(u)$ and $f_2 = f_2(u)$ are the components in each coordinate direction of the flux $\mathbf{E} = (f_1, f_2) : \mathbb{R} \to \mathbb{R}^2$ and with the initial and boundary conditions yet to be defined for the advected function $u = u(x, y, t)$. When considering the 2D linear advection equation, the components of the flux are defined as $f_1 = \lambda_1 u$ and $f_2 = \lambda_2 u$, with $\lambda_1 = \lambda_1(x, y)$ and $\lambda_2 = \lambda_2(x, y)$ the components of the velocity field $\mathbf{v} = (\lambda_1, \lambda_2)$, known inside Ω.

Using definitions in (3.2) and (3.3), the computational grid is defined and Equation (5.73) is integrated inside each cell Ω_i and within a time $\Delta t = t^{n+1} - t^n$. Application of the Gauss-Ostrogradsky theorem allows to express this integral as

$$u_i^{n+1} = u_i^n - \frac{1}{\vartheta_i}\int_0^{\Delta t}\int_{\partial\Omega_i}\mathbf{E}\hat{\mathbf{n}}d\Gamma_i dt\,. \tag{5.74}$$

If considering a quadrilateral grid composed of regular squares and constant cell area Δx^2, the TT-ADER numerical scheme can be expressed as

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x^2} \sum_{r=1}^{4} f_r^- \,, \tag{5.75}$$

where f_r^- stands for the total leaving numerical flux across each cell edge, calculated as the integral of the numerical flux along the cell edge

$$f_r^- = \int_0^{\Delta x} f_r^-(\nu) d\nu \,, \tag{5.76}$$

with $f_r^-(\nu)$ the numerical flux and $\nu \in \mathbb{R}$ the spatial position inside the edge. To construct a numerical scheme of order $K+1$-th, it is sufficient to approximate integral in (5.76) using a $K+1$-th order Gaussian quadrature rule as

$$f_r^- = \frac{\Delta x}{2} \sum_{l=1}^{k} w_l f_{r,l}^- \,, \tag{5.77}$$

where w_l are the Gaussian weights at $l = 1, ..., k$ quadrature points along the cell edge and $f_{r,l}^- = f_r^-(\nu_l)$ the computed numerical fluxes at each of these points. Quadrature points, ν_l, are taken from the general integration interval $[-1, 1]$ and transformed to the new interval $[0, \Delta x]$ by

$$\nu_l = \frac{(\mathcal{G}_l + 1)\Delta x}{2} \,, \tag{5.78}$$

where $\mathcal{G}_l \in [-1, 1]$ are the original quadrature points. When using k points of quadrature, a $2k-1 = K+1$-th order of accuracy is reached. Pointwise numerical fluxes $f_{r,l}^-$ are calculated by solving the following one dimensional DRPs at each quadrature point and for each cell edge

$$\frac{\partial u}{\partial t} + \lambda_{\hat{n}_r} \nabla u \cdot \hat{\mathbf{n}}_r = 0 \tag{5.79}$$

where $\lambda_{\hat{n}_r} = \mathbf{v} \cdot \hat{\mathbf{n}}_r$ is the projection of the velocity vector onto the normal surface direction and $\nabla u \cdot \hat{\mathbf{n}}_r$ the directional derivative of the advected variable in the normal surface direction, with $\hat{\mathbf{n}}_r$ as depicted in Figure 5.3.

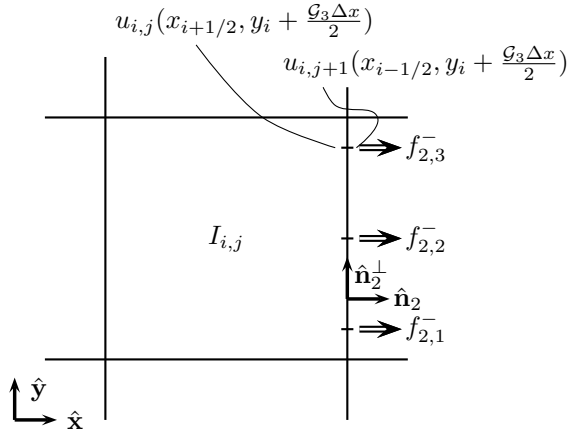

Figure 5.3: Cell edge discretization for the calculation of the numerical flux across the right edge ($r = 2$) to construct a 5-th order ($k = 3$) ADER scheme. As $k = 3$, 3 quadrature points are required. The value of the conserved variable u is also indicated at left and right sides of point $r = 2$, $l = 3$.

Let us define also the projection of the velocity vector onto a vector parallel to the cell edge, as $\lambda_{\hat{n}_r^\perp} = \mathbf{v} \cdot \hat{\mathbf{n}}_r^\perp$, where $\hat{\mathbf{n}}_r^\perp = \mathbf{R}\,\hat{\mathbf{n}}_r$ is this parallel vector expressed in terms of the normal surface vector and $\mathbf{R}$ is a rotation matrix defined as

$$\mathbf{R} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{5.80}$$

In a general framework, one dimensional DRPs, yet to be formulated, can be expressed in a new system of reference relative to the r-th cell edge given by

$$\begin{pmatrix} \breve{x}_r \\ \breve{y}_r \end{pmatrix} = \begin{pmatrix} \cos\theta_r & \sin\theta_r \\ -\sin\theta_r & \cos\theta_r \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \tag{5.81}$$

with $\theta_r = \cos^{-1}(\hat{\mathbf{n}}_r \cdot \hat{\mathbf{x}})$ the angle between the normal surface vector and the x axis. Therefore, $u(x, y, t)$ can be expressed in terms of the new coordinates as $u(\breve{x}_r, \breve{y}_r, t)$. In what follows, subscript r is dropped for the sake of simplicity. For each cell edge, a different DRP is solved at each of the k 1D gaussian quadrature points along the cell edge. For each DRP, the reference for the position $(\breve{x}, \breve{y}) = (0, 0)$ is considered to be at each quadrature point.

The one dimensional DRP is posed at each quadrature point $l = 1, ..., k$ and for each cell edge $r = 1, ..., N_{sides}$ as

$$\begin{cases} \dfrac{\partial u}{\partial t} + \lambda_{\hat{n}} \dfrac{\partial u}{\partial \breve{x}} = 0 \\ \\ u(\breve{x}, \breve{y}, t = 0) = \begin{cases} u_i(\breve{x}, \breve{y}) & \breve{x} < 0 \\ u_{i+1}(\breve{x}, \breve{y}) & \breve{x} > 0 \end{cases} \end{cases} \tag{5.82}$$

and must be solved following a similar procedure than for the 1D case. A suitable reconstruction technique is used to obtain piecewise 2D polynomial reconstruction of u inside each cell and used as initial condition for (5.82). The reconstructed piecewise polynomial function is denoted by $u_{i,j}(x, y)$ inside each cell, using global coordinates. For instance, initial condition for the one-dimensional DRP associated to quadrature point $l = 3$ in edge $r = 2$, according to Figure 5.3, is given by $u_i(\breve{x}, \breve{y}) = u_{i,j}(x_{i+1/2} + \breve{x}, y_i + \frac{\mathcal{G}_3 \Delta x}{2} + \breve{y})$ and $u_{i+1}(\breve{x}, \breve{y}) = u_{i,j+1}(x_{i-1/2} + \breve{x}, y_i + \frac{\mathcal{G}_3 \Delta x}{2} + \breve{y})$.

In this case, the Cauchy-Kowalevski theorem leads to the following expressions for time derivatives of the conserved variable and fluxes

$$\frac{\partial^k u}{\partial t^k} = \sum_{l=0}^{k} \hat{\psi}(k, l) \frac{\partial^k u}{\partial \breve{x}^l \partial \breve{y}^{k-l}}, \qquad k = 1, \ldots, K \tag{5.83}$$

$$\frac{\partial^k f}{\partial t^k} = \lambda \sum_{l=0}^{k} \hat{\psi}(k, l) \frac{\partial^k u}{\partial \breve{x}^l \partial \breve{y}^{k-l}}, \qquad k = 1, \ldots, K \tag{5.84}$$

with $\hat{\psi}(k, l)$ given by

$$\hat{\psi}(k, l) = (-1)^k \lambda_{\hat{n}}^l \, \lambda_{\hat{n}^\perp}^{k-l} \, \frac{k!}{l!(k-l)!}. \tag{5.85}$$

The solution of the DRP_K is given by

$$\begin{aligned} u^-_{i_R}(\tau) &= u^{-,(0)}_{i_R} + \sum_{k=1}^{K} \sum_{l=0}^{k} \hat{\psi}(k, l) \left[u^{-,(l,k-l)}_{i_R} \right] \frac{\tau^k}{k!}, \\ u^+_{(i+1)_L}(\tau) &= u^{+,(0)}_{(i+1)_L} + \sum_{k=1}^{K} \sum_{l=0}^{k} \hat{\psi}(k, l) \left[u^{+,(l,k-l)}_{(i+1)_L} \right] \frac{\tau^k}{k!}, \end{aligned} \tag{5.86}$$

where $u^{-,(0)}_{i_R}$ and $u^{+,(0)}_{(i+1)_L}$ are the zeroth order solutions, obtained for the following RP

$$\frac{\partial u}{\partial t} + \lambda_{\hat{n}} \frac{\partial u}{\partial \breve{x}} = 0, \qquad u(\breve{x}, 0) = \begin{cases} u^{(0)}_i & \text{if} \quad \breve{x} < 0 \\ u^{(0)}_{i+1} & \text{if} \quad \breve{x} > 0 \end{cases} \tag{5.87}$$

with

$$u_{i_R}^{(0)} = \lim_{\breve{x}\to 0^-} u_i(\breve{x},0)\,, \qquad u_{(i+1)_L}^{(0)} = \lim_{\breve{x}\to 0^+} u_{i+1}(\breve{x},0) \tag{5.88}$$

and where $u_{i_R}^{-,(l,k-l)}$ and $u_{(i+1)_L}^{+,(l,k-l)}$ are the solutions for the derivatives, obtained for the following RPs

$$\frac{\partial}{\partial t}\left(\frac{\partial^k u}{\partial \breve{x}^l \partial \breve{y}^{k-l}}\right) + \lambda_{\hat{n}} \frac{\partial}{\partial \breve{x}}\left(\frac{\partial^k u}{\partial \breve{x}^l \partial \breve{y}^{k-l}}\right) = 0\,, \qquad \frac{\partial^k u}{\partial \breve{x}^l \partial \breve{y}^{k-l}} = \begin{cases} u_i^{(l,k-l)} & \text{if} \quad \breve{x} < 0 \\ u_{i+1}^{(l,k-l)} & \text{if} \quad \breve{x} > 0 \end{cases} \tag{5.89}$$

with

$$u_{i_R}^{(l,k-l)} = \lim_{\breve{x}\to 0^-} \left.\frac{\partial^k u_i(\breve{x},\breve{y})}{\partial \breve{x}^l \partial \breve{y}^{k-l}}\right|_{\breve{y}=0}\,, \qquad u_{(i+1)_L}^{(l,k-l)} = \lim_{\breve{x}\to 0^+} \left.\frac{\partial^k u_{i+1}(\breve{x},\breve{y})}{\partial \breve{x}^l \partial \breve{y}^{k-l}}\right|_{\breve{y}=0} \tag{5.90}$$

for $k = 1, ..., K$, $l = 0, ..., k$.

Numerical fluxes are finally computed using the following expressions

$$\begin{aligned} f_{i+1/2}^{-} &= \left[f_{i_R}^{(0)} + (\lambda^- \delta u^{(0)})_{i+1/2}\right] + \sum_{k=1}^{K}\sum_{l=0}^{k} \hat{\psi}(k,l)\left[\lambda_{i+1/2} u_{i_R}^{(l,k-l)} + (\lambda^- \delta u^{(l,k-l)})_{i+1/2}\right] \frac{\Delta t^k}{(k+1)!} \\ f_{i+1/2}^{+} &= \left[f_{(i+1)_L}^{(0)} - (\lambda^+ \delta u^{(0)})_{i+1/2}\right] + \sum_{k=1}^{K}\sum_{l=0}^{k} \hat{\psi}(k,l)\left[\lambda_{i+1/2} u_{(i+1)_L}^{(l,k-l)} - (\lambda^+ \delta u^{(l,k-l)})_{i+1/2}\right] \frac{\Delta t^k}{(k+1)!} \end{aligned} \tag{5.91}$$

5.6 The AR-ADER scheme

A flux-ADER type numerical scheme for the resolution of nonlinear PDEs with source terms, called Augmented Roe ADER (AR-ADER) scheme, is presented in this section. This scheme is constructed following the flux-ADER approach, that is, instead of searching solutions of the conserved quantities at both sides of the interface to evaluate the fluxes, approximate intercell numerical fluxes are sought. Special emphasis is put on the discretization and incorporation of the source term in the solution of the DRP_K when dealing with geometric source terms of the type of (3.14).

The novelty of this solver can be summarized in the following points:

- A high-order flux-ADER type numerical scheme, named AR-ADER, is proposed as an extension of a first order solver based on weak solutions of RPs with discontinuous source terms.
- In contrast with other previously defined ADER schemes, the AR-ADER considers the presence of the source term in the solutions of the DRP, enhancing the capabilities of this types numerical schemes. Another distinctive feature is that it departs from time derivatives of the fluxes to compute the high order terms of the DRP, instead of spatial derivatives of the conserved variables.
- When applied to the shallow water equations, the numerical scheme includes the energy balanced property leading to exact numerical solutions for steady solutions with independence of the grid refinement and the order of accuracy. It ensures convergence to the exact solution in Riemann problems with source terms.

5.6.1 The AR-ADER scheme for scalar equations

The following scalar non-linear problem is considered here

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = s \tag{5.92}$$

As outlined in the previous chapter, to construct a $K+1$-th order ADER scheme the following DRP_K has to be solved at each interface

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial f(u)}{\partial x} = s \\ u(x,0) = \begin{cases} u_i(x) & x < 0 \\ u_{i+1}(x) & x > 0 \end{cases} \end{cases} \tag{5.93}$$

and the resulting numerical fluxes $f^-_{i+1/2}$ and $f^+_{i-1/2}$ are used to construct the following updating scheme

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}[f^-_{i+1/2} - f^+_{i-1/2}] + \frac{\Delta t}{\Delta x}[\bar{s}_{i_R,i} + \bar{s}_{i,i_L}] \tag{5.94}$$

with $f^-_{i+1/2}$ and $f^+_{i-1/2}$ defined as a time-integral averages at the interfaces

$$f^-_{i+1/2} = \frac{1}{\Delta t}\int_0^{\Delta t} f^-_{i_R} d\tau \qquad f^+_{i-1/2} = \frac{1}{\Delta t}\int_0^{\Delta t} f^+_{i_L} d\tau \tag{5.95}$$

and with $\bar{s}_{i_R,i}$ and $\bar{s}_{i,i_L}$ a suitable high order approximation of the integral for the source term inside the cell, split in two, according to notation in Figure 5.2. Notice that $\bar{s}_{i_R,i_L} = \bar{s}_{i_R,i} + \bar{s}_{i,i_L}$.

Adopting the flux-expansion ADER approach, we seek a truncated Taylor time expansion of the fluxes at the interfaces as done in (5.22). In this case, the scalar version reads

$$f^-_{i_R} = f^{-,0}_{i_R} + \sum_{k=1}^{K} f^{-,(k)}_{i_R} \frac{\tau^k}{k!} \qquad f^+_{(i+1)_L} = f^{+,0}_{(i+1)_L} + \sum_{k=1}^{K} f^{+,(k)}_{(i+1)_L} \frac{\tau^k}{k!} \tag{5.96}$$

where $f^{-,0}_{i_R}$ and $f^{+,0}_{(i+1)_L}$ represent the zero-th order approximate fluxes, obtained as a result of the resolution of the DRP_0 and $f^{-,(k)}_{i_R}$ and $f^{+,(k)}_{(i+1)_L}$ the approximate fluxes for the k-th order terms. The following expression of the numerical fluxes in (5.95) is obtained

$$f^-_{i+1/2} = f^{-,0}_{i_R} + \sum_{k=1}^{K} f^{-,(k)}_{i_R} \frac{\Delta t^k}{(k+1)!} \qquad f^+_{i+1/2} = f^{+,0}_{(i+1)_L} + \sum_{k=1}^{K} f^{+,(k)}_{(i+1)_L} \frac{\Delta t^k}{(k+1)!} \tag{5.97}$$

corresponding to the scalar version of (5.23).

The components of the power series expansion in time (5.97) are calculated by solving the corresponding RPs associated to the DRP_K. First, the leading terms of the expansion are computed from the DRP_0, which corresponds to the following RP

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial f(u)}{\partial x} = s \\ u(x,0) = \begin{cases} u^{(0)}_{i_R} & \text{if} \quad x < 0 \\ u^{(0)}_{(i+1)_L} & \text{if} \quad x > 0 \end{cases} \end{cases} \tag{5.98}$$

that includes the source term, unlike in the TT-ADER scheme. The augmented solver presented in Chapter 4 is used to obtain a linearized solution based on the states at both sides of the interface and on the contribution of the source term. First, an expression for the leading term of the numerical fluxes in (5.96) is provided by solving the DRP_0

$$\begin{aligned} f^{-,0}_{i_R} &= f^0_{i_R} \left(\frac{\widetilde{\lambda}^+}{\widetilde{\lambda}}\right)_{i+1/2} + \left[f^0_{(i+1)_L} - \bar{s}^0_{i+1/2}\right] \left(\frac{\widetilde{\lambda}^-}{\widetilde{\lambda}}\right)_{i+1/2} \\ f^{+,0}_{(i+1)_L} &= f^0_{(i+1)_L} \left(\frac{\widetilde{\lambda}^-}{\widetilde{\lambda}}\right)_{i+1/2} + \left[f^0_{i_R} + \bar{s}^0_{i+1/2}\right] \left(\frac{\widetilde{\lambda}^+}{\widetilde{\lambda}}\right)_{i+1/2} \end{aligned} \tag{5.99}$$

with $f^0_{i_R}$ and $f^0_{(i+1)_L}$ the physical fluxes at cell interfaces

$$f^0_{i_R} = f(u^0_{i_R}) \qquad f^0_{(i+1)_L} = f(u^0_{(i+1)_L}) \,, \tag{5.100}$$

$\widetilde{\lambda}_{i+1/2}$ given by the consistency condition as $\widetilde{\lambda}_{i+1/2} = \widetilde{\lambda}(u^0_{i_R}, u^0_{(i+1)_L})$ and $\bar{s}^0_{i+1/2}$ a suitable approximation of the integral of the source term across the interface

$$\bar{s}^0_{i+1/2} = \frac{1}{\Delta t} \int_0^{\Delta t} \int_{x^-_{i+1/2}}^{x^+_{i+1/2}} s_i(x,0)\; dx\; dt \tag{5.101}$$

Noticing that when dealing with non-geometric source terms, this integral is nil and therefore only the centered contribution of the source term has to be accounted for.

To obtain the high order terms of expansion in (5.96), it is necessary to find the solution for RPs given by the evolution equation for time derivatives in (5.40), where the Jacobian of the flux is considered a constant coefficient matrix. In the scalar case, Equation (5.40) can be rewritten as

$$\partial_t(\partial_t^{(k)} u) + \tilde{\lambda}\partial_x(\partial_t^{(k)} u) = \partial_t^{(k)} s \tag{5.102}$$

where $\tilde{\lambda}$ is given by

$$\tilde{\lambda} = \left.\frac{\partial f(u)}{\partial u}\right|_{t=0} \tag{5.103}$$

The Cauchy-Kowalewski procedure is used to construct time derivatives departing from the information provided by the interpolation method. It allows to express time derivatives of the flux, conserved quantity and source term at $\tau = 0$ as functions $R^{(k)}$, $D^{(k)}$ and $Q^{(k)}$, respectively, of spatial derivatives of u and s as a particular case of (5.29), (5.31) and (5.33) for scalar problems. At each side of the interface for each DRP_K, temporal derivatives of the fluxes are denoted by $R^{(k)}_{i_R}$ and $R^{(k)}_{(i+1)_L}$ and computed according to (5.30) and temporal derivatives of the conserved variable are denoted by $D^{(k)}_{i_R}$ and $D^{(k)}_{(i+1)_L}$ and computed as provided in (5.30). Using these reconstructions the K following RPs, where $k = 1, ..., K$, can be defined

$$\begin{cases} \partial_t(\partial_t^{(k)} u) + \tilde{\lambda}(u^{(0)})\partial_x(\partial_t^{(k)} u) = \partial_t^{(k)} s \\ \\ u(x,0) = \begin{cases} D^{(k)}_{(i+1)_L} & \text{if} \quad x < 0 \\ D^{(k)}_{i_R} & \text{if} \quad x > 0 \end{cases} \end{cases} \tag{5.104}$$

It is worth mentioning that the initial condition for the previous RP is formally given by $D^{(k)}_{(i+1)_L}$ and $D^{(k)}_{i_R}$, the k-th time derivatives of the conserved variable at both sides of the interface, but when constructing the solution for the approximate fluxes, only $R^{(k)}_{i_R}$ and $R^{(k)}_{(i+1)_L}$ are required. Using the augmented solver presented in Chapter 4 to solve (5.104), the approximate fluxes at the interface read

$$\begin{aligned} f^{-,(k)}_{i_R} &= R^{(k)}_{i_R} \begin{pmatrix} \widetilde{\lambda}^+ \\ \widetilde{\widetilde{\lambda}} \end{pmatrix}_{i+1/2} + \left[R^{(k)}_{(i+1)_L} - \bar{s}^{(k)}_{i+1/2} \right] \begin{pmatrix} \widetilde{\lambda}^- \\ \widetilde{\widetilde{\lambda}} \end{pmatrix}_{i+1/2} \\ f^{+,(k)}_{(i+1)_L} &= R^{(k)}_{(i+1)_L} \begin{pmatrix} \widetilde{\lambda}^- \\ \widetilde{\widetilde{\lambda}} \end{pmatrix}_{i+1/2} + \left[R^{(k)}_{i_R} + \bar{s}^{(k)}_{i+1/2} \right] \begin{pmatrix} \widetilde{\lambda}^+ \\ \widetilde{\widetilde{\lambda}} \end{pmatrix}_{i+1/2} \end{aligned} \tag{5.105}$$

with $R^{(k)}_{i_R}$ and $R^{(k)}_{(i+1)_L}$ previously defined as the time derivatives of the fluxes at the interface, with $\widetilde{\lambda}_{i+1/2}$ the wave speed for the leading term $\widetilde{\lambda}_{i+1/2} = \widetilde{\lambda}(u^0_{i_R}, u^0_{(i+1)_L})$ and with $\bar{s}^{(k)}_{i+1/2}$ the integral of the derivative of the source term across the interface

$$\bar{s}^{(k)}_{i+1/2} = \frac{1}{\Delta t} \int_0^{\Delta t} \int_{x^-_{i+1/2}}^{x^+_{i+1/2}} Q^{(k)}\; dx\; dt \tag{5.106}$$

that will be integrated by means of suitable approximations.

Centered contributions of the source term are included in the updating scheme (5.94) since the DRP is considered just at the interface, unlike in first order schemes where it was considered between two adjacent cell centers. The evolution in time of the source term is reconstructed by means of the power-series time expansion

$$s_i(x,\tau) = s_i(x,0) + \sum_{k=1}^{K} \left[\frac{\partial^k}{\partial t^k} s_i(x,0) \right] \frac{\tau^k}{k!} \tag{5.107}$$

where $\frac{\partial^k}{\partial t^k} s_i(x,0)$ is expressed in terms of spatial variations of the conserved variable and the source using $Q^{(k)}$ in (5.33) as outlined before. The centered contributions of the source term are denoted by $\bar{s}_{i,i_L}$ and $\bar{s}_{i_R,i}$ and for convenience will be expressed as a leading term plus K additional higher order terms as

$$\bar{s}_{i,i_L} = \bar{s}^0_{i,i_L} + \sum_{k=1}^{K} \bar{s}^{(k)}_{i,i_L} \qquad \bar{s}_{i_R,i} = \bar{s}^0_{i_R,i} + \sum_{k=1}^{K} \bar{s}^{(k)}_{i_R,i} \tag{5.108}$$

with

$$\bar{s}^0_{i,i_L} = \frac{1}{\Delta t} \int_0^{\Delta t} \int_{x_{i-1/2}}^{x_i} s_i(x,0)\, dx\, dt \qquad \bar{s}^0_{i_R,i} = \frac{1}{\Delta t} \int_0^{\Delta t} \int_{x_i}^{x_{i+1/2}} s_i(x,0)\, dx\, dt \tag{5.109}$$

$$\bar{s}^{(k)}_{i,i_L} = \frac{1}{\Delta t} \int_0^{\Delta t} \int_{x_{i-1/2}}^{x_i} Q^{(k)} \frac{\tau^k}{k!}\, dx\, dt \qquad \bar{s}^{(k)}_{i_R,i} = \frac{1}{\Delta t} \int_0^{\Delta t} \int_{x_i}^{x_{i+1/2}} Q^{(k)} \frac{\tau^k}{k!}\, dx\, dt \tag{5.110}$$

computed by suitable approximations of the integrals.

5.6.2 The AR-ADER scheme for systems of equations

The discussion is next extended to hyperbolic nonlinear systems of equations with source terms in 1D, given by (2.34). Assuming that the convective part of (2.34) is strictly hyperbolic, with N_λ real eigenvalues $\lambda^1, ..., \lambda^{N_\lambda}$ and eigenvectors $\mathbf{e}^1, ..., \mathbf{e}^{N_\lambda}$, it is possible define two matrices $\mathbf{P} = (\mathbf{e}^1, ..., \mathbf{e}^{N_\lambda})$ and $\mathbf{P}^{-1}$ with the property that they diagonalize the Jacobian $\mathbf{J}$, as done in (2.40).

Following the same approach than in section 5.6.1, the expression for the updating scheme is constructed as

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x} [\mathbf{F}^-_{i+1/2} - \mathbf{F}^+_{i-1/2}] + \frac{\Delta t}{\Delta x} [\bar{\mathbf{S}}_{i,i_R} + \bar{\mathbf{S}}_{i,i_L}] \tag{5.111}$$

with the numerical fluxes $\mathbf{F}^-_{i+1/2}$ and $\mathbf{F}^+_{i-1/2}$ as defined in (5.20). Adopting the flux-expansion ADER approach, we seek a truncated Taylor time expansion of the fluxes at the interfaces, as expressed in Equations (5.22) and (5.23). The time-series expansion of the fluxes is composed of the leading terms $\mathbf{F}^{-,0}_{i_R}$ and $\mathbf{F}^{+,0}_{(i+1)_L}$, obtained as a result of the resolution of the DRP_0 and of the high order terms $\mathbf{F}^{-,(k)}_{i_R}$ and $\mathbf{F}^{+,(k)}_{(i+1)_L}$, computed by solving the K RPs associated to the time derivatives defined in the DRP_K in (5.1).

The Augmented version of the Roe solver [7] (ARoe) presented in [1, 2] and previously outlined in Section 4.2 is used here to construct the numerical scheme. The ARoe solver takes into account the contribution of the source term in the solution, ensuring equilibrium between numerical fluxes and source term in steady cases.

In what follows, $\delta(\cdot)_{i+1/2}$ operator will represent the difference between the right and left state of the DRP centered in $i+1/2$ for a given variable, as $\delta(\cdot)_{i+1/2} = (\cdot)_{(i+1)_L} - (\cdot)_{i_R}$ and $\delta(\cdot)_{i-1/2} = (\cdot)_{i_L} - (\cdot)_{(i-1)_R}$. The ARoe solver is based on the decomposition of the approximate Jacobian of the homogeneous part at the initial time $\tilde{\mathbf{J}}_{i+1/2}(\mathbf{U}^{(0)}_{i_R}, \mathbf{U}^{(0)}_{(i+1)_L})$

$$\delta \mathbf{F}^{(0)}_{i+1/2} = \widetilde{\mathbf{J}}_{i+1/2} \delta \mathbf{U}^{(0)}_{i+1/2} \tag{5.112}$$

leading to a set of approximated eigenvalues $\tilde{\lambda}^m_{i+1/2}$ and eigenvectors and $\widetilde{\mathbf{e}}^m_{i+1/2} = (e^m_1, ..., e^m_{N_\lambda})^T$. The approximate Jacobian $\widetilde{\mathbf{J}}_{i+1/2}$ can be expressed as

$$\widetilde{\mathbf{J}}_{i+1/2} = \widetilde{\mathbf{P}}_{i+1/2} \mathbf{\Lambda}_{i+1/2} \widetilde{\mathbf{P}}^{-1}_{i+1/2} \tag{5.113}$$

with $\widetilde{\mathbf{P}}_{i+1/2} = \left(\widetilde{\mathbf{e}}^1, ..., \widetilde{\mathbf{e}}^{N_\lambda}\right)_{i+1/2}$ an invertible matrix composed by the eigenvectors of $\widetilde{\mathbf{J}}_{i+1/2}$ and $\mathbf{\Lambda}_{i+1/2}$ the diagonal matrix composed by the eigenvalues of $\widetilde{\mathbf{J}}_{i+1/2}$.

As done in the scalar case, the leading terms of the expansion are computed first from the DRP_0, which corresponds to the following RP

$$\begin{cases} \dfrac{\partial \mathbf{U}}{\partial t} + \dfrac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \mathbf{S} \\ \mathbf{U}(x,0) = \begin{cases} \mathbf{U}^{(0)}_{i_R} & \text{if} \quad x < 0 \\ \mathbf{U}^{(0)}_{(i+1)_L} & \text{if} \quad x > 0 \end{cases} \end{cases} \tag{5.114}$$

Following the ARoe solver, the leading terms of the numerical fluxes in (5.22) are given by as the solution of the DRP_0

$$\begin{aligned} \mathbf{F}^{-,(0)}_{i_R} &= \mathbf{F}^{(0)}_{i_R} + \textstyle\sum_{m=1}^{N_\lambda} \left(\widetilde{\lambda}^- \alpha^{(0)} - \beta^{-,(0)}\right)^m_{i+1/2} \widetilde{\mathbf{e}}^m_{i+1/2} \\ \mathbf{F}^{+,(0)}_{(i+1)_L} &= \mathbf{F}^{(0)}_{(i+1)_L} - \textstyle\sum_{m=1}^{N_\lambda} \left(\widetilde{\lambda}^+ \alpha^{(0)} - \beta^{+,(0)}\right)^m_{i+1/2} \widetilde{\mathbf{e}}^m_{i+1/2} \end{aligned} \tag{5.115}$$

with $\mathbf{F}^{(0)}_{i_R}$ and $\mathbf{F}^{(0)}_{(i+1)_L}$ the vectors of physical fluxes of the DRP_0,

$$\left(\tilde{\lambda}^\pm\right)^m_{i+1/2} = \left(\frac{\widetilde{\lambda} \pm |\widetilde{\lambda}|}{2}\right)^m_{i+1/2} \qquad \left(\beta^{\pm,(0)}\right)^m_{i+1/2} = \left(\frac{\widetilde{\lambda}^\pm}{\widetilde{\lambda}} \beta^{(0)}\right)^m_{i+1/2} \tag{5.116}$$

Notice that definition in (5.115) can be rewritten in terms of $\theta_{i+1/2}$ leading to (4.7). The wave strengths $\alpha^{(0)}$ are given by the projection of $\delta \mathbf{U}^{(0)}_{i+1/2}$ onto the Jacobian eigenvectors basis as

$$\delta \mathbf{U}^{(0)}_{i+1/2} = \widetilde{\mathbf{P}}_{i+1/2} \mathbf{A}_{i+1/2} \tag{5.117}$$

with $\mathbf{A}_{i+1/2} = \left(\alpha^{(0),1}, ..., \alpha^{(0),N_\lambda}\right)^T_{i+1/2}$ and $\beta^{(0)}$ the source strengths associated to each wave, given by

$$\bar{\mathbf{S}}^0_{i+1/2} = \widetilde{\mathbf{P}}_{i+1/2} \mathbf{B}^{(0)}_{i+1/2} \tag{5.118}$$

with $\mathbf{B}^{(0)}_{i+1/2} = \left(\beta^{(0),1}, ..., \beta^{(0),N_\lambda}\right)^T_{i+1/2}$. As in the scalar case, a suitable approximation of the integral of the source term across the interface

$$\bar{\mathbf{S}}^{(0)}_{i+1/2} = \frac{1}{\Delta t} \int_0^{\Delta t} \int_{x^-_{i+1/2}}^{x^+_{i+1/2}} \mathbf{S}(x,0)\, dx\, dt \tag{5.119}$$

must be found when dealing with geometric source terms.

The same procedure is extended to derive the expression of the derivative terms of the fluxes in (5.22). This is performed by solving the K RP's associated to the high order terms of the DRP_K using directly time derivatives of the fluxes as initial conditions.

$$\begin{cases} \dfrac{\partial}{\partial t}\left(\partial^{(k)}_t \mathbf{U}\right) + \widetilde{\mathbf{J}}_{i+1/2} \dfrac{\partial}{\partial x}\left(\partial^{(k)}_t \mathbf{U}\right) = \partial^{(k)}_t \mathbf{S} \\ \partial^{(k)}_t \mathbf{U}(x,0) = \begin{cases} \mathbf{D}^{(k)}_{i_R} & \text{if} \quad x < 0 \\ \mathbf{D}^{(k)}_{(i+1)_L} & \text{if} \quad x > 0 \end{cases} \end{cases} \tag{5.120}$$

As in the scalar case, the resulting fluxes are not computed using $\mathbf{D}^{(k)}_{i_R}$ and $\mathbf{D}^{(k)}_{(i+1)_L}$ but directly from derivatives of the fluxes at the interfaces instead. The solution for the fluxes is given by

$$\begin{aligned} \mathbf{F}^{-,(k)}_{i_R} &= \mathbf{R}^{(k)}_{i_R} + \textstyle\sum_{m=1}^{N_\lambda} \left(\alpha^{-,(k)} - \beta^{-,(k)}\right)^m_{i+1/2} \widetilde{\mathbf{e}}^m_{i+1/2} \\ \mathbf{F}^{+,(k)}_{(i+1)_L} &= \mathbf{R}^{(k)}_{(i+1)_L} - \textstyle\sum_{m=1}^{N_\lambda} \left(\alpha^{+,(k)} - \beta^{+,(k)}\right)^m_{i+1/2} \widetilde{\mathbf{e}}^m_{i+1/2} \end{aligned} \tag{5.121}$$

with

$$\left(\alpha^{\pm,(k)}\right)^m_{i+1/2} = \left(\frac{\widetilde{\lambda}^{\pm}}{\widetilde{\lambda}}\alpha^{(k)}\right)^m_{i+1/2} \qquad \left(\beta^{\pm,(k)}\right)^m_{i+1/2} = \left(\frac{\widetilde{\lambda}^{\pm}}{\widetilde{\lambda}}\beta^{(k)}\right)^m_{i+1/2} \tag{5.122}$$

The wave strengths, $\alpha^{(k)}$, are given in this case by the projection of the variation of $\mathbf{R}^{(k)}$ onto the Jacobian eigenvectors basis

$$\delta\mathbf{R}^{(k)}_{i+1/2} = \widetilde{\mathbf{P}}_{i+1/2}\mathbf{A}^{(k)}_{i+1/2} \tag{5.123}$$

with $\mathbf{A}^{(k)}_{i+1/2} = \left(\alpha^{(k),1}, ..., \alpha^{(k),N_\lambda}\right)^T_{i+1/2}$, and the same for the source strengths associated to each wave, $\beta^{(k)}$

$$\bar{\mathbf{S}}^{(k)}_{i+1/2} = \widetilde{\mathbf{P}}_{i+1/2}\mathbf{B}^{(k)}_{i+1/2} \tag{5.124}$$

with $\mathbf{B}^{(k)}_{i+1/2} = \left(\beta^{(k),1}, ..., \beta^{(k),N_\lambda}\right)^T_{i+1/2}$. A suitable approximation of the integral of the source term across the interface

$$\bar{\mathbf{S}}^{(k)}_{i+1/2} = \frac{1}{\Delta t}\int_0^{\Delta t}\int_{x^-_{i+1/2}}^{x^+_{i+1/2}} \mathbf{Q}^{(k)}\, dx\, dt \tag{5.125}$$

must be found when dealing with geometric source terms.

Centered contributions of the source term ($\bar{\mathbf{S}}_{i,i_L}$ and $\bar{\mathbf{S}}_{i_R,i}$) are included in the updating scheme (5.111). They are expressed as a leading term plus K additional higher order terms

$$\bar{\mathbf{S}}_{i,i_L} = \bar{\mathbf{S}}^{(0)}_{i,i_L} + \sum_{k=1}^{K}\bar{\mathbf{S}}^{(k)}_{i,i_L} \quad \bar{\mathbf{S}}_{i_R,i} = \bar{\mathbf{S}}^{(0)}_{i_R,i} + \sum_{k=1}^{K}\bar{\mathbf{S}}^{(k)}_{i_R,i} \tag{5.126}$$

with

$$\bar{\mathbf{S}}^{(0)}_{i,i_L} = \frac{1}{\Delta t}\int_0^{\Delta t}\int_{x_{i-1/2}}^{x_i} \mathbf{S}_i(x,0)\, dx\, dt \quad \bar{\mathbf{S}}^{(0)}_{i_R,i} = \frac{1}{\Delta t}\int_0^{\Delta t}\int_{x_i}^{x_{i+1/2}} \mathbf{S}_i(x,0)\, dx\, dt \tag{5.127}$$

$$\bar{\mathbf{S}}^{(k)}_{i,i_L} = \frac{1}{\Delta t}\int_0^{\Delta t}\int_{x_{i-1/2}}^{x_i} \mathbf{Q}^{(k)}\frac{\tau^k}{k!}\, dx\, dt \quad \bar{\mathbf{S}}^{(k)}_{i_R,i} = \frac{1}{\Delta t}\int_0^{\Delta t}\int_{x_i}^{x_{i+1/2}} \mathbf{Q}^{(k)}\frac{\tau^k}{k!}\, dx\, dt \tag{5.128}$$

computed by suitable approximations of the integrals.

Chapter 6

Numerical experiments

In this chapter, high order numerical schemes presented in this work are tested and compared when solving different test cases. First, the linear scalar equation is solved in 1D and 2D under a given velocity field and initial condition. As outlined before, when dealing with linear scalar problems the AR-ADER scheme is equivalent to the TT-ADER scheme and therefore this test case is not used to compare among numerical schemes but among reconstruction procedures instead, showing the improved performance of the proposed WENO-PW method. The second test case corresponds to the Burgers' equation with a source term of geometric nature, a nonlinear problem for which the treatment of the source term in the numerical scheme is decisive. Here, differences between augmented and non-augmented solvers are clearly noticeable. Finally, we move to systems of conservation laws and solve the one dimensional shallow water equations, proposing several test cases where the numerical schemes can be fully tested.

6.1 Resolution of the linear scalar equation

6.1.1 1D linear advection-reaction equation

The following initial condition is imposed to Equation (5.41)

$$u(x,0) = (\sin \pi x)^4 \tag{6.1}$$

and it is numerically solved inside $[a,b] \times [0,t] = [0,2] \times [0,2]$, setting CFL=0.45. Cyclic boundary conditions are imposed in all cases. It can be noticed that initial condition in (6.1) has four critical points. In this test λ is set equal to 1 and parameter ζ of the reactive term is set equal to 5. The transported function will suffer an exponential growth in time and classical first and second order numerical schemes do completely fail when simulating this test case. Very high order is mandatory if accurate solutions are searched.

The TT-ADER scheme is used here to compute the numerical solution and different WENO approaches will be compared. It is worth saying that the implementation of the AR-ADER scheme would lead to the TT-ADER algorithm, due to the linear nature of the problem and non-geometric nature of the source term.

Tables F.1, F.2 and F.3 show the numerical error and convergence rate for error norms L_1, L_2 and L_∞ respectively, using optimal reconstruction weights for different grid refinements, for the 3-rd, 5-th, 7-th, 9-th and 11-th order TT-ADER schemes. For all norms explored the TT-ADER scheme in combination with the sub-cell derivative reconstruction proposed in this work, provides the expected rate of convergence. Numerical results for the 3-rd TT-ADER scheme may seem to reproduce a suboptimal behavior, that can be easily overcome by setting further refinements. It is expected that a powerful WENO reconstruction method must reproduce the same level of error and converge rate.

When using WENO-JS method, the numerical results experience lack of precision in the numerical results due to the existence of critical points in the transported function. Results are shown in Tables F.1, F.2 and F.3. At the view of the numerical results, the computation of the 11-th TT-ADER scheme using the WENO-JS makes no sense.

When comparing the results provided by the WENO-JS and the WENO-Z methods, it can be observed that WENO-Z provides more accurate results for all error norms when using the 3-rd, 5-th, 7-th order TT-ADER schemes. When moving to higher orders the WENO-Z method provides worse results that the original WENO-JS method for all error norms, for instance L_∞ error norms are excessively large if compared with the optimal reconstruction (two orders of magnitude greater).

When analyzing the results of the L_1 norm for the WENO-PW ($b = 20$ in all cases) method, it can be seen that it emulates the numerical solution computed with the optimal reconstruction, reproducing the convergence rate specially for 9-th and 11-th order schemes. If considering the 7-th order TT-ADER scheme, L_1 errors are lower than those provided by the WENO-JS method, but better results are given if using the WENO-Z method in this case. When moving to 9-th and 11-th order TT-ADER schemes, numerical results evidence that the WENO-PW method recovers the optimal weights making the numerical scheme converge with the prescribed order, for all error norms.

