Higher Order Active Flux Methods with ADER-DG Technique for Hyperbolic Conservation Laws

Suresh Nisansala K.A.

Vom Fachbereich Mathematik der
Rheinland-Pfälzischen Technischen Universität Kaiserslautern-Landau
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften (Doctor rerum naturalium)
genehmigte Dissertation

Gutachter:
PD Dr. Raul Borsche, RPTU Kaiserslautern-Landau
Prof. Dr. Christiane Helzel, Heinrich Heine Universität Düsseldorf

Datum der Disputation: 10 Juli 2024

DE-386

Acknowledgements

I would like to express my sincere gratitude to my supervisor, PD Dr. Raul Borsche, for his expert guidance, support, and encouragement throughout my research. Additionally, I am immensely grateful for his exceptional support during the challenging times of the COVID-19 pandemic at the beginning of my PhD journey. I am deeply thankful for the opportunity to expand my knowledge and personal development under his mentorship.

I take this opportunity to express my gratitude to Dr. Falk Triebsch and Jessica Borsche for their continuous support throughout my entire study period in Germany. I am also thankful to the German Academic Exchange Service (DAAD) for selecting me for the doctoral program "Mathematics in Industry and Commerce" (MIC) at RPTU Kaiserslautern-Landau and for providing financial support during my PhD study.

Furthermore, I wish to express my appreciation to all the staff members and colleagues of the Technomathematics group at RPTU Kaiserslautern-Landau for establishing an excellent working environment.

Finally, I am deeply thankful to my family, relatives, and friends for their encouragement and unwavering support, which have played a crucial role in my successful completion of this work.

Abstract

This work mainly focuses on developing higher-order numerical methods for hyperbolic conservation laws. It introduces two innovative approaches aiming to enhance the accuracy and efficiency of the methods. In the explicit setting, our objective is to modify the Active Flux schemes using the ADER-DG technique. Averaged values of the conserved quantities at the cell centers (cell averages) and the interface values of the conserved quantities (point values) are considered as the degrees of freedom of the classical Active Flux method. The conservative update formula of the proposed higher-order Active Flux (hAF) methods depends on the natural degrees of freedom of the corresponding ADER-DG scheme rather than on the cell averages. We construct a highly accurate local space-time predictor by incorporating the information from point values. The proposed scheme achieves a convergence order of $N + 3$ for degree N spatial test functions, as it applies information from a more accurate local predictor in the update formula. We propose two new update strategies for the point values. In the first a Riemann problem at each interface is solved using information from local predictor. In the second, more general approach, the conservation law is integrated over time at the cell interfaces. To address non-physical oscillations in hAF schemes near discontinuities or steep gradients, we employ the MOOD limiter with some modifications to the existing one in the literature. For linear hyperbolic problems, we achieve an improvement in the CFL number compared to the ADER-DG schemes.

In the implicit context, our approach starts by extending the existing implicit Active Flux schemes to linear systems. Then, we introduce a new update strategy for the point values, creating a new class of higher-order implicit Active Flux methods. The novel Modified Implicit Active Flux (MIAF) schemes can generate various numerical schemes. The performance of the selected MIAF schemes is determined by evaluating how they behave in standard test cases. Additionally, a stability and convergence study has been carried out. Comparing the results of the existing single-step implicit Active Flux schemes with those of the MIAF schemes shows that the new approach yields similar outcomes.

Zusammenfassung

Diese Arbeit behandelt die Entwicklung numerischer Methoden höherer Ordnung für hyperbolische Erhaltungsgleichungen. Es werden zwei neue Ansätze vorgestellt, die die Genauigkeit und Effizienz der Löser verbessern. Für explizite Verfahren modifizieren wir den Active Flux Ansatz mit Hilfe der ADER-DG-Technik. Als Freiheitsgrade der klassischen Active-Flux-Methode werden gemittelte Werte der zu erhaltenden Größen in den Gitterzellen (Zellmittelwerte) und die Punktwerte an den Zell-Interfaces betrachtet. Die konservative Update-Formel der neuen Active-Flux-Methode höherer Ordnung (hAF) hängt von den Freiheitsgraden des entsprechenden ADER-DG-Schemas und nicht von den Gitterzellen ab. Wir konstruieren einen lokalen Raum-Zeit-Prädiktor, indem wir die Information von den Punktwerten mit einbeziehen. Das vorgeschlagene Schema erreicht eine Konvergenzordnung von $N + 3$ für räumliche Testfunktionen vom Grad N, da es die Information aus einem genaueren lokalen Prädiktor in der Update-Formel verwendet. Wir schlagen zwei neue Updatestrategien für die Punktwerte vor. Bei der ersten wird ein Riemann-Problem an jedem Interface mithilfe der lokalen Prädiktoren gelöst. Bei dem zweiten, etwas allgemeineren Verfahren wird die Erhaltungsgleichung in der Zeit entlang des Zellinterfaces integriert. Um nicht-physikalische Oszillationen in den hAF-Schemata in der Nähe von Unstetigkeiten oder steilen Gradienten zu vermeiden, verwenden wir den MOOD-Limiter mit einigen Modifikationen gegenüber dem Orginal-Limiter. Für lineare hyperbolische Probleme erreichen wir eine Verbesserung der CFL-Zahl im Vergleich zu den bisherigen ADER-DG-Verfahren.

Im impliziten Kontext beginnt unser Ansatz mit der Erweiterung der bestehenden impliziten Active-Flux-Verfahren auf lineare Systeme. Dann führen wir eine neue Updatestrategie für die Punktwerte ein und schaffen so eine neue Klasse impliziter Active-Flux-Verfahren höherer Ordnung. Diese neuartigen Modified Implicit Active Flux (MIAF)-Verfahren können verschiedene numerische Verfahren erzeugen. Die Qualität der ausgewählten MIAF-Verfahren wird anhand von Standard-Testfällen untersucht und Stabilitäts- und Konvergenzstudien werden durchgeführt. Der Vergleich der Ergebnisse der bestehenden einstufigen impliziten Active-Flux-Schemata mit dem der MIAF-Schemata zeigt, dass der neue Ansatz zu ähnlichen Ergebnissen führt.

Contents

1 Introduction 1

2 ADER-DG Methods 5

2.1 Higher-order accurate ADER-DG schemes 5

2.1.1 Space discretization 5

2.1.2 The local space-time predictor 6

2.1.3 Fully-discrete formulation for the ADER-DG schemes 9

2.2 MOOD limiter for the ADER-DG schemes 10

2.3 Results 12

2.3.1 Linear scalar problem 12

2.3.2 Linear system 17

2.3.3 Non-linear scalar problem 18

2.3.4 Non-linear system 20

2.3.5 Network 22

3 Higher-Order Active Flux Methods 27

3.1 Third-order Active Flux method 27

3.1.1 Advection equation 28

3.2 Arbitrary higher-order Active Flux methods (hAF) 31

3.2.1 Discrete formulations for higher-order Active Flux schemes 32

3.2.2 MOOD limiter for the higher-order Active Flux schemes 34

3.2.3 Linear scalar problem 38

3.2.4 Linear systems 42

3.2.5 Non-linear problems 44

4 Higher Order Implicit Methods 49

4.1 Implicit finite volume methods 49

4.1.1 Third order scheme 50

4.1.2 Fourth order scheme 50

4.2 Implicit Active Flux methods 53
4.2.1 Single-step methods 55
4.2.2 Linear systems 59

4.3 Modified Implicit Active Flux Methods (MIAF) 62
4.3.1 Third-order scheme 64
4.3.2 Fourth-order scheme 66
4.3.3 Fifth-order scheme 68
4.3.4 Results, stability and convergence 71
4.3.5 Exploring a case study: fifth-order scheme with CFL $= 2$ 73

5 Conclusions and Future Works 77

Bibliography 79

Akademischer Lebenslauf 85

Academic curriculum vitae 86

1 Introduction

This work aims to develop higher-order numerical methods for hyperbolic conservation laws. Over the past few decades, there has been increasing attention on such methods due to their greater accuracy, enhanced convergence properties and improved computational efficiency. We introduce two novel approaches: one emphasizing the development of fully explicit higher-order methods and the other focusing on implicit higher-order methods.

The literature presents various notable higher-order numerical methods that are specifically designed to address hyperbolic conservation laws. Among these are the ENO (Essentially Non-Oscillatory) and WENO (Weighted Essentially Non-Oscillatory) schemes [8, 55, 56], which have the advantage of taming oscillations near discontinuities due to their construction. The discontinuous Galerkin (DG) finite element method was initially proposed in [50], and Cockburn et al. subsequently developed it through a series of publications [17–19]. Classical DG schemes encounter a challenge in matching spatial accuracy equivalent to their temporal accuracy, as they also employ Runge-Kutta methods for time evolution. Ben-Artzi and Falcovitz [9–12] were the first to develop a second-order finite volume method based on the solution of generalized Riemann problems. In order to construct arbitrary higher-order methods, Titarev, Toro et al. introduced a new approach named ADER, which stands for "Arbitrary high-order schemes using derivatives" [58, 59]. By combining the ADER approach with the discontinuous Galerkin (DG) framework, Dumbser et al. [30] developed a single-step scheme (ADER-DG) that provides higher-order accuracy in both space and time and is more efficient compared to the classical Runge-Kutta DG schemes.

Moreover, another class of higher-order numerical methods named Active Flux (AF), was proposed by Roe et al. [35, 51, 52], based on the concept introduced in [62]. The cell averages and the point values at cell interfaces serve as the degrees of freedom for the classical Active Flux method, attaining third-order accuracy due to quadratic reconstruction. The update of the point values uses the characteristic form of the equations, while the update of the cell average values uses the conservative form. The classical AF method computes the averaged fluxes at the cell interfaces using Simpson's rule. In the context of linear hyperbolic problems, characteristic information is applied [36] to obtain quadrature point values along the cell interfaces. A new approach proposed in [42] introduces the use of Taylor series expansion with respect to the time variable and applies Cauchy-Kowalewski procedure to compute the quadrature point values. For more detailed discussions about the AF method, including its applications in non-linear hyperbolic problems, refer to [4, 5, 35, 38, 49]. Furthermore, [3] introduced several possible approaches to enhance accuracy by expanding the stencil or incorporating higher moments as new degrees of freedom. In [36], an extension of the AF scheme to a two-dimensional problem using an unstructured triangular grid is discussed, while [7, 15, 42] introduced Cartesian grid approaches.

Motivated by the technique proposed in [42], we introduce new higher-order Active Flux (hAF) schemes. In this approach, the conservative update relies on the natural degrees of freedom within the corresponding fully discrete formulation for the ADER-DG scheme rather than on cell averages. The higher-degree spatial reconstruction polynomial is constructed in each cell, incorporating the information from point values. This polynomial provides the necessary initial data to build a more accurate local space-time predictor. When constructing the local predictor, we can use various types of test functions, as employed in [23, 24]. We apply space-time nodal basis test functions passing through the tensor-product Gauss-Legendre quadrature points on the space-time reference element similar to that proposed in [33]. This approach yields sparse block structures in the resulting matrices and enables possible extensions to multi-dimensional problems efficiently. Applying the information from the local predictor to the update formulas in hAF methods allows us to achieve higher-order accuracy. Two new approaches are suggested for updating the point values. The first one needs to solve the Riemann problem at each interface, making it a highly problem-based approach, while the second approach is more general. In contrast to the standard ADER-DG approach, which attains a convergence order of $N+1$ for degree N spatial test functions, the new approach achieves a convergence order of $N+3$. The approach shares similarities with the $P_N P_M$ schemes proposed in [23, 32, 39]. However, in our method, the development of the spatial reconstruction polynomial is independent of neighboring cell interactions.

According to the Godunov theorem [41], any linear scheme with an accuracy greater than one generates oscillations near discontinuities. Even with smooth initial conditions, discontinuities are natural phenomena in hyperbolic conservation laws. Therefore, when employing higher-order numerical schemes, it is critical to apply so-called limiters to prevent non-physical oscillations near discontinuities or steep gradients. The *slope-limiter* methods introduced by van Leer [63] and the *flux-limiter* methods [47, 60] are widely recognized as commonly used limiting techniques. These methods are known as *a priori* limiters because they activate based on the data at the current time level t^n. Alternatively, we direct our attention to the application of an *a posteriori* limiter named Multi-dimensional Optimal Order Detection (MOOD), which is activated based on information from both the current time level (t^n) and the future time level (t^{n+1}). The literature [21, 33, 39, 48, 64, 65], extensively discussed the successful application of the MOOD limiter to higher-order methods in hyperbolic conservation laws. The exploration of the mechanism of the MOOD limiter within the context of the ADER-DG scheme with degree N spatial test functions corresponds to [39]. Initially construct a finer subgrid by dividing each cell into $N_s = 2N+1$ elements. The projection operator generates subcell averages for the discrete candidate solution. At each time step, these candidate subcell averages are checked against the detection criteria to identify troubled cells. For the troubled cells, local recomputation of the solution is performed using a simple and robust finite volume scheme, such as the ADER-WENO finite volume scheme [33] or the second-order TVD finite volume scheme [13, 29] to the subcell averages. Finally, the reconstruction operator can recover the higher-order accurate solution for the troubled cells. The choice of the number of subcells, $N_s = 2N+1$, is intentional. It corresponds to the optimal time step size of the finite volume scheme applied on subgrids, making it match with the time step size of the ADER-DG scheme applied on main grids.

In the context of Active Flux methods, various strategies for limiters have been explored [3–5, 53]. Different types of reconstructions, including hyperbolic reconstruction, piecewise polynomial reconstruction, and discontinuous reconstruction were proposed in [42]. Our objective is to apply the MOOD limiter to the hAF schemes. The MOOD limiter follows a similar approach to the innovative higher-order Active Flux methods, with modifications made for the projection operator, reconstruction operator, and detection criteria, as they involve point values. We suggest two different approaches for limiting the point values of the troubled cells. One is based on the characteristics of the problem, while the second one is more general.

Although higher-order implicit numerical methods are not yet widely known for hyperbolic conservation laws, we are incorporating the concept outlined in [6] to propose new higher-order implicit Active Flux schemes. Implicit methods typically relax the strict time restrictions imposed by higher-order explicit methods. The finite difference/volume type higher-order implicit numerical schemes applied in district heating networks, as discussed in [34], are based on the idea proposed in [57]. Classical implicit schemes such as Crank-Nicolson, Radau IA, Radau IIA, and DIRK (Crouzeix) are applied to study semi-discrete Active Flux methods discussed in [3]. The Radau and DIRK schemes are computationally expensive since they exhibit a multi-step nature. Therefore, we focus on the single-step implicit higher-order Active Flux methods as proposed in [6]. They employed a single reconstruction polynomial in time to update both cell averages and point values. We propose the Modified Implicit Active Flux (MIAF) schemes and introduce a new update strategy for the point values. In our novel approach, the update of cell averages and point values involves two distinct reconstruction polynomials in time. This method expands the potential for a wide range of implicit Active Flux schemes.

The thesis is organized as follows: In Chapter 2, it provides a comprehensive overview of the ADER-DG schemes for hyperbolic conservation laws, including the MOOD limiter. Additionally, it presents results for various test cases using ADER-DG schemes and discusses an application to a network. Chapter 3 presents our innovative higher-order Active Flux schemes, with modifications to the MOOD limiter. The validation of the proposed schemes is done through standard test cases. Chapter 4 mainly focuses on implicit higher-order numerical methods. The Modified Implicit Active Flux (MIAF) methods are introduced, and a detailed explanation is provided through the selected schemes. Chapter 5 concludes the thesis and outlines future work.

2 ADER-DG Methods

2.1 Higher-order accurate ADER-DG schemes

Initially, Ben-Artzi and Falcovitz proposed to study the Generalized Riemann problem (GRP) to achieve a second-order finite volume method. Toro's modifications further advanced the development of higher-order accurate methods in both space and time. Based on the solution of GRP, Titarev, Toro et al. [58, 59] proposed a new approach called ADER, which stands for "**A**rbitrary high-order schemes using **der**ivatives". By combining the ADER approach with the discontinuous Galerkin (DG) framework, Dumbser et al. [30] developed a single-step scheme, known as ADER-DG, which is higher-order accurate and more efficient than classical Runge-Kutta DG schemes. This chapter provides a comprehensive overview of ADER-DG schemes [14, 23–26, 37, 40, 45, 46] for one-dimensional hyperbolic conservation laws, including the processes of developing higher-order ADER-DG solvers. It explores discussions on stability, order of convergence, results for hyperbolic conservation law problems, and their applications in a network.

To develop higher-order accurate ADER-DG solvers, we consider a general nonlinear, time-dependent hyperbolic system of conservation laws expressed as

$$\begin{aligned} \frac{\partial}{\partial t}\boldsymbol{u} + \frac{\partial}{\partial x}\boldsymbol{f}(\boldsymbol{u}) = 0, \qquad x \in \Omega \subset \mathbb{R},\ t \in \mathbb{R}_0^+, \\ \boldsymbol{u}(x,0) = \boldsymbol{u}_0(x), \qquad \forall x \in \Omega. \end{aligned} \tag{2.1}$$

Where $\boldsymbol{u} = \boldsymbol{u}(x,t)$ denotes the state vector of conserved quantities, $\boldsymbol{f}(\boldsymbol{u})$ represents the possibly nonlinear flux, and Ω stands for the computational domain. The initial condition, denoted as $\boldsymbol{u}_0(x)$, defines the starting state of the problem.

2.1.1 Space discretization

In a one-dimensional domain Ω, let's denote the end points of each element as $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$, where index i ranges from 1 to the total number of elements N_E. The size of each element is represented by $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$, and the spatial elements $T_i = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]$ hold the following relation

$$\Omega = \bigcup_{i=1}^{N_E} T_i.$$

In physical coordinates, the i^{th} space-time control volume at time t^n is expressed as $\mathcal{Q}_i^n = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] \times [t^n, t^n + \Delta t]$. Here, the time step Δt is defined as $\Delta t = t^{n+1} - t^n$. The spatial reference element, denoted as T_E, is established as $T_E = [0,1]$. The spatial coordinate is given by $0 \leq \xi \leq 1$ with the transformation relation $x = x_{i-\frac{1}{2}} + \xi \cdot \Delta x$. The mapping from physical coordinates to reference coordinates and its inverse mapping concerning the spatial element T_i can be written in a concise notation as follows [23, 27, 28]

$$x = x(T_i, \xi), \qquad \xi = \xi(T_i, x).$$

At the beginning of each time step, the DG method provides the discrete representation of the solution ($\boldsymbol{u}_h(x, t^n)$) for the vector of conserved quantities in the i^{th} element using the piecewise polynomials of degree N as

$$\boldsymbol{u}_h(x, t^n) = \boldsymbol{u}_h(\xi(x), t^n) = \sum_{m=1}^{N+1} \hat{\boldsymbol{u}}_m(t^n)\Phi_m(\xi) = \hat{\boldsymbol{u}}_m(t^n)\Phi_m(\xi), \qquad x \in T_i. \tag{2.2}$$

Where $\hat{\boldsymbol{u}}_m(t^n)$ represents the degrees of freedom at time t^n, and $\Phi_m = \Phi_m(\xi)$ are the space-only dependent test functions defined in the spatial reference element T_E. We introduced Einstein summation, in which we implicitly sum over repeated indices. Hereinafter, we will use this notation to present our equations in a more concise manner. Throughout this work, we have used a set of orthogonal spatial basis functions following a similar approach presented in [33]. The Gauss-Legendre quadrature points serve as the nodal points for these basis functions. For a visual representation of the spatial basis functions for degree 2, please refer to Figure 2.1. Alternatively, one can use scaled Legendre polynomials as spatial test functions, as detailed in [24].

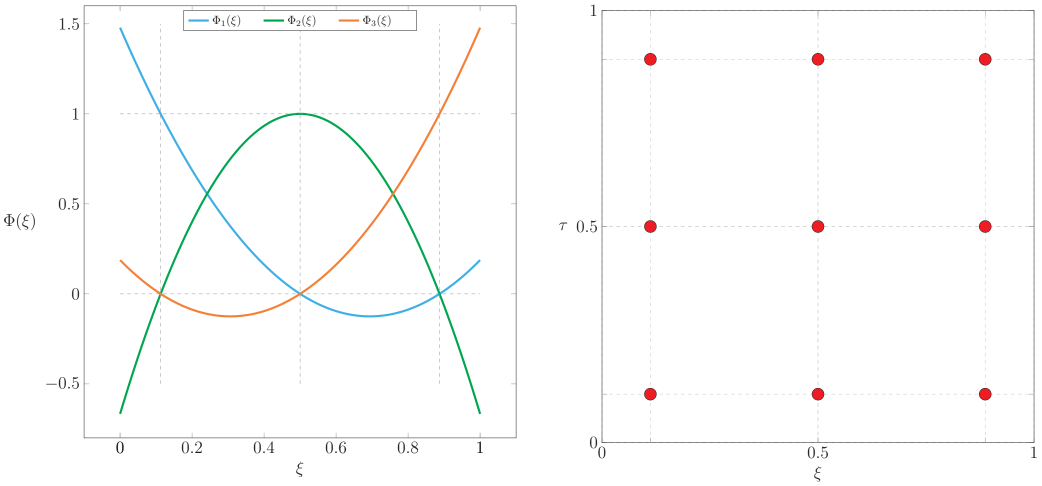


Figure 2.1: The left figure illustrates degree 2 space-only dependent test functions within T_E. The right figure represents nodal points corresponding to degree 2 space-time test functions on $\mathcal{Q}_E$.

2.1.2 The local space-time predictor

To construct the local weak formulation of the conservation law problem given in (2.1) in space and time, we define a space-time reference element $\mathcal{Q}_E = [0, 1] \times [0, 1] \in \mathbb{R}^2$. The following relations define the spatial and temporal coordinates

$$x = x_{i-\frac{1}{2}} + \xi \cdot \Delta x, \qquad t = t^n + \tau \cdot \Delta t, \tag{2.3}$$

with $0 \leq \xi \leq 1$ and $0 \leq \tau \leq 1$.

To govern the local space-time test functions $\theta_k = \theta_k(\xi, \tau)$, we employ the tensor product Gauss-Legendre quadrature points on the space-time reference element $\mathcal{Q}_E$ as the nodal points, following a similar approach to [14, 33, 37, 64]. The index k ranges from 1 to the number of degrees of freedom $N_d = (N+1)^2$. Figure 2.1 illustrates the placement of nodal points within $\mathcal{Q}_E$ used to generate space-time test functions. These test functions provide the necessary information for the full ADER-DG discretization, ensuring the scheme maintains higher-order accuracy in space and time. Alternatively, one could construct these space-time polynomials using tensor product scaled Legendre polynomials, as discussed in [24], or by considering different degrees of freedom inside space-time reference element $\mathcal{Q}_E$ as proposed in [23]. The local space-time test functions play a crucial role in constructing the higher-order Active Flux methods and the modification for the limiter discussed in Chapter 3. Figure 2.2 visualizes the set of space-time test functions corresponding to degree 2.

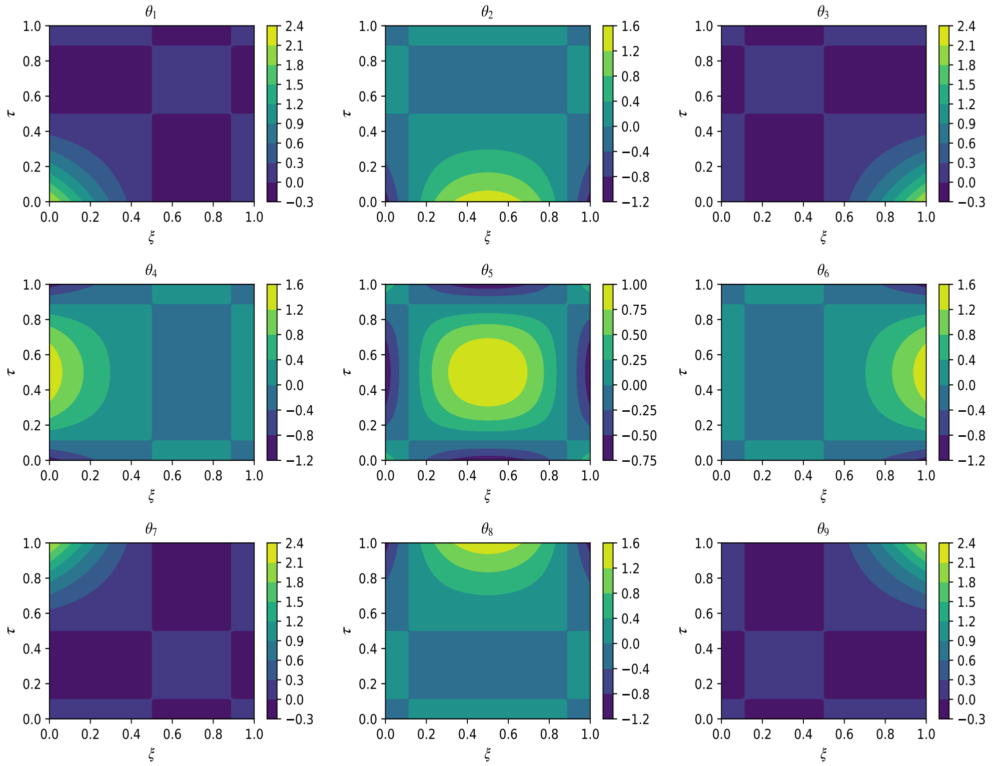


Figure 2.2: Space-time polynomials in $\mathcal{Q}_E$.

For the sake of writing lengthy expressions more simply, we define two operators

$$\langle f, g \rangle = \int_0^1 \int_0^1 f(\xi, \tau) \cdot g(\xi, \tau) d\xi d\tau,$$

$$[f, g]_\tau = \int_0^1 f(\xi, \tau) \cdot g(\xi, \tau) d\xi.$$

The first operator represents the space-time scalar product of two functions, f and g, over the space-time reference elements $\mathcal{Q}_E$. The second operator provides the spatial scalar product of two functions over the spatial reference element $T_E = [0, 1]$ at a fixed τ.

Next, we transform the conservation law problem given in (2.1) to the space-time reference element $\mathcal{Q}_E$ using the relations given by (2.3) as

$$\frac{\partial}{\partial \tau}\boldsymbol{u} + \frac{\partial}{\partial \xi}\boldsymbol{f}^*(\boldsymbol{u}) = 0 \tag{2.4}$$

with

$$\boldsymbol{f}^*(\boldsymbol{u}) = \Delta t \xi_x \boldsymbol{f}(\boldsymbol{u}). \tag{2.5}$$

To get the weak formulation, multiply (2.4) by the space-time dependent test functions θ_k and integrate over the reference element $\mathcal{Q}_E$,

$$\left\langle \theta_k, \frac{\partial}{\partial \tau}\boldsymbol{u} \right\rangle + \left\langle \theta_k, \frac{\partial}{\partial \xi}\boldsymbol{f}^*(\boldsymbol{u}) \right\rangle = 0. \tag{2.6}$$

In each space-time control volume $\mathcal{Q}_i^n$, we approximate the numerical solution $\boldsymbol{\mathcal{U}} = \boldsymbol{\mathcal{U}}(\xi, \tau)$ and the numerical flux term $\boldsymbol{\mathcal{F}} = \boldsymbol{\mathcal{F}}(\xi, \tau)$ of (2.6) using the space-time dependent test functions θ_k defined in $\mathcal{Q}_E$. i.e.

$$\boldsymbol{\mathcal{U}}(\xi, \tau) = \hat{\boldsymbol{\mathcal{U}}}_\ell \theta_\ell(\xi, \tau), \tag{2.7}$$

$$\boldsymbol{\mathcal{F}}(\xi, \tau) = \hat{\boldsymbol{\mathcal{F}}}_\ell \theta_\ell(\xi, \tau). \tag{2.8}$$

Inserting the relations (2.7) and (2.8), into the (2.6) yields

$$\left\langle \theta_k, \frac{\partial}{\partial \tau}\boldsymbol{\mathcal{U}} \right\rangle + \left\langle \theta_k, \frac{\partial}{\partial \xi}\boldsymbol{\mathcal{F}}^* \right\rangle = 0. \tag{2.9}$$

The transformation given in (2.5) can also be used for the degrees of freedom for the numerical flux term, $\hat{\boldsymbol{\mathcal{F}}}_\ell^* = \Delta t \xi_x \hat{\boldsymbol{\mathcal{F}}}_\ell$. Considering the fact that the nodal space-time test functions are governed based on the tensor product Gauss-Legendre quadrature points, we additionally have the relationship between the degrees of freedom of the numerical solution $\hat{\boldsymbol{\mathcal{U}}}_\ell$ and the degrees of freedom of the flux $\hat{\boldsymbol{\mathcal{F}}}_\ell$ as follows [23],

$$\hat{\boldsymbol{\mathcal{F}}}_\ell = \boldsymbol{f}(\hat{\boldsymbol{\mathcal{U}}}_\ell). \tag{2.10}$$

We avoid using integration by parts in space of (2.9) and instead apply integration by parts in time. This approach is consistent with our primary focus of governing the local formulation solely based on the element's information, independent of neighboring data. The following equation is obtained

$$[\theta_k, \boldsymbol{\mathcal{U}}]_1 - [\theta_k, \boldsymbol{u}_h]_0 - \left\langle \frac{\partial}{\partial \tau}\theta_k, \boldsymbol{\mathcal{U}} \right\rangle + \left\langle \theta_k, \frac{\partial}{\partial \xi}\boldsymbol{\mathcal{F}}^* \right\rangle = 0. \tag{2.11}$$

Inserting the relations (2.2), (2.7) and (2.8) into (2.11) we get

$$[\theta_k, \theta_\ell]_1 \hat{\boldsymbol{\mathcal{U}}}_\ell - [\theta_k, \Phi_m]_0 \, \hat{\boldsymbol{u}}_m^n - \left\langle \frac{\partial}{\partial \tau}\theta_k, \theta_\ell \right\rangle \hat{\boldsymbol{\mathcal{U}}}_\ell + \left\langle \theta_k, \frac{\partial}{\partial \xi}\theta_\ell \right\rangle \hat{\boldsymbol{\mathcal{F}}}_\ell^* = 0.$$

Rearranging the terms leads to

$$\left([\theta_k, \theta_\ell]_1 - \left\langle \frac{\partial}{\partial \tau}\theta_k, \theta_\ell \right\rangle \right) \hat{\boldsymbol{\mathcal{U}}}_\ell - [\theta_k, \Phi_m]_0 \, \hat{\boldsymbol{u}}_m^n + \left\langle \theta_k, \frac{\partial}{\partial \xi}\theta_\ell \right\rangle \hat{\boldsymbol{\mathcal{F}}}_\ell^* = 0. \tag{2.12}$$

Defining the following special matrices that are computed once within the space-time reference element $\mathcal{Q}_E$, and then stored for future use optimizes the efficiency of our computations

$$K_1 = [\theta_k, \theta_\ell]_1 - \left\langle \frac{\partial}{\partial \tau}\theta_k, \theta_\ell \right\rangle, \qquad K_\xi = \left\langle \theta_k, \frac{\partial}{\partial \xi}\theta_\ell \right\rangle, \qquad F_0 = [\theta_k, \Phi_m]_0 . \tag{2.13}$$

A more compact expression for (2.12) can be derived by considering the matrices defined in (2.13) as follows,

$$K_1 \hat{\boldsymbol{\mathcal{U}}}_\ell = F_0 \hat{\boldsymbol{u}}_m^n - K_\xi \hat{\boldsymbol{\mathcal{F}}}_\ell^*.$$

To calculate the $\hat{\boldsymbol{\mathcal{U}}}_\ell$, an iteration technique can be used as proposed in [32, 33, 43] with (2.10)

$$K_1 \hat{\boldsymbol{\mathcal{U}}}_\ell^{r+1} = F_0 \hat{\boldsymbol{u}}_m^n - K_\xi \hat{\boldsymbol{\mathcal{F}}}_\ell^{*,r}. \tag{2.14}$$

Numerically, we can obtain the specific result described in [23, 32], where all eigenvalues of the matrix $K_1^{-1}K_\xi$ are zero. It implies that the iteration (2.14) follows a contraction relation. The Banach fixed-point theorem guarantees the existence and uniqueness of the solution $\hat{\boldsymbol{\mathcal{U}}}_\ell$. Once we calculate $\hat{\boldsymbol{\mathcal{U}}}_\ell$ using (2.14), the space-time predictor $\boldsymbol{\mathcal{U}}(\xi,\tau)$ is fully determined for each element from (2.7).

2.1.3 Fully-discrete formulation for the ADER-DG schemes

The local space-time predictor ($\boldsymbol{\mathcal{U}}(\xi,\tau)$) cannot directly update the solution to the next time level t^{n+1}. This limitation arises because, during its construction, we neglect the effect of neighboring elements. To perform the correct update of the solution, we consider the one-step ADER-DG schemes, similar to [14, 23, 31–33, 37, 65]. The procedure is as follows: we multiply the conservation law (2.1) with a spatial test functions Φ_k and integrate over the physical space-time control volume $\mathcal{Q}_i^n = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] \times [t^n, t^n + \Delta t]$. Applying integration by parts in space leads to the following form

$$\int_{t^n}^{t^{n+1}} \int_{T_i} \Phi_k \frac{\partial \boldsymbol{u}_h}{\partial t} dx dt + \int_{t^n}^{t^{n+1}} \int_{\partial T_i} \Phi_k \boldsymbol{f}(\boldsymbol{u}_h) \cdot \boldsymbol{n} dS dt - \int_{t^n}^{t^{n+1}} \int_{T_i} \frac{\partial \Phi_k}{\partial x} \cdot \boldsymbol{f}(\boldsymbol{u}_h) dx dt = 0 \tag{2.15}$$

where, $\boldsymbol{n}$ represents the outward unit normal vector on the surface ∂T_i of the element T_i. The second term in (2.15) establishes the connection between neighboring elements and T_i through surface integration. To calculate the numerical flux function $\mathcal{G}(\boldsymbol{\mathcal{U}}^-,\boldsymbol{\mathcal{U}}^+)$, we solve a Riemann problem at each interface, considering the left and right states denoted by $\boldsymbol{\mathcal{U}}^-$ and $\boldsymbol{\mathcal{U}}^+$ respectively. We employ the Rusanov (local Lax Friedrichs) flux [23, 65] as the numerical flux function at each interface. Furthermore, we replace the physical flux in the third term of (2.15) with the numerical flux term defined in equation (2.8). Inserting the relations given by (2.2), (2.7) and (2.8) into (2.15) we can obtain,

$$[\Phi_k, \Phi_m]\left(\hat{\boldsymbol{u}}_m^{n+1} - \hat{\boldsymbol{u}}_m^n\right) + \frac{\Delta t}{\Delta x}\int_0^1 \int_{\partial T_i} \Phi_k \mathcal{G}(\boldsymbol{\mathcal{U}}^-,\boldsymbol{\mathcal{U}}^+) \cdot \boldsymbol{n} dS d\tau - \left\langle \frac{\partial \Phi_k}{\partial \xi}, \boldsymbol{\mathcal{F}}^* \right\rangle = 0.$$

In this context, we remove the subscript τ of the operator $[.,.]_\tau$, as the two arguments are solely space-dependent test functions. Finally, we construct the higher-order accurate ADER-DG schemes given by

$$[\Phi_k, \Phi_m]\left(\hat{\boldsymbol{u}}_m^{n+1} - \hat{\boldsymbol{u}}_m^n\right) + \frac{\Delta t}{\Delta x}\int_0^1 \int_{\partial T_i} \Phi_k \mathcal{G}(\boldsymbol{\mathcal{U}}^-,\boldsymbol{\mathcal{U}}^+) \cdot \boldsymbol{n} dS d\tau - \left\langle \frac{\partial \Phi_k}{\partial \xi}, \theta_\ell \right\rangle \hat{\boldsymbol{\mathcal{F}}}_\ell^* = 0. \tag{2.16}$$

Using the precomputed matrices defined in (2.13) speeds up computations. The stability of a higher-order ADER-DG scheme can be guaranteed under a suitable Courant-Friedrichs-Lewy (CFL) number. It decreases with the degree N of the test functions, as discussed in [33]. This decrease roughly follows a pattern of $\frac{1}{(2N+1)}$. The time step must fulfill the criteria

$$\Delta t \leq \frac{1}{(2N+1)} \frac{h}{|\lambda_{max}|}.$$

Where h is the characteristic mesh size, and $|\lambda_{max}|$ represents the maximum signal velocity. Further details regarding the stability of the ADER-DG schemes will be discussed in Section 2.3.1. To maintain the correct convergence order, we need to ensure an appropriate order of accuracy for the time integration in (2.15). By achieving this, when we apply degree N spatial test functions, the ADER-DG scheme attains an order of accuracy of $N+1$ in both space and time for smooth solutions. However, special attention is required near discontinuities or sharp gradients, as non-physical oscillations may arise due to the Gibbs phenomenon. The next section introduces so-called limiters, which control these non-physical oscillations.

