Lax-Wendroff Flux Reconstruction for Compressible Flows

A Thesis

submitted to the
Tata Institute of Fundamental Research, Mumbai
for the degree of **Doctor of Philosophy**
in Subject Board of **Mathematics**

by
Arpit Babbar

Centre for Applicable Mathematics
Tata Institute of Fundamental Research
Bangalore 560065, India
July 26, 2024

Acknowledgements

I begin by thanking my advisor Praveen Chandrashekar for his mentoring, teaching, patience and generous expertise which made this thesis possible.

I thank my thesis monitoring committee members Venkateswaran Krishnan, Harish Kumar and Bonfiance Nkonga, for their valuable feedback and suggestions.

I would also like to thank my colleagues: Sudarshan Kumar Kenettinkara for his collaboration and insightful discussions, Rakesh Kumar for introducing me to WENO schemes, Saurav Samantaray for patiently introducing me to the fascinating field of Asymptotic Preserving schemes and for his guidance on several occasions, Kedar Wagh for various boundless exciting discussions, Aadi Bhure for helping me learn and appreciate quality of presentation in academic discourse along with gaining an innocent appreciation for scientific software.

I am also grateful to the developers of `Trixi.jl` for creating a flexible yet powerful open source PDE solver which was used majorly in this thesis. In particular, I thank Hendrik Ranocha for his kind answers to my questions on `Trixi.jl` and numerics, some of which contributed to the work presented in this thesis.

There have been some invaluable software from which I received additional support as a researcher, namely Intend.do, Obsidian and TEXMACS.

Finally, I would like to thank my family for their love, sacrifice and support which enabled me to make it this far.

List of figures

List of tables

Table of contents

Abstract

The Lax-Wendroff (LW) method is a single stage method for evolving time dependent solutions governed by partial differential equations, in contrast to Runge-Kutta (RK) methods that need multiple stages per time step. We develop a Lax-Wendroff Flux Reconstruction (LWFR) method in combination with a Jacobian-free Lax-Wendroff procedure that is applicable to general hyperbolic conservation laws. The numerical flux is carefully constructed - a D2 dissipation scheme is introduced improving CFL numbers and **EA** scheme which improves accuracy for nonlinear problems.

A subcell based limiter is developed by blending LWFR with a lower order scheme, either first order finite volume or MUSCL-Hancock scheme. While blending with a lower order scheme suppresses spurious oscillations, it may not guarantee admissibility of the discrete solution, e.g., positivity property of quantities like density and pressure. By exploiting the subcell structure and admissibility of lower order schemes, we devise a strategy to ensure that the blended scheme is admissibility preserving for the mean values and then use a scaling limiter to obtain admissibility at all solution points. For MUSCL-Hancock scheme on non-cell-centered subcells, we develop a slope limiter, time step restrictions, and suitable blending of higher order fluxes that ensures admissibility of lower order updates and hence that of the cell averages. By using the MUSCL-Hancock scheme on subcells and Gauss-Legendre points in flux reconstruction, we improve small-scale resolution compared to the subcell-based RKDG blending scheme with first order finite volume method and Gauss-Legendre-Lobatto points.

We propose a generalized admissibility framework by performing a cell average decomposition of LWFR. By performing a flux limiting of the time averaged numerical flux, the decomposition is used to obtain an admissibility preserving LWFR scheme. The admissibility preservation framework is further extended to conservation laws with source terms.

Multiderivative Runge-Kutta (MDRK) methods generalize LW and RK as they use multiple stages and LW procedure on each stage. We extend the fourth order, two stage MDRK scheme to FR by writing both of the stages in terms of a time averaged flux and then use the approximate Lax-Wendroff procedure. The developments made for LWFR apply to MDRK by using them at each stage. Thus, accuracy and stability are improved by **EA** scheme and D2 dissipation respectively. An admissibility preserving blending scheme is developed for MDRK.

We extend the LWFR scheme to solve conservation laws on curvilinear meshes with adaptive mesh refinement (AMR). It is proven that the proposed extension of LWFR scheme to curvilinear grids preserves constant solution (free stream preservation) under the standard metric identities. For curvilinear meshes, linear Fourier stability analysis cannot be used to obtain an optimal CFL number. Thus, an embedded-error based time step computation method is proposed for LWFR method which reduces the fine tuning process required to select a stable CFL number using the wave speed based time step computation. By using the BR1 scheme, LWFR on curvilinear meshes is also applied to second order equations in conservative form like the compressible Navier-Stokes model.

Chapter 1
Introduction

Hyperbolic conservation laws arise as models of physical systems representing the conservation of mass, momentum, and energy. They are routinely solved for applications like Computational Fluid Dynamics (CFD), astrophysics and weather modeling. Thus, the development of efficient numerical methods for solving these equations is crucial. The current state of memory-bound HPC hardware [6, 169] makes a strong case for development of high order discrete methods. By incorporating more higher order terms, these methods can achieve greater numerical accuracy per degree of freedom while minimizing memory usage and data transfers. In particular, high order methods have higher arithmetic intensity and are thus less likely to be memory-bound. The superior accuracy, efficiency, and higher resolution of these methods also make them a good fit for LES (Large Eddy Simulation) and DNS (Direct Numerical Simulation) of turbulent flows. Spectral element methods like Flux Reconstruction (FR) [94] and Discontinuous Galerkin (DG) [49] are high order methods that have been successful in resolving advection-dominated flows [195, 141]. The neighbouring FR/DG elements are coupled only through the numerical flux and thus the bulk of computations are local to the element, minimizing data transfers.

In comparison to traditional semi-discrete FR/DG schemes, which mainly use Runge-Kutta (RK) time integration, this work makes developments for Lax-Wendroff Flux Reconstruction (LWFR) which is a spectral element solver for time dependent PDE that performs high order evolution to the next time level in a single stage. The single stage nature implies fewer applications of shock capturing and positivity limiters, saving computational cost. Moreover, fewer stages minimize the interelement communication [65], making the method a good fit for modern hardware. There are also some order barriers in RK methods in the sense that at high orders, we need more stages than the order of the method. The LWFR method uses a Taylor series expansion in time and there is no order barrier as any order of accuracy can be reached by performing Taylor's expansion to the same and its single stage nature makes it more efficient than RK schemes [137].

In Sections 1.1, 1.2, we do a literature review of Lax-Wendroff and Flux Reconstruction schemes respectively. In Section 1.3, we review various limiters/shock capturing techniques used for high order methods in the literature. Section 1.4 gives an overview of the contributions made in this thesis and Section 1.5 gives an outline of the subsequent chapters.

1.1. Lax-Wendroff

In the context of hyperbolic conservation laws, the Lax-Wendroff (LW) time discretization in conjunction with a wide range of spatial schemes has been extensively studied in the literature. These temporal schemes are essentially based on the classical second-

order Lax-Wendroff method [113]. The Lax-Wendroff temporal discretization, originally referred to as the Taylor-Galerkin method, was used in the continuous finite element spatial schemes by Safian et al. [154] and Tabarrok et al. [172], followed by further improvements in [201]. The case of discontinuous finite element spatial schemes was studied in [44, 45]. All these methods are confined to a certain order of accuracy in both space and time. In the finite difference framework, the LW time discretization was originally proposed by Qui and Shu [138] with the WENO approximation of spatial derivatives [163]. As an extension to this, a combination with the alternative WENO method was developed in [99]. The discontinuous Galerkin spatial discretization combined with the LW temporal scheme was originally proposed in [138, 137] (abbreviated as LWDG) with an advantage of having arbitrary order of accuracy in both space and time, in other words, with no theoretical order barrier. It was further studied in [136], where the performance of various numerical fluxes was analyzed for the Euler equations of compressible flows. It is observed that the LWDG schemes are more compact and cost effective for certain problems like the two dimensional Euler system of compressible gas dynamics, especially when nonlinear limiters are applied. In [84] it is found that the LWDG method of [137] need not exhibit the super-convergence property. To overcome this issue, a modified version of LWDG was proposed [84] using the local DG framework of Cockburn et al. [53]. The resulting scheme was found to satisfy the super-convergence property. For linear conservation laws, the stability and accuracy properties of LWDG scheme are explored in [170] with the modified LWDG scheme of Guo et al. [84].

Another significant contribution towards the single stage temporal discretization was made by Toro et al., initially for linear equations in [178] and for nonlinear systems in [175], following the idea of generalized Riemann problem (GRP) [24, 87]. These are widely known as arbitrary high order derivative (ADER) methods. Though their inception was in the finite volume spatial setup, later they were extended to finite difference and discontinuous Galerkin frameworks [67]. In the sequel, several authors have contributed to this approach with the aim of shaping up a compact single time step scheme, see [177, 100, 66, 42] and references therein. In the flavour of ADER methods, Dumbser et al. proposed an efficient DG spatial solver in [63] and a finite difference WENO spatial solver in [64]. These are compact schemes that replace the so called Cauchy-Kowalevski procedure in the original ADER scheme with an element local space-time Galerkin predictor step and a discontinuous Galerkin corrector step, which are also found to be suitable for stiff source terms and further studied by Gassner et al. in [76]. These methods have been extended to the divergence free MHD problems with a finite volume WENO spatial scheme in [21]. Through a modification of the method in [63, 76], Guthrey et al. in [85] proposed a regionally implicit ADER discontinuous Galerkin solver which is stable for higher CFL numbers. A simplified Cauchy-Kowalevski procedure is developed in [129] which is efficient, easier to implement for any system, and can be used in ADER type schemes.

The generic versions of LWDG and ADER methods require the computation of high-order flux derivatives for each hyperbolic system and may require the use of symbolic manipulation software to perform the algebra. At higher orders of accuracy, we need higher order derivatives which need the computation of flux Jacobian and other higher order tensors. This increases the computational task and the process

has to be performed for each PDE system. In order to overcome this difficulty, an approximate procedure was originally developed in [208] in the finite difference scenario and further studied by several other authors [114, 34, 39, 40]. These approximate procedures for LW type solvers are found to be computationally more efficient and easier in implementation. As a single time step method, the resulting schemes are efficient for solving hyperbolic conservation laws. Moreover, it is independent of the specific form of flux function in the governing equation as it is free from Jacobian and other higher versions of derivatives.

1.2. Flux Reconstruction

Discontinuous Galerkin (DG) is a Spectral Element Method first introduced by Reed and Hill [145] for neutron transport equations and developed for fluid dynamics equations by Cockburn and Shu and others [49]. The DG method uses an approximate solution which is a polynomial within each element and is allowed to be discontinuous across interfaces.

The Flux Reconstruction (FR) method [94] is a class of discontinuous Spectral Element Methods for the discretization of conservation laws. FR methods utilize a nodal basis which is usually based on some solution points like Gauss points, to approximate the solution with piecewise polynomials. The main idea is to construct a continuous approximation of the flux utilizing a numerical flux at the cell interfaces and a correction function. The solution at the nodes is then updated by a collocation scheme in combination with a Runge-Kutta method. The choice of the correction function affects the accuracy and stability of the method; by properly choosing the correction function and solution points, FR method can be shown to be equivalent to some discontinuous Galerkin and spectral difference schemes, as shown in [94, 182]. In [189], semidiscrete linear stability analysis of FR is performed through a broken Sobolev norm, leading to a 1-parameter family of correction functions which encompasses the stable correction functions found in [94]. The family of stable correction functions has been extended in [191, 182], see [182] for a review. For the 1-parameter family of correction functions in [189], non-linear stability for E-fluxes was studied in [97] where the significance of solution points was pointed out, with Gauss-Legendre points being the most resistant to aliasing driven instabilities. In another study on accuracy with different choices of solution points [196], the optimality of Gauss-Legendre points was again observed. In the more recent works of [47, 46], a nonlinearly stable FR scheme was constructed in split form where a key idea was the application of correction functions to the volume terms. The long term error behaviour of FR schemes has been studied in [131, 2], while dispersion and dissipation errors have been analyzed in [190, 5, 187]. The Flux Reconstruction scheme has been used on curvilinear grids [195, 1, 46]. The development of Flux Reconstruction on curvilinear grids is primarily based on its equivalence with the DG scheme; see [105, 108] for the DG scheme on curvilinear grids. Thus, the study of free stream conditions for the FR scheme on curvilinear grids is the same as in [105]. The computationally efficient performance of FR has been noted in [193, 121, 186],

which is attributed to the structured computation of finite element methods suitable for modern hardware [193]. The quadrature-free nature of FR methods together with the ability to cast the operations as matrix-vector operations that can be performed efficiently using optimized kernels makes these methods ideal for use on modern vector processors [193].

1.3. Shock capturing and admissibility preservation of FR schemes

Despite the high accuracy of high order methods, lower order methods are still routinely applied in practical applications, in part due to their robustness. Solutions to hyperbolic conservation laws contain shocks in many problems and it is well known that high order schemes produce spurious oscillations in those cases. These oscillations can lead not only to incorrect solutions but can also easily generate nonphysical solutions like negative density or pressure. In order to develop robust high order methods for conservation laws, limiters have to be used which adaptively add numerical dissipation in regions where the numerical solution has a high gradient, possibly because of a shock. Some of the limiters like [62, 61, 69, 151] have inherent mechanisms that ensure physically admissible solutions while others like [51, 110, 90] can be made admissibility preserving by relying on the admissibility preserving in means property of the numerical scheme and using the scaling limiter of [205] to guarantee admissibility of solutions.

In the pioneering work of [51, 52], a Total Variation Bounded (TVB) limiter was introduced which reduces the scheme to first order or linear in certain elements using a minmod function to enforce a local TVB property. The TVB limiter does not preserve any subcell information other than the element mean and trace values, and there have been several works that develop limiters that are better in this regard. We now give a literature review of limiters that preserve subcell information.

Moment limiters [31, 33, 110] can be seen as an extension of TVB limiters where coefficients in an orthonormal basis (moments) are limited in a decreasing sequence, from higher to lower degree. The hierarchical nature of moment limiters enables the preservation of subcell information. Another popular strategy is the (H)WENO limiting procedure [135, 20], where the DG polynomial is substituted in troubled regions by a reconstructed (H)WENO polynomial that is computed by a WENO procedure using subcell and neighboring cells information. There are also the methods of artificial viscosity where a second order diffusion term is added in elements where the solution is non-smooth, preserving the subcell information as the high order polynomial solution is still used. In [134], an artificial viscosity model was introduced for the Runge-Kutta (RK) Discontinuous Galerkin (DG) method to add dissipation to the high order method based on a modal smoothness indicator. The indicator of [134] was further refined and detailed in [102].

There have also been several schemes that limit the solution by breaking the element into subcells which offers some advantages over artificial viscosity methods, including problem independence over boundary conditions and no additional time step restric-

tions, even when high dissipation is required [90]. In [93], the modal smoothness indicator of [134] was used to adapt local basis functions, e.g., switching to finite volume basis in the presence of discontinuities. In [41], subcells were used to assign different values to artificial viscosity within each element. In [165, 58], after having detected the troubled zones using the modal indicator of [134], cells are subdivided into subcells, and a robust first-order finite volume scheme is performed on the subgrid in troubled cells. In [90], the modal smoothness indicator of [134] was used to perform limiting by blending a high order DG scheme with Gauss-Legendre-Lobatto (GLL) points with a lower order finite volume scheme on subcells. In [148], the method of [90] was extended to resistive magnetohydrodynamics (MHD) and high order reconstruction on subcells was used to improve accuracy. In [151], it was shown that the subcell FV method of [90] can be made positivity preserving by an *a posteriori* modification of the blending coefficient. In [150], the subcell finite volume method of [90] with Rusanov's flux [152] was shown to be equivalent to the sparse Invariant Domain Preserving method of Pazner [133].

The approaches explained above can be classified as *a priori* limiters. We briefly discuss *a posteriori* limiting techniques where the solution is updated to time t^{n+1}, and low order re-updates are conducted in the elements that fail certain carefully chosen admissibility checks. One of these is the MOOD technique [48, 59, 60] where the local re-updates are computed with reduced order of accuracy until the admissibility checks pass. In [62, 61], the subcell based technique of [165, 58] is applied in an *a posteriori* fashion using $2N+1$ subcells for $N+1$ degrees of freedom per element in the 1-D case, using least squares approximation to convert back to a degree N polynomial. In case the least square transformation leads to a violation of admissibility constraints, the subcell solution values are used in the next evolution and thus the scheme is guaranteed to not crash. In [188], the DG scheme was reformulated as subcell Finite Volume (FV) method with appropriate subcells. An indicator was used to mark troubled subcells and thus the solution could be modified in a very localized manner, preserving subcell information well.

Other techniques for shock capturing exist that do not fit strictly into the aforementioned categories. In [69], positivity preservation and shock capturing were achieved by filtering and enforcing the minimum entropy principle, while in [123], a numerical damping term was introduced in the DG scheme to control spurious oscillations.

1.4. Contributions

The goal of this thesis is the development of a high order single stage Lax-Wendroff scheme with novel numerical flux computation, limiters, and time stepping that enhance accuracy, stability and performance along with extension of these developments to a wide variety of problems and on adaptively refined curvilinear meshes. The scheme is developed to solve general convection domination problems in the conservative form and the numerical validation has been performed using compressible flows governed by equations like Euler, Navier-Stokes, and the ten moment problem.

Lax Wendroff Flux Reconstruction. We combine the Lax-Wendroff method for time discretization with the FR method for spatial discretization since each of these two methods has its advantages as discussed in Sections 1.1, 1.2. In this work, we propose to combine the approximate LW procedure [208] with the FR scheme in space which leads to a general method that can be applied to any PDE system unlike the work of [122], where the flux derivatives are computed by using the chain rule of differentiation. The usage of the chain rule in [122] also leads to complicated tensorial quantities, especially for large systems and high orders. In previous works like [137], the solution at the current time level has been used to estimate the dissipative part of the numerical flux; however, it does not lead to an upwind flux, even for the linear advection equation. Here we propose to use the time average solution to compute the numerical flux, which leads to an upwind scheme for linear problems, and also increases the CFL numbers, which are comparable to other single step methods like ADER-DG scheme. We also show that the scheme is in fact equivalent to the ADER-DG scheme for linear problems. An interesting observation we make is that the method at fifth order has a mild instability even though we use the CFL number determined from Fourier stability analysis. This mild instability seems to be present even in some RKDG schemes. The central part of the numerical flux can be computed either by extrapolating it from the solution points to the faces or by directly estimating them at the faces by applying the approximate Lax-Wendroff procedure. These two methods perform differently for non-linear problems, with the extrapolation method leading to loss of convergence rate at odd polynomial degrees and also having larger errors compared to RK scheme. The alternate method proposed in this work performs uniformly well at all polynomial degrees and shows comparable accuracy to RK schemes. The LW method is developed for hyperbolic systems like Euler equations, where many commonly used numerical fluxes based on approximate Riemann solvers like Roe, HLL, HLLC, are used, along with the modifications that enhance the CFL number. The method is described up to fifth order accuracy and it is cast in terms of matrix-vector operations.

Subcell based blending limiter. The above developments were initially tested for nonsmooth problems by using the TVB limiter [51, 52]. The TVB limiter is a simple approach to reduce the scheme to first order or linear in FR elements using a minmod function. It is known to have shortcomings like loss of accuracy at smooth extrema and requirement of fine tuning of the TVBM parameter. In this work, the TVB limiter is considered inadequate for the following key reasons - it does not preserve any subcell information other than the element mean and trace values, and it is not provably admissibility preserving for Lax-Wendroff schemes even when used with the scaling limiter of Zhang and Shu [205]. Some of the works that deal with the first issue have been discussed in Section 1.3. The second issue has been considered in [128, 199] by modifying the numerical flux to obtain admissibility in means making the scaling limiter applicable. In [128], admissibility in means is obtained by limiting the numerical flux. In [199], a third order maximum-principle satisfying Lax-Wendroff DG scheme is constructed using the direct DG numerical flux from [43].

We develop the *a priori* blending limiter of [90] for LWFR as its choice of subcells gives a natural correction to the time averaged numerical flux to obtain admissibility preservation in means. The key idea of the blending scheme is to reduce spurious

oscillations by using a low order scheme in regions where the solution is not smooth, as detected by a smoothness indicator. The blending limiter by itself is not guaranteed to control all oscillations and thus unphysical solutions may still be obtained. Thus, we perform additional limiting to obtain a provably admissibility preserving scheme. Special attention is also paid to improving accuracy to capture small scale structures. We use Gauss-Legendre (GL) solution points and subcells obtained from GL quadrature weights instead of the GLL points and weights used in [90]. This is because of their accuracy advantage as observed by us, and as reported in the literature. In the non-linear stability analysis for E-fluxes in [97], Gauss-Legendre points were found to be the most resistant to aliasing driven instability. In another study on accuracy with different choices of solution points [196], the optimality of Gauss-Legendre points was again observed. In [18], optimal convergence rates for some non-linear problems were observed only for Gauss-Legendre solution points.

As observed in [149], accuracy can be improved by performing a high order reconstruction on the subcells. Since LWFR is a single-stage method, we improve accuracy by using the single-stage, second order MUSCL-Hancock scheme [185] on the subcells. As explained in [90], for a DG method of degree N, maintaining conservation requires the subcell sizes to be given by the $N+1$ quadrature weights and the solution points to be the solution points of DG scheme. This implies that the subcells are non-uniform and the finite volumes are neither cell-centered nor vertex centered. Thus, as a first step to ensuring that the blended scheme is admissible, we extend the work of [26] to obtain admissibility preserving MUSCL-Hancock scheme on the non-cell centered grids that occur from demanding conservation in the blending scheme. Enforcing admissibility as in [26] requires an additional slope limiting step and we propose a problem independent procedure to do the same.

To maintain conservation, low and high order updates need to use the same numerical flux at the FR element interfaces (see Remark 5.3). This numerical flux has to be chosen by blending between the high order time averaged flux and the low order FV flux. Thus, as the next step to enforce admissibility of the blended Lax-Wendroff scheme, we carefully select the blended numerical flux using a scaling procedure to ensure that the lower order updates at solution points neighboring the interfaces are admissible.

In [151], the blending limiter of [90] has been made admissibility preserving by changing the blending coefficients in an *a posteriori* fashion. Since our choice of the blended numerical flux implies the admissibility of lower order updates at all solution points, we could take the same approach. In this work, we instead use the fact that, with the blended numerical flux, admissibility of lower order scheme implies admissibility in the means of the blended scheme and thus the scaling limiter of [205] can now be used to obtain an admissibility preserving scheme. In [128], a correction has been made to the Lax-Wendroff numerical flux enforcing the admissibility in means property and then the scaling limiter [205] has been used to obtain an admissibility preserving Lax-Wendroff scheme. Our work differs from [128] as we only target to ensure admissibility of the lower order scheme and the admissibility in means is consequently obtained. This implies that our correction requires less storage and does not require additional loops, minimizing memory reads.

Generalized admissibility preservation. The subcell based blending scheme suppresses spurious oscillations but also gives a natural flux limiting procedure to ensure admissibility preservation of the LWFR scheme. We also develop a *generalized flux limiting* process that can also be used when there is no subcell based limiting scheme. The initial argument is similar to performing a decomposition of the cell average into *fictitious finite volume updates* as in [205, 206]. The difference from [205] arises as some of the fictitious finite volume updates involve the LW high order fluxes. Then, it is seen that, if the LW numerical flux is limited to ensure that the updates with its fictitious finite volume fluxes are admissible, the scheme will be admissibility preserving in means. In addition to showing that our positivity preserving framework preserves admissibility in the presence of shocks and rarefactions, we also introduce the first LWFR scheme in the presence of source terms. The approach involves adding time averages of the sources and thus we also propose a source term limiting procedure so that admissibility is maintained. The claim is validated on the Ten Moment equations, which are derived by Levermore et al. [118] by taking a Gaussian closure of the kinetic model.

Multiderivative Runge-Kutta. In [119], a two stage fourth order Multiderivative Runge-Kutta (MDRK) scheme was introduced for solving hyperbolic conservation laws by solving a Generalized Riemann Problem (GRP). We show the first combination of MDRK with a Flux Reconstruction scheme by using the scheme of [119]. We also use the construction of the numerical flux from [18]. In particular, we use the D2 dissipation and show that it leads to enhanced Fourier CFL stability limit. We also use the **EA** scheme which leads to enhanced accuracy for non-linear problems when using Gauss-Legendre solution points. We also develop admissibility preserving subcell based blending scheme and show how it is superior to other schemes like a TVB limiter.

Adaptive, curvilinear grids and time stepping. The LWFR scheme with the above features is further developed to incorporate three new features:

1. Ability to work on curvilinear, body-fitted grids
2. Ability to work on locally and dynamically adapted grids with hanging nodes
3. Automatic error based time step computation

Curvilinear grids are defined in terms of a tensor product polynomial map from a reference element to the physical element. The conservation law is transformed to the coordinates of the reference element and then the LWFR procedure is applied leading to a collocation method that has a similar structure as on Cartesian grids. This structure also facilitates the extension of the provably admissibility preserving subcell based blending scheme to curvilinear grids. The FR formulation on curvilinear grids is based on its equivalence with the DG scheme, see [105], which also obtained certain metric identities that are required for preservation of constant solutions, that is, free stream preservation. See references in [105] for a review of earlier studies of metric terms in the context of other higher order schemes like finite difference schemes. The free stream preserving conditions for the LWFR scheme are proven to be the same discrete metric identities as that of [105]. The only requirement for the required metric identities in two dimensions is that the mappings used to define the curvilinear elements must have degree less than or equal to the degree of polynomials used to approximate the solution.

In many problems, there are non-trivial and sharp solution features only in some localized parts of the domain and these features can move with respect to time. Using a uniform mesh to resolve small scale features is computationally expensive and adaptive mesh refinement (AMR) is thus very useful. In this work, we perform adaptive mesh refinement based on some local error or solution smoothness indicator. Elements with high error indicator are flagged for refinement and those with low values are flagged for coarsening. A consequence of this procedure is that we get non-conformal elements with hanging nodes which is not a major problem with discontinuous Galerkin type methods, except that one has to ensure conservation is satisfied. For discontinuous Galerkin methods based on quadrature, conservation is ensured by performing quadrature on the cell faces from the refined side of the face [155, 202]. For FR type methods which are of collocation type, we need numerical fluxes at certain points on the element faces, which have to be computed on a refined face without loss of accuracy and such that conservation is also satisfied. For the LWFR scheme, we develop the Mortar Element Method [106, 107] to compute the solution and fluxes at non-conformal faces. The resulting method is conservative and also preserves the free-stream condition on curvilinear, adapted grids.

The choice of time step is restricted by a CFL-type condition in order to satisfy linear stability and some other non-linear stability requirements like maintaining positive solutions. Linear stability analysis can be performed on uniform Cartesian grids only, leading to some CFL-type condition that depends on wave speed estimates. In practice, these conditions are then also used for curvilinear grids but they may not be optimal and may require tuning the time step for each problem by adding a safety factor. Thus, automatic time step selection methods based on some error estimates become very relevant for curvilinear grids. Error based time stepping methods are already developed for ODE solvers; and by using a method of lines approach to convert partial differential equations to a system of ordinary differential equations, error-based time stepping schemes of ODE solvers have been applied to partial differential equations [28, 101, 194] and recent application to CFD problems can be found in [140, 142]. The LWFR scheme makes use of a Taylor expansion in time of the time averaged flux; by truncating the Taylor expansion at one order lower, we can obtain two levels of approximation, whose difference is used as a local error indicator to adapt the time step. As a consequence, the user does not need to specify a CFL number, but only needs to give some error tolerances based on which the time step is automatically decreased or increased.

Parabolic equations. We extend the LWFR scheme to second order parabolic equations on curvilinear meshes by making use of the BR1 scheme. The BR1 is known to retain the superior properties of FR/DG, is applicable to underresolved turbulent simulations [77], and was proven to be stable in [78]. We use the error based time stepping developed for LWFR, which is especially relevant here since a Fourier CFL stability limit of LWFR is also not known for second order PDE. The ADER schemes, which are another class of single stage solvers have also been extended to solve second order PDE in [75] by including additional diffusion in the numerical flux in contrast to the BR1 scheme used here. This is the first work where any single stage method has been combined with the BR1 scheme.

1.5. OUTLINE

The rest of the thesis is organized as follows:

Chapter 2 introduces the basic notations to describe the relevant equations of motion. These include first order hyperbolic systems giving the example of compressible Euler's equations, but also second order equations like compressible Navier-Stokes equations.

Chapter 3 describes the spatial discretization using Flux Reconstruction for hyperbolic conservation laws. The description of finite volume and Discontinuous Galerkin methods for hyperbolic conservation laws is also provided.

Chapter 4 describes the core Lax-Wendroff Flux Reconstruction scheme using the approximate Lax-Wendroff procedure. The D2 dissipation to compute the dissipative part of numerical flux is introduced along with a Fourier stability analysis showing enhancement of CFL numbers in comparison to previous works. The computation of **EA** scheme to compute the central part of numerical flux is also introduced which enhances accuracy for nonlinear problems. The scheme is described for 1-D and 2-D and numerically validated for accuracy and stability with various scalar problems and Euler's system of equations.

Chapter 5 describes the subcell based blending limiter for LWFR. In the direction of robustness, provable admissibility preservation is obtained by careful construction of the *blended numerical flux*. In the direction of accuracy, Gauss-Legendre points are used and MUSCL-Hancock reconstruction is performed on the subcells. An admissibility preserving MUSCL-Hancock reconstruction scheme is developed for non-cell centred grids that naturally arise as subcells to ensure the conservation property. The claims are verified by numerical experiments on Euler's equations. The admissibility preservation is verified by problems that have shocks with very high pressure ratios. The accuracy improvement is verified on problems that have small scale structures along with strong shocks.

Chapter 6 introduces a generalized admissibility preserving framework for LWFR schemes extending the scaling limiter of Zhang and Shu. The framework is extended to equations with source terms maintaining admissibility. The admissibility preservation is verified by numerical results on Ten Moment equations of gas dynamics.

Chapter 7 introduces a two stage, fourth order multiderivative Runge-Kutta (MDRK) method in Flux Reconstruction framework by writing each stage as an evolution involving a time average flux. The time average flux is approximated by performing the LWFR procedure at each stage. The D2 dissipation and **EA** flux are introduced for MDRK enhancing stability and accuracy. The blending limiter is applied at each stage to obtain a provably admissibility preserving scheme. The scheme and claims are validated by a recent test suite for high order methods on Euler's equations.

Chapter 8 extends LWFR to adaptively refined curvilinear meshes. The mortar element method is developed for LWFR to obtain a scheme that is conservative, free stream and admissibility preserving. A Fourier CFL stability analysis does not apply to curvilinear meshes and thus an error based time stepping method is introduced. The scheme is validated by numerical experiments on Compressible Euler's equations. The time stepping method is shown to be of superior performance in comparison to CFL based time stepping even though it requires less fine tuning.

Chapter 9 extends the LWFR scheme to advection-diffusion equations by using the BR1 (Bassi-Rebay) scheme. The scheme is numerically validated through test cases of compressible Navier Stokes equations on curvilinear meshes by comparing the obtained numerical solutions with reference solutions.

Chapter 2

Equations of motion

In this chapter, we give a brief overview of the PDEs of interest along with the needed notations and definitions.

2.1. Hyperbolic conservation laws

Consider

$$\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, t) \colon \Omega \times \mathbb{R}_+ \longrightarrow \mathcal{U}_{\text{ad}} \subset \mathbb{R}^m \tag{2.1}$$

to be a vector of conserved quantities satisfying a system of equations of the form

$$\partial_t \boldsymbol{u} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}(\boldsymbol{u}) = \partial_t \boldsymbol{u} + \sum_{i=1}^{d} \partial_{x_i} \boldsymbol{f}_i(\boldsymbol{u}) = \boldsymbol{0} \tag{2.2}$$

The set $\Omega \subset \mathbb{R}^d$ is the domain and $\mathcal{U}_{\text{ad}} \subset \mathbb{R}^m$ (2.1) is a convex open set containing the set of *physically admissible solutions* of (2.2). The $\boldsymbol{f}(\boldsymbol{u}) = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_d) \in \mathbb{R}^{p \times d}$ are called the fluxes with $\boldsymbol{f}_i$ being the flux in the i^{th} direction. The equations (2.2) are called a *system of conservation laws*. By fundamental theorem of calculus, $\boldsymbol{u}$ is a classical C^1 solution to (2.2) if and only if for any open set $\Omega' \subset \Omega$

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega'} \boldsymbol{u}(\boldsymbol{x}, t) \, \mathrm{d}\boldsymbol{x} = -\int_{\partial\Omega'} \boldsymbol{f} \cdot \boldsymbol{n} \, \mathrm{d}S = -\int_{\partial\Omega'} \boldsymbol{f}_i \, n_i \, \mathrm{d}S \tag{2.3}$$

where $\boldsymbol{n} = (n_i)_{i=1}^d$ is the outward unit normal across $\partial\Omega'$. The equation (2.3) is called the *integral form of* (2.2) and it says that rate of change of $\boldsymbol{u}$ in any volume $\Omega' \subset \Omega$ depends only on the flux through the boundary $\partial\Omega'$ which is why (2.2) is called a conservation law. This integral form of conservation law is how the equation (2.1) is usually derived; e.g., Euler's equations (2.13) are derived from conservation of mass, momentum and energy. In this work, we only deal with *hyperbolic* conservation laws which are defined as follows.

Definition 2.1. *Let $\boldsymbol{A}_i(\boldsymbol{u}) := \boldsymbol{f}_i'(\boldsymbol{u})$ be the flux Jacobians. Then the system (2.2) is called hyperbolic if, for any $\boldsymbol{u} \in \mathcal{U}_{\text{ad}} \subset \mathbb{R}^m$ and any $\boldsymbol{n} \in \mathbb{R}^d / \{\boldsymbol{0}\}$, the matrix*

$$\boldsymbol{A}(\boldsymbol{u}, \boldsymbol{n}) := \sum_{i=1}^{d} \boldsymbol{A}_i(\boldsymbol{u}) \, n_i$$

has m real eigenvalues $\lambda_1(\boldsymbol{u}) \leq \ldots \leq \lambda_m(\boldsymbol{u})$ and m linearly independent eigenvectors $\{\boldsymbol{r}_j(\boldsymbol{u})\}_{j=1}^m$. The eigenvalues are also called the wave speeds or characteristic speeds associated with (2.2). If, in addition, these eigenvalues are distinct, then the system is said to be "strictly hyperbolic".

The pair $(\lambda_i(\boldsymbol{u}), \boldsymbol{r}_i(\boldsymbol{u}))$ corresponding to $\boldsymbol{A}(\boldsymbol{u}, \boldsymbol{n})$ is called the λ_i-*characteristic field*.

DEFINITION 2.2. *For $i \in \{1, \dots, m\}$, the i-characteristic field of (2.2) is genuinely nonlinear when*

$$\nabla \lambda_i(\boldsymbol{u}) \cdot r_i(\boldsymbol{u}) \neq 0, \qquad \forall \boldsymbol{u} \in \mathcal{U}_{\text{ad}} \tag{2.4}$$

and linearly degenerate when

$$\nabla \lambda_i(\boldsymbol{u}) \cdot r_i(\boldsymbol{u}) = 0, \qquad \forall \boldsymbol{u} \in \mathcal{U}_{\text{ad}} \tag{2.5}$$

The *Cauchy problem* for the above system also requires the prescription of initial conditions

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^d \tag{2.6}$$

where $\boldsymbol{u}_0 \colon \mathbb{R}^d \to \mathbb{R}^m$ and boundary conditions on $\partial\Omega$.

2.1.1. Weak formulation

In many practical problems, solutions to hyperbolic conservation laws contain non-smooth solutions including shocks and rarefactions. In fact, it is well known that the solutions can develop discontinuities in finite time, even when the initial condition is smooth [115]. Thus, the class of solutions to (2.2) must be enlarged beyond the classical C^1 solutions to include discontinuous solutions. For simplicity, we take the physical domain to be $\Omega = \mathbb{R}^d$. Then, we consider solutions in the space $L^\infty(\mathbb{R}^d \times \mathbb{R}_+, \mathcal{U}_{\text{ad}})$ of bounded Lebesgue measure functions $\boldsymbol{u} \colon \mathbb{R}^d \times \mathbb{R}_+ \to \mathcal{U}_{\text{ad}}$ and define them to be solutions in a weak (distributional) sense as follows.

DEFINITION 2.3. *A function $\boldsymbol{u} \in L^\infty(\mathbb{R}^d \times \mathbb{R}_+, \mathcal{U}_{\text{ad}})$ is called a weak solution of (2.2) with initial condition $\boldsymbol{u}_0 \in L^\infty(\mathbb{R}^d, \mathcal{U}_{\text{ad}})$ if*

$$\int_0^\infty \int_{\mathbb{R}^d} \left(\boldsymbol{u} \cdot \partial_t \boldsymbol{\phi} + \sum_{i=1}^d \boldsymbol{f}_i \cdot \partial_{x_i} \boldsymbol{\phi} \right) \mathrm{d}t \, \mathrm{d}\boldsymbol{x} + \int_{\mathbb{R}^d} \boldsymbol{u}_0(\boldsymbol{x}) \cdot \boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{0}) \, \mathrm{d}\boldsymbol{x} = 0 \tag{2.7}$$

for all $\boldsymbol{\phi} \in C_c^\infty(\mathbb{R}^d \times \mathbb{R}_+)$.

The weak formulation (2.7) is obtained by taking the inner product of (2.2) with a test function $\boldsymbol{\phi} \in C_c^\infty(\mathbb{R}^d \times \mathbb{R}_+)$ and performing integration by parts in space and time. As desired, the weak formulation allows for solutions with less regularity and every C^1 solution of (2.2) satisfies (2.7). The formulation (2.7) imposes conditions on the discontinuity, known as the *Rankine-Hugoniot* conditions. Let Γ be a surface of the discontinuity in $\mathbb{R}^d \times \mathbb{R}_+$ for the solution $\boldsymbol{u}$, and $\tilde{\boldsymbol{n}} = (n_1, \dots, n_d, n_t) \neq \boldsymbol{0}$ be the normal vector to Γ. Let us denote by $\boldsymbol{u}_\pm$ the limits of $\boldsymbol{u}$ on either side of Γ

$$\boldsymbol{u}_\pm(\boldsymbol{x}, t) = \lim_{\epsilon \to 0^+} \boldsymbol{u}((\boldsymbol{x}, t) \pm \epsilon \, \tilde{\boldsymbol{n}})$$

THEOREM 2.4. ***(Rankine-Hugoniot (RH) condition).*** *Consider a $\boldsymbol{u} \in L^\infty(\mathbb{R}^d \times \mathbb{R}_+, \mathcal{U}_{\text{ad}})$ that has a surface of discontinuity Γ and is smooth everywhere else. Then, $\boldsymbol{u}$ is a solution of (2.7) if and only if it satisfies (2.2) in regions of smoothness and*

$$(\boldsymbol{u}_+ - \boldsymbol{u}_-) \, n_t + \sum_{i=1}^d \left(\boldsymbol{f}_i(\boldsymbol{u}_+) - \boldsymbol{f}_i(\boldsymbol{u}_-) \right) n_i = \boldsymbol{0} \tag{2.8}$$

across the surface of discontinuity Γ.

In the 1-D case where dimension $d=1$, the Γ can be parametrized as $(\xi(t),t)$. Thus, the normal in (t,x) plane is given by $\tilde{\boldsymbol{n}}=(1,-s)$ with $s=\mathrm{d}\xi/\mathrm{d}t$ being the speed of the discontinuity. Thus, the RH condition (2.8) becomes

$$s\,(\boldsymbol{u}_+-\boldsymbol{u}_-)=\boldsymbol{f}(\boldsymbol{u}_+)-\boldsymbol{f}(\boldsymbol{u}_-) \tag{2.9}$$

2.1.2. The Riemann problem

A particularly important special case of Cauchy problem (2.6) for 1-D conservation laws

$$\partial_t\,\boldsymbol{u}+\partial_x\,\boldsymbol{f}(\boldsymbol{u})=\boldsymbol{0}$$

is the Riemann problem with piecewise constant initial data

$$\boldsymbol{u}(\boldsymbol{x},0)=\boldsymbol{u}_0(\boldsymbol{x})=\begin{cases}\boldsymbol{u}_l, & x<0\\ \boldsymbol{u}_r & x>0\end{cases} \tag{2.10}$$

where $\boldsymbol{u}_l,\boldsymbol{u}_r$ are constant states. This is the simplest problem that can be posed for conservation laws and is also central in the theory as it exhibits many important features encountered with general solutions of (2.2).

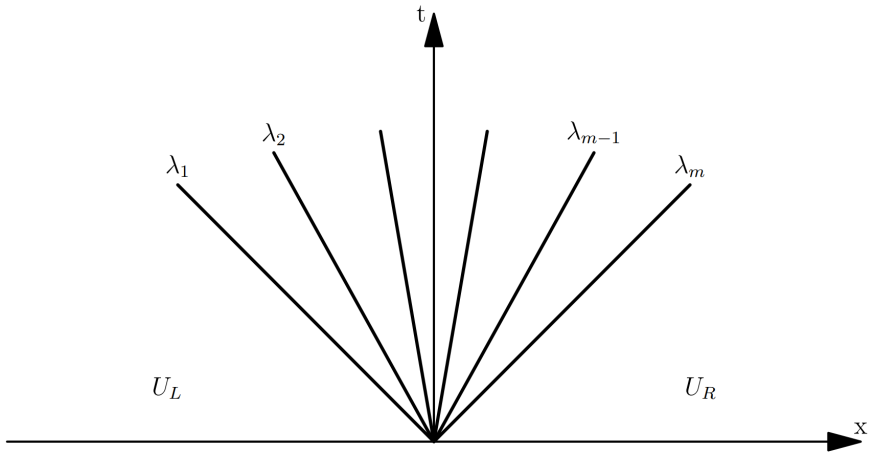


Figure 2.1. Solution structure for the Riemann problem of a system of conservation laws. The illustration is from [143].

We assume for simplicity that the 1-D system (2.2) is strictly hyperbolic (Definition 2.1) and thus has m distinct eigenvalues. This is satisfied by the compressible Euler's equations (2.11) in 1-D. The solution of the Riemann problem (2.10) is as in Figure 2.1 which consists of m distinct *waves* emanating from the origin, corresponding to each eigenvalue. The solutions to such problems are self-similar [115] in the sense that $\boldsymbol{u}(x,t)=\boldsymbol{W}(x/t)$. The $m+1$ states are connected by the following waves:

- **Shock wave:** The λ_i-wave is a shock wave if it corresponds to a genuinely nonlinear field (2.4) and connects two states $\boldsymbol{u}_-$ and $\boldsymbol{u}_+$ through a single jump discontinuity. The discontinuity moves with speed S_i satisfying 1-D RH condition (2.9) and relating to the eigenvalues by the Lax entropy condition

$$\lambda_i(\boldsymbol{u}_-)>S_i>\lambda_i(\boldsymbol{u}_+)$$

As shown in Figure 2.2a, the *characteristic* lines $\mathrm{d}x/\mathrm{d}t = \lambda_i$ on both sides collide leading to the shock wave $\mathrm{d}x/\mathrm{d}t = S_i$.

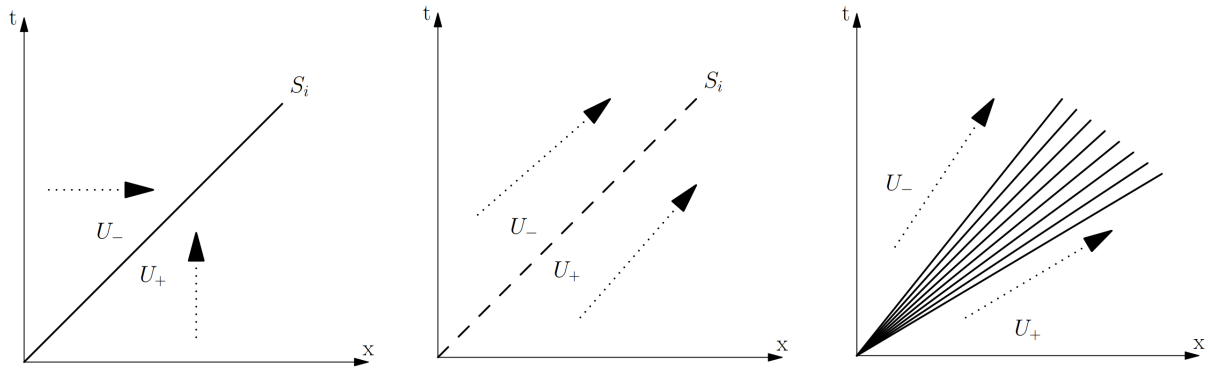
- **Contact wave:** The λ_i-wave is a contact wave, if it corresponds to a linearly degenerate field (2.5) and connects two states $\boldsymbol{u}_-$ and $\boldsymbol{u}_+$ through a single jump discontinuity. As in the case of the shock wave, the discontinuity moves with a speed S_i given by the RH condition (2.8). It additionally satisfies the *parallel characteristic condition*

$$\lambda_i(\boldsymbol{u}_-) = S_i = \lambda_i(\boldsymbol{u}_+)$$

This implies that the characteristic lines on either side of the contact line $\mathrm{d}x/\mathrm{d}t = S_i$ run parallel to it, as shown in Figure 2.2b.

- **Rarefaction:** The λ_i-wave corresponds to a rarefaction, if it connects two states $\boldsymbol{u}_-$ and $\boldsymbol{u}_+$ through a smooth transition in a genuinely nonlinear field. As shown in Figure 2.2c, the characteristic lines corresponding to a rarefaction diverge from each other, i.e.,

$$\lambda_i(\boldsymbol{u}_-) < \lambda_i(\boldsymbol{u}_+)$$



(a) Shock wave (b) Contact wave (c) Rarefaction

Figure 2.2. Characteristic lines for simple waves forming the solution to a Riemann problem. The illustration is from [143].

2.2. Compressible Euler's equations

The compressible Euler's equations of gas dynamics in 3-D are given by

$$\partial_t \begin{pmatrix} \rho \\ \rho\,\boldsymbol{v} \\ E \end{pmatrix} + \nabla_{\boldsymbol{x}} \cdot \begin{pmatrix} \rho\,\boldsymbol{v} \\ \rho\,\boldsymbol{v} \otimes \boldsymbol{v} + p\,I \\ \boldsymbol{v}\,(E+p) \end{pmatrix} = \boldsymbol{0}, \quad I = (\delta_{ij})_{\,1\le i,j\le 3} \tag{2.11}$$

The conservative variables are thus given by $\boldsymbol{u} = (\rho, \rho\,\boldsymbol{v}, E) = (\rho, \rho\,v_1, \rho\,v_2, \rho\,v_3, E)$ where $\rho, \boldsymbol{v}, p, E$ denote the fluid density, velocity, pressure and total energy per unit volume. For a polytropic gas, an equation of state $E = E(\rho, u, v, p)$ which leads to a closed system is given by

$$E = E(\rho, \boldsymbol{v}, p) = \frac{p}{\gamma - 1} + \frac{1}{2}\,\rho\,|\boldsymbol{v}|^2 \tag{2.12}$$

where $\gamma > 1$ is the adiabatic constant, which will usually be considered to be 1.4, the typical value for air. The admissible set is given by

$$\mathcal{U}_{\text{ad}} = \left\{ \boldsymbol{u} = (\rho, \rho\,\boldsymbol{v}, E) \colon \rho > 0, p = (\gamma - 1)\left(E - \frac{1}{2}\,\rho\,|\boldsymbol{v}|^2 \right) > 0 \right\}$$

Defining the flux Jacobian $\boldsymbol{A_i}(\boldsymbol{u}) = \boldsymbol{f}_i'(\boldsymbol{u})$ for $i = 1, 2, 3$, we consider the matrix

$$\boldsymbol{A}(\boldsymbol{u}, \boldsymbol{n}) \;=\; \boldsymbol{A}_1\, n_1 + \boldsymbol{A}_2\, n_2 + \boldsymbol{A}_3\, n_3, \qquad \boldsymbol{n} = (n_1, n_2, n_3) \in \mathbb{R}^3$$

The eigenvalues and eigenvectors of $\boldsymbol{A}(\boldsymbol{u}, \boldsymbol{n})$ are given by

$$\lambda_1 = v_n - a, \qquad \lambda_2 = \lambda_3 = \lambda_3 = v_n, \qquad \lambda_5 = v_n + a$$

$$\begin{aligned} \boldsymbol{R}(\boldsymbol{u}, \boldsymbol{n}) \;&=\; (\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{r}_4, \boldsymbol{r}_5) \\ &=\; \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ v_1 - a\,n_1 & v_1 & n_2 & -n_3 & v_1 + a\,n_1 \\ v_2 - a\,n_2 & v_2 & -n_1 & 0 & v_2 + a\,n_2 \\ v_3 - a\,n_3 & v_3 & 0 & n_1 & v_3 + a\,n_3 \\ H - a\,v_n & \frac{1}{2}\,|\boldsymbol{v}|^2 & v_1\,n_2 - v_2\,n_1 & v_3\,n_1 - v_1\,n_3 & H + a\,v_n \end{pmatrix} \end{aligned}$$

where $v_n = \boldsymbol{v} \cdot \boldsymbol{n}$, $a = \sqrt{\gamma\,p/\,\rho}$ is the speed of sound and $H = (\gamma - 1)^{-1}\,a^2 + |\boldsymbol{v}|^2/2$ is the specific enthalpy. Assuming the solution is admissible (i.e., $\rho, p > 0$), the eigenvalues are real and the corresponding eigenvectors are linearly independent. Thus, Euler's equations (2.11) form a hyperbolic system. In this work, we will be restricted to the 2-D compressible Euler's equations which are given by

$$\frac{\partial}{\partial t}\begin{pmatrix} \rho \\ \rho\,u \\ \rho\,v \\ E \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} \rho\,u \\ p + \rho\,u^2 \\ \rho\,u\,v \\ (E+p)\,u \end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix} \rho\,v \\ \rho\,u\,v \\ p + \rho\,v^2 \\ (E+p)\,v \end{pmatrix} = \boldsymbol{0} \tag{2.13}$$

where $u, v = v_1, v_2$.

2.3. Compressible Navier-Stokes equations

The Euler's equations (2.11) describe inviscid flows which do not account for viscosity and are thus applicable where the effect of viscosity is negligible in comparison to the advection. These advection dominated flows occur in a variety of practical problems. However, viscous effects do become important for studying flows with boundary layers near solid walls and the behaviour of fluids in turbulent regions. Thus, we consider the compressible Navier-Stokes equations which are hyperbolic-parabolic in nature. The equations are given in three dimensions as

$$\partial_t \begin{pmatrix} \rho \\ \rho\,\boldsymbol{v} \\ E \end{pmatrix} + \nabla_{\boldsymbol{x}} \cdot \begin{pmatrix} \rho\,\boldsymbol{v} \\ \rho\,\boldsymbol{v} \otimes \boldsymbol{v} + p\,I \\ \boldsymbol{v}\,(E+p) \end{pmatrix} = \nabla_{\boldsymbol{x}} \cdot \begin{pmatrix} 0 \\ \boldsymbol{\tau} \\ \boldsymbol{\tau}\,\boldsymbol{v} - \boldsymbol{Q} \end{pmatrix} \tag{2.14}$$

with the symmetric shear stress tensor $\boldsymbol{\tau}$ and heat flux $\boldsymbol{Q}$ given by Newtonian and Fourier constitutive relations respectively

$$\boldsymbol{\tau} = \mu\,(\nabla\,\boldsymbol{v} + (\nabla\,\boldsymbol{v})^T) - \frac{2}{3}\,\mu\,(\nabla\cdot\boldsymbol{v})\,I\,, \qquad \boldsymbol{Q} = (Q_1, Q_2, Q_3) = -\kappa\,\nabla\theta \tag{2.15}$$

Here, $I = (\delta_{ij})_{1\le i,j\le 3}$, μ is the coefficient of dynamic viscosity and κ is the coefficient of heat conductance. The θ denotes temperature of the flow which is obtained using the *ideal gas law* $p = \rho\,R\,\theta$ where R is the gas constant with $R = c_p - c_v$. The coefficient of heat conductance can be determined from μ using the relation

$$\kappa = \frac{\mu\,c_p}{\mathrm{Pr}}$$

where Pr is the Prandtl number, which is assumed to be constant for a given gas. The Euler's equations (2.11) can be recovered from the Navier-Stokes equations (2.14) by setting $\mu = 0$.

An important non-dimensional number for viscous flows is the *Reynolds number* given by

$$\mathrm{Re} = \frac{L\,U}{\nu}$$

where L and U are the respective characteristic length and velocity scales of the flow, $\nu = \mu/\,\rho_0$ is the coefficient of kinematic viscosity given the free stream density ρ_0. The Reynolds number can be seen as a measure of the ratio of advection and diffusion. High Reynolds number flows are advection dominated flows, while low Reynolds number flows are diffusion dominated.

In this work, we will be restricted to the Navier-Stokes equations in two dimensions which are given by

$$\begin{aligned}
&\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho\,u\\ \rho\,v\\ E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho\,u\\ p+\rho\,u^2\\ \rho\,u\,v\\ (E+p)\,u\end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix}\rho\,v\\ \rho\,u\,v\\ p+\rho\,v^2\\ (E+p)\,v\end{pmatrix}\\
&= \frac{\partial}{\partial x}\begin{pmatrix}0\\ \tau_{11}\\ \tau_{12}\\ u\,\tau_{11}+u_2\,\tau_{12}-Q_1\end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix}0\\ \tau_{21}\\ \tau_{22}\\ u_1\,\tau_{21}+u_2\,\tau_{22}-Q_2\end{pmatrix}
\end{aligned} \tag{2.16}$$

where $u, v = v_1, v_2$ and from (2.15)

$$\tau_{11} = \frac{4}{3}\,\mu\,\partial_x\,u - \frac{2}{3}\,\mu\,\partial_y\,v\,, \quad \tau_{12} = \tau_{21} = \mu\,(\partial_y\,u + \partial_x\,v), \quad \tau_{22} = \frac{4}{3}\,\mu\,\partial_y\,v - \frac{2}{3}\,\mu\,\partial_x\,u$$

$$Q_1 = -\kappa\,\partial_x\,\theta\,, \quad Q_2 = -\kappa\,\partial_y\,\theta$$

where θ is the temperature specified by ideal gas law $p = \rho\,R\,\theta$.

Chapter 3
Flux Reconstruction

In this chapter, we discuss the flux reconstruction scheme and its corresponding finite element basis that will be used in the subsequent chapters. The same basis is used to describe the discontinuous Galerkin method. The finite volume method is also briefly reviewed.

3.1. Conservation law

Let us consider a conservation law of the form

$$\boldsymbol{u}_t + \boldsymbol{f}(\boldsymbol{u})_x = \boldsymbol{0} \tag{3.1}$$

where $\boldsymbol{u}$ is some conserved quantity, $\boldsymbol{f}(\boldsymbol{u})$ is the corresponding flux, together with some initial and boundary conditions. The physically correct solution to (3.1) is going to be in the admissible set $\mathcal{U}_{\text{ad}}$ (2.1) whose detailed discussion is in Chapter 5. In this chapter, we focus on description of the finite element grid and basis.

We will divide the computational domain Ω into disjoint elements Ω_e, with

$$\Omega_e = [x_{e-\frac{1}{2}}, x_{e+\frac{1}{2}}] \qquad \text{and} \qquad \Delta x_e = x_{e+\frac{1}{2}} - x_{e-\frac{1}{2}}$$

Let us map each element to a reference element, $\Omega_e \to [0,1]$, by

$$x \mapsto \xi = \frac{x - x_{e-\frac{1}{2}}}{\Delta x_e}$$

Inside each element, we approximate the solution by degree $N \geq 0$ polynomials belonging to the set $\mathbb{P}_N$. For this, choose $N+1$ distinct nodes

$$0 \leq \xi_0 < \xi_1 < \cdots < \xi_N \leq 1 \tag{3.2}$$

which will be taken to be Gauss-Legendre (GL) or Gauss-Lobatto-Legendre (GLL) nodes, and will also be referred to as *solution points*. There are associated quadrature weights w_j such that the quadrature rule is exact for polynomials of degree up to $2N+1$ for GL points and up to degree $2N-1$ for GLL points. Note that the nodes and weights we use are with respect to the interval $[0,1]$ whereas they are usually defined for the interval $[-1,+1]$. The solution inside an element Ω_e is given by

$$x \in \Omega_e: \qquad \boldsymbol{u}_h(\xi, t) = \sum_{p=0}^{N} \boldsymbol{u}_{e,p}(t)\, \ell_p(\xi) \tag{3.3}$$

where $\{\ell_p\}$ are degree N Lagrange polynomials given by

$$\ell_p(\xi) = \prod_{q=0, q\neq p}^{N} \frac{\xi - \xi_q}{\xi_p - \xi_q} \in \mathbb{P}_N, \qquad \ell_p(\xi_q) = \delta_{pq}, \qquad 0 \leq p \leq N \tag{3.4}$$

Figure (3.1a) illustrates a piecewise polynomial solution at some time t_n with discontinuities at the element boundaries. Note that the coefficients $\boldsymbol{u}_{e,p}$ which are the basic unknowns or *degrees of freedom* (dof), are the solution values at the solution points.

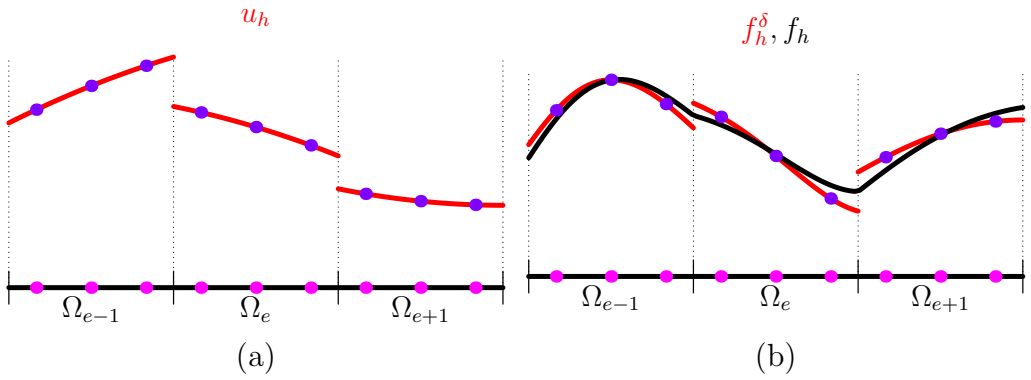


Figure 3.1. (a) Piecewise polynomial solution at time t_n, and (b) discontinuous and continuous flux.

The numerical method will require spatial derivatives of certain quantities. We can compute the spatial derivatives on the reference interval using a differentiation matrix $\mathsf{D}=[D_{pq}]$ whose entries are given by

$$D_{pq} = \ell_q'(\xi_p), \qquad 0 \le p, q \le N \tag{3.5}$$

For example, we can obtain the spatial derivatives of the solution $\boldsymbol{u}_h$ at all the solution points by a matrix-vector product as follows

$$\begin{bmatrix} \partial_x \boldsymbol{u}_h(\xi_0, t) \\ \vdots \\ \partial_x \boldsymbol{u}_h(\xi_N, t) \end{bmatrix} = \frac{1}{\Delta x_e} \mathsf{D}\mathsf{u}(t), \qquad \mathsf{u} = \begin{bmatrix} \boldsymbol{u}_{e,0} \\ \vdots \\ \boldsymbol{u}_{e,N} \end{bmatrix}$$

We will use symbols in sans serif font like D, u, etc. to denote matrices or vectors defined with respect to the solution points. The entries of the differentiation matrix are given by

$$D_{pq} = \frac{W_q}{W_p} \frac{1}{(\xi_p - \xi_q)}, \qquad p \ne q \qquad \text{and} \qquad D_{pp} = -\sum_{q=0, q\ne p}^{N} D_{pq}$$

where the W_p are barycentric weights given by

$$W_p = \frac{1}{\prod_{q=0, q\ne p}^{N} (\xi_p - \xi_q)}, \qquad 0 \le p \le N$$

Define the Vandermonde matrices corresponding to the left and right boundaries of a cell by

$$\mathsf{V}_L = [\ell_0(0), \ell_1(0), \ldots, \ell_N(0)]^\top, \qquad \mathsf{V}_R = [\ell_0(1), \ell_1(1), \ldots, \ell_N(1)]^\top \tag{3.6}$$

which is used to extrapolate the solution and/or flux to the cell faces for the computation of inter-cell fluxes.

3.2. FINITE VOLUME METHOD

Define the cell center of an element $\Omega_e = [x_{e-\frac{1}{2}}, x_{e+\frac{1}{2}}]$ as

$$x_e = \frac{1}{2}\left(x_{e-\frac{1}{2}} + x_{e+\frac{1}{2}}\right)$$

In a finite volume method, the unknowns to solve for are the *element averages* $\bar{\boldsymbol{u}}_e(t)$

$$\bar{\boldsymbol{u}}_e(t) \approx \frac{1}{\Delta x_e}\int_{\Omega_e} \boldsymbol{u}(x,t)\,\mathrm{d}x, \qquad \bar{\boldsymbol{u}}_e^n := \bar{\boldsymbol{u}}_e(t^n)$$

where t^n is the current time level for $n \geq 0$. Integrating the conservation law (3.1) over the element Ω_e gives

$$\frac{\mathrm{d}\bar{\boldsymbol{u}}_e}{\mathrm{d}t} + \frac{\boldsymbol{f}_{e+\frac{1}{2}} - \boldsymbol{f}_{e-\frac{1}{2}}}{\Delta x_e} = \boldsymbol{0} \tag{3.7}$$

where $\boldsymbol{f}_{e+\frac{1}{2}} \approx \boldsymbol{f}(\boldsymbol{u}(x_{e+\frac{1}{2}}, t))$ is the *numerical flux* function that couples neighbouring elements. The fundamental case is the one where the numerical flux is computed using only the adjacent elements

$$\boldsymbol{f}_{e+\frac{1}{2}} = \boldsymbol{f}(\bar{\boldsymbol{u}}_e, \bar{\boldsymbol{u}}_{e+1}) \tag{3.8}$$

In this case, the temporal discretization of (3.7) can be performed by the forward Euler method to get a first order accurate method

$$\bar{\boldsymbol{u}}_e^{n+1} = \bar{\boldsymbol{u}}_e^n - \frac{\Delta t^n}{\Delta x}\left(\boldsymbol{f}_{e+\frac{1}{2}} - \boldsymbol{f}_{e-\frac{1}{2}}\right), \qquad \Delta t^n := t^{n+1} - t^n \tag{3.9}$$

The choice of numerical flux (3.8) is typically made taking the specific conservation law (3.1) into consideration. It is based on the solution of a Riemann problem (Section 2.1.2) of the conservation law (3.1)

$$\boldsymbol{u}(x,0) = \begin{cases} \boldsymbol{u}_l, & x < 0 \\ \boldsymbol{u}_r, & x > 0 \end{cases}$$

To be precise, recalling the self-similarity of solutions of Riemann problem, we denote the exact solution of the Riemann problem as $\boldsymbol{u}(x/t; \boldsymbol{u}_l, \boldsymbol{u}_r)$. Then, the *Godunov's flux* for (3.8) is denoted by

$$\boldsymbol{f}(\bar{\boldsymbol{u}}_e, \bar{\boldsymbol{u}}_{e+1}) = \boldsymbol{f}(\boldsymbol{u}(0; \bar{\boldsymbol{u}}_e, \bar{\boldsymbol{u}}_{e+1}))$$

and an approximate Riemann solver is based on

$$\boldsymbol{f}(\bar{\boldsymbol{u}}_e, \bar{\boldsymbol{u}}_{e+1}) = \boldsymbol{f}(\boldsymbol{u}^{\text{approx}}(0; \bar{\boldsymbol{u}}_e, \bar{\boldsymbol{u}}_{e+1}))$$

Some numerical fluxes/approximate Riemann solvers for compressible Euler's equations (2.11) like Roe, HLL and HLLC are discussed in Appendix D. A numerical flux that applies to general conservation law is the Lax-Friedrichs flux. For a general conservation law with uniform grid size $\Delta x = \Delta x_e$ for all e, the *global* Lax-Friedrichs flux [116] is given by

$$\boldsymbol{f}_{e+\frac{1}{2}} = \frac{1}{2}\left(\boldsymbol{f}(\bar{\boldsymbol{u}}_e) + \boldsymbol{f}(\bar{\boldsymbol{u}}_{e+1})\right) - \frac{\Delta x}{2\,\Delta t^n}\left(\bar{\boldsymbol{u}}_{e+1} - \bar{\boldsymbol{u}}_e\right) \tag{3.10}$$

In the absence of the term $\frac{\Delta x}{2\,\Delta t^n}(\bar{\boldsymbol{u}}_{e+1}-\bar{\boldsymbol{u}}_e)$, the scheme (3.9) using (3.10) becomes the Forward Time Central Scheme (FTCS) which is unconditionally unstable. This term is called the dissipation term as its contribution to (3.7) gives a central approximation to $(\Delta x^2/2\,\Delta t)\,\partial_{xx}\boldsymbol{u}$. Following a von Neumann stability analysis, the time step size Δt^n of the scheme (3.9, 3.10) is usually computed to satisfy

$$\Delta t^n \max_e \frac{1}{\Delta x_e}\sigma(\boldsymbol{f}'(\bar{\boldsymbol{u}}_e)) \le 1$$

where $\sigma(A)$ is the maximum eigenvalue of a matrix A in absolute values. In practice, the time step Δt^n is taken to be close to the CFL limit and thus corresponding to each element e, we would like to have $\Delta x_e/\Delta t^n \approx \sigma(\boldsymbol{f}'(\bar{\boldsymbol{u}}_e))$ so that (3.10) motivates the *local Lax-Friedrichs*/Rusanov flux [152]

$$\begin{aligned}\boldsymbol{f}_{e+\frac{1}{2}} = \boldsymbol{f}^{\text{Rusanov}}(\bar{\boldsymbol{u}}_e,\bar{\boldsymbol{u}}_{e+1}) := \frac{1}{2}\left(\boldsymbol{f}(\bar{\boldsymbol{u}}_e)+\boldsymbol{f}(\bar{\boldsymbol{u}}_{e+1})\right) - \frac{1}{2}\lambda_{e+\frac{1}{2}}(\bar{\boldsymbol{u}}_{e+1}-\bar{\boldsymbol{u}}_e)\\ \lambda_{e+\frac{1}{2}} = \max\left\{\sigma(\boldsymbol{f}'(\bar{\boldsymbol{u}}_e)),\sigma(\boldsymbol{f}'(\bar{\boldsymbol{u}}_{e+1}))\right\}\end{aligned} \tag{3.11}$$

A numerical flux can use more neighbouring elements and get higher order accuracy

$$\begin{aligned}\boldsymbol{f}_{e+\frac{1}{2}} &= \boldsymbol{f}(\bar{\boldsymbol{u}}_{e-k},\ldots,\bar{\boldsymbol{u}}_{e-1},\bar{\boldsymbol{u}}_e,\bar{\boldsymbol{u}}_{e+1},\ldots,\bar{\boldsymbol{u}}_{e+l})\\ \boldsymbol{f}_{e+\frac{1}{2}} &= \boldsymbol{f}(\boldsymbol{u}(x_{e+\frac{1}{2}},t^n)) + O(\Delta x_e^{k+l+1})\end{aligned} \tag{3.12}$$

The approach where we obtain a semidiscrete scheme by discretizing only in space (3.7) is called the *method of lines*. Once a high order flux is chosen as in (3.12), in order to get high order accuracy in time, a multistage Runge-Kutta method for solving ODEs is used for solving the semidiscrete equation (3.7). There are many ways in which high order accuracy in space can be obtained. For second order accuracy, a MUSCL scheme [54] can be used that is based on performing linear reconstructions of the solution. For higher order accuracy, piecewise parabolic [55], ENO [88] and WENO [163] schemes can be used. While maintaining accuracy, the finite volume methods need to be chosen to preserve the admissibility set $\mathcal{U}_{\text{ad}}$ (2.1) of the conservation law (3.1) and thus we define admissibility preserving finite volume schemes as follows.

DEFINITION 3.1. *The finite volume method with flux approximation* (3.12) *is said to be admissibility preserving if* $\bar{\boldsymbol{u}}^n_{e-k-1},\bar{\boldsymbol{u}}^n_{e-k},\ldots,\bar{\boldsymbol{u}}^n_{e-1},\bar{\boldsymbol{u}}^n_e,\bar{\boldsymbol{u}}^n_{e+1},\ldots,\bar{\boldsymbol{u}}^n_{e+l}\in\mathcal{U}_{\text{ad}}$ *and*

$$\Delta t^n \le \Delta t_*(\bar{\boldsymbol{u}}^n) \tag{3.13}$$

imply

$$\bar{\boldsymbol{u}}^{n+1}_e = \bar{\boldsymbol{u}}^n_e - \frac{\Delta t^n}{\Delta x_e}\left(\boldsymbol{f}_{e+\frac{1}{2}}-\boldsymbol{f}_{e-\frac{1}{2}}\right)\in\mathcal{U}_{\text{ad}} \tag{3.14}$$

Thus, if solution at current time level is admissible at all points in the stencil and the time step restriction (3.13) *is satisfied, the finite volume evolution under forward Euler method* (3.9) *will also be admissible.*

A finite volume scheme using an admissibility preserving finite volume flux will preserve admissibility of solutions if the system of ODE (3.7) is solved with a *strong stability preserving Runge-Kutta (SSPRK) method* [162, 163]. This is because SSPRK methods are convex combinations of forward euler methods in each stage [205].

3.3. Runge-Kutta DG

This section introduces the Discontinuous Galerkin (DG) method with Runge-Kutta discretization in time. We multiply the conservation law (3.1) by a test function $v \in \mathbb{P}_N$ and integrate over element Ω_e

$$\int_{\Omega_e} \left(\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}}{\partial x} \right) v \, \mathrm{d}x = \boldsymbol{0}$$

An integration by parts is performed on the flux derivative term to get

$$\begin{aligned} \int_{\Omega_e} \frac{\partial \boldsymbol{u}}{\partial t} v \, \mathrm{d}x - \int_{\Omega_e} \boldsymbol{f}(\boldsymbol{u}) \frac{\partial v}{\partial x} \mathrm{d}x \\ + \boldsymbol{f}(x_{e+\frac{1}{2}}, t)\, v(x^-_{e+\frac{1}{2}}) - \boldsymbol{f}(x_{e-\frac{1}{2}}, t)\, v(x^+_{e-\frac{1}{2}}) = \boldsymbol{0} \end{aligned}$$

We now replace $\boldsymbol{u}$ with the numerical approximation $\boldsymbol{u}_h$ (3.3). At $x = x_{e+\frac{1}{2}}$, $\boldsymbol{u}_h$ may be discontinuous, i.e.,

$$\boldsymbol{u}_h(x^-_{e+\frac{1}{2}}, t) \neq \boldsymbol{u}_h(x^+_{e+\frac{1}{2}}, t)$$

Following the finite volume method, we will approximate the flux by a *numerical flux function* (3.8) denoted as

$$\boldsymbol{f}_{e+\frac{1}{2}}(t) = \boldsymbol{f}(\boldsymbol{u}_h(x^-_{e+\frac{1}{2}}, t), \boldsymbol{u}_h(x^+_{e+\frac{1}{2}}, t))$$

For example, the numerical flux can be taken to be $\boldsymbol{f}_{e+\frac{1}{2}}(t) = \boldsymbol{f}^{\text{Rusanov}}(\boldsymbol{u}_h(x^-_{e+\frac{1}{2}}, t), \boldsymbol{u}_h(x^+_{e+\frac{1}{2}}, t))$ (3.11). Thus, the semi-discrete DG scheme is given by

$$\begin{aligned} \int_{\Omega_e} \frac{\partial \boldsymbol{u}_h}{\partial t} v \, \mathrm{d}x - \int_{\Omega_e} \boldsymbol{f}(\boldsymbol{u}_h) \frac{\partial v}{\partial x} \mathrm{d}x \\ + \boldsymbol{f}_{e+\frac{1}{2}}(t)\, v(x^-_{e+\frac{1}{2}}) - \boldsymbol{f}_{e-\frac{1}{2}} v(x^+_{e-\frac{1}{2}}) = \boldsymbol{0} \end{aligned} \tag{3.15}$$

The scheme (3.15) is implemented by performing quadrature in space. It is explicit in the sense that the quadrature in the temporal derivative term will only require a *local mass matrix* to be inverted. If degree N quadrature with Gauss-Legendre points is performed, the integral on temporal derivative can be computed exactly. The integral on the flux term cannot be performed exactly because the flux $\boldsymbol{f}$ is usually nonlinear. We define a *discontinuous flux approximation* taking flux values at solution points giving a degree N polynomial represented in Lagrange basis (3.4) as

$$\boldsymbol{f}^\delta_h(\xi, t) = \sum_{p=0}^{N} \boldsymbol{f}(\boldsymbol{u}_{e,p}(t))\, \ell_p(\xi) \tag{3.16}$$

If quadrature is performed at the solution points, the equation (3.15) is equivalent to

$$\begin{aligned} \int_{\Omega_e} \frac{\partial \boldsymbol{u}_h}{\partial t} v \, \mathrm{d}x - \int_{\Omega_e} \boldsymbol{f}^\delta_h \frac{\partial v}{\partial x} \mathrm{d}x \\ + \boldsymbol{f}_{e+\frac{1}{2}}(t)\, v(x^-_{e+\frac{1}{2}}) - \boldsymbol{f}_{e-\frac{1}{2}} v(x^+_{e-\frac{1}{2}}) = \boldsymbol{0} \end{aligned}$$

Since we use Gauss-Legendre (GL) solution points or Gauss-Lobatto-Legendre (GLL), the integral on flux derivative is exact. Thus, we can perform an integration by parts in space to get the *strong form DG*

$$\begin{aligned}\int_{\Omega_e} \frac{\partial \boldsymbol{u}_h}{\partial t} v \,\mathrm{d}x + \int_{\Omega_e} \frac{\mathrm{d}\boldsymbol{f}_h^\delta}{\mathrm{d}x} v \,\mathrm{d}x \\ + (\boldsymbol{f}_{e+\frac{1}{2}}(t) - \boldsymbol{f}_h^\delta(x^-_{e+\frac{1}{2}}))\, v(x^-_{e+\frac{1}{2}}) - (\boldsymbol{f}_{e-\frac{1}{2}} - \boldsymbol{f}_h^\delta(x^+_{e-\frac{1}{2}}))\, v(x^+_{e-\frac{1}{2}}) = \boldsymbol{0}\end{aligned} \tag{3.17}$$

The scheme (3.17) is equivalent to the Flux Reconstruction (FR) scheme (Section 3.4) when GL/GLL points are used as solution and quadrature points. The proof is detailed in Appendix B, but the crucial idea is to take the test function to be $v = \ell_p$ (3.4) and use the identities (B.4).

3.4. Runge-Kutta FR

The Runge-Kutta Flux Reconstruction (RKFR) scheme is based on an FR spatial discretization leading to a system of ODE followed by the application of an RK scheme to march forward in time. The key idea is to construct a continuous polynomial approximation of the flux which is then used in a collocation scheme to update the nodal solution values. At some time t, we have the piecewise polynomial solution defined inside each cell; the FR scheme can be described by the following steps.

Step 1. In each element, we construct the flux approximation by interpolating the flux at the solution points leading to a polynomial of degree N, given by (3.16). The flux (3.16) is in general discontinuous across the elements similar to the red curve in Figure 3.1b.

Step 2. We build a continuous flux approximation by adding some correction terms at the element boundaries

$$\boldsymbol{f}_h(\xi,t) = \Big[\, \boldsymbol{f}_{e-\frac{1}{2}}(t) - \boldsymbol{f}_h^\delta(0,t) \Big]\, g_L(\xi) + \boldsymbol{f}_h^\delta(\xi,t) + \Big[\, \boldsymbol{f}_{e+\frac{1}{2}}(t) - \boldsymbol{f}_h^\delta(1,t) \Big]\, g_R(\xi)$$

where

$$\boldsymbol{f}_{e+\frac{1}{2}}(t) = \boldsymbol{f}(\boldsymbol{u}_h(x^-_{e+\frac{1}{2}}, t), \boldsymbol{u}_h(x^+_{e+\frac{1}{2}}, t))$$

is a numerical flux function that makes the flux unique across the cells. The continuous flux approximation is illustrated by the black curve in Figure 3.1b. The functions g_L, g_R are the correction functions that must be chosen to obtain a stable scheme.

Step 3. We obtain the system of ODE by collocating the PDE at the solution points

$$\frac{\mathrm{d}\boldsymbol{u}_{e,p}}{\mathrm{d}t}(t) = -\frac{1}{\Delta x_e} \frac{\partial \boldsymbol{f}_h}{\partial \xi}(\xi_p, t), \qquad 0 \le p \le N$$

which is solved in time by a Runge-Kutta scheme.

Correction functions. The correction functions g_L, g_R should satisfy the end point conditions

$$\begin{aligned} g_L(0) &= 1, & g_R(0) &= 0 \\ g_L(1) &= 0, & g_R(1) &= 1 \end{aligned} \tag{3.18}$$

which ensures the continuity of the flux, i.e., $\boldsymbol{f}_h(x^-_{e+\frac{1}{2}},t)=\boldsymbol{f}_h(x^+_{e+\frac{1}{2}},t)=\boldsymbol{f}_{e+\frac{1}{2}}(t)$. Moreover, we want them to be close to zero inside the element. There is a wide family of correction functions available in the literature [94, 189]. A family of correction functions depending on a parameter c was developed in [189] based on stability in a Sobolev-type norm. Two of these functions, the Radau and g_2 correction functions, are of major interest since they correspond to commonly used DG formulations. The Radau correction function is a polynomial of degree $N+1$ which belongs to the family of [189] corresponding to the parameter $c=0$ and given by

$$\begin{aligned} g_L(\xi) &= \frac{(-1)^N}{2}\left[L_N(2\,\xi-1)-L_{N+1}(2\,\xi-1)\right] \\ g_R(\xi) &= \frac{1}{2}\left[L_N(2\,\xi-1)+L_{N+1}(2\,\xi-1)\right] \end{aligned} \tag{3.19}$$

where $L_N\colon[-1,1]\to\mathbb{R}$ is the Legendre polynomial of degree N. The resulting RKFR scheme can be shown to be identical to the nodal RKDG scheme using Gauss-Legendre nodes for solution points and quadrature. In the general class of [189], g_2 correction function of degree $N+1$ corresponds to $c=\frac{2\,(N+1)}{(2\,N+1)\,N\,(a_N\,N!)^2}$ where a_N is the leading coefficient of L_N; they are given by

$$\begin{aligned} g_L(\xi) &= \frac{(-1)^N}{2}\left[L_N(2\,\xi-1)-\frac{(N+1)\,L_{N-1}(2\,\xi-1)+N\,L_{N+1}(2\,\xi-1)}{2\,N+1}\right] \\ g_R(\xi) &= \frac{1}{2}\left[L_N(2\,\xi-1)+\frac{(N+1)\,L_{N-1}(2\,\xi-1)+N\,L_{N+1}(2\,\xi-1)}{2\,N+1}\right] \end{aligned} \tag{3.20}$$

The resulting RKFR scheme can be shown to be identical to the nodal RKDG scheme using Gauss-Lobatto-Legendre points as solution points and for quadrature. We will perform a Fourier stability analysis of the Lax-Wendroff scheme based on these correction functions in a later section. Note that the correction functions are usually defined in the interval $[-1,1]$ but here we have written them for our reference interval which is $[0,1]$.

Chapter 4

Lax-Wendroff Flux Reconstruction

4.1. Introduction

In this chapter, we introduce the Lax-Wendroff (LW) scheme with the Flux Reconstruction (FR) method used for spatial discretization, since each of these two methods has the advantages discussed in the introduction. In brief, the advantage of Lax-Wendroff schemes arises from their single stage nature which minimizes interelement communication. The Flux Reconstruction is a quadrature free, vectorized scheme that generalizes variants of Discontinuous Galerkin and spectral difference schemes. We use the approximate Lax-Wendroff procedure of [208] so that, unlike the work of [122], the method does not require using chain rule which can lead to complicated Jacobians. This chapter uses discretization of the domain and function approximation by polynomials presented in Section 3.1 for solving the hyperbolic conservation law (3.1). The one dimensional Runge-Kutta Flux Reconstruction (RKFR) scheme from Section 3.4 is used for motivation as we introduce the Lax-Wendroff FR (LWFR) method in Section 4.2.

The numerical flux used on finite element interfaces has been improved for Lax-Wendroff schemes. We introduce a D2 dissipation numerical flux that improves Fourier CFL stability and **EA** flux that improves accuracy for nonlinear problems. The description of the numerical flux computation and how it improves over existing methods is presented in Section 4.3. The Fourier stability analysis in 1-D to demonstrate enhancement of CFL numbers is performed in Section 4.4. In Section 4.5, the treatment of boundary conditions is described. This chapter uses TVD limiter for problems with nonsmooth solution and it is described in Section 4.6. Sections 4.7, 4.8 present some numerical results in 1-D for scalar and system problems, to demonstrate the convergence rates and effect of correction functions, solution points and numerical flux schemes. Section 4.9 presents the LW scheme in two dimensions and Sections 4.10, 4.11 present numerical results in two dimensions. Section 4.12 presents a summary of the new scheme.

4.2. Lax-Wendroff FR scheme

In contrast to the spatial discretization described in Section 3.4, where a multistage Runge-Kutta scheme was needed to obtain high order accuracy, the LWFR scheme described here is a fully discrete high order scheme.

The Lax-Wendroff scheme combines spatial and temporal discretization into a single step. The starting point is a Taylor expansion in time following the Cauchy-Kowalewski procedure where the PDE is used to rewrite some of the time derivatives in the Taylor expansion as spatial derivatives. Using Taylor expansion in time around $t = t_n$, we can write the solution at the next time level as

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \sum_{m=1}^{N+1} \frac{\Delta t^m}{m!} \partial_t^m \boldsymbol{u}^n + O(\Delta t^{N+2})$$

Since the spatial error is expected to be of $O(\Delta x^{N+1})$, we retain terms up to $O(\Delta t^{N+1})$ in the Taylor expansion, so that the overall accuracy is of order $N+1$ both in space and time. Using the PDE, $\partial_t \boldsymbol{u} = -\partial_x \boldsymbol{f}$, we re-write time derivatives of the solution in terms of spatial derivatives of the flux

$$\partial_t^m \boldsymbol{u} = -\partial_t^{m-1} \partial_x \boldsymbol{f} = -(\partial_t^{m-1} \boldsymbol{f})_x, \qquad m = 1, 2, \ldots$$

so that

$$\begin{aligned} \boldsymbol{u}^{n+1} &= \boldsymbol{u}^n - \sum_{m=1}^{N+1} \frac{\Delta t^m}{m!} (\partial_t^{m-1} \boldsymbol{f})_x + O(\Delta t^{N+2}) \\ &= \boldsymbol{u}^n - \Delta t \left[\sum_{m=0}^{N} \frac{\Delta t^m}{(m+1)!} \partial_t^m \boldsymbol{f} \right]_x + O(\Delta t^{N+2}) \\ &= \boldsymbol{u}^n - \Delta t \frac{\partial \boldsymbol{F}}{\partial x}(\boldsymbol{u}^n) + O(\Delta t^{N+2}) \end{aligned} \tag{4.1}$$

where

$$\boldsymbol{F}(\boldsymbol{u}) = \sum_{m=0}^{N} \frac{\Delta t^m}{(m+1)!} \partial_t^m \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u}) + \frac{\Delta t}{2} \partial_t \boldsymbol{f}(\boldsymbol{u}) + \ldots + \frac{\Delta t^N}{(N+1)!} \partial_t^N \boldsymbol{f}(\boldsymbol{u}) \tag{4.2}$$

Note that $\boldsymbol{F}(\boldsymbol{u}^n)$ is an approximation to the time average flux in the interval $[t_n, t_{n+1}]$ since it can be written as

$$\boldsymbol{F}(\boldsymbol{u}^n) = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \left[\boldsymbol{f}(\boldsymbol{u}^n) + (t - t_n) \partial_t \boldsymbol{f}(\boldsymbol{u}^n) + \ldots + \frac{(t - t_n)^N}{N!} \partial_t^N \boldsymbol{f}(\boldsymbol{u}^n) \right] \mathrm{d}t \tag{4.3}$$

where the quantity inside the square brackets is the truncated Taylor expansion of the flux $\boldsymbol{f}$ in time. Equation (4.1) is the basis for the construction of the Lax-Wendroff method. Following the ideas in the RKFR scheme, we will first reconstruct the time average flux $\boldsymbol{F}$ inside each element by a continuous polynomial $\boldsymbol{F}_h(\xi)$. Then truncating equation (4.1), the solution at the nodes is updated by a collocation scheme as follows

$$\boldsymbol{u}_{e,p}^{n+1} = \boldsymbol{u}_{e,p}^n - \frac{\Delta t}{\Delta x_e} \frac{\mathrm{d}\boldsymbol{F}_h}{\mathrm{d}\xi}(\xi_p), \qquad 0 \le p \le N \tag{4.4}$$

where $\boldsymbol{u}_{e,p}$ are degrees of freedom of the approximate solution $\boldsymbol{u}_h$ (3.3) which is degree N in each finite element and is allowed to be discontinuous across element interfaces (Figure 3.1a). This is the single stage Lax-Wendroff update scheme for any order of accuracy. The major work in the above scheme is involved in the construction of the time average flux approximation $\boldsymbol{F}_h$ which is explained in subsequent sections.

4.2.1. Conservation property

The computation of correct weak solutions for non-linear conservation laws in the presence of discontinuous solutions requires the use of conservative numerical schemes. The Lax-Wendroff theorem shows that if a consistent, conservative method converges, then the limit is a weak solution. The method (4.4) is also conservative though it is not directly apparent; to see this multiply (4.4) by the quadrature weights associated with the solution points and sum overall the points in the e^{th} element,

$$\sum_{p=0}^{N} w_p \boldsymbol{u}_{e,p}^{n+1} = \sum_{p=0}^{N} w_p \boldsymbol{u}_{e,p}^{n} - \frac{\Delta t}{\Delta x_e} \sum_{p=0}^{N} w_p \frac{\partial \boldsymbol{F}_h}{\partial \xi}(\xi_p)$$

The correction functions are of degree $N+1$ and the flux $\boldsymbol{F}_h$ is a polynomial of degree $\leq N+1$. If the quadrature is exact for polynomials of degree at least N, which is true for both GLL and GL points, then the quadrature is exact for the flux derivative term and we can write it as an integral, which leads to

$$\int_{\Omega_e} \boldsymbol{u}_h^{n+1} \,\mathrm{d}x = \int_{\Omega_e} \boldsymbol{u}_h^{n} \,\mathrm{d}x - \Delta t \left[\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}}\right] \tag{4.5}$$

This shows that the total "mass" inside the cell changes only due to the boundary fluxes and the scheme is hence conservative.

4.2.2. Reconstruction of the time average flux

To complete the description of the LW method (4.4), we must explain the method for the computation of the time average flux $\boldsymbol{F}_h$. The flux reconstruction $\boldsymbol{F}_h(\xi)$ for a time interval $[t_n, t_{n+1}]$ is performed in three steps.

Step 1. Use the approximate Lax-Wendroff procedure to compute the time average flux $\boldsymbol{F}$ at all the solution points

$$\boldsymbol{F}_{e,p} \approx \boldsymbol{F}(\xi_p), \qquad 0 \leq p \leq N \tag{4.6}$$

The approximate LW procedure is explained in a subsequent section.

Step 2. Build a local approximation of the time average flux inside each element by interpolating at the solution points

$$\boldsymbol{F}_h^{\delta}(\xi) = \sum_{p=0}^{N} \boldsymbol{F}_{e,p} \ell_p(\xi) \tag{4.7}$$

which however may not be continuous across the elements. This is illustrated in Figure 3.1b.

Step 3. Modify the flux approximation $\boldsymbol{F}_h^{\delta}(\xi)$ so that it becomes continuous across the elements. Let $\boldsymbol{F}_{e+\frac{1}{2}}$ be some numerical flux function that approximates the flux $\boldsymbol{F}$ at $x = x_{e+\frac{1}{2}}$. Then the continuous flux approximation is given by

$$\boldsymbol{F}_h(\xi) = \left[\boldsymbol{F}_{e-\frac{1}{2}} - \boldsymbol{F}_h^{\delta}(0)\right] g_L(\xi) + \boldsymbol{F}_h^{\delta}(\xi) + \left[\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_h^{\delta}(1)\right] g_R(\xi) \tag{4.8}$$

which is illustrated in Figure 3.1b. The correction functions g_L, g_R are chosen from the FR literature [94, 182, 189] (Section 3.4).

Step 4. Let F denote the values of time average flux approximation at solution points (4.6). The derivatives of the continuous flux approximation at the solution points can be obtained as

$$\partial_\xi \mathsf{F}_\mathsf{h} = \left[\boldsymbol{F}_{e-\frac{1}{2}} - \mathsf{V}_L^\top \mathsf{F} \right] \mathsf{b}_L + \mathsf{D}\mathsf{F} + \left[\boldsymbol{F}_{e+\frac{1}{2}} - \mathsf{V}_R^\top \mathsf{F} \right] \mathsf{b}_R$$

$$\mathsf{b}_L = \begin{bmatrix} g_L'(\xi_0) \\ \vdots \\ g_L'(\xi_N) \end{bmatrix}, \qquad \mathsf{b}_R = \begin{bmatrix} g_R'(\xi_0) \\ \vdots \\ g_R'(\xi_N) \end{bmatrix}$$

which can also be written as

$$\partial_\xi \mathsf{F}_\mathsf{h} = \boldsymbol{F}_{e-\frac{1}{2}} \mathsf{b}_L + \mathsf{D}_1 \mathsf{F} + \boldsymbol{F}_{e+\frac{1}{2}} \mathsf{b}_R, \qquad \mathsf{D}_1 = \mathsf{D} - \mathsf{b}_L \mathsf{V}_L^\top - \mathsf{b}_R \mathsf{V}_R^\top \tag{4.9}$$

where $\mathsf{V}_L, \mathsf{V}_R$ are Vandermonde matrices defined in (3.6) and D is the differentiation matrix defined in (3.5). The quantities $\mathsf{D}, \mathsf{b}_L, \mathsf{b}_R, \mathsf{V}_L, \mathsf{V}_R$ can be computed once and re-used in all subsequent computations. They do not depend on the element and are computed on the reference element. Equation (4.9) contains terms that can be computed inside a single cell (middle term) and those computed at the faces (first and third terms) where it is required to use the data from two adjacent cells. The computation of the flux derivatives can thus be performed by looping over cells and then the faces. In the case of a system of equations, the differentiation matrices are applied to each variable; see Appendix E for a performance efficient implementation of these operations.

4.2.3. Direct flux reconstruction (DFR) scheme

An alternate approach to flux reconstruction which does not require the choice of a correction function is based on the idea of direct flux reconstruction [147], which we adopt in the Lax-Wendroff scheme as follows. Let us take the solution points to be the $N+1$ Gauss-Legendre nodes, and define

$$\xi_{-1} = 0, \qquad \xi_{N+1} = 1$$

The Lagrange polynomials corresponding to the $N+3$ points $\{\xi_i, i = -1, 0, \ldots, N+1\}$ are given by

$$\tilde{\ell}_j(\xi) = \prod_{i=-1, i\neq j}^{N+1} \frac{\xi - \xi_i}{\xi_j - \xi_i} \in \mathbb{P}_{N+2}$$

We approximate the continuous flux in terms of these polynomials

$$\boldsymbol{F}_h(\xi) = \boldsymbol{F}_{e-\frac{1}{2}} \tilde{\ell}_{-1}(\xi) + \sum_{p=0}^{N} \boldsymbol{F}_{e,p} \tilde{\ell}_p(\xi) + \boldsymbol{F}_{e+\frac{1}{2}} \tilde{\ell}_{N+1}(\xi)$$

We can compute the spatial derivatives using a differentiation matrix $\tilde{\mathsf{D}} \in \mathbb{R}^{(N+1)\times(N+3)}$

$$\tilde{D}_{pq} = \tilde{\ell}_q'(\xi_p), \qquad 0 \le p \le N, \quad -1 \le q \le N+1$$

Define $\mathsf{b}_L, \mathsf{b}_R$ to be the first and last column of the matrix $\tilde{\mathsf{D}}$ and D_1 to be the remaining columns

$$\mathsf{b}_L = \tilde{\mathsf{D}}\texttt{(:,-1)}, \qquad \mathsf{b}_R = \tilde{\mathsf{D}}\texttt{(:,N+1)}, \qquad \mathsf{D}_1 = \tilde{\mathsf{D}}\texttt{(0:N, 0:N)}$$

The flux derivatives at all the solution points can be computed as follows

$$\partial_\xi \mathsf{F}_\mathsf{h} = \boldsymbol{F}_{e-\frac{1}{2}}\, \mathsf{b}_L + \mathsf{D}_1 \mathsf{F} + \boldsymbol{F}_{e+\frac{1}{2}}\, \mathsf{b}_R$$

Note that the above equation has the same structure as (4.9) from the FR procedure but $\mathsf{b}_L, \mathsf{b}_R, \mathsf{D}_1$ are obtained using different idea. In this DFR approach, we cannot use GLL points since then the boundary points $\xi_{-1} = \xi_0$, $\xi_{N+1} = \xi_N$ would be repeated and the Lagrange interpolation is not well-defined; if we use GL points, the resulting scheme is identical to the LWFR approach using Radau correction function in combination with GL points as solution points, as shown in Appendix C.

4.2.4. Approximate Lax-Wendroff procedure

The time average flux at the solution points $\boldsymbol{F}_{e,p}$ must be computed using (4.2). The usual approach is to use the PDE and replace time derivatives with spatial derivatives, but this leads to a large amount of algebraic computations since we need to evaluate the flux Jacobian and its higher tensor versions. To avoid this process, we follow the ideas in [208, 34] and adopt an approximate Lax-Wendroff procedure. To present this idea in a concise and efficient form, we introduce the notation

$$\boldsymbol{u}^{(m)} = \Delta t^m\, \partial_t^m\, \boldsymbol{u}, \qquad \boldsymbol{f}^{(m)} = \Delta t^m\, \partial_t^m\, \boldsymbol{f}, \qquad m = 1, 2, \ldots$$

The time derivatives of the solution are computed using the PDE

$$\boldsymbol{u}^{(m)} = -\Delta t\, \partial_x\, \boldsymbol{f}^{(m-1)}, \qquad m = 1, 2, \ldots$$

Let the vector f below contain the flux values at solution points

$$\mathsf{f}_p = \boldsymbol{f}(\boldsymbol{u}_p)$$

The approximate Lax-Wendroff procedure uses a finite difference approximation applied at the solution points to compute the time derivatives of the fluxes. For example, a second order approximation is given by

$$\boldsymbol{f}_t(\xi, t) \approx \frac{\boldsymbol{f}(\boldsymbol{u}(\xi, t+\Delta t)) - \boldsymbol{f}(\boldsymbol{u}(\xi, t-\Delta t))}{2\,\Delta t}$$

The arguments to the flux are in turn approximated by a Taylor expansion in time

$$\boldsymbol{u}(\xi, t \pm \Delta t) \approx \boldsymbol{u}\,(\xi, t) \pm \boldsymbol{u}_t(\xi, t)\,\Delta t$$

Using this approximation at the p^{th} solution point in an element, we get

$$\begin{aligned} \boldsymbol{f}_p^{(1)} &= \boldsymbol{f}_t(\xi_p, t)\,\Delta t \approx \frac{1}{2}\left[\boldsymbol{f}(\boldsymbol{u}_p + \boldsymbol{u}_p^{(1)}) - \boldsymbol{f}(\boldsymbol{u}_p - \boldsymbol{u}_p^{(1)})\right] \\ \boldsymbol{u}_p^{(1)} &= \boldsymbol{u}_t(\xi_p, t)\,\Delta t = -\frac{\Delta t}{\Delta x_e}\,\boldsymbol{f}_\xi(\xi_p, t) \approx -\frac{\Delta t}{\Delta x_e}(\mathsf{Df})_p \end{aligned}$$

It can be shown that the above approximation to $\boldsymbol{f}_t$ is second order accurate in Δt. Such approximations can be written for higher accuracy and for higher time derivatives [208, 34], and we summarize them below at different orders of accuracy which are used in this thesis. The neglected term in the Taylor expansion (4.2) is of $O(\Delta t^{N+1})$, and hence the derivative approximation $\partial_t^m \boldsymbol{f}$ must be computed to at least $O(\Delta t^{N+1-m})$ accuracy. The Lax-Wendroff procedure is applied in each element and so for simplicity of notation, we do not show the element index in the following sub-sections.

4.2.4.1. Second order scheme, $N = 1$

The time average flux at the solution points is given by

$$\mathsf{F} = \mathsf{f} + \frac{1}{2}\mathsf{f}^{(1)}$$

where

$$\begin{aligned} \mathsf{u}^{(1)} &= -\frac{\Delta t}{\Delta x_e}\mathsf{Df} \\ \mathsf{f}^{(1)} &= \frac{1}{2}\left[\boldsymbol{f}\left(\mathsf{u}+\mathsf{u}^{(1)}\right) - \boldsymbol{f}\left(\mathsf{u}-\mathsf{u}^{(1)}\right)\right] \end{aligned}$$

4.2.4.2. Third order scheme, $N = 2$

The time average flux at the solution points is given by

$$\mathsf{F} = \mathsf{f} + \frac{1}{2}\mathsf{f}^{(1)} + \frac{1}{6}\mathsf{f}^{(2)}$$

where

$$\begin{aligned} \mathsf{u}^{(1)} &= -\frac{\Delta t}{\Delta x_e}\mathsf{Df} \\ \mathsf{f}^{(1)} &= \frac{1}{2}\left[\boldsymbol{f}(\mathsf{u}+\mathsf{u}^{(1)}) - \boldsymbol{f}(\mathsf{u}-\mathsf{u}^{(1)})\right] \\ \mathsf{u}^{(2)} &= -\frac{\Delta t}{\Delta x_e}\mathsf{Df}^{(1)} \\ \mathsf{f}^{(2)} &= \boldsymbol{f}\left(\mathsf{u}+\mathsf{u}^{(1)}+\frac{1}{2}\mathsf{u}^{(2)}\right) - 2\,\boldsymbol{f}(\mathsf{u}) + \boldsymbol{f}\left(\mathsf{u}-\mathsf{u}^{(1)}+\frac{1}{2}\mathsf{u}^{(2)}\right) \end{aligned}$$

4.2.4.3. Fourth order scheme, $N = 3$

For the fourth order scheme, the time average flux at the solution points reads as

$$\mathsf{F} = \mathsf{f} + \frac{1}{2}\mathsf{f}^{(1)} + \frac{1}{6}\mathsf{f}^{(2)} + \frac{1}{24}\mathsf{f}^{(3)}$$

where

$$
\begin{aligned}
\mathsf{u}^{(1)} &= -\frac{\Delta t}{\Delta x_e}\,\mathsf{Df} \\
\mathsf{f}^{(1)} &= \frac{1}{12}\left[-\boldsymbol{f}(\mathsf{u}+2\,\mathsf{u}^{(1)})+8\,\boldsymbol{f}(\mathsf{u}+\mathsf{u}^{(1)})-8\,\boldsymbol{f}(\mathsf{u}-\mathsf{u}^{(1)})+\boldsymbol{f}(\mathsf{u}-2\,\mathsf{u}^{(1)})\right] \\
\mathsf{u}^{(2)} &= -\frac{\Delta t}{\Delta x_e}\,\mathsf{Df}^{(1)} \\
\mathsf{f}^{(2)} &= \boldsymbol{f}\left(\mathsf{u}+\mathsf{u}^{(1)}+\frac{1}{2}\,\mathsf{u}^{(2)}\right)-2\,\boldsymbol{f}(\mathsf{u})+\boldsymbol{f}\left(\mathsf{u}-\mathsf{u}^{(1)}+\frac{1}{2}\,\mathsf{u}^{(2)}\right) \\
\mathsf{u}^{(3)} &= -\frac{\Delta t}{\Delta x_e}\,\mathsf{Df}^{(2)} \\
\mathsf{f}^{(3)} &= \frac{1}{2}\left[\boldsymbol{f}\left(\mathsf{u}+2\,\mathsf{u}^{(1)}+\frac{2^2}{2!}\,\mathsf{u}^{(2)}+\frac{2^3}{3!}\,\mathsf{u}^{(3)}\right)-2\,\boldsymbol{f}\left(\mathsf{u}+\mathsf{u}^{(1)}+\frac{1}{2!}\,\mathsf{u}^{(2)}+\frac{1}{3!}\,\mathsf{u}^{(3)}\right)\right. \\
&\qquad \left.+2\,\boldsymbol{f}\left(\mathsf{u}-\mathsf{u}^{(1)}+\frac{1}{2!}\,\mathsf{u}^{(2)}-\frac{1}{3!}\,\mathsf{u}^{(3)}\right)-\boldsymbol{f}\left(\mathsf{u}-2\,\mathsf{u}^{(1)}+\frac{2^2}{2!}\,\mathsf{u}^{(2)}-\frac{2^3}{3!}\,\mathsf{u}^{(3)}\right)\right]
\end{aligned}
$$

4.2.4.4. Fifth order scheme, $N = 4$

The time average flux at the solution points for the fifth order scheme takes the form

$$\mathsf{F}=\mathsf{f}+\frac{1}{2}\,\mathsf{f}^{(1)}+\frac{1}{6}\,\mathsf{f}^{(2)}+\frac{1}{24}\,\mathsf{f}^{(3)}+\frac{1}{120}\,\mathsf{f}^{(4)}$$

where

$$
\begin{aligned}
\mathsf{u}^{(1)} &= -\frac{\Delta t}{\Delta x_e}\,\mathsf{Df} \\
\mathsf{f}^{(1)} &= \frac{1}{12}\left[-\boldsymbol{f}(\mathsf{u}+2\,\mathsf{u}^{(1)})+8\,\boldsymbol{f}(\mathsf{u}+\mathsf{u}^{(1)})-8\,\boldsymbol{f}(\mathsf{u}-\mathsf{u}^{(1)})+\boldsymbol{f}(\mathsf{u}-2\,\mathsf{u}^{(1)})\right] \\
\mathsf{u}^{(2)} &= -\frac{\Delta t}{\Delta x_e}\,\mathsf{Df}^{(1)} \\
\mathsf{f}^{(2)} &= \frac{1}{12}\left[-\boldsymbol{f}\left(\mathsf{u}+2\mathsf{u}^{(1)}+\frac{2^2}{2!}\mathsf{u}^{(2)}\right)+16\,\boldsymbol{f}\left(\mathsf{u}+\mathsf{u}^{(1)}+\frac{1}{2!}\mathsf{u}^{(2)}\right)-30\,\boldsymbol{f}(\mathsf{u})\right. \\
&\qquad \left.+16\,\boldsymbol{f}\left(\mathsf{u}-\mathsf{u}^{(1)}+\frac{1}{2!}\mathsf{u}^{(2)}\right)-\boldsymbol{f}\left(\mathsf{u}-2\mathsf{u}^{(1)}+\frac{2^2}{2!}\mathsf{u}^{(2)}\right)\right] \\
\mathsf{u}^{(3)} &= -\frac{\Delta t}{\Delta x_e}\,\mathsf{Df}^{(2)} \\
\mathsf{f}^{(3)} &= \frac{1}{2}\left[\boldsymbol{f}\left(\mathsf{u}+2\,\mathsf{u}^{(1)}+\frac{2^2}{2!}\,\mathsf{u}^{(2)}+\frac{2^3}{3!}\,\mathsf{u}^{(3)}\right)-2\,\boldsymbol{f}\left(\mathsf{u}+\mathsf{u}^{(1)}+\frac{1}{2!}\,\mathsf{u}^{(2)}+\frac{1}{3!}\,\mathsf{u}^{(3)}\right)\right. \\
&\qquad \left.+2\,\boldsymbol{f}\left(\mathsf{u}-\mathsf{u}^{(1)}+\frac{1}{2!}\,\mathsf{u}^{(2)}-\frac{1}{3!}\,\mathsf{u}^{(3)}\right)-\boldsymbol{f}\left(\mathsf{u}-2\,\mathsf{u}^{(1)}+\frac{2^2}{2!}\,\mathsf{u}^{(2)}-\frac{2^3}{3!}\,\mathsf{u}^{(3)}\right)\right] \\
\mathsf{u}^{(4)} &= -\frac{\Delta t}{\Delta x_e}\,\mathsf{Df}^{(3)} \\
\mathsf{f}^{(4)} &= \left[\boldsymbol{f}\left(\mathsf{u}+2\,\mathsf{u}^{(1)}+\frac{2^2}{2!}\,\mathsf{u}^{(2)}+\frac{2^3}{3!}\,\mathsf{u}^{(3)}+\frac{2^4}{4!}\,\mathsf{u}^{(4)}\right)\right. \\
&\qquad \left.-4\,\boldsymbol{f}\left(\mathsf{u}+\mathsf{u}^{(1)}+\frac{1}{2!}\,\mathsf{u}^{(2)}+\frac{1}{3!}\,\mathsf{u}^{(3)}+\frac{1}{4!}\,\mathsf{u}^{(4)}\right)+6\,\boldsymbol{f}(\mathsf{u})\right.
\end{aligned}
$$

$$-4\,\boldsymbol{f}\left(\mathsf{u}-\mathsf{u}^{(1)}+\frac{1}{2!}\,\mathsf{u}^{(2)}-\frac{1}{3!}\,\mathsf{u}^{(3)}+\frac{1}{4!}\,\mathsf{u}^{(4)}\right)$$
$$+\boldsymbol{f}\left(\mathsf{u}-2\,\mathsf{u}^{(1)}+\frac{2^2}{2!}\,\mathsf{u}^{(2)}-\frac{2^3}{3!}\,\mathsf{u}^{(3)}+\frac{2^4}{4!}\,\mathsf{u}^{(4)}\right)\Bigg]$$

The above set of formulae shows the sequence of steps that have to be performed to compute the time average flux at various orders. The arguments of the fluxes used on the right hand side in these steps are built in a sequential manner. Note that all the equations are vectorial equations and are applied at each solution point.