We can conclude that the use of the WENO-PW reconstruction is more adequate for very high order schemes (such as 7-th, 9-th and 11-th order schemes) where a large number of stencils is used, although it can be also used for lower order schemes showing a good performance. In such cases (specially 5-th order), the WENO-Z seems to provide the best reconstruction, leading to the most accurate numerical solution among the proposed methods.

6.1.2 1D linear advection of a discontinuous function

For this test case, the reactive term of (5.41) is set to 0 leading to the scalar linear advection equation and $\lambda = 1$. A discontinuous function composed of a square, triangular, Gaussian and sinusoidal wave is used as initial condition. Figure 6.1 shows the numerical results provided by a 1-st, 3-rd, 5-th, 7-th and 9-th order TT-ADER scheme at t=2000, using the WENO-PW method and setting $b = 20$, $\Delta x = 1$ and CFL= 0.45. For all cases, the essentially non-oscillatory property is retained and spurious oscillations do not appear. It is observed that numerical diffusion is dramatically reduced when increasing the order of the numerical scheme. When analyzing the result provided by the 1-st order scheme, it can be seen that the original shape of the function is not recovered. When moving to 3-rd order numerical scheme, the shape of the function is recovered but sharp discontinuities are not accurately captured. This issue is addressed when using the 5-th, 7-th, 9-th and 11-th order numerical schemes.

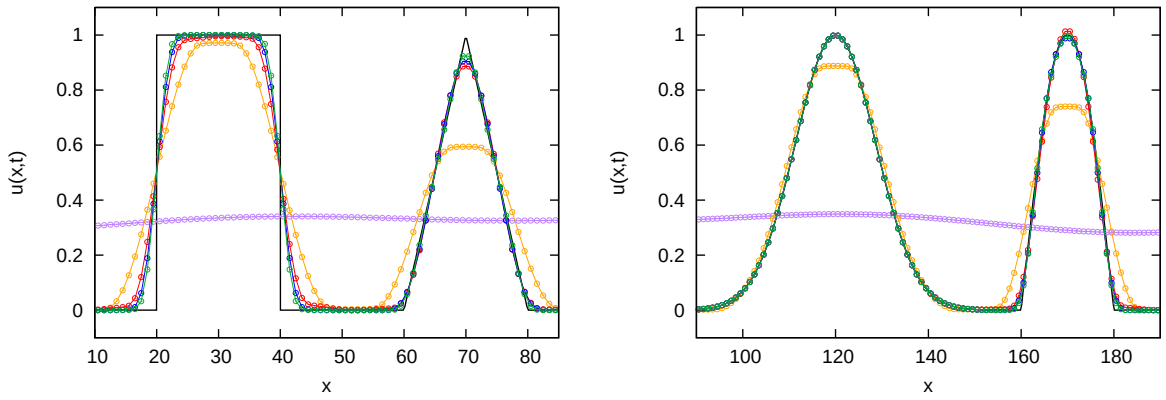

Figure 6.1: Section 6.1.2. Computational results for the advection equation with a discontinous initial condition using a 1-st (– • –), 3-rd (– • –), 5-th (– • –), 7-th (– • –) and 9-th (– • –) order TT-ADER numerical scheme and the WENO-PW method with $b = 20$. Results are compared with the exact solution (–), using a grid size $\Delta x = 1$.

6.1.3 2D linear advection of a Gaussian pulse

The following Gaussian function

$$u(x,y) = \exp\left(-\frac{(x-15)^2 + (y-15)^2}{10}\right) \tag{6.2}$$

is used as initial condition for the linear scalar Equation in (5.73), setting $\lambda_1 = \lambda_2 = 1$. It is computed using the 2D ADER numerical scheme presented in Section 5.5 inside the spatial domain $\Omega = [0, 30] \times [0, 30]$, imposing cyclic boundary conditions.

Convergence rate tests for the solution at $t = 30$ are presented in Tables F.4, F.5, F.6 and F.7 where the optimal reconstruction, traditional WENO-JS reconstruction, WENO-PW reconstruction and WENO-Z reconstruction are used respectively, in combination with the 2D ADER numerical scheme. Four refinement levels $\Delta x = \{15, 30, 60, 120\}$ have been used, setting $CFL = 0.45$. It is observed that, in general, best convergence results and lower numerical errors are achieved when using the optimal reconstruction since the initial condition is smooth and the problem does not lead to discontinuous solutions. When using the WENO-PW method, good convergence results and low numerical errors are still preserved, unlike for the WENO-JS, that leads to higher numerical errors although the convergence rate is roughly maintained. Notice that the convergence rate using L_2 error norm is higher than when using L_1 or L_∞ error norms, even leading to a faster convergence rate than it is prescribed. On the other hand, L_1 and L_∞ error norms do converge at barely the prescribed convergence rate. The use of the WENO-Z reconstruction is reported to provide a less accurate solution (higher numerical errors) as the order of the numerical scheme is increased. Notice that the numerical errors provided by the numerical scheme when using the WENO-PW reconstruction and $N = 120$ coincide with those provided by the same scheme when the WENO-Z reconstruction is used.

Figure 6.2 shows the numerical solution at $t = 60$ provided by a 1-st order Godunov scheme and by the 3-rd, 5-th and 7-th order 2D ADER numerical schemes in combination with the WENO-PW method, with a grid of size 30×30 cells and setting $CFL = 0.45$.

6.1.4 2D linear advection with space-dependent coefficients: Doswell frontogenesis

The kinematic approach to frontogenesis proposed by Doswell [41] provides a reliable benchmark for numerical models in meteorology. In [41], an idealized model of a vortex interacting with a initially straight frontal zone was developed. Local advection and frontogenesis were calculated analytically at the initial time and used to find the evolution of the system in time. In [40], an analytical solution for the advected scalar at a given time was obtained by solving the linear transport PDE for the scalar. Both publications explore the frontogenesis solution for a general nondivergent vortex flow.

In the present work, we use those results to reproduce numerically the advection of a scalar quantity under the effect of the frontogenesis using the 2D ADER scheme in Section 5.5. Numerical results are compared with the exact solution derived in the mentioned publications.

The kinematic model proposed in [41] consists of a hyperbolic vortex that represents a smooth approximation to the Rankine combined vortex. It is worth mentioning that in many studies, the flow in real atmospheric vortices has been assumed to fit the Rankine Combined Vortex. The hyperbolic vortex in [41] is given by the following velocity profile in polar coordinates

$$\mathbf{v}(r,\theta) = \begin{pmatrix} 0 \\ V_T(r) \end{pmatrix} \tag{6.3}$$

where $V_T(r)$ represents a tangential wind given by

$$V_T(r) = V_{max}\,\mathrm{sech}^2(r)\tanh(r) \tag{6.4}$$

with $V_{max} = 2.5980762$ in order to normalize the maximum value of the wind profile. When expressing the velocity field on a cartesian coordinate system, it reads

$$\mathbf{v}(x,y) = \begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix} = \begin{pmatrix} -V_T(r)\frac{y}{r} \\ V_T(r)\frac{x}{r} \end{pmatrix} \tag{6.5}$$

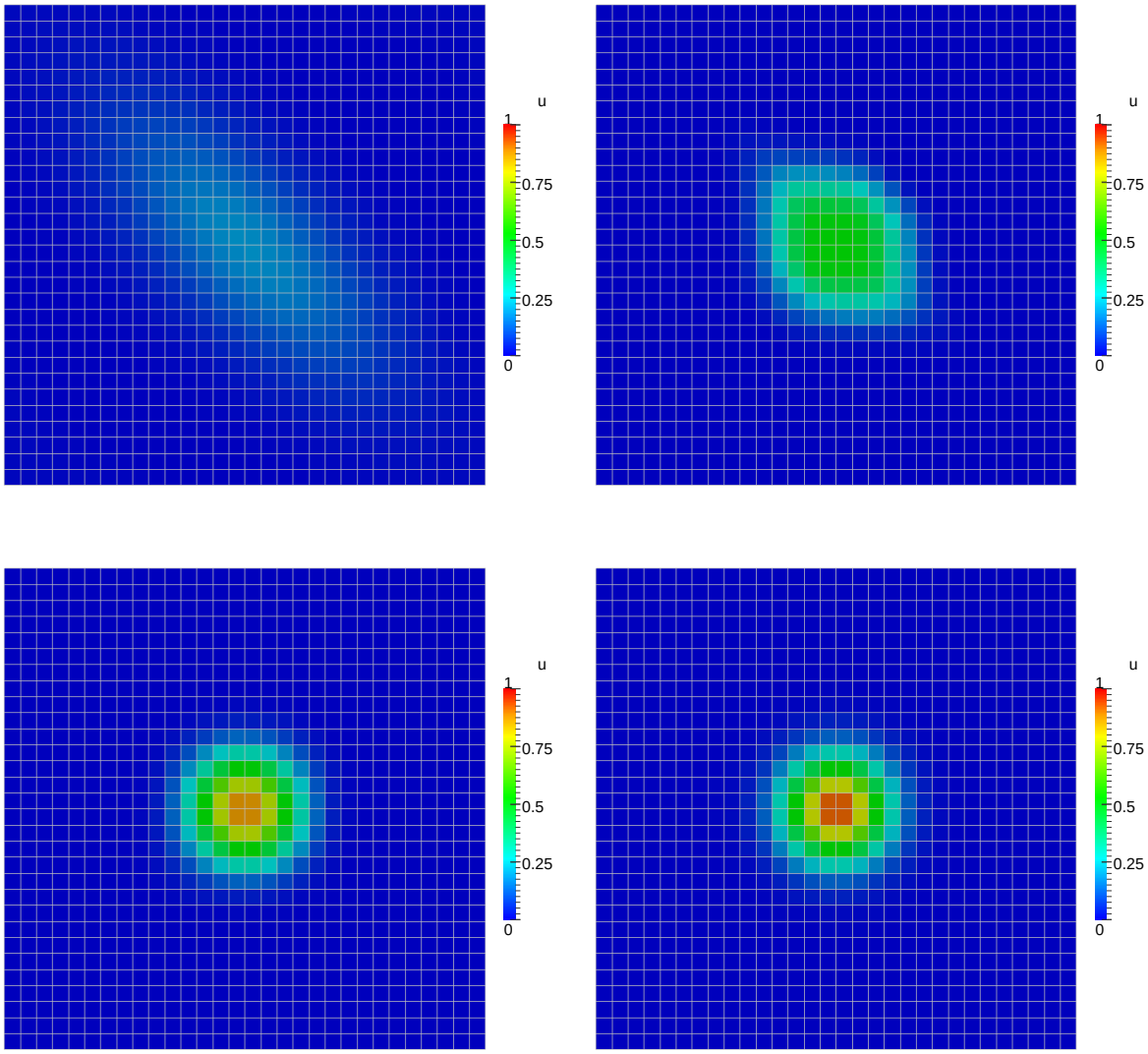

Figure 6.2: Numerical solution for the advection of the gaussian pulse at $t = 60$, using a 1-st order Godunov scheme and the 3-rd, 5-th and 7-th order 2D ADER numerical schemes. The computational grid is composed of 30×30 cells and CFL number is set to 0.45.

with $r = \sqrt{x^2 + y^2}$. The kinematic properties of the vortex field can be analyzed by studying the linear representation of the velocity field, using the first order Taylor series expansion

$$\mathbf{v}(x_0 + \delta x, y_0 + \delta y) = \mathbf{v}(x_0, y_0) + \nabla(\mathbf{v}) \cdot (\delta x, \delta y)^T \tag{6.6}$$

where $\nabla(\mathbf{v})$ is the gradient of the velocity vector, with components $\frac{\partial u_i}{\partial x_j}$, that can be expressed as

$$\nabla(\mathbf{v}) = \mathbf{D} + \mathbf{R} \tag{6.7}$$

where $\mathbf{D} \in \mathbb{R}^{2\times 2}$ is the deformation matrix with components $d_{i,j} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$ and $\mathbf{R} \in \mathbb{R}^{2\times 2}$ is the rotation matrix with components $r_{i,j} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right)$, with $i, j = 1, 2$, $u_1 \equiv u$, $u_2 \equiv v$, $x_1 \equiv x$, $x_2 \equiv y$. From that, it is straightforward to notice [41] that for the hyperbolic vortex flow (6.5)

$$\mathbf{D} = \begin{pmatrix} \beta/2 & \alpha/2 \\ \alpha/2 & -\beta/2 \end{pmatrix}, \qquad \mathbf{R} = \begin{pmatrix} 0 & \gamma/2 \\ -\gamma/2 & 0 \end{pmatrix} \tag{6.8}$$

with

$$\alpha = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = V_{max}\,\mathrm{sech}^2(r)\left(\mathrm{sech}^2(r) - 2\tanh^2(r) - \frac{\tanh(r)}{r}\right)\cos(2\theta) \tag{6.9}$$

$$\beta = 2\frac{\partial u}{\partial x} = -2\frac{\partial v}{\partial y} = V_{max}\,\mathrm{sech}^2(r)\left(\frac{\tanh(r)}{r} - \frac{1}{2}\right)\sin(2\theta) \tag{6.10}$$

$$\gamma = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} = -V_{max}\,\mathrm{sech}^2(r)\left(\mathrm{sech}^2(r) - 2\tanh^2(r) + \frac{\tanh(r)}{r}\right) \tag{6.11}$$

being $-\gamma$ the component of the vorticity vector $\nabla \times \mathbf{v}$ normal to the $x-y$ plane. It is worth mentioning the incompressible (non-divergent) characteristic of the flow, noticed as $\nabla \cdot \mathbf{v} = \mathrm{tr}(\mathbf{D}) = \beta/2 - \beta/2 = 0$. The vector field associated to the vortex in (6.5), centered at $(x, y) = (5, 5)$, is depicted in Figure 6.3, including a longitudinal cut of the velocity magnitude along a line passing by the center.

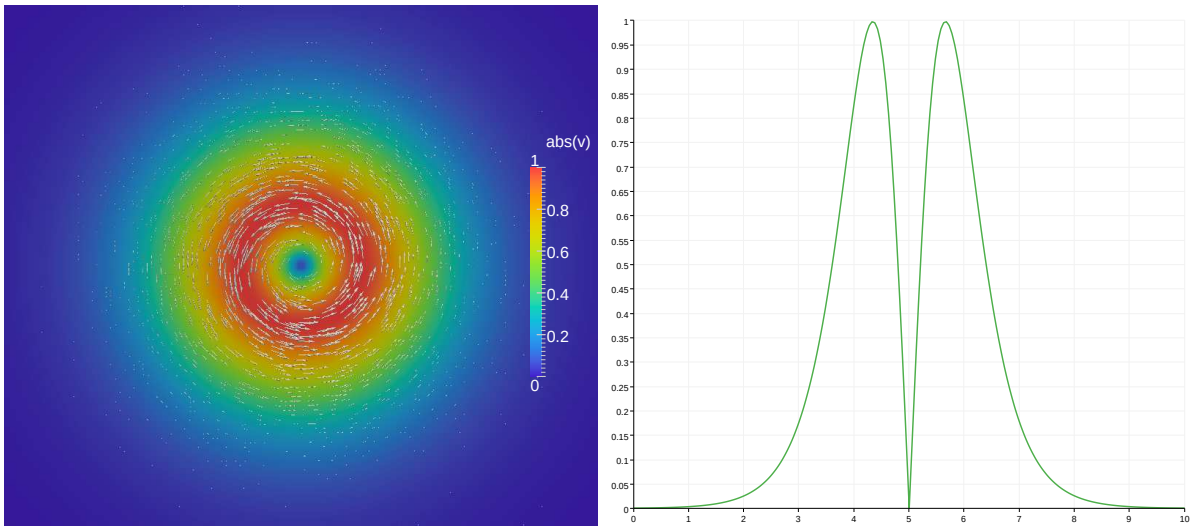

Figure 6.3: Left: vector field and plot of the magnitude of $\mathbf{v}$ inside the computational domain. Right: plot of the tangential velocity along the radial direction, with the vortex centered at $(x, y) = (5, 5)$.

Once the kinematic model has been studied, the initial condition for the scalar quantity $u = u(x, y, t)$ (e.g. temperature) must be provided. In [41], the following initial condition is proposed

$$u(x, y, 0) = \tanh\left(\frac{y}{\delta}\right), \tag{6.12}$$

modelling a straight front configuration. The evolution in time of the scalar field $u(x, y, t)$ with initial condition in (6.12) under the action of the vortex is given by the following PDE

$$\begin{cases} \dfrac{\partial u}{\partial t} + \lambda_1 \dfrac{\partial u}{\partial x} + \lambda_2 \dfrac{\partial u}{\partial y} = 0 \qquad x, y \in \Omega \subseteq \mathbb{R}^2\,,\ t \geq 0 \\ \\ u(x, y, 0) = \tanh\left(\dfrac{y}{\delta}\right) \end{cases} \tag{6.13}$$

where $\lambda_1 = u(x, y)$ and $\lambda_2 = v(x, y)$ are the components of the velocity vector in (6.5).

Problem in (6.13) is solved using the 2D ADER numerical scheme in combination with the WENO-PW reconstruction inside the spatial and temporal domains $\Omega = [0, 10] \times [0, 10]$ and $t \in [0, T]$ respectively, with the velocity field centered at $(x, y) = (5, 5)$. Parameter δ is set to 10^{-6}. Results of the computation of (6.13) using a 1-st order Godunov scheme and the 3-rd, 5-th and 7-th order 2D ADER numerical schemes at $t = 4$ and $t = 6$ are included in Figures 6.4 and 6.5 respectively, using $CFL = 0.45$ and a grid of 201 cells in each coordinate direction. It is observed that numerical diffusion is drastically reduced

when moving from a 1-st order scheme to a 3-rd order ADER scheme. As the order of the numerical scheme is increased, the discontinuous solution of the frontogenesis is more accurately captured. The evolution in time of the solution is shown in `https://youtu.be/U_zN7bluqYQ` .

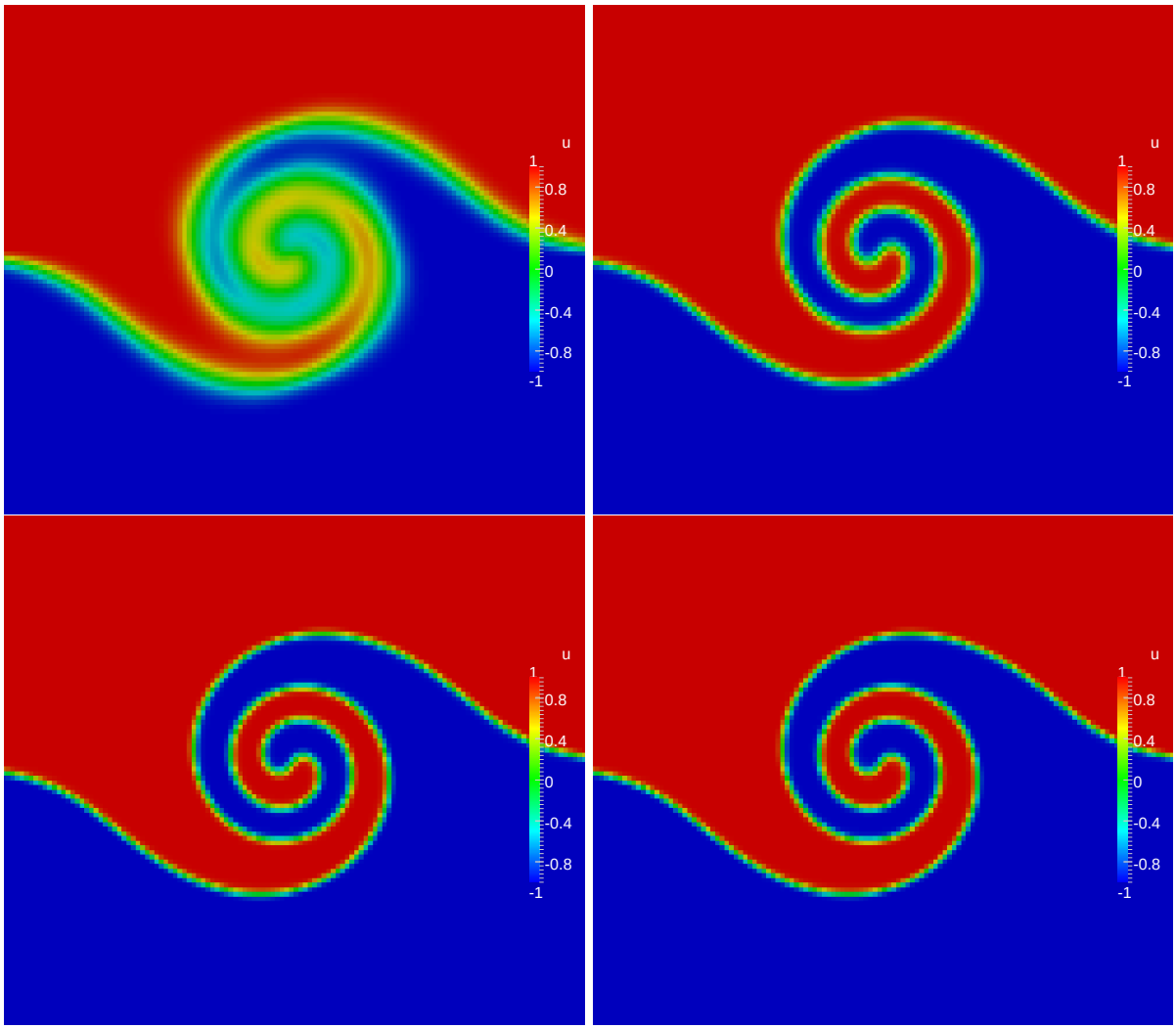

Figure 6.4: Numerical results for the Doswell frontogenesis test case in (6.13) at $t = 4$, using a 1-st order Godunov scheme and the 3-rd, 5-th and 7-th order 2D ADER numerical schemes. The computational grid is composed of 201×201 cells and CFL number is set to 0.45.

Longitudinal cuts in the y-direction at $x = 5$ of the solutions at $t = 4$ and $t = 6$ provided by a 1-st order Godunov scheme and the 3-rd, 5-th and 7-th order 2D ADER numerical schemes are presented in Figure 6.6, including the exact solution. It is observed that discontinuities are more accurately captured when increasing the order of the numerical scheme. Some oscillations are noticed when computing the solution using the 5-th order ADER scheme at $t = 6$, due to the fact that more than one discontinuity of the solution is included in the stencil of the non-oscillatory reconstruction, as reported in [35]. This issue appears to be masked when using the 7-th order ADER scheme, probably due to the even number of stencils.

6.2 Resolution of Burgers' equation

In this section, numerical results of the proposed AR-ADER method are presented. All test cases have been performed with the following Burgers' equation with source term in [4]

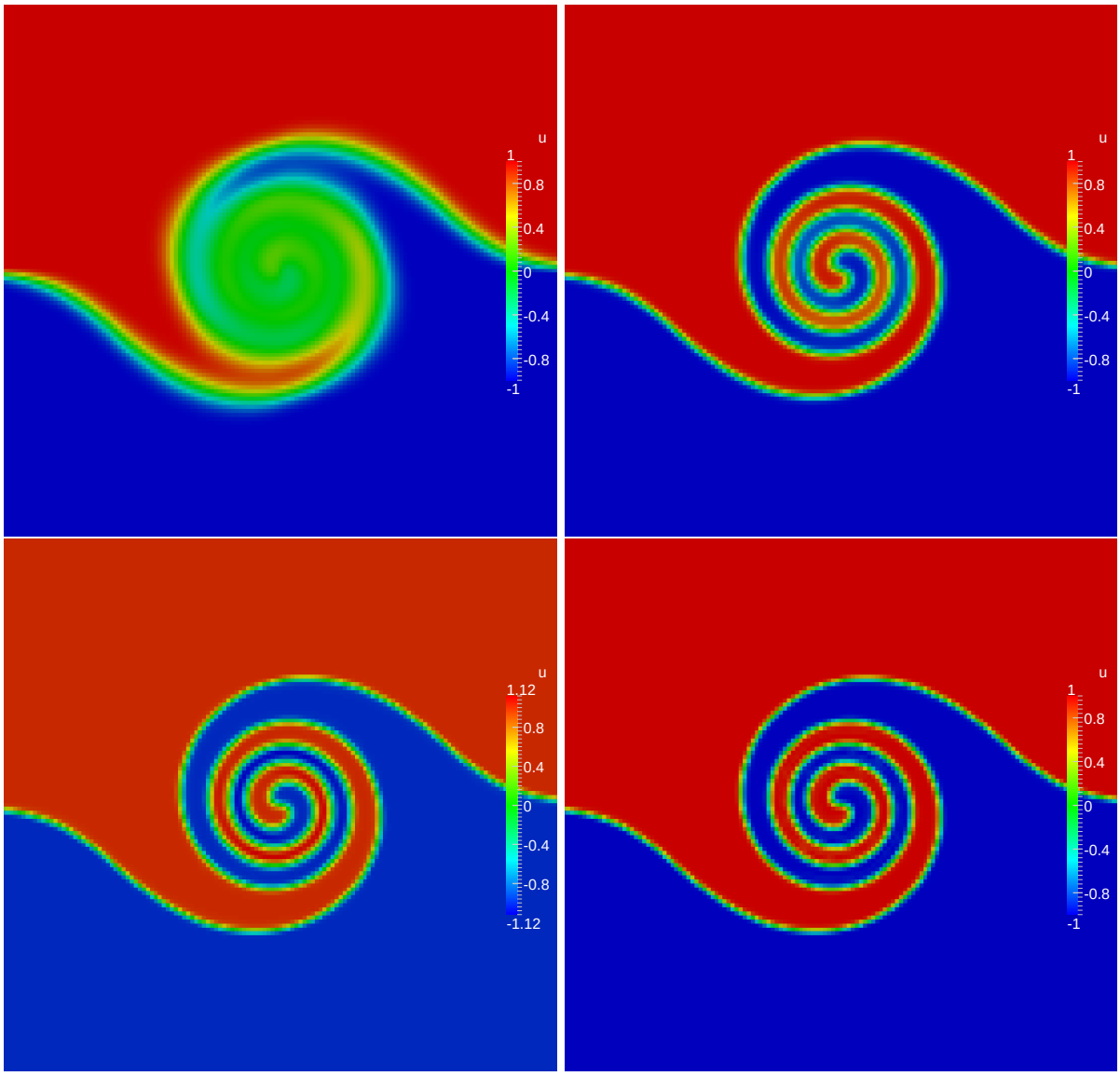

Figure 6.5: Numerical results for the Doswell frontogenesis test case in (6.13) at $t = 6$, using a 1-st order Godunov scheme and the 3-rd, 5-th and 7-th order 2D ADER numerical schemes. The computational grid is composed of 201×201 cells and CFL number is set to 0.45.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -u\frac{\partial z}{\partial x} \tag{6.14}$$

where $u = u(x,t)$ is the computed variable and $z = z(x)$ is provided. The solution of (6.14) under steady state conditions, $\partial_t u = 0$, leads to a equilibrium among fluxes and source terms, that involves the definition of a new variable $e = u + z$, constant in space and time.

When moving to the discrete approximation, it must be verified that the numerical scheme is able to keep the initial equilibrium of the solution in time. The formulation of the AR-ADER method presented in this work in (5.94) is designed to enforce discrete equilibrium in steady state, but also to converge to the exact solution. To ensure both properties, the approximations made over the the source terms in (5.106), (5.109), (5.110) and their appearance in the weak solution presented, are decisive.

Considering that the updating expression to compute the cell average value u^{n+1} can be written as a sum of contributions of the form $\delta m = \delta\bar{f} - \bar{s}$, in equilibrium, all these δm contributions must become zero. This condition is enforced here for an arbitrary interval $[x_I, x_J]$ when defining the source term in the DRP_0

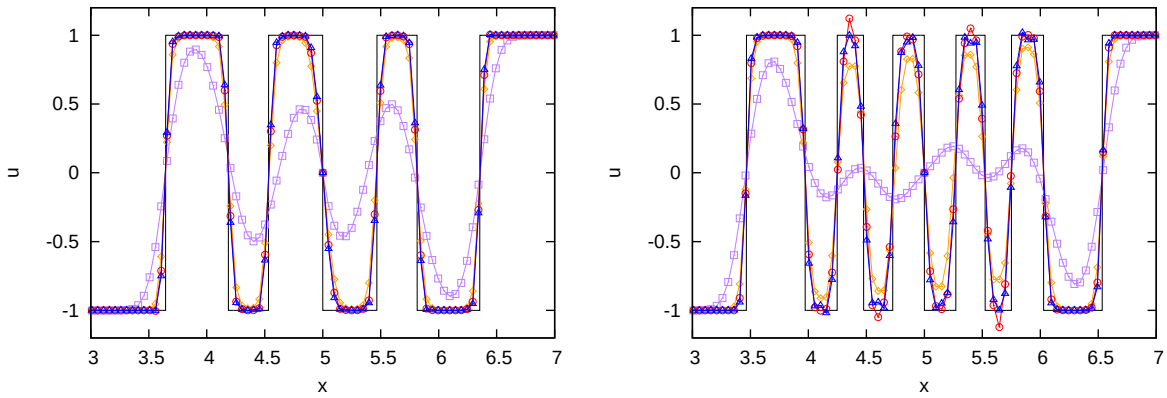

Figure 6.6: Numerical solution for the Doswell frontogenesis in (6.13) at $t = 4$ (left) and $t = 6$ (right) along the y-axis, using a 1-st order Godunov scheme and the 3-rd, 5-th and 7-th order 2D ADER numerical schemes. The computational grid is composed of 201 cells in the y-direction.

$$\delta m^0_{J,I} = \left(\delta \bar{f}^0 - \bar{s}^0\right)_{J,I} = \left(\frac{u^0_J + u^0_I}{2}\right)\left(u^0_J - u^0_I\right) - \bar{s}^0_{J,I} = 0 \tag{6.15}$$

by approaching integrals in (5.106) and (5.109) as follows

$$\bar{s}^0_{J,I} \approx -\left(\frac{u^0_J + u^0_I}{2}\right)\left(z^0_J - z^0_I\right) \tag{6.16}$$

When using definition in (6.16) equation (6.15) becomes

$$\delta m^0_{J,I} = \left(\frac{u^0_J + u^0_I}{2}\right)\delta e^0_{J,I} \tag{6.17}$$

ensuring equilibrium. Higher order terms of the contributions in each δm are expressed by

$$\delta m^{(k)}_{J,I} = \left(\delta f^{(k)} - \bar{s}^{(k)}\right)_{J,I} = (R^{(k)}_J - R^{(k)}_I)\frac{\Delta t^k}{(k+1)!} - \frac{1}{\Delta t}\int_0^{\Delta t}\int_{x_I}^{x_J} Q^{(k)}\frac{\tau^k}{k!}\,dx\,dt \tag{6.18}$$

where $Q^{(k)}$ and $R^{(k)}$ are defined using the Cauchy-Kowalewski procedure. The two first time derivatives of f are given by

$$R^{(1)} \equiv \partial_t f = -u^2 e_x \tag{6.19}$$

$$R^{(2)} \equiv \partial_{tt} f = u^2 e_x^2 + u^2\left(2e_x^2 - z_x e_x + u e_{xx}\right) \tag{6.20}$$

Temporal derivatives have been derived up to 4-th order for the Burger's equation in this work. Time derivatives of the source term are reduced to find time derivatives of u, $u^{(k)} = \partial^k_t u(x,t)$, as parameter z is constant in time. Therefore, function $Q^{(k)}$ is given by

$$Q^{(k)} = -u^{(k)} z_x \tag{6.21}$$

and will be expressed as a function of spatial derivatives of e. The two first time derivatives of s are

$$u^{(1)} = -u e_x \tag{6.22}$$

$$u^{(2)} = u\left(2e_x^2 - z_x e_x + u e_{xx}\right) \tag{6.23}$$

multiplied by spatial derivatives of e. This observation can be extended to any k-th time derivative for both s and f. If equilibrium is enforced, all time derivatives of e must be zero in this particular case. Depending on data reconstruction procedure, numerical computation of time derivatives may lead to non-zero results. In order to have nil time derivatives and nil contributions of the leading term under steady conditions, the reconstruction procedure must be carried out for the variable e instead of u.

High order terms of the integral of the source term at cell interfaces, $\bar{s}^{(k)}_{i+1/2}$, are approximated by the expression

$$\bar{s}^{(k)}_{i+1/2} \approx -\left(\frac{u^{(k)}_{(i+1)_L} + u^{(k)}_{i_R}}{2}\right)\left(z^0_{(i+1)_L} - z^0_{i_R}\right) \tag{6.24}$$

following the same approach done for the leading terms.

The performance of the AR-ADER method proposed in this work is compared with the MUSCLS method with source terms in [4] in combination with the minmod slope limiter or the superbee limiter [45] and with the TT-ADER scheme method [35].

6.2.1 RP with a right moving shock.

In this test case the following RP is defined. The initial data is

$$u(x,0) = \begin{cases} 2.0 & if \quad x < 0 \\ 1.0 & if \quad x > 0 \end{cases} \qquad z(x,0) = \begin{cases} 0 & if \quad x < 0 \\ 0.5 & if \quad x > 0 \end{cases} \tag{6.25}$$

and the exact solution consists of a steady discontinuity at $x = 0$ plus a linear shock traveling with velocity $\lambda = 1.25$, connecting an intermediate state $u^* = 1.5$ with the right initial state [1]. Numerical results are plotted in Figure 6.7 (left) at time $t = 5s$ using $CFL = 0.8$ and Δx=0.02. The simulation times used in the different RP's presented in this work are selected avoiding the interference of the boundary cells with the evolution of the solution, so imposition of boundary conditions are not necessary. The MUSCLS and the 1st, 3rd and 5th order AR-ADER methods do converge to the exact solution. When the order of the AR-ADER method is increased the shock is more accurately captured. When using the TT-ADER scheme the intermediate state u^* does not appear in the numerical solution since the source term is accounted for as a cell-centered contribution, and therefore it is unable to converge to the exact solution.

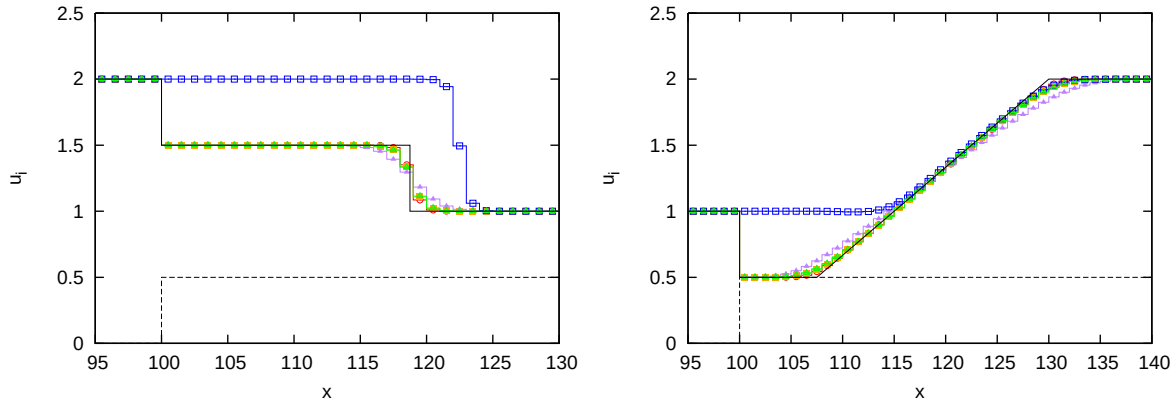

Figure 6.7: Sections 6.2.1 and 6.2.2.. Exact (—) and numerical solutions at $t = 15$ using a 1-st (−▲−) , 3rd (−■−) and 5th order AR-ADER method (− ○ −), 3rd order TT-ADER scheme (−□−) and the MUSCLS (superbee) method (−△−).

6.2.2 RP with a right moving rarefaction wave.

In this test case a different type of RP is defined, with the following initial data

$$u(x,0) = \begin{cases} 1.0 & if \quad x < 100 \\ 2.0 & if \quad x > 100 \end{cases} \qquad z(x,0) = \begin{cases} 0 & if \quad x < 100 \\ 0.5 & if \quad x > 100 \end{cases} \tag{6.26}$$

The weak self-similar solution consists of a steady discontinuity plus a right moving rarefaction wave expanding with a velocity x/t between two constant states $u^* = 0.5$ and with the right initial state [1]. Figure (6.7) (right) shows the numerical results at $t = 5s$ using $CFL = 0.8$ and Δx=0.02, using the different methods proposed above. When the order of the method is increased a better approximation of the rarefaction wave is obtained. The TT-ADER scheme is unable to converge to the exact solution, as in the previous case.

6.2.3 Smooth initial conditions with cyclic boundary conditions

In the previous sections, the performance of the numerical method has been tested in transient cases with large discontinuities. The convergence rate of the proposed numerical scheme is tested in a case with smooth initial condition. In order to compute the exact solution, (6.14) is expressed as

$$\frac{D}{Dt}(e) = \frac{\partial e}{\partial t} + u\frac{\partial e}{\partial x} = 0 \tag{6.27}$$

Analytical solutions of (6.27) are computed using numerical integration of very high order. The initial condition is given by $z(x) = 0.1\sin^4(\pi x)$ and $u(x,0) = 1.0$ in a domain $[0,2]$. Numerical results are computed setting $CFL = 0.4$ and Δx=0.05. Cyclic boundary conditions are imposed.

Error norms and convergence rates are shown in Table F.8 at $t = 0.1$s. At this time the solution remains smooth. The prescribed order of accuracy is reached for the AR-ADER method up to 5-th order. Second order MUSCLS scheme does operate as expected too. Although the TT-ADER scheme reaches the expected order of convergence for the L_2 error norm, difficulties appear for L_1 and L_∞ error norms. Convergence rates are not sensitive to variations of the CFL number.

6.3 Application to the Shallow Water Equations

In this section, the proposed AR-ADER method is applied to the shallow water equations

$$\mathbf{U} = \begin{pmatrix} h \\ q \end{pmatrix}, \ \mathbf{F} = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{pmatrix}, \ \mathbf{S} = \begin{pmatrix} 0 \\ S_z \end{pmatrix}, \tag{6.28}$$

where h is the water depth, u is the depth averaged velocity and g is the acceleration of gravity. The source term S_z involves the variations in bed geometry S_z

$$S_z = -gh\frac{dz}{dx} \tag{6.29}$$

where z represents the bed elevation. For the sake of simplicity, shear stress will be neglected in this work.

When applied to the shallow water equations, the Augmented Roe solver provides a linearized solution that can be straightforward expanded from the homogeneous case. The approximate Jacobian $\widetilde{\mathbf{J}}$ of the homogeneous part is given by [7]

$$\tilde{\mathbf{J}}_{i+1/2} = \begin{pmatrix} 0 & 1 \\ \widetilde{c}^2 - \widetilde{u}^2 & 2\widetilde{u} \end{pmatrix}_{i+1/2} \qquad \delta\mathbf{F}_{i+1/2} = \widetilde{\mathbf{J}}_{i+1/2}\delta\mathbf{U}_{i+1/2} \tag{6.30}$$

where

$$\begin{array}{cc} \widetilde{\lambda}^1 = \widetilde{u} - \widetilde{c} & \widetilde{\lambda}^2 = \widetilde{u} + \widetilde{c} \\ \widetilde{\mathbf{e}}^1 = \begin{pmatrix} 1 \\ \widetilde{u} - \widetilde{c} \end{pmatrix} & \widetilde{\mathbf{e}}^2 = \begin{pmatrix} 1 \\ \widetilde{u} + \widetilde{c} \end{pmatrix} \end{array} \tag{6.31}$$

with

$$\widetilde{c} = \sqrt{g\frac{h_{i_R}^{(0)} + h_{(i+1)_L}^{(0)}}{2}} \quad \widetilde{u} = \frac{u_{(i+1)_L}^{(0)}\sqrt{h_{(i+1)_L}^{(0)}} + u_{i_R}^{(0)}\sqrt{h_{i_R}^{(0)}}}{\sqrt{h_{(i+1)_L}^{(0)}} + \sqrt{h_{i_R}^{(0)}}} \tag{6.32}$$

Numerical discretization must be performed to ensure convergence to the exact solutions. As in the scalar case, the definition of the numerical scheme as a sum of updating contributions in provides a suitable procedure, as under steady conditions all terms must become nil.

In order to extend the well balanced property for static equilibrium to ensure exact equilibrium in all steady states and geometries, the numerical approximation done over the integral of the source term, S_z, that will referred to as $\bar{S}_z^{(0)}$ for the leading term, is that proposed in [39]. This approach is first applied to the leading term. The method evaluates the discrete source term at the cell interfaces in order to ensure the energy balance property. It proposes a combination of two alternatives: one possibility is to compute $\bar{S}_z$ considering a smooth variation of the variables inside an arbitrary interval $[x_I, x_J]$ as

$$\bar{S}_z^a = -g\tilde{h}\delta z \tag{6.33}$$

with $\tilde{h} = 1/2(h_J^{(0)} + h_I^{(0)})$, $\delta z = z_J - z_I$ and second possibility is to define S_z^b as

$$\bar{S}_z^b = -gh_j\delta z \tag{6.34}$$

where

$$h_j = \begin{cases} h_I^{(0)} & \text{if} \quad \delta h > 0 \\ h_J^{(0)} & \text{if} \quad \delta h \leq 0 \end{cases} \tag{6.35}$$

with $\delta h^{(0)} = h_J^{(0)} - h_I^{(0)}$. In cases of still water with a continuous water level surface both estimations in (6.33) and (6.34) ensure quiescent equilibrium. In this particular case hydrostatic forces are equilibrated exactly. The second possibility is to evaluate the source term using the following linear combination

$$\bar{S}_z^{(0)} = (1 - \mathcal{A})S_z^a + \mathcal{A}S_z^b \tag{6.36}$$

where $0 \leq \mathcal{A} \leq 1$. The calculation of $\mathcal{A}$ is detailed in [39] and discriminates between smooth solutions and transcritical jumps.