2.2 MOOD limiter for the ADER-DG schemes

In this section, we focus on limiters designed to tame the Gibbs phenomenon that arises near discontinuities or steep gradients. There is a vast literature concerning limiters, as nonlinear hyperbolic problems can generate discontinuities even when starting with a smooth initial condition. Various reconstruction techniques, such as ENO (Essentially Non-Oscillatory)/WENO (Weighted Essentially Non-Oscillatory) schemes [55, 56], slope limiters (linear reconstruction in MUSCL schemes), and flux limiters [60], are recognized for producing stable outcomes without introducing non-physical oscillations. These limiters are commonly referred to as *a priori* limiters, implying their activation by concerning information from the current time level t^n and produce non-oscillatory results at the next time level t^{n+1}. Throughout this work, we apply a recently developed MOOD (Multi-dimensional Optimal Order Detection) limiter [16, 21, 22, 33, 39, 48] to reduce non-physical oscillations. The MOOD limiter is considered as *a posteriori* limiter, activated based on information from the current time level t^n and the approximation of the solution at the subsequent time level t^{n+1}. When implementing a limiter, we must address the following two crucial questions:

1. In which specific areas is it necessary to apply limiting?
2. How can we minimize non-physical oscillations occurring in the problematic regions while preserving higher-order accuracy elsewhere?

Now, we focus on answering these two questions by describing the MOOD paradigm [33, 39]. First, consider the unlimited ADER-DG numerical solution at t^{n+1} as the *candidate* solution, denoted by $\boldsymbol{u}_h^{n+1,*}$. Then, create a fine subgrid $\mathcal{S}_i = \bigcup_j S_{i,j}$ by dividing each element T_i into $N_s = 2N+1$ equidistant, non-overlapping subcells with the volume $|S_{i,j}|$. At time t^n, the piece-wise constant subcell averages in each element T_i, denoted by $\boldsymbol{v}_{i,j}^n(x, t^n)$, can be calculated considering the L_2 projection $(\mathcal{P})$ of the discrete solution $\boldsymbol{u}_h(x, t^n)$ as

$$\boldsymbol{v}_{i,j}^n(x, t^n) = \frac{1}{|S_{i,j}|} \int_{S_{i,j}} \boldsymbol{u}_h(x, t^n) dx := \mathcal{P}(\boldsymbol{u}_h^n), \qquad \forall j \in [1, N_s]\,. \tag{2.17}$$

Similarly at t^{n+1}, the candidate subcell averages are computed

$$\boldsymbol{v}_{i,j}^{n+1,*}(x,t^{n+1}) = \frac{1}{|S_{i,j}|}\int_{S_{i,j}} \boldsymbol{u}_h^{n+1,*}(x,t^{n+1})dx := \mathcal{P}(\boldsymbol{u}_h^{n+1,*}), \qquad \forall j \in [1, N_s]\,.$$

Since this operator is linear, we compute its associated matrices just once and store them for future use, optimizing efficiency. The identification of *problem cells* involves checking the candidate subcell averages against a set of *detection criteria*, as defined in [33, 39].

- The *physical admissibility detection* criteria (PAD) ensures that the candidate solution provides physically valid results. For instance, in the context of shallow water equations, the height of the water level must remain non-negative. Similarly, when dealing with the Euler equations of compressible gas dynamics, both density and pressure must be positive at all points and times.
- Applying the relaxed discrete maximum principle (DMP) enforces the *numerical admissibility detection* criteria (NAD)

  $$\min_{m\in\nu(T_i)}\left(\min_{\gamma\in[1,N_s]} \boldsymbol{v}_{m,\gamma}^n\right) - \delta \le \boldsymbol{v}_{i,j}^{n+1,*} \le \max_{m\in\nu(T_i)}\left(\max_{\gamma\in[1,N_s]} \boldsymbol{v}_{m,\gamma}^n\right) + \delta, \qquad \forall j \in [1, N_s]\,,$$

  where $\nu(T_i)$ is a set containing element T_i with its neighbors. We use the parameter values as specified in [39]

  $$\delta = \max\left(\delta_0, \epsilon \cdot \left[\max_{m\in\nu(T_i)}\left(\max_{\gamma\in[1,N_s]} \boldsymbol{v}_{m,\gamma}^n\right) - \min_{m\in\nu(T_i)}\left(\min_{\gamma\in[1,N_s]} \boldsymbol{v}_{m,\gamma}^n\right)\right]\right),$$

  with $\delta_0 = 10^{-5}$ and $\epsilon = 10^{-4}$.

If the candidate subcell averages $\boldsymbol{v}_{i,j}^{n+1,*}$ in element T_i do not satisfy the specified detection criteria, then it is marked as a troubled cell. Following this process at each time step allows us to identify the regions where limiting is required. Once we have identified the troubled cells, let's explore methods to eliminate the non-physical oscillations.

One possible approach is to iterate the MOOD algorithm for the problematic cells, gradually decrementing the degrees of the test functions until the candidate subcell averages become valid. For more details, see [21, 22, 48]. In the worst case of this strategy, the first-order (degree zero test functions) solution is acceptable. On the other hand, one could use more accurate ADER-WENO finite volume methods [33, 39] or any other TVD finite volume (FV) methods [60] directly on the problematic cells. To prioritize efficiency and accuracy, we employ a second-order accurate finite volume method with the *minmod* flux limiter scheme to compute $\boldsymbol{v}_{i,j}^{n+1}$ starting from $\boldsymbol{v}_{i,j}^{n}$. We denote it

$$v_{i,j}(x,t^{n+1}) = \Psi(v_{i,j}(x,t^n)).$$

An initial data is necessary to perform the lower-order finite volume update at time level t^{n+1}. Therefore, the previously limited $\boldsymbol{v}_{i,j}^{n+1}$ values are kept in memory to be used in the next time level.

Next, let's highlight a strategy applied to a fine element T_i, which is not marked as a troubled cell but contains at least one problematic cell in its $\nu(T_i)$. In this situation, we can not accept the higher order ADER-DG solution $\boldsymbol{u}_h^{n+1,*}(x,t^{n+1})$ in T_i, as the correct update values to the next time level t^{n+1}. Because the flux calculation across the interface between T_i and the troubled cell follows two different strategies, violating the conservation property of the scheme. To fix this issue, we assign the lower-order flux to that interface. After making these modifications, we need to update cell T_i again with the adjusted fluxes. In rare cases, it is possible to generate troubled cells in this step. Therefore, a loop procedure is imposed to ensure that no problematic cells pass to the next time level t^{n+1} without being treated by the second-order FV method. After successfully treating the troubled cells with a second-order FV method, we can recover the higher-order solution $\boldsymbol{u}_h^{n+1}$, using the corrected subcell averages. This is done by the reconstruction operator defined as follows

$$\int_{S_{i,j}} \boldsymbol{u}_h^{n+1}(x, t^{n+1})dx = \int_{S_{i,j}} \boldsymbol{v}_{i,j}^{n+1}(x, t^{n+1})dx := \mathcal{R}(\boldsymbol{v}_{i,j}^{n+1}(x, t^{n+1})), \qquad \forall j \in [1, N_s], \tag{2.18}$$

which satisfy the conservation property on the main cell T_i under additional linear constraint

$$\int_{T_i} \boldsymbol{u}_h^{n+1}(x, t^{n+1})dx = \int_{T_i} \boldsymbol{v}_{i,j}^{n+1}(x, t^{n+1})dx. \tag{2.19}$$

Here, we can use the least-square technique discussed in [27] to solve the system of equations given in (2.18) and (2.19). In this case, it might be possible to have some oscillations. If so, in the next time level, that cell is captured again as a troubled cell and treated with a second-order finite volume method. Moreover, we can identify that the matrix corresponding to the reconstruction operator in (2.18) is the pseudo-inverse of the matrix corresponding to the projection operator given in (2.17). Hence, these operators satisfy the following property

$$\mathcal{R} \circ \mathcal{P} = \mathcal{I} \tag{2.20}$$

where $\mathcal{I}$ represents the identity operator. Finally, we can combine all these processes to describe the solution for the troubled cell at t^{n+1},

$$\boldsymbol{u}_h^{n+1}(x, t^{n+1}) = \mathcal{R}\big(\Psi(\boldsymbol{v}_{i,j}^{n}(x, t^{n}))\big).$$

The finite volume method applied to troubled cells must satisfy the following stability criteria

$$\Delta t_{FV} = \frac{1}{N_s}\frac{h}{|\lambda_{max}|}$$

where, $\frac{h}{N_s}$ denotes the subcell size. Choosing the number of subcells as $N_s = 2N + 1$ matches the maximum time step size of the finite volume scheme with the maximum time step size of the ADER-DG scheme.

2.3 Results

In this section, we apply the higher-order accurate ADER-DG schemes presented in Section 2.1.3 to some of the standard problems of hyperbolic conservation laws.

2.3.1 Linear scalar problem

As our first test problem, we consider the simple and widely used linear advection equation to validate the numerical method. Using this model equation, we explore additional properties of the higher-order ADER-DG schemes, such as stability analysis and the order of convergence. The linear scalar advection equation, expressed with the simple flux function, takes the following form

$$\begin{gathered}\frac{\partial}{\partial t}u(x,t)+\frac{\partial}{\partial x}f(u(x,t))=0, \qquad f(u(x,t))=au(x,t), \qquad a\in\mathbb{R}, \qquad 0\le x\le 2, \\ u(x,0)=u_0(x).\end{gathered} \tag{2.21}$$

We impose a special initial condition which is originally present in [54], defined in the domain $[-1,1]$ as

$$u_0(x)=\begin{cases}\frac{1}{6}\big(G(x,\beta,z-\delta)+G(x,\beta,z+\delta)+4G(x,\beta,z)\big), & -0.8\le x\le -0.6;\\ 1, & -0.4\le x\le -0.2;\\ 1-|10(x-0.1)|, & 0\le x\le 0.2;\\ \frac{1}{6}\big(F(x,\alpha,\tilde{a}-\delta)+F(x,\alpha,\tilde{a}+\delta)+4F(x,\alpha,\tilde{a})\big), & 0.4\le x\le 0.6;\\ 0, & \text{otherwise.}\end{cases} \tag{2.22}$$

Where $G(x,\beta,z)=\exp(-\beta(x-z)^2)$ and $F(x,\alpha,\tilde{a})=\sqrt{\max(1-\alpha^2(x-\tilde{a})^2,0)}$. The constants are taken as $\tilde{a}=0.5$, $z=-0.7$, $\delta=0.005$, $\alpha=10$, and $\beta=\log(2)/36\delta^2$.

To ensure compatibility with the domain of the model given in (2.21), we transform the initial condition's domain from $[-1,1]$ to $[0,2]$. The wave speed is set to one ($a=1$). Figure 2.3 represents the results for different numbers of elements. The numerical solution obtained using the 6^{th}-order unlimited ADER-DG scheme with 34 elements accurately captures the exact solution in smooth regions. We intentionally choose 34 elements for the computational domain to match the total number of degrees of freedom to approximately 200, following the approach outlined in [54]. The 6^{th}-order unlimited ADER-DG scheme generates non-physical oscillations near the discontinuities. When the MOOD limiter is applied with a smaller number of elements, it captures a relatively higher number of troubled cells. As a result, the overall scheme behaves more closely to a second-order scheme. Increasing the number of elements in the domain overcomes this issue. Finally, the results obtained for 200 elements demonstrate that the ADER-DG scheme with the MOOD limiter effectively suppresses the oscillations and precisely captures the exact solution.

Linear stability and convergence study

Let's shift our focus to the stability of the ADER-DG schemes presented in Section 2.1.3. There, we briefly discuss the stability requirements by addressing the choice of the time step according to the CFL number. Now, we aim to explore the stability in more detail. Our first objective is to express the full discretization presented in (2.16), in the form of a linear recursion

$$\hat{\boldsymbol{u}}^{n+1}=\mathbb{A}\hat{\boldsymbol{u}}^{n}. \tag{2.23}$$

Here, $\hat{\boldsymbol{u}}$ represents a vector, including all the degrees of freedom in the domain. Analytically, converting the full ADER-DG discretization into this form is a challenging task. Alternatively, with the help of linear algebra technique the matrix $\mathbb{A}$ is constructed. Assume our numerical method (ADER-DG solver) $\mathcal{H}$ provides an update for the degrees of freedoms such that

$$\hat{\boldsymbol{u}}^{n+1}=\mathcal{H}(\hat{\boldsymbol{u}}^{n}).$$

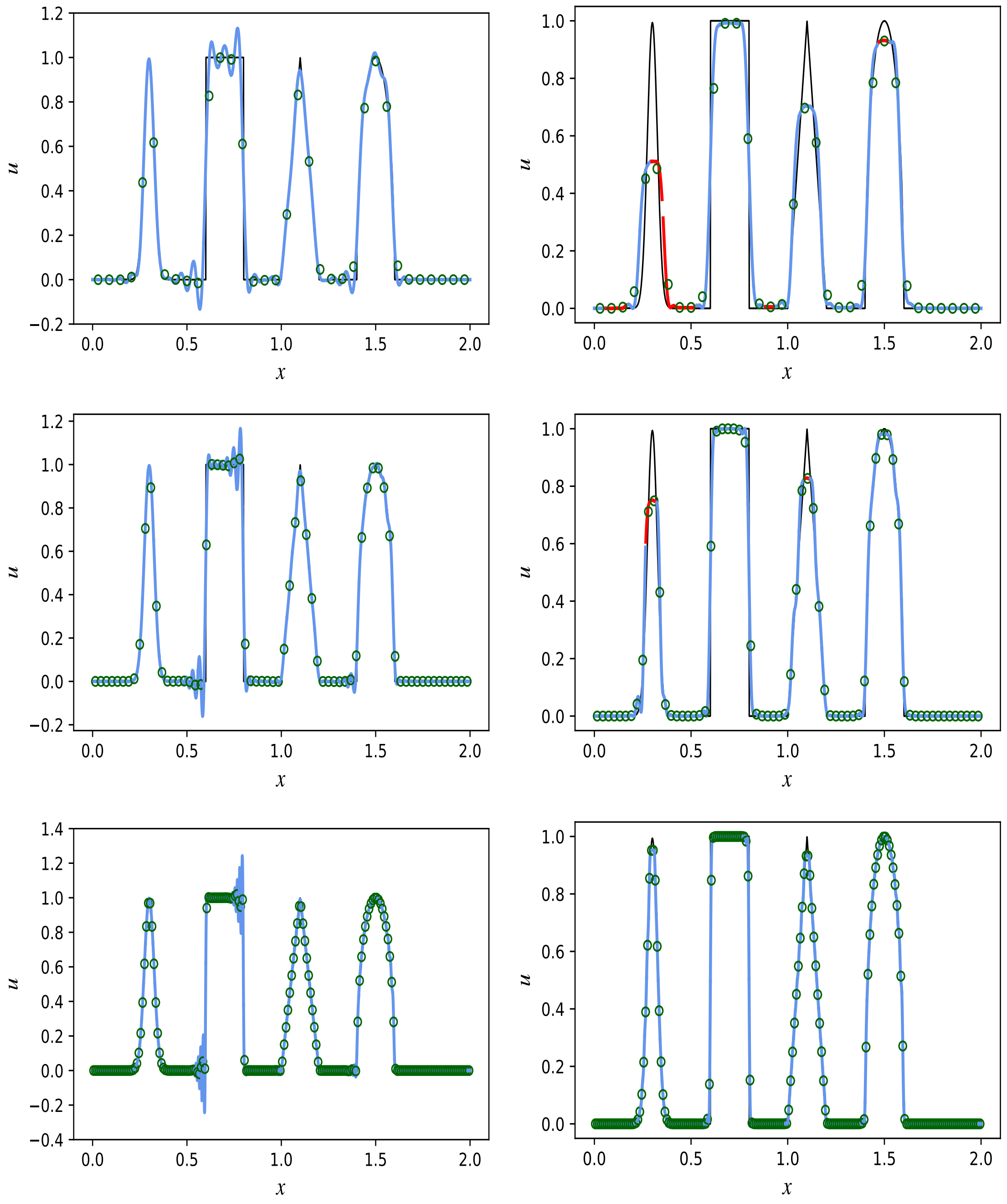


Figure 2.3: The final time is set as $T = 8$. The blue lines (—) represent the 6^{th}-order ADER-DG solution, red lines (—) highlight troubled cells, green circles (○) mark the cell averages of the conserved variable and the exact solution shows in black line (—). The left column of figures illustrates the results obtained with the unlimited ADER-DG scheme, with configurations of 34 elements (*top*), 68 elements (*middle*), and 200 elements (*bottom*). Similarly, the right column of figures demonstrates outcomes of the ADER-DG scheme with MOOD limiter, corresponding to the same number of elements.

In general, we have the freedom to choose the initial conditions for the problem. Therefore, we used a set of standard basis vectors, $e_i = (0, 0, \ldots, 1, \ldots, 0, 0)^T$ as initial conditions, where the components are all zeros, except for one that is equal to one. For a single time step, j^{th} column of the iterative matrix $\mathbb{A}$, represented by $\mathbb{A}_j$, can be obtained from the corresponding basis vector e_j such that, $\mathcal{H}(e_j) = \mathbb{A}e_j = \mathbb{A}_j$. After progressing through all the standard basis vectors, we end up with the complete iteration matrix $\mathbb{A}$. Subsequently, we introduce key terminology and state the theorem regarding the stability of the general linear recursion. For more details, please refer to [20].

Definition 2.1 Consider a matrix $\mathbb{A} \in \mathbb{C}^{d\times d}$.

- The **spectrum** of a matrix $\mathbb{A}$ is the set of all eigenvalues of $\mathbb{A}$

$$\sigma(\mathbb{A}) = \{\lambda \in \mathbb{C} : \det(\lambda \mathbf{I} - \mathbb{A}) = 0\}\,.$$

- The **index** $i(\lambda)$ of an eigenvalue $\lambda \in \sigma(\mathbb{A})$ is the maximal dimension of the Jordan blocks of $\mathbb{A}$ containing λ.
- The **spectral radius** of the matrix $\mathbb{A}$

$$\rho(\mathbb{A}) = \max_{\lambda\in\sigma(\mathbb{A})} |\lambda|.$$

Theorem 2.2 — Stability of the linear recursion. The linear recursion $x^{n+1} = \mathbb{A}x^n$ with $\mathbb{A} \in \mathbb{C}^{d\times d}$ is **stable** if and only if the spectral radius satisfies $\rho(\mathbb{A}) \le 1$ and all eigenvalues $\lambda \in \sigma(\mathbb{A})$ with $|\lambda| = 1$ have index $i(\lambda) = 1$. It is asymptotically stable exactly if $\rho(\mathbb{A}) < 1$ holds for the spectral radius.

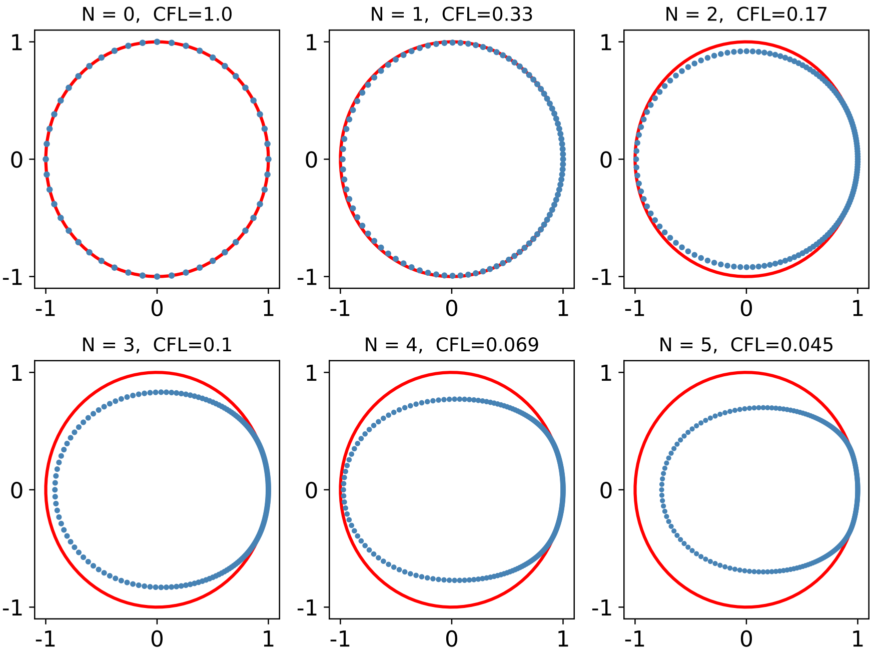


Figure 2.4: The spectrum of the iteration matrix $\mathbb{A}$ corresponds to the degree of the space-dependent test functions and optimal CFL numbers.

We are examining the spectrum of the matrix $\mathbb{A}$ under periodic boundary conditions. For each ADER-DG solver, the CFL number is selected based on the degree of the space-dependent test functions similar to [23]. Figure 2.4 illustrates the spectrum of the iterative matrix $\mathbb{A}$ and its correlation with the optimal CFL number for the respective degree of the space-dependent test functions. There, the eigenvalues are represented by blue dots (●), and the reference unit circle is shown in red. These carefully chosen CFL numbers guarantee the stability of the numerical scheme by ensuring that the spectrum radius does not exceed 1.

To investigate the convergence behaviour of the schemes, we begin with a smooth initial condition

$$u(x, 0) = \sin(2\pi x).$$

ADER-DG	N_E	L^1	$\mathcal{O}_{L^1}$	L^2	$\mathcal{O}_{L^2}$	L^∞	$\mathcal{O}_{L^\infty}$
$\mathcal{O}2$ $(N = 1)$	10	1.1230E-01	-	8.8790E-02	-	8.8189E-02	-
	15	5.0416E-02	1.97	3.9564E-02	1.99	3.9606E-02	1.97
	20	2.8495E-02	1.98	2.2295E-02	1.99	2.2124E-02	2.02
	25	1.8185E-02	2.01	1.4283E-02	1.99	1.4327E-02	1.94
	30	1.2626E-02	2.00	9.9198E-03	1.99	9.9222E-03	2.01
$\mathcal{O}3$ $(N = 2)$	10	6.9694E-03	-	4.4437E-03	-	3.8638E-03	-
	15	2.0357E-03	3.03	1.3062E-03	3.01	1.1298E-03	3.03
	20	8.5766E-04	3.00	5.4980E-04	3.00	4.7346E-04	3.02
	25	4.3816E-04	3.00	2.8126E-04	3.00	2.4134E-04	3.01
	30	2.5363E-04	2.99	1.6268E-04	3.00	1.3914E-04	3.02
$\mathcal{O}4$ $(N = 3)$	10	1.9648E-04	-	1.1115E-04	-	9.7024E-05	-
	15	3.9190E-05	3.97	2.1910E-05	4.00	1.8046E-05	4.14
	20	1.2452E-05	3.98	6.9296E-06	4.00	5.4156E-06	4.18
	25	5.1370E-06	3.96	2.8619E-06	3.96	2.2968E-06	3.84
	30	2.4858E-06	3.98	1.3854E-06	3.97	1.1005E-06	4.03
$\mathcal{O}5$ $(N = 4)$	10	7.1048E-06	-	3.5316E-06	-	2.3695E-06	-
	15	9.4626E-07	4.97	4.7121E-07	4.96	3.2199E-07	4.92
	20	2.2543E-07	4.98	1.1213E-07	4.99	7.4454E-08	5.09
	25	7.3749E-08	5.00	3.6699E-08	5.00	2.4646E-08	4.95
	30	2.9841E-08	4.96	1.4849E-08	4.96	9.8680E-09	5.02
$\mathcal{O}6$ $(N = 5)$	10	1.4376E-07	-	6.6963E-08	-	5.1882E-08	-
	15	1.2841E-08	5.95	5.9371E-09	5.97	4.4743E-09	6.04
	20	2.3081E-09	5.96	1.0597E-09	5.98	7.6217E-10	6.15
	25	6.0756E-10	5.98	2.7779E-10	6.00	1.9242E-10	6.16
	30	2.0411E-10	5.98	9.3326E-11	5.98	6.4608E-11	5.98

Table 2.1: Numerical convergence rates for the advection equation with ADER-DG schemes evaluated from second to sixth order of accuracy in both space and time.

We then refine the mesh from 10 to 30 elements and proceed to compute the numerical solution with the final time set to $T = 2$. The errors and convergence rates in L^1,L^2 and L^∞

norms are presented in table Table 2.1. The computation of error norms follows the expressions discussed in [47]

$$||u - u_h||_p = \left(\int_0^1 |u(x,t) - u_h(x,t)|^p dx \right)^{\frac{1}{p}},$$

where $u(x,t)$ and $u_h(x,t)$ represent the exact (reference) and the discrete solution respectively.

2.3.2 Linear system

We use linear acoustic equations detailed in [47] to validate our numerical scheme for a linear system. Let us consider the general linear hyperbolic system as

$$\frac{\partial}{\partial t}\boldsymbol{u} + A\frac{\partial}{\partial x}\boldsymbol{u} = 0,$$

with

$$\boldsymbol{u}(x,t) = \begin{bmatrix} p(x,t) \\ v(x,t) \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & K_0 \\ \frac{1}{\rho_0} & 0 \end{bmatrix}.$$

We set the constants $K_0 = \rho_0 = 1.4$. The periodic boundary conditions are imposed for the simulation. The initial condition is selected similarly to [42], with the domain $x \in [-1, 1]$.

$$\begin{aligned} p(x,0) &= \exp\big(-100x^2\big)\sin\big(80(x-0.5)\big) \\ v(x,0) &= 0. \end{aligned} \tag{2.24}$$

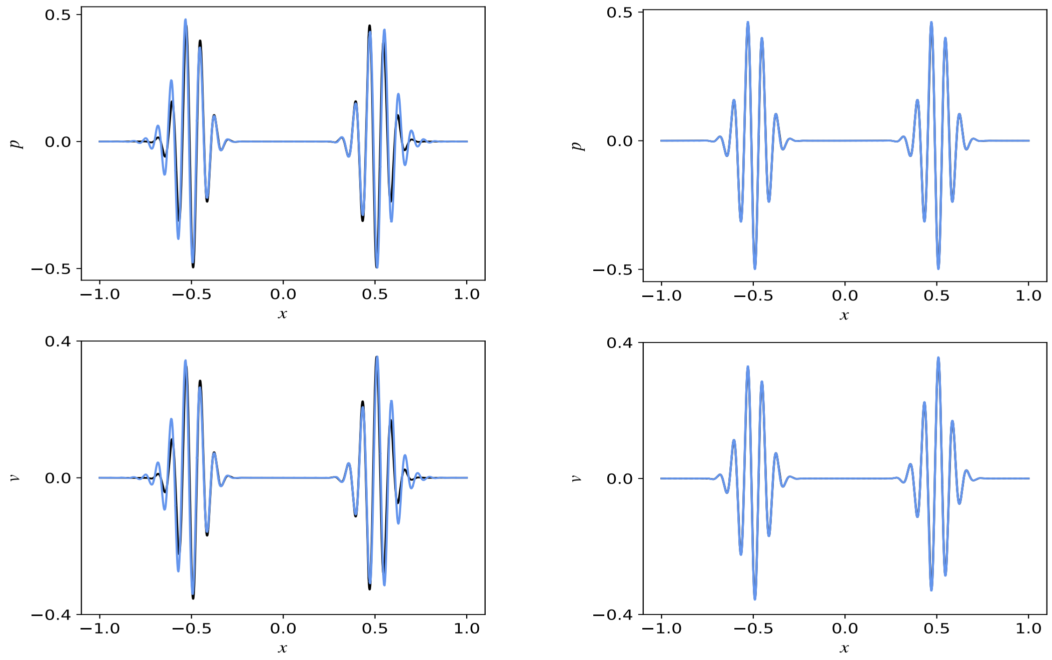


Figure 2.5: The results are presented at the final time $T = 7.5$. The figures on the left show the results with 34 elements, while the figures on the right represent the results with 68 elements. The blue lines (——) depict the 6^{th}-order ADER-DG solutions and the black line (——) represents the exact solution.

The variables p and v represent pressure and velocity, respectively. The constant K_0 is referred to as the *bulk modulus of compressibility* of the material, and $\frac{1}{\rho_0}$ represents constant density. In acoustics, the *impedance* of the medium is denoted by Z_0, where $Z_0 = \rho_0 c_0$ and c_0 is the speed of sound defined as $c_0 = \sqrt{K_0/\rho_0}$. Considering these parameters, we can express the exact solution in a simplified form. After starting with the initial condition,

$$\boldsymbol{u}(x,0) = \begin{bmatrix} \overset{\circ}{p} \\ \overset{\circ}{v} \end{bmatrix}$$

we can express the exact solution as

$$\begin{aligned} p(x,t) &= \frac{1}{2}\left[\overset{\circ}{p}(x+c_0t) + \overset{\circ}{p}(x-c_0t)\right] - \frac{Z_0}{2}\left[\overset{\circ}{v}(x+c_0t) - \overset{\circ}{v}(x-c_0t)\right], \\ v(x,t) &= -\frac{1}{2Z_0}\left[\overset{\circ}{p}(x+c_0t) - \overset{\circ}{p}(x-c_0t)\right] + \frac{1}{2}\left[\overset{\circ}{v}(x+c_0t) + \overset{\circ}{v}(x-c_0t)\right]. \end{aligned}$$

Figure 2.5 illustrates the 6^{th}-order ADER-DG scheme solution for 34 elements and 68 elements. Note that the scheme with 34 elements has approximately 200 degrees of freedom. Doubling the number of elements significantly enhances the accuracy of the solution and ensures the exact solution is captured. Since the MOOD limiter relies on the discrete maximum principle, it identifies smooth extrema as troubled cells. Therefore, we do not apply the MOOD limiter for this example. Although it is possible to incorporate a smooth extrema detector, as discussed in [21], this approach is computationally costly because it relies on curvature-based detection criteria.

2.3.3 Non-linear scalar problem

Here, we are examining the most straightforward non-linear scalar conservation law problem, namely inviscid Burgers' equation as our next test problem. The appropriate initial condition aligns with the insights presented in [2]

$$\begin{gathered} \frac{\partial}{\partial t}u + u\frac{\partial}{\partial x}u = 0, \qquad x \in X \in \mathbb{R}, \quad ,t \geq 0, \\ u(x,0) = 2.5\exp\Big(-\frac{(x-0.5)^2}{0.1^2}\Big) - 0.2. \end{gathered}$$

Here, $u : X \times \mathbb{R}^+ \to \mathbb{R}$. At the final time T=0.1, even with a smooth (Gaussian) initial condition for Burgers' equation, discontinuities naturally emerge. This phenomenon is illustrated in Figure 2.6. The 6^{th}-order ADER-DG scheme can sharply capture the discontinuity even with 34 elements. The results show the effect of the MOOD limiter, which removes oscillations near the discontinuity by considering only a few troubled cells. When the domain is equipped with 100 elements, there is a slight improvement in capturing the exact solution at the discontinuity.

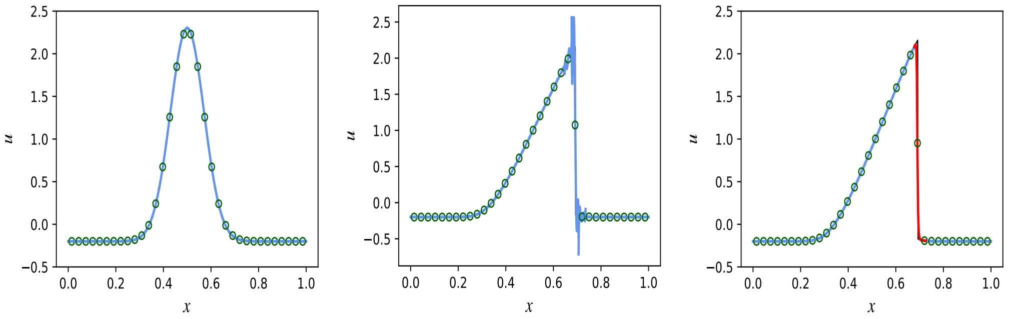
For the convergence study, we adopt the initial condition used in [2]

$$u(x,0) = 0.8 + \exp\Big(-\frac{(x-0.5)^2}{0.05^2}\Big).$$

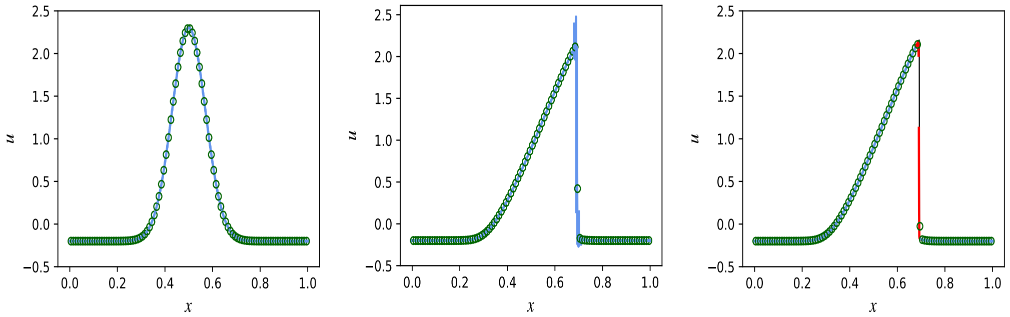
The errors and convergence rates are presented in Table 2.2. The simulation is run for a short time period, with the final time set at $T = 0.01$, ensuring the absence of any discontinuity.

ADER-DG	N_E	L^1	$\mathcal{O}_{L^1}$	L^2	$\mathcal{O}_{L^2}$	L^∞	$\mathcal{O}_{L^\infty}$
$\mathcal{O}2$ $(N=1)$	16	6.8269E-02	-	7.1878E-02	-	2.4041E-01	-
	32	1.7781E-02	1.94	1.8977E-02	1.92	6.0052E-02	2.00
	64	4.7072E-03	1.91	5.2699E-03	1.84	2.1628E-02	1.47
	128	1.2147E-03	1.95	1.3942E-03	1.91	6.6041E-03	1.71
	256	3.1083E-04	1.96	3.6131E-04	1.94	1.6402E-03	2.00
$\mathcal{O}3$ $(N=2)$	16	2.0305E-02	-	2.0643E-02	-	4.9406E-02	-
	32	2.9214E-03	2.79	3.4023E-03	2.60	1.6742E-02	1.56
	64	3.5004E-04	3.06	4.1333E-04	3.04	1.5827E-03	3.40
	128	4.4039E-05	2.99	5.3553E-05	2.94	2.7501E-04	2.52
	256	5.6608E-06	2.95	6.8580E-06	2.96	3.4984E-05	2.97
$\mathcal{O}4$ $(N=3)$	16	5.0856E-03	-	6.2232E-03	-	1.6044E-02	-
	32	2.9457E-04	4.10	3.0853E-04	4.33	1.0155E-03	3.98
	64	2.2633E-05	3.70	2.8551E-05	3.43	9.8535E-05	3.36
	128	1.4865E-06	3.92	1.9869E-06	3.84	9.3929E-06	3.39
	256	9.8170E-08	3.92	1.3259E-07	3.90	6.3808E-07	3.87
$\mathcal{O}5$ $(N=4)$	16	1.3970E-03	-	1.4977E-03	-	3.3695E-03	-
	32	7.2854E-05	4.26	9.6857E-05	3.95	2.8161E-04	3.58
	64	1.7814E-06	5.35	2.7504E-06	5.13	1.0285E-05	4.77
	128	5.4641E-08	5.02	8.9219E-08	4.94	4.4367E-07	4.53
	256	1.6443E-09	5.05	2.6628E-09	5.06	1.4880E-08	4.89
$\mathcal{O}6$ $(N=5)$	16	4.9065E-04	-	6.2543E-04	-	1.6424E-03	-
	32	8.4737E-06	5.85	1.2074E-05	5.69	4.8502E-05	5.08
	64	1.2979E-07	6.02	2.2540E-07	5.74	1.2979E-06	5.22
	128	1.9261E-09	6.07	3.2389E-09	6.12	1.6241E-08	6.32
	256	4.7120E-11	5.35	5.8295E-11	5.79	2.9072E-10	5.80

Table 2.2: Numerical convergence rates for the Burgers' equation with ADER-DG schemes evaluated from second to sixth order of accuracy in both space and time. The reference solution considers the 7^{th}-order ADER-DG solution with 1024 cells.



Situation A



Situation B

Figure 2.6: Situation A consists of 34 elements, while Situation B consists of 100 elements for the computational domain. Numerical solutions to the Burgers' equation: the left figure represents the initial condition, the middle figure depicts the solution with the unlimited 6^{th}-order ADER-DG scheme, and the right figure shows the solution using the 6^{th}-order ADER-DG scheme with MOOD limiter. The blue lines (——) represent the 6^{th}-order ADER-DG solution, red lines (——) highlight troubled cells, green circles (○) mark the cell averages, and the black line (——) represents the reference solution using a first-order upwind scheme with 4000 cells.