4.3. NUMERICAL FLUX

The numerical flux couples the solution between two neighbouring cells in a discontinuous Galerkin type method. In RK methods, the numerical flux is a function of the trace values of the solution at the faces. In the Lax-Wendroff scheme, we have constructed the time average flux at all the solution points inside the element and we want to use this information to compute the time averaged numerical flux at the element faces. The simplest numerical flux is based on Lax-Friedrich type approximation and is given by [137]

$$\boldsymbol{F}_{e+\frac{1}{2}} = \frac{1}{2}[\boldsymbol{F}^-_{e+\frac{1}{2}} + \boldsymbol{F}^+_{e+\frac{1}{2}}] - \frac{1}{2}\lambda_{e+\frac{1}{2}}\,[\boldsymbol{u}_h(x^+_{e+\frac{1}{2}}, t_n) - \boldsymbol{u}_h(x^-_{e+\frac{1}{2}}, t_n)] \tag{4.10}$$

which consists of a central flux and a dissipative part. For linear advection equation $\boldsymbol{u}_t + a\,\boldsymbol{u}_x = 0$, the coefficient in the dissipative part of the flux is taken as $\lambda_{e+\frac{1}{2}} = |a|$, while for a non-linear PDE like Burgers' equation, we take it to be

$$\lambda_{e+\frac{1}{2}} = \max\{|\boldsymbol{f}'(\bar{\boldsymbol{u}}^n_e)|, |\boldsymbol{f}'(\bar{\boldsymbol{u}}^n_{e+1})|\}$$

where $\bar{\boldsymbol{u}}^n_e$ is the cell average solution in element Ω_e at time t_n, and will be referred to as Rusanov or local Lax-Friedrich [152] approximation. Note that the dissipation term in the above numerical flux is evaluated at time t_n whereas the central part of the flux uses the time average flux. Since the dissipation term contains the solution difference at faces, we still expect to obtain optimal convergence rates, which is verified in numerical experiments. This numerical flux depends on the following quantities: $\{\bar{\boldsymbol{u}}^n_e, \bar{\boldsymbol{u}}^n_{e+1}, \boldsymbol{u}_h\,(x^-_{e+\frac{1}{2}}, t_n), \boldsymbol{u}_h\,(x^+_{e+\frac{1}{2}}, t_n), \boldsymbol{F}^-_{e+\frac{1}{2}}, \boldsymbol{F}^+_{e+\frac{1}{2}}\}$.

The numerical flux of the form (4.10) leads to somewhat reduced CFL numbers as shown by Fourier stability analysis in a later section, and also does not have upwind property even for linear advection equation. An alternate form of the numerical flux is obtained by evaluating the dissipation term using the time average solution, leading to the formula

$$\boldsymbol{F}_{e+\frac{1}{2}} = \frac{1}{2}\,[\boldsymbol{F}^-_{e+\frac{1}{2}} + \boldsymbol{F}^+_{e+\frac{1}{2}}] - \frac{1}{2}\,\lambda_{e+\frac{1}{2}}\,[\boldsymbol{U}^+_{e+\frac{1}{2}} - \boldsymbol{U}^-_{e+\frac{1}{2}}] \tag{4.11}$$

where

$$\boldsymbol{U} = \sum_{m=0}^{N} \frac{\Delta t^m}{(m+1)!}\,\partial_t^m\,\boldsymbol{u} = \boldsymbol{u} + \frac{\Delta t}{2}\,\partial_t\,\boldsymbol{u} + \ldots + \frac{\Delta t^N}{(N+1)!}\,\partial_t^N\,\boldsymbol{u} \tag{4.12}$$

is the time average solution. In this case, the numerical flux depends on the following quantities: $\{\bar{\boldsymbol{u}}_e^n, \bar{\boldsymbol{u}}_{e+1}^n, \boldsymbol{U}^-_{e+\frac{1}{2}}, \boldsymbol{U}^+_{e+\frac{1}{2}}, \boldsymbol{F}^-_{e+\frac{1}{2}}, \boldsymbol{F}^+_{e+\frac{1}{2}}\}$. We will refer to the above two forms of dissipation as D1 and D2, respectively. The dissipation model D2 is not computationally expensive compared to the D1 model since all the quantities required to compute the time average solution $\boldsymbol{U}$ are available during the Lax-Wendroff procedure. Some numerical fluxes for the case of systems of hyperbolic equations are described in Appendix D. It remains to explain how to compute $\boldsymbol{F}^{\pm}_{e+\frac{1}{2}}$ appearing in the central part of the numerical flux, which can be accomplished in two different ways, which we term **AE** and **EA** in the next two sub-sections.

Remark 4.1. In case of constant linear advection equation, $\boldsymbol{u}_t + a\,\boldsymbol{u}_x = \boldsymbol{0}$, $\boldsymbol{f}^{(m)} = a\,\boldsymbol{u}^{(m)}$ so that $\boldsymbol{F}_{e,p} = a\,\boldsymbol{U}_{e,p}$. Then, since $\lambda_{e+\frac{1}{2}} = |a|$, the numerical flux (4.11) becomes the upwind flux

$$\boldsymbol{F}_{e+\frac{1}{2}} = \begin{cases} \boldsymbol{F}_h^\delta(x^-_{e+\frac{1}{2}}), & a \geq 0 \\ \boldsymbol{F}_h^\delta(x^+_{e+\frac{1}{2}}), & a < 0 \end{cases}$$

but the flux (4.10) does not have this upwind property. For a variable coefficient advection problem with flux $\boldsymbol{f} = a(x)\,\boldsymbol{u}$, we get $\boldsymbol{F}^e_p = a(x_p)\,\boldsymbol{U}^e_p$, the numerical flux (4.11) is

$$\boldsymbol{F}_{e+\frac{1}{2}} = \frac{1}{2}\left[\boldsymbol{F}^-_{e+\frac{1}{2}} + \boldsymbol{F}^+_{e+\frac{1}{2}}\right] - \frac{1}{2}\left|a(x_{e+\frac{1}{2}})\right|\left[\boldsymbol{U}^+_{e+\frac{1}{2}} - \boldsymbol{U}^-_{e+\frac{1}{2}}\right] \tag{4.13}$$

which does not reduce to an upwind flux due to interpolation errors, though it will be close to it in the well resolved cases. In this case, we can define the upwind numerical flux as

$$\boldsymbol{F}_{e+\frac{1}{2}} = \begin{cases} \boldsymbol{F}^-_{e+\frac{1}{2}}, & a(x_{e+\frac{1}{2}}) \geq 0 \\ \boldsymbol{F}^+_{e+\frac{1}{2}}, & a(x_{e+\frac{1}{2}}) < 0 \end{cases} \tag{4.14}$$

which is defined in terms of the time average flux only and does not make use of the solution.

Remark 4.2. For non-linear problems, we can also consider the global Lax-Friedrich and Roe type dissipation models which are given by

$$\lambda_{e+\frac{1}{2}} = \lambda = \max_e |\boldsymbol{f}'(\bar{\boldsymbol{u}}_e)|, \qquad \lambda_{e+\frac{1}{2}} = \left|\boldsymbol{f}'\Big(\frac{\bar{\boldsymbol{u}}_e + \bar{\boldsymbol{u}}_{e+1}}{2}\Big)\right|$$

respectively. In the global Lax-Friedrich flux, the maximum is taken over the whole grid. For Burgers' equation, we can consider an Osher type flux [74] which is given by

$$\boldsymbol{F}_{e+\frac{1}{2}} = \begin{cases} \boldsymbol{F}^-_{e+\frac{1}{2}} & \bar{\boldsymbol{u}}_e, \bar{\boldsymbol{u}}_{e+1} > 0 \\ \boldsymbol{F}^+_{e+\frac{1}{2}} & \bar{\boldsymbol{u}}_e, \bar{\boldsymbol{u}}_{e+1} < 0 \\ \boldsymbol{F}^-_{e+\frac{1}{2}} + \boldsymbol{F}^+_{e+\frac{1}{2}} & \bar{\boldsymbol{u}}_e \geq 0 \geq \bar{\boldsymbol{u}}_{e+1} \\ 0 & \text{otherwise} \end{cases}$$

4.3.1. Numerical flux – average and extrapolate to face (AE)

In each element, the time average flux $\boldsymbol{F}_h^\delta$ has been constructed using the Lax-Wendroff procedure. The simplest approximation that can be used for $\boldsymbol{F}^{\pm}_{e+\frac{1}{2}}$ in the central part of the numerical flux is to extrapolate the flux $\boldsymbol{F}_h^\delta$ to the faces,

$$\boldsymbol{F}^{\pm}_{e+\frac{1}{2}} = \boldsymbol{F}_h^\delta(x^{\pm}_{e+\frac{1}{2}})$$

We will refer to this approach with the abbreviation **AE**. However, as shown in the numerical results, this approximation can lead to sub-optimal convergence rates for some non-linear problems. Hence we propose another method for the computation of the inter-cell flux which overcomes this problem as explained next.

4.3.2. Numerical flux – extrapolate to face and average (EA)

Instead of extrapolating the time average flux from the solution points to the faces, we can instead build the time average flux at the faces directly using the approximate Lax-Wendroff procedure that is used at the solution points. The flux at the faces is constructed after the solution is evolved at all the solution points. In the following equations, α denotes either the left face (L) or the right face (R) of a cell. For $\alpha \in \{L, R\}$, we compute the time average flux at the faces of the e'th element by the following steps, where we suppress the element index since all the operations are performed inside one element.

Degree $N=1$

$$\begin{aligned}
\boldsymbol{u}_\alpha &= \mathsf{V}_\alpha^\top \mathsf{u} \\
\boldsymbol{u}_\alpha^\pm &= \mathsf{V}_\alpha^\top (\mathsf{u} \pm \mathsf{u}^{(1)}) \\
\boldsymbol{f}_\alpha^{(1)} &= \tfrac{1}{2}\left[\boldsymbol{f}(u_\alpha^+) - \boldsymbol{f}(u_\alpha^-)\right] \\
\boldsymbol{F}_\alpha &= \boldsymbol{f}(u_\alpha) + \tfrac{1}{2}\,\boldsymbol{f}_\alpha^{(1)}
\end{aligned}$$

Degree $N=2$

$$\begin{aligned}
\boldsymbol{u}_\alpha &= \mathsf{V}_\alpha^\top \mathsf{u} \\
\boldsymbol{u}_\alpha^\pm &= \mathsf{V}_\alpha^\top (\mathsf{u} \pm \mathsf{u}^{(1)} + \tfrac{1}{2}\,\mathsf{u}^{(2)}) \\
\boldsymbol{f}_\alpha^{(1)} &= \tfrac{1}{2}\left[\boldsymbol{f}(u_\alpha^+) - \boldsymbol{f}(u_\alpha^-)\right] \\
\boldsymbol{f}_\alpha^{(2)} &= \boldsymbol{f}(\boldsymbol{u}_\alpha^+) - 2\,\boldsymbol{f}(\boldsymbol{u}_\alpha) + \boldsymbol{f}(\boldsymbol{u}_\alpha^-) \\
\boldsymbol{F}_\alpha &= \boldsymbol{f}(\boldsymbol{u}_\alpha) + \tfrac{1}{2}\,\boldsymbol{f}_\alpha^{(1)} + \tfrac{1}{6}\,\boldsymbol{f}_\alpha^{(2)}
\end{aligned}$$

Degree $N=3$

$$\begin{aligned}
u_\alpha &= \mathsf{V}_\alpha^\top \mathsf{u} \\
u_\alpha^\pm &= \mathsf{V}_\alpha^\top \left(\mathsf{u} \pm \mathsf{u}^{(1)} + \frac{1}{2!}\mathsf{u}^{(2)} \pm \frac{1}{3!}\mathsf{u}^{(3)} \right) \\
u_\alpha^{\pm 2} &= \mathsf{V}_\alpha^\top \left(\mathsf{u} \pm 2\mathsf{u}^{(1)} + \frac{2^2}{2!}\mathsf{u}^{(2)} \pm \frac{2^3}{3!}\mathsf{u}^{(3)} \right) \\
\boldsymbol{f}_\alpha^{(1)} &= \frac{1}{12}\left[-\boldsymbol{f}(u_\alpha^{+2}) + 8\,\boldsymbol{f}(u_\alpha^+) - 8\,\boldsymbol{f}(u_\alpha^-) + \boldsymbol{f}(u_\alpha^{-2})\right] \\
\boldsymbol{f}_\alpha^{(2)} &= \boldsymbol{f}(u_\alpha^-) - 2\,\boldsymbol{f}(u_\alpha) + \boldsymbol{f}(u_\alpha^+) \\
\boldsymbol{f}_\alpha^{(3)} &= \frac{1}{2}\left[\boldsymbol{f}(u_\alpha^{+2}) - 2\,\boldsymbol{f}(u_\alpha^+) + 2\,\boldsymbol{f}(u_\alpha^-) - \boldsymbol{f}(u_\alpha^{-2})\right] \\
\boldsymbol{F}_\alpha &= \boldsymbol{f}(u_\alpha) + \frac{1}{2}\,\boldsymbol{f}_\alpha^{(1)} + \frac{1}{6}\,\boldsymbol{f}_\alpha^{(2)} + \frac{1}{24}\,\boldsymbol{f}_\alpha^{(3)}
\end{aligned}$$

Degree $N = 4$

$$
\begin{aligned}
u_\alpha &= \mathsf{V}_\alpha^\top \mathsf{u} \\
u_\alpha^{\pm} &= \mathsf{V}_\alpha^\top \left(\mathsf{u} \pm \mathsf{u}^{(1)} + \frac{1}{2!}\mathsf{u}^{(2)} \pm \frac{1}{3!}\mathsf{u}^{(3)} + \frac{1}{4!}\mathsf{u}^{(4)} \right) \\
u_\alpha^{\pm 2} &= \mathsf{V}_\alpha^\top \left(\mathsf{u} \pm 2\mathsf{u}^{(1)} + \frac{2^2}{2!}\mathsf{u}^{(2)} \pm \frac{2^3}{3!}\mathsf{u}^{(3)} + \frac{2^3}{3!}\mathsf{u}^{(3)} \right) \\
\boldsymbol{f}_\alpha^{(1)} &= \frac{1}{12}\left[-\boldsymbol{f}(u_\alpha^{+2}) + 8\,\boldsymbol{f}(u_\alpha^{+}) - 8\,\boldsymbol{f}(u_\alpha^{-}) + \boldsymbol{f}(u_\alpha^{-2}) \right] \\
\boldsymbol{f}_\alpha^{(2)} &= \frac{1}{12}\left[-\boldsymbol{f}(u_\alpha^{+2}) + 16\,\boldsymbol{f}(u_\alpha^{+}) - 30\,\boldsymbol{f}(u_\alpha) + 16\,\boldsymbol{f}(u_\alpha^{-}) - \boldsymbol{f}(u_\alpha^{-2}) \right] \\
\boldsymbol{f}_\alpha^{(3)} &= \frac{1}{2}\left[\boldsymbol{f}(u_\alpha^{+2}) - 2\,\boldsymbol{f}(u_\alpha^{+}) + 2\,\boldsymbol{f}(u_\alpha^{-}) - \boldsymbol{f}(u_\alpha^{-2}) \right] \\
\boldsymbol{f}_\alpha^{(4)} &= \left[\boldsymbol{f}(u_\alpha^{+2}) - 4\,\boldsymbol{f}(u_\alpha^{+}) + 6\,\boldsymbol{f}(u_\alpha) - 4\,\boldsymbol{f}(u_\alpha^{-}) + \boldsymbol{f}(u_\alpha^{-2}) \right] \\
\boldsymbol{F}_\alpha &= \boldsymbol{f}(u_\alpha) + \frac{1}{2}\,\boldsymbol{f}_\alpha^{(1)} + \frac{1}{6}\,\boldsymbol{f}_\alpha^{(2)} + \frac{1}{24}\,\boldsymbol{f}_\alpha^{(3)} + \frac{1}{120}\,\boldsymbol{f}_\alpha^{(4)}
\end{aligned}
$$

We see that the solution is first extrapolated to the cell faces and the same finite difference formulae for the time derivatives of the flux which are used at the solution points, are also used at the faces. The numerical flux is computed using the time average flux built as above at the faces; the central part of the flux $\boldsymbol{F}^{\pm}_{e+\frac{1}{2}}$ in equations (4.10), (4.11) are computed as

$$\boldsymbol{F}^{-}_{e+\frac{1}{2}} = (\boldsymbol{F}_R)_e, \qquad \boldsymbol{F}^{+}_{e+\frac{1}{2}} = (\boldsymbol{F}_L)_{e+1}$$

We will refer to this method with the abbreviation **EA**.

Remark 4.3. The two methods **AE** and **EA** are different only when there are no solution points at the faces. E.g., if we use GLL solution points, then the two methods yield the same result since there is no interpolation error. For the constant coefficient advection equation, the above two schemes for the numerical flux lead to the same approximation but they differ in case of variable coefficient advection problems and when the flux is non-linear with respect to u. The effect of this non-linearity and the performance of the two methods are shown later using some numerical experiments.

4.4. Fourier stability analysis in 1-D

We now perform Fourier stability analysis of the LW schemes applied to the linear advection equation $u_t + a\,u_x = 0$ where a is the constant advection speed. We will assume that the advection speed a is positive and denote the CFL number by

$$\sigma = \frac{a\,\Delta t}{\Delta x}$$

Since $f^{(m)} = a\,u^{(m)}$, the time average flux at all the solution points is given by

$$\mathsf{F}_e = a\,\mathsf{U}_e \quad \text{where} \quad \mathsf{U}_e = \mathsf{T}\mathsf{u}_e \quad \text{and} \quad \mathsf{T} = \sum_{m=0}^{N} \frac{1}{(m+1)!}\,(-\sigma\,\mathsf{D})^m$$

Then the discontinuous flux at the cell boundaries are given by

$$F_h^\delta(x^{+}_{e-\frac{1}{2}}) = \mathsf{V}_L^\top \mathsf{F}_e, \qquad F_h^\delta(x^{-}_{e+\frac{1}{2}}) = \mathsf{V}_R^\top \mathsf{F}_e$$

We can write the update equation in the form

$$\mathsf{u}_e^{n+1} = -\sigma \mathsf{A}_{-1}\mathsf{u}_{e-1}^n + (\mathsf{I} - \sigma \mathsf{A}_0)\mathsf{u}_e^n - \sigma \mathsf{A}_{+1}\mathsf{u}_{e+1}^n \tag{4.15}$$

where the matrices $\mathsf{A}_{-1}, \mathsf{A}_0, \mathsf{A}_{+1}$ depend on the choice of the dissipation model in the numerical flux. The **EA** and **AE** schemes for the flux are identical for this linear problem, and hence we do not make any distinction between them for Fourier stability analysis.

Dissipation model D1. The numerical flux is given by

$$F_{e+\frac{1}{2}} = \frac{1}{2}[\mathsf{V}_R^\top \mathsf{F}_e + \mathsf{V}_L^\top \mathsf{F}_{e+1}] - \frac{1}{2} a\, (\mathsf{V}_L^\top \mathsf{u}_{e+1} - \mathsf{V}_R^\top \mathsf{u}_e)$$

Since the flux difference at the faces is

$$F_{e-\frac{1}{2}} - F_h^\delta(x^+_{e-\frac{1}{2}}) = \frac{1}{2} a\, \mathsf{V}_R^\top(\mathsf{T}+\mathsf{I})\mathsf{u}_{e-1} - \frac{1}{2} a\, \mathsf{V}_L^\top(\mathsf{T}+\mathsf{I})\mathsf{u}_e$$

$$F_{e+\frac{1}{2}} - F_h^\delta(x^-_{e+\frac{1}{2}}) = \frac{1}{2} a\, \mathsf{V}_L^\top(\mathsf{T}-\mathsf{I})\mathsf{u}_{e+1} - \frac{1}{2} a\, \mathsf{V}_R^\top(\mathsf{T}-\mathsf{I})\mathsf{u}_e$$

the flux derivative at the solution points is given by

$$\begin{aligned}\partial_\xi \mathsf{F}_h \;=\; & \frac{1}{2} a\, \mathsf{b}_L \mathsf{V}_R^\top(\mathsf{T}+\mathsf{I})\mathsf{u}_{e-1} + \left[a\, \mathsf{D}\mathsf{T} - \frac{1}{2} a\, \mathsf{b}_L \mathsf{V}_L^\top(\mathsf{T}+\mathsf{I}) - \frac{1}{2} a\, \mathsf{b}_R \mathsf{V}_R^\top(\mathsf{T}-\mathsf{I}) \right] \mathsf{u}_e \\ & + \frac{1}{2} a\, \mathsf{b}_R \mathsf{V}_L^\top(\mathsf{T}-\mathsf{I})\mathsf{u}_{e+1}\end{aligned}$$

Thus the matrices in (4.15) are given by

$$\mathsf{A}_{-1} = \frac{1}{2}\mathsf{b}_L \mathsf{V}_R^\top(\mathsf{T}+\mathsf{I}), \quad \mathsf{A}_{+1} = \frac{1}{2}\mathsf{b}_R \mathsf{V}_L^\top(\mathsf{T}-\mathsf{I}), \quad \mathsf{A}_0 = \mathsf{D}\mathsf{T} - \frac{1}{2}\mathsf{b}_L \mathsf{V}_L^\top(\mathsf{T}+\mathsf{I}) - \frac{1}{2}\mathsf{b}_R \mathsf{V}_R^\top(\mathsf{T}-\mathsf{I})$$

Dissipation model D2. Since $a > 0$, the numerical flux is given by

$$F_{e+\frac{1}{2}} = \mathsf{V}_R^\top \mathsf{F}_e = a\, \mathsf{V}_R^\top \mathsf{T}\mathsf{u}_e$$

and the flux differences at the face are

$$F_{e-\frac{1}{2}} - F_h^\delta(x^+_{e-\frac{1}{2}}) = a\, \mathsf{V}_R^\top \mathsf{T}\mathsf{u}_{e-1} - a\, \mathsf{V}_L^\top \mathsf{T}\mathsf{u}_e, \qquad F_{e+\frac{1}{2}} - F_h^\delta(x^-_{e+\frac{1}{2}}) = 0$$

so that the flux derivative at the solution points is given by

$$\partial_\xi \mathsf{F}_h = (a\, \mathsf{V}_R^\top \mathsf{T}\mathsf{u}_{e-1} - a\, \mathsf{V}_L^\top \mathsf{T}\mathsf{u}_e)\mathsf{b}_L + a\, \mathsf{D}\mathsf{T}\mathsf{u}_e = a\, \mathsf{b}_L \mathsf{V}_R^\top \mathsf{T}\mathsf{u}_{e-1} + a\, (\mathsf{D}\mathsf{T} - \mathsf{b}_L \mathsf{V}_L^\top \mathsf{T})\mathsf{u}_e$$

Thus the matrices in (4.15) are given by

$$\mathsf{A}_{-1} = \mathsf{b}_L \mathsf{V}_R^\top \mathsf{T}, \qquad \mathsf{A}_0 = \mathsf{D}\mathsf{T} - \mathsf{b}_L \mathsf{V}_L^\top \mathsf{T}, \qquad \mathsf{A}_{+1} = \mathbf{0}$$

The upwind character of the flux leads to $\mathsf{A}_+ = \mathbf{0}$ and the right cell does not appear in the update equation.

Stability analysis. We assume a solution of the form $\mathsf{u}_e^n = \hat{\mathsf{u}}_k^n \exp(\mathsf{i} k\, x_e)$ where $\mathsf{i} = \sqrt{-1}$, k is the wave number and $\hat{\mathsf{u}}_k^n \in \mathbb{R}^{N+1}$ are the Fourier amplitudes; substituting this ansatz in (4.15), we find that the amplitudes evolve according to the equation

$$\hat{\mathsf{u}}_k^{n+1} = \mathsf{H}(\sigma, \kappa)\, \hat{\mathsf{u}}_k^n, \qquad \mathsf{H} = \mathsf{I} - \sigma\, \mathsf{A}_0 - \sigma\, \mathsf{A}_{-1} \exp(-\mathsf{i}\kappa) - \sigma\, \mathsf{A}_{+1} \exp(\mathsf{i}\kappa), \qquad \kappa = k\, \Delta x$$

where κ is the non-dimensional wave number. The eigenvalues of H depend on the CFL number σ and the non-dimensional wave number κ, i.e., $\lambda = \lambda(\sigma, \kappa)$; for stability, all the eigenvalues of H must have magnitude less than or equal to one for all $\kappa \in [0, 2\pi]$, i.e.,

$$\lambda(\sigma) = \max_{\kappa} |\lambda(\sigma, \kappa)| \leq 1$$

The CFL number is the maximum value of σ for which the above stability condition is satisfied. This CFL number is determined approximately by sampling in the wave number space; we partition $\kappa \in [0, 2\pi]$ into a large number of uniformly spaced points κ_p and determine

$$\lambda(\sigma) = \max_{p} |\lambda(\sigma, \kappa_p)|$$

The values of σ are also sampled in some interval $[\sigma_{\min}, \sigma_{\max}]$ and the largest value of $\sigma_l \in [\sigma_{\min}, \sigma_{\max}]$ for which $\lambda(\sigma_l) \leq 1$ is determined in a Python code. We start with a large interval $[\sigma_{\min}, \sigma_{\max}]$ and then progressively reduce the size of this interval so that the CFL number is determined to about three decimal places. The results of this numerical investigation of stability are shown in Table 4.1 for two correction functions and different polynomial degrees.

N	Radau			g_2		
	D1	D2	Ratio	D1	D2	Ratio
1	0.226	0.333	1.47	0.465	1.000	2.15
2	0.117	0.170	1.45	0.204	0.333	1.63
3	0.072	0.103	1.43	0.116	0.170	1.47
4	0.049	0.069	1.40	0.060	0.103	1.72

Table 4.1. CFL numbers for 1-D LWFR using the two dissipation models and correction functions

We see that dissipation model D2 has a higher CFL number compared to dissipation model D1. The CFL numbers for the g_2 correction function are also significantly higher than those for the Radau correction function. The LWFR scheme with Radau correction function is identical to DG scheme and the CFL numbers found here agree with those from the ADER-DG scheme [63, 76]. The optimality of these CFL numbers has been verified by experiment on the linear advection test case (Section 4.7.1), i.e., the solution eventually blows up if the time step is slightly higher than what is allowed by the CFL condition.

4.5. Boundary conditions

The boundary conditions for hyperbolic problems are usually implemented in a weak manner through the fluxes (4.11). We explain the implementation for the 1-D scheme which is applied to higher dimensions across normal direction. Consider a grid $\{\Omega_e\}_{e=1}^{M}$ where Ω_1, Ω_M are the left, right boundary elements. In some cases, the boundary conditions are enforced by the choice of *ghost values* which are $\boldsymbol{F}^{+}_{M+\frac{1}{2}}, \boldsymbol{U}^{+}_{M+\frac{1}{2}}, \bar{\boldsymbol{u}}_{M+1}$ for the right boundary and $\boldsymbol{F}^{-}_{\frac{1}{2}}, \boldsymbol{U}^{-}_{\frac{1}{2}}, \bar{\boldsymbol{u}}_0$ for the left boundary. Here we describe the treatment in various cases.

Periodic boundary.

$$\begin{aligned} \boldsymbol{F}^+_{M+\frac{1}{2}}, \boldsymbol{U}^+_{M+\frac{1}{2}}, \bar{\boldsymbol{u}}_{M+1} &= \boldsymbol{F}^+_{\frac{1}{2}}, \boldsymbol{U}^+_{\frac{1}{2}}, \bar{\boldsymbol{u}}_1 \\ \boldsymbol{F}^-_{\frac{1}{2}}, \boldsymbol{U}^-_{\frac{1}{2}}, \bar{\boldsymbol{u}}_0 &= \boldsymbol{F}^-_{M+\frac{1}{2}}, \boldsymbol{U}^-_{M+\frac{1}{2}}, \bar{\boldsymbol{u}}_M \end{aligned}$$

The numerical flux at boundary face can now be computed with (4.11).

Dirichlet/Inflow boundary. The boundary condition on the solution can be specified only at inflow boundaries, i.e., where the characteristics are entering the domain. For example, at the left boundary of the domain, say $x_{1/2}=0$, the boundary condition can be specified if $f'>0$ for a scalar problem and if eigenvalues of $\boldsymbol{f}'$ are positive for system of equations. Assuming this is the case for our problem, let the boundary condition be given as $\boldsymbol{u}(0,t)=\boldsymbol{g}(t)$. It will be enforced by defining an upwind numerical flux at the boundary face, which is given by

$$\boldsymbol{F}_{\frac{1}{2}} = \boldsymbol{F}^-_{\frac{1}{2}} \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \boldsymbol{f}(\boldsymbol{g}(t))\,\mathrm{d}t$$

If the integral cannot be computed analytically, then it is approximated by quadrature in time. From (4.3), we see that integral must be at least accurate to $O(\Delta t^{N+1})$ which is of the same order as the neglected terms in (4.3). In the numerical tests, we use $(N+1)$-point Gauss-Legendre quadrature which ensures the required accuracy. In this case, we specify the numerical flux at boundary face directly, and do not need to compute the numerical flux using (4.11).

Outflow boundary. The right boundary is an outflow boundary if eigenvalues of $\boldsymbol{f}'$ are positive. In this case, the flux across the right boundary is computed in an upwind manner using the interior solution, i.e., $\boldsymbol{F}_{M+1/2}=\boldsymbol{F}^-_{M+1/2}$ where $\boldsymbol{F}^-_{M+1/2}$ is obtained from the Lax-Wendroff procedure.

Numerical flux for boundaries. There are cases when the characteristics are a mix of inflow and outflow, and it is known that the inflow is given by a function $\boldsymbol{g}(t)$. In these cases, we use an upwind numerical flux like Roe (Appendix D) which will distinguish between inflow and outflow characteristics. We explain the treatment for the left boundary, say $x_{1/2}=0$. The time averaging of inflow quantities is performed to obtain the ghost values as follows

$$\boldsymbol{F}^-_{\frac{1}{2}} \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \boldsymbol{f}(\boldsymbol{g}(t))\,\mathrm{d}t, \qquad \boldsymbol{U}^-_{\frac{1}{2}} \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \boldsymbol{g}(t)\,\mathrm{d}t, \qquad \bar{\boldsymbol{u}}_0 = \bar{\boldsymbol{u}}_1$$

Then, the numerical flux at $x_{1/2}$ is computed as in (4.11) but now with an upwind flux from Appendix D.

Solid wall / reflective boundaries. This is a type of boundary condition for compressible Euler's equations (4.16). The velocity at a solid wall interface, say $x_{1/2}$, is set to zero. To satisfy the property in the numerical flux, we reflect the velocity along the origin in the ghost values. For brevity, we define $\boldsymbol{F},\boldsymbol{U},\boldsymbol{u} := \boldsymbol{F}^+_{\frac{1}{2}}, \boldsymbol{U}^+_{\frac{1}{2}}, \bar{\boldsymbol{u}}_1$ and set the ghost values as follows

$$\boldsymbol{F}^-_{\frac{1}{2}} = (-\boldsymbol{F}_1, -\boldsymbol{F}_2, \boldsymbol{F}_3), \quad \boldsymbol{U}^-_{\frac{1}{2}} = (\boldsymbol{U}_1, -\boldsymbol{U}_2, \boldsymbol{U}_3), \quad \bar{\boldsymbol{u}}_0 = (\boldsymbol{u}_1, -\boldsymbol{u}_2, \boldsymbol{u}_3)$$

Then, the numerical flux at $x_{1/2}$ is computed as in (4.11).

4.6. TVD LIMITER

The computation of discontinuous solutions with high order methods can give oscillatory solutions which must be limited by some non-linear process. The a posteriori limiters developed in the context of RKDG schemes [52, 51] can be adopted in the framework of LWFR schemes. The limiter is applied in a post-processing step after the solution is updated to the new time level. The limiter is thus applied only once for each time step unlike in RKDG scheme where it has to be applied after each RK stage update. Let $\boldsymbol{u}_h(x)$ denote the solution at time t_{n+1}. In element Ω_e, let the average solution be $\bar{\boldsymbol{u}}_e$; define the backward and forward differences of the solution and cell averages by

$$\Delta^- \boldsymbol{u}_e = \bar{\boldsymbol{u}}_e - \boldsymbol{u}_h(x^+_{e-\frac{1}{2}}), \qquad \Delta^+ \boldsymbol{u}_e = \boldsymbol{u}_h(x^-_{e+\frac{1}{2}}) - \bar{\boldsymbol{u}}_e$$

$$\Delta^- \bar{\boldsymbol{u}}_e = \bar{\boldsymbol{u}}_e - \bar{\boldsymbol{u}}_{e-1}, \qquad \Delta^+ \bar{\boldsymbol{u}}_e = \bar{\boldsymbol{u}}_{e+1} - \bar{\boldsymbol{u}}_e$$

We limit the solution by comparing its variation within the cell with the difference of the neighbouring cell averages through a limiter function,

$$\Delta^- \boldsymbol{u}_e^m = \text{minmod}(\Delta^- \boldsymbol{u}_e, \Delta^- \bar{\boldsymbol{u}}_e, \Delta^+ \bar{\boldsymbol{u}}_e), \qquad \Delta^+ \boldsymbol{u}_e^m = \text{minmod}(\Delta^+ \boldsymbol{u}_e, \Delta^- \bar{\boldsymbol{u}}_e, \Delta^+ \bar{\boldsymbol{u}}_e)$$

which is defined for each component as

$$\text{minmod}(a,b,c) = \begin{cases} s \min(|a|,|b|,|c|), & \text{if } s = \text{sign}(a) = \text{sign}(b) = \text{sign}(c) \\ 0, & \text{otherwise} \end{cases}$$

If $\Delta^- \boldsymbol{u}_e^m \neq \Delta^- \boldsymbol{u}_e$ or $\Delta^+ \boldsymbol{u}_e^m \neq \Delta^+ \boldsymbol{u}_e$, then the solution is deemed to be locally oscillatory and we modify the solution inside the cell by replacing it as a linear polynomial with a limited slope, which is taken to be the average limited slope. The limited solution polynomial in cell Ω_e is given by

$$\boldsymbol{u}_h|_{\Omega_e} = \overline{\boldsymbol{u}}_e + \frac{\Delta^- \boldsymbol{u}_e^m + \Delta^+ \boldsymbol{u}_e^m}{2}(2\,\xi - 1), \qquad \xi \in [0,1]$$

This limiter is known to clip smooth extrema since it cannot distinguish them from jump discontinuities. A small modification based on the idea of TVB limiters [52] can be used to relax the amount of limiting that is performed which leads to improved resolution of smooth extrema. The minmod function is replaced by

$$\overline{\text{minmod}}(a,b,c) = \begin{cases} a, & |a| \leq M\,\Delta x^2 \\ \text{minmod}(a,b,c), & \text{otherwise} \end{cases}$$

which requires the choice of a parameter M, which is an estimate of the second derivative of the solution at smooth extrema. In the case of systems of equations, the limiter is applied to the characteristic variables, which is known to yield better control on the spurious numerical oscillations [50]. The limiters used in this chapter are not able to provide high order accuracy and the development of better limiters, with the idea of a subcell based scheme, is discussed in Chapter 5.

Remark 4.4. For Runge-Kutta FR/DG schemes with a monotone numerical flux, the limiters of [50] can be used to obtain a Total Variation Diminishing (TVD) in means property. Although such a property cannot be shown for the LWFR scheme, the numerical results with the TVB limiter are similar for RKFR and LWFR schemes.

4.7. NUMERICAL RESULTS IN 1-D: SCALAR PROBLEMS

In this section, we present some numerical results to show the convergence rates and the comparison of different scheme parameters like correction function, solution points and dissipation model. For each problem in this section, the corresponding CFL number is chosen from Table 4.1. Here after, when we use the CFL numbers obtained using the Fourier stability analysis, we multiply it with a safety factor of 0.95. When D1 and D2 schemes are compared together, the CFL numbers of D1 schemes are used as these are lower; the same CFL numbers are used for the RKFR schemes. Up to degree $N = 3$, RKFR schemes use Runge-Kutta time integration of order $N+1$ with $N+1$ stages. In the $N = 4$ case, for non-linear problems there is no five stage Runge-Kutta method of order 5, see Chapter 32 of [35]. However, for linear, autonomous problems, the five stage SSPRK method in [82] is fifth order accurate, and we make use of it for the constant advection test cases with periodic boundary conditions and refer to it as SSPRK55. For non-linear or non-periodic problems, to make a fair comparison of LW and RK, we make use of the six stage, fifth order Runge-Kutta (RK65) time integration introduced in [183].

The RKFR and LWFR schemes are illustrated at a high level in Algorithm 4.1 and Algorithm 4.2, respectively, for solving a hyperbolic conservation law in a time interval $[0, T]$. Here we assume that an a posteriori limiter like a TVD/TVB limiter and a positivity limiter are applied in a post-processing step after the solution is updated. The LWFR scheme requires the application of the limiter only once per time step while the RKFR scheme requires multiple applications of the limiter depending on the number of RK stages. The limiter can be costly to apply for systems of equations where a characteristic approach and/or WENO-type limiter is used. In the present work, we use a simple TVD/TVB limiter but use characteristic limiting for systems.

Algorithm 4.1

Runge-Kutta Flux Reconstruction

```
t = 0;
while t < T do
    for each RK stage do
        Loop over cells and assemble rhs;
        Loop over faces and assemble rhs;
        Update solution to next RK stage;
        Apply a posteriori limiter;
        Apply positivity limiter;
    t = t + Δt;
```

Algorithm 4.2

Lax-Wendroff Flux Reconstruction

$t = 0$;
while $t < T$ **do**
    Loop over cells and assemble rhs;
    Loop over faces and assemble rhs;
    Update solution to next time step;
    Apply *a posteriori* limiter;
    Apply positivity limiter;
    $t = t + \Delta t$;

4.7.1. Linear advection equation: constant speed

We first consider the 1-D linear advection equation $u_t + a\, u_x = 0$ with speed $a = 1$ and periodic boundary condition.

4.7.1.1. Smooth solutions

For the initial condition $u(x, 0) = \sin(2\,\pi\, x)$ with periodic boundaries on $[0, 1]$, we perform grid convergence studies using various parameters like correction functions and solution points. The error norms are computed at time $t = 2$ units. In Figure 4.1 we compare the convergence behaviour for Radau and g_2 correction functions and for both choices of solution points using the D2 dissipation model. It is clear that the errors due to Radau are consistently smaller than those with g_2 correction function. The choice of the solution points does not significantly affect the error in the solution.

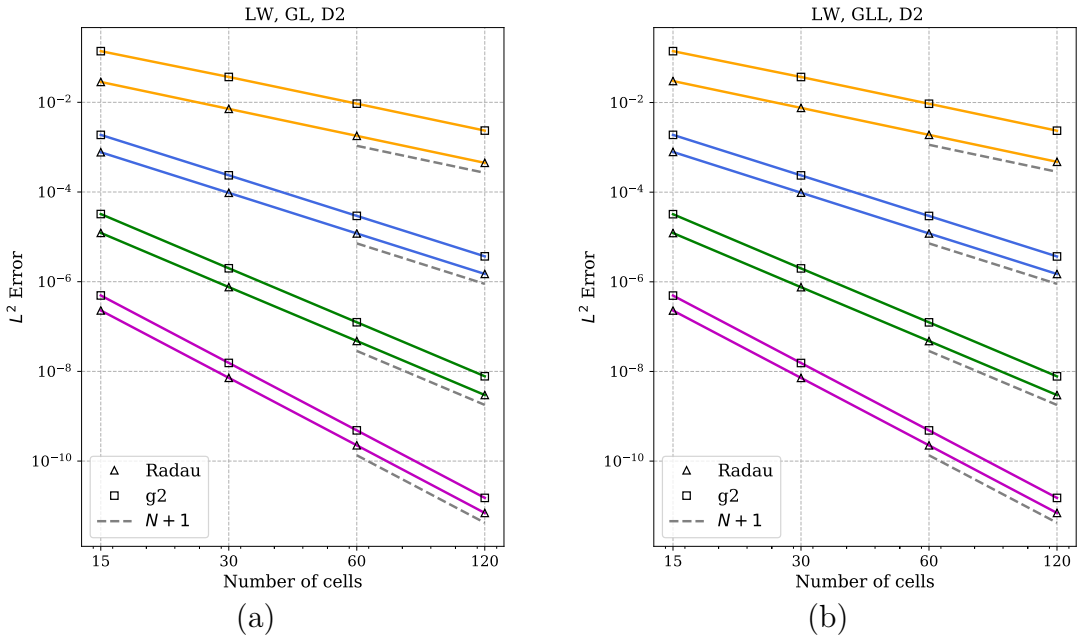


Figure 4.1. Error convergence for constant linear advection; (a) GL points, (b) GLL points. The different colors correspond to degrees $N = 1, 2, 3, 4$ from top to bottom.

Figure 4.2 shows the comparison of LW and RK schemes using Radau correction and two types of solution points. There is a small difference in the error levels between the two dissipation models, with the D2 model performing better for odd N. The RK scheme has slightly smaller errors than the LW scheme. We can see this more clearly by plotting the error norm versus time as shown in Figure 4.3, where all the four degrees consist of the same number of total dofs which is 200. We see that the error growth with time is higher for the LW scheme than for the RK scheme. The superior performance of the RK scheme is already known in the literature [84] and is due to its super-convergence properties. It is possible to construct LW schemes that are also super-convergent as done in [84] but the resulting schemes are computationally more expensive as they involve a stronger coupling with the neighbouring cell solutions, than what is used in the standard LW schemes. Hence we do not pursue that approach for our LW schemes. Note that this super-convergence occurs for constant coefficient linear problems on uniform grids and these advantages of RK schemes are not present when we consider non-linear problems, as shown in later results.

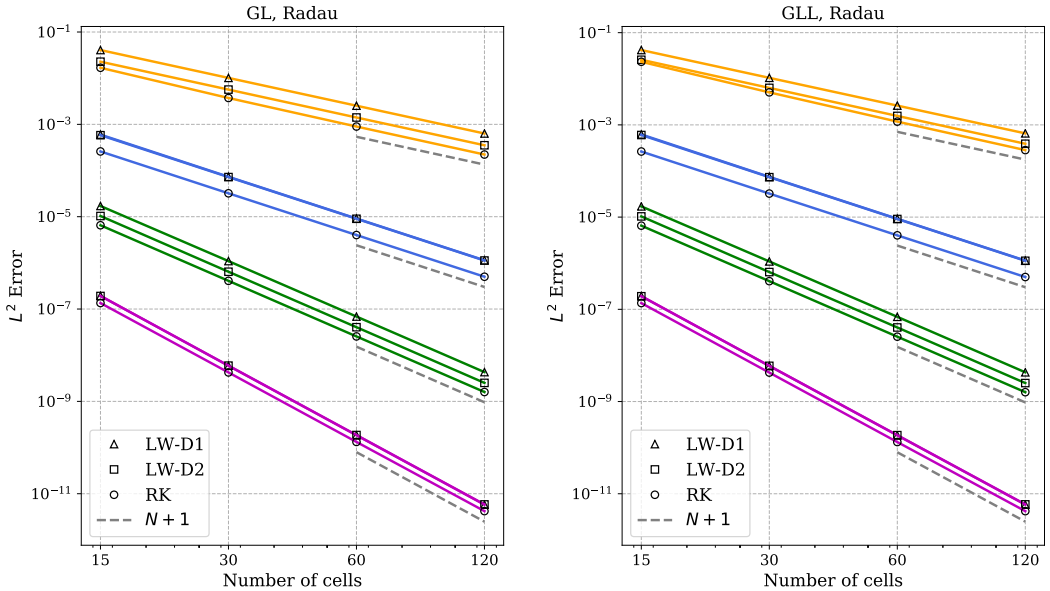


Figure 4.2. Error convergence for constant linear advection; (a) GL points , (b) GLL points. The different colors correspond to degrees $N = 1, 2, 3, 4$ from top to bottom.

Figure 4.4 analyzes the behaviour of L^2 norm of the solution where we plot the relative change in the L^2 norm with respect to the initial value, defined as $\|u_h(t)\|_2^* = \frac{\|u_h(t)\|_2 - \|u_h(0)\|_2}{\|u_h(0)\|_2}$. For $N = 1$, we see that LW is less dissipative than RK and thus better at conserving energy while for $N = 2, 3$, RK schemes perform better. For $N = 4$, we see a mild instability for both LWFR and RKFR schemes. For $N = 4$, we compare LW with SSPRK55 which is fifth order only for linear problems and SSPRK54 [167], which is more relevant for non-linear problems and is fourth order accurate. Choosing time step size by CFL numbers of the LW-D1 scheme, we observe the mild instability for

both the time integration schemes. The instability of RKDG scheme has been studied in [198] which can be remedied by using an RK scheme with different number of stages; however the use of six stage RK65 method with a limiter may dampen the solution too much, as we discuss later in Figure 4.11. The solution norm grows linearly, with a very small slope for both LW-D1 and LW-D2 (approximately 6.177e-10 and 5.415e-10) schemes, and also for SSPRK54 and SSPRK55 (approximately 2.862e-13, 1.908e-13) schemes. This type of mild instability for $N=4$ seems to be present in other single step methods like ADER-DG schemes also.

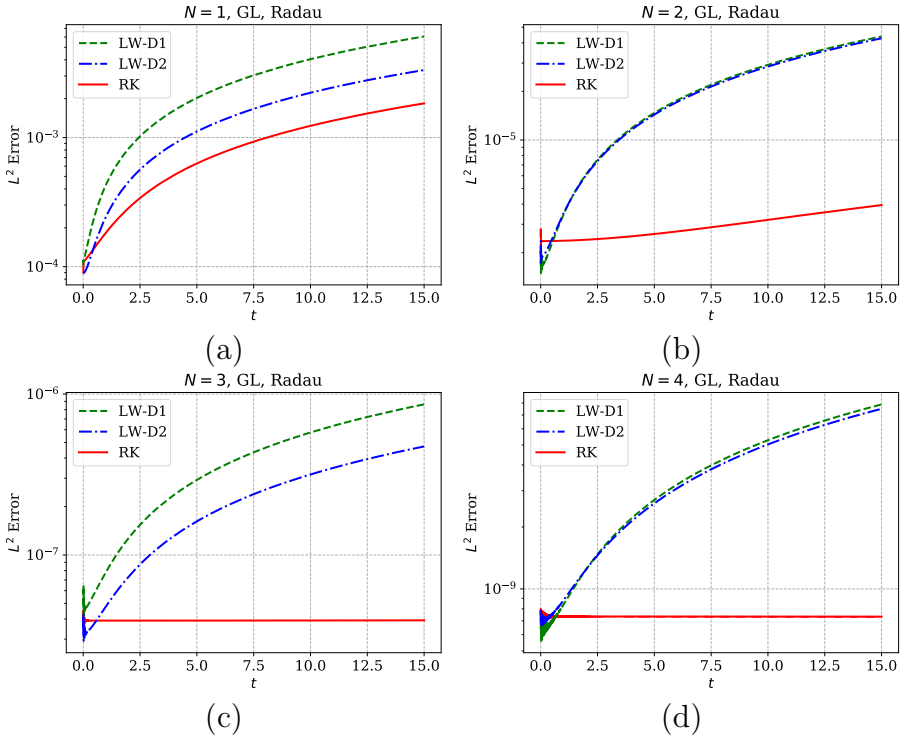


Figure 4.3. Error versus time for constant linear advection $u_t + u_x = 0$, initial condition $u(x,0) = \sin(2\pi x)$, $x \in [0,1]$, periodic boundary conditions, for different polynomial degrees, each with 200 degrees of freedom (dofs); GL solution points and Radau correction. (a) $N=1$, (b) $N=2$, (c) $N=3$, (d) $N=4$.

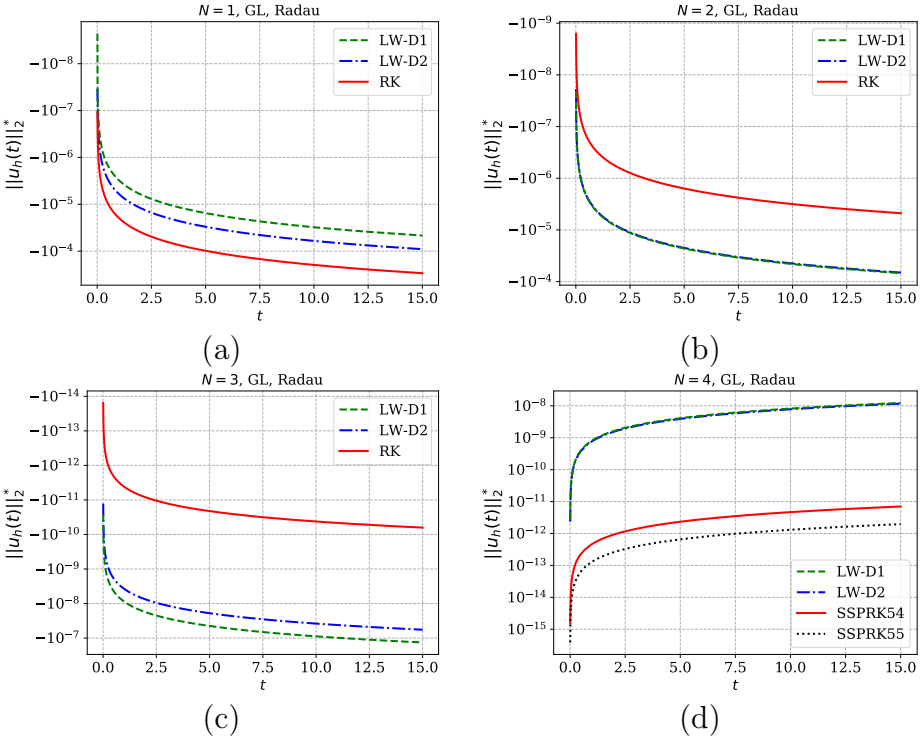


Figure 4.4. Semi-log plot of relative change in L_2 norm versus time for constant linear advection with the initial condition $u(x,0) = \sin(2\pi x), x \in [0,1]$ for different polynomial degrees, each with 200 dofs; GL solution points and Radau correction.

We now solve the problem with the same initial condition but using Dirichlet boundary condition at the left side of the domain; the exact solution remains same as before. The fifth order SSPRK scheme of [82] is only for autonomous systems, so here we use RK65 [183] for $N = 4$. Figure 4.5a shows the error convergence when we use the CFL of LW-D1 scheme for all schemes, since this is the smallest. All the schemes show optimal convergence rates with the LW-D2 and RK schemes showing very similar errors. In Figure 4.5b, we perform the same error convergence study but using the stable CFL number of each scheme; we still observe optimal convergence rates in each scheme, but the RK scheme shows slightly larger errors at degrees $N = 3, 4$, when the error level has become small. The issue with RK schemes may be related with the way Dirichlet conditions are implemented inside the RK stages as studied in [38]. Figure 4.6 shows the time history of the relative change in L^2 norm of the numerical solution; for degrees $N = 1, 2$ the norm does not increase relative to the initial value but for degrees $N = 3, 4$, there is some increase in the norm at initial times for some schemes. In Figure 4.6, we also make comparison of LW and RK for $N = 4$, with RK time integration performed with SSPRK54 [167]. However, in all cases, the norm does not grow monotonically but reaches a periodic oscillatory behaviour. Unlike the case of periodic boundary conditions, the inflow and outflow boundary conditions may lead to increase and decrease in energy respectively; if the two mechanisms aren't exactly balanced, we can observe the oscillatory behaviour in the solution norm.

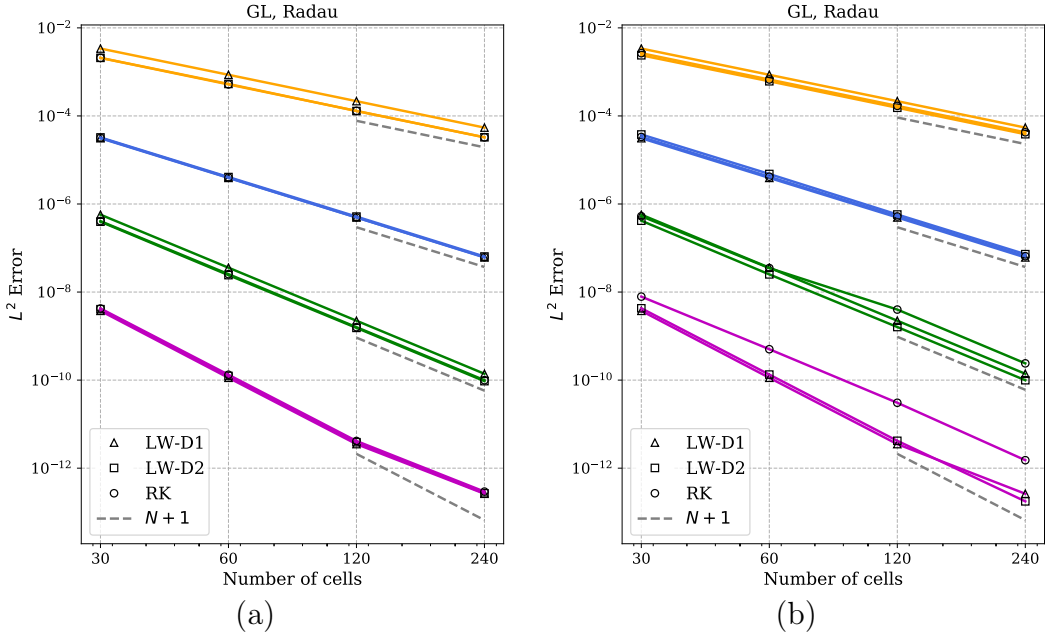


Figure 4.5. Convergence for constant linear advection with Dirichlet boundary conditions; (a) using CFL numbers of LW-D1 for all schemes, (b) using corresponding CFL number for each scheme. The different colors correspond to degrees N = 1, 2, 3, 4 from top to bottom.

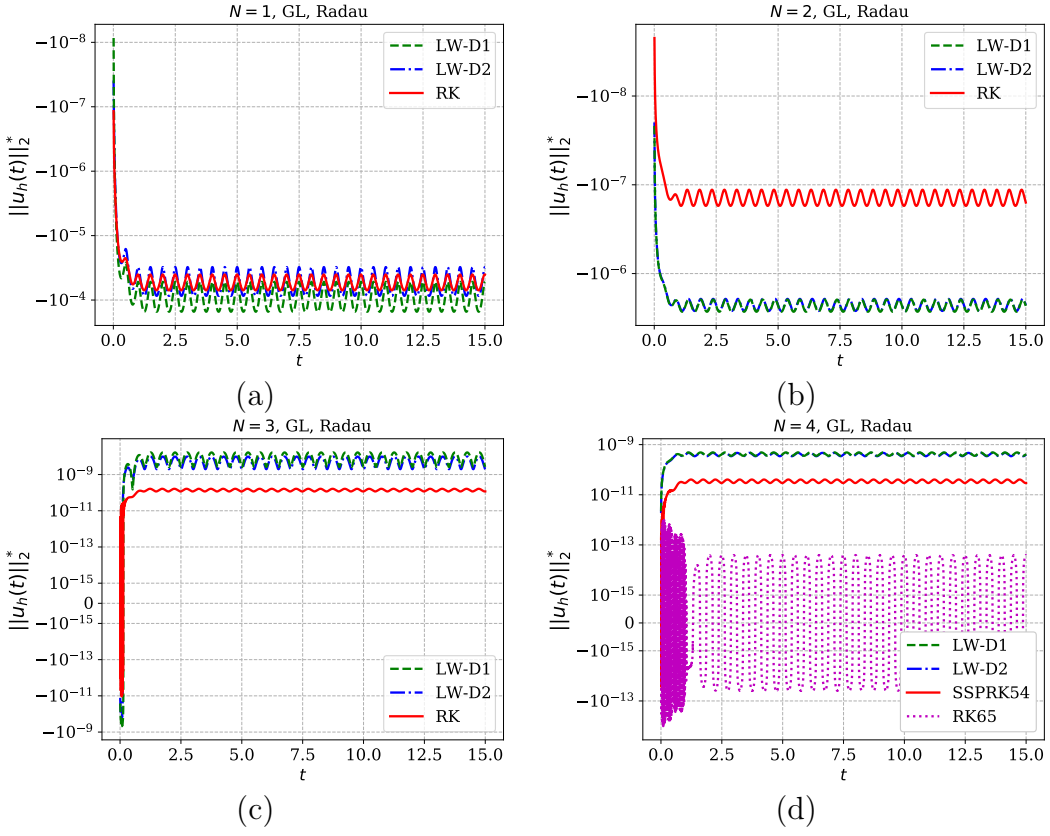
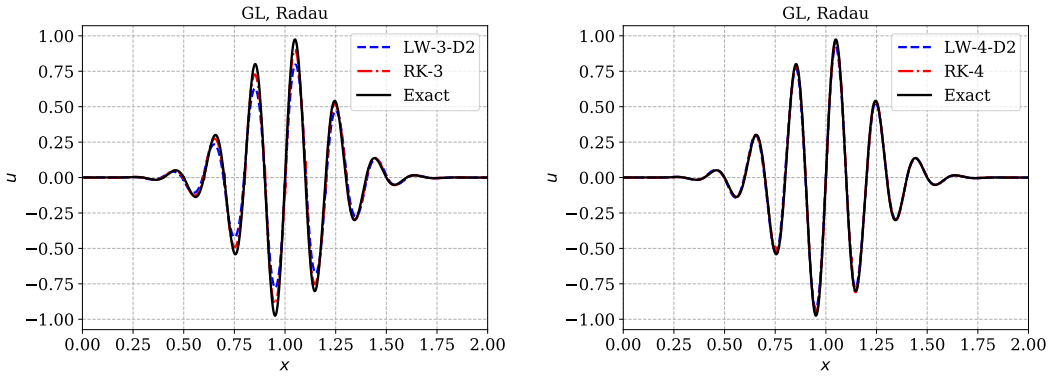


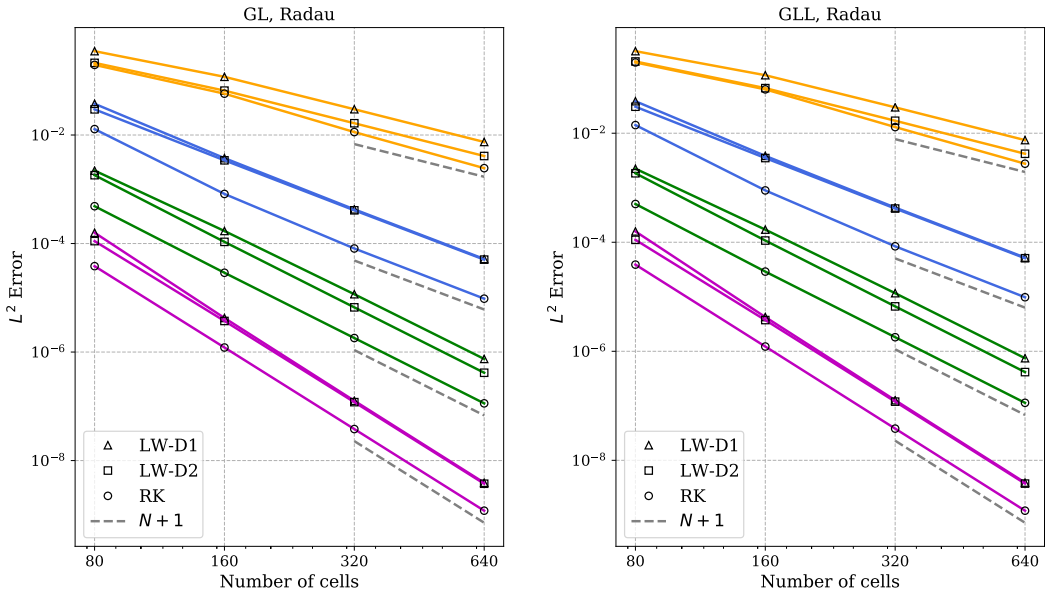
Figure 4.6. Semi-log plot of relative change in L_2 norm versus time for constant linear advection, with initial condition $u(x,0)=\sin(2\pi x), x\in[0,1]$ together with Dirichlet boundary conditions, for different polynomial degrees, each with 200 dofs; GL solution points and Radau correction.

Next we perform error convergence studies for an initial condition of a wave packet given by $u(x,0) = e^{-10x^2} \sin(10\,\pi\,x)$ with periodic boundary conditions. This initial condition has a more broadband Fourier spectrum than the previous case which had only one Fourier mode. Figure 4.7 shows the solutions obtained for $N = 3, 4$ and using 200 dofs in each case. The solutions are more accurate in case of $N = 4$ compared to $N = 3$ showing the benefits of a higher order method. We see that RK schemes are able to capture the peak solution more accurately than LW schemes, especially in case of $N = 3$, but the difference between the two schemes reduces for $N = 4$ case. Figure 4.8 shows the error convergence plot with GL points; as before, we see that RK schemes show smaller errors than the LW schemes due to their super-convergence property. For odd degrees, the D2 dissipation has slightly smaller errors than the D1 model, while for even degrees, the difference between the two models is negligible.



(a) (b)

Figure 4.7. Constant linear advection of a wave packet; solution at time $t=1$ with 160 dofs using polynomial degree (a) $N=3$, (b) $N=4$.



(a) (b)

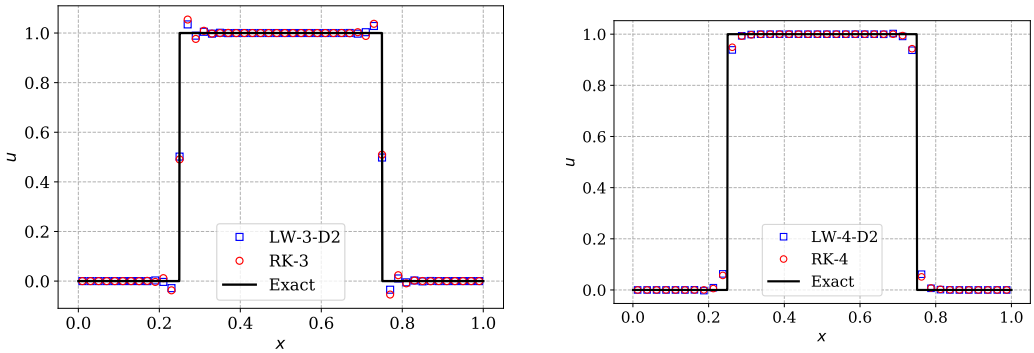
Figure 4.8. Error convergence for constant linear advection of a wave packet; (a) GL points, (b) GLL points. The different colors correspond to degrees $N=1, 2, 3, 4$ from top to bottom.

4.7.1.2. Non-smooth solutions

If the initial condition is not smooth and has a jump discontinuity, then high order methods will generate oscillatory solutions due to the non-monotone property of the schemes. For such problems, we need some form of limiter to control the oscillations and we use the TVB-type limiters which are applied in an *a posteriori* manner as explained in Section 4.6. Consider the initial condition consisting of a square hat function,

$$u(x,0) = \begin{cases} 1, & x \in (0.25, 0.75) \\ 0, & x \in [0, 0.25) \cup (0.75, 1] \end{cases}$$

and which is extended by periodicity. We compute the solution up to the time $t = 1$ unit when the solution returns to its initial position. Figure 4.9 shows the solutions obtained with degree $N = 3, 4$ and without applying any limiter. We observe oscillations in case of $N = 3$ but no significant oscillations are seen for the $N = 4$ case. The oscillations are however localized around the discontinuity and do not corrupt the rest of the solution. When TVB limiter is applied, these oscillations disappear as seen in Figure (4.10) but the jumps are smeared over more cells. If we use the RK65 scheme which is fifth order accurate but has six stages, then the results are shown in Figure (4.11) where we observe increased smearing of the jump in the RK scheme. Overall, we see that the limiter smears the discontinuity over a few cells in case of both LW and RK schemes; but we also observe that the solutions obtained with the LW schemes are very similar to the RK schemes.



(a) (b)

Figure 4.9. Constant linear advection of hat profile without limiter. The solution is shown at time t = 1 with 200 dofs using polynomial degree (a) $N = 3$, (b) $N = 4$.

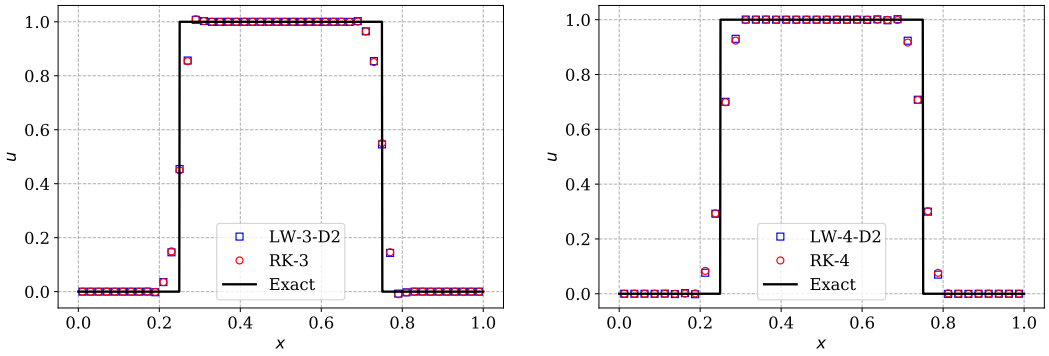


Figure 4.10. Constant linear advection of hat profile with TVB limiter ($M=100$). The solution is shown at time $t=1$ and 200 dofs using polynomial degree (a) $N=3$, (b) $N=4$.

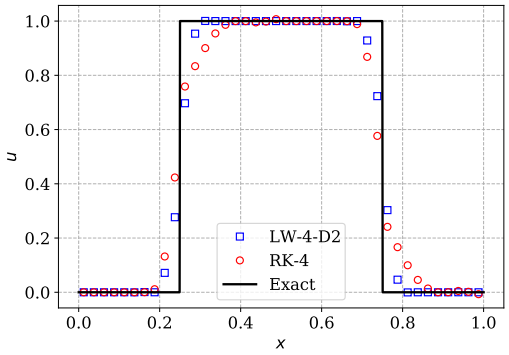
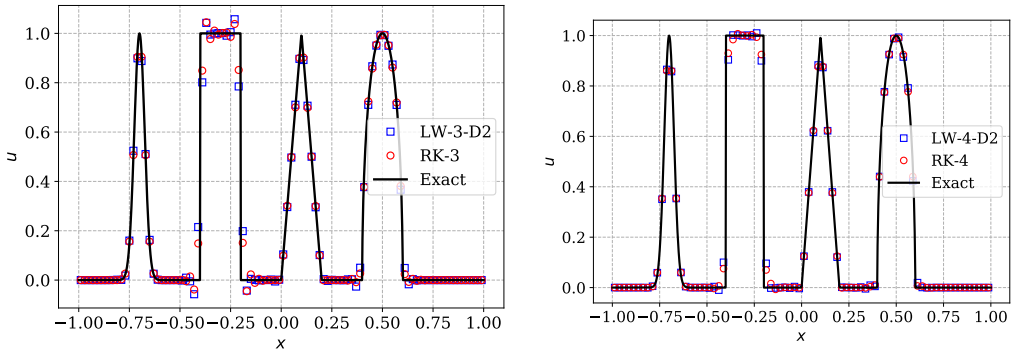


Figure 4.11. Constant linear advection of hat profile with TVB limiter ($M=100$) where Runge-Kutta time integration is performed using RK65 [183]. The solution is shown at time $t=1$ and 200 dofs using polynomial degree $N=4$.

We next consider a composite signal consisting of profiles with different levels of smoothness whose initial condition is a slightly different version of [98] given by

$$u(x,0)=\begin{cases} G\left(x,\beta,z\right), & -0.8\leq x\leq -0.6\\ 1, & -0.4\leq x\leq -0.2\\ 1-|10\left(x-0.1\right)|, & 0\leq x\leq 0.2\\ F\left(x,\alpha,a\right), & 0.4\leq x\leq 0.6\\ 0, & \text{elsewhere}\end{cases}$$

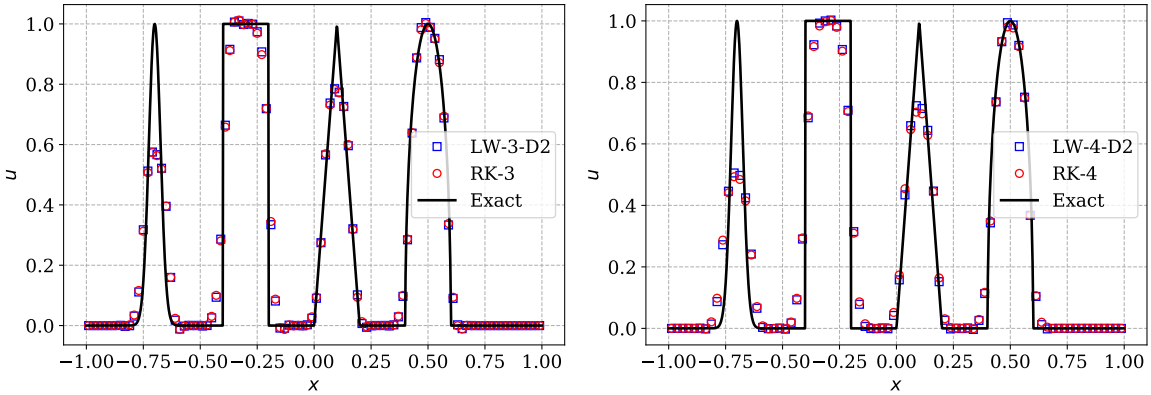
where $G(x,\beta,z)=e^{-\beta(x-z)^2}$, $F\left(x,\alpha,a\right)=\sqrt{1-\alpha^2\left(x-a\right)^2}$ with the constants $a=0.5$, $z=-0.7$, $\delta=0.005$, $\alpha=10$ and $\beta=\frac{\log 2}{36\,\delta^2}$. This initial condition is composed of the succession of a Gaussian, rectangular, triangular and parabolic signals. We compute the numerical solutions at $t=8$ (after 4 periods) and for degrees $N=3,4$ but with 400 dofs in total for each case. The results without any limiter are shown in Figure 4.12; the profiles which are more regular are captured accurately by the numerical schemes, while the hat profile shows some oscillations. These oscillations are larger for $N=3$ than for $N=4$ case.



(a) (b)

Figure 4.12. Constant linear advection of a composite profile without limiter. The solution is shown at time $t=8$ using 400 dofs in each case and polynomial degree (a) $N=3$, (b) $N=4$.

When the TVB limiter is used, the corresponding solutions are shown in Figure 4.13. Now the oscillations in the hat profile are controlled but there is more numerical dissipation as is evident in the reduced amplitude of the smooth profiles. We observe that the results from the LW scheme are very similar to those of the RK scheme.



(a) (b)

Figure 4.13. Constant linear advection of a composite profile with TVB limiter ($M=50$). The solution is shown at time $t=8$ using 400 dofs in each case and polynomial degree (a) $N=3$, (b) $N=4$.

4.7.2. Linear equation with variable coefficient

Now we consider the linear equation with spatially varying coefficient which is given by

$$u_t + f(x,u)_x = 0, \qquad f(x,u) = a(x)\,u$$

This problem is non-linear in the spatial variable, i.e., if I_h is the interpolation operator, then $I_h\,(a\,\boldsymbol{u}_h) \neq I_h(a)\,I_h(u_h)$. This can lead to different behaviour of the numerical schemes compared to the linear case, depending on **AE** and **EA** methods for the numerical flux. To study the effect of non-linearity, we consider different types of speeds with different degree of non-linearity from [131].

Figure 4.14 shows the error convergence for the **AE** and **EA** schemes, and for the speed $a(x)=x$ with initial condition $u_0(x)=\sin(12\,(x-0.1))$. The domain is $[0.1, 2\,\pi]$ and we use Dirichlet boundary conditions at $x=0.1$ and outflow condition at $x=2\,\pi$ so

that the exact solution is given by $u(x,t) = e^{-t}\, u_0(x\, e^{-t})$. As mentioned earlier, upwind flux is used to enforce the boundary condition at inflow boundaries. The LW scheme with either **AE** or **EA** method yields correct convergence rates, while the RK scheme exhibits a small super-convergence. Figure 4.14c shows that the error levels with **AE** and **EA** are nearly same. The non-linearity in this problem is small enough that it does not spoil the error and convergence behaviour of the LW schemes, for both **AE** and **EA** methods.

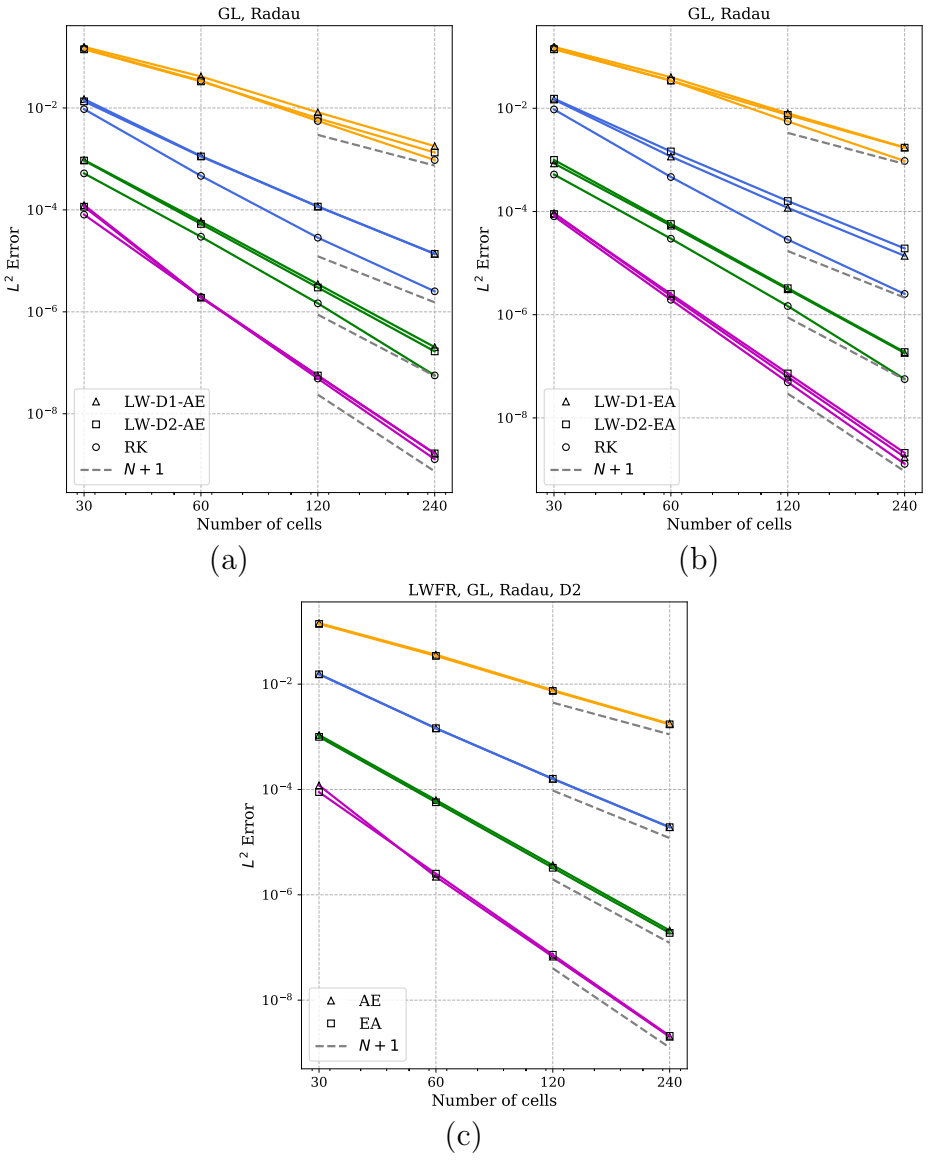


Figure 4.14. Error convergence for variable linear advection with $a(x)=x$: (a) **AE** scheme, (b) **EA** scheme, (c) **AE** vs **EA**.

Figure 4.15 shows the error convergence for the **AE** and **EA** schemes, and for the non-linear speed $a(x) = x^2$, with initial condition $u_0(x) = \cos(\pi x/2)$. The domain is [0.1, 1], and we use Dirichlet boundary conditions at $x = 0.1$ which is an inflow boundary, and outflow condition at $x = 1$ so that the exact solution is given by $u(x,t) = u_0(x/(1+t\,x))/(1+t\,x)^2$. For odd degrees, the LW scheme with **AE** shows larger errors compared to the RK scheme though the convergence rate is optimal. The LW scheme with **EA**

shown in Figure 4.15b, is as accurate as the RK scheme at all degrees. Figure 4.15c compares **AE** and **EA** schemes using GL solution points, Radau correction function and D2 dissipation; we clearly see that **EA** scheme has smaller errors than **AE** scheme at odd degrees, while they are very similar for even degrees. Figure 4.16 shows the error versus time plots for degrees $N = 3, 4$; we see that the LW and RK schemes have very similar error levels and the superior performance of RK schemes observed for constant linear advection is not realized in this non-linear case.

We have observed the same behaviour in all other non-linear test cases given in [131] but the results are not shown here, i.e., the LW schemes with **EA** perform at par with RK schemes for non-linear problems.

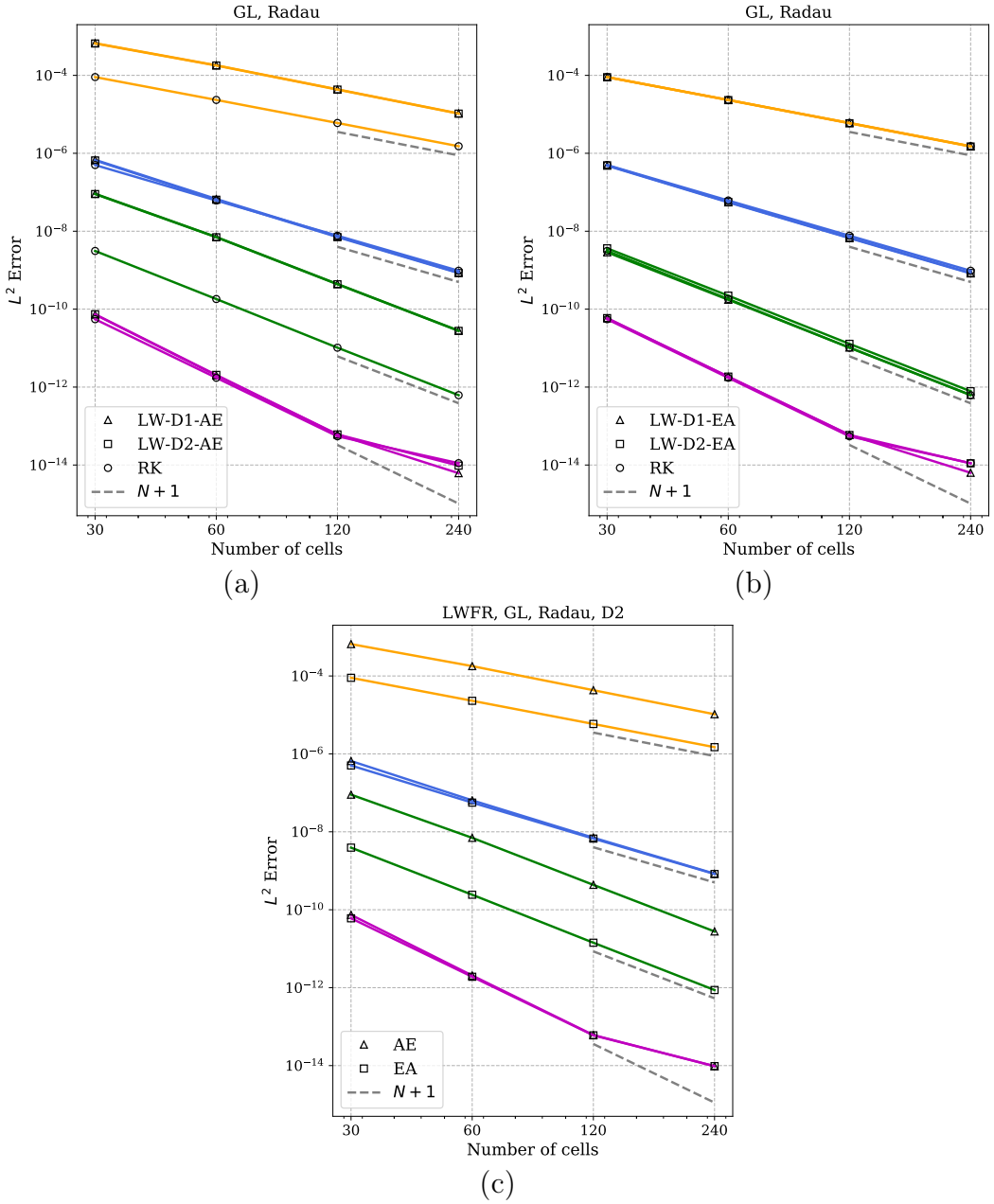
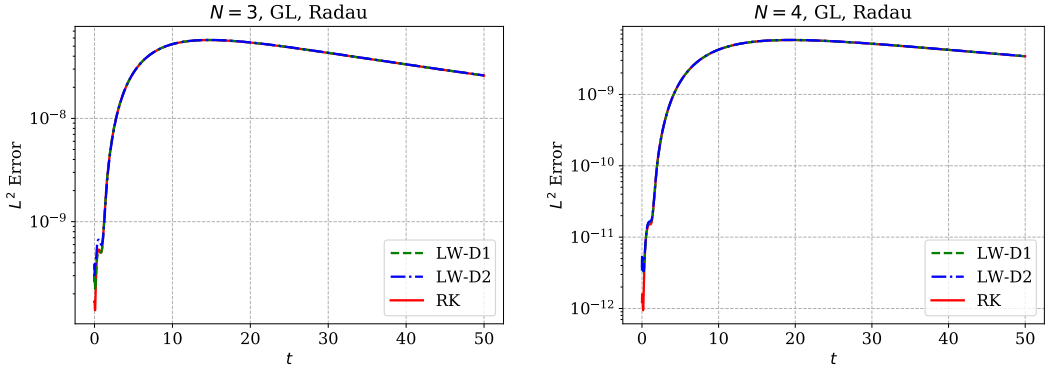


Figure 4.15. Error convergence for variable linear advection with $a(x) = x^2$: (a) **AE** scheme, (b) **EA** scheme, (c) **AE** vs **EA**.

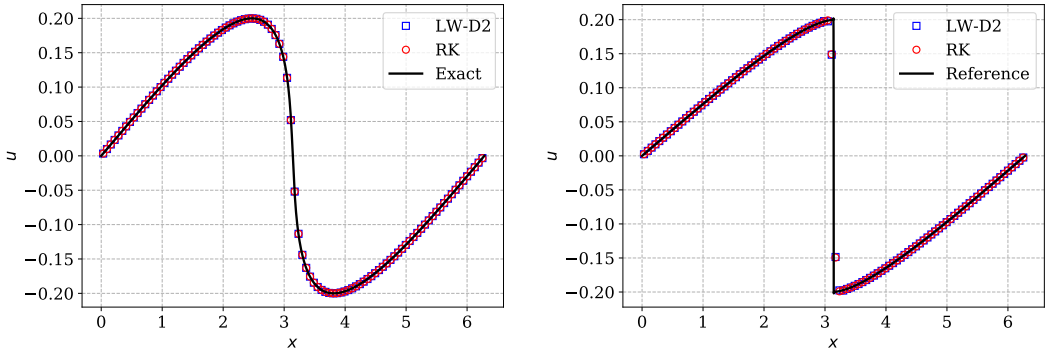


(a) $N=3$ (b) $N=4$

Figure 4.16. Error versus time for linear advection with wave speed $a(x)=x^2$ for different polynomial degrees; GL solution points, Radau correction and polynomial degree (a) $N=3$, (b) $N=4$.

4.7.3. Inviscid Burgers' equation

The one dimensional Burgers' equation is a conservation law of the form $u_t + f(u)_x = 0$ with the quadratic flux $f(u) = u^2/2$. For the smooth initial condition $u(x,0) = 0.2\sin(x)$, we compute the numerical solution at different times $t \in \{2.0, 4.5, 8.0\}$ with periodic boundary condition in the domain $[0, 2\pi]$. The TVB limiter with parameter $M = 1$ is used. A stationary discontinuity is formed at $x = \pi$ and time $t_c = 5$. The solutions are shown in Figure 4.17 for degree $N = 3$ and compared with the results from the RK method. We see that the discontinuity is captured accurately and without any oscillations, and the LW results compare very well with the RK results.



(a) (b)

Figure 4.17. Solution of 1-D Burgers' equation with $N=3$ and 100 cells at different time instants (a) $t=4.5$, (b) $t=8$. TVB limiter ($M=1$) is used. The reference solution is computed using RKFR, degree $N=1$, on a mesh of 3500 cells.

At time $t=2$, the solution is still smooth and we can obtain the exact solution, using which, error norms and convergence rates can be estimated, see Figure 4.18. Figure 4.18a compares the error norms for the **AE** and **EA** methods for the Rusanov numerical flux, and using GL solution points, Radau correction and D2 dissipation; at

odd degrees, the convergence rate of **AE** is less than optimal and close to $O(h^{N+1/2})$, while at even degrees, we obtain the optimal $O(h^{N+1})$ rate. In Figure 4.18b, we see that error norms of LW-**EA** and RK schemes are very close. In Figure 4.18c, we compare the two correction functions using the **EA** scheme and Rusanov flux; it is clear that the errors with Radau correction are significantly smaller than those with g_2 correction.

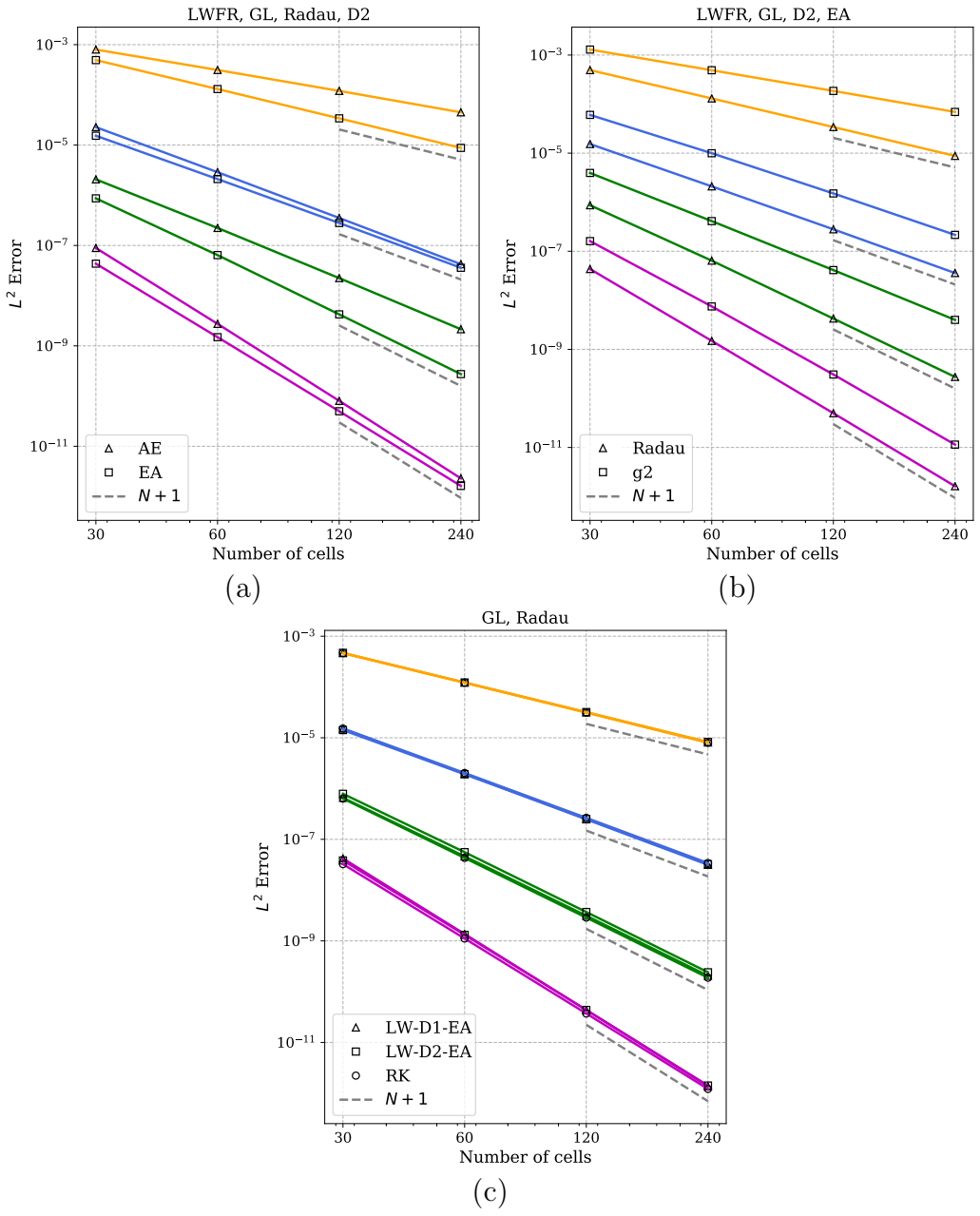


Figure 4.18. Error convergence for 1-D Burgers' equation at time $t=2$. (a) **AE** vs **EA**, (b) Radau vs g2, **EA** scheme, (c) LW-**EA** vs RK.

Next, we study the effect of different numerical fluxes in Figure 4.19 for odd degrees $N=1,3$. With the **AE** scheme, only the global Lax-Friedrich flux is able to achieve the correct convergence rates and has the smallest errors compared to other fluxes which is

a surprising result since it is a very dissipative flux. When the **EA** scheme is used as shown in the right of Figure 4.19, all the numerical fluxes perform very similarly and achieve the optimal convergence rate. An examination of the error distribution in space shows that the **AE** scheme in combination with any numerical flux other than global Lax-Friedrich, produces large errors around the region of sonic points where $f'(u) = 0$; however this happens only for odd degrees and the reason for this behaviour is not known at present. For initial data where the solution does not have a sonic point as in Example 2 of [122], we get optimal convergence rates for all degrees even with the **AE** scheme.

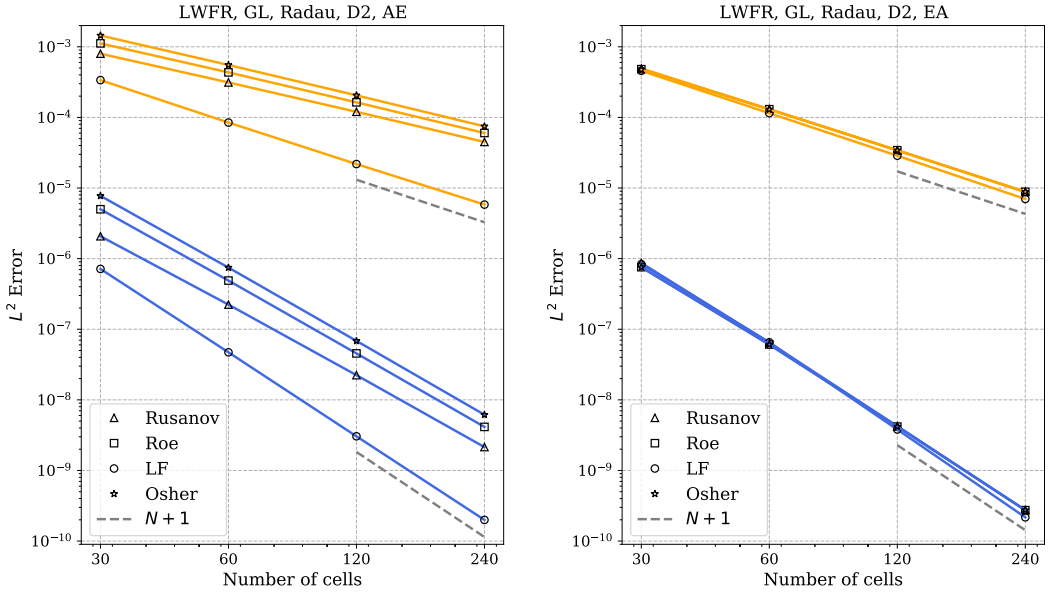


Figure 4.19. Error convergence for 1-D Burgers' equation at time $t = 2$; effect of numerical fluxes for $N = 1, 3$. (a) **AE** scheme, (b) **EA** scheme.

4.7.4. Non-convex problem: Buckley-Leverett equation

We consider the Buckley-Leverett equation $u_t + f(u)_x = 0$, where the flux $f(u) = \dfrac{4u^2}{4u^2 + (1-u)^2}$ is convex and concave in different regions of the solution space. The numerical solution is computed up to the time $t = 0.4$ with the initial condition

$$u(x,0) = \begin{cases} 1, & x \in [-1/2, 0] \\ 0, & \text{elsewhere} \end{cases}$$

At $t = 0.4$, the characteristics that originate from the two discontinuities do not intersect, and thus we only have to deal with the two uncoupled Riemann problems. Solutions to Riemann problems for piecewise strictly convex-concave fluxes can be computed explicitly. In case of the Buckley-Leverett model, we can split the state-space $[0,1]$ into $[0, u_{\text{buck}}]$ and $[u_{\text{buck}}, 1]$ so that f is strictly convex in $(0, u_{\text{buck}})$ and strictly

concave in $(u_{\text{buck}}, 1)$. Thus, the solution to a Riemann problem with states $0, 1$ would compose of a rarefaction and shock, and the exact solution corresponding to the above defined initial condition is composed of two rarefaction-shock combinations. Since the solution measures saturation of displacing fluid, it should lie in the interval $[0, 1]$ and we try to enforce this by applying a positivity preserving scaling limiter [205]. For the LW schemes, we cannot strictly prove the positivity of the resulting scheme[4.1] but numerical results show that this holds in practice, with slightly reduced CFL number compared to the Fourier CFL number. For $N = 4$, the optimal CFL conditions preserve the bounds, while a slightly reduced CFL of 0.079 was needed for $N = 3$. Figure 4.20 shows the results at the final time obtained using degree $N = 3, 4$, respectively. Since the flux is monotone in solution space, an upwind numerical flux is used at cell interfaces, i.e., $F_{e+\frac{1}{2}} = F^{-}_{e+\frac{1}{2}}$. The numerical solutions are able to resolve all the waves well including correct shock location, and they compare well with the results from the RK scheme.

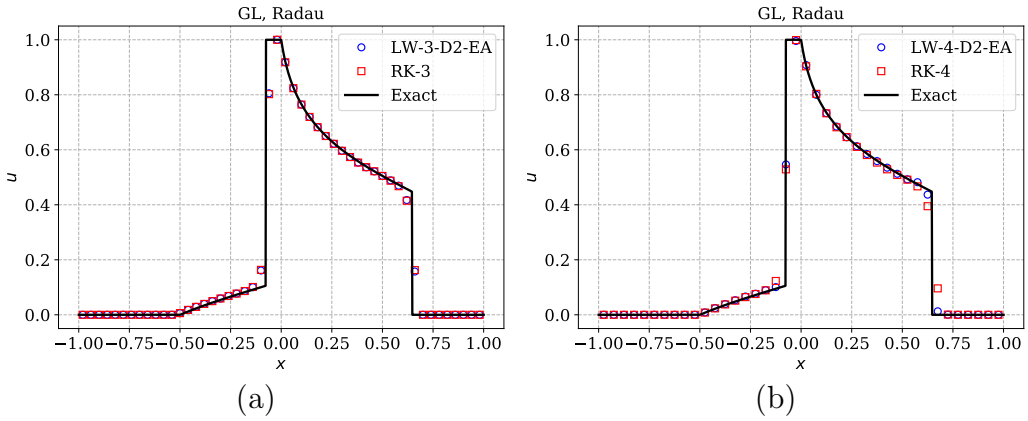


Figure 4.20. Solution of Buckley-Leverett model with TVD limiter using polynomial degrees $N = 3$, 4 with 200 dofs in each case. A positivity preserving scaling limiter [205] has been used to keep the solution in $[0, 1]$.

4.8. Numerical results in 1-D: Euler equations

As an example of a system of non-linear hyperbolic equations, we consider the one-dimensional Euler equations of gas dynamics given by

$$\frac{\partial}{\partial t}\begin{pmatrix} \rho \\ \rho\, u \\ E \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} \rho\, u \\ p + \rho\, u^2 \\ (E + p)\, u \end{pmatrix} = \mathbf{0} \tag{4.16}$$

where ρ, u, p and E denote the density, velocity, pressure and total energy per unit volume of the gas, respectively. For a polytropic gas, an equation of state $E = E(\rho, u,$

4.1. This issue is later overcome by switching to a subcell based blending limiter in Chapter 5, but also without the blending limiter in Chapter 6.

p) which leads to a closed system is given by

$$E = E(\rho, u, p) = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2 \tag{4.17}$$

where $\gamma > 1$ is the adiabatic constant, which will be taken as 1.4, the value for air. The time step size for polynomial degree N is computed as

$$\Delta t = C_{\text{CFL}} \min_e \left(\frac{\Delta x_e}{|\bar{v}_e| + \bar{c}_e} \right) \text{CFL}(N) \tag{4.18}$$

where e is the element index, $\bar{v}_e, \bar{c}_e$ are velocity and sound speed of element mean in element e, CFL(N) is the optimal CFL number obtained by Fourier stability analysis (Table 4.1) and $C_{\text{CFL}} \leq 1$ is a safety factor. The CFL safety factor of 0.95 is used in all results, unless specified otherwise.

The numerical fluxes for Euler equations are explained in the Appendix D. In the following results, wherever it is not mentioned, we use the Rusanov flux.

In the scalar results, we see that the LW scheme with Radau correction function is superior to that with g_2 correction function in terms of error reduction. In light of this, for the 1-D Euler case we compare the performance of LW scheme with RK scheme using the Radau correction function. It is also observed that the **EA** scheme is more accurate than **AE** in the scalar case. So, for the 1-D Euler case we present only those results obtained using **EA** schemes.

Note that wherever it is not specified we use the CFL numbers of Table 4.1 and whenever we compare the numerical solutions of LW scheme with that of RK scheme, both are run with the CFL numbers of LW scheme. Specifically, In the time integration of the RK schemes, for degree $N = 1$ and 2, we use $(N+1)$-stage SSPRK method of order $N+1$. For $N = 3$, we use a five stage, SSPRK method of order four [167] as there is no four stage SSPRK method. In smooth test cases and for $N = 4$, we use a six stage Runge-Kutta method of order five [183]. In those test cases where the solution is not sufficiently smooth, the SSP property of the RK time integration is useful in obtaining non-oscillatory solutions. As we do not have SSPRK method of order five with positive coefficients [153], we use the SSPRK method [167] of order four when $N = 4$.

4.8.1. Smooth solution

To verify the accuracy of the proposed scheme, we solve the Euler equations (4.16, 4.17) with a smooth initial condition

$$\rho(x, 0) = 1 + 0.5 \sin(2\pi x), \qquad u(x, 0) = 1, \qquad p(x, 0) = 1$$

in the domain $[0, 1]$ with periodic boundary conditions for all the variables. The corresponding exact solution is a density wave, i.e., it consists of a translation of the initial

density at constant speed of one, and is given by

$$\rho(x,t) = 1 + 0.5\sin(2\pi(x-t)), \qquad u(x,t) = 1, \qquad p(x,t) = 1$$

We compute the solution up to the time $t = 1$ and estimate the error norms. The linear nature of this test case makes **EA** and **AE** schemes equivalent, and we show only the **EA** results. We plot the error in the density obtained using the LW and RK schemes and the corresponding results are given in Figure (4.21). In each case we observe the expected order of accuracy and the error reduction is close to that of RK scheme.