In cases of still water with a continuous/discontinuous water level surface (which is a particular case of energy conservation) quiescent equilibrium is guaranteed, while in steady cases with smooth solutions exact conservation of energy is preserved. In presence of hydraulic jumps, the numerical discretization proposed in [39] only considers momentum conservation and energy is dissipated at the correct rate. With this numerical technique it is possible to evaluate the different source strengths, $\beta^{(0),m}$ ensuring an exact balance between fluxes and source terms.

Considering that steady solutions are provided by a constant value of water discharge and under smooth conditions by a constant value of mechanical energy, in order to preserve the energy-balanced property, the spatial reconstruction is carried out for the specific mechanical energy, E, and the unitary discharge, q,

$$E = \frac{u^2}{2g} + h + z \qquad q = hu \tag{6.37}$$

Then, when necessary, h and u are computed departing from spatial reconstructions of E and q.

In order to compute the high order terms, the expression of time derivatives of the fluxes and source in terms of spatial derivatives of the main variables is obtained applying the Cauchy-Kowalewski theorem. A proper execution of this procedure is crucial to ensure a good performance of the proposed numerical scheme. For this purpose, it is advised to get a final expression for time derivatives of the fluxes and source term given by a sum of products in which spatial derivatives of E and q are always present.

Therefore, under steady conditions, functions $\mathbf{R}^{(k)}$ and $\mathbf{Q}^{(k)}$ are enforced to be zero since there is no spatial variation of energy or discharge, ensuring the equilibrium state for higher order terms.

With this information, it is possible to compute the contribution of the source term across the cell edge in (5.27), by means of (6.36) for the leading term and

$$\bar{S}_{z\,i+1/2}^{(k)} = \frac{1}{\Delta t}\int_0^{\Delta t}\int_{x_{i+1/2}^-}^{x_{i+1/2}^+} -g\,h^{(k)}\,\frac{dz}{dx}dx \approx -g\left(\frac{h_{(i+1)_L}^{(k)} + h_{i_R}^{(k)}}{2}\right)\delta z \tag{6.38}$$

for high order terms, with $\delta z = z_{(i+1)_L} - z_{i_R}$. Other suitable approximations are possible if using spatial derivatives of both energy and discharge.

6.3.1 Steady flow over a hump

The energy balanced approach in [39] ensures a constant value of mechanical energy, providing the exact solutions at every computational cell, with independence of the grid refinement under steady conditions. The numerical scheme presented here preserves this property, as high derivative terms become nil in stationary conditions. Numerical results up to 5th order are presented here for three different configurations of steady flow widely used as reference test cases in a wide number of works. Three different configurations of steady flow are defined in a domain 25 m long, where in all cases the bed level is given by

$$z(x) = \begin{cases} 0 & if \quad x < 8 \\ 0.2 - 0.05(x-10)^2 & if \quad 8 \leq x \leq 12 \\ 0 & if \quad x > 12 \end{cases} \tag{6.39}$$

The first considers subcritical flow in all the domain. Numerical solutions are generated in this test case by imposing a constant discharge $q = 4.42$ m^2/s upstream and a constant water depth $h = 2m$ downstream the channel. A mesh refinement of $\Delta x = 0.25$ m is used in the results presented in Figure 6.8, where the numerical solutions provided by the energy balanced 1st, 3rd and 5th order AR-ADER methods are plotted. As expected, the numerical scheme reproduces the exact solution for all orders.

In the second test case, the flow passes from subcritical conditions to supercritical conditions, leading to a transcritical flow without shocks, maintaining a constant value of total head energy in the entire domain. The transition from subcritical to supercritical conditions is present at $z = 0.2$ m. Flow conditions evolve from subcritical to supercritical conditions, therefore, only the unit discharge, q=0.18m^3/s, is imposed upstream the domain. Numerical solutions provided by the AR-ADER method reproduce the exact solution for all orders in the whole computational domain as plotted in Figure 6.9.

The third test case considers transcritical flow followed by a shock over the hump, where energy is dissipated. The total head energy before the hydraulic jump depends on the water discharge imposed upstream and the total head energy after the hydraulic jump is determined by the water depth imposed downstream, $q = 0.18$ m^2/s and $h = 0.33$ m respectively in this case. Figure 6.10 shows the numerical solutions provided by the AR-ADER method. In all cases the position of the shock is correctly computed. The numerical solutions reproduce the exact solution, except in the cell where the hydraulic jump is developed.

6.3.2 Numerical performance in RP

Following [39] comparisons between exact solutions of the Riemann problem for system (6.28) and numerical solutions obtained using the AR-ADER method are presented. The examples involve different combinations of wave patterns in presence of bed discontinuities and are summarized in Table 6.1. Test cases 2 and 3 are included in a list of RPs defined by LeFloch and Duc-Thanh [47]. The domain is defined by $[-1, 1]$ m, the bottom step is positioned at $x = 0$, has a variable height and g=9.8 m/s^2. The domain is divided in 500 cells and 1000 cells . Numerical solutions are plotted at time $t = 0.01$ s and the time step is computed using CFL = 0.2.

RPs 1 to 3 are non-resonance problems and admit only one solution. RP 1, Figure 6.11, is a dam-break type problem that contains a left moving rarefaction wave, a stationary shock at the step and a

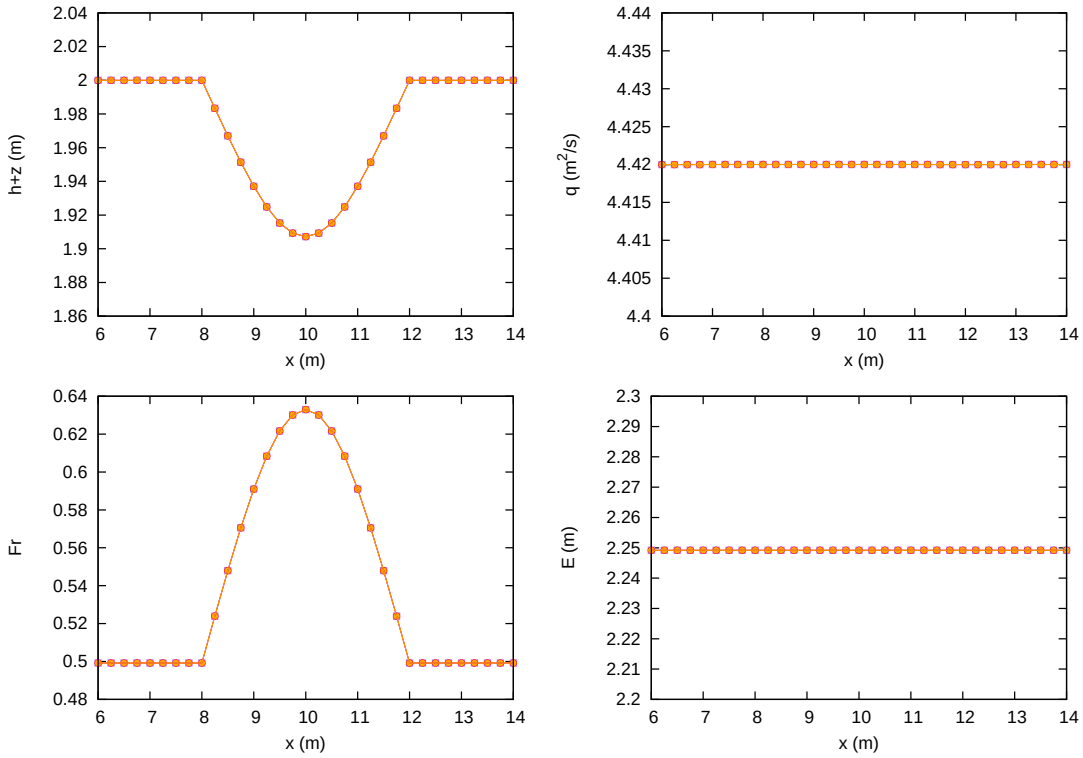

Figure 6.8: Section 6.3.1. Subcritical flow. Exact solution (—) and numerical solutions using 1st (–□–), 3rd (– ● –) and 5th (– ● –) order AR-ADER method. $\Delta x = 0.25$.

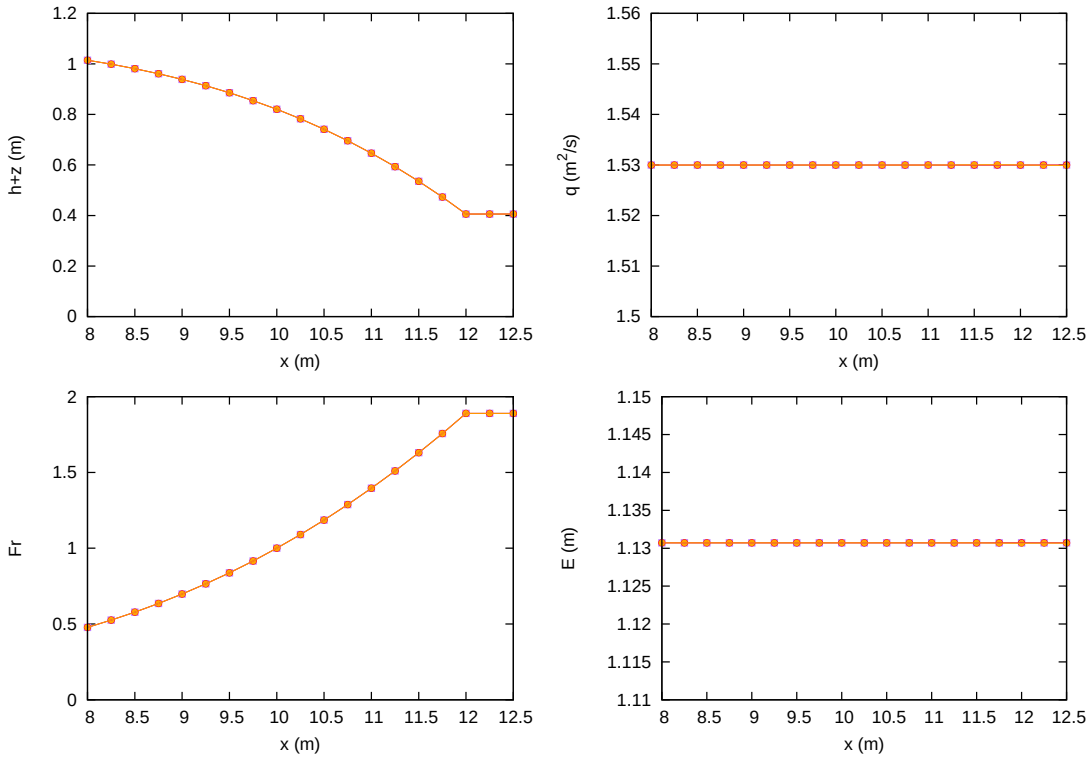

Figure 6.9: Section 6.3.1. Transcritical flow. Exact solution (—) and numerical solutions using 1st (–□–), 3rd (– ● –) and 5th (– ● –) order AR-ADER method. $\Delta x = 0.25$m.

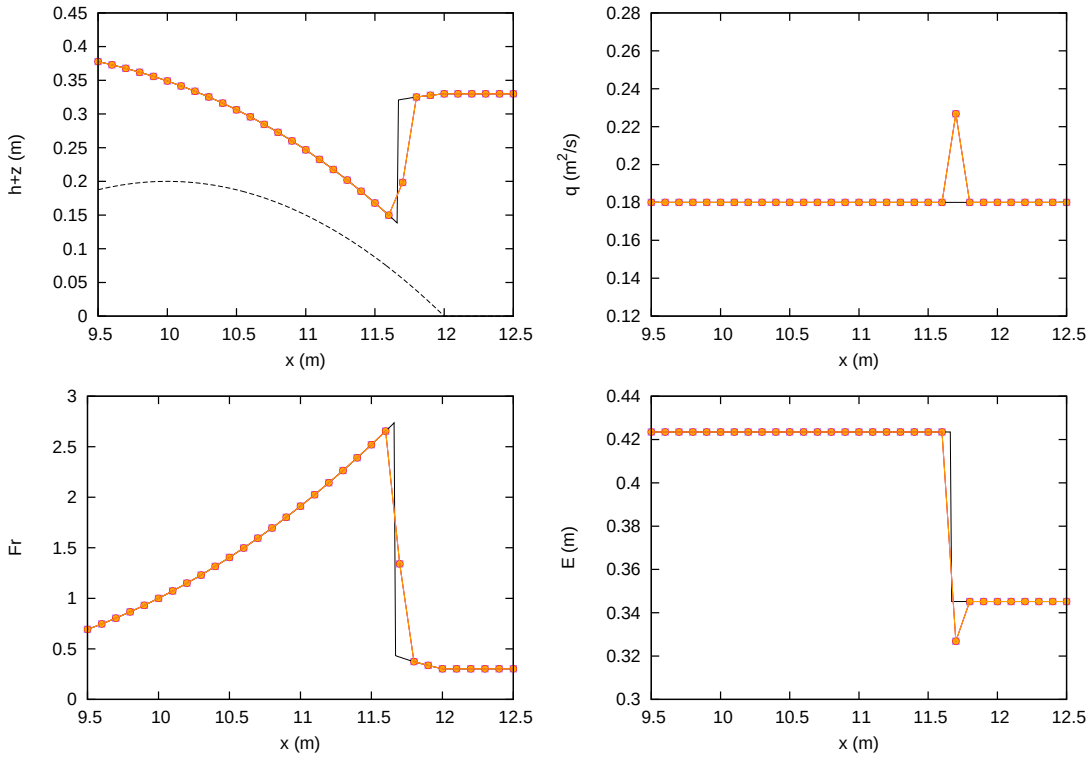

Figure 6.10: Section 6.3.1. Transcritical flow with a shock. Exact solution (—) and numerical solutions using 1st (–□–), 3rd (–●–) and 5th (–●–) order AR-ADER method. $\Delta x = 0.25$m.

Table 6.1: Section 6.3.2. Summary of test cases.

RP	h_L	h_R	u_L	u_R	z_L	z_R
1	4.0	0.69196567	0.0	0.0	0.0	1.5
2	0.3	0.39680194	2.0	2.2	1.0	1.0
3	1.0	1.2	3.0	0.1	1.1	1.0
4	1.0	2.0	2.0	4.0	1.1	1.0

right-moving shock wave. Supercritical motion from left to right is considered in RP 2, Figure 6.12. In RP 3, Figure 6.13, is a resonance problem that admits only one solution given by a sequence of shocks. In all cases the proposed numerical scheme provides accurate results for the water level surface at the bed discontinuity. Convergence is ensured with mesh refinement or when numerical order is increased. RP 4, in Figure 6.14, also is a resonant case with a unique solution. The solution begins with a rarefaction, followed by a stationary contact, continued by a shock wave and finally ends in a rarefaction. Again, the proposed scheme converges to the solution with mesh refinement or when the numerical order is increased.

6.3.3 Smooth case and convergence rate test

In this section, a convergence test is performed. The numerical solutions are compared with a solution computed with a very refined mesh, setting 5th order. A smooth initial condition is desirable in order to perform a proper analysis of convergence. The following function is proposed for the bed profile

$$z(x) = \begin{cases} 0 & if \quad x < 1 \\ 0.1(\sin \pi x)^4 & if \quad 1 \leq x \leq 2 \\ 0 & if \quad x > 2 \end{cases} \tag{6.40}$$

and the initial condition for the water level surface $h + z$

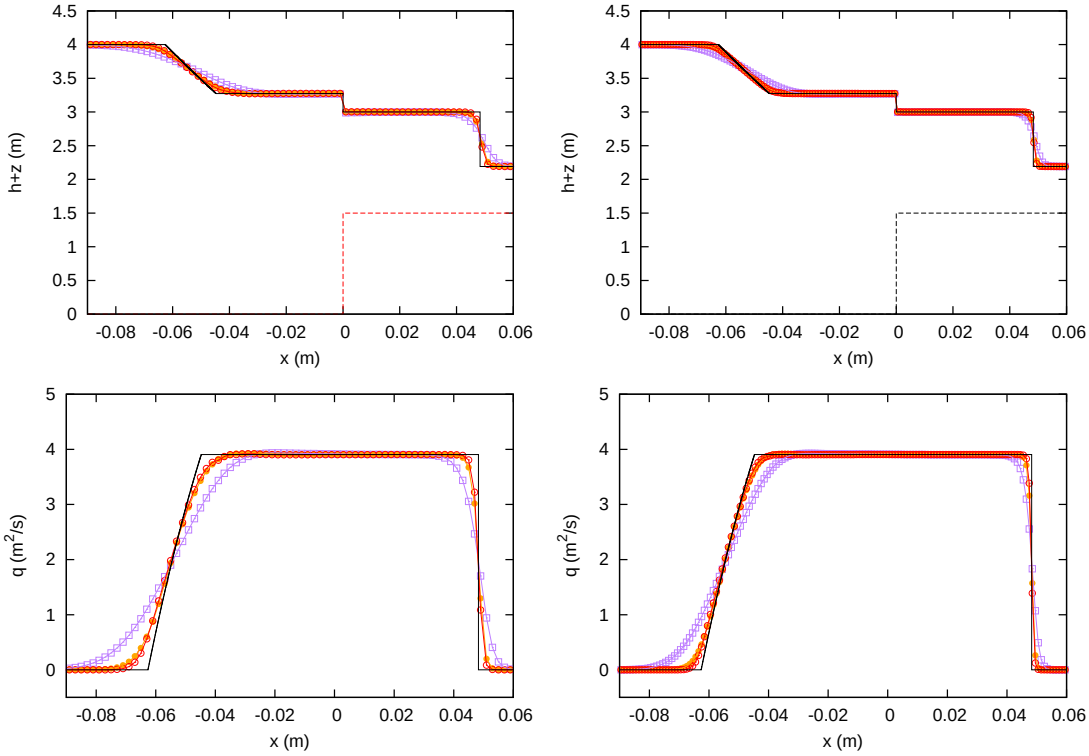

Figure 6.11: Section 6.3.2. RP 1. Exact solution (—) and numerical solutions using the 1st (–□–), 3rd (–●–) and 5th (–●–) order AR-ADER method using (left) 500 and (right) 1000 cells.

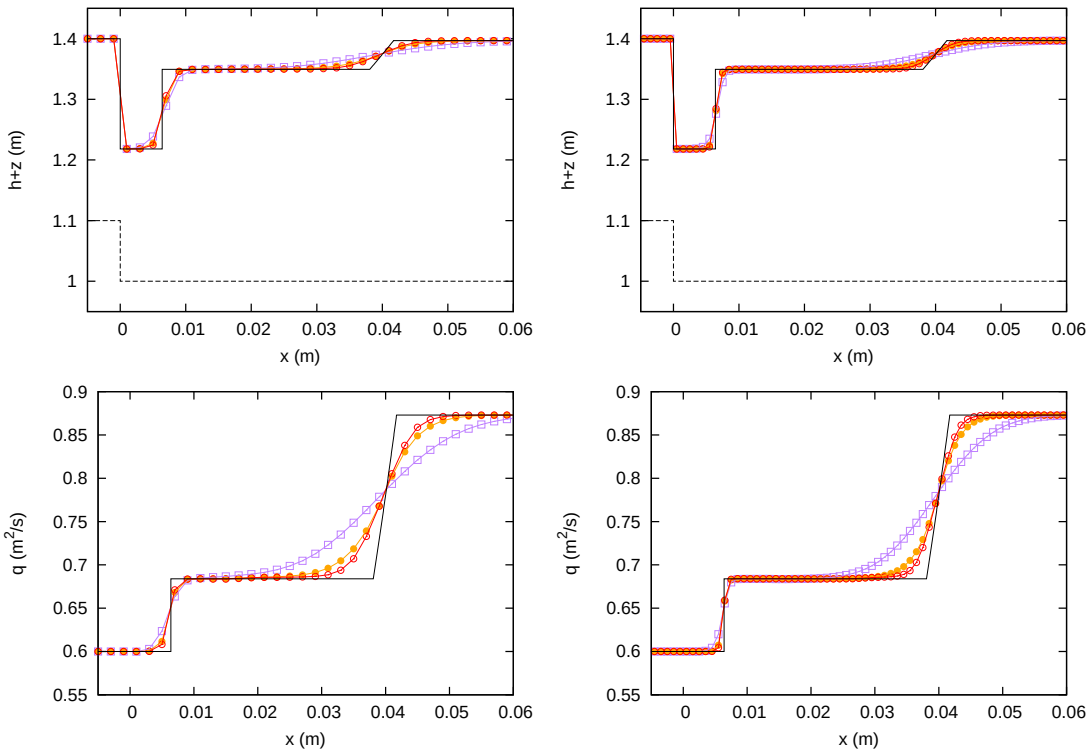

Figure 6.12: Section 6.3.2. RP 2. Exact solution (—) and numerical solutions using the 1st (–□–), 3rd (–●–) and 5th (–●–) order AR-ADER method using (left) 500 and (right) 1000 cells.

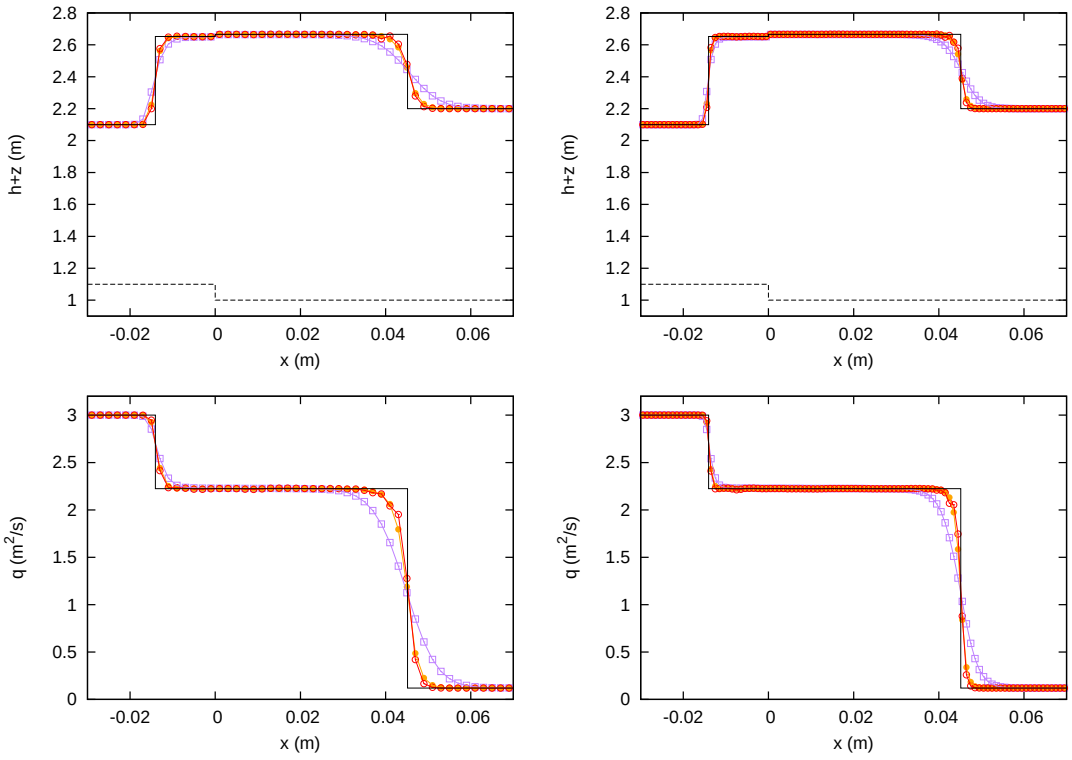

Figure 6.13: Section 6.3.2. RP 3. Exact solution (—) and numerical solutions using the 1st (–□–), 3rd (–●–) and 5th (–●–) order AR-ADER method using (left) 500 and (right) 1000 cells.

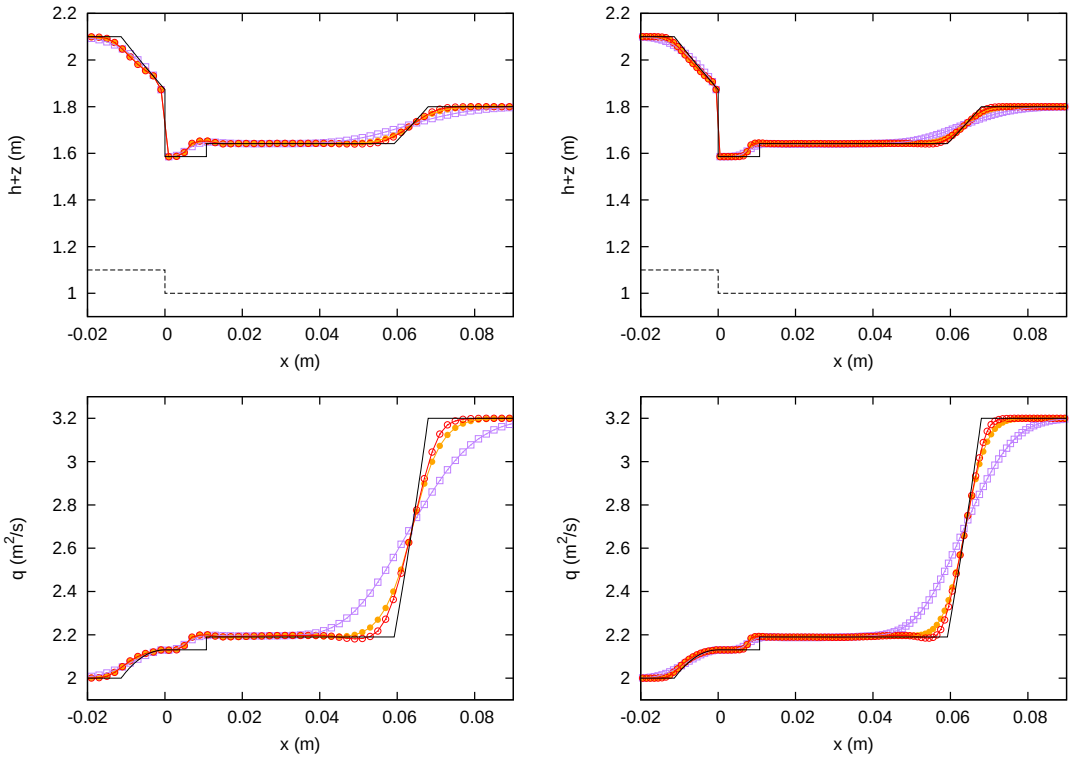

Figure 6.14: Section 6.3.2. RP 4. Exact solution (—) and numerical solutions using the 1st (–□–), 3rd (–●–) and 5th (–●–) order AR-ADER method using (left) 500 and (right) 1000 cells.

$$h(x,0)+z(x)=\begin{cases} 0.5 & if \quad x<1 \\ 0.5+0.08(\sin\pi x)^4 & if \quad 1\leq x\leq 2 \\ 0.5 & if \quad x>2 \end{cases} \tag{6.41}$$

in the computational domain $[0,3]$ meters.

Figure 6.15 (top) shows numerical results for the water level surface and discharge at time $t=0.05$ s, using the 1st, 3rd and 5th order AR-ADER method. CFL number is set equal to 0.5 in all cases. Figure 6.15 (bottom) shows water level surface and discharge for a larger time, $t=0.3$ s. The grid size used is $\Delta x=0.1$. Large differences appear among first and higher orders. First order is unable to capture maximum and minimum values so the predicted values of $h+z$ and q are strongly attenuated. Differences between 3rd and 5th order become more appreciable in this test case as time evolves.

Numerical errors have been computed for h and q using L_1, L_2 and L_∞ error norms. The results of the convergence rate test for the 3rd and 5th order AR-ADER method are shown in Tables F.9 and F.10 for h and q respectively. The prescribed order of accuracy is reached for the schemes up to 5th order, being suboptimal in certain circumstances.

Numerical results evidence that it is less efficient to use a low-order method on a fine mesh than a high-order method on a coarse mesh.

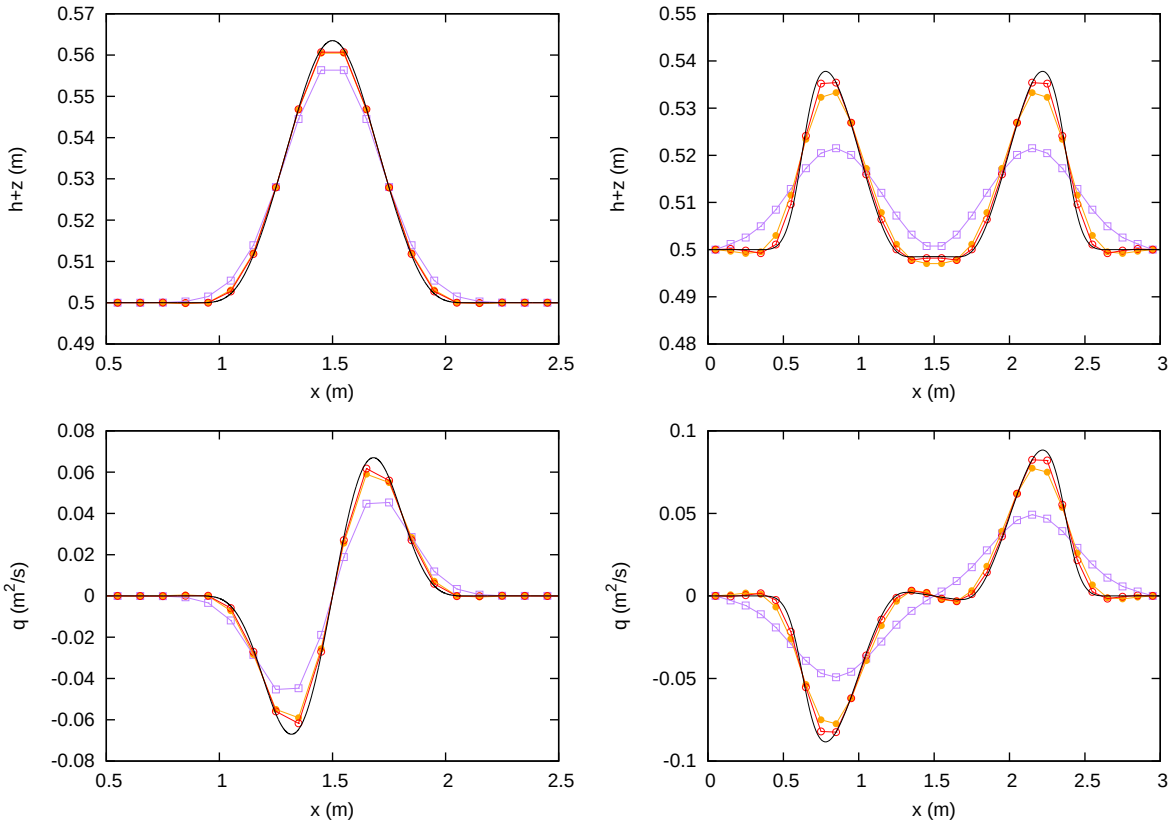

Figure 6.15: Section 6.3.3. Exact solution (−) and computed solutions for (upper) water surface level, $h+z$, and (lower) discharge, q, using 1st (−□−), 3rd order (−●−) and 5th AR-ADER method (−●−), at $t=0.05$ s (left) and $t=0.3$ s (right).

More numerical results for the shallow water equations including computational costs can be found in [25].

Chapter 7

Concluding remarks

In this work, numerical methods for the computation of hyperbolic conservation laws within the framework of fluid mechanics are studied. First order Godunov type numerical schemes and the corresponding Riemann solvers were presented first, serving as framework for the subsequent extension of such schemes to arbitrary order of accuracy carried out in the second part of the work. The ADER methodology was chosen. The necessity of suitable non-oscillatory reconstruction procedures when constructing high order numerical schemes was evidenced and the WENO reconstruction procedure was selected for the spatial reconstructions required for ADER schemes.

A novel improvement of the WENO-JS method has been introduced. The proposed technique recovers the optimal reconstruction when the reconstruction data is smooth and ensures the non-oscillatory property when discontinuities are present by emulating the WENO-JS reconstruction. The keystone of this improvement is the modification of the power exponent in the calculation of the α^{JS} weights by means of a suitable indicator that measures the smoothness of the function in the whole reconstruction domain. This global smoothness indicator provides a better measure of the smoothness of the function inside it and allows to identify either if real discontinuities are present or if the function is smooth, with independence of the existence of critical points.

In ADER schemes, a proper computation of the derivatives of the reconstructed function is essential to converge to the exact solution and to achieve the prescribed order of convergence. The sub-cell derivative reconstruction technique proposed in [21] has been used defining a inner grid that avoids the presence of negative optimal weights. For 2D cases, this method has been extended and is presented in Appendix E. For a good performance of such derivative reconstruction method, the departing data provided by a WENO reconstruction must be accurate. When critical points are present and the traditional WENO method fails, derivatives obtained using this procedure are erroneous. Only when using the WENO-PW reconstruction technique, it is ensured that we recover the proper values of the derivatives as the reconstructed polynomial does not contain spurious oscillations, with independence of the mesh refinement.

Numerical results using a Flux-ADER numerical scheme (TT-ADER scheme) for the resolution of the linear scalar transport equation are provided for different improved WENO methods such as the WENO-PW, WENO-5M, WENO-Z and WENO-MZ. When analyzing the numerical results, it is evidenced that the utilization of the WENO-PW reconstruction technique normally leads to more accurate results than the other existent approaches, specially for very high order schemes (9-th and 11-th orders). The computational cost of this method does not differ significantly from that of the WENO-JS method, however it is worth saying that the use of a real power exponent produces a higher computational cost than when it is an integer. The WENO-PW method offers a simple and inexpensive way to achieve the theoretical order of convergence when using the TT-ADER scheme for smooth cases, as well as to provide accurate results when computing discontinuous cases, always satisfying the non-oscillatory property.

The main novel point of this work is the high order extension of weak solutions for classical RPs involving geometric source terms, to compute the DRP_K, allowing to generate a flux-ADER type numerical scheme named AR-ADER. For that purpose, the DRP_K is decomposed in $K+1$ RPs: the DRP_0, for the leading term, and K RPs related to the derivative terms. The new solver is an extension of the ARoe solver in [1] and computes the $K+1$ solutions of the DRP_K used to form the truncated Taylor

power-series expansion in time of the fluxes, needed to construct a $K+1$-th order AR-ADER scheme. It is worth recalling some distinctive features of the AR-ADER scheme when compared to other traditional ADER schemes. The solver directly computes the solution of the derivative RPs departing from time derivatives of the fluxes, unlike classical ADER solvers, where spatial derivatives of the conserved variables are used instead. Another important feature of this solver, inherited from the ARoe solver, is that the contribution of the source term across the cell boundaries is accounted for by adding an extra wave in the solution.

It is worth mentioning that the leading terms in the AR-ADER scheme are computed ensuring an exact balance between fluxes and source terms. Numerical approximations done over the source term for the shallow water model ensure an energy balanced scheme, that is, when applied to steady flows, the exact solution is recovered with independence of the mesh refinement. High order terms are constructed replacing temporal derivatives by spatial derivatives written in terms of those variables that are constant in space under the steady regime, in order to preserve the discrete equilibrium. For instance, in the shallow water model, the reconstruction procedure is performed over the mechanical energy and discharge. In this way, convergence to the exact solution can be guaranteed in both steady and transient conditions. It is worth mentioning that numerical issues may arise when considering transcritical cases since time derivatives in terms of spatial derivatives lead to infinity. In this work, a reduction of the order of the numerical scheme to first order-accurate is proposes as a first approach.

The AR-ADER method has been implemented and tested up to a fifth order of accuracy in space and time and a thorough verification of the expected orders of accuracy of the schemes for the Burger's equation and the shallow water equations has been carried out, obtaining satisfactory results. For a given mesh, we conclude that when using the high-order version of the AR–ADER method instead of a low-order AR-ADER method, the error is lower but the computational cost is much greater. However, numerical results evidence that it is less efficient to use a low-order method on a fine mesh than a high-order method on a coarse mesh. This observation is a good argument for the justification of the use and development of high-order numerical schemes, such as the proposed AR-ADER method.

Future work must focus on the high order extension of other Riemann solvers such as the HLLS solver, as done for the ARoe solver, studying at the same time different kinds of discretization techniques for the integral of the source terms that must be designed considering the physical characteristics of the fluid flow. The 2D extension of such schemes for their application to nonlinear systems of equations such as the 2D shallow water equations also remains. To do it, the Cauchy-Kowalevski procedure must be carried out for the 2D system of equations but the derivation of the analytical expressions for time derivatives of the fluxes and conserved variables may become very tedious. Therefore, the implementation of an automatic differentiation algorithm could be suggested for that purpose, also making the application of the proposed numerical scheme to other systems of equations straightforward. It is worth mentioning that when addressing the multidimensional extension of the method, adaptable reconstruction procedures that provide an approximation of the variables and their derivatives inside arbitrary-shaped computational cells should be used, since quadrilateral grids are not adequate for the resolution of certain flow patterns and cannot be adapted to most domain geometries.

Bibliography

[1] J. Murillo, P. García-Navarro, Weak solutions for partial differential equations with source terms: application to the shallow water equations, J. Comput. Phys. 229 (2010) 4327–4368.

[2] J. Murillo, P. García-Navarro, Augmented versions of the HLL and HLLC Riemann Solvers including source terms in one and two dimensions for shallow flow applications, J. Comput. Phys. 231 (2012) 6861–6906.

[3] D.L. George. Augmented Riemann solvers for the shallow water equations over variable topography with steady states and inundation. Journal of Computational Physics 227 (2008) 3089–3113.

[4] J. Murillo, J. Burguete, P. Brufau, and P. García-Navarro. The influence of source terms on stability, accuracy and conservation in two-dimensional shallow flow simulation using triangular finite volumes, Int. J. Numer. Meth. Fluids (2007) 54 543–590.

[5] J. Murillo, P. García-Navarro, Augmented Roe's approaches for Riemann problems including source terms: definition of stability region with application to the shallow water equations with rigid and deformable bed. In M. E. Vázquez-Cendón and A. Hidalgo and P. García-Navarro and L. Cea, eds., Numerical Methods for Hyperbolic Equations. Theory and Applications, pages 149–154. Taylor-Francis Group, 2013.

[6] A. Bermudez and M.E. Vázquez-Cendón, Upwind methods for hyperbolic conservation laws with source terms, Comput. Fluids. 23 (1994) 1049–1071.

[7] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.

[8] E.F. Toro, R.C. Millington, and L.A.M. Nejad. Primitive upwind methods for hyperbolic partial differential equations. In C. H. Bruneau, editor, Sixteenth International Conference on Numerical Methods for Fluid Dynamics. Lecture Notes in Physics, pages 421–426. Springer-Verlag, 1998.

[9] E.F. Toro, R.C. Millington, and L.A.M. Nejad. Towards very high order Godunov schemes. In E. F. Toro, editor, Godunov Methods. Theory and Applications, pages 907–940. Kluwer/Plenum Academic Publishers, 2001.

[10] E.F. Toro, V.A. Titarev, Solution of the generalised Riemann problem for advection-reaction equations, Proc. Roy. Soc. London A 458 (2002) 271–281.

[11] E.F. Toro, V.A. Titarev, ADER schemes for scalar hyperbolic conservation laws with source terms in three space dimensions, J. Comput. Phys. 202 (1) (2005) 196–215.

[12] E.F. Toro, V.A. Titarev, Derivative Riemann solvers for systems of conservation laws and ADER methods, J. Comput Phys. 212 (1) (2006) 150–165.

[13] S.K. Godunov, A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics, Mat. Sb. 47 (1959) 357–393

[14] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput. Phys. 49 (1983) 357-393.

[15] B. Van Leer, Towards the ultimate conservative difference scheme II, monotonicity and conservation combined in a second order scheme, J. Comput. Phys. 14 (1974) 361-470.

[16] B. Van Leer, Towards the ultimate conservative difference scheme V, a second order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101-136.

[17] A. Harten, B. Enquist, S. Osher, S. Chakravarthy , Uniformly high order accurate essentially non-oscillatory schemes, J. Comput Phys. 131 (1997) 3-17

[18] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput Phys. 115 (1994) 200-212.

[19] G.S. Jiang, C.W. Shu, Efficient implementation of weighted ENO schemes, J. Comput Phys. 126 (1996) 202-228.

[20] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in *Advanced Numerical Aproximation of Nonlinear Hyperbolic Equations*, edited by B. Cockburn, C. Johnson, C.-W. Shu, and E. Tadmor, Lect. Notes in Math. Springer-Verlag, Berlin/New York, 1998 Vol. 1697.

[21] J.B. Cheng, E. F. Toro, S. Jiang, W. Tang, A sub-cell WENO reconstruction method for spatial derivatives in the ADER scheme , *Journal of Computational Physics*, 251 (2013) 53–80.

[22] A.K. Henrick, T.D. Aslam, J.M. Powers, Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points, J. Comput Phys. 207 (2005) 542-567.

[23] R. Borges, C. Carmona, B. Costa, W.S. Don, An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws, J. Comput Phys. 227 (6) (2008) 3101-3211.

[24] E.F. Toro, V.A. Titarev, Derivative Riemann solvers for systems of conservation laws and ADER methods, *Journal of Computational Physics*, Volume 212, Issue 1, 10 February 2006, Pages 150-165, ISSN 0021-9991

[25] A. Navas-Montilla, J. Murillo, Energy balanced numerical schemes with very high order. The Augmented Roe Flux ADER scheme. Application to the shallow water equations, J. Comput. Phys. 290 (2015) 188–218

[26] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput Phys. 49 (1983) 357-393.