2.3.4 Non-linear system

Many physical problems are governed by systems of hyperbolic conservation laws, which are often represented according to the principles of the well-known shallow water model. Here, we consider the one-dimensional shallow water system of the form

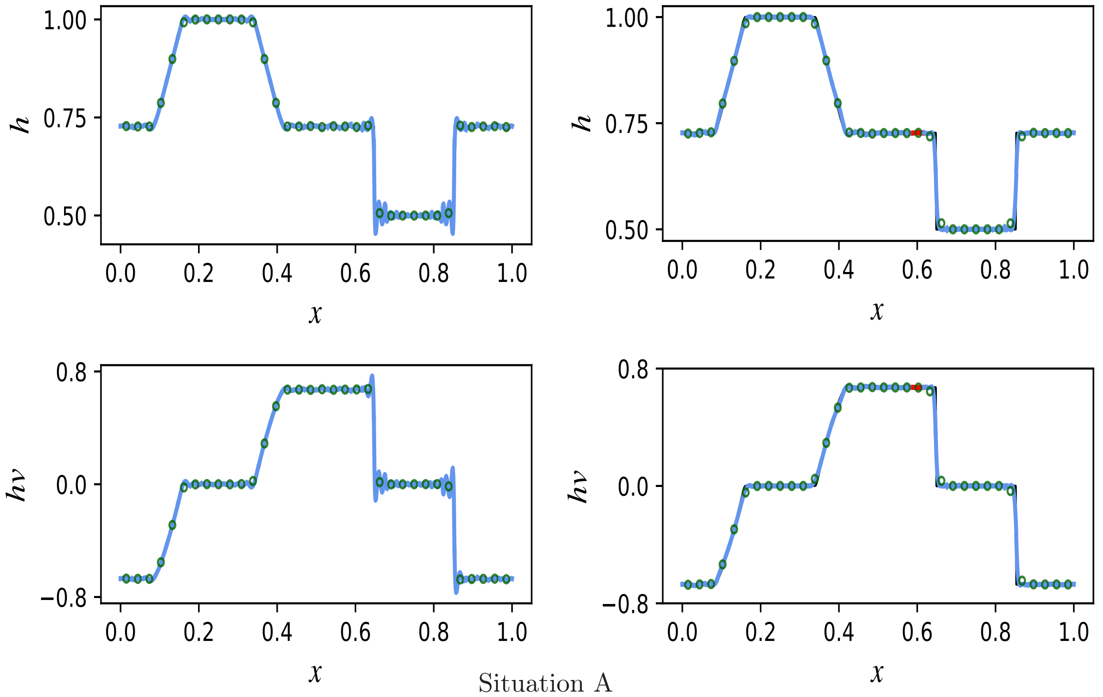
$$\begin{pmatrix} h \\ hv \end{pmatrix}_t + \begin{pmatrix} hv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}_x = 0$$

where h represents the height of the water surface, v indicates the velocity of the water flow in the x direction and g is the gravitational acceleration. The initial condition is given as follows

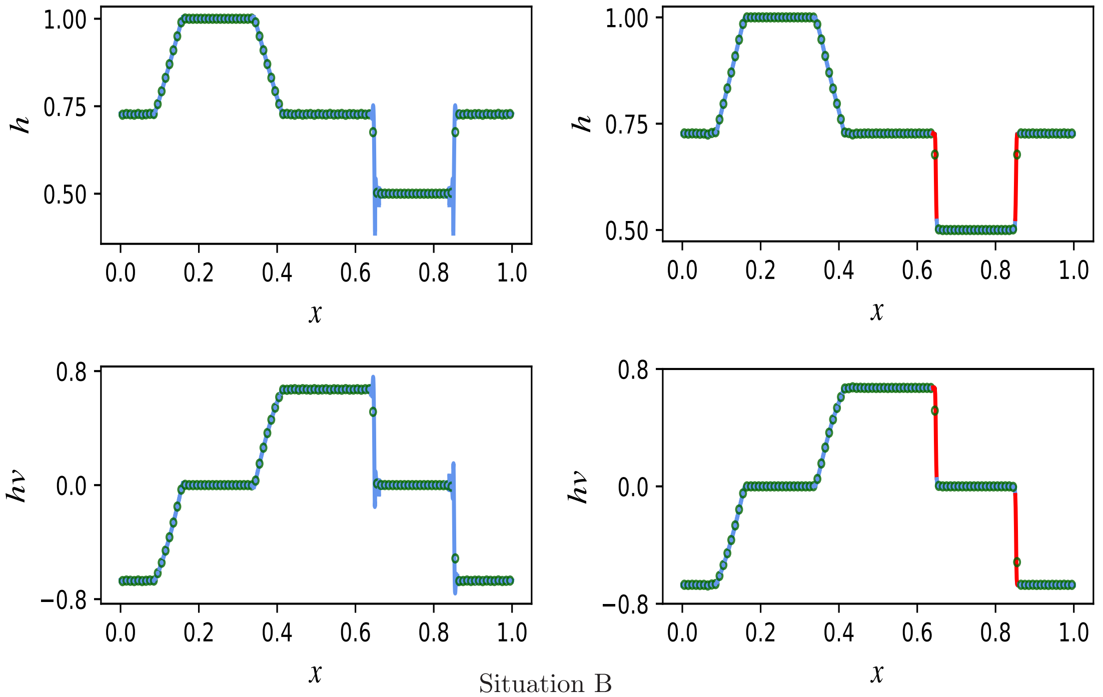
$$h(x,0) = \begin{cases} h_0 = 1 & x < 0.5 \\ h_1 = 0.5 & x > 0.5 \end{cases}, \qquad v(x,0) = 0.$$

For a comprehensive exploration of theoretical and numerical aspects, please refer [61]. The simulation applies periodic boundary conditions. We are examining two scenarios.

First, the 6^{th}-order ADER-DG scheme is applied using 34 elements within the domain $x \in [0,1]$. Even with less number of elements, it accurately captures the exact solution. Next, the ADER-DG scheme is implemented with 100 elements. In both scenarios, the unlimited numerical solution exhibits oscillations near the discontinuities. However, the application of the MOOD limiter effectively eliminates these oscillations, resulting in sharper outcomes. Figure 2.7 represents the results.



Situation A



Situation B

Figure 2.7: The numerical results for the shallow water system are presented at the final time $T = 0.05$. Situation A consists of 34 elements while Situation B consists of 100 elements. In each situation, the left figures show the unlimited ADER-DG solutions and the right figures represent the solutions with MOOD limiter. The blue lines (—) represent the 6^{th}-order ADER-DG solution, red lines (—) emphasize troubled cells, and green circles (○) indicate the cell averages. The black line (—) represents the reference solution (Exact Riemann solver) with 5000 cells.

The convergence study follows a similar approach discussed in Section 2.3.1 and Section 2.3.3 with the smooth initial condition

$$
\begin{aligned}
h(x,0) &= 0.2\sin(2\pi x) + 0.5,\\
v(x,0) &= 0.
\end{aligned}
$$

We compute the errors and convergence rates for the first component (h) of the system at time $T = 0.05$ and summarize the results in the Table 2.3. The numerical schemes provide expected convergence orders.

ADER-DG	N_E	L^1	$\mathcal{O}_{L^1}$	L^2	$\mathcal{O}_{L^2}$	L^∞	$\mathcal{O}_{L^\infty}$
$\mathcal{O}2$ $(N=1)$	8	1.8317E-02	-	8.3374E-03	-	1.0282E-02	-
	16	3.9714E-03	2.20	1.8925E-03	2.13	2.6850E-03	1.93
	32	1.0124E-03	1.97	4.8044E-04	1.97	7.2463E-04	1.88
	64	2.6478E-04	1.93	1.248E-04	1.94	1.8340E-04	1.98
	128	6.5845E-05	2.00	3.1618E-05	1.98	4.4996E-05	2.02
$\mathcal{O}3$ $(N=2)$	8	1.3721E-03	-	6.8095E-04	-	1.1589E-03	-
	16	1.8458E-04	2.89	8.6319E-05	2.97	8.3208E-05	3.79
	32	2.2274E-05	3.05	1.0481E-05	3.04	1.1335E-05	2.87
	64	2.8307E-06	2.97	1.3267E-06	2.98	1.3565E-06	3.06
	128	3.5822E-07	2.98	1.7040E-07	2.96	1.5819E-07	3.10
$\mathcal{O}4$ $(N=3)$	8	1.1659E-04	-	5.4632E-05	-	5.0894E-05	-
	16	6.3229E-06	4.20	3.1670E-06	4.10	4.8931E-06	3.37
	32	3.8148E-07	4.05	1.9928E-07	3.99	3.6861E-07	3.73
	64	2.3666E-08	4.01	1.2591E-08	3.98	2.5182E-08	3.87
	128	1.5425E-09	3.93	8.2474E-10	3.93	1.6088E-09	3.96
$\mathcal{O}5$ $(N=4)$	4	4.1973E-04	-	1.8409E-04	-	1.3115E-04	-
	8	9.5250E-06	5.46	5.3532E-06	5.10	1.3726E-05	3.25
	16	3.2398E-07	4.87	1.9951E-07	4.74	4.7506E-07	4.85
	32	9.9311E-09	5.02	6.1107E-09	5.02	1.5423E-08	4.94
	64	3.1412E-10	4.98	1.8814E-10	5.02	4.0571E-10	5.24
$\mathcal{O}6$ $(N=5)$	4	5.1513E-05	-	3.2086E-05	-	8.7117E-05	-
	8	8.3077E-07	5.95	4.8777E-07	6.03	8.7476E-07	6.63
	16	9.6000E-09	6.43	5.8023E-09	6.39	1.4893E-08	5.87
	32	1.5602E-10	5.94	9.7092E-11	5.90	3.3294E-10	5.48

Table 2.3: Numerical convergence rates for the height (h) of the shallow water equations with ADER-DG schemes evaluated from second to sixth order of accuracy in both space and time. The reference solution is computed using the 7^{th}-order ADER-DG scheme with 512 cells.

2.3.5 Network

Here, we apply the higher-order accurate ADER-DG solver to solve one-dimensional linear scalar conservation law problems within a network. The network's structure is explained in Figure 2.8 as a directed graph, denoted by $\mathcal{G} = \left(\mathcal{V}, \mathcal{E}\right)$ [6]. The symbols $\mathcal{V}$ and $\mathcal{E}$ represent the sets of nodes and edges, respectively.

For each edge $e \in \mathcal{E}$, we use a spatially equidistant grid with cells $\left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]$, where $i \in \mathbb{Z}$, and Δx represents the size of the grid cells. In this example, we solve the advection equation by considering the direction of the edges shown in Figure 2.8. We provide all the parameters and necessary relations in Example 2.3.

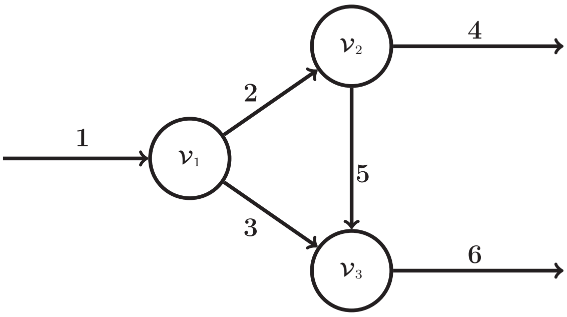


Figure 2.8: Network with three nodes and six edges.

■ **Example 2.3** Solve,

$$\begin{aligned} &\frac{\partial}{\partial t}u_e + \frac{\partial}{\partial x}f(u_e) = 0 \quad \text{with} \quad f(u_e) = a_e u_e \quad \forall e \in \{1,2,\ldots,6\} = \mathcal{E}, \;\; a_e \in \mathbb{R} \\ &\bullet(\text{IC}) \;\; u_e(0,x) = 0, \quad \forall e \in \{1,2,\ldots,6\} \\ &\bullet(\text{BC}) \;\; u_1(t,0) = \sin(\Omega t) \quad \text{with} \quad \Omega = \frac{2\pi}{3}. \end{aligned} \tag{2.25}$$

The lengths of the edges are

$$\ell_1 = 5, \qquad \ell_2 = \ell_3 = \ell_5 = 20, \qquad \ell_4 = \ell_6 = 30.$$

The advection velocities are

$$a_1 = a_3 = a_4 = a_6 = 1, \qquad a_2 = 2, \qquad a_5 = \frac{40}{23}.$$

These coupling conditions are used at the three nodes

$$\text{at } \mathcal{V}_1, \begin{cases} f\big(u_2(t,0)\big) = \alpha_1 f\big(u_1(t,\ell_1)\big) \\ f\big(u_3(t,0)\big) = (1-\alpha_1) f\big(u_1(t,\ell_1)\big) \end{cases} \qquad \text{at } \mathcal{V}_2, \begin{cases} f\big(u_4(t,0)\big) = \alpha_2 f\big(u_2(t,\ell_2)\big) \\ f\big(u_5(t,0)\big) = (1-\alpha_2) f\big(u_2(t,\ell_2)\big) \end{cases}$$

$$\text{at } \mathcal{V}_3, \left\{ f\big(u_6(t,0)\big) = f\big(u_3(t,\ell_3)\big) + f\big(u_5(t,\ell_5)\big), \right. \quad \text{the parametes } \alpha_1 = \frac{3}{4} \text{ and } \alpha_2 = \frac{2}{3}.$$

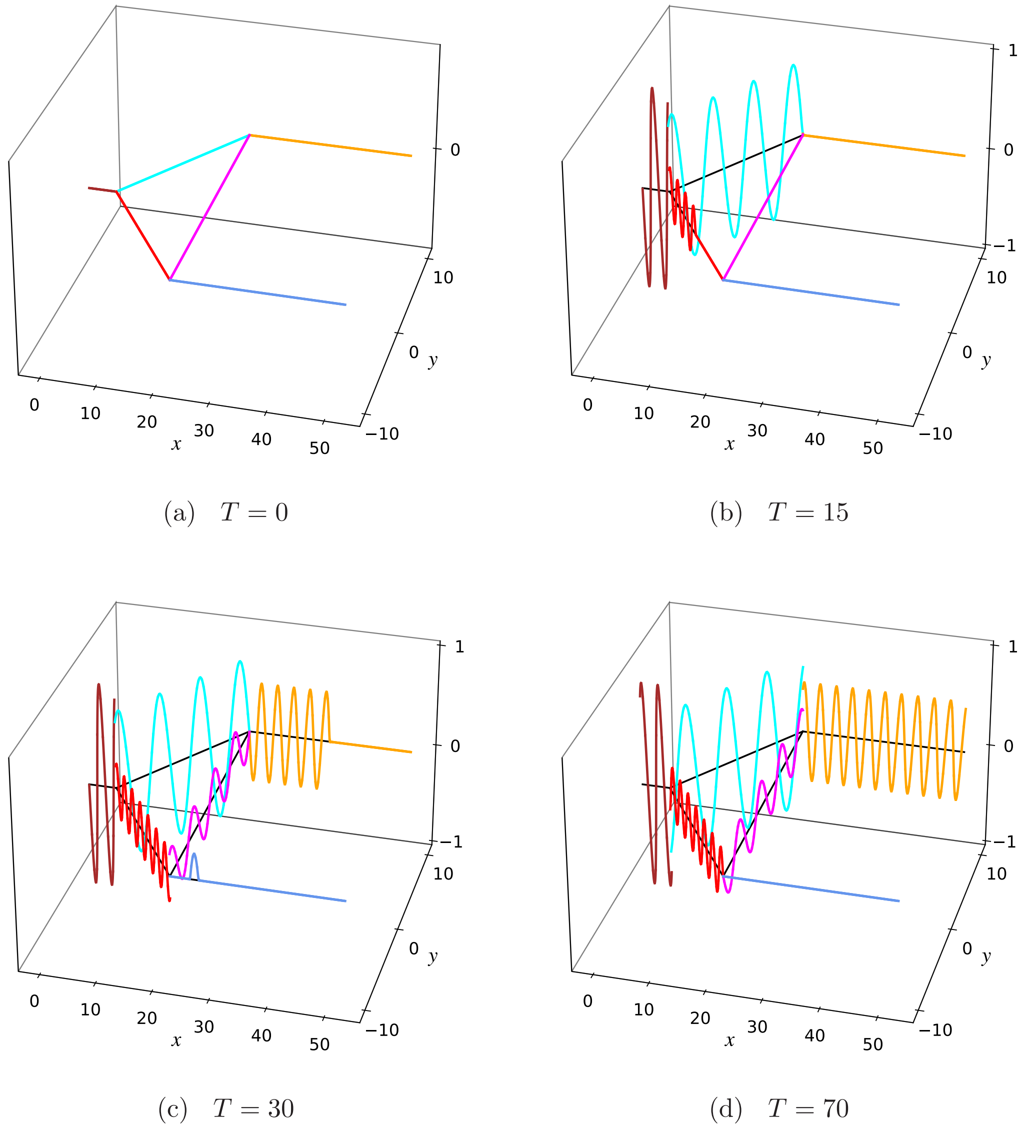
According to Example 2.3, we formulate the initial-boundary value problem for the network as shown in (2.25). Initially, we set the entire network to a zero state, with a Dirichlet-type boundary condition imposed at the left end of the first edge. The objective is to generate sinusoidal signals at that boundary and propagate them through the network as time evolves. Nodes $\mathcal{V}1$ and $\mathcal{V}2$ exhibit similar behavior, each dividing the incoming signal into two parts using the fixed parameters α_1 and α_2, respectively. Velocities are assigned to induce destructive interference at node $\mathcal{V}_3$, resulting in the exact solution on edge 6 being zero.

Next, let's focus on determining the appropriate fixed parameters. Given that we know the exact solution for the advection equation, it becomes straightforward to express the solution at nodes $\mathcal{V}3$ for sufficiently large time t. We define the time taken for a signal to pass each edge e as $\tau_e := \frac{\ell_e}{a_e}$. Using that fact, one can write the exact solution for the edge 6 as follows:

$$u_6(t,0) = \alpha_1(1-\alpha_2)\sin\big(\Omega(t-\tau_1-\tau_2-\tau_5-\tau_6)\big) + (1-\alpha_1)\sin\big(\Omega(t-\tau_1-\tau_3-\tau_6)\big).$$

The signals must have the same amplitude in opposite directions to achieve destructive interference. Therefore, the parameters α_1, α_2, and Ω must satisfy the following conditions

$$\alpha_1(1-\alpha_2) = 1-\alpha_1,$$
$$\Omega(\tau_2+\tau_5) = \Omega\tau_3 + \pi.$$



(a) $T = 0$ (b) $T = 15$

(c) $T = 30$ (d) $T = 70$

Figure 2.9: Network simulation results using 5^{th}-order ADER-DG scheme.

The numerical simulations run using the fifth order accurate ADER-DG scheme with different cell sizes $\Delta x = \frac{1}{2}$ and $\frac{1}{4}$. In this scenario, we do not activate any limiter. This example was proposed in [6], as an illustration of the implicit Active Flux approach. With the highly accurate ADER-DG scheme, we can achieve excellent accuracy without the need for significant mesh refinement.

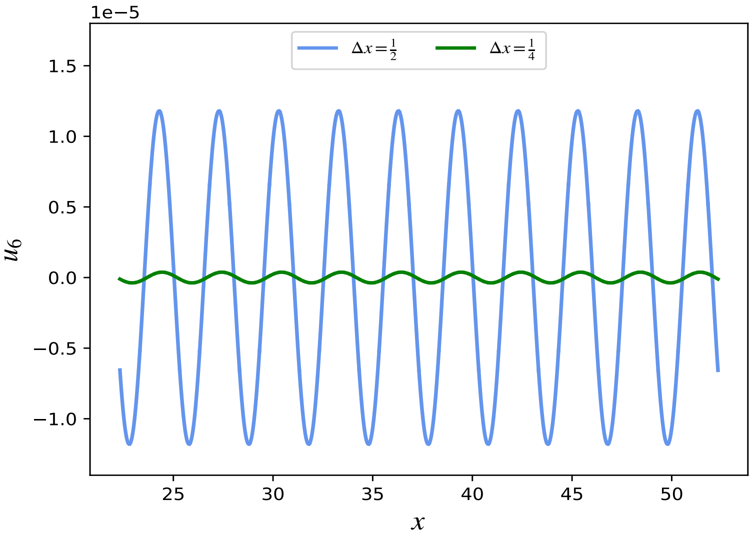


Figure 2.10: Result for edge 6 at the final time $T = 70$.

3 Higher-Order Active Flux Methods

This chapter introduces new higher-order numerical solvers designed to yield highly accurate and efficient results for hyperbolic conservation laws. The third-order Active Flux method proposed in [42] establishes a numerical approach that bridges classical Active Flux method with the ADER approach. Our novel approach, which combines the Active Flux methods with the ADER-DG technique, is motivated by the methodology outlined in that paper.

To develop higher-order accurate Active Flux methods, we initiate our exploration by considering the general one-dimensional system of hyperbolic conservation laws, represented as

$$\begin{aligned} \frac{\partial}{\partial t}\boldsymbol{u} + \frac{\partial}{\partial x}\boldsymbol{f}(\boldsymbol{u}) = 0, &\qquad x \in \Omega \subset \mathbb{R},\ t \in \mathbb{R}_0^+, \\ \boldsymbol{u}(x,0) = \boldsymbol{u}_0(x), &\qquad \forall x \in \Omega, \end{aligned} \tag{3.1}$$

where $\boldsymbol{u} = \boldsymbol{u}(x,t)$ denotes the state vector of conserved quantities and $\boldsymbol{f}(\boldsymbol{u})$ represents the possibly nonlinear flux. The initial condition is $\boldsymbol{u}_0(x)$. The classical Active Flux method has two kinds of degrees of freedom. They are cell averaged values of the conserved quantities and the point values of the conserved quantities at the cell interfaces. We define these quantities for the i^{th} element as follows

- Cell averages : $\bar{\boldsymbol{u}}_i^n \approx \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \boldsymbol{u}(x, t^n) dx,$
- Point values : $\boldsymbol{u}_{i+\frac{1}{2}}^n \approx \boldsymbol{u}(x_{i+\frac{1}{2}}, t^n).$

Compared to finite difference/volume methods, the point values offer greater flexibility for the scheme to adapt to a wide range of problems. The cell averages represent the solution across the entire cell, whereas point values directly capture local information at the cell interfaces. We can ensure that mass is conserved within each element by using the cell average update formula.

3.1 Third-order Active Flux method

Our motivation for adopting a new approach is rooted in the methodology proposed in [42]. Before we move forward with our new approach, let's take a moment to summarize the main ideas and principles of third-order Active Flux methods for linear hyperbolic conservation law problems. Considering an equidistant grid covering the domain, the averaged values can be updated using the conservative finite volume update formula

$$\bar{\boldsymbol{u}}_i^{n+1} = \bar{\boldsymbol{u}}_i^n - \frac{\Delta t}{\Delta x}\left(F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}\right). \tag{3.2}$$

Here, $F_{i+\frac{1}{2}}$ represents the averaged flux across the cell interface at $x_{i+\frac{1}{2}}$, defined as

$$F_{i+\frac{1}{2}} \approx \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \boldsymbol{f}\left(\boldsymbol{u}(x_{i+\frac{1}{2}}, t)\right) dt.$$

The order of accuracy of the scheme mainly depends on the approximation of the flux term. Here, Simpson's rule is applied to evaluate the integral as

$$\frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \boldsymbol{f}\left(\boldsymbol{u}(x_{i+\frac{1}{2}}, t)\right) dt \approx \frac{1}{6}\Big[\boldsymbol{f}\left(\boldsymbol{u}(x_{i+\frac{1}{2}}, t^n)\right) + 4\boldsymbol{f}\left(\boldsymbol{u}(x_{i+\frac{1}{2}}, t^n + \frac{\Delta t}{2})\right) \\ + \boldsymbol{f}\left(\boldsymbol{u}(x_{i+\frac{1}{2}}, t^n + \Delta t)\right)\Big].$$

Introducing the variables,

$$\begin{aligned} \boldsymbol{u}^n_{i+\frac{1}{2}} &= \boldsymbol{u}(x_{i+\frac{1}{2}}, t^n) \\ \boldsymbol{u}^{n+\frac{1}{2}}_{i+\frac{1}{2}} &= \boldsymbol{u}(x_{i+\frac{1}{2}}, t^n + \frac{\Delta t}{2}) \\ \boldsymbol{u}^{n+1}_{i+\frac{1}{2}} &= \boldsymbol{u}(x_{i+\frac{1}{2}}, t^n + \Delta t) \end{aligned}$$

leads to

$$\frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \boldsymbol{f}\left(\boldsymbol{u}(x_{i+\frac{1}{2}}, t)\right) dt \approx \frac{1}{6}\left[\boldsymbol{f}\left(\boldsymbol{u}^n_{i+\frac{1}{2}}\right) + 4\boldsymbol{f}\left(\boldsymbol{u}^{n+\frac{1}{2}}_{i+\frac{1}{2}}\right) + \boldsymbol{f}\left(\boldsymbol{u}^{n+1}_{i+\frac{1}{2}}\right)\right]. \tag{3.3}$$

Let's focus on evaluating the time integral along the cell interface. To accomplish this, we rely on the data at the points $\boldsymbol{u}(x_{i+\frac{1}{2}}, t^n + \tau)$, where $\tau \in \{0, \Delta t/2, \Delta t\}$. We acquire these data through a polynomial, constructed based on the degrees of freedom within the cell. Specifically, the piece-wise quadratic reconstruction polynomial in cell i takes the following form

$$p_i(\xi) = \boldsymbol{u}^n_{i-\frac{1}{2}}\left(3\xi^2 - 4\xi + 1\right) + \bar{\boldsymbol{u}}^n_i\left(6\xi - 6\xi^2\right) + \boldsymbol{u}^n_{i+\frac{1}{2}}\left(3\xi^2 - 2\xi\right) \tag{3.4}$$

where $\xi = \left(x - x_{i-\frac{1}{2}}\right)/\Delta x$ with the relation $0 \le \xi \le 1$.

3.1.1 Advection equation

Consider the advection equation of the form

$$\frac{\partial}{\partial t}u + a\frac{\partial}{\partial x}u = 0, \tag{3.5}$$

with the conserved quantity $u = u(x,t) : \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}$ and $a \in \mathbb{R}$.

Approach 1

First, the characteristic information of the problem is analyzed. Then, the flux is computed using Simpson's rule, as demonstrated in equation (3.3), following a similar approach as described in [36]. Using the reconstruction polynomial defined in (3.4), we can evaluate interface values at $x_{i+\frac{1}{2}}$

$$u(x_{i+\frac{1}{2}}, t^n + \tau) \approx \begin{cases} p_i\left(1 - \frac{a\tau}{\Delta x}\right), & a > 0 \\ p_{i+1}\left(-\frac{a\tau}{\Delta x}\right), & a < 0. \end{cases} \tag{3.6}$$

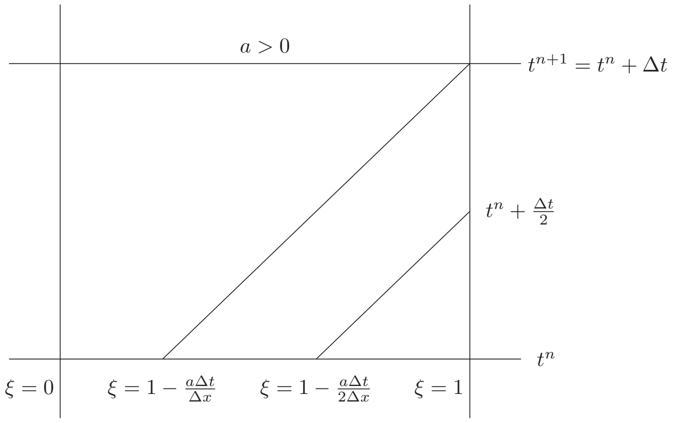

$a > 0$
$t^{n+1} = t^n + \Delta t$
$t^n + \frac{\Delta t}{2}$
t^n
$\xi = 0$
$\xi = 1 - \frac{a\Delta t}{\Delta x}$
$\xi = 1 - \frac{a\Delta t}{2\Delta x}$
$\xi = 1$

Figure 3.1: Characteristic lines accurately pass through the required points at the cell interface $x_{i+\frac{1}{2}}$.

The flux can be represented for the positive speed ($a > 0$) in the form,

$$\begin{aligned} F_{i+\frac{1}{2}} = \frac{a}{\Delta t}\int_{t^n}^{tn+1} u(x_{i+\frac{1}{2}}, t)dt &= \frac{a}{\Delta t}\int_{t^n}^{tn+1} u(x_{i+\frac{1}{2}} - a(t-t^n), t^n)dt \\ &= \frac{1}{\Delta t}\int_{x_{i+\frac{1}{2}}-a\Delta t}^{x_{i+\frac{1}{2}}} u(x, t^n)dx \\ &\approx \frac{\Delta x}{\Delta t}\int_{1-\frac{a\Delta t}{\Delta x}}^{1} p_i(\xi)d\xi \\ &= \frac{a}{6}\Big(p_i(1-\frac{a\Delta t}{\Delta x}) + 4p_i(1-\frac{a\Delta t}{2\Delta x}) + p_i(1)\Big). \end{aligned}$$

After computing the fluxes, the relation provided by (3.2) updates the cell averaged values. Computing the update of the point value involves setting $\tau = \Delta t$ in (3.6). Incorporating the information extracted from the quadratic reconstruction polynomial enables the Active Flux scheme to achieve a third-order convergence rate.

Approach 2

As an alternative approach suggested in [42], we can use the Taylor series expansion concerning the time variable

$$u(x_{i+\frac{1}{2}}, t^n+\tau) = u(x_{i+\frac{1}{2}}, t^n) + \tau u_t(x_{i+\frac{1}{2}}, t^n) + \frac{1}{2}\tau^2 u_{tt}(x_{i+\frac{1}{2}}, t^n) + \mathcal{O}(\tau^3). \tag{3.7}$$

Then, the temporal derivatives are converted to spatial derivatives following the Cauchy-Kowalewski procedure. The purpose is to approximate the spatial derivatives using the derivatives of the reconstruction polynomial. Considering the advection equation (3.5), we establish the following relations

$$u_t(x_{i+\frac{1}{2}}, t^n) = -au_x(x_{i+\frac{1}{2}}, t^n), \qquad u_{tt}(x_{i+\frac{1}{2}}, t^n) = a^2 u_{xx}(x_{i+\frac{1}{2}}, t^n). \tag{3.8}$$

The spatial derivatives are approximated as follows

$$\begin{aligned} u_x\left(x_{i+\frac{1}{2}}, t^n\right) &= \begin{cases} \frac{1}{\Delta x} p_i'(1), & a > 0 \\ \frac{1}{\Delta x} p_{i+1}'(0), & a < 0, \end{cases} \\ u_{xx}\left(x_{i+\frac{1}{2}}, t^n\right) &= \begin{cases} \frac{1}{\Delta x^2} p_i''(1), & a > 0 \\ \frac{1}{\Delta x^2} p_{i+1}''(0), & a < 0, \end{cases} \end{aligned} \tag{3.9}$$

where,

$$\begin{aligned} p_i'(\xi) &= u_{i-\frac{1}{2}}^n (6\xi - 4) + \bar{u}_i^n (6 - 12\xi) + u_{i+\frac{1}{2}} (6\xi - 2), \\ p_i''(\xi) &= 6u_{i-\frac{1}{2}}^n - 12\bar{u}_i^n + 6u_{i+\frac{1}{2}}. \end{aligned}$$

By substituting (3.8) and (3.9) into (3.7) and varying $\tau \in \left\{0, \frac{\Delta t}{2}, \Delta t\right\}$, we can derive the averaged flux at the cell interface $x_{i+\frac{1}{2}}$

$$F_{i+\frac{1}{2}} = au\left(x_{i+\frac{1}{2}}, t^n\right) - \frac{1}{2}\Delta t a^2 u_x\left(x_{i+\frac{1}{2}}, t^n\right) + \frac{1}{6}\Delta t^2 a^3 u_{xx}\left(x_{i+\frac{1}{2}}, t^n\right).$$

Moreover, in [42] it is demonstrated that these two approaches are equivalent. Here, we provide numerical results for Approach 2 using the initial condition specified in (2.22). We discretize the domain into 200 cells, covering the interval $[0, 2]$. The results obtained from the unlimited third-order Active Flux scheme are presented here. In later sections, the strategy for eliminating non-physical oscillations will be explored by implementing the MOOD limiter for the Active Flux method.

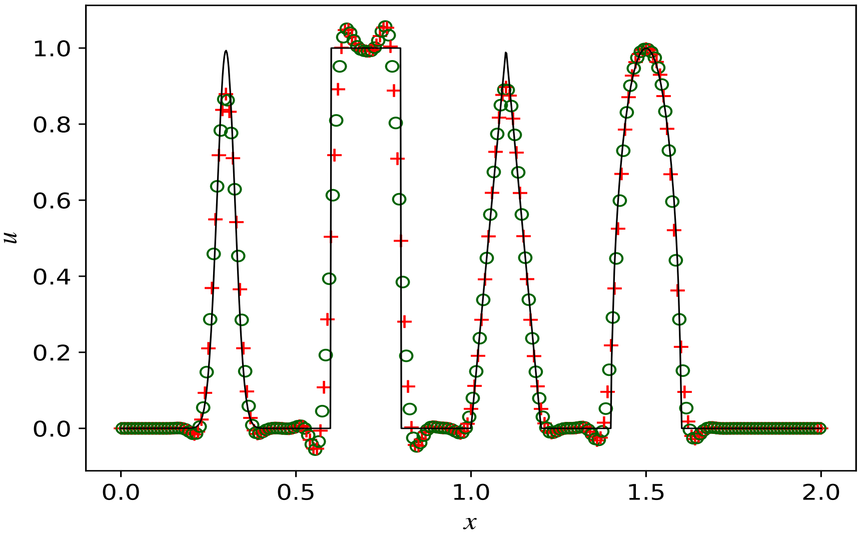


Figure 3.2: The numerical results for the advection equation using "Approach 2" are presented at the final time $T = 8$. The wave speed is set to one ($a = 1$). The green circles (○) denote the averaged values of the conserved quantity, while the symbol (+) indicates the point values at the cell interfaces.

In Approach 2, the stability of the scheme is investigated using linear advection equation with periodic boundary conditions. Following Section 2.3, the Active Flux scheme can be written in the form of (2.23). The spectrum of the iteration matrix $\mathbb{A}$ is shown in Figure 3.3. Considering Theorem 2.2, the Active Flux scheme remains stable for CFL < 1.

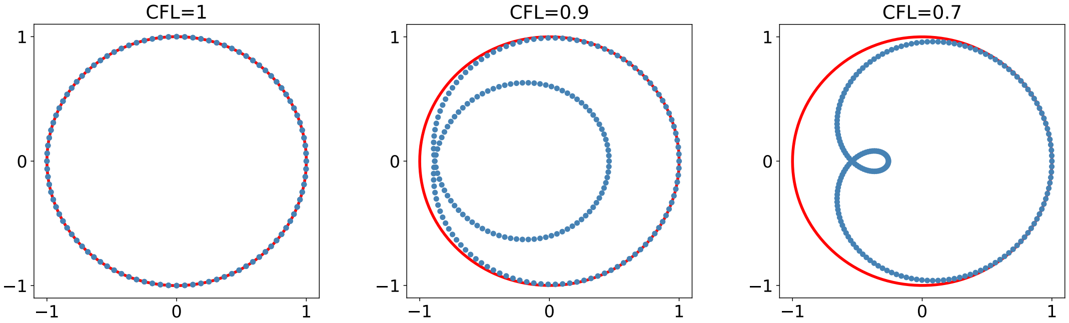


Figure 3.3: Spectrum of the iterative matrix $\mathbb{A}$ for different CFL numbers. Eigenvalues are represented by blue dots (●) and the reference unit circle is shown in red.

3.2 Arbitrary higher-order Active Flux methods (hAF)

Here, we propose a new approach to attain arbitrary high-order accuracy by combining the Active Flux method with ADER-DG techniques. In the classical Active Flux method, cell averages and the point values serve as the degrees of freedom. In this novel approach, the conservative update is not based on the averaged values of conserved quantities but rather on the degrees of freedom of the relevant fully discrete formulation for the ADER-DG scheme. For general hyperbolic conservation law problems (3.1), the discrete solution is expressed through space-only dependent test functions of degree N, mirroring the approach outlined in Chapter 2

$$\boldsymbol{u}_h(x,t^n) = \boldsymbol{u}_h(\xi(x),t^n) = \sum_{m=1}^{N+1} \hat{\boldsymbol{u}}_m(t^n)\Phi_m^N(\xi) = \hat{\boldsymbol{u}}_m(t^n)\Phi_m^N(\xi), \qquad x \in T_i. \tag{3.10}$$

Here, we specify the degree of the test functions N because the hAF methods involve constructing a more accurate local space-time predictor in each cell that corresponds to the higher-degree $M > N$ spatial test functions, reflecting our specific goal.