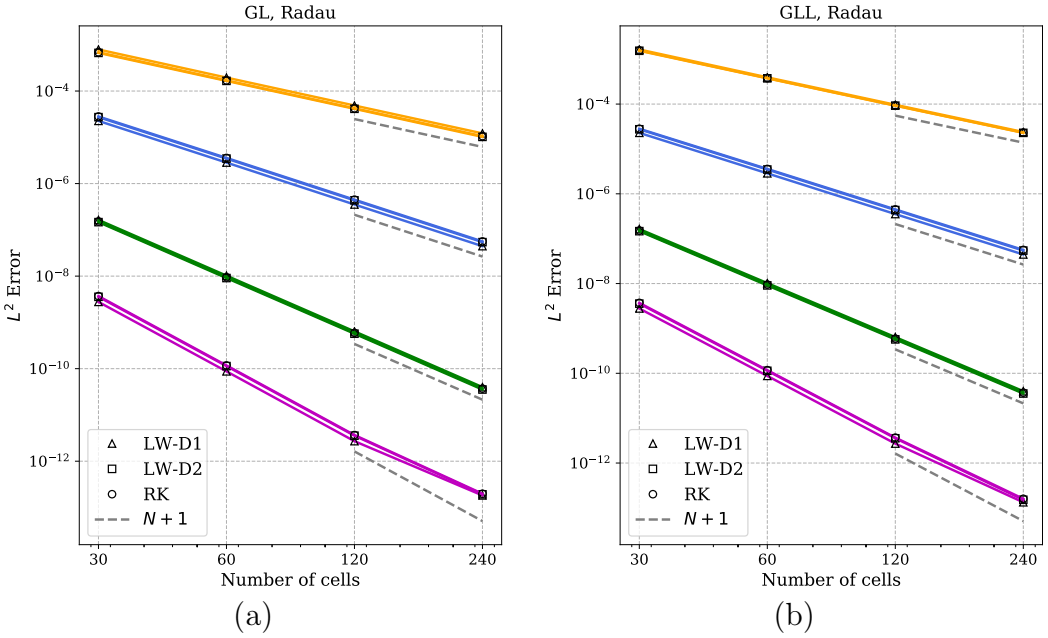


Figure 4.21. Density error convergence for 1-D Euler's equation at time $t = 1$. The different colors correspond to degrees $N = 1, 2, 3, 4$ from top to bottom. (a) GL points, (b) GLL points.

Remark 4.5. Based on the scalar test cases and the above smooth test case for Euler equations, in all the remaining 1-D Euler test cases, we present only the results obtained using D2 scheme since it is advantageous in terms of having higher CFL number and performs as well as or better than the D1 scheme.

4.8.2. Sod's shock tube problem

The Sod's shock tube problem [164] is a Riemann problem which models a shock tube where gas at two different conditions initially is allowed to interact, with the formation of shock, contact and rarefaction waves. The Euler equations (4.16) are solved with the initial condition

$$(\rho, u, p) = \begin{cases} (1, 0, 1), & \text{if } x < 0.5 \\ (0.125, 0, 0.1), & \text{if } x > 0.5 \end{cases} \tag{4.19}$$

for which the exact solution is composed of a left rarefaction, a contact discontinuity and a right shock wave. The approximate solution is computed in the domain [0, 1] with the outflow boundary conditions on both the ends $x = 0$ and $x = 1$. We run the numerical scheme up to time $t = 0.2$ with 100 cells using the TVB limiter with parameter $M = 10$. The density profile obtained using the LW and RK schemes for $N = 3$ and $N = 4$ are shown in Figure 4.22 together with the exact solution. From the plots it is visible that the results obtained using the LW scheme agree very well with that of RK scheme.

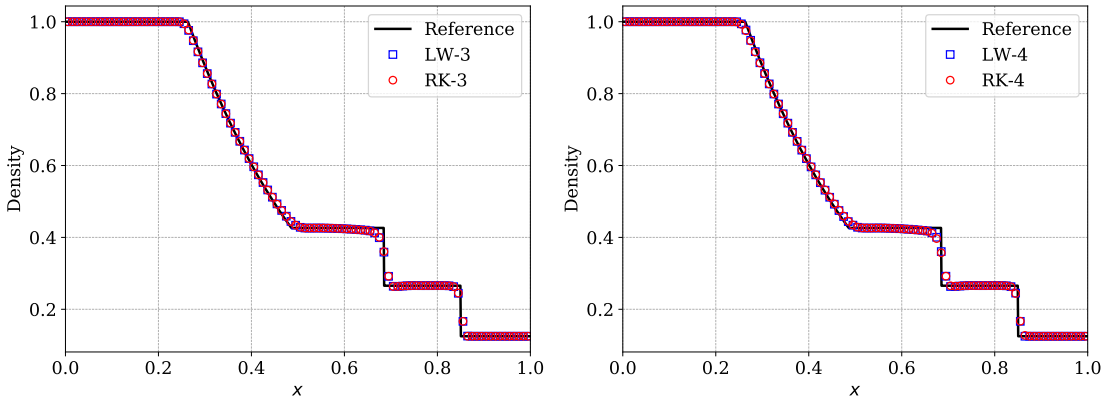


Figure 4.22. Numerical solutions of 1-D Euler equations (Sod's test case) obtained by LW and RK schemes for polynomial degree (a) $N = 3$ and (b) $N = 4$ using Radau correction function and GL solution points. The solutions are shown at time $t = 0.2$ on a mesh of 100 cells with Rusanov flux and D2 dissipation. The TVB limiter is used with the parameter $M = 10$.

4.8.3. Lax problem

We consider the Lax problem given in [111, 92] which solves a Riemann problem for the system of equations (4.16) with initial condition

$$(\rho, u, p) = \begin{cases} (0.445, 0.698, 3.528), & \text{if } x < 0 \\ (0.5, 0, 0.571), & \text{if } x > 0 \end{cases} \tag{4.20}$$

where, unlike the Sod's shock tube problem, the initial velocity is not zero. The exact solution of this Riemann problem is known and it consists of a rarefaction, a right moving contact discontinuity and shock. For a detailed description of this problem, see [92]. This is a demanding test case in the sense that, high order schemes are prone to produce oscillations near the contact discontinuity. The numerical solution is computed up to time $t = 1.3$ in the domain $[-5, 5]$ using 100 cells and using TVB limiter with parameter $M = 1$. The approximate solutions are computed for polynomial degrees $N = 3$ and $N = 4$, and the corresponding density profiles are shown in Figure 4.23 along with the exact solution. We observe that the LW scheme captures the wave structures accurately without oscillations and the numerical solutions are very close to that of

RK scheme.

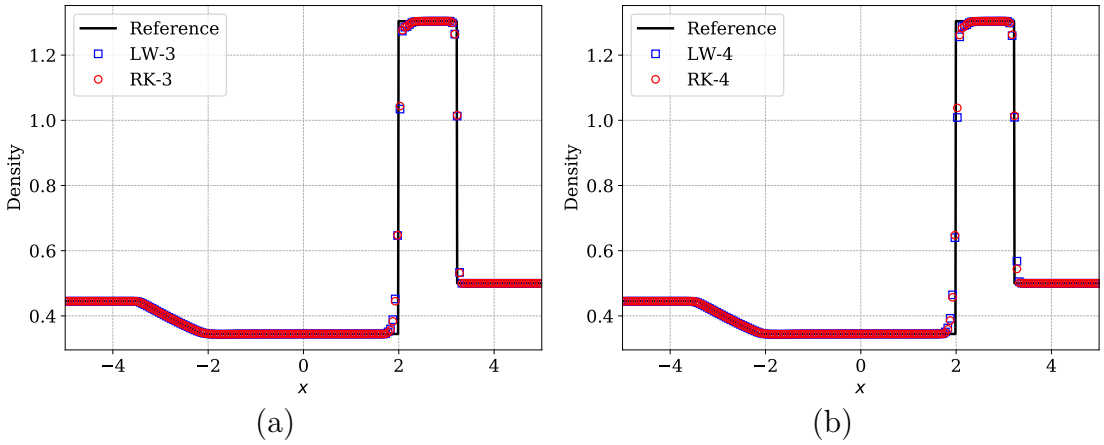


Figure 4.23. Numerical solutions of 1-D Euler equations (Lax's test case) obtained by LW and RK schemes for polynomial degree (a) $N=3$ and (b) $N=4$ with Radau correction function and GL solution points. The solutions are computed on a mesh of 200 cells with dissipation model D2 and Rusanov numerical flux and are shown at time $t=1.3$. The TVB limiter is used with parameter $M=1$.

4.8.4. Shu-Osher problem

This problem was developed in [163] to test the numerical scheme's capability to accurately capture a shock wave and its interaction with a smooth density field, which then propagates downstream of the shock. Here, we compute the numerical solution of (4.16) with initial condition

$$(\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.333333), & \text{if } x < -4 \\ (1 + 0.2\sin(5x), 0, 1), & \text{if } x \geq -4 \end{cases} \tag{4.21}$$

prescribed in the domain $[-5, 5]$ at time $t = 1.8$. The smooth density profile passes through the shock and appears on the other side, and its accurate computation is challenging due to numerical dissipation introduced by limiters at the shock. We discretize the spatial domain with 400 cells and to control the spurious oscillations we use the TVB limiter with parameter $M = 300$ [137]. The density component of the approximate solutions computed using LW and RK schemes for $N = 3$ and $N = 4$ are plotted against a reference solution obtained using a very fine mesh, and are given in Figure 4.24. We observe that the post shock wave patterns are accurately captured by the proposed LW scheme. Furthermore, the enlarged plots of the oscillatory portion indicate that

the numerical solutions corresponding to LW scheme are comparable with that of RK schemes.

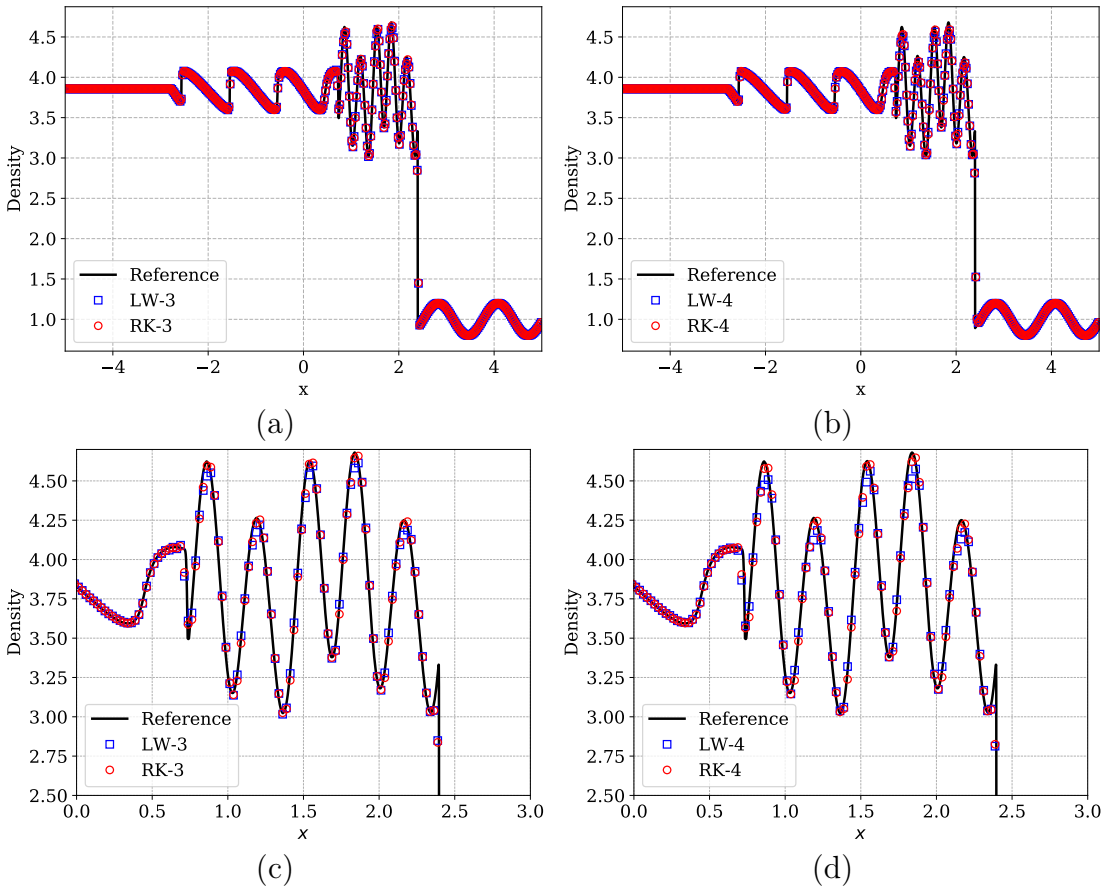


Figure 4.24. Numerical solutions of 1-D Euler equations (Shu-Osher problem) obtained by LW and RK schemes for (a, c) $N=3$ and (b, d) $N=4$ with Radau correction function and GL solution points. The enlarged plot of the oscillatory portion is given in the bottom row. The solutions are shown at time $t=1.8$ on a mesh of 400 cells with dissipation model D2 and Rusanov numerical flux. The TVB limiter is used with parameter $M=300$.

4.8.5. Blast wave

In this test case the Euler equations (4.16) are solved with the initial condition

$$(\rho,u,p)=\begin{cases}(1,0,1000), & \text{if } x<0.1\\ (1,0,0.01), & \text{if } 0.1<x<0.9\\ (1,0,100), & \text{if } x>0.9\end{cases}$$

in the domain $[0,1]$. It is originally introduced in [197] to check the capability of the numerical scheme to accurately capture the shock-shock interaction scenario. The boundaries are set as solid walls by imposing the reflecting boundary conditions at

$x=0$ and $x=1$. The solution consists of reflection of shocks and expansion waves off the boundary wall and several wave interactions inside the domain. With a grid of 400 cells, we run the simulation till the time $t=0.038$ where a high density peak profile is produced. The TVB limiter as in [137] with parameter $M=300$ is used along with a positivity preserving scaling limiter [205]. The scaling limiter is used without the flux limiting introduced in Chapters 5, 6 and is thus not provably positive. We compare the performance of the LW scheme with the RK scheme and analyze how well they predict the density profile and its peak amplitude. For $N=3$ and $N=4$ cases, the results are given in Figure 4.25 where the approximated density profiles are compared with a reference solution computed using a very fine mesh. From the plots it is evident that the computed density profile obtained using LW scheme are close to that of RK scheme.

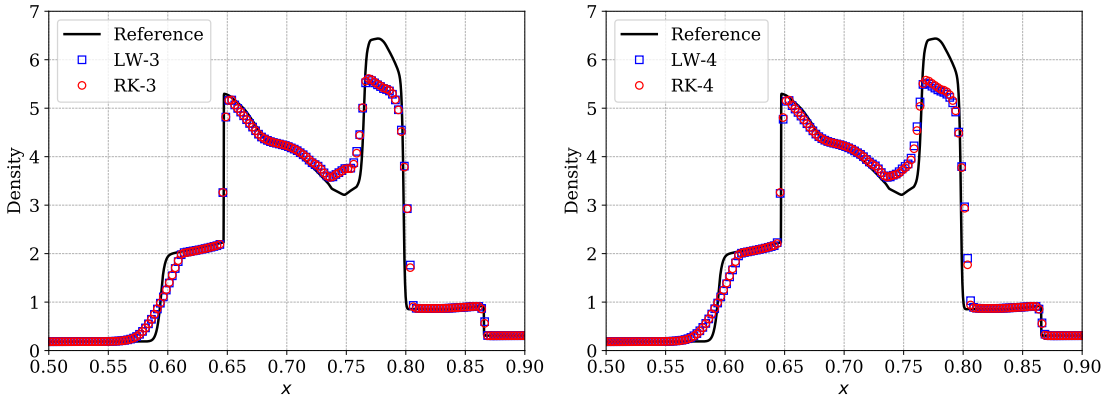


Figure 4.25. Numerical solutions of 1-D Euler equations (Blast wave) obtained by LWFR and RKFR schemes for (a) $N=3$ and (b) $N=4$ using Radau correction function and GL solution points. The solutions are plotted at time $t=0.038$ on a mesh of 400 cells with dissipation model D2 and Rusanov numerical flux. The corresponding CFL numbers of LWFR scheme are chosen from the Table 4.1 and kept same for the RKFR schemes. TVB limiter is used with the parameter $M=300$ and the **EA** scheme is used for numerical flux.

4.8.6. Numerical fluxes: LF, Roe, HLL and HLLC

The previous Euler results used Rusanov flux. Here, we show performance of other numerical fluxes like HLL, HLLC, Roe and global Lax-Friedrichs which were described for LWFR in Appendix D. Fluxes like HLL, HLLC and Roe may be desirable in some problems due to their upwind character, unlike Lax-Friedrich/Rusanov type fluxes. Moreover, HLLC and Roe fluxes also model the linear contact and shear waves which can lead to better approximations of these waves. We have tested the numerical fluxes in all the test cases, however to save space we present only the blast test case for $N=3$. The results are given in Figure 4.26 which compare these fluxes with Rusanov flux. The high density peak region is better approximated by the LW schemes using HLL, HLLC and Roe fluxes, as compared to the Rusanov flux. The global LF flux is found to be less accurate in this respect when compared to the Rusanov flux, which is expected

due to the larger amount of numerical dissipation in the global Lax-Friedrich flux.

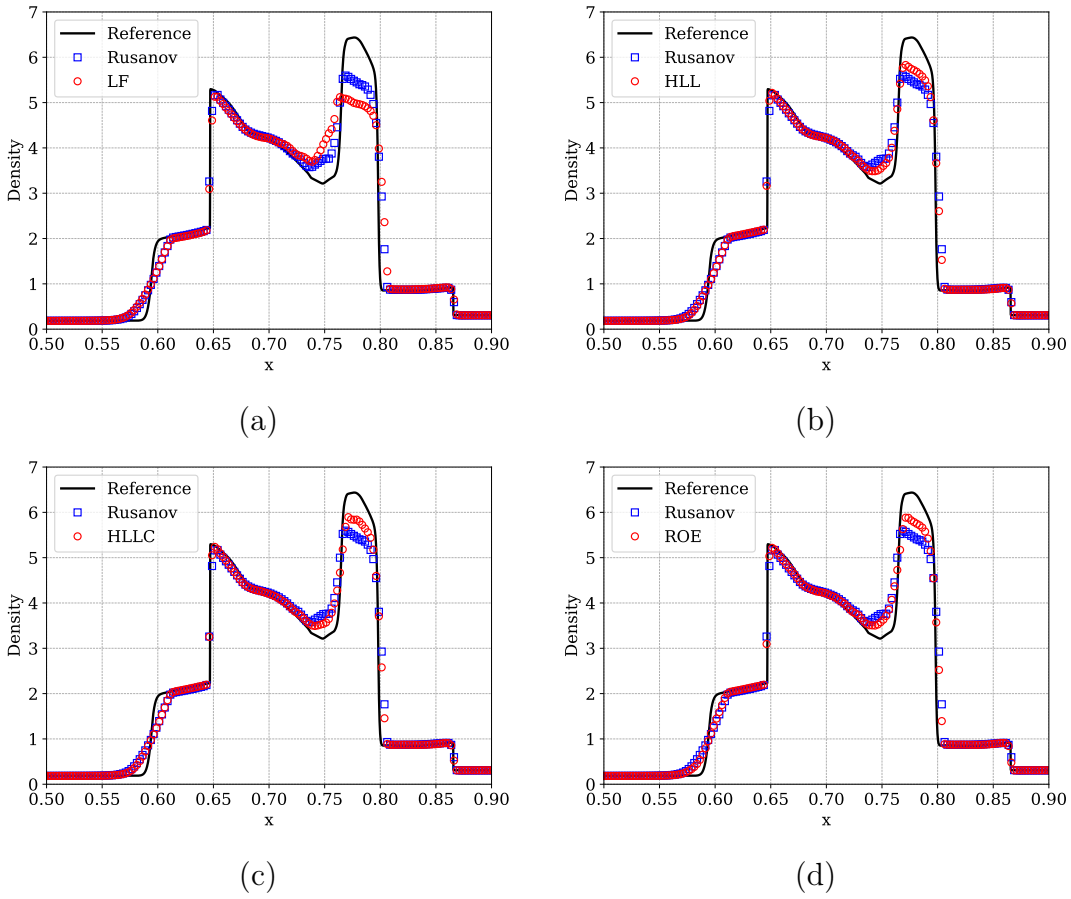


Figure 4.26. Numerical solutions of 1-D Euler equations (Blast wave) obtained by LW schemes with different numerical fluxes (a) LF, (b) HLL, (c) HLLC and (d) ROE compared with Rusanov flux, for $N=3$ using Radau correction function and GL solution points. All other parameters remain the same as in Figure 4.25.

Since HLLC and Roe schemes contain more information about the wave structure, they are better at resolving contact discontinuities which are linearly degenerate waves that can be severely affected by numerical dissipation. We illustrate this through two Riemann problems containing stationary contact waves. The first one consists of an initial density jump that leads to a stationary contact wave, with initial condition given by,

$$(\rho, u, p) = \begin{cases} (1,0,1), & \text{if } x < 0.5 \\ (2,0,1), & \text{if } x > 0.5 \end{cases}$$

In Figure 4.27a, we show the comparison of numerical fluxes for this stationary solution test case, zoomed near the discontinuity, at $t=1.0$ using LW schemes with D2 dissipation model for degree $N=4$ on a grid of 100 cells together with TVB ($M=1$) limiter. As expected, we see that Roe and HLLC fluxes are able to resolve the contact discontinuity exactly, while the other fluxes smear the jump over two cells.

The second Riemann problem is a tough test case with respect to maintaining positivity of pressure and is taken from [181]. The initial condition is given by

$$(\rho, u, p) = \begin{cases} (1, -19.59745, 1000), & \text{if } x < 0.8 \\ (1, -19.59745, 0.01), & \text{if } x > 0.8 \end{cases}$$

The solution develops a stationary contact at the location of the initial discontinuity $x = 0.8$ and a right moving shock wave. In Figure 4.27b, we show the comparison of numerical fluxes, zoomed near the contact discontinuity, at $t = 0.012$ obtained using LW scheme with D2 dissipation model for polynomial degree $N = 4$ on a grid of 100 cells and TVB ($M = 1$) limiter. As in the previous case, the HLLC flux captures the contact discontinuity more accurately than the other fluxes.

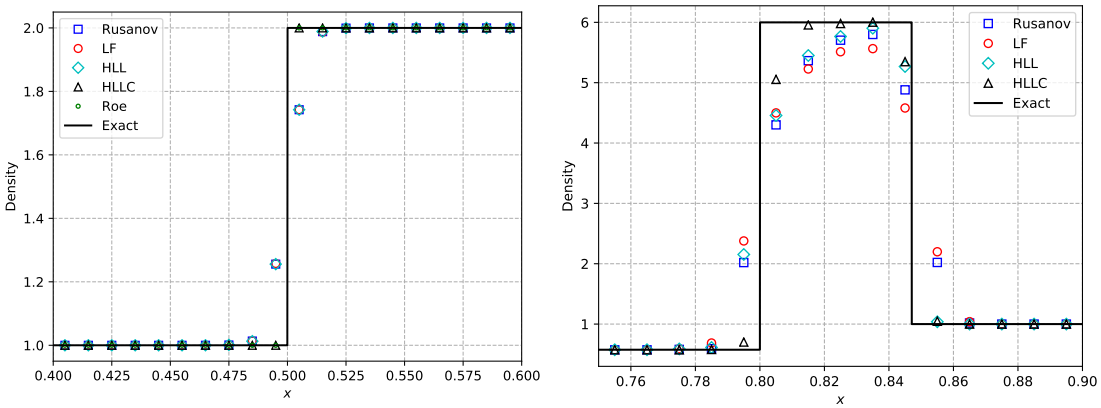


Figure 4.27. Numerical solutions of 1-D Euler's equations for (a) stationary contact test, (b) Toro's Test 5 obtained by LW schemes with different numerical fluxes for polynomial degree $N = 4$ using Radau correction function and GL solution points, TVB ($M = 1$) limiter on a grid of 100 cells.

4.8.7. Comparison of correction functions

We compare the robustness and accuracy of the two correction functions, Radau and g_2, in the LW scheme when applied to the Euler equations (4.16) with GL solution points. The numerical experiments are conducted for the Shu-Osher test case and the corresponding results are obtained with the HLLC numerical flux for polynomial degrees $N = 1, 2, 3, 4$, see Figure 4.28. For this test case, it is observed that the LW scheme with g_2 correction function fails to work for $N = 1$ with the optimal CFL of Table 4.1 due to loss of positivity of pressure. So, we use a smaller CFL number of 0.44 to compute the $N = 1$ case in Figure 4.28. For $N \geq 2$, the solutions computed using the g_2 correction function are found to be close to that of Radau correction function. However, it fails to perform consistently, as we see in the $N = 1$ case. With this observation and also the behaviour for other problems, we see that it is desirable to use the Radau correction function in the LW scheme.

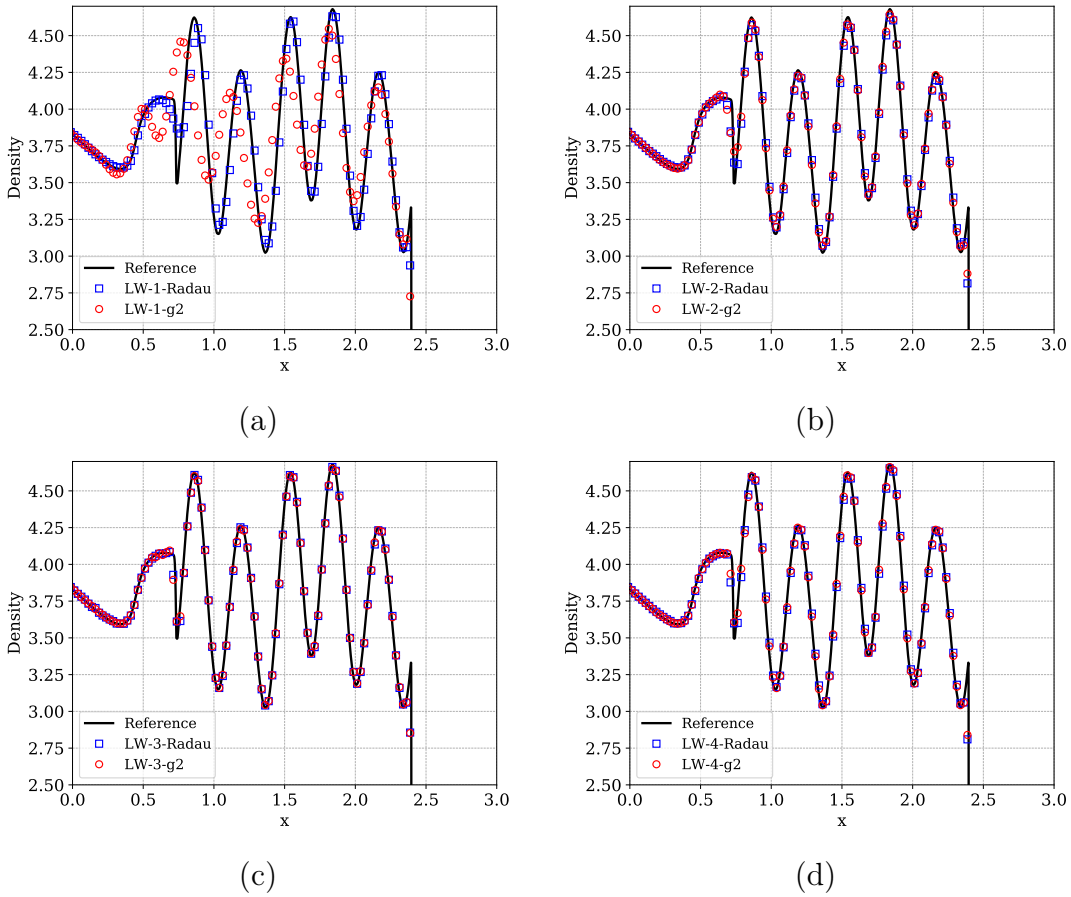


Figure 4.28. Numerical solutions of 1-D Euler equations (Shu-Osher problem) for (a) $N=1$, (b) $N=2$, (c) $N=3$, (d) $N=4$. Comparison of LW scheme with GL solution points for two correction functions, Radau and g2, with their own CFL numbers chosen from Table 4.1, except for g_2 correction function with $N=1$, where we choose CFL=0.44. The enlarged oscillatory portion of the solutions is shown. The solutions are computed at time $t=1.8$ on a mesh of 400 cells with dissipation model D2 and HLLC numerical flux. The TVB limiter is used with parameter $M=300$.

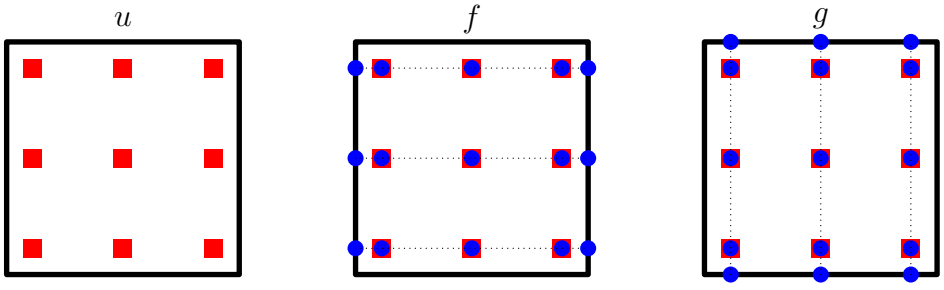


Figure 4.29. Location of solution and flux points on a 2-D FR element for the degree $N=2$ case. Numerical flux is required at the blue points on the faces.

4.9. TWO DIMENSIONAL SCHEME

The extension of the 1-D scheme to two dimensions is performed by applying the 1-D ideas along each coordinate direction. Consider a 2-D conservation law of the form

$$\boldsymbol{u}_t + \boldsymbol{f}(\boldsymbol{u})_x + \boldsymbol{g}(\boldsymbol{u})_y = \boldsymbol{0} \tag{4.22}$$

where $(\boldsymbol{f}, \boldsymbol{g})$ are Cartesian components of the flux vector. Using Taylor expansion in time, we can write the solution at $t = t_{n+1}$ as

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \Delta t \left[\frac{\partial \boldsymbol{F}}{\partial x}(\boldsymbol{u}^n) + \frac{\partial \boldsymbol{G}}{\partial y}(\boldsymbol{u}^n) \right] + O(\Delta t^{N+2})$$

where $\boldsymbol{F}, \boldsymbol{G}$ are time average fluxes given by

$$\boldsymbol{F}(\boldsymbol{u}) = \sum_{m=0}^{N} \frac{\Delta t^m}{(m+1)!} \partial_t^m \boldsymbol{f}(\boldsymbol{u}), \qquad \boldsymbol{G}(\boldsymbol{u}) = \sum_{m=0}^{N} \frac{\Delta t^m}{(m+1)!} \partial_t^m \boldsymbol{g}(\boldsymbol{u})$$

We will consider a Cartesian mesh and map each element Ω_e to the reference element $\hat{K} = [0,1] \times [0,1]$. Inside the reference element, the solution points are chosen to be tensor product of 1-D solution points, which may be either GL or GLL points. Figure 4.29 shows an example of 2-D solution points based on tensor product of 1-D GL points. The solution inside an element Ω_e is approximated by a tensor product polynomial of degree N,

$$(x, y) \in \Omega_e: \qquad \boldsymbol{u}_h = \sum_{p=0}^{N} \sum_{q=0}^{N} \boldsymbol{u}_{e,pq} \ell_p(\xi) \, \ell_q(\eta)$$

where (ξ, η) are coordinates in the reference element, and $\ell_p(\xi), \ell_q(\eta)$ are the 1-D Lagrange polynomials based on the solution points. The discontinuous fluxes are approximated by interpolating at the solution points,

$$\boldsymbol{F}_h^\delta(\xi, \eta) = \sum_{p=0}^{N} \sum_{q=0}^{N} \boldsymbol{F}_{e,pq} \ell_p(\xi) \, \ell_q(\eta), \qquad \boldsymbol{G}_h^\delta(\xi, \eta) = \sum_{p=0}^{N} \sum_{q=0}^{N} \boldsymbol{G}_{e,pq} \ell_p(\xi) \, \ell_q(\eta)$$

where $\boldsymbol{F}_{e,pq}, \boldsymbol{G}_{e,pq}$ are time average fluxes obtained from the Lax-Wendroff procedure applied at each solution point. The continuous flux along the x and y axes are constructed using the one dimensional algorithm along the $\eta = \xi_q = \text{constant}$ and $\xi = \xi_p = \text{constant}$ lines, respectively, see Figure 4.29,

$$\boldsymbol{F}_h(\xi, \xi_q) = [\boldsymbol{F}_{e-\frac{1}{2},q} - \boldsymbol{F}_h^\delta(0, \xi_q)] \, g_L(\xi) + \boldsymbol{F}_h^\delta(\xi, \xi_q) + [\boldsymbol{F}_{e+\frac{1}{2},q} - \boldsymbol{F}_h^\delta(1, \xi_q)] \, g_R(\xi), \ 0 \le q \le N$$

$$\boldsymbol{G}_h(\xi_p, \eta) = [\boldsymbol{G}_{e-\frac{1}{2},p} - \boldsymbol{G}_h^\delta(\xi_p, 0)] \, g_L(\eta) + \boldsymbol{G}_h^\delta(\xi_p, \eta) + [\boldsymbol{G}_{e+\frac{1}{2},p} - \boldsymbol{G}_h^\delta(\xi_p, 1)] \, g_R(\eta), \ 0 \le p \le N$$

Note that the above equations are obtained by applying the FR idea along the horizontal and vertical lines in Figure 4.29. The quantities $\boldsymbol{F}_{e-\frac{1}{2}}, \boldsymbol{F}_{e+\frac{1}{2}}$ are x-directional numerical fluxes on the left and right faces, while $\boldsymbol{G}_{e-\frac{1}{2}}, \boldsymbol{G}_{e+\frac{1}{2}}$ are the y-directional numerical fluxes across the bottom and top faces, respectively. The update equation is given by a collocation procedure at each solution point,

$$\boldsymbol{u}_{e,pq}^{n+1} = \boldsymbol{u}_{e,pq}^n - \Delta t \left[\frac{1}{\Delta x_e} \frac{\partial \boldsymbol{F}_h}{\partial \xi}(\xi_p, \xi_q) + \frac{1}{\Delta y_e} \frac{\partial \boldsymbol{G}_h}{\partial \eta}(\xi_p, \xi_q) \right], \qquad 0 \le p, q \le N \tag{4.23}$$

where the flux derivatives can be computed from

$$\partial_\xi \boldsymbol{F}_h(:,\xi_q) = \left[\boldsymbol{F}_{e-\frac{1}{2},q} - \boldsymbol{F}_h^\delta(0,\xi_q)\right]\mathsf{b}_L + \partial_\xi \boldsymbol{F}_h^\delta(:,\xi_q) + \left[\boldsymbol{F}_{e+\frac{1}{2},q} - \boldsymbol{F}_h^\delta(1,\xi_q)\right]\mathsf{b}_R, \quad 0 \le q \le N$$

$$\partial_\eta \boldsymbol{G}_h(\xi_p,:) = \left[\boldsymbol{G}_{e-\frac{1}{2},p} - \boldsymbol{G}_h^\delta(\xi_p,0)\right]\mathsf{b}_L + \partial_\eta \boldsymbol{G}_h^\delta(\xi_p,:) + \left[\boldsymbol{G}_{e+\frac{1}{2},p} - \boldsymbol{G}_h^\delta(\xi_p,1)\right]\mathsf{b}_R, \quad 0 \le p \le N$$

We can cast the update equation in matrix form. For this, define the flux matrices

$$\mathsf{F}_e(p,q) = \boldsymbol{F}_{e,pq}, \qquad \mathsf{G}_e(p,q) = \boldsymbol{G}_{e,pq}, \qquad 0 \le p,q \le N$$

Then we can compute the derivatives of the discontinuous flux at all the solution points by a matrix-matrix product

$$\partial_\xi \boldsymbol{F}_h^\delta(:,:) = \mathsf{D}\mathsf{F}_e, \qquad \partial_\eta \boldsymbol{G}_h^\delta(:,:) = \mathsf{G}_e\mathsf{D}^\top$$

where D is the 1-D differentiation matrix. Note that D acts on each component for systems of equations, see Appendix E for its efficient implementation. The update equation can be written in matrix form,

$$\begin{aligned} \mathsf{u}_e^{n+1} \;=\; \mathsf{u}_e^n &- \left[\frac{\Delta t}{\Delta x_e}\mathsf{D}_1\mathsf{F}_e + \frac{\Delta t}{\Delta y_e}\mathsf{G}_e\mathsf{D}_1^\top\right] - \frac{\Delta t}{\Delta x_e}\left[\mathsf{b}_L\mathsf{F}_{e-\frac{1}{2}}^\top + \mathsf{b}_R\mathsf{F}_{e+\frac{1}{2}}^\top\right] \\ &- \frac{\Delta t}{\Delta y_e}\left[\mathsf{G}_{e-\frac{1}{2}}\mathsf{b}_L^\top + \mathsf{G}_{e+\frac{1}{2}}\mathsf{b}_R^\top\right] \end{aligned} \tag{4.24}$$

where the quantities $\mathsf{D}_1, \mathsf{b}_L, \mathsf{b}_R$ have been defined before in the description of the 1-D scheme in Section 4.2.2.

The time average fluxes are computed by the approximate Lax-Wendroff procedure. To describe this, let us define the flux matrices

$$\mathsf{f}_e(p,q) = \boldsymbol{f}(\boldsymbol{u}_{e,pq}), \qquad \mathsf{g}_e(p,q) = \boldsymbol{g}(\boldsymbol{u}_{e,pq}), \qquad 0 \le p,q \le N$$

The time derivatives of the solution at all solution points are obtained from the PDE by the following matrix equation,

$$\mathsf{u}_e^{(m)} = -\frac{\Delta t}{\Delta x_e}\mathsf{D}\mathsf{f}_e^{(m-1)} - \frac{\Delta t}{\Delta y_e}\mathsf{g}_e^{(m-1)}\mathsf{D}^\top, \qquad m = 1,2,\ldots,N \tag{4.25}$$

and the time average solution and fluxes are given by

$$\mathsf{U}_e = \sum_{m=0}^{N} \frac{\mathsf{u}_e^{(m)}}{(m+1)!}, \qquad \mathsf{F}_e = \sum_{m=0}^{N} \frac{\mathsf{f}_e^{(m)}}{(m+1)!}, \qquad \mathsf{G}_e = \sum_{m=0}^{N} \frac{\mathsf{g}_e^{(m)}}{(m+1)!}$$

The time derivatives of the fluxes $\mathsf{f}_e^{(m)}$, $\mathsf{g}_e^{(m)}$ are approximated using finite differences in time as in the 1-D case given in Section 4.2.4; those formulae are applied to both components of the flux. The stable time step is determined by considering the linear advection equation in 2-D and applying Fourier stability analysis to the LW scheme, as described in Section 4.9.1.

4.9.1. Fourier analysis in 2-D

Consider the linear advection equation

$$u_t + a_1\, u_x + a_2\, u_y = 0 \tag{4.26}$$

where (a_1, a_2) is a constant velocity. We first write the LW scheme for (4.26) in matrix form which helps to derive the Fourier amplification term. Let us define the matrix of solution and flux values by

$$\mathsf{u}_e(p,q) = u_{e,pq}, \qquad \mathsf{f}_e(p,q) = a_1\, u_{e,pq}, \qquad \mathsf{g}_e(p,q) = a_2\, u_{e,pq}$$

In the Lax-Wendroff procedure, the time derivative of the solution at all the solution points is given by (4.25)

$$\mathsf{u}_e^{(m)} = -\sigma_1\, \mathsf{D}\mathsf{u}_e^{(m-1)} - \sigma_2\, \mathsf{u}_e^{(m-1)}\mathsf{D}^\top, \quad m = 1, 2, \ldots, N$$

where σ_1, σ_2 are the CFL numbers along x, y directions, respectively, which are given by

$$\sigma_1 = \frac{a_1\, \Delta t}{\Delta x_e}, \qquad \sigma_2 = \frac{a_2\, \Delta t}{\Delta y_e}$$

Then the time average solution and fluxes are given by

$$\mathsf{U}_e = \sum_{m=0}^{N} \frac{\mathsf{u}_e^{(m)}}{(m+1)!}, \qquad \mathsf{F}_e = a_1\, \mathsf{U}_e, \qquad \mathsf{G}_e = a_2\, \mathsf{U}_e$$

To perform the Fourier analysis, we must write the scheme (4.23) in matrix-vector form. To do this, let us renumber the two dimensional indices (p, q) which denote solution points, into the one dimensional numbering by the following transformation

$$k = p + (N+1)\, q, \qquad 0 \le p, q \le N$$

Then k takes the values 0 to $M = (N+1)^2 - 1$. If $\phi_e \in \mathbb{R}^{(N+1)\times(N+1)}$ is some quantity defined at the solution points, we let $[\![\phi_e]\!] \in \mathbb{R}^{M+1}$ denote the same renumbered as above. After renumbering, the matrix-matrix products become

$$[\![\mathsf{A}\phi_e]\!] = R_1(\mathsf{A})[\![\phi_e]\!], \qquad [\![\phi_e\mathsf{A}]\!] = R_2(\mathsf{A})[\![\phi_e]\!]$$

where

$$R_1(\mathsf{A}) = \mathsf{I} \otimes \mathsf{A}, \qquad R_2(\mathsf{A}) = \mathsf{A}^\top \otimes \mathsf{I}$$

with $\otimes$ denoting the kronecker product. Then the renumbering of the solution time derivatives and time average solution and fluxes are given by

$$[\![\mathsf{u}_e^{(m)}]\!] = (-\sigma_1 R_1(\mathsf{D}) - \sigma_2 R_2(\mathsf{D}^\top))[\![\mathsf{u}_e^{(m-1)}]\!] =: \mathsf{H}_1 [\![\mathsf{u}_e^{(m-1)}]\!], \quad m = 1, 2, \ldots, N$$

$$[\![\mathsf{U}_e]\!] = \left(\sum_{m=0}^{N} \frac{\mathsf{H}_1^m}{(m+1)!} \right) [\![\mathsf{u}_e]\!] =: \mathsf{T} [\![\mathsf{u}_e]\!], \qquad [\![\mathsf{F}_e]\!] = a_1 [\![\mathsf{U}_e]\!], \qquad [\![\mathsf{G}_e]\!] = a_2 [\![\mathsf{U}_e]\!]$$

Finally, the renumbered terms in the update equation (4.24) are given by

$$[\![\mathsf{D}_1 \mathsf{F}_e]\!] = R_1(\mathsf{D}_1)[\![\mathsf{F}_e]\!] = a_1 R_1(\mathsf{D}_1)\mathsf{T}[\![\mathsf{u}_e]\!], \qquad [\![\mathsf{G}_e \mathsf{D}_1^\top]\!] = R_2(\mathsf{D}_1^\top)[\![\mathsf{G}_e]\!] = a_2 R_2(\mathsf{D}_1^\top)\mathsf{T}[\![\mathsf{u}_e]\!]$$

so that the cell terms can be written as

$$\left[\!\!\left[\frac{\Delta t}{\Delta x_e} \mathsf{D}_1 \mathsf{F}_e + \frac{\Delta t}{\Delta y_e} \mathsf{G}_e \mathsf{D}_1^\top \right]\!\!\right] = (\sigma_1 R_1(\mathsf{D}_1)\mathsf{T} + \sigma_2 R_2(\mathsf{D}_1^\top)\mathsf{T})[\![\mathsf{u}_e]\!]$$

For the terms involving the numerical flux, let us consider the case $a_1 \geq 0$, $a_2 \geq 0$. Let $\mathsf{u}_l, \mathsf{u}_r, \mathsf{u}_b$ denote the solution in the elements to the left, right and bottom of the e'th element. Then, for the upwind flux which is obtained for dissipation model D2, we can renumber the terms involving the numerical flux as follows

$$\begin{aligned} \frac{\Delta t}{\Delta x_e} \left[\!\!\left[\mathsf{b}_L \mathsf{F}^\top_{e-\frac{1}{2}} + \mathsf{b}_R \mathsf{F}^\top_{e+\frac{1}{2}} \right]\!\!\right] &= \frac{\Delta t}{\Delta x_e} [\![a_1 \mathsf{b}_L \mathsf{V}_R^\top \mathsf{U}_l + a_1 \mathsf{b}_R \mathsf{V}_R^\top \mathsf{U}_e]\!] \\ &= \sigma_1 R_1(\mathsf{b}_L \mathsf{V}_R^\top)\mathsf{T}[\![\mathsf{u}_l]\!] + \sigma_1 R_1(\mathsf{b}_R \mathsf{V}_R^\top)\mathsf{T}[\![\mathsf{u}_e]\!] \end{aligned}$$

$$\begin{aligned} \frac{\Delta t}{\Delta y_e} \left[\!\!\left[\mathsf{G}_{e-\frac{1}{2}} \mathsf{b}_L^\top + \mathsf{G}_{e+\frac{1}{2}} \mathsf{b}_R^\top \right]\!\!\right] &= \frac{\Delta t}{\Delta y_e} [\![a_2 \mathsf{U}_b \mathsf{V}_R \mathsf{b}_L^\top + a_2 \mathsf{U}_e \mathsf{V}_R \mathsf{b}_R^\top]\!] \\ &= \sigma_2 R_2(\mathsf{V}_R \mathsf{b}_L^\top)\mathsf{T}[\![\mathsf{u}_b]\!] + \sigma_2 R_2(\mathsf{V}_R \mathsf{b}_R^\top)\mathsf{T}[\![\mathsf{u}_e]\!] \end{aligned}$$

The update equation can be written as

$$[\![\mathsf{u}_e^{n+1}]\!] = \mathsf{A}_l [\![\mathsf{u}_l^n]\!] + \mathsf{A}_e [\![\mathsf{u}_e^n]\!] + \mathsf{A}_b [\![\mathsf{u}_b^n]\!]$$

where the coefficient matrices are given by

$$\mathsf{A}_l = -\sigma_1 R_1(\mathsf{b}_L \mathsf{V}_R^\top)\mathsf{T}, \qquad \mathsf{A}_b = -\sigma_2 R_2(\mathsf{V}_R \mathsf{b}_L^\top)\mathsf{T}$$

$$\mathsf{A}_e = I - \sigma_1 R_1(\mathsf{D}_1)\mathsf{T} - \sigma_2 R_2(\mathsf{D}_1^\top)\mathsf{T} - \sigma_1 R_1(\mathsf{b}_R \mathsf{V}_R^\top)\mathsf{T} - \sigma_2 R_2(\mathsf{V}_R \mathsf{b}_R^\top)\mathsf{T}$$

Assuming a solution of the form $\mathsf{u}_e^n = \hat{\mathsf{u}}_k^n \exp(\mathsf{i}(k_1 x_e + k_2 y_e))$, we get the amplification equation

$$[\![\hat{\mathsf{u}}_k^{n+1}]\!] = (\mathsf{A}_l \exp(-\mathsf{i}\kappa_1) + \mathsf{A}_e + \mathsf{A}_b \exp(-\mathsf{i}\kappa_2))[\![\hat{\mathsf{u}}_k^n]\!] =: H(\sigma_1, \sigma_2; \kappa_1, \kappa_2)[\![\hat{\mathsf{u}}_k^n]\!]$$

where $\kappa_1 = k_1 \Delta x$ and $\kappa_2 = k_2 \Delta y$. For stability, it is required that the spectral radius of the matrix $H(\sigma_1, \sigma_2; \kappa_1, \kappa_2)$ is less than or equal to one for all wave numbers $\kappa_1, \kappa_2 \in [0, 2\pi]$. Numerically, we compute the region consisting of the pairs (σ_1, σ_2) that ensure the stability. These regions for different degrees with dissipation model D2 are given in Figures (4.30) and (4.31) for the Radau and g_2 correction functions, respectively. We set CFL$=2c$, where $c := \max\{\sigma : (\sigma, \sigma) \text{is a stable pair}\}$ which is the CFL limit when the advection velocity is in the direction $(1, 1)$. These CFL numbers for different degrees are given in Table 4.2. We see from the figures that the stable domain is bounded by a straight line except in case of degree $N = 1$ so that this region is given by

$$|\sigma_1| + |\sigma_2| \leq \text{CFL} \tag{4.27}$$

If the advection velocity is along the x or y axis, the CFL corresponds to that of the 1-D case, but if the velocity is at an angle to the grid, then the allowed time step is reduced. This is because of the one dimensional numerical fluxes employed at the cell faces which couple each cell only to its left/right and bottom/top cells, without any coupling with the diagonal neighbours.

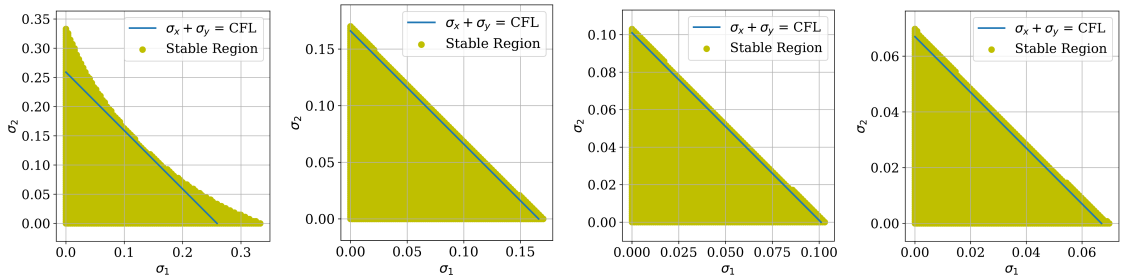


Figure 4.30. Stability regions of LWFR scheme with the Radau correction function and D2 dissipation model in two dimensions. (a) $N = 1$, (b) $N = 2$, (c) $N = 3$, (d) $N = 4$.

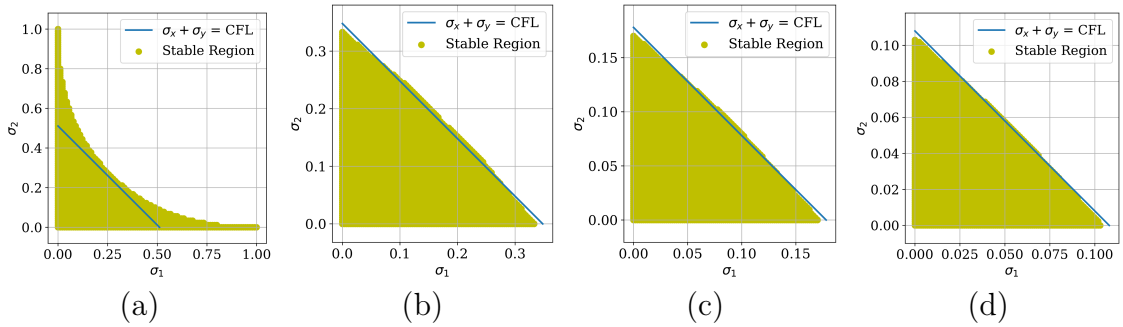


Figure 4.31. Stability regions of LWFR scheme with g_2 correction function and D2 dissipation model in two dimensions. (a) $N = 1$, (b) $N = 2$, (c) $N = 3$, (d) $N = 4$.

N	1	2	3	4
Radau	0.259	0.166	0.101	0.067
g_2	0.511	0.348	0.178	0.108

Table 4.2. Two dimensional CFL numbers for LWFR scheme (satisfying (4.27)) with dissipation model D2 and two correction functions.

4.10. NUMERICAL RESULTS IN 2D: SCALAR PROBLEMS

We present results to test the error convergence properties of the LW schemes for some 2-D problems and compare them to RK scheme. For each problem in this section, the corresponding CFL numbers are chosen based on Fourier stability analysis which are given in Table 4.2. We compare Lax-Wendroff scheme with D2 dissipation model and Runge-Kutta schemes in this section, and the CFL numbers of the former are used for both schemes. For the RKFR scheme, we use SSP Runge-Kutta time integration [82] for $N=1$ and 2, the classical four stage Runge-Kutta method of order four for $N=3$, and six-stage, fifth order Runge-Kutta (RK65) time integration for $N=4$ [183] implemented in `DifferentialEquations.jl` [139]. All the results in this section are produced using code written in Julia [29]; the design and optimization of the code was inspired by `Trixi.jl` [141].

4.10.1. Advection of a smooth signal

We consider the advection equation in two dimensions

$$u_t + \nabla \cdot [\boldsymbol{a}(x,y)\, u] = 0, \tag{4.28}$$

with two types of divergence-free advection velocity, namely a constant velocity $\boldsymbol{a}=(1,1)$ and a variable velocity $\boldsymbol{a}=(-y,x)$. For the second velocity, the flux components are $(f,g)=(-y\,u, x\,u)$ so that F_h^δ is of degree N in x and G_h^δ is of degree N in y variable. Since F_h^δ is interpolated along x direction and G_h^δ is interpolated along y direction, there is no interpolation error due to non-linearity of the flux and the **EA** and **AE** schemes are equivalent. Due to this reason, we only show the results with **EA** scheme. In order to verify the accuracy of the LWFR scheme we consider the equation (4.28) with a smooth initial condition and perform the simulation for both the advection velocities. For the velocity $\boldsymbol{a}=(1,1)$, the characteristic curves are straight lines and we use periodic boundary conditions on the domain $[0,1]\times[0,1]$ with initial condition $u_0(x,y)=\sin(2\pi x)\sin(2\pi y)$. The error convergence plots are shown in Figure 4.32 using Radau correction function. The optimal convergence rate is attained by both LW and RK schemes and there is no significant difference between GL and GLL points. The errors of RK scheme are slightly smaller than those of the LW scheme, similar to the 1-D case.

For the variable velocity $\boldsymbol{a}=(-y,x)$, the characteristic curves are circles whose center is at the origin and we take the domain $\Omega=[0,1]\times[0,1]$. The exact solution is given by $u(x,y,t)=u_0(x\cos(t)+y\sin(t), -x\sin(t)+y\cos(t))$ with the initial con-

dition $u_0(x,y) = 1 + \exp(-50\,((x-1/2)^2 + y^2))$. At the bottom and right boundaries, we use inflow conditions while on top and left side of the boundary, we use outflow conditions. The initial condition advects along the circular characteristic curves in the counter clock-wise direction. A contour plot of the numerical solution is visualized in Figure 4.33, and the error convergence analysis is made in Figure 4.34. The error convergence agrees with the optimal convergence rates and the error values of the LW scheme are comparable to those from the RK scheme at all orders shown in the figures.

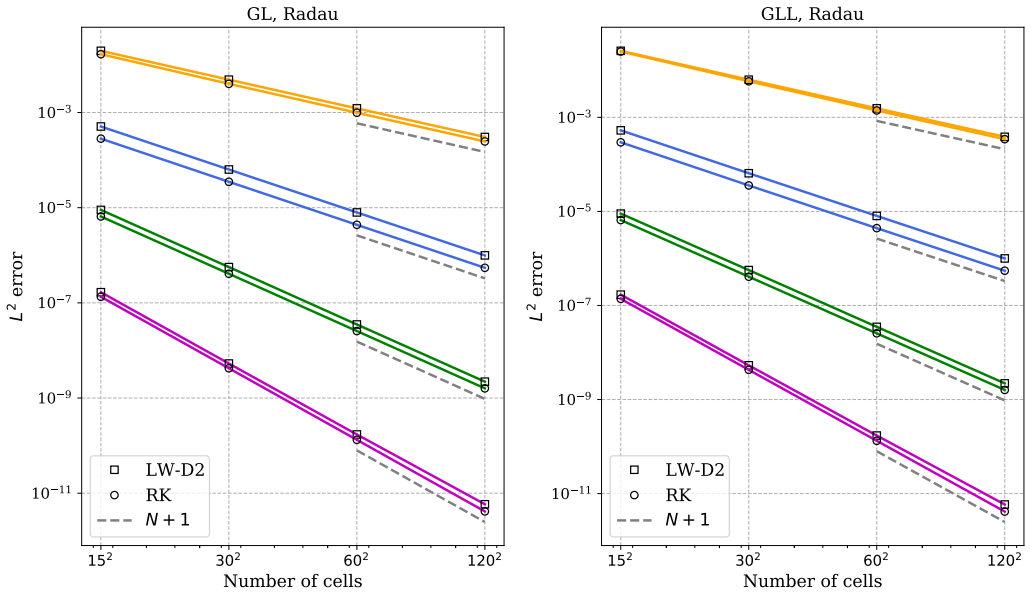


Figure 4.32. Error convergence test for 2-D linear advection equation with velocity $\boldsymbol{a} = (1,1)$ at $t = 1$, initial data $u_0(x,y) = \sin(2\pi x)\sin(2\pi y)$; (a) GL points, (b) GLL points. The different colors correspond to degrees $N = 1,2,3,4$ from top to bottom.

4.10.2. Rotation of a composite signal

In this example, we consider a classical test case [117] where the equation (4.28) is solved with a divergence free velocity field $\boldsymbol{a} = (\frac{1}{2} - y, x - \frac{1}{2})$ and an initial condition which consists of a slotted disc, a cone and a smooth hump, given as follows

$$
\begin{aligned}
u_0(x,y) &= u_1(x,y) + u_2(x,y) + u_3(x,y), \quad (x,y) \in [0,1]\times[0,1] \\
u_1(x,y) &= \frac{1}{4}\left(1 + \cos(\pi\, q(x,y))\right) \\
q(x,y) &= \min\left(\sqrt{(x-\bar{x})^2 + (y-\bar{y})^2}, r_0\right)/r_0, \quad (\bar{x},\bar{y}) = (0.25, 0.5), \quad r_0 = 0.15 \\
u_2(x,y) &= \begin{cases} 1 - \dfrac{1}{r_0}\sqrt{(x-\bar{x})^2 + (y-\bar{y})^2} & \text{if } (x-\bar{x})^2 + (y-\bar{y})^2 \le r_0^2 \\ 0 & \text{otherwise} \end{cases}, \\
& \qquad (\bar{x},\bar{y}) = (0.5, 0.25), \quad r_0 = 0.15 \\
u_3(x,y) &= \begin{cases} 1 & \text{if } (x,y) \in \mathrm{C} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

where C is a slotted disc with center at $(0.5, 0.75)$ and radius of 0.15. The initial condition is shown in Figure 4.35a. The numerical solutions of LWFR and RKFR after one rotation, without limiter, degree $N = 3$ and 100×100 cells, are shown in Figures 4.35b,d respectively. The same results with a TVB limiter ($M = 100$) are shown in Figures 4.35c,e. Without the limiter, the solution is captured well but there are some oscillations that take the solution outside the initial range of values. With the TVB limiter, the oscillations are reduced though it is not completely eliminated and results in increased numerical dissipation that smears the discontinuous profiles. However, in all cases, LWFR scheme performs comparably with RKFR scheme with the same limiter settings.

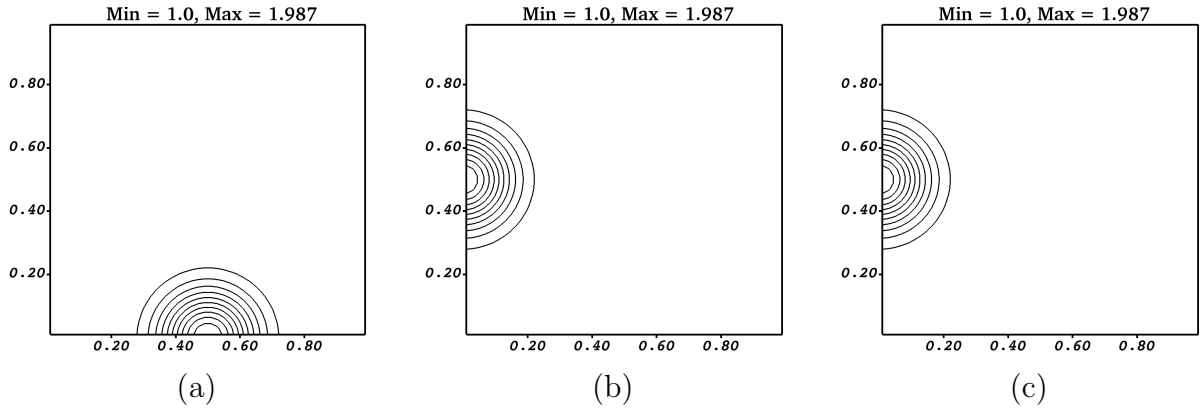


Figure 4.33. Linear advection with velocity $\boldsymbol{a} = (-y, x)$ on $[0, 1] \times [0, 1]$ with inflow/outflow boundary condition. The solutions are shown on a mesh of 50×50 cells with polynomial degree $N = 3$; (a) initial solution, (b) LWFR, $t = \frac{\pi}{2}$ (c) RKFR, $t = \frac{\pi}{2}$.

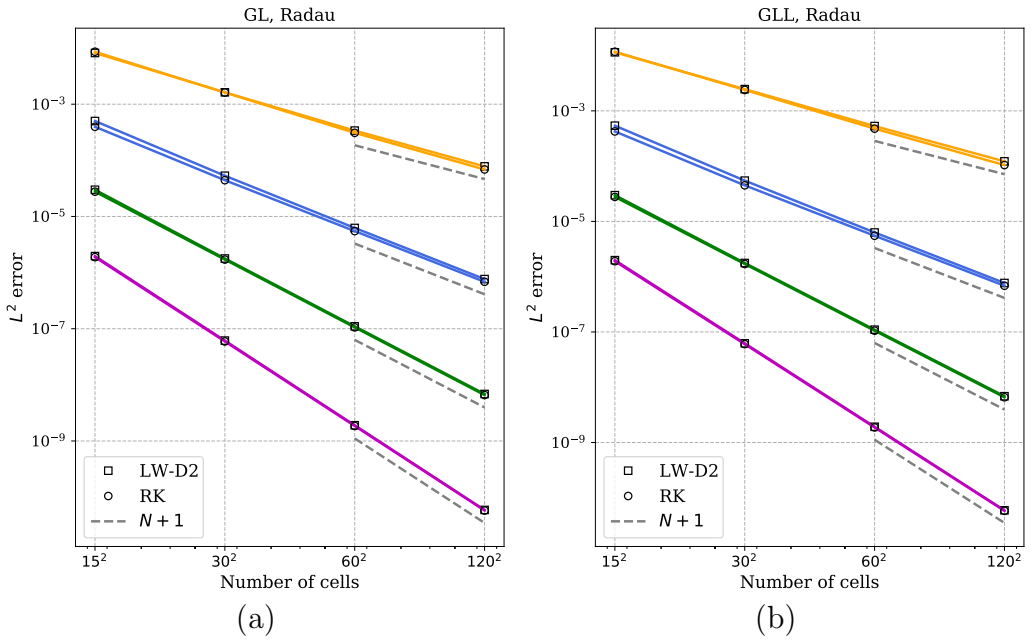
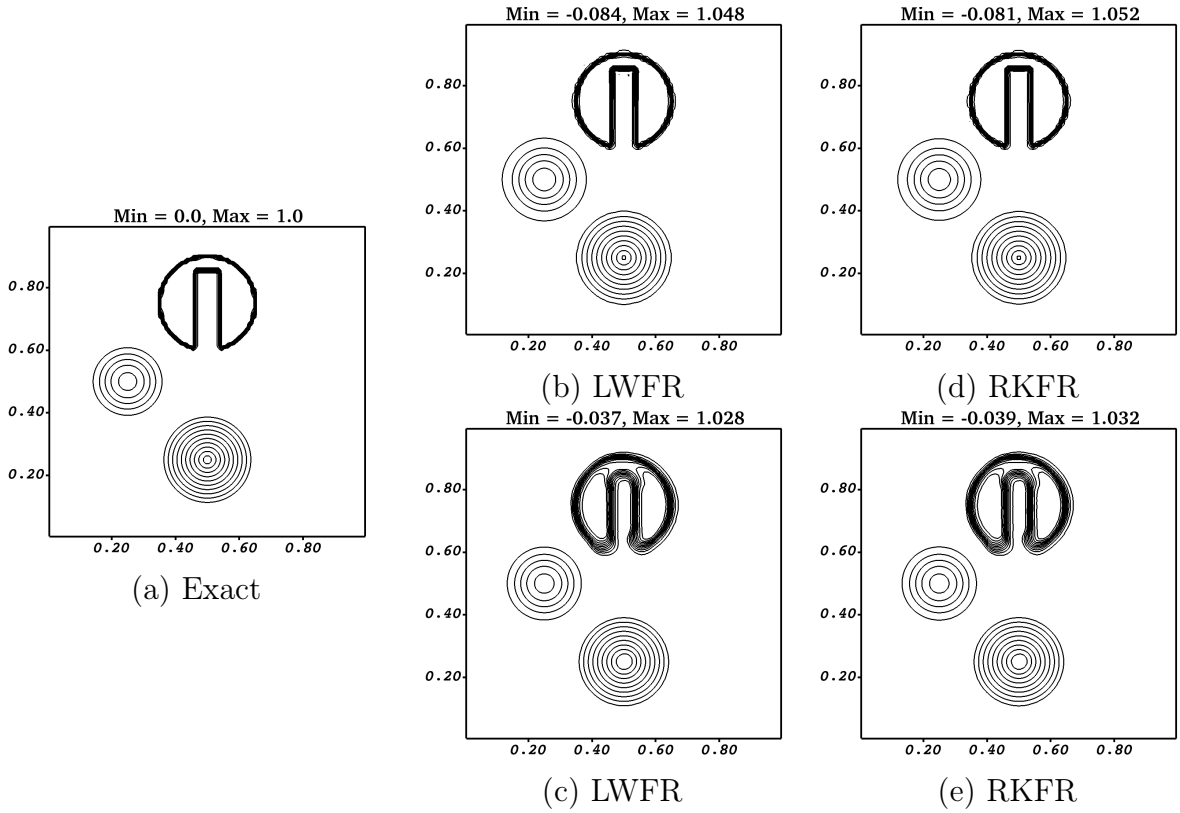


Figure 4.34. Error convergence test for 2-D linear advection equation with velocity $\boldsymbol{a} = (-y, x)$ at $t = \frac{\pi}{2}$, initial data $u_0(x, y) = 1 + \exp(-50((x - 1/2)^2 + y^2))$ using (a) GL points, (b) GLL points.



(a) Exact (b) LWFR (d) RKFR (c) LWFR (e) RKFR

Figure 4.35. Numerical solutions of composite signal with velocity $\boldsymbol{a} = \left(\frac{1}{2} - y, x - \frac{1}{2}\right)$ obtained for degree $N = 3$ using Radau correction function and GL solution points. The solutions are plotted after 1 period of rotation on a mesh of 100×100 cells; No limiter is used in (b), (c) and TVB limiter ($M = 100$) is used in (d), (e).

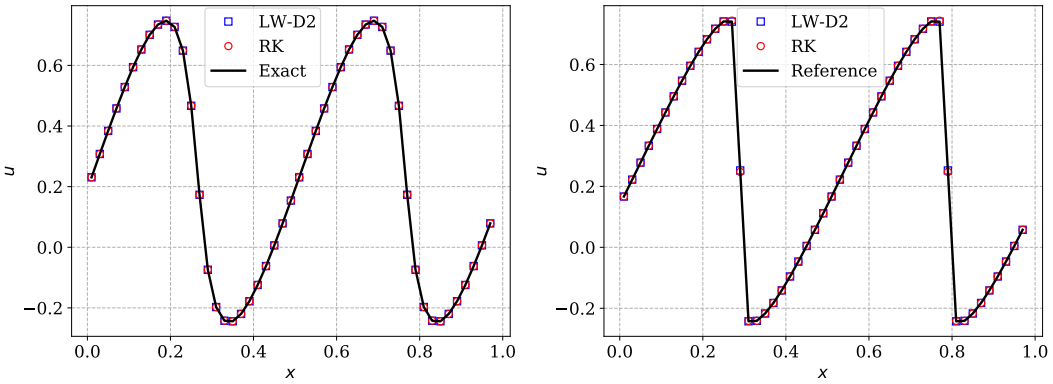
4.10.3. Inviscid Burgers' equation

We test the accuracy and robustness of the LWFR scheme for the two dimensional nonlinear scalar problem by considering a Burger-type equation [137]

$$u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0 \tag{4.29}$$

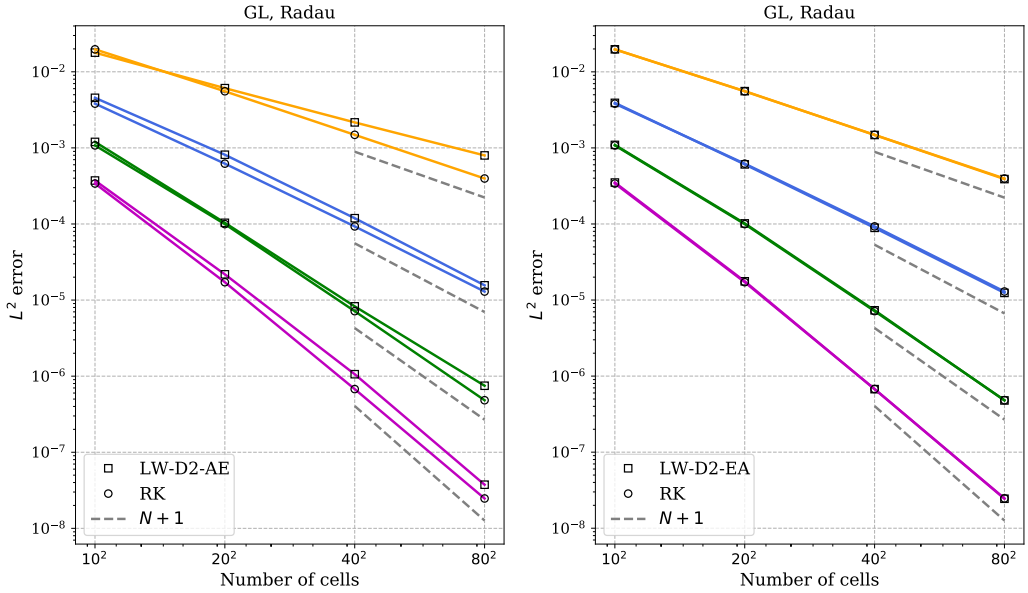
with an initial condition $u(x, y, 0) = \frac{1}{4} + \frac{1}{2}\sin(2\,\pi\,(x + y))$ in the domain $\Omega = [0, 1] \times [0, 1]$. The boundary conditions are set to be periodic in both directions. To test the error convergence, the solutions are computed up to time $t = 0.1$ as shown in Figure 4.36a, when the solutions are still smooth and the exact solution is available. The error convergence results up to degree four are given in Figure 4.37 using D2 dissipation model and Radau correction function. Similar to that in the 1-D case, the **AE** scheme shows optimal convergence rate for even polynomial degrees but sub-optimal convergence rates for odd degrees. The **EA** scheme on the other hand shows optimal convergence rates at all degrees and the error values are also comparable to those of RK scheme. In order to show the robustness of the LWFR scheme, we compute the numerical solution at time $t = 0.2$ where the solution is discontinuous. The corresponding solution across the diagonal of the domain for mesh size 50×50 with $N = 3$ is shown in Figure 4.36b which shows that the shock is captured accurately and

without spurious oscillations. In each case, when the interface fluxes are computed with **EA** scheme, the LWFR schemes perform at par with the RKFR schemes.



(a) $t = 0.1$ (b) $t = 0.2$

Figure 4.36. Line plot across the diagonal of $[0, 1] \times [0, 1]$ of the solution of 2-D Burgers' equation with 50 × 50 cells and degree $N = 3$. The reference solution for $t = 0.2$ is computed using RKFR scheme with degree $N = 1$ on a mesh of 1000 × 1000 cells.



(a) (b)

Figure 4.37. Error convergence test for 2-D Burgers' equation with initial condition $u(x, y, 0) = \frac{1}{4} + \frac{1}{2}\sin(2\pi(x + y))$ in the domain $[0, 1] \times [0, 1]$ comparing the two boundary fluxes of LWFR (a) **AE**, (b) **EA**. The errors are computed at $t = 0.1$.

4.11. NUMERICAL RESULTS IN 2-D: EULER EQUATIONS

We consider the two-dimensional Euler equations of gas dynamics (2.13). We present results to test the accuracy and computational performance of the LW schemes for some 2-D problems and compare them to RK scheme. The time step size for polynomial degree N is computed as

$$\Delta t = C_{\text{CFL}} \min_e \left(\frac{|\bar{u}_e| + \bar{c}_e}{\Delta x_e} + \frac{|\bar{v}_e| + \bar{c}_e}{\Delta y_e} \right)^{-1} \text{CFL}(N) \tag{4.30}$$

where e is the element index, $(\bar{u}_e, \bar{v}_e), \bar{c}_e$ are velocity and sound speed of element mean in element e, CFL(N) is the optimal CFL number of the scheme and $C_{\text{CFL}} \leq 1$ is a safety factor. For each problem in this section, the corresponding CFL numbers of Lax-Wendroff schemes are chosen based on the Fourier stability analysis which are given in Table 4.2. For a fair performance comparison, the Runge-Kutta schemes use their optimal CFL numbers [76]. The C_{CFL} is taken to be 0.98 in all results. The time averaged flux is always computed using the **EA** scheme. For RKFR, we use SSP Runge-Kutta time integration [82] for degrees $N = 1$ and 2, the five stage SSP Runge-Kutta method of order four for $N = 3$ [167], and six-stage, fifth order Runge-Kutta (RK65) time integration for $N = 4$ [183] implemented in `DifferentialEquations.jl` [139]. We make use of the HLLC flux with wave speeds from [23]. All the results in this section are produced using code written in Julia [29].

4.11.1. Isentropic vortex

We perform error convergence and Wall Clock Time (WCT) analysis using the isentropic vortex test case [200, 166]. This problem consists of a vortex that advects at a constant velocity while the entropy is constant in both space and time. The initial condition is given by

$$\rho = \left[1 - \frac{\beta^2 (\gamma - 1)}{8 \gamma \pi^2} \exp\left(1 - r^2\right)\right]^{\frac{1}{\gamma - 1}}, \qquad u = M \cos\alpha - \frac{\beta (y - y_c)}{2\pi} \exp\left(\frac{1 - r^2}{2}\right)$$

$$v = M \sin\alpha + \frac{\beta (x - x_c)}{2\pi} \exp\left(\frac{1 - r^2}{2}\right), \qquad r^2 = (x - x_c)^2 + (y - y_c)^2$$

and the pressure is given by $p = \rho^\gamma$. We choose the parameters $\beta = 5$, $M = 0.5$, $\alpha = 45^o$, $(x_c, y_c) = (0, 0)$ and the domain is taken to be $[-10, 10] \times [-10, 10]$ with periodic boundary conditions. For this configuration, the vortex returns to its initial position after a time interval of $T = 20\sqrt{2}/M$ units. We run the computations up to a time $t = T$ when the vortex has crossed the domain once in the diagonal direction.

The L^2 error and Wall Clock Time (WCT) against grid resolution is shown in Figure 4.38. We observe optimal convergence rates for all new LW schemes proposed in this paper. The WCT scales as the total number of cells to the power of 1.5, which is the expected rate and the LW-D2 scheme shows smallest time as seen in Figure 4.38b. We denote by WCT(LW-D1) the Wall Clock Time corresponding to LW-D1 scheme, and similarly for other schemes.

The WCT versus L^2 error comparison has been made in Figure 4.39 and the ratios WCT(LW-D1)/WCT(LW-D2) and WCT(RK)/WCT(LW-D2) are plotted against grid resolution in Figures 4.40a and 4.40b, respectively. We see that the newly proposed LW-D2 scheme is more efficient in comparison to the LW-D1 scheme since it can use a larger CFL number. As we expect from Table 4.1 comparing the CFL ratios, the explicit time ratios of LW-D1 and LW-D2 are consistently in the range of 1.4 and 1.5 for $N > 1$, as shown in Figure 4.40a.

Figure 4.39 shows that LW-D2 has smaller Wall Clock Time than RK for all degrees and Figure 4.40b shows that the WCT ratio WCT(RK)/WCT(LW-D2) is close to

1.1, 1.4, 1.7 for $N = 1, 2, 3$ respectively. Thus, the ratio improves as we increase the degree up to 3. However, at $N = 4$, the ratio deteriorates to approximately 1.2. The low CFL number of LW at $N = 4$ relative to the RK scheme plays a role in this loss of performance. Figure 4.39 shows that when the error levels are small, the higher order schemes are more efficient in terms of WCT than lower order methods.

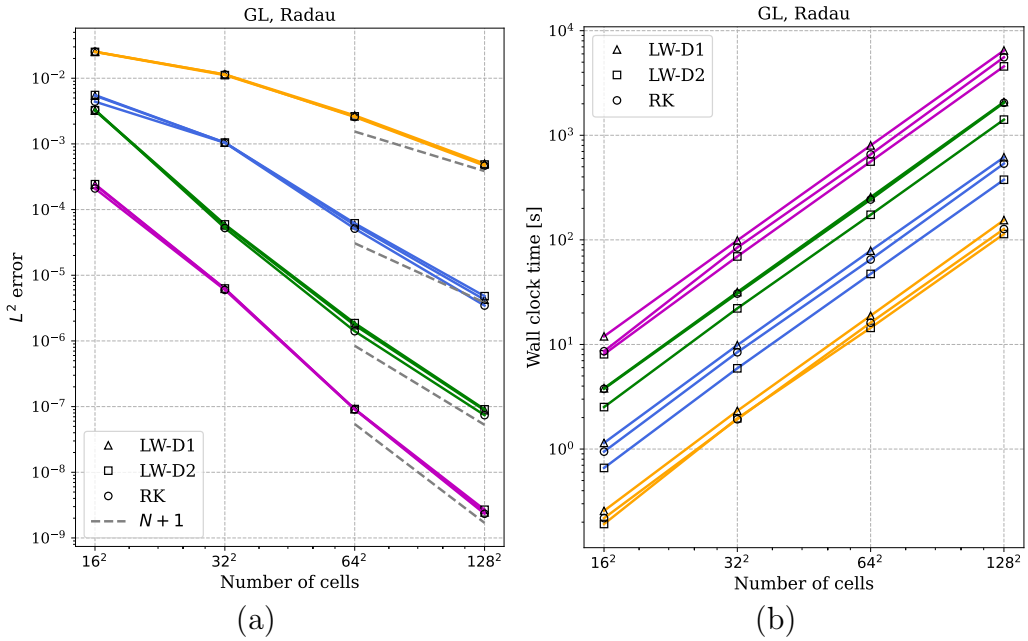


Figure 4.38. L^2 error and Wall Clock Time (WCT) analysis of 2-D Euler equations (isentropic vortex) against grid resolution comparing LW-D1, LW-D2 and RK is shown in (a) and (b) respectively. The error is computed after one period. The time step size of each scheme is computed using its optimal CFL from Fourier stability analysis.

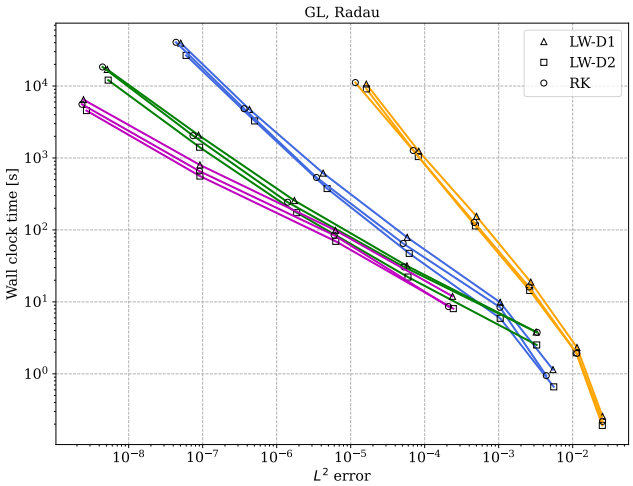


Figure 4.39. Wall Clock Time (WCT) versus L^2 error for 2-D Euler equations (isentropic vortex) comparing LW-D1, LW-D2 and RK for degrees $N = 1, 2, 3, 4$. The different colors correspond to different degrees, with the degree increasing from right to left. The error is computed after one period. The time step size of each scheme is computed using its optimal CFL number from Fourier stability analysis.

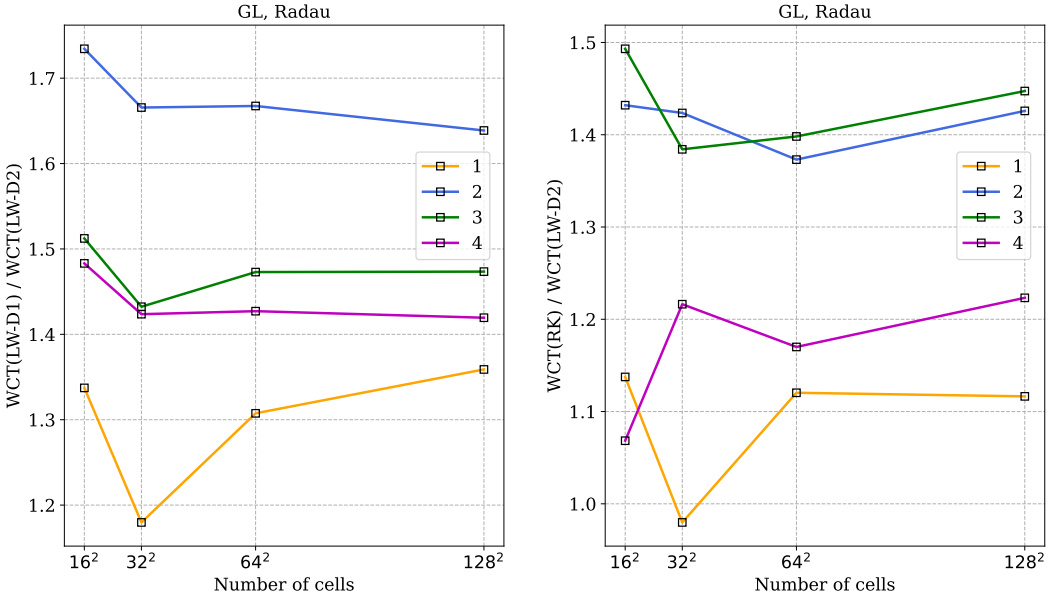


Figure 4.40. Wall Clock Time (WCT) ratios versus grid resolution for 2-D Euler equations (isentropic vortex). (a) WCT ratio of LW-D1 and LW-D2, (b) WCT ratio of RK and LW-D2. The error is computed after one period. The time step size of each scheme is computed using its optimal CFL number from Fourier stability analysis.

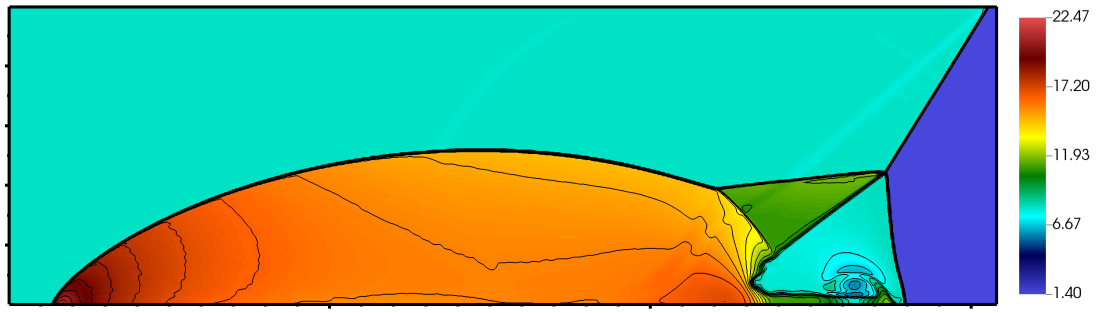
4.11.2. Double Mach reflection

We now test the double Mach reflection problem which was originally proposed by Woodward and Colella [197]. The problem consists of a shock impinging on a wedge/ramp which is inclined by 30 degrees. An equivalent problem is obtained on the rectangular domain $\Omega = [0,4] \times [0,1]$ obtained by rotating the wedge so that the initial condition now consists of a shock angled at 60 degrees. The solution consists of a self similar shock structure with two triple points. Defining $\boldsymbol{u}_b = \boldsymbol{u}_b(x,y,t)$ with primitive variables given by

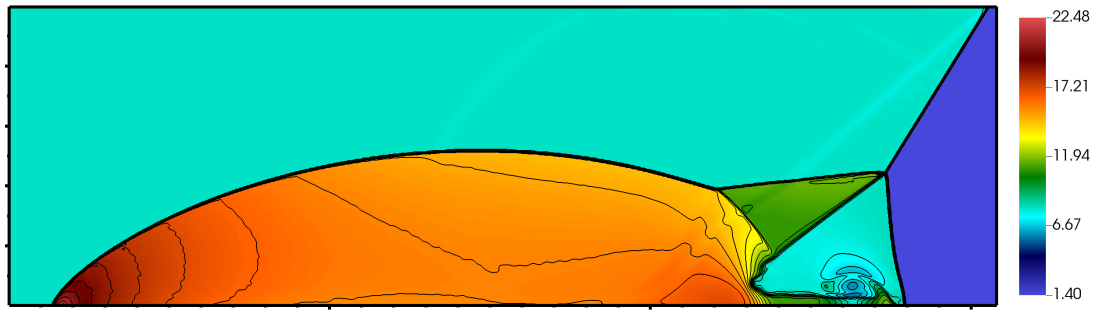
$$(\rho,u,v,p) = \begin{cases} \left(8, 8.25\cos\left(\frac{\pi}{6}\right), -8.25\sin\left(\frac{\pi}{6}\right), 116.5\right) & \text{if } x < \frac{1}{6} + \frac{y+20t}{\sqrt{3}} \\ (1.4,0,0,1) & \text{if } x > \frac{1}{6} + \frac{y+20t}{\sqrt{3}} \end{cases}$$

we define the initial condition to be $\boldsymbol{u}_0(x,y) = \boldsymbol{u}_b(x,y,0)$. With $\boldsymbol{u}_b$, we impose inflow boundary conditions at the left side $\{0\} \times [0,1]$, outflow boundary conditions both at $[0, 1/6] \times \{0\}$ and $\{4\} \times [0,1]$, reflecting boundary conditions at $[1/6, 4] \times \{0\}$ and inflow boundary conditions at the upper side $[0,4] \times \{1\}$. In Figure (4.41), we compare the density plots obtained using the LWFR and RKFR schemes for $N = 2$ at a resolution of 960×240 cells at $t = 0.2$. The non-linear TVB limiter is used with the parameter $M = 100$ [137]. The Lax-Wendroff solution is computed using D2 dissipation and **EA** scheme. We use GL points and Radau corrector in both LW and RK schemes. We observe similar resolution for both schemes; the similarity holds for other degrees also, which we have not shown to save space. In Figure 4.43a, we plot grid resolution against

the Wall Clock Time for degrees $N = 1, 2, 3, 4$ which shows the expected dependence of time with grid size. Figure 4.43b shows the ratio of WCT for RK and LW-D2 schemes, indicating better efficiency of LW scheme, and these results are similar to what is observed in [137].

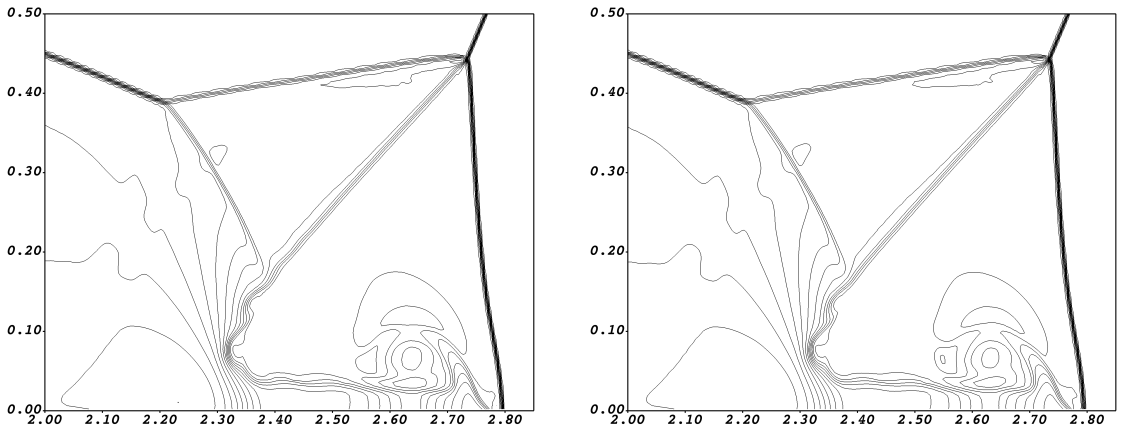


(a) LWFR-D2 with **EA** scheme



(b) RKFR

Figure 4.41. Density profile of numerical solutions of 2-D Euler equations (double Mach reflection problem) at $t = 0.2$ for $N = 2$, with $\Delta x = \Delta y = 1/240$. Contours of 30 steps from 1.4 to 22.5 are printed.



(a) (b)

Figure 4.42. Enlarged contours of density (2-D Euler equations, double Mach reflection problem) at $t = 0.2$ for $N = 2$, with $\Delta x = \Delta y = 1/240$. Contours of 30 steps from 1.4 to 22.5 are printed.

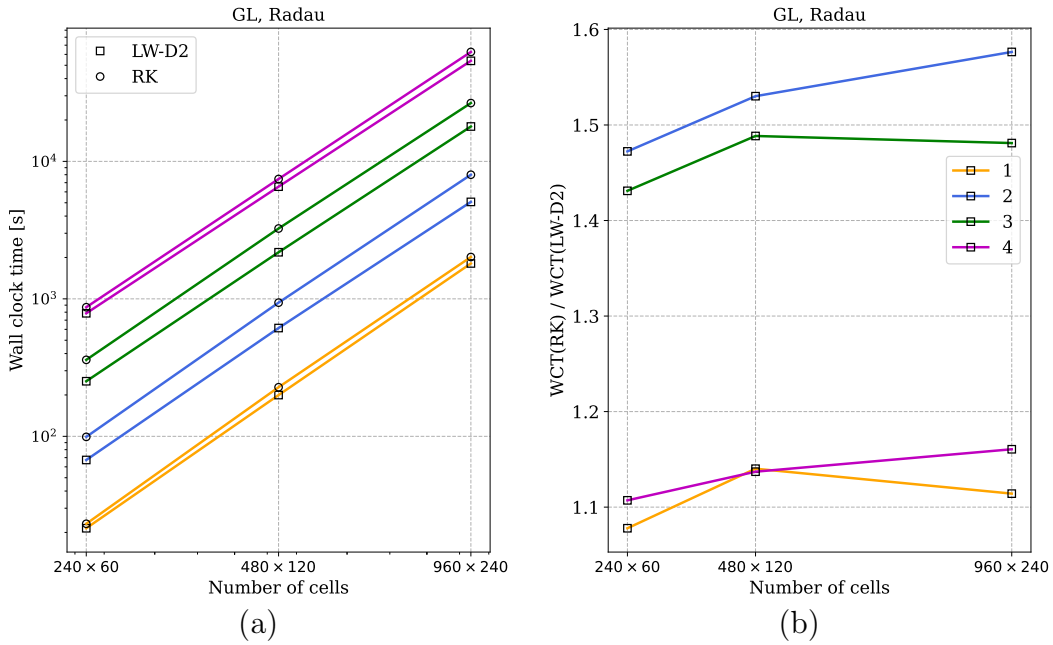


Figure 4.43. Grid size versus WCT comparison of RK and LW schemes for 2-D Euler equations (double Mach reflection problem). Time step of each scheme has been chosen with its optimal CFL number from Fourier stability analysis.

4.12. SUMMARY

A conservative, Jacobian-free and single step, explicit Lax-Wendroff method has been constructed in a flux reconstruction context, and its implementation has been demonstrated for solving hyperbolic conservation laws in one and two dimensions. The Jacobian-free property is achieved by using a finite difference approach to compute time derivatives of the fluxes that are needed in the Taylor expansion. The method requires only the time average flux and its corresponding numerical fluxes. It is written in matrix-vector form that is useful for computer implementation. We have studied the effect of two commonly used correction functions and solution points. The stable CFL numbers are computed using Fourier stability analysis in one and two dimensions. The numerical fluxes are computed using both the time average flux and the time average solution which leads to improved CFL numbers compared to other existing methods which use the solution at previous time level to compute the dissipative part of the numerical flux. At fifth order ($N = 4$), there is a mild linear instability for periodic problems, which seems to be present in other single step methods and also in RKDG schemes. For non-linear problems, we identify a loss of optimal convergence rate when a simple average-extrapolate (**AE**) approach is used to compute the central part of the numerical flux. We show that this can be improved to optimal rates by using an extrapolate-average (**EA**) procedure, and the resulting schemes perform comparably with RK schemes in terms of their error levels. The performance of the method is also demonstrated on 1-D and 2-D non-linear systems like Euler equations, where it is able to resolve all the waves at comparable accuracy to RK schemes. Many commonly used numerical fluxes based on approximate Riemann solvers and modeling even contact waves can be developed and used in these schemes. These

studies show that the Radau correction function in combination with Gauss-Legendre solution points and the extrapolate-average (**EA**) technique leads to uniformly accurate LW scheme for non-linear problems. The method has a simple structure which makes it easy to develop a general code that can be used to solve any conservation law; the user has to supply subroutines for the flux, numerical flux, and maximum wave speed estimate used in the CFL condition.

Chapter 5

Admissibility preserving subcell limiter

5.1. Introduction

In this chapter, we develop a subcell based blending limiter for Lax-Wendroff Flux Reconstruction (LWFR) motivated by the work of [90]. The idea is to break each element into subcells and construct a robust low order method on the subcells. A smoothness indicator is then used to blend the high order LWFR scheme with the low order scheme, getting a robust limited scheme. In the development of the blending scheme for LWFR, special attention has been paid to improving accuracy and obtaining provable admissibility preservation. In contrast to [90], we use Gauss-Legendre solution points because of their accuracy benefit known in the literature and also observed by us in Chapter 4. The low order scheme on subcells is a finite volume method. A natural choice is to use a first order finite volume method, but to enhance accuracy we develop a MUSCL-Hancock scheme on the subcells. For admissibility preservation, we exploit the subcell structure of the blending scheme to develop a problem independent flux limiter that guarantees admissibility preservation in means.

The rest of this chapter is organized as follows. In Section 5.2, we formalize the concept of admissibility of a physical and numerical (FR) solution of hyperbolic conservation laws (3.1). In Section 5.3, we explain the blending limiter including a review of the smoothness indicator used in [90] and then MUSCL-Hancock reconstruction performed on the subcells in Section 5.4. Maintaining conservation requires that at the faces of FR elements, both the lower and high order schemes must use the same numerical flux (see Remark 5.3). In Section 5.5, we show how to construct the numerical flux to ensure admissibility preservation in means. In Section 5.6, we explain our implementation of the Lax-Wendroff blended scheme as an algorithm. The numerical results verifying accuracy and robustness of our scheme with 1-D and 2-D compressible Euler equations are shown in Sections 5.7, 5.8, 5.9. Section 5.10 gives a summary of the proposed blending scheme.

5.2. Admissibility preservation

The solution $\boldsymbol{u} \in \mathbb{R}^p$ of the conservation law (3.1) that is physically correct is assumed to belong to an admissible set, denoted by $\mathcal{U}_{\text{ad}}$. For example, in case of compressible flows, the density and pressure (or internal energy) must remain positive. In case of shallow water equations, the water depth must remain positive. In most of the models that are of interest, the admissible set is a convex subset of $\mathbb{R}^p$, and can be written as

$$\mathcal{U}_{\text{ad}} = \{\boldsymbol{u} \in \mathbb{R}^p : P_k(\boldsymbol{u}) > 0, 1 \leq k \leq K\} \tag{5.1}$$

Moreover, in most cases, the admissibility constraints P_k are concave functions of the conservative variables. In particular, we may have concavity of P_k if $P_j > 0$ for all $j < k$. For Euler's equations, $K = 2$ and P_1, P_2 are density, pressure functions respectively; if the density is positive then pressure is a concave function of the conserved variables. This structure simplifies the slope and flux limiting steps to enforce admissibility (Section 5.4.1, 5.5) and was assumed in [19]. However, there are models of interest where the admissibility constraints are not concave functions of the conservative variables, like ten moment equations which are considered in Chapter 6. For those models, our admissibility enforcing procedure will instead use the following weaker assumption

$$P_j(\boldsymbol{u}_a), P_j(\boldsymbol{u}_b) > 0, \ \forall j \leq k \qquad \Longrightarrow \qquad P_j(\theta \boldsymbol{u}_a + (1-\theta)\boldsymbol{u}_b) > \epsilon_j(\boldsymbol{u}_a, \boldsymbol{u}_b), \quad \forall j \leq k \tag{5.2}$$

In case of the Ten moment problem, $\epsilon_3(\boldsymbol{u}_a, \boldsymbol{u}_b) = \frac{1}{2}\min\left(P_3(\boldsymbol{u}_a), P_3(\boldsymbol{u}_b)\right)$ (see (2.9) of [125]). Thus, although the numerical experiments in this chapter are performed on Compressible Euler's equations (2.13), we also discuss admissibility preservation in case the admissibility constraints P_k are not concave functions of the conservative variables.

The high order Lax-Wendroff Flux Reconstruction scheme to solve the conservation law (3.1) is as described in Chapter 4. In particular, we use the discrete scheme described in Section 4.2 with the time averaged numerical flux constructed using the D2 dissipation and **EA** flux described in Section 4.3. We define the element mean value of the numerical solution $\{\boldsymbol{u}_{e,p}\}$ (3.3) as

$$\bar{\boldsymbol{u}}_e = \sum_{p=0}^{N} \boldsymbol{u}_{e,p} w_p$$

where w_p are the weights associated to the solution points (3.2). Then, looking at the LWFR update (4.4), it is easy to show that the scheme is conservative in the sense that

$$\bar{\boldsymbol{u}}_e^{n+1} = \bar{\boldsymbol{u}}_e^{n} - \frac{\Delta t}{\Delta x_e}\left(\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}}\right) \tag{5.3}$$

The admissibility preserving property, also known as convex set preservation property since $\mathcal{U}_{\text{ad}}$ is convex, of the conservation law can be written as

$$\boldsymbol{u}(\cdot, t_0) \in \mathcal{U}_{\text{ad}} \qquad \Longrightarrow \qquad \boldsymbol{u}(\cdot, t) \in \mathcal{U}_{\text{ad}}, \qquad t > t_0 \tag{5.4}$$

and thus we define an admissibility preserving scheme to be

DEFINITION 5.1. *The flux reconstruction scheme is said to be admissibility preserving if*

$$\boldsymbol{u}_{e,p}^{n} \in \mathcal{U}_{\text{ad}} \quad \forall e, p \qquad \Longrightarrow \qquad \boldsymbol{u}_{e,p}^{n+1} \in \mathcal{U}_{\text{ad}} \quad \forall e, p$$

where $\mathcal{U}_{\text{ad}}$ is the admissible set of the conservation law.

To obtain an admissibility preserving scheme, we exploit the weaker admissibility preservation in means property defined as

Definition 5.2. *The flux reconstruction scheme is said to be admissibility preserving in the means if*

$$\boldsymbol{u}_{e,p}^n \in \mathcal{U}_{\text{ad}} \quad \forall e, p \qquad \Longrightarrow \qquad \bar{\boldsymbol{u}}_e^{n+1} \in \mathcal{U}_{\text{ad}} \quad \forall e$$

where $\mathcal{U}_{\text{ad}}$ is the admissible set of the conservation law.

The focus of this chapter is to obtain the admissibility preservation in means property for the Lax-Wendroff Flux Reconstruction scheme. Once the scheme is admissibility preserving in means, the scaling limiter of [206] can be used to obtain an admissibility preserving scheme in the sense of Definition 5.1.

5.3. On controlling oscillations

High order methods for hyperbolic problems necessarily produce Gibbs oscillations at discontinuities. In particular, it was shown by Godunov [81] that an oscillation free *linear scheme* can be at most first order accurate. The cure is to make the schemes to be non-linear even in the case of linear equations. For one dimensional problems, total variation diminishing approach provides a framework to construct non-oscillatory schemes. This is achieved by incorporating some non-linear limiting strategy into the scheme which locally reduces the order of the scheme when a discontinuity is detected. In discontinuous Galerkin type methods, the limiting is performed by modifying the solution in each element so as to ensure a TVD property for the element means, which was a strategy introduced by Cockburn and Shu [52, 51] and used in Chapter 4, as described in Section 4.6. In this chapter, we introduce the blending scheme, motivated and described in Section 5.3.1.

5.3.1. Blending scheme

In Chapter 4, TVD-type limiters of [52, 51] for DG methods were used. These limiters lose a lot of information when the limiter is active, since the polynomial solution of degree N is replaced either by a solution of degree 1 or a constant solution if a strong discontinuity is detected in an element. This is especially problematic near smooth extrema which may be wrongly detected as a discontinuity. It would be desirable to use more information inside each element while applying some limiting process. Let us write the LWFR update equation (4.4) as

$$\mathsf{u}_e^{H,n+1} = \mathsf{u}_e^n - \frac{\Delta t}{\Delta x_e} \mathsf{R}_e^H$$

where u_e is the vector of nodal values in the element. Suppose we also have a lower order and non-oscillatory scheme available to us in the form

$$\mathsf{u}_e^{L,n+1} = \mathsf{u}_e^n - \frac{\Delta t}{\Delta x_e} \mathsf{R}_e^L \tag{5.5}$$

Then a blended scheme is given by

$$\mathsf{u}_e^{n+1} = (1-\alpha_e)\,\mathsf{u}_e^{H,n+1} + \alpha_e\,\mathsf{u}_e^{L,n+1} = \mathsf{u}_e^n - \frac{\Delta t}{\Delta x_e}\left[(1-\alpha_e)\,\mathsf{R}_e^H + \alpha_e\,\mathsf{R}_e^L\right] \tag{5.6}$$

where $\alpha_e \in [0,1]$ must be chosen based on some local smoothness indicator. If $\alpha_e = 0$ then we obtain the high order LWFR scheme, while if $\alpha_e = 1$ then the scheme becomes the low order scheme that is less oscillatory. In subsequent sections, we explain the details of the lower order scheme and the design of smoothness indicators. The lower order scheme will either be a first order finite volume scheme (Section 5.3.3) or a high resolution scheme based on MUSCL-Hancock idea (Section 5.4). In either case, the common structure of the low order scheme can be explained as follows.