[27] M. Castro, B. Costa, W.S. Don, High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws, J. Comput Phys. 230 (6) (2011) 1766-1792.

[28] Leveque, R. *Finite Volume Methods for Hyperbolic Problem*. Cambridge University Press, New York, 2002.

[29] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer-Verlag, Berlin, 1999.

[30] E. Godlewski, P.-A. Raviart *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer Science and Business Media, Berlin, 2013.

[31] E. Carlini, R. Ferretti and G. Russo, A weighted essentially non-oscillatory, large time-step scheme for Hamilton-Jacobi equations, *SIAM Journal of Scientific Computing*, 2005.

[32] Y. Liu., C-W. Shu, M. Zhang, On the positivity of linear weights in WENO approximations. Springer-Verlag, 25 (2009) 503-538.

[33] G.S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, *Journal of Computational Physics*, 126 (1996) 202–228.

[34] E.F. Toro, V.A. Titarev, ADER schemes for three-dimensional non-linear hyperbolic systems, J. Comput Phys. 204 (2) (2005) 715–736.

[35] V. A. Titarev. Derivative Riemann Problem and ADER Schemes. PhD thesis, Department of Mathematics, University of Trento, Italy, 2005.

[36] V. A. Titarev and E. F. Toro. ADER: Arbitrary high order Godunov approach. *J. Sci. Comput.*, 17(1-4):609–618, 2002.

[37] Y Takakura, EF Toro: Arbitrarily Accurate Non-oscillatory Schemes for Nonlinear Scalar Conservation Laws with Source Terms. AIAA paper 2002-2736 (2002)

[38] P.G. LeFloch, M.D. Thanh, A Godunov-type method for the shallow water equations with discontinuous topography in the resonant regime, J. Comput. Phys. 230 (2011) 7631–7660.

[39] J. Murillo, P. García-Navarro, Energy balance numerical schemes for shallow water equations with discontinuous topography , J. Comput. Phys. 236 (2012) 119–142.

[40] Davies-Jones R. Comments on " Kinematic analysis of frontogenesis associated with a nondivergent vortex". Journal of the Atmospheric Sciences 1985 42(19) 2073-–5.

[41] Charles A. Doswell III, 1984: A Kinematic Analysis of Frontogenesis Associated with a Nondivergent Vortex. J. Atmos. Sci., 41, 1242–1248.

[42] S. Zhao, N. Lardjane, I. Fedioun, Comparison of improved finite-difference WENO schemes for the implicit large eddy simulation of turbulent non-reacting and reacting high-speed shear flows, Comput Fluids, 95 (2014) 74-87.

[43] Y. Ha, C.H. Kim, Y. J. Lee, J. Yoon, An improved weighted essentially non-oscillatory scheme with a new smoothness indicator, J. Comput Phys. 232 (2013) 68–86.

[44] J. Shi, Y.T. Zhang, C.W. Shu, Resolution of high order WENO schemes for complicated flow structures, J. Comput Phys. 186 (2003) 690–696.

[45] E.F. Toro, Riemann solvers and numerical methods for fluid dynamics: a practical introduction, third ed., Springer-Verlag, Berlin, Heidelberg, 2009.

[46] A. Harten, P. Lax, B. van Leer, On upstream differencing and Godunov type methods for hyperbolic conservation laws, SIAM Rev., 25 (1983), 35-–61

[47] P.G. LeFloch, M.D. Thanh, A Godunov-type method for the shallow water equations with discontinuous topography in the resonant regime, J. Comput. Phys. 230 (2011) 7631–7660.

List of Figures

List of Tables

Appendix A

First order approximate Riemann solvers

A.1 First order Augmented solver for scalar equations

Scalar version of RP in (3.17) reads

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial f(u)}{\partial x} = s \\ \\ u(x,0) = \begin{cases} u_i & x < 0 \\ u_{i+1} & x > 0 \end{cases} \end{cases} \tag{A.1}$$

where $u \in \mathbb{R}$ is the conserved variable, $s \in \mathbb{R}$ the source term and $f(u) : \mathbb{R} \to \mathbb{R}$ the physical flux, which is a nonlinear function of the conserved variable.

The integral form of (A.1) over the control volume $[0, \Delta t] \times [-x_L, x_R]$ is given by

$$\int_{-x_L}^{x_R} \int_0^{\Delta t} \left(\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} - s \right) dxdt = 0 \tag{A.2}$$

and the following expression for the integral volume of $u(x, \Delta t)$ inside $[-x_L, x_R]$ is obtained

$$\int_{-x_L}^{x_R} u(x, \Delta t)\, dx = x_R u_{i+1} + x_L u_i - (\delta f - \bar{s})_{i+\frac{1}{2}} \Delta t \tag{A.3}$$

with $f_{i+1} = f(u_{i+1})$ and $f_i = f(u_i)$ and the source term integrated as

$$\int_{-x_L}^{x_R} \int_0^{\Delta t} s(u_i, u_{i+1}, t = 0)\, dxdt = \Delta t \bar{s}_{i+\frac{1}{2}}\,. \tag{A.4}$$

Problem in (A.1) can be approximated by the following constant coefficient linear RP

$$\begin{cases} \dfrac{\partial \hat{u}}{\partial t} + \widetilde{\lambda}_{i+\frac{1}{2}} \dfrac{\partial \hat{u}}{\partial x} = s \\ \\ \hat{u}(x,0) = \begin{cases} u_i & x < 0 \\ u_{i+1} & x > 0 \end{cases} \end{cases} \tag{A.5}$$

where $\hat{u}(x,t)$ is the approximate solution of (A.1) and $\widetilde{\lambda}_{i+\frac{1}{2}}$ is a constant wave velocity defined as a function of left and right states (u_i and u_{i+1}) that represents an approximation of the propagation velocity $\lambda(u) = \partial_u f(u)$ at $x_{i+\frac{1}{2}}$.

If expressing the integral form of (A.19) over the same control volume than in the previous case

$$\int_{-x_L}^{x_R} \hat{u}(x,\Delta t)\,dx = x_R u_{i+1} + x_L u_i - \left(\widetilde{\lambda}\delta u + \bar{s}\right)_{i+\frac{1}{2}} \Delta t \tag{A.6}$$

and imposing *consistency condition* between (A.3) and (A.6)

$$\int_{-x_L}^{x_R} \hat{u}(x,\Delta t)\,dx = \int_{-x_L}^{x_R} u(x,\Delta t)\,dx \tag{A.7}$$

the following constraint is noticed

$$\delta f_{i+\frac{1}{2}} = \widetilde{\lambda}_{i+\frac{1}{2}} \delta u_{i+\frac{1}{2}}\,. \tag{A.8}$$

that allows to compute the value of $\widetilde{\lambda}_{i+\frac{1}{2}}$.
The solution for $\hat{u}$ in (A.5) consists of three regions, as depicted in Figure A.1 for the particular case when $\widetilde{\lambda}_{i+\frac{1}{2}} > 0$.

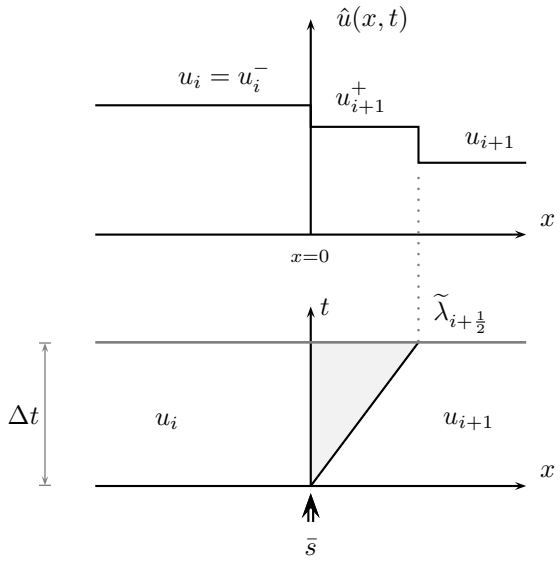

Figure A.1: Values of the solution $\hat{u}(x,t)$ in each wedge of the (x,t) plane.

It is possible to define the solution for each characteristic RP on the left and right sides of the t axis, denoted by u_i^- and u_{i+1}^+ respectively as depicted in Figure A.1. These values are defined as

$$u_i^- = \lim_{x\to 0^-} \hat{u}(x,t) \qquad u_{i+1}^+ = \lim_{x\to 0^+} \hat{u}(x,t) \tag{A.9}$$

In Figure A.1, it is observed that the solution on the left hand side of the interface, u_i^-, is equal to the left state since the wave propagates to the right. However, a new state on the right hand side of the interface, u_{i+1}^+, appears. To find the value for u_{i+1}^+ the RH condition across the steady wave at the interface must be obtained first

$$f_{i+1}^+ - f_i^- - \bar{s}_{i+\frac{1}{2}} = 0 \tag{A.10}$$

On the other hand, if we assume that the difference of states and fluxes across the discontinuity are related using the approximate wave velocity in the following way

$$f_{i+1}^+ - f_i^- = \widetilde{\lambda}_{i+\frac{1}{2}}(u_{i+1}^+ - u_i^-) \tag{A.11}$$

then, the right state can be obtained by substitution of (A.11) in (A.10), yielding

$$u_{i+1}^{+} = u_i^{-} + \begin{pmatrix} \bar{s} \\ \widetilde{\lambda} \end{pmatrix}_{i+\frac{1}{2}} = u_i + \begin{pmatrix} \bar{s} \\ \widetilde{\lambda} \end{pmatrix}_{i+\frac{1}{2}} \tag{A.12}$$

It is also possible to apply the RH condition across the positive moving wave as

$$f_{i+1} - f_{i+1}^{+} = \widetilde{\lambda}_{i+\frac{1}{2}}(u_{i+1} - u_{i+1}^{+}) \tag{A.13}$$

and substitution of (A.12) in (A.13) leads to the expression for the right state flux

$$f_{i+1}^{+} = f_i + \bar{s}_{i+\frac{1}{2}} \tag{A.14}$$

The solution in the $x-t$ plane can be expressed as a piecewise constant function that depends upon x and t as

$$\hat{u}(x,t) = \begin{cases} u_i & \text{if} \quad x < \widetilde{\lambda}_{i+\frac{1}{2}} t \\ u_i + (\theta \delta u)_{i+\frac{1}{2}} & \text{if} \quad \widetilde{\lambda}_{i+\frac{1}{2}} t < x < 0 \\ u_{i+1} & \text{if} \quad 0 < x \end{cases} \tag{A.15}$$

when $\widetilde{\lambda}_{i+\frac{1}{2}} < 0$, and

$$\hat{u}(x,t) = \begin{cases} u_i & \text{if} \quad x < 0 \\ u_{i+1} - (\theta \delta u)_{i+\frac{1}{2}} & \text{if} \quad 0 < x < \widetilde{\lambda}_{i+\frac{1}{2}} t \\ u_{i+1} & \text{if} \quad \widetilde{\lambda}_{i+\frac{1}{2}} t < x \end{cases} \tag{A.16}$$

when $\widetilde{\lambda}_{i+\frac{1}{2}} > 0$, with

$$\theta_{i+\frac{1}{2}} = 1 - \left(\frac{\bar{s}}{\delta f} \right)_{i+\frac{1}{2}} \tag{A.17}$$

and from Equations (A.15) and (A.16) they yield

$$\begin{aligned} u_i^{-} &= \begin{cases} u_i & if \quad \widetilde{\lambda}_{i+\frac{1}{2}} > 0 \\ u_i + (\theta \delta u)_{i+\frac{1}{2}} & if \quad \widetilde{\lambda}_{i+\frac{1}{2}} < 0 \end{cases} \\ u_{i+1}^{+} &= \begin{cases} u_{i+1} - (\theta \delta u)_{i+\frac{1}{2}} & if \quad \widetilde{\lambda}_{i+\frac{1}{2}} > 0 \\ u_{i+1} & if \quad \widetilde{\lambda}_{i+\frac{1}{2}} < 0 \end{cases} \end{aligned} \tag{A.18}$$

A.2 First order Augmented solvers for systems

A.2.1 Approximate solution using ARoe solver

RP in (3.17) can be approximated by exactly solving the following constant coefficient linear RP

$$\begin{cases} \dfrac{\partial \hat{\mathbf{U}}}{\partial t} + \widetilde{\mathbf{J}}_{i+\frac{1}{2}} \dfrac{\partial \hat{\mathbf{U}}}{\partial x} = \mathbf{S} \\ \\ \hat{\mathbf{U}}(x,0) = \begin{cases} \mathbf{U}_i & x < 0 \\ \mathbf{U}_{i+1} & x > 0 \end{cases} \end{cases} \tag{A.19}$$

where $\hat{\mathbf{U}}(x,t)$ is the approximate solution of (3.17) and $\widetilde{\mathbf{J}}_{i+\frac{1}{2}} = \widetilde{\mathbf{J}}_{i+\frac{1}{2}}(\mathbf{U}_i, \mathbf{U}_{i+1})$ is a constant matrix defined as a function of left and right states ($\mathbf{U}_i$ and $\mathbf{U}_{i+1}$) that represents an approximation of the Jacobian at $x_{i+\frac{1}{2}}$.

If expressing the integral form of (A.19) over the same control volume than in the previous case

$$\int_{-x_L}^{x_R} \hat{\mathbf{U}}(x,\Delta t)\,dx = x_R\mathbf{U}_{i+1} + x_L\mathbf{U}_i - \left(\widetilde{\mathbf{J}}\delta\mathbf{U} + \bar{\mathbf{S}}\right)_{i+\frac{1}{2}}\Delta t \tag{A.20}$$

and imposing the *consistency condition*

$$\int_{-x_L}^{x_R} \hat{\mathbf{U}}(x,\Delta t)\,dx = \int_{-x_L}^{x_R} \mathbf{U}(x,\Delta t)\,dx \tag{A.21}$$

the following constraint is noticed

$$\delta\mathbf{F}_{i+\frac{1}{2}} = \widetilde{\mathbf{J}}_{i+\frac{1}{2}}\delta\mathbf{U}_{i+\frac{1}{2}}\,. \tag{A.22}$$

Matrix $\widetilde{\mathbf{J}}_{i+\frac{1}{2}}$ is considered to be diagonalizable with N_λ approximate real eigenvalues

$$\widetilde{\lambda}^1_{i+\frac{1}{2}} < \ldots < \widetilde{\lambda}^I_{i+\frac{1}{2}} < 0 < \widetilde{\lambda}^{I+1}_{i+\frac{1}{2}} < ... < \widetilde{\lambda}^{N_\lambda}_{i+\frac{1}{2}} \tag{A.23}$$

and N_λ eigenvectors $\widetilde{\mathbf{e}}^1, ..., \widetilde{\mathbf{e}}^{N_\lambda}$. With them, two approximate matrices, $\widetilde{\mathbf{P}}_{i+\frac{1}{2}} = (\widetilde{\mathbf{e}}^1, ..., \widetilde{\mathbf{e}}^{N_\lambda})_{i+\frac{1}{2}}$ and $\widetilde{\mathbf{P}}^{-1}_{i+\frac{1}{2}}$ are built with the following property

$$\widetilde{\mathbf{J}}_{i+\frac{1}{2}} = (\widetilde{\mathbf{P}}\widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{P}}^{-1})_{i+\frac{1}{2}}, \qquad \widetilde{\mathbf{\Lambda}}_{i+\frac{1}{2}} = \begin{pmatrix} \widetilde{\lambda}^1 & & 0 \\ & \ddots & \\ 0 & & \widetilde{\lambda}^{N_\lambda} \end{pmatrix}_{i+\frac{1}{2}} \tag{A.24}$$

where $\widetilde{\mathbf{\Lambda}}_{i+\frac{1}{2}}$ is a diagonal matrix with approximate eigenvalues in the main diagonal. As done in Section 2.4.1, system in (A.19) can be transformed using $\widetilde{\mathbf{P}}^{-1}$ matrix as follows

$$\widetilde{\mathbf{P}}^{-1}_{i+\frac{1}{2}}\left(\frac{\partial\hat{\mathbf{U}}}{\partial t} + \widetilde{\mathbf{J}}_{i+\frac{1}{2}}\frac{\partial\hat{\mathbf{U}}}{\partial x}\right) = \widetilde{\mathbf{P}}^{-1}_{i+\frac{1}{2}}\mathbf{S} \tag{A.25}$$

expressing (A.19) in terms of the characteristic variables $\hat{\mathbf{W}} = \widetilde{\mathbf{P}}^{-1}_{i+\frac{1}{2}}\hat{\mathbf{U}}$, with $\hat{\mathbf{W}} = (\hat{w}^1, ..., \hat{w}^{N_\lambda})$. This transformation leads to a decoupled system that generates the following linear RP

$$\begin{cases} \dfrac{\partial\hat{\mathbf{W}}}{\partial t} + \widetilde{\mathbf{\Lambda}}_{i+\frac{1}{2}}\dfrac{\partial\hat{\mathbf{W}}}{\partial x} = \mathbf{B}_{i+\frac{1}{2}} \\ \hat{\mathbf{W}}(x,0) = \begin{cases} \mathbf{W}_i = \widetilde{\mathbf{P}}^{-1}_{i+\frac{1}{2}}\mathbf{U}_i & \text{if} \quad x<0 \\ \mathbf{W}_{i+1} = \widetilde{\mathbf{P}}^{-1}_{i+\frac{1}{2}}\mathbf{U}_{i+1} & \text{if} \quad x>0 \end{cases} \end{cases} \tag{A.26}$$

with $\mathbf{B}_{i+\frac{1}{2}} = \widetilde{\mathbf{P}}^{-1}_{i+\frac{1}{2}}\mathbf{S} = (\beta^1, ..., \beta^{N_\lambda})_{i+\frac{1}{2}}$, where each equation

$$\frac{\partial\hat{w}^m}{\partial t} + \widetilde{\lambda}^m_{i+\frac{1}{2}}\frac{\partial\hat{w}^m}{\partial x} = \beta^m_{i+\frac{1}{2}}, \qquad m = 1, ..., N_\lambda \tag{A.27}$$

involves the variable $\hat{w}^m$ and the source term $\beta^m_{i+\frac{1}{2}}$. As equations in (A.27) are decoupled, RP in (A.26) can be decomposed in N_λ independent RPs

$$\begin{cases} \dfrac{\partial\hat{w}^m}{\partial t} + \widetilde{\lambda}^m_{i+\frac{1}{2}}\dfrac{\partial\hat{w}^m}{\partial x} = \beta^m_{i+\frac{1}{2}} \\ \hat{w}^m(x,0) = \begin{cases} w^m_i & \text{if} \quad x<0 \\ w^m_{i+1} & \text{if} \quad x>0 \end{cases} \end{cases} \tag{A.28}$$

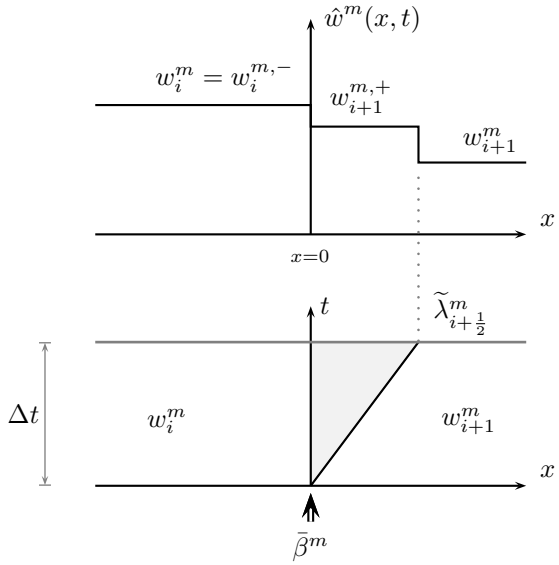

Figure A.2: Values of the solution $\hat{w}(x,t)$ in the (x,t) plane.

The solution for each $\hat{w}^m$ characteristic variable is given by the solution of the scalar RP (A.28) [weaksol] and and consist of three regions as depicted in Figure A.2.

The solution can be expressed as a piecewise constant function that depends upon x and t as

$$\hat{w}^m(x,t) = \begin{cases} w_i^m & \text{if} \quad x < \widetilde{\lambda}_{i+\frac{1}{2}}^m t \\ w_i^m + (\theta\delta w)_{i+\frac{1}{2}}^m & \text{if} \quad \widetilde{\lambda}_{i+\frac{1}{2}}^m t < x < 0 \\ w_{i+1}^m & \text{if} \quad 0 < x \end{cases} \tag{A.29}$$

when $\widetilde{\lambda}_{i+\frac{1}{2}}^m < 0$, and

$$\hat{w}^m(x,t) = \begin{cases} w_i^m & \text{if} \quad x < 0 \\ w_{i+1}^m - (\theta\delta w)_{i+\frac{1}{2}}^m & \text{if} \quad 0 < x < \widetilde{\lambda}_{i+\frac{1}{2}}^m t \\ w_{i+1}^m & \text{if} \quad \widetilde{\lambda}_{i+\frac{1}{2}}^m t < x \end{cases} \tag{A.30}$$

when $\widetilde{\lambda}_{i+\frac{1}{2}}^m > 0$, with

$$\theta_{i+\frac{1}{2}}^m = 1 - \left(\frac{\bar{\beta}^m}{\widetilde{\lambda}^m \alpha^m}\right)_{i+\frac{1}{2}} \tag{A.31}$$

where the set of wave strengths is defined as

$$\mathbf{A}_{i+\frac{1}{2}} = (\alpha^1, ..., \alpha^{N_\lambda})_{i+\frac{1}{2}}^T = \delta\mathbf{W}_{i+\frac{1}{2}} = (\widetilde{\mathbf{P}}^{-1}\delta\mathbf{U})_{i+\frac{1}{2}} \tag{A.32}$$

and the set of source strengths

$$\bar{\mathbf{B}}_{i+\frac{1}{2}} = (\bar{\beta}^1, ..., \bar{\beta}^{N_\lambda})_{i+\frac{1}{2}}^T = (\widetilde{\mathbf{P}}^{-1}\bar{\mathbf{S}})_{i+\frac{1}{2}} \tag{A.33}$$

Analogously, it is possible to define the solution for each characteristic RP on the left and right sides of the t axis, denoted by $w_i^{m,-}$ and $w_{i+1}^{m,+}$ respectively as depicted in Figure A.2. These values are defined as

$$w_i^{m,-} = \lim_{x\to 0^-} \hat{w}^m(x,t) \qquad w_{i+1}^{m,+} = \lim_{x\to 0^+} \hat{w}^m(x,t) \tag{A.34}$$

and from Equations (A.29) and (A.30) they yield

$$w_i^{m,-} = \begin{cases} w_i^m & if \quad \widetilde{\lambda}_{i+\frac{1}{2}}^m > 0 \\ w_i^m + (\theta\delta w)_{i+\frac{1}{2}}^m & if \quad \widetilde{\lambda}_{i+\frac{1}{2}}^m < 0 \end{cases}$$
$$w_{i+1}^{m,+} = \begin{cases} w_{i+1}^m - (\theta\delta w)_{i+\frac{1}{2}}^m & if \quad \widetilde{\lambda}_{i+\frac{1}{2}}^m > 0 \\ w_{i+1}^m & if \quad \widetilde{\lambda}_{i+\frac{1}{2}}^m < 0 \end{cases} \tag{A.35}$$

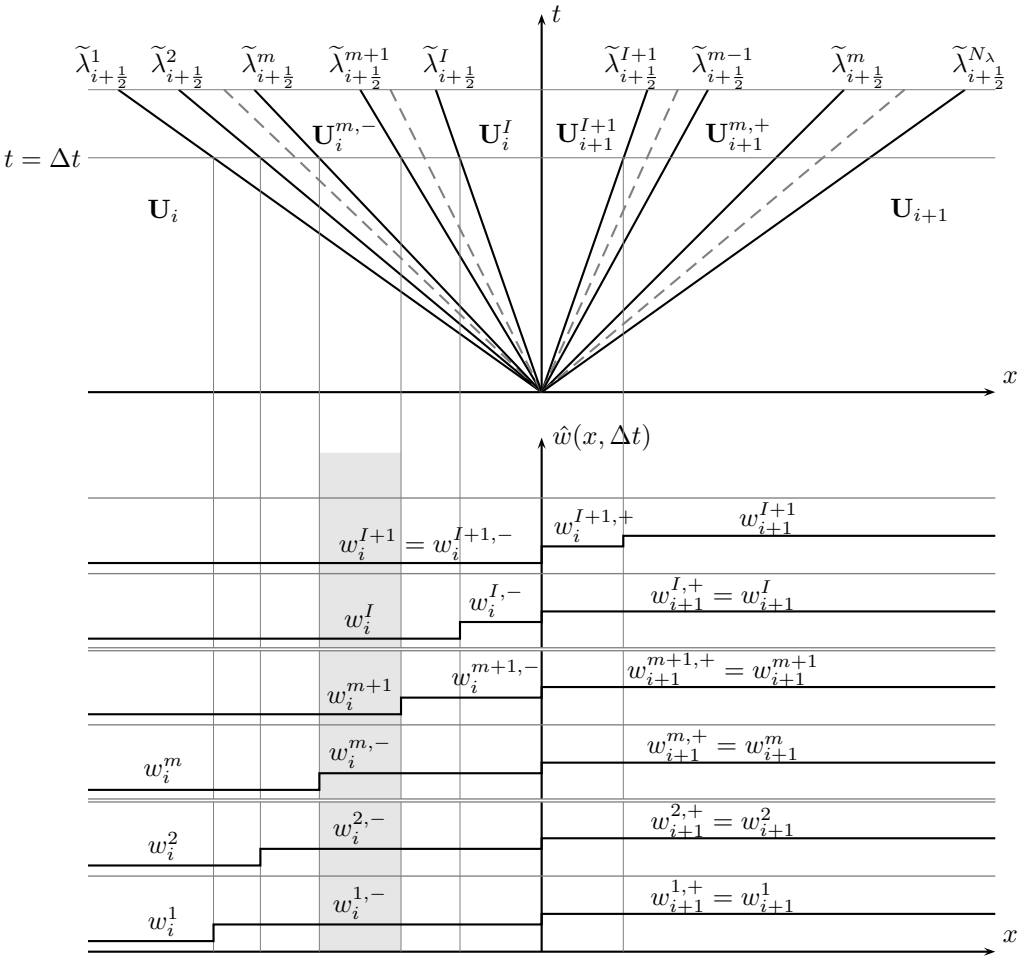

Figure A.3: Upper: Approximate solution $\hat{\mathbf{U}}(x,t)$. The solution consist of N_λ inner constant states separated by a stationary shock wave, with celerity $S = 0$ at $x = 0$. Lower: The solution for characteristic variables $\hat{w}^m(x,t)$ for $m = 1, ..., I+1$ is depicted at $t = \Delta t$.

The derivation of the general solution $\hat{\mathbf{U}}(x,t)$ for a linear system is based on the expansion of the solution as a linear combination of the vectors that compose the Jacobian's eigenvectors basis, using the relation $\mathbf{U} = \widetilde{\mathbf{P}}\mathbf{W}$, as follows

$$\hat{\mathbf{U}}(x,t) = \sum_{m_1=1}^{N_\lambda} \hat{w}^{m_1}(x,t)\, \widetilde{\mathbf{e}}_{i+\frac{1}{2}}^{m_1}\,, \tag{A.36}$$

where the scalar values $\hat{w}^{m_1}(x,t)$ are the characteristic approximate solutions at the sought point and represent the strength of each wave.

If focusing on a constant state on the left hand side of the t-axis, $\mathbf{U}^{m,-}$, defined between characteristic lines $\widetilde{\lambda}^{m}_{i+\frac{1}{2}}t$ and $\widetilde{\lambda}^{m+1}_{i+\frac{1}{2}}t$, the solution is given by the combination of the characteristic solutions in the spatial domain $[\widetilde{\lambda}^{m}_{i+\frac{1}{2}}t, \widetilde{\lambda}^{m+1}_{i+\frac{1}{2}}t]$. Considering that $\mathbf{U}_i = \sum_{m_1=1}^{N_\lambda} w_i^{m_1}\widetilde{\mathbf{e}}^{m_1}_{i+\frac{1}{2}}$, $\delta w^{m_1}_{i+\frac{1}{2}} = \alpha^{m_1}_{i+\frac{1}{2}}$ and the scalar solutions provided before, an arbitrary left state can be expressed as

$$\mathbf{U}_i^{m,-} = \mathbf{U}_i + \sum_{m_1=1}^{m} (\theta\alpha\widetilde{\mathbf{e}})^{m_1}_{i+\frac{1}{2}}\,, \tag{A.37}$$

When seeking the primitive vector solution for a state defined on the right hand side of the t-axis, $\mathbf{U}^{m,+}_{i+1}$, it has to be defined between characteristic lines $\widetilde{\lambda}^{m-1}_{i+\frac{1}{2}}t$ and $\widetilde{\lambda}^{m}_{i+\frac{1}{2}}t$. Following expansion in (A.36), the combination of the characteristic solutions in the spatial domain $[\widetilde{\lambda}^{m-1}_{i+\frac{1}{2}}t, \widetilde{\lambda}^{m}_{i+\frac{1}{2}}t]$ provides

$$\mathbf{U}^{m,+}_{i+1} = \mathbf{U}_{i+1} - \sum_{m_1=m}^{N_\lambda} (\theta\alpha\widetilde{\mathbf{e}})^{m_1}_{i+\frac{1}{2}} \tag{A.38}$$

Expressions for $\mathbf{U}_i^-$ and $\mathbf{U}^+_{i+1}$ can be derived from the previous results, setting $m = I$ in (A.37) and $m = I+1$ in (A.38) respectively, leading to

$$\begin{aligned} \mathbf{U}_i^- &= \mathbf{U}_i + \sum_{m_1=1}^{I} (\theta\alpha\widetilde{\mathbf{e}})^{m_1}_{i+\frac{1}{2}} \\ \mathbf{U}^+_{i+1} &= \mathbf{U}_{i+1} - \sum_{m_1=I+1}^{N_\lambda} (\theta\alpha\widetilde{\mathbf{e}})^{m_1}_{i+\frac{1}{2}} \end{aligned} \tag{A.39}$$

The difference between left and right states across the interface can be expressed as

$$\mathbf{U}^+_{i+1} - \mathbf{U}_i^- = \mathbf{U}_{i+1} - \mathbf{U}_i - \sum_{m_1=1}^{N_\lambda} (\theta\alpha\widetilde{\mathbf{e}})^{m_1}_{i+\frac{1}{2}} \tag{A.40}$$

In order to provide a complete description of the approximate flux function $\hat{\mathbf{F}}(x,t)$, the inner constant fluxes on the left side of the (x,t) plane will be denoted by $\mathbf{F}_i^{m,-}$, where $1 \le m \le I$. On the right side of the (x,t) plane solution, inner constant states are denoted by $\mathbf{F}^{m,+}_{i+1}$, where $I+1 \le m \le N_\lambda$. The approximate solution for the fluxes can be constructed defining appropriate RH condition across each moving wave.

Approximate fluxes on the left and right side of the t axis, $\mathbf{F}_i^-$ and $\mathbf{F}^+_{i+1}$, can be derived using the telescopic property as

$$\begin{aligned} \mathbf{F}_i^- &= \mathbf{F}_i + \sum_{m_1=1}^{I} \left(\widetilde{\lambda}^- \alpha\theta\widetilde{\mathbf{e}}\right)^{m_1}_{i+\frac{1}{2}} \\ \mathbf{F}^+_{i+1} &= \mathbf{F}_{i+1} - \sum_{m_1=I+1}^{N_\lambda} \left(\widetilde{\lambda}^+ \alpha\theta\widetilde{\mathbf{e}}\right)^{m_1}_{i+\frac{1}{2}} \end{aligned} \tag{A.41}$$

The corresponding intercell numerical fluxes for the approximate first order Godunov's method are given by

$$\mathbf{F}^-_{i+\frac{1}{2}} = \mathbf{F}_i^- \qquad \mathbf{F}^+_{i-\frac{1}{2}} = \mathbf{F}_i^+ \tag{A.42}$$

and updating expression in (3.13) yields

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - (\mathbf{F}_i^- - \mathbf{F}_i^+)\frac{\Delta t}{\Delta x} \tag{A.43}$$

Notice that source term is accounted for in the numerical fluxes and therefore no explicit contribution of the source appears in (A.43) as in (3.13).

Appendix B

WENO reconstruction procedures

The preservation of high accuracy in both space and time for system of conservation laws with source terms has been and is a major step in the resolution of complex flows. If a reconstruction procedure is performed to provide a high order approximation of the conserved variables, fluxes and source terms, it must be considered that discontinuous solutions may be present. Discontinuities may introduce spurious oscillations in the numerical solution and the choice of a proper reconstruction technique is decisive for their rejection.

In this chapter, the WENO method is introduced. The acronym of WENO stands for *Weighted Essentially Non-Oscillatory*. Its name arises from the way data is reconstructed and how the solution behaves around discontinuities: any possible oscillatory behavior is eliminated, leading to a very stable non-oscillatory reconstruction.

Before the appearance of the WENO method, many other approaches addressed the issue of the generation of spurious oscillations in finite differences schemes, leading to the family of total-variation diminishing (TVD) schemes [26]. Later on, in the search of appropriate reconstruction techniques, the essentially non-oscillatory (ENO) method was proposed by Harten et al. [17]. Based on the definition of an smoothness indicator, the ENO method selects among different candidate stencils. The stencil in which the solution is smoothest is selected, avoiding oscillatory effects produced by the discontinuities. Founded in the ENO approach, the WENO method was then developed by Liu et al. in [18], allowing a r-th order ENO reconstruction be transformed into an $(r+1)$-th order WENO reconstruction.

The WENO reconstruction procedure uses a dynamic set of stencils where lower order polynomials are constructed first. These lower order polynomials are combined either to create a higher order polynomial in smooth regions (optimal reconstruction) or an off-center reconstruction able to capture discontinuities in non-smooth regions. The definition of a smoothness indicator allows to distinguish between both cases. Also, it is desirable that the selected indicator preserves the desired order of accuracy in smooth regions while retaining the essentially non-oscillatory property. Focusing in the preservation of the order of accuracy, Jiang and Shu [33] proposed a smoothness indicator linked to each small stencil, leading to an improved 5-th order WENO method. This indicator was established as the basis of an arbitrary order WENO method, referred here to as WENO-JS.

We will first review simple data reconstruction in 1D, focusing on the WENO method afterward. The two first sections of this chapter are based on the work of C.W. Shu presented in [20].

B.1 Interpolation and reconstruction in 1D

In this section, the problem of data reconstruction at an arbitrary point inside a cell by means of polynomial interpolation when departing from cell averages is considered.

The function $u(x)$ will be defined departing from the starting data, that will be considered as the average value of this function in each cell. The definition of $u(x)$ is useful for the derivation of the reconstruction procedure but its analytical expression will be unknown in most cases. The computational grid, shown in Figure B.1, is composed by N cells as

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < ... < x_{N-\frac{1}{2}} < x_{N+\frac{1}{2}} = b \tag{B.1}$$

with cells and cell sizes defined by

$$I_i = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] \tag{B.2}$$

$$\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} \equiv \text{constant} \tag{B.3}$$

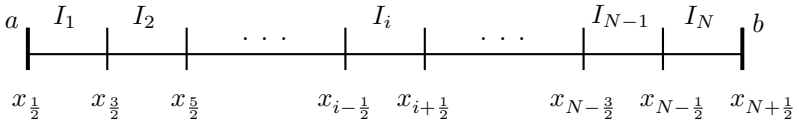

Figure B.1: Mesh discretization

With the previous definitions, the starting data set is now defined as the the average value of the function $u(x)$ in each cell

$$\bar{u}_i = \frac{1}{\Delta x_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(\xi)\, d\xi, \quad i = 1, 2, ..., N \tag{B.4}$$

The problem we face is to find a polynomial $p_r(x)$ of **degree at most** $k-1$ for each cell I_i, such that it is a k**-th order accurate** approximation of the function $u(x)$ inside I_i

$$p_r(x) = u(x) + O\left(\Delta x^k\right), \quad x \in I_i, \quad i = 1, 2, ..., N \tag{B.5}$$

The polynomial in (B.5) provides an approximation to the values of the function at the boundaries of cell I_i when evaluating $p_r(x)$ at $x_{i+\frac{1}{2}}$ and $x_{i-\frac{1}{2}}$, as follows

$$u^+_{i-\frac{1}{2}} = p_r\left(x_{i-\frac{1}{2}}\right), \quad u^-_{i+\frac{1}{2}} = p_r\left(x_{i+\frac{1}{2}}\right), \quad i = 1, 2, ..., N \tag{B.6}$$

Due to the properties of this form of interpolation, the resulting values for the approximation at the cell boundaries (B.6) will be defined as a linear combination of cell averages [20]. This linear combination is given by a set of constants c_{rj} which depend on the polynomial degree and the grid geometry, but not on the function $u(x)$. The expression for the approximation to the values of the function at the cell boundaries is written as

$$u^-_{i+\frac{1}{2}} = \sum_{j=0}^{k-1} c_{rj} \bar{u}_{i-r+j}, \quad u^+_{i-\frac{1}{2}} = \sum_{j=0}^{k-1} \tilde{c}_{rj} \bar{u}_{i-r+j} \tag{B.7}$$

with $\tilde{c}_{rj} = c_{r-1,j}$.

The reconstructed values at the cell boundaries are a k-th approximation to those of the function $u(x)$ at these points

$$u^-_{i+\frac{1}{2}} = u\left(x_{i+\frac{1}{2}}\right) + O\left(\Delta x^k\right), \quad u^+_{i-\frac{1}{2}} = u\left(x_{i-\frac{1}{2}}\right) + O\left(\Delta x^k\right) \tag{B.8}$$

The derivation of (B.7) is detailed next. First of all, it is necessary to clarify some notions. First, the concept of *stencil* is introduced. A stencil is defined as a group of connected cells. In this section, the reconstruction will be performed using information contained in only one stencil. Therefore, each cell, I_i, will be linked to a stencil $S_r(i)$ composed by cell I_i plus r cells to the left and s cells to the right. Hence, **the number of cells in the stencil** will be $r + s + 1$ which **agrees with the order of accuracy of**

the polynomial for that stencil, $k = r + s + 1$. For all cases, the condition $r, s \geq 0$ must be satisfied. The stencil will be denoted by:

$$S_r\left(i\right) = \{I_{i-r}, ..., I_i, ..., I_{i+s}\} \tag{B.9}$$

It is worth mentioning that polynomial $p_r(x)$ refers to stencil $S_r\left(i\right)$. Therefore, it is possible to define k $p_r(x)$ independent polynomials of k-th order, with r variable, that will be used to provide information inside cell I_i. This will be used for the WENO reconstruction procedure in the next chapter.

The steps required to generate the reconstructing polynomial departing from cell averages are listed below:

a) Stencil selection.

Given the cell I_i and the order of accuracy required k, we must first choose a stencil $S_r\left(i\right)$ with $k = r + s + 1$ cells.

There is a unique polynomial $p_r\left(x\right)$ of degree at most $k - 1$ whose cell average value for each cell in the stencil agrees with that of the function $u\left(x\right)$ [20]

$$\frac{1}{\Delta x_j}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} p_r\left(\xi\right) d\xi = \bar{u}_j, \qquad j = i - r, ..., i + s \tag{B.10}$$

b) Definition of the primitive function.

In order to find the interpolating polynomial $p_r(x)$ of degree $k - 1$ and k-th order of accuracy, a new function is introduced. This new function is the primitive function of $u(x)$, denoted by $U(x)$, which is defined as the cumulative integral of $u(x)$ from $-\infty$ to x.

$$U\left(x\right) = \int_{-\infty}^{x} u\left(\xi\right) d\xi \tag{B.11}$$

For a random location in the grid, i, the value of this cumulative integral at the right boundary of the cell I_i can be computed by the summation of the average values of each cell multiplied by the cell size, from $-\infty$ to the cell I_i, as follows:

$$U\left(x_{i+\frac{1}{2}}\right) = \int_{-\infty}^{x_{i+\frac{1}{2}}} u\left(\xi\right) d\xi = \sum_{j=-\infty}^{i} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u\left(\xi\right) d\xi = \sum_{j=-\infty}^{i} \bar{u}_j \Delta x_j \tag{B.12}$$

Also, a polynomial $P_r(x)$ is defined as the **unique polynomial of degree at most k which interpolates $U(x)$ with $k + 1$-th order of accuracy in $k + 1$ nodes** (which are all the cell boundaries in the stencil) and we denote its derivative by $p_r(x)$:

$$p_r(x) = P_r'(x) \tag{B.13}$$

Note that $p_r(x)$ is a polynomial of degree $k - 1$ and k-th order, defined by k cells. Polynomial $P_r(x)$ is one order greater, and as the number of cells does not change, $k + 1$ interpolation points are necessary. It is worth noticing that this new $k + 1$ points are defined in the nodes, even the value of $u(x)$ is not a priori defined at these locations.