- The discrete solution at time t^n is represented using spatial test functions of degree N.
- The initial values $(\boldsymbol{w}_h(x,t^n))$ for the local space-time predictor are computed using spatial test functions of degree M as follows

$$\boldsymbol{w}_h(x,t^n) = \sum_{m=1}^{M+1} \hat{\boldsymbol{w}}_m(t^n)\Phi_m^M(\xi) = \hat{\boldsymbol{w}}_m(t^n)\Phi_m^M(\xi), \qquad x \in T_i. \tag{3.11}$$

In addition to the degrees of freedom $(\hat{\boldsymbol{u}}_m(t^n))$, we also consider two point values at the cell interfaces in each cell. This capability allows for the representation of a polynomial of degree M within the space-reference element $T_E = [0,1]$, with the relevant degrees of freedom situated at the quadrature points corresponding to degree M. The spatial test functions define the relationship between the higher and lower degrees as $M = N + 2$. We visually depict the placement of various degrees of freedom for the 4^{th}-order Active Flux scheme in Figure 3.4. The construction of space-dependent higher-degree polynomial in the i^{th} cell denoted as P_i, follows the constraints mentioned below

$$
\begin{aligned}
P_i(\xi) = \sum_{j=1}^{M} \hat{\boldsymbol{w}}_j(t^n)\Phi_j^M(\xi) &= \hat{\boldsymbol{w}}_j(t^n)\Phi_j^M(\xi), \qquad (3.12)\\
P_i(\xi = 0) &= P_L,\\
P_i(\xi = 1) &= P_R,\\
\int_0^1 P_i(\xi)\cdot\Phi_j^N d\xi &= \int_0^1 \boldsymbol{u}_h(\xi)\cdot\Phi_j^N d\xi.
\end{aligned}
$$

where P_L and P_R denote the left and right cell interface values respectively.

For the i^{th} cell, we can construct more accurate local space-time predictor $\left(\boldsymbol{\mathcal{U}}^M(\xi,\tau)\right)$ inside the reference element $\mathcal{Q}_E$ using the information from P_i. Next, the fully discrete ADER-DG formulation (2.16) is modified by incorporating more precise fluxes and volume integrals. First, let's discuss the conservative update formulation. Then, we'll move on to the step-by-step procedures for computing the more accurate local space-time predictor for each of the cases listed below.

- Linear scalar hyperbolic problems.
- Linear hyperbolic systems.
- Nonlinear hyperbolic problems.

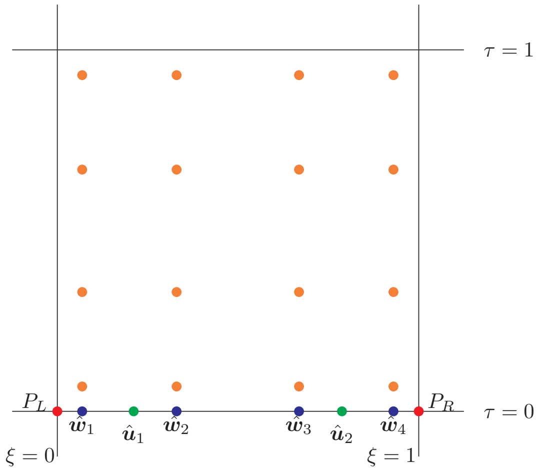


Figure 3.4: The degrees of freedom inside $\mathcal{Q}_E$ relate to the 4th-order Active Flux method. The red dots (●) denote the left and right point values at the interfaces, green dots (●) indicate the degrees of freedom related to degree N test functions and blue dots (●) denote the degrees of freedom used for degree M test functions. The values represented at (●) serve as the initial condition for the local space-time predictor. Orange dots (●) depict the degrees of freedom for creating local space-time test functions.

3.2.1 Discrete formulations for higher-order Active Flux schemes

This section will provide a more detailed explanation of the two types of update formulations used by the higher-order Active Flux methods.

For the interior degrees of freedom

The space-time dependent test functions, denoted by $\theta_k^M = \theta_k^M(\xi, \tau)$ for degree M, can approximate the numerical solution and the numerical flux term analogous to ADER-DG methods discussed in Section 2.1.2. The index k ranges from 1 to $(M+1)^2$.

$$\boldsymbol{\mathcal{U}}^M(\xi, \tau) = \sum_{\ell} \hat{\boldsymbol{\mathcal{U}}}_\ell^M \theta_\ell^M(\xi, \tau) := \hat{\boldsymbol{\mathcal{U}}}_\ell^M \theta_\ell^M \tag{3.13}$$

$$\boldsymbol{\mathcal{F}}^M(\xi, \tau) = \sum_{\ell} \hat{\boldsymbol{\mathcal{F}}}_\ell^M \theta_\ell^M(\xi, \tau) := \hat{\boldsymbol{\mathcal{F}}}_\ell^M \theta_\ell^M \tag{3.14}$$

We multiply the conservation law (3.1) with a spatial test function Φ_k^N and integrate over the physical space-time control volume $\mathcal{Q}_i^n = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] \times [t^n, t^n + \Delta t]$. By applying integration by parts

$$\int_{t^n}^{t^{n+1}} \int_{T_i} \Phi_k^N \frac{\partial \boldsymbol{u}_h}{\partial t} dx dt + \int_{t^n}^{t^{n+1}} \int_{\partial T_i} \Phi_k^N \boldsymbol{f}(\boldsymbol{u}_h) \cdot \boldsymbol{n} dS dt - \int_{t^n}^{t^{n+1}} \int_{T_i} \frac{\partial \Phi_k^N}{\partial x} \cdot \boldsymbol{f}(\boldsymbol{u}_h) dx dt = 0. \tag{3.15}$$

Here, the outward unit normal vector on the surface ∂T_i is denoted by $\boldsymbol{n}$. By inserting the relations (3.10), (3.14) and (3.14) into (3.15), we get

$$[\Phi_k^N, \Phi_m^N] \left(\hat{\boldsymbol{u}}_m^{n+1} - \hat{\boldsymbol{u}}_m^n \right) + \frac{\Delta t}{\Delta x} \int_0^1 \int_{\partial T_i} \Phi_k^N \mathcal{G}(\boldsymbol{\mathcal{U}}^{M-}, \boldsymbol{\mathcal{U}}^{M+}) \cdot \boldsymbol{n} dS d\tau - \left\langle \frac{\partial \Phi_k^N}{\partial \xi}, \boldsymbol{\mathcal{F}}^{M*} \right\rangle = 0.$$

We introduce numerical flux function $\mathcal{G}(\boldsymbol{\mathcal{U}}^{M-}, \boldsymbol{\mathcal{U}}^{M+})$. Here, $\boldsymbol{\mathcal{U}}^{M-}$ and $\boldsymbol{\mathcal{U}}^{M+}$ represent the left and right states at each cell interface. Since we consider nodal basis polynomials the relation $\hat{\boldsymbol{\mathcal{F}}}_\ell^{M*} = \Delta t \xi_x \hat{\boldsymbol{\mathcal{F}}}_\ell^M$ holds. A more simplified version of the fully-discrete formulation can be represented as follows

$$[\Phi_k^N, \Phi_m^N] \left(\hat{\boldsymbol{u}}_m^{n+1} - \hat{\boldsymbol{u}}_m^n \right) + \frac{\Delta t}{\Delta x} \int_0^1 \int_{\partial T_i} \Phi_k^N \mathcal{G}(\boldsymbol{\mathcal{U}}^{M-}, \boldsymbol{\mathcal{U}}^{M+}) \cdot \boldsymbol{n} dS d\tau - \left\langle \frac{\partial \Phi_k^N}{\partial \xi}, \theta_\ell^M \right\rangle \hat{\boldsymbol{\mathcal{F}}}_\ell^{M*} = 0. \tag{3.16}$$

Referring to (3.16), the local prediction remains pivotal, as the relation $\hat{\boldsymbol{\mathcal{F}}}_\ell = \boldsymbol{f}(\hat{\boldsymbol{\mathcal{U}}}_\ell)$ still holds. Moreover, while the ADER-DG schemes (2.16) provide $N+1$ order accuracy in both space and time for degree N spatial test functions, the hAF schemes achieve $N+3$ order accuracy. In an upcoming section, we will explore the stability and convergence of hAF schemes in greater detail.

For the Point Values

Since there is no need to apply a conservative update for the point values, we have tested two approaches for updating them and compared the resulting outcomes.

(P1) The first approach involves computing the local space-time predictor values at the cell interface $x_{i+\frac{1}{2}}$ for the time level t^{n+1} considering cells i and $i+1$. We set these values as the two states for the Riemann problem at each interface.

- $U_L = \boldsymbol{\mathcal{U}}_i^M(\xi = 1, \tau = 1)$,
- $U_R = \boldsymbol{\mathcal{U}}_{i+1}^M(\xi = 0, \tau = 1)$,

$$\boldsymbol{u}_{i+\frac{1}{2}}^{n+1} = \text{solution for RP}\left(U_L, U_R\right).$$

(P2) The second approach involves integrating the conservation law specified in (3.1) over time at the cell interface $x_{i+\frac{1}{2}}$

$$\int_{t^n}^{t^{n+1}} \frac{\partial}{\partial t}\boldsymbol{u}(x_{i+\frac{1}{2}},t)\,dt + \int_{t^n}^{t^{n+1}} \frac{\partial}{\partial x}\boldsymbol{f}\left(\boldsymbol{u}(x_{i+\frac{1}{2}},t)\right)dt = 0.$$

Simplifying,

$$\begin{aligned} \boldsymbol{u}^{n+1}_{i+\frac{1}{2}} &= \boldsymbol{u}^{n}_{i+\frac{1}{2}} - \int_{t^n}^{t^{n+1}} \frac{\partial}{\partial x}\boldsymbol{f}\left(\boldsymbol{u}(x_{i+\frac{1}{2}},t)\right)dt \\ &= \boldsymbol{u}^{n}_{i+\frac{1}{2}} - \int_{0}^{1} \frac{\partial}{\partial \xi}\boldsymbol{f}^{*}\left(\boldsymbol{u}\right)d\tau \end{aligned} \tag{3.17}$$

where $\boldsymbol{f}^{*}(\boldsymbol{u}) = \Delta t \xi_x \boldsymbol{f}(\boldsymbol{u})$. In the second step of (3.17), the integration transformed into the space-time reference coordinate system.

The spatial derivative of the general hyperbolic conservation law (3.1) can be expressed in the reference coordinates as follows

$$\frac{\partial}{\partial \tau}\left(\frac{\partial}{\partial \xi}\boldsymbol{u}\right) + \frac{\Delta t}{\Delta x}\frac{\partial}{\partial \xi}\left(\frac{\partial}{\partial \xi}\boldsymbol{f}(\boldsymbol{u})\right) = 0. \tag{3.18}$$

The corresponding Rusanov flux of (3.18) is presented in the following form

$$\begin{aligned} \frac{\partial}{\partial \xi}\boldsymbol{f} \approx \frac{1}{2}&\left[\boldsymbol{f}\left((\hat{\boldsymbol{\mathcal{U}}}^{M}_{R})_k\right)\frac{\partial}{\partial \xi}\theta^{M}_{k}\left(\xi=0,\tau\right) + \boldsymbol{f}\left((\hat{\boldsymbol{\mathcal{U}}}^{M}_{L})_k\right)\frac{\partial}{\partial \xi}\theta^{M}_{k}\left(\xi=1,\tau\right)\right] \\ &- \frac{1}{2}\sigma\left[(\hat{\boldsymbol{\mathcal{U}}}^{M}_{R})_k\frac{\partial}{\partial \xi}\theta^{M}_{k}(\xi=0,\tau) - (\hat{\boldsymbol{\mathcal{U}}}^{M}_{L})_k\frac{\partial}{\partial \xi}\theta^{M}_{k}(\xi=1,\tau)\right] \end{aligned}$$

where σ is the maximum eigenvalue of the Jacobian matrix related to the problem. $\hat{\boldsymbol{\mathcal{U}}}^{M}_{L}$ and $\hat{\boldsymbol{\mathcal{U}}}^{M}_{R}$ represent the degrees of freedom of the left and right cells, respectively. The integration in (3.17) can be computed by evaluating the numerical fluxes at the quadrature points along the interface in $\mathcal{Q}_E$.

3.2.2 MOOD limiter for the higher-order Active Flux schemes

The literature has applied various limiting strategies [3, 42] to address non-physical oscillations observed in Active Flux methods. Inspired by its success in the ADER-DG approach, we aim to apply the MOOD limiter with some modifications to the newly proposed higher-order Active Flux schemes. We will explore the adjustments made to the MOOD paradigm in this section.

We start the discussion similar to the approach outlined in Section 2.2. In this new context, we first consider an unlimited higher-order Active Flux solution $\boldsymbol{u}_h^{n+1,*}$ derived from (3.16), along with updated point values of the element T_i, denoted as $\boldsymbol{u}^{n+1,*}_{i-\frac{1}{2}}$ and $\boldsymbol{u}^{n+1,*}_{i+\frac{1}{2}}$, using P1 or P2 approaches as discussed previously. These quantities are considered as the candidate solutions for the next time level t^{n+1}. The fine subgrid $\mathcal{S}_i = \bigcup_j S_{i,j}$ is obtained by dividing each element T_i into $N_s = 2N+1$ equidistant, non-overlapping subcells. We denote the volume of each subcell as $|S_{i,j}|$. The subgrid creation is consistent with the MOOD limiter for the ADER-DG approach to ensure that the optimal time step remains valid for both higher-order and lower-order numerical schemes. At time level t^n, the subcell averages $(\boldsymbol{v}^{n}_{i,j})$ in each element T_i, can be compute by using the projection operator. We consider two situations.

- **Situation 1 :** The degree of the spatial test functions $N = 1$.

  $$\boldsymbol{v}^n_{i,j}(x, t^n) = \frac{1}{|S_{i,j}|} \int_{S_{i,j}} \boldsymbol{u}_h(x, t^n) dx := \mathcal{P}^N(\boldsymbol{u}^n_h), \qquad \forall j \in [1, N_s],$$

  where $\mathcal{P}^N$ denotes the projection operator related to degree N spatial test functions. Similarly at t^{n+1}, the candidate subcell averages are computed

  $$\boldsymbol{v}^{n+1,*}_{i,j}(x, t^{n+1}) = \frac{1}{|S_{i,j}|} \int_{S_{i,j}} \boldsymbol{u}^{n+1,*}_h(x, t^{n+1}) dx := \mathcal{P}^N(\boldsymbol{u}^{n+1,*}_h), \qquad \forall j \in [1, N_s].$$

- **Situation 2 :** The degree of the spatial test functions $N \geq 2$.

  Our primary objective is the construction of a more accurate projection operator. By referring to (3.11), we can achieve a better approximation by employing degree M spatial test functions for the values at time level t^n

  $$\boldsymbol{v}^n_{i,j}(x, t^n) = \frac{1}{|S_{i,j}|} \int_{S_{i,j}} \boldsymbol{w}_h(x, t^n) dx := \mathcal{P}^M(\boldsymbol{w}^n_h), \qquad \forall j \in [1, N_s],$$

  where $\mathcal{P}^M$ denotes the projection operator related to degree M spatial test functions. Similarly at t^{n+1}, the candidate subcell averages are computed

  $$\boldsymbol{v}^{n+1,*}_{i,j}(x, t^{n+1}) = \frac{1}{|S_{i,j}|} \int_{S_{i,j}} \boldsymbol{w}^{n+1,*}_h(x, t^{n+1}) dx := \mathcal{P}^M(\boldsymbol{w}^{n+1,*}_h), \qquad \forall j \in [1, N_s].$$

These candidate subcell averages are tested against the detection criteria to identify the problem cells. At this point, it's natural to wonder why we're considering two situations. Later in this section, we will address this problem.

Detection criteria

We kept the physical admissibility detection criteria (PAD) unchanged and made modifications only to the numerical admissibility detection criteria (NAD). We imposed the relaxed discrete maximum principle considering the element T_i as follows. The notations are shown in Figure 3.5.

$$\boldsymbol{v}_{(S,\min)} = \min_{m \in \nu(T_i)} \left(\min_{\gamma \in [1,N_s]} \boldsymbol{v}^n_{m,\gamma} \right), \qquad \boldsymbol{v}_{(S,\max)} = \max_{m \in \nu(T_i)} \left(\max_{\gamma \in [1,N_s]} \boldsymbol{v}^n_{m,\gamma} \right),$$

$$\boldsymbol{v}_{(L,\min)} = \min_{m \in \nu(T_{L,i})} \left(\min_{\gamma \in [1,N_s]} \boldsymbol{v}^n_{m,\gamma} \right), \qquad \boldsymbol{v}_{(L,\max)} = \max_{m \in \nu(T_{L,i})} \left(\max_{\gamma \in [1,N_s]} \boldsymbol{v}^n_{m,\gamma} \right),$$

$$\boldsymbol{v}_{(R,\min)} = \min_{m \in \nu(T_{R,i})} \left(\min_{\gamma \in [1,N_s]} \boldsymbol{v}^n_{m,\gamma} \right), \qquad \boldsymbol{v}_{(R,\max)} = \max_{m \in \nu(T_{R,i})} \left(\max_{\gamma \in [1,N_s]} \boldsymbol{v}^n_{m,\gamma} \right).$$

- Subcell averages

$$CV_{\min} = \min\left(\boldsymbol{v}_{(S,\min)}, P_{(L,i)}, P_{(R,i)}\right),$$
$$CV_{\max} = \max\left(\boldsymbol{v}_{(S,\max)}, P_{(L,i)}, P_{(R,i)}\right),$$

$$CV_{\min} - \delta_v \leq \boldsymbol{v}^{n+1,*}_{i,j} \leq CV_{\max} + \delta_v \qquad \forall j \in [1, N_s]. \tag{3.19}$$

- Left point value

$$\begin{aligned} CP_{L,\min} &= \min\left(\boldsymbol{v}_{(L,\min)}, P_{(L,i-1)}, P_{(L,i)}, P_{(R,i)}\right), \\ CP_{L,\max} &= \max\left(\boldsymbol{v}_{(L,\max)}, P_{(L,i-1)}, P_{(L,i)}, P_{(R,i)}\right), \end{aligned}$$

$$CP_{L,\min} - \delta_L \le \boldsymbol{u}^{n+1,*}_{i-\frac{1}{2}} \le CP_{L,\max} + \delta_L. \tag{3.20}$$

- Right point value

$$\begin{aligned} CP_{R,\min} &= \min\left(\boldsymbol{v}_{(R,\min)}, P_{(L,i)}, P_{(R,i)}, P_{(R,i+1)}\right), \\ CP_{R,\max} &= \max\left(\boldsymbol{v}_{(R,\max)}, P_{(L,i)}, P_{(R,i)}, P_{(R,i+1)}\right), \end{aligned}$$

$$CP_{R,\min} - \delta_R \le \boldsymbol{u}^{n+1,*}_{i+\frac{1}{2}} \le CP_{R,\max} + \delta_R. \tag{3.21}$$

The parameters are computed as follows

$$\begin{aligned} \delta_v &= \max\Big(\delta_0, \epsilon \cdot (CV_{\max} - CV_{\min})\Big), \\ \delta_L &= \max\Big(\delta_0, \epsilon \cdot (CP_{L,\max} - CP_{L,\min})\Big), \\ \delta_R &= \max\Big(\delta_0, \epsilon \cdot (CP_{R,\max} - CP_{R,\min})\Big), \end{aligned}$$

where $\delta_0 = 1e-5$ and $\epsilon = 1e-4$.

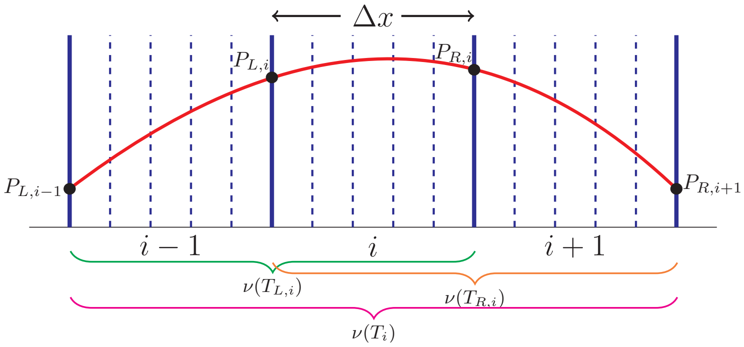


Figure 3.5: Information regarding the detection procedure in hAF methods. $\nu(T_i)$ denotes a set containing element T_i with its neighbors, $\nu(T_{L,i})$ represents a set containing element T_i with its left neighboring element and $\nu(T_{R,i})$ represents a set containing element T_i with its right neighboring element. The point values at time t^n are denoted by P with appropriate subscript indicating their locations.

If the candidate solutions fail to satisfy any of the criteria outlined above (3.19), (3.20), or (3.21), we identify the corresponding element as a problem cell. These problem cells need to be updated using a lower-order scheme as discussed in Section 2.2. The second-order finite volume scheme with *minmod* limiter is applied to update the subcell averages in the problem cells and denoted by

$$\boldsymbol{v}_{i,j}(x, t^{n+1}) = \Psi(\boldsymbol{v}_{i,j}(x, t^n)). \tag{3.22}$$

Subsequently, we must recover the higher-order solution from the subcell averages which are updated using the lower-order finite volume method given in (3.22). This is done by using the reconstruction operator $\mathcal{R}^N$ (or $\mathcal{R}^M$) corresponding to the projection operators $\mathcal{P}^N$ (or $\mathcal{P}^M$). The operators satisfy the relation given in (2.20).

At this point, we can explain why we consider two situations for creating the projection operator. The problem arises at the recovery step. In situation 1, if we consider the matrix corresponding to the degree M reconstruction operator, we have to solve an *underdetermined* system of equations. However, this problem no longer occurs when the degree of the spatial test functions is greater than or equal to 2. To address this issue, we consider the two situations mentioned at the beginning of this section in order to create the projection operator.

Furthermore, in situation 1, the matrix related to $\mathcal{R}^N$ directly provides the $\hat{\boldsymbol{u}}(t^{n+1})$ values in which we are interested. But in situation 2, the matrix related to $\mathcal{R}^M$ provides the $\hat{\boldsymbol{w}}(t^{n+1})$ values at the degrees of freedoms related to the degree M. We need to do one more step to compute the $\hat{\boldsymbol{u}}(t^{n+1})$ values from them.

From (3.12) the higher-degree polynomial approximation for the values at time level t^{n+1} can be written as follows,

$$P_i(\xi) = \sum_{j=1}^{M} \hat{\boldsymbol{w}}_j(t^{n+1})\Phi_j^M(\xi) = \hat{\boldsymbol{w}}_j(t^{n+1})\Phi_j^M(\xi), \tag{3.23}$$

with the relation

$$\int_0^1 P_i(\xi) \cdot \Phi_j^N d\xi = \int_0^1 \boldsymbol{u}_h(\xi) \cdot \Phi_j^N d\xi. \tag{3.24}$$

Substituting (3.10) and (3.23) into (3.24) , we can compute the $\hat{\boldsymbol{u}}(t^{n+1})$ values.

After following the procedure, we reduce the non-physical oscillations that may arise in the solution $\boldsymbol{u}_h(x, t^{n+1})$ at the subsequent time level t^{n+1}. However, we have not yet implemented any limiting strategy for the point values in problem cells. Here we suggest two approaches for limiting the point values.

(L1) Once the lower-order updated subcell averages are obtained from (3.22), we proceed to solve the Riemann problem at each cell interface of the troubled cells. We select the two states for the Riemann problem as the nearest left and right subcell averages at the cell interface. The solution of the Riemann problem can be assigned as the limited point value at time level t^{n+1}. Figure 3.6 illustrates the situation.

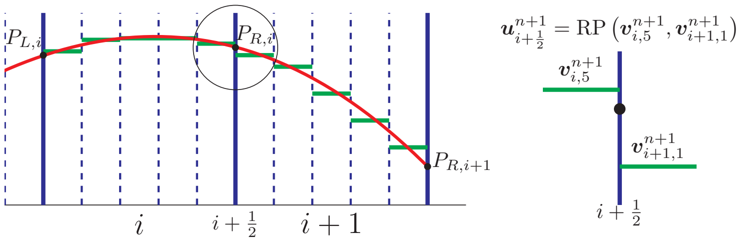

$P_{L,i}$
$P_{R,i}$
$P_{R,i+1}$
i
$i+\frac{1}{2}$
$i+1$
$\boldsymbol{u}^{n+1}_{i+\frac{1}{2}} = \text{RP}\left(\boldsymbol{v}^{n+1}_{i,5}, \boldsymbol{v}^{n+1}_{i+1,1}\right)$
$\boldsymbol{v}^{n+1}_{i,5}$
$\boldsymbol{v}^{n+1}_{i+1,1}$
$i+\frac{1}{2}$

Figure 3.6: Limiting the point values considering the solution of the Riemann problem at the cell interface. This representation corresponds to the 5^{th}- order Active Flux scheme.

(L2) Alternatively, we can extend the limitation on point values in a broader manner. We determine the point values at t^{n+1} by averaging the values of the left and right subcell averages from (3.22) near the cell interface. In one-dimensional scenarios, this involves computing the arithmetic mean of the two subcell averages under consideration.

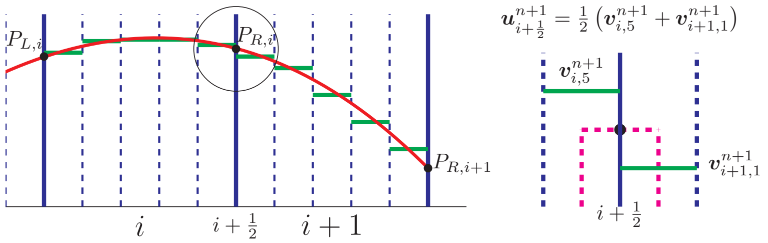

$P_{L,i}$
$P_{R,i}$
$P_{R,i+1}$
i
$i+\frac{1}{2}$
$i+1$
$\boldsymbol{u}^{n+1}_{i+\frac{1}{2}} = \frac{1}{2}\left(\boldsymbol{v}^{n+1}_{i,5} + \boldsymbol{v}^{n+1}_{i+1,1}\right)$
$\boldsymbol{v}^{n+1}_{i,5}$
$\boldsymbol{v}^{n+1}_{i+1,1}$
$i+\frac{1}{2}$

Figure 3.7: Limiting the point values considering the arithmetic mean of the nearest subcell averages at the cell interface. This representation corresponds to the 5^{th}- order Active Flux scheme.

3.2.3 Linear scalar problem

In this section, we discuss our proposed method by considering a linear scalar hyperbolic problem with the flux function $f(u) = au$

$$\frac{\partial}{\partial t}u + a\frac{\partial}{\partial x}u = 0.$$

For simplicity, we first transform the problem into the space-time reference element $\mathcal{Q}_E$

$$\frac{\partial}{\partial \tau}u + a^*\frac{\partial}{\partial \xi}u = 0, \quad \text{with} \quad a^* = \Delta t \xi_x a. \tag{3.25}$$

The weak formulation relies on the local space-time dependent test functions denoted by $\theta^M_k(\xi,\tau)$, corresponding to degree M. We multiply (3.25) by space-time dependent test functions and integrate over the space-time reference element $\mathcal{Q}_E$

$$\left\langle \theta^M_k, \frac{\partial}{\partial \tau}u \right\rangle + a^*\left\langle \theta^M_k, \frac{\partial}{\partial \xi}u \right\rangle = 0.$$

We apply integration by parts in time and it leads to

$$\left[\theta^M_k, \boldsymbol{\mathcal{U}}^M\right]_1 - \left[\theta^M_k, \boldsymbol{w}_h\right]_0 - \left\langle \frac{\partial}{\partial \tau}\theta^M_k, \boldsymbol{\mathcal{U}}^M \right\rangle + a^*\left\langle \theta^M_k, \frac{\partial}{\partial \xi}\boldsymbol{\mathcal{U}}^M \right\rangle = 0.$$

Substituting the relation (3.13) and (3.11),

$$\left[\theta^M_k, \theta^M_\ell\right]_1 \hat{\boldsymbol{\mathcal{U}}}^M_\ell - \left[\theta^M_k, \Phi^M_m\right]_0 \hat{\boldsymbol{w}}_m(t^n) - \left\langle \frac{\partial}{\partial \tau}\theta^M_k, \theta^M_\ell \right\rangle \hat{\boldsymbol{\mathcal{U}}}^M_\ell + a^*\left\langle \theta^M_k, \frac{\partial}{\partial \xi}\theta^M_\ell \right\rangle \hat{\boldsymbol{\mathcal{U}}}^M_\ell = 0.$$

By rearranging the terms, we can obtain the following expression

$$\left(\left[\theta^M_k, \theta^M_\ell\right]_1 - \left\langle \frac{\partial}{\partial \tau}\theta^M_k, \theta^M_\ell \right\rangle + a^*\left\langle \theta^M_k, \frac{\partial}{\partial \xi}\theta^M_\ell \right\rangle\right) \hat{\boldsymbol{\mathcal{U}}}^M_\ell = \left[\theta^M_k, \Phi^M_m\right]_0 \hat{\boldsymbol{w}}_m(t^n). \tag{3.26}$$

Introducing shorthand notation for the matrices: $K^\tau_{k\ell} = \left\langle \frac{\partial}{\partial \tau}\theta^M_k, \theta^M_\ell \right\rangle$, $K^\xi_{k\ell} = \left\langle \theta^M_k, \frac{\partial}{\partial \xi}\theta^M_\ell \right\rangle$, $F^0_{km} = \left[\theta^M_k, \Phi^M_m\right]_0$, $F^1_{k\ell} = \left[\theta^M_k, \theta^M_\ell\right]_1$, we can write (3.26) in the compact form

$$Y_{k\ell}\hat{\boldsymbol{\mathcal{U}}}_\ell^M = F_{km}^0\hat{\boldsymbol{w}}_m(t^n) \tag{3.27}$$

with the system matrix

$$Y_{k\ell} = F_{k\ell}^1 - K_{k\ell}^\tau + a^* K_{k\ell}^\xi.$$

The system matrix, always invertible as mentioned in [24], ensures that the solution to (3.27) can be computed directly. Subsequently, using (3.13), we can compute the local spacetime predictor $\boldsymbol{\mathcal{U}}^M(\xi,\tau)$. With all the necessary information at hand, we are prepared to proceed with the validation of the hAF method for the linear scalar problem.

Results

First, we present the results obtained through the unlimited higher-order Active Flux schemes proposed in Section 3.2.1. Both approaches for updating the point values (P1 and P2) are examined. We apply the standard initial conditions proposed in [44] and set the wave speed to one ($a = 1$).

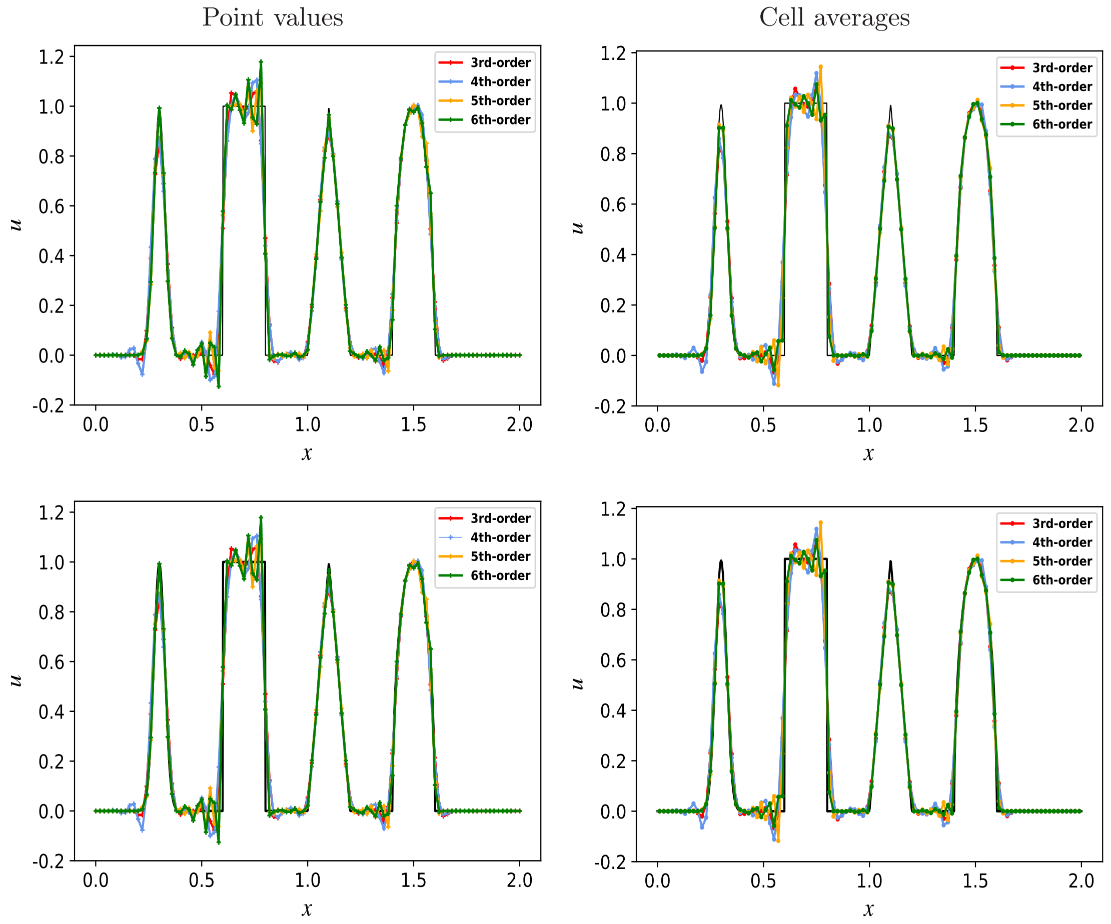


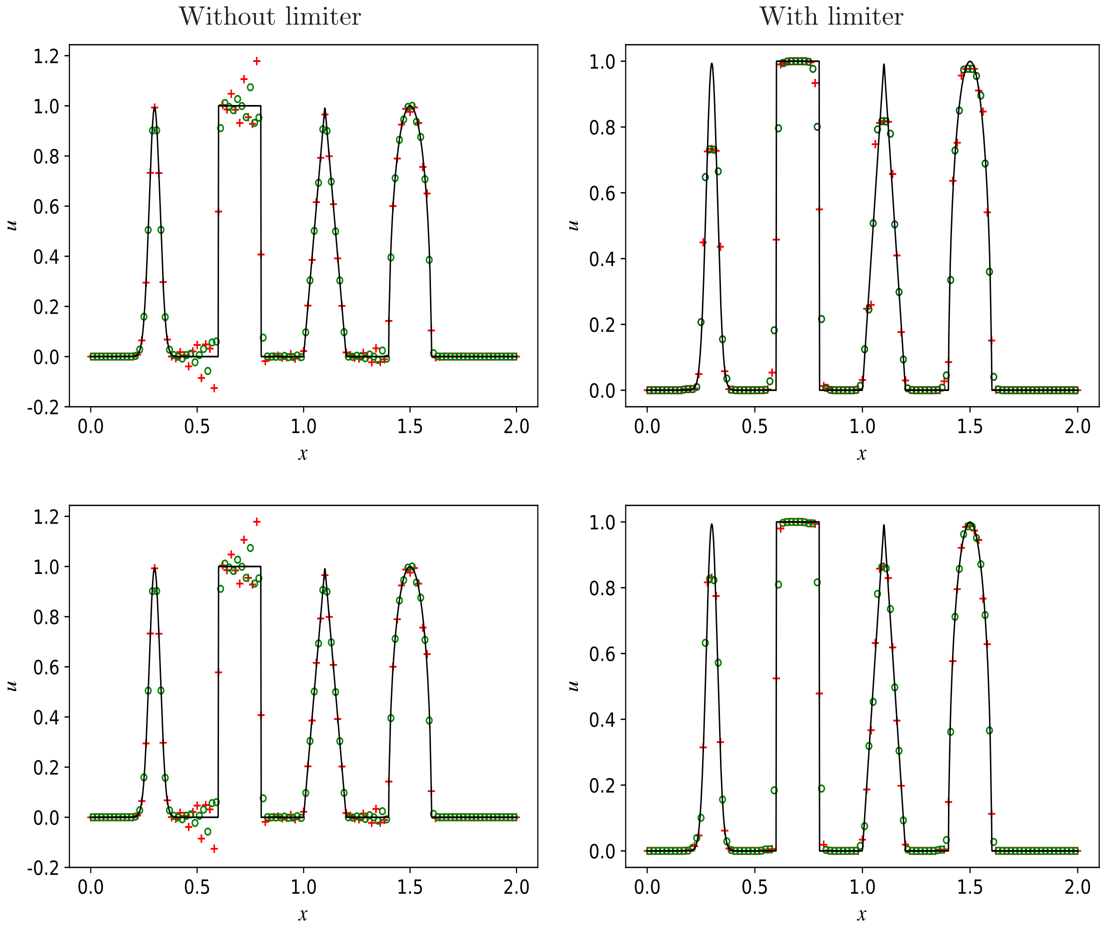
Figure 3.8: The results obtained from the unlimited hAF schemes are compared. The P1 approach is shown in the top row, while the P2 approach is shown in the bottom row. The domain consists of 100 elements and the results are presented at the final time $T = 8$. Black line represents the exact solution (——).