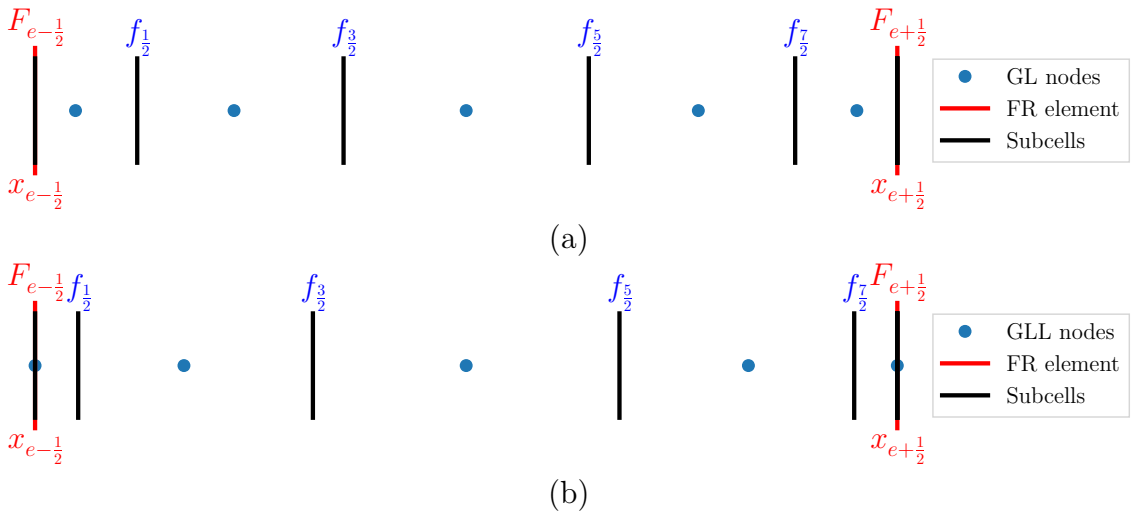


Figure 5.1. Subcells used by lower order scheme for degree $N = 4$ using (a) Gauss-Legendre (GL) solution points, (b) Gauss-Legendre-Lobatto (GLL) solution points

Let us subdivide each element Ω_e into $N+1$ subcells associated to the solution points $\{x_p^e, p = 0, 1, \ldots, N\}$ of the LWFR scheme. Thus, we will have $N+2$ subfaces denoted by $\{x^e_{p+\frac{1}{2}}, p = -1, 0, \ldots, N\}$ with $x^e_{-\frac{1}{2}} = x_{e-\frac{1}{2}}$ and $x^e_{N+\frac{1}{2}} = x_{e+\frac{1}{2}}$. For maintaining a conservative scheme, the p^{th} subcell is chosen so that

$$x^e_{p+\frac{1}{2}} - x^e_{p-\frac{1}{2}} = w_p\,\Delta x_e, \qquad 0 \le p \le N \tag{5.7}$$

where w_p is the p^{th} quadrature weight associated with the solution points. Figure 5.1 gives an illustration of the subcells for degree $N = 4$ case. The low order scheme is obtained by updating the solution in each of the subcells by a finite volume scheme,

$$\begin{aligned}
\boldsymbol{u}_{e,0}^{L,n+1} &= \boldsymbol{u}_{e,0}^n - \frac{\Delta t}{w_0\,\Delta x_e}\left[\boldsymbol{f}^e_{\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}}\right] \\
\boldsymbol{u}_{e,p}^{L,n+1} &= \boldsymbol{u}_{e,p}^n - \frac{\Delta t}{w_p\,\Delta x_e}\left[\boldsymbol{f}^e_{p+\frac{1}{2}} - \boldsymbol{f}^e_{p-\frac{1}{2}}\right], \qquad 1 \le p \le N-1 \\
\boldsymbol{u}_{e,N}^{L,n+1} &= \boldsymbol{u}_{e,N}^n - \frac{\Delta t}{w_N\,\Delta x_e}\left[\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{f}^e_{N-\frac{1}{2}}\right]
\end{aligned} \tag{5.8}$$

The inter-element fluxes $\boldsymbol{F}_{e+\frac{1}{2}}$ used in the low order scheme are same as those used in the high order LWFR scheme in equation (4.8). The lower order fluxes $\boldsymbol{f}^e_{p+\frac{1}{2}}$ will be taken to be admissibility preserving finite volume fluxes (Definition 3.1). The element mean value obtained by the low order scheme satisfies

$$\bar{\boldsymbol{u}}_e^{L,n+1} = \sum_{p=0}^{N} \boldsymbol{u}_{e,p}^{L,n+1} w_p = \bar{\boldsymbol{u}}_e^n - \frac{\Delta t}{\Delta x_e} \left(\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}} \right) \tag{5.9}$$

which is identical to the update equation by the LWFR scheme given in equation (5.3). The element mean in the blended scheme evolves according to

$$\begin{aligned} \bar{\boldsymbol{u}}_e^{n+1} &= (1-\alpha_e) \left(\bar{\boldsymbol{u}}_e\right)^{H,n+1} + \alpha_e \left(\bar{\boldsymbol{u}}_e\right)^{L,n+1} \\ &= (1-\alpha_e) \left[\bar{\boldsymbol{u}}_e^n - \frac{\Delta t}{\Delta x_e} \left(\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}} \right) \right] + \alpha_e \left[\bar{\boldsymbol{u}}_e^n - \frac{\Delta t}{\Delta x_e} \left(\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}} \right) \right] \\ &= \bar{\boldsymbol{u}}_e^n - \frac{\Delta t}{\Delta x_e} \left(\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}} \right) \end{aligned} \tag{5.10}$$

and hence the blended scheme is also conservative; all three schemes, i.e., lower order, LWFR and the blended scheme, predict the same mean value.

The inter-element flux $\boldsymbol{F}_{e+\frac{1}{2}}$ is used both in the low and high order schemes. To achieve high order accuracy in smooth regions, this flux needs to be high order accurate, however it may produce numerical oscillations near discontinuities when used in the low order scheme. A natural choice to balance accuracy and spurious oscillations is to take

$$\boldsymbol{F}_{e+\frac{1}{2}} = (1-\alpha_{e+\frac{1}{2}}) \, \boldsymbol{F}^{\mathrm{LW}}_{e+\frac{1}{2}} + \alpha_{e+\frac{1}{2}} \, \boldsymbol{f}_{e+\frac{1}{2}}, \qquad \alpha_{e+\frac{1}{2}} \in [0,1] \tag{5.11}$$

where $\boldsymbol{F}^{\mathrm{LW}}_{e+\frac{1}{2}}$ is the high order inter-element time-averaged numerical flux of the LWFR scheme (4.8) and $\boldsymbol{f}_{e+\frac{1}{2}}$ is an admissibility preserving low order flux (Definition 3.1) at the face $x_{e+\frac{1}{2}}$ shared between FR elements and subcells (5.14, 5.20). The blending coefficient $\alpha_{e+\frac{1}{2}}$ will be based on a local smoothness indicator which will bias the flux towards the lower order flux $\boldsymbol{f}_{e+\frac{1}{2}}$ near regions of lower solution smoothness. However, to enforce admissibility in means (Definition 5.2), the flux has to be further corrected, as explained in Section 5.5.

Remark 5.3.

a) It is essential to use the same inter-element flux in both the low and high order schemes in order to have conservation. Suppose we use numerical fluxes $\boldsymbol{F}^L_{e+\frac{1}{2}}$, $\boldsymbol{F}^H_{e+\frac{1}{2}}$ in the low and high order schemes, respectively; then the element mean in the blended scheme will become

$$\bar{\boldsymbol{u}}_e^{n+1} = \bar{\boldsymbol{u}}_e^n - \frac{\Delta t}{\Delta x_e} \left[\left((1-\alpha_e) \, \boldsymbol{F}^H_{e+\frac{1}{2}} + \alpha_e \, \boldsymbol{F}^L_{e+\frac{1}{2}} \right) - \left((1-\alpha_e) \, \boldsymbol{F}^H_{e-\frac{1}{2}} + \alpha_e \, \boldsymbol{F}^L_{e-\frac{1}{2}} \right) \right]$$

For conservation the flux leaving element Ω_e through $x_{e+\frac{1}{2}}$ must enter the neighbouring element Ω_{e+1}, i.e.,

$$(1-\alpha_e) \, \boldsymbol{F}^H_{e+\frac{1}{2}} + \alpha_e \, \boldsymbol{F}^L_{e+\frac{1}{2}} = (1-\alpha_{e+1}) \, \boldsymbol{F}^H_{e+\frac{1}{2}} + \alpha_{e+1} \, \boldsymbol{F}^L_{e+\frac{1}{2}}$$

i.e., $(\alpha_e - \alpha_{e+1})\, \boldsymbol{F}^L_{e+\frac{1}{2}} = (\alpha_e - \alpha_{e+1})\, \boldsymbol{F}^H_{e+\frac{1}{2}}$ which must hold for all values of α_e, α_{e+1} and hence we need $\boldsymbol{F}^L_{e+\frac{1}{2}} = \boldsymbol{F}^H_{e+\frac{1}{2}}$.

b) The contribution to $\mathtt{R}^L_e, \mathtt{R}^H_e$ of the flux $\boldsymbol{F}_{e+\frac{1}{2}}$ has coefficients given by $\frac{\Delta t}{w_N \Delta x_e}$, $\frac{\Delta t}{\Delta x_e} g_R'(\xi_N)$ respectively, as can be seen from (5.8, 8.28). If we use g_2 correction functions with Gauss-Legendre-Lobatto solution points, we have from (B.6), $g_R'(\xi_N) = \ell_N(1)/w_N = 1/w_N$. Thus, the coefficient is the same for both higher and lower order residuals and we add the contribution without a blending coefficient. This is different from the case of Gauss-Legendre solution points where the coefficients disagree as $1/w_N \neq \ell_N(1)/w_N = g_R'(\xi_N)$ (B.6).

5.3.2. Smoothness indicator

The numerical approximation of the PDE solution is in the form of piecewise polynomials of degree N. The polynomial can be written in terms of an orthogonal basis like Legendre polynomials. The smoothness of the solution can be assessed by analyzing the decay of the coefficients of the orthogonal expansion, a technique originally proposed by Persson and Peraire [134] and subsequently refined by Klöckner et al. [102] and Henemann et al. [90]. For a scalar problem, the solution $\boldsymbol{u}$ itself can be used to design a smoothness indicator. For a system of PDE, we can use any one or all components of the solution vector. Alternatively, some derived quantity that can indicate the smoothness of all solution components can be chosen. For the Euler equations, a good choice seems to be the product of density and pressure [90].

Let $q = q(\boldsymbol{u})$ be the quantity used to measure the solution smoothness. We first project this onto Legendre polynomials,

$$q_h(\xi) = \sum_{j=0}^{N} \hat{q}_j\, L_j(2\,\xi - 1), \quad \xi \in [0,1], \qquad \hat{q}_j = \int_0^1 q(\boldsymbol{u}_h(\xi))\, L_j(2\,\xi - 1)\, \mathrm{d}\xi$$

The Legendre coefficients $\hat{q}_j$ are computed using the quadrature induced by the solution points,

$$\hat{q}_j = \sum_{q=0}^{N} q(\boldsymbol{u}_{e,q})\, L_j\,(2\,\xi_q - 1)\, w_q$$

Then the energy contained in the highest modes relative to the total energy of the polynomial is computed as follows,

$$\mathbb{E} = \max\left(\frac{\hat{q}_{N-1}^2}{\sum_{j=0}^{N-1} \hat{q}_j^2}, \frac{\hat{q}_N^2}{\sum_{j=0}^{N} \hat{q}_j^2} \right) \tag{5.12}$$

The N^{th} Legendre coefficient $\hat{q}_N$ of a function which is in the Sobolev space H^2 decays as $O(1/N^2)$ (see Chapter 5, Section 5.4.2 of [37]). We consider smooth functions to be those whose Legendre coefficients $\hat{q}_N$ decay at a rate proportional to $1/N^2$ or faster so that their squares decay proportional to $1/N^4$ [134] or faster. Thus, the following dimensionless threshold for smoothness is proposed in [90]

$$\mathbb{T}(N) = a \cdot 10^{-c(N+1)^4}$$

where parameters $a = \frac{1}{2}$ and $c = 1.8$ are obtained through numerical experiments. To convert the highest mode energy indicator $\mathbb{E}$ and threshold value $\mathbb{T}$ into a value in $[0, 1]$, the logistic function (Figure 5.2) is used

$$\tilde{\alpha}(\mathbb{E}) = \frac{1}{1 + \exp\left(-\frac{s}{\mathbb{T}}(\mathbb{E} - \mathbb{T})\right)}$$

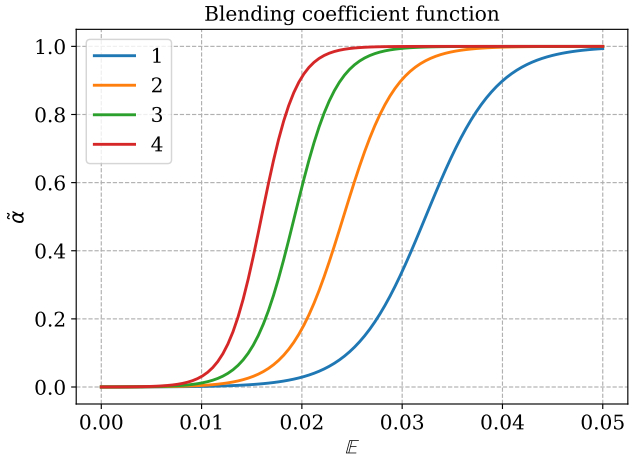


Figure 5.2. Logistic function used to map energy to a smoothness coefficient $\alpha \in [0, 1]$ shown for various solution polynomial degrees N.

The sharpness factor s was chosen to be $s = 9.21024$ so that blending coefficient equals $\alpha = 0.0001$ when highest energy indicator $\mathbb{E} = 0$. In regions where $\tilde{\alpha} = 0$ or $\tilde{\alpha} = 1$, computational cost can be saved by performing only the lower order or higher order scheme respectively. Thus, the values of α are clipped as

$$\alpha_e := \begin{cases} 0, & \text{if } \tilde{\alpha} < \alpha_{\min} \\ \tilde{\alpha}, & \text{if } \alpha_{\min} \leq \tilde{\alpha} \leq 1 - \alpha_{\min} \\ 1, & \text{if } 1 - \alpha_{\min} < \tilde{\alpha} \end{cases}$$

with $\alpha_{\min} = 0.001$. Finally, since shocks can spread to the neighbouring cells as time advances, some smoothening of α is performed as

$$\alpha_e^{\text{final}} = \max_{E \in \mathcal{E}_e} \left\{ \alpha_e, \frac{1}{2}\alpha_E \right\} \tag{5.13}$$

where $\mathcal{E}_e$ denotes the set of elements sharing a face with Ω_e.

5.3.3. First order blending

The lower order scheme is taken to be a first order finite volume scheme, for which the subcell fluxes in (5.8) are given by

$$\boldsymbol{f}^e_{p+\frac{1}{2}} = \boldsymbol{f}(\boldsymbol{u}_{e,p}, \boldsymbol{u}_{e,p+1})$$

At the interfaces that are shared with FR elements, we define the lower order flux used in computing inter-element flux (Section 5.5) as

$$\boldsymbol{f}_{e+\frac{1}{2}} = \boldsymbol{f}(\boldsymbol{u}_{e,N}, \boldsymbol{u}_{e+1,0}) \tag{5.14}$$

In this chapter, the numerical flux $\boldsymbol{f}(\cdot,\cdot)$ is taken to be Rusanov's flux [152], which is the same flux used by the higher order scheme at the element interfaces.

5.4. Higher order blending

The MUSCL-Hancock scheme is a single-stage and second order accurate scheme, originally introduced in [185], and proven to be robust under appropriate slope restrictions [26]. We can expect better accuracy by blending the LWFR scheme with the MUSCL-Hancock scheme. Following the slope correction procedure of Berthon [26], the MUSCL-Hancock scheme can mimic the admissible set preservation of the solutions of conservation laws (5.4). The extension of Berthon's work to non-cell centered grids (G.3) which arise in the blending scheme is given in Theorem 5.4 whose proof is given in Appendix G. In this section, we give algorithmic details of the 1-D procedure and details of the 2-D procedure can be found in Appendix G.6.

Essentially, the MUSCL-Hancock scheme provides a high order estimate of the subcell fluxes $\boldsymbol{f}^e_{p+\frac{1}{2}}$ used in the low order scheme (5.8) and we now explain the procedure for estimating these fluxes. The procedure below can be used for any choice of solution points. However, in this thesis, all results with MUSCL-Hancock scheme have been generated using Gauss-Legendre solution points. To simplify the notation, let us suppress the element index e and set

$$\boldsymbol{u}_{-2} = \boldsymbol{u}^{e-1}_{N-1}, \quad \boldsymbol{u}_{-1} = \boldsymbol{u}^{e-1}_{N}, \quad \{\boldsymbol{u}_p = \boldsymbol{u}_{e,p}, \quad 0 \le p \le N\}, \quad \boldsymbol{u}_{N+1} = \boldsymbol{u}^{e+1}_{0}, \quad \boldsymbol{u}_{N+2} = \boldsymbol{u}^{e+1}_{1}$$

Using the mid-point rule in time to integrate the conservation law (3.1) over the space-time element $[x_{p-\frac{1}{2}}, x_{p+\frac{1}{2}}] \times [t^n, t^{n+1}]$, we get

$$\boldsymbol{u}_p^{n+1} = \boldsymbol{u}_p^n - \frac{\Delta t}{\Delta x_p}\left(\boldsymbol{f}^{n+\frac{1}{2}}_{p+\frac{1}{2}} - \boldsymbol{f}^{n+\frac{1}{2}}_{p-\frac{1}{2}}\right) \tag{5.15}$$

where

$$\boldsymbol{f}^{n+\frac{1}{2}}_{p+\frac{1}{2}} = \boldsymbol{f}(\boldsymbol{u}_{p-1}^{n+\frac{1}{2},+}, \boldsymbol{u}_p^{n+\frac{1}{2},-}) \tag{5.16}$$

is obtained from a numerical flux function. The numerical flux in (5.16) is taken to be Rusanov's flux [152] in this work, but any admissibility preserving finite volume flux (Definition 3.1) can be used. The $\boldsymbol{u}_p^{n+\frac{1}{2},\pm}$ denote the approximations of solutions in subcell p at right, left faces respectively, evolved to time level $n+\frac{1}{2}$. Aiming to first approximate the solution at t^n on the faces, we create a linear approximation of the solution in each subcell as

$$\boldsymbol{r}_p^n(x) = \boldsymbol{u}_p^n + (x - x_p)\,\boldsymbol{\delta}_p, \qquad \boldsymbol{\delta}_p = \text{minmod}(\beta\,\Delta_+\,\boldsymbol{u}_p, \Delta_c\,\boldsymbol{u}_p, \beta\,\Delta_-\,\boldsymbol{u}_p) \tag{5.17}$$

where, for $h_1 = x_p - x_{p-1}$, $h_2 = x_{p+1} - x_p$,[5.1]

$$\Delta_+ \boldsymbol{u}_p = \frac{\boldsymbol{u}_{p+1}^n - \boldsymbol{u}_p^n}{h_2}, \qquad \Delta_- \boldsymbol{u}_p = \frac{\boldsymbol{u}_p^n - \boldsymbol{u}_{p-1}^n}{h_1}$$
$$\Delta_c \boldsymbol{u}_p = -\frac{h_2}{h_1 (h_1 + h_2)} \boldsymbol{u}_{p-1}^n + \frac{h_2 - h_1}{h_1 h_2} \boldsymbol{u}_p^n + \frac{h_1}{h_2 (h_1 + h_2)} \boldsymbol{u}_{p+1}^n$$

The $\Delta_\pm \boldsymbol{u}_p$ are forward and backward approximations of slope respectively, and $\Delta_c \boldsymbol{u}_p$ is the second order approximation of the slope. The value β is chosen to lie between 1 and 2; for $\beta = 1$, we reduce to the minmod limiter and $\beta = 2$ corresponds to the MC (monotonized central-difference) limiter of van Leer [184]. A higher value of β tips the slope closer to the second order approximation, gaining accuracy but also increasing the risk of spurious oscillations. For all results in this chapter, the choice of $\beta = 2 - \alpha_e$ is made. Thus, β will be close to 2 in regions where smoothness indicator only detects mild irregularities in the solution, while it will be near 1 in regions with strong discontinuities. With the linear reconstructions, we can define

$$\boldsymbol{u}_p^{n,-} = \boldsymbol{r}_p^n(x_{p-\frac{1}{2}}) = \boldsymbol{u}_p^n + \boldsymbol{\delta}_p (x_{p-\frac{1}{2}} - x_p), \qquad \boldsymbol{u}_p^{n,+} = \boldsymbol{r}_p^n(x_{p+\frac{1}{2}}) = \boldsymbol{u}_p^n + \boldsymbol{\delta}_p (x_{p+\frac{1}{2}} - x_p) \tag{5.18}$$

Using the conservation law, we approximate the temporal derivatives as

$$\partial_t \boldsymbol{u}_p^n := -\frac{\boldsymbol{f}(\boldsymbol{u}_p^{n,+}) - \boldsymbol{f}(\boldsymbol{u}_p^{n,-})}{x_{p+\frac{1}{2}} - x_{p-\frac{1}{2}}}$$

and finally use Taylor's expansion to evolve the face values in time as

$$\boldsymbol{u}_p^{n+\frac{1}{2},-} = \boldsymbol{u}_p^{n,-} + \frac{\Delta t}{2} \partial_t \boldsymbol{u}_p^n, \qquad \boldsymbol{u}_p^{n+\frac{1}{2},+} = \boldsymbol{u}_p^{n,+} + \frac{\Delta t}{2} \partial_t \boldsymbol{u}_p^n \tag{5.19}$$

At the interfaces shared with the FR elements, the lower order flux used in computing inter-element flux (Section 5.5) is given by $\boldsymbol{f}_{e+\frac{1}{2}} = \boldsymbol{f}_{N+\frac{1}{2}}^{n+\frac{1}{2}}$; the dependence on neighbouring states can be made explicit as

$$\boldsymbol{f}_{e+\frac{1}{2}} = \boldsymbol{f}(\boldsymbol{u}_{N-1}^e, \boldsymbol{u}_N^e, \boldsymbol{u}_0^{e+1}, \boldsymbol{u}_1^{e+1}) \tag{5.20}$$

For admissibility of the lower order method, we rely on the following generalization of Berthon [26], proved in Appendix G.

THEOREM 5.4. *Consider a conservation law of the form (3.1) which preserves the admissible set $\mathcal{U}_{\text{ad}}$ (5.4). Let $\{\boldsymbol{u}_p^n\}_p$ be the approximate solution at time level n and assume that $\boldsymbol{u}_p^n \in \mathcal{U}_{\text{ad}}$ for all p. Consider conservative reconstructions*

$$\boldsymbol{u}_p^{n,+} = \boldsymbol{u}_p^n + (x_{p+\frac{1}{2}} - x_p)\boldsymbol{\delta}_p, \qquad \boldsymbol{u}_p^{n,-} = \boldsymbol{u}_p^n + (x_{p-\frac{1}{2}} - x_p)\boldsymbol{\delta}_p$$

Define $\boldsymbol{u}_p^{*,\pm}$ by

$$\mu_- \boldsymbol{u}_p^{n,-} + \boldsymbol{u}_p^{*,\pm} + \mu_+ \boldsymbol{u}_p^{n,+} = 2\, \boldsymbol{u}_p^{n,\pm} \tag{5.21}$$

5.1. In case of Gauss-Legendre-Lobatto points, the slope $\boldsymbol{\delta}_p = \boldsymbol{0}$ is used in (5.17) whenever $h_1 = 0$ or $h_2 = 0$.

where

$$\mu_- = \frac{x_{p+\frac{1}{2}} - x_p}{x_{p+\frac{1}{2}} - x_{p-\frac{1}{2}}}, \qquad \mu_+ = \frac{x_p - x_{p-\frac{1}{2}}}{x_{p+\frac{1}{2}} - x_{p-\frac{1}{2}}}$$

Assume that the slope $\boldsymbol{\delta}_p$ *is chosen so that*

$$\boldsymbol{u}_p^{*,\pm} \in \mathcal{U}_{\text{ad}} \tag{5.22}$$

Then, assuming that the first order finite volume flux used in (5.16) *is admissibility preserving (Definition 3.1), under appropriate time step restrictions (G.10, G.13, G.19), the updated solution* $\boldsymbol{u}_p^{n+1}$ *defined by the MUSCL-Hancock procedure* (5.15) *is in* $\mathcal{U}_{\text{ad}}$.

5.4.1. Slope limiting in practice

A problem-independent procedure for slope limiting to ensure admissibility preservation is proposed, in contrast to the original procedure for Euler's equations in [26] that was extended to the 10-moment problem in [124]. For the MUSCL-Hancock scheme to be admissibility preserving, the slope $\boldsymbol{\delta}_p$ given by the minmod limiter (5.17) has to be further limited so that $\boldsymbol{u}_p^{*,\pm} = \boldsymbol{u}_p^n + 2\,(x_{p\pm\frac{1}{2}} - x_p)\,\boldsymbol{\delta}_p \in \mathcal{U}_{\text{ad}}$ (5.21). Let $\{P_k, 1 \le 1 \le K\}$ be the admissibility constraints (5.1) for the conservation law (3.1) to be in $\mathcal{U}_{\text{ad}}$. The slope is limited by iterating over the constraints. For each constraint, we can solve an optimization problem to find the largest $\theta_\pm \in [0,1]$ satisfying

$$P_k(\boldsymbol{u}_p^n + 2\,\theta_\pm\,(x_{p\pm\frac{1}{2}} - x_p)\,\boldsymbol{\delta}_p) = P_k(\theta_\pm\,\boldsymbol{u}_p^{*,\pm} + (1-\theta_\pm)\,\boldsymbol{u}_p^n) \ge \epsilon_p, \qquad p = 0, N \tag{5.23}$$

where ϵ_p is a tolerance, taken to be $\frac{1}{10}\,P_k(\boldsymbol{u}_p^n)$ [151]. The optimization problem is usually a polynomial equation in θ, and can be solved for its root. In this work, we use a general iterative solver that is independent of choice of P_k (Appendix F). If P_k is a concave function of the conserved variables, we can follow [19] and use the simpler but possibly sub-optimal approach of defining

$$\theta_\pm = \min\left(\min_{p=0,N} \left| \frac{\epsilon_p - P_k(\boldsymbol{u}_p^n)}{P_k(\boldsymbol{u}_p^{*,\pm}) - P_k(\boldsymbol{u}_p^n)} \right|, 1 \right) \tag{5.24}$$

In either case, by iterating over the admissibility constraints $\{P_k\}$ of the conservation law, the flux slope limiting is performed by the following `for` loop

---

$\boldsymbol{\delta}_p \leftarrow \text{minmod}(\beta\,\Delta_+\,\boldsymbol{u}_p, \Delta_c\,\boldsymbol{u}_p, \beta\,\Delta_-\,\boldsymbol{u}_p)$
$\boldsymbol{u}_p^{*,\pm} \leftarrow \boldsymbol{u}_p^n + 2\,(x_{p\pm\frac{1}{2}} - x_p)\boldsymbol{\delta}_p$
for $k = 1{:}\,K$ **do**
  $\epsilon_k = \frac{1}{10}\,P_k(\boldsymbol{u}_p^n)$
  Find $\theta_\pm$ by solving (5.23) or by using (5.24) if P_k is concave
  $\theta_k \leftarrow \min\{\theta_+, \theta_-\}$
  $\boldsymbol{\delta}_p \leftarrow \theta_k\,\boldsymbol{\delta}_p$
  $\boldsymbol{u}_p^{*,\pm} \leftarrow \boldsymbol{u}_p^n + 2\,(x_{p\pm\frac{1}{2}} - x_p)\,\boldsymbol{\delta}_p$
end for

At the k^{th} iteration, solving the optimization problem (5.23) will satisfy the constraint P_k by definition. On the other hand, if we use (5.24) in case P_k is concave, the $\boldsymbol{u}_p^{*,\pm}$ computed with the corrected slope $\boldsymbol{\delta}_p$ will satisfy

$$P_k(\boldsymbol{u}_p^{*,\pm}) = P_k\left(\theta_k(\boldsymbol{u}_p^{*,\pm})^{\text{prev}} + (1-\theta_k)\,\boldsymbol{u}_p^n\right) \geq \theta_k\, P_k\left((\boldsymbol{u}_p^{*,\pm})^{\text{prev}}\right) + (1-\theta_k)\, P_k(\boldsymbol{u}_p^n) \geq \epsilon_k \tag{5.25}$$

so that the k^{th} admissibility constraint is satisfied; here $(\boldsymbol{u}_p^{*,\pm})^{\text{prev}}$ denotes $\boldsymbol{u}_p^{*,\pm}$ before the k^{th} correction. The choice of $\epsilon_k = \frac{1}{10}\, P_k(\boldsymbol{u}_p^n)$ was made following [151] to allow only a certain deviation below the *safe solution*, imposing a stricter requirement than positivity. Note that this limiting is performed on the slope used for reconstruction in the MUSCL-Hancock scheme, and not on the updated solution. We now use an inductive argument to show that the k^{th} correction will continue to satisfy the previous admissibility constraints. Thus, we assume that constraint P_l is satisfied by $(\boldsymbol{u}_p^{*,\pm})^{\text{prev}}$ for all $l < k$ and we perform k^{th} correction on it to obtain $\boldsymbol{u}_p^{*,\pm}$. In case of concave admissibility constraints,

$$\begin{aligned} P_l(\boldsymbol{u}_p^{*,\pm}) &= P_l(\theta_k\,(\boldsymbol{u}_p^{*,\pm})^{\text{prev}} + (1-\theta_k)\,\boldsymbol{u}_p^n) \\ &\geq \theta_k\, P_l((\boldsymbol{u}_p^{*,\pm})^{\text{prev}}) + (1-\theta_k)\, P_l(\boldsymbol{u}_p^n) \geq \theta_k\,\epsilon_l + (1-\theta_k)\,\epsilon_l = \epsilon_l \end{aligned} \tag{5.26}$$

In case of non-concave P_l, we use (5.2) to obtain $P_l(\boldsymbol{u}_p^{*,\pm}) > \epsilon_l((\boldsymbol{u}_p^{*,\pm})^{\text{prev}}, \boldsymbol{u}_p^n) > 0$ from (5.25). Thus, in both cases, constraints P_l are satisfied for all $l < k$ and the slope $\boldsymbol{\delta}_p$ obtained at the end of K iterations satisfies all admissibility constraints ensuring $\boldsymbol{u}_p^{*,\pm} \in \mathcal{U}_{\text{ad}}$.

5.5. Flux limiter for admissibility preservation

The first step in obtaining an admissibility preserving blending scheme is to ensure that the lower order scheme preserves the admissible set $\mathcal{U}_{\text{ad}}$. This is always true if all the fluxes in the lower order method are computed with a finite volume method that is proven to be admissibility preserving. However, the LWFR scheme uses a time average numerical flux and maintaining conservation requires that we use the same numerical flux at the element interfaces for both lower and higher order schemes (see Remark 5.3). To maintain accuracy and admissibility, we have to carefully choose a blended numerical flux $\boldsymbol{F}_{e+\frac{1}{2}}$ as in (5.11) but this choice may not ensure admissibility, and further limitation is required. Our proposed procedure for choosing the blended numerical flux will give us an admissibility preserving lower order scheme. After this step, there are two possibilities for obtaining admissibility of the blending scheme. We could follow the procedure of [151] to *a posteriori* modify the blending coefficient α to obtain admissibility relying directly on the admissibility of the lower order scheme. The other option which we take in this thesis is to note that, as a result of using the same numerical flux in both high and low order schemes, element means of both schemes are the same (Theorem 5.5). A consequence of this is that our scheme now preserves admissibility of element means and thus we can use the scaling limiter of [205]. The latter approach of correcting element means to obtain a positivity preserving Lax-Wendroff scheme has been used in [128], where the numerical flux is corrected to directly make element means admissible. In comparison to [128], our procedure for ensuring admissibility of element means requires less storage and loops.

The theoretical basis for flux limiting can be summarised in the following Theorem 5.5.

THEOREM 5.5. *Consider the LWFR blending scheme (5.6) where low and high order schemes use the same numerical flux $\boldsymbol{F}_{e+\frac{1}{2}}$ at every element interface. Then the following can be said about admissibility preserving in means property (Definition 5.2) of the scheme:*

1. *element means of both low and high order schemes are same and thus the blended scheme (5.6) is admissibility preserving in means if and only if the lower order scheme is admissibility preserving in means;*

2. *if the finite volume method using the lower order flux $\boldsymbol{f}_{e+\frac{1}{2}}$ as the interface flux is admissibility preserving, such as the first-order finite volume method or the MUSCL-Hancock scheme with CFL restrictions and slope correction from Theorem 5.4, and the blended numerical flux $\boldsymbol{F}_{e+\frac{1}{2}}$ is chosen to preserve the admissibility of lower-order updates at solution points adjacent to the interfaces, then the blending scheme (5.6) will preserve admissibility in means.*

Proof. By (5.3, 5.9), element means are the same for both low and high order schemes. Thus, admissibility in means of one implies the same for other, proving the first claim. For the second claim, note that our assumptions imply $\boldsymbol{u}_{e,p}^{L,n+1}$ given by (5.8) is in $\mathcal{U}_{\text{ad}}$ for $0 \le p \le N$ implying admissibility in means property of the lower order scheme by (5.9) and thus admissibility in means for the blended scheme. □

We now explain the procedure of ensuring that the update obtained by the lower order scheme will be admissible. The lower order scheme is computed with a first order finite volume method or MUSCL-Hancock with slope correction from Theorem 5.4 so that admissibility is already ensured for inner solution points; i.e., we already have

$$\boldsymbol{u}_{e,p}^{L,n+1} \in \mathcal{U}_{\text{ad}}, \quad 1 \le p \le N-1$$

The remaining admissibility constraints for the first ($p=0$) and last solution points ($p=N$) will be satisfied by appropriately choosing the inter-element flux $\boldsymbol{F}_{e+\frac{1}{2}}$. The first step is to choose a candidate for $\boldsymbol{F}_{e+\frac{1}{2}}$ which is heuristically expected to give reasonable control on spurious oscillations, i.e.,

$$\boldsymbol{F}_{e+\frac{1}{2}} = (1-\alpha_{e+\frac{1}{2}})\,\boldsymbol{F}^{\text{LW}}_{e+\frac{1}{2}} + \alpha_{e+\frac{1}{2}}\,\boldsymbol{f}_{e+\frac{1}{2}}, \qquad \alpha_{e+\frac{1}{2}} = \frac{\alpha_e + \alpha_{e+1}}{2}$$

where $\boldsymbol{f}_{e+\frac{1}{2}}$ is the lower order flux at the face $e+\frac{1}{2}$ shared between FR elements and subcells (5.14, 5.20), and α_e is the blending coefficient (5.6) based on element-wise smoothness indicator (Section 5.3.2).

The next step is to correct $\boldsymbol{F}_{e+\frac{1}{2}}$ to enforce the admissibility constraints. The guiding principle of our approach is to perform the correction within the face loops, minimizing storage requirements and additional memory reads. The lower order updates in subcells neighbouring the $e+\frac{1}{2}$ face with the candidate flux are

$$\begin{aligned} \hat{\boldsymbol{u}}_0^{n+1} &= \boldsymbol{u}_{e+1,0}^n - \frac{\Delta t}{w_0\,\Delta x_{e+1}}\,(\boldsymbol{f}_{\frac{1}{2}}^e - \boldsymbol{F}_{e+\frac{1}{2}}) \\ \hat{\boldsymbol{u}}_N^{n+1} &= \boldsymbol{u}_{e,N}^n - \frac{\Delta t}{w_N\,\Delta x_e}\,(\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{f}_{N-\frac{1}{2}}^e) \end{aligned} \tag{5.27}$$

To correct the interface flux, we will again use the fact that first order finite volume method and MUSCL-Hancock with slope correction from Theorem 5.4 preserve admissibility, i.e.,

$$\begin{aligned} \hat{\boldsymbol{u}}_0^{\text{low},n+1} &= \boldsymbol{u}_{e+1,0}^n - \frac{\Delta t}{w_0\,\Delta x_{e+1}}\,(\boldsymbol{f}_{\frac{1}{2}}^e - \boldsymbol{f}_{e+\frac{1}{2}}) \in \mathcal{U}_{\text{ad}} \\ \hat{\boldsymbol{u}}_N^{\text{low},n+1} &= \boldsymbol{u}_{e,N}^n - \frac{\Delta t}{w_N\,\Delta x_e}\,(\boldsymbol{f}_{e+\frac{1}{2}} - \boldsymbol{f}_{N-\frac{1}{2}}^e) \in \mathcal{U}_{\text{ad}} \end{aligned}$$

Let $\{P_k, 1 \le 1 \le K\}$ be the admissibility constraints (5.1) of the conservation law (3.1). For each constraint, we can solve an optimization problem to find the largest $\theta \in [0,1]$ satisfying

$$P_k(\theta\,\hat{\boldsymbol{u}}_p^{n+1} + (1-\theta)\,\hat{\boldsymbol{u}}_p^{\text{low},n+1}) > \epsilon_p, \qquad p = 0, N \tag{5.28}$$

where ϵ_p is a tolerance, taken to be $\frac{1}{10}\,P_k(\hat{\boldsymbol{u}}_p^{\text{low},n+1})$ [151]. The optimization problem is usually a polynomial equation in θ, and can be solved for its root. In this work, we use a general iterative solver that is independent of choice of P_k (Appendix F). If P_k is a concave function of the conserved variables, we can follow [19] and use the simpler but possibly sub-optimal approach of defining

$$\theta = \min\left(\min_{p=0,N}\left|\frac{\epsilon_p - P_k(\hat{\boldsymbol{u}}_p^{\text{low},n+1})}{P_k(\hat{\boldsymbol{u}}_p^{n+1}) - P_k(\hat{\boldsymbol{u}}_p^{\text{low},n+1})}\right|, 1\right) \tag{5.29}$$

In either case, by iterating over the admissibility constraints $\{P_k\}$ of the conservation law, the flux $\boldsymbol{F}_{e+\frac{1}{2}}^{\text{LW}}$ can be corrected using the iterative limiting procedure in Algorithm 5.1.

Algorithm 5.1

Flux limiter

$\boldsymbol{F}_{e+\frac{1}{2}} \leftarrow (1-\alpha_{e+\frac{1}{2}})\,\boldsymbol{F}_{e+\frac{1}{2}}^{\text{LW}} + \alpha_{e+\frac{1}{2}}\,\boldsymbol{f}_{e+\frac{1}{2}}$ ▷ Initial guess

for $k = 1{:}\,K$ **do**

$\epsilon_0, \epsilon_N \leftarrow \frac{1}{10}\,P_k(\hat{\boldsymbol{u}}_0^{\text{low},n+1}), \frac{1}{10}\,P_k(\hat{\boldsymbol{u}}_N^{\text{low},n+1})$

Find θ by solving (5.28) or by using (5.29) if P_k is concave

$\boldsymbol{F}_{e+\frac{1}{2}} \leftarrow \theta\,\boldsymbol{F}_{e+\frac{1}{2}} + (1-\theta)\,\boldsymbol{f}_{e+\frac{1}{2}}$

$\hat{\boldsymbol{u}}_0^{n+1} \leftarrow \boldsymbol{u}_{e+1,0}^n - \frac{\Delta t}{w_0\,\Delta x_{e+1}}\,(\boldsymbol{f}_{\frac{1}{2}}^{e+1} - \boldsymbol{F}_{e+\frac{1}{2}})$

$\hat{\boldsymbol{u}}_N^{n+1} \leftarrow \boldsymbol{u}_{e,N}^n - \frac{\Delta t}{w_N\,\Delta x_e}\,(\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{f}_{N-\frac{1}{2}}^e)$

end for

In case of solving an optimization problem (5.28), the admissibility constraint P_k will be satisfied after the k^{th} iteration of Algorithm 5.1 by definition of the optimization problem. In the case of concave P_k, if (5.29) is used, after the k^{th} iteration, the updates computed using flux $\boldsymbol{F}_{e+\frac{1}{2}}$ will satisfy for $p = 0, N$

$$\begin{aligned} P_k(\hat{\boldsymbol{u}}_p^{n+1}) &= P_k(\theta(\hat{\boldsymbol{u}}_p^{n+1})^{\text{prev}} + (1-\theta)\hat{\boldsymbol{u}}_p^{\text{low},n+1}) \\ &\geq \theta\, P_k((\hat{\boldsymbol{u}}_p^{n+1})^{\text{prev}}) + (1-\theta)\, P_k(\hat{\boldsymbol{u}}_p^{\text{low},n+1}) \geq \epsilon_p \end{aligned}$$

satisfying the k^{th} admissibility constraint; here $(\hat{\boldsymbol{u}}_p^{n+1})^{\text{prev}}$ denotes $\hat{\boldsymbol{u}}_p^{n+1}$ before the k^{th} correction and the choice of $\epsilon_p = \frac{1}{10} P_k(\hat{\boldsymbol{u}}_p^{\text{low},n+1})$ is made following [151]. After the K iterations, all admissibility constraints will be satisfied and the resulting flux $\boldsymbol{F}_{e+\frac{1}{2}}$ will be used as the interface flux keeping the lower order updates and thus the element means admissible. Thus, by Theorem 5.5, the choice of blended numerical flux gives us admissibility preservation in means. We now use the scaling limiter of [205] to obtain an admissibility preserving scheme as defined in Definition 5.1, an overview of the complete scheme can be found in Algorithm 5.2. The above procedure is for 1-D conservation laws; the extension to 2-D is performed by breaking the update into convex combinations of 1-D updates and adding additional time step restrictions; the details are given in Appendix H.

5.6. Some implementation details

In Section 5.5, the procedure for computing the blended numerical flux to achieve admissibility preservation in means for LWFR (Definition 5.2) was presented. In this section, we present an overview of the complete LWFR blended scheme which employs the computed blended flux and the scaling limiter of [205] to obtain an admissibility preserving scheme (Definition 5.1) in Algorithm 5.2.

The residual in (5.6) is computed by performing an element loop and a face loop, incorporating blending within each of these loops. Within the element loop, we compute lower order fluxes on the subcell faces not shared by the FR elements. The fluxes for the faces shared by FR elements are computed within the face loop, and subsequently blended with the LW flux. This approach enables direct computation and use of each quantity, without the need for intermediate storage. However, to compute (5.27), admissibility preservation requires storage of lower order fluxes $\boldsymbol{f}^e_{\frac{1}{2}}$ and $\boldsymbol{f}^e_{N-\frac{1}{2}}$, which are computed during the element loop.

In Algorithm 5.2, we give a high level overview of the LWFR with blending scheme. In the implementation, some operations are avoided by computing only high or low order residuals in the cases where $\alpha_e = 0$ or $\alpha_e = 1$, but we did not include this optimization in Algorithm 5.2 to maintain simplicity in our explanation. The correction procedure of numerical flux for admissibility preservation (Section 5.5) is performed

within the interface iteration. The contribution of numerical flux to the residual is added in a different element loop to avoid race conditions in a multi-threaded loop; only one loop would be needed in a serial code. After the solution update in Algorithm 5.2, the blended flux will ensure that our purely low order update and the element means are admissible. However, the updates at solution points need not be admissible at this stage and must be corrected. The correction at solution points could now be performed as an *a posteriori* modification of the blending coefficients [151] or using the scaling limiter of [205]; we use the scaling limiter for all results in this work.

Algorithm 5.2

High-level overview of the Lax-Wendroff with blending scheme

$t = 0$;
while $t < T$ **do**
  Compute $\{\alpha_e\}$ (Section 5.3.2)

  ▷ Assemble element residual
  for e in `eachelement(mesh)` **do**
    Add LW element residual to rhs scaled with $1 - \alpha_e$
    Add FV subcell residual to rhs scaled with α_e
    Store $\boldsymbol{f}^e_{1/2}, \boldsymbol{f}^e_{N-1/2}$ (5.27)
  end for

  ▷ Compute numerical fluxes at all interfaces
  for $e + \frac{1}{2}$ in `eachinterface(mesh)` **do**
    Compute $\boldsymbol{F}^{\mathrm{LW}}_{e+\frac{1}{2}}$, $\boldsymbol{f}_{e+\frac{1}{2}}$ and blend them into $\boldsymbol{F}_{e+\frac{1}{2}}$ (Section 5.5)
  end for

  ▷ Assemble face residual
  for e in `eachelement(mesh)` **do**
    Add contribution of $\boldsymbol{F}_{e\pm\frac{1}{2}}$ to high, low order residual scaled with $1 - \alpha_e, \alpha_e$
  end for
  Update solution
  Apply positivity correction at solution points using [205] or [151]
  $t = t + \Delta t$;
end while

5.7. Numerical results

We perform various tests to show the robustness and accuracy of the proposed blending scheme. The LWFR results are always obtained with D2 dissipation and EA flux [18] with Rusanov's numerical flux using Gauss-Legendre solutions point and Radau cor-

rection functions. All numerical simulations were run with the first order blending (Section 5.3.3), MUSCL-Hancock blending (Section 5.4) and TVB limiter with fine-tuned parameter M plotted with legends FO, MH and TVB-M. We also made comparison with the results of first order blending scheme using Gauss-Legendre-Lobatto points of [90] implemented in `Trixi.jl` [141, 158]. Our code is publicly available at [17], and the scripts for generated results in this chapter are available at [16]. The user only needs to install `Julia` [29] and the remaining dependencies are automatically handled by `Julia` environments and its package manager.

5.7.1. 1-D Euler equations

As an example of system of non-linear hyperbolic equations, consider the one-dimensional Euler equations of gas dynamics given by (4.16). Unless otherwise specified, the gas constant γ will be taken as 1.4 which is the value for air. The time step size for polynomial degree N is computed as in (4.18). Most of the numerical results presented in this chapter use degree $N = 4$ for which $\mathrm{CFL}(N) = 0.069$. The admissibility preservation of subcell based MUSCL-Hancock imposes a time restriction (Theorem 5.4) which depends on several quantities other than element means, including some evolved quantities, see equations (G.10, G.13, G.19). The CFL coefficient of MUSCL-Hancock admissibility is also smaller than $\mathrm{CFL}(N)$ in (4.18), see Remark G.7. However, we have found the time step given by (4.18) with $C_{\mathrm{CFL}} = 0.98$ to be sufficient for admissibility preservation in all the simulations we have performed. Thus, we do not explicitly impose the CFL restrictions in Theorem 5.4 as they are more severe and expensive to compute. If the admissibility is violated in any cell, then the time update can be repeated in those cells by lowering the time step by some fraction.

5.7.1.1. Shu-Osher problem

This problem was described in Section 4.8.4. Due to presence of both spurious oscillations and smooth extremums, this becomes a good test for testing robustness and

accuracy of limiters. We discretize the spatial domain with 400 cells using polynomial degree $N = 4$ and compare blending schemes and TVB limiter with parameter $M = 300$ [137]. The density component of the approximate solutions computed for the compared limiters are plotted against a reference solution obtained using a very fine mesh, and are given in Figures 5.3, 5.4. The three limiters show similar performance in Figure 5.3 on the large scale. The enlarged plots in Figure 5.4 show that the MUSCL-Hancock blending scheme is able to capture smooth extrema better than the first order blending and the TVB scheme.

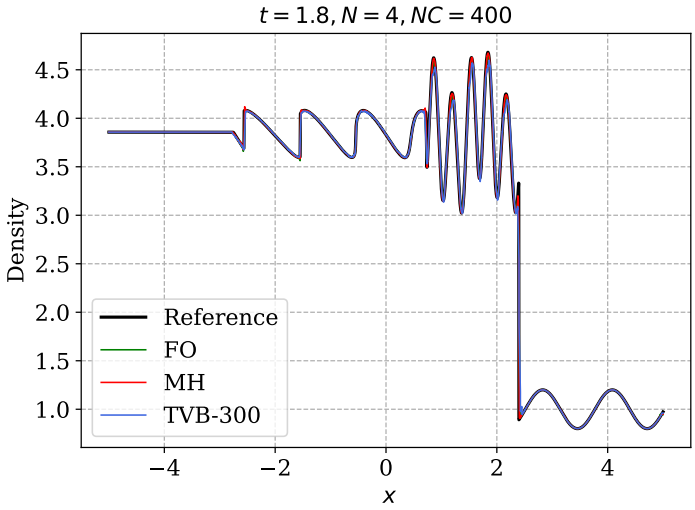


Figure 5.3. Shu-Osher problem, density plot of numerical solution with degree $N = 4$ using first order (FO) and MUSCL-Hancock (MH) blending schemes, and TVB limited scheme (TVB-300) with parameter $M = 300$.

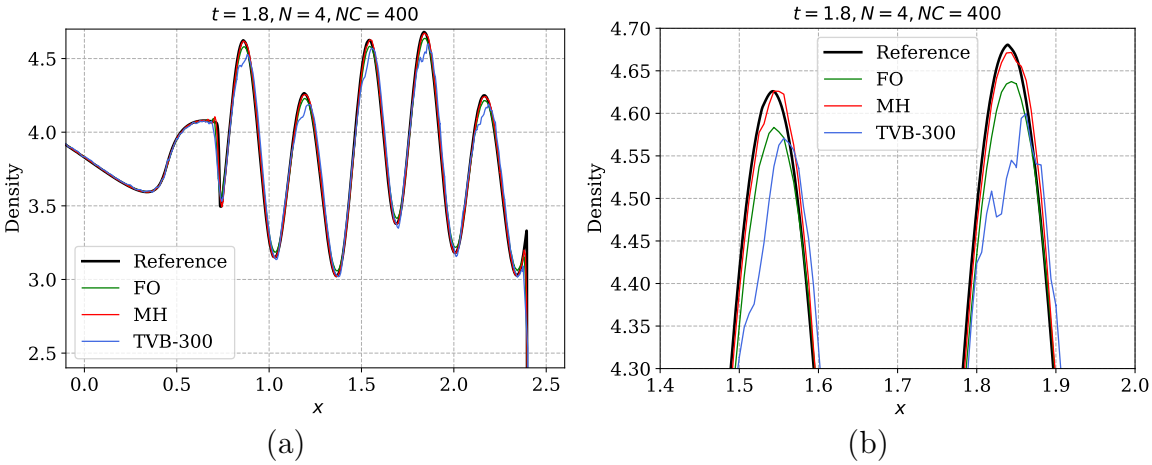


Figure 5.4. Shu-Osher problem, density plot of numerical solution with degree $N = 4$ using first order (FO) and MUSCL-Hancock (MH) blending schemes, and TVB limited scheme (TVB-300) with parameter $M = 300$ at time $t = 1.8$ on a mesh of 400 cells.. (a) Zoomed near smooth extrema, (b) Zoomed to only show two extrema.

5.7.1.2. Blast wave

This test case was described in Section 4.8.5. The solution consists of reflection of shocks and expansion waves off the boundary wall and several wave interactions inside the domain. The numerical solutions are inadmissible if the positivity correction is not

applied. With a grid of 400 cells using polynomial degree $N=4$, we run the simulation till the time $t=0.038$ where a high density peak profile is produced. As in the previous test, we compare first order (FO) and MUSCL-Hancock (MH) blending schemes, and TVB limiter with parameter $M=300$ [137] (TVB-300). We compare the performance of limiters in Figure 5.5 where the approximated density and pressure profiles are compared with a reference solution computed using a very fine mesh. Looking at the peak amplitude and contact discontinuity of the density profile which is also shown in the zoomed inset, it is clear that MUSCL-Hancock blending scheme gives the best resolution, especially when compared with the TVB limiter.

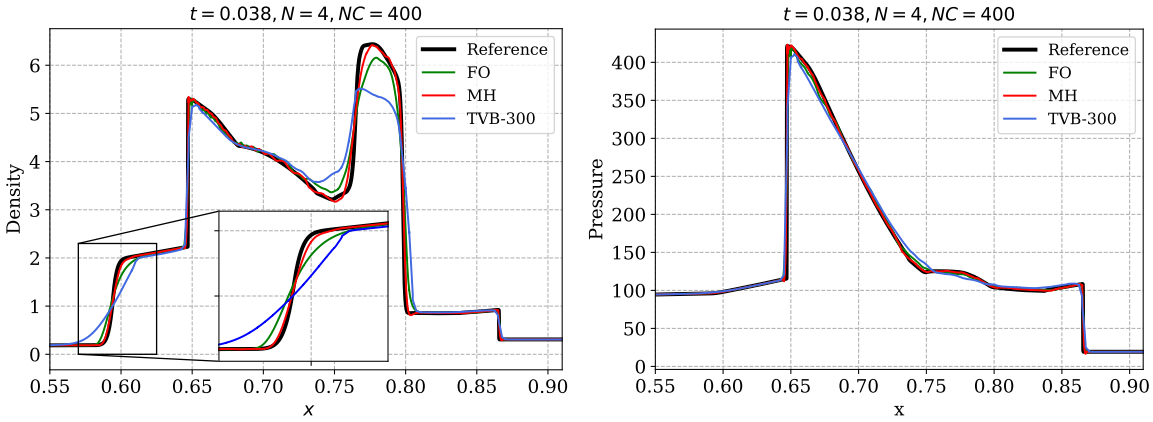


Figure 5.5. Blast wave problem, numerical solution with degree $N=4$ using first order (FO) and MUSCL-Hancock (MH) blending schemes, and TVB limited scheme (TVB-300) with parameter $M=300$. (a) Density, (b) pressure profiles are shown at time $t=0.038$ on a mesh of 400 cells.

5.7.1.3. Sedov's blast wave

To demonstrate the admissibility preserving property of our scheme, we simulate Sedov's blast wave [160]; the test describes the explosion of a point-like source of energy in a gas. The explosion generates a spherical shock wave that propagates outwards, compressing the gas and reaching extreme temperatures and pressures. The problem can be formulated in one dimension as a special case, where the explosion occurs at $x=0$ and the gas is confined to the interval $[-1,1]$ by solid walls. For the simulation, on a grid of 201 cells with solid wall boundary conditions, we use the following initial data [207],

$$\rho=1, \qquad v=0, \qquad E(x)=\begin{cases} \frac{3.2\times 10^6}{\Delta x}, & |x|\le\frac{\Delta x}{2} \\ 10^{-12}, & \text{otherwise} \end{cases}$$

where Δx is the element width. This is a difficult test for positivity preservation because of the high pressure ratios. Nonphysical solutions are obtained if the proposed admissibility preservation corrections are not applied. The density and pressure profiles at $t=0.001$ obtained using blending schemes are shown in Figure 5.6. Results of

TVD limiter are not shown as it fails to preserve positivity in this test because the admissibility correction of Lax-Wendroff scheme depends on the blended numerical flux (Section 5.5).

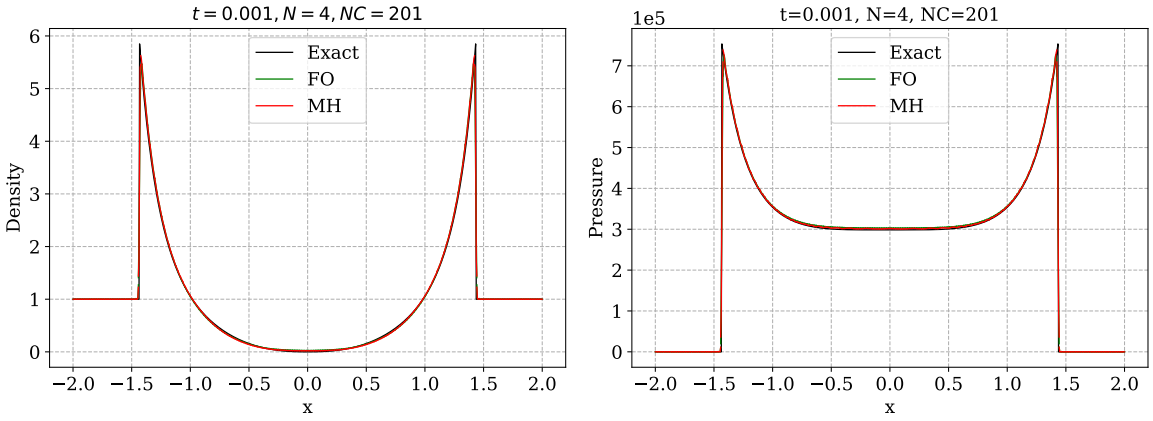


Figure 5.6. Sedov's blast wave problem, numerical solution with degree $N = 4$ using first order (FO) and MUSCL-Hancock blending schemes. (a) Density and (b) pressure profiles of numerical solutions are plotted at time $t = 0.001$ on a mesh of 201 cells.

5.7.1.4. Riemann problems

We test two extreme Riemann problems from [205] to demonstrate admissibility preservation of our scheme. The first is a Riemann problem with no shocks and two rarefactions, which move away from each other leading to a near vacuum state in the exact solution. The low densities make it a challenging test, as the oscillations can easily cause negative density values. As in the previous test, results of TVD limiter are not shown as it fails to preserve admissibility. We run the simulation on the domain $[-1, 1]$ with initial data

$$(\rho, v, p) = \begin{cases} (7.0, -1.0, 0.2), & -1 \leq x \leq 0 \\ (7.0, 1.0, 0.2), & \text{otherwise} \end{cases}$$

The results obtained using blending schemes are shown in Figure 5.7 on a mesh of 200 cells with transmissive boundary conditions at time $t = 0.6$.

The second test is a 1D Leblanc shock tube problem with initial data

$$(\rho, v, p) = \begin{cases} (2, 0, 10^9), & -1 \leq x \leq 0 \\ (0.001, 0, 1), & \text{otherwise} \end{cases}$$

The solution has extremely high density and pressure ratios across the shock and the numerical solutions give negative pressure if the proposed admissibility preservation techniques are not applied. The log-scaled results obtained using blending schemes are shown in Figure 5.8 at time $t = 0.001$ on a mesh of 800 cells with transmissive boundary conditions.

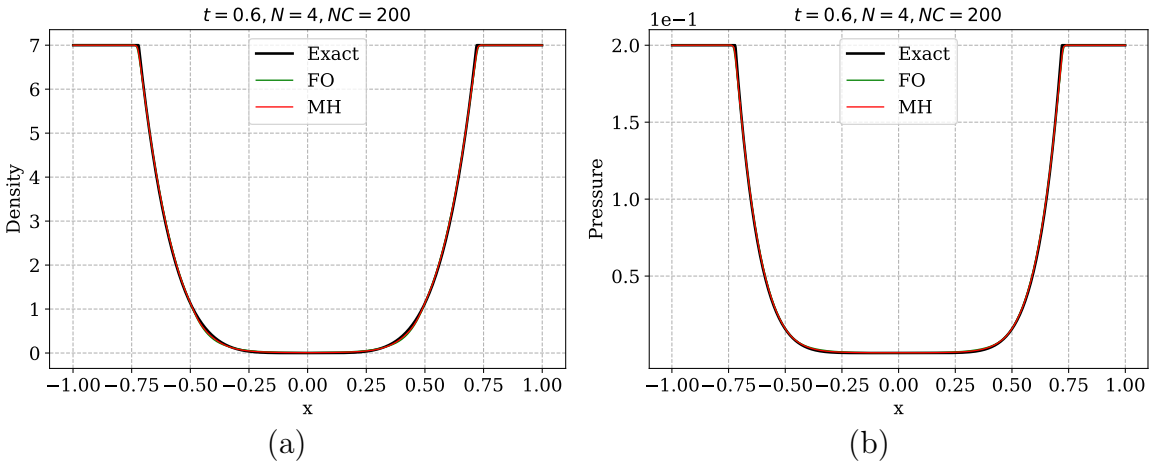


Figure 5.7. Double rarefaction problem, numerical solution with degree $N=4$ using first order (FO) and MUSCL-Hancock (MH) blending. (a) Density and (b) pressure profiles of numerical solutions are plotted at $t=0.6$ on a mesh of 200 cells.

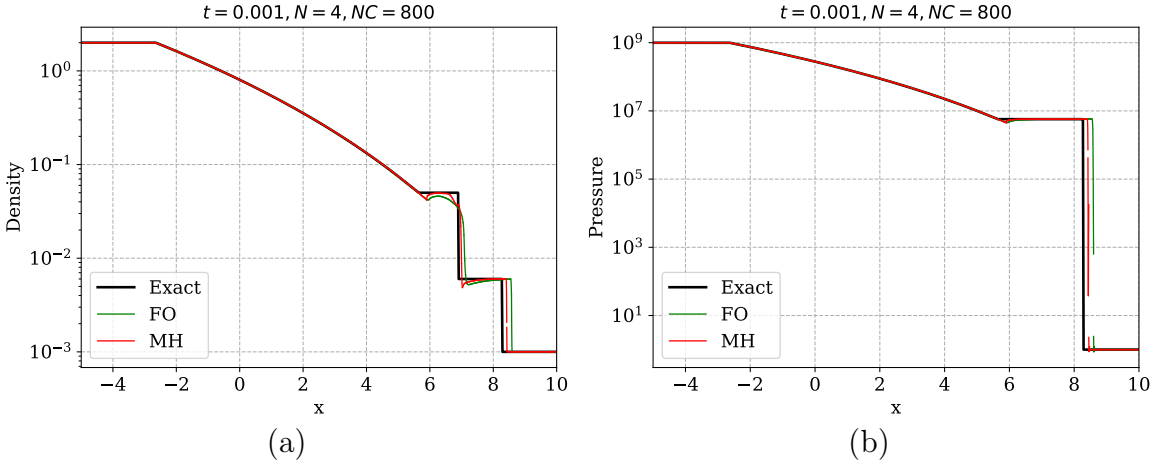
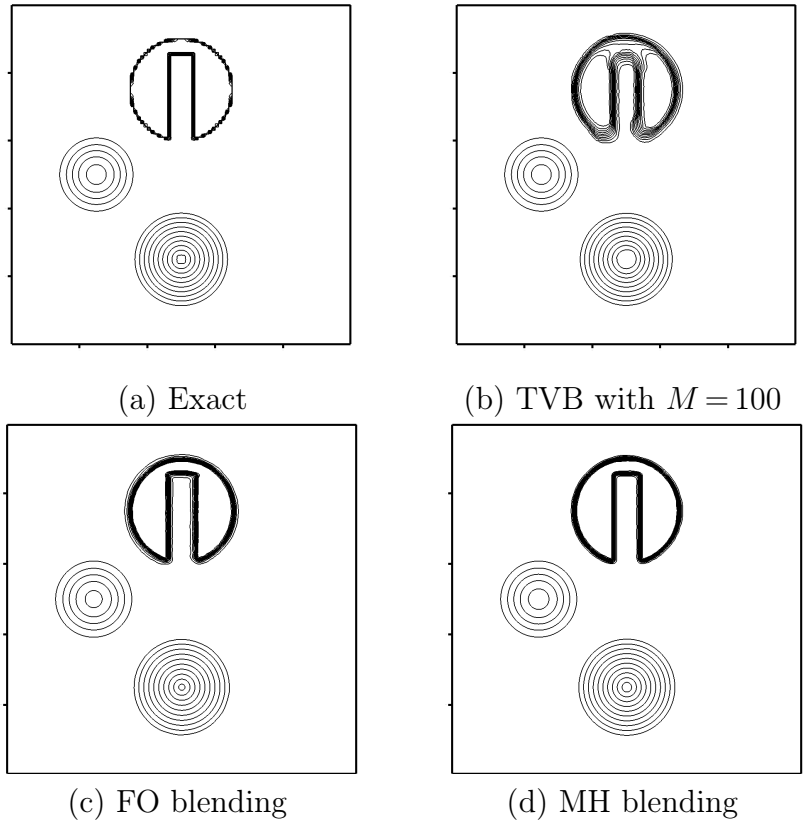


Figure 5.8. Leblanc's test, numerical solution with polynomial degree $N=4$ using first order (FO) and MUSCL-Hancock (MH) blending. (a) Density and (b) pressure profiles of numerical solutions with log scales are plotted at $t=0.001$ on a mesh of 800 cells.

5.8. 2-D ADVECTION EQUATION

The description of this test is provided in Section 4.10.2. The numerical solution is computed at $t=2\pi$ and shown in Figure 5.9a after one time period, comparing different limiters Figure 5.9b-c. To be specific, Figure 5.9 compares contour plots of polynomial solutions obtained using the LWFR method of degree $N=4$ with TVB limiter using a fine-tuned parameter $M=100$, and with blending limiter using first order and MUSCL-Hancock reconstructions, after one time period. The blending limiter with MUSCL-Hancock reconstruction is shown to produce more accurate solutions among the three profiles especially when compared to the TVB limiter, as the TVB limiter results in greater smearing of the profile. The sharp features of slotted disc profile show the most

notable improvement.



(a) Exact (b) TVB with $M = 100$

(c) FO blending (d) MH blending

Figure 5.9. Rotation of a composite signal with velocity $\boldsymbol{a} = (\frac{1}{2} - y, x - \frac{1}{2})$, numerical solution with polynomial degree $N = 4$ on a mesh of 100^2 elements.

5.9. 2-D Euler equations

We consider the two-dimensional Euler equations of gas dynamics given by (2.13). Unless otherwise specified, the adiabatic constant γ will be taken as 1.4 in the numerical tests, which is the typical value for air. The time step size for polynomial degree N is computed as in (4.30). Most of the numerical results presented in this chapter use degree $N = 4$ for which $\mathrm{CFL}(N) = 0.069$ (4.30). As in the 1-D case, (4.30) will not guarantee that the time step restriction for admissibility of MUSCL-Hancock scheme on the subcells is satisfied. However, we have found all tests to work with (4.30) using $C_{\mathrm{CFL}} = 0.98$ and the results are shown with that safety factor unless otherwise specified.

For verification of numerical results and to demonstrate the accuracy gain of our proposed Lax-Wendroff blending scheme with MUSCL-Hancock reconstruction using Gauss-Legendre points, we will compare our results with the first order blending scheme using Gauss-Legendre-Lobatto (GLL) points of [90] available in `Trixi.jl` [141]. Both solvers use the same time step sizes in all results. We have also performed experiments using LWFR with first order blending scheme and Gauss-Legendre (GL) points,

and observed lower accuracy than the MUSCL-Hancock blending scheme, but higher accuracy than the first order blending scheme implementation of `Trixi.jl` using GLL points. These results are expected since GL points and quadrature are more accurate than GLL points, and MUSCL-Hancock is also more accurate than first order finite volume method. However, to save space, we have not presented the results of LWFR with first order blending.

5.9.1. Isentropic vortex convergence test

The description of this test containing a smooth solution with an analytic solution has been given in Section 4.11.1. We run the computations up to a time $t = T$ when the vortex has crossed the domain once in the diagonal direction. Figure 4.38a compares the L^2 error of density sampled at $N + 3$ equispaced points against grid resolution when using the blending limiter. It can be seen that the limiter does not activate for adequately high resolution, yielding the same optimal convergence rates as those achieved without the limiter, as shown in Figure 4.38b.

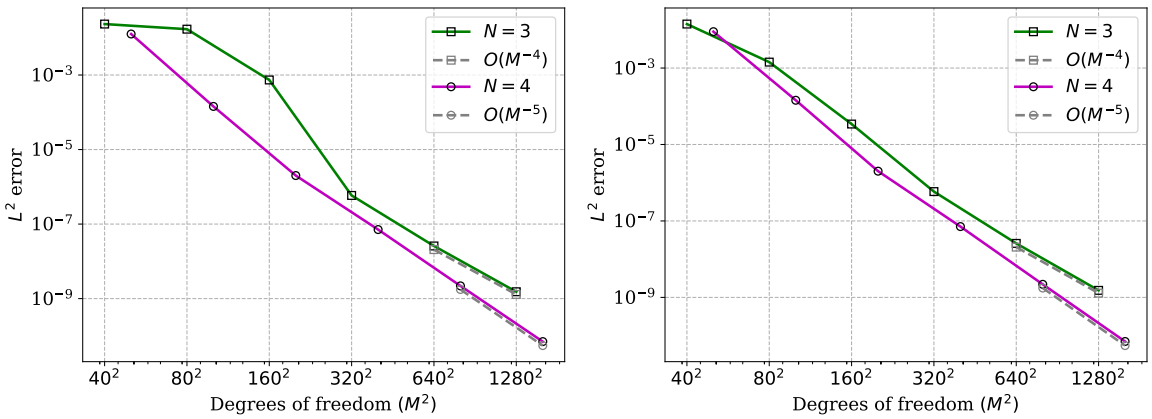


Figure 5.10. Convergence analysis of isentropic vortex test for polynomial degrees $N = 3, 4$ when (a) the blending limiter is active (b) no limiter is active.

5.9.2. Double Mach reflection

The description and significance of this test have been given in Section 4.11.2. The simulation is run on a mesh of 600×150 elements using degree $N = 4$ polynomials up to time $t = 0.2$. In Figure 5.11, the density plot generated using LWFR with MUSCL-Hancock blending scheme is shown. In Figure 5.12, we compare the results of `Trixi.jl` with the MUSCL-Hancock blended scheme zoomed near the primary triple point with the same 600×150 mesh resolution in Figure 5.12a,b. In Figure 5.12c, we show a solution generated with `Trixi.jl` on a finer mesh of 1600×400 elements. The MUSCL-Hancock (Figure 5.12b) captures more of the small scale structures present in the reference solution (Figure 5.12c) than the first order blending scheme of `Trixi.jl` (Figure 5.12a).

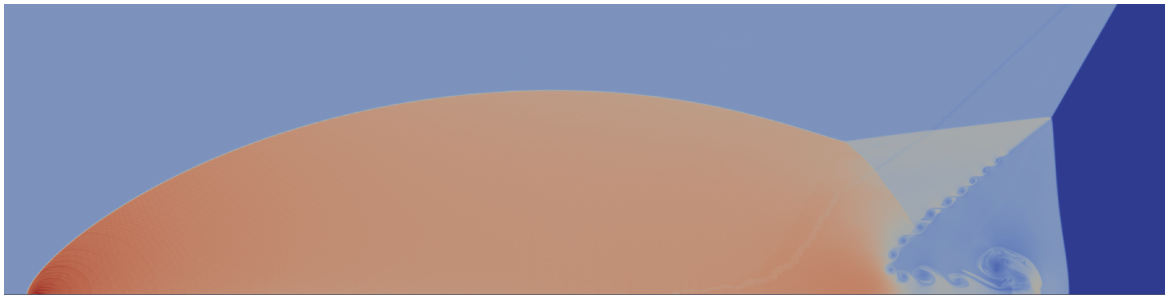
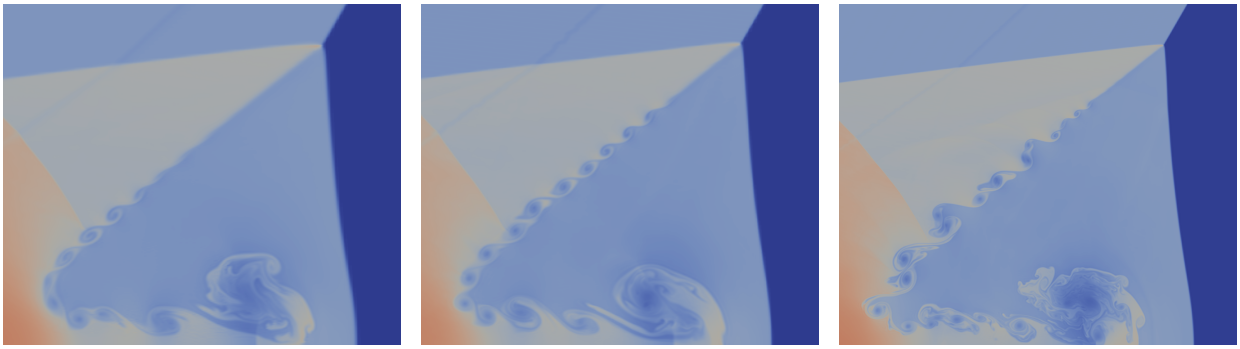


Figure 5.11. Double Mach reflection problem, density plot of numerical solution at $t = 0.2$ using polynomial degree $N = 4$ on a 600×150 mesh generated using Lax-Wendroff Flux Reconstruction with MUSCL-Hancock blending scheme.



(a) `Trixi.jl` zoomed (b) LW-MH zoomed (c) Reference solution

Figure 5.12. Double Mach reflection problem, density plots of numerical solution at $t = 0.2$ using polynomial degree $N = 4$ on a 600×150 mesh zoomed near the primary triple point.

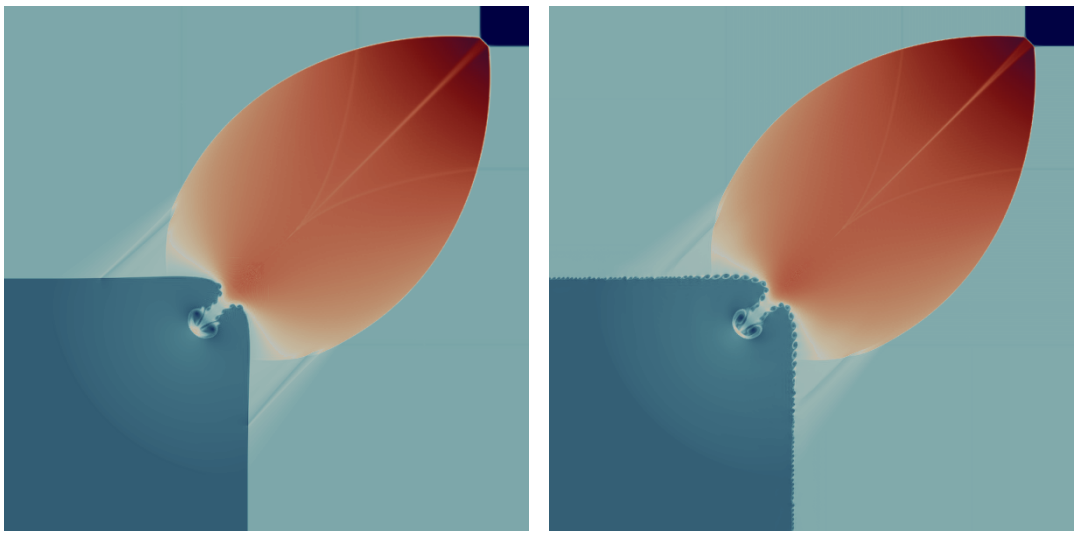
5.9.3. 2-D Riemann problem

2-D Riemann problems consist of four constant states and have been studied theoretically and numerically for gas dynamics in [80]. We consider this problem in the square domain $[0, 1]^2$ where each of the four quadrants has one constant initial state and every jump in initial condition leads to an elementary planar wave, i.e., a shock, rarefaction or contact discontinuity. There are only 19 such genuinely different configurations possible [203, 112]. As studied in [203], a bounded region of subsonic flows is formed by interaction of different planar waves leading to appearance of many complex structures depending on the elementary planar flow. We consider configuration 12 of [112] consisting of 2 positive slip lines and two forward shocks, with initial condition

$$(\rho, u, v, p) = \begin{cases} (0.5313, 0, 0, 0.4) & \text{if } x \geq 0.5, y \geq 0.5 \\ (1, 0.7276, 0, 1) & \text{if } x < 0.5, y \geq 0.5 \\ (0.8, 0, 0, 1) & \text{if } x < 0.5, y < 0.5 \\ (1, 0, 0.7276, 1) & \text{if } x \geq 0.5, y < 0.5 \end{cases}$$

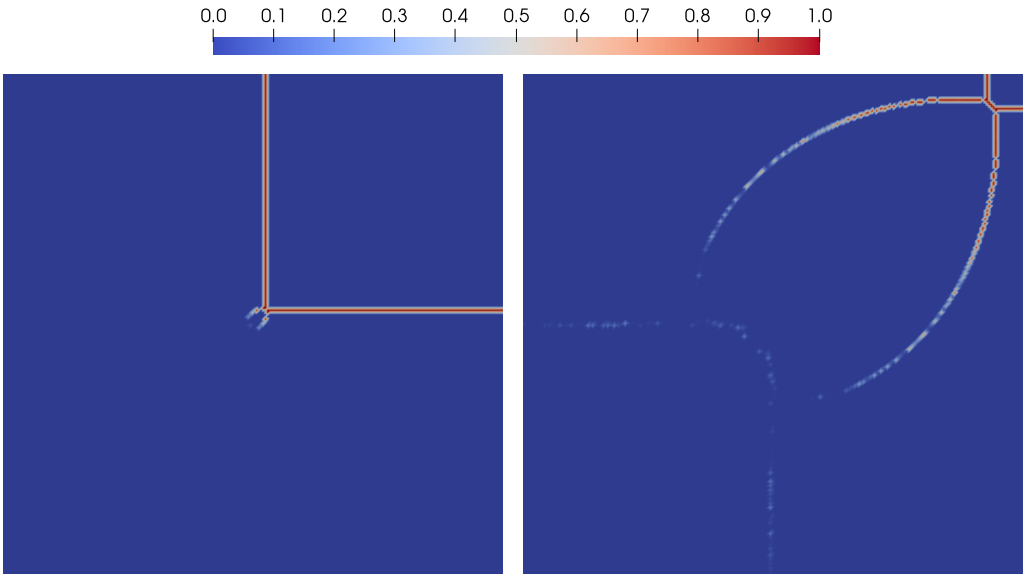
The simulations are performed with transmissive boundary conditions on an enlarged domain up to time $t = 0.25$. The density profiles obtained from the MUSCL-Hancock blending scheme and `Trixi.jl` are shown in Figure 5.13. We see that both schemes give similar resolutions in most regions. The MUSCL-Hancock blending scheme gives better resolution of the small scale structures arising across the slip lines.

A plot of the blending coefficients computed by the smoothness indicator is shown in Figure 5.14 at an early time $t = 0.025$ (5.14a) and the final time $t = 0.25$ (5.14b). The blending coefficient takes values close to $\alpha = 1$ in the vicinity of shocks while smaller values are seen near the stationary contact discontinuities. Figure 5.15 shows the percentage of cells in which the indicator function $\alpha > 0$ as a function of time. From these figures, we see that limiting is only performed in a small subset of the elements in the grid and the indicator is able to track the sharp features and ignore the smooth regions.



(a) `Trixi.jl` (b) LW-MH

Figure 5.13. 2-D Riemann problem, density plots of numerical solution at $t = 0.25$ for degree $N = 4$ on a 256×256 mesh.



(a) `Trixi.jl` (b) LW-MH

Figure 5.14. 2-D Riemann problem, blending coefficient α for degree $N = 4$ on a 256×256 mesh.

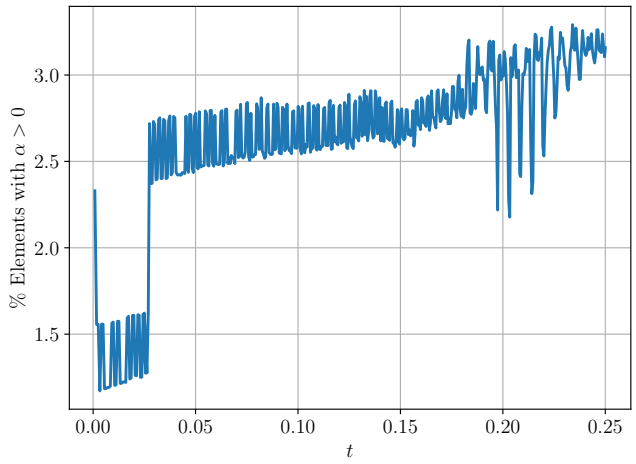


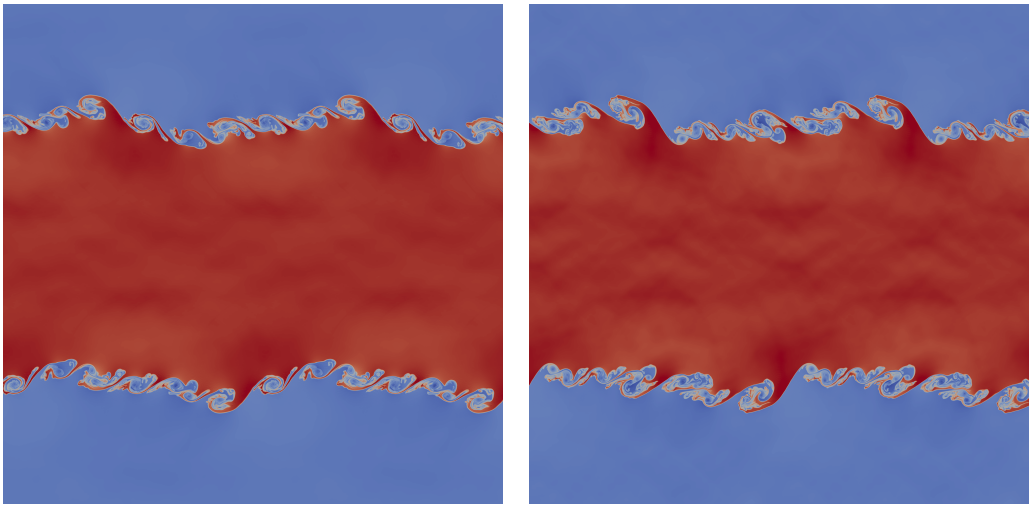
Figure 5.15. 2-D Riemann problem, percentage of elements where the blending coefficient $\alpha > 0$ vs time t, for approximate solution with polynomial degree $N = 4$ on a 256×256 mesh.

5.9.4. Kelvin-Helmholtz instability

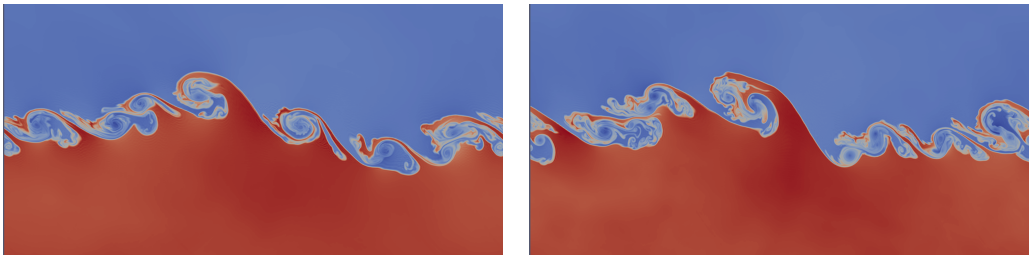
Fluid instabilities are essential for mixing processes and turbulence production, and play a significant role in many astrophysical phenomena. They are crucial for accurately modeling stripping of gas from satellite galaxies, as well as calculating the expected levels of turbulence and entropy in the intracluster gas of galaxy clusters [168]. The Kelvin-Helmholtz instability is a common fluid instability that occurs across contact discontinuities in the presence of a tangential shear flow. This instability leads to the formation of vortices that grow in amplitude and can eventually lead to the onset of turbulence. We adopt the initial conditions for this instability from [168] over the domain $[0, 1]^2$,

$$
\begin{aligned}
\rho\,(x, y) &= \begin{cases} 2, & \text{if} \quad 0.25 < y < 0.75 \\ 1, & \text{otherwise} \end{cases} \\
u\,(x, y) &= \begin{cases} 0.5, & \text{if} \quad 0.25 < y < 0.75 \\ -0.5, & \text{otherwise,} \end{cases} \\
v(x, y) &= w_0 \sin(4\,\pi\,x) \left\{ \exp\left[-\frac{(y - 0.25)^2}{2\,\sigma^2} \right] + \exp\left[-\frac{(y - 0.75)^2}{2\,\sigma^2} \right] \right\} \\
p\,(x, y) &= 2.5
\end{aligned}
$$

with $w_0 = 0.1$, $\sigma = 0.05/\sqrt{2}$ and the adiabatic index $\gamma = 7/5$ corresponding to diatomic gases. The initial conditions consist of a single strongly excited mode in the y velocity concentrated near the interfaces. The wavelength is chosen to be equal to half the domain size so that the single mode dominates the linear growth of instability. This instability leads to shearing and small scale, self-similar vortex structures. We run this test using solution polynomial degree $N = 4$ on a mesh of 512^2 elements with periodic boundary conditions. We compare the density profiles of `Trixi.jl` and our MUSCL-Hancock blending scheme in Figure 5.16. The presence of more vortex structures with the MUSCL-Hancock scheme suggests that the scheme has lesser dissipation errors and is capable of capturing small scale features.



(a) `Trixi.jl` (b) LW-MH



(c) `Trixi.jl` zoomed around top left (d) LW-MH zoomed around top left

Figure 5.16. Kelvin-Helmholtz instability, density plots of numerical solution at $t = 0.4$ using polynomial degree $N = 4$ with Rusanov flux on a 512^2 element mesh.

5.9.5. Astrophysical jet

In this test, a hypersonic jet is injected into a uniform medium with a Mach number of 2000 relative to the sound speed in the medium. Following [86, 206], the domain is taken to be $[0, 1] \times [-0.5, 0.5]$, the ambient gas in the interior has state $\boldsymbol{u}_a$ defined in primitive variables as

$$(\rho, u, v, p)_a = (0.5, 0, 0, 0.4127)$$

and inflow state $\boldsymbol{u}_j$ is defined in primitive variables as

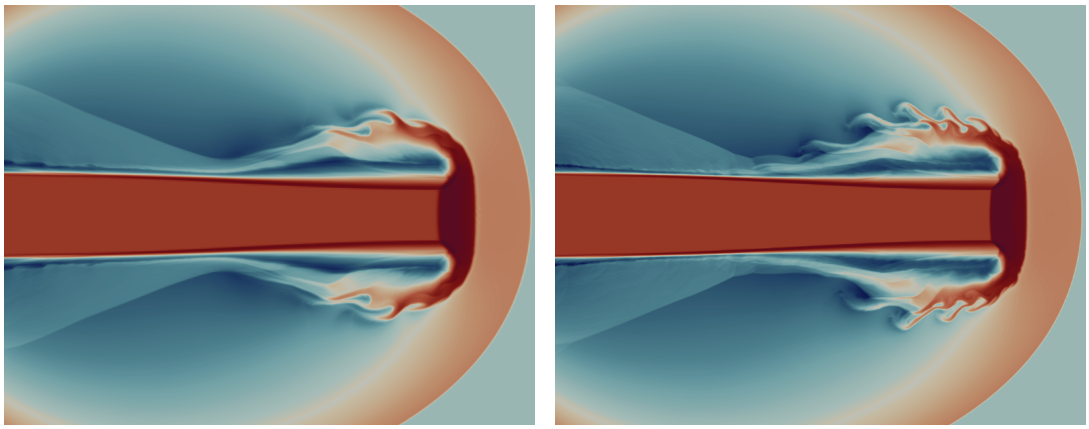
$$(\rho, u, v, p)_j = (5, 800, 0, 0.4127)$$

On the left boundary, we impose the boundary conditions

$$\boldsymbol{u}_b = \begin{cases} \boldsymbol{u}_a, & \text{if} \quad y \in [-0.05, 0.05] \\ \boldsymbol{u}_j, & \text{otherwise} \end{cases}$$

and outflow conditions on the right, top and bottom. The HLLC numerical flux was used in the left most cells to distinguish between characteristics entering and exiting the domain. To get better resolution of vortices, we used a smaller time step with

$C_{\text{CFL}} = 0.5$ in (4.30) and included ghost elements in time step computation to handle the cold start. The high velocity makes the kinetic energy much higher than internal energy. Thus, it is very likely for numerical solvers to give negative pressures. At the same time, a Kelvin-Helmholtz instability arises before the bow shock. Thus, it is a good test both for admissibility preservation and capturing small scale structures. The simulation gives negative pressures if used without the proposed admissibility preservation techniques. While the large scale structures are captured similarly by both the schemes as seen in Figure 5.17, the LWFR with MH blending scheme shows more small scales near the front of the jet.



(a) `Trixi.jl` (b) LW-MH

Figure 5.17. Mach 2000 astrophysical jet, density plot of numerical solution in log scales on 400×400 mesh at time $t = 0.001$.

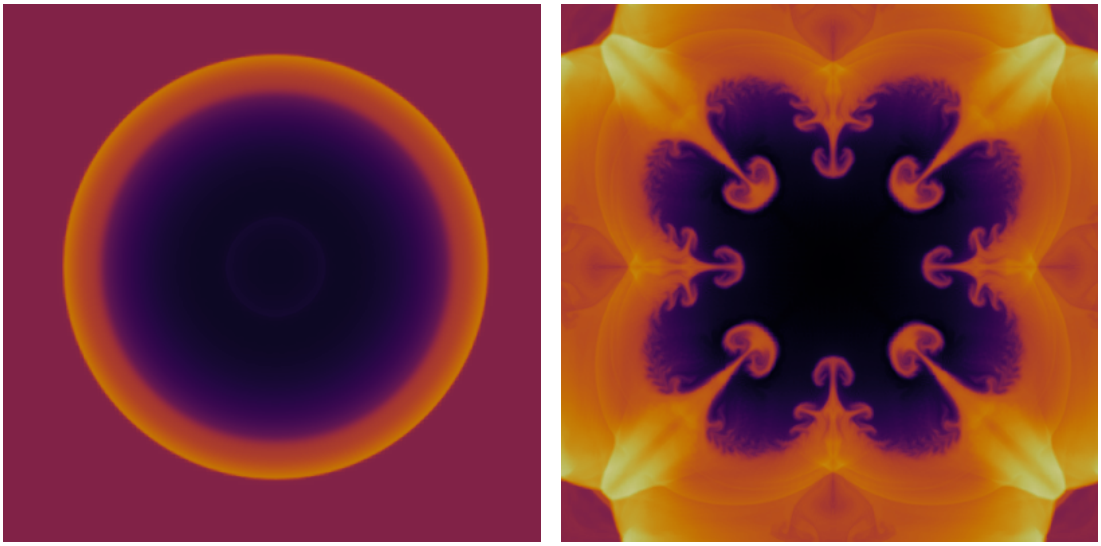
5.9.6. Sedov's blast case with periodic boundary conditions

Similar to Sedov's blast test in Section 5.7.1.3 this test from [151] on domain $[-1.5, 1.5]^2$ has energy concentrated at the origin. More precisely, for the initial condition, we assume that the gas is at rest ($u = v = 0$) with Gaussian distribution of density and pressure

$$\rho(x,y) = \rho_0 + \frac{1}{4\pi\sigma_\rho^2} e^{-\frac{r^2}{2\sigma_\rho^2}}, \quad p(x,y) = p_0 + \frac{\gamma - 1}{4\pi\sigma_p^2} e^{-\frac{r^2}{2\sigma_p^2}}, \quad r^2 = x^2 + y^2 \tag{5.30}$$

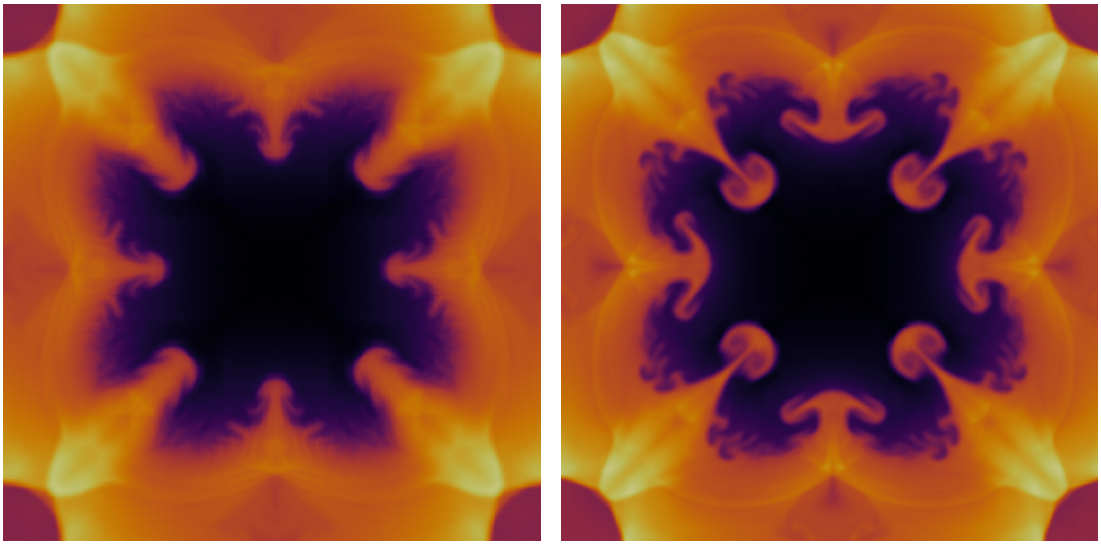
where $\sigma_\rho = 0.25$ and $\sigma_p = 0.15$. The ambient density and ambient pressure are set to $\rho_0 = 1$, $p_0 = 10^{-5}$. There are two differences in this Sedov's test compared to the previous one - the energy concentrated at the origin is lesser, and domain is assumed to be periodic. When shocks collide at the periodic boundaries, the resulting interaction leads to the formation of small scale structures. A reference solution on a 128^2 element mesh with polynomial degree $N = 4$ is shown in Figure 5.18. In Figure 5.19, we compare the density profiles of the numerical solutions of polynomial degree $N = 4$ on a mesh of 64^2 elements using `Trixi.jl` and the proposed MUSCL-Hancock blending scheme in log scales. The solution on the coarse mesh generated by the proposed scheme is able to resolve small scale structures better than the solution of `Trixi.jl` on the coarse

mesh. This is most evidently seen by looking at the *mushroom structures* as some of the mushroom structures in the MUSCL-Hancock scheme (Figure 5.19b) look very similar to the reference solution (Figure 5.18b).



(a) $t = 2$ (b) $t = 20$

Figure 5.18. Sedov's blast test with periodic domain, density plot of numerical solution on 128×128 mesh in log scales with degree $N = 4$ at (a) $t = 2$ and (b) $t = 20$ with polynomial degree $N = 4$ computed using `Trixi.jl`.



(a) `Trixi.jl` (b) LW-MH

Figure 5.19. Sedov's blast test with periodic domain, density plot of numerical solution on 64×64 mesh in log scales at $t = 20$ with degree $N = 4$.

5.9.7. Detonation shock diffraction

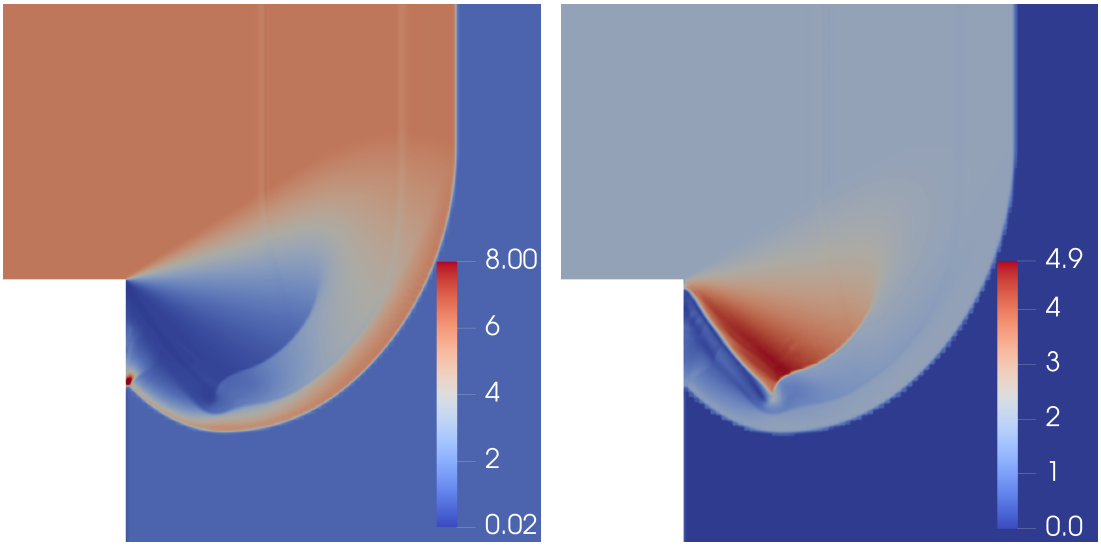
This test [173] involves a planar detonation wave that interacts with a wedge-shaped

corner and diffracts around it, resulting in a complicated wave pattern comprising of transmitted and reflected shocks, as well as rarefaction waves. The computational domain is $\Omega = [0,2]^2 \setminus ([0,0.5] \times [0,1])$ and following [90], the simulation is performed by taking the initial condition to be a pure right-moving shock with Mach number of 100 initially located at $x = 0.5$ and travelling through a channel of resting gas. The post shock states are computed by normal relations [4], so that the initial data is

$$\rho(x,y) = \begin{cases} 5.9970, & \text{if } x \le 0.5 \\ 1, & \text{if } x > 0.5 \end{cases}, \qquad u(x,y) = \begin{cases} 98.5914, & \text{if } x \le 0.5 \\ 0, & \text{if } x > 0.5 \end{cases}$$

$$v(x,y) = 0, \qquad p(x,y) = \begin{cases} 11666.5, & \text{if } x \le 0.5 \\ 1, & \text{if } x > 0.5 \end{cases}$$

The left boundary is set as inflow and right boundary is set as outflow, all other boundaries are solid walls. The numerical results at $t = 0.01$ with polynomial degree $N = 4$ on a Cartesian grid consisting of uniformly sized squares with $\Delta x = \Delta y = 1/200$ are shown in Figure 5.20. The results look similar to [90]; the strong shock makes this a tough test for the admissibility preservation and negative pressure values are obtained if the proposed admissibility correction is not applied.



(a) Density (b) Mach number

Figure 5.20. Shock diffraction test, numerical solution at time $t = 0.01$ with polynomial degree $N = 4$.

5.9.8. Forward facing step

Forward facing step is a classical test case from [73, 197] where a uniform supersonic flow passes through a channel with a forward facing step generating several phenomena like a strong bow shock, shock reflections and a Kelvin-Helmholtz instability. It is a good test for demonstrating a shock capturing scheme's capability of capturing small

scale vortex structures while suppressing spurious oscillations arising from shocks. The step is simulated by taking the domain to be $\Omega = ([0,3] \times [0,1]) \setminus ([0.6,3] \times [0,0.2])$ and the initial conditions are taken to be

$$(\rho, u, v, p) = (1.4, 3, 0, 1)$$

The initial conditions are taken to be constant over the whole domain Ω. The left boundary condition is taken as an inflow and the right one is an outflow, the rest are solid walls. The corner $(0.6, 0.2)$ of the step is the center of a rarefaction fan and is thus a singular point leading to formation of a spurious boundary layer. The modern treatment of this issue is to use a more refined mesh near the corner point, which is what we do in Chapter 8. For now, we obtain the same outcome by forming 1-D meshes in $[0,1], [0,3]$ with the same grid spacing $\Delta x_{\max}$ away from the singularity and the smaller grid spacing $\Delta x_{\min} = \frac{1}{4}\Delta x_{\max}$ in $[0.15, 0.25], [0.45, 0.75]$. Then, the 2-D mesh is formed by taking a tensor product of the two 1-D meshes with cells from $[0.6, 3] \times [0, 0.2]$ removed. We show the density profile of numerical solutions in Figure (5.22a, b, c) for solution polynomial degrees $N = 2, 3, 4$ with $\Delta x_{\max} = 1/160$. The scheme captures both the shock and the small scale vortices, with better resolution of shear structures from the triple shock point near the top wall as the overall resolution is increased. The corner point singularity causes an artificial boundary layer and Mach stem to occur but these numerical artifacts decrease as we increase mesh resolution by increasing the polynomial degree. Figure 5.21 shows the time evolution of the percentage of cells in the grid where the blending coefficient $\alpha > 0$ and Figure 5.22d plots the blending coefficient for degree $N = 4$ solution at the final time; these figures show that the blending limiter is activated in a small fraction of the cells and only in the vicinity of shocks.

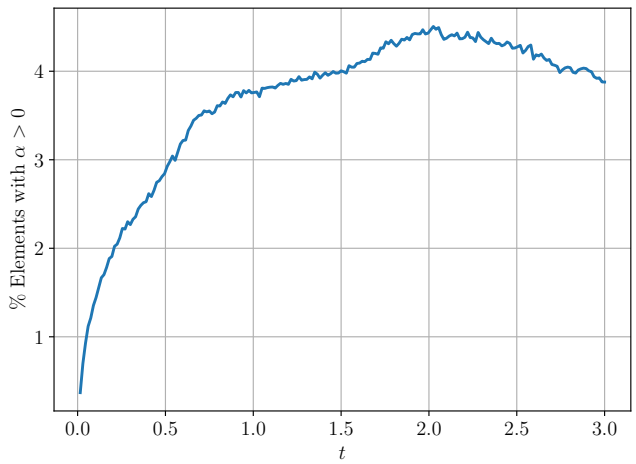
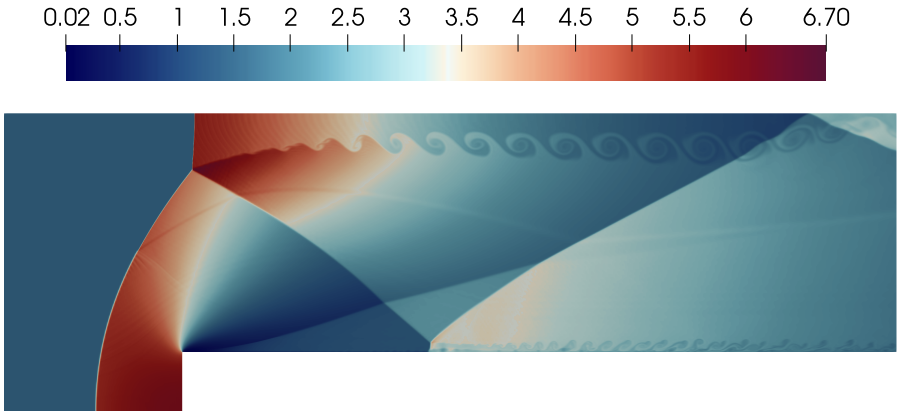


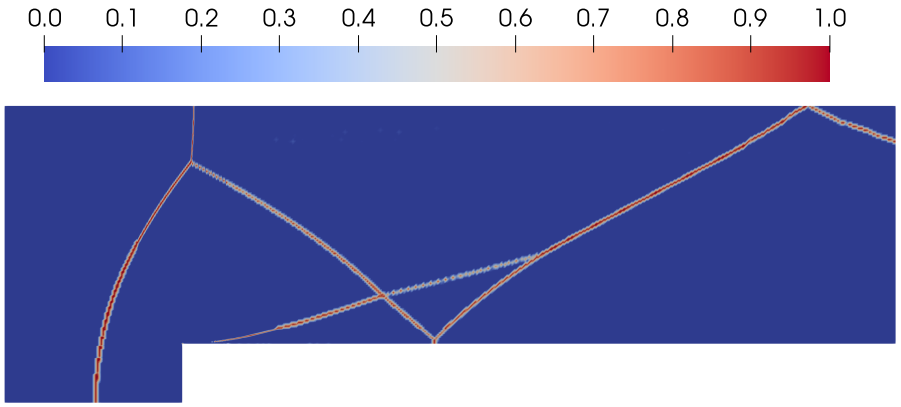
Figure 5.21. Forward facing step test case, percentage of elements where the blending coefficient α is non-zero versus time t for approximate solution with polynomial degree $N = 4$ on a mesh with $\Delta x_{\max} = 1/160$.



(a) $N = 2$

(b) $N = 3$

(c) $N = 4$



(d) Blending coefficient α for $N = 4$

Figure 5.22. Forward facing step, density plots of numerical solution at time $t = 3$ with solution polynomial degrees $N = 2, 3, 4$ (a, b, c) and blending coefficient plot for degree $N = 4$ (d). The meshes are formed by taking grid spacing $\Delta x_{\max} = \Delta y_{\max}$ away from the corner and smaller grid spacing $\Delta x_{\min} = \Delta y_{\min} = \frac{1}{4}\Delta x_{\max}$ near the corner.