Using $P_r'(x)$ it is possible to prove the equality in (B.10)

$$\begin{aligned}
&\frac{1}{\Delta x_j}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} p_r\left(\xi\right) d\xi = \frac{1}{\Delta x_j}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} P_r'\left(\xi\right) d\xi = \frac{1}{\Delta x_j}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} dP_r\left(\xi\right) = \\
&= \frac{1}{\Delta x_j}\left(P_r\left(x_{j+\frac{1}{2}}\right) - P_r\left(x_{j-\frac{1}{2}}\right)\right) \approx \frac{1}{\Delta x_j}\left(U\left(x_{j+\frac{1}{2}}\right) - U\left(x_{j-\frac{1}{2}}\right)\right) = \\
&= \frac{1}{\Delta x_j}\left(\int_{-\infty}^{x_{j+\frac{1}{2}}} u\left(\xi\right) d\xi - \int_{-\infty}^{x_{j-\frac{1}{2}}} u\left(\xi\right) d\xi\right) = \frac{1}{\Delta x_j}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u\left(\xi\right) d\xi = \bar{u}_j
\end{aligned} \tag{B.14}$$

for any $j = i - r, ..., i + s$, being j the subscript that indicates the cell of the stencil we are dealing with. The approximation symbol stands for the approximation of $U(x)$ by the interpolating polynomial $P_r(x)$. This interpolation is a $k+1$-th order approximation

$$P_r(x) = U(x) + O\left(\Delta x^{k+1}\right), \qquad x \in I_i \tag{B.15}$$

and that of its derivative, a k-th order approximation

$$P_r'(x) = U'(x) + O\left(\Delta x^{k}\right), \qquad x \in I_i \tag{B.16}$$

Therefore it can be concluded that we must first get $P_r(x)$ by interpolating the primitive function $U(x)$ and then we must take the derivative of $P_r(x)$ to find $p_r(x)$.

c) Lagrange interpolation

In [20], the use of the Lagrange form of the interpolating polynomial is proposed to achieve what it is conveyed in the previous lines. This kind of interpolation is said to be nodal since each weight takes the value of 1 in its node and 0 in the rest of the nodes. Therefore, the result of the interpolation at each node is the value of the function at that node, since the other terms of the summation will be zero and have no contribution. The generic expression for the Lagrange interpolating polynomial, at $b+1$ nodes $(x_0, y(x_0))...(x_b, y(x_b))$, for a function $y(x)$, is as follows:

$$L(x) = \sum_{i=0}^{b} y(x_i) l_i(x) \tag{B.17}$$

with the weighting functions

$$l_i(x) = \prod_{\substack{l=0 \\ l \neq i}}^{b} \frac{(x - x_l)}{(x_i - x_l)} \tag{B.18}$$

The plots of the weighting functions $l_i(x)$ is shown in Figure B.2. In this case, the number of nodes for the interpolation is 4 reaching a 4-th order of accuracy. The cell size Δx is constant. As we can see in Figure B.2, $l_1(x)$ is equal to 1 at $x = 0$, $l_2(x)$ is 1 at $x = 1$ and so on.

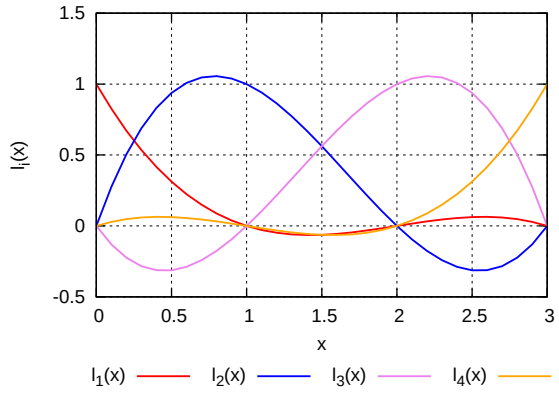

Figure B.2: Weighting functions for $k = 3$, with nodes $x_i = \{0, 1, 2, 3\}$

Now, we write the expression for $P_r(x)$ following the Lagrange interpolating polynomial in (B.17), by imposing the values of function $U(x)$ at the $k+1$ nodes of the stencil $S(i)$:

$$P_r(x) = \sum_{m=0}^{k} U(x_{i-r+m-\frac{1}{2}}) \prod_{\substack{l=0 \\ l \neq m}}^{k} \frac{(x - x_{i-r+l-\frac{1}{2}})}{(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}})} \tag{B.19}$$

For an easier manipulation, the constant value $U(x_{i-r-\frac{1}{2}})$ is going to be subtracted from the previous expression (B.19). This way, the starting point for the calculation of the integral will shift from $-\infty$ to the first wall of the stencil.

$$P_r(x) - U(x_{i-r-\frac{1}{2}}) = \sum_{m=0}^{k} \left(U(x_{i-r+m-\frac{1}{2}}) - U(x_{i-r-\frac{1}{2}}) \right) \prod_{\substack{l=0 \\ l\neq m}}^{k} \frac{(x - x_{i-r+l-\frac{1}{2}})}{(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}})} \tag{B.20}$$

The difference between the primitive function evaluated in any wall of the stencil, $U(x_{i-r+m-\frac{1}{2}})$, and the same function evaluated in the first wall of the stencil, $U(x_{i-r-\frac{1}{2}})$, will be a measure of the cumulative integral from the beginning of the stencil to that wall at $x_{i-r+m-\frac{1}{2}}$. This can be clearly seen in the following expression

$$U(x_{i-r+m-\frac{1}{2}}) - U(x_{i-r-\frac{1}{2}}) = \sum_{j=0}^{m-1} \bar{u}_{i-r+j} \Delta x_{i-r+j} \tag{B.21}$$

Taking the derivative on both terms of (B.20) and noticing the previous equality, we get the expression for the polynomial $p_r(x)$, which performs the reconstruction using the average values of $u(x)$ in the cells, unlike $P_r(x)$, which used boundary values

$$p_r(x) = \sum_{m=0}^{k} \sum_{j=0}^{m-1} \bar{u}_{i-r+j} \Delta x_{i-r+j} \left(\frac{\sum_{\substack{l=0 \\ l\neq m}}^{k} \prod_{\substack{q=0 \\ q\neq m,l}}^{k} \left(x - x_{i-r+q-\frac{1}{2}} \right)}{\prod_{\substack{l=0 \\ l\neq m}}^{k} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}} \right)} \right) \tag{B.22}$$

A simpler expression for $p_r(x)$ can be derived from equation (B.22) taking the cell averages as common factors. The resulting expression represents the reconstructing polynomial as a linear combination of the cell averages as

$$p_r(x) = \sum_{j=0}^{k-1} \left(\sum_{m=j+1}^{k} \frac{\sum_{\substack{l=0 \\ l\neq m}}^{k} \prod_{\substack{q=0 \\ q\neq m,l}}^{k} \left(x - x_{i-r+q-\frac{1}{2}} \right)}{\prod_{\substack{l=0 \\ l\neq m}}^{k} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}} \right)} \right) \bar{u}_{i-r+j} \Delta x_{i-r+j}, \qquad r = 0, ..., k-1 \tag{B.23}$$

If defining

$$C_{rj}^{(k)}(x) = \left(\sum_{m=j+1}^{k} \frac{\sum_{\substack{l=0 \\ l\neq m}}^{k} \prod_{\substack{q=0 \\ q\neq m,l}}^{k} \left(x - x_{i-r+q-\frac{1}{2}} \right)}{\prod_{\substack{l=0 \\ l\neq m}}^{k} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}} \right)} \right) \Delta x_{i-r+j} \tag{B.24}$$

it is possible to express Equation (B.23) as

$$p_r(x) = \sum_{j=0}^{k-1} C_{rj}^{(k)}(x) \bar{u}_{i-r+j}, \qquad r = 0, ..., k-1 \tag{B.25}$$

Where $C_{rj}^{(k)}(x)$ are constants at a given x and provide the weights for the linear combination of cell averages. The superscript k of the linear coefficient $C_{rj}^{(k)}(x)$ stands for the dimension of the stencil where $p_r(x)$ is defined, it is useful to not mix up these coefficients when different stencils are used at the same time.

For the sake of clarity, the evaluation of $C_{rj}^{(k)}(x)$ at $x_{i+\frac{1}{2}}$ or $x_{i-\frac{1}{2}}$ will be denoted as $c_{rj}^{(k)}$ and $\tilde{c}_{rj}^{(k)}$, respectively, as follows

$$C_{rj}^{(k)}(x = x_{i+\frac{1}{2}}) = c_{rj}^{(k)}, \qquad C_{rj}^{(k)}(x = x_{i-\frac{1}{2}}) = \tilde{c}_{rj}^{(k)} \tag{B.26}$$

Expression in (B.25) can expressed in a more compact form as

$$p(r, k, \nu, \bar{\mathbf{v}}) = p_r(\nu) = \sum_{j=0}^{k-1} C_{rj}^{(k)}(\nu)\, \bar{v}_j, \tag{B.27}$$

where r and k describe the stencil and the position of the reconstruction cell, ν stands for the spatial variable and $\bar{\mathbf{v}}$ for the vector of cell averages in the stencil, with components $\bar{v}_j = \bar{u}_{i-r+j}$ for $j = 0, ..., k-1$.

d) Computation of the linear $c_{rj}^{(k)}$ coefficients.

The evaluation of Equation (B.23) at $x_{i+\frac{1}{2}}$ (right boundary of the cell I_i) provides the approximation to the value $u(x_{i+\frac{1}{2}})$, denoted by $u_{i+\frac{1}{2}}$

$$u_{i+\frac{1}{2}} = p_i(x_{i+\frac{1}{2}}) = \sum_{j=0}^{k-1} \left(\sum_{m=j+1}^{k} \frac{\sum_{\substack{l=0 \\ l\neq m}}^{k} \prod_{\substack{q=0 \\ q\neq m,l}}^{k} \left(x_{i+\frac{1}{2}} - x_{i-r+q-\frac{1}{2}} \right)}{\prod_{\substack{l=0 \\ l\neq m}}^{k} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}} \right)} \right) \bar{u}_{i-r+j} \Delta x_{i-r+j} \tag{B.28}$$

As outlined before, this expression may be seen as a summation of constants, denoted by $c_{rj}^{(k)}$, multiplied by the cell averages $\bar{u}_{i-r+j}$, following the equation (B.7). The expression for these constants $c_{rj}^{(k)}$ results from the evaluation of (B.24) at $x_{i+\frac{1}{2}}$

$$c_{rj}^{(k)} = \left(\sum_{m=j+1}^{k} \frac{\sum_{\substack{l=0 \\ l\neq m}}^{k} \prod_{\substack{q=0 \\ q\neq m,l}}^{k} \left(x_{i+\frac{1}{2}} - x_{i-r+q-\frac{1}{2}} \right)}{\prod_{\substack{l=0 \\ l\neq m}}^{k} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}} \right)} \right) \Delta x_{i-r+j} \tag{B.29}$$

If we now rewrite this expression for the particular case of uniform grid (constant Δx), we finally obtain

$$c_{rj}^{(k)} = \sum_{m=j+1}^{k} \frac{\sum_{\substack{l=0 \\ l\neq m}}^{k} \prod_{\substack{q=0 \\ q\neq m,l}}^{k} (r - q + 1)}{\prod_{\substack{l=0 \\ l\neq m}}^{k} (m - l)} \tag{B.30}$$

B.2 Weighted Essentially Non-Oscillatory (WENO) reconstruction

In this section, the procedure to construct a WENO reconstruction are provided. Before starting, the reader must notice that there are different sorts of problems for which WENO procedures are designed such as WENO interpolation, WENO integration, WENO approximation to the first derivative and WENO reconstruction [32]. WENO interpolation departs from pointwise information instead of cell-averaged values and it is used in finite difference methods. WENO integration provides an approximation to the integral of a function, given its values at grid points. WENO approximation to the first derivative and WENO reconstruction are equivalent and depart from the cell averages of a function.

The case analyzed in this text is the WENO reconstruction: from cell averages, we have to compute the value of the function at the cell boundaries. This procedure is widely used in the numerical solution of conservation laws.

In the previous section it was detailed how to perform a simple data reconstruction using linear interpolation: from cell averages in the stencil $S_r(i)$, an approximation to the cell boundary values of cell I_i was computed. Now, the procedure goes further and the reconstruction will depend on the

shape of the function (on its smoothness), preventing the solution from being oscillatory. Moreover, the starting data set for the interpolation will be broader than in the previous case. Instead of computing the approximation of the function inside one cell with the data stored in only one stencil of k cells, a combination of k different stencils composed of k cells each one will be used. This leads to a reconstruction of $2k-1$-th order of accuracy.

The smoothness of the function inside each stencil is measured by a suitable smoothness indicator. The final reconstruction combines the k different stencils, where the weight associated to each of them is determined by this indicator.

The reconstruction is computed in two steps. The first one is related to the calculation of the coefficients that ensure the equality between the polynomial high order approximation in the big stencil and the linear combination of polynomial lower order approximations in the smaller stencils. These coefficients will be referred to as *optimal weights*. The second step focuses on the calculation of the non-oscillatory weights, modifying the optimal weights by means of the smoothness indicators.

B.2.1 First part: Computation of the optimal weights

Before starting, a reconstruction domain must be chosen. In the previous section, the reconstruction procedure used data from only one stencil (composed of k cells). It was shown that depending on the selection of r, and keeping in mind that $k = r+s+1$, k different $p_r(x)$ polynomials associated to k different stencils could be found to approximate the value of $u(x)$ inside a cell. The keystone of the WENO method is to combine these k different $p_r(x)$ polynomials to generate a $(2k-1)$-th order reconstruction.

To construct a **WENO reconstruction of $(2k-1)$-th order** on the cell $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ for the function $u(x)$, we need the k **different stencils** for $r = 0, ..., k-1$, denoted by $S_r(i)$ and defined as

$$S_r(i) = \{I_{i-r}, ..., I_{i+k-r-1}\}, \qquad r = 0, ..., k-1 \tag{B.31}$$

where $s = k - r - 1$.

Stencils $S_r(i)$ are overlapped on the interval $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, as follows

$$\cap_{r=0}^{k-1} S_r(i) = I_i \tag{B.32}$$

Then, they are used to generate a bigger stencil that will contain all the cells from the smaller stencils, denoted by

$$\mathcal{T}(i) = \cup_{r=0}^{k-1} S_r(i) = \{I_{i-k+1}, ..., I_{i+k-1}\} \tag{B.33}$$

As it was defined in equation (B.10), there is a **unique polynomial $p_r(x)$ associated to each stencil S_r**, which is a **k-th order approximation** to the function $u(x)$ on the stencil $S_r(i)$ if this function is smooth inside it, as follows

$$p_r(x) = u(x) + O\left(\Delta x^k\right), \qquad x \in S_r,\ r = 0, ..., k-1 \tag{B.34}$$

The expression for $p_r(x)$ was derived in equation (B.23). If we evaluate it at $x_{i+\frac{1}{2}}$ or $x_{i-\frac{1}{2}}$, it provides approximations to the cell boundary values

$$u^{(r)}_{i+\frac{1}{2}} = p_r(x_{i+\frac{1}{2}}) = \sum_{j=0}^{k-1} c^{(k)}_{rj} \bar{u}_{i-r+j}, \qquad u^{(r)}_{i-\frac{1}{2}} = p_r(x_{i-\frac{1}{2}}) = \sum_{j=0}^{k-1} \tilde{c}^{(k)}_{rj} \bar{u}_{i-r+j} \tag{B.35}$$

The procedure in (B.23) can be extended to obtain a **polynomial $q(x)$, which is a $2k-1$-th order accurate approximation of the function $u(x)$ on the big stencil $\mathcal{T}(i)$**, denoted by

$$q(x) = \sum_{j=1}^{2k-1} \left(\sum_{m=j+1}^{2k} \frac{\sum_{\substack{l=1 \\ l\neq m}}^{2k} \prod_{\substack{q=1 \\ q\neq m,l}}^{2k} \left(x - x_{i-k+q-\frac{1}{2}}\right)}{\prod_{\substack{l=1 \\ l\neq m}}^{2k} \left(x_{i-k+m-\frac{1}{2}} - x_{i-k+l-\frac{1}{2}}\right)} \right) \bar{u}_{i-k+j} \Delta x_{i-k+j} \tag{B.36}$$

that can be written as

$$q(x) = \sum_{j=1}^{2k-1} C_{rj}^{(2k-1)}(x)\bar{u}_{i-k+j} \tag{B.37}$$

where superscript $2k-1$ only refers to the order of the approximation. The approximation at the right boundary of I_i is denoted as

$$u_{i+\frac{1}{2}} = q(x = x_{i+\frac{1}{2}}) = \sum_{j=1}^{2k-1} c_{k-1,j}^{(2k-1)}\bar{u}_{i-k+j} \tag{B.38}$$

Note that in (B.38), the value of r is fixed, $r = k-1$, as the big stencil $\mathcal{T}(i)$ is always symmetric. Now, the goal is to express the coefficients $c_{k-1,j}^{(2k-1)}$ of the big stencil as a linear combination of the previously computed coefficients $c_{rj}^{(k)}$ obtained for the small stencils. By doing this, it will be possible to express polynomial $q(x)$ in terms of the k $p_r(x)$ polynomials. At a certain point x, the evaluation of $q(x)$ will be expressed as a linear combination of the evaluations provided by $p_r(x)$ and the coefficients that provide this linear combination are the so called optimal weights.

Computation of optimal weights for a 5-th Order WENO reconstruction

For the sake of clarity, a simple example of the procedure for the computation of the optimal weights is given in this text. The details concerning the calculation of the *optimal weights* for a 5-th order WENO reconstruction, based on 3-cell stencil reconstruction with 3-th order polynomials ($k = 3$) are given below. A uniform grid will be assumed.

The three different stencils are given by

$$\begin{array}{l} S_0(i) = \{I_i, I_{i+1}, I_{i+2}\} \\ S_1(i) = \{I_{i-1}, I_i, I_{i+1}\} \\ S_2(i) = \{I_{i-2}, I_{i-1}, I_i\} \end{array} \tag{B.39}$$

and the stencil $\mathcal{T}(i)$ can be constructed

$$\mathcal{T}(i) = S_0 \cup S_1 \cup S_2 = \{I_{i-2}, I_{i-1}, I_i, I_{i+1}, I_{i+2}\} \tag{B.40}$$

Stencils in (B.39) and (B.40) are depicted in Figure B.3 for a random cell I_i.

For each stencil, boundary values of $u(x)$ in I_i are obtained using (B.35). At the right boundary, the 3 polyomials $p_r(x)$ provide the following approximations

$$u_{i+\frac{1}{2}}^{(0)} = \sum_{j=0}^{2} c_{0j}^{(3)}\bar{u}_{i+j} = \frac{1}{3}\bar{u}_i + \frac{5}{6}\bar{u}_{i+1} - \frac{1}{6}\bar{u}_{i+2} \tag{B.41}$$

$$u_{i+\frac{1}{2}}^{(1)} = \sum_{j=0}^{2} c_{1j}^{(3)}\bar{u}_{i-1+j} = -\frac{1}{6}\bar{u}_{i-1} + \frac{5}{6}\bar{u}_i + \frac{1}{3}\bar{u}_{i+1} \tag{B.42}$$

$$u_{i+\frac{1}{2}}^{(2)} = \sum_{j=0}^{2} c_{2j}^{(3)}\bar{u}_{i-2+j} = \frac{1}{3}\bar{u}_{i-2} - \frac{7}{6}\bar{u}_{i-1} - \frac{11}{6}\bar{u}_i \tag{B.43}$$

which are a 3-th order approximation to the value of the function $u(x)$ at $x_{i+\frac{1}{2}}$ if the function is smooth inside each stencil [32]

$$u_{i+\frac{1}{2}}^{(0)} = u\left(x_{i+\frac{1}{2}}\right) + O\left(\Delta x^3\right), \qquad \text{if } u(x) \text{ is smooth inside } S_0(i) \tag{B.44}$$

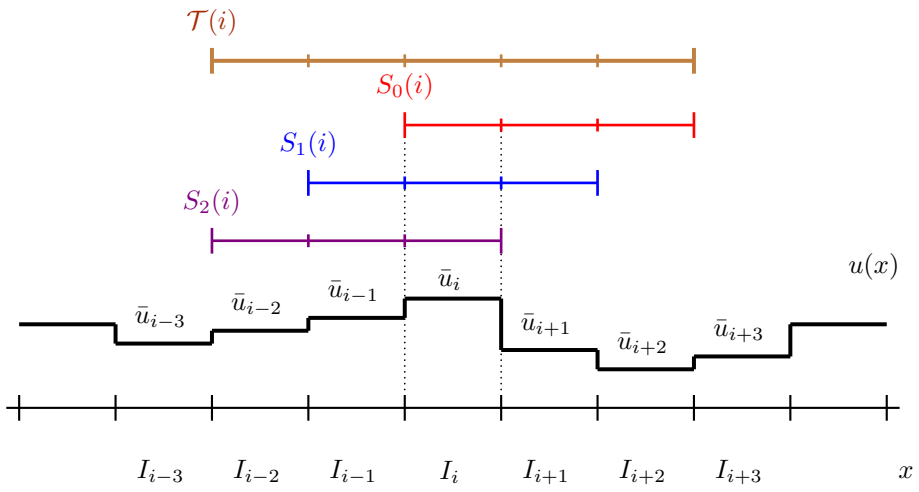

Figure B.3: Stencil combination for a 5-*th* order WENO reconstruction

$$u^{(1)}_{i+\frac{1}{2}} = u\left(x_{i+\frac{1}{2}}\right) + O\left(\Delta x^3\right), \qquad \text{if } u(x) \text{ is smooth inside } S_1(i) \tag{B.45}$$

$$u^{(2)}_{i+\frac{1}{2}} = u\left(x_{i+\frac{1}{2}}\right) + O\left(\Delta x^3\right), \qquad \text{if } u(x) \text{ is smooth inside } S_2(i) \tag{B.46}$$

When the 5-th order centered reconstruction generated using $q(x)$, Equation (B.35) leads to

$$u_{i+\frac{1}{2}} = \sum_{j=1}^{5} c^{(5)}_{2j}\,\bar{u}_{i-3+j} = \frac{1}{30}\bar{u}_{i-2} - \frac{13}{60}\bar{u}_{i-1} + \frac{47}{60}\bar{u}_i + \frac{9}{20}\bar{u}_{i+1} - \frac{1}{20}\bar{u}_{i+2} \tag{B.47}$$

which is a 5-th order approximation to the value of the function $u(x)$ at the boundary $x_{i+\frac{1}{2}}$ if the function is smooth inside $\mathcal{T}(i)$

$$u_{i+\frac{1}{2}} = u\left(x_{i+\frac{1}{2}}\right) + O\left(\Delta x^5\right) \tag{B.48}$$

The 5-th order reconstruction in (B.47) may also be expressed as a convex combination of the 3-th order approximations, $u^{(r)}_{i+\frac{1}{2}}$ in (B.41-B.43). The coefficients that determine this combination will be unique and denoted by γ_0, γ_1, γ_2, giving

$$u_{i+\frac{1}{2}} = \gamma_0 u^{(0)}_{i+\frac{1}{2}} + \gamma_1 u^{(1)}_{i+\frac{1}{2}} + \gamma_2 u^{(2)}_{i+\frac{1}{2}} \tag{B.49}$$

These coefficients are called *optimal weights*. They can be easily computed by imposing the equality between (B.47) and (B.49) as

$$\begin{aligned}\frac{1}{30}\bar{u}_{i-2} - \frac{13}{60}\bar{u}_{i-1} + \frac{47}{60}\bar{u}_i + \frac{9}{20}\bar{u}_{i+1} - \frac{1}{20}\bar{u}_{i+2} = \gamma_0\left(\frac{1}{3}\bar{u}_i + \frac{5}{6}\bar{u}_{i+1} - \frac{1}{6}\bar{u}_{i+2}\right) + \\ +\gamma 1\left(-\frac{1}{6}\bar{u}_{i-1} + \frac{5}{6}\bar{u}_i + \frac{1}{3}\bar{u}_{i+1}\right) + \gamma_2\left(\frac{1}{3}\bar{u}_{i-2} - \frac{7}{6}\bar{u}_{i-1} - \frac{11}{6}\bar{u}_i\right)\end{aligned} \tag{B.50}$$

From equation (B.50) we can obtain the 3 coefficients in (B.49) formulating 2 different equations to satisfy the equality of weights for 2 cell averages in this particular case, and one more equation to satisfy the unit sum of the weights $\sum_{r=0}^{k-1}\gamma_r = 1$. The following equations, starting from $\bar{u}_{i-2}$, appear

a) The equality of weights for $\bar{u}_{i-2}$ leads to

$$\begin{aligned} \gamma_2 \frac{1}{3} &= \frac{1}{30} \\ \gamma_2 &= \frac{1}{10} \end{aligned} \tag{B.51}$$

b) And the same for $\bar{u}_{i-1}$ gives

$$\begin{aligned} -\gamma_2 \frac{7}{6} - \gamma_1 \frac{1}{6} &= -\frac{13}{60} \\ \gamma_1 &= \frac{6}{10} \end{aligned} \tag{B.52}$$

c) Finally, an additional equation to fulfill the unit sum allows to compute γ_0

$$\begin{aligned} \gamma_2 + \gamma_1 + \gamma_0 &= 1 \\ \gamma_0 &= \frac{3}{10} \end{aligned} \tag{B.53}$$

The same could be done to compute the weights at the left boundary, denoted by $\tilde{\gamma}_r$.

Generalization of the procedure

The generalization of the procedure and the steps for the computation of the optimal weights for a $(2k-1)$-th order accurate approximation are presented next. First, the $(2k-1)$-th order polynomial, $q(x)$, is expressed in terms of the k-th order polynomials, $p_r(x)$, as

$$q\left(x\right) = \sum_{r=0}^{k-1} \Gamma_r\left(x\right) p_r\left(x\right) \tag{B.54}$$

where $\Gamma_r\left(x\right)$ are the weights, which are rational functions [32]. Inserting in this equation the expressions for the polynomials $p_r(x)$ in (B.25) and $q(x)$ in (B.37), it yields

$$\sum_{j=1}^{2k-1} C_{k-1,j}^{(2k-1)}(x)\bar{u}_{i-k+j} = \sum_{r=0}^{k-1} \Gamma_r\left(x\right) \sum_{j=0}^{k-1} C_{rj}^{(k)}(x)\bar{u}_{i-r+j} \tag{B.55}$$

For the sake of clarity, Equation (B.55) is evaluated at $x = x_{i+\frac{1}{2}}$ though any other point could be used for this derivation. The sought optimal weights are only valid at the point where the evaluation is carried out, in this case at $x = x_{i+\frac{1}{2}}$. As a result of this evaluation, Equation (B.55) becomes

$$\sum_{j=1}^{2k-1} c_{k-1,j}^{(2k-1)}\bar{u}_{i-k+j} = \sum_{r=0}^{k-1} \gamma_r \sum_{j=0}^{k-1} c_{rj}^{(k)}\bar{u}_{i-r+j} \equiv u_{i+\frac{1}{2}} \tag{B.56}$$

where the reconstruction coefficients $c_{k-1,j}^{(2k-1)}$ and $c_{rj}^{(k)}$ are now constant according to (B.26) and with $\gamma_r = \Gamma_r\left(x = x_{i+\frac{1}{2}}\right)$ the sought weights.

From (B.56) and taking into account the condition $\sum_{r=0}^{k-1}\gamma_r = 1$, the following system of equations for the optimal weights γ_r is formulated

$$\mathbf{M}\cdot\gamma = \mathbf{c} \tag{B.57}$$

where $\mathbf{M} \in \mathbb{R}^{k\times k}$, $\gamma \in \mathbb{R}^k$ and $\mathbf{c} \in \mathbb{R}^k$

k	γ_0	γ_1	γ_2	γ_3	γ_4
5	$\frac{5}{126}$	$\frac{20}{63}$	$\frac{10}{21}$	$\frac{10}{63}$	$\frac{1}{126}$
4	$\frac{4}{35}$	$\frac{18}{35}$	$\frac{12}{35}$	$\frac{1}{35}$	
3	$\frac{3}{10}$	$\frac{6}{10}$	$\frac{1}{10}$		
2	$\frac{2}{3}$	$\frac{1}{3}$			
1	1				

Table B.1: Linear coefficients γ_r for $k = 1, 2, 3, 4, 5$

$$\begin{pmatrix} c_{k-1,0}^{(k)} & 0 & 0 & 0 & \cdots & 0 \\ c_{k-1,1}^{(k)} & c_{k-2,0}^{(k)} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & & \vdots \\ \vdots & \vdots & & \ddots & & \vdots \\ c_{k-1,k-2}^{(k)} & c_{k-2,k-3}^{(k)} & c_{k-3,k-4}^{(k)} & \cdots & c_{1,0}^{(k)} & 0 \\ 1 & 1 & 1 & \cdots & \cdots & 1 \end{pmatrix} \begin{pmatrix} \gamma_{k-1} \\ \gamma_{k-2} \\ \vdots \\ \gamma_0 \end{pmatrix} = \begin{pmatrix} c_{k-1,0}^{(2k-1)} \\ c_{k-1,1}^{(2k-1)} \\ \vdots \\ c_{k-1,k-1}^{(2k-1)} \\ 1 \end{pmatrix}$$

The components of the matrix $\mathbf{M}$ at a location (α, β) are given by

$$M_{\alpha,\beta} = \begin{cases} c_{k-\beta,\alpha-\beta}^{(k)} & \text{if } \alpha \geq \beta, \alpha \neq k \\ 1 & \text{if } \alpha = k \\ 0 & \text{if } \alpha < \beta \end{cases}$$

for $1 \leq \alpha \leq k$ and $1 \leq \beta \leq k$, where α stands for the row and β stands for the column inside the matrix.

Once the coefficients γ_r have been computed, the expression for the approximation to the value of $u(x)$ at the right boundary of I_i can be computed as follows

$$u_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} \gamma_r u_{i+\frac{1}{2}}^{(r)} \tag{B.58}$$

where $u_{i+\frac{1}{2}}$ is $(2k-1)$-th order accurate as long as the function $u(x)$ is smooth inside the stencil $\mathcal{T}(i)$. Table B.1 shows the coefficients γ_r computed using (B.57) for five different values of k, from 1 to 5. They can be used to construct up to a 9-th order reconstruction at $x_{i+1/2}$.

Recall that different optimal weights are obtained depending on the point at which the reconstruction has to be computed. Therefore, the same procedure should be repeated at each point where the reconstruction has to be calculated. For instance, to compute a $(2k-1)$-th order reconstruction at the left boundary of I_i we use

$$u_{i-\frac{1}{2}} = \sum_{r=0}^{k-1} \tilde{\gamma}_r u_{i-\frac{1}{2}}^{(r)} \tag{B.59}$$

and for the particular case of a uniform grid, $\tilde{\gamma}_r = \gamma_{k-1-r}$, for $r = 0, ..., k-1$, due to the symmetry of stencil $\mathcal{T}(i)$.

B.2.2 Second part: Calculation of the non-oscillatory weights

As outlined before, the approximation in (B.58) of the value of the function at the cell boundaries will be $(2k-1)$-th accurate as long as the function is smooth inside the big stencil $\mathcal{T}(i)$. If the function is

non-smooth or discontinuous, a lower order of accuracy is reached. Moreover, oscillations will appear due to the presence of discontinuities. This fact motivates the idea of using a modified set of coefficients instead of the optimal weights in order to reduce the weight of the contributions associated to those stencils including discontinuities.

Thus, instead of computing the $(2k-1)$-th order approximation using the optimal weights, γ_r, as in (B.58), non-oscillatory WENO weights, denoted by ω_r, will be used. The non-oscillatory reconstruction of $u(x)$ at the right cell boundary will be computed now as

$$u_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} \omega_r u_{i+\frac{1}{2}}^{(r)} \tag{B.60}$$

where the set of non-oscillatory weights ω_r is sought to provide a linear convex combination using the k different low order reconstructions. Therefore we require

$$\sum_{r=0}^{k-1} \omega_r = 1 \qquad \text{and} \qquad \omega_r \geq 0 \tag{B.61}$$

In the case when $u(x)$ is smooth, both expressions (B.58) and (B.60) are equivalent and provide a $(2k-1)$-th order approximation. According to the properties of the WENO reconstruction [33], WENO weights ω_r are a $k-1$-th order approximation to the optimal weights γ_r,

$$\omega_r = \gamma_r + O(\Delta x^{k-1}) \tag{B.62}$$

in smooth monotone regions. If there is a discontinuity in the stencil, then

$$\omega_r = \gamma_r + O(\Delta x) \tag{B.63}$$

In order to compute the WENO nonlinear weights ω_r, the nonlinear coefficients α_r are formulated first

$$\alpha_r = \frac{\gamma_r}{(\beta_r + \epsilon)^2}, \qquad r = 0, ..., k-1 \tag{B.64}$$

with ϵ a properly defined small parameter (see [31], p.4). The smoothness indicator, β_r, can be computed following [33],

$$\beta_r = \sum_{l=1}^{k-1} \int_{x_{i+\frac{1}{2}}}^{x_{i-\frac{1}{2}}} \Delta x^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial x^l} \right)^2 dx, \qquad r = 0, ..., k-1 \tag{B.65}$$

defined as the sum of the L^2 norms of all derivatives of the interpolating polynomial $p_r(x)$ over the interval $\left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]$. The term Δx^{2l-1} is included to remove Δx-dependent factors in the derivatives of the polynomials.

Once computed, the α_r coefficients are normalized so that their sum is equal to the unity, leading to the desired non-oscillatory weights

$$\omega_r = \frac{\alpha_r}{\sum_{l=0}^{k-1} \alpha_l}, \qquad r = 0, ..., k-1 \tag{B.66}$$

Repeating the procedure for $\tilde{\omega}_r$, which are the non-oscillatory weights associated to the linear weights at the left interface, $\tilde{\gamma}_r$, the WENO reconstruction of $u(x)$ at the cell boundaries will be computed as

$$u_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} \omega_r u_{i+\frac{1}{2}}^{(r)}, \qquad u_{i-\frac{1}{2}} = \sum_{r=0}^{k-1} \tilde{\omega}_r u_{i-\frac{1}{2}}^{(r)} \tag{B.67}$$

Considering again the general case, it was shown that $q(x)$ can be constructed in two different ways: in Equation (B.54) it was defined as a linear combination of lower order polynomials $p_r(x)$ while in

Equation (B.37) it was defined by directly constructing a $(2k-1)$-th order polynomial using the general polynomial reconstruction procedure. Following the first approach, it is possible to construct a polynomial $q(x)$ reducing the contribution of certain $p_r(x)$ polynomials which may generate oscillations. This new polynomial is defined, $q_{\text{WENO}}(x)$

$$q_{\text{WENO}}(x) = \sum_{r=0}^{k-1} \Omega_r(x)\, p_r(x) \tag{B.68}$$

where $\Omega_r(x)$ is the general expression for the WENO non-oscillatory weights, for instance

$$\Omega_r\left(x_{i+\frac{1}{2}}\right) = \omega_r \qquad \Omega_r\left(x_{i-\frac{1}{2}}\right) = \tilde{\omega}_r \tag{B.69}$$

For the sake of clarity when moving to 2D WENO reconstruction procedures, expression in (B.68) can be expressed in a more compact form as

$$q(k,\nu,\mathbf{p}) = q_{\text{WENO}}(\nu) = \sum_{r=0}^{k-1} \Omega_r(\nu)\, p_r(\nu) \tag{B.70}$$

where k is the size of the small stencils, ν stands for the spatial variable ($\nu \equiv x$ in this case) and $\mathbf{p} = \{p_r(\nu)\}_{r=0,\ldots,k-1}$ for the vector of low order polynomials used to generate the high order reconstruction.

Computation of β_r

In order to compute the smoothness indicator, β_r, using (B.65), a general expression for the the n-th derivative of $p_r(x)$ must be obtained departing from formulation in (B.25), as

$$\frac{\partial^n (p_r(x))}{\partial x^n} = \frac{\partial^n}{\partial x^n}\left(\sum_{j=0}^{k-1} C_{rj}^{(k)}(x)\bar{u}_{i-r+j}\right) = \sum_{j=0}^{k-1} \frac{\partial^n}{\partial x^n}\left(C_{rj}^{(k)}(x)\right)\bar{u}_{i-r+j} \tag{B.71}$$

with $C_{rj}^{(k)}(x)$ defined in (B.24). The term $\frac{\partial^n}{\partial x^n}\left(C_{rj}^{(k)}(x)\right)$ is the n-th derivative of the expression in (B.24), which is expressed as

$$\frac{\partial^n}{\partial x^n}\left(C_{rj}^{(k)}(x)\right) = \left(\sum_{m=j+1}^{k} \frac{\sum_{\substack{l_1=0\\ l_1\neq m}}^{k}\sum_{\substack{l_2=0\\ l_2\neq m,l_1}}^{k}\cdots\sum_{\substack{l_{n+1}=0\\ l_{n+1}\neq m,l_1,\ldots,l_n}}^{k}\prod_{\substack{q=0\\ q\neq m,l_1,\ldots,l_{n+1}}}^{k}\left(x - x_{i-r+q-\frac{1}{2}}\right)}{\prod_{\substack{l=0\\ l\neq m}}^{k}\left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}}\right)}\right)\Delta x_{i-r+j} \tag{B.72}$$

with $n = 1, \ldots, k-2$.

To compute β_r using (B.65), numerical integration must be carried out. A suitable quadrature formula must be used for the integration of the $k-2$ first terms of the summation. For the last term ($l = k-1$), numerical integration is not needed since the derivative is a constant value.

For instance, when using the 3-point Newton-Cotes quadrature rule, known as Simpson's rule and given by

$$\int_a^b f(x)\,dx \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right], \tag{B.73}$$

an approximate expression for the computations of β_r can be obtained by using (B.73) in (B.65). In case of a uniform grid, (B.65) becomes

$$\beta_r = \sum_{l=1}^{k-1} \int_{x_{i+\frac{1}{2}}}^{x_{i-\frac{1}{2}}} \Delta x^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial x^l} \right)^2 dx \approx$$

$$\approx \sum_{l=1}^{k-2} \frac{\Delta x}{6} \left[\Delta x^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial x^l} \right)^2_{x=x_{i-\frac{1}{2}}} + 4\Delta x^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial x^l} \right)^2_{x=x_i} + \Delta x^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial x^l} \right)^2_{x=x_{i+\frac{1}{2}}} \right] +$$

$$+\Delta x^{2k-3} \left(\frac{\partial^{k-1} p_r(x)}{\partial x^{k-1}} \right)^2 \Delta x \tag{B.74}$$

noticing that it is necessary the evaluation of the spatial derivatives of $p_r(x)$ at three different points: at the cell boundaries and at the center of the cell.

Using (B.71) it is possible to compute the value of the n-th derivative of $p_r(x)$ at a certain point. When having uniform grid, derivatives of the coefficients for (B.71) at the right boundary are given by

$$\frac{\partial^n}{\partial x^n} \left(c_{rj}^{(k)} \right) = \left(\sum_{m=j+1}^{k} \frac{\sum_{l_1=0}^{k} \sum_{l_2=0}^{k} \cdots \sum_{l_{n+1}=0}^{k} \prod_{q=0}^{k} (r-q+1)}{\prod_{l=0}^{k} (m-l)} \right) \Delta x^{-n} \tag{B.75}$$

where $c_{rj}^{(k)} = C_{rj}^{(k)}(x = x_{i+\frac{1}{2}})$ and $n = 1, ..., k-2$. Derivatives of degree up to $k-2$ at the cell center are given by

$$\frac{\partial^n}{\partial x^n} \left(\breve{c}_{rj}^{(k)} \right) = \left(\sum_{m=j+1}^{k} \frac{\sum_{l_1=0}^{k} \sum_{l_2=0}^{k} \cdots \sum_{l_{n+1}=0}^{k} \prod_{q=0}^{k} (r-q+1/2)}{\prod_{l=0}^{k} (m-l)} \right) \Delta x^{-n} \tag{B.76}$$

with $\breve{c}_{rj}^{(k)} = C_{rj}^{(k)}(x_i)$. Derivatives of degree up to $k-2$ at the left boundary can be obtained using

$$\frac{\partial^n}{\partial x^n} \left(\tilde{c}_{rj}^{(k)} \right) = \frac{\partial^n}{\partial x^n} \left(c_{r-1,j}^{(k)} \right) \tag{B.77}$$

with $\tilde{c}_{rj}^{(k)} = C_{rj}^{(k)}(x = x_{i-\frac{1}{2}})$

Equation (B.72) is only valid for the derivatives of order $n = 1, ..., k-2$, since the product term cannot be computed for the case when $n = k-1$. To calculate the $(k-1)$-th derivative, which is constant inside the cell, the following equation is used

$$\frac{\partial^{k-1}}{\partial x^{k-1}} \left(C_{rj}^{(k)}(x) \right) = \left(\sum_{m=j+1}^{k} \frac{k!}{\prod_{l=0}^{k} (m-l)} \right) \Delta x^{-(k-1)} \tag{B.78}$$

Numerical results of the computation of derivatives for Gaussian type function

The following Gaussian function is considered:

$$f(x) = 1 + e^{-\frac{(x+100)^2}{150}} \tag{B.79}$$

The first four derivatives of (B.79) are computed in the domain $x = [0, 200]$ using $k = 5$, $\Delta x = 2$ and $N = 100$. Numerical results at cell boundaries and cell center are plotted in Figure B.4 and compared with the exact solution. Notice that the fourth derivative shown in Figure B.4 is constant inside each cell since reconstructing polynomials are of 4-th degree when setting $k = 5$.