According to Figure 3.8, both approaches provide identical results for the linear advection equation. Next, we consider higher-order Active Flux methods with the MOOD limiter proposed in Section 3.2.2. Multiple combinations are available to represent these outcomes, employing different types of point value updates (P1 or P2) and limiting strategies (L1 or L2). We specifically highlight two combinations, with others operating similarly.

1. The P1 point value update and the L1 limiting strategy are both highly dependent on the characteristics of the problem. This combination is denoted as P1L1.

2. In order to adopt a more generalized approach, we apply P2 point value update and L2 limiting strategy. We denote it as P2L2.

We considered only 6^{th}-order Active Flux scheme with 100 elements to compare the performance of the two combinations.


Without limiter

With limiter

Figure 3.9: Results at the final time $T = 8$. The point values (+), cell averages (○), and exact solution (—) are displayed. Top row: the unlimited results with P1 approach (*left*) and results with P1L1 approach (*right*) are presented. Bottom row: the unlimited results with P2 approach (*left*) and results with P2L2 approach (*right*) are displayed.

Stability and convergence

The stability ensures that the numerical method does not amplify errors as the computation progresses. We can write the hAF scheme in the form of (2.23) and apply the Theorem 2.2 for the stability analysis. The eigenvalues of the matrix $\mathbb{A}$ for different schemes are depicted in Figure 3.10.

Moreover, in Remark 3.1, we compare some properties of the higher-order Active Flux method and the ADER-DG method considering linear hyperbolic problems.

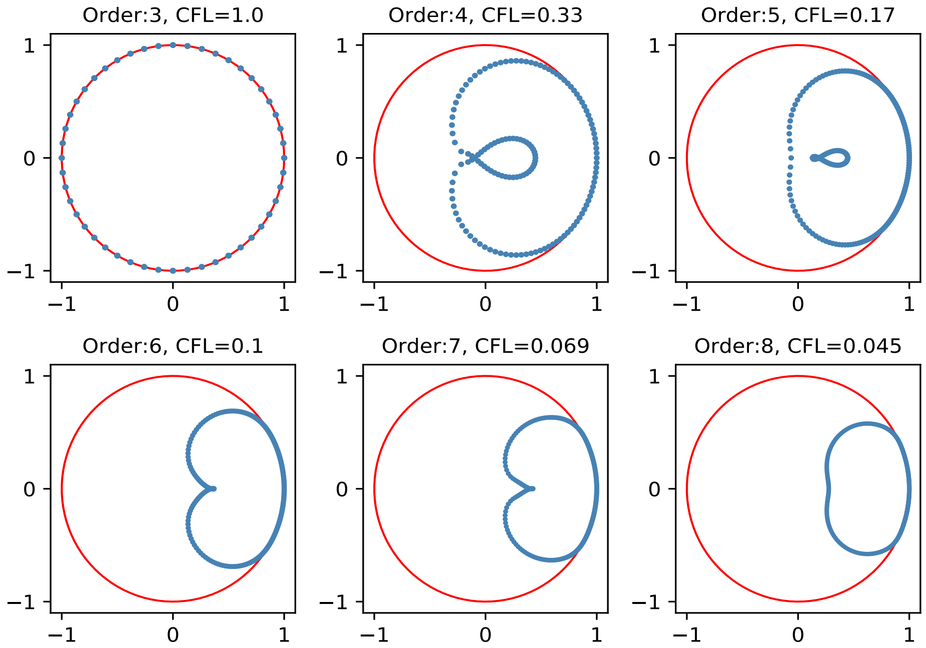


Figure 3.10: The eigenvalues (●) of the iterative matrix $\mathbb{A}$ for different CFL numbers and related convergence orders. The reference unit circle is shown in red.

Remark 3.1 Here, we provide a summary of the comparison between the ADER-DG method and the higher-order Active Flux method, considering linear scalar problems. Let's consider a domain with $\mathcal{Z}$ cells.

Comparison between the ADER-DG and hAF methods						
Order	Degree (N)		DOF		CFL	
	hAF	ADER-DG	hAF	ADER-DG	hAF	ADER-DG
3	0	2	$2\mathcal{Z}+1$	$3\mathcal{Z}$	1.0	0.17
4	1	3	$3\mathcal{Z}+1$	$4\mathcal{Z}$	0.33	0.10
5	2	4	$4\mathcal{Z}+1$	$5\mathcal{Z}$	0.17	0.069
6	3	5	$5\mathcal{Z}+1$	$6\mathcal{Z}$	0.10	0.045
7	4	6	$6\mathcal{Z}+1$	$7\mathcal{Z}$	0.069	0.038

The convergence study measures whether the numerical solution approaches the exact solution as we refine the number of elements in the grid. We list the convergence results in Table 3.1 for the smooth initial condition $u(x,0) = \sin(2\pi x)$. As anticipated, hAF schemes achieve the expected convergence orders. We start with 4 elements in the grid and refine it to 20 elements. The periodic boundary conditions are imposed. We calculate the data with the final time set at $T = 5$.

hAF	N_E	L^1	$\mathcal{O}_{L^1}$	L^2	$\mathcal{O}_{L^2}$	L^∞	$\mathcal{O}_{L^\infty}$
$\mathcal{O}3$ $(N=0)$	4	8.5982E-02	-	1.0022E-01	-	1.3748E-01	-
	8	1.2195E-02	2.81	1.4259E-02	2.81	2.0166E-02	2.76
	12	3.8726E-03	2.82	4.2997E-03	2.95	6.0417E-03	2.97
	16	1.6244E-03	3.02	1.7967E-03	3.03	2.5198E-03	3.03
	20	8.1406E-04	3.09	9.0369E-04	3.07	1.2746E-03	3.05
$\mathcal{O}4$ $(N=1)$	4	1.3748E-01	-	1.3748E-01	-	1.3844E-01	-
	8	9.4489E-03	3.86	1.0229E-02	3.74	1.3474E-02	3.36
	12	1.8843E-03	3.97	2.0692E-03	3.94	2.8335E-03	3.84
	16	5.9711E-04	3.99	6.5898E-04	3.97	9.1510E-04	3.92
	20	2.4508E-04	3.99	2.7110E-04	3.98	3.7872E-04	3.95
$\mathcal{O}5$ $(N=2)$	4	5.6625E-03	-	6.1488E-03	-	8.0592E-03	-
	8	1.5934E-04	5.15	1.7756E-04	5.11	2.4818E-04	5.02
	12	2.0531E-05	5.05	2.2867E-05	5.05	3.2197E-05	5.03
	16	4.8478E-06	5.01	5.3969E-06	5.01	7.6165E-06	5.01
	20	1.5860E-06	5.00	1.7653E-06	5.00	2.4940E-06	5.00
$\mathcal{O}6$ $(N=3)$	4	1.6970E-04	-	1.7321E-04	-	2.0439E-04	-
	8	2.6033E-06	6.02	2.8304E-06	5.93	3.8261E-06	5.73
	12	2.2928E-07	5.99	2.5236E-07	5.96	3.5024E-07	5.89
	16	4.0847E-08	5.99	4.5143E-08	5.98	6.3193E-08	5.95
	20	1.0715E-08	5.99	1.1864E-08	5.98	1.6673E-08	5.97
$\mathcal{O}7$ $(N=4)$	4	6.3602E-06	-	6.5791E-06	-	8.0432E-06	-
	8	4.7258E-08	7.07	5.1680E-08	6.99	7.0820E-08	6.82
	12	2.7627E-09	7.00	3.0454E-09	6.98	4.2418E-09	6.94
	16	3.7181E-10	6.97	4.1060E-10	6.96	5.7328E-10	6.95
	20	8.0069E-11	6.88	8.8612E-11	6.87	1.2434E-10	6.84

Table 3.1: Numerical convergence rates for the advection equation with higher-order Active Flux schemes evaluated from third to seventh order of accuracy in both space and time.

3.2.4 Linear systems

The extension of the linear scalar problem to the linear systems can be done as follows. Let us consider general linear hyperbolic systems of the form

$$\frac{\partial}{\partial t}\boldsymbol{u} + A\frac{\partial}{\partial x}\boldsymbol{u} = 0. \tag{3.28}$$

The variable $\boldsymbol{u}$ represents a vector with n unknowns, and the matrix A is a square matrix of size $n \times n$ with n real eigenvalues and corresponding n linear independent eigenvectors. The linear system given in (3.28) can be transformed to the space-time reference element using the transformation relations given in (2.3). We get

$$\frac{\partial}{\partial \tau}\boldsymbol{u} + A^{*}\frac{\partial}{\partial \xi}\boldsymbol{u} = 0,$$

with $A^{*} = \Delta t \xi_x A$. Introducing the indices p and q for the variables, the matrix A is defined as $A = \left(a_{qp}\right)_{qp}$. Furthermore, indices k,ℓ and m correspond to the degrees of freedom. Similar to (3.26) in Section 3.2.3, we can represent the system as

$$\left(\delta_{qp}\left(F^{1}_{k\ell} - K^{\tau}_{k\ell}\right) + a^{*}_{qp}K^{\xi}_{k\ell}\right)\hat{\boldsymbol{\mathcal{U}}}^{M}_{p\ell} = \delta_{qp}F^{0}_{km}\hat{\boldsymbol{w}}_{pm}(t^{n}) \tag{3.29}$$

where δ_{qp} denotes the Kronecker symbol. Introducing the system matrix

$$Y_{qpkl} = \left(\delta_{qp}\left(F^{1}_{k\ell} - K^{\tau}_{k\ell}\right) + a^{*}_{qp}K^{\xi}_{k\ell}\right),$$

compact form of (3.29) can be written as follows

$$Y_{qpkl}\hat{\boldsymbol{\mathcal{U}}}^{M}_{p\ell} = \delta_{qp}F^{0}_{km}\hat{\boldsymbol{w}}_{pm}(t^{n}). \tag{3.30}$$

We can compute $\hat{\boldsymbol{\mathcal{U}}}^{M}_{p\ell}$ from (3.30) and then obtain the local space-time predictor from (3.13). First, apply the unlimited higher-order Active Flux schemes considering the system (3.28) with

$$\boldsymbol{u} = \begin{bmatrix} p \\ v \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & K_0 \\ \frac{1}{\rho_0} & 0 \end{bmatrix},$$

where the constants take the values $K_0 = \rho_0 = 1.4$. The initial conditions are defined in (2.24). Figure 3.11 represents the first component p using 3^{rd} and 6^{th}-order Active Flux schemes. Here, we use approximately 400 degrees of freedom for both schemes. In other words, the domain contains 200 cells for 3^{rd}-order scheme and 80 cells for 6^{th}-order scheme. This demonstrates that the 6^{th}-order Active Flux scheme is able to capture all the features of the solution more sharply than the 3^{rd}-order scheme. We consider the P2 approach for the point value update.

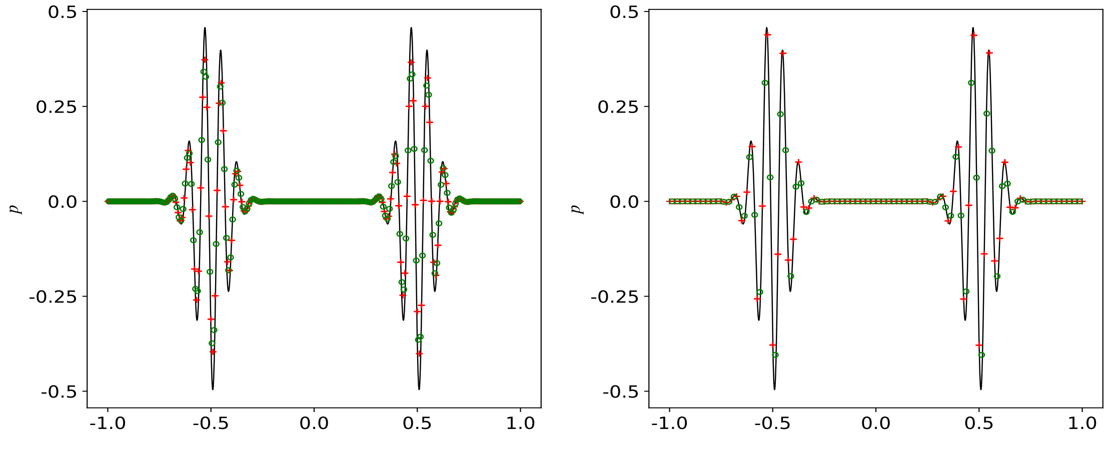


Figure 3.11: The 3^{rd}-order AF solution (*left*) and the 6^{th}-order AF solution (*right*). Results are presented at the final time $T = 7.5$. The point values (+) and cell averages (○) are displayed. The black line represents the exact solution (——).

Next, we aim to apply the MOOD limiter. When applying the MOOD limiter to linear scalar problems, it demonstrates more favorable outcomes for the general limiting strategy P2L2 than the Riemann solution-based strategy P1L1. This general approach P2L2 involves using the averaged value of nearest subcell averages as limited point value, making it a preferred choice for linear systems. Here, we introduce another example with discontinuities. Let's consider a linear hyperbolic system (3.28) with,

$$\boldsymbol{u} = \begin{bmatrix} p \\ v \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & 1 \\ \bar{c}^2 - \bar{v}^2 & 2\bar{v} \end{bmatrix}.$$

The parameters take the values $\bar{c} = 3/2$ and $\bar{v} = -1/2$. The system has two waves with the speeds $\lambda_1 = -2$ and $\lambda_2 = 1$. To capture the two waves more accurately, we imposed a Roe-type flux function for the simulation. Figure 3.12 shows that the numerical solution sharply captured the discontinuities without producing oscillations. The domain consists of 100 elements. We impose the following initial conditions

$$\rho(x,0) = \begin{cases} 1, & 0.4 < x < 0.6 \\ 0, & \text{otherwise}, \end{cases}$$
$$q(x,0) = 0.$$

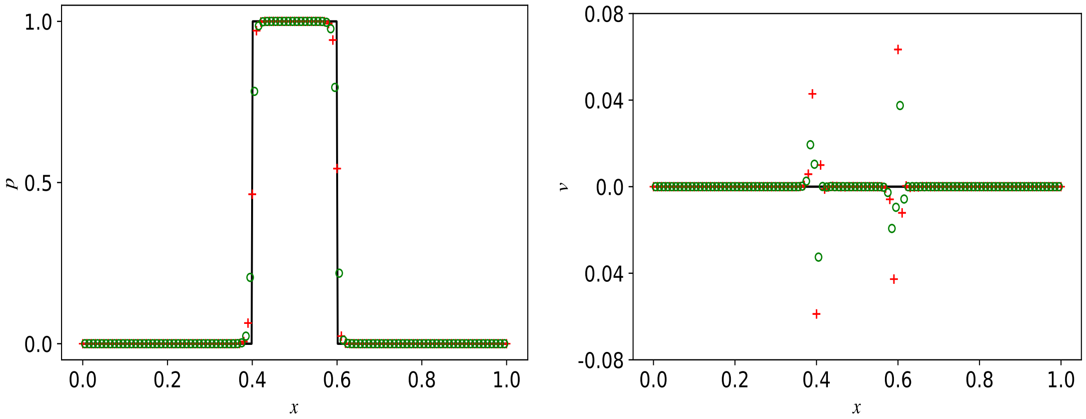


Figure 3.12: The results are represented at the final time $T = 1$. The cell averages are denoted by (o) and the point values are indicated by (+). These results correspond to the 6^{th}-order Active Flux scheme. The black line represents the exact solution (——).

3.2.5 Non-linear problems

The general non-linear hyperbolic systems given in (3.1) can be written in the space-time reference coordinate system as

$$\frac{\partial}{\partial \tau}\boldsymbol{u} + \frac{\partial}{\partial \xi}\boldsymbol{f}^*(\boldsymbol{u}) = 0, \tag{3.31}$$

where $\boldsymbol{f}^*(\boldsymbol{u}) = \Delta t \xi_x \boldsymbol{f}(\boldsymbol{u})$. We can multiply (3.31) with the higher degree space-time test functions θ_k^M and integrate over the $\mathcal{Q}_E$. Applying the integration by parts for the time derivative term we obtain the following relation

$$\left[\theta_k^M, \boldsymbol{u}\right]_1 - \left[\theta_k^M, \boldsymbol{w}\right]_0 - \left\langle \frac{\partial}{\partial \tau}\theta_k^M, \boldsymbol{u} \right\rangle + \left\langle \theta_k^M, \frac{\partial}{\partial \xi}\boldsymbol{f}^*(\boldsymbol{u}) \right\rangle = 0. \tag{3.32}$$

Substituting the relations (3.10), (3.11), (3.13) and (3.14) into (3.32), we get

$$\left[\theta_k^M, \theta_\ell^M\right]_1 \hat{\boldsymbol{\mathcal{U}}}_\ell^M - \left[\theta_k^M, \Phi_m^M\right]_0 \hat{\boldsymbol{w}}_m(t^n) - \left\langle \frac{\partial}{\partial \tau}\theta_k^M, \theta_\ell^M \right\rangle \hat{\boldsymbol{\mathcal{U}}}_\ell^M + \left\langle \theta_k^M, \frac{\partial}{\partial \xi}\theta_\ell^M \right\rangle \hat{\boldsymbol{\mathcal{F}}}_\ell^{M*} = 0.$$

The relation between the degrees of freedom of the numerical solution and those of the numerical flux can be expressed similarly to Chapter 2, as follows

$$\hat{\boldsymbol{\mathcal{F}}}_\ell^M = \boldsymbol{f}(\hat{\boldsymbol{\mathcal{U}}}_\ell^M).$$

Since we consider a non-linear flux function, the iteration procedure is imposed similarly to the discussion in Section 2.1.2. Considering the notations defined in Section 3.2.3 we can obtain

$$\left(F_{k\ell}^1 - K_{k\ell}^\tau\right)\left(\hat{\boldsymbol{\mathcal{U}}}_\ell^M\right)^{r+1} = F_{km}^0 \hat{\boldsymbol{w}}_m(t^n) - K_{k\ell}^\xi \left(\hat{\boldsymbol{\mathcal{F}}}_\ell^{M*}\right)^r \tag{3.33}$$

where r represents the iteration index. In general, the convergence of the iteration (3.33) can not be guaranteed with the improved CFL numbers presented in Remark 3.1. We consider the same CFL numbers which we consider in ADER-DG methods (see Section 2.1.3) to ensure the convergence of the (3.33).

Burgers' equation

First, we consider the well-known inviscid Burgers' equation as the non-linear scalar problem. The initial condition is applied similarly to that in [2].

$$\frac{\partial}{\partial t}u + u\frac{\partial}{\partial x}u = 0,$$
$$u(x,0) = 2.5\exp\left(-\frac{(x-0.5)^2}{0.1^2}\right) - 0.2.$$

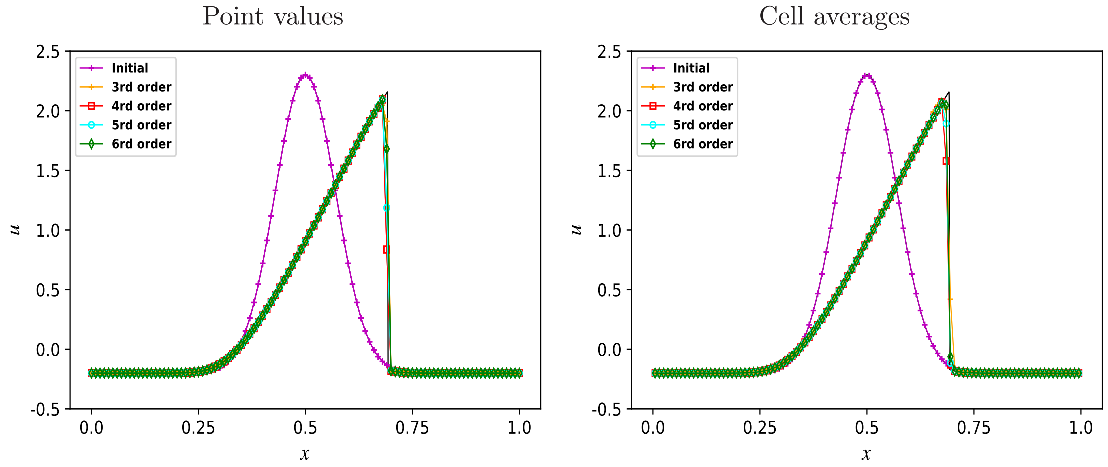


Figure 3.13: The results are presented for 3^{rd} to 6^{th}-order Active Flux schemes under the P1L1 approach using 100 elements. The black line (——) represents the reference solution using a first-order upwind scheme with 4000 cells. The final time is set at $T = 0.1$.

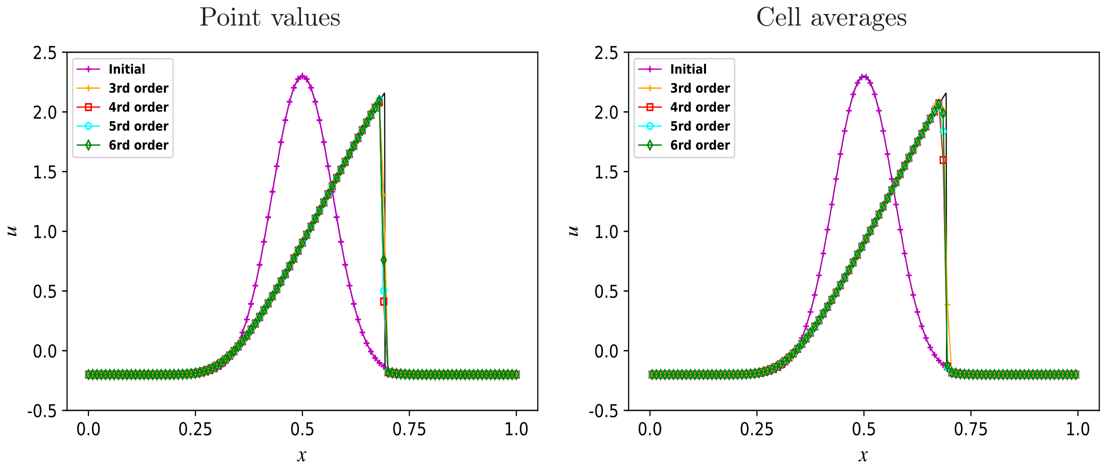


Figure 3.14: The results are presented for 3^{rd} to 6^{th}-order Active Flux schemes under the P2L2 approach using 100 elements. The black line (——) represents the reference solution using a first-order upwind scheme with 4000 cells. The final time is set at $T = 0.1$.

Figure 3.13 and Figure 3.14 correspond to P1L1 and P2L2 approach respectively. All the hAF schemes yield similar results. A simulation of Burgers' equation with an initial Gaussian distribution shows a notable transformation after time $T = 0.1$. The MOOD limiter successfully reduces the oscillations near the discontinuity.

Shallow water equations

Exploring nonlinear hyperbolic systems, we turn our attention to the shallow water equations, which play a critical role in understanding fluid behavior in shallow water environments. The shallow water system can be written in the following form

$$\begin{pmatrix} h \\ hv \end{pmatrix}_t + \begin{pmatrix} hv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}_x = 0,$$

where h represents the height of the water surface, v indicates the velocity of the water flow in the x direction and g is the gravitational acceleration. The initial condition is given as follows

$$h(x,0) = \begin{cases} h_0 = 1, & x < 0.5 \\ h_1 = 0.5, & x > 0.5 \end{cases}, \qquad v(x,0) = 0.$$

The two combinations, based on the point values update and the limiting strategies, are tested using a smaller number of elements (32) and a sufficiently larger number of elements (100). Figure 3.15 shows the results with the more problem-dependent approach P1L1, while Figure 3.16 represents the more general approach P2L2. Both approaches sharply capture the exact solution, specially when the number of elements is large. However, hAF schemes are able to capture all features of the exact solutions with fewer elements.

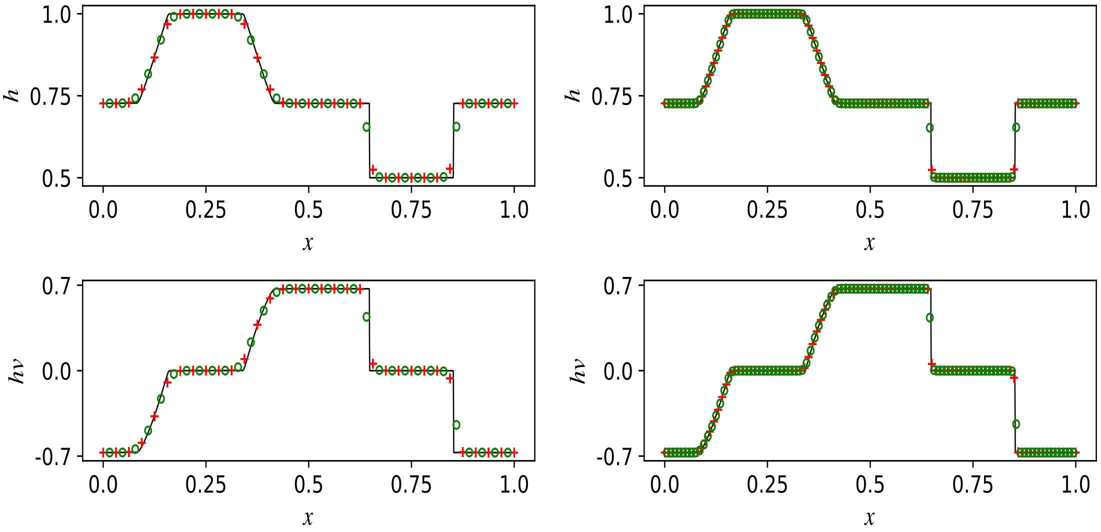


Figure 3.15: At the final time $T = 0.05$, results are shown for the P1L1 approach for the 6^{th}-order Active Flux scheme with 32 elements (*left*) and 100 elements (*right*) in the domain. The black line (——) represents the reference solution (Exact Riemann solver) with 5000 cells.

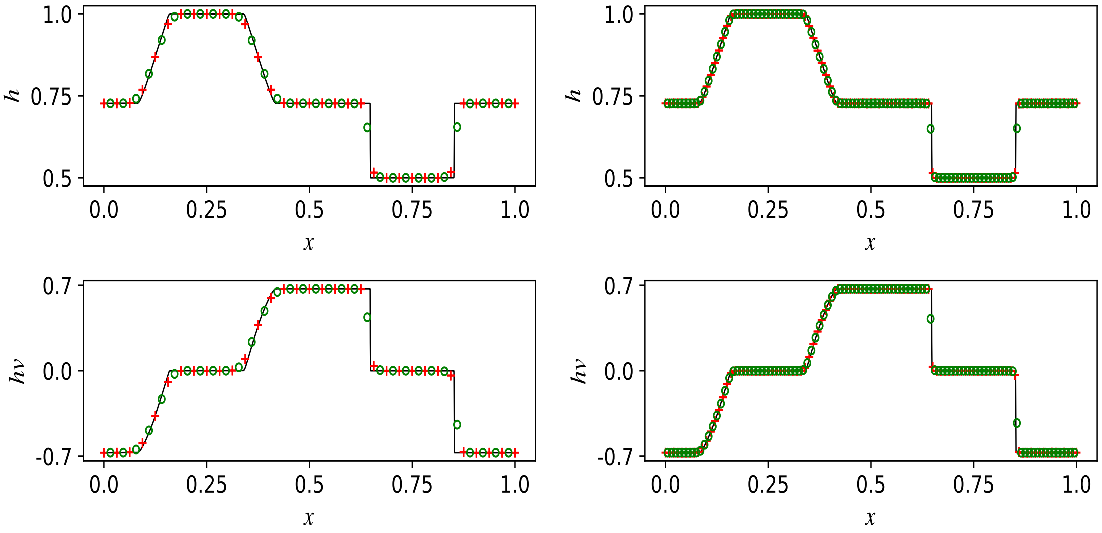


Figure 3.16: At the final time $T = 0.05$, results are shown for the P2L2 approach for the 6^{th}-order Active Flux scheme with 32 elements (*left*) and 100 elements (*right*) in the domain. The black line (——) represents the reference solution (Exact Riemann solver) with 5000 cells.

4 Higher Order Implicit Methods

So far, we have studied explicit numerical methods with higher-order accuracy. This chapter explores higher-order implicit approaches, focusing on implicit finite volume methods [34, 57] and Active Flux methods [3, 6] for solving hyperbolic conservation law problems. When employing explicit higher-order methods, the CFL number significantly decreases to ensure the stability of the scheme. Consequently, we must consider small time steps and thereby reducing the overall efficiency. However, by adopting implicit approaches, we can overcome this limitation. These implicit numerical methods use a narrow stencil to achieve higher-order accuracy. While explicit schemes update the solution directly to the subsequent time level, implicit schemes require the solution of a system of equations at each time step. The main challenges of implicit methods include solving a system of equations at each time step and implementing limiters. This chapter primarily concentrates on the implicit approach to linear hyperbolic problems.

4.1 Implicit finite volume methods

We initiate our discussion by examining higher-order accurate implicit finite volume (FV) methods with a narrow stencil. Typically, the numerical schemes presented in this section focus on the linear scalar advection equation of the form

$$\begin{aligned} \frac{\partial}{\partial t}u + a\frac{\partial}{\partial x}u &= 0, \qquad\qquad a \in \mathbb{R}, \\ u(x,0) &= u_0(x). \end{aligned} \tag{4.1}$$

In this context, $u = u\big(x,t\big)$ represents the conserved quantity with $u_0(x)$ denoting the given initial condition. For simplicity in explanation, we assume the velocity to be a positive real number ($a > 0$). Applying one-dimensional uniform spatial discretization with N_E elements, where cells denoted by $\left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]$, and index i ranges from 1 to N_E, with a cell size of Δx. The CFL number c is defined as $c = a\frac{\Delta t}{\Delta x}$, where $\Delta t = t^{n+1} - t^n$ represents the time step. The classical update formula for the finite volume schemes is as follows

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\Big(F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}\Big), \qquad\qquad i = 1, 2, 3, \ldots, N_E, \tag{4.2}$$

where, $u_i^n = u(x_i, t^n)$ is the cell averages defined as $u_i^n = \frac{1}{\Delta x}\displaystyle\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x,t^n)dx$ and $F_{i\pm\frac{1}{2}}$ are the numerical fluxes at the cell interfaces. The study on higher-order finite volume schemes is based on [34].

4.1.1 Third order scheme

We aim to achieve a third-order scheme by ensuring highly accurate numerical fluxes at the interfaces. Constructing a time-dependent quadratic polynomial p at the cell interface is crucial, and it must satisfy the following constraints

$$\frac{c}{\Delta t}\int_{t^n}^{t^n+\frac{\Delta t}{c}} p(t)dt = u_i^n, \qquad \frac{c}{\Delta t}\int_{t^{n+1}}^{t^{n+1}+\frac{\Delta t}{c}} p(t)dt = u_i^{n+1}, \qquad \frac{c}{\Delta t}\int_{t^{n+1}+\frac{\Delta t}{c}}^{t^{n+1}+\frac{2\Delta t}{c}} p(t)dt = u_{i-1}^{n+1}.$$

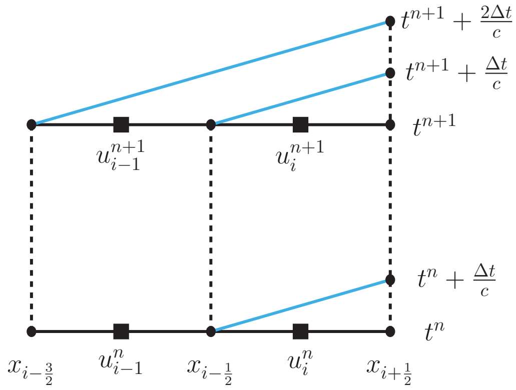


Figure 4.1: Characteristic information for 3^{rd} order scheme.

The numerical flux is computed at cell interface by $F_{i+\frac{1}{2}} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} ap(t)dt$ and has the form as in [34]

$$F_{i+\frac{1}{2}} = a\left[\left(\frac{c^2+3c-4}{6c}\right)u_i^{n+1} - \left(\frac{c^2-c}{6c}\right)u_{i-1}^{n+1} + \left(\frac{c+2}{3c}\right)u_i^n\right].$$

We compute the final update formula as follows,

$$\begin{aligned} u_i^{n+1} = u_i^n - \Bigg[\left(\frac{c^2+3c-4}{6}\right)u_i^{n+1} - \left(\frac{2c^2+2c-4}{6}\right)u_{i-1}^{n+1} + \left(\frac{c+2}{3}\right)u_i^n \\ + \left(\frac{c^2-c}{6}\right)u_{i-2}^{n+1} - \left(\frac{c+2}{3}\right)u_{i-1}^n\Bigg]. \end{aligned}$$

4.1.2 Fourth order scheme

We can calculate the fourth-order scheme by employing a similar approach to the one described above. To compute the numerical fluxes, we construct a time-dependent cubic polynomial p at the cell interface, subject to the following constraints

$$\begin{aligned} &\frac{c}{\Delta t}\int_{t^n}^{t^n+\frac{\Delta t}{c}} p(t)dt = u_i^n, &&\frac{c}{\Delta t}\int_{t^n+\frac{\Delta t}{c}}^{t^n+\frac{2\Delta t}{c}} p(t)dt = u_{i-1}^n, \\ &\frac{c}{\Delta t}\int_{t^{n+1}}^{t^{n+1}+\frac{\Delta t}{c}} p(t)dt = u_i^{n+1}, &&\frac{c}{\Delta t}\int_{t^{n+1}+\frac{\Delta t}{c}}^{t^{n+1}+\frac{2\Delta t}{c}} p(t)dt = u_{i-1}^{n+1}. \end{aligned}$$

After computing the more accurate fluxes across the cell interfaces, we apply the general finite volume update formula (4.2).

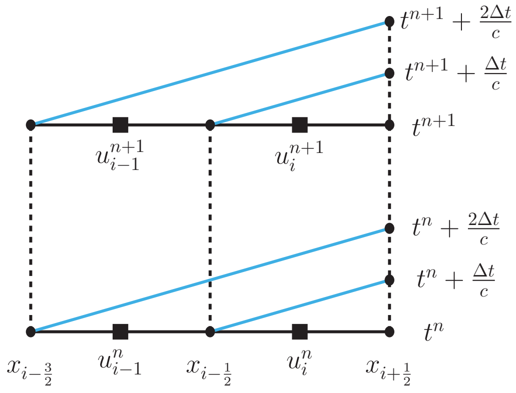


Figure 4.2: Characteristic information for 4^{th} order scheme.

We represent the flux in the following form as given in [34]

$$F_{i+\frac{1}{2}} = a\left[\left(\frac{-c^2+3c+10}{12c}\right)u_i^n + \left(\frac{c^2+3c+2}{12c}\right)u_{i-1}^n + \left(\frac{c^2+3c-10}{12c}\right)u_i^{n+1} + \left(\frac{-c^2+3c-2}{12c}\right)u_{i-1}^{n+1}\right].$$

We can express the final update formula as follows

$$u_i^{n+1} = u_i^n - \left[\left(\frac{-c^2+3c+10}{12}\right)u_i^n + \left(\frac{2c^2-8}{12}\right)u_{i-1}^n + \left(\frac{c^2+3c-10}{12}\right)u_i^{n+1} + \left(\frac{-2c^2+8}{12}\right)u_{i-1}^{n+1} - \left(\frac{c^2+3c+2}{12}\right)u_{i-2}^n + \left(\frac{c^2-3c+2}{12}\right)u_{i-2}^{n+1}\right].$$

To investigate the performance of the two methods, we analyze their behaviour using the standard initial condition proposed in [54]. The simulations run with 200 cells and a CFL number of approximately 6. As Figure 4.3 depicts, the fourth-order implicit finite volume (FV) method shows more oscillations. We expect these oscillations because all eigenvalues corresponding to the fourth-order scheme are equal to one as shown in Figure 4.4. Additionally, we analyze the convergence of the schemes by applying the smooth initial condition $u(x,0) = \sin(2\pi x)$. We start with 32 cells and refine the mesh to 512 cells. We outline the results obtained in this study in Table Table 4.1.