5.10. SUMMARY AND CONCLUSIONS

An admissibility preserving subcell-based blending limiter for the high order Lax-Wendroff Flux Reconstruction (LWFR) scheme has been constructed by extending the LWFR scheme proposed in Chapter 4 using the blending limiter of [90]. The scheme uses a smoothness indicator to blend two single-stage solvers on the FR grid, one based on the high order LWFR method and the other based on a finite volume update on the subcells. At the FR element interfaces, a *blended numerical flux* is constructed using the Lax-Wendroff time averaged flux and lower order numerical flux. The same blended numerical flux is used by both schemes at the element interfaces to maintain conservation. The crucial observation used for obtaining admissibility preservation was that admissibility preservation in means is a consequence of admissibility of the lower order updates. A simple and efficient procedure to obtain admissibility preservation in means was thus proposed, where lower-order updates are made admissible by appropriately constructing the blending numerical flux within the face loop. This approach eliminates the need for additional element or interface loops, minimizing storage requirements. The user only needs to provide the admissibility constraints $\{P_k, k = 1, \ldots, K\}$ of the conservative variables and whose positivity implies that the solution is in the admissible set $\mathcal{U}_{\text{ad}}$, making the correction problem-independent. Once admissibility preservation in means is obtained, we use the scaling limiter of [206] to enforce admissibility of the polynomial values. To enhance accuracy, we modified the blending scheme of [90] to use Gauss-Legendre solution points and used the second-order MUSCL-Hancock scheme to compute the lower-order residual. We extended the slope restriction criterion of [26] for admissibility of the MUSCL-Hancock scheme to non-cell-centered grids that arise in the blending scheme to maintain the conservation property. We also proposed a problem-independent procedure to enforce the slope restriction. The scheme is robust and the higher resolution of MUSCL-Hancock is more superior in capturing small scale structures, as was demonstrated by numerical experiments on compressible Euler equations.

Chapter 6

Generalized admissibility preservation and source terms

6.1. Introduction

In Chapter 5, we developed an admissibility preserving subcell based blending scheme for LWFR by exploiting the subcell structure to appropriately construct the *blended numerical flux*. The subcell based limiter was initially introduced to control oscillations and then also used to obtain admissibility preservation. In this chapter, we will show that the role of subcell based blending in admissibility preservation was primarily to derive and motivate the construction of the blended numerical flux. That is, we now propose a *generalized procedure* of limiting the time average flux to obtain provable admissibility preservation. This procedure can be combined with any choice of limiter to control oscillations and get a robust, admissibility preserving scheme. In this chapter, we use it with the TVB limiter to verify admissibility preservation. The idea of the generalized procedure is to perform a cell average decomposition like in [205] for LWFR and perform flux limiting to enforce admissibility in means. The LWFR scheme is extended to apply to conservation laws with source terms by performing time averaging of source terms. The extension is made provably admissibility preserving by limiting the time average source terms. To numerically validate our claims, we test LWFR on the Ten Moment equations, which are derived by Levermore et al. [118] by taking a Gaussian closure of the kinetic model.

The rest of the chapter is organized as follows. Section 6.2 describes the LWFR scheme for conservation laws with source terms, and notions of admissibility preservation. Section 6.3 describes the additional limiting required in LW scheme for admissibility preservation, i.e., for the time averaged flux (Section 6.3.1) and time averaged sources (Section 6.3.2). Section 6.4 shows the numerical results for the Ten Moment equations model and a summary of the the chapter is given in Section 6.5.

6.2. LWFR for source terms

Consider a conservation law of the form

$$\boldsymbol{u}_t + \boldsymbol{f}_x = \boldsymbol{s} \tag{6.1}$$

where $\boldsymbol{u} \in \mathbb{R}^p$ is the vector of conserved quantities, $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{u})$ is the corresponding flux, $\boldsymbol{s} = \boldsymbol{s}(\boldsymbol{u}, t, x)$ is the source term, together with some initial and boundary conditions. As in the case of $\boldsymbol{s} = \boldsymbol{0}$ (3.1), the solution that is physically correct is assumed to belong to an admissibility set $\mathcal{U}_{\text{ad}}$ (5.1).

Following Chapter 4, the LWFR scheme for source terms is derived from a Taylor's expansion in time at $t = t_{n+1}$ around $t = t_n$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \sum_{m=1}^{N+1} \frac{\Delta t^m}{m!} \partial_t^m \boldsymbol{u}^n + O(\Delta t^{N+2})$$

Since the spatial error is expected to be of $O(\Delta x^{N+1})$, we retain terms up to $O(\Delta t^{N+1})$ in the Taylor expansion, so that the overall formal accuracy is of order $N+1$ in both space and time. Using the conservation law with source terms, $\partial_t \boldsymbol{u} = -\partial_x \boldsymbol{f} + \boldsymbol{s}$ (6.1), we re-write time derivatives of the solution in terms of spatial derivatives of the flux and source terms

$$\partial_t^m \boldsymbol{u} = -(\partial_t^{m-1} \boldsymbol{f})_x + \partial_t^{m-1} \boldsymbol{s}, \qquad m = 1, 2, \ldots$$

so that

$$\begin{aligned} \boldsymbol{u}^{n+1} &= \boldsymbol{u}^n - \sum_{m=1}^{N+1} \frac{\Delta t^m}{m!} (\partial_t^{m-1} \boldsymbol{f})_x + \sum_{m=1}^{N+1} \frac{\Delta t^m}{m!} \partial_t^{m-1} \boldsymbol{s} + O(\Delta t^{N+2}) \\ &= \boldsymbol{u}^n - \Delta t \frac{\partial \boldsymbol{F}}{\partial x}(\boldsymbol{u}^n) + \Delta t\, \boldsymbol{S}(\boldsymbol{u}^n, t^n) + O(\Delta t^{N+2}) \end{aligned} \tag{6.2}$$

where

$$\boldsymbol{F} = \sum_{m=0}^{N} \frac{\Delta t^m}{(m+1)!} \partial_t^m \boldsymbol{f} = \boldsymbol{f} + \frac{\Delta t}{2} \partial_t \boldsymbol{f} + \ldots + \frac{\Delta t^N}{(N+1)!} \partial_t^N \boldsymbol{f} \tag{6.3}$$

$$\boldsymbol{S} = \sum_{m=0}^{N} \frac{\Delta t^m}{(m+1)!} \partial_t^m \boldsymbol{s} = \boldsymbol{s} + \frac{\Delta t}{2} \partial_t \boldsymbol{s} + \ldots + \frac{\Delta t^N}{(N+1)!} \partial_t^N \boldsymbol{s} \tag{6.4}$$

Note that $\boldsymbol{F}(\boldsymbol{u}^n), \boldsymbol{S}(\boldsymbol{u}^n, t^n)$ are approximations to the time average flux and source term in the interval $[t_n, t_{n+1}]$ since they can be written as

$$\boldsymbol{F}(\boldsymbol{u}^n) = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \left[\boldsymbol{f}(\boldsymbol{u}^n) + \ldots + \frac{(t-t_n)^N}{N!} \partial_t^N \boldsymbol{f}(\boldsymbol{u}^n) \right] \mathrm{d}t \tag{6.5}$$

$$\boldsymbol{S}(\boldsymbol{u}^n, t^n) = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \left[\boldsymbol{s}(\boldsymbol{u}^n, t^n) + \ldots + \frac{(t-t_n)^N}{N!} \partial_t^N \boldsymbol{s}(\boldsymbol{u}^n, t^n) \right] \mathrm{d}t \tag{6.6}$$

where the quantity inside the square brackets is the truncated Taylor expansion of the flux $\boldsymbol{f}$ or source $\boldsymbol{s}$ in time. Following equation (6.2) we need to specify the construction of the time averaged flux (6.3) and the time averaged source terms (6.4). The first step of approximating (6.2) is the predictor step where a local degree N approximation $\boldsymbol{F}^\delta$ of the time averaged flux is computed by the approximate Lax-Wendroff procedure (Section 4.2.4). Then, as in the standard RKFR scheme, we perform the Flux Reconstruction procedure on $\boldsymbol{F}^\delta$ to construct a local degree $N+1$ and globally continuous flux approximation $\boldsymbol{F}_h(\xi)$. The time average source $\boldsymbol{S}$ will also be approximated locally as a degree N polynomial using the approximate Lax-Wendroff procedure and denoted with a single notation $\boldsymbol{S}^\delta(\xi)$ since it needs no correction. The scheme for local approxi-

mation is discussed in Section 6.2.1. After computing $\boldsymbol{F}_h, \boldsymbol{S}^\delta$, truncating equation (6.2), the solution at the nodes is updated by a collocation scheme as follows

$$\boldsymbol{u}_{e,p}^{n+1} = \boldsymbol{u}_{e,p}^{n} - \frac{\Delta t}{\Delta x_e} \frac{\mathrm{d}\boldsymbol{F}_h}{\mathrm{d}\xi}(\xi_p) + \Delta t\, \boldsymbol{S}^\delta(\xi_p), \qquad 0 \le p \le N \tag{6.7}$$

This is the single step Lax-Wendroff update scheme for any order of accuracy.

6.2.1. Approximate Lax-Wendroff procedure for degree $N = 2$

The approximations of temporal derivatives of $\boldsymbol{s}$ are made in a similar fashion as those of $\boldsymbol{f}$ in [208, 18] (Section 4.2.4). For example, to obtain second order accuracy, $\partial_t \boldsymbol{s}$ can be approximated as

$$\partial_t \boldsymbol{s}(\boldsymbol{u}, \boldsymbol{x}, t) \;\approx\; \frac{\boldsymbol{s}(\boldsymbol{u}^n + \Delta t \boldsymbol{u}_t^n, \boldsymbol{x}, t^{n+1}) - \boldsymbol{s}(\boldsymbol{u}^n - \Delta t \boldsymbol{u}_t^n, \boldsymbol{x}, t^{n-1})}{2\,\Delta t}$$

where $\boldsymbol{u}_t = -\partial_x \boldsymbol{f} + \boldsymbol{s}(\boldsymbol{u}, \boldsymbol{x}, t)$. Denoting $\mathsf{g}^{(k)}$ as an approximation for $\Delta t^k\, \partial_t^k\, g$, we explain the local flux and source term approximation procedure for degree $N = 2$

$$\mathsf{F} = \mathsf{f} + \frac{1}{2}\mathsf{f}^{(1)} + \frac{1}{6}\mathsf{f}^{(2)}, \qquad \mathsf{S} = \mathsf{s} + \frac{1}{2}\mathsf{s}^{(1)} + \frac{1}{6}\mathsf{s}^{(2)}$$

where

$$\begin{aligned}
\mathsf{u}^{(1)} &= -\frac{\Delta t}{\Delta x_e}\mathsf{D}\mathsf{f} + \Delta t\, \mathsf{s} \\
\mathsf{f}^{(1)}, \mathsf{s}^{(1)} &= \frac{1}{2}\left[\boldsymbol{f}(\mathsf{u}+\mathsf{u}^{(1)}) - \boldsymbol{f}(\mathsf{u}-\mathsf{u}^{(1)})\right], \frac{1}{2}\left[\boldsymbol{s}(\mathsf{u}+\mathsf{u}^{(1)}) - \boldsymbol{s}(\mathsf{u}-\mathsf{u}^{(1)})\right] \\
\mathsf{u}^{(2)} &= -\frac{\Delta t}{\Delta x_e}\mathsf{D}\mathsf{f}^{(1)} + \Delta t\, \mathsf{s}^{(1)} \\
\mathsf{f}^{(2)} &= \boldsymbol{f}\left(\mathsf{u}+\mathsf{u}^{(1)}+\frac{1}{2}\mathsf{u}^{(2)}\right) - 2\,\boldsymbol{f}(\mathsf{u}) + \boldsymbol{f}\left(\mathsf{u}-\mathsf{u}^{(1)}+\frac{1}{2}\mathsf{u}^{(2)}\right) \\
\mathsf{s}^{(2)} &= \boldsymbol{s}\left(\mathsf{u}+\mathsf{u}^{(1)}+\frac{1}{2}\mathsf{u}^{(2)}\right) - 2\,\boldsymbol{s}(\mathsf{u}) + \boldsymbol{s}\left(\mathsf{u}-\mathsf{u}^{(1)}+\frac{1}{2}\mathsf{u}^{(2)}\right)
\end{aligned}$$

The local approximation of the flux $\boldsymbol{F}$ for all degrees and then its FR correction using the time numerical flux $\boldsymbol{F}_{e+\frac{1}{2}}$ is as in Chapter 4.

6.2.2. Admissibility preservation

As in Chapter 5, the idea is to obtain admissibility preservation in means (Definition 5.2) and then use the scaling limiter of [205] (Appendix F)to obtain an admissibility preserving LWFR scheme (Definition 5.1). The following conservation property of the LWFR scheme will be crucial in obtaining admissibility preservation in means

$$\bar{\boldsymbol{u}}_e^{n+1} = \bar{\boldsymbol{u}}_e^{n} - \frac{\Delta t}{\Delta x_e}\left(\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}}\right) + \Delta t \bar{\boldsymbol{S}}_e \tag{6.8}$$

where $\bar{\boldsymbol{u}}_e^n$ is the cell average of solution, $\bar{\boldsymbol{S}}_e := \sum_{p=0}^N w_p \boldsymbol{S}_e^\delta(\xi_p)$ is the cell average of the source term. As in (4.5), the conservation property (6.8) is obtained by multiplying (6.7) by the quadrature weights associated with the solution points and sum over all the points in the e^{th} element. In the subsequent sections, we discuss limiting of time average sources and fluxes in order to obtain admissibility in means.

6.3. LIMITING TIME AVERAGES

6.3.1. Limiting time average flux

In this section, we describe the approach to obtain admissibility preservation in means property (5.2) for the LWFR update (6.7) in the case where source term $\boldsymbol{s}$ in (6.1) is zero. In Chapter 5, a subcell based blending limiter was used which helped in controlling spurious oscillations but also motivated construction of the blended numerical flux that gave us admissibility preservation in means. The admissibility preserving scheme used in Chapter 5 was a combination of the subcell based blending scheme and a flux limiter. The approach we now describe generalizes Chapter 5 in the sense that we can use the blended numerical flux to obtain admissibility preservation in means even without using the subcell based blending limiter, allowing us to use a different limiter for controlling oscillations. The procedure begins by following the work of Zhang and Shu [205] to define *fictitious finite volume updates*

$$\begin{aligned}
\hat{\boldsymbol{u}}_{e,0}^{n+1} &= \boldsymbol{u}_{e,0}^n - \frac{\Delta t}{w_0 \Delta x_e}\left[\boldsymbol{f}_{\frac{1}{2}}^e - \boldsymbol{F}_{e-\frac{1}{2}}^{\text{LW}}\right] \\
\hat{\boldsymbol{u}}_{e,p}^{n+1} &= \boldsymbol{u}_{e,p}^n - \frac{\Delta t}{w_p \Delta x_e}\left[\boldsymbol{f}_{p+\frac{1}{2}}^e - \boldsymbol{f}_{p-\frac{1}{2}}^e\right], \qquad 1 \le p \le N-1 \\
\hat{\boldsymbol{u}}_{e,N}^{n+1} &= \boldsymbol{u}_{e,N}^n - \frac{\Delta t}{w_N \Delta x_e}\left[\boldsymbol{F}_{e+\frac{1}{2}}^{\text{LW}} - \boldsymbol{f}_{N-\frac{1}{2}}^e\right]
\end{aligned} \tag{6.9}$$

where $\boldsymbol{f}_{p+\frac{1}{2}}^e = \boldsymbol{f}(\boldsymbol{u}_{e,p}^n, \boldsymbol{u}_{e,p+1}^n)$ is an admissibility preserving finite volume numerical flux (Definition 3.1). The fictitious updates of (6.9) look similar to a lower scheme on subcells (5.8) and can indeed be seen as finite volume updates on subcells. The argument for admissibility preservation in means of the scheme will in fact be obtained by viewing $\hat{\boldsymbol{u}}_{e,p}^{n+1}$ for $p = 0, N$ as evolutions on subcells. However, $\hat{\boldsymbol{u}}_{e,p}^{n+1}$ for $p = 1, \ldots, N-1$ are never explicitly computed. The relation of (6.9) to element means of the scheme is the following

$$\bar{\boldsymbol{u}}_e^{n+1} = \sum_{p=0}^N w_p \, \hat{\boldsymbol{u}}_{e,p}^{n+1} \tag{6.10}$$

Thus, if we can ensure that $\hat{\boldsymbol{u}}_{e,p}^{n+1} \in \mathcal{U}_{\text{ad}}$ for all p, the scheme will be admissibility preserving in means (5.2). We do have $\hat{\boldsymbol{u}}_{e,p}^{n+1} \in \mathcal{U}_{\text{ad}}$ for $1 \le p \le N-1$ under appropriate CFL conditions because the finite volume fluxes are admissibility preserving (3.14). In order

to ensure that the updates $\hat{\boldsymbol{u}}_{e,0}^{n+1}, \hat{\boldsymbol{u}}_{e,N}^{n+1}$ are also admissible, the flux limiting procedure of Chapter 5 is followed so that the high order numerical fluxes $\boldsymbol{F}_{e\pm\frac{1}{2}}^{\mathrm{LW}}$ are replaced by *blended numerical fluxes* $\boldsymbol{F}_{e\pm\frac{1}{2}}$. The procedure is explained here for completeness. We define an admissibility preserving lower order flux at the interface $e+\frac{1}{2}$

$$\boldsymbol{f}_{e+\frac{1}{2}} = \boldsymbol{f}(\boldsymbol{u}_{e+1,0}^n, \boldsymbol{u}_{e,N}^n)$$

Note that, for an RKFR scheme using Gauss-Legendre-Lobatto (GLL) solution points, the definition of $\hat{\boldsymbol{u}}_{e,N}^{n+1}$ will use $\boldsymbol{f}_{e+\frac{1}{2}}$ in place of $\boldsymbol{F}_{e+\frac{1}{2}}^{\mathrm{LW}}$ and thus admissibility preserving in means property will always be present. That is the argument of [205] and here we demonstrate that the same argument can be applied to LWFR schemes by limiting $\boldsymbol{F}_{e+\frac{1}{2}}^{\mathrm{LW}}$. We will explain the procedure for limiting $\boldsymbol{F}_{e+\frac{1}{2}}^{\mathrm{LW}}$ to obtain $\boldsymbol{F}_{e+\frac{1}{2}}$; it will be similar in the case of $\boldsymbol{F}_{e-\frac{1}{2}}$. Note that we want $\boldsymbol{F}_{e+\frac{1}{2}}$ to be such that the following are admissible

$$\begin{aligned} \hat{\boldsymbol{u}}_0^{n+1} &= \boldsymbol{u}_{e+1,0}^n - \frac{\Delta t}{w_0 \Delta x_{e+1}} (\boldsymbol{f}_{\frac{1}{2}}^{e+1} - \boldsymbol{F}_{e+\frac{1}{2}}) \\ \hat{\boldsymbol{u}}_N^{n+1} &= \boldsymbol{u}_{e,N}^n - \frac{\Delta t}{w_N \Delta x_e} (\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{f}_{N-\frac{1}{2}}^{e}) \end{aligned} \tag{6.11}$$

We will exploit the admissibility preserving property of the finite volume fluxes to get

$$\begin{aligned} \hat{\boldsymbol{u}}_0^{\mathrm{low},n+1} &= \boldsymbol{u}_{e+1,0}^n - \frac{\Delta t}{w_0 \Delta x_{e+1}} (\boldsymbol{f}_{\frac{1}{2}}^{e+1} - \boldsymbol{f}_{e+\frac{1}{2}}) \in \mathcal{U}_{\mathrm{ad}} \\ \hat{\boldsymbol{u}}_N^{\mathrm{low},n+1} &= \boldsymbol{u}_{e,N}^n - \frac{\Delta t}{w_N \Delta x_e} (\boldsymbol{f}_{e+\frac{1}{2}} - \boldsymbol{f}_{N-\frac{1}{2}}^{e}) \in \mathcal{U}_{\mathrm{ad}} \end{aligned} \tag{6.12}$$

Thus, to enforce admissibility preservation in means, the flux $\boldsymbol{F}_{e+\frac{1}{2}}^{\mathrm{LW}}$ can be limited by Algorithm 5.1 by using the initial guess $\boldsymbol{F}_{e+\frac{1}{2}} \leftarrow \boldsymbol{F}_{e+\frac{1}{2}}^{\mathrm{LW}}$ and the $\hat{\boldsymbol{u}}_i^{\mathrm{low},n+1}, \hat{\boldsymbol{u}}_i^{n+1}$ defined for $i = 0, N$ in (6.11, 6.12).

6.3.2. Limiting time average sources

After the flux limiting performed in Section 6.3.1, we will have an admissibility preserving in means scheme (5.2) if the source term average $\bar{\boldsymbol{S}}_e$ in (6.8) is zero. In order to get an admissibility preserving scheme in the presence of source terms, we will make a splitting of the cell average update (6.8), which is similar to that of [124]

$$\bar{\boldsymbol{u}}_e^{n+1} = \frac{1}{2}\left(\bar{\boldsymbol{u}}_e^n - \frac{2\,\Delta t}{\Delta x_e} (\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}}) \right) + \frac{1}{2} (\bar{\boldsymbol{u}}_e^n + 2\,\Delta t\, \overline{\boldsymbol{S}_e^{\mathrm{LW}}}) =: \bar{\boldsymbol{u}}_e^{\boldsymbol{F}} + \bar{\boldsymbol{u}}_e^{\boldsymbol{S}^{\mathrm{LW}}} \tag{6.13}$$

where $\boldsymbol{S}_e^{\mathrm{LW}}$ denotes the time average source term in element e computed with the approximate Lax-Wendroff procedure in Section 6.2.1. With the flux limiting per-

formed in Section 6.3.1, we can ensure that cell average $\bar{\boldsymbol{u}}_e^{\boldsymbol{F}} \in \mathcal{U}_{\text{ad}}$ if half the standard CFL is assumed[6.1]. In order to enforce $\bar{\boldsymbol{u}}_e^{\boldsymbol{S}} \in \mathcal{U}_{\text{ad}}$, $\boldsymbol{S}_e^{\text{LW}}$ will be limited as follows. We will use the admissibility of the first order update using the source term

$$\bar{\boldsymbol{u}}_e^{\text{low},n+1} := \bar{\boldsymbol{u}}_e^n + 2\,\Delta t\,\bar{\boldsymbol{s}}_e \in \mathcal{U}_{\text{ad}}, \qquad \bar{\boldsymbol{s}}_e = \sum_{p=0}^{N} w_p\,\boldsymbol{s}(\boldsymbol{u}_{e,p}, \boldsymbol{x}_{e,p}, t^n) \tag{6.14}$$

which will be true under some problem dependent time step restrictions (e.g., Theorem 3.3.1 of [125]). Then, we will find a $\theta \in [0,1]$ so that for $\boldsymbol{S} = \boldsymbol{S}^\theta := \theta\,\boldsymbol{s} + (1-\theta)\,\boldsymbol{S}^{\text{LW}}$, we will have $\bar{\boldsymbol{u}}_e^{\boldsymbol{S}} \in \mathcal{U}_{\text{ad}}$. The θ can be found by iterating over admissibility constraints P_k (5.1). For the constraint P_k we can solve an optimization problem as in (5.23) to find the largest θ satisfying

$$P_k(\bar{\boldsymbol{u}}_e^n + 2\,\Delta t\,\boldsymbol{S}^\theta) = P_k(\theta\,\bar{\boldsymbol{u}}_e^{\boldsymbol{S}^{\text{LW}}} + (1-\theta)\,\bar{\boldsymbol{u}}_e^{\text{low},n+1}) \geq \epsilon \tag{6.15}$$

where ϵ is a tolerance, taken to be $\frac{1}{10} P_k(\bar{\boldsymbol{u}}_e^{\text{low},n+1})$ [151] . We solve (6.15) using a general iterative solver that is independent of choice of P_k (Appendix F). If P_k is a concave function of the conserved variables, as in (5.24), we can use the simpler but possibly sub-optimal approach of defining

$$\theta = \min\left(\min_{p=0,N}\left|\frac{\epsilon_p - P_k(\bar{\boldsymbol{u}}_e^{\text{low},n+1})}{P_k(\bar{\boldsymbol{u}}_e^{\boldsymbol{S}^{\text{LW}}}) - P_k(\bar{\boldsymbol{u}}_e^{\text{low},n+1})}\right|, 1\right) \tag{6.16}$$

Thus, a procedure analogous to Algorithm 5.1 is used for limiting source terms, which we write here for completeness.

Algorithm 6.1

Source limiter

$\overline{\boldsymbol{S}}_e \leftarrow \overline{\boldsymbol{S}}_e^{\text{LW}}$ ▷ Initial guess
for $k = 1{:}\,K$ **do**
  $\epsilon_0 \leftarrow \frac{1}{10} P_k(\bar{\boldsymbol{u}}_e^{\text{low},n+1})$
  Find θ by solving (6.15) or by using (6.16) if P_k is concave
  $\overline{\boldsymbol{S}}_e \leftarrow \theta\,\overline{\boldsymbol{s}}_e + (1-\theta)\,\overline{\boldsymbol{S}}_e$
  $\bar{\boldsymbol{u}}_e^{\boldsymbol{S}} \leftarrow \bar{\boldsymbol{u}}_e^n + 2\,\Delta t\,\overline{\boldsymbol{S}}_e$
end for

After replacing $\boldsymbol{S}^{\text{LW}}$ by $\boldsymbol{S}$ obtained from Algorithm 6.1 in (6.13), we will have $\bar{\boldsymbol{u}}^{\boldsymbol{S}} \in \mathcal{U}_{\text{ad}}$ and since $\boldsymbol{F}$ has been corrected to ensure $\bar{\boldsymbol{u}}_e^{\boldsymbol{F}} \in \mathcal{U}_{\text{ad}}$ following Section 6.3.1, we will also have $\bar{\boldsymbol{u}}_e^{n+1} \in \mathcal{U}_{\text{ad}}$. Thus, we have an admissibility preserving in means LWFR scheme (5.2) even in the presence of source terms. Then, the scaling limiter of [205] (Appendix F) will be used to obtain an admissibility preserving scheme.

6.1. In the experiments we conducted, the CFL restriction used in Chapter 5 preserved admissibility.

6.4. NUMERICAL RESULTS

The numerical tests for admissibility preservation with 2-D Euler's equations in Chapter 5 were repeated with the generalized admissbility enforcing procedure of Section 6.3.1 and it was seen that admissibility of numerical solution was preserved in all test cases. For further numerical verification of admissibility preserving flux limiter (Section 6.3.1) and for validation of admissibility of LWFR with source terms (Section 6.3.2), we test our scheme with the Ten Moment equations [118] which we describe here. Here, the energy tensor is defined by the ideal equation of state $\boldsymbol{E} = \frac{1}{2}\,\boldsymbol{p} + \frac{1}{2}\,\rho\,\boldsymbol{v} \otimes \boldsymbol{v}$ where ρ is the density, $\boldsymbol{v}$ is the velocity vector, $\boldsymbol{p}$ is the symmetric pressure tensor. Thus, we can define the 2-D conservation law with source terms

$$\partial_t \boldsymbol{u} + \partial_{x_1} \boldsymbol{f}_1 + \partial_{x_2} \boldsymbol{f}_2 = \boldsymbol{s}^{x_1}(\boldsymbol{u}) + \boldsymbol{s}^{x_2}(\boldsymbol{u})$$

where $\boldsymbol{u} = (\rho, \rho\,\boldsymbol{v}, \boldsymbol{E}) = (\rho, \rho\,v_1, \rho\,v_2, E_{11}, E_{12}, E_{22})$ and

$$\boldsymbol{f}_1 = \begin{bmatrix} \rho\,v_1 \\ p_{11} + \rho\,v_1^2 \\ p_{12} + \rho\,v_1\,v_2 \\ (E_{11} + p_{11})\,v_1 \\ E_{12}\,v_1 + \frac{1}{2}\,(p_{11}\,v_2 + p_{12}\,v_1) \\ E_{22}\,v_1 + p_{12}\,v_2 \end{bmatrix}, \quad \boldsymbol{f}_2 = \begin{bmatrix} \rho\,v_2 \\ p_{12} + \rho\,v_1\,v_2 \\ p_{22} + \rho\,v_2^2 \\ E_{11}\,v_2 + p_{12}\,v_1 \\ E_{12}\,v_2 + \frac{1}{2}\,(p_{12}\,v_2 + p_{22}\,v_1) \\ (E_{22} + p_{22})\,v_2 \end{bmatrix} \tag{6.17}$$

The source terms are given by

$$\boldsymbol{s}^{x_1} = \begin{bmatrix} 0 \\ -\frac{1}{2}\,\rho\,\partial_x W \\ 0 \\ -\frac{1}{2}\,\rho\,v_1\,\partial_x W \\ -\frac{1}{4}\,\rho\,v_2\,\partial_x W \\ 0 \end{bmatrix}, \quad \boldsymbol{s}^{x_2} = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{2}\,\rho\,\partial_y W \\ 0 \\ -\frac{1}{4}\,\rho\,v_1\,\partial_y W \\ -\frac{1}{2}\,\rho\,v_2\,\partial_y W \end{bmatrix} \tag{6.18}$$

where $W(x, y, t)$ is a given function, which models electron quiver energy in the laser [27]. These equations are relevant in many applications, especially related to plasma flows in cases where the *local thermodynamic equilibrium* used to close the Euler equations of compressible flows is not valid, and anisotropic nature of the pressure needs to be accounted for. More details about the significance of these models can be found in [25, 27] and further references in [124]. The admissibility set is given by

$$\mathcal{U}_{\text{ad}} = \{\boldsymbol{u} \in \mathbb{R}^6 | \rho(\boldsymbol{u}) > 0, \quad \boldsymbol{x}^T \boldsymbol{p}(\boldsymbol{u})\,\boldsymbol{x} > 0, \quad \boldsymbol{x} \in \mathbb{R}^2 \setminus \{\boldsymbol{0}\}\}$$

which contains the states $\boldsymbol{u}$ with positive density and positive definite pressure tensor. The positive definiteness of $\boldsymbol{p}$ is equivalent to that $\text{Tr}(\boldsymbol{p}) = p_{11} + p_{22} > 0$ and $\det \boldsymbol{p} = p_{11}p_{22} - p_{12}^2 > 0$. Following the notation of (5.1), the $K = 3$ admissibility constraints P_1, P_2, P_3 are density, Trace($\boldsymbol{p}$), and det($\boldsymbol{p}$). However, although density and trace functions are concave functions of the conserved variables, det ($\boldsymbol{p}$) is not so.

The hyperbolicity of the system without source terms, along with its eigenvalues are presented in Lemma 2.0.2 of [125]. The conditions for admissibiltiy preservation of the forward Euler method for the source terms, which are the basis for the source term limiting described in Section 6.3.2, are Lemma 5.1 of [124]. For completeness, they are stated here.

LEMMA 6.1. ***(Lemma 2.0.2 of [125]).***
The system (6.17) without source terms is hyperbolic for $\boldsymbol{u} \in \mathcal{U}_{\text{ad}}$ and admits the eigenvalues

$$\boldsymbol{v}\cdot\boldsymbol{n}, \quad \boldsymbol{v}\cdot\boldsymbol{n} \pm \sqrt{\frac{3\,(\boldsymbol{p}\cdot\boldsymbol{n})\cdot\boldsymbol{n}}{\rho}}, \quad \boldsymbol{v}\cdot\boldsymbol{n} \pm \sqrt{\frac{(\boldsymbol{p}\cdot\boldsymbol{n})\cdot\boldsymbol{n}}{\rho}}$$

along the unitary vector $\boldsymbol{n}$ (The definition of eigenvalues along a direction $\boldsymbol{n}$ is in the sense of Definition 2.1) where $\rho, \boldsymbol{v}, \boldsymbol{p}$ denote the density, velocity and pressure tensor of (6.17) respectively. The eigenvalue $\boldsymbol{v}\cdot\boldsymbol{n}$ has a multiplicity of two while the rest have a multiplicity of one. The eigenvalues $\boldsymbol{v}\cdot\boldsymbol{n}, \boldsymbol{v}\cdot\boldsymbol{n} \pm \sqrt{\frac{(\boldsymbol{p}\cdot\boldsymbol{n})\cdot\boldsymbol{n}}{\rho}}$ are associated to linearly degenerate fields (2.5). The eigenvalues $\boldsymbol{v}\cdot\boldsymbol{n} \pm \sqrt{\frac{3\,(\boldsymbol{p}\cdot\boldsymbol{n})\cdot\boldsymbol{n}}{\rho}}$ are associated to a genuinely nonlinear field (2.4).

THEOREM 6.2. ***(Lemma 5.1 of [124]).***
Define source term updates in the the two coordinate directions as

$$\begin{aligned} \boldsymbol{u}_{e,\boldsymbol{p}}^{\boldsymbol{s}^{x_1},n+1} &= \boldsymbol{u}_{e,\boldsymbol{p}}^{n} + 2\,\Delta t\, \boldsymbol{s}_{e,\boldsymbol{p}}^{x_1}, & \boldsymbol{s}_{e,\boldsymbol{p}}^{x_1} &= \boldsymbol{s}^{x_1}(\boldsymbol{u}_{e,\boldsymbol{p}}^{n}, \boldsymbol{x}_{e,\boldsymbol{p}}, t^n) \\ \boldsymbol{u}_{e,\boldsymbol{p}}^{\boldsymbol{s}^{x_2},n+1} &= \boldsymbol{u}_{e,\boldsymbol{p}}^{n} + 2\,\Delta t\, \boldsymbol{s}_{e,\boldsymbol{p}}^{x_2}, & \boldsymbol{s}_{e,\boldsymbol{p}}^{x_2} &= \boldsymbol{s}^{x_2}(\boldsymbol{u}_{e,\boldsymbol{p}}^{n}, \boldsymbol{x}_{e,\boldsymbol{p}}, t^n) \end{aligned}$$

for $\boldsymbol{s}^{x_1}, \boldsymbol{s}^{x_2}$ defined in (6.18). Then, for $\boldsymbol{s}_{e,\boldsymbol{p}} = \boldsymbol{s}_{e,\boldsymbol{p}}^{x_1} + \boldsymbol{s}_{e,\boldsymbol{p}}^{x_2}$, the source term update in 2-D can be written as

$$\boldsymbol{u}_{e,\boldsymbol{p}}^{\boldsymbol{s},n+1} = \boldsymbol{u}_{e,\boldsymbol{p}}^{n} + \Delta t\, \boldsymbol{s}_{e,\boldsymbol{p}}^{n} = \frac{1}{2}\,(\boldsymbol{u}_{e,\boldsymbol{p}}^{\boldsymbol{s}^{x_1},n+1} + \boldsymbol{u}_{e,\boldsymbol{p}}^{\boldsymbol{s}^{x_2},n+1}) \tag{6.19}$$

Assume $\boldsymbol{u}_{e,\boldsymbol{p}}^{n} \in \mathcal{U}_{\text{ad}}$. Then, we will have $\boldsymbol{u}_{e,\boldsymbol{p}}^{\boldsymbol{s}^{x_1},n+1} \in \mathcal{U}_{\text{ad}}$ if the the following time step conditions are satisfied

$$\Delta t \le \frac{1}{2}\sqrt{\frac{(p_{11}^{n})_{e,\boldsymbol{p}}}{(\rho^{n})_{e,\boldsymbol{p}}\,((\partial_x W_x^{n})_{e,\boldsymbol{p}})^2}}, \qquad \Delta t \le \frac{1}{2}\sqrt{\frac{(p_{11}^{n})_{e,\boldsymbol{p}}\,(p_{22}^{n})_{e,\boldsymbol{p}} - ((p_{12}^{n})_{e,\boldsymbol{p}})^2}{(\rho^{n})_{e,\boldsymbol{p}}\,(p_{22}^{n})_{e,\boldsymbol{p}}\,((\partial_x W_x^{n})_{e,\boldsymbol{p}})^2}}$$

Similarly, $\boldsymbol{u}_{e,\boldsymbol{p}}^{\boldsymbol{s}^{x_2},n+1} \in \mathcal{U}_{\text{ad}}$ if the following time step conditions are satisfied.

$$\Delta t \le \frac{1}{2}\sqrt{\frac{(p_{22}^{n})_{e,\boldsymbol{p}}}{(\rho^{n})_{e,\boldsymbol{p}}\,((\partial_y W_y^{n})_{e,\boldsymbol{p}})^2}}, \qquad \Delta t \le \frac{1}{2}\sqrt{\frac{(p_{11}^{n})_{e,\boldsymbol{p}}\,(p_{22}^{n})_{e,\boldsymbol{p}} - ((p_{12}^{n})_{e,\boldsymbol{p}})^2}{(\rho^{n})_{e,\boldsymbol{p}}\,(p_{11}^{n})_{e,\boldsymbol{p}}\,((\partial_y W_y^{n})_{e,\boldsymbol{p}})^2}}$$

By (6.19), these time step restrictions will imply $\boldsymbol{u}_{e,\boldsymbol{p}}^{\boldsymbol{s},n+1} \in \mathcal{U}_{\text{ad}}$.

All distinct numerical experiments from [125, 124, 126] were performed and observed to validate the accuracy and robustness of the proposed scheme, but only some are shown here. The experiments were performed both with the TVB limiter used in Chapter 4 and the subcell-based blending scheme developed in Chapter 5. As seen in Chapter 5, the subcell based limiter preserves small scale structures well compared to the TVB

limiter. The use of TVB limiter is only made in this chapter to numerically validate that the flux limiting procedure of Section 6.3.1 preserves admissibility. The results shown are produced with TVB limiter unless specified otherwise.

The developments made in this chapter have been contributed to the package `Tenkai.jl` [17] developed in Chapter 5 and the setup files used for generating the results in this chapter are available in [8].

6.4.1. Convergence test

This is a smooth convergence test from [30] and requires no limiter. The domain is taken to be $\Omega = [-0.5, 0.5]$ and the potential for source terms (6.18) is $W = \sin(2\pi(x-t))$. With periodic boundary conditions, the exact solution is given by

$$\rho(x,t) = 2 + \sin(2\pi(x-t)), \quad v_1(x,t) = 1, \quad v_2(x,t) = 0$$
$$p_{11} = 1.5 + \frac{1}{8}[\cos(4\pi(x-t)) - 8\sin(2\pi(x-t))], \quad p_{12}(x,t) = 0, \quad p_{22}(x,t) = 1$$

The solutions are computed at $t = 0.5$ and the convergence results for variable ρ and p_{11} are shown in Figure 6.1 where optimal convergence rates are seen.

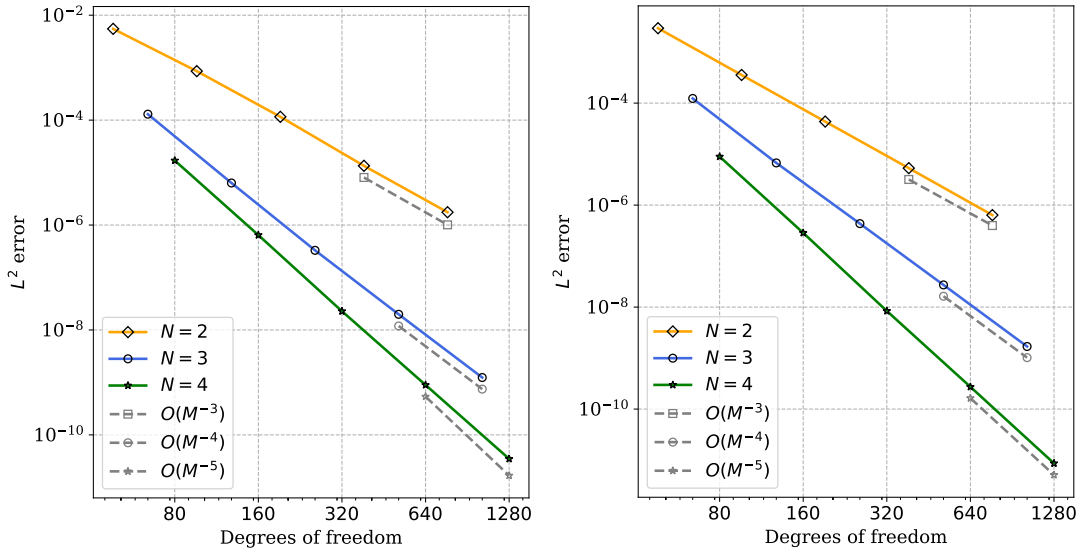


Figure 6.1. Error convergence analysis of a smooth test with source terms for (a) ρ, (b) p_{11} variable

6.4.2. Riemann problems

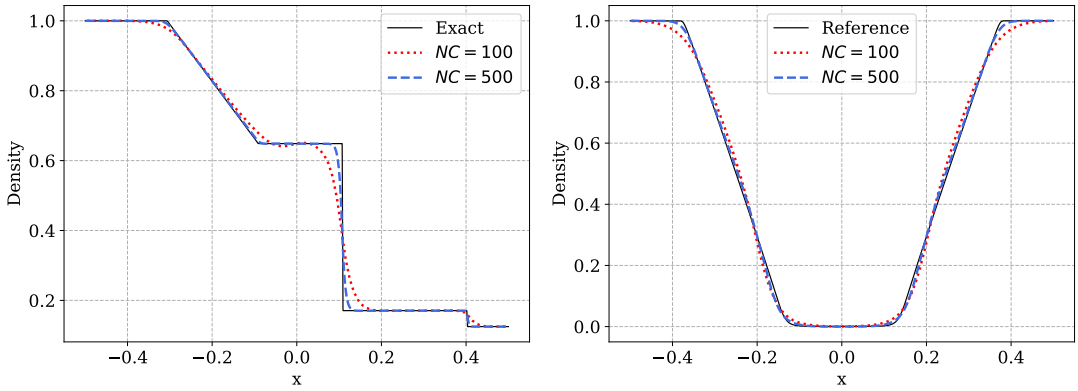
Here, we test the scheme on Riemann problems in the absence of source terms with the TVB limiter. The domain is $\Omega = [-0.5, 0.5]$. The first problem is Sod's test

$$(\rho, v_1, v_2, p_{11}, p_{12}, p_{22}) = \begin{cases} (1,0,0,2,0.05,0.6), & x < 0 \\ (0.125,0,0,0.2,0.1,0.2), & x > 0 \end{cases}$$

The second is a problem from [125] with two rarefaction waves containing both low-density and low-pressure, leading to a near vacuum solution

$$(\rho, v_1, v_2, p_{11}, p_{12}, p_{22}) = \begin{cases} (1,-5,0,2,0,2), & x < 0 \\ (1,5,0,2,0,2), & x > 0 \end{cases}$$

The scheme is able to maintain admissibility in the near vacuum test and the results for both Riemann problems are shown in Figure 6.2 where convergence is seen under grid refinement.



(a) (b)

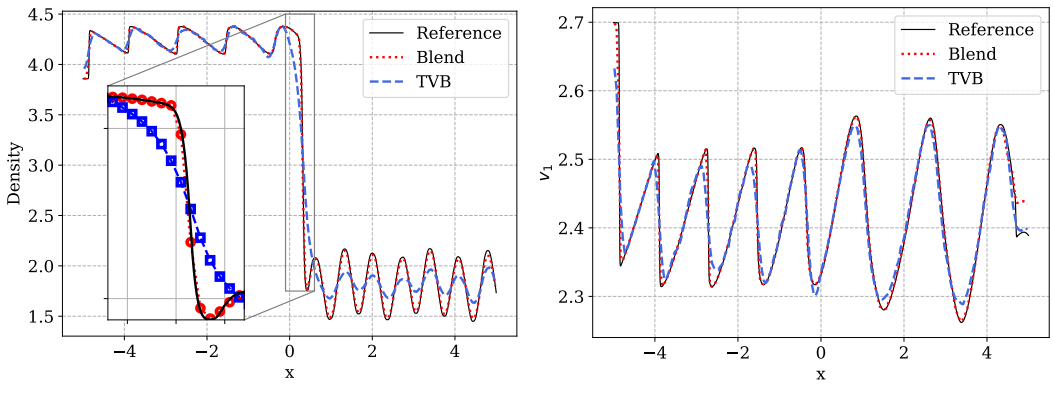
Figure 6.2. Density plots of numerical solutions with polynomial degree $N = 2$ for (a) Sod's problem, (b) Two rarefaction (near vacuum) problem

6.4.3. Shu-Osher test

This is a modified version of the standard Shu-Osher test (Section 4.8.4), taken from [126]. The solution is initialized in domain $[-5, 5]$ in terms of primitive variables as

$$(\rho, v_1, v_2, p_{11}, p_{12}, p_{22}) = \begin{cases} (3.857143, 2.699369, 0, 10.33333, 0, 10.33333), & \text{if } x \leq -4 \\ (1 + 0.2\sin(5x), 0, 0, 1, 0, 1), & \text{if } x > -4 \end{cases}$$

The simulation is performed with polynomial degree $N = 4$ using 200 elements and run till time $t = 1.8$ and the results with both blending and TVB limiter are shown in Figure 6.3 where, as expected, the blending limiter is giving a much better resolution of the shock and high-frequency wave.



(a) (b)

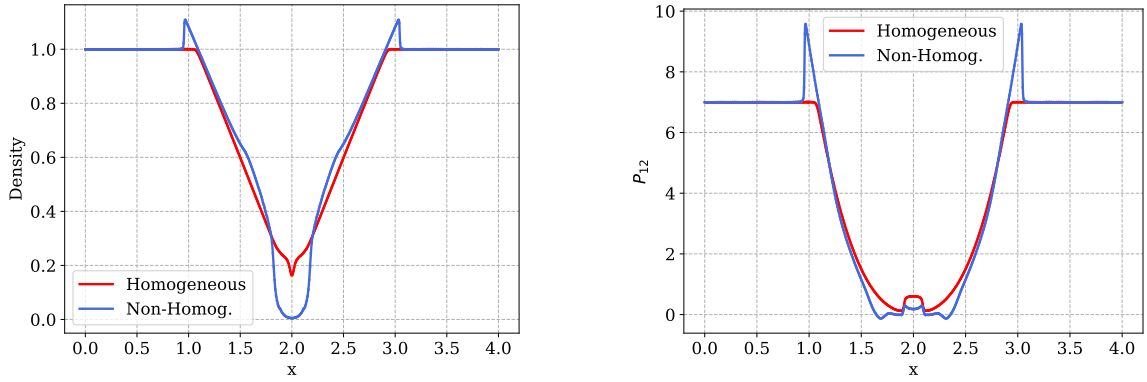
Figure 6.3. Numerical solution for Shu-Osher problem with polynomial degree $N = 4$ using TVB and blending limiter and we show (a) Density, (b) v_1 profiles. The density plot has an inset plot near the shock which compares the number of cells smeared across the shock by blending and TVB limiter.

6.4.4. Two rarefactions with source terms

The Riemann problem is given by

$$(\rho, v_1, v_2, p_{11}, p_{12}, p_{22}) = \begin{cases} (1, -4, 0, 9, 7, 9), & x < 0 \\ (1, 4, 0, 9, 7, 9), & x > 0 \end{cases}$$

with source terms as in (6.18) with $W(x, y, t) = 25 \exp(-200\,(x-2)^2)$. We show the numerical solutions with degree $N = 4$ and 500 elements at $t = 0.1$ in Figure 6.4 with and without the source terms using the blending limiter. The solution with source terms has a near vacuum state at the centre. Thus, this is a test where low density is caused by the presence of source terms verifying that our scheme is able to capture admissibility even in the presence of source terms. The positivity limiter had to be used to maintain admissibility of the solution.



(a) (b)

Figure 6.4. Two rarefactions with source terms using polynomial degree $N = 4$ on a mesh of 500 element at time $t = 0.1$, where we show (a) Density Profile, (b) P_{12}

6.4.5. Two dimensional near vacuum test

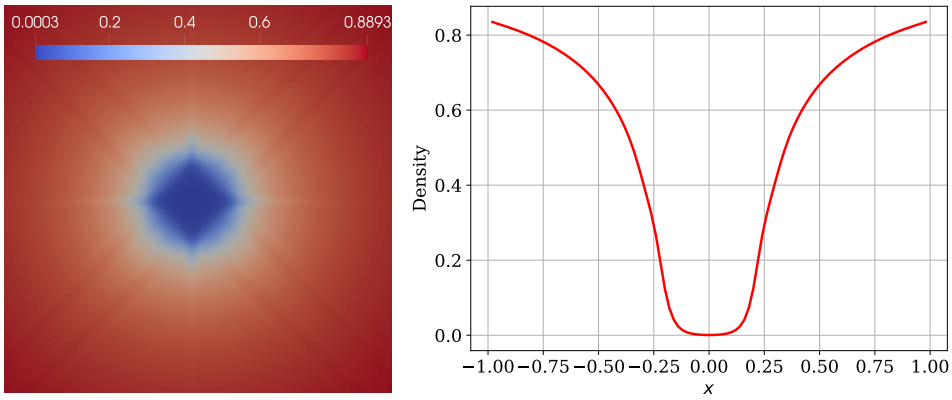
This is a near vacuum test taken from [125] which is simulated using the TVB limiter, and is thus another verification of the admissibility preserving framework of Section 6.3.1. The domain is $\Omega = [-1, 1]^2$ with outflow boundary conditions. The initial conditions are

$$\rho = 1, \quad p_{11} = p_{22} = 1, \quad p_{12} = 0, \quad v_1 = 8\, f_s(r) \cos\theta, \quad v_2 = 8\, f_s(r) \sin\theta$$

where $r = \sqrt{x^2 + y^2}$, $\theta = \arctan(y/x) \in [-\pi, \pi]$ and $s = 0.06\,\Delta x$ for mesh size $\Delta x = \Delta y$ of the uniform mesh. The $f_s(r)$ smoothens the velocity profile near the origin as θ is not defined there

$$f_s(r) = \begin{cases} -2\left(\frac{r}{s}\right)^3 + 3\left(\frac{r}{s}\right)^2, & \text{if } r < s \\ 1, & \text{otherwise} \end{cases}$$

The numerical solution computed using polynomial degree $N = 2$ and 100 elements is shown at the time $t = 0.02$. The results are shown in Figure 6.5 and are similar to those seen in the literature. Since this is a near vacuum problem, the numerical method is not able to maintain admissibility of solution without the positivity limiter.



(a) (b)

Figure 6.5. 2-D near vacuum test. Density plot of numerical solution with degree $N=2$ on a 100^2 element mesh (a) Pseudocolor plot (b) Solution cut along the line $y=0$.

6.4.6. Uniform plasma state with Gaussian source

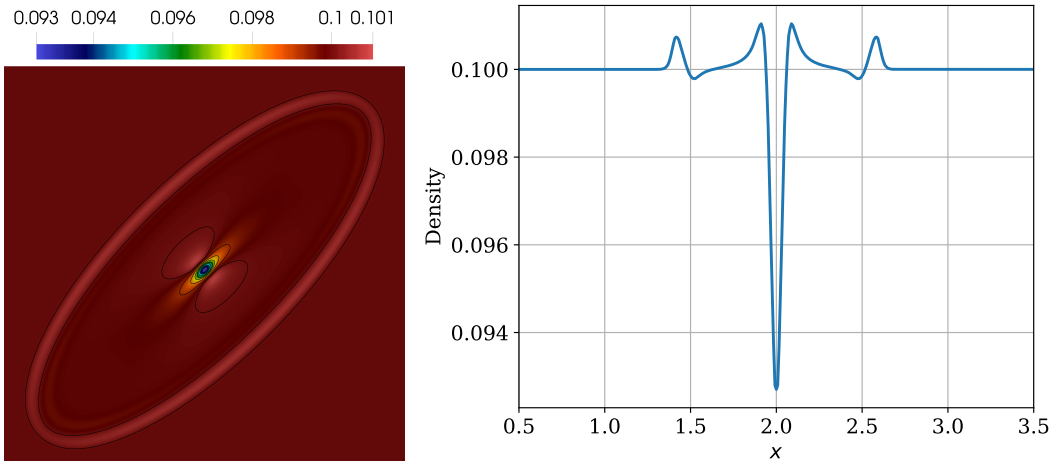
The initial condition is a uniform plasma state characterized by

$$\rho=0.1, \qquad v_1=v_2=0, \qquad p_{11}=p_{22}=9, \qquad p_{12}=7$$

in the domain $\Omega=[0,4]^2$ with outflow boundary conditions and source terms with potential

$$W(x,y,t)=25\exp\left(-200\left((x-2)^2+(y-2)^2\right)\right)$$

Since W depends on both x and y variable, the uniform state will be affected anisotropically. The simulation is run till $t=0.1$ and the density profile is shown in Figure 6.6 with degree $N=2$ on a 200×200 mesh using the blending limiter. In Figure 6.6, we show the line plot across the diagonal.



(a) (b)

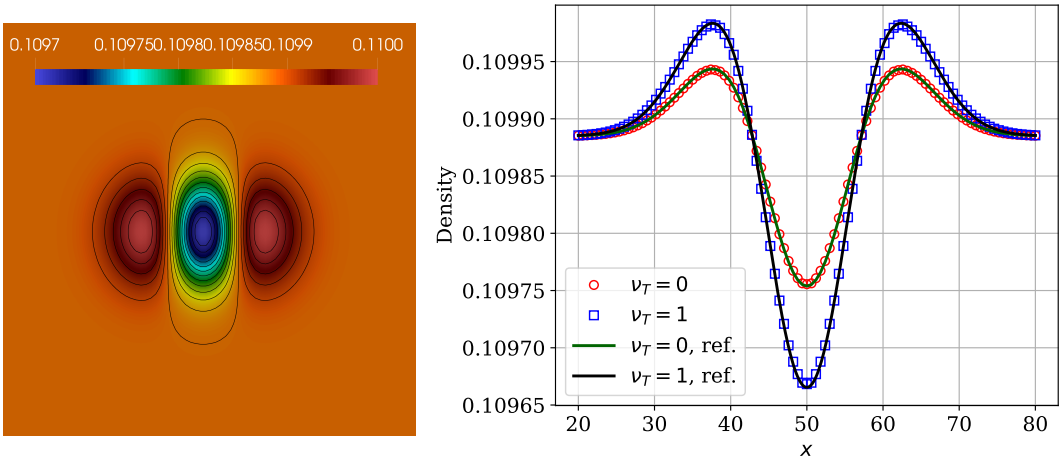
Figure 6.6. Uniform plasma state with Gaussian source (a) Density pseudocolour plot (b) Line plot across $x+y=4$.

6.4.7. Realistic simulation with inverse bremsstrahlung

Consider the domain $\Omega = [0, 100]^2$ with outflow boundary conditions. The uniform initial condition is taken to be

$$\rho = 0.109885, \quad v_1 = v_2 = 0, \quad p_{11} = p_{22} = 1, \quad p_{12} = 0$$

with the electron quiver energy $W(x, y, t) = \exp\left(-0.01\left((x-50)^2 + (y-50)^2\right)\right)$. The source term is taken from [27], and only has the x component, i.e., $\boldsymbol{s}^y(\boldsymbol{u}) = \boldsymbol{0}$, even though W continues to depend on x and y. An additional source corresponding to energy components $\boldsymbol{s}_E = (0, 0, 0, \nu_T \rho W, 0, 0)$ is also added where ν_T is an absorption coefficient. Thus, the source terms are $\boldsymbol{s} = \boldsymbol{s}_x + \boldsymbol{s}_E$. The simulation is run till $t = 0.5$ on a grid of 100 cells. The blending limiter from Chapter 5 was used in this test as it captured the smooth extrema better. The density plot with a cut at $y = 4$ is shown in Figure 6.7.



(a) (b)

Figure 6.7. Realistic simulation. Density profile computed with degree $N = 2$ on 100^2 element mesh. (a) Pseudocolor color plot (b) Cut at $y = 4$ comparing different absorption coefficients ν_T.

6.5. Summary and Conclusions

A generalized framework was developed for the LWFR scheme. The framework can be seen as an extension of [205] to LWFR. It is a generalization of Chapter 5 as it can be used in combination with any limiter for controlling spurious oscillations. As a demonstration of this, results with TVB limiter that is made admissibility preserving were presented, though the best accuracy is obtained with blending limiter. The LWFR scheme was extended to be applicable to problems with source terms while maintaining high order accuracy. Provable admissibility preservation in the presence of source terms was also obtained by limiting the time average sources. The claims were numerically verified on the Ten Moment problem where the scheme showed high order accuracy and robustness.

Chapter 7
Multi-derivative Runge-Kutta

7.1. Introduction

Lax-Wendroff schemes discussed in earlier chapters perform a Taylor's expansion in time to the order of the desired accuracy and compute the temporal derivatives locally. Multiderivative Runge-Kutta (MDRK) schemes also make use of temporal derivatives but they combine them with multiple stages to obtain the desired order of accuracy. As MDRK schemes use both temporal derivatives and multiple stages, they are a generalization of LW and standard multistage RK methods [159]. MDRK methods typically require fewer temporal derivatives in contrast to the Lax-Wendroff schemes and fewer stages in contrast to the standard RK methods, which is what makes them promising. In this chapter, we propose a multiderivative Runge-Kutta scheme in a Flux Reconstruction framework to solve hyperbolic conservation laws (3.1). The idea is to cast the fourth order multi-derivative Runge-Kutta scheme of [119] in the form of

$$\begin{aligned} \boldsymbol{u}^{*} &= \boldsymbol{u}^{n} - \frac{\Delta t}{2}\,\partial_x \boldsymbol{F} \\ \boldsymbol{u}^{n+1} &= \boldsymbol{u}^{n} - \Delta t\,\partial_x \boldsymbol{F}^{*} \end{aligned} \tag{7.1}$$

where

$$\partial_x \boldsymbol{F} = \partial_x \boldsymbol{F}(\boldsymbol{u}^n) \approx \frac{1}{\Delta t/2}\,\partial_x \int_{t^n}^{t^{n+1/2}} \boldsymbol{f}\,\mathrm{d}t, \qquad \partial_x \boldsymbol{F}^{*} = \partial_x \boldsymbol{F}^{*}(\boldsymbol{u}^n, \boldsymbol{u}^{*}) \approx \frac{1}{\Delta t}\,\partial_x \int_{t^n}^{t^{n+1}} \boldsymbol{f}\,\mathrm{d}t$$

The method is two-stage; in the first stage, $\boldsymbol{F}$ is locally approximated and then Flux Reconstruction (FR) (4.8) is used to construct a globally continuous approximation of $\boldsymbol{F}$ which is used to perform evolution to $\boldsymbol{u}^{*}$ (7.1); and the same procedure is then performed using $\boldsymbol{F}^{*}$ for evolution to $\boldsymbol{u}^{n+1}$. The developments of Chapters 4, 5 are applied to each stage of MDRK. In particular, the numerical flux has been constructed with D2 dissipation (4.11) and **EA** scheme (Section 4.3.2) to enhance accuracy and stability. Admissibility preserving blending limiter performing MUSCL-Hancock on subcells is also developed showing good accuracy like in Chapter 5. The scheme is validated with a modern test suite for high order methods [132].

The rest of the chapter is organized as follows. The MDRK scheme in FR framework is introduced in Section 7.2. In particular, Section 7.2.3 discusses the approximate Lax-Wendroff procedure applied to MDRK, Sections 7.2.4 discuss the D2 dissipation for computing the dissipative part of the numerical flux to enhance Fourier CFL stability limit and Section 7.2.6 discusses the **EA** scheme for computing the central part of numerical flux to enhance stability. The Fourier stability analysis is performed in Section 7.3 to demonstrate the improved stability of D2 dissipation. In Section 7.4, we show how the admissibility preserving blending limiter of Chapter 5 applies to the MDRK scheme. The numerical results validating the order of accuracy and capability of the blending scheme are shown in Section 7.5 and a summary of the new MDRK scheme is presented in Section 7.6.

7.2. MULTI-DERIVATIVE RUNGE-KUTTA FR SCHEME

Multiderivative Runge-Kutta [130] methods were initially developed to solve systems of ODE like

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \boldsymbol{L}(\boldsymbol{u}) \tag{7.2}$$

that use temporal derivatives of $\boldsymbol{L}$. They were first used for temporal discretization of hyperbolic conservation laws in [159] by using Weighted Essentially Non-Oscillatary (WENO) [163] and Discontinuous Galerkin [49] methods for spatial discretization.

In this work, we use the two stage fourth order multiderivative Runge-Kutta method from [119]. For the system of ODE (7.2), the MDRK scheme of [119] to evolve from t^n to t^{n+1} is given by

$$\begin{aligned}
\boldsymbol{u}^* &= \boldsymbol{u}^n + \frac{1}{2}\Delta t\, \boldsymbol{L}(\boldsymbol{u}^n) + \frac{\Delta t^2}{8}\frac{\mathrm{d}\boldsymbol{L}}{\mathrm{d}t}(\boldsymbol{u}^n) \\
\boldsymbol{u}^{n+1} &= \boldsymbol{u}^n + \Delta t\, \boldsymbol{L}(\boldsymbol{u}^n) + \frac{\Delta t^2}{6}\left(\frac{\mathrm{d}\boldsymbol{L}}{\mathrm{d}t}(\boldsymbol{u}^n) + 2\frac{\mathrm{d}\boldsymbol{L}}{\mathrm{d}t}(\boldsymbol{u}^*)\right)
\end{aligned}$$

In order to solve the 1-D conservation law (3.1) using the above scheme, we formally set $\boldsymbol{L} = -\boldsymbol{f}(\boldsymbol{u})_x$ to get the following two stage procedure

$$\boldsymbol{u}^* = \boldsymbol{u}^n - \frac{\Delta t}{2}\partial_x \boldsymbol{F} \tag{7.3}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \Delta t\, \partial_x \boldsymbol{F}^* \tag{7.4}$$

where

$$\begin{aligned}
\boldsymbol{F} &:= \boldsymbol{f}(\boldsymbol{u}^n) + \frac{1}{4}\Delta t\frac{\partial}{\partial t}\boldsymbol{f}(\boldsymbol{u}^n) \\
\boldsymbol{F}^* &:= \boldsymbol{f}(\boldsymbol{u}^n) + \frac{1}{6}\Delta t\left(\frac{\partial}{\partial t}\boldsymbol{f}(\boldsymbol{u}^n) + 2\frac{\partial}{\partial t}\boldsymbol{f}(\boldsymbol{u}^*)\right)
\end{aligned} \tag{7.5}$$

The formal order of accuracy of the scheme (Appendix I) is obtained from

$$\partial_x \boldsymbol{F}^* = \frac{1}{\Delta t}\partial_x \int_{t^n}^{t^{n+1}} \boldsymbol{f} + O(\Delta t^4)$$

The idea is to use (7.2 7.3) to obtain solution update at the nodes written as a collocation scheme

$$\begin{aligned}
\boldsymbol{u}^*_{e,p} &= \boldsymbol{u}^n_{e,p} - \frac{\Delta t}{2\,\Delta x_e}\frac{\mathrm{d}\boldsymbol{F}_h}{\mathrm{d}\xi}(\xi_p) \\
\boldsymbol{u}^{n+1}_{e,p} &= \boldsymbol{u}^n_{e,p} - \frac{\Delta t}{\Delta x_e}\frac{\mathrm{d}\boldsymbol{F}^*_h}{\mathrm{d}\xi}(\xi_p)
\end{aligned}, \qquad 0 \le p \le N \tag{7.6}$$

where we take $N = 3$ to get fourth order accuracy in both space and time. As was the case for Chapter 4, the major work is in the construction of the time average flux approximations $\boldsymbol{F}_h, \boldsymbol{F}^*_h$ which is explained in subsequent sections.

7.2.1. Conservation property

The computation of correct weak solutions for non-linear conservation laws in the presence of discontinuous solutions requires the use of conservative numerical schemes. In order to see the conservation property of (7.6), multiply each equation by the quadrature weights associated with the solution points and sum over all the points in the e^{th} element,

$$\begin{aligned} \sum_{p=0}^{N} w_p \boldsymbol{u}^*_{e,p} &= \sum_{p=0}^{N} w_p \boldsymbol{u}^n_{e,p} - \frac{\Delta t}{2\,\Delta x_e} \sum_{p=0}^{N} w_p \frac{\partial \boldsymbol{F}_h}{\partial \xi}(\xi_p) \\ \sum_{p=0}^{N} w_p \boldsymbol{u}^{n+1}_{e,p} &= \sum_{p=0}^{N} w_p \boldsymbol{u}^n_{e,p} - \frac{\Delta t}{\Delta x_e} \sum_{p=0}^{N} w_p \frac{\partial \boldsymbol{F}^*_h}{\partial \xi}(\xi_p) \end{aligned} \tag{7.7}$$

The correction functions are of degree $N+1$ and thus the fluxes $\boldsymbol{F}_h, \boldsymbol{F}^*_h$ are polynomials of degree $\leq N+1$. If the quadrature is exact for polynomials of degree at least N, which is true for both GLL and GL points, then the quadrature is exact for the flux derivative term and we can write it as an integral, which leads to

$$\begin{aligned} \int_{\Omega_e} \boldsymbol{u}^*_h \,\mathrm{d}x &= \int_{\Omega_e} \boldsymbol{u}^n_h \,\mathrm{d}x - \frac{\Delta t}{2} \left[\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}}\right] \\ \int_{\Omega_e} \boldsymbol{u}^{n+1}_h \,\mathrm{d}x &= \int_{\Omega_e} \boldsymbol{u}^n_h \,\mathrm{d}x - \Delta t \left[\boldsymbol{F}^*_{e+\frac{1}{2}} - \boldsymbol{F}^*_{e-\frac{1}{2}}\right] \end{aligned} \tag{7.8}$$

This shows that the total mass inside the cell changes only due to the boundary fluxes and the scheme is hence conservative. The conservation property is crucial in the proof of admissibility preservation studied in Section 5.5.

7.2.2. Reconstruction of the time average flux

To complete the description of the MDRK method (7.6), we must explain the method for the computation of the time average fluxes $\boldsymbol{F}_h, \boldsymbol{F}^*_h$ when evolving from t^n to t^{n+1}. In the first stage (7.2), we compute $\boldsymbol{F}_h$ which is then used to evolve to $\boldsymbol{u}^*$. In the second stage (7.3), $\boldsymbol{u}^n, \boldsymbol{u}^*$ are used to compute $\boldsymbol{F}^*_h$ which is used for evolution to $\boldsymbol{u}^{n+1}$. The procedure for both $\boldsymbol{F}_h, \boldsymbol{F}^*_h$ is the same, and is in fact the same as Steps 1-4 in Section 4.2.2. The procedure is not fully described but for readability, we briefly mention that the steps are the following.

1. Approximate Lax-Wendroff procedure (Section 4.2.4) to approximate time average fluxes $\boldsymbol{F}_h, \boldsymbol{F}^*_h$ at all solution points.
2. Use Lagrange interpolation to construct discontinuous time average flux approximations $\boldsymbol{F}^\delta_h, \boldsymbol{F}^{*\delta}_h$ (4.7).
3. Use FR correction functions g_L, g_R (3.18) to construct to continuous time average fluxes $\boldsymbol{F}_h, \boldsymbol{F}^*_h$ (4.8).
4. Plug continuous fluxes $\boldsymbol{F}_h, \boldsymbol{F}^*_h$ into (7.6) to get an LWFR scheme using matrix vector operations (4.9).

7.2.3. Approximate Lax-Wendroff procedure

The time average fluxes $\boldsymbol{F}_p, \boldsymbol{F}_p^*$ must be computed using (7.5), which involves temporal derivatives of the flux. The approximate Lax-Wendroff is used due to its advantages discussed in Section 4.2.4. To present this idea in a concise and efficient form, we recall the notation

$$\boldsymbol{u}^{(1)} = \Delta t\, \partial_t \boldsymbol{u}, \qquad \boldsymbol{f}^{(1)} = \Delta t\, \partial_t \boldsymbol{f}$$

The time derivatives of the solution are computed using the PDE

$$\boldsymbol{u}^{(1)} = -\Delta t\, \partial_x \boldsymbol{f}$$

The approximate Lax-Wendroff procedure is applied in each element and so for simplicity of notation, we do not show the element index in the following. The vector f below contains the flux values at solution points.

First stage.

$$\boldsymbol{F} := \boldsymbol{f}(\boldsymbol{u}^n) + \frac{1}{4}\Delta t \frac{\partial}{\partial t}\boldsymbol{f}(\boldsymbol{u}^n) \approx \frac{1}{\Delta t/2}\int_{t^n}^{t^{n+1/2}} \boldsymbol{f}(\boldsymbol{u})\,\mathrm{d}t \tag{7.9}$$

To obtain fourth order accuracy, the approximation for $\frac{\partial}{\partial t}\boldsymbol{f}(\boldsymbol{u}^n)$ needs to be third order accurate (Appendix I) which we obtain as

$$\begin{aligned}
&\boldsymbol{f}_t(\xi,t)\\
&\approx \frac{-\boldsymbol{f}(\boldsymbol{u}(\xi,t+2\,\Delta t)) + 8\,\boldsymbol{f}(\boldsymbol{u}(\xi,t+\Delta t)) - 8\,\boldsymbol{f}(\boldsymbol{u}(\xi,t-\Delta t)) + \boldsymbol{f}(\boldsymbol{u}(\xi,t-2\,\Delta t))}{12\,\Delta t}\\
&\approx \left.\frac{-\boldsymbol{f}(\boldsymbol{u}+2\,\Delta t\,\boldsymbol{u}_t) + 8\,\boldsymbol{f}(\boldsymbol{u}+\Delta t\,\boldsymbol{u}_t) - 8\,\boldsymbol{f}(\boldsymbol{u}-\Delta t\,\boldsymbol{u}_t) + \boldsymbol{f}(\boldsymbol{u}-2\,\Delta t\,\boldsymbol{u}_t)}{12\,\Delta t}\right|_{(\xi,t)}
\end{aligned}$$

Thus, the time averaged flux is computed as

$$\mathsf{F} = \mathsf{f} + \frac{1}{4}\,\mathsf{f}^{(1)}$$

where

$$\begin{aligned}
\mathsf{u}^{(1)} &= -\frac{\Delta t}{\Delta x_e}\,\mathsf{D}\mathsf{f}\\
\mathsf{f}^{(1)} &= \frac{1}{12}\left[-\boldsymbol{f}(\mathsf{u}+2\,\mathsf{u}^{(1)}) + 8\,\boldsymbol{f}(\mathsf{u}+\mathsf{u}^{(1)}) - 8\,\boldsymbol{f}(\mathsf{u}-\mathsf{u}^{(1)}) + \boldsymbol{f}(\mathsf{u}-2\,\mathsf{u}^{(1)})\right]
\end{aligned}$$

Second stage.

The time averaged flux is computed as

$$\mathsf{F}^* = \mathsf{f} + \frac{1}{6}\,(\mathsf{f}^{(1)} + 2\,\mathsf{f}^{*(1)})$$

where

$$\begin{aligned}
\mathsf{u}^{*(1)} &= -\frac{\Delta t}{\Delta x_e}\,\mathsf{D}\mathsf{f}^*\\
\mathsf{f}^{*(1)} &= \frac{1}{12}\left[-\boldsymbol{f}(\mathsf{u}^*+2\,\mathsf{u}^{*(1)}) + 8\,\boldsymbol{f}(\mathsf{u}^*+\mathsf{u}^{*(1)}) - 8\,\boldsymbol{f}(\mathsf{u}^*-\mathsf{u}^{*(1)}) + \boldsymbol{f}(\mathsf{u}^*-2\,\mathsf{u}^{*(1)})\right]
\end{aligned}$$

Remark 7.1. The $\mathsf{f}, \mathsf{f}^{(1)}$ computed in the first stage are reused in the second.

7.2.4. Numerical flux

The numerical flux couples the solution between two neighbouring cells in a discontinuous Galerkin type method. In RK methods, the numerical flux is a function of the trace values of the solution at the faces. In the MDRK scheme, we have constructed time average fluxes at all the solution points inside the element and we want to use this information to compute the time averaged numerical flux at the element faces. The simplest numerical flux is based on Lax-Friedrich type approximation and is similar to the one used for LW [137] and is termed D1 dissipation as in Section 4.3

$$\begin{aligned}\boldsymbol{F}_{e+\frac{1}{2}} &= \frac{1}{2}\left[\boldsymbol{F}^{-}_{e+\frac{1}{2}} + \boldsymbol{F}^{+}_{e+\frac{1}{2}}\right] - \frac{1}{2}\lambda_{e+\frac{1}{2}}\left[\boldsymbol{u}_h(x^{+}_{e+\frac{1}{2}}, t_n) - \boldsymbol{u}_h(x^{-}_{e+\frac{1}{2}}, t_n)\right] \\ \boldsymbol{F}^{*}_{e+\frac{1}{2}} &= \frac{1}{2}\left[\boldsymbol{F}^{*-}_{e+\frac{1}{2}} + \boldsymbol{F}^{*+}_{e+\frac{1}{2}}\right] - \frac{1}{2}\lambda_{e+\frac{1}{2}}\left[\boldsymbol{u}_h(x^{+}_{e+\frac{1}{2}}, t_n) - \boldsymbol{u}_h(x^{-}_{e+\frac{1}{2}}, t_n)\right]\end{aligned} \tag{7.10}$$

which consists of a central flux and a dissipative part. As in Chapter 4, the numerical flux of the form (7.10) leads to somewhat reduced CFL numbers which is experimentally verified and discussed in Section 7.3. The flux (7.10) also lacks the upwind property even for linear advection equation. An alternate form of the numerical flux is obtained by evaluating the dissipation term using the time average solution, leading to the formula similar to D2 dissipation of Section 4.3

$$\begin{aligned}\boldsymbol{F}_{e+\frac{1}{2}} &= \frac{1}{2}\left[\boldsymbol{F}^{-}_{e+\frac{1}{2}} + \boldsymbol{F}^{+}_{e+\frac{1}{2}}\right] - \frac{1}{2}\lambda_{e+\frac{1}{2}}\left[\boldsymbol{U}^{+}_{e+\frac{1}{2}} - \boldsymbol{U}^{-}_{e+\frac{1}{2}}\right] \\ \boldsymbol{F}^{*}_{e+\frac{1}{2}} &= \frac{1}{2}\left[\boldsymbol{F}^{*-}_{e+\frac{1}{2}} + \boldsymbol{F}^{*+}_{e+\frac{1}{2}}\right] - \frac{1}{2}\lambda_{e+\frac{1}{2}}\left[\boldsymbol{U}^{*+}_{e+\frac{1}{2}} - \boldsymbol{U}^{*-}_{e+\frac{1}{2}}\right]\end{aligned} \tag{7.11}$$

where

$$\begin{aligned}\boldsymbol{U} &= \mathsf{u} + \frac{1}{4}\,\mathsf{u}^{(1)} \\ \boldsymbol{U}^{*} &= \mathsf{u} + \frac{1}{6}\left(\mathsf{u}^{(1)} + 2\,\mathsf{u}^{\star(1)}\right)\end{aligned} \tag{7.12}$$

are the time average solutions. Following Chapter 4, we will refer to the above two forms of dissipation as D1 and D2, respectively. The dissipation model D2 is not computationally expensive compared to the D1 model since all the quantities required to compute the time average solutions $\boldsymbol{U}, \boldsymbol{U}^*$ are available during the Lax-Wendroff procedure. It remains to explain how to compute $\boldsymbol{F}^{\pm}_{e+\frac{1}{2}}, \boldsymbol{F}^{*\pm}_{e+\frac{1}{2}}$ appearing in the central part of the numerical flux. There were two ways introduced for Lax-Wendroff in Chapter 4 to compute the central flux, termed **AE** and **EA**. We explain how the two apply to MDRK in the next two subsections.

7.2.5. Numerical flux – average and extrapolate to face (AE)

In each element, the time average fluxes $\boldsymbol{F}^{\delta}_h, \boldsymbol{F}^{*\delta}_h$ corresponding to each stage have been constructed using the Lax-Wendroff procedure. The simplest approximation that can be used for $\boldsymbol{F}^{\pm}_{e+\frac{1}{2}}, \boldsymbol{F}^{*\pm}_{e+\frac{1}{2}}$ in the central part of the numerical flux is to extrapolate the fluxes $\boldsymbol{F}^{\delta}_h, \boldsymbol{F}^{*\delta}_h$to the faces

$$\boldsymbol{F}^{\pm}_{e+\frac{1}{2}}, \boldsymbol{F}^{*\pm}_{e+\frac{1}{2}} = \boldsymbol{F}^{\delta}_h(x^{\pm}_{e+\frac{1}{2}}), \boldsymbol{F}^{*\delta}_h(x^{\pm}_{e+\frac{1}{2}})$$

As in Chapter 4, we will refer to this approach with the abbreviation **AE**. However, as shown in the numerical results, this approximation can lead to sub-optimal convergence rates for some non-linear problems. Hence we propose another method for the computation of the inter-cell flux which overcomes this problem as explained next.

7.2.6. Numerical flux – extrapolate to face and average (EA)

Instead of extrapolating the time average flux from the solution points to the faces, we can instead build the time average flux at the faces directly using the approximate Lax-Wendroff procedure that is used at the solution points. The flux at the faces is constructed after the solution is evolved at all the solution points. In the following equations, α denotes either the left face (L) or the right face (R) of a cell. For $\alpha \in \{L, R\}$, we compute the time average flux at the faces of the e^{th} element by the following steps, where we suppress the element index since all the operations are performed inside one element.

Stage 1.

$$\begin{aligned}
\boldsymbol{u}_\alpha^{\pm} &= \mathsf{V}_\alpha^\top(\mathsf{u} \pm \mathsf{u}^{(1)}) \\
\boldsymbol{f}_\alpha^{(1)} &= \frac{1}{12}\left[-\boldsymbol{f}(\boldsymbol{u}_\alpha^{+2}) + 8\,\boldsymbol{f}(\boldsymbol{u}_\alpha^{+}) - 8\,\boldsymbol{f}(\boldsymbol{u}_\alpha^{-}) + \boldsymbol{f}(\boldsymbol{u}_\alpha^{-2})\right] \\
\boldsymbol{F}_\alpha &= \boldsymbol{f}(\boldsymbol{u}_\alpha) + \frac{1}{4}\,\boldsymbol{f}_\alpha^{(1)}
\end{aligned}$$

Stage 2.

$$\begin{aligned}
\boldsymbol{u}_\alpha^{*\pm} &= \mathsf{V}_\alpha^\top(\mathsf{u}^{\star} \pm \mathsf{u}^{\star(1)}) \\
\boldsymbol{u}_\alpha^{*\pm 2} &= \mathsf{V}_\alpha^\top(\mathsf{u}^{\star} \pm 2\,\mathsf{u}^{\star(1)}) \\
\boldsymbol{f}_\alpha^{*(1)} &= \frac{1}{12}\left[-\boldsymbol{f}(\boldsymbol{u}_\alpha^{*+2}) + 8\,\boldsymbol{f}(\boldsymbol{u}_\alpha^{*+}) - 8\,\boldsymbol{f}(\boldsymbol{u}_\alpha^{*-}) + \boldsymbol{f}(\boldsymbol{u}_\alpha^{*-2})\right] \\
\boldsymbol{F}_\alpha^{*} &= \boldsymbol{f}(\boldsymbol{u}_\alpha) + \frac{1}{6}\,(\boldsymbol{f}_\alpha^{(1)} + 2\,\boldsymbol{f}_\alpha^{*(1)})
\end{aligned}$$

Remark 7.2. The $\boldsymbol{f}(\boldsymbol{u}_\alpha)$, $\boldsymbol{f}_\alpha^{(1)}$ computed in the first stage are reused in the second stage.

We see that the solution is first extrapolated to the cell faces and the same finite difference formulae for the time derivatives of the flux which are used at the solution points, are also used at the faces. The numerical flux is computed using the time average flux built as above at the faces; the central parts of the fluxes $\boldsymbol{F}_{e+\frac{1}{2}}^{\pm}, \boldsymbol{F}_{e+\frac{1}{2}}^{*\pm}$ in equations (7.10), (7.11) are computed as

$$\begin{aligned}
\boldsymbol{F}_{e+\frac{1}{2}}^{-} &= (\boldsymbol{F}_R)_e, \qquad & \boldsymbol{F}_{e+\frac{1}{2}}^{+} &= (\boldsymbol{F}_L)_{e+1} \\
\boldsymbol{F}_{e+\frac{1}{2}}^{*-} &= (\boldsymbol{F}_R^*)_e, \qquad & \boldsymbol{F}_{e+\frac{1}{2}}^{*+} &= (\boldsymbol{F}_R^*)_{e+1}
\end{aligned}$$

We will refer to this method with the abbreviation **EA**.

7.3. Fourier stability analysis

We now perform Fourier stability analysis of the MDRK scheme applied to the linear advection equation, $u_t + a\, u_x = 0$, where a is the constant advection speed. We will assume that the advection speed a is positive and denote the CFL number by

$$\sigma = \frac{a\,\Delta t}{\Delta x}$$

We will perform the stability analysis for the MDRK scheme with D2 dissipation flux (7.11) and thus will like to write the two stage scheme as

$$\mathsf{u}_e^{n+1} = -\sigma^2\,\mathsf{A}_{-2}\mathsf{u}_{e-2} - \sigma\,\mathsf{A}_{-1}\mathsf{u}_{e-1}^n + (1-\sigma\,\mathsf{A}_0)\,\mathsf{u}_e^n - \sigma\,\mathsf{A}_{+1}\mathsf{u}_{e+1}^n - \sigma^2\,\mathsf{A}_{+2}\mathsf{u}_{e+2}^n \tag{7.13}$$

where the matrices $\mathsf{A}_{-2}, \mathsf{A}_{-1}, \mathsf{A}_0, \mathsf{A}_{+1}, \mathsf{A}_{+2}$ depend on the choice of the dissipation model in the numerical flux. We will perform the final evolution by studying both the stages.

7.3.1. Stage 1

We will try to write the first stage as

$$\mathsf{u}_e^* = -\sigma\,\mathsf{A}_{-1}^{(1)}\mathsf{u}_{e-1}^n + (1-\sigma\,\mathsf{A}_0^{(1)})\,\mathsf{u}_e^n - \sigma\,\mathsf{A}_{+1}^{(1)}\mathsf{u}_{e+1}^n \tag{7.14}$$

Since $f_t = a\,u_t$, the time average flux for the first stage at all solution points is given by

$$\mathsf{F}_e = a\,\mathsf{U}_e \quad \text{where} \quad \mathsf{U}_e = \mathsf{T}^{(1)}\mathsf{u}_e \quad \text{and} \quad \mathsf{T}^{(1)} = \mathsf{I} - \frac{\sigma}{4}\,\mathsf{D}$$

The extrapolations to the cell boundaries are given by

$$F_h^\delta(x^+_{e-\frac{1}{2}}) = \mathsf{V}_L^T\mathsf{F}_e, \qquad F_h^\delta(x^+_{e+\frac{1}{2}}) = \mathsf{V}_R^T\mathsf{F}_e$$

The D2 dissipation numerical flux is given by

$$\mathsf{F}_{e+\frac{1}{2}} = \mathsf{V}_R^T\mathsf{F}_e = a\,\mathsf{V}_R^T\mathsf{T}^{(1)}\mathsf{u}_e$$

and the flux differences at the face as

$$F_{e-\frac{1}{2}} - F_h^\delta(x^+_{e-\frac{1}{2}}) = a\,\mathsf{V}_R^T\mathsf{u}_{e-1} - a\,\mathsf{V}_L^T\mathsf{T}^{(1)}\mathsf{u}_e, \qquad F_{e+\frac{1}{2}} - F_h^\delta(x^-_{e+\frac{1}{2}}) = 0$$

so that the flux derivative at the solution points is given by

$$\begin{aligned} \partial_\xi\,\mathsf{F}_h &= \mathsf{b}_L(a\mathsf{V}_R^T\mathsf{T}^{(1)}\mathsf{u}_{e-1} - a\mathsf{V}_L^T\mathsf{T}^{(1)}\mathsf{u}_e) + a\,\mathsf{D}\mathsf{T}^{(1)}\mathsf{u}_e \\ &= a\,\mathsf{b}_L\mathsf{V}_R^T\mathsf{T}^{(1)}\mathsf{u}_{e-1} + a\,(\mathsf{D}\mathsf{T}^{(1)} - \mathsf{b}_L\mathsf{V}_L^T\mathsf{T}^{(1)})\,\mathsf{u}_e \end{aligned}$$

Since the evolution to u^* is given by

$$\mathsf{u}^* = \mathsf{u}^n - \frac{\Delta t/2}{\Delta x_e}\,\partial_\xi\,\mathsf{F}_h \tag{7.15}$$

the matrices in (7.14) are given by

$$\mathsf{A}^{(1)}_{-1}=\frac{1}{2}\,\mathsf{b}_L\mathsf{V}_R^T\mathsf{T}^{(1)}, \qquad \mathsf{A}^{(1)}_{0}=\frac{1}{2}\,(\mathsf{D}\mathsf{T}^{(1)}-\mathsf{b}_L\mathsf{V}_L^T\mathsf{T}^{(1)}), \qquad \mathsf{A}^{(1)}_{+1}=0 \tag{7.16}$$

The upwind character of the D2 dissipation flux leads to $\mathsf{A}^{(1)}_{+1}=0$ and the right cell does not appear in the update equation.

7.3.2. Stage 2

After stage 1, we have $\mathsf{u}^*, \mathsf{u}^n$ and both are used to obtain u^{n+1}. In this case,

$$\mathsf{F}^*_e=a\,\mathsf{U}^*_e, \qquad \mathsf{U}^*_e=\mathsf{u}^n_e-\frac{1}{6}\,\sigma\,\mathsf{D}\mathsf{u}^n_e-\frac{1}{3}\,\sigma\,\mathsf{D}\mathsf{u}^*_e=\mathsf{T}^{(2)}\mathsf{u}^n_e+\mathsf{T}^{(2,*)}\mathsf{u}^*_e$$

where

$$\mathsf{T}^{(2)}=\mathsf{I}-\frac{1}{6}\,\sigma\,\mathsf{D}, \qquad \mathsf{T}^{(2,*)}=-\frac{1}{3}\,\sigma\,\mathsf{D}$$

The numerical fluxes are given by

$$\begin{aligned}
\mathsf{F}^*_{e+\frac{1}{2}} &= \frac{1}{2}\,[\mathsf{V}_R^T\mathsf{F}^*_e+\mathsf{V}_L^T\mathsf{F}^*_{e+1}]-\frac{1}{2}\,a\,(\mathsf{V}_L^T\mathsf{U}^*_{e+1}-\mathsf{V}_R^T\mathsf{U}^*_e)\\
&= \frac{1}{2}\,a\,[\mathsf{V}_R^T\mathsf{U}^*_e+\mathsf{V}_L^T\mathsf{U}^*_{e+1}]-\frac{1}{2}\,a\,(\mathsf{V}_L^T\mathsf{U}^*_{e+1}-\mathsf{V}_R^T\mathsf{U}^*_e)\\
&= a\,\mathsf{V}_R^T\mathsf{U}^*_e\\
\mathsf{F}^*_{e-\frac{1}{2}} &= a\,\mathsf{V}_R^T\mathsf{U}^*_{e-1}
\end{aligned}$$

and the face extrapolations are

$$\begin{aligned}
F^{*\delta}_h(x^-_{e+\frac{1}{2}}) &= \mathsf{V}_R^T\mathsf{F}^*_e = a\,\mathsf{V}_R^T\mathsf{U}^*_e\\
F^{*\delta}_h(x^+_{e-\frac{1}{2}}) &= \mathsf{V}_L^T\mathsf{F}^*_e = a\,\mathsf{V}_L^T\mathsf{U}^*_e
\end{aligned}$$

Thus, the flux difference at the faces is

$$\begin{aligned}
F^*_{e+\frac{1}{2}}-F^{*\delta}_h(x^-_{e+\frac{1}{2}}) &= 0\\
F^*_{e-\frac{1}{2}}-F^{*\delta}_h(x^+_{e-\frac{1}{2}}) &= a\,(\mathsf{V}_R^T\mathsf{U}^*_{e-1}-\mathsf{V}_L^T\mathsf{U}^*_e)
\end{aligned} \tag{7.17}$$

the flux derivative at the solution points is given by

$$\begin{aligned}
\partial_\xi\,\mathsf{F}^*_h &= a\,\mathsf{D}\mathsf{U}^*_e+a\,\mathsf{b}_L(\mathsf{V}_R^T\mathsf{U}^*_{e-1}-\mathsf{V}_L^T\mathsf{U}^*_e)\\
&= a\,\mathsf{b}_L\mathsf{V}_R^T\mathsf{U}^*_{e-1}+a\,(\mathsf{D}-\mathsf{b}_L\mathsf{V}_L^T)\,\mathsf{U}^*_e
\end{aligned} \tag{7.18}$$

We now expand U^*_e in terms of u^n_e as follows

$$\mathsf{U}^2_e=\mathsf{T}^{(2)}\mathsf{u}^n_e+\mathsf{T}^{(2,*)}\mathsf{u}^*_e$$

where

$$\mathsf{T}^{(2)}=\mathsf{I}-\frac{1}{6}\,\sigma\,\mathsf{D}, \qquad \mathsf{T}^{(2,*)}=-\frac{1}{3}\,\sigma\,\mathsf{D}$$

Thus, by

$$\mathsf{u}^{n+1}=\mathsf{u}^n-\frac{\Delta t}{\Delta x_e}\,\partial_\xi\,\mathsf{F}_h^*$$

and also expanding u^* from (7.15), the matrices in (7.13) are given by

$$\begin{aligned}
\mathsf{A}_{-2} &= -\mathsf{b}_L\mathsf{V}_R^T\mathsf{T}^{(2,*)}\mathsf{A}_{-1}^{(1)}\\
\mathsf{A}_{-1} &= \mathsf{b}_L\mathsf{V}_R^T(\mathsf{T}^{(2)}+\mathsf{T}^{(2,*)}(1-\sigma\,\mathsf{A}_0^{(1)}))-\sigma\,(\mathsf{D}-\mathsf{b}_L\mathsf{V}_L^T)\,\mathsf{T}^{(2,*)}\mathsf{A}_{-1}^{(1)}\\
\mathsf{A}_0 &= -(\mathsf{D}-\mathsf{b}_L\mathsf{V}_L^T)\,(\mathsf{T}^{(2)}+\mathsf{T}^{(2,*)}(\mathsf{I}-\sigma\,\mathsf{A}_0^{(1)}))\\
\mathsf{A}_{+1}=\mathsf{A}_{+2} &= 0
\end{aligned}$$

where $\mathsf{A}_i^{(1)}$ are defined in (7.16). The upwind character of D2 flux is the reason why we have $\mathsf{A}_{+1}=\mathsf{A}_{+2}=0$.

Stability analysis. We assume a solution of the form $\mathsf{u}_e^n=\hat{\mathsf{u}}_k^n\exp(i\,k\,x_e)$ where $i=\sqrt{-1}$, k is the wave number which is an integer and $\hat{\mathsf{u}}_k^n\in\mathbb{R}^{N+1}$ are the Fourier amplitudes; substituting this ansatz in the update equation (7.13), we get

$$\hat{\mathsf{u}}_k^{n+1}=H(\sigma,k)\,\hat{\mathsf{u}}_k^n$$

where

$$\mathsf{H}=-\sigma^2\,\mathsf{A}_{-2}\exp(-2\,i\,\kappa)-\sigma\,\mathsf{A}_{-1}\exp(-i\,\kappa)+\mathsf{I}-\sigma\,\mathsf{A}_0-\sigma\,\mathsf{A}_{+1}\exp(i\,\kappa)-\sigma^2\,\mathsf{A}_{+2}\exp(2\,i\,\kappa)$$

and $\kappa=k\,\Delta x$ is the non-dimensional wave number. The explicit expression of H is then used to numerically compute the CFL number as in Section 4.4. The results of this numerical investigation of stability are shown in Table 7.1 for two correction functions with polynomial degree $N=3$. The comparison is made with CFL numbers of MDRK-D1 (7.10) which are experimentally obtained from the linear advection test case (Section 7.5.1.1), i.e., using time step size that is slightly larger than these numbers causes the solution to blow up.

Correction	D1 (Experimentally obtained)	D2	$\frac{\text{D2}}{\text{D1}}$	LW-D2 ($N=3$)	$\frac{\text{MDRK-D2}}{\text{LW-D2}}$
Radau	≈0.09	0.107	1.19	0.103	1.04
g_2	≈0.16	0.224	1.4	0.170	1.31

Table 7.1. CFL numbers for MDRK scheme with Radau and g_2 correction functions.

We see that dissipation model D2 has a higher CFL number compared to dissipation model D1. The CFL numbers for the g_2 correction function are also significantly higher than those for the Radau correction function. The MDRK scheme also has higher CFL numbers than the single stage LW method for degree $N=3$, which is especially true with the g_2 correction function. The optimality of these CFL numbers has been verified by experiment on the linear advection test case (Section 7.5.1.1), i.e., the solution eventually blows up if the time step is slightly higher than what is allowed by the CFL condition.

7.4. BLENDING SCHEME

The MDRK scheme (7.6) gives a high (fourth) order method for smooth problems, but most practical problems involving hyperbolic conservation laws consist of non-smooth solutions like shocks. Thus, we develop the blending scheme used for LWFR from Section 5.3.1 for the MDRK scheme. The idea is to apply the limiter at each MDRK stage.

Let us write the MDRK update equation (7.6)

$$\mathsf{u}_e^{H,*} = \mathsf{u}_e^n - \frac{\Delta t/2}{\Delta x_e}\,\mathsf{R}_e^H, \qquad \mathsf{u}_e^{H,n+1} = \mathsf{u}_e^n - \frac{\Delta t}{\Delta x_e}\,\mathsf{R}_e^{*,H} \tag{7.19}$$

where u_e is the vector of nodal values in the element. We use the lower order schemes as

$$\mathsf{u}_e^{L,*} = \mathsf{u}_e^n - \frac{\Delta t/2}{\Delta x_e}\,\mathsf{R}_e^L, \qquad \mathsf{u}_e^{L,n+1} = \mathsf{u}_e^n - \frac{\Delta t}{\Delta x_e}\,\mathsf{R}_e^{*,L} \tag{7.20}$$

Then the two-stage blended scheme is given by

$$\begin{aligned} \mathsf{u}_e^* &= (1-\alpha_e)\,\mathsf{u}_e^{H,*} + \alpha_e\,\mathsf{u}_e^{L,*} = \mathsf{u}_e^n - \frac{\Delta t/2}{\Delta x_e}\left[(1-\alpha_e)\,\mathsf{R}_e^H + \alpha_e\,\mathsf{R}_e^L\right] \\ \mathsf{u}_e^{n+1} &= (1-\alpha_e)\,\mathsf{u}_e^{H,n+1} + \alpha_e\,\mathsf{u}_e^{L,n+1} = \mathsf{u}_e^n - \frac{\Delta t}{\Delta x_e}\left[(1-\alpha_e)\,\mathsf{R}_e^{*,H} + \alpha_e\,\mathsf{R}_e^{*,L}\right] \end{aligned} \tag{7.21}$$

where $\alpha_e \in [0,1]$ must be chosen based on the local smoothness indicator of Section 5.3.2. As in Section 5.3.1, if $\alpha_e = 0$ then we obtain the high order MDRK scheme, while if $\alpha_e = 1$ then the scheme becomes the low order scheme that is less oscillatory. The lower order scheme will either be a first order finite volume scheme (Section 5.3.3) or a high resolution scheme based on MUSCL-Hancock idea (Section 5.4). In either case, the common structure of the low order scheme at each stage will be the same as in Section 5.3.1. However, there is one thing that we would like to clarify in the structure of the lower order method (7.20). In the first stage, the lower order residual R_e^L performs evolution from time t^n to $t^{n+\frac{1}{2}}$ while, in the second stage, $\mathsf{R}_e^{*,L}$ performs evolution from t^n to t^{n+1}. Intuition may suggest evolving from $t^{n+\frac{1}{2}}$ to t^{n+1} in the next stage, but that will violate the conservation property because of the expression of second stage of MDRK (7.3, 7.19).

Note that the subcells will be the same as in the single stage LWFR scheme, see Figure 5.1. Since the lower order scheme for the second stage is an evolution from t^n to t^{n+1}, its explanation will be exactly the same as in Section 5.3.1. With a slight modification, we will obtain the lower order scheme used in the first stage, but we write it here for clarity. The low order scheme is obtained by updating the solution in each of the subcells by a finite volume scheme,

$$\begin{aligned} \boldsymbol{u}_{e,0}^{L,*} &= \boldsymbol{u}_{e,0}^n - \frac{\Delta t/2}{w_0\,\Delta x_e}\left[\boldsymbol{f}_{\frac{1}{2}}^e - \boldsymbol{F}_{e-\frac{1}{2}}\right] \\ \boldsymbol{u}_{e,p}^{L,*} &= \boldsymbol{u}_{e,p}^n - \frac{\Delta t/2}{w_p\,\Delta x_e}\left[\boldsymbol{f}_{p+\frac{1}{2}}^e - \boldsymbol{f}_{p-\frac{1}{2}}^e\right], \qquad 1 \le p \le N-1 \\ \boldsymbol{u}_{e,N}^{L,*} &= \boldsymbol{u}_{e,N}^n - \frac{\Delta t/2}{w_N\,\Delta x_e}\left[\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{f}_{N-\frac{1}{2}}^e\right] \end{aligned} \tag{7.22}$$

The inter-element fluxes $\boldsymbol{F}_{e+\frac{1}{2}}$ used in the low order scheme are same as those used in the high order MDRK scheme in equation (7.5). As in Chapter 5, Rusanov's flux [152] will be used for the inter-element fluxes and in the lower order scheme. The element mean value obtained by the low order scheme satisfies

$$\bar{\boldsymbol{u}}_e^{L,*} = \sum_{p=0}^{N} \boldsymbol{u}_{e,p}^{L,*} w_p = \bar{\boldsymbol{u}}_e^n - \frac{\Delta t/2}{\Delta x_e} \left(\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}} \right) \tag{7.23}$$

which is identical to the update equation by the MDRK scheme given in equation (7.8). The element mean in the blended scheme evolves according to

$$\begin{aligned}
\bar{\boldsymbol{u}}_e^* &= (1-\alpha_e) \left(\bar{\boldsymbol{u}}_e \right)^{H,*} + \alpha_e \left(\bar{\boldsymbol{u}}_e \right)^{L,*} \\
&= (1-\alpha_e) \left[\bar{\boldsymbol{u}}_e^n - \frac{\Delta t/2}{\Delta x_e} \left(\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}} \right) \right] + \alpha_e \left[\bar{\boldsymbol{u}}_e^n - \frac{\Delta t/2}{\Delta x_e} \left(\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}} \right) \right] \\
&= \bar{\boldsymbol{u}}_e^n - \frac{\Delta t/2}{\Delta x_e} \left(\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}} \right)
\end{aligned}$$

and hence the blended scheme is also conservative. The similar arguments will apply to the second stage, where the lower order scheme is as described in Section 5.3.1, and we will have by (7.8)

$$\begin{aligned}
\bar{\boldsymbol{u}}_e^* &= \bar{\boldsymbol{u}}_e^n - \frac{\Delta t/2}{\Delta x_e} \left(\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{F}_{e-\frac{1}{2}} \right) \\
\bar{\boldsymbol{u}}_e^{n+1} &= \bar{\boldsymbol{u}}_e^n - \frac{\Delta t}{\Delta x_e} \left(\boldsymbol{F}_{e+\frac{1}{2}}^* - \boldsymbol{F}_{e-\frac{1}{2}}^* \right)
\end{aligned}$$

Overall, all three schemes, i.e., lower order, MDRK and the blended scheme, predict the same mean value.

The inter-element fluxes $\boldsymbol{F}_{e+\frac{1}{2}}, \boldsymbol{F}_{e+\frac{1}{2}}^*$ are used both in the low and high order schemes. To achieve high order accuracy in smooth regions, this flux needs to be high order accurate, however it may produce spurious oscillations near discontinuities when used in the low order scheme. A natural choice to balance accuracy and oscillations is to take

$$\begin{aligned}
\boldsymbol{F}_{e+\frac{1}{2}} &= (1-\alpha_{e+\frac{1}{2}}) \, \boldsymbol{F}_{e+\frac{1}{2}}^{\mathrm{HO}} + \alpha_{e+\frac{1}{2}} \, \boldsymbol{f}_{e+\frac{1}{2}} \\
\boldsymbol{F}_{e+\frac{1}{2}}^* &= (1-\alpha_{e+\frac{1}{2}}) \, \boldsymbol{F}_{e+\frac{1}{2}}^{\mathrm{HO}*} + \alpha_{e+\frac{1}{2}} \, \boldsymbol{f}_{e+\frac{1}{2}}
\end{aligned} \tag{7.24}$$

where $\boldsymbol{F}_{e+\frac{1}{2}}^{\mathrm{HO}}, \boldsymbol{F}_{e+\frac{1}{2}}^{\mathrm{HO}*}$ are the high order inter-element time-averaged numerical fluxes used in the MDRK scheme (7.11) and $\boldsymbol{f}_{e+\frac{1}{2}}$ is a lower order flux at the face $x_{e+\frac{1}{2}}$ shared between FR elements and subcells (5.14, 5.20). The construction of specific lower order schemes as first order (Section 5.3.3) or MUSCL-Hancock (Section 5.4) remains as in Chapter 5, and the same goes for flux limiting of (7.24) to enforce admissibility in means (Definition 5.2). Once admissibility preservation in means is obtained, the scaling limiter of [205] (Appendix F), is used to obtain an admissibility preserving scheme (Definition 5.1).

7.5. NUMERICAL RESULTS

In this section, we test the MDRK scheme with numerical experiments using polynomial degree $N=3$ in all results. Most of the test cases from the previous chapters were tried and were seen to validate our claims, but we only show the important results here. We also tested all the benchmark problems for higher order methods in [132], and show some of the results from there.

7.5.1. Scalar equations

We perform convergence tests with scalar equations. The MDRK scheme with D1 and D2 dissipation is tested using the optimal CFL numbers from Table 7.1. We make a comparison with RKFR scheme with polynomial degree $N=3$ described in Section 3.4 using the SSPRK scheme from [167]. The CFL number for the fourth order RK scheme is taken from [76]. In many problems, we compare with Gauss-Legendre (GL) solution points and Radau correction functions, and Gauss-Legendre-Lobatto (GLL) solution points with g_2 correction functions. These combinations are important because they are both variants of Discontinuous Galerkin methods [94, 57] (Appendix B).

7.5.1.1. Linear advection equation

The initial condition $u(x,0)=\sin(2\pi x)$ is taken with periodic boundaries on $[0,1]$. The error norms are computed at time $t=2$ units and shown in Figure 7.1. The MDRK scheme with D2 dissipation (MDRK-D2) scheme shows optimal order of convergence and has errors close to that MDRK-D1 and the RK scheme for all the combinations of solution points and correction functions.