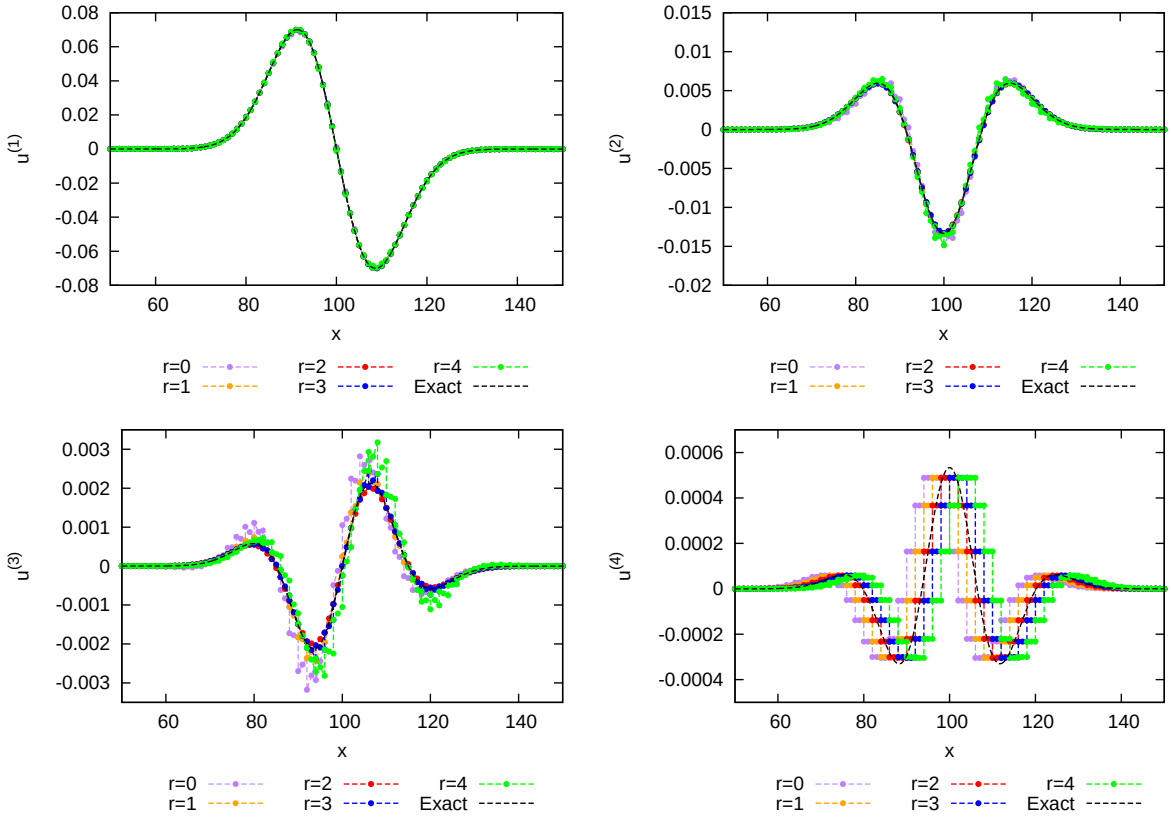

Figure B.4: Numerical results of the computation of first four derivatives of function in (B.79) using $k = 5$, $\Delta x = 2$ and $N = 100$.

B.3 Improved WENO procedures

B.3.1 WENO-5M

The mapped WENO approach was first introduced in [22] as a fix for the convergence issues that appeared at critical points when using the WENO-JS finite differences scheme. The resulting numerical scheme was only designed to reach fifth order of accuracy and was called WENO-5M [22].

In the WENO-JS scheme conditions in (B.61) and (B.62) were required to ensure the formal order of accuracy of the WENO reconstruction. But the achievement of formal order of accuracy require that WENO weights become a 3-rd order approximation of the optimal weights γ_r at critical points [22]. This constrain can be enforced through a mapping procedure. In [22] the following mapping function was proposed

$$g_r(\omega) = \frac{\omega(\gamma_r + \gamma_r^2 - 3\gamma_r\omega + \omega^2)}{\gamma_r^2 + \omega(1 - 2\gamma_r)} \qquad r = 0, 1, 2 \tag{B.80}$$

and combines the WENO-JS weights and the optimal weights. The WENO-5M weights, ω_r^M, are computed as follows

$$\alpha_r^M = g_r(\omega_r^{JS}) \qquad \omega_r^M = \frac{\alpha_r^M}{\sum_{l=0}^{k-1} \alpha_l^M}, \qquad r = 0, 1, 2 \tag{B.81}$$

Function $g_r(\omega)$ in (B.80) becomes flat in the neighborhood of the r-th optimal weight γ_r, and in smooth regions where the deviation of the original WENO-JS weights ω_r^{JS} from the optimal weights γ_r is relatively small, function $g_r(\omega)$ maps those weights providing more accurate values, closer to γ_r. In non-smooth regions the original weights ω_r^{JS} may get extreme values (close to 0 or 1), and $g_r(\omega)$ provides

a mapping close to the identity mapping ensuring $g_r(0) = 0$ and $g_r(1) = 1$. A drawback of this new technique is the extra computational cost needed for the rendering of the new weights [23].

B.3.2 WENO-Z

The WENO-Z weights in [23, 27] provide an alternative to the smoothness indicator β_r in (B.65), defining a more sophisticated indicator β_r^Z

$$\beta_r^Z = \frac{\beta_r + \epsilon}{\beta_r + \tau_{2k-1} + \epsilon} \tag{B.82}$$

where τ_{2k-1} is the global smoothness indicator, derived from the examination of the Taylor expansions of the Lagrange polynomials that provide the $c_{rj}^{(k)}$ coefficients in (B.35). This indicator will be either computed as $\tau_{2k-1} = |\beta_0 - \beta_{k-1}|$ when k is odd or computed as $\tau_{2k-1} = |\beta_0 - \beta_1 + \beta_{k-2} - \beta_{k-1}|$ when k is even. The general expression for the α_r^Z and ω_r^Z weights is given by

$$\alpha_r^Z = \frac{\gamma_r}{\beta_r^Z} = \gamma_r \left(1 + \left(\frac{\tau_{2r-1}}{\beta_r + \epsilon}\right)^{p_Z}\right) \qquad \omega_r^Z = \frac{\alpha_r^Z}{\sum_{l=0}^{k-1} \alpha_l^Z}, \qquad r = 0, ..., k-1 \tag{B.83}$$

with $p_Z = k-1$, ensuring the necessary order of accuracy of the non-oscillatory weights at critical points. Even both the WENO-5M and the WENO-Z schemes ensure all conditions to achieve the formal order of convergence, the WENO-Z scheme provides more accurate results around shocks avoiding the extra computational cost of a mapping procedure [23, 27].

B.3.3 The WENO-MZ method

In [42], an improved technique based on the combination of the WENO-M and WENO-Z methods was proposed. First, the non-oscillatory weights are calculated using the WENO-Z approach, following the procedure in Section B.3.2. Then, the ω_r^Z weights are mapped into new weights that should be closer to the optimal weights in smooth regions. These new weights will be denoted by ω_r^{MZ} weights and are computed following the procedure in Section B.3.1 as

$$\alpha_r^{MZ} = g_r(\omega_r^Z) \qquad \omega_r^{MZ} = \frac{\alpha_r^{MZ}}{\sum_{l=0}^{k-1} \alpha_l^{MZ}}, \qquad r = 0, ..., k-1 \tag{B.84}$$

where $g_r(\omega)$ is the mapping function in (B.80).

B.3.4 The WENO-PW method

The WENO-PW presented here is an extension of the WENO-JS method proposed in [19]. This new method achieves the desired rate of convergence in presence of critical points. WENO-PW is based on the definition of a new global smoothness indicator that accounts for the variation of smoothness of the function among the k different stencils that compose the big stencil $\mathcal{T}(i)$. This information is used to decide when the reconstruction must retain the non-oscillatory property or provide the optimal reconstruction.

The global smoothness indicator presented here and denoted by ξ, is defined as follows

$$\xi = \chi^b\,, \qquad \chi = \left(\frac{|\beta_0 - \beta_{k-1}|}{\beta_0 + \beta_{k-1} + \epsilon}\right) \tag{B.85}$$

where ϵ is a small constant to avoid division by zero, selected in this work as 10^{-m}, with m the number of digits of precision of the machine. Parameter b is a positive constant that enhances the ratio inside the parenthesis. The global smoothness indicator is defined to ensure that the ratio inside the parenthesis is always less than unity and greater or equal zero. The α_r coefficients in (B.64) are reformulated using parameter ξ as a power exponent, and then normalized leading to the WENO-PW weights, ω_r^{PW}

$$\alpha_r^{PW} = \frac{\gamma_r}{(\beta_r + \epsilon)^{p\xi}} \qquad \omega_r^{PW} = \frac{\alpha_r^{PW}}{\sum_{l=0}^{k-1} \alpha_l^{PW}}, \qquad r = 0, ..., k-1 \tag{B.86}$$

where parameter p is a positive constant and ϵ is set as in (B.85). Therefore suitable values of b and p are required.

In order to observe the influence of b in the behavior of the global smoothness indicator ξ, Figure B.5 plots ξ for different values of b, expressed in terms of the ratio β_{k-1}/β_0. Depending on the relative values of the different β_r smoothness indicators, one can observe that

- when the function is smooth in $\mathcal{T}(i)$, then $\beta_0 \approx \beta_{k-1}$ making ξ tend to 0^+ and α_r^{PW} coefficients become closer to the optimal weights.
- when the function has a discontinuity in $\mathcal{T}(i)$, then $\beta_0 \ll \beta_{k-1}$ or $\beta_0 \gg \beta_{k-1}$ making ξ tend to 1^-, and the WENO-JS strategy is recovered avoiding oscillatory reconstructions.
- when the function is symmetric in $\mathcal{T}(i)$ with respect to x_i, then $\beta_0 \approx \beta_{k-1}$ making ξ tend to 0^+, recovering the optimal weights.

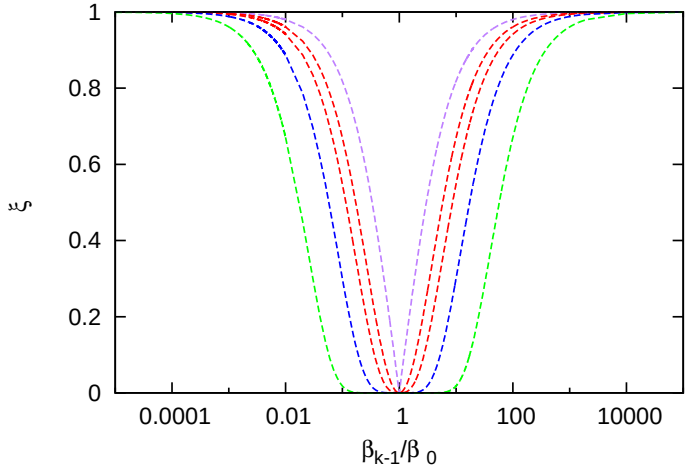

Figure B.5: Global smoothness indicator ξ versus β_{k-1}/β_0 for different values of b: $b = 1$ (violet), $b = 2$ (red), $b = 3$ (orange), $b = 6$ (blue), $b = 20$ (green).

Therefore, the introduction of ξ in the definition of the α_r^{PW} coefficients allows to control the influence of the smoothness indicator β_r in the WENO weights.

In the WENO-Z scheme, only one user control parameter is defined, the power parameter p_Z. When p_Z is increased over 1, the non-oscillatory property of the reconstruction and the recovery of the order accuracy in smooth regions are strengthened. On the other hand, when power parameter, p_Z is decreased below 1, the WENO property starts to vanish and the reconstruction in smooth areas does not experiment any improvement if compared to the original WENO-JS scheme.

Using the approach proposed in this work, the user can modify separately the capability to recover the optimal weights in smooth regions by changing b in (B.85), and the capability of the WENO reconstruction by changing p in (B.86). Different combinations of parameters p and b provide different effects in the reconstruction, and can be analyzed considering two possible alternatives, one with $p < 1$ and the other one with $p > 1$:

- When the power exponent p is lower than 1, the essentially non-oscillatory property cannot be fully retained due to the homogenization of weights and discontinuities are less accurately captured as p tends to 0. At this limit, the approximation procedure provides the optimal reconstruction since

Parameters	Recovers smooth regions	Captures discontinuities
$b \gg 1,\ p \gg 1$	✓	✓
$b \ll 1,\ p \gg 1$	✗	✓
$b \gg 1,\ p \ll 1$	✓	✗
$b \ll 1,\ p \ll 1$	✓	✗

Table B.2: Summary of the possible combinations of b and p in the WENO-PW method, and their effect in the reconstruction.

the optimal weights are fully recovered. The only effect that ξ (by modifying b) can have when $p < 1$ is to increment the homogenization of the weights since its upper bound is 1, strengthening the recovery of optimal weights.

- If moving to values of p greater than 1, the essentially non-oscillatory property is enhanced since differences among β_r weights are increased. Here, the role of ξ (through the modification of b) is decisive:
  - In regions where discontinuities are present, χ in (B.85) tends to 1. Although values of $b > 1$ decrease the rate of convergence of ξ to 1, imposing values of $p > 1$ reinforce the non oscillatory property when evaluating weights in (B.86), as the final power exponent q is increased.
  - In smooth regions, χ in (B.85) tends to 0. Values of $b > 1$ increase the rate of convergence of ξ to 0, enhancing the homogenization of the weights and leading to a faster recovery of the optimal weights when evaluating weights in (B.86). The effect of using values of $p > 1$ does not have a noticeable impact in the solution as ξ tends to 0 in this case.

The performance of the reconstruction for the different combinations of parameters b and p explained above is summarized in Table B.2.

When comparing with the WENO-JS method, it becomes clear that the introduction of ξ in the definition of the α_r^{PW} coefficients allows to control the influence of the smoothness indicator β_r in the calculation of the WENO weights. In smooth regions the β_r indicators loose their influence in the calculation of the weights. On the other hand, in presence of discontinuities, the WENO-JS scheme is recovered as the β_r indicators are used to compute the weights. Numerical tests indicate that values of power exponent $b \geq 10$ ensure the desired rate of convergence. The value of p is selected using $p = k - 1$, as in the WENO-Z method.

Appendix C

Sub-cell WENO reconstruction of derivatives for the ADER scheme

WENO sub-cell derivative reconstruction procedures in [21, 10] provide suitable approximations of the derivatives of the function in ADER schemes. Reconstruction procedure of spatial derivatives in [21] is more efficient and leads to a better solution of the ADER scheme. The reconstruction of the $2k-2$ derivatives for a $(2k-1)$-th order ADER scheme is performed by means of a $(2k-1)$-th order WENO reconstruction using k stencils in [21], while in [10] $(2k-1)$ stencils are needed.

This method is based on the construction of a polynomial $\phi_i(x)$ inside each cell I_i. As $2k-2$ points inside I_i are defined, $2k-2$ WENO reconstructions of $(2k-1)$-th order are required. Derivatives of polynomial $\phi_i(x)$ are an approximation of the exact derivatives of function $u(x)$. The procedure for the estimation of the $2k-2$ derivatives is summarized in the next subsection.

C.1 Procedure for the reconstruction of the derivatives

The procedure for the WENO sub-cell derivative reconstruction procedure in [21] is composed of the following four steps:

a) *Define sub-cell points*

Sub-cell points for the cell I_i are denoted by $x_i^{(b)}$ for $b = 1, ..., 2k-2$. In [21], uniformly distributed points inside the cell are proposed, but this leads to negative optimal weights, γ_r, in the WENO reconstruction. Thus the WENO procedure needs to be modified in order to give a good treatment to the negative weights.

So as not to embark on a more complex WENO reconstruction, all sub-cell points are chosen to be inside the positive interval of the optimal weights [32]. Note that the positive intervals of the optimal weights widely cover the cell boundaries but the center, as depicted in Figure C.1, therefore sub-cell points are taken close to cell interfaces. Moreover, it is worth mentioning that there points inside the cell where optimal weights do not exist. The asymptotic behavior of the weights around these points can be observed in Figure C.1 that plots the minimum optimal weight against x inside a normalized cell with $\Delta x = 1$. Notice that the number of singular points is equal to $k-1$ and also that the center of the cell is a singular point when k is even.

The following formula is proposed, with a geometrical refinement of 1/2 between consecutive points of the same cell side:

$$x_i^{(b)} = \begin{cases} x_{i-\frac{1}{2}} & \text{if } b = 1 \\ x_{i-\frac{1}{2}} + \frac{\Delta x}{2 \cdot 2^{k-b}} & \text{if } 2 \leq b \leq k-1 \\ x_{i+\frac{1}{2}} - \frac{\Delta x}{2 \cdot 2^{b-k+1}} & \text{if } k \leq b \leq 2k-3 \\ x_{i+\frac{1}{2}} & \text{if } b = 2k-2 \end{cases}$$

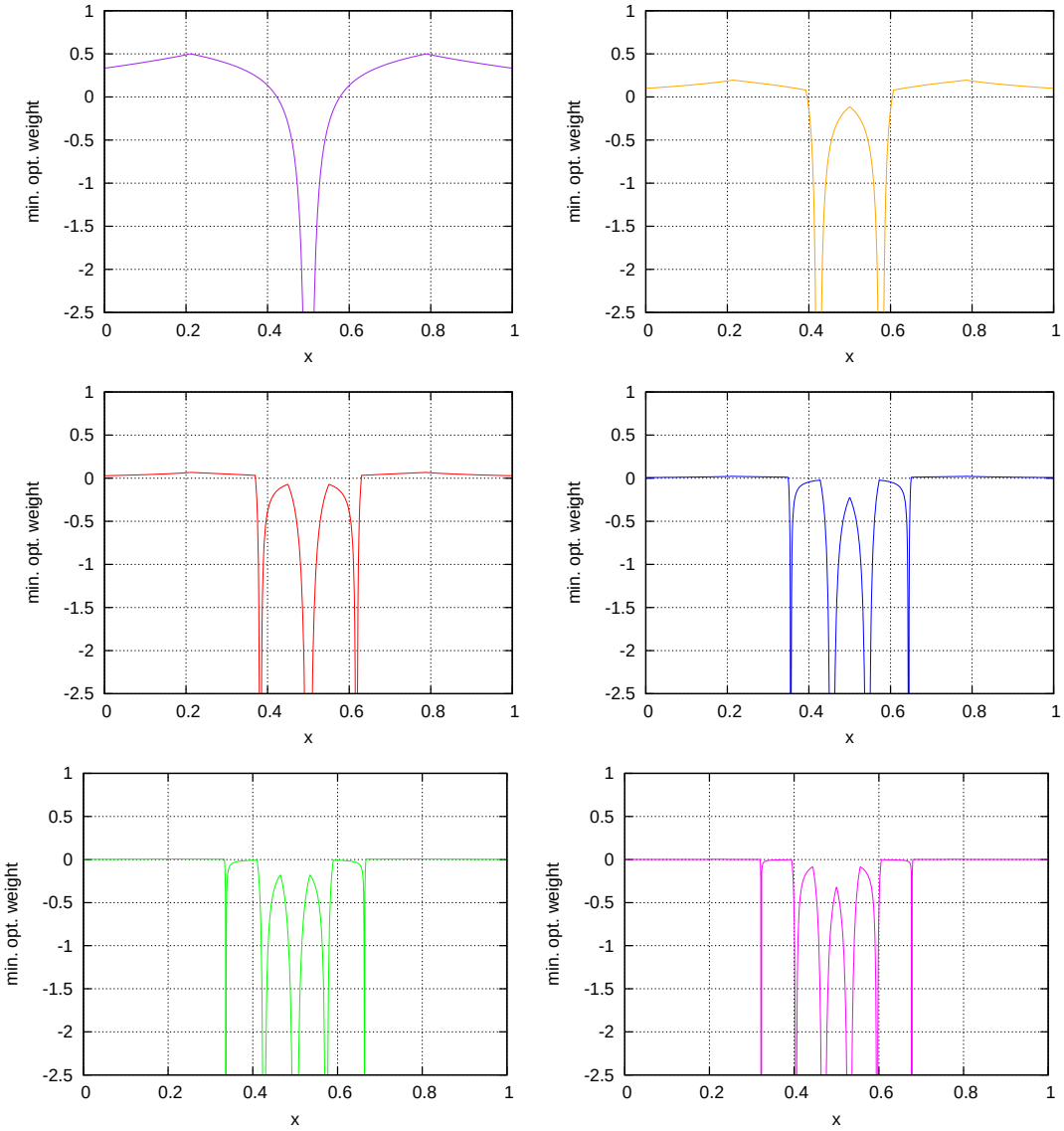

Figure C.1: Minimum optimal weight value inside a cell with cell size $\Delta x = 1$ for a 3-rd, 5-th, 7-th, 9-th, 11-th and 13-th polynomial reconstruction procedure.

b) *Reconstruct* $(2k-2)$ *point-wise values of* $u(x)$

A $(2k-1)$-th order WENO approximation in (B.68) has to be used to get $u_i^{(b)}$ with $b = 1, 2, ..., 2k-2$, the reconstructed point-wise values of u at the sub-cell points $x_i^{(b)}$ inside I_i. Different WENO weights, denoted by $\omega_r^{(b)} = \Omega_r(x_i^{(b)})$, will appear at each sub-cell point. Expression in (B.64) is used to compute $\alpha_r^{(b)}$ weights

$$\alpha_r^{(b)} = \frac{\gamma_r^{(b)}}{(\beta_r + \epsilon_i)^2}, \qquad r = 0, ..., k-1, \quad b = 1, ..., 2k-2 \tag{C.1}$$

where $\gamma_r^{(b)} = \Gamma_r(x_i^{(b)})$. Then, expression in (B.66) is used to compute the non-oscillatory weights as

$$\omega_r^{(b)} = \frac{\alpha_r^{(b)}}{\sum_{l=0}^{k-1} \alpha_l^{(b)}}, \qquad r = 0, ..., k-1, \quad b = 1, ..., 2k-2 \tag{C.2}$$

The $2k-1$-th WENO reconstruction at $x_i^{(b)}$ is given by (B.60), yielding

$$u_i^{(b)} = \sum_{r=0}^{k-1} \omega_r^{(b)} u_i^{(b)^{(r)}} \tag{C.3}$$

c) *Construct* $\phi_i(x)$

Following [21], the expression for the polynomial that approximates $u(x)$ in I_i is

$$\phi_i(x) = \sum_{l=0}^{2k-2} a_l \left(\frac{x - x_{i-\frac{1}{2}}}{\Delta x} \right)^l \tag{C.4}$$

where the coefficients a_l $(l = 0, ..., 2k-2)$ have to be determined with the following $2k-1$ equations for each cell I_i:

- From the $2k-2$ reconstructed values of $u(x)$ at the $2k-2$ sub-cell points, the following equations are formulated

$$\phi_i \left(x_i^{(b)} \right) = u_i^{(b)}, \qquad b = 1, ..., 2k-2 \tag{C.5}$$

- From the cell average, $\bar{u}_i$, the following equation is formulated

$$\frac{1}{\Delta x} \int_{I_r} \phi_i(x)\, dx = \sum_{l=0}^{2k-2} \frac{a_l}{l+1} \left[\left(\frac{x - x_{i-\frac{1}{2}}}{\Delta x} \right)^{l+1} \right]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} = \bar{u}_i \tag{C.6}$$

These $2k-2$ equations in (C.5) and the equation in (C.6) are used to formulate a linear system with the coefficients a_l $(l = 0, ..., 2k-2)$ in the vector of unknowns.

d) *Evaluate derivatives at the cell boundaries*

We get the approximation to the m-th derivative of $u(x)$ $(m = 1, ..., 2k-2)$ at any desired point taking the m-th derivative of $\phi_i(x)$

$$\frac{d^m \phi_i(x)}{dx^m} = \frac{d^m}{dx^m} \left[\sum_{l=0}^{2k-2} a_l \left(\frac{x - x_{i-\frac{1}{2}}}{\Delta x} \right)^l \right] = \sum_{l=m}^{2k-2} \frac{l!}{(l-m)!} \frac{a_l}{\Delta x^l} \left(x - x_{i-\frac{1}{2}} \right)^{(l-m)} \tag{C.7}$$

From (C.7), it becomes clear that the quality of the WENO the reconstruction at the $2k-2$ points will determine the accuracy in the computation of the $2k-2$ derivatives or the equivalent a_l coefficients in (C.7), and the actual convergence to an ADER scheme of $(2k-1)$-th order of accuracy.

Appendix D

2D extension of the WENO reconstruction method

D.1 Interpolation and reconstruction in 2D

In this section, the problem of data reconstruction in 2D at an arbitrary point inside a cell by means of polynomial interpolation when departing from cell averages is considered.

The function $u(x,y)$ will be defined departing from the starting data, that will be considered as the average value of this function in each cell. The definition of $u(x,y)$ is useful for the derivation of the reconstruction procedure but its analytical expression will be unknown in most cases. The computational grid, shown in Figure D.1, is composed by $N_x \times N_y$ cells as

$$\Omega = [a,b] \times [c,d] \tag{D.1}$$

with

$$\begin{array}{c} a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < ... < x_{N_x-\frac{1}{2}} < x_{N_x+\frac{1}{2}} = b \\ c = y_{\frac{1}{2}} < y_{\frac{3}{2}} < ... < y_{N_y-\frac{1}{2}} < y_{N_y+\frac{1}{2}} = d \end{array} \tag{D.2}$$

with cells and cell sizes defined by

$$I_{i,j} = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] \times \left[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}\right] \tag{D.3}$$

$$\begin{array}{c} \Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} \equiv \text{constant} \\ \Delta y_j = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}} \equiv \text{constant} \end{array} \tag{D.4}$$

With the previous definitions, the starting data set is now defined as the the average value of the function $u(x,y)$ in each cell

$$\bar{u}_i = \frac{1}{\Delta x_i \Delta y_j} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u(\xi,\eta)\, d\eta d\xi, \quad i = 1,2,...,N_x \quad j = 1,2,...,N_y \tag{D.5}$$

The problem we face is to find a polynomial function $p_{r_1,r_2}(x,y)$ of **degree at most** $k-1$ for each cell $I_{i,j}$, such that it is a k**-th order accurate** approximation of the function $u(x,y)$ inside $I_{i,j}$

$$p_r(x,y) = u(x,y) + O\left(\Delta x^k\right), \quad x \in I_{i,j}, \quad i = 1,2,...,N_x \quad j = 1,2,...,N_y \tag{D.6}$$

In the two-dimensional case, the concept of *stencil* is generalized to a group of surface connected cells. For each cell, $I_{i,j}$, it is possible to define a stencil $S_{r_1,r_2}(i,j)$ composed by cell $I_{i,j}$ plus r_1 cells to the left, s_1 cells to the right, r_2 cells to the top and s_2 cells to the bottom. If considering $S_{r_1,r_2}(i,j)$ with

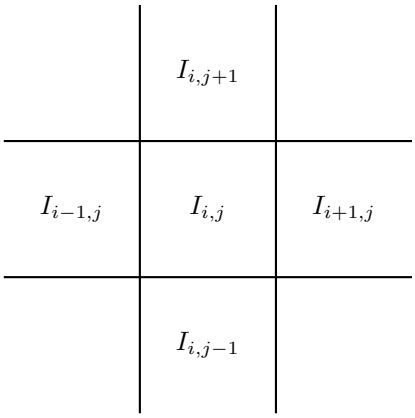

Figure D.1: Mesh discretization

the same number of cells k, in both directions, we can affirm $k = r_1 + s_1 + 1 = r_2 + s_2 + 1$. For all cases, the condition $r, s \geq 0$ must be satisfied. The stencil can be expressed as

$$S_{r_1,r_2}(i,j) = \bigcup_{l,m \in [0,\ldots,k-1]} I_{i-r_1+l,j-r_2+m} \tag{D.7}$$

The steps required to generate the reconstructing polynomial departing from cell averages are listed below:

a) Stencil selection.

Given the cell $I_{i,j}$ and the order of accuracy required k, we must first choose a stencil $S_{r_1,r_2}(i,j)$ with $k = r_1 + s_1 + 1 = r_2 + s_2 + 1$ cells.

There is a unique polynomial $p_{r_1,r_2}(x,y)$ of degree at most $k-1$ whose cell average value for each cell in the stencil agrees with that of the function $u(x,y)$ [20]

$$\frac{1}{\Delta x_m \Delta y_l} \int_{x_{m-\frac{1}{2}}}^{x_{m+\frac{1}{2}}} \int_{y_{l-\frac{1}{2}}}^{y_{l+\frac{1}{2}}} p_{r_1,r_2}(\xi,\eta)\, d\eta d\xi = \bar{u}_{m,l} \tag{D.8}$$

with $m = i - r_1, ..., i + s_1$ and $l = j - r_2, ..., i + s_2$.

b) Definition of the primitive function.

In order to find the interpolating polynomial $p_{r_1,r_2}(x,y)$ of degree $k-1$ and k-th order of accuracy, a new function is introduced. This new function is the primitive function of $u(x,y)$, denoted by $U(x,y)$, which is defined as the cumulative integral of $u(x,y)$ from $-\infty$ to x and y

$$U(x,y) = \int_{-\infty}^{x} \int_{-\infty}^{y} u(\xi,\eta)\, d\eta d\xi \tag{D.9}$$

For a random location in the grid, i, j, the value of this cumulative integral at the right boundary of the cell $I_{i,j}$ can be computed by the summation of the average values of each cell multiplied by the cell size, from $-\infty$ to the cell $I_{i,j}$, as follows:

$$U\left(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}\right) = \int_{-\infty}^{x_{i+\frac{1}{2}}} \int_{-\infty}^{y_{j+\frac{1}{2}}} u(\xi,\eta)\, d\eta d\xi = \sum_{m=-\infty}^{i} \sum_{l=-\infty}^{j} \bar{u}_{m,l} \Delta x_m \Delta y_l \tag{D.10}$$

Also, a polynomial $P_{r_1,r_2}(x,y)$ is defined as the **unique polynomial function of degree at most k which interpolates $U(x,y)$ with $k+1$-th order of accuracy in $k+1$ nodes** (which are all the cell boundaries in the stencil) and we denote its derivative by $p_{r_1,r_2}(x,y)$:

$$p_{r_1,r_2}(x,y) = \frac{\partial^2}{\partial x \partial y} P_{r_1,r_2}(x,y) \tag{D.11}$$

Note that $p_{r_1,r_2}(x,y)$ is a polynomial of degree $k-1$ and k-th order, defined by k^2 cells. Polynomial $P_r(x)$ is one order greater, and as the number of cells does not change, $k+1$ interpolation points are necessary. It is worth noticing that this new $k+1$ points are defined at the nodes, even though the value of $u(x,y)$ is not initially defined at these locations.

Using (D.11) it is possible to prove the equality in (D.8)

$$\begin{aligned}
&\frac{1}{\Delta x_m \Delta y_l} \int_{x_{m-\frac{1}{2}}}^{x_{m+\frac{1}{2}}} \int_{y_{l-\frac{1}{2}}}^{y_{l+\frac{1}{2}}} p_{r_1,r_2}(\xi,\eta)\, d\eta d\xi = \frac{1}{\Delta x_m \Delta y_l} \int_{x_{m-\frac{1}{2}}}^{x_{m+\frac{1}{2}}} \int_{y_{l-\frac{1}{2}}}^{y_{l+\frac{1}{2}}} \frac{\partial^2}{\partial \xi \partial \eta} P_{r_1,r_2}(\xi,\eta)\, d\eta d\xi = \\
&\frac{1}{\Delta x_m \Delta y_l} \int_{x_{m-\frac{1}{2}}}^{x_{m+\frac{1}{2}}} \partial \int_{y_{l-\frac{1}{2}}}^{y_{l+\frac{1}{2}}} \partial P_{r_1,r_2}(\xi,\eta) = \frac{1}{\Delta x_m \Delta y_l} \int_{x_{m-\frac{1}{2}}}^{x_{m+\frac{1}{2}}} \partial \left[P_{r_1,r_2}\left(\xi, y_{l+\frac{1}{2}}\right) - P_{r_1,r_2}\left(\xi, y_{l-\frac{1}{2}}\right) \right] = \\
&\frac{1}{\Delta x_m \Delta y_l} \left[P_{r_1,r_2}\left(x_{m+\frac{1}{2}}, y_{l+\frac{1}{2}}\right) - P_{r_1,r_2}\left(x_{m+\frac{1}{2}}, y_{l-\frac{1}{2}}\right) - P_{r_1,r_2}\left(x_{m-\frac{1}{2}}, y_{l+\frac{1}{2}}\right) + P_{r_1,r_2}\left(x_{m-\frac{1}{2}}, y_{l-\frac{1}{2}}\right) \right] \\
&\approx \frac{1}{\Delta x_m \Delta y_l} \left[U\left(x_{m+\frac{1}{2}}, y_{l+\frac{1}{2}}\right) - U\left(x_{m+\frac{1}{2}}, y_{l-\frac{1}{2}}\right) - U\left(x_{m-\frac{1}{2}}, y_{l+\frac{1}{2}}\right) + U\left(x_{m-\frac{1}{2}}, y_{l-\frac{1}{2}}\right) \right] = \\
&= \frac{1}{\Delta x_m \Delta y_l} \int_{x_{m-\frac{1}{2}}}^{x_{m+\frac{1}{2}}} \int_{y_{l-\frac{1}{2}}}^{y_{l+\frac{1}{2}}} \frac{\partial^2}{\partial \xi \partial \eta} U(\xi,\eta)\, d\eta d\xi = \frac{1}{\Delta x_m \Delta y_l} \int_{x_{m-\frac{1}{2}}}^{x_{m+\frac{1}{2}}} \int_{y_{l-\frac{1}{2}}}^{y_{l+\frac{1}{2}}} u(\xi,\eta)\, d\eta d\xi = \bar{u}_{m,l}
\end{aligned} \tag{D.12}$$

for any $m = i - r_1, ..., i + s_1$ and $l = j - r_2, ..., i + s_2$. The approximation symbol stands for the approximation of $U(x,y)$ by the interpolating polynomial $P_{r_1,r_2}(x,y)$. This interpolation is a $k+1$-th order approximation

$$P_{r_1,r_2}(x,y) = U(x,y) + O\left(\Delta x^{k+1}\right), \qquad x, y \in I_{i,j} \tag{D.13}$$

and that of its derivative, a k-th order approximation

$$\frac{\partial^2}{\partial x \partial y} P_{r_1,r_2}(x,y) = \frac{\partial^2}{\partial x \partial y} U(x,y) + O\left(\Delta x^{k}\right), \qquad x, y \in I_{i,j}\,. \tag{D.14}$$

c) Lagrange interpolation

In [20], the use of the Lagrange form of the interpolating polynomial is proposed. This kind of interpolation is said to be nodal since each weight takes the value of 1 at the corresponding node and 0 at the rest of the nodes. The expression for the 2D Lagrange interpolating polynomial for structured meshes of $n_x \cdot n_y$ points is given by

$$L(x) = \sum_{i=0}^{n_x} \sum_{j=0}^{n_y} y(x_i, y_j) L_{i,j}(x,y) \tag{D.15}$$

where

$$L_{i,j}(x,y) = l_i(x) \cdot l_j(y) \tag{D.16}$$

considering the 2D mesh as the intersection of two 1D meshes defined by the points $\{x_1, x_2, ..., x_{n_x}\}$ and $\{y_1, y_2, ..., y_{n_y}\}$ respectively.

The weighting functions are defined as in the 1D case as

$$
\begin{aligned}
l_i(x) &= \prod_{\substack{d=0 \\ d\neq i}}^{n_x} \frac{(x - x_d)}{(x_i - x_d)} \\
l_j(y) &= \prod_{\substack{d=0 \\ d\neq j}}^{n_y} \frac{(y - y_d)}{(y_j - y_d)}
\end{aligned}
\tag{D.17}
$$

Making use of the Lagrange formula in (D.15), it is possible to write the expression for the interpolating polynomial $P_{r_1,r_2}(x,y)$ at all nodes of the stencil $S(i,j)$ where the values of function $U(x,y)$ are known, yielding

$$
P_{r_1,r_2}(x,y) = \sum_{m=0}^{k}\sum_{l=0}^{k} \tilde{U}_{m,l} \prod_{\substack{d=0 \\ d\neq m}}^{k} \frac{(x - x_{i-r_1+d-\frac{1}{2}})}{(x_{i-r_1+m-\frac{1}{2}} - x_{i-r_1+d-\frac{1}{2}})} \prod_{\substack{d=0 \\ d\neq l}}^{k} \frac{(y - y_{j-r_2+d-\frac{1}{2}})}{(y_{j-r_2+l-\frac{1}{2}} - y_{j-r_2+d-\frac{1}{2}})} \tag{D.18}
$$

where $\tilde{U}_{m,l}$ is a redefined primitive function with origin at $(x_{i-\frac{1}{2}-r_1}, y_{j-\frac{1}{2}-r_2})$ and given by

$$
\tilde{U}_{m,l} = \int_{x_{i-\frac{1}{2}-r_1}}^{x_{i-\frac{1}{2}-r_1+m}} \int_{y_{j-\frac{1}{2}-r_2}}^{y_{j-\frac{1}{2}-r_2+l}} u\,(\xi,\eta)\, d\eta d\xi \tag{D.19}
$$

The use of $\tilde{U}_{m,l}$ allows to express (D.18) in terms of exclusively cell averages inside the stencil, since

$$
\tilde{U}_{m,l} = \sum_{e=0}^{m-1}\sum_{d=0}^{l-1} \bar{u}_{i-r_1+e,j-r_2+d} \Delta x_{i-r_1+e} \Delta y_{j-r_2+d} \tag{D.20}
$$

Taking the cross derivative of (D.18) with respect to x and y and inserting the previous result, a expression for polynomial $p_{r_1,r_2}(x,y)$ is obtained

$$
p_{r_1,r_2}(x,y) = \sum_{m=0}^{k}\sum_{l=0}^{k} \left(\mathcal{L}_m\, \mathcal{L}_l \sum_{e=0}^{m-1}\sum_{d=0}^{l-1} \bar{u}_{i-r_1+e,j-r_2+d} \Delta x_{i-r_1+e} \Delta y_{j-r_2+d} \right) \tag{D.21}
$$

with

$$
\mathcal{L}_m = \left(\frac{\sum_{\substack{t=0 \\ t\neq m}}^{k} \prod_{\substack{n=0 \\ n\neq m,t}}^{k} \left(x - x_{i-r_1+n-\frac{1}{2}} \right)}{\prod_{\substack{n=0 \\ n\neq m}}^{k} \left(x_{i-r_1+m-\frac{1}{2}} - x_{i-r_1+n-\frac{1}{2}} \right)} \right) \tag{D.22}
$$

$$
\mathcal{L}_l = \left(\frac{\sum_{\substack{t=0 \\ t\neq l}}^{k} \prod_{\substack{n=0 \\ n\neq l,t}}^{k} \left(y - y_{j-r_2+n-\frac{1}{2}} \right)}{\prod_{\substack{n=0 \\ n\neq l}}^{k} \left(y_{j-r_2+l-\frac{1}{2}} - y_{j-r_2+n-\frac{1}{2}} \right)} \right) \tag{D.23}
$$

A simpler expression for $p_{r_1,r_2}(x,y)$ can be derived from equation (D.21) taking the cell averages as common factors. The resulting expression represents the reconstructing polynomial function as a linear combination of the cell averages inside the stencil as

$$
p_{r_1,r_2}(x,y) = \sum_{e=0}^{k-1}\sum_{d=0}^{k-1} \left(\sum_{m=e+1}^{k}\sum_{l=d+1}^{k} \mathcal{L}_m\, \mathcal{L}_l \right) \bar{u}_{i-r_1+e,j-r_2+d} \Delta x_{i-r_1+e} \Delta y_{j-r_2+d} \tag{D.24}
$$

that can be rewritten as

$$
p_{r_1,r_2}(x,y) = \sum_{e=0}^{k-1}\sum_{d=0}^{k-1} \left(\sum_{m=e+1}^{k} \mathcal{L}_m \sum_{l=d+1}^{k} \mathcal{L}_l \right) \bar{u}_{i-r_1+e,j-r_2+d} \Delta x_{i-r_1+e} \Delta y_{j-r_2+d} \tag{D.25}
$$

If defining

$$C^{(k)}_{r_1,e}(x) = \left(\sum_{m=e+1}^{k} \mathcal{L}_m \right) \Delta x_{i-r_1+e} \tag{D.26}$$

$$C^{(k)}_{r_2,d}(y) = \left(\sum_{l=d+1}^{k} \mathcal{L}_l \right) \Delta y_{j-r_2+d} \tag{D.27}$$

it is possible to express Equation (D.25) as

$$p_{r_1,r_2}(x,y) = \sum_{e=0}^{k-1} \sum_{d=0}^{k-1} C^{(k)}_{r_1,e}(x)\, C^{(k)}_{r_2,d}(y)\ \bar{u}_{i-r_1+e,j-r_2+d} \tag{D.28}$$

Where $C^{(k)}_{r_1,e}(x)$ and $C^{(k)}_{r_2,d}(y)$ are constants at a given x and provide the weights for the linear combination of cell averages. The superscript k of these coefficients stands for the dimension of the stencil in each direction. Remark that coefficients $C^{(k)}_{r_1,e}(x)$ and $C^{(k)}_{r_2,d}(y)$ are equivalent to those obtained for the 1D case in (B.24).

d) Calculation of $C^{(k)}_{r_1,e}(x)$ and $C^{(k)}_{r_2,d}(y)$ coefficients at the sought point and computation of the reconstruction using (D.28).

D.2 Dimension-by-dimension 2D reconstruction

In the previous part, the procedure for the generation of a 2D reconstruction departing from cell averages was shown. The expression for the reconstructing polynomial function was obtained in (D.28). Considering this result, it is straightforward to compute the reconstruction at a certain point by calculating first the coefficients $C^{(k)}_{r_1,e}(x)$ and $C^{(k)}_{r_2,d}(y)$ at the desired point and substituting then in (D.28), getting the sought value.

Another possibility would be to obtain the 2D reconstruction by carrying out two 1D reconstructions recursively, for each of the variables, x and y, each time. For instance, let us consider the first 1D reconstruction for variable y. First, for a fixed x value, the function $u(x,y)$ can be reconstructed along the y coordinate using Lagrange interpolation as in the 1D case. Then, the resulting set of reconstructed values at each x position, which depends upon y, can be used to generate another Lagrange interpolation polynomial that depends upon x and y and corresponds to the sought reconstructing function.