3^{rd} order FV			4^{th} order FV		
Cells	L^1- error	Order of convergence	Cells	L^1- error	Order of convergence
32	0.7283	-	32	0.0864	-
64	0.1367	2.41	64	0.0057	3.91
128	0.0182	2.90	128	0.0003	3.99
256	0.0023	2.98	256	2.26E-05	3.99
512	0.0003	3.00	512	1.41E-06	4.00

Table 4.1: The convergence results for the implicit finite volume methods.

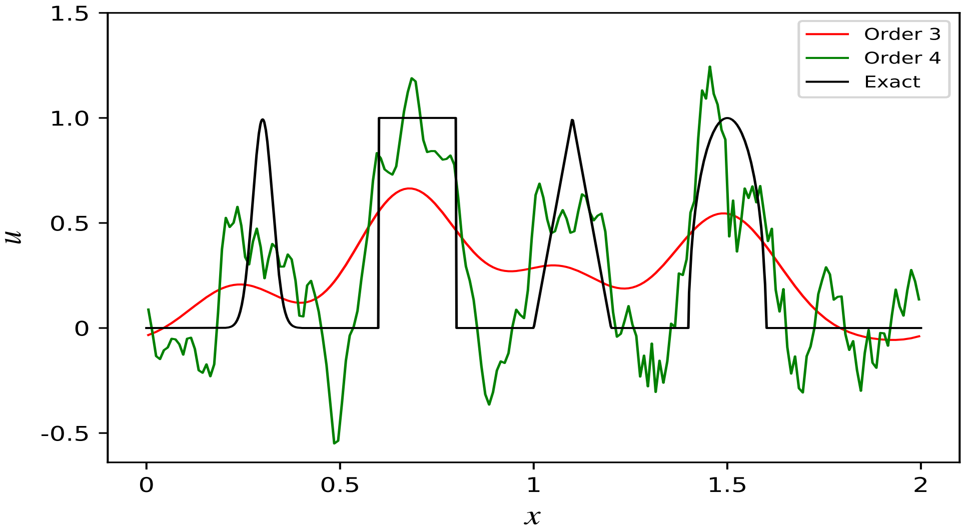


Figure 4.3: Comparison of two schemes using 200 cells with a CFL number of about 6. Results are shown at final time $T = 8$.

A detailed discussion on the stability of the two implicit finite volume methods can be found in [34]. Therefore, we write the implicit finite volume schemes in the following form

$$A\boldsymbol{u}^{n+1} = B\boldsymbol{u}^{n} \tag{4.3}$$

where $\boldsymbol{u}$ represents all degrees of freedom in the domain.

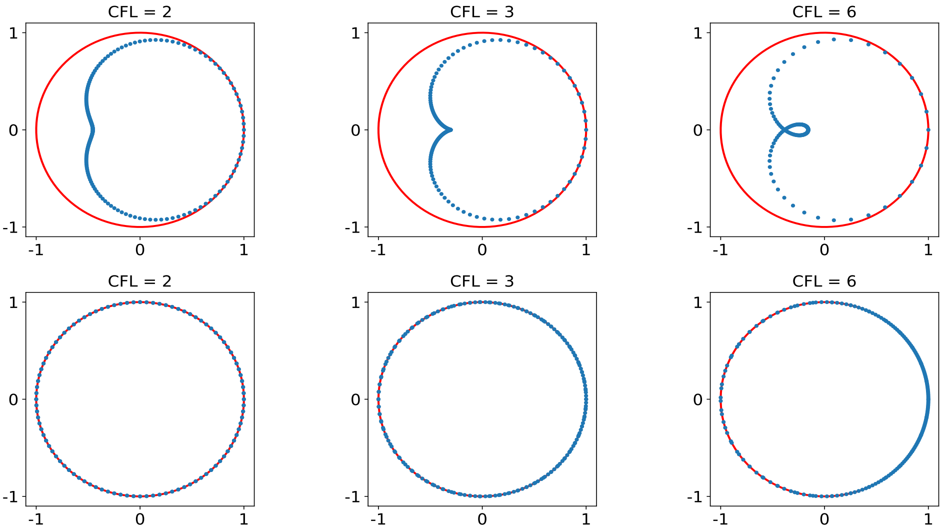


Figure 4.4: Spectrum of the $A^{-1}B$ matrix for 3^{rd} (*top row*) and 4^{th} (*bottom row*) order finite volume schemes for different CFL numbers. The reference unit circle is shown in red.

The stability of these finite volume schemes can be verified using Theorem 2.2 by analyzing the spectrum of the matrix $A^{-1}B$. Figure 4.4 illustrates the spectrum for some CFL numbers greater than 1. Based on these findings, we can conclude that the 3^{rd} and 4^{th}-order schemes remain stable for CFL > 1. Additionally, in the case of the 4^{th}-order finite volume method, all eigenvalues are equal to 1.

4.2 Implicit Active Flux methods

The Active Flux method employs different degrees of freedom than finite volume methods, as discussed in Chapter 3. Let's briefly recap the two degrees of freedom for the i^{th} cell:

1. Cell averages of the conserved quantity, denoted as $\bar{u}_i(t) \approx \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x,t)dx$.

2. Point values of the conserved quantities at the cell interface, denoted as $u_{i+\frac{1}{2}}(t) \approx u(x_{i+\frac{1}{2}}, t)$

Our starting point introduces the semi-discrete Active Flux methods, as outlined in [1, 2]. The spatial integration of the general hyperbolic conservation law $\partial_t u + \partial_x f(u) = 0$ yields the following expression

$$\frac{d}{dt}\bar{u}_i(t) + \frac{f\big(u_{i+\frac{1}{2}}(t)\big) - f\big(u_{i-\frac{1}{2}}(t)\big)}{\Delta x} = 0. \tag{4.4}$$

In line with the approach adopted throughout this chapter, we assume the flux function to be $f(u) = au$. To derive the third-order approximation for the spatial derivative, we first transform each element to the spatial reference element T_E using $\xi = \big(x - x_{i-\frac{1}{2}}\big)/\Delta x$, where $0 \le \xi \le 1$. Within T_E, we construct a polynomial $p(\xi)$ under the constraints

$$p_i(0) = u_{i-\frac{1}{2}}(t), \qquad p_i(1) = u_{i+\frac{1}{2}}(t), \qquad \int_0^1 p_i(\xi)d\xi = \bar{u}_i(t).$$

It takes the following form

$$p_i(\xi) = u_{i-\frac{1}{2}}(t) + \xi\left(-4u_{i-\frac{1}{2}}(t) + 6\bar{u}_i(t) - 2u_{i+\frac{1}{2}}(t)\right) + \xi^2\left(3u_{i-\frac{1}{2}}(t) - 6\bar{u}_i(t) + 3u_{i+\frac{1}{2}}(t)\right).$$

After constructing the spatial polynomial, we then take its derivative at the cell interface at $x_{i+\frac{1}{2}}$, providing a third-order approximation for the spatial derivative

$$\frac{d}{dt}u_{i+\frac{1}{2}}(t) = -a\frac{2u_{i-\frac{1}{2}}(t) - 6\bar{u}_i(t) + 4u_{i+\frac{1}{2}}(t)}{\Delta x}. \tag{4.5}$$

We applied more fundamental implicit numerical methods to solve the semi-discrete coupled ODE system represented in (4.4) and (4.5) with $a > 0$. Figure 4.5 illustrates the numerical results for the following classical implicit methods as in [6].

- Crank-Nicolson (CN) method (2^{nd} order)

- Radau IA method (3^{rd} order)

$$\begin{array}{c|cc} 0 & 1/4 & -1/4 \\ 2/3 & 1/4 & 5/12 \\ \hline & 1/4 & 3/4 \end{array}$$

- Radau IIA method (3^{rd} order)

$$\begin{array}{c|cc} 1/3 & 5/12 & -1/12 \\ 1 & 3/4 & 1/4 \\ \hline & 3/4 & 1/4 \end{array}$$

- DIRK method (3^{rd} order)

$$\begin{array}{c|cc} 1/2+\sqrt{3}/6 & 1/2+\sqrt{3}/6 & 0 \\ 1/2-\sqrt{3}/6 & -\sqrt{3}/3 & 1/2+\sqrt{3}/6 \\ \hline & 1/2 & 1/2 \end{array}$$

The convergence study follows a similar approach to the one we discussed for implicit finite volume methods.

	CN		Radau IA		Radau IIA		DIRK	
Cells	L^1- error	OC	L^1- error	OC	L^1- error	OC	L^1- error	OC
16	1.0050	-	0.1020	-	0.1020	-	0.3395	-
32	0.4214	1.25	0.0127	2.99	0.0127	2.99	0.0649	2.38
64	0.1133	1.89	0.0015	3.01	0.0015	3.01	0.0093	2.79
128	0.0294	1.94	1.89E-4	3.05	1.89E-4	3.05	0.0011	2.99
256	0.0072	2.01	2.34E-5	3.01	2.34E-5	3.01	1.46E-4	3.00

Table 4.2: The errors are calculated using the point values. "OC" stands for Order of convergence.

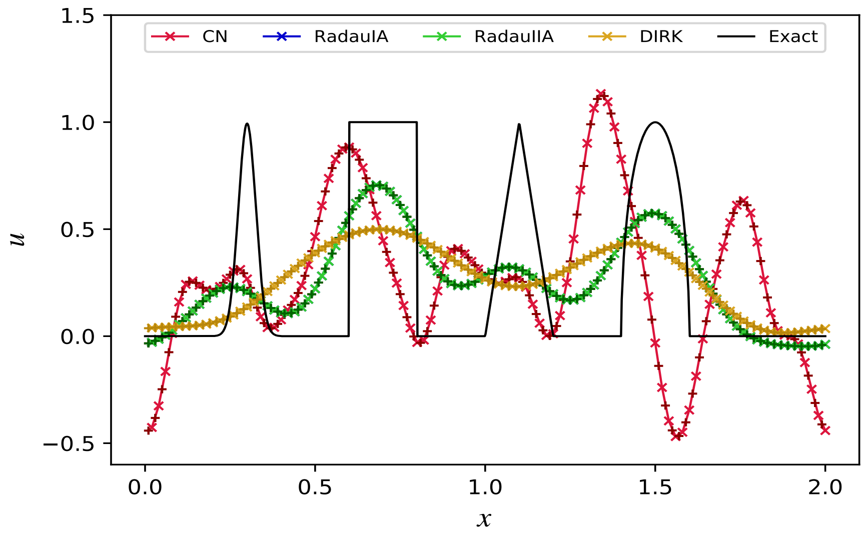


Figure 4.5: Results for semi-discrete Active Flux method solving by classical implicit methods using 100 cells with CFL number around 3. The symbol (+) indicates cell averages, and (×) represents point values. The final time is set at $T = 8$.

4.2.1 Single-step methods

The semi-discrete implicit Active Flux methods achieve maximum third-order accuracy because the quadratic polynomial provides information for updating the point values. Also, some of them employ multistep behaviour. These reasons direct our attention to a recently developed method proposed in [6]. Within this framework, a time-dependent polynomial is constructed along the cell interface using the characteristic information provided by Figure 4.6. This reconstruction polynomial involves both the averages and point values update.

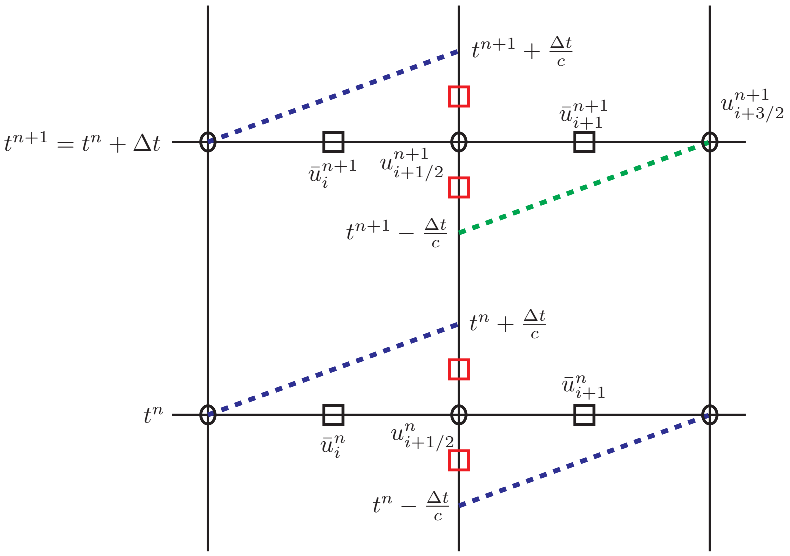

$t^{n+1} + \frac{\Delta t}{c}$
$u_{i+3/2}^{n+1}$
$\bar{u}_{i+1}^{n+1}$
$t^{n+1} = t^n + \Delta t$
$\bar{u}_i^{n+1}$
$u_{i+1/2}^{n+1}$
$t^{n+1} - \frac{\Delta t}{c}$
$t^n + \frac{\Delta t}{c}$
$\bar{u}_{i+1}^n$
t^n
$\bar{u}_i^n$
$u_{i+1/2}^n$
$t^n - \frac{\Delta t}{c}$

Figure 4.6: Dashed lines depict characteristic lines. Cell averages and point values are denoted by black squares and circles, respectively. Red squares represent the averages of the reconstruction polynomial in time. Both the black and red squares hold identical values

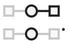
Following Figure 4.6, one can obtain six data points along the cell interface (which may exceed the limit $\left[t^n, t^{n+1}\right]$) by applying information from time levels t^n and t^{n+1}. We can select the necessary relations from the following six options to find the reconstruction polynomial in time, $u_{i+\frac{1}{2}}^{recon}(t)$, depending on its degree

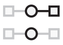
$$\begin{aligned}
u_{i+\frac{1}{2}}^{recon}(t^n) &= u_{i+\frac{1}{2}}^n, \\
u_{i+\frac{1}{2}}^{recon}(t^{n+1}) &= u_{i+\frac{1}{2}}^{n+1}, \\
\frac{c}{\Delta t} \int_{t^{n+1}}^{t^{n+1}+\frac{\Delta t}{c}} u_{i+\frac{1}{2}}^{recon}(t) dt &= \bar{u}_i^{n+1}, \\
\frac{c}{\Delta t} \int_{t^{n+1}-\frac{\Delta t}{c}}^{t^{n+1}} u_{i+\frac{1}{2}}^{recon}(t) dt &= \bar{u}_{i+1}^{n+1}, \\
\frac{c}{\Delta t} \int_{t^n}^{t^n+\frac{\Delta t}{c}} u_{i+\frac{1}{2}}^{recon}(t) dt &= \bar{u}_i^n, \\
\frac{c}{\Delta t} \int_{t^n-\frac{\Delta t}{c}}^{t^n} u_{i+\frac{1}{2}}^{recon}(t) dt &= \bar{u}_{i+1}^n.
\end{aligned}$$

In this approach, a reconstruction polynomial of up to degree five is generated to achieve sixth-order accuracy. Once the reconstruction polynomial is obtained, updates are performed in the following manner.

- Averages
$$\bar{u}_i^{n+1} = \bar{u}_i^n - \frac{\Delta t}{\Delta x}\left(\tilde{f}_{i+\frac{1}{2}} - \tilde{f}_{i-\frac{1}{2}}\right),$$
  with
$$\tilde{f}_{i+\frac{1}{2}} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(u_{i+\frac{1}{2}}^{recon}(t))dt.$$

- Point values
$$u_{i+\frac{3}{2}}^{n+1} = u_{i+\frac{1}{2}}^{recon}\left(t^{n+1} - \frac{\Delta t}{c}\right).$$

Next, we will discuss three examples using this approach. To simplify the representation of the stencil, we will consider a pictorial representation similar to the one shown in . We use boxes to denote the average values and circles to denote the point values in that representation. The symbols on the upper line correspond to the time level t^{n+1}, while those below are related to the time level t^n. To generate the reconstruction polynomial, we only consider highlighted degrees of freedom as shown in the symbols.

Example 1: Third-Order Scheme denoted by 

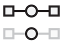
In the first example, we construct a third-order scheme using the stencil as shown in the pictorial representation. The reconstruction polynomial is of degree 2 and involves $u_{i+\frac{1}{2}}^n$, $u_{i+\frac{1}{2}}^{n+1}$, and $\bar{u}_{i+1}^{n+1}$. We summarize the necessary steps by considering the CFL number $c = \frac{a\Delta t}{\Delta x}$.

- Reconstruction polynomial in time at $x_{i+\frac{1}{2}}$,
$$\begin{aligned} u_{i+\frac{1}{2}}^{recon}(t) = u_{i+\frac{1}{2}}^n &+ \frac{(t-t^n)}{(3c-2)\Delta t}\Big(-2(3c^2-3c+1)u_{i+\frac{1}{2}}^{n+1} + 6c^2\bar{u}_{i+1}^{n+1} - 2(3c-1)u_{i+\frac{1}{2}}^n\Big) \\ &+ \frac{(t-t^n)^2}{(3c-2)\Delta t^2}\Big(3c(2c-1)u_{i+\frac{1}{2}}^{n+1} - 6c^2\bar{u}_{i+1}^{n+1} + 3cu_{i+\frac{1}{2}}^n\Big). \end{aligned}$$

- The point values update formula is as follows,
$$(3c-1)(c-1)u_{i+\frac{1}{2}}^{n+1} + 6c(1-c)\bar{u}_{i+1}^{n+1} + c(3c-2)u_{i+\frac{3}{2}}^{n+1} - u_{i+\frac{1}{2}}^n = 0.$$

- Numerical flux function,
$$\tilde{f}_{i+\frac{1}{2}} = \frac{a}{3c-2}\Big(-(c-1)^2u_{i+\frac{1}{2}}^{n+1} + c^2\bar{u}_{i+1}^{n+1} + (c-1)u_{i+\frac{1}{2}}^n\Big).$$

- We update the average values in the following manner
$$\begin{aligned} -c(c-1)^2u_{i-\frac{1}{2}}^{n+1} + (c-1)^2(c+2)\bar{u}_i^{n+1} + c(c-1)^2u_{i+\frac{1}{2}}^{n+1} - c^3\bar{u}_{i+1}^{n+1} \\ + c(c-1)u_{i-\frac{1}{2}}^n + (3c-2)\bar{u}_i^n - c(c-1)u_{i+\frac{1}{2}}^n = 0. \end{aligned}$$

Example 2: Fourth-Order Scheme denoted by 

Here, we summarize the updated formulas for the fourth-order scheme. The values $u^n_{i+\frac{1}{2}}$, $u^{n+1}_{i+\frac{1}{2}}$, $\bar{u}^{n+1}_i$, and $\bar{u}^{n+1}_{i+1}$ are used to generate the reconstruction polynomial of degree 3 in time.

- For the point values update,

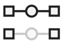
$$\begin{aligned}-c(c-1)(2c-1)\bar{u}^{n+1}_i + 2(c-1)(4c^2+c-1)u^{n+1}_{i+\frac{1}{2}} - c(c-1)(10c+7)\bar{u}^{n+1}_{i+1}&\\ +2c(2c^2-1)u^{n+1}_{i+\frac{3}{2}} - 2u^n_{i+\frac{1}{2}} = 0.&\end{aligned}$$

- Numerical flux function,

$$\begin{aligned}\tilde{f}_{i+\frac{1}{2}} = \frac{a}{4(2c^2-1)}\Big(&-2(c-1)^2(c+1)^2u^{n+1}_{i+\frac{1}{2}} + c^2(c+1)^2\bar{u}^{n+1}_{i+1}\\ &+c^2(c-1)^2\bar{u}^{n+1}_i + 2(c-1)(c+1)u^n_{i+\frac{1}{2}}\Big).\end{aligned}$$

- For the averages update,

$$\begin{aligned}c^3(c-1)^2\bar{u}^{n+1}_{i-1} - 2c(c-1)^2(c+1)^2u^{n+1}_{i-\frac{1}{2}} + 4(c-1)^2(c+1)^2\bar{u}^{n+1}_i&\\ +2c(c-1)^2(c+1)^2u^{n+1}_{i+\frac{1}{2}} - c^3(c+1)^2\bar{u}^{n+1}_{i+1} + 2c(c-1)(c+1)u^n_{i-\frac{1}{2}}&\\ -4(1-2c^2)\bar{u}^n_i - 2c(c-1)(c+1)u^n_{i+\frac{1}{2}} = 0.&\end{aligned}$$

Example 3: Fifth-Order Scheme denoted by 

In the fifth-order scheme, the values $\bar{u}^n_i$, $\bar{u}^n_{i+1}$, $\bar{u}^{n+1}_i$, $u^{n+1}_{i+\frac{1}{2}}$ and $\bar{u}^{n+1}_{i+1}$ are used to generate a reconstruction polynomial of degree 4 in time. For the simulations, we consider the following formulas.

- For the point values,

$$\begin{aligned}-c(c-1)^2(5c^2-5c-1)\bar{u}^{n+1}_i + 2c(c-1)(c+1)(10c^2-15c+2)u^{n+1}_{i+\frac{1}{2}}&\\ -(c+1)(25c^4-40c^3-4c^2+17c-4)\bar{u}^{n+1}_{i+1} + 2c(c-1)(c+1)(5c^2-2)u^{n+1}_{i+\frac{3}{2}}&\\ -c(c+1)(5c+1)\bar{u}^n_i + (c-1)^2(5c-4)\bar{u}^n_{i+1} = 0.&\end{aligned}$$

- Numerical flux function,

$$\begin{aligned}\tilde{f}_{i+\frac{1}{2}} = \frac{a}{4(5c^2-2)}\Big(&(c-1)(c+1)^2(c+2)\bar{u}^{n+1}_{i+1} - 2(c-2)(c-1)(c+1)(c+2)u^{n+1}_{i+\frac{1}{2}}\\ &+(c-2)(c-1)^2(c+1)\bar{u}^{n+1}_i - (c-2)(c-1)^2\bar{u}^n_{i+1} + (c+2)(c+1)^2\bar{u}^n_i\Big).\end{aligned}$$

- For the averages,

$$\begin{aligned}-c(c-2)(c-1)^2(c+1)\bar{u}^{n+1}_{i-1} + 2c(c-2)(c-1)(c+1)(c+2)u^{n+1}_{i-\frac{1}{2}}&\\ -2(c-2)(c+2)(3c^2-1)\bar{u}^{n+1}_i - 2c(c-2)(c-1)(c+1)(c+2)u^{n+1}_{i+\frac{1}{2}}&\\ +c(c-1)(c+1)^2(c+2)\bar{u}^{n+1}_{i+1} - c(c+1)^2(c+2)\bar{u}^n_{i-1}&\\ +2(c-2)(c-1)(c+1)(c+2)\bar{u}^n_i - c(c-2)(c-1)^2\bar{u}^n_{i+1} = 0.&\end{aligned}$$

We validate these schemes by considering the Jiang-Shu initial condition [44] with 100 cells and setting the CFL number to approximately 3. The convergence study begins with a smooth initial condition $u(x, 0) = \sin(x)$ using 16 cells and progressively refines the domain, reaching 512 cells for simulation. The results demonstrate the L^1-error computed based on the point values. As expected, the described schemes achieve the correct convergence rates, as shown in Figure 4.8. The third-order and fourth-order schemes guarantee stability for CFL > 1, while the fifth-order scheme ensures stability for CFL > 2. For further investigation in which combinations are stable, please refer to [6].

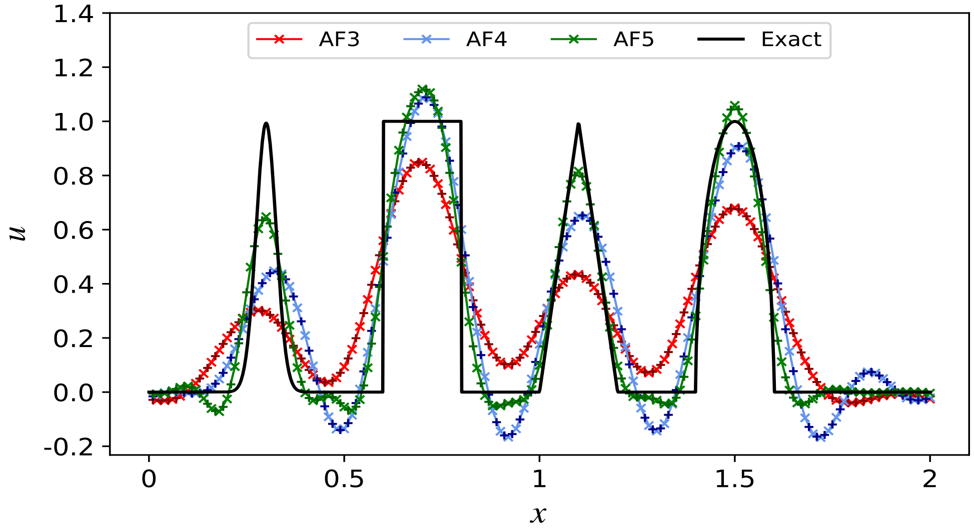


Figure 4.7: Results for single-step implicit Active Flux methods: (+) denotes cell averages, (×) represents point values, at the final time $T = 8$. Results of the third-order (□–○–□ □–○–□, —×—), fourth-order (□–○–□ □–○–□, —×—), and fifth-order (□–○–□ □–○–□, —×—) schemes are presented.

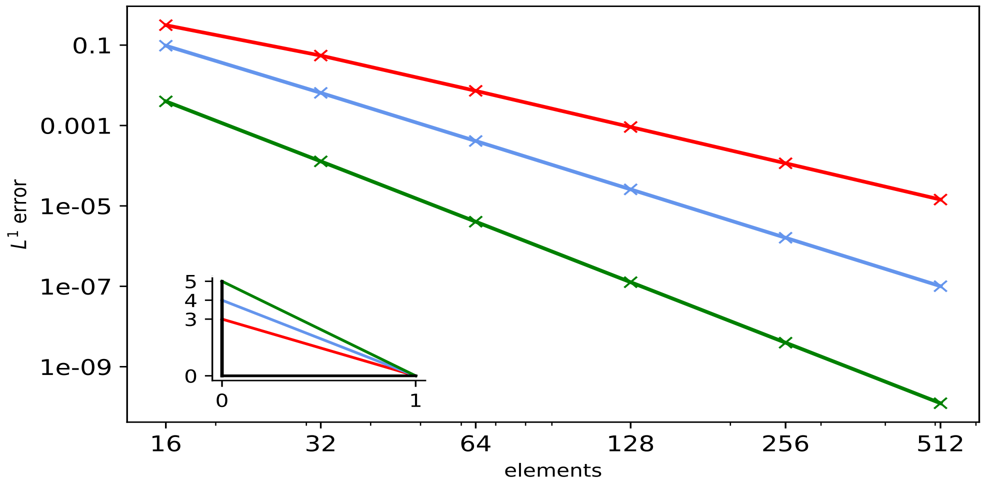


Figure 4.8: The L^1-errors are present for the third-order scheme —×—, the fourth-order scheme —×—, and the fifth-order scheme —×—.

For stability analysis, the single-step implicit Active Flux methods can be represented in the general form outlined in equation (4.3). The eigenvalues of the matrix $A^{-1}B$, as shown in Figure 4.9, satisfy the stability criteria specified in Theorem 2.2.

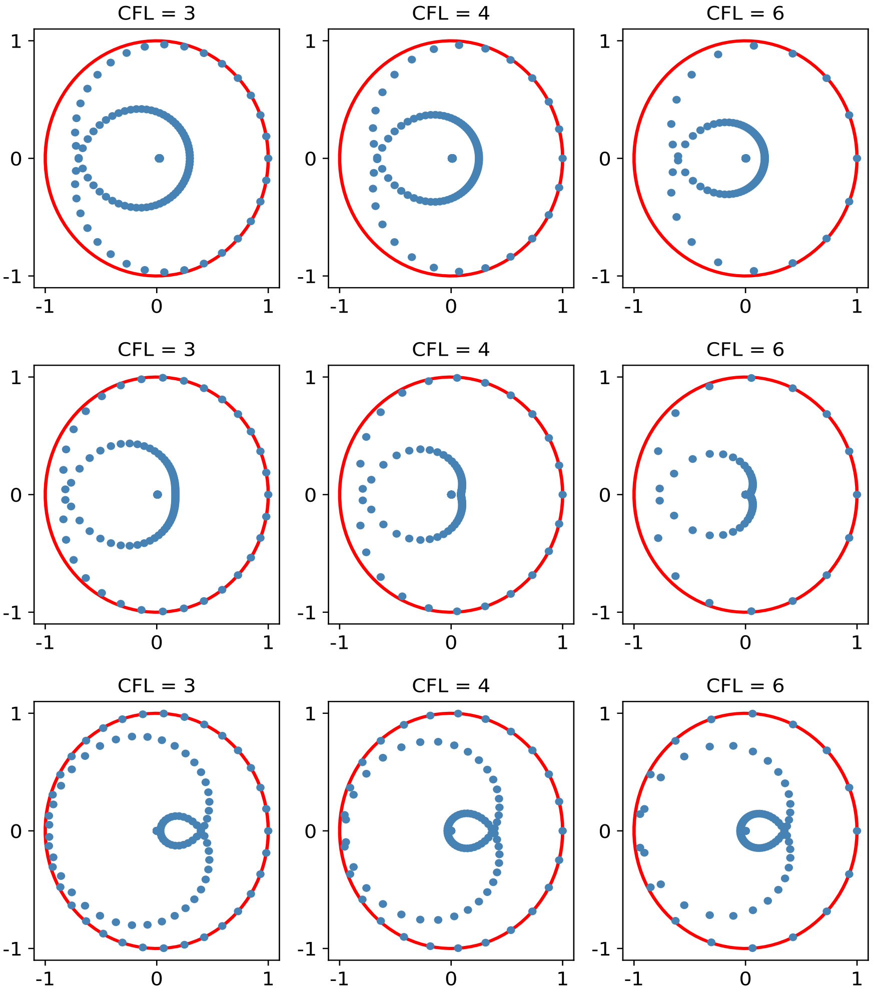


Figure 4.9: The spectrum of the matrix $A^{-1}B$ related to 3^{rd}-order (*top row*), 4^{th}-order (*middle row*) and 5^{th}-order (*bottom row*) single-step implicit Active Flux schemes for different CFL numbers.

4.2.2 Linear systems

Let's extend the single-step implicit Active Flux method to linear systems of hyperbolic problems. In this discussion, our focus is on a system characterized by two waves with different speeds. Let's consider the following form of the linear system

$$\boldsymbol{u}_t + A\boldsymbol{u}_x = 0 \tag{4.6}$$

where

$$\boldsymbol{u} = \boldsymbol{u}(x,t) = \begin{bmatrix} \rho \\ q \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & 1 \\ \tilde{c}^2 - v^2 & 2v \end{bmatrix},$$

with the given initial condition

$$\rho(x,0) = \begin{cases} 1, & 0.4 < x < 0.6 \\ 0, & \text{otherwise} \end{cases}, \qquad q(x,0) = 0.$$

We select parameters $\tilde{c} = 3.5$ and $v = 0.5$ to establish two distinct wave speeds, with waves propagating in opposing directions. The left-going and right-going waves have different CFL numbers, denoted by c_l and c_r respectively. Considering the eigenvectors r_1 and r_2 of the matrix A, we construct a matrix $R = \big(r_1|r_2\big)$. Then we introduce the characteristic variables $\boldsymbol{w} = R^{-1}\boldsymbol{u}$ to decoupled the system (4.6) into two linear advection equations. We have detailed the process of solving the linear advection equation in Section 4.2.1 with positive speed ($a > 0$). Likewise, we can proceed for the negative speed ($a < 0$). First, solve for the characteristic variables $\boldsymbol{w} = [\mathcal{W}1, \mathcal{W}2]^T$ then we can transform back to the original variables. The relationships concerning the averages and point value updates are summarized as follows. $\mathcal{W}1$ and $\mathcal{W}2$ represent the left-going and right-going characteristic waves, respectively.