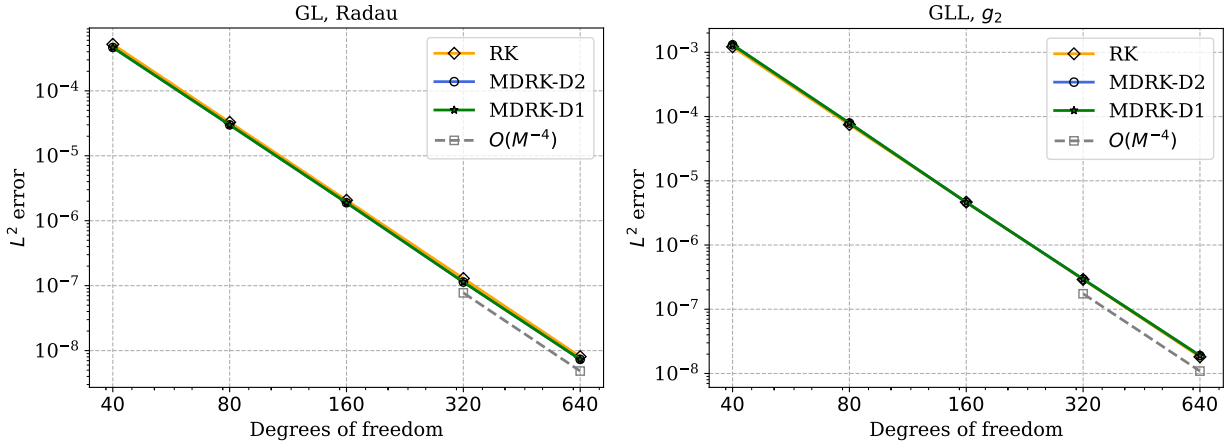


Figure 7.1. Error convergence for constant linear advection equation comparing MDRK and RK - (a) GL points with Radau correction, (b) GLL points with g_2 correction

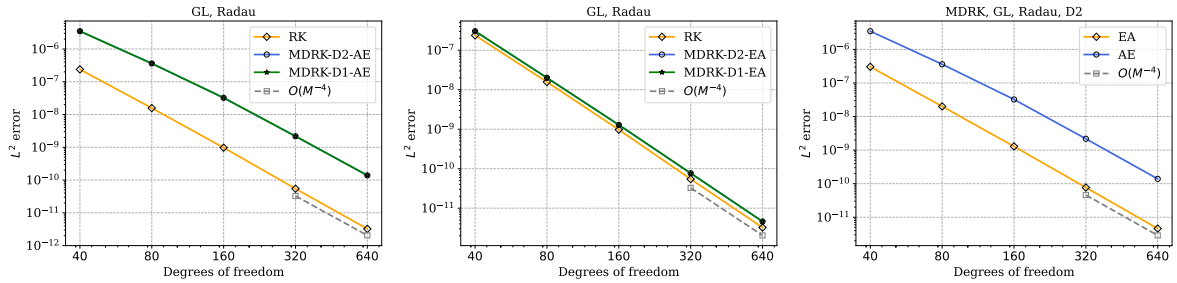
7.5.1.2. Variable advection equation

$$u_t + f(x,u)_x = 0, \qquad f(x,u) = a(x)\,u$$

The velocity is $a(x)=x^2$, $u_0(x)=\cos(\pi\,x/2)$, $x\in[0.1,1]$ and the exact solution is $u(x,t)=u_0(x/(1+t\,x))/(1+t\,x)^2$. Dirichlet boundary conditions are imposed on the left boundary and outflow boundary conditions on the right. This problem is non-linear in the spatial variable, i.e., if I_h is the interpolation operator, $I_h(a\,u_h)\neq I_h(a)\,I_h(u_h)$.

Thus, the **AE** and **EA** schemes are expected to show different behavior, unlike the previous test.

The grid convergence analysis is shown in Figure 7.2. In Figure 7.2a, the scheme with **AE** shows larger errors compared to the RK scheme though the convergence rate is optimal. The MDRK scheme with **EA** shown in the middle figure, is as accurate as the RK scheme. The last figure compares **AE** and **EA** schemes using GL solution points, Radau correction function and D2 dissipation; we clearly see that **EA** scheme has smaller errors than **AE** scheme.

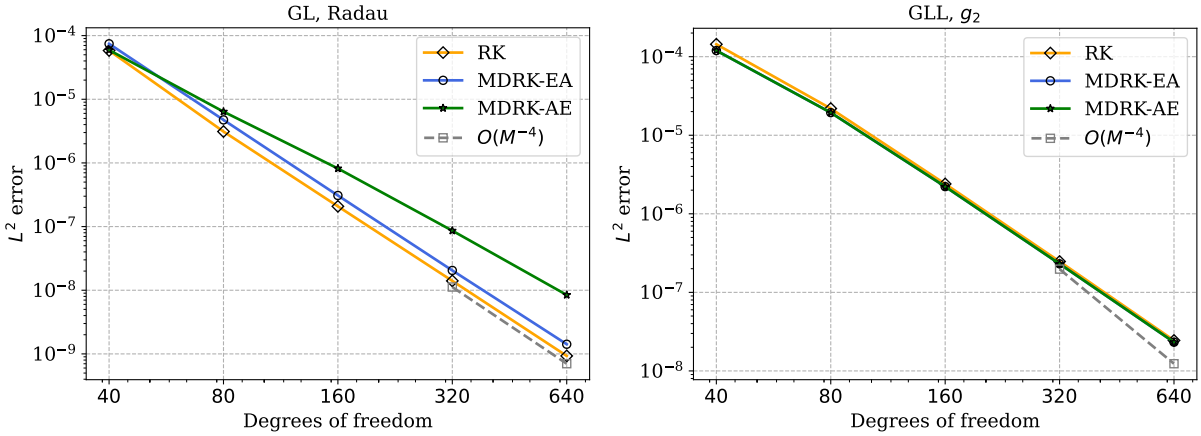


(a) (b) (c)

Figure 7.2. Error convergence for variable linear advection equation with $a(x) = x^2$; (a) **AE** scheme, (b) **EA** scheme, (c) **AE** versus **EA**

7.5.1.3. Burgers' equations

The one dimensional Burger's equation is a conservation law of the form $u_t + f(u)_x = 0$ with the quadratic flux $f(u) = u^2/2$. For the smooth initial condition $u(x,0) = 0.2\sin(x)$ with periodic boundary condition in the domain $[0, 2\pi]$, we compute the numerical solution at time $t = 2$ when the solution is still smooth. Figure 7.3a compares the error norms for the **AE** and **EA** methods for the Rusanov numerical flux, and using GL solution points, Radau correction and D2 dissipation. The convergence rate of **AE** is less than optimal and close to $O(h^{3+1/2})$. In Figure 7.3b, we see that no scheme shows optimal convergence rates when g_2 correction + GLL points is used. The comparison between D1, D2 dissipation is made in Figure 7.4 and their performances are found to be similar.



(a) (b)

Figure 7.3. Comparing **AE** and **EA** schemes using D2 dissipation for 1-D Burgers' equation at $t = 2$. (a) GL points with Radau correction, (b) GLL points with g_2 correction.

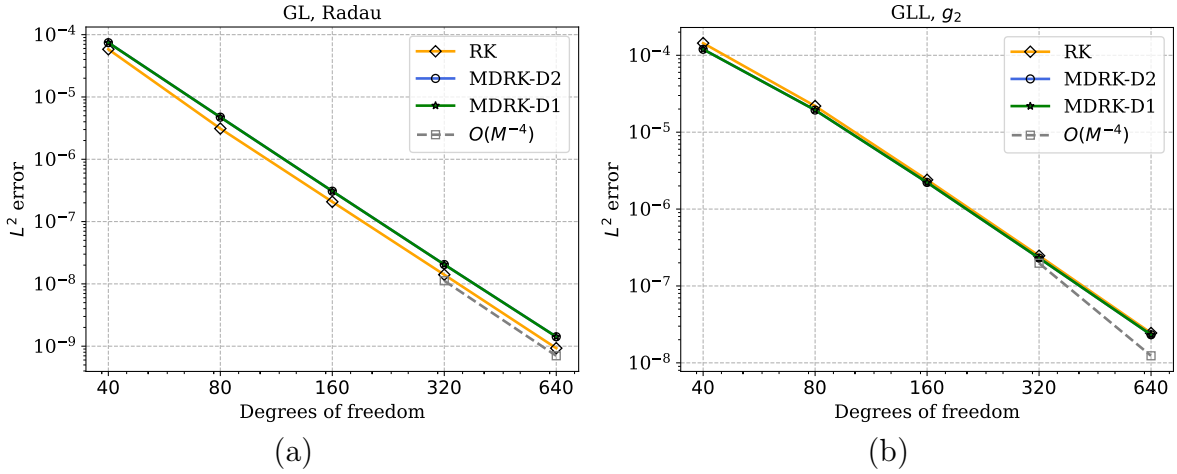


Figure 7.4. Comparing D1 and D2 dissipation for 1-D Burgers' equation at $t = 2$. (a) GL points with Radau correction, (b) GLL points with g_2 correction

7.5.2. 1-D Euler equations

We now consider the 1-D Euler's equations (4.16) and compute the time step size with (4.18) with CFL=0.107 (Table 7.1). The coefficient $C_{\text{CFL}} = 0.98$ is used in tests, unless specified otherwise.

7.5.2.1. Blast wave

This test is as described in Section 4.8.5. As in the case of the LWFR scheme, the numerical solutions give negative pressure if the positivity correction is not applied. With a grid of 400 cells using polynomial degree $N = 3$, we run the simulation till the time $t = 0.038$ where a high density peak profile is produced. As tested in Section 4.8.5, we compare first order (FO) and MUSCL-Hancock (MH) blending schemes, and TVB limiter with parameter $M = 300$ [137] (TVB-300). We compare the performance of limiters in Figure (7.5) where the approximated density and pressure profiles are compared with a reference solution computed using a very fine mesh. Looking at the peak amplitude and contact discontinuity, it is clear that MUSCL-Hancock blending scheme gives the best resolution, especially when compared with the TVB limiter.

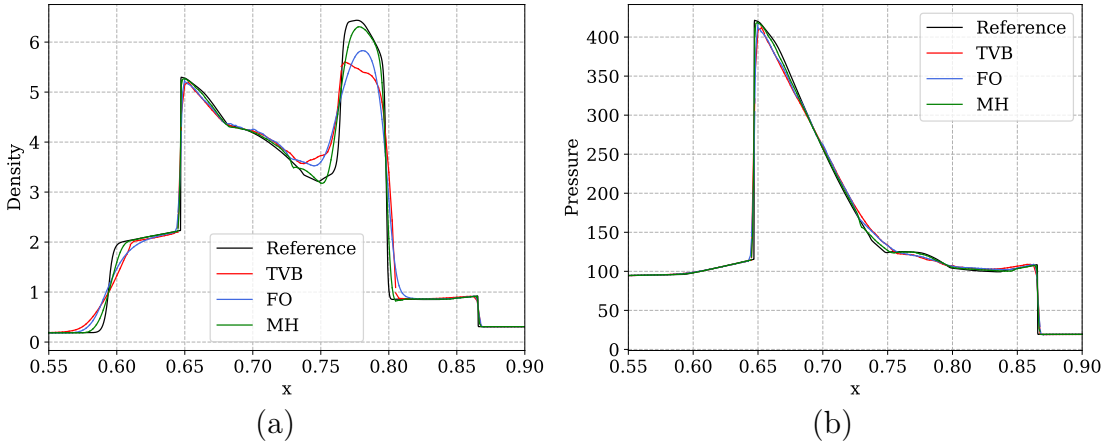


Figure 7.5. Blast wave problem using first order (FO) and MUSCL-Hancock blending schemes, and TVB limited scheme (TVB-300) with parameter $M = 300$. (a) Density, (b) Pressure profiles are shown at $t = 0.038$ on a mesh of 400 cells.

7.5.2.2. Titarev Toro

This is an extension of the Shu-Osher (Section 4.8.4) problem given by Titarev and Toro [176] and the initial data comprises of a severely oscillatory wave interacting with a shock

$$(\rho, v, p) = \begin{cases} (1.515695, 0.523346, 1.805), & -5 \leq x \leq -4.5 \\ (1 + 0.1\sin(20\,\pi\,x), 0, 1), & -4.5 < x \leq 5 \end{cases}$$

The physical domain is $[-5, 5]$ and transmissive boundary condition is used at both ends. This problem tests the ability of a high-order numerical scheme to capture the extremely high frequency waves. The smooth density profile passes through the shock and appears on the other side, and its accurate computation is challenging due to numerical dissipation. Due to presence of both spurious oscillations and smooth extrema, this becomes a good test for testing robustness and accuracy of limiters. We discretize the spatial domain with 800 cells using polynomial degree $N = 3$ and compare blending schemes. The density profile at $t = 5$ is shown in Figure 7.6. As expected, the MUSCL-Hancock (MH) blending scheme is superior to the First Order (FO) blending scheme and has nearly resolved the smooth extrema.

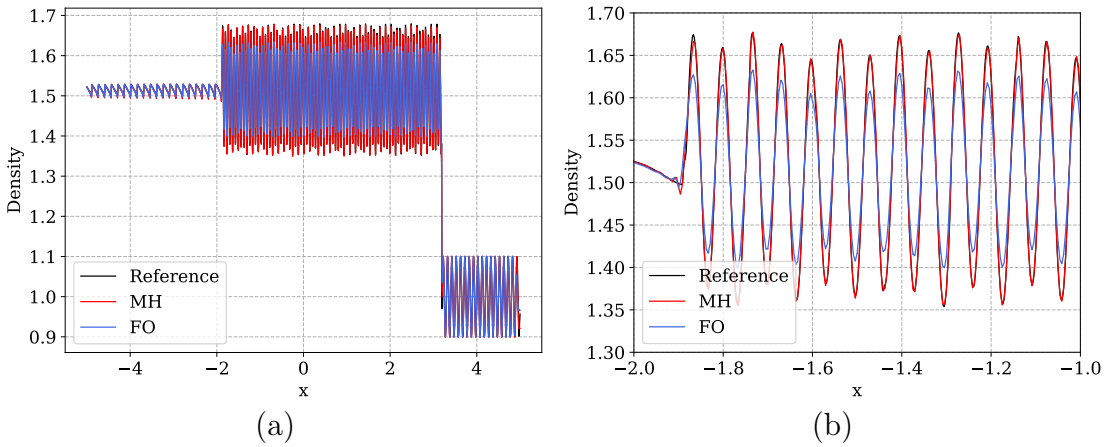


Figure 7.6. Titarev-Toro problem, comparing First Order (FO) and MUSCL-Hancock (MH) blending (a) Complete plot, (b) Profile zoomed near smooth extrema on a mesh of 800 cells.

7.5.2.3. Large density ratio Riemann problem

The second example is the large density ratio problem with a very strong rarefaction wave [174]. The initial condition is given by

$$(\rho, v, p) = \begin{cases} (1000, 0, 1000), & x < 0.3 \\ (1, 0, 1), & 0.3 < x \end{cases}$$

The computational domain is $[0, 1]$ and transmissive boundary condition is used at both ends. The density and pressure profile on a mesh of 500 elements at $t = 0.15$ is shown in Figure 7.7. The MH blending scheme is giving better accuracy even in this tough problem.

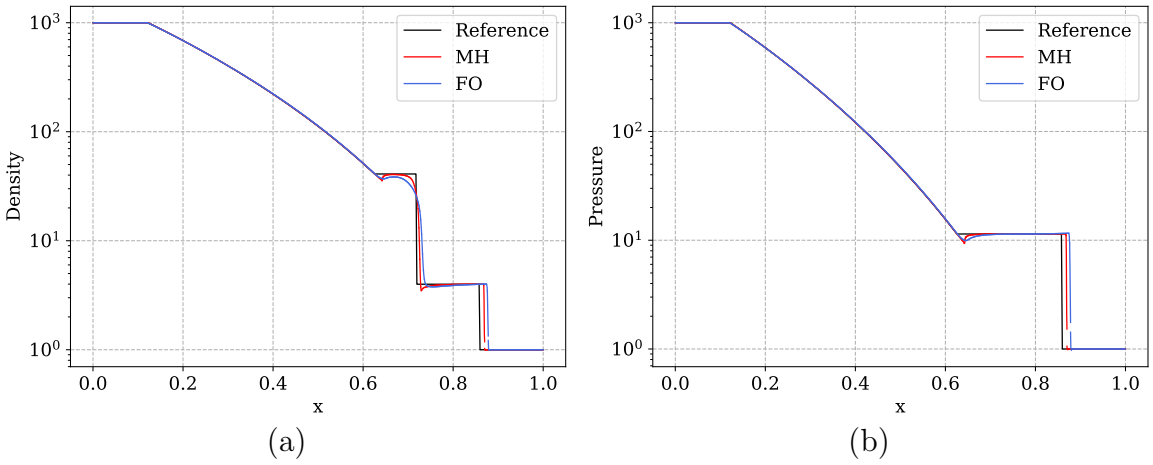


Figure 7.7. High density problem at $t = 0.15$ on a mesh of 500 elements (a) Density plot, (b) Pressure plot

7.5.2.4. Sedov's blast

This test case is as described in Section 5.7.1.3. Nonphysical solutions are obtained if the proposed admissibility preservation corrections are not applied. The density and pressure profiles at $t = 0.001$ are obtained using blending schemes are shown in Figure 7.8. In Chapter 5, the TVB limiter was not used in this test as the proof of admissibility preservation depended on the blending scheme. Here, by using the generalized admissibility preserving scheme of Chapter 6, [11], we are able to use the TVB limiter. However, as expected, the TVB limiter is less accurate and unable to control the oscillations.

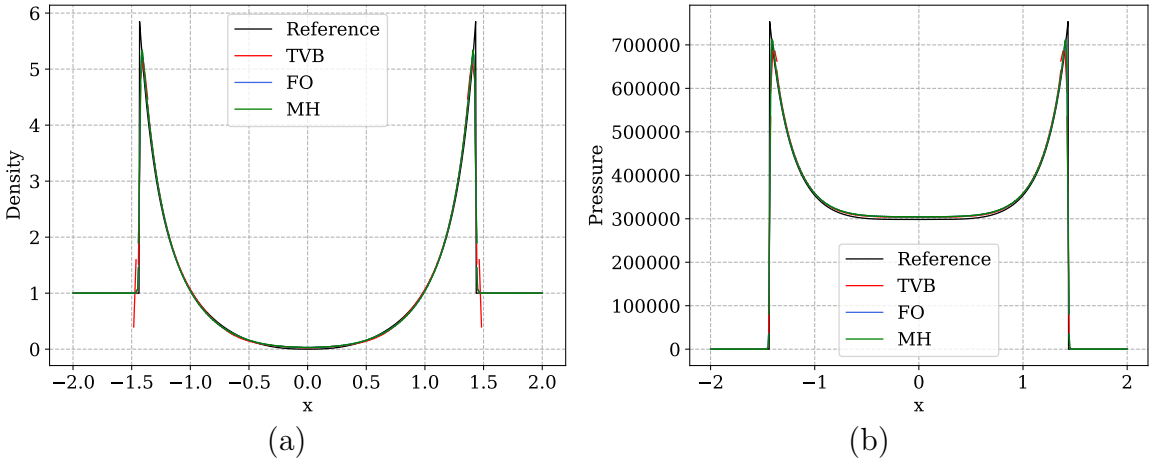


Figure 7.8. Sedov's blast wave problem, numerical solution using first order (FO) and MUSCL-Hancock blending schemes, and TVD (a) Density, (b) Pressure profiles are shown at $t = 0.001$ on a mesh of 201 cells.

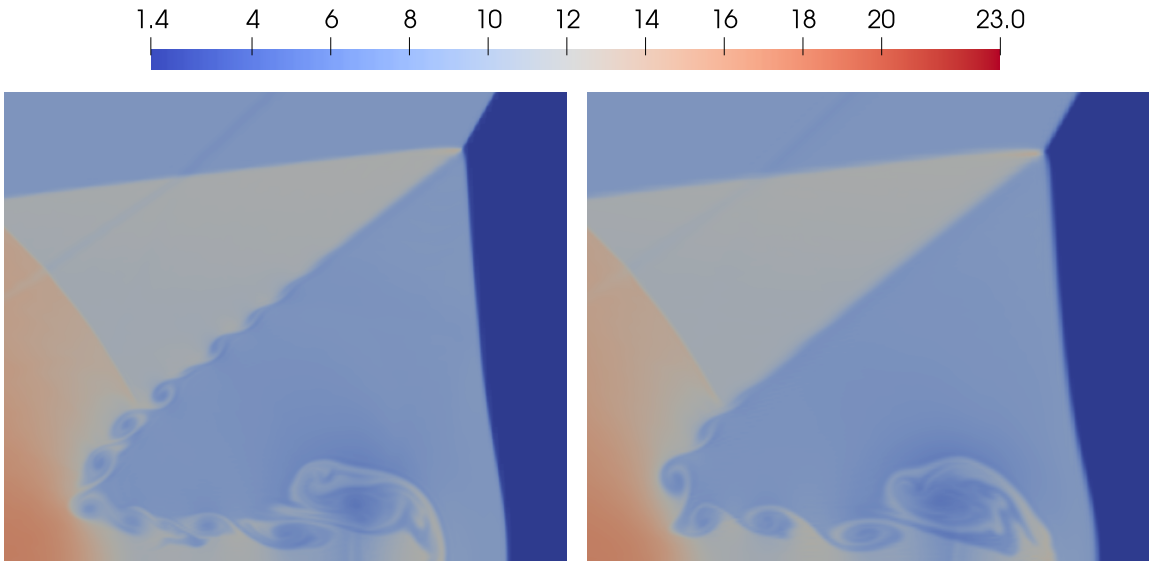
7.5.3. 2-D Euler's equations

We consider the two-dimensional Euler equations of gas dynamics given by (2.13). The time step size is computed as in (4.30) with CFL=0.107 (Table 7.1) and $C_{\text{CFL}} = 0.98$ unless otherwise specified.

As in Chapter 5, for verification of some of our numerical results and to demonstrate the accuracy gain observed in [19] of using MUSCL-Hancock reconstruction using Gauss-Legendre points, we will compare our results with the first order blending scheme using Gauss-Legendre-Lobatto (GLL) points of [90] available in `Trixi.jl` [141]. The accuracy benefit is expected since GL points and quadrature are more accurate than GLL points, and MUSCL-Hancock is also more accurate than first order finite volume method.

7.5.3.1. Double Mach reflection

The description and significance of this test have been given in Section 4.11.2. The simulation is run on a mesh of 600×150 elements using degree $N = 3$ polynomials up to time $t = 0.2$. In Figure 7.9, we compare the results of `Trixi.jl` with the MUSCL-Hancock blended scheme zoomed near the primary triple point. As expected, the small scale structures are captured better by the MUSCL-Hancock blended scheme.



(a) MDRK (b) `Trixi.jl`

Figure 7.9. Double Mach reflection problem, density plot of numerical solution at $t = 0.2$ on a 600×150 mesh zoomed near the primary triple point.

7.5.3.2. Rotational low density problem

These problems are taken from [132] where the solution consists of hurricane-like flow evolution and has one-point vacuum in the center with rotational velocity field. The initial condition is given by

$$(\rho, u, v, p) = (\rho_0, v_0 \sin\theta, -v_0 \cos\theta, A\,\rho_0^{\gamma})$$

where $\theta = \arctan(y/x)$, $A = 25$ is the initial entropy, $\rho = 1$ is the initial density, gas constant $\gamma = 2$. The initial velocity distribution has a nontrivial transversal component, which makes the flow rotational. The solutions are classified [204] into three types

according to the initial Mach number $M_0 = |v_0| / c_0$, where $c_0 = p'(\rho_0) = A\,\gamma\,\rho_0^{\gamma-1}$ is the sound speed.

1. **Critical rotation with $M_0 = \sqrt{2}$.** This test has an exact solution with explicit formula. The solution consists of two parts: a far field solution and a near-field solution. The former far field solution is defined for $r \geq 2\,t\,\sqrt{p'(\rho_0)}$, $r = \sqrt{x^2+y^2}$,

$$\begin{cases} U(x,y,t) &= \frac{1}{r}\left(2\,t\,p_0'\cos\theta + \sqrt{2\,p_0'}\,\sqrt{r^2 - 2\,t^2\,p_0'}\sin\theta\right) \\ V(x,y,t) &= \frac{1}{r}\left(2\,t\,p_0'\sin\theta - \sqrt{2\,p_0'}\,\sqrt{r^2 - 2\,t^2\,p_0'}\cos\theta\right) \\ \rho(x,y,t) &= \rho_0 \\ p(x,y,t) &= A\,\rho_0^{\gamma} \end{cases} \tag{7.25}$$

   and the near-field solution is defined for $r < 2\,t\,\sqrt{p'(\rho_0)}$

$$U(x,y,t) = \frac{x+y}{2\,t}, \quad V(x,y,t) = \frac{-x+y}{2\,t}, \quad \rho(x,y,t) = \frac{r^2}{8\,A\,t^2}$$

   The curl of the velocity in the near-field is

$$\mathrm{curl}(U,V) = V_x - U_y = -\frac{1}{2\,t} \neq 0$$

   and the solution has one-point vacuum at the origin $r = 0$. This is typical hurricane-like solution that is singular, particularly near the origin $r = 0$. There are two issues here challenging the numerical schemes: one is the presence of the vacuum state which examines whether a high order scheme can keep the positivity preserving property; the other is the rotational velocity field for testing whether a numerical scheme can preserve the symmetry. In this regime, we take $v_0 = 10$ on the computational domain $[-1,1]^2$ with $\Delta x = \Delta y = 1/100$. The boundary condition is given by the far field solution in (7.25).

2. **High-speed rotation with $M_0 > \sqrt{2}$.** For this case, $v_0 = 12.5$, so that the density goes faster to the vacuum and the fluid rotates severely. The physical domain is $[-2,2]^2$ and the grid spacing is $\Delta x = \Delta y = 1/100$. Outflow boundary conditions are given on the boundaries.Because of the higher rotation speed, this case is tougher than the first one, and can be used to validate the robustness of the higher-order scheme.

3. **Low-speed rotation with $M_0 < \sqrt{2}$.** In this test case, we take $v_0 = 7.5$ making it a rotation with lower speed than the previous tests. The outflow boundary conditions are given as in the previous tests. The simulation is performed in the domain $[-1,1]^2$ till $t = 0.045$.

The density profile for the flow with critical speed are shown in Figure 7.10 including a comparison with exact solution at a line cut of $y = 0$ in Figure 7.10b, showing near overlap. In Figure 7.11a, we show the line cut of density profile at $y = 0$ for the

three rotation speeds. In Figure 7.11b, we show streamlines for high rotational speed, showing symmetry.

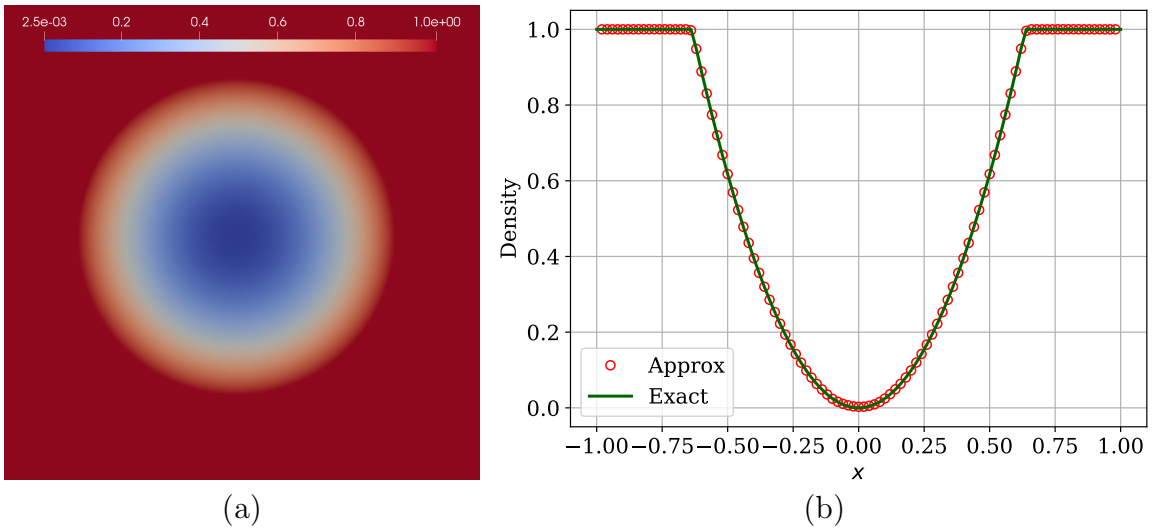


Figure 7.10. Density profile of rotational low density problem at critical speed (a) Pseudocolor plot (b) Line cut at $y = 0$ on a mesh with $\Delta x = \Delta y = 1/100$.

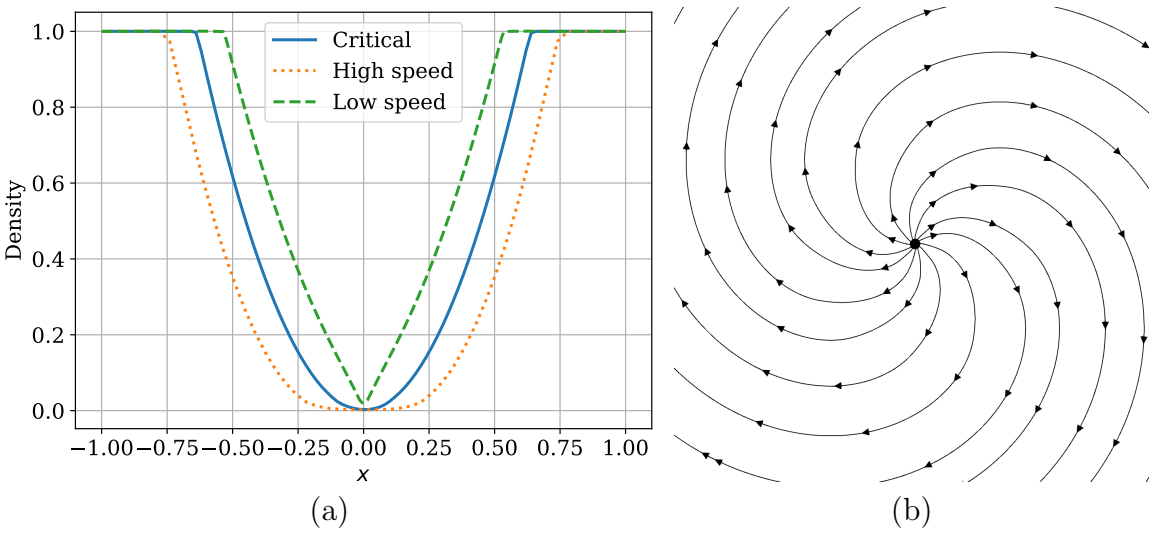
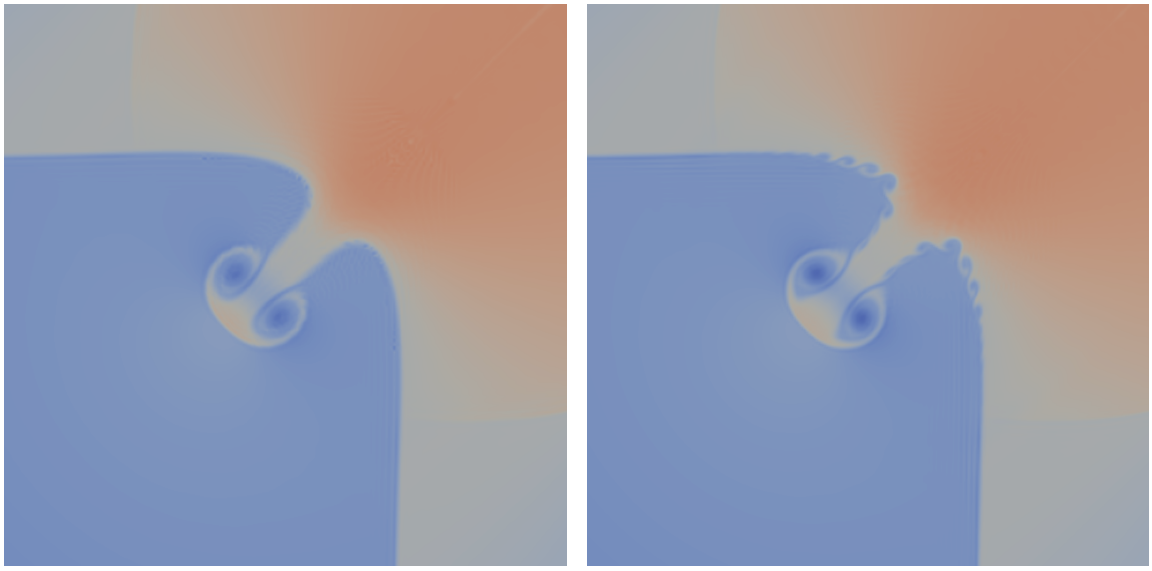


Figure 7.11. Rotational low density problem (a) Density profile line cut at $y = 0$ for different rotational speeds, (b) Stream lines for high rotational speed.

7.5.3.3. Two Dimensional Riemann problem

This test case is as described in Section 5.9.3. The simulations are performed with transmissive boundary conditions on an enlarged domain up to time $t = 0.25$. The density profiles obtained from the MUSCL-Hancock blending scheme and `Trixi.jl` are shown in Figure 7.12. We see that both schemes give similar resolution in most regions. As in LWFR scheme, the MUSCL-Hancock blending scheme gives better resolution of the small scale structures arising across the slip lines.



(a) `Trixi.jl` (b) MDRK

Figure 7.12. 2-D Riemann problem, density plots of numerical solution at $t = 0.25$ for degree $N = 3$ on a 256×256 mesh (a) `Trixi.jl`, (b) MDRK

7.5.3.4. Rayleigh-Taylor instability

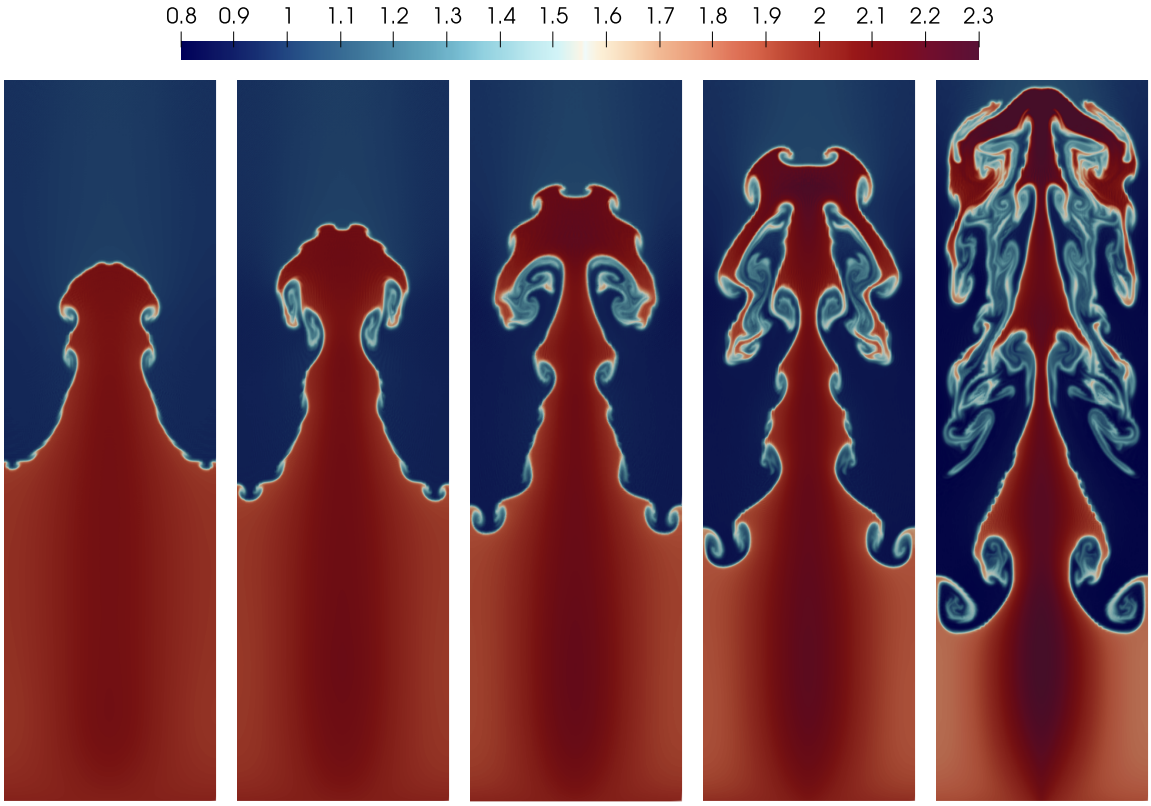
The last problem is the Rayleigh-Taylor instability to test the performance of higher-order scheme for the conservation laws with source terms, and the governing equations are written as

$$\frac{\partial}{\partial t}\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} \rho u \\ p + \rho u^2 \\ \rho u v \\ (E+p) u \end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix} \rho v \\ \rho u v \\ p + \rho v^2 \\ (E+p) v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \rho \\ \rho v \end{pmatrix}$$

The implementation of MDRK with source terms is based on [11] where an approximate Lax-Wendroff procedure is also applied to the source term. The following description is taken from [132]. The Rayleigh-Taylor instability happens on the interface between fluids with different densities when an acceleration is directed from the heavy fluid to the light one. The instability with fingering nature generates bubbles of light fluid rising into the ambient heavy fluid and spikes of heavy fluid falling into the light fluid. The initial condition of this problem [161] is given as follows

$$(\rho, u, v, p) = \begin{cases} (2, 0, -0.025\, a \cos(8\,\pi\, x), 2\, y + 1), & y \leq 0.5, \\ (1, 0, -0.025\, a \cos(8\,\pi\, x), y + 1.5), & y > 0.5 \end{cases}$$

where $a = \sqrt{\gamma p / \rho}$ is the sound speed and $\gamma = 5/3$. The computational domain is $[0, 0.25] \times [0, 1]$. The reflecting boundary conditions are imposed for the left and right boundaries. At the top boundary, the flow variables are set as $(\rho, u, v, p) = (1, 0, 0, 2.5)$. At the bottom boundary, they are $(\rho, u, v, p) = (2, 0, 0, 1)$. The uniform mesh with 64×256 elements is used in the simulation. The density distributions at $t = 1.5$, $1.75, 2, 2.25, 2.5$ are presented in Figure 7.13. It is a test to check the suitability of higher-order schemes for the capturing of interface instabilities.



(a) $t = 1.5$ (b) $t = 1.75$ (c) $t = 2$ (d) $t = 2.25$ (e) $t = 2.5$

Figure 7.13. Rayleigh-Taylor instability on a 64×256 mesh

7.6. Summary and conclusions

This chapter introduces fourth order multiderivative Runge-Kutta (MDRK) scheme of [119] in the conservative, quadrature free Flux Reconstruction framework to solve hyperbolic conservation laws. The idea is to cast each MDRK stage as an evolution involving a time average flux which is approximated by the Jacobian free Approximate Lax-Wendroff procedure. The numerical flux is carefully computed with accuracy and stability in mind. In particular, the D2 dissipation and **EA** flux of Chapter 4 have been introduced which enhance stable CFL numbers and accuracy for nonlinear problems respectively. The stable CFL numbers are computed using Fourier stability analysis for two commonly used correction functions g_{Radau} and g_2, showing the improved CFL numbers. Convergence analysis for non-linear problems was performed which revealed that optimal convergence rates were only shown when using the **EA** flux. The shock capturing blending scheme of Chapter 5 has also been introduced for the MDRK scheme applied at each stage. The scheme is provably admissibility preserving and good at capturing small scale structures. The claims are validated by numerical experiments for compressible Euler's equations with the modern test suite [132] of high order methods.

Chapter 8
Curvilinear grids

8.1. Introduction

This chapter extends the LWFR scheme to curvilinear grids that are body-fitted to handle curved geometries. These geometries occur in practical problems, especially in CFD. Adaptive mesh refinement is also developed for LWFR using the Mortar Element Method [106] in order to have efficient resolution of localized flows. The extension is conservative, free stream and admissibility preserving. In the previous chapters, the time step size was computed based on a wave speed estimate by using optimal CFL numbers obtained from a Fourier stability analysis. The Fourier stability analysis is based on uniform Cartesian grids and does not apply to the curvilinear case. Thus, usage of a wave speed based formula for computing time step sizes for curvilinear grids requires fine tuning of the CFL number for each problem and geometry. In order to minimize fine tuning, we propose an error based time step computation method for the LWFR method. Numerical results for compressible Euler's equations are used to validate LWFR on adaptively refined, curvilinear grids with error based time stepping. The performance improvement of error based time stepping over CFL based time stepping is also shown.

The chapter is organized as follows. In Section 8.2, we review notations and the transformation of conservation laws from curved elements to a reference cube following [105, 103]. In Section 8.3, the LWFR scheme of Chapter 4 is extended to curvilinear grids. In Section 8.3.1, we review FR on curvilinear grids and use it to construct LWFR on curvilinear grids in Section 8.3.2. Section 8.3.4 shows that the free stream preservation condition of LWFR is the standard metric identity of [105]. In Section 8.4, the admissibility preserving subcell limiter for LWFR Chapter 5 is extended to curvilinear grids. In Section 8.5, the Mortar Element Method for treatment of non-conformal interfaces in AMR of [106] is extended to LWFR. In Section 8.6, error-based time stepping methods are discussed; Section 8.6.1 reviews error-based time stepping methods for Runge-Kutta and Section 8.6.2 introduces an embedded error-based time stepping method for LWFR. In Section 8.7, numerical results are shown to demonstrate the scheme's capability of handling adaptively refined curved grids and benefits of error-based time stepping. Section 8.8 gives a summary and draws conclusions from the work.

8.2. Conservation laws and curvilinear grids

The developments in this work are applicable to a wide class of hyperbolic conservation laws but the numerical experiments are performed on 2-D compressible Euler's equations (2.13). For the sake of simplicity and generality, we subsequently explain the development of the algorithms for a general hyperbolic conservation law written as

$$\boldsymbol{u}_t + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{0} \tag{8.1}$$

where $\boldsymbol{u} \in \mathbb{R}^p$ is the vector of conserved quantities, $\boldsymbol{f}(\boldsymbol{u}) = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_d) \in \mathbb{R}^{p \times d}$ is the corresponding physical flux, $\boldsymbol{x}$ is in domain $\Omega \subset \mathbb{R}^d$ and

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{f} = \sum_{i=1}^{d} \partial_{x_i} \boldsymbol{f}_i \tag{8.2}$$

Let us partition Ω into M non-overlapping quadrilateral/hexahedral elements Ω_e such that

$$\Omega = \bigcup_{e=1}^{M} \Omega_e \tag{8.3}$$

The elements Ω_e are allowed to have curved boundaries in order to match curved boundaries of the problem domain Ω. In this chapter, we take the reference element to be $\Omega_o = [-1, 1]^d$ in contrast to the previous chapters where it was $[0, 1]^d$. This choice is made for compatibility with `Trixi.jl` [141]. To construct the numerical approximation, we map each element Ω_e to a reference element $\Omega_o = [-1, 1]^d$ by a bijective map $\Theta_e \colon \Omega_o \to \Omega_e$

$$\boldsymbol{x} = \Theta_e(\boldsymbol{\xi})$$

where $\boldsymbol{\xi} = (\xi^i)_{i=1}^d$ are the coordinates in the reference element, and the subscript e will usually be suppressed. We will denote a d-dimensional multi-index as $\boldsymbol{p} = (p_i)_{i=1}^d$. In this work, the reference map is defined using tensor product Lagrange interpolation of degree $N \geq 1$,

$$\Theta(\boldsymbol{\xi}) = \sum_{\boldsymbol{p} \in \mathbb{N}_N^d} \hat{\boldsymbol{x}}_{\boldsymbol{p}} \, \ell_{\boldsymbol{p}}(\boldsymbol{\xi}) \tag{8.4}$$

where

$$\mathbb{N}_N^d = \{\boldsymbol{p} = (p_1, \ldots, p_d) \colon p_i \in \{0, 1, \ldots, N\}, 1 \leq i \leq d\} \tag{8.5}$$

and $\{\ell_{\boldsymbol{p}}\}_{\boldsymbol{p} \in \mathbb{N}_N^d}$ is the degree N Lagrange polynomial corresponding to the Gauss-Legendre-Lobatto (GLL) points $\{\boldsymbol{\xi}_{\boldsymbol{p}}\}_{\boldsymbol{p} \in \mathbb{N}_N^d}$ so that $\Theta(\boldsymbol{\xi}_{\boldsymbol{p}}) = \hat{\boldsymbol{x}}_{\boldsymbol{p}}$ for all $\boldsymbol{p} \in \mathbb{N}_N^d$. Thus, the points $\{\boldsymbol{\xi}_{\boldsymbol{p}}\}_{\boldsymbol{p} \in \mathbb{N}_N^d}$ are where the reference map will be specified and they will also be taken to be the solution points of the Flux Reconstruction scheme throughout this chapter. The functions $\{\ell_{\boldsymbol{p}}\}_{\boldsymbol{p} \in \mathbb{N}_N^d}$ can be written as a tensor product of the 1-D Lagrange polynomials $\{\ell_{p_i}\}_{p_i=0}^N$ of degree N corresponding to the GLL points $\{\xi_{p_i}\}_{p_i=0}^N$

$$\ell_{\boldsymbol{p}}(\boldsymbol{\xi}) = \prod_{i=1}^{d} \ell_{p_i}(\xi^i), \qquad \ell_{p_i}(\xi^i) = \prod_{k=0, k \neq i}^{N} \frac{\xi^i - \xi_{p_k}}{\xi_{p_i} - \xi_{p_k}} \tag{8.6}$$

The numerical approximation of the conservation law will be developed by first transforming the PDE in terms of the coordinates of the reference cell. To do this, we need to introduce covariant and contravariant basis vectors with respect to the reference coordinates.

Definition 8.1. (Covariant basis) *The coordinate basis vectors* $\{\boldsymbol{a}_i\}_{i=1}^d$ *are defined so that* $\boldsymbol{a}_i, \boldsymbol{a}_j$ *are tangent to* $\{\xi^k = \text{const}\}$ *where* i, j, k *are cyclic. They are explicitly given as*

$$\boldsymbol{a}_i = (a_{i,1}, \ldots, a_{i,d}) = \frac{\partial \boldsymbol{x}}{\partial \xi^i}, \qquad 1 \le i \le d \tag{8.7}$$

Definition 8.2. (Contravariant basis) *The contravariant basis vectors* $\{\boldsymbol{a}^i\}_{i=1}^d$ *are the respective normal vectors to the coordinate planes* $\{\xi_i = \text{const}\}_{i=1}^3$. *They are explicitly given as*

$$\boldsymbol{a}^i = (a_1^i, \ldots, a_d^i) = \nabla_{\boldsymbol{x}} \xi^i, \qquad 1 \le i \le d \tag{8.8}$$

The covariant basis vectors $\boldsymbol{a}_i$ can be computed by differentiating the reference map $\Theta(\xi)$. The contravariant basis vectors can be computed using [105, 103]

$$J\boldsymbol{a}^i = \boldsymbol{a}_j \times \boldsymbol{a}_k \tag{8.9}$$

where (i, j, k) are cyclic, and J denotes the Jacobian of the transformation which also satisfies

$$J = \det\left[\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right] = \boldsymbol{a}_i \cdot (\boldsymbol{a}_j \times \boldsymbol{a}_k) \qquad (i, j, k) \text{ cyclic}$$

The divergence of a flux vector can be computed in reference coordinates using the contravariant basis vectors as [105, 103]

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{f} = \frac{1}{J} \sum_{i=1}^{d} \frac{\partial}{\partial \xi^i} (J\boldsymbol{a}^i \cdot \boldsymbol{f}) \tag{8.10}$$

Consequently, the gradient of a scalar function ϕ becomes

$$\nabla \phi = \frac{1}{J} \sum_{i=1}^{d} \frac{\partial}{\partial \xi^i} [(J\boldsymbol{a}^i)\, \phi] \tag{8.11}$$

Within each element Ω_e, performing change of variables with the reference map Θ_e (8.10), the transformed conservation law is given by

$$\tilde{\boldsymbol{u}}_t + \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}} = \boldsymbol{0} \tag{8.12}$$

where

$$\tilde{\boldsymbol{u}} = J\boldsymbol{u}, \qquad \tilde{\boldsymbol{f}}^i = J\boldsymbol{a}^i \cdot \boldsymbol{f} = \sum_{n=1}^{d} J a_n^i \boldsymbol{f}_n \tag{8.13}$$

The flux $\tilde{\boldsymbol{f}}$ is referred to as the contravariant flux.

The vectors $\{J\boldsymbol{a}^i\}_{i=1}^d$ are called the metric terms and the *metric identity* is given by

$$\sum_{i=1}^{d} \frac{\partial (J\boldsymbol{a}^i)}{\partial \xi^i} = \boldsymbol{0} \tag{8.14}$$

The metric identity can be obtained by reasoning that the gradient of a constant function is zero and using (8.11) or that a constant solution must remain constant in (8.12). The metric identity is crucial for studying free stream stream preservation of a numerical scheme.

Remark 8.3. The equations for two dimensional case can be obtained by setting $(\Theta(\boldsymbol{\xi}))_3 = x_3(\boldsymbol{\xi}) = \xi^3$ so that $\boldsymbol{a}_3 = (0, 0, 1)$.

8.3. LWFR ON CURVILINEAR GRIDS

The solution of the conservation law will be approximated by piecewise polynomial functions which are allowed to be discontinuous across the elements. In each element Ω_e, the solution is approximated by

$$\hat{\boldsymbol{u}}_e^{\delta}(\boldsymbol{\xi}) = \sum_{\boldsymbol{p}} \boldsymbol{u}_{e,\boldsymbol{p}}\, \ell_{\boldsymbol{p}}(\boldsymbol{\xi}) \tag{8.15}$$

where the $\ell_{\boldsymbol{p}}$ are tensor-product polynomials of degree N which have been already introduced before to define the map to the reference element. The hat will be used to denote functions written in terms of the reference coordinates and the delta denotes functions that are possibly discontinuous across the element boundaries. Note that the coefficients $\boldsymbol{u}_{e,\boldsymbol{p}}$ are the values of the function at the solution points which are GLL points.

8.3.1. Flux Reconstruction (FR)

Recall that we defined the multi-index $\boldsymbol{p} = (p_i)_{i=1}^d$ (8.5) where $p_i \in \{0, 1 \ldots, N\}$. Let $i \in \{1, \ldots, d\}$ denote a coordinate direction and $S \in \{L, R\}$ so that (S, i) corresponds to the face $\partial\Omega_{o,i}^s$ in direction i on side S which has the reference outward normal $\hat{\boldsymbol{n}}_{S,i}$, see Figure 8.1. Thus, $\partial\Omega_{o,i}^R$ denotes the face where reference outward normal is $\hat{\boldsymbol{n}}_{R,i} = \boldsymbol{e}_i$ and $\partial\Omega_{o,i}^L$ has outward unit normal $\hat{\boldsymbol{n}}_{L,i} = -\hat{\boldsymbol{n}}_{R,i}$.

The FR scheme is a collocation scheme at each of the solution points $\{\boldsymbol{\xi}_{\boldsymbol{p}} = (\xi_{p_i})_{i=1}^d$, $p_i = 0, \ldots, N\}$. We will thus explain the scheme for a fixed $\boldsymbol{\xi} = \boldsymbol{\xi}_{\boldsymbol{p}}$ and denote $\boldsymbol{\xi}_i^S$ as the projection of $\boldsymbol{\xi} = (\xi_j)_{j=1}^d$ to the face $S = L, R$ in the i^{th} direction (see Figure 8.1), i.e.,

$$(\boldsymbol{\xi}_i^S)_j = \begin{cases} \xi_j, & j \neq i \\ -1, & j = i, S = L \\ +1, & j = i, S = R \end{cases} \tag{8.16}$$

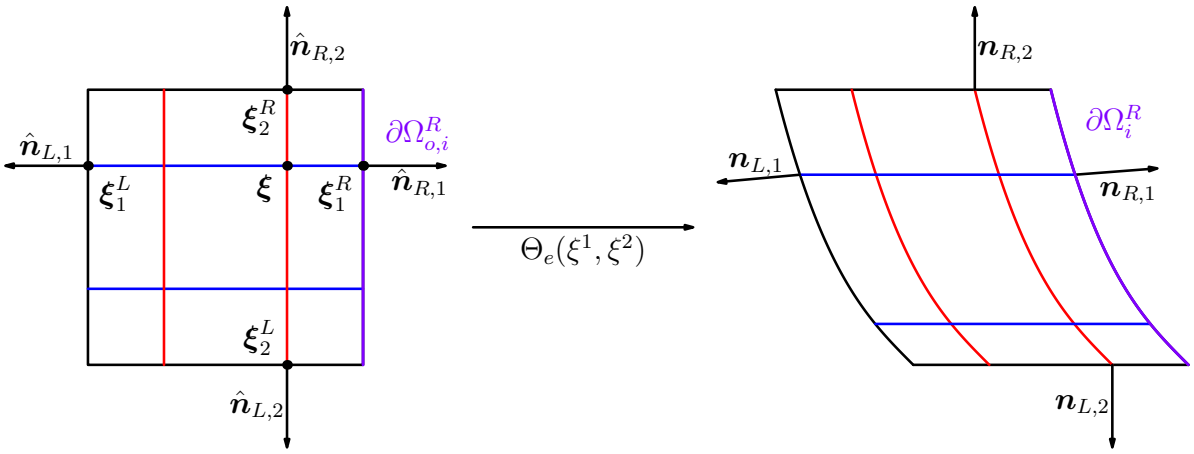


Figure 8.1. Illustration of reference map, solution point projections, reference and physical normals.

The first step is to construct an approximation to the flux by interpolating at the solution points

$$(\tilde{\boldsymbol{f}}_e^{\delta})_i(\boldsymbol{\xi}) = \sum_{\boldsymbol{q}} (J\boldsymbol{a}^i \cdot \boldsymbol{f})(\boldsymbol{\xi}_{\boldsymbol{q}})\, \ell_{\boldsymbol{q}}(\boldsymbol{\xi}) \tag{8.17}$$

which may be discontinuous across the element interfaces. In order to couple the neighbouring elements and ensure conservation property, continuity of the normal flux at the interfaces is enforced by constructing the *continuous flux approximation* using the FR correction functions g_L, g_R [94]. We construct this for the contravariant flux $\tilde{\boldsymbol{f}}^{\delta}$ (8.17) by performing correction along each direction i,

$$\begin{aligned}(\tilde{\boldsymbol{f}}_e(\boldsymbol{\xi}))^i \;=\; (\tilde{\boldsymbol{f}}_e^{\delta}(\boldsymbol{\xi}))^i \;&+\; ((\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{R,i})^* - \tilde{\boldsymbol{f}}_e^{\delta} \cdot \hat{\boldsymbol{n}}_{R,i})(\boldsymbol{\xi}_i^R)\, g_R(\xi_{p_i}) \\ &-\; ((\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{L,i})^* - \tilde{\boldsymbol{f}}_e^{\delta} \cdot \hat{\boldsymbol{n}}_{L,i})(\boldsymbol{\xi}_i^L)\, g_L(\xi_{p_i})\end{aligned} \tag{8.18}$$

where $\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{S,i}(\boldsymbol{\xi}_i^S)$ denotes the trace value of the normal flux in element Ω_e and $(\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_i)^*(\boldsymbol{\xi}_i^S)$ denotes the numerical flux. We will use Rusanov's numerical flux [152] which for the face (S,i) is given by

$$(\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{S,i})^* = \tilde{\boldsymbol{f}}^*(\boldsymbol{u}_{S,i}^-, \boldsymbol{u}_{S,i}^+, \hat{\boldsymbol{n}}_{S,i}) = \frac{1}{2}[(\tilde{\boldsymbol{f}}^{\delta} \cdot \hat{\boldsymbol{n}}_{S,i})^+ + (\tilde{\boldsymbol{f}}^{\delta} \cdot \hat{\boldsymbol{n}}_{S,i})^-] - \frac{\lambda_{S,i}}{2}(\boldsymbol{u}_{S,i}^+ - \boldsymbol{u}_{S,i}^-) \tag{8.19}$$

The $(\tilde{\boldsymbol{f}}^{\delta} \cdot \boldsymbol{n}_{S,i})^{\pm}$ and $\boldsymbol{u}_{S,i}^{\pm}$ denote the trace values of the normal flux and solution from outer, inner directions respectively; the inner direction corresponds to the element Ω_e while the outer direction corresponds to its neighbour across the interface (S,i). The $\lambda_{S,i}$ is a local wave speed estimate at the interface (S,i). For compressbile Euler's equations (2.13), the wave speed is estimated as [141]

$$\lambda = \max(|v^-|, |v^+|) + \max(|c^-|, |c^+|), \qquad v^{\pm} = \boldsymbol{v} \cdot \boldsymbol{n}^{\pm}, \qquad c^{\pm} = \sqrt{\gamma p^{\pm} / \rho^{\pm}} \tag{8.20}$$

where $\boldsymbol{n}$ is the physical unit normal at the interface. The FR correction functions g_L, g_R in the degree $N+1$ polynomial space $\mathbb{P}_{N+1}$ are a crucial ingredient of the FR scheme and have the property

$$g_L(-1) = g_R(1) = 1, \qquad g_L(1) = g_R(-1) = 0$$

Reference [94] gives a discussion on how the choice of correction functions leads to equivalence between FR and variants of DG scheme. In this work, the correction functions known as g_2 or g_{HU} from [94] are used since along with Gauss-Legendre-Lobatto (GLL) solution points, they lead to an FR scheme which is equivalent to a DG scheme using the same GLL solution and quadrature points (Appendix B). Once the continuous flux approximation is obtained, the FR scheme is given by

$$\frac{\mathrm{d}\boldsymbol{u}^{\delta}_{e,\boldsymbol{p}}}{\mathrm{d}t} + \frac{1}{J_{e,\boldsymbol{p}}} \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}_e(\boldsymbol{\xi}_{\boldsymbol{p}}) = \boldsymbol{0}, \qquad \forall \boldsymbol{p} \tag{8.21}$$

where $J_{e,\boldsymbol{p}}$ is the Jacobian of the transformation at solution points $\boldsymbol{x}_{e,\boldsymbol{p}}$. The FR scheme is explicitly written as

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{u}^{\delta}_{e,\boldsymbol{p}}}{\mathrm{d}t} \; & + \frac{1}{J_{e,\boldsymbol{p}}} \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}^{\delta}_e(\boldsymbol{\xi}) \\ & + \frac{1}{J_{e,\boldsymbol{p}}} \sum_{i=1}^{d} \left((\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{R,i})^* - \tilde{\boldsymbol{f}}^{\delta}_e \cdot \hat{\boldsymbol{n}}_{R,i} \right)(\boldsymbol{\xi}^R_i)\, g'_R(\xi_{p_i}) \\ & - \frac{1}{J_{e,\boldsymbol{p}}} \sum_{i=1}^{d} \left((\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{L,i})^* - \tilde{\boldsymbol{f}}^{\delta}_e \cdot \hat{\boldsymbol{n}}_{L,i} \right)(\boldsymbol{\xi}^L_i)\, g'_L(\xi_{p_i}) = \boldsymbol{0} \end{aligned} \tag{8.22}$$

8.3.2. Lax-Wendroff Flux Reconstruction (LWFR)

The LWFR scheme is obtained by following the Lax-Wendroff procedure for Cartesian domains [208, 18] (Chapter 4) on the transformed equation (8.12). With $\boldsymbol{u}^n$ denoting the solution at time level $t = t_n$, the solution at the next time level can be written using Taylor expansion in time as

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \sum_{k=1}^{N+1} \frac{\Delta t^k}{k!} \partial_t^{(k)} \boldsymbol{u}^n + O(\Delta t^{N+2})$$

where N is the solution polynomial degree. Then, use $\boldsymbol{u}_t = -\frac{1}{J} \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}$ (8.12) to swap a temporal derivative with a spatial derivative and retain terms up to $O(\Delta t^{N+1})$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \frac{1}{J} \sum_{k=1}^{N+1} \frac{\Delta t^k}{k!} \partial_t^{(k-1)} (\nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}})$$

Shifting indices and writing in a conservative form

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \frac{\Delta t}{J} \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{F}} \tag{8.23}$$

where $\tilde{\boldsymbol{F}}$ is a time averaged approximation of the contravaraint flux $\tilde{\boldsymbol{f}}$ given by

$$\tilde{\boldsymbol{F}} = \sum_{k=0}^{N} \frac{\Delta t^k}{(k+1)!} \partial_t^k \tilde{\boldsymbol{f}} \approx \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \tilde{\boldsymbol{f}} \, \mathrm{d}t \tag{8.24}$$

We first construct an element local order $N+1$ approximation $\tilde{\boldsymbol{F}}_e^\delta$ to $\tilde{\boldsymbol{F}}$ using the Approximate Lax-Wendroff procedure described in Section 8.3.3 to get

$$\tilde{\boldsymbol{F}}_e^\delta(\boldsymbol{\xi}) = \sum_{\boldsymbol{p}} \tilde{\boldsymbol{F}}_{e,\boldsymbol{p}}\, \ell_{\boldsymbol{p}}(\boldsymbol{\xi})$$

The $\tilde{\boldsymbol{F}}_e^\delta$ will be, in general, discontinuous across the element interfaces. Then, we construct the *continuous time averaged flux approximation* by performing a correction along each direction i, analogous to the case of FR (8.18), leading to

$$\begin{aligned}(\tilde{\boldsymbol{F}}_e(\boldsymbol{\xi}))^i \;=\; (\tilde{\boldsymbol{F}}_e^\delta(\boldsymbol{\xi}))^i \;&+\; ((\tilde{\boldsymbol{F}}_e\cdot\hat{\boldsymbol{n}}_{R,i})^* - \tilde{\boldsymbol{F}}_e^\delta\cdot\hat{\boldsymbol{n}}_{R,i})(\boldsymbol{\xi}_i^R)\, g_R(\xi_{p_i}) \\ &-\; ((\tilde{\boldsymbol{F}}_e\cdot\hat{\boldsymbol{n}}_{L,i})^* - \tilde{\boldsymbol{F}}_e^\delta\cdot\hat{\boldsymbol{n}}_{L,i})(\boldsymbol{\xi}_i^L)\, g_L(\xi_{p_i})\end{aligned} \tag{8.25}$$

where, as in (8.19), the numerical flux $(\tilde{\boldsymbol{F}}_e\cdot\hat{\boldsymbol{n}}_{S,i})^*$ is an approximation to the time average flux and is computed by a Rusanov-type approximation,

$$(\tilde{\boldsymbol{F}}_e\cdot\hat{\boldsymbol{n}}_{S,i})^* = \frac{1}{2}\left[(\tilde{\boldsymbol{F}}^\delta\cdot\hat{\boldsymbol{n}}_{S,i})^+ + (\tilde{\boldsymbol{F}}^\delta\cdot\hat{\boldsymbol{n}}_{S,i})^-\right] - \frac{\lambda_{S,i}}{2}\left(\boldsymbol{U}_{S,i}^+ - \boldsymbol{U}_{S,i}^-\right) \tag{8.26}$$

and $\boldsymbol{U}$ is the approximation of time average solution given by

$$\boldsymbol{U} = \sum_{k=0}^{N} \frac{\Delta t^k}{(k+1)!}\,\partial_t^k \boldsymbol{u} \approx \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \boldsymbol{u}\,\mathrm{d}t \tag{8.27}$$

The computation of dissipative part of (8.26) using the time averaged solution instead of the solution at time t_n was introduced in Section 4.3 and was termed D2 dissipation. It is a natural choice in approximating the time averaged numerical flux and does not add any significant computational cost because the temporal derivatives of $\boldsymbol{u}$ are already available when computing the local approximation $\tilde{\boldsymbol{F}}^\delta$. The choice of D2 dissipation reduces to an upwind scheme in case of constant advection equation and leads to enhanced Fourier CFL stability limit (Section 4.4).

The Lax-Wendroff update is performed following (8.21) for (8.23)

$$\boldsymbol{u}_{e,\boldsymbol{p}}^{n+1} = \boldsymbol{u}_{e,\boldsymbol{p}}^n - \frac{\Delta t}{J_{e,\boldsymbol{p}}}\nabla_{\boldsymbol{\xi}}\cdot\tilde{\boldsymbol{F}}_e(\boldsymbol{\xi}_{\boldsymbol{p}})$$

which can be explicitly written as

$$\begin{aligned}\boldsymbol{u}_{e,\boldsymbol{p}}^{n+1} = \boldsymbol{u}_{e,\boldsymbol{p}}^n \;&-\frac{\Delta t}{J_{e,\boldsymbol{p}}}\nabla_{\boldsymbol{\xi}}\cdot\tilde{\boldsymbol{F}}_e^\delta(\boldsymbol{\xi}_{\boldsymbol{p}}) \\ &-\frac{\Delta t}{J_{e,\boldsymbol{p}}}\sum_{i=1}^{d}\left((\tilde{\boldsymbol{F}}_e\cdot\hat{\boldsymbol{n}}_{R,i})^* - \tilde{\boldsymbol{F}}_e^\delta\cdot\hat{\boldsymbol{n}}_{R,i}\right)(\boldsymbol{\xi}_i^R)\, g_R'(\xi_{p_i}) \\ &+\frac{\Delta t}{J_{e,\boldsymbol{p}}}\sum_{i=1}^{d}\left((\tilde{\boldsymbol{F}}_e\cdot\hat{\boldsymbol{n}}_{L,i})^* - \tilde{\boldsymbol{F}}_e^\delta\cdot\hat{\boldsymbol{n}}_{L,i}\right)(\boldsymbol{\xi}_i^L)\, g_L'(\xi_{p_i})\end{aligned} \tag{8.28}$$

By multiplying (8.28) by quadrature weights $J_{e,\boldsymbol{p}}\,w_{\boldsymbol{p}}$ and summing over $\boldsymbol{p}$, it is easy to see that the scheme is conservative in the sense that

$$\bar{\boldsymbol{u}}_e^{n+1} = \bar{\boldsymbol{u}}_e^n - \frac{\Delta t}{|\Omega_e|}\left(\sum_{i=1}^{d}\int_{\partial\Omega_{o,i}^R}(\tilde{\boldsymbol{F}}_e\cdot\hat{\boldsymbol{n}}_{R,i})^*\mathrm{d}S_{\boldsymbol{\xi}} + \int_{\partial\Omega_{o,i}^L}(\tilde{\boldsymbol{F}}_e\cdot\hat{\boldsymbol{n}}_{L,i})^*\mathrm{d}S_{\boldsymbol{\xi}}\right) \tag{8.29}$$

where the element mean value $\bar{\boldsymbol{u}}_e$ is defined to be

$$\bar{\boldsymbol{u}}_e = \frac{1}{|\Omega_e|} \sum_{\boldsymbol{p}} \boldsymbol{u}_{e,\boldsymbol{p}} J_{e,\boldsymbol{p}} w_{\boldsymbol{p}} \tag{8.30}$$

We provide the proof of (8.29) for completeness. Multiply (8.28) with $J_{e,\boldsymbol{p}} \boldsymbol{w}_{\boldsymbol{p}}$ and sum over $\boldsymbol{p} \in \mathbb{N}_N^d$ to get, using the exactness of Gauss-Legendre-Lobatto (GLL) quadrature

$$\begin{aligned}
\bar{\boldsymbol{u}}_e^{n+1} = \bar{\boldsymbol{u}}_e^n &- \frac{\Delta t}{|\Omega_e|} \int_{\Omega_o} \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{F}}_e^\delta(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi} \\
&- \frac{\Delta t}{|\Omega_e|} \int_{\Omega_o} \sum_{i=\overline{1}}^{d} ((\tilde{\boldsymbol{F}}_e \cdot \hat{\boldsymbol{n}}_{R,i})^* - \tilde{\boldsymbol{F}}_e^\delta \cdot \hat{\boldsymbol{n}}_{R,i})(\boldsymbol{\xi}_i^R) \, g_R'(\xi_{p_i}) \, \mathrm{d}\boldsymbol{\xi} \\
&+ \frac{\Delta t}{|\Omega_e|} \int_{\Omega_o} \sum_{i=1}^{d} ((\tilde{\boldsymbol{F}}_e \cdot \hat{\boldsymbol{n}}_{L,i})^* - \tilde{\boldsymbol{F}}_e^\delta \cdot \hat{\boldsymbol{n}}_{L,i})(\boldsymbol{\xi}_i^L) \, g_L'(\xi_{p_i}) \, \mathrm{d}\boldsymbol{\xi}
\end{aligned} \tag{8.31}$$

where $\boldsymbol{\xi}_i^S$ are as defined in (8.16). Then, note the following integral identities that are an application of Fubini's theorem followed by fundamental theorem of Calculus

$$\begin{aligned}
\int_{\Omega_o} \partial_{\xi^i} \tilde{\boldsymbol{F}}_e^\delta(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi} &= \int_{\partial\Omega_{o,i}^L} [\tilde{\boldsymbol{F}}_e^\delta \cdot \hat{\boldsymbol{n}}_{L,i}] \, \mathrm{d}S_{\boldsymbol{\xi}} + \int_{\partial\Omega_{o,i}^R} [\tilde{\boldsymbol{F}}_e^\delta \cdot \hat{\boldsymbol{n}}_{R,i}] \, \mathrm{d}S_{\boldsymbol{\xi}} \\
\int_{\Omega_o} ((\tilde{\boldsymbol{F}}_e \cdot \hat{\boldsymbol{n}}_{R,i})^* - \tilde{\boldsymbol{F}}_e^\delta \cdot \hat{\boldsymbol{n}}_{R,i})(\boldsymbol{\xi}_i^R) \, g_R'(\xi_{p_i}) \, \mathrm{d}\boldsymbol{\xi} &= \int_{\partial\Omega_{o,i}^R} [(\tilde{\boldsymbol{F}}_e \cdot \hat{\boldsymbol{n}}_{R,i})^* - \tilde{\boldsymbol{F}}_e^\delta \cdot \hat{\boldsymbol{n}}_{R,i}] \, \mathrm{d}S_{\boldsymbol{\xi}} \\
\int_{\Omega_o} ((\tilde{\boldsymbol{F}}_e \cdot \hat{\boldsymbol{n}}_{L,i})^* - \tilde{\boldsymbol{F}}_e^\delta \cdot \hat{\boldsymbol{n}}_{L,i})(\boldsymbol{\xi}_i^L) \, g_L'(\xi_{p_i}) \, \mathrm{d}\boldsymbol{\xi} &= -\int_{\partial\Omega_{o,i}^L} [(\tilde{\boldsymbol{F}}_e \cdot \hat{\boldsymbol{n}}_{L,i})^* - \tilde{\boldsymbol{F}}_e^\delta \cdot \hat{\boldsymbol{n}}_{L,i}] \, \mathrm{d}S_{\boldsymbol{\xi}}
\end{aligned}$$

where $\partial\Omega_{o,i}^s$ is as in Figure 8.1 and we used $g_L(-1) = g_R(1) = 1$, $g_L(-1) = g_R(1) = 0$. Then substituting these identities into (8.31) gives us the conservative update of the cell average (8.29).

8.3.3. Approximate Lax-Wendroff procedure

We now illustrate how to approximate the time average flux at the solution points $\tilde{\boldsymbol{F}}_{e,\boldsymbol{p}}$ which is required to construct the element local approximation $\tilde{\boldsymbol{F}}_e^\delta(\boldsymbol{\xi})$ using the approximate Lax-Wendroff procedure [208]. For $N = 1$, (8.24) requires $\partial_t \tilde{\boldsymbol{f}}$ which is approximated as

$$\partial_t \tilde{\boldsymbol{f}}^\delta(\boldsymbol{\xi}_{\boldsymbol{p}}) = \frac{\tilde{\boldsymbol{f}}(\boldsymbol{u}_{\boldsymbol{p}} + \Delta t \, (\boldsymbol{u}_t)_{\boldsymbol{p}}) - \tilde{\boldsymbol{f}}(\boldsymbol{u}_{\boldsymbol{p}} - \Delta t \, (\boldsymbol{u}_t)_{\boldsymbol{p}})}{2 \, \Delta t} \tag{8.32}$$

where element index e is suppressed as all these operations are local to each element. The time index is also suppressed as all quantities are used from time level t_n. The $\boldsymbol{u}_t$ above is approximated using (8.12)

$$(\boldsymbol{u}_t)_{\boldsymbol{p}} = -\frac{1}{J_{\boldsymbol{p}}} \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}^\delta(\boldsymbol{\xi}_{\boldsymbol{p}}) \tag{8.33}$$

where $\tilde{\boldsymbol{f}}^{\delta}$ is the cell local approximation to the flux $\tilde{\boldsymbol{f}}$ given in (8.17). For $N=2$, (8.24) additionally requires $\partial_{tt}\,\tilde{\boldsymbol{f}}$

$$\begin{aligned}&\partial_{tt}\,\tilde{\boldsymbol{f}}^{\delta}(\boldsymbol{\xi}_{\boldsymbol{p}})\\ &= \frac{1}{\Delta t^2}\left[\tilde{\boldsymbol{f}}\left(\boldsymbol{u}_{\boldsymbol{p}}+\Delta t(\boldsymbol{u}_t)_{\boldsymbol{p}}+\frac{\Delta t}{2}(\boldsymbol{u}_{tt})_{\boldsymbol{p}}\right)-2\,\tilde{\boldsymbol{f}}(\boldsymbol{u}_{\boldsymbol{p}})+\tilde{\boldsymbol{f}}\left(\boldsymbol{u}_{\boldsymbol{p}}-\Delta t(\boldsymbol{u}_t)_{\boldsymbol{p}}+\frac{\Delta t}{2}(\boldsymbol{u}_{tt})_{\boldsymbol{p}}\right)\right]\end{aligned}$$

where the element index e is again suppressed. We approximate $\boldsymbol{u}_{tt}$ as

$$(\boldsymbol{u}_{tt})_{\boldsymbol{p}} = -\frac{1}{J_{\boldsymbol{p}}}\,\nabla_{\boldsymbol{\xi}}\cdot\partial_t\tilde{\boldsymbol{f}}^{\delta}(\boldsymbol{\xi}_{\boldsymbol{p}}) \tag{8.34}$$

The procedure for other degrees will be similar and the derivatives $\nabla_{\boldsymbol{\xi}}$ are computed using a differentiation matrix. The implementation is made efficient by accounting for cancellations of Δt terms, which will be the same as in Section 4.2.4 for Cartesian meshes.

8.3.4. Free stream preservation for LWFR

If the solution of the conservation law (8.1) is spatially constant at a time level, then its evolution at later time levels remains constant[8.1]. This is the free stream property and it is crucial for the numerical scheme to satisfy it in order to avoid very large errors [105]. In this section, we discuss the conditions under which the LWFR scheme is free stream preserving.

The free stream preservation of a conservation law is equivalent to the metric identity (8.14). The divergence in a Flux Reconstruction (FR) scheme (8.22) is computed as the derivative of a polynomial. Thus, denoting $\partial^h_{\xi_i}$ as a polynomial derivative, which is computed using a differentiation matrix (3.5), the following *discrete metric identity* (8.14) is at least required for our scheme to preserve a constant state

$$\sum_{i=1}^{d}\partial^h_{\xi_i}(J\boldsymbol{a}^i) = \sum_{i=1}^{d}\partial_{\xi_i} I_N(J\boldsymbol{a}^i) = \boldsymbol{0} \tag{8.35}$$

where I_N is the degree N interpolation operator defined as

$$I_N(f)(\boldsymbol{\xi}) = \sum_{\boldsymbol{p}}\ell_{\boldsymbol{p}}(\boldsymbol{\xi})\,f(\boldsymbol{\xi}_{\boldsymbol{p}}) \tag{8.36}$$

The study of free-stream preservation for Discontinuous Galerkin (DG) methods was made in [105] showing that satisfying (8.35) gives free stream preservation. However, it was also shown that the identities impose additional constraints on the degree of the reference map Θ. The remedy given in (8.35) is to replace the metric terms $J\boldsymbol{a}^i$ by a different degree N approximation $\mathcal{I}_N(J\boldsymbol{a}^i)$ so that (8.35) reduces to

$$\sum_{i=1}^{d}\partial^h_{\xi_i}\mathcal{I}_N(J\boldsymbol{a}^i) = \sum_{i=1}^{d}\partial_{\xi_i} I_N\mathcal{I}_N(J\boldsymbol{a}^i) = \sum_{i=1}^{d}\partial_{\xi_i}\mathcal{I}_N(J\boldsymbol{a}^i) = \boldsymbol{0} \tag{8.37}$$

8.1. This assumes that boundary conditions are either specified by the same constant, or are periodic.

In [105], choices of $\mathcal{I}_N$ like the *conservative curl form* were proposed which ensured (8.37) without any additional constraints on the degree of the reference map Θ. Those choices are only relevant in 3-D as, in 2-D, they are equivalent to interpolating Θ to a degree N polynomial before computing the metric terms which is the choice of $\mathcal{I}_N$ we make in this work by defining the reference map as in (8.4).

In this section, we show that the identities (8.35) are actually enough to ensure free stream preservation for LWFR. Throughout this section, we assume that the mesh is *well-constructed* [105] which is a property that follows from the natural assumption of global continuity of the reference map.

DEFINITION 8.4. *Consider a mesh where element faces in reference element Ω_o are denoted as $\{\partial\Omega^S_{o,i}\}$ for coordinate directions $1\le i\le d$ and $S=L/R$ chosen so that the corresponding reference normals $\{\hat{\boldsymbol{n}}_{S,i}\}$ are $\hat{\boldsymbol{n}}_{R,i}=\boldsymbol{e}_i$ and $\hat{\boldsymbol{n}}_{L,i}=-\hat{\boldsymbol{n}}_{R,i}$ where $\{\boldsymbol{e}_i\}_{i=1}^d$ is the Cartesian basis, see Figure 8.1. The mesh is said to be well-constructed if the following is satisfied*

$$\sum_{m=1}^{d}\left(\mathcal{I}_N(J\boldsymbol{a}^m)^+-\mathcal{I}_N(J\boldsymbol{a}^m)^-\right)(\hat{\boldsymbol{n}}_{S,i})_m=\boldsymbol{0},\qquad\forall 1\le i\le d,\quad S=L,R \tag{8.38}$$

where $\pm$ are used to denote trace values from Ω_o or from the neighbouring element respectively.

Remark 8.5. From (8.9), the identity (8.38) can be seen as a property of continuity of the tangential derivatives of the reference map at the faces and is thus obtained if the reference map is globally continuous. Also, since the unit normal vector of an element at interface i is given by $J\boldsymbol{a}^i/|J\boldsymbol{a}^i|$, (8.38) also gives us continuity of the unit normal across interfaces.

Assuming the current solution is constant in space, $\boldsymbol{u}^n=\boldsymbol{c}$, we will begin by proving that the approximate time averaged flux and solution satisfy

$$\tilde{\boldsymbol{F}}^\delta=\tilde{\boldsymbol{f}}^\delta(\boldsymbol{c}),\qquad \boldsymbol{U}=\boldsymbol{u}^\delta=\boldsymbol{c} \tag{8.39}$$

For the constant physical flux $\boldsymbol{f}(\boldsymbol{c})$, the contravariant flux $\tilde{\boldsymbol{f}}$ will be

$$\tilde{\boldsymbol{f}}_i=\mathcal{I}_N(J\boldsymbol{a}^i)\cdot\boldsymbol{f}(\boldsymbol{c})=\sum_{n=1}^{d}\mathcal{I}_N(Ja^i_n)\boldsymbol{f}_n(\boldsymbol{c}) \tag{8.40}$$

Note that the contravariant flux (8.40) is not constant as the metric terms spatially vary for curvilinear grids. However, using (8.33), we obtain at each solution point

$$\begin{aligned}\boldsymbol{u}_t &= -\frac{1}{J}\nabla_{\boldsymbol{\xi}}\cdot\tilde{\boldsymbol{f}}^\delta=-\frac{1}{J}\sum_{i=1}^{d}\partial_{\xi^i}(\tilde{\boldsymbol{f}}^\delta)^i\\ &= -\frac{1}{J}\sum_{i=1}^{d}\partial_{\xi^i}\left(\mathcal{I}_N(J\boldsymbol{a}^i)\cdot\boldsymbol{f}(\boldsymbol{c})\right)=-\frac{1}{J}\left(\sum_{i=1}^{d}\partial_{\xi^i}\mathcal{I}_N(J\boldsymbol{a}^i)\right)\cdot\boldsymbol{f}(\boldsymbol{c})\\ &= \boldsymbol{0}\end{aligned}$$

where the last equality follows from using the metric identity (8.35). For polynomial degree $N=1$, recalling (8.32), this proves that

$$\partial_t \tilde{\boldsymbol{f}}^\delta = \frac{\tilde{\boldsymbol{f}}(\boldsymbol{u}+\Delta t\,\boldsymbol{u}_t)-\tilde{\boldsymbol{f}}(\boldsymbol{u}-\Delta t\,\boldsymbol{u}_t)}{2\,\Delta t}=\boldsymbol{0}$$

Thus, we obtain

$$\tilde{\boldsymbol{F}}^\delta=\tilde{\boldsymbol{f}}^\delta+\frac{\Delta t}{2}\,\partial_t\,\tilde{\boldsymbol{f}}^\delta=\tilde{\boldsymbol{f}}^\delta,\qquad \boldsymbol{U}=\boldsymbol{u}^\delta+\frac{\Delta t}{2}\,\partial_t\,\boldsymbol{u}^\delta=\boldsymbol{u}^\delta$$

Building on this, for $N=2$, by (8.34),

$$\boldsymbol{u}_{tt}=-\frac{1}{J}\nabla_{\boldsymbol{\xi}}\cdot\partial_t\,\tilde{\boldsymbol{f}}^\delta=\boldsymbol{0}$$

which will prove $\partial_{tt}\tilde{\boldsymbol{f}}^\delta=\boldsymbol{0}$ and we similarly obtain the following for all degrees

$$\tilde{\boldsymbol{F}}^\delta = \sum_{k=0}^{N}\frac{\Delta t^k}{(k+1)!}\,\partial_t^k\,\tilde{\boldsymbol{f}}^\delta=\tilde{\boldsymbol{f}}^\delta=\{J\boldsymbol{a}^i\cdot\boldsymbol{f}(\boldsymbol{c})\}_{i=1}^d \tag{8.41}$$

$$\boldsymbol{U} = \sum_{k=0}^{N}\frac{\Delta t^k}{(k+1)!}\,\partial_t^k\,\boldsymbol{u}^\delta=\boldsymbol{u}^\delta=\boldsymbol{c} \tag{8.42}$$

To prove free stream preservation, we argue that the update (8.28) vanishes as the volume terms involving divergence of $\tilde{\boldsymbol{F}}^\delta$ and the surface terms involving trace values and numerical flux vanish. By (8.41), the volume terms in (8.28) are given by

$$\frac{1}{J}\Delta t\left(\sum_{i=1}^{d}\partial_{\xi^i}\mathcal{I}_N(J\boldsymbol{a}^i)\right)\cdot\boldsymbol{f}(\boldsymbol{c})$$

and vanish by the metric identity (8.37). By (8.42), the dissipative part of the numerical flux (8.26) is computed with the constant solution $\boldsymbol{u}^n=\boldsymbol{c}$ and will thus vanish. For the central part of the numerical flux, as the mesh is well-constructed (Definition 8.4), the trace values are given by

$$\begin{aligned}(\tilde{\boldsymbol{F}}^\delta\cdot\hat{\boldsymbol{n}}_{S,i})^+ &= \sum_{m=1}^{d}(\mathcal{I}_N(J\boldsymbol{a}^m)\cdot\boldsymbol{f}(\boldsymbol{c}))^+(\hat{\boldsymbol{n}}_{S,i})_m=\sum_{m=1}^{d}(\mathcal{I}_N(J\boldsymbol{a}^m)\cdot\boldsymbol{f}(\boldsymbol{c}))^-(\hat{\boldsymbol{n}}_{S,i})_m\\ &= (\tilde{\boldsymbol{F}}^\delta\cdot\hat{\boldsymbol{n}}_{S,i})^-\end{aligned}$$

Overall, the numerical flux (8.26) satisfies

$$(\tilde{\boldsymbol{F}}_e\cdot\hat{\boldsymbol{n}}_{S,i})^*=(\tilde{\boldsymbol{F}}^\delta\cdot\hat{\boldsymbol{n}}_{S,i})^+=(\tilde{\boldsymbol{F}}^\delta\cdot\hat{\boldsymbol{n}}_{S,i})^-$$

That is, the numerical flux agrees with the physical flux at element interfaces, ensuring that the surface terms in (8.28) vanish.

8.4. Shock capturing and admissibility preservation

As is the case for problems involving Cartesian grids (Chapters 4-7), most practical problems involving hyperbolic conservation laws on curvilinear grids consist of non-smooth solutions containing shocks and other discontinuities. Thus, we develop the

blending scheme from Section 5.3.1 for curvilinear grids. In order to be compatible with `Trixi.jl` [141] and make use of this excellent code, we introduce LWFR with blending scheme for Gauss-Legendre-Lobatto solution points, which are also used in `Trixi.jl`. As in Chapter 5, the blending scheme has to be constructed to be provably admissibility preserving (Definition 5.1), which is obtained by the weaker admissibility preservation in means (Definition 5.2) property using the scaling limiter ([205], Appendix F).

As in Section 5.3.1, we will subdivide each element into subcells (Figure 8.2) and perform a first order finite volume evolution on the subcells which will be blended with the high order scheme LWFR scheme using a smoothness indicator as in (5.6). We begin by discussing how to construct the low order scheme on the curved element and subcells.

8.4.1. Low order scheme for curvilinear grids

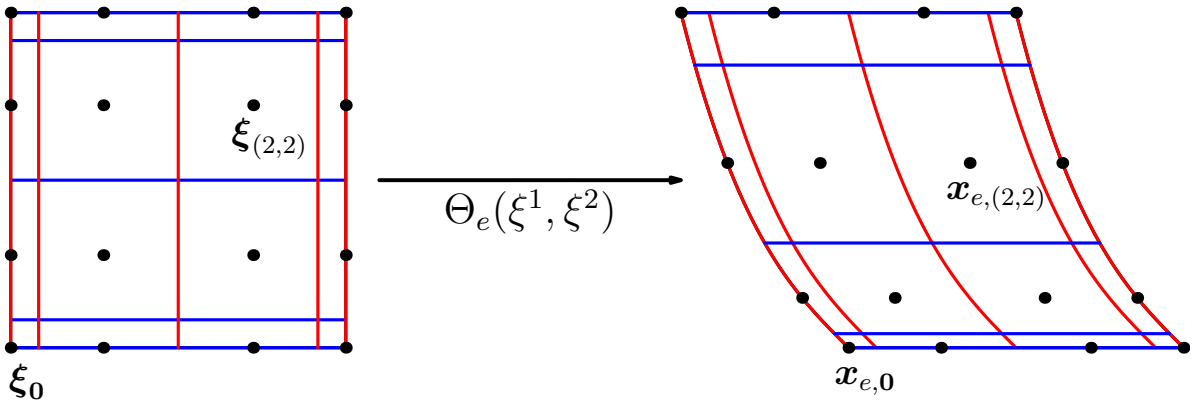


Figure 8.2. Subcells in a curved FR element

The subcells for a curved element will be defined by the reference map, as shown in Figure 8.2. As in Appendix B.3 of [90], the finite volume formulation on subcells is obtained by an integral formulation of the transformed conservation law (8.12). In the reference element, consider subcells $C_{\boldsymbol{p}}$ of size $w_{\boldsymbol{p}} = \prod_{i=1}^{d} w_{p_i}$ associated to the solution point $(\xi_{p_i})_{i=1}^{d}$ corresponding to the multi-index $\boldsymbol{p} = (p_i)_{i=1}^{d}$, where $p_i \in \{0, 1 \ldots, N\}$. Fix a subcell $C = C_{\boldsymbol{p}}$ around the solution point $\boldsymbol{\xi} = (\xi_{p_i})_{i=1}^{d}$ and denote $\boldsymbol{\xi}_i^{L/R}$ as in (8.16). Integrate the conservation law on the fixed subcell C

$$\int_C J \boldsymbol{u}_t \, \mathrm{d}V + \int_C \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}} \, \mathrm{d}V = \mathbf{0}$$

Next, perform one point quadrature in the first term and apply the Gauss divergence theorem on the second term to get

$$J_{e,\boldsymbol{p}} \frac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{p}}}{\mathrm{d}t} w_{\boldsymbol{p}} + \int_{\partial C} \tilde{\boldsymbol{f}} \cdot \hat{\boldsymbol{n}} \, \mathrm{d}A = \mathbf{0} \tag{8.43}$$

where $\hat{\boldsymbol{n}}$ is the reference normal vector on the subcell surface. Now evaluate this surface integral by approximating fluxes in each direction with numerical fluxes

$$\int_{\partial C} \tilde{\boldsymbol{f}} \cdot \hat{\boldsymbol{n}} \mathrm{d}A = \sum_{i=1}^{d} \frac{w_{\boldsymbol{p}}}{w_{p_i}} [(\tilde{\boldsymbol{f}}_C^{\delta} \cdot \hat{\boldsymbol{n}}_{R,i})^*(\boldsymbol{\xi}_i^R) + (\tilde{\boldsymbol{f}}_C^{\delta} \cdot \hat{\boldsymbol{n}}_{L,i})^*(\boldsymbol{\xi}_i^L)], \quad \hat{\boldsymbol{n}}_{R,i} = \boldsymbol{e}_i, \quad \hat{\boldsymbol{n}}_{L,i} = -\boldsymbol{e}_i \tag{8.44}$$

The explicit lower order method using forward Euler update is thus given by

$$\boldsymbol{u}_{\boldsymbol{p}}^{n+1} = \boldsymbol{u}_{\boldsymbol{p}}^{n} - \frac{\Delta t}{J_{e,\boldsymbol{p}}} \sum_{i=1}^{d} \frac{1}{w_{p_i}} [(\tilde{\boldsymbol{f}}_{C_{\boldsymbol{p}}}^{\delta} \cdot \hat{\boldsymbol{n}}_{R,i})^*(\boldsymbol{\xi}_i^R) + (\tilde{\boldsymbol{f}}_{C_{\boldsymbol{p}}}^{\delta} \cdot \hat{\boldsymbol{n}}_{L,i})^*(\boldsymbol{\xi}_i^L)] \tag{8.45}$$

For the subcells whose interfaces are not shared by the FR element, the fluxes are computed, following [90], as

$$\begin{aligned}
(\tilde{\boldsymbol{f}}_{C_{\boldsymbol{p}}}^{\delta} \cdot \hat{\boldsymbol{n}}_{R,i})^*(\boldsymbol{\xi}_i^R) &= \|(\boldsymbol{n}_{R,i})_{\boldsymbol{p}}\| \, \boldsymbol{f}^*\left(\boldsymbol{u}_{\boldsymbol{p}}, \boldsymbol{u}_{\boldsymbol{p}_{i+}}, \frac{(\boldsymbol{n}_{R,i})_{\boldsymbol{p}}}{\|(\boldsymbol{n}_{R,i})_{\boldsymbol{p}}\|} \right) \\
(\tilde{\boldsymbol{f}}_{C_{\boldsymbol{p}}}^{\delta} \cdot \hat{\boldsymbol{n}}_{L,i})^*(\boldsymbol{\xi}_i^L) &= \|(\boldsymbol{n}_{L,i})_{\boldsymbol{p}}\| \, \boldsymbol{f}^*\left(\boldsymbol{u}_{\boldsymbol{p}_{i-}}, \boldsymbol{u}_{\boldsymbol{p}}, \frac{(\boldsymbol{n}_{L,i})_{\boldsymbol{p}}}{\|(\boldsymbol{n}_{L,i})_{\boldsymbol{p}}\|} \right) \\
(\boldsymbol{p}_{i\pm})_m &= \begin{cases} p_m & m \neq i \\ p_{i\pm 1} & m = i \end{cases}
\end{aligned} \tag{8.46}$$

where $(\boldsymbol{n}_{S,i})_{\boldsymbol{p}}$ is the normal vector of subcell $C_{\boldsymbol{p}}$ in direction i and side $S \in \{L, R\}$. The numerical fluxes (8.46) are taken to be Rusanov's flux (8.19)

$$\tilde{\boldsymbol{f}}^*(\boldsymbol{u}^-, \boldsymbol{u}^+, \boldsymbol{n}) = \frac{1}{2}[(\boldsymbol{f} \cdot \boldsymbol{n})(\boldsymbol{u}^+) + (\boldsymbol{f} \cdot \boldsymbol{n})(\boldsymbol{u}^-)] - \frac{\lambda}{2}(\boldsymbol{u}^+ - \boldsymbol{u}^-) \tag{8.47}$$

where λ is the wave speed estimate at the subcell face computed using $\boldsymbol{u}^{\pm}$ (8.20). At the interfaces shared by FR elements, the first order numerical flux is computed by setting $\boldsymbol{u}^{\pm}$ in (8.47) to element trace values as in (8.19). However, the lower order residual needs to be computed using the same inter-element flux as the higher order scheme at interfaces of the Flux Reconstruction (FR) elements. Thus, for example, for an element Ω_e at solution point $\boldsymbol{\xi} = \boldsymbol{\xi}_{\boldsymbol{p}}$ with $\boldsymbol{p} = \boldsymbol{0}$, the subcell update will be given by

$$\boldsymbol{u}_{e,\boldsymbol{0}}^{n+1} = \boldsymbol{u}_{e,\boldsymbol{0}}^{n} - \frac{\Delta t}{J_{e,\boldsymbol{p}}} \sum_{i=1}^{d} \frac{1}{w_{p_i}} [(\tilde{\boldsymbol{f}}_{C_{\boldsymbol{0}}}^{\delta} \cdot \hat{\boldsymbol{n}}_{R,i})^*(\boldsymbol{\xi}_i^R) + (\tilde{\boldsymbol{F}}_e^{\delta} \cdot \hat{\boldsymbol{n}}_{L,i})^*(\boldsymbol{\xi}_i^L)] \tag{8.48}$$

where $(\tilde{\boldsymbol{F}}_e^{\delta} \cdot \hat{\boldsymbol{n}}_i)^*(\boldsymbol{\xi}_i^L)$ is the blended numerical flux and is computed by taking a convex combination of the lower order flux chosen as in (8.19) and the time averaged flux from LWFR scheme (8.26). An initial guess is made as in 1-D (5.11) and then further correction is performed to ensure admissibility, as explained in Section 8.4.3.2. Other subcells neighbouring the element interfaces will also use the blended numerical fluxes at the element interfaces and thus have an update similar to (8.48). Then, by multiplying each update equation of each subcell $\boldsymbol{p}$ by $w_{\boldsymbol{p}}$ and summing over $\boldsymbol{p}$, the conservation property is obtained

$$\bar{\boldsymbol{u}}_e^{L,n+1} = \sum_{\boldsymbol{p}} \boldsymbol{u}_{e,\boldsymbol{p}}^{L,n+1} w_{\boldsymbol{p}} = \bar{\boldsymbol{u}}_e^{n} - \frac{\Delta t}{|\Omega_e|} \left(\sum_{i=1}^{d} \int_{\partial\Omega_{o,i}^R} (\tilde{\boldsymbol{F}}_e^{\delta} \cdot \hat{\boldsymbol{n}}_{R,i})^* \mathrm{d}S_{\boldsymbol{\xi}} + \int_{\partial\Omega_{o,i}^L} (\tilde{\boldsymbol{F}}_e^{\delta} \cdot \hat{\boldsymbol{n}}_{L,i})^* \mathrm{d}S_{\boldsymbol{\xi}} \right) \tag{8.49}$$

Since we also have the conservation property in the higher order scheme (8.29), the blended scheme will be conservative, analogous to the 1-D case (5.9, 5.10).

The expressions for normal vectors on the subcells needed to compute (8.46) are taken from Appendix B.4 of [90] where they were derived by equating the high order flux difference and Discontinuous Galerkin split form. We directly state the normal vectors here, denoting $(\boldsymbol{n}_{R,i})_{\boldsymbol{p}}$ as the outward normal direction in subcell $C_{\boldsymbol{p}}$ along the positive i direction

$$(\boldsymbol{n}_{R,i})_{\boldsymbol{p}} = \mathcal{I}_N(J\boldsymbol{a}^i)_{\boldsymbol{p}_{i|0}} + \sum_{l=0}^{p_i} w_l\, \partial_{\xi^i} \mathcal{I}_N(J\boldsymbol{a}^i)_{\boldsymbol{p}_{i|l}}, \qquad (\boldsymbol{p}_{i|l})_m = \begin{cases} p_m & m \neq i \\ p_l & m = i \end{cases}$$

where $\{w_l\}_{l=0}^N$ are quadrature weights corresponding to solution points, $\mathcal{I}_N$ is the approximation operator for metric terms (8.37), and $(\boldsymbol{n}_{L,i})_{\boldsymbol{p}}$ can be obtained by the relation $(\boldsymbol{n}_{L,i})_{\boldsymbol{p}} = -(\boldsymbol{n}_{R,i})_{\boldsymbol{p}_{i-}}$, where $\boldsymbol{p}_{i-}$ was defined in (8.46).

Free-stream preservation. To show the free stream preservation of the lower order scheme with the chosen normal vectors, we consider a constant initial state $\boldsymbol{u} = \boldsymbol{c}$ and show that the finite volume residual will be zero. A constant state implies that the time average of the contravariant flux will be the contravariant flux itself (8.39). Thus, all numerical fluxes including element interface fluxes are first order fluxes like in (8.46) and the residual at $\boldsymbol{p}$ in direction i is given by

$$\begin{aligned} & \frac{\boldsymbol{f}(\boldsymbol{c})}{w_{p_i}} \cdot ((\boldsymbol{n}_{R,i})_{\boldsymbol{p}} + (\boldsymbol{n}_{L,i})_{\boldsymbol{p}}) \\ = \; & \frac{\boldsymbol{f}(\boldsymbol{c})}{w_{p_i}} \cdot (\mathcal{I}_N(J\boldsymbol{a}^i)_{\boldsymbol{p}_{i|0}} + \sum_{l=0}^{p_i} w_l\, \partial_{\xi^i} \mathcal{I}_N(J\boldsymbol{a}^i)_{\boldsymbol{p}_{i|l}} - \mathcal{I}_N(J\boldsymbol{a}^i)_{\boldsymbol{p}_{i|0}} - \sum_{l=0}^{p_i-1} w_l\, \partial_{\xi^i} \mathcal{I}_N(J\boldsymbol{a}^i)_{\boldsymbol{p}_{i|l}}) \\ = \; & \boldsymbol{f}(\boldsymbol{c}) \cdot \partial_{\xi^i} \mathcal{I}_N(J\boldsymbol{a}^i)_{\boldsymbol{p}} \end{aligned}$$

The residuals in other directions give similar terms and summing them gives

$$\boldsymbol{f}(\boldsymbol{c}) \cdot \sum_{i=1}^{d} \frac{\partial}{\partial \xi^i} \mathcal{I}_N(J\boldsymbol{a}^i)_{\boldsymbol{p}} = \boldsymbol{0}$$

by the metric identities, thus satisfying the free stream preservation condition.

8.4.2. Smoothness indicator

As in Section 5.3.2, we measure the smoothness of degree N approximate solution within each element and in terms of the orthonormal Legendre basis and analyze the decay of its coefficients. In this section, we write the smoothness indicator for d dimensions, using the multi-index notation (8.5).

Let $q = q(\boldsymbol{u})$ be the quantity used to measure the solution smoothness. With $\{L_j\}_{j=0}^N$ being the 1-D Legendre polynomial basis of degree N, taking tensor product gives the degree N Legendre basis

$$L_{\boldsymbol{p}}(\boldsymbol{\xi}) = \prod_{i=1}^{d} L_{p_i}(\xi^i), \qquad p_i \in \{0, 1, \ldots, N\}$$

The Legendre basis representation of q can be obtained as

$$q_h(\boldsymbol{\xi}) = \sum_{\boldsymbol{p}} \hat{q}_{\boldsymbol{p}}\, L_{\boldsymbol{p}}(\boldsymbol{\xi}), \quad \boldsymbol{\xi} \in \Omega_o, \qquad \hat{q}_{\boldsymbol{p}} = \int_{\Omega_o} q(\boldsymbol{u}^\delta(\boldsymbol{\xi}))\, L_{\boldsymbol{p}}(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi}$$

The Legendre coefficients $\{\hat{q}_{\boldsymbol{p}}\}$ are computed using the quadrature induced by the solution points,

$$\hat{q}_{\boldsymbol{p}} = \sum_{\boldsymbol{q}} q(\boldsymbol{u}_{e,\boldsymbol{q}})\, L_{\boldsymbol{p}}(\boldsymbol{\xi}_{\boldsymbol{q}})\, w_{\boldsymbol{q}}$$

Define

$$\mathbb{S}_K = \sum_{\boldsymbol{p}, p_i \leq K} \hat{q}_{\boldsymbol{p}}^2$$

which measures the "energy" in q_h. Then the energy contained in the highest modes relative to the total energy of the polynomial is computed as follows

$$\mathbb{E} = \max\left\{ \frac{\mathbb{S}_N - \mathbb{S}_{N-1}}{\mathbb{S}_N}, \frac{\mathbb{S}_{N-1} - \mathbb{S}_{N-2}}{\mathbb{S}_{N-1}} \right\}$$

In 1-D, this simplifies into the expression of (5.12) and the remaining steps to obtain the blending coefficient $\alpha_e \in [0, 1]$ are the same as in Section 5.3.2.

8.4.3. Flux limiter for admissibility preservation

We first briefly review the flux limiting process for admissibility preservation from Chapter 5 for 1-D and then do a natural extension to curvilinear meshes. The first step in obtaining an admissibility preserving blending scheme is to ensure that the lower order scheme preserves the admissibility set $\mathcal{U}_{\text{ad}}$. This is always true if all the fluxes in the lower order method are computed with an admissibility preserving low order finite volume method. But the LWFR scheme uses a time average numerical flux and maintaining conservation requires that we use the same numerical flux at the element interfaces for both lower and higher order schemes (Remark 1 of [19]). To maintain accuracy and admissibility, we carefully choose a blended numerical flux $\boldsymbol{F}_{e+\frac{1}{2}}$ as in (5.11) but this choice may not ensure admissibility and further limitation is required. Our proposed procedure for choosing the blended numerical flux will give us an admissibility preserving lower order scheme. As a result of using the same numerical flux at element interfaces in both high and low order schemes, element means of both schemes are the same (Theorem 8.6). A consequence of this is that our scheme now preserves admissibility of element means and thus we can use the scaling limiter of [205] to get admissibility at all solution points.