For instance, let us consider the first polynomial reconstruction for variable y. The interpolation polynomial will be denoted by $P^m_{r_1,r_2}(y)$ and constructed using the Lagrange basis, leading to the following expression

$$P^m_{r_1,r_2}(y) = \sum_{l=0}^{k-1} \tilde{U}_{m,l} \prod_{\substack{d=0 \\ d\neq l}}^{k} \frac{(y - y_{j-r_2+d-\frac{1}{2}})}{(y_{j-r_2+l-\frac{1}{2}} - y_{j-r_2+d-\frac{1}{2}})} \tag{D.29}$$

where m stands for the x position. Substitution of $\tilde{U}_{m,l}$ by the expression provided in (D.20) leads to

$$P^m_{r_1,r_2}(y) = \sum_{l=0}^{k-1} \left(\sum_{e=0}^{m-1} \sum_{d=0}^{l-1} \bar{u}_{i-r_1+e,j-r_2+d} \Delta x_{i-r_1+e} \Delta y_{j-r_2+d} \right) \prod_{\substack{d=0 \\ d\neq l}}^{k} \frac{(y - y_{j-r_2+d-\frac{1}{2}})}{(y_{j-r_2+l-\frac{1}{2}} - y_{j-r_2+d-\frac{1}{2}})} \tag{D.30}$$

Taking the derivative of (D.30) with respect to y, it yields

$$\frac{\partial P^m_{r_1,r_2}(y)}{\partial y} = \sum_{l=0}^{k-1} \left[\mathcal{L}_l \sum_{d=0}^{l-1} \left(\sum_{e=0}^{m-1} \bar{u}_{i-r_1+e,j-r_2+d} \Delta x_{i-r_1+e} \right) \Delta y_{j-r_2+d} \right] \tag{D.31}$$

with $\mathcal{L}_l$ defined in (D.23). This expression can be rewritten as

$$\frac{\partial P^m_{r_1,r_2}(y)}{\partial y} = \sum_{d=0}^{k-1} \sum_{l=d+1}^{k} \mathcal{L}_l \Delta y_{j-r_2+d} \sum_{e=0}^{m-1} \bar{u}_{i-r_1+e,j-r_2+d} \Delta x_{i-r_1+e} \tag{D.32}$$

and making use of the coefficient $C^{(k)}_{r_2,d}(y)$ defined in (D.27), Equation (D.32) can be expressed as

$$\frac{\partial P^m_{r_1,r_2}(y)}{\partial y} = \sum_{d=0}^{k-1} C^{(k)}_{r_2,d}(y) \sum_{e=0}^{m-1} \bar{u}_{i-r_1+e,j-r_2+d} \Delta x_{i-r_1+e} \,. \tag{D.33}$$

Factorization of the previous expression for each value of e yields

$$\frac{\partial P^m_{r_1,r_2}(y)}{\partial y} = \sum_{e=0}^{m-1} \left(\sum_{d=0}^{k-1} C^{(k)}_{r_2,d}(y) \bar{u}_{i-r_1+e,j-r_2+d} \right) \Delta x_{i-r_1+e} \,. \tag{D.34}$$

noticing that the term inside brackets corresponds to a 1D polynomial reconstruction for a given value of e. The general expression for a 1D reconstruction is provided in (B.27) and referred to as $p(r, k, \nu, \bar{\mathbf{v}})$, where in this case $r = r_2$, $\nu = y$ and the vector of cell averages will include the superscript e, r_1 and r_2 to denote dependency upon the x position and the stencils respectively, becoming $\bar{\mathbf{v}}^e_{r_1,r_2}$ and with components $\left\{\bar{v}^e_{r_1,r_2}\right\}_d = \bar{u}_{i-r_1+e,j-r_2+d}$, for $d = 0, ..., k-1$. In this case, Equation (B.27) becomes

$$p(r_2, k, y, \bar{\mathbf{v}}^e_{r_1,r_2}) = \sum_{d=0}^{k-1} C^{(k)}_{r_2,d}(y) \bar{u}_{i-r_1+e,j-r_2+d} \,. \tag{D.35}$$

Making use of (D.35), the derivative $\partial P^m_{r_1,r_2}(y)/\partial y$ in Equation (D.34) is finally expressed as

$$\frac{\partial P^m_{r_1,r_2}(y)}{\partial y} = \sum_{e=0}^{m-1} p(r_2, k, y, \bar{\mathbf{v}}^e_{r_1,r_2}) \Delta x_{i-r_1+e} \,. \tag{D.36}$$

On the other hand, an analogous polynomial interpolation can be carried out in the x direction by means of the Lagrange formula, given by the following expression

$$P_{r_1,r_2}(x, y) = \sum_{m=0}^{k-1} P^m_{r_1,r_2}(y) \prod_{\substack{d=0 \\ d \neq m}}^{k} \frac{(x - x_{i-r_1+d-\frac{1}{2}})}{(x_{i-r_1+m-\frac{1}{2}} - x_{i-r_1+d-\frac{1}{2}})} \tag{D.37}$$

where $P^m_{r_1,r_2}(y)$ are the values of the 1D interpolating polynomial in (D.30) along y at the k different x positions according to the selected stencil.

In order to obtain the sought 2D reconstructing function, $p_{r_1,r_2}(x, y)$, according to definition in (D.11), we take the second order cross derivative of (D.37) and obtain

$$p_{r_1,r_2}(x, y) = \sum_{m=0}^{k-1} \frac{\partial}{\partial y} \left(P^m_{r_1,r_2}(y) \right) \frac{\partial}{\partial x} \left(\prod_{\substack{d=0 \\ d \neq m}}^{k} \frac{(x - x_{i-r_1+d-\frac{1}{2}})}{(x_{i-r_1+m-\frac{1}{2}} - x_{i-r_1+d-\frac{1}{2}})} \right) \tag{D.38}$$

Inserting (D.36) in (D.38), the latter yields

$$p_{r_1,r_2}(x, y) = \sum_{m=0}^{k-1} \left(\sum_{e=0}^{m-1} p(r_2, k, y, \bar{\mathbf{v}}^e_{r_1,r_2}) \Delta x_{i-r_1+e} \right) \frac{\partial}{\partial x} \left(\prod_{\substack{d=0 \\ d \neq m}}^{k} \frac{(x - x_{i-r_1+d-\frac{1}{2}})}{(x_{i-r_1+m-\frac{1}{2}} - x_{i-r_1+d-\frac{1}{2}})} \right) \tag{D.39}$$

and noticing that the derivative of the product with respect to x is equal to $\mathcal{L}_m$ in (D.22), Equation (D.39) can be expressed more compactly as

$$p_{r_1,r_2}(x,y) = \sum_{m=0}^{k-1} \left(\sum_{e=0}^{m-1} p(r_2,k,y,\bar{\mathbf{v}}^e_{r_1,r_2}) \Delta x_{i-r_1+e} \right) \mathcal{L}_m \tag{D.40}$$

that can be rewritten taking cell averages as common factors, leading to

$$p_{r_1,r_2}(x,y) = \sum_{e=0}^{k-1} \sum_{m=e+1}^{k} \mathcal{L}_m \Delta x_{i-r_1+e}\, p(r_2,k,y,\bar{\mathbf{v}}^e_{r_1,r_2}) \tag{D.41}$$

Making use of the definition of $C^{(k)}_{r_1,e}(x)$ in (D.26), Equation (D.41) can be expressed in a more compact form as

$$p_{r_1,r_2}(x,y) = \sum_{e=0}^{k-1} C^{(k)}_{r_1,e}(x)\, p(r_2,k,y,\bar{\mathbf{v}}^e_{r_1,r_2}) \tag{D.42}$$

that corresponds to a 1D reconstruction along x with departing data provided by the 1D reconstruction $p(r_2,k,y,\bar{\mathbf{v}}^e_{r_1,r_2})$. Equation (D.42) can be expressed in its compact form using (B.27), as

$$p_{r_1,r_2}(x,y) = p\left(r_1,k,x,\left\{p(r_2,k,y,\bar{\mathbf{v}}^e_{r_1,r_2})\right\}_{e=0,\dots,k-1}\right) \tag{D.43}$$

Notice that substitution of (D.35) in (D.42) leads to the general expression for the reconstruction in 2D presented in (D.28), that can be rewritten in recursive (dimension-by-dimension) form as

$$p_{r_1,r_2}(x,y) = \sum_{e=0}^{k-1} C^{(k)}_{r_1,e}(x) \left(\sum_{d=0}^{k-1} C^{(k)}_{r_2,d}(y)\ \bar{u}_{i-r_1+e,j-r_2+d} \right) \tag{D.44}$$

A simpler and analogous derivation of the dimension-by-dimension approach for 2D polynomial reconstruction can be carried out by introducing the variable $\bar{\zeta}_i(y)$ as

$$\bar{\zeta}_i(y) = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x,y)\, dx \tag{D.45}$$

that stands for the x-line averages along y, inside cells $I_{i,j}$ at column i. The use of this variable makes possible to rewrite cell averages as

$$\bar{u}_{i,j} = \frac{1}{\Delta x \Delta y} \int\int_{x,y\in I_{i,j}} u(x,y)\, dx\, dy = \frac{1}{\Delta y} \int_{y\in I_{i,j}} \bar{\zeta}_i(y)\, dy \tag{D.46}$$

and to notice that the application of the dimension-by-dimension reconstruction is straightforward. First, line averages $\bar{\zeta}_i(y)$ are reconstructed for a certain y value by means of Equation (D.35) and departing from cell averages $\bar{u}_{i,j}$. This reconstruction is carried out for all columns composing the stencil, given by parameter e, and provides the new 1D averages used as departing data in the second reconstruction. Finally, this second reconstruction is carried out using polynomial in (D.42) for a certain x value, leading to Equation (D.44).

D.3 Dimension-by-dimension 2D WENO reconstruction

As outlined in the previous section, it is possible to construct a conventional 2D reconstruction by means of two nested 1D reconstructions in each of the coordinate directions. In the same way, it will be possible to generate a 2D WENO reconstruction by carrying out two successive 1D WENO reconstructions.

For the generation of a $2k-1$-th order 2D WENO reconstruction inside the cell $I_{i,j}$, k^2 different stencils will be needed. The candidate stencils are given by

$$S_{r_1,r_2}(i,j) \qquad \forall r_1,r_2 = 0,\dots,k-1 \tag{D.47}$$

with $S_{r_1,r_2}(i,j)$ defined in (D.7). Moreover, a bigger stencil is defined as the union of the smaller stencils $S_{r_1,r_2}(i,j)$

$$\mathcal{T}(i,j) = \bigcup_{r_1,r_2\in[0,\ldots,k-1]} S_{r_1,r_2}(i,j) \tag{D.48}$$

noticing the following property

$$\bigcap_{r_1,r_2\in[0,\ldots,k-1]} S_{r_1,r_2}(i,j) = I_{i,j} \tag{D.49}$$

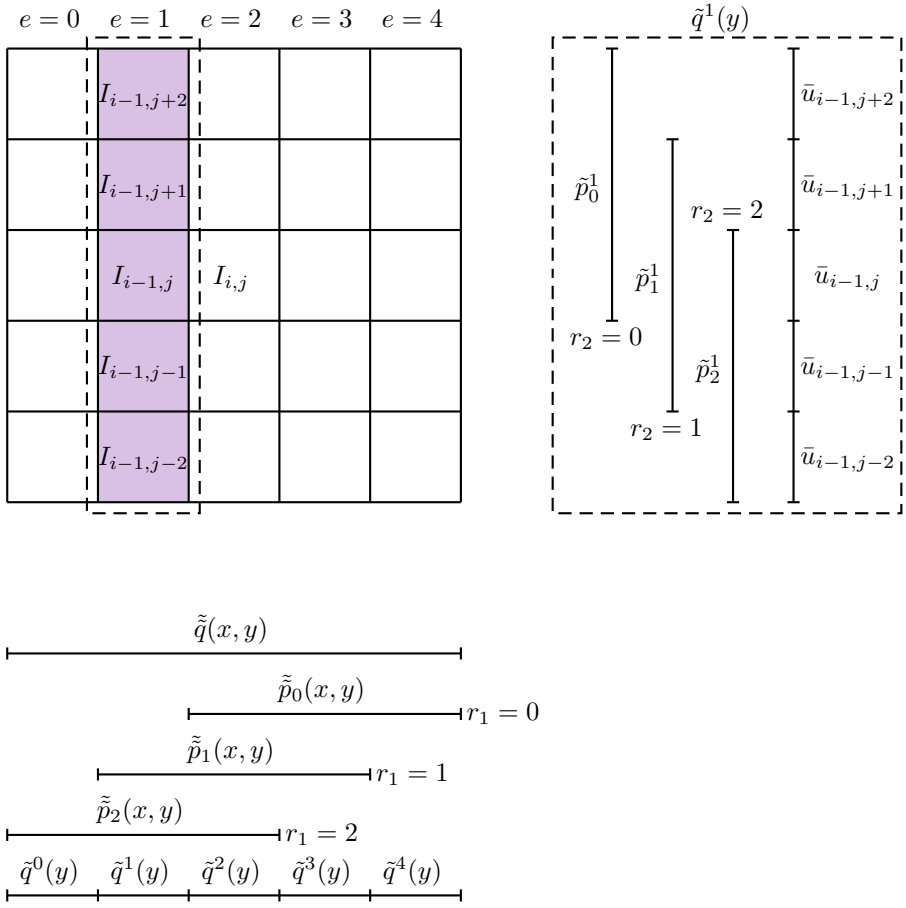

Figure D.2: 5-th order ($k=3$) 2D WENO reconstruction for cell $I_{i,j}$ inside stencil $\mathcal{T}(i,j)$ using two 1D sweeps. The first 1D sweep, along y direction, is depicted for $e=1$.

To compute a 2D WENO reconstruction inside cell $I_{i,j}$, the first step is to obtain $2k-1$ 1D WENO reconstructions along y, referred to as $\tilde{q}^e_{r_1}(y)$, for each x column of $\mathcal{T}(i,j)$ departing from cell averages grouped in k 1D stencils according to parameter r_2. These reconstructions will provide new one-dimensional average-like values, grouped in k 1D stencils according to parameter r_1, to generate another 1D WENO reconstruction along x, referred to as $\tilde{\tilde{q}}(x,y)$. The procedure to compute a 5-th order 2D WENO reconstruction inside cell $I_{i,j}$ is entirely depicted in Figure D.2.

As pointed out in the previous paragraph, the 1D reconstructions along y are carried out for each of the $2k-1$ columns of the big stencil $\mathcal{T}(i,j)$. For each column, k different stencils are taken in the y direction according to parameter r_2. Moreover, for each stencil characterized by r_2, k-th order 1D reconstructing polynomials $\tilde{p}^e_{r_2}(y)$ are calculated using Equation (D.35) as

$$\tilde{p}^e_{r_2}(y) = p\left(r_2, k, y, \{\bar{v}^e_{r_1,r_2}\}_{d=0,\dots,k-1}\right) \tag{D.50}$$

with $r_1 = k-1$, $e = 0, ..., 2k-2$ and $r_2 = 0, ..., k-1$.

The $2k-1$-th order WENO reconstruction $\tilde{q}^e_{r_1}(y)$ based on polynomials $\tilde{p}^e_{r_2}(y)$ is expressed according to (B.70) as

$$\tilde{q}^e(y) = q\left(k, y, \{\tilde{p}^e_{r_2}(y)\}_{r_2=0,\dots,k-1}\right) \tag{D.51}$$

for each $e = 0, ..., 2k-2$.

WENO reconstruction in (D.51) provides new one-dimensional average-like values along the x direction. Therefore, it is possible to repeat the previous procedure but in this case departing from $\tilde{q}^e(y)$ instead of cell averages of u. Now, as in the previous case k different stencils are taken according to parameter r_1 with corresponds to the x direction. Inside each stencil, the the following polynomials are constructed

$$\tilde{\tilde{p}}_{r_1}(x, y) = p\left(r_1, k, x, \{\tilde{q}^e(y)\}_{e=i-r_1,\dots,i-r_1+k-1}\right) \tag{D.52}$$

Finally, a $2k-1$-th order WENO reconstruction is carried out using polynomials in (D.52) as

$$\tilde{\tilde{q}}(x, y) = q\left(k, x, \{\tilde{\tilde{p}}_{r_1}(x, y)\}_{r_1=0,\dots,k-1}\right) \tag{D.53}$$

Up to this point, the reconstruction procedures have been only considered inside a cell $I_{i,j}$ and therefore subscripts denoting for row and column position, i and j respectively, were dropped from polynomials. In what follows, the 1D WENO reconstruction in y direction will be denoted by $\tilde{q}_{i,j}(y)$, which is equivalent to $\tilde{q}^{k-1}(y)$ according to Equation (D.51). Similarly, the 2D WENO reconstruction will be denoted by $\tilde{\tilde{q}}_{i,j}(x, y)$.

A FORTRAN subroutine for the complete WENO 2D reconstruction is presented below. The algorithm implemented here corresponds to the procedure outlined before and generates the reconstruction at every point of the grid. The reconstruction is obtained at $(2k-2)\cdot(2k-2)$ points inside each cell.

```
      SUBROUTINE  WENO_PROCEDURE2D(
      !INPUTS:
     &  UCELL,K,NCELLX,NCELLY,DELTAX,EPSI,X,Y,C,D,GAMMA,TIPO,
      !OUTPUTS:
     &  UWENO_ALL                              )

      IMPLICIT NONE

      INTEGER K,NCELLX,NCELLY,L1,L2,L3,L4,L5,L6,N1,N2,J,TIPO
      INTEGER ORD1,ORD2

      DOUBLE PRECISION EPSI,DELTAX,AUX1
      DOUBLE PRECISION UCELL     (1:NCELLX,1:NCELLY)
      DOUBLE PRECISION UCELL_Y   (1:NCELLY)
      DOUBLE PRECISION UCELL_X   (1:NCELLX)
      DOUBLE PRECISION GAMMA     (0:K-1, 1:2*K-2          )
      DOUBLE PRECISION C         (0:K-1  ,0:K-1   ,1:2*K-2)
      DOUBLE PRECISION D         (0:K-1,0:K-1 ,1:K-1, 1:3)
      DOUBLE PRECISION UWENO_Y   (1:NCELLY,1:2*K-2)
      DOUBLE PRECISION UWENO_ALL_AUX (1:NCELLX,1:NCELLY,1:2*K-2)
      DOUBLE PRECISION UWENO_X   (1:NCELLX,1:2*K-2)
      DOUBLE PRECISION UWENO_ALL(1:NCELLX,1:NCELLY,1:2*K-2,1:2*K-2)
      DOUBLE PRECISION X         (1:NCELLX,          1:2*K-2)
      DOUBLE PRECISION Y         (1:NCELLY,          1:2*K-2)
```

```
!I/O VARIABLE LIST:
!UCELL: 2D AVERAGES
!K: FOR ORDER SELECTION. WENO OF (2K-1)-TH ORDER
!NCELLX,NCELLY: NUMBER OF CELLS
!DELTAX: CELL LENGTH
!EPSI: EPSILON COEFFICIENT FOR WENO WEIGHTS CALCULATION
!X,Y: GRIDS AT WHICH THE RECONSTRUCTION IS COMPUTED
!C: COEFFICIENTS CRJ FOR LOWER ORDER RECONSTRUCTION
!D: DERIVATIVES OF COEFFICIENTS CRJ
!GAMMA: OPTIMAL WEIGHTS
!TIPO: TO SPECIFY THE TYPE OF WENO METHOD
!UWENO_ALL: ARRAY OF RECONSTRUCTED VALUES


   DO N1=1,NCELLX

       DO N2=1,NCELLY
           UCELL_Y(N2)=UCELL   (N1,N2)
       END DO

       CALL  WENO_PROCEDURE1D(
       !INPUTS:
&      UCELL_Y,K,NCELLY,DELTAX,EPSI,C,D,GAMMA,TIPO,
       !OUTPUTS:
&      UWENO_Y                              )

       DO N2=1,NCELLY
           DO L4=1,2*K-2
             UWENO_ALL_AUX(N1,N2,L4)=UWENO_Y(N2,L4)
           END DO
       END DO

   END DO

   DO N2=1,NCELLY
     DO L4=1,2*K-2

       DO N1=1,NCELLX
           UCELL_X(N1)=UWENO_ALL_AUX(N1,N2,L4)
       END DO

       CALL  WENO_PROCEDURE1D(
       !INPUTS:
&      UCELL_X,K,NCELLX,DELTAX,EPSI,C,D,GAMMA,TIPO,
       !OUTPUTS:
&      UWENO_X                              )

       DO N1=1,NCELLX
           DO L3=1,2*K-2
           UWENO_ALL(N1,N2,L3,L4)=UWENO_X(N1,L3)
           END DO
       END DO

    END DO
   END DO
```

```
      END SUBROUTINE
```

The previous procedure makes use of the 1D WENO reconstruction subroutine, which provides the values of the reconstruction at every point of the 1D grid departing from cell averages. This subroutine is invoked as

```
        CALL  WENO_PROCEDURE1D(
        !INPUTS
&       UCELL,K,NCELL,DELTAX,EPSI,C,D,GAMMA,TIPO,
        !OUTPUTS:
&       UWENO                                 )
```

where `UCELL` is the array of cell averages (row or column) and `UWENO` the array of reconstructed values in the same direction.

Appendix E

2D extension of the sub-cell WENO reconstruction of derivatives

The previously presented 1D sub-cell WENO reconstruction of derivatives is now extended to 2 spatial dimensions, again by means of a dimension-by-dimension reconstruction approach.

E.1 Derivation and description of the procedure

The procedure for the WENO sub-cell derivative reconstruction procedure is composed of the following steps:

a) *Define sub-cell points*

Sub-cell points for the cell $I_{i,j}$ are denoted by $[x_i^{(b)}, y_j^{(v)}]$ for $b, v = 1, ..., 2k-2$. In Section 1, it was shown that choosing uniformly distributed points inside the cell leads to negative optimal weights, γ_r, in the WENO reconstruction procedure. Therefore, the WENO procedure would require some modifications in order to give a good treatment to the negative weights.

As done in the 1D case, all sub-cell points are chosen to be inside the positive interval of the optimal weights, according to Figure C.1. The same formula is used for the generation of the subcell grid, for each of the coordinate directions:

$$x_i^{(b)} = \begin{cases} x_{i-\frac{1}{2}} & \text{if } b = 1 \\ x_{i-\frac{1}{2}} + \frac{\Delta x}{2 \cdot 2^{k-b}} & \text{if } 2 \leq b \leq k-1 \\ x_{i+\frac{1}{2}} - \frac{\Delta x}{2 \cdot 2^{b-k+1}} & \text{if } k \leq b \leq 2k-3 \\ x_{i+\frac{1}{2}} & \text{if } b = 2k-2 \end{cases}$$

$$y_j^{(v)} = \begin{cases} x_{j-\frac{1}{2}} & \text{if } v = 1 \\ x_{j-\frac{1}{2}} + \frac{\Delta x}{2 \cdot 2^{k-v}} & \text{if } 2 \leq v \leq k-1 \\ x_{j+\frac{1}{2}} - \frac{\Delta x}{2 \cdot 2^{v-k+1}} & \text{if } k \leq v \leq 2k-3 \\ x_{j+\frac{1}{2}} & \text{if } v = 2k-2 \end{cases}$$

b) *Reconstruct an interpolating polynomial that approximates line averages of $u(x, y)$ and its derivatives, in y direction, inside each cell:*

Line averages of $u(x, y)$ in y direction are given by $\bar{\zeta}_i(y)$, defined in (D.45). The 1D WENO sub-cell derivative reconstruction procedure in Section .. is applied to obtain an approximation of $\bar{\zeta}_i(y)$ and its spatial derivatives departing from cell averages $\bar{u}_{i,j}$. The polynomial that approximates $\bar{\zeta}_i(y)$ inside cell $I_{i,j}$ is denoted by $\bar{\phi}_{i,j}(y)$ and defined as

$$\bar{\phi}_{i,j}(y) = \sum_{l=0}^{2k-2} a_l \left(\frac{y - y_{j-\frac{1}{2}}}{\Delta y} \right)^l \tag{E.1}$$

where the coefficients a_l $(l = 0, ..., 2k-2)$ have to be determined with the following $2k-1$ equations for each cell $I_{i,j}$:

- Using the $2k-2$ reconstructed values of $\bar{\zeta}_i(y)$ at the $2k-2$ sub-cell points obtained with the WENO procedure, given by

$$\bar{\zeta}_{i,j}^{(v)} \approx \bar{\zeta}_i(y_j^{(v)}) \tag{E.2}$$

we set

$$\bar{\phi}_{i,j}\left(y_j^{(v)}\right) = \bar{\zeta}_{i,j}^{(v)}, \qquad v = 1, ..., 2k-2 \tag{E.3}$$

- Using the cell average, $\bar{u}_i$, the following equation is formulated

$$\frac{1}{\Delta y}\int_{I_{i,j}} \bar{\phi}_{i,j}(y)\, dy = \sum_{l=0}^{2k-2} \frac{a_l}{l+1}\left[\left(\frac{y - y_{j-\frac{1}{2}}}{\Delta y}\right)^{l+1}\right]_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} = \bar{u}_{i,j} \tag{E.4}$$

Polynomial $\bar{\phi}_{i,j}(y)$ is determined, providing the following approximations

$$\begin{aligned} \bar{\phi}_{i,j}(y) &\approx \bar{\zeta}_i(y) \\ \frac{\partial^m}{\partial y^m}\bar{\phi}_{i,j}(y) &\approx \frac{\partial^m}{\partial y^m}\bar{\zeta}_i(y) \end{aligned} \tag{E.5}$$

c) *Reconstruct an interpolating polynomial that approximates $u(x, y)$ and its derivatives in x, inside each cell and at each quadrature point:*

The 1D WENO sub-cell derivative reconstruction procedure in Section .. is applied now in the x direction to obtain an approximation of $u(x, y)$ and its spatial derivatives in x, at each quadrature y-point $\mathcal{G}_{y_j}^{(e_2)}$, departing from line averages $\bar{\zeta}_i(\mathcal{G}_{y_j}^{(e_2)})$ obtained in (E.5). The polynomial that approximates $u(x, y)$ inside cell $I_{i,j}$, along x and at $\mathcal{G}_{y_j}^{(e_2)}$, is denoted by $\phi_{i,j}^{(0),(e_2)}(x)$ and defined as

$$\phi_{i,j}^{(0),(e_2)}(x) = \sum_{l=0}^{2k-2} b_l^{(0)}\left(\frac{x - x_{i-\frac{1}{2}}}{\Delta x}\right)^l \tag{E.6}$$

where the coefficients $b_l^{(0)}$ $(l = 0, ..., 2k-2)$ have to be determined with the following $2k-1$ equations for each cell $I_{i,j}$ and for each quadrature point:

- Using the $2k-2$ reconstructed values of $u(x, \mathcal{G}_{y_j}^{(e_2)})$ at the $2k-2$ x-sub-cell points obtained with the WENO procedure, denoted by

$$u_{i,j}^{(b,e_2)} \approx u(x_i^{(b)}, \mathcal{G}_{y_j}^{(e_2)}) \tag{E.7}$$

we set

$$\phi_{i,j}^{(0),(e_2)}\left(x_i^{(b)}\right) = u_{i,j}^{(b,e_2)}, \qquad b = 1, ..., 2k-2 \tag{E.8}$$

- Using the cell average, $\bar{\zeta}_i(\mathcal{G}_{y_j}^{(e_2)})$, the following equation is formulated

$$\frac{1}{\Delta x}\int_{I_{i,j}} \phi_{i,j}^{(0),(e_2)}(x)\, dx = \sum_{l=0}^{2k-2} \frac{b_l^{(0)}}{l+1}\left[\left(\frac{x - x_{i-\frac{1}{2}}}{\Delta x}\right)^{l+1}\right]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} = \bar{\zeta}_i(\mathcal{G}_{y_j}^{(e_2)}) \tag{E.9}$$

Polynomials $\phi_{i,j}^{(0),(e_2)}(x)$ are determined at each $\mathcal{G}_{y_j}^{(e_2)}$, providing the following approximations

$$\begin{aligned} \phi_{i,j}^{(0),(e_2)}(x) &\approx u(x, \mathcal{G}_{y_j}^{(e_2)}) \\ \frac{\partial^n}{\partial x^n}\phi_{i,j}^{(0),(e_2)}(x) &\approx \frac{\partial^n}{\partial x^n} u(x, \mathcal{G}_{y_j}^{(e_2)}) \end{aligned} \tag{E.10}$$

and allow to compute the approximation of $u(x,y)$ and its derivatives at the quadrature points $(x,y) = (\mathcal{G}_{x_i}^{(e_1)}, \mathcal{G}_{y_j}^{(e_2)})$ by evaluating (E.10) at $x = \mathcal{G}_{x_i}^{(e_1)}$.

d) *Reconstruct interpolating polynomials for cross derivatives and y derivatives of $u(x,y)$, along x, inside each cell and for each quadrature point:*

The calculation of derivatives of $u(x,y)$ in the x direction is straightforward when departing from information related to x-averaged values, as done in the previous step. However, the computation of derivatives in the y direction as well as cross derivatives require an additional step since the former were already calculated in the second step as averages in x.

Now, we seek derivatives of the type

$$\frac{\partial^{n+m}}{\partial x^n \partial y^m} u(x,y) \tag{E.11}$$

with $m+n \leq 2k-2$ and $m > 0$, since the case when $m = 0$ corresponds to the previous step. The departing data will be x-averages of (E.11) with $n = 0$ and $m = 1, ..., 2k-2$, that can be expressed as derivatives of line averages $\bar{\zeta}_i(y)$

$$\frac{\partial^m}{\partial y^m}\bar{\zeta}_i(y) = \int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial^m}{\partial y^m} u(x,y) dx \tag{E.12}$$

computed straightforward from derivatives of $\bar{\phi}_{i,j}(y)$ in (E.5).

For each value of m, the 1D WENO sub-cell derivative reconstruction procedure is applied in the x direction to obtain an approximation of $\frac{\partial^m}{\partial y^m}u(x,y)$ and its spatial derivatives in x (cross derivatives). This procedure will be carried out at each quadrature y-point $\mathcal{G}_{y_j}^{(e_2)}$.

The polynomial that approximates $\frac{\partial^m}{\partial y^m}u(x,y)$ inside cell $I_{i,j}$, along x and at $\mathcal{G}_{y_j}^{(e_2)}$, is denoted by $\phi_{i,j}^{(m),(e_2)}(x)$ and defined as

$$\phi_{i,j}^{(m),(e_2)}(x) = \sum_{l=0}^{2k-2} b_l^{(m)} \left(\frac{x - x_{i-\frac{1}{2}}}{\Delta x} \right)^l \tag{E.13}$$

where the coefficients $b_l^{(m)}$ $(l = 0, ..., 2k-2)$ have to be determined with the following $2k-1$ equations for each cell $I_{i,j}$ and for each quadrature point:

- Using the $2k-2$ reconstructed values of $\frac{\partial^m}{\partial y^m}u(x,y)\Big|_{y=\mathcal{G}_{y_j}^{(e_2)}}$ at the $2k-2$ x-sub-cell points obtained with the WENO procedure, we set

$$\phi_{i,j}^{(m),(e_2)}\left(x_i^{(b)}\right) = \frac{\partial^m}{\partial y^m}u(x,y)\Big|_{\substack{y=\mathcal{G}_{y_j}^{(e_2)} \\ x=x_i^{(b)}}}, \qquad b = 1, ..., 2k-2 \tag{E.14}$$

- Using the cell average of the m-th derivative in y direction, $\frac{\partial^m}{\partial y^m}\bar{\zeta}_i(y)\Big|_{y=\mathcal{G}_{y_j}^{(e_2)}}$, the following equation is formulated

$$\frac{1}{\Delta x}\int_{I_{i,j}} \phi_{i,j}^{(m),(e_2)}(x)\, dx = \sum_{l=0}^{2k-2} \frac{b_l^{(m)}}{l+1} \left[\left(\frac{x - x_{i-\frac{1}{2}}}{\Delta x} \right)^{l+1} \right]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} = \frac{\partial^m \bar{\zeta}_i(y)}{\partial y^m}\Big|_{y=\mathcal{G}_{y_j}^{(e_2)}} \tag{E.15}$$

Polynomials $\phi_{i,j}^{(m),(e_2)}(x)$ are determined at each quadrature point $\mathcal{G}_{y_j}^{(e_2)}$, providing the following approximations

$$\begin{aligned} \phi_{i,j}^{(m),(e_2)}(x) &\approx \frac{\partial^m}{\partial y^m} u(x, \mathcal{G}_{y_j}^{(e_2)}) \\ \frac{\partial^n}{\partial x^n} \phi_{i,j}^{(m),(e_2)}(x) &\approx \frac{\partial^n}{\partial x^n} \left(\frac{\partial^m}{\partial y^m} u(x, \mathcal{G}_{y_j}^{(e_2)}) \right) \end{aligned} \tag{E.16}$$

with $n \leq 2k - 2 - m$. They allow to compute the approximation of $u(x, y)$ and its derivatives at the quadrature points $(x, y) = (\mathcal{G}_{x_i}^{(e_1)}, \mathcal{G}_{y_j}^{(e_2)})$ by evaluating (E.16) at $x = \mathcal{G}_{x_i}^{(e_1)}$.

E.2 Numerical results

The 2D extension of the WENO sub-cell derivative reconstruction procedure proposed in this work is used to reconstruct the following function

$$u(x, y) = \cos\left(\frac{2\pi}{100} x\right) \cos\left(\frac{2\pi}{100} y\right), \tag{E.17}$$

depicted in Figure E.1 and its derivatives, inside the spatial domain $(x, y) = [0, 100] \times [0, 100]$ using a grid size of $\Delta x = \Delta y = 5$. The derivation of analytical expressions for partial derivatives of (E.17) is straightforward, leading to

$$\frac{\partial^n u(x,y)}{\partial x^n} = \begin{cases} (-1)^{\lceil \frac{n}{2} \rceil} \left(\frac{2\pi}{100}\right)^n \sin\left(\frac{2\pi}{100} x\right) \cos\left(\frac{2\pi}{100} y\right) & \text{if} \quad \left\lceil \frac{n}{2} \right\rceil \neq \frac{n}{2} \\ (-1)^{\lceil \frac{n}{2} \rceil} \left(\frac{2\pi}{100}\right)^n \cos\left(\frac{2\pi}{100} x\right) \cos\left(\frac{2\pi}{100} y\right) & \text{if} \quad \left\lceil \frac{n}{2} \right\rceil = \frac{n}{2} \end{cases} \tag{E.18}$$

$$\frac{\partial^m u(x,y)}{\partial y^m} = \begin{cases} (-1)^{\lceil \frac{m}{2} \rceil} \left(\frac{2\pi}{100}\right)^m \sin\left(\frac{2\pi}{100} y\right) \cos\left(\frac{2\pi}{100} x\right) & \text{if} \quad \left\lceil \frac{m}{2} \right\rceil \neq \frac{m}{2} \\ (-1)^{\lceil \frac{m}{2} \rceil} \left(\frac{2\pi}{100}\right)^m \cos\left(\frac{2\pi}{100} y\right) \cos\left(\frac{2\pi}{100} x\right) & \text{if} \quad \left\lceil \frac{m}{2} \right\rceil = \frac{m}{2} \end{cases} \tag{E.19}$$

where $\lceil a \rceil = \text{ceiling}(a)$ is the ceiling function, which provides the largest integer not greater than $a \in \mathbb{R}$. It can be noticed that both derivatives in x and y are alike and periodic for any n and m. For a later analysis of the numerical derivatives, it is worth presenting the absolute values of global maxima (or minima, due to simetry) for derivatives in (E.18) and (E.18), denoted by

$$M^n(u) = \left| \max\left(\frac{\partial^n u(x,y)}{\partial x^n} \right) \right| = \left| \max\left(\frac{\partial^n u(x,y)}{\partial y^n} \right) \right| \qquad \forall x, y \in [0, 100] \times [0, 100]. \tag{E.20}$$

Table E.1 shows numerical values for $M^n(u)$ up to $n = 6$, noticing that

$$M^n(u) = \left(\frac{2\pi}{100}\right)^n. \tag{E.21}$$

Numerical results for the reconstruction of the test function in (E.17) and pointwise numerical errors at gaussian quadrature points using a 3-rd, 5-th and 7-th 2D WENO method are presented in Figure E.2. The pointwise numerical error is calculated as the difference between numerical solution and exact solution at a given point. It can be observed that increasing the order in one level produces a decrement of the numerical error in one order of magnitude for this grid size, approximately.

Pointwise numerical errors at gaussian quadrature points in the reconstruction of partial derivatives with respect to x of the test function in (E.17) using a 3-rd, 5-th and 7-th 2D WENO method are presented

n	$M^n(u)$
0	1
1	0.0628
2	3.9478E-03
3	0.4805E-04
4	1.5585E-05
5	9.7926E-07
6	6.1528E-08

Table E.1: Absolute values of global maxima for derivatives in (E.18) and (E.18).

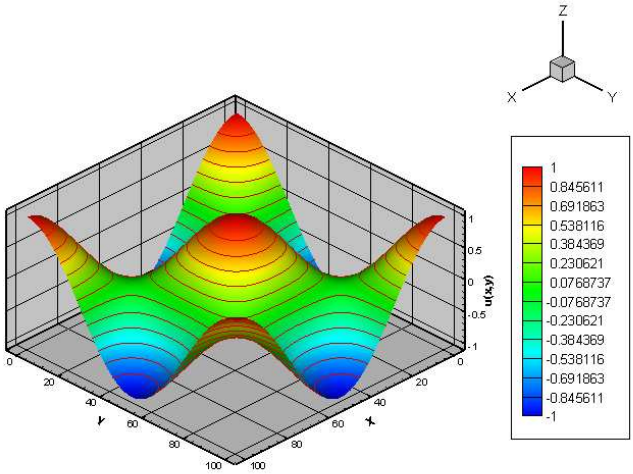

Figure E.1: Test function presented in Equation (E.17).

in Figures E.3, E.4 and E.5 respectively. It is noticed how the shape of the grid determines the error pattern. Numerical errors are within the range of acceptability that produces a proper reconstruction of derivatives. On the other hand, pointwise numerical errors at gaussian quadrature points in the reconstruction of partial derivatives with respect to y of the test function in (E.17) using a 3-rd, 5-th and 7-th 2D WENO method are presented in Figures E.6, E.7 and E.8 respectively. The same error patterns, but in y direction, are observed now.

A convergence rate test is also carried out for L_1 and L_∞ error norms using five different grid sizes: $N_x = N_y = 10, 20, 40, 80, 160$. Numerical results are presented in Table E.2 and evidence that the dimension-by-dimension WENO reconstruction algorithm achieves the prescribed order of accuracy. Remark that when computing the 9-th order WENO reconstruction, a level of error of the same order of the machine precision is reached for $N = 80$, therefore further refinements will not produce the prescribed decrement of the numerical error according to the order of the method.

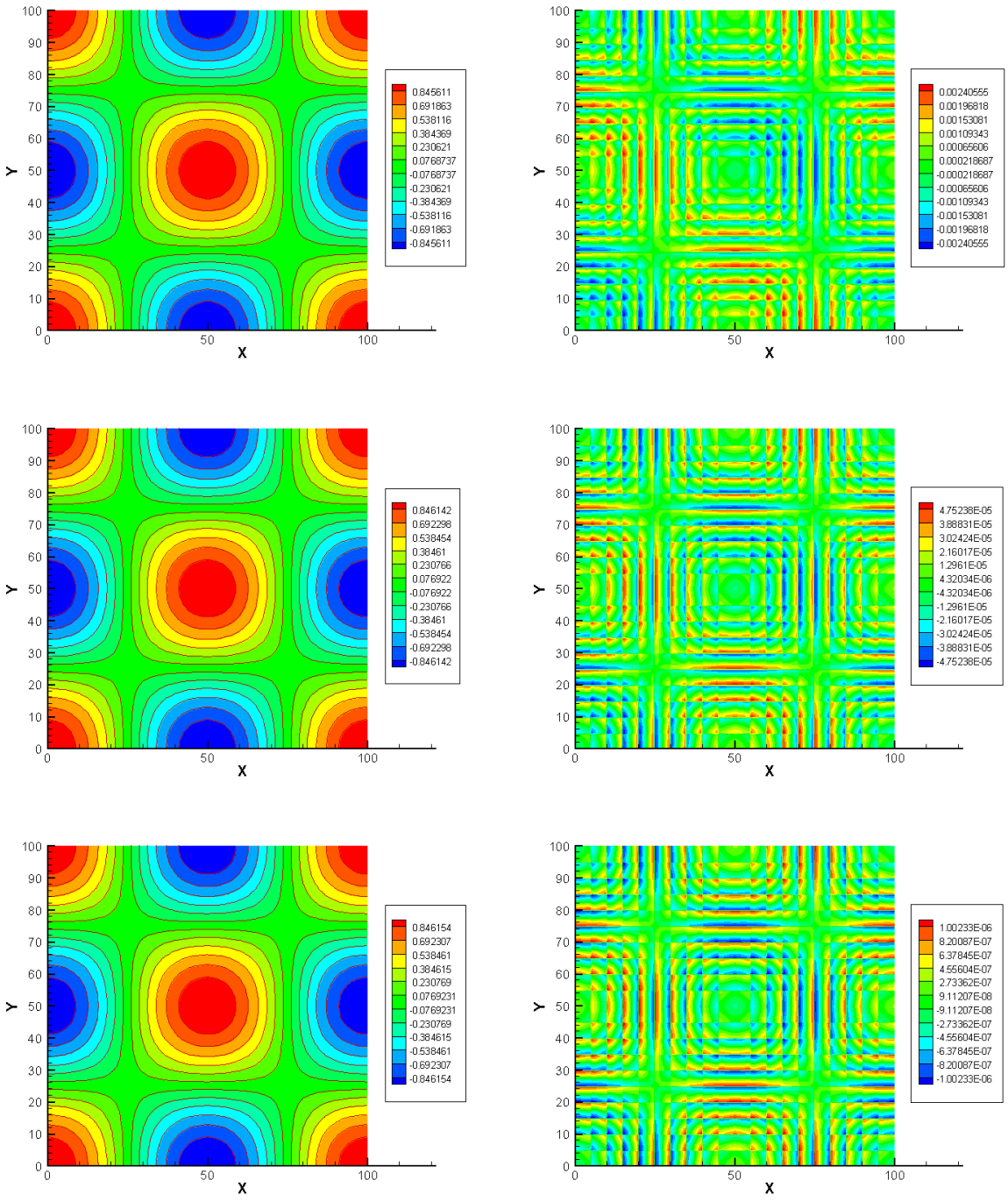

Figure E.2: Numerical results for the reconstruction of (E.17) (left) and pointwise numerical error (right) using a 3-rd, 5-th and 7-th 2D WENO method.