Third-order scheme

- Averages

(←)

$$c_l^3\bar{\mathcal{W}}1_{i-1}^{n+1} - c_l(c_l+1)^2\mathcal{W}1_{i-\frac{1}{2}}^{n+1} - (c_l-2)(c_l+1)^2\bar{\mathcal{W}}1_{i}^{n+1} + c_l(c_l+1)^2\mathcal{W}1_{i+\frac{1}{2}}^{n+1}$$
$$-c_l(c_l+1)\mathcal{W}1_{i-\frac{1}{2}}^{n} - (3c_l+2)\bar{\mathcal{W}}1_{i}^{n} + c_l(c_l-1)\mathcal{W}1_{i+\frac{1}{2}}^{n} = 0.$$

(→)

$$-c_r(c_r-1)^2\mathcal{W}2_{i-\frac{1}{2}}^{n+1} + (c_r-1)^2(c_r+2)\bar{\mathcal{W}}2_{i}^{n+1} + c_r(c_r-1)^2\mathcal{W}2_{i+\frac{1}{2}}^{n+1} - c_r^3\bar{\mathcal{W}}2_{i+1}^{n+1}$$
$$+c_r(c_r-1)\mathcal{W}2_{i-\frac{1}{2}}^{n} + (3c_r-2)\bar{\mathcal{W}}2_{i}^{n} - c_r(c_r-1)\mathcal{W}2_{i+\frac{1}{2}}^{n} = 0.$$

- Point values

(←)

$$c_l(3c_l+2)\mathcal{W}1_{i-\frac{3}{2}}^{n+1} - 6c_l(c_l+1)\bar{\mathcal{W}}1_{i-1}^{n+1} + (3c_l+1)(c_l+1)\mathcal{W}1_{i-\frac{1}{2}}^{n+1} - \mathcal{W}1_{i-\frac{1}{2}}^{n} = 0.$$

(→)

$$(3c_r-1)(c_r-1)\mathcal{W}2_{i+\frac{1}{2}}^{n+1} + 6c_r(1-c_r)\bar{\mathcal{W}}2_{i+1}^{n+1} + c_r(3c_r-2)\mathcal{W}2_{i+\frac{3}{2}}^{n+1} - \mathcal{W}2_{i+\frac{1}{2}}^{n} = 0.$$

Fourth-order scheme

- Averages

(←)

$$c_l^3(1-c_l)^2\bar{\mathcal{W}}1_{i-1}^{n+1} - 2c_l(c_l+1)^2(1-c_l)^2\mathcal{W}1_{i-\frac{1}{2}}^{n+1} + 4(c_l+1)^2(1-c_l)^2\bar{\mathcal{W}}1_{i}^{n+1}$$
$$+2c_l(c_l+1)^2(1-c_l)^2\mathcal{W}1_{i+\frac{1}{2}}^{n+1} - c_l^3(c_l+1)^2\bar{\mathcal{W}}1_{i+1}^{n+1} + 2c_l(c_l+1)(c_l-1)\mathcal{W}1_{i-\frac{1}{2}}^{n}$$
$$-4(1-2c_l^2)\bar{\mathcal{W}}1_{i}^{n} - 2c_l(c_l+1)(c_l-1)\mathcal{W}1_{i+\frac{1}{2}}^{n} = 0.$$

$(\rightarrow)$

$$\begin{aligned}
&c_r^3(c_r-1)^2\bar{\mathcal{W}}2_{i-1}^{n+1} - 2c_r(c_r-1)^2(c_r+1)^2\mathcal{W}2_{i-\frac{1}{2}}^{n+1} + 4(c_r-1)^2(c_r+1)^2\bar{\mathcal{W}}2_i^{n+1}\\
&+2c_r(c_r-1)^2(c_r+1)^2\mathcal{W}2_{i+\frac{1}{2}}^{n+1} - c_r^3(c_r+1)^2\bar{\mathcal{W}}2_{i+1}^{n+1} + 2c_r(c_r-1)(c_r+1)\mathcal{W}2_{i-\frac{1}{2}}^{n}\\
&\qquad -4(1-2c_r^2)\bar{\mathcal{W}}2_i^n - 2c_r(c_r-1)(c_r+1)\mathcal{W}2_{i+\frac{1}{2}}^n = 0.
\end{aligned}$$

- Point values

$(\leftarrow)$

$$\begin{aligned}
&-2c_l(2c_l^2-1)\mathcal{W}1_{i-\frac{3}{2}}^{n+1} + c_l(c_l+1)(10c_l-7)\bar{\mathcal{W}}1_{i-1}^{n+1} - 2(c_l+1)(4c_l^2-c_l-1)\mathcal{W}1_{i-\frac{1}{2}}^{n+1}\\
&\qquad +c_l(c_l+1)(2c_l+1)\bar{\mathcal{W}}1_i^{n+1} - 2\mathcal{W}1_{i-\frac{1}{2}}^n = 0.
\end{aligned}$$

$(\rightarrow)$

$$\begin{aligned}
&-c_r(c_r-1)(2c_r-1)\bar{\mathcal{W}}2_i^{n+1} + 2(c_r-1)(4c_r^2+c_r-1)\mathcal{W}2_{i+\frac{1}{2}}^{n+1}\\
&-c_r(c_r-1)(10c_r+7)\bar{\mathcal{W}}2_{i+1}^{n+1} + 2c_r(2c_r^2-1)\mathcal{W}2_{i+\frac{3}{2}}^{n+1} - 2\mathcal{W}2_{i+\frac{1}{2}}^n = 0.
\end{aligned}$$

Fifth-order scheme

- Averages

$(\leftarrow)$

$$\begin{aligned}
&-c_l(c_l+1)(1-c_l)^2(c_l-2)\bar{\mathcal{W}}1_{i-1}^{n+1} + 2c_l(c_l+2)(c_l+1)(c_l-1)(c_l-2)\mathcal{W}1_{i-\frac{1}{2}}^{n+1}\\
&-2(c_l+2)(c_l-2)(3c_l^2-1)\bar{\mathcal{W}}1_i^{n+1} - 2c_l(c_l+2)(c_l+1)(c_l-1)(c_l-2)\mathcal{W}1_{i+\frac{1}{2}}^{n+1}\\
&\qquad +c_l(c_l+2)(c_l+1)^2(c_l-1)\bar{\mathcal{W}}1_{i+1}^{n+1} - c_l(c_l+2)(c_l+1)^2\bar{\mathcal{W}}1_{i-1}^n\\
&\qquad +2(c_l+2)(c_l+1)(c_l-1)(c_l-2)\bar{\mathcal{W}}1_i^n - c_l(1-c_l)^2(c_l-2)\bar{\mathcal{W}}1_{i+1}^n = 0.
\end{aligned}$$

$(\rightarrow)$

$$\begin{aligned}
&-c_r(c_r-2)(c_r-1)^2(c_r+1)\bar{\mathcal{W}}2_{i-1}^{n+1} + 2c_r(c_r-2)(c_r-1)(c_r+1)(c_r+2)\mathcal{W}2_{i-\frac{1}{2}}^{n+1}\\
&-2(c_r-2)(c_r+2)(3c_r^2-1)\bar{\mathcal{W}}2_i^{n+1} - 2c_r(c_r-2)(c_r-1)(c_r+1)(c_r+2)\mathcal{W}2_{i+\frac{1}{2}}^{n+1}\\
&\qquad +c_r(c_r-1)(c_r+1)^2(c_r+2)\bar{\mathcal{W}}2_{i+1}^{n+1} - c_r(c_r+1)^2(c_r+2)\bar{\mathcal{W}}2_{i-1}^n\\
&\qquad +2(c_r-2)(c_r-1)(c_r+1)(c_r+2)\bar{\mathcal{W}}2_i^n - c_r(c_r-2)(c_r-1)^2\bar{\mathcal{W}}2_{i+1}^n = 0.
\end{aligned}$$

- Point values

$(\leftarrow)$

$$\begin{aligned}
&-2c_l(c_l+1)(c_l-1)(5c_l^2-2)\mathcal{W}1_{i-\frac{3}{2}}^{n+1} + (c_l-1)(25c_l^4+40c_l^3-4c_l^2-17c_l-4)\bar{\mathcal{W}}1_{i-1}^{n+1}\\
&-2c_l(c_l+1)(c_l-1)(10c_l^2+15c_l+2)\mathcal{W}1_{i-\frac{1}{2}}^{n+1} + c_l(c_l+1)^2(5c_l^2+5c_l-1)\bar{\mathcal{W}}1_i^{n+1}\\
&\qquad -(c_l+1)^2(5c_l+4)\bar{\mathcal{W}}1_{i-1}^n + c_l(c_l-1)(5c_l-1)\bar{\mathcal{W}}1_i^n = 0.
\end{aligned}$$

$(\rightarrow)$

$$\begin{aligned}-c_r(c_r-1)^2(5c_r^2-5c_r-1)\bar{\mathcal{W}}2_i^{n+1}+2c_r(c_r-1)(c_r+1)(10c_r^2-15c_r+2)\mathcal{W}2_{i+\frac{1}{2}}^{n+1}\\-(c_r+1)(25c_r^4-40c_r^3-4c_r^2+17c_r-4)\bar{\mathcal{W}}2_{i+1}^{n+1}+2c_r(c_r-1)(c_r+1)(5c_r^2-2)\mathcal{W}2_{i+\frac{3}{2}}^{n+1}\\-c_r(c_r+1)(5c_r+1)\bar{\mathcal{W}}2_i^{n}+(c_r-1)^2(5c_r-4)\bar{\mathcal{W}}2_{i+1}^{n}=0.\end{aligned}$$

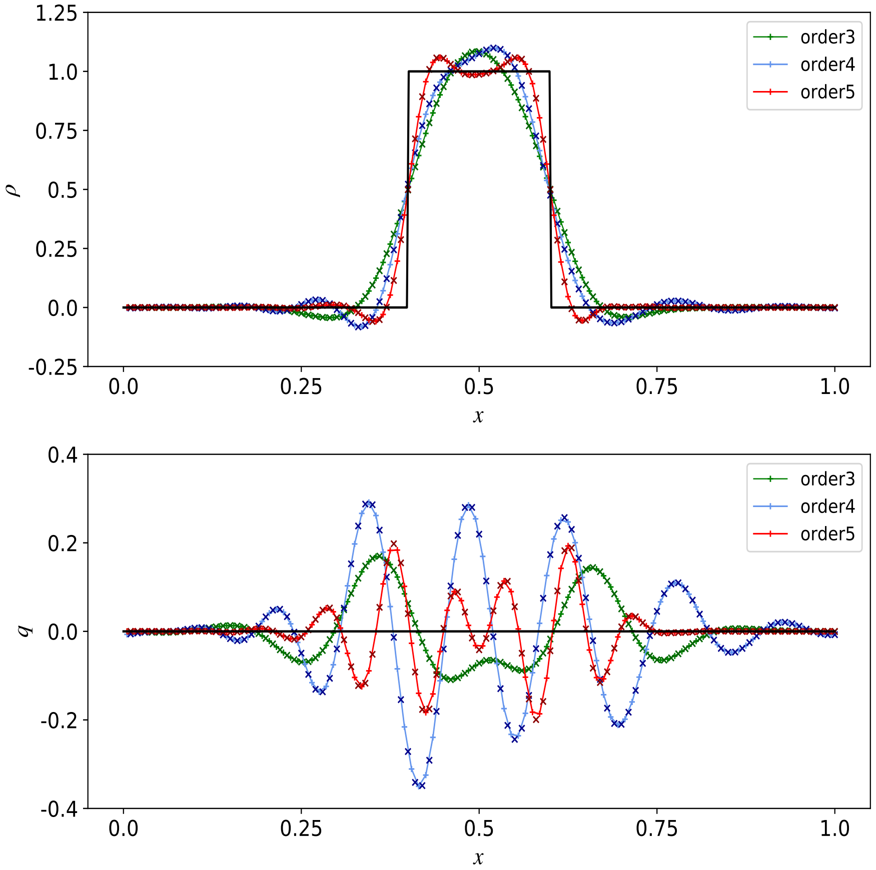


Figure 4.10: The results are displayed at the final time $T = 1$. The domain consists of 100 cells. The left-going wave has a CFL number 3, while the right-going wave has a CFL number 4. The cell averages are denoted by $(+)$ and the point values are indicated by $(\times)$.

4.3 Modified Implicit Active Flux Methods (MIAF)

In this section, we present a novel approach for updating point values in an alternative manner. In the previous configuration, we updated point values and averages using a single time-dependent reconstruction polynomial along the cell interface. In our new approach, we employ two distinct reconstruction polynomials: one for updating cell averages and the other for updating point values. The term 'Modified Implicit Active Flux method' is applied to this approach because of its broader scope, which enables the generation of a variety of higher-order implicit schemes compared to the previous approach. Our goal is to introduce the concept of the Modified Implicit Active Flux method and validate selected schemes for scalar linear hyperbolic problems.

- The averaged values can be updated using the finite volume update formula,

  $$\bar{u}_i^{n+1} = \bar{u}_i^n - \frac{\Delta t}{\Delta x}\Big(F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}\Big), \qquad i = 1, 2, 3, \ldots, N_E,$$

  where the flux calculation at cell interfaces involves the reconstruction polynomial created in Section 4.2.1. Hence, the final average update formula remains unchanged.

- Let's concentrate on the specifics of the proposed approach for updating the point values. Firstly, we'll integrate the conservation law provided in (4.1) in time at the cell interface $x_{i+\frac{1}{2}}$,

  $$\int_{t^n}^{t^{n+1}} \frac{\partial}{\partial t} u(x_{i+\frac{1}{2}}, t)dt + a\int_{t^n}^{t^{n+1}} \frac{\partial}{\partial x} u(x_{i+\frac{1}{2}}, t)dt = 0.$$

  Simplifying the first term,

  $$u_{i+\frac{1}{2}}^{n+1} = u_{i+\frac{1}{2}}^{n} - a\int_{t^n}^{t^{n+1}} \frac{\partial}{\partial x} u(x_{i+\frac{1}{2}}, t)dt. \tag{4.7}$$

  Our objective is to approximate the time integration that appears in (4.7) to ensure the preservation of higher-order accuracy in the scheme. To achieve this, we adopt a two-step approach, which we will discuss next. We conduct all computations in the space-time reference coordinate system for simplicity. The coordinate transformations can be written as, $\xi = (x - x_{i-\frac{1}{2}})/\Delta x$ and $\tau = (t - t^n)/\Delta t$.

The reconstruction polynomial we establish in Section 4.2.1 remains unchanged for updating the averaged values. However, when it comes to updating the point values, we generate a new time-dependent reconstruction polynomial at the cell interface using derivative values. As mentioned before, the creation of this new reconstruction polynomial involves two steps.

- ***step 1*** : We consider both cell averages and point values to govern the space-dependent polynomials at time levels t^n and t^{n+1}. We determine the degree of these polynomials based on the desired convergence order of the scheme. In higher-order schemes, we incorporate information from neighboring cells when creating these spatial polynomials. Our objective is to calculate spatial derivatives at specific points. This calculation depends on the stencil chosen for updating the point values.

- ***step 2***: We generate a time-dependent reconstruction polynomial along the cell interface using the spatial-derivative values computed in *step 1*. Guided by the characteristic information of the problem, we derive this polynomial by applying interpolation techniques.

Next, we examine the process of the Modified Implicit Active Flux method in detail, focusing on selected examples. For each problem, we walk through the step-by-step application of the two-step process outlined above. As before, we consider the conservation law given in (4.1) with positive speed ($a > 0$).

4.3.1 Third-order scheme

We apply quadratic spatial and temporal polynomials to achieve third-order accurate schemes. We maintain the same procedure discussed in Section 4.2.1 for updating cell averages because the modification only applies to the point values update. Therefore, we can directly write the cell averages update formula as follows

$$\begin{aligned}-c(c-1)^2u^{n+1}_{i-\frac{1}{2}}+(c-1)^2(c+2)\bar{u}^{n+1}_i+c(c-1)^2u^{n+1}_{i+\frac{1}{2}}-c^3\bar{u}^{n+1}_{i+1}\\ +c(c-1)u^n_{i-\frac{1}{2}}+(3c-2)\bar{u}^n_i-c(c-1)u^n_{i+\frac{1}{2}}=0.\end{aligned} \tag{4.8}$$

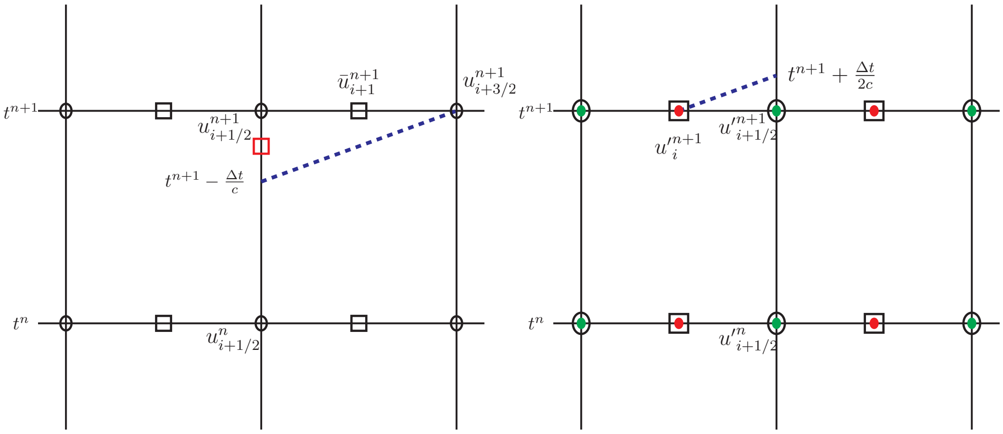


Figure 4.11: On the left side, the figure shows the stencil related to the cell average update, while the right side depicts the stencil related to the point value update. We display the cell averages (□) and the point values (○). Additionally, the red circles (●) represent the spatial derivative values at the cell centers, and the green circles (●) represent the spatial derivative values at the cell interfaces.

Step 1

We outline the procedure for constructing the space-dependent quadratic polynomial $p^n_i(\xi)$, for the i^{th} cell at time level t^n. There we consider the values $u^n_{i-\frac{1}{2}}, \bar{u}^n_i, u^n_{i+\frac{1}{2}}$. Analogously, we can construct the space-dependent polynomial at time level t^{n+1}. The following constraints are imposed

$$p^n_i(0)=u^n_{i-\frac{1}{2}}, \qquad p^n_i(1)=u^n_{i+\frac{1}{2}}, \qquad \int_0^1 p^n_i(\xi)d\xi=\bar{u}^n_i.$$

The desired polynomial and its spatial derivative, represented in the reference coordinates, are provided as follows

$$p^n_i(\xi)=u^n_{i-\frac{1}{2}}\left(3\xi^2-4\xi+1\right)+\bar{u}^n_i\left(6\xi-6\xi^2\right)+u^n_{i+\frac{1}{2}}\left(3\xi^2-2\xi\right),$$

$$\frac{\partial}{\partial\xi}p^n_i(\xi)=u^n_{i-\frac{1}{2}}(6\xi-4)+\bar{u}^n_i(6-12\xi)+u^n_{i+\frac{1}{2}}(6\xi-2).$$

At this point, we introduce new variables to approximate the derivative of the conserved quantity at certain places (see Figure 4.11)

$$\begin{aligned}
\frac{\partial}{\partial x} u\left(x_{i+\frac{1}{2}}, t^n\right) &\approx \frac{1}{\Delta x}\frac{\partial}{\partial \xi} p_i^n(\xi)\Big|_{\xi=1} &&= u'^{n}_{i+\frac{1}{2}},\\
\frac{\partial}{\partial x} u\left(x_{i+\frac{1}{2}}, t^{n+1}\right) &\approx \frac{1}{\Delta x}\frac{\partial}{\partial \xi} p_i^{n+1}(\xi)\Big|_{\xi=1} &&= u'^{n+1}_{i+\frac{1}{2}},\\
\frac{\partial}{\partial x} u\left(x_{i}, t^{n+1}\right) &\approx \frac{1}{\Delta x}\frac{\partial}{\partial \xi} p_i^{n+1}(\xi)\Big|_{\xi=\frac{1}{2}} &&= u'^{n+1}_{i}.
\end{aligned}$$

Step 2

Next, we can construct the time-dependent quadratic reconstruction polynomial using the derivative values $u'^{n}_{i+\frac{1}{2}}$, $u'^{n+1}_{i+\frac{1}{2}}$, and u'^{n+1}_{i}.

Let

$$q(\tau) = a_0 + a_1\tau + a_2\tau^2$$

be the time-dependent reconstruction polynomial along the cell interface. Our goal is to approximate the integral,

$$I = \int_{t^n}^{t^{n+1}} \frac{\partial}{\partial x} u\left(x_{i+\frac{1}{2}}, t\right) dt$$

as described in equation (4.7) using this reconstruction polynomial. To achieve this, we employ the following interpolation strategy

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & (1+\frac{1}{2c}) & (1+\frac{1}{2c})^2 \end{pmatrix}}_{V} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} u'^{n}_{i+\frac{1}{2}} \\ u'^{n+1}_{i+\frac{1}{2}} \\ u'^{n+1}_{i} \end{pmatrix}. \tag{4.9}$$

We simplify the integration as,

$$\begin{aligned}
I &= \int_{t^n}^{t^{n+1}} \frac{\partial}{\partial x} u\left(x_{i+\frac{1}{2}}, t\right) dt\\
&\approx \frac{\Delta t}{\Delta x}\int_0^1 q(\tau)d\tau\\
&= \frac{\Delta t}{\Delta x}\left[a_0\tau + \frac{a_1}{2}\tau^2 + \frac{a_2}{3}\tau^3\right]_0^1\\
&= \frac{\Delta t}{\Delta x}\left(1, \frac{1}{2}, \frac{1}{3}\right)\begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}.
\end{aligned}$$

Substituting from (4.9),

$$I = \frac{\Delta t}{\Delta x}\left(1, \frac{1}{2}, \frac{1}{3}\right) V^{-1} \begin{pmatrix} u'^{n}_{i+\frac{1}{2}} \\ u'^{n+1}_{i+\frac{1}{2}} \\ u'^{n+1}_{i} \end{pmatrix}.$$

Introducing the weights $W = [w_0, w_1, w_2] = \left(1, \frac{1}{2}, \frac{1}{3}\right) V^{-1}$ we can write,

$$I = \frac{\Delta t}{\Delta x}\left[w_0 u'^{n}_{i+\frac{1}{2}} + w_1 u'^{n+1}_{i+\frac{1}{2}} + w_2 u'^{n+1}_{i}\right].$$

Insert it in (4.7),

$$u_{i+\frac{1}{2}}^{n+1} = u_{i+\frac{1}{2}}^{n} - \underbrace{a\frac{\Delta t}{\Delta x}}_{c}\left[w_0 {u'}_{i+\frac{1}{2}}^{n} + w_1 {u'}_{i+\frac{1}{2}}^{n+1} + w_2 {u'}_{i}^{n+1}\right].$$

We can express the modified point value update formula as follows

$$u_{i+\frac{1}{2}}^{n+1} - u_{i+\frac{1}{2}}^{n} + c\left[w_0 {u'}_{i+\frac{1}{2}}^{n} + w_1 {u'}_{i+\frac{1}{2}}^{n+1} + w_2 {u'}_{i}^{n+1}\right] = 0. \tag{4.10}$$

The averaged values update formula in (4.8), combined with the point values update in (4.10), forms an implicit Active Flux scheme of order 3.

4.3.2 Fourth-order scheme

In this section, we provide a detailed exploration of all the crucial aspects of the fourth-order accurate Modified Implicit Active Flux scheme.

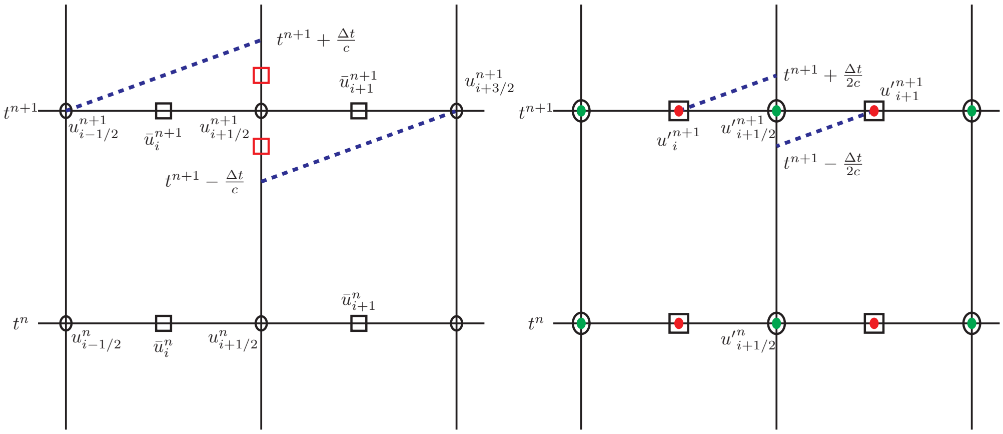


Figure 4.12: On the left side, the figure shows the stencil related to the cell average update □–○–□, while the right side depicts the stencil related to the point value update □–○–□. We display the cell averages (□) and the point values (○). Additionally, the red circles (🔴) represent the spatial derivative values at the cell centers, and the green circles (🟢) represent the spatial derivative values at the cell interfaces.

Initially, we can proceed by directly formulating the update formula for the cell averages, as discussed in Section 4.2.1

$$\begin{aligned} &c^3(c-1)^2\bar{u}_{i-1}^{n+1} - 2c(c-1)^2(c+1)^2 u_{i-\frac{1}{2}}^{n+1} + 4(c-1)^2(c+1)^2\bar{u}_{i}^{n+1} \\ &+2c(c-1)^2(c+1)^2 u_{i+\frac{1}{2}}^{n+1} - c^3(c+1)^2\bar{u}_{i+1}^{n+1} + 2c(c-1)(c+1)u_{i-\frac{1}{2}}^{n} \\ &\qquad -4(1-2c^2)\bar{u}_{i}^{n} - 2c(c-1)(c+1)u_{i+\frac{1}{2}}^{n} = 0. \end{aligned} \tag{4.11}$$

Next, let's concentrate on updating the point values using the new approach, ensuring that the overall scheme maintains fourth-order accuracy.

Step 1

The space-dependent polynomial of degree 3, denoted by $p_i^n(\xi)$ for the i^{th} cell at time level t^n, is governed under the following constraints. We consider the values $u_{i-\frac{1}{2}}^n, \bar{u}_i^n, u_{i+\frac{1}{2}}^n, \bar{u}_{i+1}^n$. It's worth noting that the usual bounds for ξ $(0 \leq \xi \leq 1)$ may be exceeded when dealing with neighboring cells. Similarly, we construct the space-dependent polynomial at time level t^{n+1} and denote it $p_i^{n+1}(\xi)$.

$$p_i^n(\xi=0) = u_{i-\frac{1}{2}}^n, \quad p_i^n(\xi=1) = u_{i+\frac{1}{2}}^n, \quad \int_{\xi=0}^{\xi=1} p_i^n(\xi) d\xi = \bar{u}_i^n, \quad \int_{\xi=1}^{\xi=2} p_i^n(\xi) d\xi = \bar{u}_{i+1}^n.$$

The polynomial $p_i^n(\xi)$ and its spatial derivative can be expressed in the following form,

$$\begin{aligned} p_i^n(\xi) = {} & u_{i-\frac{1}{2}}^n \left(1 - 5\xi + 6\xi^2 - 2\xi^3\right) + \bar{u}_i^n \left(\frac{17}{2}\xi - \frac{27}{2}\xi^2 + 5\xi^3\right) \\ & + u_{i+\frac{1}{2}}^n \left(-4\xi + 9\xi^2 - 4\xi^3\right) + \bar{u}_{i+1}^n \left(\frac{1}{2}\xi - \frac{3}{2}\xi^2 + \xi^3\right), \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial \xi} p_i^n(\xi) = {} & u_{i-\frac{1}{2}}^n \left(-5 + 12\xi - 6\xi^2\right) + \bar{u}_i^n \left(\frac{17}{2} - 27\xi + 15\xi^2\right) \\ & + u_{i+\frac{1}{2}}^n \left(-4 + 18\xi - 12\xi^2\right) + \bar{u}_{i+1}^n \left(\frac{1}{2} - 3\xi + 3\xi^2\right). \end{aligned}$$

Introducing the variables, ${u'}_{i+\frac{1}{2}}^n$, ${u'}_{i+1}^{n+1}$,${u'}_{i+\frac{1}{2}}^{n+1}$, and ${u'}_i^{n+1}$,

$$\begin{aligned} \frac{\partial}{\partial x} u(x_{i+\frac{1}{2}}, t^n) &\approx \frac{1}{\Delta x} \frac{\partial}{\partial \xi} p_i^n(\xi)\Big|_{\xi=1} &= {u'}_{i+\frac{1}{2}}^n, \\ \frac{\partial}{\partial x} u(x_{i+1}, t^{n+1}) &\approx \frac{1}{\Delta x} \frac{\partial}{\partial \xi} p_i^{n+1}(\xi)\Big|_{\xi=\frac{3}{2}} &= {u'}_{i+1}^{n+1}, \\ \frac{\partial}{\partial x} u(x_{i+\frac{1}{2}}, t^{n+1}) &\approx \frac{1}{\Delta x} \frac{\partial}{\partial \xi} p_i^{n+1}(\xi)\Big|_{\xi=1} &= {u'}_{i+\frac{1}{2}}^{n+1}, \\ \frac{\partial}{\partial x} u(x_i, t^{n+1}) &\approx \frac{1}{\Delta x} \frac{\partial}{\partial \xi} p_i^{n+1}(\xi)\Big|_{\xi=\frac{1}{2}} &= {u'}_i^{n+1}, \end{aligned}$$

we can formulate the time-dependent cubic polynomial along the cell interface $x_{i+\frac{1}{2}}$.

Step 2

Let

$$q(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 \tag{4.12}$$

be the cubic reconstruction polynomial in time along the cell interface. Using interpolation techniques allows us to find the integral in (4.7)

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & (1-\frac{1}{2c}) & (1-\frac{1}{2c})^2 & (1-\frac{1}{2c})^3 \\ 1 & 1 & 1 & 1 \\ 1 & (1+\frac{1}{2c}) & (1+\frac{1}{2c})^2 & (1+\frac{1}{2c})^3 \end{pmatrix}}_{V} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} {u'}_{i+\frac{1}{2}}^n \\ {u'}_{i+1}^{n+1} \\ {u'}_{i+\frac{1}{2}}^{n+1} \\ {u'}_i^{n+1} \end{pmatrix}. \tag{4.13}$$

Compute the integral presented in (4.7), considering the cubic polynomial defined in (4.12).

$$\begin{aligned} I = &\int_{t^n}^{t^{n+1}} \frac{\partial}{\partial x} u(x_{i+\frac{1}{2}}, t) dt \\ \approx &\frac{\Delta t}{\Delta x} \int_0^1 q(\tau) d\tau \\ = &\frac{\Delta t}{\Delta x} \left[a_0 \tau + \frac{a_1}{2} \tau^2 + \frac{a_2}{3} \tau^3 + \frac{a_3}{4} \tau^4 \right]_0^1 \\ = &\frac{\Delta t}{\Delta x} \left(1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4} \right) \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_4 \end{pmatrix}. \end{aligned}$$

Substituting from (4.13),

$$I = \frac{\Delta t}{\Delta x} \left(1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4} \right) V^{-1} \begin{pmatrix} u'^{n}_{i+\frac{1}{2}} \\ u'^{n+1}_{i+1} \\ u'^{n+1}_{i+\frac{1}{2}} \\ u'^{n+1}_{i} \end{pmatrix}.$$

Introducing the weights $W = [w_0, w_1, w_2, w_3] = \left(1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}\right) V^{-1}$ we can write,

$$I = \frac{\Delta t}{\Delta x} \left[w_0 u'^{n}_{i+\frac{1}{2}} + w_1 u'^{n+1}_{i+1} + w_2 u'^{n+1}_{i+\frac{1}{2}} + w_3 u'^{n+1}_{i} \right].$$

Insert it in (4.7),

$$u^{n+1}_{i+\frac{1}{2}} = u^{n}_{i+\frac{1}{2}} - \underbrace{a \frac{\Delta t}{\Delta x}}_{c} \left[w_0 u'^{n}_{i+\frac{1}{2}} + w_1 u'^{n+1}_{i+1} + w_2 u'^{n+1}_{i+\frac{1}{2}} + w_3 u'^{n+1}_{i} \right].$$

Below, we present the modified formula for updating the point values

$$u^{n+1}_{i+\frac{1}{2}} - u^{n}_{i+\frac{1}{2}} + c \left[w_0 u'^{n}_{i+\frac{1}{2}} + w_1 u'^{n+1}_{i+1} + w_2 u'^{n+1}_{i+\frac{1}{2}} + w_3 u'^{n+1}_{i} \right] = 0. \tag{4.14}$$

The averaged values update formula in (4.11), combined with the point values update in (4.14), forms an implicit Active Flux scheme of order 4.

4.3.3 Fifth-order scheme

Here, we investigate the fifth-order accurate scheme to illustrate the effectiveness of our new strategy.

We can express the formula for updating averaged values as follows

$$\begin{aligned} -c(c-2)(c-1)^2(c+1)\bar{u}^{n+1}_{i-1} + 2c(c-2)(c-1)(c+1)(c+2)u^{n+1}_{i-\frac{1}{2}} \\ -2(c-2)(c+2)(3c^2-1)\bar{u}^{n+1}_{i} - 2c(c-2)(c-1)(c+1)(c+2)u^{n+1}_{i+\frac{1}{2}} \\ +c(c-1)(c+1)^2(c+2)\bar{u}^{n+1}_{i+1} - c(c+1)^2(c+2)\bar{u}^{n}_{i-1} \\ +2(c-2)(c-1)(c+1)(c+2)\bar{u}^{n}_{i} - c(c-2)(c-1)^2\bar{u}^{n}_{i+1} = 0. \end{aligned} \tag{4.15}$$

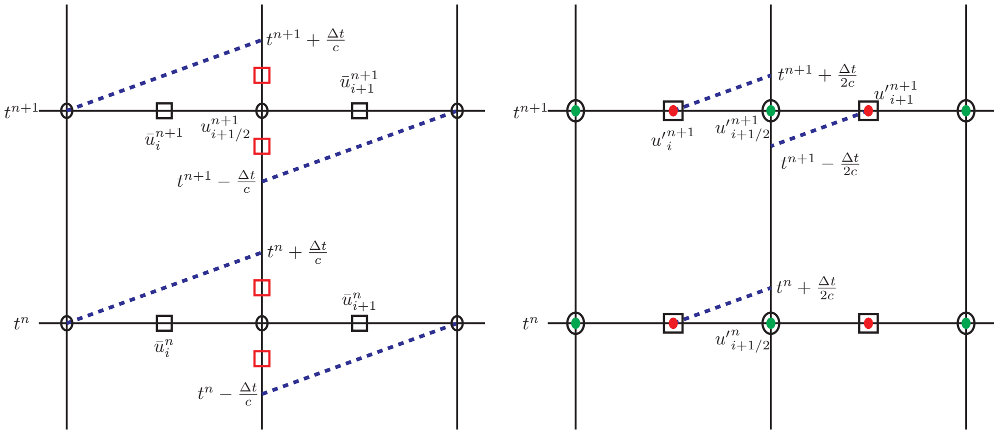

$t^{n+1}+\frac{\Delta t}{c}$
$\bar{u}^{n+1}_{i+1}$
t^{n+1}
$\bar{u}^{n+1}_{i}$
$u^{n+1}_{i+1/2}$
$t^{n+1}-\frac{\Delta t}{c}$
$t^{n}+\frac{\Delta t}{c}$
$\bar{u}^{n}_{i+1}$
t^{n}
$\bar{u}^{n}_{i}$
$t^{n}-\frac{\Delta t}{c}$
$t^{n+1}+\frac{\Delta t}{2c}$
u'^{n+1}_{i+1}
t^{n+1}
u'^{n+1}_{i}
$u'^{n+1}_{i+1/2}$
$t^{n+1}-\frac{\Delta t}{2c}$
$t^{n}+\frac{\Delta t}{2c}$
t^{n}
$u'^{n}_{i+1/2}$

Figure 4.13: On the left side, the figure shows the stencil related to the cell average update □–○–□ □–○–□, while the right side depicts the stencil related to the point value update □–○–□ □–○–□. We display the cell averages (□) and the point values (○). Additionally, the red circles (●) represent the spatial derivative values at the cell centers, and the green circles (●) represent the spatial derivative values at the cell interfaces.

Step 1

For fifth-order accurate schemes, it is necessary to incorporate degree four polynomials in both spatial and temporal dimensions. We'll start by constructing the space-dependent polynomial of degree 4 for the i^{th} cell at time t^n, denoted by $p_i^n(\xi)$, subject to the following constraints. Similarly, we can construct the space-dependent polynomial at time level t^{n+1}.

$$p_i^n(\xi=0)=u^n_{i-\frac{1}{2}}, \qquad \int_{\xi=0}^{\xi=1} p_i^n(\xi)d\xi = \bar{u}_i^n, \qquad p_i^n(\xi=1)=u^n_{i+\frac{1}{2}},$$
$$\int_{\xi=1}^{\xi=2} p_i^n(\xi)d\xi = \bar{u}^n_{i+1}, \qquad p_i^n(\xi=2)=u^n_{i+\frac{3}{2}}.$$

The spatial polynomial and its derivative take the following form at time level t^n

$$\begin{aligned} p_i^n(\xi) = {} & u^n_{i-\frac{1}{2}}\left(1-6\xi+\frac{39}{4}\xi^2-6\xi^3+\frac{5}{4}\xi^4\right)+\bar{u}_i^n\left(\frac{23}{2}\xi-\frac{99}{4}\xi^2+17\xi^3-\frac{15}{4}\xi^4\right)\\ & +u^n_{i+\frac{1}{2}}\left(-8\xi+24\xi^2-20\xi^3+5\xi^4\right)+\bar{u}^n_{i+1}\left(\frac{7}{2}\xi-\frac{51}{4}\xi^2+13\xi^3-\frac{15}{4}\xi^4\right)\\ & +u^n_{i+\frac{3}{2}}\left(-\xi+\frac{15}{4}\xi^2-4\xi^3+\frac{5}{4}\xi^4\right), \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial\xi}p_i^n(\xi) = {} & u^n_{i-\frac{1}{2}}\left(-6+\frac{39}{2}\xi-18\xi^2+5\xi^3\right)+\bar{u}_i^n\left(\frac{23}{2}-\frac{99}{2}\xi+51\xi^2-15\xi^3\right)\\ & +u^n_{i+\frac{1}{2}}\left(-8+48\xi-60\xi^2+20\xi^3\right)+\bar{u}^n_{i+1}\left(\frac{7}{2}-\frac{51}{2}\xi+39\xi^2-15\xi^3\right)\\ & +u^n_{i+\frac{3}{2}}\left(-1+\frac{15}{2}\xi-12\xi^2+5\xi^3\right). \end{aligned}$$

In the standard procedure, we introduce the variables $u'^n_{i+\frac{1}{2}}$, u'^n_i, u'^{n+1}_{i+1}, $u'^{n+1}_{i+\frac{1}{2}}$, and u'^{n+1}_i. Considering the notations as before, we construct a time-dependent polynomial along the cell interface to approximate the integral in equation (4.7).

Step 2

Let

$$q(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 + a_4\tau^4$$

be the reconstruction polynomial in time along the cell interface. When applying an interpolation technique, we obtain the following Vandermonde matrix

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & ,0 \\ 1 & (\frac{1}{2c}) & (\frac{1}{2c})^2 & (\frac{1}{2c})^3 & (\frac{1}{2c})^4 \\ 1 & (1-\frac{1}{2c}) & (1-\frac{1}{2c})^2 & (1-\frac{1}{2c})^3 & (1-\frac{1}{2c})^4 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & (1+\frac{1}{2c}) & (1+\frac{1}{2c})^2 & (1+\frac{1}{2c})^3 & (1+\frac{1}{2c})^4 \end{pmatrix}}_{V} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} u'^{n}_{i+\frac{1}{2}} \\ u'^{n}_{i} \\ u'^{n+1}_{i+1} \\ u'^{n+1}_{i+\frac{1}{2}} \\ u'^{n+1}_{i} \end{pmatrix}. \tag{4.16}$$

Consider the integral,

$$\begin{aligned} I =& \int_{t^n}^{t^{n+1}} \frac{\partial}{\partial x} u(x_{i+\frac{1}{2}}, t)dt \\ \approx& \frac{\Delta t}{\Delta x} \int_0^1 q(\tau)d\tau \\ =& \frac{\Delta t}{\Delta x} \left[a_0\tau + \frac{a_1}{2}\tau^2 + \frac{a_2}{3}\tau^3 + \frac{a_3}{4}\tau^4 + \frac{a_4}{5}\tau^5 \right]_0^1 \\ =& \frac{\Delta t}{\Delta x} \left(1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}\right) \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_4 \\ a_5 \end{pmatrix}. \end{aligned}$$

Substituting the relation from (4.16),

$$I = \frac{\Delta t}{\Delta x} \left(1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}\right) V^{-1} \begin{pmatrix} u'^{n}_{i+\frac{1}{2}} \\ u'^{n}_{i} \\ u'^{n+1}_{i+1} \\ u'^{n+1}_{i+\frac{1}{2}} \\ u'^{n+1}_{i} \end{pmatrix}.$$

Introducing the weights $W = [w_0, w_1, w_2, w_3, w_4] = \left(1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}\right) V^{-1}$ we can write,

$$I = \frac{\Delta t}{\Delta x} \left[w_0 u'^{n}_{i+\frac{1}{2}} + w_1 u'^{n}_{i} + w_2 u'^{n+1}_{i+1} + w_3 u'^{n+1}_{i+\frac{1}{2}} + w_4 u'^{n+1}_{i} \right].$$

Insert it in (4.7),

$$u^{n+1}_{i+\frac{1}{2}} = u^{n}_{i+\frac{1}{2}} - \underbrace{a\frac{\Delta t}{\Delta x}}_{c} \left[w_0 u'^{n}_{i+\frac{1}{2}} + w_1 u'^{n}_{i} + w_2 u'^{n+1}_{i+1} + w_3 u'^{n+1}_{i+\frac{1}{2}} + w_4 u'^{n+1}_{i} \right]$$

The following provides the modified formula for updating point values.

$$u^{n+1}_{i+\frac{1}{2}} - u^{n}_{i+\frac{1}{2}} + c\left[w_0 {u'}^{n}_{i+\frac{1}{2}} + w_1 {u'}^{n}_{i} + w_2 {u'}^{n+1}_{i+1} + w_3 {u'}^{n+1}_{i+\frac{1}{2}} + w_4 {u'}^{n+1}_{i}\right] = 0. \tag{4.17}$$

The averaged values update formula in (4.15), combined with the point values update in (4.17), forms an implicit Active Flux scheme of order 5.