Once the low order scheme on subcells is constructed as in Section 8.4.1, the blending scheme with smoothness coefficient can be written as (5.6). The theoretical basis for flux limiting summarized in Theorem 5.5 also applies. For clarity, we rewrite Theorem 5.5 in the notation of general curvilinear specialized to first order blending in Theorem 8.6.

Theorem 8.6. *Consider the LWFR blending scheme on curved meshes where low and high order schemes use the same numerical flux* $(\tilde{\boldsymbol{F}}_e^{\delta} \cdot \hat{\boldsymbol{n}}_i)^*(\boldsymbol{\xi}_i^s)$ *at every element interface and the lower order residual is computed using the first order finite volume scheme* (8.45)*. Then the following can be said about admissibility preserving in means property (Definition 5.2) of the scheme:*

1. *element means of both low and high order schemes are same, and thus the blended scheme is admissibility preserving in means if and only if the lower order scheme is admissibility preserving in means;*

2. *if the blended numerical flux* $(\tilde{\boldsymbol{F}}_e^{\delta} \cdot \hat{\boldsymbol{n}}_i)^*(\boldsymbol{\xi}_i^s)$ *is chosen to preserve the admissibility of lower-order updates at solution points adjacent to the interfaces, then the blending scheme will preserve admissibility in means.*

Proof. By (8.29, 8.49), element means are the same for both low and high order schemes. Thus, admissibility in means of one implies the same for the other, proving the first claim. For the second claim, note that our assumptions imply $\boldsymbol{u}_{e,\boldsymbol{p}}^{L,n+1}$ given by (8.45, 8.48) are in $\mathcal{U}_{\text{ad}}$ for all $\boldsymbol{p}$. Therefore, we obtain admissibility in means property of the lower order scheme by (8.49) and thus admissibility in means for the blended scheme. □

8.4.3.1. Flux limiter for admissibility preservation in 1-D

To make the general case of curved meshes easier to understand, we keep the 1-D Flux limiter in Section 5.5 in a self-contained version in Algorithm 8.1. For simplicity, we only consider the case where the admissibility constraints P_k (5.1) are concave functions of the conservative variables.

Algorithm 8.1

Computation of blended flux $\boldsymbol{F}_{e+\frac{1}{2}}$

Input: $\boldsymbol{F}_{e+\frac{1}{2}}^{\text{LW}}, \boldsymbol{f}_{e+\frac{1}{2}}, \boldsymbol{f}_{\frac{1}{2}}^{e+1}, \boldsymbol{f}_{N-\frac{1}{2}}^{e}, \boldsymbol{u}_{e+1,0}^{n}, \boldsymbol{u}_{e,N}^{n}, \alpha_e, \alpha_{e+1}$

Output: $\boldsymbol{F}_{e+\frac{1}{2}}$

$\alpha_{e+\frac{1}{2}} = \frac{\alpha_e + \alpha_{e+1}}{2}$

$\boldsymbol{F}_{e+\frac{1}{2}} \leftarrow (1 - \alpha_{e+\frac{1}{2}})\, \boldsymbol{F}_{e+\frac{1}{2}}^{\text{LW}} + \alpha_{e+\frac{1}{2}}\, \boldsymbol{f}_{e+\frac{1}{2}}$ ▷Heuristic guess to control oscillations

▷FV inner updates with guessed $\boldsymbol{F}_{e+\frac{1}{2}}$

$\hat{\boldsymbol{u}}_0^{n+1} \leftarrow \boldsymbol{u}_{e+1,0}^{n} - \frac{\Delta t}{w_0 \Delta x_{e+1}} (\boldsymbol{f}_{\frac{1}{2}}^{e} - \boldsymbol{F}_{e+\frac{1}{2}})$

$\hat{\boldsymbol{u}}_N^{n+1} \leftarrow \boldsymbol{u}_{e,N}^{n} - \frac{\Delta t}{w_N \Delta x_e} (\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{f}_{N-\frac{1}{2}}^{e})$

▷FV inner updates with $\boldsymbol{f}_{e+\frac{1}{2}}$ which are admissibile

$\hat{\boldsymbol{u}}_0^{\text{low},n+1} = \boldsymbol{u}_{e+1,0}^{n} - \frac{\Delta t}{w_0 \Delta x_{e+1}} (\boldsymbol{f}_{\frac{1}{2}}^{e+1} - \boldsymbol{f}_{e+\frac{1}{2}})$

$\hat{\boldsymbol{u}}_N^{\text{low},n+1} = \boldsymbol{u}_{e,N}^{n} - \frac{\Delta t}{w_N \Delta x_e} (\boldsymbol{f}_{e+\frac{1}{2}} - \boldsymbol{f}_{N-\frac{1}{2}}^{e})$

for $k = 1{:}K$ **do**

▷Correct $\boldsymbol{F}_{e+\frac{1}{2}}$ for K admissibility constraints

$\epsilon_0, \epsilon_N \leftarrow \frac{1}{10} P_k(\hat{\boldsymbol{u}}_0^{\text{low},n+1}), \frac{1}{10} P_k(\hat{\boldsymbol{u}}_N^{\text{low},n+1})$

$\theta \leftarrow \min\left(\min_{j=0,N} \left| \frac{\epsilon_j - P_k(\hat{\boldsymbol{u}}_j^{\text{low},n+1}))}{P_k(\hat{\boldsymbol{u}}_j^{n+1}) - P_k(\hat{\boldsymbol{u}}_j^{\text{low},n+1})} \right|, 1 \right)$

$\boldsymbol{F}_{e+\frac{1}{2}} \leftarrow \theta\, \boldsymbol{F}_{e+\frac{1}{2}} + (1 - \theta)\, \boldsymbol{f}_{e+\frac{1}{2}}$

▷FV inner updates with $\boldsymbol{F}_{e+\frac{1}{2}}$ corrected for P_k

$\hat{\boldsymbol{u}}_0^{n+1} \leftarrow \boldsymbol{u}_{e+1,0}^{n} - \frac{\Delta t}{w_0 \Delta x_{e+1}} (\boldsymbol{f}_{\frac{1}{2}}^{e} - \boldsymbol{F}_{e+\frac{1}{2}})$

$\hat{\boldsymbol{u}}_N^{n+1} \leftarrow \boldsymbol{u}_{e,N}^{n} - \frac{\Delta t}{w_N \Delta x_e} (\boldsymbol{F}_{e+\frac{1}{2}} - \boldsymbol{f}_{N-\frac{1}{2}}^{e})$

end

8.4.3.2. Flux limiter for admissibility preservation on curved meshes

Consider the calculation of the blended numerical flux for a corner solution point of the element, see Figure 8.2. A corner solution point is adjacent to interfaces in all d directions, making its admissibility preservation procedure different from 1-D. In particular, let us consider the corner solution point $\boldsymbol{p}=\mathbf{0}$ and show how we can apply the 1-D procedure in Section 8.4.3.1 to ensure admissibility at such points. The same procedure applies to other corner and non-corner points. The lower order update at the corner is given by (8.48)

$$\hat{\boldsymbol{u}}_{e,\mathbf{0}}^{n+1} \;=\; \boldsymbol{u}_{e,\mathbf{0}}^{n} - \frac{\Delta t}{J_{e,\boldsymbol{p}}} \sum_{i=1}^{d} \frac{1}{w_{p_i}} [(\tilde{\boldsymbol{f}}_{C_{\mathbf{0}}}^{\delta} \cdot \hat{\boldsymbol{n}}_{R,i})^*(\boldsymbol{\xi}_i^R) + (\hat{\boldsymbol{F}}_e^{\delta} \cdot \hat{\boldsymbol{n}}_{L,i})^*(\boldsymbol{\xi}_i^L)] \tag{8.50}$$

where $\hat{\boldsymbol{n}}_i = \boldsymbol{e}_i$ is the reference normal vector on the subcell interface in direction i, $(\tilde{\boldsymbol{f}}_{C_{\mathbf{0}}}^{\delta} \cdot \hat{\boldsymbol{n}}_{R,i})^*$ denotes the lower order flux (8.44) at the subcell $C_{\mathbf{0}}$ surrounding $\boldsymbol{\xi}_{\mathbf{0}}$, $(\hat{\boldsymbol{F}}_e^{\delta} \cdot \hat{\boldsymbol{n}}_{L,i})^*(\boldsymbol{\xi}_i^L)$ is the initial guess candidate for the blended numerical flux. Pick $k_i > 0$ such that $\sum_{i=1}^{d} k_i = 1$ and

$$\hat{\boldsymbol{u}}_i^{\text{low},n+1} := \boldsymbol{u}_{e,\mathbf{0}}^{n} - \frac{\Delta t}{k_i\, w_{p_i}\, J_{e,\boldsymbol{p}}} [(\tilde{\boldsymbol{f}}_{C_{\mathbf{0}}}^{\delta} \cdot \hat{\boldsymbol{n}}_{R,i})^*(\boldsymbol{\xi}_i^R) + (\tilde{\boldsymbol{f}} \cdot \hat{\boldsymbol{n}}_{L,i})^*(\boldsymbol{\xi}_i^L)], \qquad 1 \le i \le d \tag{8.51}$$

satisfy

$$\hat{\boldsymbol{u}}_i^{\text{low},n+1} \in \mathcal{U}_{\text{ad}}, \qquad 1 \le i \le d \tag{8.52}$$

where $(\tilde{\boldsymbol{f}} \cdot \hat{\boldsymbol{n}}_i)^*(\boldsymbol{\xi}_i^L)$ is the first order finite volume flux computed at the FR element interface.

The $\{k_i\}$ that ensure (8.52) will exist provided the appropriate CFL restrictions are satisfied because the lower order scheme using the first order numerical flux at element interfaces is admissibility preserving. The choice of $\{k_i\}$ should be made so that (8.52) is satisfied with the least time step restriction. However, we make the trivial choice of equal k_i's motivated by the experience of Chapter 5, where it was found that even this choice does not impose any additional time step constraints over the Fourier stability limit. After choosing k_i's, we have reduced the update to 1-D and can repeat the same procedure as in Algorithm 8.1 where for all directions i, the neighbouring element is chosen along the normal direction. After the flux limiting is performed following Algorithm 8.1, we obtain $(\hat{\boldsymbol{F}}_e^{\delta} \cdot \hat{\boldsymbol{n}}_{L,i})^*(\boldsymbol{\xi}_i^L)$ such that

$$\hat{\boldsymbol{u}}_i^{n+1} := \boldsymbol{u}_{e,\mathbf{0}}^{n} - \frac{\Delta t}{k_i\, w_{p_i}\, J_{e,\boldsymbol{p}}} [(\tilde{\boldsymbol{f}}_{C_{\mathbf{0}}}^{\delta} \cdot \hat{\boldsymbol{n}}_{R,i})^*(\boldsymbol{\xi}_i^R) + (\hat{\boldsymbol{F}}^{\delta} \cdot \hat{\boldsymbol{n}}_{L,i})^*(\boldsymbol{\xi}_i^L)] \in \mathcal{U}_{\text{ad}} \tag{8.53}$$

Then, we will get

$$\sum_{i=1}^{d} k_i\, \hat{\boldsymbol{u}}_i^{n+1} = \hat{\boldsymbol{u}}_{e,\mathbf{0}}^{n+1} \in \mathcal{U}_{\text{ad}} \tag{8.54}$$

along with admissibility of all other corner and non-corner solution points where the flux $(\hat{\boldsymbol{F}}^{\delta} \cdot \hat{\boldsymbol{n}}_i)^*(\boldsymbol{\xi}_i^R)$ is used. Finally, by Theorem 8.6, admissibility in means (Definition 8.30) is obtained and the scaling limiter of [205] can be used to obtain an admissibility preserving scheme (Definition 5.1).

8.5. ADAPTIVE MESH REFINEMENT

Adaptive mesh refinement helps resolve flows where the relevant features are localized to certain regions of the physical domain by increasing the mesh resolution in those regions and coarsening in the rest of the domain. In this work, we allow the adaptively refined meshes to be non-conforming, i.e., element neighbours need not have coinciding solution points at the interfaces (Figure 8.3a). We handle the non-conformality using the *mortar element method* first introduced for hyperbolic PDEs in [106].

In order to perform the transfer of solution during coarsening and refinement, we introduce some notations and operators. Define the 1-D reference elements

$$I_0 = [-1, 0], \qquad I_1 = [0, 1], \qquad I = [-1, 1], \qquad \mathbb{N}_1^d = \{0, 1\}^d \tag{8.55}$$

and the bijections $\phi_s \colon I_s \to I$ for $s = 0, 1$ as

$$\phi_0(\xi) = 2\,\xi + 1, \quad \xi \in I_0, \qquad \phi_1(\xi) = 2\,\xi - 1, \quad \xi \in I_1 \tag{8.56}$$

so that the inverse maps $\phi_s^{-1} \colon I \to I_s$ are given by

$$\phi_0^{-1}(\xi) = \frac{\xi - 1}{2}, \quad \xi \in I, \qquad \phi_1^{-1}(\xi) = \frac{\xi + 1}{2}, \quad \xi \in I \tag{8.57}$$

Denoting the 1-D solution points and Lagrange basis for I as $\{\xi_p\}_{p=0}^N$ and $\{\ell_p(\xi)\}_{p=0}^N$ respectively, the same for I_s are given by $\{\phi_s^{-1}(\xi_p)\}_{p=0}^N$ and $\{\ell_p(\phi_s(\xi))\}_{p=0}^N$ respectively. We also define $⨍$ to be integration under quadrature at solution points. Thus,

$$⨍_I u(\xi)\,\mathrm{d}\xi = \sum_{p=0}^{N} u(\xi_p)\,w_p, \qquad ⨍_{I_s} u(\xi)\,\mathrm{d}\xi = \sum_{p=0}^{N} 2\,u(\phi_s^{-1}(\xi_p))\,w_p$$

In order to get the solution point values of the refined elements, we will perform interpolation. All integrals in this section are approximated by quadrature at solution points which are the degree N Gauss-Legendre-Lobatto points. The interpolation operator from I to $\{I_s\}_{s=0,1}$ is given by $\mathtt{V}_{\Xi_s}$ defined as the Vandermonde matrix corresponding to the Lagrange basis

$$(\mathtt{V}_s)_{pq} = \ell_q(\phi_s^{-1}(\xi_p)), \qquad 0 \le p, q \le N, \qquad s = 0, 1 \tag{8.58}$$

For the process of coarsening, we also define the L^2 projection operators $\{\mathcal{P}^s\}_{s=0,1}$ which projects a polynomial u defined on the Lagrange basis of I_s to the Lagrange basis of I as

$$⨍_I \mathcal{P}^s(u)(\xi)\,\ell_i(\xi)\,\mathrm{d}\xi = ⨍_{I_s} u(\xi)\,\ell_i(\xi)\,\mathrm{d}\xi, \qquad 0 \le i \le N$$

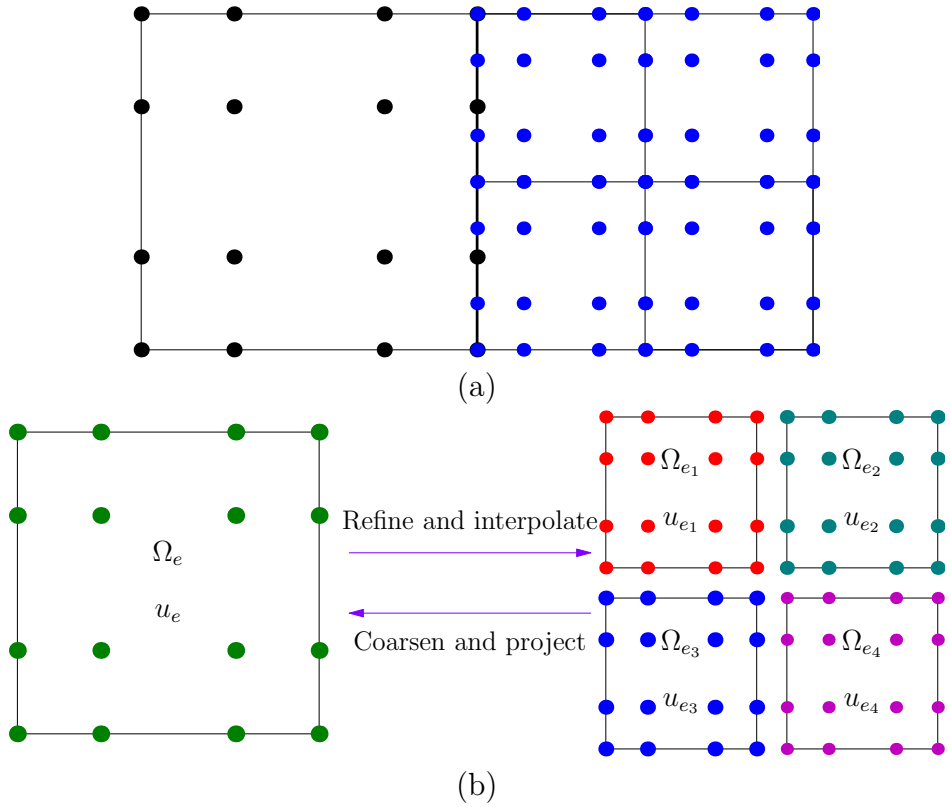


Figure 8.3. (a) Neighbouring elements with hanging nodes (b) Illustration of refinement and coarsening

Approximating the integrals by quadrature on solution points, we obtain the matrix representations corresponding to the basis

$$\mathcal{P}^s_{pq} = \frac{1}{2}\frac{w_q}{w_p}\,\ell_p(\phi_s^{-1}(\xi_q)), \qquad 0 \le i, j \le N, \quad s = 0, 1 \tag{8.59}$$

where $\{w_p\}_{p=0}^N$ are the quadrature weights corresponding to solution points. The transfer of solution during coarsening and refinement is performed by matrix-vector operations using the operators (8.58, 8.59). Thus, the operators (8.58, 8.59) are stored as matrices for the reference element at the beginning of the simulation and reused for the adaptation operations in all elements. Lastly, we introduce the notation of a product of matrix operators $\{A_i\}_{i=1}^d$ acting on $\boldsymbol{b} = (b_{\boldsymbol{p}})_{\boldsymbol{p}\in\mathbb{N}_N^d} = (b_{p_1p_2p_3})_{\boldsymbol{p}\in\mathbb{N}_N^d}$ as

$$(A_i\,\boldsymbol{b})_{\boldsymbol{p}} = \sum_{\boldsymbol{q}\in\mathbb{N}_N^d} \left(\prod_{i=1}^{d} (A_i)_{p_iq_i}\right) b_{\boldsymbol{q}} \tag{8.60}$$

8.5.1. Solution transfer between element and subelements

Corresponding to the element Ω_e, we denote the 2^d subdivisions as (Figure 8.3b)

$$\Omega_{e_{\boldsymbol{s}}} = \Theta_e\left(\prod_{i=1}^{d} I_{s_i}\right), \qquad \forall \boldsymbol{s} \in \mathbb{N}_1^d$$

where I_s are defined in (8.55). We also define $\phi_{\boldsymbol{s}}(\boldsymbol{\xi}) = (\phi_{s_i}(\xi^i))_{i=1}^d$ so that $\phi_{\boldsymbol{s}}$ is a bijection between $\Omega_{e_{\boldsymbol{s}}}$ and Ω_e in reference coordinates[8.2]. Recall that $\{\ell_{\boldsymbol{p}}\}_{\boldsymbol{p}\in\mathbb{N}_N^d}$ are Lagrange polynomials of degree N with variables $\boldsymbol{\xi} = (\xi^i)_{i=1}^d$. Thus, the reference solution points and Lagrange basis for $\Omega_{e_{\boldsymbol{s}}}$ are given by $\{\phi_{\boldsymbol{s}}^{-1}(\boldsymbol{\xi}_{\boldsymbol{p}})\}_{\boldsymbol{p}\in\mathbb{N}_N^d}$ and $\{\ell_{\boldsymbol{p}}(\phi_{\boldsymbol{s}}(\boldsymbol{\xi}))\}_{\boldsymbol{p}\in\mathbb{N}_N^d}$, respectively. The respective representations of solution approximations in $\Omega_e, \Omega_{e_{\boldsymbol{s}}}$ in reference coordinates are thus given by

$$\boldsymbol{u}_e(\boldsymbol{\xi}) = \sum_{\boldsymbol{q}\in\mathbb{N}_N^d} \ell_{\boldsymbol{q}}(\boldsymbol{\xi})\, \boldsymbol{u}_{e,\boldsymbol{q}}, \qquad \boldsymbol{u}_{e_{\boldsymbol{s}}}(\boldsymbol{\xi}) = \sum_{\boldsymbol{q}\in\mathbb{N}_N^d} \ell_{\boldsymbol{q}}(\phi_{\boldsymbol{s}}(\boldsymbol{\xi}))\, \boldsymbol{u}_{e_{\boldsymbol{s}},\boldsymbol{q}} \tag{8.61}$$

8.5.1.1. Interpolation for refinement

After refining an element Ω_e into child elements $\{\Omega_{e_{\boldsymbol{s}}}\}_{\boldsymbol{s}\in\mathbb{N}_1^d}$, the solution $\boldsymbol{u}_e$ has to be interpolated on the solution points of child elements to obtain $\{\boldsymbol{u}_{e_{\boldsymbol{s}}}\}_{\boldsymbol{s}\in\mathbb{N}_1^d}$. The scheme will be specified by writing $\boldsymbol{u}_{e_{\boldsymbol{s}},\boldsymbol{q}}$ in terms of $\boldsymbol{u}_{e,\boldsymbol{q}}$, which were defined in (8.61). The interpolation is performed as

$$\begin{aligned}
\boldsymbol{u}_{e_{\boldsymbol{s}},\boldsymbol{p}} &= \sum_{\boldsymbol{q}\in\mathbb{N}_N^d} \ell_{\boldsymbol{q}}(\phi_{\boldsymbol{s}}^{-1}(\boldsymbol{\xi}_{\boldsymbol{p}}))\boldsymbol{u}_{e,\boldsymbol{q}} = \sum_{\boldsymbol{q}\in\mathbb{N}_N^d} \left(\prod_{i=1}^d \ell_{q_i}(\phi_{s_i}^{-1}(\boldsymbol{\xi}_{p_i}))\right)\boldsymbol{u}_{e,\boldsymbol{q}} \\
&= \sum_{\boldsymbol{q}\in\mathbb{N}_N^d} \left(\prod_{i=1}^d (\mathtt{V}_{\Xi_{s_i}})_{p_i q_i}\right)\boldsymbol{u}_{e,\boldsymbol{q}}
\end{aligned}$$

In the product of operators notation (8.60), the interpolation can be written as

$$\boldsymbol{u}_{e_{\boldsymbol{s}}} = \left(\prod_{i=1}^d \mathtt{V}_{\Xi_{s_i}}\right)\boldsymbol{u}_e$$

8.5.1.2. Projection for coarsening

When 2^d elements are joined into one single bigger element Ω_e, the solution transfer is performed using L^2 projection of $\{\boldsymbol{u}_{e_{\boldsymbol{s}}}\}_{\boldsymbol{s}\in\mathbb{N}_1^d}$ into $\boldsymbol{u}_e$, which is given by

$$\sum_{\boldsymbol{s}\in\mathbb{N}_1^d} \fint_{\Omega_{e_{\boldsymbol{s}}}} \boldsymbol{u}_{e_{\boldsymbol{s}}} \ell_{\boldsymbol{p}}(\boldsymbol{\xi})\mathrm{d}\boldsymbol{x} = \fint_{\Omega_e} \boldsymbol{u}_e \ell_{\boldsymbol{p}}(\boldsymbol{\xi})\mathrm{d}\boldsymbol{x}, \qquad \forall \boldsymbol{p}\in\mathbb{N}_N^d \tag{8.62}$$

Substituting (8.61) into (8.62) gives

$$\sum_{\boldsymbol{s}\in\mathbb{N}_1^d}\sum_{\boldsymbol{q}\in\mathbb{N}_N^d} \fint_{\Omega_{e_{\boldsymbol{s}}}} \ell_{\boldsymbol{p}}(\boldsymbol{\xi})\, \ell_{\boldsymbol{q}}(\phi_{\boldsymbol{s}}(\boldsymbol{\xi}))\boldsymbol{u}_{e_{\boldsymbol{s}},\boldsymbol{q}}\mathrm{d}\boldsymbol{x} = \sum_{\boldsymbol{q}\in\mathbb{N}_N^d} \fint_{\Omega_e} \ell_{\boldsymbol{p}}(\boldsymbol{\xi})\, \ell_{\boldsymbol{q}}(\boldsymbol{\xi})\boldsymbol{u}_{e,\boldsymbol{q}}\mathrm{d}\boldsymbol{x} \tag{8.63}$$

Note the 1-D identities

$$\begin{aligned}
\fint_I \ell_p(\xi)\, \ell_q(\xi)\mathrm{d}\xi &= \delta_{pq}\, w_p \\
\fint_{I_s} \ell_p(\xi)\, \ell_q(\phi_s(\xi))\mathrm{d}\xi = \frac{1}{2}\fint_{-1}^{+1} \ell_p(\phi_s^{-1}(\xi))\, \ell_q(\xi)\mathrm{d}\xi &= \frac{1}{2}\, \ell_p(\phi_s^{-1}(\xi_q))\, w_q = \mathcal{P}_{pq}^s\, w_p
\end{aligned}$$

8.2. That is, $\Theta_e\, \phi_{\boldsymbol{s}}\, \Theta_e^{-1}$ is a bijection from $\Omega_{e_{\boldsymbol{s}}}$ to Ω_e.

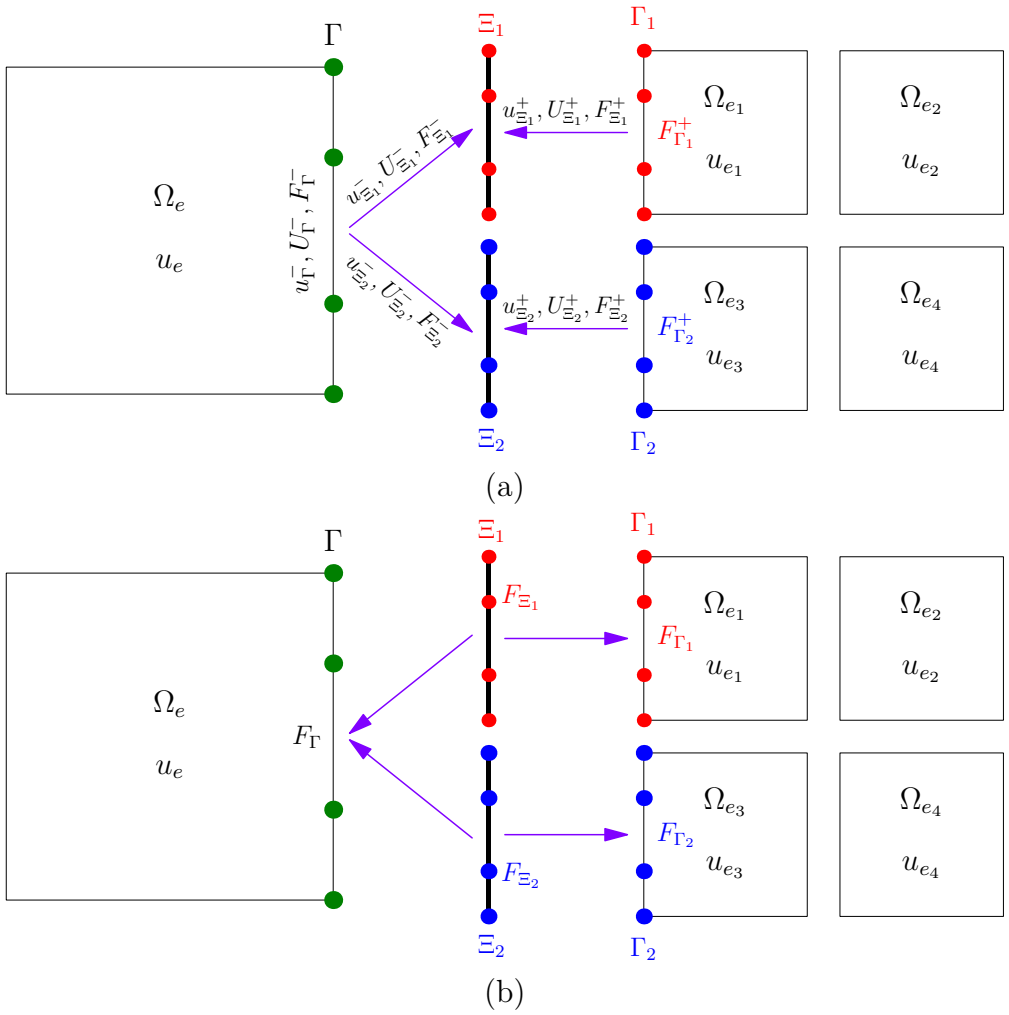


Figure 8.4. (a) Prolongation to mortar and computation of numerical flux $\boldsymbol{F}_{\Xi_1}, \boldsymbol{F}_{\Xi_2}$, (b) Projection of numerical flux to interfaces.

where the projection operator $\{\mathcal{P}^s\}_{s=0,1}$ is defined in (8.59). Then, by change of variables, we have the following

$$\int_{\Omega_e} \ell_{\boldsymbol{p}}(\boldsymbol{\xi})\,\ell_{\boldsymbol{q}}(\boldsymbol{\xi}) = J_{e,\boldsymbol{p}} \prod_{i=1}^{d} w_{p_i}\,\delta_{p_i q_i}, \qquad \int_{\Omega_{e_{\boldsymbol{s}}}} \ell_{\boldsymbol{p}}(\boldsymbol{\xi})\,\ell_{\boldsymbol{q}}(\phi_{\boldsymbol{s}}(\boldsymbol{\xi})) = J_{e,\boldsymbol{p}} \prod_{i=1}^{d} w_{p_i}\,\mathcal{P}^{s_i}_{p_i q_i} \tag{8.64}$$

Using (8.64) in (8.63) and dividing both sides by $J_{e,\boldsymbol{p}}$ gives

$$\boldsymbol{u}_{e,\boldsymbol{p}} = \sum_{\boldsymbol{s}\in\mathbb{N}_1^d} \sum_{\boldsymbol{q}\in\mathbb{N}_N^d} \left(\prod_{i=1}^{d} \mathcal{P}^{s_i}_{p_i q_i} \right) \boldsymbol{u}_{e_{\boldsymbol{s}},\boldsymbol{q}} = \sum_{\boldsymbol{s}\in\mathbb{N}_1^d} \left(\prod_{i=1}^{d} \mathcal{P}^{s_i} \right) \boldsymbol{u}_{e_{\boldsymbol{s}}}$$

where the last equation follows using the product of operators notation (8.60).

8.5.2. Mortar element method (MEM)

8.5.2.1. Motivation and notation

When the mesh is adaptively refined, there will be elements with different refinement levels sharing a face; in this work, we assume that the refinement levels of those elements only differ by 2 (Figure 8.3a). Since the neighbouring elements do not have

a common face, the solution points on their faces do not coincide (Figure 8.4). We will use the Mortar Element Method (MEM) for computing the numerical flux at all the required points on such a face, while preserving accuracy and the conservation property (8.29). There are two steps to the method.

1. Prolong $\tilde{\boldsymbol{F}}^{\delta}\cdot\hat{\boldsymbol{n}}_{S,i}, \boldsymbol{U}_{S,i}, \boldsymbol{u}_{S,i}$ (8.26) from the neighbouring elements to a set of common solution points known as mortar solution points (Figure 8.4a).
2. Compute the numerical flux at the mortar solution points as in (8.26) and map it back to the interfaces (Figure 8.4b).

In Sections 8.5.2.2, 8.5.2.3, we will explain these two steps through the specific case of Figure 8.4 and we first introduce notations for the same.

Consider the multi-indices $\boldsymbol{s}\in\mathbb{N}_1^{d-1}=\{0,1\}^{d-1}$ and the interface in right (positive) $i=1$ direction of element Ω_e, denoted as Γ (Figure 8.4). We assume that the elements neighbouring Ω_e at the interface Γ are finer and thus we have non-conforming subinterfaces $\{\Gamma_{\boldsymbol{s}}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$ which, by continuity of the reference map, can be written as $\Gamma_{\boldsymbol{s}}=\Theta_e(\{1\}\times\prod_{i=1}^{d-1}I_{s_i})=\Theta_e(\{1\}\times\phi_{\boldsymbol{s}}^{-1}(I^{d-1}))$. Thus, in reference coordinates, $\phi_{\boldsymbol{s}}$ (8.56) is a bijection from $\Gamma_{\boldsymbol{s}}$ to Γ. The interface Γ can be parametrized as $\boldsymbol{y}=\boldsymbol{\gamma}(\boldsymbol{\eta})=\Theta_e(1,\boldsymbol{\eta})$ for $\boldsymbol{\eta}\in I^{d-1}$ and thus the reference variable of interface is denoted $\boldsymbol{\eta}=\boldsymbol{\gamma}^{-1}(\boldsymbol{y})$. The subinterfaces can also be written by using the same parametrization so that $\Gamma_{\boldsymbol{s}}=\{\gamma(\boldsymbol{\eta})\colon\boldsymbol{\eta}\in\prod_{i=1}^{d-1}I_{s_i}\}$. For the reference solution points on Γ being $\{\boldsymbol{\eta}_{\boldsymbol{s}}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$, the solution points in $\Gamma_{\boldsymbol{s}}$ are respectively given by $\{\phi_{\boldsymbol{s}}^{-1}(\boldsymbol{\eta}_{\boldsymbol{p}})\}_{\boldsymbol{p}\in N_1^{d-1}}$ and for $\{\ell_{\boldsymbol{p}}(\boldsymbol{\eta})\}_{\boldsymbol{p}\in\mathbb{N}_N^{d-1}}$ being Lagrange polynomials in Γ, the Lagrange polynomials in $\Gamma_{\boldsymbol{s}}$ are given by $\{\ell_{\boldsymbol{p}}(\phi_{\boldsymbol{s}}(\boldsymbol{\eta}))\}_{\boldsymbol{p}\in\mathbb{N}_1^{d-1}}$ respectively. Since the solution points between Γ and $\Gamma_{\boldsymbol{s}}$ do not coincide, they will be mapped to common solution points in the mortars $\Xi_{\boldsymbol{s}}$ and then back to $\Gamma,\Gamma_{\boldsymbol{s}}$ after computing the common numerical flux. The solution points in $\Xi_{\boldsymbol{s}}$ are actually given by $\{\phi_{\boldsymbol{s}}^{-1}(\boldsymbol{\eta}_{\boldsymbol{p}})\}_{\boldsymbol{p}\in N_1^{d-1}}$, i.e., they are the same as $\Gamma_{\boldsymbol{s}}$. The quantities with subscripts $\Xi_{\boldsymbol{s}}^{-}$, $\Xi_{\boldsymbol{s}}^{+}$ will denote trace values from larger, smaller elements respectively.

8.5.2.2. Prolongation to mortars

We will explain the prolongation procedure for a quantity $\boldsymbol{F}$ which could be the normal flux $\tilde{\boldsymbol{F}}^{\delta}\cdot\hat{\boldsymbol{n}}_{S,i}$, time average solution $\boldsymbol{U}_{S,i}$ or the solution $\boldsymbol{u}_{S,i}$. The first step of MEM of mapping of solution point values from solution points at element interfaces $\Gamma,\Gamma_{\boldsymbol{s}}$ to solution points at mortars $\Xi_{\boldsymbol{s}}^{-},\Xi_{\boldsymbol{s}}^{+}$ is known as prolongation. The prolongation of $\{\boldsymbol{F}_{\Gamma_{\boldsymbol{s}}}^{\delta}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$ from small elements $\Gamma_{\boldsymbol{s}}$ to mortar values $\{\boldsymbol{F}_{\Xi_{\boldsymbol{s}}^{+}}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$ is the identity map since both have the same solution points, and the prolongation of $\boldsymbol{F}_{\Gamma}^{\delta}$ from the large interface Γ to the $\{\boldsymbol{F}_{\Xi_{\boldsymbol{s}}^{-}}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$ is an interpolation to the mortar solution points. Accuracy is maintained by the interpolation as the mortar elements are finer. Below, we explain the matrix operations used to perform the interpolation.

The prolongation of $\{\boldsymbol{F}_{\Gamma_{\boldsymbol{s}}}^{\delta}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$ to the mortar values $\{\boldsymbol{F}_{\Xi_{\boldsymbol{s}}^{+}}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$ is the identity map. The $\{\boldsymbol{F}_{\Xi_{\boldsymbol{s}}^{-}}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$ in Lagrange basis are given by

$$\boldsymbol{F}_{\Xi_{\boldsymbol{s}}^{-}}(\boldsymbol{\eta})=\sum_{\boldsymbol{p}\in\mathbb{N}_N^{d-1}}\ell_{\boldsymbol{p}}(\phi_{\boldsymbol{s}}(\boldsymbol{\eta}))\boldsymbol{F}_{\Xi_{\boldsymbol{s}}^{-},\boldsymbol{p}},\qquad\boldsymbol{\eta}\in\prod_{i=1}^{d-1}I_{s_i}\tag{8.65}$$

The coefficients $\{\boldsymbol{F}^{-}_{\Xi_{\boldsymbol{s}},\boldsymbol{p}}\}_{\boldsymbol{p}\in\mathbb{N}_N^{d-1}}$ are computed by interpolation

$$\begin{aligned}\boldsymbol{F}_{\Xi_{\boldsymbol{s}}^-,\boldsymbol{p}} = \boldsymbol{F}_{\Gamma^-}(\phi_{\boldsymbol{s}}^{-1}(\boldsymbol{\eta}_{\boldsymbol{p}})) &= \sum_{\boldsymbol{q}\in\mathbb{N}_N^{d-1}} \ell_{\boldsymbol{q}}(\phi_{\boldsymbol{s}}^{-1}(\boldsymbol{\eta}_{\boldsymbol{p}}))\boldsymbol{F}_{\Gamma}^{\delta}(\boldsymbol{\eta}_{\boldsymbol{q}}) \\ &= \sum_{\boldsymbol{q}\in\mathbb{N}_N^{d-1}} \left(\prod_{i=1}^{d-1} \ell_{q_i}(\phi_{s_i}^{-1}(\boldsymbol{\eta}_{p_i}))\right)\boldsymbol{F}_{\Gamma}^{\delta}(\boldsymbol{\eta}_{\boldsymbol{q}}) \\ &= \sum_{\boldsymbol{q}\in\mathbb{N}_N^{d-1}} \left(\prod_{i=1}^{d-1} (\mathtt{V}_{s_i})_{p_i q_i}\right)\boldsymbol{F}_{\Gamma}^{\delta}(\boldsymbol{\eta}_{\boldsymbol{q}})\end{aligned} \tag{8.66}$$

where the interpolation operators $\{\mathtt{V}_{\Xi_s}\}_{s=0,1}$ were defined in (8.58). Using the product of operators notation (8.60), we can compactly write (8.66) as

$$\boldsymbol{F}_{\Xi_{\boldsymbol{s}}^-} = \left(\prod_{i=1}^{d-1} \mathtt{V}_{s_i}\right)\boldsymbol{F}_{\Gamma}^{\delta} \tag{8.67}$$

The same procedure is performed for obtaining $\boldsymbol{U}_{\Xi_{\boldsymbol{s}}^{\pm}}$, $\boldsymbol{u}_{\Xi_{\boldsymbol{s}}^{\pm}}$. The numerical fluxes $\{\boldsymbol{F}^{*}_{\Xi_{\boldsymbol{s}}}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$ are then computed as in (8.26).

8.5.2.3. Projection of numerical fluxes from mortars to faces

In this section, we use the notation $\boldsymbol{F}^{*} := (\tilde{\boldsymbol{F}}_e \cdot \hat{\boldsymbol{n}}_{S,i})^{*}$ to denote the numerical flux (8.26). In the second step of MEM, the numerical fluxes $\{\boldsymbol{F}^{*}_{\Xi_{\boldsymbol{s}}}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$ computed using values at $\{\Xi^{\pm}_{\boldsymbol{s}}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$ are mapped back to interfaces $\Gamma_{\boldsymbol{s}}, \Gamma$. Since the solution points on $\Gamma_{\boldsymbol{s}}$ are the same as those of $\Xi^{\pm}_{\boldsymbol{s}}$, the mapping from $\{\boldsymbol{F}^{*}_{\Xi_{\boldsymbol{s}}}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$ to $\{\boldsymbol{F}^{*}_{\Gamma_{\boldsymbol{s}}}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$ is the identity map. In order to maintain the conservation property, an L^2 projection is performed to map all the fluxes $\{\boldsymbol{F}^{*}_{\Xi_{\boldsymbol{s}}}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$ into one numerical flux $\boldsymbol{F}^{*}_{\Gamma}$ on the larger interface.

An L^2 projection of these fluxes to $\boldsymbol{F}^{*}_{\Gamma}$ on Γ is performed as

$$\sum_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}} \int_{\Gamma_{\boldsymbol{s}}} \boldsymbol{F}^{*}_{\Xi_{\boldsymbol{s}}}\,\ell_{\boldsymbol{p}} = \int_{\Gamma} \boldsymbol{F}^{*}_{\Gamma}\ell_{\boldsymbol{p}}, \qquad \forall \boldsymbol{p}\in\mathbb{N}_N^{d-1} \tag{8.68}$$

where integrals are computed with quadrature at solution points. As in (8.65), we write the mortar fluxes as

$$\begin{aligned}\boldsymbol{F}^{*}_{\Xi_{\boldsymbol{s}}}(\boldsymbol{\eta}) &= \sum_{\boldsymbol{q}\in\mathbb{N}_1^{d-1}} \ell_{\boldsymbol{q}}(\phi_{\boldsymbol{s}}(\boldsymbol{\eta}))\boldsymbol{F}^{*}_{\Xi_{\boldsymbol{s}},\boldsymbol{q}}, \qquad \boldsymbol{\eta}\in\Xi_{\boldsymbol{s}} \\ \boldsymbol{F}^{*}_{\Gamma}(\boldsymbol{\eta}) &= \sum_{\boldsymbol{q}\in\mathbb{N}_1^{d-1}} \ell_{\boldsymbol{q}}(\boldsymbol{\eta})\boldsymbol{F}^{*}_{\Gamma,\boldsymbol{q}}, \qquad \boldsymbol{\eta}\in\Gamma\end{aligned}$$

Thus, the integral identity (8.68) can be written as

$$\sum_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}} \sum_{\boldsymbol{q}\in\mathbb{N}_N^{d-1}} \int_{\Gamma_{\boldsymbol{s}}} \ell_{\boldsymbol{p}}(\boldsymbol{\eta})\,\ell_{\boldsymbol{q}}(\phi_{\boldsymbol{s}}(\boldsymbol{\eta}))\boldsymbol{F}^{*}_{\Xi_{\boldsymbol{s}},\boldsymbol{q}} = \sum_{\boldsymbol{q}\in\mathbb{N}_{\boldsymbol{p}}^{d-1}} \int_{\Gamma} \ell_{\boldsymbol{p}}(\boldsymbol{\eta})\,\ell_{\boldsymbol{q}}(\boldsymbol{\eta})\boldsymbol{F}^{*}_{\Gamma,\boldsymbol{q}}, \quad \forall \boldsymbol{p}\in\mathbb{N}_N^{d-1} \tag{8.69}$$

Using the identities (8.69), the equations (8.69) become

$$\sum_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}\sum_{\boldsymbol{q}\in\mathbb{N}_N^{d-1}}\left(\prod_{i=1}^{d-1} w_{p_i}\mathcal{P}_{p_iq_i}^{s_i}\right)\boldsymbol{F}^*_{\Xi_{\boldsymbol{s}},\boldsymbol{q}}J^S_{e,\boldsymbol{p}} = w_{\boldsymbol{p}}\,\boldsymbol{F}^*_{\Gamma,\boldsymbol{p}}\,J^S_{e,\boldsymbol{p}}$$

where $J^S_{e,\boldsymbol{p}}$ is the surface Jacobian, given by $||(J\,\boldsymbol{a}^1)_{e,\boldsymbol{p}}||$ in this case ((6.29) of [103]). Then, dividing both sides by $J^S_{e,\boldsymbol{p}}\,w_{\boldsymbol{p}}$ gives

$$\boldsymbol{F}^*_{\Gamma,\boldsymbol{p}} = \sum_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}\sum_{\boldsymbol{q}\in\mathbb{N}_N^{d-1}}\left(\prod_{i=1}^{d-1}\mathcal{P}_{p_iq_i}^{s_i}\right)\boldsymbol{F}^*_{\Xi_{\boldsymbol{s}},\boldsymbol{q}} = \sum_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}\left(\prod_{i=1}^{d-1}\mathcal{P}^{s_i}\right)\boldsymbol{F}^*_{\Xi_{\boldsymbol{s}}} \tag{8.70}$$

where the last identity is obtained by the product of operators notation (8.60). Note that the identity (8.68) implies

$$\sum_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}\fint_{\Gamma_{\boldsymbol{s}}}\boldsymbol{F}^*_{\Xi_{\boldsymbol{s}}}\,v = \fint_{\Gamma}\boldsymbol{F}^*_{\Gamma}\,v, \qquad v\in\mathbb{P}_N$$

Then, taking $v=1$ shows that the total fluxes over an interface Γ are the same as over $\{\Gamma_{\boldsymbol{s}}\}_{\boldsymbol{s}\in\mathbb{N}_1^{d-1}}$ and thus the conservation property (8.29) of LWFR is maintained by the LWFR scheme.

Remark 8.7. (FREESTREAM AND ADMISSIBILITY PRESERVATION UNDER AMR) Under the adaptively refined meshes, free stream preservation and provable admissibility preservation are respectively ensured.

1. When refining/coarsening, there are two ways to compute the metric terms - interpolate/project the metric terms directly or interpolate/project the reference map Θ at solution points and use the newly obtained reference map to recompute the metric terms. The latter, which is the approach taken in this work, can lead to violation of free stream preservation as we can have $(\mathcal{I}_N)_{e_L}(J\,\boldsymbol{a}^i) \neq (\mathcal{I}_N)_{e_R}(J\,\boldsymbol{a}^i)$ where Ω_{e_L} and Ω_{e_S} are two neighbouring large and small elements respectively. Thus, the interface terms may not vanish in the update equation (8.28) with constant $\boldsymbol{u}^n$ leading to a violation of free stream preservation. This issue only occurs in 3-D and is thus beyond the scope of this work, but some remedies are to interpolate/project the metric terms when refining/coarsening or to use the reference map $\Theta\in\mathbb{P}_{N/2}$, as explained in [104]. Another solution has been studied in [109] where a common finite element space with mixed degree $N-1$ and N is used with continuity at the non-conformal interfaces. Since this work only deals with problems in 2-D, we always have $(\mathcal{I}_N)_{e_L}(J\,\boldsymbol{a}^i) = (\mathcal{I}_N)_{e_R}(J\,\boldsymbol{a}^i)$ ensuring that the interface terms in (8.28) vanish when $\boldsymbol{u}=\boldsymbol{c}$. Further, since the metric terms are recomputed in this work, the volume terms will vanish by the same arguments as in Section 8.3.4. Thus, free stream preservation is maintained even with the non-conformal, adaptively refined meshes.

2. The flux limiting explained in Section 8.4.3 ensures admissibility in means (Definition 5.2) and then uses the scaling limiter of [205] to enforce admissibility of solution polynomial at all solution points to obtain an admissibility preserving scheme (Definition 5.1). However, the procedure doesn't ensure that the polynomial is admissible at points which are not the solution points. Adaptive mesh refinement introduces such points into the numerical method and can thus cause a failure of admissibility preservation in the following situations: (a) mortar solution values $\{\boldsymbol{u}_{\Xi_s}^-\}$ obtained by interpolation as in (8.65) are not admissible, (b) mean values $\{\bar{\boldsymbol{u}}_{e_s}\}$ of the solution values $\{\boldsymbol{u}_{e_s}\}$ obtained by interpolating from the larger element as in (8.5.1.1) are not admissible. Since the scaling limiter [205] can be used to enforce admissibility of solution at any desired points, the remedy to both issues is further scaling; we simply perform scaling of solution point values $\{\boldsymbol{u}_{\Xi_s}^-\}, \{\boldsymbol{u}_{e_s}\}$ with the admissible mean value $\bar{\boldsymbol{u}}_e$. This will ensure that the mortar solution point values and the mean values $\{\boldsymbol{u}_{e_s}\}$ are admissible.

8.5.3. AMR indicators

The process of adaptively refining and coarsening the mesh requires a local solution smoothness indicator. In this work, two smoothness indicators have been used for adaptive mesh refinement. The first is the indicator of [90], explained in Section 8.4.2. The second is Löhner's smoothness indicator [120] which uses the central finite difference formula for second derivative, which is given by

$$\begin{aligned}
\alpha_e &= \max_{\boldsymbol{p}\in\mathbb{N}_N^d} \max_{1\le i\le d} \frac{|q(\boldsymbol{u}_{\boldsymbol{p}_{i+}}) - 2\,q(\boldsymbol{u}_{\boldsymbol{p}}) + q(\boldsymbol{u}_{\boldsymbol{p}_{i-}})|}{\text{Normalizer}(i,\boldsymbol{p})}, \\
\text{Normalizer}(i,\boldsymbol{p}) &= \begin{pmatrix} |q(\boldsymbol{u}_{\boldsymbol{p}_{i+}}) - q(\boldsymbol{u}_{\boldsymbol{p}})| + |q(\boldsymbol{u}_{\boldsymbol{p}}) - q(\boldsymbol{u}_{\boldsymbol{p}_{i-}})| \\ + f_{\text{wave}}\left(|q(\boldsymbol{u}_{\boldsymbol{p}_{i+}})| + 2|q(\boldsymbol{u}_{\boldsymbol{p}})| + |q(\boldsymbol{u}_{\boldsymbol{p}_{i-}})|\right) \end{pmatrix} \\
(\boldsymbol{p}_{i\pm})_m &= \begin{cases} p_m, & m \ne i \\ p_{i\pm 1}, & m = i \end{cases}
\end{aligned} \tag{8.71}$$

where $\{\boldsymbol{u}_{\boldsymbol{p}}\}_{\boldsymbol{p}\in\mathbb{N}_N^d}$ are the degrees of freedom in element Ω_e and q is a derived quantity like the product of density and pressure used in Section 8.4.2. The value $f_{\text{wave}} = 0.2$ has been chosen in all the tests [120].

Once a smoothness indicator is chosen, the three level controller implemented in `Trixi.jl` [140] is used to determine the local refinement level. The mesh begins with an initial refinement level and the effective refinement level is prescribed by how much further refinement has been done to the initial mesh, The mesh is created with two thresholds `med_threshold` and `max_threshold` and three refinement levels `base_level`, `med_level` and `max_level`. Then, we have

$$\texttt{level}_e = \begin{cases} \texttt{base\_level}, & \alpha_e \le \texttt{med\_threshold} \\ \texttt{med\_level}, & \texttt{med\_threshold} \le \alpha_e \le \texttt{max\_threshold} \\ \texttt{max\_level}, & \texttt{max\_threshold} \le \alpha_e \end{cases}$$

Beyond these refinement levels, further refinement is performed to make sure that two neighbouring elements only differ by a refinement level of 1.

8.6. TIME STEPPING

This section introduces an embedded error approximation method to compute the time step size Δt for the single stage Lax-Wendroff Flux Reconstruction method. First, recall that a standard way to compute the time step size Δt^n is to follow Chapter 4 and use

$$\Delta t_n = C_s \min_{e,\boldsymbol{p}} \frac{|J_{e,\boldsymbol{p}}|}{\sigma(\boldsymbol{u}^n_{e,\boldsymbol{p}})} \mathrm{CFL}(N) \tag{8.72}$$

where the minimum is taken over all elements $\{\Omega_e\}_e$, J_e is the Jacobian of the change of variable map, $\sigma(\boldsymbol{u}^n_e)$ is the largest eigenvalue of the flux jacobian at state $\boldsymbol{u}^n_e$, approximating the local wave speed, $\mathrm{CFL}(N)$ is the optimal CFL number dependent on solution polynomial degree N and $C_s \leq 1$ is a safety factor. In Section 4.4, a Fourier stability analysis of the LWFR scheme was performed on Cartesian grids, and the optimal CFL numbers were obtained for each degree N (Table 4.1) which guaranteed the stability of the scheme. However, the Fourier stability analysis does not apply to curvilinear grids and formula (8.72) need not guarantee L^2 stability with CFL numbers from Table 4.1. Thus, formula (8.72) may require the CFL number to be fine-tuned for each problem. Along with the L^2 stability, the time step has to be chosen so that the scheme does not give inadmissible solutions. An error-based time stepping method inherently minimizes the parameter tuning process in time step computation. The parameters in an error-based time stepping scheme that a user has to specify are the absolute and relative error tolerances τ_a, τ_r, and they only affect the time step size logarithmically. In particular, because of the weak dependence, tolerances $\tau_a = \tau_r = 10^{-6}$ worked reasonably for all tests with shocks; although, it was possible to enhance performance by choosing larger tolerances for some problems. Secondly, if inadmissibility is detected during any step in the scheme or if errors are too large, the time step is redone with a reduced time step size provided by the error estimate. The scheme also has the capability of increasing and decreasing the time step size.

We begin by reviewing the error-based time stepping scheme for the Runge-Kutta ODE solvers from [140, 142] in Section 8.6.1 and explain our extension of the same to LWFR in Section 8.6.2.

8.6.1. Error estimation for Runge-Kutta schemes

Consider an explicit Runge-Kutta method used for solving ordinary differential equations by evolving the numerical solution from time level n to $n+1$. For error estimation, the method is constructed to have an embedded lower order update $\hat{\boldsymbol{u}}^{n+1}$, as described

in equation (3) of [140]. The difference in the two updates, $\boldsymbol{u}^{n+1} - \hat{\boldsymbol{u}}^{n+1}$, gives an indication of the time integration error, which is used to build a Proportional Integral Derivative (PID) controller to compute the new time step size,

$$\widetilde{\Delta} t_{n+1} = \kappa(\varepsilon_{n+1}^{\beta_1/k}\, \varepsilon_n^{\beta_2/k}\, \varepsilon_{n-1}^{\beta_3/k})\, \Delta t_n \tag{8.73}$$

where for q being the order of main method, $\hat{q}$ being the order of embedded method, we have

$$k = \min(q, \hat{q}) + 1$$

and β_i are called control parameters which are optimized for the particular Runge-Kutta scheme [140]. With M being the number of degrees of freedom in $\boldsymbol{u}$, we pick absolute and relative tolerances τ_a, τ_r and then error approximation is made as

$$\varepsilon_{n+1} = \frac{1}{w_{n+1}}, \qquad w_{n+1} = \left(\frac{1}{M} \sum_{i=1}^{M} \left(\frac{\boldsymbol{u}_i^{n+1} - \hat{\boldsymbol{u}}_i^{n+1}}{\tau_a + \tau_r \max\{|\boldsymbol{u}_i^{n+1}|, |\breve{\boldsymbol{u}}|\}} \right)^2 \right)^{\frac{1}{2}} \tag{8.74}$$

where the sum is over all degrees of freedom, including solution points and conservative variables. The tolerances are to be chosen by the user but their influence on the scheme is logarithmic, unlike the CFL based scheme (8.72).

The limiting function $\kappa(x) = 1 + \tan^{-1}(x-1)$ is used to prevent sudden increases in time step sizes. For normalization, `PETSc` uses $\breve{\boldsymbol{u}} = \hat{\boldsymbol{u}}^{n+1}$ while `OrdinaryDiffEq.jl` uses $\breve{\boldsymbol{u}} = \boldsymbol{u}^n$. Following [140], if the time step factor $\widetilde{\Delta} t^{n+1} / \Delta t^n \geq 0.9^2$, the new time step is accepted and used in the next level as $\Delta t^{n+1} = \widetilde{\Delta} t^{n+1}$. If not, or if admissibility is violated, evolution is redone with time step size $\Delta t^n = \widetilde{\Delta} t^{n+1}$ computed from (8.73).

8.6.2. Error based time stepping for Lax-Wendroff flux reconstruction

Consider the LWFR scheme (8.28) with polynomial degree N and formal order of accuracy $N+1$

$$\boldsymbol{u}_{e,\boldsymbol{p}}^{n+1} = \boldsymbol{u}_{e,\boldsymbol{p}}^{n} - \frac{\Delta t}{J_{e,\boldsymbol{p}}} \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{F}}_e^{\delta}(\boldsymbol{\xi}_{\boldsymbol{p}}) - \boldsymbol{\mathcal{C}}_{e,\boldsymbol{p}}$$

where $\boldsymbol{\mathcal{C}}_{e,\boldsymbol{p}}$ contains contributions at element interfaces. In order to construct a lower order embedded scheme without requiring additional inter-element communication, consider an evolution where the interface correction terms $\boldsymbol{\mathcal{C}}_{e,\boldsymbol{p}}$ are not used, i.e., consider the element local update

$$\boldsymbol{u}_{\mathrm{loc},e,\boldsymbol{p}}^{n+1} = \boldsymbol{u}_{e,\boldsymbol{p}}^{n} - \frac{\Delta t}{J_{e,\boldsymbol{p}}} \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{F}}_e^{\delta}(\boldsymbol{\xi}_{\boldsymbol{p}}) \tag{8.75}$$

Truncating the locally computed time averaged flux $\tilde{\boldsymbol{F}}_e^\delta$ (8.24) at one order lower

$$\widehat{\boldsymbol{F}}_e^\delta = \sum_{k=0}^{N-1} \frac{\Delta t^k}{(k+1)!}\, \partial_t^k\, \tilde{\boldsymbol{f}}_e^\delta \tag{8.76}$$

we can consider another update

$$\widehat{\boldsymbol{u}_{\mathrm{loc},e,\boldsymbol{p}}^{n+1}} = \boldsymbol{u}_{e,\boldsymbol{p}}^n - \frac{\Delta t}{J_{e,\boldsymbol{p}}} \nabla_{\boldsymbol{\xi}} \cdot \widehat{\boldsymbol{F}}_e^\delta(\boldsymbol{\xi}_{\boldsymbol{p}}) \tag{8.77}$$

which is also locally computed but is one order of accuracy lower. We thus use $\boldsymbol{u}_e^{n+1} = \boldsymbol{u}_{\mathrm{loc},e}^{n+1}$ and $\hat{\boldsymbol{u}}_e^{n+1} = \widehat{\boldsymbol{u}_{\mathrm{loc},e}^{n+1}}$ in the formula (8.74) along with $\breve{\boldsymbol{u}} = \hat{\boldsymbol{u}}^{n+1}$; then we use the same procedure of redoing the time step sizes as in Section 8.6.1. That is, after using the error estimate (8.74) to compute $\widetilde{\Delta t}_{n+1}$ (8.73) we redo the time step if $\widetilde{\Delta t}^{n+1}/\Delta t^n \geq 0.9^2$ or if admissibility is violated; otherwise we set Δt^{n+1} to be used at the next time level. The complete process is also detailed in Algorithm 8.4. In this work, we have used the control parameters $\beta_1 = 0.6, \beta_2 = -0.2, \beta_3 = 0.0$ for all numerical results which are the same as those used in [140] for BS3(2)3$_{\mathrm{F}}$, the third-order, four-stage RK method of [32]. We tried the other control parameters from [140] but found the present choice to be either superior or only slightly different in performance, measured by the number of iterations taken to reach the final time.

Algorithm 8.2

Overview of LWFR element residual of order $N+1$ & error approximation using $\boldsymbol{u}_{\mathrm{loc}}^{n+1}, \widehat{\boldsymbol{u}_{\mathrm{loc}}^{n+1}}$

for e in `eachelement(mesh)` **do**
- Compute $\{\partial_t^k \tilde{\boldsymbol{f}}_e^\delta\}_{k=0}^{N-1}$ using approximate LW procedure (8.32) to obtain $\widehat{\boldsymbol{F}}_e^\delta$ (8.76)
- Compute $\widehat{\boldsymbol{u}_{\mathrm{loc},e}^{n+1}}$ using $\widehat{\boldsymbol{F}}_e^\delta$ (8.77)
- Compute $\partial_t^{N+1} \tilde{\boldsymbol{f}}_e^\delta$ using approximate LW procedure (8.32) to obtain $\tilde{\boldsymbol{F}}_e^\delta$ (8.24)
- Compute $\boldsymbol{u}_{\mathrm{loc},e}^{n+1}$ using $\tilde{\boldsymbol{F}}_e^\delta$ as in (8.75)
- `temporal_error[e]` $= \sum_i \left(\dfrac{\boldsymbol{u}_{\mathrm{loc},e,i}^{n+1} - \widehat{\boldsymbol{u}_{\mathrm{loc},e,i}^{n+1}}}{\tau_a + \tau_r \max\left\{ |\boldsymbol{u}_{\mathrm{loc},e,i}^{n+1}|, \left|\widehat{\boldsymbol{u}_{\mathrm{loc},e,i}^{n+1}}\right| \right\}} \right)^2$, sum is over dofs in e
- Compute and add local contribution of $\tilde{\boldsymbol{F}}_e^\delta$ to the residual (8.28)

end

Algorithm 8.3

High level overview of LWFR residual (Within time integration)

Compute $\{\alpha_e\}$ (Section 8.4.2)
Assemble cell residual (Algorithm 8.2)
for Γ in `eachinterface(mesh)` **do**
- Compute $\boldsymbol{F}_\Gamma^{\mathrm{LW}}$, $\boldsymbol{f}_\Gamma$ and blend them into $\boldsymbol{F}_\Gamma$ (Algorithm 8.1)

end
for e in `eachelement(mesh)` **do**
- Add contribution of numerical fluxes to residual of element e (Remark 5.3b)

done
Update solution
Apply positivity limiter

Algorithm 8.4

Lax-Wendroff Flux Reconstruction at a high level to explain error based time stepping

Initialize $t \leftarrow 0$, time step number $n \leftarrow 0$, and initial state $\boldsymbol{u}^0$
Initialize PID controller with $\varepsilon_0 \leftarrow 1, \varepsilon_{-1} \leftarrow 1$
Initialize $\Delta t_0 = \widetilde{\Delta t}$ with a user supplied value
Initialize `accept_step` $\leftarrow$ `false`
while $t < T$ **do**
  if `accept_step` **then**
    `accept_step` $\leftarrow$ `false`
    $t \leftarrow \tilde{t}$
    $\Delta t_{n+1} \leftarrow \widetilde{\Delta t}$
    $n \leftarrow n+1$
  else
    $\Delta t_n \leftarrow \widetilde{\Delta t}$
  end
  if $t + \Delta t_n >$ `final_time` **then**
    $\Delta t_n \leftarrow$ `final_time` $- t$
  end if
  $\boldsymbol{u}^n \xrightarrow{\Delta t_n} \boldsymbol{u}^{n+1}$ (Algorithm 8.3, 8.2) computing `temporal_error`, checking admissibility
  $w_{n+1} \leftarrow \left(\frac{1}{M}\sum_e$ `temporal_error[e]` $\right)^{\frac{1}{2}}$ ▷ M is the total number of dofs
  $w_{n+1} \leftarrow \max\{w_{n+1}, 10^{-10}\}$ ▷ To avoid division by zero
  $\varepsilon_{n+1} \leftarrow \frac{1}{w_{n+1}}$
  `dt_factor` $\leftarrow \kappa\left(\varepsilon_{n+1}^{\beta_1/k}\, \varepsilon_n^{\beta_2/k}\, \varepsilon_{n-1}^{\beta_3/k}\right)$ ▷ $\kappa(x) = 1 + \tan^{-1}(x-1)$
  $\widetilde{\Delta t} \leftarrow$ `dt_factor` $\cdot \Delta t_n$
  if `dt_factor` $\geq$ `accept_safety && no inadmissibility` **then**
    `accept_step` $\leftarrow$ `true`
  else
    `accept_step` $\leftarrow$ `false`
  end
  if `accept_step` **then**
    $\tilde{t} \leftarrow t + \Delta t_n$
      if $\tilde{t} \approx$ `final_time` **then**
        $\tilde{t} \leftarrow$ `final_time`
      end if
    Apply callbacks ▷ Analyze and postprocess solution, AMR
    Positivity correction for AMR (Remark 8.7)
  end if
end while

8.7. Numerical results

The numerical experiments are performed on 2-D Euler's equations (2.13). Unless

specified otherwise, the adiabatic constant γ will be taken as 1.4 in the numerical tests, which is the typical value for air. A comparison will be made between error and CFL based time stepping schemes where the CFL based time stepping schemes use the following formula for the time step (see (2.5) of [142], but also [96, 140])

$$\Delta t_n = \frac{2}{N+1} C_{\text{CFL}} \min_{e,\boldsymbol{p}} \left(\frac{1}{|J_{e,\boldsymbol{p}}|} \sum_{i=1}^{d} \tilde{\lambda}^i_{e,\boldsymbol{p}} \right), \qquad C_{\text{CFL}} \leq 1 \tag{8.78}$$

where $\{\tilde{\lambda}^i_{e,\boldsymbol{p}}\}_{i=1}^d$ are wave speed estimates computed by the transformation

$$\tilde{\lambda}^i_{e,\boldsymbol{p}} = \sum_{n=1}^{d} (J a^i_n)_{e,\boldsymbol{p}} \lambda^i_{e,\boldsymbol{p}}$$

for $\{J\boldsymbol{a}^i\}_{i=1}^d$ being the contravariant vectors (8.2) and $\lambda^i_{e,\boldsymbol{p}}$ the absolute maximum eigenvalue of $\boldsymbol{f}_i'(\boldsymbol{u}_{e,\boldsymbol{p}})$. For Euler's equations with velocity vector $\boldsymbol{v} = \{v_i\}$ and sound speed c, $\lambda^i = |v_i| + c$. The C_{CFL} in (8.78) may need to be fine-tuned depending on the problem. Other than the convergence test (Section 8.7.2.2), the results shown below have been generated with error-based time stepping (Section 8.6.2). The scheme is implemented in a Julia package `TrixiLW.jl` written using `Trixi.jl` [141, 158, 157] as a library. `Trixi.jl` is a high order PDE solver package in `Julia` [29] and uses the Runge-Kutta Discontinuous Galerkin method; `TrixiLW.jl` uses `Julia`'s multiple dispatch to borrow features like curved meshes support and postprocessing from `Trixi.jl`. `TrixiLW.jl` is not a fork of `Trixi.jl` but only uses it through `Julia`'s package manager without modifying its internal code. The setup files for the numerical experiments in this work are available at [10]. The animations of the results presented in this chapter can be viewed at

www.youtube.com/playlist?list=PLHg8S7nd3rfvI1Uzc3FDaTFtQo5VBUZER

8.7.1. Results on Cartesian grids

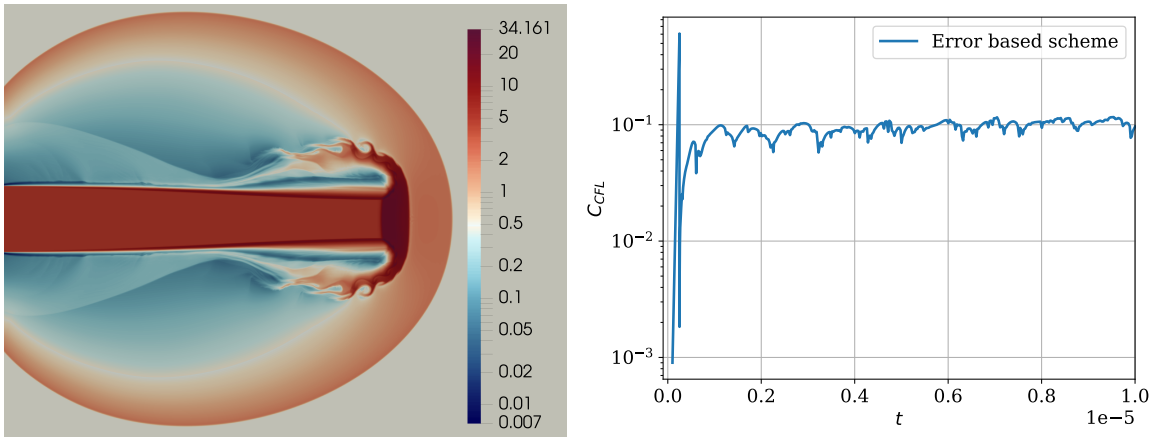
8.7.1.1. Mach 2000 astrophysical jet

The test is as described in Section 5.9.5. The simulation is performed on a uniform 512^2 element mesh. This test requires admissibility preservation to be enforced to avoid solutions with negative pressure. This is a cold-start problem as the solution is constant with zero velocity in the domain at time $t = 0$. However, there is a high speed inflow at the boundary, which the standard wave speed estimate for time step approximation (8.79) does not account for. Thus, in order to use the CFL based time stepping, lower values of C_{CFL} (8.78) have to be used in the first few iterations of the simulations. Once the high speed flow has entered the domain, this value needs to be raised since otherwise, the simulation will use much smaller time steps than the linear stability limit permits. In Section 5.9.5, this was handled by including the inflow wave speed for computation of time step. Error based time stepping schemes automate this process by their adaptivity and ability to redo the time steps. The simulation is run till $t = 10^{-2}$ and the log scaled density plot for degree $N = 4$ solution obtained on the

uniform mesh is shown in Figure 8.5a. For an error-based time stepping scheme, we define the effective C_{CFL} as

$$C_{\text{CFL}} := \Delta t_n \left[\frac{2}{N+1} \min_{e,\boldsymbol{p}} \left(\frac{1}{|J_{e,\boldsymbol{p}}|} \sum_{i=1}^{d} \tilde{\lambda}^i_{e,\boldsymbol{p}} \right) \right]^{-1} \tag{8.79}$$

which is a reverse computation so that its usage in (8.72) will get Δt_n chosen in the error-based time stepping scheme (Algorithm 8.4). In Figure 8.5b, time t versus effective C_{CFL} (8.79) is plotted up to $t = 10^{-5}$ to demonstrate that the scheme automatically uses a smaller C_{CFL} of $\sim 10^{-3}$ at the beginning which later increases and stabilizes at $\sim 10^{-1}$. Thus, the error based time stepping is automatically doing what would have to be manually implemented for a CFL based time stepping scheme which would be problem-dependent and require smart user intervention.



(a) (b)

Figure 8.5. Mach 2000 astrophysical jet (a) Density plot (b) Effective C_{CFL}

8.7.1.2. Kelvin-Helmholtz instability

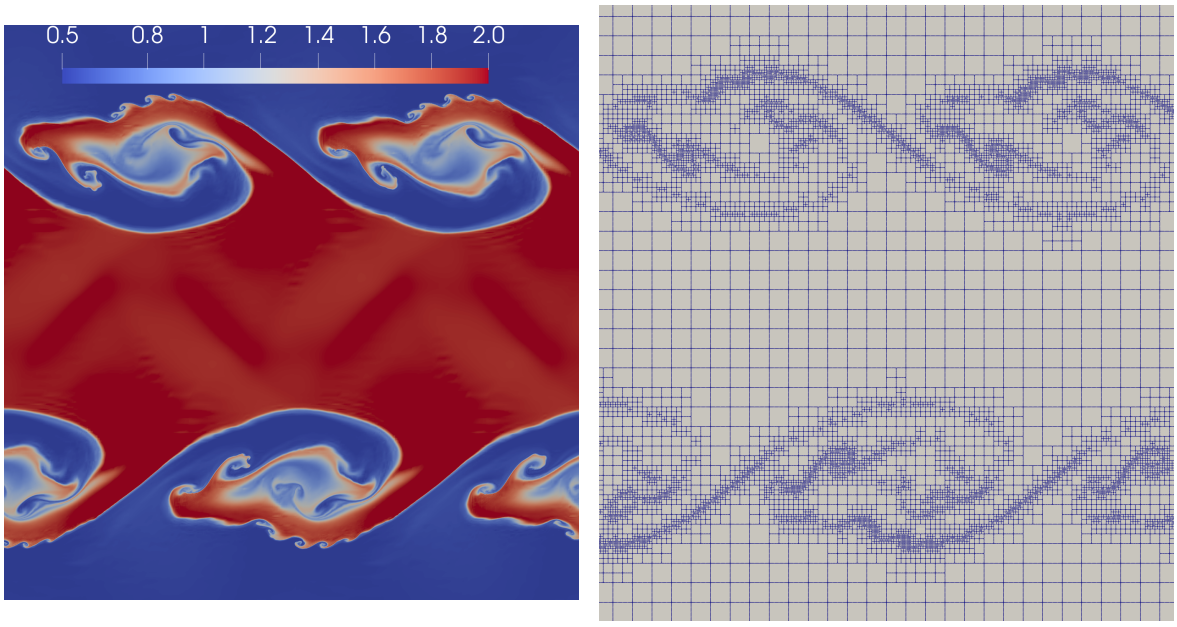
This is a variant of the Kelvin-Helmholtz instability like in Section 5.9.4. The initial condition is given by [151]

$$(\rho, u, v, p) = \left(\frac{1}{2} + \frac{3}{4} B, \frac{1}{2}(B-1), \frac{1}{10} \sin(2\pi x), 1 \right)$$

with $B = \tanh(15\, y + 7.5) - \tanh(15\, y - 7.5)$ in domain $\Omega = [-1, 1]^2$ with periodic boundary conditions. The initial condition has a Mach number $M \leq 0.6$ which makes compressibility effects relevant but does not cause shocks to develop. Thus, a very mild shock capturing scheme is used by setting $\alpha_e = \min\{\alpha_e, \alpha_{\max}\}$ (Section 8.4.2) where $\alpha_{\max} = 0.002$. The same smoothness indicator of Section 8.4.2 is used for AMR indicator with parameters from Section 8.5.3 chosen to be

$$(\texttt{base\_level}, \texttt{med\_level}, \texttt{max\_level}) = (4, 0, 8)$$
$$(\texttt{med\_threshold}, \texttt{max\_threshold}) = (0.0003, 0.003)$$

where `base_level` $= 0$ refers to a 2×2 mesh. This test case, along with indicators' configuration was taken from the examples of `Trixi.jl` [140]. The simulation is run till $t = 3$ using polynomial degree $N = 4$. There is a shear layer at $y = \pm 0.5$ which rolls up and develops smaller scale structures as time progresses. The results are shown in Figure 8.6 and it can be seen that the AMR indicator is able to track the small scale structures. The simulation starts with a mesh of 1024 elements which steadily increases to 13957 at the final time; the mesh is adaptively refined or coarsened at every time step. The solution has non-trivial variations in small regions around the rolling structures which an adaptive mesh algorithm can capture efficiently, while a uniform mesh with similar resolution would require 262144 elements.



(a) (b)

Figure 8.6. Kelvin-Helmholtz instability at $t = 3$ using polynomial degree $N = 4$ (a) density plots, (b) adaptively refined mesh

8.7.1.3. Double mach reflection

The description and significance of this test have been given in Section 4.11.2. The setup of Löhner's smoothness indicator (8.71) is taken from an example of `Trixi.jl` [140]

$$(\texttt{base\_level}, \texttt{med\_level}, \texttt{max\_level}) = (0, 3, 6)$$
$$(\texttt{med\_threshold}, \texttt{max\_threshold}) = (0.05, 0.1)$$

where `base_level` $= 0$ corresponds to a 16×5 mesh. The density solution obtained using polynomial degree $N = 4$ is shown in Figure 8.7 where it is seen that AMR is tracing the shocks and small scale shearing well. The initial mesh consists of 80 elements and is refined in first iteration in the vicinity of the shock to get 2411 elements. In later iterations, the mesh is refined and coarsened in each iteration, and the number of elements keeps increasing up to 7793 elements at the final time $t = 0.2$. In order to capture the same effective refinement, a uniform mesh will require 327680 elements.

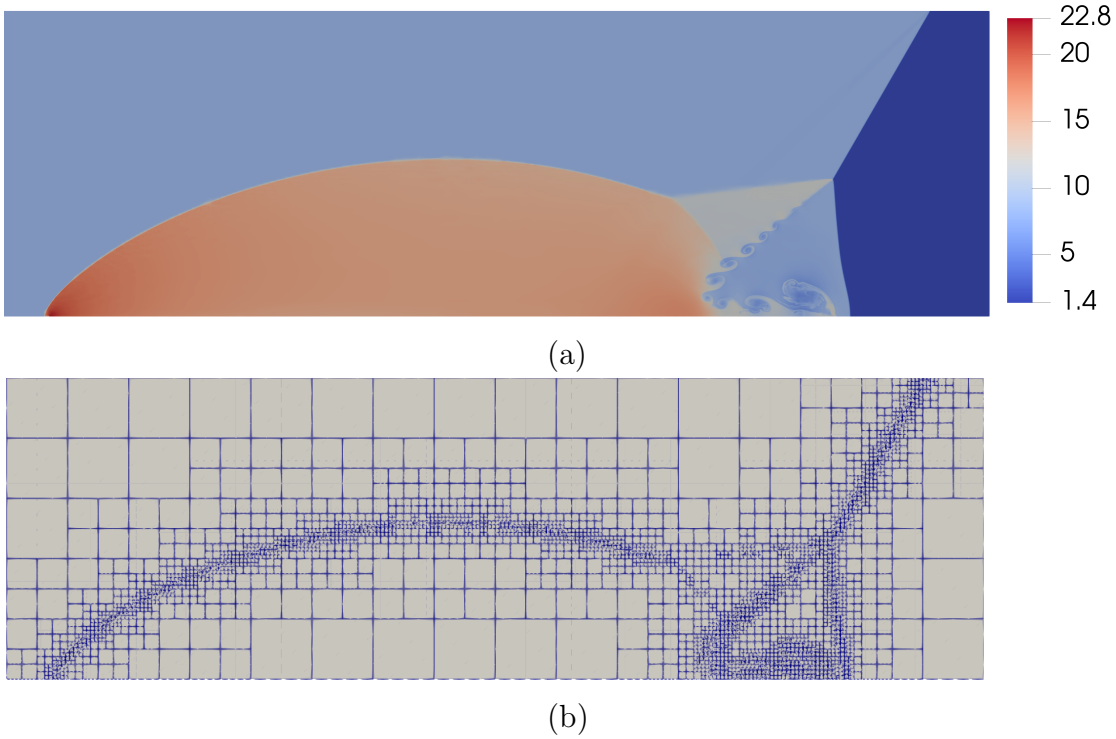


Figure 8.7. Double Mach reflection with solution polynomial degree $N = 4$ at $t = 0.2$ (a) Density plot, (b) Adaptively refined mesh at final time

8.7.1.4. Forward facing step

The description and significance of this test have been given in Section 5.9.8. Here, we repeat the setup to make this chapter self-contained. The step is simulated in the domain $\Omega = ([0,3] \times [0,1]) \setminus ([0.6,3] \times [0,0.2])$ and the initial conditions are taken to be

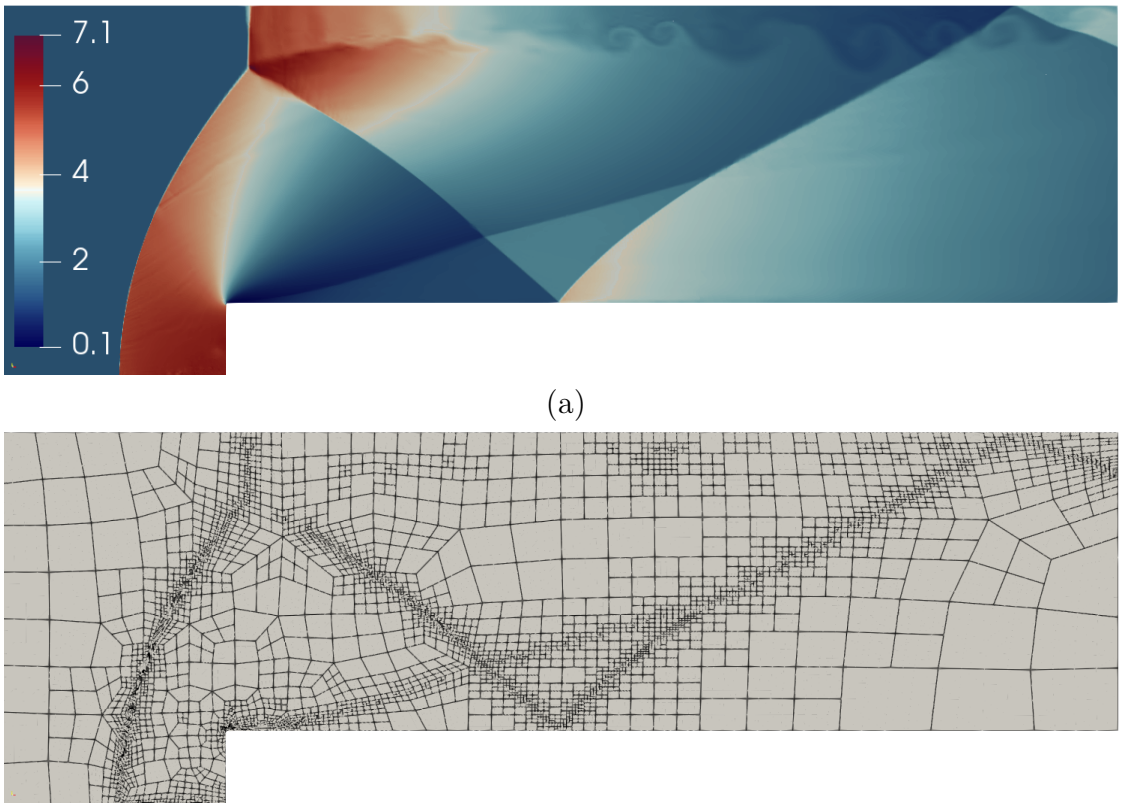
$$(\rho, u, v, p) = (1.4, 3, 0, 1) \qquad \text{in } \Omega$$

The left boundary condition is taken as an inflow and the right one is an outflow, while the rest are solid walls. The corner $(0.6, 0.2)$ of the step is the center of a rarefaction fan and can lead to large errors and the formation of a spurious boundary layer, as shown in Figure 7a-7d of [197] and also in the results of Section 5.9.8. These errors can be reduced by refining the mesh near the corner, which is automated here with the AMR algorithm.

The setup of Löhner's smoothness indicator (8.71) is taken from an example of `Trixi.jl` [140]

$$(\texttt{base\_level}, \texttt{med\_level}, \texttt{max\_level}) = (0, 2, 5)$$
$$(\texttt{med\_threshold}, \texttt{max\_threshold}) = (0.05, 0.1)$$

The density at $t = 3$ obtained using polynomial degree $N = 4$ and Löhner's smoothness indicator (8.71) is plotted in Figure 8.8. The shocks have been well-traced and resolved by AMR and the spurious boundary layer and Mach stem do not appear. The simulation starts with a mesh of 198 elements and the number peaks at 6700 elements during the simulation then and decreases to 6099 at the final time $t = 3$. The mesh is adaptively refined or coarsened once every 100 time steps. In order to capture the same effective refinement, a uniform mesh will require 202752 elements.



(a)

(b)

Figure 8.8. Mach 3 flow over forward facing step at time $t = 3$ using solution polynomial degree $N = 4$ with Löhner's indicator for mesh refinement. (a) Density plot (b) Adaptively refined mesh

8.7.2. Results on curved grids

8.7.2.1. Free stream preservation

In this section, free stream preservation is tested for meshes with curved elements. Since we use a reference map of degree N in (8.4), free stream will be preserved following the discussion in Section 8.3.4. We numerically verify the same for the meshes taken from `Trixi.jl` which are shown in Figure 8.9. The mesh in Figure 8.9a consists of curved boundaries and only the elements adjacent to the boundary are curved, while the one in Figure 8.9b is a non-conforming mesh with curved elements everywhere, and is used to verify that free stream preservation holds with adaptively refined meshes. The mesh in Figure 8.9b is a 2-D reduction of the one used in Figure 3 of [149] and is defined by the global map $(\xi, \eta) \mapsto (x, y)$ from $[0, 3]^2 \to \Omega$ described as

$$x = \xi + \frac{3}{8}\cos\left(\frac{\pi}{2}\frac{2\,\xi - 3}{3}\right)\cos\left(2\,\pi\,\frac{2\,y - 3}{3}\right)$$
$$y = \eta + \frac{3}{8}\cos\left(\frac{3\,\pi}{2}\frac{2\,\xi - 3}{3}\right)\cos\left(\frac{\pi}{2}\frac{2\,\eta - 3}{3}\right)$$

The free stream preservation is verified on these meshes by solving the Euler's equation with constant initial data

$$(\rho, u, v, p) = (1, 0.1, -0.2, 10)$$

and Dirichlet boundary conditions. Figure 8.9 shows the density at time $t = 10$ which is constant throughout the domain.

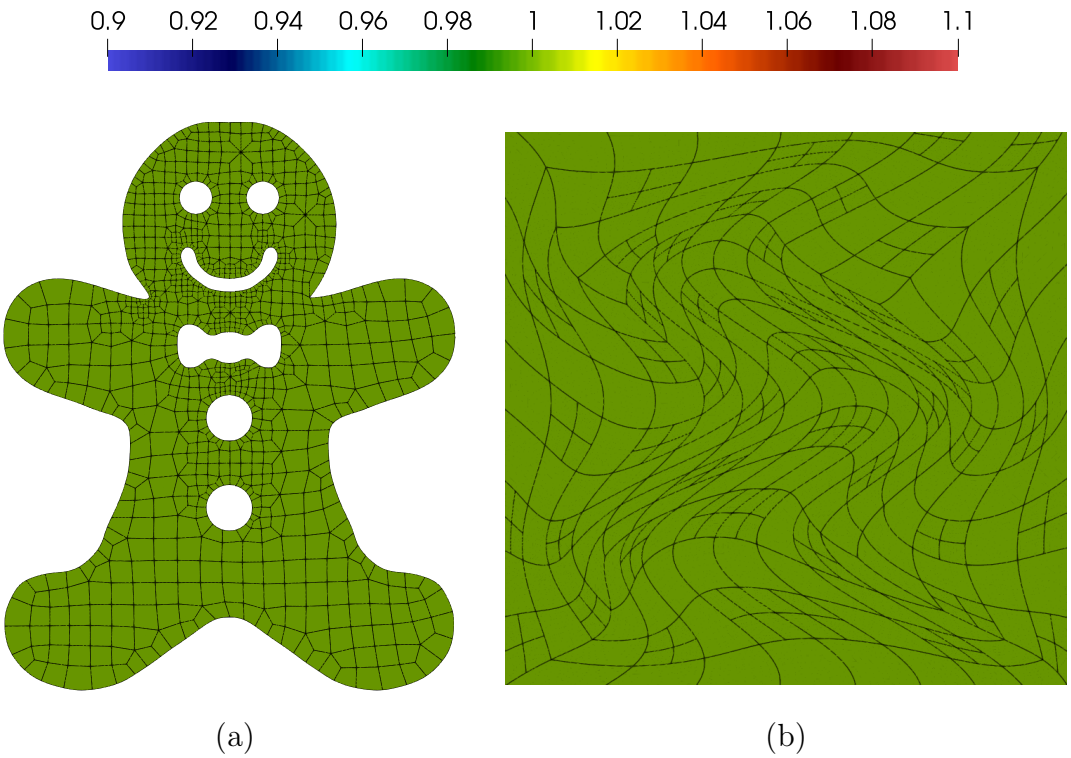


Figure 8.9. Density plots of free stream tests with mesh and solution polynomial degree $N = 6$ at $t = 10$ on (a) mesh with curved boundaries, (b) mesh with refined curved elements

8.7.2.2. Isentropic vortex

This is a test with exact solution taken from [90] where the domain is specified by the following transformation from $[0, 1]^2 \to \Omega$

$$\boldsymbol{x}(\xi, \eta) = \begin{pmatrix} \xi\, L_x - A_x\, L_y \sin(2\,\pi\,\eta) \\ \eta\, L_y + A_y\, L_x \sin(2\,\pi\,\xi) \end{pmatrix}$$

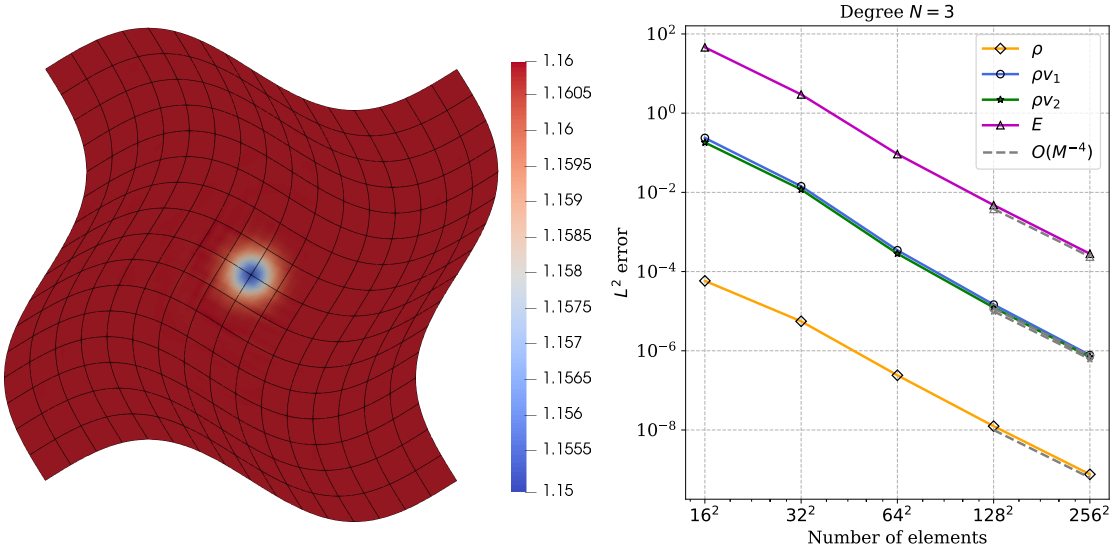
which is a distortion of the square $[0, L_x] \times [0, L_y]$ with sine waves of amplitudes A_x, A_y. Following [90], we choose length $L_x = L_y = 0.1$ and amplitudes $A_x = A_y = 0.1$. The boundaries are set to be periodic. A vortex with radius $R_v = 0.005$ is initialized in the curved domain with center $(x_v, y_v) = (L_x/2, L_y/2)$. The gas constant is taken to be $R_{\text{gas}} = 287.15$ and specific heat ratio $\gamma = 1.4$ as before. The free stream state is defined by the Mach number $M_0 = 0.5$, temperature $T_0 = 300$, pressure $p_0 = 10^5$, velocity $u_0 = M_0 \sqrt{\gamma\, R_{\text{gas}}\, T_0}$ and density $\rho_0 = \frac{p_0}{R_{\text{gas}}\, T_0}$. The initial condition $\boldsymbol{u}_0$ is given by

$$(\rho, u, v, p) = \left(\rho_0 \left(\frac{T}{T_0} \right)^{\frac{1}{\gamma - 1}}, u_0 \left(1 - \beta \frac{y - y_v}{R_v}\, e^{\frac{-r^2}{2}} \right), u_0\, \beta\, \frac{x - x_v}{R_v}\, e^{\frac{-r^2}{2}}, \rho(x, y)\, R_{\text{gas}}\, T \right)$$

$$T(x, y) = T_0 - \frac{(u_0\, \beta)^2}{2\, C_p}\, e^{-r^2}, \qquad r = \sqrt{(x - x_v)^2 + (y - y_v)^2}\,/\,R_v$$

where $C_p = R_{\text{gas}}\, \gamma\,/\,(\gamma - 1)$ is the heat capacity at constant pressure and $\beta = 0.2$ is the

vortex strength. The vortex moves in the positive x direction with speed u_0 so that the exact solution at time t is $\boldsymbol{u}(x,y,t) = \boldsymbol{u}_0(x - u_0 t, y)$ where $\boldsymbol{u}_0$ is extended outside Ω by periodicity. We simulate the propagation of the vortex for one time period $t_p = L_x / u_0$ and perform numerical convergence analysis for degree $N = 3$ in Figure 8.10b, showing optimal rates in grid versus L^2 error norm for all the conserved variables.



(a) (b)

Figure 8.10. Convergence analysis for isentropic vortex problem with polynomial degree $N = 3$. (a) Density plot, (b) L^2 error norm of conserved variables

8.7.2.3. Supersonic flow over cylinder

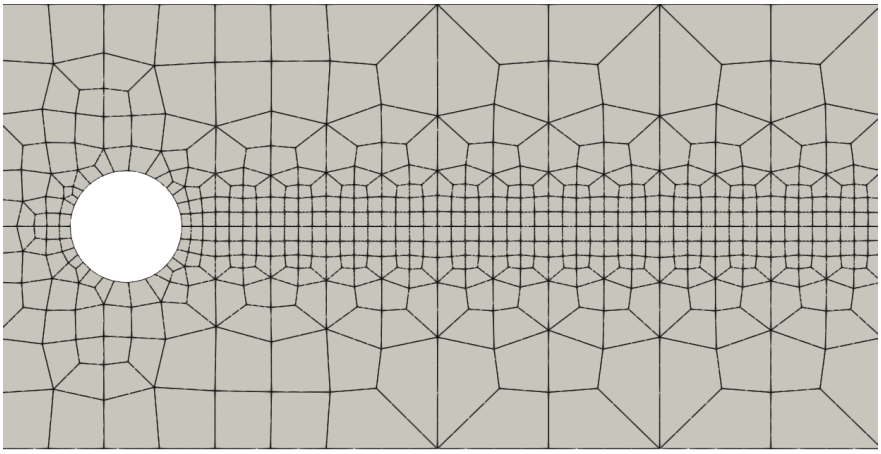
Supersonic flow over a cylinder is computed at a free stream Mach number of 3 with the initial condition

$$(\rho, u, v, p) = (1.4, 3, 0, 1)$$

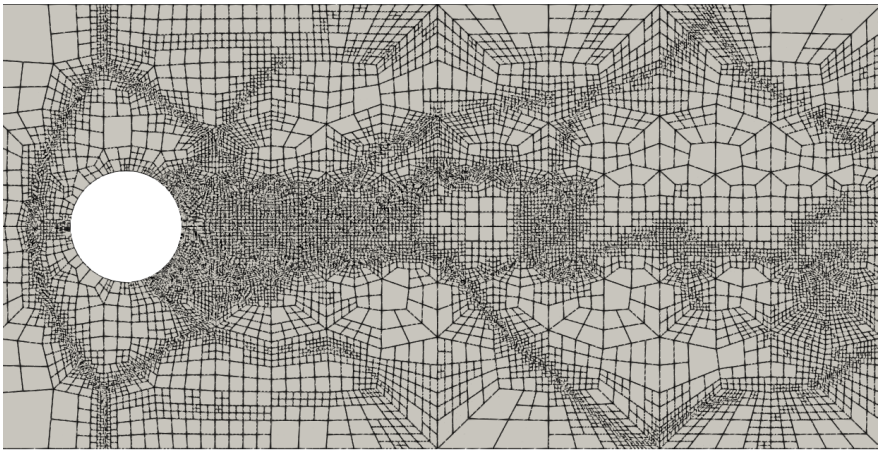
Solid wall boundary conditions are used at the top and bottom boundaries. A bow shock forms which reflects across the solid walls and interacts with the small vortices forming in the wake of the cylinder. The setup of Löhner's smoothness indicator (8.71) is taken from an example of `Trixi.jl` [140]

$$(\texttt{base\_level}, \texttt{med\_level}, \texttt{max\_level}) = (0, 3, 5)$$
$$(\texttt{med\_threshold}, \texttt{max\_threshold}) = (0.05, 0.1)$$

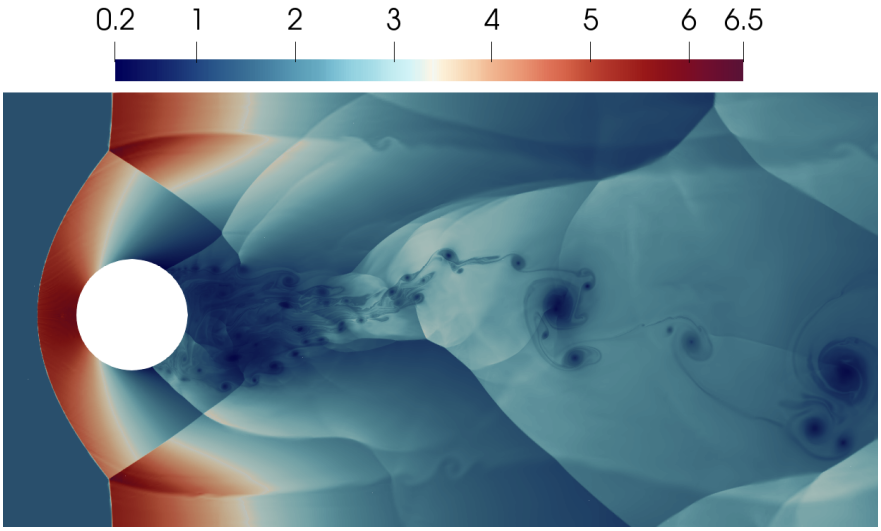
where `base_level` $= 0$ refers to mesh in Figure 8.11a. The flow consists of a strong shock and thus the positivity limiter had to be used to enforce admissibility. The flow behind the cylinder is highly unsteady, with reflected shocks and vortices interacting continuously. The density profile of the numerical solution at $t = 10$ is shown in Figure 8.11 with mesh and solution polynomial degree $N = 4$ using Löhner's indicator (8.71) for AMR. The AMR indicator is tracing the shocks and the vortex structures forming in the wake well. The initial mesh has 561 elements which first increase to 63000 elements followed by a fall to 39000 elements and then a steady increase to the peak of 85000 elements from which it steadily falls to 36000 elements by the end of the simulation. The mesh is refined or coarsened once every 100 time steps. In order to capture the same effective refinement, a uniform mesh will require 574464 elements.



(a)



(b)



(c)

Figure 8.11. Mach 3 flow over cylinder using solution and mesh polynomial degree $N = 4$ at $t = 10$ (a) Initial mesh, (b) adaptively refined mesh at final time, (c) density plot at final time.

8.7.2.4. Inviscid bow shock upstream of a blunt body

This test simulates steady supersonic flow over a blunt body and is taken from [90] which followed the description proposed by the high order computational fluid dynamics workshop [36]. The domain, also shown in Figure 8.12 consists of a left and a right boundary. The left boundary is an arc of a circle with origin $(3.85, 0)$ and radius 5.9 extended till $x = 0$ on both ends. The right boundary consists of (a) the blunt body and (b) straight-edged outlets. The straight-edged outlets are $\{(0, y): |y| > 0.5\}$ extended till the left boundary arc. The blunt body consists of a front of length 1 and two quarter circles of radius 0.5. The domain is initialized with a Mach 4 flow, which is given in primitive variables by

$$(\rho, u, v, p) = (1.4, 4, 0, 1) \tag{8.80}$$

The left boundary is set as supersonic inflow, the blunt body is a reflecting wall and the straight edges at $x = 0$ are supersonic outflow boundaries. Löhner's smoothness indicator (8.71) for AMR is set up as

$$(\texttt{base\_level}, \texttt{med\_level}, \texttt{max\_level}) = (0, 1, 2)$$
$$(\texttt{med\_threshold}, \texttt{max\_threshold}) = (0.05, 0.1)$$

where $\texttt{base\_level} = 0$ refers to mesh in Figure 8.12a. Since this is a test case with a strong bow shock, the positivity limiter had to be used to enforce admissibility. The pressure obtained with polynomial degree $N = 4$ is shown in Figure 8.12 with adaptive mesh refinement performed using Löhner's smoothness indicator (8.71) where the AMR procedure is seen to be refining the mesh in the region of the bow shock. The initial mesh (Figure 8.12a) has 244 elements which steadily increases to ∼1600 elements till $t \approx 1.5$ and then remains nearly constant as the solution reaches steady state. The mesh is adaptively refined or coarsened at every time step.

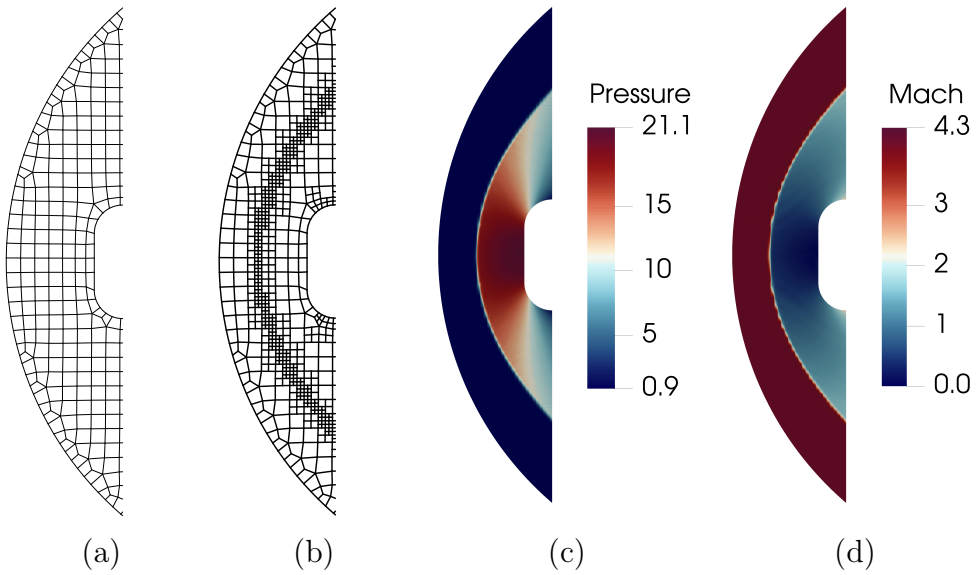


Figure 8.12. Mach 4 flow over blunt body using polynomial degree $N = 4$ showing (a) initial mesh, (b) adaptively refined mesh, (c) pressure plot, (d) Mach number plot

8.7.2.5. Transonic flow over NACA0012 airfoil

This is a steady transonic flow over the symmetric NACA0012 airfoil. The initial

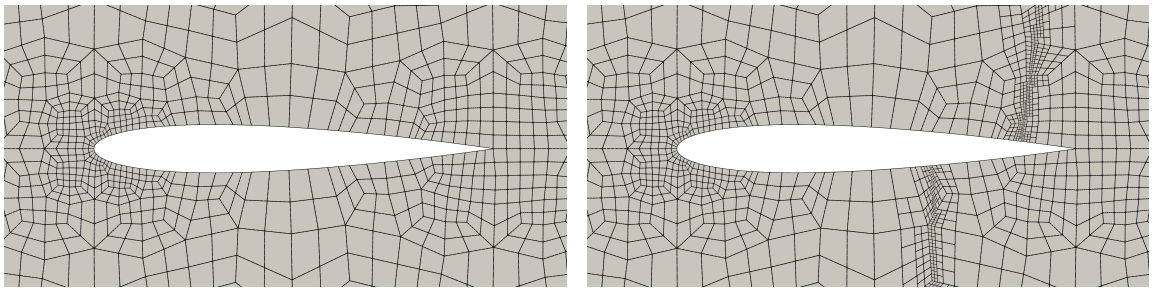
condition is taken to have Mach number $M_0 = 0.85$ and it is given in primitive variables as

$$(\rho, u, v, p) = \left(\frac{p_0}{T_0 R}, U_0 \cos\theta, U_0 \sin\theta, p_0 \right)$$

where $p_0 = 1, T_0 = 1, R = 287.87, \theta = \pi/180$, $U_0 = M_0 c_0$ and sound speed $c_0 = \sqrt{\gamma p_0 / \rho_0}$. The airfoil is of length 1 unit located in the rectangular domain $[-20, 20]^2$ and the initial mesh has 728 elements. We run the simulation with mesh and solution polynomial degree $N = 6$ using Löhner's smoothness indicator (8.71) for AMR with the setup

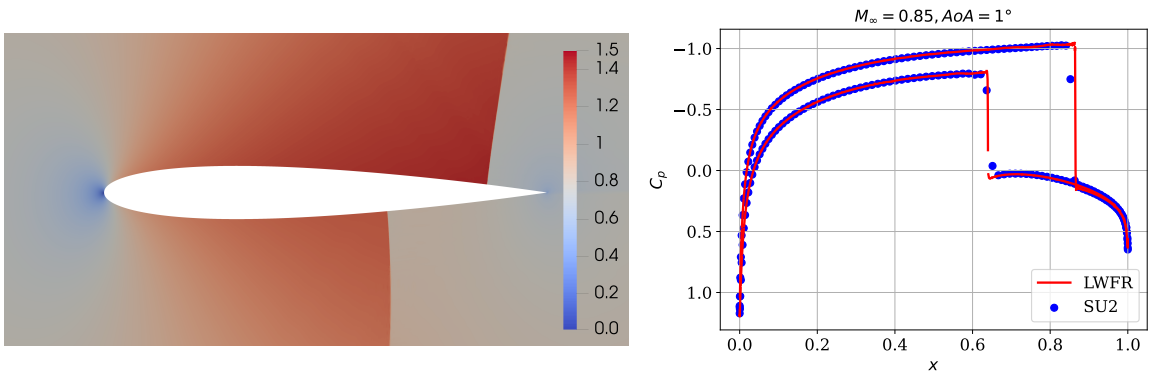
$$(\texttt{base\_level}, \texttt{med\_level}, \texttt{max\_level}) = (1, 3, 4)$$
$$(\texttt{med\_threshold}, \texttt{max\_threshold}) = (0.05, 0.1)$$

where $\texttt{base\_level} = 1$ refers to the mesh in Figure 8.13a . In Figure 8.13, we show the initial and adaptively refined mesh. In Figure 8.14, we show the Mach number and compare the coefficient of pressure C_p on the surface of airfoil with SU2 [70] results, seeing reasonable agreement in terms of the values and shock locations. The AMR procedure is found to steadily increase the number of elements till they peak at ≈4200 and decrease to stabilize at ∼3750; the region of the shocks is being refined by the AMR process. The mesh is adaptively refined or coarsened once every 100 time steps. In order to capture the same effective refinement, a uniform mesh will require 186368 elements.