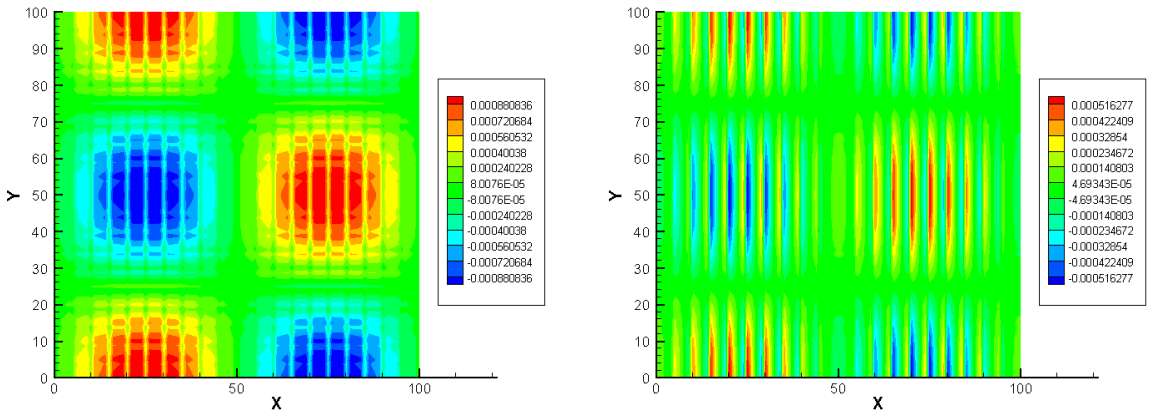

Figure E.3: Pointwise numerical error for 1-st (left) and 2-nd (right) x-derivatives of function in (E.17) using the dimension-by-dimension WENO sub-cell derivative reconstruction procedure for $k = 2$.

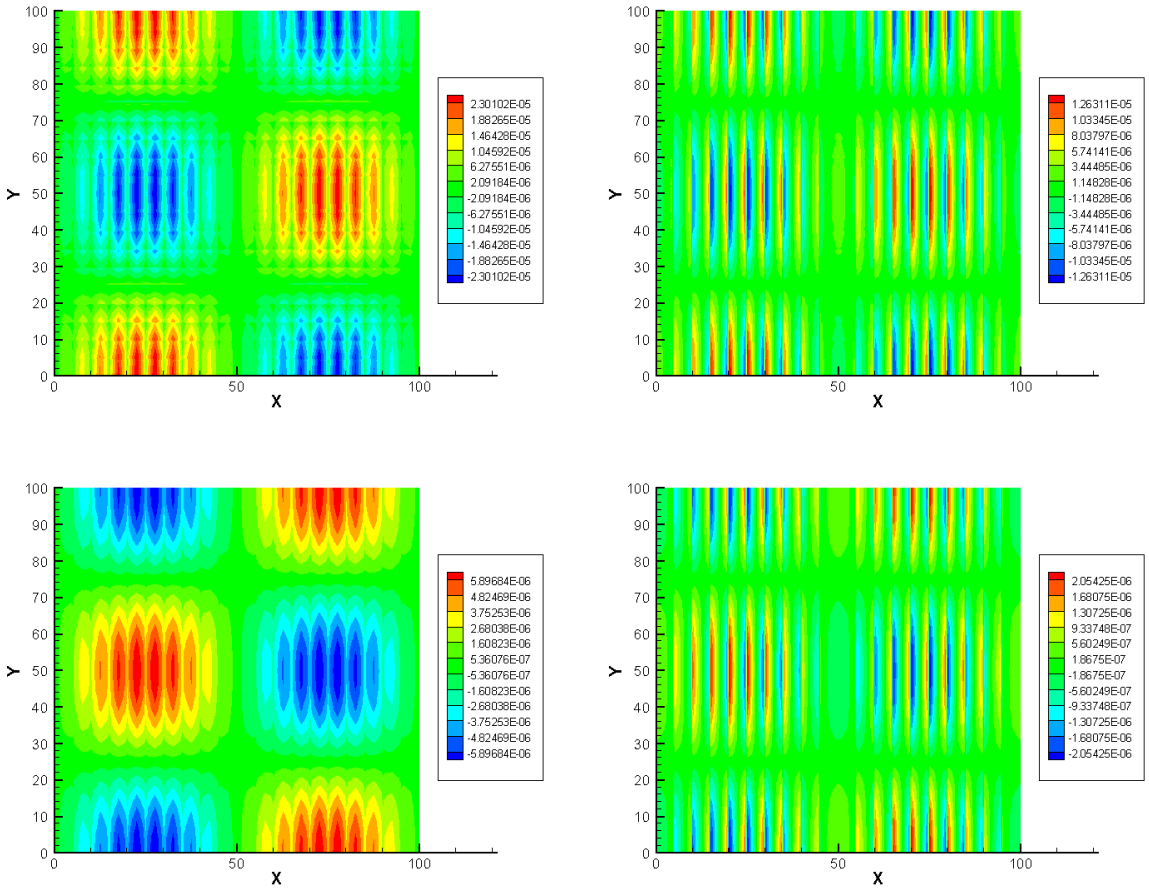

Figure E.4: Pointwise numerical error for 1-st (upper left), 2-nd (upper right), 3-rd (lower left) and 4-th (lower right) x-derivatives of function in (E.17) using the dimension-by-dimension WENO sub-cell derivative reconstruction procedure for $k = 3$.

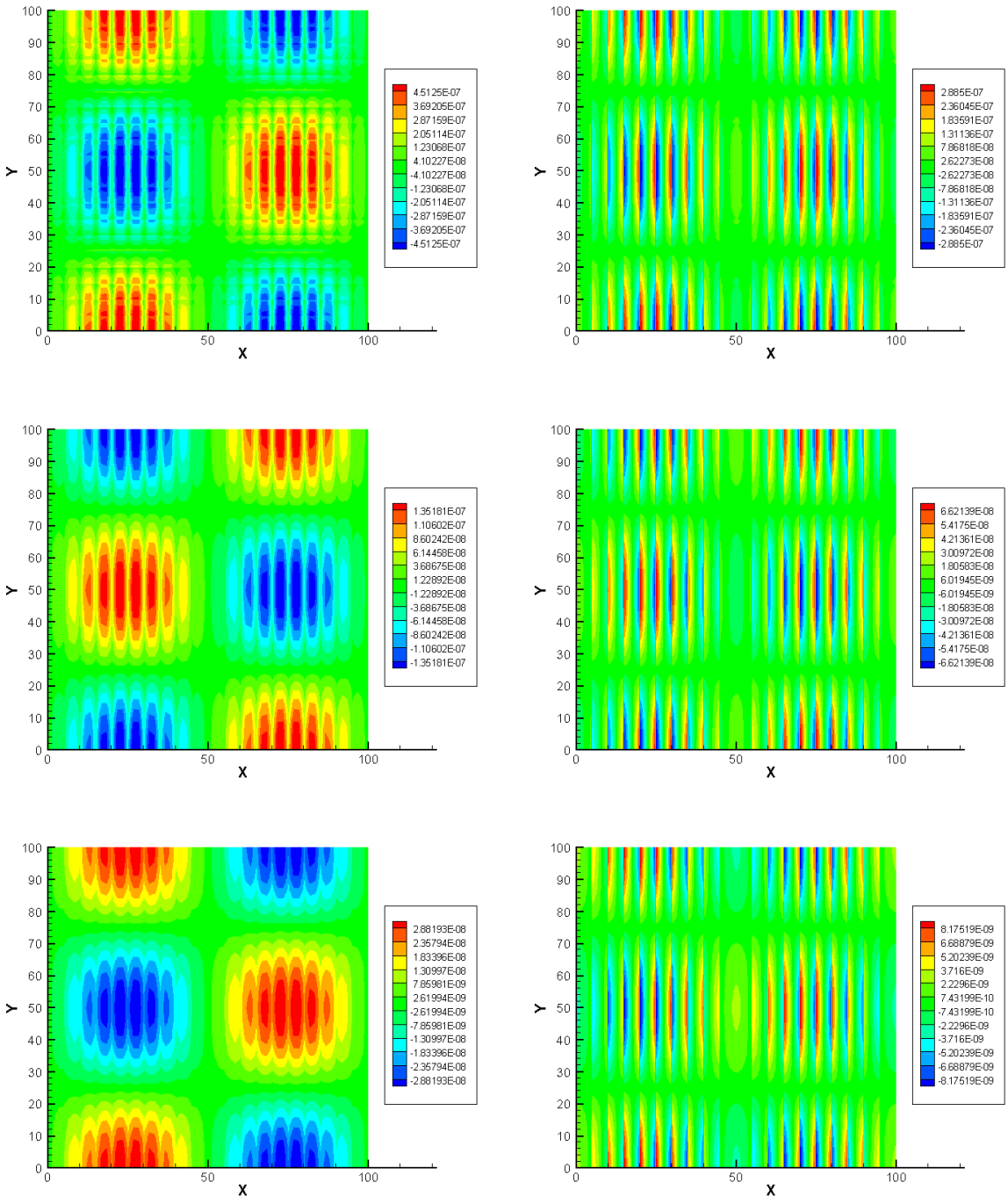

Figure E.5: Pointwise numerical error for 1-st (upper left), 2-nd (upper right), 3-rd (medium left), 4-th (medium right), 5-th (lower left) and 6-th (lower right) x-derivatives of function in (E.17) using the dimension-by-dimension WENO sub-cell derivative reconstruction procedure for $k = 4$.

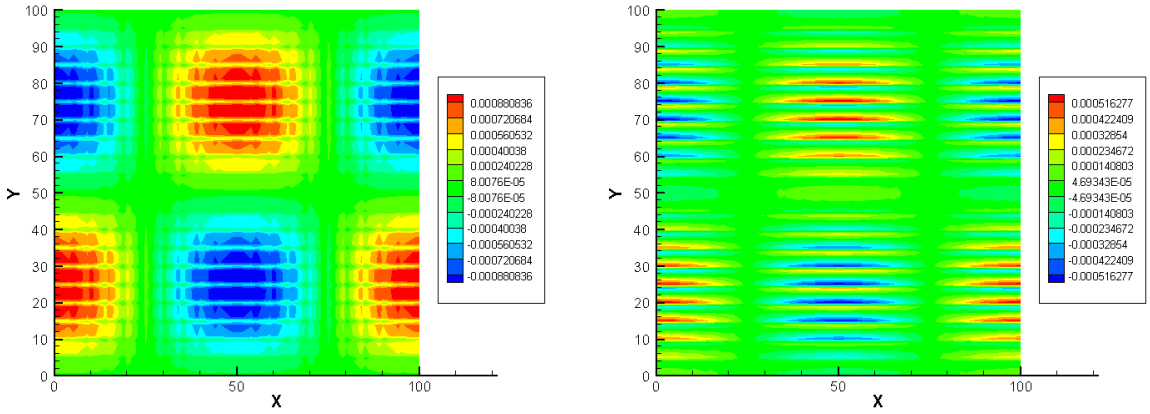

Figure E.6: Pointwise numerical error for 1-st (left) and 2-nd (right) y-derivatives of function in (E.17) using the dimension-by-dimension WENO sub-cell derivative reconstruction procedure for $k = 2$.

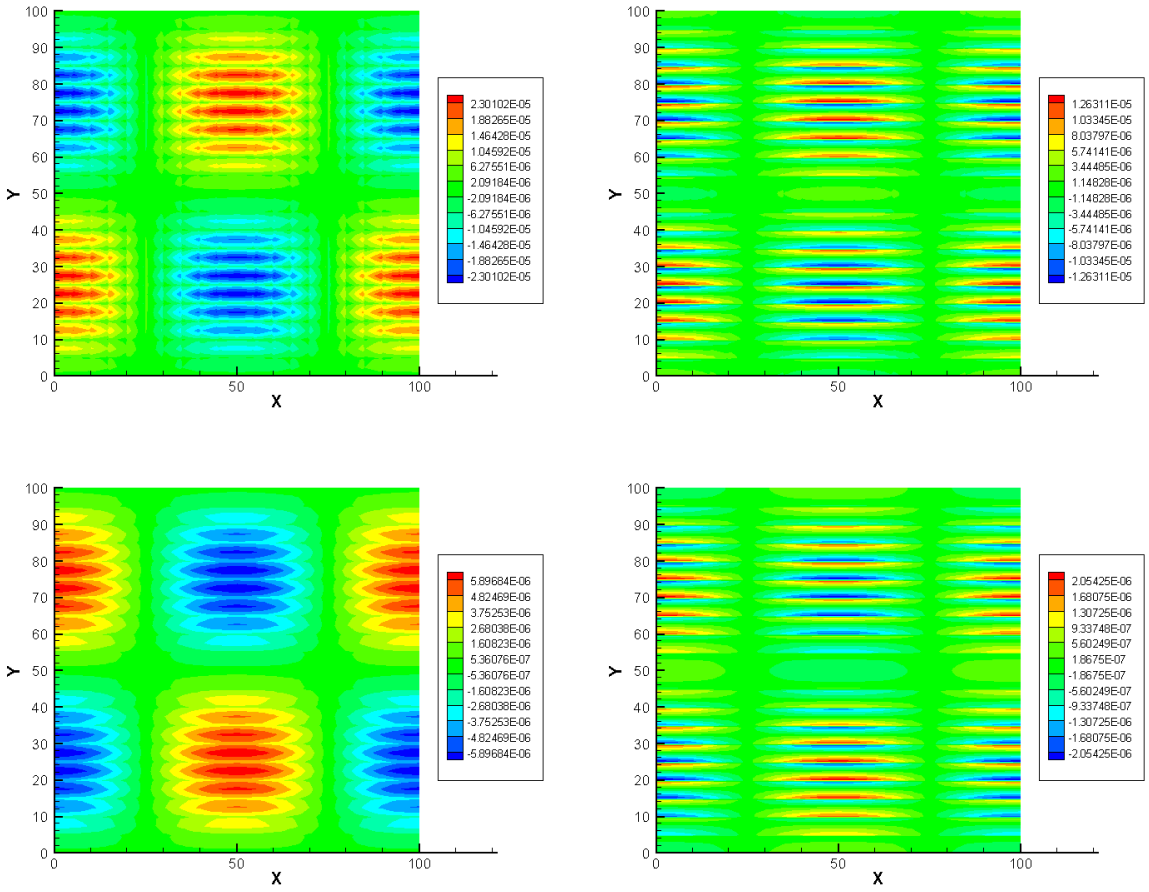

Figure E.7: Pointwise numerical error for 1-st (upper left), 2-nd (upper right), 3-rd (lower left) and 4-th (lower right) x-derivatives of function in (E.17) using the dimension-by-dimension WENO sub-cell derivative reconstruction procedure for $k = 3$.

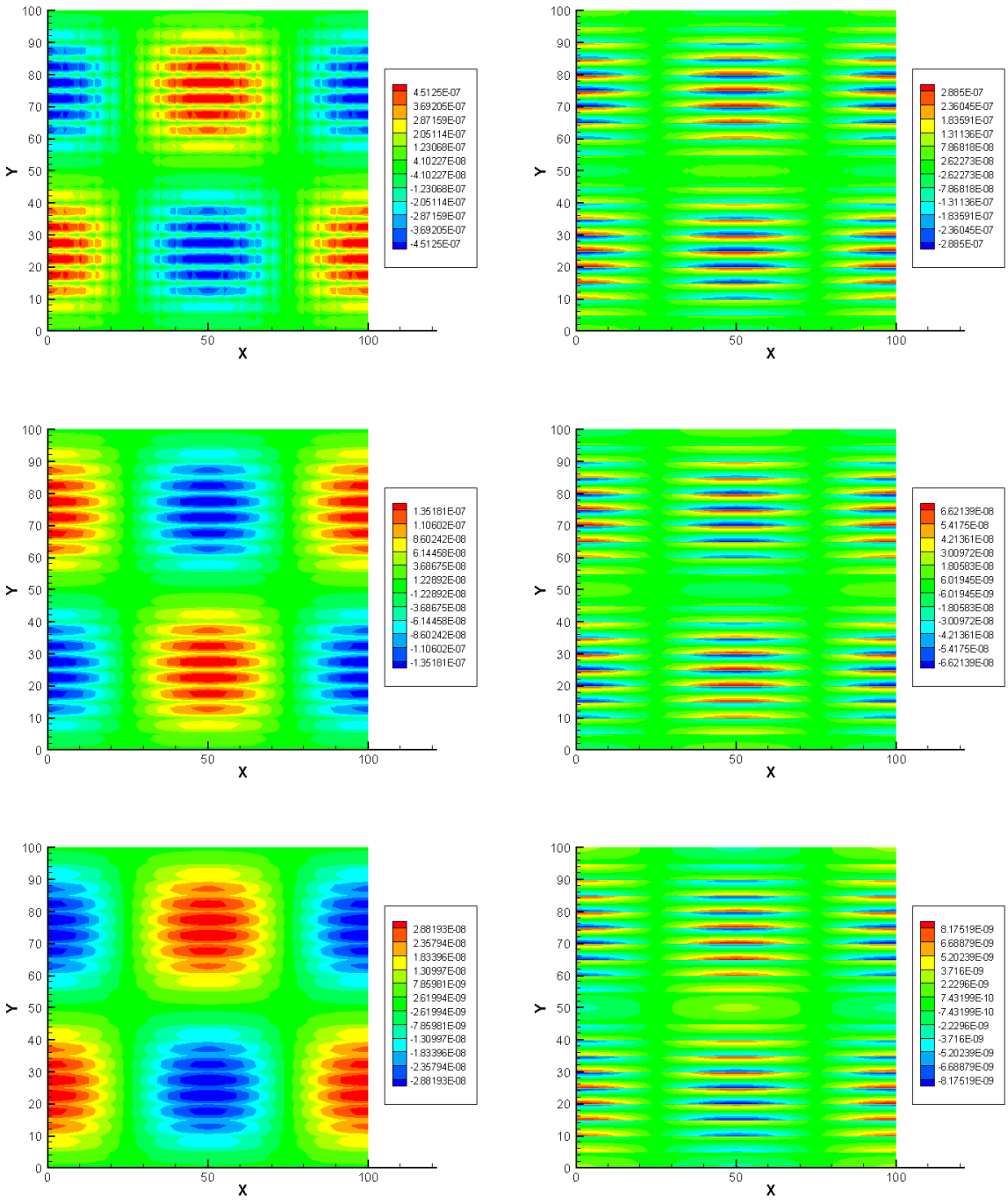

Figure E.8: Pointwise numerical error for 1-st (upper left), 2-nd (upper right), 3-rd (medium left), 4-th (medium right), 5-th (lower left) and 6-th (lower right) y-derivatives of function in (E.17) using the dimension-by-dimension WENO sub-cell derivative reconstruction procedure for $k = 4$.

k	N	L_1 error	L_1 order	L_∞ error	L_∞ order
2	10	1.08E-02	-	2.47E-02	-
	20	1.37E-03	2.97	2.84E-03	3.12
	40	1.73E-04	2.99	3.42E-04	3.06
	80	2.16E-05	3.00	4.16E-05	3.04
	160	2.70E-06	3.00	5.12E-06	3.02
3	10	7.83E-04	-	1.94E-03	-
	20	2.55E-05	4.94	5.62E-05	5.11
	40	8.03E-07	4.99	1.69E-06	5.05
	80	2.51E-08	5.00	5.14E-08	5.04
	160	7.86E-10	5.00	1.58E-09	5.02
4	10	6.59E-05	-	1.60E-04	-
	20	5.47E-07	6.91	1.18E-06	7.08
	40	4.34E-09	6.98	8.94E-09	7.05
	80	3.40E-11	6.99	6.80E-11	7.04
	160	2.66E-13	7.00	5.34E-13	6.99
5	10	5.47E-06	-	1.37E-05	-
	20	1.17E-08	8.87	2.59E-08	9.05
	40	2.32E-11	8.97	4.90E-11	9.04
	80	4.57E-14	8.99	1.03E-13	8.90
	160	3.78E-15	3.60	4.02E-14	1.36

Table E.2: L_1 and L_∞ error norms and convergence rates of the numerical solution using a 3-rd, 5-th, 7-th and 9-th order 2D WENO reconstruction method. N stands for the number of cells in each coordinate direction, which are equal in this case.

Appendix F

Convergence rate tests: tables

F.1 Linear scalar equation in Section 6.1

F.1.1 1D linear advection-reaction equation in Section 6.1.1

Approach		Optimal rec.		WENO-JS		WENO-Z		WENO-PW	
Scheme	N	L_1 error	L_1 order	L_1 error	L_1 order	L_1 error	L_1 order	L_1 error	L_1 order
ADER-3	8	13306.1047	-	14230.0031	-	13204.5891	-	14281.3691	-
	16	4236.8962	1.65	11305.9088	0.33	4227.95311	1.64	8516.08041	0.75
	20	3360.81026	1.04	9063.78775	0.99	3360.80944	1.03	5803.50283	1.72
	25	2503.53129	1.32	6964.92308	1.18	2503.53089	1.32	3668.59238	2.06
	32	1557.4721	1.92	4824.47974	1.49	1557.47206	1.92	2208.2659	2.06
ADER-5	8	8555.41998	-	13338.9284	-	11987.5178	-	8555.82623	-
	16	2975.91094	1.52	3432.32973	1.96	3226.90381	1.89	2975.89141	1.52
	20	1633.53804	2.69	2610.60903	1.23	1796.28241	2.63	1632.29859	2.69
	25	733.521245	3.59	1280.586	3.19	564.476899	5.19	732.310052	3.59
	32	246.629784	4.42	500.379951	3.81	171.345515	4.83	253.296022	4.30
ADER-7	8	4598.10939	-	10619.2407	-	10929.9357	-	4603.75039	-
	16	1927.94242	1.25	2601.30069	2.03	2290.76604	2.25	1927.51329	1.26
	20	607.305053	5.18	1172.9671	3.57	936.039569	4.01	373.166084	7.36
	25	162.613454	5.90	368.269248	5.19	479.523701	3.00	304.087784	0.92
	32	32.5639728	6.51	125.740234	4.35	97.9384238	6.43	128.807402	3.48
ADER-9	8	2273.11647	-	8048.1982	-	8717.25759	-	2280.46308	-
	16	1051.29586	1.11	2104.74364	1.94	1957.18923	2.16	850.253584	1.42
	20	202.364589	7.38	572.835132	5.83	514.762861	5.99	206.828779	6.34
	25	34.6830785	7.90	89.7847532	8.30	253.23115	3.18	31.8029298	8.39
	32	4.31840113	8.44	31.5346655	4.24	115.92998	3.17	4.3775994	8.03
ADER-11	8	1091.2802	-	6327.61844	-	6504.90881	-	1117.71902	-
	16	528.751847	1.05	1972.29691	1.68	1998.0091	1.70	595.861849	0.91
	20	65.701758	9.35	590.042993	5.41	824.56314	3.97	82.1361779	8.88
	25	7.42236012	9.77	99.2149885	7.99	131.965944	8.21	7.42233128	10.77
	32	0.57961132	10.33	11.488342	8.73	22.4695764	7.17	0.57962058	10.33

Table F.1: Section 6.1.1. L_1 error norm and convergence rate at $t = 2$ using 3-rd, 5-th, 7-th, 9-th and 11-th order TT-ADER schemes comparing the utilization of optimal reconstruction, WENO-JS, WENO-Z ($p = k - 1$) and WENO-PW ($b = 20$) approaches.

Approach		Optimal rec.		WENO-JS		WENO-Z		WENO-PW	
Scheme	N	L_2 error	L_2 order	L_2 error	L_2 order	L_2 error	L_2 order	L_2 error	L_2 order
ADER-3	8	4759.58774	-	5038.1092	-	4725.08665	-	5060.89152	-
	16	1323.21171	1.85	3179.03571	0.66	1321.05349	1.84	2375.83893	1.09
	20	886.633836	1.79	2318.98543	1.41	886.633235	1.79	1542.31598	1.94
	25	574.629197	1.94	1604.43783	1.65	574.628914	1.94	932.676931	2.25
	32	313.094188	2.46	1015.12672	1.85	313.094059	2.46	556.594848	2.09
ADER-5	8	3122.6441	-	4745.50662	-	4269.01971	-	3122.77434	-
	16	761.750568	2.04	1103.03019	2.11	877.934143	2.28	761.747763	2.04
	20	408.443423	2.79	670.120226	2.23	447.716181	3.02	408.160008	2.80
	25	162.340847	4.13	303.925439	3.54	129.802673	5.55	162.057995	4.14
	32	48.7885328	4.87	100.590569	4.48	33.3078644	5.51	50.0053902	4.76
ADER-7	8	1714.71682	-	3827.76599	-	3939.0258	-	1716.57488	-
	16	489.249109	1.81	779.734398	2.30	676.761412	2.54	489.092169	1.81
	20	152.39525	5.23	311.605999	4.11	238.448553	4.67	100.33072	7.10
	25	36.046324	6.46	87.970384	5.67	115.366855	3.25	78.6870906	1.09
	32	6.42804441	6.98	27.8324681	4.66	19.9362607	7.11	30.9293347	3.78
ADER-9	8	857.084144	-	3006.70032	-	3177.90249	-	859.822547	-
	16	268.686755	1.67	577.566108	2.38	608.508336	2.38	220.583806	1.96
	20	50.9261143	7.45	138.398012	6.40	131.143709	6.88	52.0458916	6.47
	25	7.69651749	8.47	20.5360405	8.55	74.094523	2.56	7.18481309	8.87
	32	0.85158016	8.92	7.38855907	4.14	32.8842092	3.29	0.86815679	8.56
ADER-11	8	413.427296	-	2244.72979	-	2308.22085	-	421.749877	-
	16	135.717083	1.61	511.216055	2.13	566.978749	2.03	152.301875	1.47
	20	16.5456542	9.43	145.37151	5.64	204.751979	4.56	21.1423435	8.85
	25	1.64781225	10.34	23.7583428	8.12	32.7433333	8.21	1.64780569	11.44
	32	0.1141083	10.82	3.00323624	8.38	6.04132706	6.85	0.11411019	10.82

Table F.2: Section 6.1.1. L_2 error norm and convergence rate at $t = 2$ using 3-rd, 5-th, 7-th, 9-th and 11-th order TT-ADER schemes comparing the utilization of optimal reconstruction, WENO-JS, WENO-Z ($p = k - 1$) and WENO-PW ($b = 20$) approaches.

Approach		Optimal rec.		WENO-JS		WENO-Z		WENO-PW	
Scheme	N	L_∞ error	L_∞ order	L_∞ error	L_∞ order	L_∞ error	L_∞ order	L_∞ error	L_∞ order
ADER-3	8	7307.20372	-	6765.40332	-	7186.49638	-	7072.72278	-
	16	3827.48013	0.93	7532.58033	-0.15	3822.23383	0.91	6301.01467	0.17
	20	3083.27638	0.97	6275.29631	0.82	3083.27474	0.96	4951.67022	1.08
	25	2254.00581	1.40	5603.94577	0.51	2254.00477	1.40	3763.09794	1.23
	32	1296.67019	2.24	4169.92031	1.20	1296.66944	2.24	2646.88318	1.43
ADER-5	8	5370.64667	-	7412.51587	-	6713.69222	-	5370.7992	-
	16	1654.08525	1.70	3103.85335	1.26	2150.20934	1.64	1654.10011	1.70
	20	1322.88364	1.00	2282.90301	1.38	1504.62283	1.60	1322.17174	1.00
	25	577.550628	3.71	1137.29451	3.12	459.295065	5.32	577.304705	3.71
	32	196.022863	4.38	361.683265	4.64	104.08427	6.01	195.049419	4.40
ADER-7	8	3070.30591	-	6363.88799	-	6544.75383	-	3072.97999	-
	16	1129.24995	1.44	1972.33878	1.69	1688.80656	1.95	1131.34191	1.44
	20	474.282723	3.89	871.036255	3.66	750.446362	3.63	312.909342	5.76
	25	126.967194	5.91	316.263468	4.54	513.110004	1.70	475.587709	-1.88
	32	25.5530369	6.49	163.775733	2.67	92.1308406	6.96	230.472601	2.93
ADER-9	8	1557.77701	-	5397.82803	-	5454.29866	-	1562.68283	-
	16	637.372485	1.29	1321.08832	2.03	1387.2897	1.98	531.677024	1.56
	20	156.7645	6.29	390.838957	5.46	593.380801	3.81	162.224215	5.32
	25	27.0140921	7.88	88.6007066	6.65	439.965274	1.34	27.0289199	8.03
	32	3.37478248	8.43	39.2029349	3.30	218.828296	2.83	3.99076233	7.75
ADER-11	8	755.69455	-	3424.45647	-	3530.50047	-	767.243016	-
	16	325.89132	1.21	1210.54645	1.50	1475.5338	1.26	361.693344	1.08
	20	50.7909813	8.33	462.682011	4.31	638.035073	3.76	58.3904858	8.17
	25	5.79205975	9.73	89.5268543	7.36	143.515805	6.69	5.7920388	10.36
	32	0.46043911	10.26	11.7024685	8.24	28.5317431	6.54	0.46043381	10.26

Table F.3: Section 6.1.1. L_∞ error norm and convergence rate at $t = 2$ using 3-rd, 5-th, 7-th, 9-th and 11-th order TT-ADER schemes comparing the utilization of optimal reconstruction, WENO-JS, WENO-Z ($p = k - 1$) and WENO-PW ($b = 20$) approaches.

F.1.2 2D linear advection of a Gaussian pulse in Section 6.1.3

Scheme	N. of cells	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
ADER-3	15	2.12E-02	-	3.49E-03	-	0.45553108	-
	30	6.36E-03	1.74	6.41E-04	2.45	0.17686875	1.36
	60	1.06E-03	2.59	6.16E-05	3.38	4.10E-02	2.11
	120	1.40E-04	2.92	4.25E-06	3.86	6.00E-03	2.77
ADER-5	15	1.15E-02	-	1.84E-03	-	0.24951337	-
	30	1.27E-03	3.18	1.30E-04	3.83	3.77E-02	2.73
	60	5.19E-05	4.61	3.04E-06	5.42	2.11E-03	4.16
	120	1.70E-06	4.93	5.05E-08	5.91	7.29E-05	4.86
ADER-7	15	7.81E-03	-	1.15E-03	-	0.15742517	-
	30	3.40E-04	4.52	3.46E-05	5.06	9.97E-03	3.98
	60	3.75E-06	6.50	2.20E-07	7.30	1.55E-04	6.00
	120	3.17E-08	6.89	9.40E-10	7.87	1.38E-06	6.81
ADER-9	15	5.58E-03	-	7.85E-04	-	0.10728565	-
	30	1.10E-04	5.67	1.12E-05	6.14	3.13E-03	5.10
	60	3.52E-07	8.28	2.07E-08	9.08	1.46E-05	7.75
	120	1.94E-09	7.51	5.43E-11	8.57	5.38E-08	8.08

Table F.4: L_1, L_2 and L_∞ error norms and corresponding convergence rates at $t = 30$ using a 3-rd, 5-th, 7-th and 9-th order ADER scheme in combination with the optimal reconstruction. CFL is set to 0.45. The number of cells appearing in the table corresponds to the number of cells in each direction when using a regular grid.

Scheme	N. of cells	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
ADER-3	15	2.89E-02	-	5.27E-03	-	0.65902957	-
	30	1.47E-02	0.98	1.62E-03	1.70	0.4337996	0.60
	60	4.72E-03	1.64	3.28E-04	2.30	0.24546794	0.82
	120	1.46E-03	1.69	5.81E-05	2.50	0.11402813	1.11
ADER-5	15	1.49E-02	-	2.93E-03	-	0.38028925	-
	30	2.15E-03	2.79	2.55E-04	3.52	7.50E-02	2.34
	60	1.64E-04	3.71	9.80E-06	4.70	5.74E-03	3.71
	120	6.66E-06	4.62	2.03E-07	5.60	2.93E-04	4.29
ADER-7	15	9.25E-03	-	2.06E-03	-	0.29682279	-
	30	6.92E-04	3.74	8.56E-05	4.59	2.86E-02	3.37
	60	1.20E-05	5.85	9.00E-07	6.57	9.59E-04	4.90
	120	1.70E-07	6.14	7.71E-09	6.87	2.30E-05	5.38
ADER-9	15	7.76E-03	-	1.58E-03	-	0.20997635	-
	30	3.16E-04	4.62	2.90E-05	5.77	7.33E-03	4.84
	60	9.26E-07	8.41	4.65E-08	9.28	2.64E-05	8.11
	120	2.49E-09	8.54	7.49E-11	9.28	1.30E-07	7.67

Table F.5: L_1, L_2 and L_∞ error norms and corresponding convergence rates at $t = 30$ using a 3-rd, 5-th, 7-th and 9-th order ADER scheme in combination with the WENO-JS reconstruction. CFL is set to 0.45. The number of cells appearing in the table corresponds to the number of cells in each direction when using a regular grid.

Scheme	N. of cells	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
ADER-3	15	2.13E-02	-	4.41E-03	-	0.58535917	-
	30	7.74E-03	1.46	1.04E-03	2.08	0.31367631	0.90
	60	2.14E-03	1.86	1.65E-04	2.67	0.15028625	1.06
	120	4.94E-04	2.12	2.12E-05	2.96	5.59E-02	1.43
ADER-5	15	7.70E-03	-	1.69E-03	-	0.2442285	-
	30	1.16E-03	2.73	1.28E-04	3.72	3.77E-02	2.70
	60	5.19E-05	4.49	3.04E-06	5.40	2.11E-03	4.16
	120	1.70E-06	4.93	5.05E-08	5.91	7.29E-05	4.86
ADER-7	15	4.19E-03	-	9.47E-04	-	0.14847004	-
	30	3.34E-04	3.65	3.45E-05	4.78	9.97E-03	3.90
	60	3.76E-06	6.47	2.20E-07	7.29	1.55E-04	6.00
	120	3.17E-08	6.89	9.40E-10	7.87	1.38E-06	6.81
ADER-9	15	3.89E-03	-	8.21E-04	-	0.10486486	-
	30	1.11E-04	5.13	1.11E-05	6.20	3.13E-03	5.07
	60	3.55E-07	8.29	2.07E-08	9.07	1.46E-05	7.75
	120	1.94E-09	7.52	5.43E-11	8.57	5.38E-08	8.08

Table F.6: L_1, L_2 and L_∞ error norms and corresponding convergence rates at $t = 30$ using a 3-rd, 5-th, 7-th and 9-th order ADER scheme in combination with the WENO-PW reconstruction. CFL is set to 0.45. The number of cells appearing in the table corresponds to the number of cells in each direction when using a regular grid.

Scheme	N. of cells	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
ADER-3	15	2.12E-02	-	3.49E-03	-	0.45553108	-
	30	6.36E-03	1.74	6.41E-04	2.45	0.17686875	1.36
	60	1.06E-03	2.59	6.16E-05	3.38	4.10E-02	2.11
	120	1.40E-04	2.92	4.25E-06	3.86	6.00E-03	2.77
ADER-5	15	1.30E-02	-	2.59E-03	-	0.33800925	-
	30	1.19E-03	3.45	1.27E-04	4.35	3.97E-02	3.09
	60	5.78E-05	4.36	3.13E-06	5.34	2.13E-03	4.22
	120	1.72E-06	5.07	5.06E-08	5.95	7.30E-05	4.87
ADER-7	15	9.15E-03	-	1.97E-03	-	0.2835448	-
	30	5.44E-04	4.07	5.85E-05	5.07	1.85E-02	3.94
	60	4.02E-06	7.08	2.36E-07	7.95	1.93E-04	6.58
	120	3.17E-08	6.99	9.41E-10	7.97	1.39E-06	7.12
ADER-9	15	9.06E-03	-	1.78E-03	-	0.21798476	-
	30	4.48E-04	4.34	3.66E-05	5.60	9.35E-03	4.54
	60	2.08E-05	4.43	1.33E-06	4.78	1.35E-03	2.79
	120	1.94E-09	13.39	5.43E-11	14.58	5.38E-08	14.62

Table F.7: L_1, L_2 and L_∞ error norms and corresponding convergence rates at $t = 30$ using a 3-rd, 5-th, 7-th and 9-th order ADER scheme in combination with the WENO-Z ($p = k - 1$) reconstruction. CFL is set to 0.45. The number of cells appearing in the table corresponds to the number of cells in each direction when using a regular grid.

F.2 Resolution of Burgers' equation in Section 6.2

Method	Δx	L_1 error	L_1 Order	L_2 error	L_2 Order	L_∞ error	L_∞ Order
AR-ADER 1	0.2	2.03E-02	-	6.90E-03	-	1.73E-02	-
	0.1	1.12E-02	0.9	2.93E-03	1.2	1.25E-02	0.5
	0.05	6.07E-03	0.9	1.14E-03	1.3	7.14E-03	0.8
	0.025	3.17E-03	0.9	4.74E-04	1.2	3.77E-03	0.9
AR-ADER 3	0.2	7.92E-03	-	3.03E-03	-	8.03E-03	-
	0.1	2.55E-03	1.6	6.83E-04	2.2	2.77E-03	1.5
	0.05	3.71E-04	2.8	7.94E-05	3.1	6.77E-04	2.0
	0.025	4.95E-05	2.9	7.81E-06	3.3	1.36E-04	2.3
AR-ADER 5	0.2	7.10E-03	-	2.54E-03	-	5.47E-03	-
	0.1	6.63E-04	3.4	1.66E-04	3.9	5.44E-04	3.3
	0.05	2.75E-05	4.6	4.81E-06	5.1	2.35E-05	4.5
	0.025	1.23E-06	4.5	1.73E-07	4.8	1.51E-06	4.0
MUSCLS (2^{nd})	0.2	1.25E-02	-	4.54E-03	-	1.22E-02	-
	0.1	5.04E-03	1.3	1.47E-03	1.6	4.44E-03	1.5
	0.05	1.81E-03	1.5	4.06E-04	1.8	1.50E-03	1.6
	0.025	5.53E-04	1.7	9.93E-05	2.0	5.95E-04	1.4
TT-ADER 3	0.2	1.69E-02	-	6.12E-03	-	1.44E-02	-
	0.1	8.59E-03	1.0	2.87E-03	1.1	1.29E-02	0.2
	0.05	2.57E-03	1.7	5.70E-04	2.3	3.94E-03	1.7
	0.025	5.20E-04	2.3	8.02E-05	2.8	1.16E-03	1.8

Table F.8: Section 6.2.3. Order of convergence for the 1st, 3rd and 5th order AR-ADER method, MUSCLS method (minmod) and TT-ADER scheme at $t = 0.1$s.

F.3 Application to the Shallow Water Model in Section 6.3

Variable	Δx	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
$h(m)$	0.2	1.37E-03	-	5.45E-04	-	1.55E-03	-
	0.1	8.65E-04	1.6	2.78E-04	2.3	9.12E-04	1.8
	0.05	3.25E-04	2.4	1.00E-04	2.5	5.27E-04	1.4
	0.025	8.63E-05	2.6	2.24E-05	3.0	1.41E-04	2.6
$q(m^2/s)$	0.2	9.15E-03	-	4.45E-03	-	1.36E-02	-
	0.1	6.10E-03	1.4	2.23E-03	2.5	6.96E-03	1.7
	0.05	2.23E-03	2.5	7.11E-04	2.8	3.67E-03	1.6
	0.025	5.42E-04	2.8	1.40E-04	3.2	9.17E-04	2.7

Table F.9: Section 6.3.3. Errors and orders of convergence for h and q at $t = 0.05$ s using 3-th order AR-ADER method.

Variable	Δx	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
$h(m)$	0.2	9.93E-04	-	3.70E-04	-	9.21E-04	-
	0.15	4.73E-04	2.6	1.56E-04	3.0	4.66E-04	2.4
	0.1	1.33E-04	3.3	4.00E-05	3.4	2.05E-04	2.0
	0.06	3.06E-05	2.9	7.31E-06	3.3	4.64E-05	2.9
$q(m^2/s)$	0.2	6.90E-03	-	3.04E-03	-	8.23E-03	-
	0.15	3.07E-03	2.8	1.12E-03	3.5	3.48E-03	3.0
	0.1	6.36E-04	3.9	1.91E-04	4.4	9.28E-04	3.3
	0.06	7.08E-05	4.3	2.17E-05	4.3	1.99E-04	3.1

Table F.10: Section 6.3.3. Errors and orders of convergence for h and q at $t = 0.05$ s using 5th order AR-ADER method.