4.3.4 Results, stability and convergence

First, we present the results for the modified implicit Active Flux methods discussed in Section 4.3.1, Section 4.3.2, and Section 4.3.3. We consider the initial condition presented in [54], which includes both smooth and discontinuous regions. The computational domain $[0, 2]$ is discretized into 100 cells. The CFL number is set to approximately 3. The results display in Figure 4.14. The higher-order schemes have overshoots and undershoots compared to the 3^{rd}-order scheme. However, they capture the exact solution more sharply than the 3^{rd}-order scheme.

Next, we compare the results of the Modified Implicit Active Flux schemes with those of the implicit Active Flux schemes discussed in Section 4.2.1. According to the results shown in Figure 4.15, the modified implicit Active Flux schemes yield similar outcomes.

The convergence study involves a smooth initial condition $u(x, 0) = \sin(x)$ to examine the convergence of the modified implicit Active Flux methods. We discretize the domain $[0, 1]$ into 100 cells and set the CFL number to 3 for the simulations. We compute the L^1 error by considering the point values at the final time $T = 5$. As illustrated in Figure 4.16, the schemes exhibit the expected convergence orders.

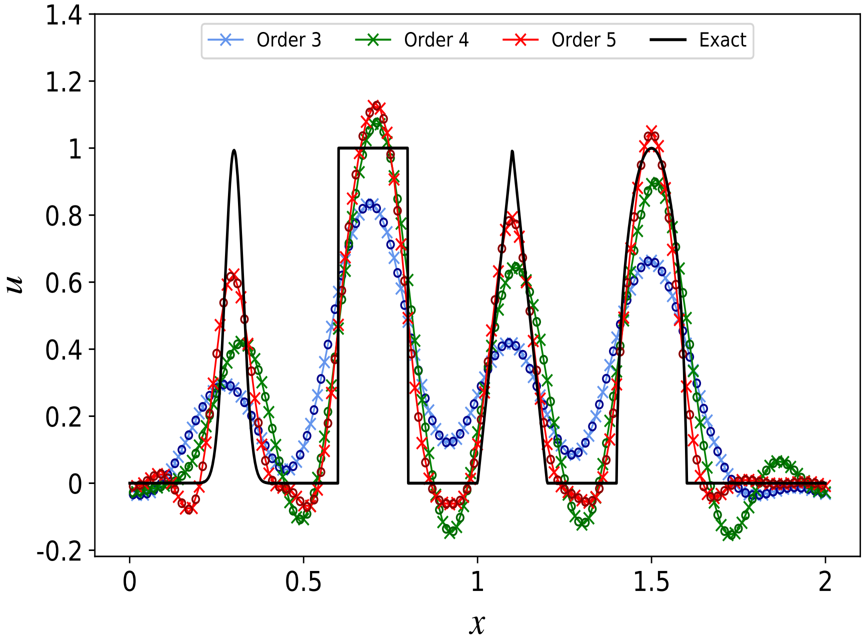
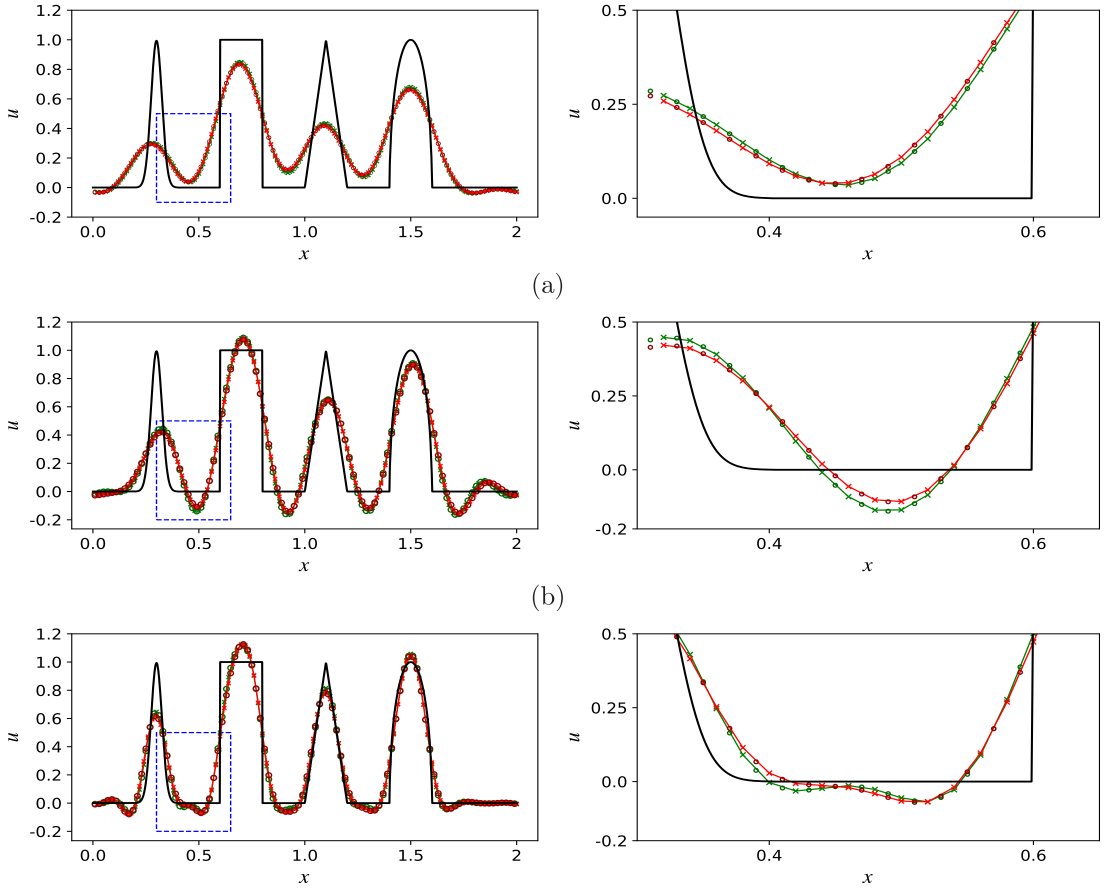


Figure 4.14: The results of the Modified Implicit Active Flux methods are presented at the final time $T = 8$. The symbol ($\circ$) denotes cell averages, while ($\times$) indicates point values.



(a)

(b)

(c)

Figure 4.15: Comparison between the single-step implicit Active Flux method (*green*) and the MIAF method (*red*) at $T = 8$ with CFL number approximately 3. (a) 3^{rd}-order schemes, (b): 4^{th}-order schemes and (c): 5^{th}-order schemes.The domain consists of 100 cells. The figures on the right correspond to zoomed-in views of the rectangle region mentioned in the left figures.

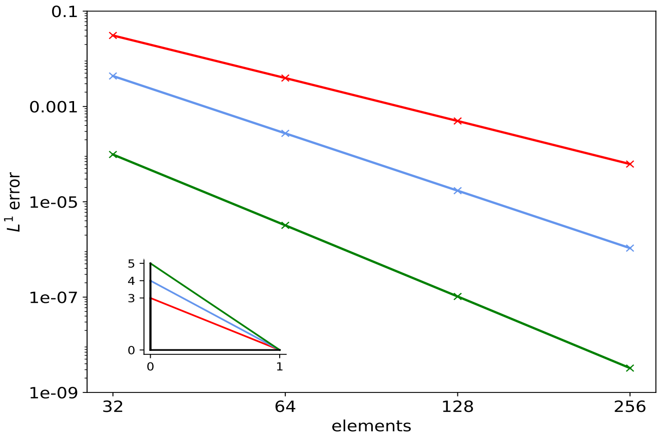


Figure 4.16: The L^1-errors are present for the third-order scheme —×—, the fourth-order scheme —×—, and the fifth-order scheme —×—.

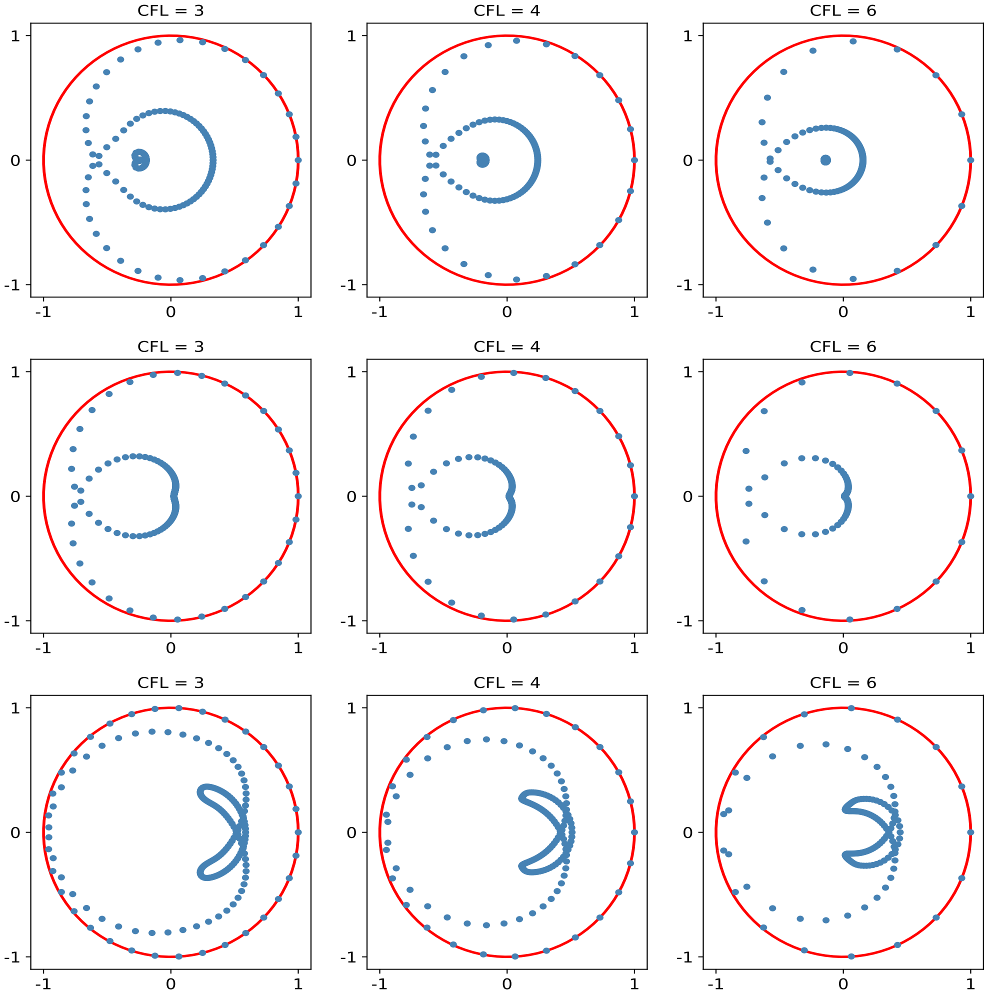


Figure 4.17: The spectrum of the matrix $A^{-1}B$ related to 3^{rd}-order (*top row*), 4^{th}-order (*middle row*) and 5^{th}-order (*bottom row*) MIAF schemes for different CFL numbers.

For completeness, the schemes are formulated according to the structure outlined in equation (4.3), and a stability analysis is conducted. Regarding the Modified Implicit Active Flux methods, the third-order scheme remains stable for CFL > 1, the fourth-order scheme for CFL > 1.18, and the fifth-order scheme for CFL > 2.43. The stable CFL numbers are determined using a bisection-type approach, relying on initial guesses. While the schemes demonstrate stability for CFL numbers greater than those identified, it's possible that stable regions exist for CFL numbers lower than those specified. Identifying these regions would require an analytical approach, which we have not yet explored and remains a topic for future investigation.

4.3.5 Exploring a case study: fifth-order scheme with CFL $= 2$

As discussed earlier, the Modified Implicit Active Flux methods offer a variety of schemes. Now, we compare two different fifth-order schemes, where the only difference lies in the method used to compute the spatial derivatives at a single point. For the fifth-order scheme, we consider the variables $u'^{n}_{i+\frac{1}{2}}$, u'^{n}_{i}, u'^{n+1}_{i+1}, $u'^{n+1}_{i+\frac{1}{2}}$, and u'^{n+1}_{i}. To compare the effect of the spatial derivatives, two approaches are used to approximate the variables u'^{n+1}_{i+1}, as described in Table 4.3. We construct the time-dependent reconstruction polynomial accordingly. Since here we only focus on a particular example that sets the CFL = 2, the fifth-order scheme is marginally stable. The spectrum of the iterative matrix, depicted in Figure 4.19, shows a circle of eigenvalues equal to one and others strictly less than one. This suggests that, in this scenario, the updating of point values and cell averages happens in a decoupled manner.

Similar results can be obtained from the single-step implicit Active Flux scheme Section 4.2.1 as well, and they are presented to conclude the results.

Case 1	Case 2
$u'^{n}_{i+\frac{1}{2}} \approx \frac{1}{\Delta x}\frac{\partial}{\partial \xi} p^n_i(\xi)\Big\|_{\xi=1}$	$u'^{n}_{i+\frac{1}{2}} \approx \frac{1}{\Delta x}\frac{\partial}{\partial \xi} p^n_i(\xi)\Big\|_{\xi=1}$
$u'^{n}_{i} \approx \frac{1}{\Delta x}\frac{\partial}{\partial \xi} p^n_i(\xi)\Big\|_{\xi=\frac{1}{2}}$	$u'^{n}_{i} \approx \frac{1}{\Delta x}\frac{\partial}{\partial \xi} p^n_i(\xi)\Big\|_{\xi=\frac{1}{2}}$
$u'^{n+1}_{i+1} \approx \frac{1}{\Delta x}\frac{\partial}{\partial \xi} p^{n+1}_i(\xi)\Big\|_{\xi=\frac{3}{2}}$	$u'^{n+1}_{i+1} \approx \frac{1}{\Delta x}\frac{\partial}{\partial \xi} p^{n+1}_{i+1}(\xi)\Big\|_{\xi=\frac{1}{2}}$
$u'^{n+1}_{i+\frac{1}{2}} \approx \frac{1}{\Delta x}\frac{\partial}{\partial \xi} p^{n+1}_i(\xi)\Big\|_{\xi=1}$	$u'^{n+1}_{i+\frac{1}{2}} \approx \frac{1}{\Delta x}\frac{\partial}{\partial \xi} p^{n+1}_i(\xi)\Big\|_{\xi=1}$
$u'^{n+1}_{i} \approx \frac{1}{\Delta x}\frac{\partial}{\partial \xi} p^{n+1}_i(\xi)\Big\|_{\xi=\frac{1}{2}}$	$u'^{n+1}_{i} \approx \frac{1}{\Delta x}\frac{\partial}{\partial \xi} p^{n+1}_i(\xi)\Big\|_{\xi=\frac{1}{2}}$

Table 4.3: The approximation for the derivatives at specific points considering the degree 4 space-dependent polynomial p.

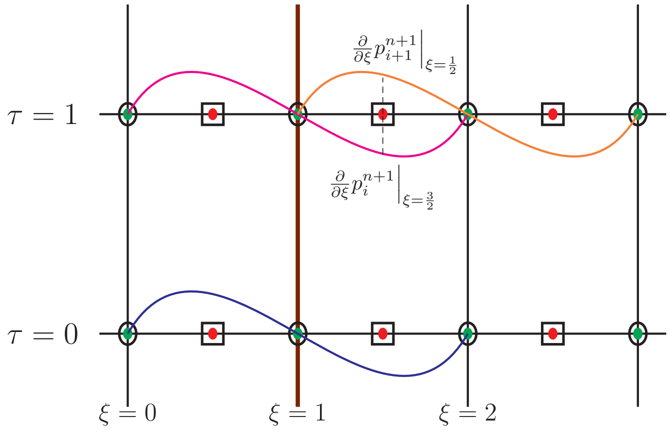

$\frac{\partial}{\partial \xi} p^{n+1}_{i+1}\Big|_{\xi=\frac{1}{2}}$
$\tau = 1$
$\frac{\partial}{\partial \xi} p^{n+1}_{i}\Big|_{\xi=\frac{3}{2}}$
$\tau = 0$
$\xi = 0$
$\xi = 1$
$\xi = 2$

Figure 4.18: The graphical representation of the two different polynomial approximations for the derivative.

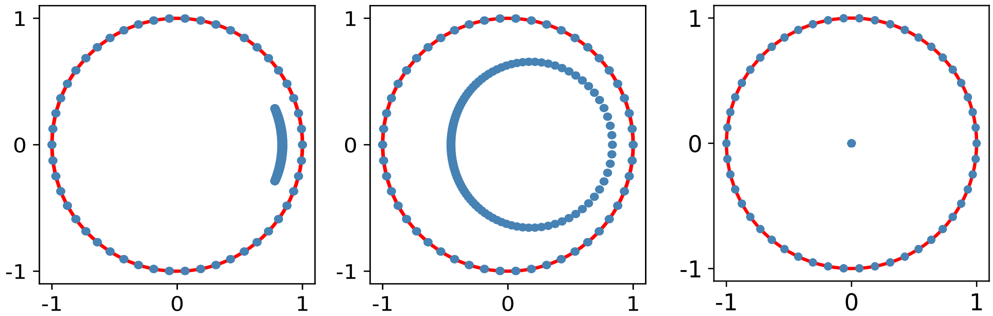


Figure 4.19: Spectrum of the matrix $A^{-1}B$ of the fifth-order MIAF schemes and single-step implicit Active Flux scheme. The left figure corresponds to the "Case1" and the middle figure corresponds to the "Case2". The right figure is related to the single-step implicit Active Flux scheme.

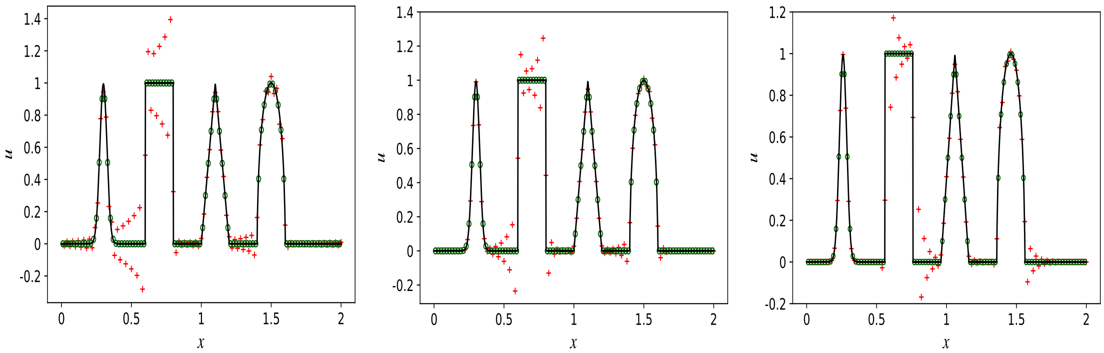


Figure 4.20: Results for the 5^{th}-order schemes with CFL$= 2$. The left figure shows the results for "Case 1", the middle figure shows "Case 2" and the right figure provides results for the single-step implicit Active Flux scheme. The domain consists of 100 cells. The final time is set at $T = 8$. The circle (○) denotes the cell averages and (+) represents the point values.

5 Conclusions and Future Works

In summary, this thesis has explored higher-order numerical methods for hyperbolic conservation laws. Based on the existing literature, two innovative higher-order numerical methods have been developed.

Chapter 2 provides a comprehensive overview of the ADER-DG methods, which are considered one of the foundations of the new approach. Several standard test cases are used to validate the numerical implementations. The application of the MOOD limiter helps in preventing oscillations near discontinuities.

Chapter 3 introduces the new higher-order Active Flux (hAF) schemes, developed by extending the classical Active Flux method using the ADER-DG technique. Within each cell, the cell average value is replaced with the corresponding degrees of freedom from the ADER-DG scheme. Two approaches are suggested for the point value update. The first suggested approach (P1) involves solving the Riemann problem at each cell interface, a method that highly depends on the specific problem. The second approach (P2), which is more general, integrates the conservation law at cell interfaces and utilizes the differential form of the flux function. The hAF schemes achieve a convergence rate of $N+3$ for degree N spatial test functions. Additionally, it is demonstrated that the CFL number improves for linear problems compared to the ADER-DG schemes. However, this improvement does not extend to nonlinear hyperbolic scenarios yet. Difficulties are encountered in achieving convergence in the iteration procedure (Equation 3.33) for local prediction with the improved CFL number. These aspects will be addressed in future work. To address the non-physical oscillations that appear near the discontinuities or steep gradients, we apply the MOOD limiter with some modifications. The detection procedure is improved by incorporating information from the point values. Two approaches for limiting the point values are tested. The first approach (L1) requires to solve Riemann problems at the interfaces of the troubled cells. The second approach (L2), a more general method, uses the volume averaged values of the corrected subcells as the limited point values. The modified MOOD limiter successfully performs across all proposed test cases. The schemes, which represent a more general approach for updating and limiting the point values, yield slightly sharper numerical results compared to highly problem-specific approaches, particularly for linear scalar hyperbolic problems.

Chapter 4 focuses on implicit higher-order numerical methods. Based on the implicit Active Flux methods proposed in [6], we extend the concept to linear systems of hyperbolic conservation laws. Further, a new update strategy has been proposed for the point values in the implicit setting. The newly proposed Modified Implicit Active Flux (MIAF) schemes yield similar results to those of the single-step Active Flux methods proposed in [6] but the MIAF approach allows for a larger variety of implicit numerical schemes. A complete stability analysis of the MIAF schemes will be conducted in future work.

Bibliography

[1] R. Abgrall. *A combination of residual distribution and the active flux formulations or a new class of schemes that can combine several writings of the same hyperbolic problem: application to the 1d euler equations. Communications on Applied Mathematics and Computation*, 5: 370–402, 2023. (see p. 53)

[2] R. Abgrall **and** W. Barsukow. *Extensions of Active Flux to arbitrary order of accuracy. ESAIM: Mathematical Modelling and Numerical Analysis*, 57: 991–1027, 2023. (see pp. 18, 45, 53)

[3] R. Abgrall **and** W. Barsukow. *Extensions of Active Flux to arbitrary order of accuracy. arXiv preprint arXiv:2208.14476*, 2022. (see pp. 1, 3, 34, 49)

[4] W. Barsukow. *The active flux scheme for nonlinear problems. Journal of Scientific Computing*, 86: 3, 2021. (see pp. 1, 3)

[5] W. Barsukow **and** J. P. Berberich. *A well-balanced Active Flux method for the shallow water equations with wetting and drying. Communications on Applied Mathematics and Computation*, 1–46, 2023. (see pp. 1, 3)

[6] W. Barsukow **and** R. Borsche. *Implicit Active Flux methods for linear advection. arXiv preprint arXiv:2303.13318*, 2023. (see pp. 3, 22, 25, 49, 53, 55, 58, 77)

[7] W. Barsukow, J. Hohm, C. Klingenberg, **and** P. L. Roe. *The active flux scheme on Cartesian grids and its low Mach number limit. Journal of Scientific Computing*, 81: 594–622, 2019. (see p. 2)

[8] T. J. Barth **and** H. Deconinck. *High-order methods for computational physics.* **volume** 9 Springer Science & Business Media, 2013. (see p. 1)

[9] M. Ben-Artzi **and** J. Falcovitz. *A high-resolution upwind scheme for quasi 1-D flows. IN: Numerical methods for the Euler equations of fluid dynamics (A87-14085 03-34). Philadelphia*, 66–83, 1985. (see p. 1)

[10] M. Ben-Artzi **and** J. Falcovitz. *A second-order Godunov-type scheme for compressible fluid dynamics. Journal of Computational Physics*, 55: 1–32, 1984. (see p. 1)

[11] M. Ben-Artzi **and** J. Falcovitz. *An upwind second-order scheme for compressible duct flows. SIAM journal on scientific and statistical computing*, 7: 744–768, 1986. (see p. 1)

[12] M. Ben-Artzi **and** J. Falcovitz. *GRP—An analytic approach to high-resolution upwind schemes for compressible fluid flow.* **in**: *Ninth International Conference on Numerical Methods in Fluid Dynamics.* Springer 2005. 87–91 (see p. 1)

[13] W. Boscheri **and** M. Dumbser. *Arbitrary-Lagrangian–Eulerian discontinuous Galerkin schemes with a posteriori subcell finite volume limiting on moving unstructured meshes. Journal of Computational Physics*, 346: 449–479, 2017. (see p. 2)

[14] S. Busto, M. Dumbser, C. Escalante, N. Favrie, **and** S. Gavrilyuk. *On high order ADER discontinuous Galerkin schemes for first order hyperbolic reformulations of nonlinear dispersive systems. Journal of Scientific Computing*, 87: 48, 2021. (see pp. 5, 7, 9)

[15] D. Calhoun, E. Chudzik, **and** C. Helzel. *The Cartesian grid Active Flux method with adaptive mesh refinement. Journal of Scientific Computing*, 94: 54, 2023. (see p. 2)

[16] S. Claina, S. Diota, **and** R. Louberea. *A very high-order finite volume method for hyperbolic systems: Multi-dimensional Optimal Order Detection (MOOD).* (see p. 10)

[17] B. Cockburn, S. Hou, **and** C.-W. Shu. *The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case. Mathematics of Computation*, 54: 545–581, 1990. (see p. 1)

[18] B. Cockburn **and** C.-W. Shu. *The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. Journal of computational physics*, 141: 199–224, 1998. (see p. 1)

[19] B. Cockburn **and** C.-W. Shu. *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. Mathematics of computation*, 52: 411–435, 1989. (see p. 1)

[20] P. Deuflhard **and** F. Bornemann. *Scientific computing with ordinary differential equations.* **volume** 42 Springer Science & Business Media, 2012. (see p. 15)

[21] S. Diot, S. Clain, **and** R. Loubère. *Improved detection criteria for the multi-dimensional optimal order detection (MOOD) on unstructured meshes with very high-order polynomials. Computers & Fluids*, 64: 43–63, 2012. (see pp. 2, 10, 11, 18)

[22] S. Diot, R. Loubère, **and** S. Clain. *The Multidimensional Optimal Order Detection method in the three-dimensional case: very high-order finite volume method for hyperbolic systems. International Journal for Numerical Methods in Fluids*, 73: 362–392, 2013. (see pp. 10, 11)

[23] M. Dumbser, D. S. Balsara, E. F. Toro, **and** C.-D. Munz. *A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. Journal of Computational Physics*, 227: 8209–8253, 2008. (see pp. 2, 5–9, 16)

[24] M. Dumbser, C. Enaux, **and** E. F. Toro. *Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws. Journal of Computational Physics*, 227: 3971–4001, 2008. (see pp. 2, 5–7, 39)

[25] M. Dumbser, F. Fambri, M. Tavelli, M. Bader, **and** T. Weinzierl. *Efficient implementation of ADER discontinuous Galerkin schemes for a scalable hyperbolic PDE engine. Axioms*, 7: 63, 2018. (see p. 5)

[26] M. Dumbser **and** M. Käser. *An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes—II. The three-dimensional isotropic case. Geophysical Journal International*, 167: 319–336, 2006. (see p. 5)

[27] M. Dumbser **and** M. Käser. *Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. Journal of Computational Physics*, 221: 693–723, 2007. (see pp. 6, 12)

[28] M. Dumbser, M. Käser, V. A. Titarev, **and** E. F. Toro. *Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. Journal of Computational Physics*, 226: 204–243, 2007. (see p. 6)

[29] M. Dumbser **and** R. Loubère. *A simple robust and accurate a posteriori sub-cell finite volume limiter for the discontinuous Galerkin method on unstructured meshes. Journal of Computational Physics*, 319: 163–199, 2016. (see p. 2)

[30] M. Dumbser **and** C.-D. Munz. *ADER discontinuous Galerkin schemes for aeroacoustics. Comptes Rendus Mécanique*, 333: 683–687, 2005. (see pp. 1, 5)

[31] M. Dumbser **and** C.-D. Munz. *Building blocks for arbitrary high order discontinuous Galerkin schemes. Journal of Scientific Computing*, 27: 215–230, 2006. (see p. 9)

[32] M. Dumbser **and** O. Zanotti. *Very high order PNPM schemes on unstructured meshes for the resistive relativistic MHD equations. Journal of Computational Physics*, 228: 6991–7006, 2009. (see pp. 2, 9)

[33] M. Dumbser, O. Zanotti, R. Loubère, **and** S. Diot. *A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws. Journal of Computational Physics*, 278: 47–75, 2014. (see pp. 2, 6, 7, 9–11)

[34] M. Eimer, R. Borsche, **and** N. Siedow. *Implicit finite volume method with a posteriori limiting for transport networks. Advances in Computational Mathematics*, 48: 21, 2022. (see pp. 3, 49–52)

[35] T. Eymann **and** P. Roe. *Active flux schemes for systems.* **in**: *20th AIAA computational fluid dynamics conference*. 2011. 3840 (see p. 1)

[36] T. A. Eymann **and** P. L. Roe. *Multidimensional active flux schemes.* **in**: *21st AIAA computational fluid dynamics conference*. 2013. 2940 (see pp. 1, 28)

[37] F. Fambri, M. Dumbser, **and** O. Zanotti. *Space–time adaptive ADER-DG schemes for dissipative flows: Compressible Navier–Stokes and resistive MHD equations. Computer Physics Communications*, 220: 297–318, 2017. (see pp. 5, 7, 9)

[38] D. Fan. *On the acoustic component of active flux schemes for nonlinear hyperbolic conservation laws.* phdthesis. 2017. (see p. 1)

[39] E. Gaburro **and** M. Dumbser. *A Posteriori Subcell Finite Volume Limiter for General P_NP_M PNPM Schemes: Applications from Gasdynamics to Relativistic Magnetohydrodynamics. Journal of Scientific Computing*, 86: 1–41, 2021. (see pp. 2, 10, 11)

[40] E. Gaburro, P. Öffner, M. Ricchiuto, **and** D. Torlo. *High order entropy preserving ADER-DG schemes. Applied Mathematics and Computation*, 440: 127644, 2023. (see p. 5)

[41] S. K. Godunov **and** I. Bohachevsky. *Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. Matematičeskij sbornik*, 47: 271–306, 1959. (see p. 2)

[42] C. Helzel, D. Kerkmann, **and** L. Scandurra. *A new ADER method inspired by the active flux method. Journal of Scientific Computing*, 80: 1463–1497, 2019. (see pp. 1–3, 17, 27, 29, 30, 34)

[43] A. Hidalgo **and** M. Dumbser. *ADER schemes for nonlinear systems of stiff advection–diffusion–reaction equations. Journal of Scientific Computing*, 48: 173–189, 2011. (see p. 9)

[44] G.-S. Jiang **and** C.-W. Shu. *Efficient implementation of weighted ENO schemes. Journal of computational physics*, 126: 202–228, 1996. (see pp. 39, 58)

[45] M. Käser **and** M. Dumbser. *An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes—I. The two-dimensional isotropic case with external source terms. Geophysical Journal International*, 166: 855–877, 2006. (see p. 5)

[46] M. Käser, M. Dumbser, J. De La Puente, **and** H. Igel. *An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes—III. Viscoelastic attenuation. Geophysical Journal International*, 168: 224–242, 2007. (see p. 5)

[47] R. J. LeVeque. *Finite volume methods for hyperbolic problems.* **volume** 31 Cambridge university press, 2002. (see pp. 2, 17)

[48] R. Loubere, M. Dumbser, **and** S. Diot. *A new family of high order unstructured MOOD and ADER finite volume schemes for multidimensional systems of hyperbolic conservation laws. Communications in Computational Physics*, 16: 718–763, 2014. (see pp. 2, 10, 11)

[49] J. Maeng. *On the advective component of active flux schemes for nonlinear hyperbolic conservation laws.* phdthesis. 2017. (see p. 1)

[50] W. H. Reed **and** T. Hill. *Triangularmesh methodsfor the neutrontransportequation. Los Alamos Report LA-UR-73-479*, 1973. (see p. 1)

[51] P. Roe. *Did numerical methods for hyperbolic problems take a wrong turning?* **in**: *Theory, Numerics and Applications of Hyperbolic Problems II: Aachen, Germany, August 2016.* Springer 2018. 517–534 (see p. 1)

[52] P. Roe. *Is discontinuous reconstruction really a good idea? Journal of Scientific Computing*, 73: 1094–1114, 2017. (see p. 1)

[53] P. L. Roe, T. Lung, **and** J. Maeng. *New approaches to limiting.* **in**: *22nd AIAA Computational Fluid Dynamics Conference.* 2015. 2913 (see p. 3)

[54] C.-W. Shu **and** G.-S. Jiang. *Efficient implementation of weighted ENO schemes. J. Comput. Phys*, 126: 202–228, 1996. (see pp. 13, 51, 71)

[55] C.-W. Shu **and** S. Osher. *Efficient implementation of essentially non-oscillatory shock-capturing schemes. Journal of computational physics*, 77: 439–471, 1988. (see pp. 1, 10)

[56] C.-W. Shu **and** S. Osher. *Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. Journal of Computational Physics*, 83: 32–78, 1989. (see pp. 1, 10)

[57] P. J. Steinle. *Finite difference methods for the advection equation.* phdthesis. 1993. (see pp. 3, 49)

[58] V. A. Titarev **and** E. F. Toro. *ADER: Arbitrary high order Godunov approach. Journal of Scientific Computing*, 17: 609–618, 2002. (see pp. 1, 5)

[59] E. F. Toro, R. Millington, **and** L. Nejad. *Towards very high order Godunov schemes.* **in**: *Godunov methods: theory and applications.* Springer 2001. 907–940 (see pp. 1, 5)

[60] E. F. Toro. *A Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction.* Springer, 2004. (see pp. 2, 10, 11)

[61] E. F. Toro **andothers**. *Shock-capturing methods for free-surface shallow flows.* Wiley **and** Sons Ltd., 2001. (see p. 20)

[62] B. Van Leer. *Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. Journal of computational physics*, 23: 276–299, 1977. (see p. 1)

[63] B. Van Leer. *Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. Journal of computational Physics*, 32: 101–136, 1979. (see p. 2)

[64] O. Zanotti, F. Fambri, **and** M. Dumbser. *Solving the relativistic magnetohydrodynamics equations with ADER discontinuous Galerkin methods, a posteriori subcell limiting and adaptive mesh refinement. Monthly Notices of the Royal Astronomical Society*, 452: 3010–3029, 2015. (see pp. 2, 7)

[65] O. Zanotti, F. Fambri, M. Dumbser, **and** A. Hidalgo. *Space–time adaptive ADER discontinuous Galerkin finite element schemes with a posteriori sub-cell finite volume limiting. Computers & Fluids*, 118: 204–224, 2015. (see pp. 2, 9)

Akademischer Lebenslauf

Suresh Nisansala K.A.

2020	Annahme als Doktorand am Fachbereich Mathematik an der Rheinland-Pfälzischen Technischen Universiät Kaiserslautern-Landau, Kaiserslautern, Deutschland.
2018 - 2020	Master-Abschluss in Mathematik am Fachbereich Mathematik an der Technischen Universität Kaiserslautern, Deutschland.
2013-2016	Bachelor-Abschluss in Mathematik an der Universität Ruhuna. Matara. Sri Lanka
2011	G.C.E. Advanced Level an der Henegama Central College, Henegama, Sri Lanka.

Academic curriculum vitae

Suresh Nisansala K.A.

2020	Acceptance as a doctoral candidate at the Department of Mathematics at the Rheinland-Pfälzische Technische Universiät Kaiserslautern-Landau, Kaiserslautern, Germany.
2018 - 2020	Master's degree in Mathematics, form the Department of Mathematics at Technische Universität Kaiserslautern, Germany.
2013-2016	Bachelor's degree in Mathematics from the University of Ruhuna, Matara, Sri Lanka.
2011	G.C.E. Advanced Level, Henegama Central College, Henegama, Sri Lanka.