(a) (b)

Figure 8.13. Meshes for transonic flow over NACA0012 airfoil. (a) Initial mesh (b) adaptively refined mesh



(a) (b)

Figure 8.14. Transonic flow over airfoil using degree $N = 6$ on adapted mesh (a) Mach number (b) Coefficient of pressure on the surface of the airfoil

8.7.3. Performance comparison of time stepping schemes

In Table 8.1, we show comparison of total time steps needed by error (Algorithm 8.4) and CFL (8.78) based time stepping methods for test cases where non-Cartesian meshes are used. The total time steps give a complete description of the cost because our experiments have shown that error estimation procedure only adds an additional computational cost of ∼4%. The relative and absolute tolerances τ_a, τ_r in (8.74) are taken to be the same, and denoted `tolE`. The iterations that are redone because of error or admissibility criterion in Algorithm 8.4 are counted as *failed* (shown in Table 8.1 in red) while the rest as *successful* (shown in Table 8.1 in blue). The comparisons are made between the two time stepping schemes as follows - the constant C_{CFL} in (8.78) is experimentally chosen to be the largest which can be used without admissibility violation while error based time stepping is shown with `tolE = 1e-6` and the best tolerance for the particular test case (which is either `1e-6` or `5e-6`). Note that the choice of `tolE = 1e-6` is made in all the results shown in previous sections. A poor quality (nearly degenerate) mesh (Figure 8.12b) was used in the flow over blunt body (Section 8.7.2.4) and thus the CFL based scheme could not run till the final time $t = 10$ without admissibility violation for any choice of C_{CFL}. However, the error-based time stepping scheme is able to finish the simulation by its ability to redo time steps; although there are many failed time steps as is to be expected. The error-based time stepping scheme is giving superior performance with `tolE = 1e-6` for the supersonic flow over cylinder and transonic flow over airfoil (curved meshes tests) with ratio of total time steps being 1.755 and 1.43 respectively. However, for the forward facing step test with a straight sided quadrilateral mesh, error based time stepping with `tolE = 1e-6` takes more time steps than the fine-tuned CFL based time stepping. However, increasing the tolerance to `tolE = 5e-6` gives the same performance as the CFL based time stepping. By using `tolE = 5e-6`, the performance of supersonic flow cylinder can be further obtained to get a ratio of 2.327. These results show the robustness of error-based time stepping and even improved efficiency in meshes with curved elements.

	CFL (8.78)	Error (Alg. 8.4) (Pass + Fail)			Ratio
		`tolE=1e-6`	$\frac{\text{CFL}}{\texttt{tolE=1e-6}}$	Best `tolE`	$\frac{\text{CFL}}{\text{Best tolE}}$
FF Step (8.7.1.4)	5706455	7661457 (7661453 + 4)	0.74	5646355 `(5e-6)` (5646355 + 5)	1.01
Cylinder (8.7.2.3)	1529064	871262 (871124 + 138)	1.755	657170 `(5e-6)` (651118+6052)	2.327
Blunt (8.7.2.4)	-	4200 (3800 + 400)	-	4200 `(1e-6)` (3800 + 400)	-
Airfoil (8.7.2.5)	6856828	4778674 (4778651 + 23)	1.43	4778674 `(1e-6)` (4778651 + 23)	1.43

Table 8.1. Number of time steps comparing error and CFL based methods

8.8. Summary and conclusions

The Lax-Wendroff Flux Reconstruction (LWFR) scheme has been extended to curvilinear and dynamic, locally adapted meshes. On curvilinear meshes, it is shown that satisfying the standard metric identities gives free stream preservation for the LWFR scheme. The subcell based blending scheme of Chapter 5 has been extended to curvilinear meshes along with the provable admissibility preservation (Section 5.2) based on the idea of appropriately choosing the *blended numerical flux* (Section 5.5) at the element interfaces. Adaptive Mesh Refinement has been introduced for LWFR scheme using the Mortar Element Method (MEM) of [106]. Fourier stability analysis to compute the optimal CFL number as in Section 4.4 is based on uniform Cartesian meshes and does not apply to curvilinear grids. Thus, in order to use a wave speed based time step computation, the CFL number has to be fine tuned for every problem, especially for curved grids. In order to decrease the fine-tuning process, an embedded error-based time step computation method was introduced for LWFR by taking difference between two element local evolutions of the solutions using the local time averaged flux approximations - one which is order $N+1$ and the other truncated to be order N. This is the first time error-based time stepping has been introduced for a single stage evolution method for solving time dependent equations. Numerical results using compressible Euler equations were shown to validate the claims. It was shown that free stream condition is satisfied on curvilinear meshes even with non-conformal elements and that the LWFR scheme shows optimal convergence rates on domains with curved boundaries and meshes. The AMR with shock capturing was tested on various problems to show the scheme's robustness and capability to automatically refine in regions comprising of relevant features like shocks and small scale structures. The error based time stepping scheme is able to run with fewer time steps in comparison to the CFL based scheme and with less fine tuning. The speed-up obtained by error based time stepping for curvilinear meshes was in the range of 1.43 and 2.33.

Chapter 9

Parabolic equations

9.1. Introduction

In this chapter, we develop the LWFR scheme to solve second order PDE in conservative form using the BR1 scheme [22]. Examples of such equations include compressible Navier-Stokes and resistive magnetohydrodynamics. Following Chapter 8, we solve second order equations on curvilinear meshes and use error based time stepping.

This chapter is organized as follows. In Section 9.2, the notations and transformations of second order equations from curved element to a reference cube are reviewed. In Section 9.3, the LWFR scheme for second order equations is introduced. In Section 9.4, the treatment of boundary conditions is described. The numerical results are shown in Section 9.5 and a summary of the chapter is provided in Section 9.6.

9.2. Curvilinear coordinates for parabolic equations

We work with a system of parabolic equations in conservative form in d dimensions

$$\partial_t \boldsymbol{u} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}^a(\boldsymbol{u}) = \nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}^v(\boldsymbol{u}, \nabla_{\boldsymbol{x}} \boldsymbol{u}) \tag{9.1}$$

with some initial and boundary conditions. Here, $\boldsymbol{u} \in \mathbb{R}^p$ is the solution vector and its gradient is the matrix $\nabla_{\boldsymbol{x}} \boldsymbol{u} = (\partial_{x_1} \boldsymbol{u}, \dots, \partial_{x_d} \boldsymbol{u}) \in \mathbb{R}^{p \times d}$. The $\boldsymbol{f}^a$, $\boldsymbol{f}^v$ are the advective and viscous fluxes and can be seen as matrices $\boldsymbol{f} = (\boldsymbol{f}_1, \dots, \boldsymbol{f}_d) \in \mathbb{R}^{p \times d}$, $\boldsymbol{x}$ is in domain $\Omega \subset \mathbb{R}^d$ and divergence of a flux is given by $\nabla_{\boldsymbol{x}} \cdot \boldsymbol{f} = \sum_{i=1}^d \partial_{x_i} \boldsymbol{f}_i$. Following [78], we introduce some notations to describe the scheme. The action of a vector $\boldsymbol{b} \in \mathbb{R}^d$ on $\boldsymbol{v} \in \mathbb{R}^p$ gives $\boldsymbol{b}\,\boldsymbol{v} \in \mathbb{R}^{p \times d}$ which is defined component-wise as

$$\boldsymbol{b}\,\boldsymbol{v} = (b_j\,\boldsymbol{v})_{j=1}^d \tag{9.2}$$

Further, the action of a matrix $\mathcal{B} = (\boldsymbol{b}_1, \dots, \boldsymbol{b}_d) \in \mathbb{R}^{d \times d}$ on $\boldsymbol{v} = (\boldsymbol{v}_1, \dots, \boldsymbol{v}_d) \in \mathbb{R}^{p \times d}$ also gives $\mathcal{B}\,\boldsymbol{v} \in \mathbb{R}^{p \times d}$ defined as

$$\mathcal{B}\,\boldsymbol{v} = \sum_{i=1}^d \boldsymbol{b}_i\,\boldsymbol{v}_i, \qquad \boldsymbol{b}_i\,\boldsymbol{v}_i = (b_{ij}\,\boldsymbol{v}_i)_{j=1}^d \tag{9.3}$$

The second order system (9.1) will be reduced to a first order system of equations following the BR1 scheme [22] where an auxiliary variable $\boldsymbol{q}$ is introduced

$$\begin{aligned} \boldsymbol{q} - \nabla_{\boldsymbol{x}} \boldsymbol{u} &= \boldsymbol{0} \\ \partial_t \boldsymbol{u} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{q}) &= \boldsymbol{0} \end{aligned} \tag{9.4}$$

We will be using the same FR elements $\{\Omega_e\}$ (8.3), reference map $\boldsymbol{x} = \Theta(\boldsymbol{\xi})$ (8.4), multi-index $\boldsymbol{p} \in \mathbb{N}_N^d$ (8.5), Lagrange polynomial basis $\{\ell_{\boldsymbol{p}}\}_{\boldsymbol{p}\in\mathbb{N}_N^d}$ as in Section 8.2 and again denote the covariant and contravariant basis vectors as $\{\boldsymbol{a}_i\}_{i=1}^3, \{\boldsymbol{a}^i\}_{i=1}^3$ (Definition 8.1, 8.2). The covariant and contravariant vectors will now be used to map the equations (9.4) to the reference element $\Omega_o = [-1, 1]^d$.

Using the Leibniz product rule of differentiation, and the metric identity (8.14) on the transformation of a gradient transformation (8.11), we get the non-conservative form of gradient of a scalar ϕ and thus of a vector $\boldsymbol{u}$ in vector action notation (9.2)

$$\nabla_{\boldsymbol{x}} \phi = \frac{1}{J} \sum_{i=1}^{d} J \boldsymbol{a}^i \frac{\partial \phi}{\partial \xi^i}, \qquad \nabla_{\boldsymbol{x}} \boldsymbol{u} = \frac{1}{J} \sum_{i=1}^{d} J \boldsymbol{a}^i \frac{\partial \boldsymbol{u}}{\partial \xi^i} \tag{9.5}$$

Following the notation of [78], define the transformation matrix $\mathcal{M} = (J\boldsymbol{a}^1, \ldots, J\boldsymbol{a}^d) \in \mathbb{R}^{d\times d}$ so that with the matrix action notation (9.3)

$$\nabla_{\boldsymbol{x}} \boldsymbol{u} = \frac{1}{J} \mathcal{M} \nabla_{\boldsymbol{\xi}} \boldsymbol{u} \tag{9.6}$$

Within each element Ω_e, performing change of variables with the reference map Θ_e (8.11, 8.10), the first order system (9.4) transforms into

$$\begin{aligned} J\boldsymbol{q} - \mathcal{M} \nabla_{\boldsymbol{\xi}} \boldsymbol{u} &= \boldsymbol{0} \\ J\partial_t \boldsymbol{u} + \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}^a &= \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}^v(\boldsymbol{u}, \boldsymbol{q}) \end{aligned} \tag{9.7}$$

where, as in (8.13), we have

$$(\tilde{\boldsymbol{f}}^\alpha)^i = J\boldsymbol{a}^i \cdot \tilde{\boldsymbol{f}}^\alpha = \sum_{n=1}^{d} J a_n^i \boldsymbol{f}_n^\alpha, \qquad \alpha = a, v \tag{9.8}$$

9.3. Lax-Wendroff flux reconstruction

As in the earlier chapters, the solution of (9.1) will be approximated by piecewise polynomial functions which are allowed to be discontinuous across the elements. Within each element Ω_e, the solution will be represented by degree N Lagrange basis in the reference coordinates (8.15) and $\boldsymbol{u}_{e,\boldsymbol{p}}$ are the unknown values at solution points which are taken to be GLL points.

9.3.1. Solving for $\boldsymbol{q}$

The system (9.7) is solved at each time step for evolving the numerical solution from time t^n to t^{n+1}, where the first step is solving the equations for $\boldsymbol{q}$. The BR1 scheme was initially introduced for Discontinuous Galerkin (DG) method in [22] and is used as a Flux Reconstruction (FR) scheme by exploiting the equivalence between FR and DG [94], see for example [195]. Here, we show the first step in the FR framework[9.1], repeating some notations from Chapter 8. Recall that we defined the multi-index $\boldsymbol{p}=(p_i)_{i=1}^d$ where $p_i\in\{0,1\ldots,N\}$. Let $i\in\{1,\ldots,d\}$ denote a coordinate direction and $S\in\{L,R\}$ so that (S,i) corresponds to the face $\partial\Omega_{o,i}^s$ in direction i on side S which has the reference outward normal $\hat{\boldsymbol{n}}_{S,i}$, see Figure 8.1. Thus, $\partial\Omega_{o,i}^R$ denotes the face where reference outward normal is $\hat{\boldsymbol{n}}_{R,i}=\boldsymbol{e}_i$ and $\partial\Omega_{o,i}^L$ has outward unit normal $\hat{\boldsymbol{n}}_{L,i}=-\hat{\boldsymbol{n}}_{R,i}$.

The FR scheme is a collocation at each of the solution points $\{\boldsymbol{\xi_p}=(\xi_{p_i})_{i=1}^d, p_i=0,\ldots,N\}$. We will thus explain the scheme for a fixed $\boldsymbol{\xi}=\boldsymbol{\xi_p}$ and denote $\boldsymbol{\xi}_i^S$ as the projection of $\boldsymbol{\xi}$ to the face $S=L,R$ in the i^{th} direction (Figure 8.1), as defined in (8.16). A correction of $\boldsymbol{u}_e^\delta$ is performed to obtain $\boldsymbol{u}_e$ as

$$\boldsymbol{u}_e(\boldsymbol{\xi})=\boldsymbol{u}_e^\delta(\boldsymbol{\xi})+(\boldsymbol{u}_e^*-\boldsymbol{u}_e^\delta)(\boldsymbol{\xi}_i^R)\,g_R(\xi_{p_i})+(\boldsymbol{u}_e^*-\boldsymbol{u}_e^\delta)(\boldsymbol{\xi}_i^L)\,g_L(\xi_{p_i}) \tag{9.9}$$

where $\boldsymbol{u}_e^\delta(\boldsymbol{\xi}_i^S)$ denotes the trace value of the normal flux in element e and $\boldsymbol{u}_e^*(\boldsymbol{\xi}_i^S)$ denotes an approximation of the solution at the interface (S,i) which is chosen to be

$$\boldsymbol{u}_e^*(\boldsymbol{\xi}_i^S)=\frac{1}{2}\left(\boldsymbol{u}_{S,i}^+ + \boldsymbol{u}_{S,i}^-\right) \tag{9.10}$$

and g_L, g_R are FR correction functions [94] (Section 3.4). Thus, $\boldsymbol{q}$ can be obtained from (9.7) as

$$\boldsymbol{q}=\frac{1}{J}\mathcal{M}\,\nabla_{\boldsymbol{\xi}}\,\boldsymbol{u}_e(\boldsymbol{\xi}) \tag{9.11}$$

9.3.2. Time averaging

The LWFR scheme is obtained by performing the Lax-Wendroff procedure for Cartesian domains on the transformed equation (9.7). Let $\boldsymbol{u}^n$ denote the solution at time $t=t_n$ and $\boldsymbol{q}^n$ denoting the gradient computed from (9.11). As we did in the previous chapters, the solution at the next time level can be written as

$$\boldsymbol{u}^{n+1}=\boldsymbol{u}^n+\sum_{k=1}^{N+1}\frac{\Delta t^k}{k!}\,\partial_t^{(k)}\,\boldsymbol{u}^n+O(\Delta t^{N+2})$$

where N is the solution polynomial degree. Then, use $\boldsymbol{u}_t=-\frac{1}{J}\nabla_{\boldsymbol{\xi}}\cdot(\tilde{\boldsymbol{f}}^a-\tilde{\boldsymbol{f}}^v)$ from (9.7) to swap a temporal derivative with a spatial derivative and retain terms up to Δt to get

$$\boldsymbol{u}^{n+1}=\boldsymbol{u}^n-\frac{1}{J}\sum_{k=1}^{N+1}\frac{\Delta t^k}{k!}\,\partial_t^{(k-1)}\left(\nabla_{\boldsymbol{\xi}}\cdot\tilde{\boldsymbol{f}}^a\right)+\frac{1}{J}\sum_{k=1}^{N+1}\frac{\Delta t^k}{k!}\,\partial_t^{(k-1)}\left(\nabla_{\boldsymbol{\xi}}\cdot\tilde{\boldsymbol{f}}^v\right)$$

9.1. Unlike FR, there is no *physical flux* in the first step of the BR1 scheme where we solve for $\boldsymbol{q}$. We call this step FR due to the application of correction functions to enforce global continuity.

Shifting indices and writing in a conservative form gives

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \frac{\Delta t}{J}\nabla_{\boldsymbol{\xi}}\cdot\tilde{\boldsymbol{F}}^a + \frac{\Delta t}{J}\nabla_{\boldsymbol{\xi}}\cdot\tilde{\boldsymbol{F}}^v \tag{9.12}$$

where $\tilde{\boldsymbol{F}}^a, \tilde{\boldsymbol{F}}^v$ are time averages of the contravariant advective and viscous fluxes $\tilde{\boldsymbol{f}}^a, \tilde{\boldsymbol{f}}^v$

$$\tilde{\boldsymbol{F}}^\alpha = \sum_{k=0}^{N} \frac{\Delta t^k}{(k+1)!}\partial_t^k \tilde{\boldsymbol{f}}^\alpha \approx \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \tilde{\boldsymbol{f}}^\alpha \,\mathrm{d}t, \qquad \alpha = a, v \tag{9.13}$$

We will find local order $N+1$ approximations $\tilde{\boldsymbol{F}}_e^{a\delta}, \tilde{\boldsymbol{F}}_e^{v\delta}$ to $\tilde{\boldsymbol{F}}^a, \tilde{\boldsymbol{F}}^v$ (Section 9.3.2.1) which will be discontinuous across element interfaces. Then, in order to couple the neighbouring elements, continuity of the normal fluxes at the interfaces will be enforced by constructing the *continuous flux approximation* using the FR correction functions g_L, g_R [94] (Section 3.4) and the numerical fluxes. Thus, once the local approximations $\tilde{\boldsymbol{F}}_e^{a\delta}, \tilde{\boldsymbol{F}}_e^{v\delta}$ are computed, we construct the advective and viscous numerical fluxes for element e in coordinate direction $i \in \{1, \dots d\}$ at the side $S \in \{L, R\}$ (following the notation from (8.16)) by using Rusanov's [152] and the central flux respectively

$$(\tilde{\boldsymbol{F}}_e^a \cdot \hat{\boldsymbol{n}}_{S,i})^* = \frac{1}{2}\left((\tilde{\boldsymbol{F}}^{a\delta}\cdot\hat{\boldsymbol{n}}_{S,i})^+ + (\tilde{\boldsymbol{F}}^{a\delta}\cdot\hat{\boldsymbol{n}}_{S,i})^-\right) - \frac{\lambda_{S,i}}{2}\left(\boldsymbol{U}_{S,i}^+ - \boldsymbol{U}_{S,i}^-\right) \tag{9.14}$$

$$(\tilde{\boldsymbol{F}}_e^v \cdot \hat{\boldsymbol{n}}_{S,i})^* = \frac{1}{2}\left((\tilde{\boldsymbol{F}}^{v\delta}\cdot\hat{\boldsymbol{n}}_{S,i})^+ + (\tilde{\boldsymbol{F}}^{v\delta}\cdot\hat{\boldsymbol{n}}_{S,i})^-\right) \tag{9.15}$$

The $(\tilde{\boldsymbol{F}}^\delta \cdot \hat{\boldsymbol{n}}_{S,i})^\pm$ and $\boldsymbol{U}_{S,i}^\pm$ denote the trace values of the normal flux and time average solution from outer, inner directions respectively; the inner direction corresponds to the element e while the outer direction corresponds to its neighbour across the interface (S, i). The Rusanov's flux (9.13) is exactly as discussed in the inviscid case (8.26).

The advective and viscous fluxes have been treated separately so far to keep a simple implementation of the different boundary conditions of the two. However, for the final evolution (9.12), we can combine them to define $\tilde{\boldsymbol{F}} = \tilde{\boldsymbol{F}}^a - \tilde{\boldsymbol{F}}^v$, so that the interface numerical fluxes can also be summed as $(\tilde{\boldsymbol{F}}_e \cdot \hat{\boldsymbol{n}}_{S,i})^* = (\tilde{\boldsymbol{F}}_e^a \cdot \hat{\boldsymbol{n}}_{S,i})^* - (\tilde{\boldsymbol{F}}_e^v \cdot \hat{\boldsymbol{n}}_{S,i})^*$ and thus the continuous time averaged numerical flux is constructed as

$$\begin{aligned}(\tilde{\boldsymbol{F}}_e(\boldsymbol{\xi}))^i = (\tilde{\boldsymbol{F}}_e^\delta(\boldsymbol{\xi}))^i \;& + ((\tilde{\boldsymbol{F}}_e\cdot\hat{\boldsymbol{n}}_{R,i})^* - \tilde{\boldsymbol{F}}_e^\delta\cdot\hat{\boldsymbol{n}}_{R,i})(\boldsymbol{\xi}_i^R)\, g_R(\xi_{p_i}) \\ & - ((\tilde{\boldsymbol{F}}_e\cdot\hat{\boldsymbol{n}}_{L,i})^* - \tilde{\boldsymbol{F}}_e^\delta\cdot\hat{\boldsymbol{n}}_{L,i})(\boldsymbol{\xi}_i^L)\, g_L(\xi_{p_i})\end{aligned} \tag{9.16}$$

Substituting $\tilde{\boldsymbol{F}}_e$ in (9.12), the explicit LWFR update is

$$\begin{aligned}\boldsymbol{u}_{e,\boldsymbol{p}}^{n+1} = \boldsymbol{u}_{e,\boldsymbol{p}}^n - \frac{\Delta t}{J_{e,\boldsymbol{p}}}\nabla_{\boldsymbol{\xi}}\cdot\tilde{\boldsymbol{F}}_e^\delta(\boldsymbol{\xi}_{\boldsymbol{p}}) \;& - \frac{\Delta t}{J_{e,\boldsymbol{p}}}\sum_{i=1}^{d}\left((\tilde{\boldsymbol{F}}_e\cdot\hat{\boldsymbol{n}}_{R,i})^* - \tilde{\boldsymbol{F}}_e^\delta\cdot\hat{\boldsymbol{n}}_{R,i}\right)(\boldsymbol{\xi}_i^R)\, g_R'(\xi_{p_i}) \\ & + \frac{\Delta t}{J_{e,\boldsymbol{p}}}\sum_{i=1}^{d}\left((\tilde{\boldsymbol{F}}_e\cdot\hat{\boldsymbol{n}}_{L,i})^* - \tilde{\boldsymbol{F}}_e^\delta\cdot\hat{\boldsymbol{n}}_{L,i}\right)(\boldsymbol{\xi}_i^L)\, g_L'(\xi_{p_i})\end{aligned} \tag{9.17}$$

9.3.2.1. Approximate Lax-Wendroff procedure

We now illustrate how to locally approximate $\tilde{\boldsymbol{F}}^{a\delta}, \tilde{\boldsymbol{F}}^{v\delta}$ (9.13) for degree $N=1$ using the approximate Lax-Wendroff procedure [208]. For $N=1$, (9.13) requires $\partial_t \tilde{\boldsymbol{f}}^a, \partial_t \tilde{\boldsymbol{f}}^v$ which are

$$\begin{aligned} \partial_t \tilde{\boldsymbol{f}}^a &\approx \frac{\tilde{\boldsymbol{f}}^a(\boldsymbol{u}+\Delta t \boldsymbol{u}_t) - \tilde{\boldsymbol{f}}^a(\boldsymbol{u}-\Delta t \boldsymbol{u}_t)}{2\,\Delta t} =: \partial_t \tilde{\boldsymbol{f}}^{a\delta} \\ \partial_t \tilde{\boldsymbol{f}}^v &\approx \frac{\tilde{\boldsymbol{f}}^v(\boldsymbol{u}+\Delta t\, \boldsymbol{u}_t, \nabla \boldsymbol{u}+\Delta t\, (\nabla \boldsymbol{u})_t) - \tilde{\boldsymbol{f}}^v(\boldsymbol{u}-\Delta t\, \boldsymbol{u}_t, \nabla \boldsymbol{u}-\Delta t\, (\nabla \boldsymbol{u})_t)}{2\,\Delta t} =: \partial_t \tilde{\boldsymbol{f}}^{v\delta} \end{aligned} \tag{9.18}$$

and $\boldsymbol{u}_t, (\nabla \boldsymbol{u})_t$ are approximated using (9.7)

$$\boldsymbol{u}_t = -\frac{1}{J} \nabla_{\boldsymbol{\xi}} \cdot (\tilde{\boldsymbol{f}}^{a\delta} - \tilde{\boldsymbol{f}}^{v\delta}), \qquad (\nabla \boldsymbol{u})_t = \frac{1}{J} \mathcal{M}\, \nabla_{\boldsymbol{\xi}}\, \boldsymbol{u}_t \tag{9.19}$$

where $\tilde{\boldsymbol{f}}_e^{a\delta}, \tilde{\boldsymbol{f}}_e^{v\delta}$ are degree N cell local approximations to the fluxes $\tilde{\boldsymbol{f}}^a, \tilde{\boldsymbol{f}}^v$ given in (9.7) constructed by interpolation (8.17)

$$(\tilde{\boldsymbol{f}}_e^{a\delta})_i = \sum_{\boldsymbol{p}} \tilde{\boldsymbol{f}}^a(\boldsymbol{u}_{e,p})\, \ell_{\boldsymbol{p}}, \qquad (\tilde{\boldsymbol{f}}^{v\delta})_i = \sum_{\boldsymbol{p}} \tilde{\boldsymbol{f}}^v(\boldsymbol{u}_{e,p}, \boldsymbol{q}_{e,p})\, \ell_{\boldsymbol{p}} \tag{9.20}$$

where $\boldsymbol{q}$ is obtained in (9.11). The procedure for other degrees will be similar following Section 4.2.4.

9.3.3. Free stream preservation

Assume a constant solution $\boldsymbol{u}^n = \boldsymbol{c}$. The correction terms in (9.9) will be zero since a globally constant solution will be continuous across element interfaces. Thus, $\boldsymbol{u}_e = \boldsymbol{u}_e^\delta = \boldsymbol{c}$, proving that $\boldsymbol{q} = \boldsymbol{0}$ from (9.11). Thus, we can now suppress dependence of $\boldsymbol{q}$ on $\boldsymbol{f}^v$ and, in particular, write $\tilde{\boldsymbol{f}}^v = \tilde{\boldsymbol{f}}^v(\boldsymbol{u}_e)$. Thus, the equation we are solving for evolution from t^n to t^{n+1} is now

$$\boldsymbol{u}_t + \frac{1}{J} \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}(\boldsymbol{u}) = \boldsymbol{0}, \qquad \tilde{\boldsymbol{f}} = \tilde{\boldsymbol{f}}^a - \tilde{\boldsymbol{f}}^v$$

implying that the arguments for free stream preservation for LWFR on hyperbolic conservation laws used in Chapter 8 apply to parabolic equations. Thus, as proven in Section 8.3.4, the free stream will be preserved as long the metric identity (8.14) is satisfied for interpolated metric terms (8.35).

9.4. Boundary conditions

In this section, we discuss the treatment of additional boundary conditions required to solve second order equations (9.1). We explain the implementation for the 1-D scheme which is applied to higher dimensions across normal direction. Consider a grid with elements $\{\Omega_e\}_{e=1}^M$ where Ω_1, Ω_M are the left, right boundary elements. In addition to the advective numerical flux (9.13), application of boundary conditions is needed in the first step of BR1 scheme when computing (9.10) and when computing the viscous central flux (9.14). These additional boundary conditions are discussed in this section. We denote the discontinuous numerical solution as $\boldsymbol{u}_e^\delta(\xi)$ and the globally continuous approximation is given by

$$\boldsymbol{u}_e(\xi) = \boldsymbol{u}_e^\delta(\xi) + (\boldsymbol{u}_{e+\frac{1}{2}} - \boldsymbol{u}^-_{e+\frac{1}{2}})\, g_R(\xi) + (\boldsymbol{u}_{e-\frac{1}{2}} - \boldsymbol{u}^+_{e-\frac{1}{2}})\, g_L(\xi) \tag{9.21}$$

where $\boldsymbol{u}_{e+\frac{1}{2}}$ is the interface value given by

$$\boldsymbol{u}_{e+\frac{1}{2}} = \frac{1}{2}\left(\boldsymbol{u}^+_{e+\frac{1}{2}} + \boldsymbol{u}^-_{e+\frac{1}{2}}\right) \tag{9.22}$$

Similarly, the discontinuous viscous flux approximation is denoted as $\boldsymbol{F}^{v\delta}_e(\xi)$ and its globally continuous approximation is given by

$$\boldsymbol{F}^v_e(\xi) = \boldsymbol{F}^{v\delta}_e(\xi) + \left(\boldsymbol{F}^v_{e+\frac{1}{2}} - \boldsymbol{F}^{v-}_{e+\frac{1}{2}}\right) g_R(\xi) + \left(\boldsymbol{F}^v_{e-\frac{1}{2}} - \boldsymbol{F}^{v+}_{e+\frac{1}{2}}\right) g_L(\xi) \tag{9.23}$$

where $\boldsymbol{F}^v_{e+\frac{1}{2}}$ is the interface value which we compute as

$$\boldsymbol{F}^v_{e+\frac{1}{2}} = \frac{1}{2}\left(\boldsymbol{F}^{v+}_{e+\frac{1}{2}} + \boldsymbol{F}^{v-}_{e+\frac{1}{2}}\right) \tag{9.24}$$

With these notations, application of boundary conditions involves specification of $\boldsymbol{u}_{\frac{1}{2}}$, $\boldsymbol{u}_{M+\frac{1}{2}}$ and $\boldsymbol{F}^v_{\frac{1}{2}}, \boldsymbol{F}^v_{M+\frac{1}{2}}$. In some cases, the boundary conditions are enforced through the *ghost values* which are $\boldsymbol{F}^{v+}_{M+\frac{1}{2}}, \boldsymbol{u}^+_{M+\frac{1}{2}}$ for the right boundary and $\boldsymbol{F}^{v-}_{\frac{1}{2}}, \boldsymbol{u}^-_{\frac{1}{2}}$ for the left boundary. After specification of the ghost values, (9.22, 9.24) can be used to compute the boundary values. In other cases, the boundary values $\boldsymbol{u}_{\frac{1}{2}}, \boldsymbol{u}_{M+\frac{1}{2}}$ and $\boldsymbol{F}^v_{\frac{1}{2}}, \boldsymbol{F}^v_{M+\frac{1}{2}}$ are specified directly.

Periodic boundary. In case of periodic boundaries, the ghost values are specified as follows.

$$\begin{aligned} \boldsymbol{F}^{v+}_{M+\frac{1}{2}}, \boldsymbol{u}^+_{M+\frac{1}{2}} &= \boldsymbol{F}^{v+}_{\frac{1}{2}}, \boldsymbol{u}^+_{\frac{1}{2}} \\ \boldsymbol{F}^{v-}_{\frac{1}{2}}, \boldsymbol{u}^-_{\frac{1}{2}} &= \boldsymbol{F}^{v-}_{M+\frac{1}{2}}, \boldsymbol{u}^-_{M+\frac{1}{2}} \end{aligned}$$

This enables us to compute (9.21, 9.23) at the boundary faces.

Dirichlet/Inflow boundary. Assume that the left boundary is an inflow boundary. Let the boundary condition be given by $\boldsymbol{u}(0,t) = \boldsymbol{g}(t)$. The solution at the boundary is given by

$$\boldsymbol{u}_{\frac{1}{2}} = \boldsymbol{g}(t)$$

The viscous flux at boundary is computed as

$$\boldsymbol{F}_{\frac{1}{2}} \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \boldsymbol{f}^v(\boldsymbol{g}(t), (\nabla \boldsymbol{u}_1(t_n))^-)\,\mathrm{d}t$$

If the integral cannot be computed analytically, then it is approximated by quadrature in time. From (9.13), we see that integral must be at least accurate to $O(\Delta t^{N+1})$ which is of the same order as the neglected terms in (9.13). In the numerical tests, we use

$(N+1)$-point Gauss-Legendre quadrature which ensures the required accuracy.

Outflow boundary. Assume that the right boundary is an outflow boundary. In this case, the values across the right boundary are computed using the interior solution and flux so that $\boldsymbol{u}_{M+\frac{1}{2}} = \boldsymbol{u}^{-}_{M+\frac{1}{2}}$ and $\boldsymbol{F}^{v}_{M+\frac{1}{2}} = \boldsymbol{F}^{v-}_{M+\frac{1}{2}}$ where $\boldsymbol{F}^{v-}_{M+\frac{1}{2}}$ is obtained from the Lax-Wendroff procedure.

The remaining boundary conditions used in this chapter are specific to the Navier-Stokes equations (2.14) and are explained for 2-D. The viscous flux is given by

$$\boldsymbol{f}^{v} = \begin{pmatrix} 0 \\ \boldsymbol{\tau} \\ \boldsymbol{\tau v} - \nabla \boldsymbol{Q} \end{pmatrix},$$

We will assume that the respective boundary conditions are on left boundary element with index $\boldsymbol{e} = (1, e_y)$ whose left face is given by $\boldsymbol{e}_f = (1/2, e_y)$.

No-slip, adiabatic walls.

At no-slip boundaries, tangential component of velocity vector $\boldsymbol{v}$ is set to be the speed of the wall, while the normal component is set to zero. In case of the left face $\boldsymbol{e}_f = (1/2, e_y)$, the velocity is set to $\boldsymbol{v}_{\boldsymbol{e}_f} = (0, v_{e_y})$. Thus, the boundary value of solution is specified in primitive variables as

$$\boldsymbol{u}^{\text{prim}}_{\boldsymbol{e}_f} = (\rho^{+}_{\boldsymbol{e}_f}, \boldsymbol{v}_{\boldsymbol{e}_f}, p^{+}_{\boldsymbol{e}_f})$$

Adiabatic walls are those where the normal heat flux is zero and thus the viscous flux is specified as

$$\begin{aligned} (\boldsymbol{F}^{v}_{\boldsymbol{e}_f} \cdot \boldsymbol{n}) &= ((\boldsymbol{F}^{v+}_{\boldsymbol{e}_f} \cdot \boldsymbol{n})^1, (\boldsymbol{F}^{v+}_{\boldsymbol{e}_f} \cdot \boldsymbol{n})^2, (\boldsymbol{F}^{v+}_{\boldsymbol{e}_f} \cdot \boldsymbol{n})^3, (\boldsymbol{\tau}^{+}_{\boldsymbol{e}_f} \boldsymbol{n}) \cdot \boldsymbol{v}_{\boldsymbol{e}_f}) \\ &= (0, (\boldsymbol{F}^{v+}_{\boldsymbol{e}_f} \cdot \boldsymbol{n})^2, (\boldsymbol{F}^{v+}_{\boldsymbol{e}_f} \cdot \boldsymbol{n})^3, (\boldsymbol{\tau}^{+}_{\boldsymbol{e}_f} \boldsymbol{n}) \cdot \boldsymbol{v}_{\boldsymbol{e}_f}) \end{aligned}$$

where $\boldsymbol{n}$ is the normal vector at the face $\boldsymbol{e}_f$.

In numerical results, unless specified otherwise, the velocity in no-slip walls is zero. We will refer to a wall as *moving with speed* v if it is no-slip and the tangential component is specified to have speed v.

No-slip, Isothermal walls.

The velocity $\boldsymbol{v}_{\boldsymbol{e}_f}$ is treated the same as in the case of no-slip, adiabatic walls. Additionally, in an isothermal wall, the temperature $T_{\boldsymbol{e}_f}$ is specified. The temperature is enforced by setting the pressure to be $p_{\boldsymbol{e}_f} = \rho^{+}_{\boldsymbol{e}_f} R T_{\boldsymbol{e}_f}$. The solution is thus specified at boundaries in terms of primitive variables as

$$\boldsymbol{u}^{\text{prim}}_{\boldsymbol{e}_f} = (\rho^{+}_{\boldsymbol{e}_f}, \boldsymbol{v}_{\boldsymbol{e}_f}, p_{\boldsymbol{e}_f})$$

The viscous flux is specified using the inner values as follows.

$$(\boldsymbol{F}^{v}_{\boldsymbol{e}_f} \cdot \boldsymbol{n}) = (\boldsymbol{F}^{v+}_{\boldsymbol{e}_f} \cdot \boldsymbol{n})$$

9.5. NUMERICAL RESULTS

The numerical experiments were made with the compressible Navier Stokes equations (2.16), with most test cases taken from [144]. The error based time stepping from Section 8.6.2 is used in all experiments other than the convergence tests. The error based time stepping is applied to parabolic equations by using the local time averaged flux (8.75, 8.24) to be the total of time averaged advective and viscous fluxes (9.13). The subsequent step of truncating the time averaged flux (8.76) to obtain an embedded lower order method (8.77) remains the same. Absolute and relative tolerances (8.74) of $\tau_a = \tau_r = 10^{-8}$ are used in all experiments.

The results have been generated by extending the package `TrixiLW.jl` developed in Chapter 8 to solve parabolic equations. The setup files for the numerical experiments in this chapter are available at [15].

9.5.1. Convergence test

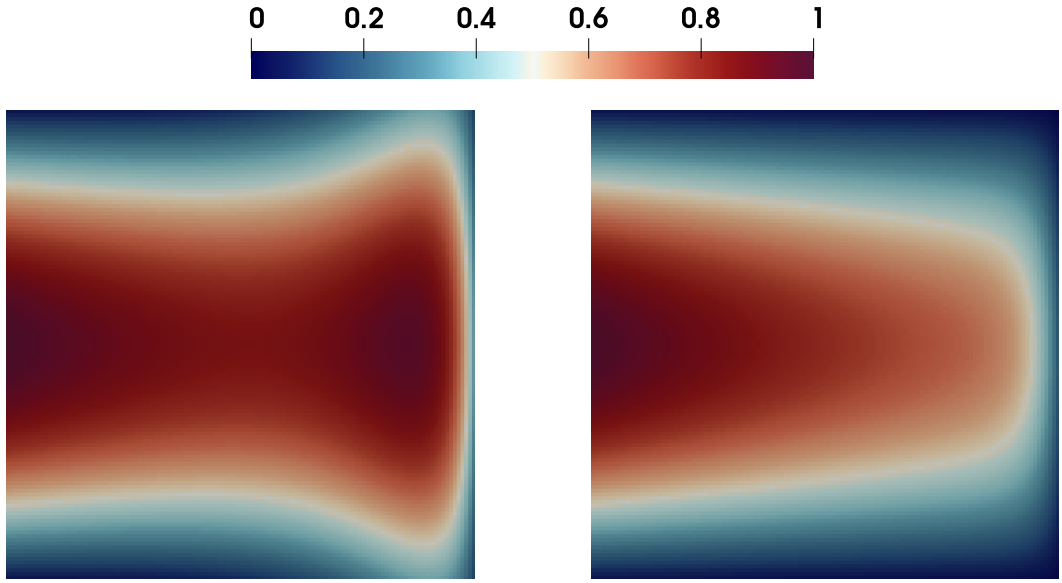
Consider the scalar advection-diffusion equation $u_t + \boldsymbol{a} \cdot \nabla u = \nu \Delta u$ where $\boldsymbol{a} = (1.5, 1)$. For the initial condition $u_0(x, y) = 1 + 0.5 \sin(\pi (x + y))$ on $[-1, 1]^2$ with periodic boundary conditions, the exact solution is given by

$$u(x, y) = 1 + 0.5\, e^{-2\nu\pi^2 t} \sin(\pi (x - a_1 t + y - a_2 t))$$

A convergence analysis with ν chosen to be in diffusion and advection dominated regimes is performed and shown in Figure 9.3, and optimal convergence rates are seen for all solution polynomial degrees. For non-periodic boundaries, we use the Eriksson-Johnson test [72] which is also a scalar advection diffusion with $\boldsymbol{a} = (1, 0)$ and $\nu = 0.05$ on domain $[-1, 0] \times [-0.5, 0.5]$ with exact solution that decays to a steady state

$$\begin{aligned} u(x, y) &= \exp(-l\, t)\,(e^{\lambda_1 x} - e^{\lambda_2 x}) + \cos(\pi\, y)\, \frac{e^{\pi x} - e^{r_1 x}}{e^{-s_1} - e^{-r_1}} \\ \lambda_1, \lambda_2 &= \frac{(-1 \pm \sqrt{1 - 4\,\nu\, l})}{-2\,\nu}, \qquad l = 4 \\ r_1, s_1 &= \frac{1 \pm \sqrt{1 + 4\,\pi^2\,\nu^2}}{2\,\nu} \end{aligned} \tag{9.25}$$

Dirichlet boundary conditions are used on left, bottom and top boundaries and outflow conditions on the right. The initial and numerical solution at $t = 1$ on a 128^2 mesh are shown in Figure 9.1. The convergence results are shown in Figure 9.4a where degree $N = 2, 4$ show optimal rates while degree $N = 3$ nears 3.54 order accuracy. The suboptimal accuracy for this test using $N = 3$ is also seen for Runge-Kutta FR/DG solvers of `Trixi.jl`. The phenomenon is similar to the nonlinear Burgers' convergence test (Section 4.7.3) where suboptimal convergence rates were seen for odd degrees, especially when Gauss-Legendre-Lobatto points were used.



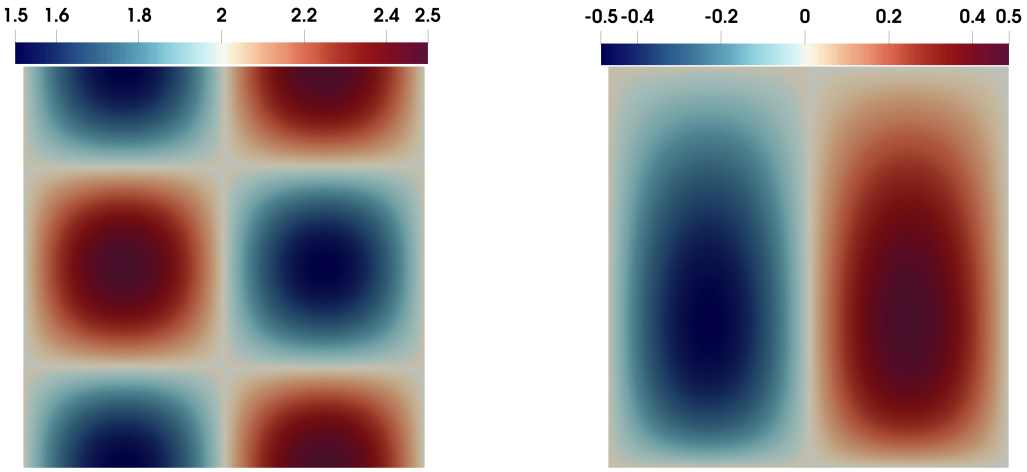
(a) (b)

Figure 9.1. Errikson-Johnson test (a) Initial condition (b) Numerical solution at $t=1$

A convergence analysis is also made for the Navier-Stokes equations on the domain $[-1,1]^2$ with a manufactured solution taken from one of the examples in `Trixi.jl` [141] given by

$$\begin{aligned} \rho &= c + A\sin(\pi x)\cos(\pi y)\cos(\pi t) \\ v_1 = v_2 &= \sin(\pi x)\log(y+2)\left(1-e^{-A(y-2)}\right)\cos(\pi t) \\ p &= \rho^2 \end{aligned} \tag{9.26}$$

The *manufacturing* of solution will lead to source terms which will be treated as in Chapter 6. The vertical boundaries are periodic and horizontal boundaries are no slip, adiabatic walls. The density and v_1 plot of numerical solution at $t=1$ are shown in Figure 9.2. The error convergence analysis for density profile is shown in Figure 9.4b, where optimal convergence rates are seen for all polynomial degrees.



(a) (b)

Figure 9.2. Numerical solution for Navier-Stokes equations with manufactured exact solution (9.26). (a) Density plot, (b) v_x plot

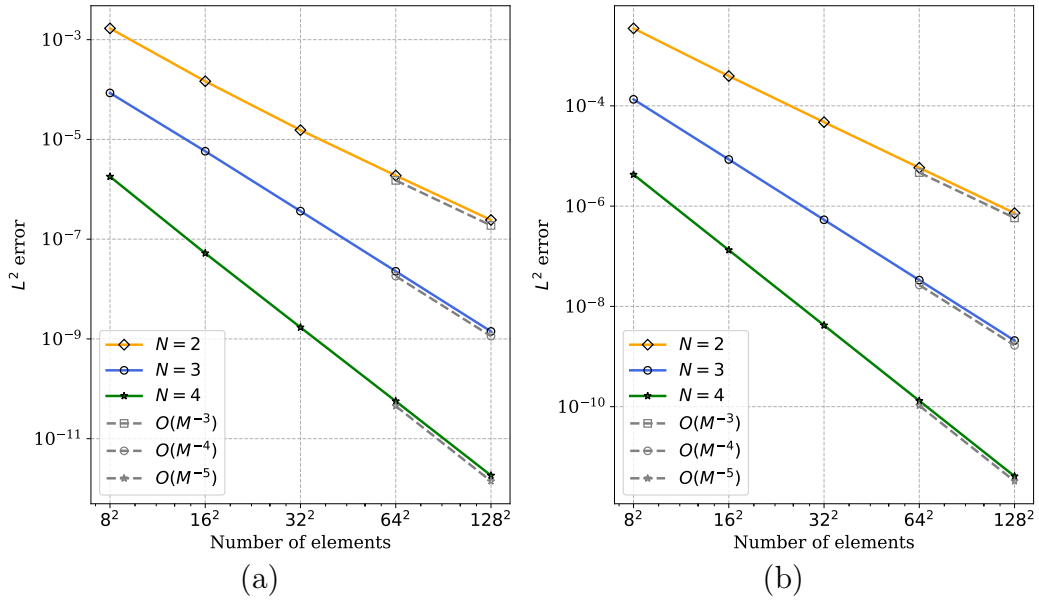


Figure 9.3. Convergence analysis for scalar advection-diffusion equation with $\boldsymbol{a} = (1.5, 1)$ and coefficient (a) $\nu = 5 \times 10^{-2}$ (b) $\nu = 10^{-12}$

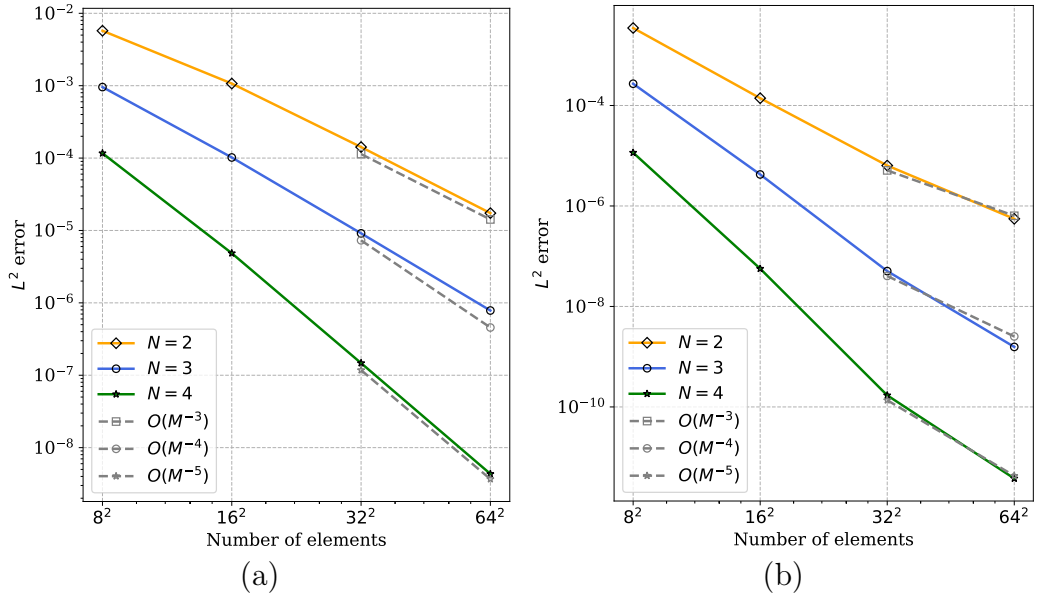


Figure 9.4. Convergence analysis with non-periodic boundary conditions. (a) Eriksson-Johnson test (Section 4 of [72]) and (b) Navier-Stokes equations with manufactured solution.

9.5.2. Lid driven cavity

This is a steady state test case for the Navier Stokes equations in the square domain $[0, 1]^2$. We take the setup from [79] where Pr=0.7, $\mu = 0.001$ and the initial condition is the solution at rest

$$(\rho, u, v, p) = \left(1, 0, 0, \frac{1}{M_\infty^2 \gamma}\right)$$

where $M_\infty = 0.1$. The top boundary is moving with a velocity of $(1, 0)$ and the other three have no-slip boundary conditions. All boundaries are adiabatic. The problem has a Reynolds number of 1000 corresponding to the top moving wall. The laminar solution of the problem is steady, with Mach number 0.1 corresponding to the moving lid. We compare the solution with the numerical data of Ghia et al. [79] by plotting the horizontal velocity profile along the vertical line through the midpoint of the domain, and vertical velocity profile along the horizontal line through the midpoint of the domain. The x-velocity plot along with velocity vectors are shown in Figure 9.5. The comparison is shown in Figure 9.6, where a good agreement with [79] is seen.

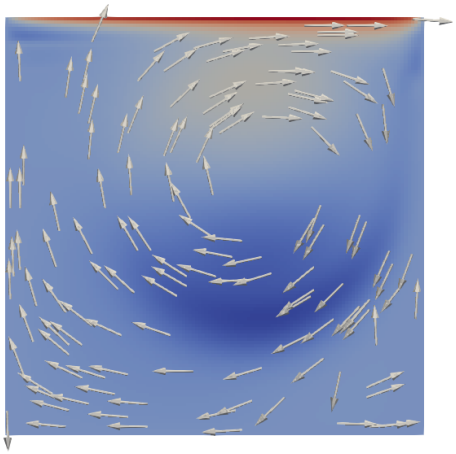


Figure 9.5. Lid driven cavity, x-velocity pseudocolor plot and velocity vectors.

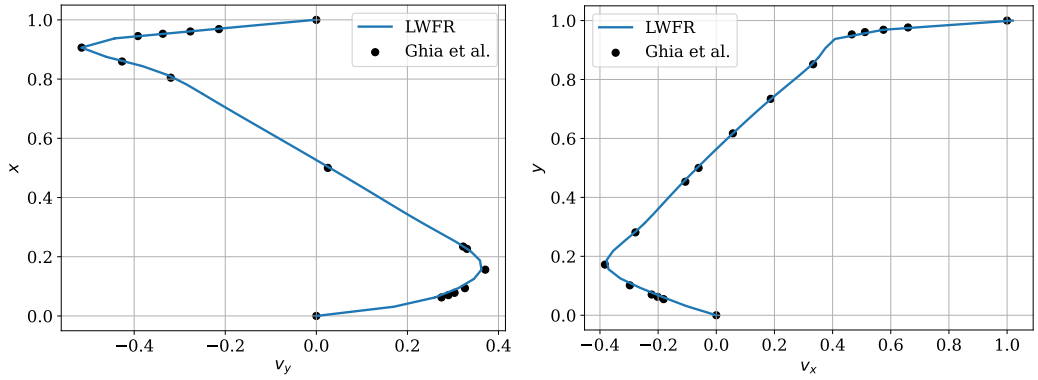


Figure 9.6. Velocity profiles of lid driven cavity test. (a) v_y cut at $y = 0.5$ (b) v_x cut at $x = 0.5$.

9.5.3. Transonic flow past NACA-0012 airfoil

This test case involves a steady flow past a symmetric NACA-0012 airfoil. We choose the free stream density and pressure as $\rho_\infty = 1, p_\infty = 2.85$ and Prandtl number $\mathrm{Pr} = 0.72$, and simulate a flow corresponding to a Reynolds number 500, free-stream Mach number

of 0.8 and 10° angle of attack. The free-stream velocity is set at $\boldsymbol{u}_\infty = (1.574, 0.277)$. The following additional quantities are considered for validation of the scheme:

- Surface pressure coefficient C_p and surface skin friction coefficient C_f along the airfoil surface

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty |\boldsymbol{v}_\infty|^2 l_\infty}, \qquad C_f = \frac{(\tau \boldsymbol{n})\,\boldsymbol{n}^\perp}{\frac{1}{2}\rho_\infty |\boldsymbol{v}_\infty|^2 l_\infty}$$

- Pressure induced lift and drag force coefficients

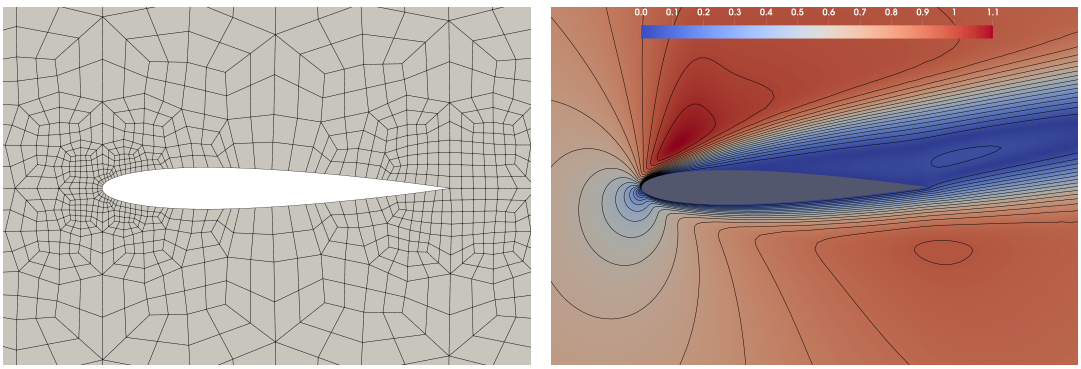
$$c_{dp} = \frac{\int_S p\,(\boldsymbol{n}\,\Psi_d)\,\mathrm{d}s}{\frac{1}{2}\rho_\infty |\boldsymbol{v}_\infty|^2 l_\infty}, \qquad c_{dp} = \frac{\int_S p\,(\boldsymbol{n}\,\Psi_l)\,\mathrm{d}s}{\frac{1}{2}\rho_\infty |\boldsymbol{v}_\infty|^2 l_\infty}$$

- Lift and drag force coefficients due to viscous forces

$$c_{dp} = \frac{\int_S (\tau \boldsymbol{n}) \cdot \Psi_d\,\mathrm{d}s}{\frac{1}{2}\rho_\infty |\boldsymbol{v}_\infty|^2 l_\infty}, \qquad c_{dp} = \frac{\int_S (\tau \boldsymbol{n}) \cdot \Psi_l\,\mathrm{d}s}{\frac{1}{2}\rho_\infty |\boldsymbol{v}_\infty|^2 l_\infty}$$

where $\boldsymbol{n}$ is the inward unit normal at domain boundary, $\boldsymbol{n}^\perp$ is the tangent at the domain boundary, $\Psi_d = (\cos\alpha, \sin\alpha)^\perp$, $\Psi_l = (-\sin\alpha, \cos\alpha)^\perp$, with α being the angle of attack.

The simulation is performed with 728 elements and polynomial degree $N = 4$. The mesh and Mach number contour plot are shown in Figure 9.7. In Figure 9.8, coefficient of pressure C_p and coefficient of friction C_f over the airfoil surface are compared with [171], showing good agreement for C_p and same for C_f everywhere other than the leading edge where there are some errors. The coefficients of pressure induced drag and lift (c_{dp}, c_{lp}), and drag and lift due to viscous forces (c_{df}, c_{lf}) are shown in Table 9.1 and a good agreement with [171] is seen.



(a) (b)

Figure 9.7. Transonic flow over a NACA-0012 airfoil with $M_\infty = 0.8$ solved on a mesh with 728 elements using solution polynomial degree $N = 4$. (a) Mesh (b) Mach number contour.

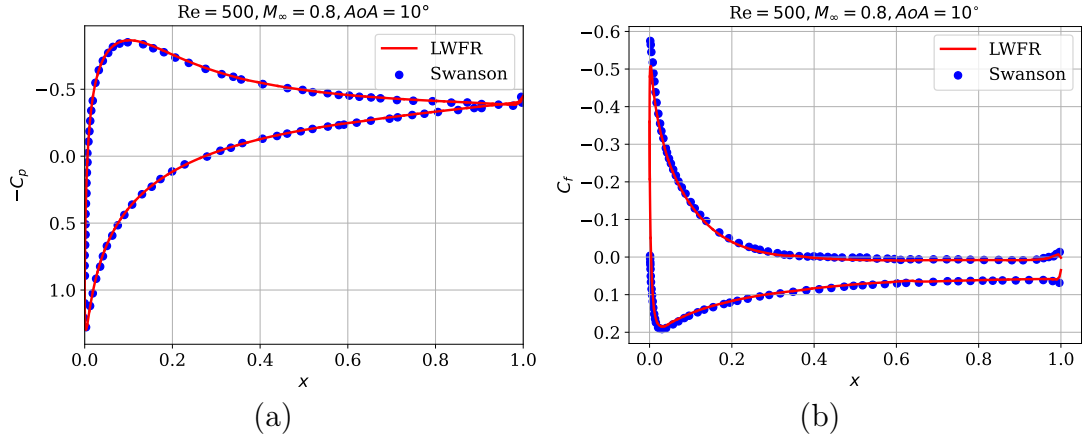


Figure 9.8. Transonic flow over an airfoil, quantities of interest on surface. (a) Coefficient of pressure and (b) Coefficient of friction

	c_{dp}	c_{df}	c_{lp}	c_{lf}	c_{lp}+c_{lf}
LWFR	0.1467	0.1242	0.4416	-0.0043	0.4373
Reference	0.1475	0.1275	–	–	0.4363

Table 9.1. Transonic flow over an airfoil compared with data from [171].

9.5.4. Flow past a cylinder

This test involves a laminar, unsteady flow past a cylinder inside a channel [156]. On the left, the inflow boundary condition is imposed with $p = \theta = 160.7143$, $v_1 = 4\,v_m\,y\,(H-y)\,/\,H^2$ where $H = 0.41$m and $v_m = 1.5$m/s is the maximum velocity, and $v_2 = 0$. The Mach number corresponding to v_m is 0.1. The v_1 velocity has a quadratic profile in y and is symmetric for $y \in [0, H]$. The cylinder is placed so that its center is at $(H/2, H/2) - (0.005, 0.005)$ so that it is slightly offset in y from the center of the channel to destabilize the otherwise steady symmetric flow (Figure 9.9).

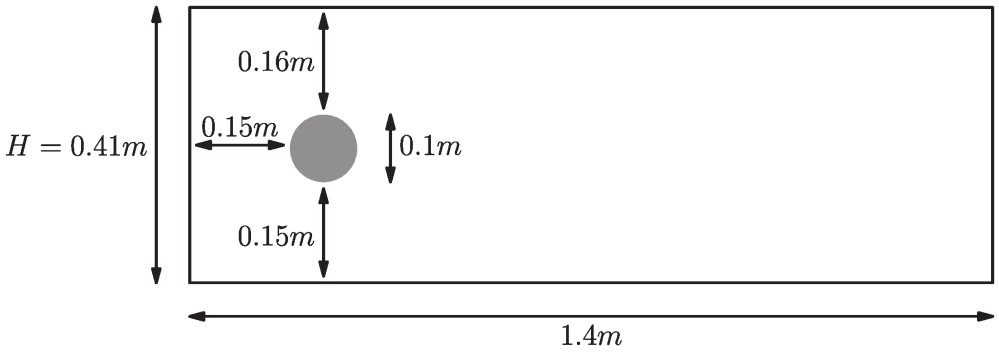


Figure 9.9. Physical domain used in Von Karman street.

Isothermal, no-slip boundary conditions are imposed on the cylinder surface, and the top and bottom boundaries. The viscosity coefficient is $\nu = 10^{-3}$ so that Reynolds number of the flow corresponding to the mean velocity is 100. The simulation is performed on a mesh with 5692 elements and polynomial degree $N = 4$ so that $\Delta x \approx 0.01$. After some time, a Von Karman vortex street appears with a periodic shedding of eddies from alternate sides of the cylinder. This is typical for slow flows past a slender

body. The vorticity profile is shown in Figure 9.10, which clearly depicts the periodic vortex shedding. The periodic behavior can also be observed in Figure 9.11 where the evolution of the coefficient of total lift $c_l = c_{lp} + c_{lf}$ and the coefficient of total drag $c_d = c_{dp} + c_{df}$ on the surface of cylinder is shown. The time period of the c_l profile is $\tau \approx 0.33759$ so that the Strouhal number is $\mathrm{St} = \mathcal{F}\mathcal{D}/\bar{u} = \mathcal{D}/(\tau\bar{u}) = 0.29621$ where $\mathcal{D} = 0.1, \bar{u} = 1$ are the diameter of cylinder and mean velocity. This value is in the reference range of [156]. The values Max c_l and Max c_l are not in the reference range but are close, as shown in Table 9.2.

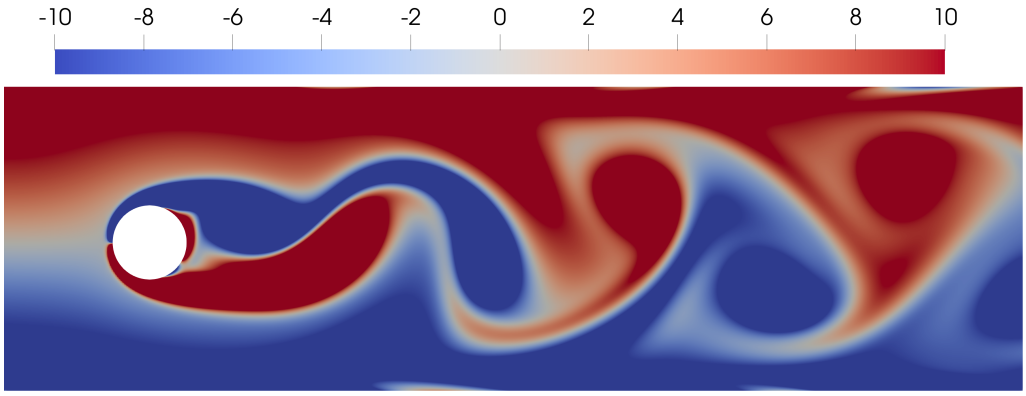
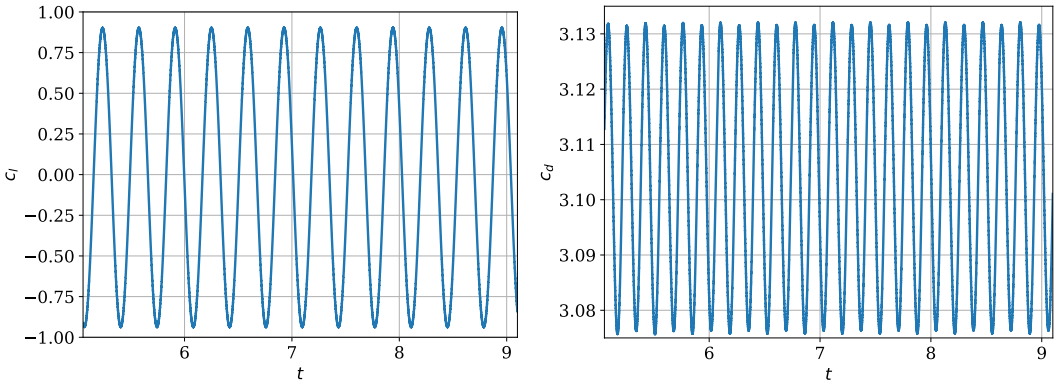


Figure 9.10. Vorticity plot of Von Karman street



(a) (b)

Figure 9.11. Flow over cylinder. (a) Coefficient of lift c_l (b) Coefficient of drag c_d.

	Max c_l	Max c_d	St
LWFR	0.906	3.136	0.29621
Reference range	$[0.99, 1.01]$	$[3.22, 3.24]$	$[0.284, 0.3]$

Table 9.2. Comparison of quantities of interest for flow past cylinder

9.6. SUMMARY

The Lax-Wendroff Flux Reconstruction (LWFR) scheme has been extended to parabolic equations along with its capability of handling curved meshes and error based time stepping proposed in Chapter 8. The scheme has been numerically validated by performing convergence and other validation tests on the compressible Navier Stokes equations.

Chapter 10

Conclusions

A conservative, Jacobian-free and single step, explicit Lax-Wendroff method has been constructed in flux reconstruction context. The Jacobian free property is achieved by using the approximate Lax-Wendroff procedure, leading to a procedure where the scheme only requires the specification of physical flux and numerical flux. Special attention was paid to construction of numerical flux; the dissipative part of the numerical flux was computed with the time averaged solution (called D2 dissipation) leading to an upwind flux in the linear case and improved CFL numbers at no additional computational cost. The scheme with D2 dissipation is proven to be equivalent to ADER schemes for linear problems. The central part of the Lax-Wendroff numerical flux was computed by performing the approximate Lax-Wendroff procedure at the faces (**EA** scheme) rather than using the extrapolated time averaged flux from solution points (**AE** scheme). It was observed that the **EA** scheme improved accuracy of the LWFR scheme and some tests showed optimal order of convergence only with the **EA** scheme. The development of various numerical fluxes like HLL, HLLC and Roe for the LWFR scheme with the improvements was made. The new scheme and its benefits were validated by performing convergence analysis, analyzing error and energy growth, and various test cases on compressible Euler's equations.

A subcell based shock capturing blending scheme was introduced for LWFR based on [90]. The idea was to construct a robust low order scheme on subcells and blend it with the high order LWFR scheme using a smoothness indicator. To enhance accuracy, we used Gauss-Legendre solution points and performed MUSCL-Hancock reconstruction on the subcells. The MUSCL-Hancock reconstruction was only possible due to the single stage nature of LWFR. Since the subcells of the blending scheme were inherently non-cell centred, the MUSCL-Hancock scheme was extended to non-cell centred grids along with the proof of [26] for admissibility preservation. The subcell structure was exploited to obtain a provably admissibility preserving LWFR scheme by careful construction of the *blended numerical flux* at the element interfaces. The procedure for enforcing admissibility only requires the user to specify what the admissibility constraints of the equation are, and the process is problem independent beyond that. The numerical experiments were made on compressible Euler's equations verifying the enhancement of accuracy and admissibility preservation. In particular, the experiments revealed that the MUSCL-Hancock blending scheme was more accurate than the first order blending scheme and that LWFR was able to simulate difficult tests for admissibility like a Mach 2000 astrophysical jet and strong Sedov's blast waves.

A generalized framework for admissibility preservation was introduced by performing a cell average decomposition followed by flux limiting, extending the positivity limiter of Zhang and Shu to Lax-Wendroff schemes. The scheme was extended to handle source terms by constructing time average source terms. Provable admissi-

bility was also obtained for the extension by introducing a limiter for the time average sources. The scheme with source terms was validated using the ten moment problem of gas dynamics with tests for accuracy and robustness.

The multiderivative Runge-Kutta (MDRK) method of Li and Du [119] was written as an evolution involving time average fluxes in both stages. This allowed us to apply approximate Lax-Wendroff procedure to each stage and obtain an MDRK scheme in Flux Reconstruction framework. The key developments made for LWFR, namely D2 dissipation, **EA** scheme and blending limiter were applied to each stage. This gave us an MDRK scheme with improved stability, robustness and provable admissibility preservation. The scheme was validated on the modern test suite of [132] for high order methods.

The LWFR scheme was extended to handle body-fitted, adaptively refined curvilinear meshes. The curvilinear grids were defined by a reference map for each element which was used to apply the Lax-Wendroff procedure in reference coordinates. The adaptively refined mesh was allowed to be nonconformal and Mortar element method for LWFR was developed to obtain a scheme that was conservative, admissibility and free stream preserving. A Fourier stability analysis does not apply to such meshes and hence an error based time stepping method was developed for LWFR. A performance comparison between error based time stepping and CFL based time was made and, even with less fine tuning, the error based time stepping gave superior performance. The extension is validated by performing tests for Euler's equations on curvilinear grids.

The scheme was also extended to second order equations in conservative form by using the BR1 scheme, along with capability to handle curvilinear meshes and error based time stepping. The extension is validated with convergence analysis and various standard tests for Navier-Stokes equations like lid driven cavity and flow over cylinder and airfoil.

10.1. FUTURE SCOPE

There are several extensions possible for this work:

1. This work is restricted to partial differential equations in conservative form. There are many practical hyperbolic problems like shear shallow water equations which contain *non-conservative products* and are usually solved with path conservative schemes that satisfy *generalized Rankine-Hugoniot* conditions. There are already Runge-Kutta Discontinuous Galerkin methods that can solve non-conservative hyperbolic equations and thus development of high order Lax-Wendroff schemes for such problems will be an important area of research. Flux reconstruction cannot be applied to such systems but we can possibly use correction functions g_L, g_R to obtain a quadrature free scheme.

2. The LWFR scheme has been developed to explicitly handle source terms while maintaining high order accuracy. This explicit treatment imposes additional conditions on time step size for stability of the scheme. Stiff source terms are those which impose a time step restriction that is more severe than the CFL restriction from the wave speeds. They occur in a variety of problems like those involving chemical reactions. The time step restriction by the source terms can

be avoided by adding local implicitness to the source terms. This is the idea of IMEX schemes. The development of such locally implicit solvers for LWFR on source terms that maintain high order accuracy will be an important area of research.

3. We made a comparison of accuracy between Gauss-Legendre (GL) and Gauss-Legendre-Lobatto (GLL) solution points which showed us the superior accuracy of the former. Another accuracy improvement that we could make was the development of a blending scheme that performs MUSCL-Hancock reconstruction on subcells. These accuracy improvements were only applied to problems on Cartesian meshes and their extension can be made to curvilinear meshes.

4. The multiderivative Runge-Kutta scheme in Flux Reconstruction form was only developed for Cartesian meshes and its extension to adaptively refined curvilinear meshes with error based time stepping should be possible.

5. The subcell based blending limiter was developed to add dissipation to inviscid problems. However, there are many advection-diffusion problems that require additional dissipation especially for underresolved flows. Thus, it is practical to develop a subcell based blending scheme for such problems. This involves development of a low order scheme on subcells that can solve advection diffusion equations.

6. The description of LWFR scheme on curvilinear grids was dimension independent and can thus be applied to 3-D. However, its numerical validation with practical problems to test accuracy, robustness and free stream conditions needs to be performed.

7. This work most generally applies to quadrilateral meshes (with curved boundaries) even though triangular meshes are also attractive due to availability of better mesh generation algorithms. Thus, the extension of LWFR to triangular and hybrid meshes will be fruitful.

8. The numerical experimentation of LWFR for other models of interest like Relativistic Hydrodynamics (RHD), Magnetohydrodynamics (MHD), Relativistic Magnetohydrodynamics (RMHD) will be worth exploring due to their practical significance and need for admissibility preserving schemes like the ones developed in this work.

Appendix A

ADER-FR and LWFR for linear problems

A.1. Introduction

In addition to high order Lax-Wendroff schemes which have been studied in this thesis, this appendix considers the family of Arbitrary high order schemes using DERivatives (ADER) initially introduced by the idea of a generalized Riemann solver [178] but later extended to Finite Volume / DG framework to obtain high order accuracy by using a predictor-corrector approach [63]. The local evolution in ADER schemes is performed by solving an element local implicit equation while the LW scheme uses an explicit Taylor's expansion. In this work, we prove, for linear problems, the equivalence of the ADER-DG scheme introduced in [63] with Lax-Wendroff FR (LWFR) using D2 dissipation numerical flux introduced in Chapter 4; the key observation used is that the space time predictor polynomial can be explicitly determined for linear problems. We remark that there are some works where both these ideas are considered as types of ADER schemes. However, in this work, we refer to ADER schemes as those that use a local implicit solver like in [63] while LW schemes as those that use a local Taylor's expansion like in Chapter 4 and [137, 18]. The rest of this appendix is organized as follows. In Section A.2, we review the ADER-DG scheme of [63] for 1-D scalar conservation laws and cast it in an FR framework for simplicity of the proof. In Section A.3, we show the equivalence of ADER-FR scheme and the LWFR scheme with D2 dissipation flux of Chapter 4 for linear problems. In Section A.4, we verify the equivalence numerically and draw conclusions in Section A.5.

A.2. ADER Discontinuous Galerkin and Flux Reconstruction

The arguments in this work apply to linear conservation laws of any dimension but for simplicity we restrict ourselves to 1-D linear scalar conservation law

$$u_t + f(u)_x = 0, \qquad f(u) = a\,u, \quad a = \text{const} \tag{A.1}$$

where u is some conserved quantity, together with some initial and boundary conditions. In this work, we consider the ADER-DG framework of [63]. We will divide the physical domain Ω into disjoint elements Ω_e, with $\Omega_e = [x_{e-\frac{1}{2}}, x_{e+\frac{1}{2}}]$ and $\Delta x_e = x_{e+\frac{1}{2}} - x_{e-\frac{1}{2}}$. The temporal discretization is performed by denoting the n^{th} time interval as $[t_n, t_{n+1}]$ and $\Delta t_n = t_{n+1} - t_n$. Let us map all spatial and temporal elements to reference elements $\Omega_e \to [0,1]$, $[t_n, t_{n+1}] \to [0,1]$ by

$$x \mapsto \xi = \frac{x - x_{e-\frac{1}{2}}}{\Delta x_e}, \qquad t \mapsto \tau = \frac{t - t_n}{\Delta t_n}$$

Thus, x, t are physical variables in space and time and ξ, τ are the respective reference variables. Inside each element, we approximate the solution as $\mathbb{P}_N$ functions which are polynomials of degree $N \geq 0$. For this, choose $N+1$ distinct nodes $\{\xi_i\}_{i=0}^N$ in $[0,1]$ which will be taken to be Gauss-Legendre (GL) or Gauss-Lobatto-Legendre (GLL) nodes, and will also be referred to as *solution points*. There are associated quadrature weights w_j such that the quadrature rule is exact for polynomials of degree up to $2N+1$ for GL points and up to degree $2N-1$ for GLL points. Note that the nodes and weights we use are with respect to the interval $[0,1]$ whereas they are usually defined for the interval $[-1,+1]$. For constructing the space-time predictor, we use the same solution points in time. The numerical solution inside an element Ω_e at $t = t^n$ is given by

$$x \in \Omega_e: \qquad u_h^n(\xi) = \sum_{p=0}^{N} u_{e,p}\, \ell_p(\xi)$$

where each ℓ_p is a Lagrange polynomial of degree N in $[0,1]$ defined to satisfy $\ell_q(\xi_p) = \delta_{pq}$ for $0 \leq p, q \leq N$.

Predictor step. The predictor inside a space-time element is given by

$$(x,t) \in \Omega_e \times [t_n, t_{n+1}]: \qquad \tilde{u}_h(\xi,\tau) = \sum_{p,q=0}^{N} \tilde{u}_{e,pq}\, \ell_p(\xi)\, \ell_q(\tau) \tag{A.2}$$

Within each element Ω_e, we take a local space-time test function ℓ_{pq}

$$\ell_{pq}(\xi,\tau) = \ell_p(\xi)\, \ell_q(\tau)$$

To compute the cell-local predictor, we multiply the conservation law (A.1) by ℓ_{pq} and do an integration by parts in time

$$\begin{aligned} -\int_{t^n}^{t^{n+1}} \int_{\Omega_e} \tilde{u}_h\, \partial_t \ell_{pq}\, \mathrm{d}x\, \mathrm{d}t + \int_{\Omega_e} \tilde{u}_h(\xi,1)\, \ell_{pq}\, \mathrm{d}x - \int_{\Omega_e} u_h^n(\xi)\, \ell_{pq}\, \mathrm{d}x \\ + \int_{t^n}^{t^{n+1}} \int_{\Omega_e} (\partial_x \tilde{f}_h)\, \ell_{pq}\, \mathrm{d}x\, \mathrm{d}t = 0 \end{aligned} \tag{A.3}$$

where $\tilde{f}_h = a\tilde{u}_h$. The above system of equations (A.3) is solved for all $\tilde{u}_{e,pq}$ (A.2).

Corrector step. Integrate (A.1) over the space-time element $\Omega_e \times [t_n, t_{n+1}]$ with the test function $\ell_p = \ell_p(\xi)$ and perform an integration by parts in space to get

$$\begin{aligned} \int_{\Omega_e} u_h^{n+1}\, \ell_p\, \mathrm{d}x \;=\; & \int_{\Omega_e} u_h^n\, \ell_p\, \mathrm{d}x + \int_{t^n}^{t^{n+1}} \int_{\Omega_e} \tilde{f}_h\, \partial_x \ell_p\, \mathrm{d}x\, \mathrm{d}t \\ & - \ell_p(1) \int_{t^n}^{t^{n+1}} f_{e+\frac{1}{2}}(\tilde{u}_h(t))\, \mathrm{d}t + \ell_p(0) \int_{t^n}^{t^{n+1}} f_{e-\frac{1}{2}}(\tilde{u}_h(t))\, \mathrm{d}t \end{aligned} \tag{A.4}$$

where, for the linear case, $f_{e+\frac{1}{2}}(\tilde{u}_h(t))$ is the upwind flux

$$f_{e+\frac{1}{2}}(\tilde{u}_h(t)) = \frac{a}{2}\left(\tilde{u}_h(x^-_{e+\frac{1}{2}}, t) + \tilde{u}_h(x^+_{e+\frac{1}{2}}, t)\right) - \frac{|a|}{2}\left(\tilde{u}_h(x^+_{e+\frac{1}{2}}, t) - \tilde{u}_h(x^-_{e+\frac{1}{2}}, t)\right) \tag{A.5}$$

The complete numerical scheme is given by space-time quadrature on (A.4) at the solution points. By linearity of the flux, quadrature on the flux term in (A.4) is exact as we use GL / GLL quadrature points and can thus perform another integration by parts in space to write

$$\begin{aligned}\int_{\Omega_e} u_h^{n+1}\,\ell_p\,\mathrm{d}x \;=\; & \int_{\Omega_e} u_h^n\,\ell_p\,\mathrm{d}x - \int_{t^n}^{t^{n+1}}\int_{\Omega_e} (\partial_x \tilde{f}_h)\,\ell_p\,\mathrm{d}x\,\mathrm{d}t \\ & -\ell_p(1)\int_{t^n}^{t^{n+1}} (f_{e+\frac{1}{2}}(\tilde{u}_h(t)) - \tilde{f}_h(1,t))\,\mathrm{d}t \\ & +\ell_p(0)\int_{t^n}^{t^{n+1}} (f_{e-\frac{1}{2}}(\tilde{u}_h(t)) - \tilde{f}_h(0,t))\,\mathrm{d}t\end{aligned}$$

Performing quadrature in space at solution points gives

$$\begin{aligned}u_p^{n+1} \;=\; & u_p^n - \left[\partial_x \int_{t^n}^{t^{n+1}} \tilde{f}_h(t)\,\mathrm{d}t\right]_p \\ & -\frac{\ell_p(1)}{\Delta x_e\, w_p}\int_{t^n}^{t^{n+1}} \Big(f_{e+\frac{1}{2}}(\tilde{u}_h(t)) - \tilde{f}_h(1,t)\Big)\,\mathrm{d}t \\ & +\frac{\ell_p(0)}{\Delta x_e\, w_p}\int_{t^n}^{t^{n+1}} \Big(f_{e-\frac{1}{2}}(\tilde{u}_h(t)) - \tilde{f}_h(0,t)\Big)\,\mathrm{d}t\end{aligned}$$

We choose correction functions $g_L, g_R \in \mathbb{P}_{N+1}$ to be g_{Radau} (3.19) if the solution points are GL points and g_2 (3.20) if solution points are GLL. Then, by the identities (Appendix B)

$$g_R'(\xi_p) = \ell_p(1)\,/\,w_p, \qquad g_L'(\xi_p) = -\ell_p(0)\,/\,w_p$$

and thus the correction step can be written in the FR form as

$$u_p^{n+1} = u_p^n - \Delta t_n\,\partial_x \tilde{F}_h(\xi_p) \tag{A.6}$$

where we define

$$\begin{aligned}\tilde{F}_h(\xi) \;=\; & \frac{1}{\Delta t_n}\int_{t^n}^{t^{n+1}} \tilde{f}_h(\xi,t)\,\mathrm{d}t \\ & +\frac{1}{\Delta t_n}\int_{t^n}^{t^{n+1}} \Big[g_R(\xi)\Big(f_{e+\frac{1}{2}}(\tilde{u}_h(t)) - \tilde{f}_h(1,t)\Big)\Big]\,\mathrm{d}t \\ & +\frac{1}{\Delta t_n}\int_{t^n}^{t^{n+1}} \Big[g_L(\xi)\Big(f_{e-\frac{1}{2}}(\tilde{u}_h(t)) - \tilde{f}_h(0,t)\Big)\Big]\,\mathrm{d}t\end{aligned} \tag{A.7}$$

which is the ADER time-averaged flux corrected by FR. The g_L, g_R satisfy $g_L(0) = g_R(1) = 1$, $g_L(1) = g_R(0) = 0$ so that

$$\tilde{F}_h(0) = \frac{1}{\Delta t_n}\int_{t^n}^{t^{n+1}} f_{e-\frac{1}{2}}(\tilde{u}_h(t))\,\mathrm{d}t, \qquad \tilde{F}_h(1) = \frac{1}{\Delta t_n}\int_{t^n}^{t^{n+1}} f_{e+\frac{1}{2}}(\tilde{u}_h(t))\,\mathrm{d}t$$

making $\tilde{F}_h$ a globally continuous flux approximation. The equations (A.3, A.6, A.7) describe the ADER-FR scheme.

A.3. Equivalence

Since $f(u) = a\,u$ in (A.1), the numerical flux function is linear and thus the corrected ADER time-averaged flux of (A.7) can be written as

$$\begin{aligned}\tilde{F}_h(\xi) &= \frac{1}{\Delta t_n}\int_{t^n}^{t^{n+1}} a\,\tilde{u}_h(\xi,t)\,\mathrm{d}t \\ &+ g_R(\xi)\left[f_{e+\frac{1}{2}}\left(\frac{1}{\Delta t_n}\int_{t^n}^{t^{n+1}}\tilde{u}_h(t)\,\mathrm{d}t\right) - \frac{1}{\Delta t_n}\int_{t^n}^{t^{n+1}} a\,\tilde{u}_h(1,t)\mathrm{d}t\right] \\ &+ g_L(\xi)\left[f_{e-\frac{1}{2}}\left(\frac{1}{\Delta t_n}\int_{t^n}^{t^{n+1}}\tilde{u}_h(t)\,\mathrm{d}t\right) - \frac{1}{\Delta t_n}\int_{t^n}^{t^{n+1}} a\,\tilde{u}_h(0,t)\mathrm{d}t\right]\end{aligned} \tag{A.8}$$

We will prove equivalence assuming that both schemes have the same solution at time $t = t^n$. Now, by (4.4), Lax-Wendroff Flux Reconstruction (LWFR) in an element can be written as

$$u_p^{n+1} = u_p^n - \Delta t_n\,\partial_x F_h(\xi_p) \tag{A.9}$$

where F_h is the continuous LW time averaged flux

$$F_h(\xi) = F_h^\delta(\xi) + g_R(\xi)\,[F_{e+\frac{1}{2}} - F_h^\delta(1)] + g_L(\xi)\,[F_{e-\frac{1}{2}} - F_h^\delta(0)] \tag{A.10}$$

and F_h^δ is the discontinuous time averaged flux computed by the approximate Lax-Wendroff procedure, described in Section 4.2.4, [208, 34], which gives the following for linear flux

$$F_h^\delta = \sum_{k=0}^{N}\frac{\Delta t^k}{(k+1)!}\,\partial_t^k f(u^n) = a\sum_{k=0}^{N}\frac{\Delta t^k}{(k+1)!}\,\partial_t^k u^n = a\sum_{k=0}^{N}\frac{(-a\,\Delta t)^k}{(k+1)!}\,\partial_x^k u_h^n =: a\,U_h^n \tag{A.11}$$

where U_h^n is the approximate time averaged solution, and all spatial derivatives are computed as local polynomial derivatives. The numerical flux with D2 dissipation introduced in (4.11) is given by

$$\begin{aligned}F_{e+\frac{1}{2}} &= \frac{1}{2}\left(F_h^\delta(x^-_{e+\frac{1}{2}}) + F_h^\delta(x^+_{e+\frac{1}{2}})\right) - \frac{|a|}{2}\left(U_h^n(x^+_{e+\frac{1}{2}}) - U_h^n(x^-_{e+\frac{1}{2}})\right) \\ &= \frac{a}{2}\left(U_h^n(x^-_{e+\frac{1}{2}}) + U_h^n(x^+_{e+\frac{1}{2}})\right) - \frac{|a|}{2}\left(U_h^n(x^+_{e+\frac{1}{2}}) - U_h^n(x^-_{e+\frac{1}{2}})\right) \\ &= f_{e+\frac{1}{2}}(U_h^n)\end{aligned} \tag{A.12}$$

where $f_{e+\frac{1}{2}}(U_h^n)$ is as defined in (A.5). Thus, the time averaged flux (A.10) in LWFR scheme (A.9) can be written as

$$F_h(\xi) = a\, U_h^n(\xi) + g_R(\xi)\,[f_{e+\frac{1}{2}}(U_h^n) - U_h^n(1)] + g_L(\xi)\,[f_{e-\frac{1}{2}}(U_h^n) - U_h^n(0)] \tag{A.13}$$

Remark A.1. The D1 dissipation numerical flux, as termed in Chapter 4, was used in earlier works like [137] and is given by

$$F_{e+\frac{1}{2}} = \frac{1}{2}\,(F_h^\delta(x^-_{e+\frac{1}{2}}) + F_h^\delta(x^+_{e+\frac{1}{2}})) - \frac{|a|}{2}\,(u_h^n(x^+_{e+\frac{1}{2}}) - u_h^n(x^-_{e+\frac{1}{2}})) \tag{A.14}$$

The D2 flux (A.12) enhances the Fourier CFL stability limit (Section 4.4). The equivalence between LW and ADER only holds with the D2 dissipation.

Looking at (A.6, A.9), to prove the claimed equivalence, we need to show that (A.8) and (A.13) are the same, which will be true if we show that the time averaged solution U_h^n defined in (A.11) is given by

$$U_h^n(\xi) = \frac{1}{\Delta t_n}\int_{t^n}^{t^{n+1}} \tilde{u}_h(\xi, t)\,\mathrm{d}t \tag{A.15}$$

For simplicity of explanation, extend the cell local polynomial $x \mapsto \sum_{p=0}^N u_p^e\,\ell_p(\xi(x))$ as a polynomial in whole of $\mathbb{R}$, now denoted u_e^n. Then, defined in physical coordinates, $(x, t) \mapsto u_e^n\,(x - a\,(t - t_n))$ is a degree N space-time polynomial which satisfies the predictor equation (A.3) for $f(u) = a\,u$. Since the predictor equation has a unique solution [68, 95], the solution of (A.3) is indeed given in physical coordinates as

$$\tilde{u}_h(x,t) = u_e^n\,(x - a\,(t - t^n)), \qquad x \in \Omega_e \tag{A.16}$$

Thus, we have $\partial_t \tilde{u}_h = -a\,\partial_x \tilde{u}_h$ and $\tilde{u}_h|_{t=t_n, x\in\Omega_e} = u_e^n = u_h^n$ which we will now exploit to obtain (A.15). Since $\tilde{u}_h$ is a degree N polynomial, its Taylor's expansion gives

$$\begin{aligned}
\tilde{u}_h(\xi,t) &= \sum_{k=0}^N \frac{(t-t^n)^k}{k!}\,\partial_t^k\,\tilde{u}_h(\xi,t^n) \\
&= \sum_{k=0}^N \frac{(-a\,(t-t^n))^k}{k!}\,\partial_x^k\,\tilde{u}_h(\xi,t^n) \\
&= \sum_{k=0}^N \frac{(-a\,(t-t^n))^k}{k!}\,\partial_x^k\,u_h^n \qquad (A.16) \\
\Longrightarrow \frac{1}{\Delta t_n}\int_{t^n}^{t^{n+1}} \tilde{u}_h(\xi,t)\,\mathrm{d}t &= \sum_{k=0}^N \frac{(-a\,\Delta t)^k}{(k+1)!}\,\partial_x^k\,u_h^n = U_h^n \qquad (A.11)
\end{aligned}$$

Thus, we have obtained (A.15) proving equivalence of the two schemes.

Remark A.2. The above steps are not valid for a non-linear flux because the identity $\tilde{u}_t = -\tilde{f}_x$ need not hold at $t = t^n$.

A.4. Numerical validation

The ADER-FR scheme described in Section A.2 is implemented, tested and validated for general scalar conservation laws like Burgers' equations with smooth solutions. For numerical validation of equivalence, the Lax-Wendroff scheme with D1, D2 (A.14, A.12) dissipation (called LW-D1, LW-D2) and ADER scheme are tested for scalar linear advection equation (A.1) with $a = 5$ and wave packet initial condition $u(x, 0) = e^{-10x^2} \sin(10\pi x)$ on domain $[-1, 1]$ with periodic and Dirichlet boundary conditions for degrees $N = 1, 2, 3$. The non-periodic boundaries for LWFR are treated as in Section 4.5. The Radau correction functions [94] and Gauss-Legendre solution points are used in the results shown, although we have also tested other correction functions and solution points where same behavior was seen. Each scheme uses the same time step size, and is within the stability limit. The LW-D2 (A.12) and ADER schemes are found to match to $O(10^{-14})$ in L^∞ norm, verifying equivalence. In Figure A.1, we show the L^2 error $\|u_h - u_{\text{exact}}\|_2$ versus time plot for LW-D1, LW-D2 (A.14, A.12) and the ADER scheme for periodic (Figure A.1a, b, c) and non-periodic (Figure A.1d, e, f) boundaries. Since the ADER and LW-D2 schemes are equivalent, we see their L^2 error curves overlap, while for D1 dissipation, we see differences of up to $O(10^{-3})$ for both periodic and non-periodic boundaries. Thus, equivalence holds precisely with the D2 dissipation. The code used to generate these results is available online at [7].

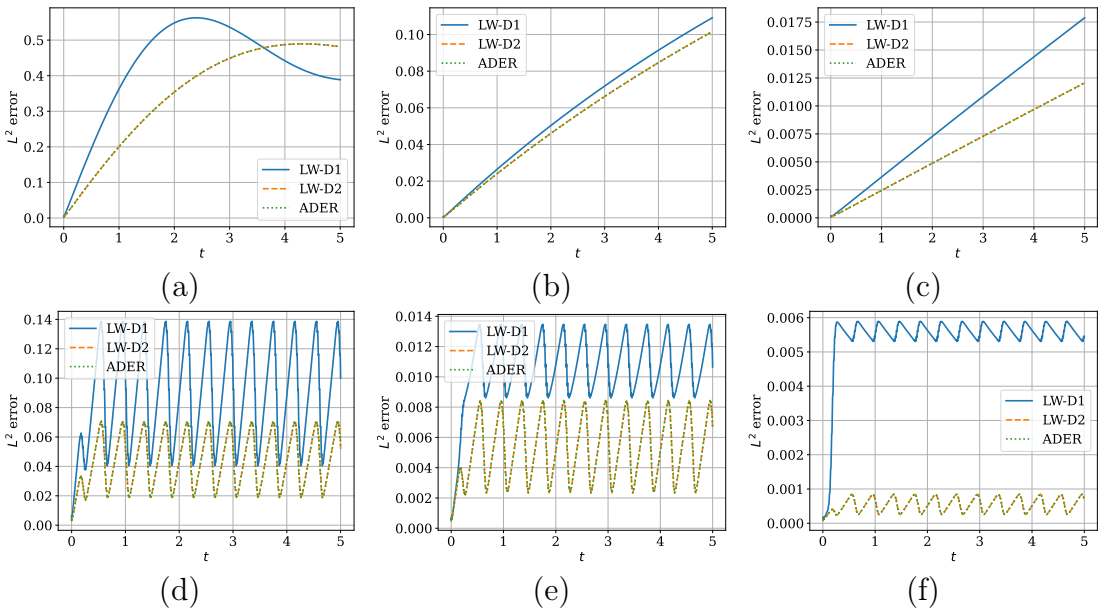


Figure A.1. L^2 error $\|u_h - u_{\text{exact}}\|_2$ versus time for wave packet test for different polynomial degrees with 240 degrees of freedom. Periodic : (a) $N = 1$, (b) $N = 2$, (c) $N = 3$. Non-periodic : (d) $N = 1$, (e) $N = 2$, (f) $N = 3$

A.5. Conclusions

This appendix proves linear equivalence of high order ADER and Lax-Wendroff (LW) schemes in Discontinuous Galerkin / Flux Reconstruction framework when the numerical flux in LW is computed using the D2 dissipation introduced in Chapter 4. This is consistent with the Fourier stability analysis performed in Section 4.4 where it was observed that the CFL numbers of LWFR scheme with D2 dissipation are the same as those of ADER-DG schemes obtained in [63, 76]. The equivalence was also numerically validated for a wave packet test. The crucial observation needed for the proof is that the solution of the predictor equation of ADER scheme has the same expression as exact solution of the linear problem. Thus, this work relates two single stage methods which are based on very different ideas and is thus a contribution to our understanding of these numerical schemes. A natural but important research question for further comparison of these two schemes is whether it can be proven that they agree up to optimal order of accuracy for smooth solutions, which is numerically observed.

Appendix B

Equivalence of DG and FR

As proven in [94, 127], the Discontinuous Galerkin (DG) method can be cast in a Flux Reconstruction (FR) framework when Gauss-Legendre or Gauss-Legendre-Lobatto points are used as solution and quadrature points. The same is proven here for the general case of curvilinear grids in part to justify the FR formulation in Section 8.3.1. The proof is provided here for Runge-Kutta Flux Reconstruction for simplicity although the same arguments apply for Lax-Wendroff Flux Reconstruction. That is, following the ideas in this appendix and Section 8.3.2, a Lax-Wendroff Discontinuous Galerkin method on curvilinear grids can be defined which will be equivalent to the Lax-Wendroff Flux Reconstruction method of Chapter 8.

B.1. Discontinuous Galerkin on curvilinear grids

Consider the degree N Lagrange polynomial basis $\{\ell_{\boldsymbol{p}}\}$ on the reference cell $\Omega_o = [-1, 1]^d$ (8.6). Let $\boldsymbol{u}^\delta$, $\tilde{\boldsymbol{f}}^\delta$ be the degree N approximate solution and contravariant flux, defined in (8.15, 8.17) respectively. The DG scheme can either be formulated for the transformed PDE (8.12) or weak formulation can be constructed for the conservation law in the physical space (3.1) and transformed to the reference cell. It is easy to see that the two are equivalent, and we will only show the DG scheme for the transformed PDE (8.12).

We will show that both can be formulated in a way that the obtained schemes are equivalent. We first derive the DG scheme for the transformed conservation law. The first step is to multiply the transformed conservation law (8.12) with a test function φ which is a degree N polynomial in reference space

$$\int_{\Omega_o} J \frac{\partial \boldsymbol{u}_e^\delta}{\partial t} \varphi \,\mathrm{d}\boldsymbol{\xi} + \int_{\Omega_o} \varphi \,\nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}_e^\delta \,\mathrm{d}\boldsymbol{\xi} = \boldsymbol{0}$$

Performing a formal integration by parts to derive the DG scheme gives

$$\begin{aligned} &\int_{\Omega_o} J \frac{\partial \boldsymbol{u}_e^\delta}{\partial t} \varphi(\boldsymbol{\xi}) \,\mathrm{d}\boldsymbol{\xi} - \int_{\Omega_o} \tilde{\boldsymbol{f}}_e^\delta \cdot (\nabla_{\boldsymbol{\xi}} \varphi) \,\mathrm{d}\boldsymbol{\xi} \\ &\quad + \sum_{i=1}^{d} \left[\int_{\partial\Omega_{o,i}^L} (\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{L,i})^* \,\ell_{\boldsymbol{p}} \,\mathrm{d}S_{\boldsymbol{\xi}} + \int_{\partial\Omega_{o,i}^R} (\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{R,i})^* \,\ell_{\boldsymbol{p}} \,\mathrm{d}S_{\boldsymbol{\xi}} \right] = \boldsymbol{0} \end{aligned} \tag{B.1}$$

where i denotes the coordinate direction and s denotes the side $\{L, R\}$, $\partial\Omega_{o,i}^R$ denotes the face where reference outward normal is $\hat{\boldsymbol{n}}_{R,i} = \boldsymbol{e}_i$ and $\partial\Omega_{o,i}^L$ has outward unit normal $\hat{\boldsymbol{n}}_{L,i} = -\hat{\boldsymbol{n}}_{R,i}$. The face of element e in a direction i on the side s will be referred to as the face (s, i) and $(\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{s,i})^*$ denotes the numerical flux. The numerical flux is usually taken to be Rusanov's flux [152] in this work which we discussed in (8.19).

B.2. Equivalence with Flux Reconstruction

We derive the collocation based Flux Reconstruction [94] scheme directly from the DG scheme. For the multi-indices $\boldsymbol{p} = (p_i)_{i=1}^d$ where $p_i \in \{0, 1 \ldots, N\}$, take the test function to be

$$\ell_{\boldsymbol{p}}(\boldsymbol{\xi}) = \prod_{i=1}^{d} \ell_{p_i}(\xi^i)$$

so that the DG scheme (B.1) becomes

$$\begin{aligned} &\int_{\Omega_o} J \frac{\partial \boldsymbol{u}_e^\delta}{\partial t} \ell_{\boldsymbol{p}} \, \mathrm{d}\boldsymbol{\xi} - \int_{\Omega_o} \tilde{\boldsymbol{f}}_e^\delta \cdot (\nabla_{\boldsymbol{\xi}} \ell_{\boldsymbol{p}}) \, \mathrm{d}\boldsymbol{\xi} \\ &\quad + \sum_{i=1}^{d} \left[\int_{\partial\Omega_{o,i}^L} (\tilde{\boldsymbol{f}}_e^\delta \cdot \hat{\boldsymbol{n}}_{L,i})^* \, \ell_{\boldsymbol{p}} \, \mathrm{d}S_{\boldsymbol{\xi}} + \int_{\partial\Omega_{o,i}^R} (\tilde{\boldsymbol{f}}_e^\delta \cdot \hat{\boldsymbol{n}}_{R,i})^* \, \ell_{\boldsymbol{p}} \, \mathrm{d}S_{\boldsymbol{\xi}} \right] = \boldsymbol{0} \end{aligned} \tag{B.2}$$

The scheme in (B.2) requires quadrature to be implemented; for equivalence with Flux Reconstruction, quadrature points are taken to be the same as solution points. Integration by parts can be performed if the volume integral with flux is exact. This will be true if we use Gauss-Legendre (GL) quadrature points (integrals will be exact) or Gauss-Legendre-Lobatto (GLL) quadrature points (integrals will be exact along the direction of the derivative, also used in [108]). Thus, (B.2) is equivalent to the *strong form* [108]

$$\begin{aligned} &\int_{\Omega_o} J \frac{\partial \boldsymbol{u}_e^\delta}{\partial t} \ell_{\boldsymbol{p}} \, \mathrm{d}\boldsymbol{\xi} + \int_{\Omega_o} (\nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}_e^\delta) \, \ell_{\boldsymbol{p}} \, \mathrm{d}\boldsymbol{\xi} \\ &\quad + \sum_{i=1}^{d} \left[\int_{\partial\Omega_{o,i}^R} ((\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{R,i})^* - \tilde{\boldsymbol{f}}_e^\delta \cdot \hat{\boldsymbol{n}}_{R,i}) \, \ell_{\boldsymbol{p}} \, \mathrm{d}S_{\boldsymbol{\xi}} + \int_{\partial\Omega_{o,i}^L} ((\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{L,i})^* - \tilde{\boldsymbol{f}}_e^\delta \cdot \hat{\boldsymbol{n}}_{L,i}) \, \ell_{\boldsymbol{p}} \, \mathrm{d}S_{\boldsymbol{\xi}} \right] = \boldsymbol{0} \end{aligned}$$

Recall that the solution points are given by $\{\boldsymbol{\xi}_{\boldsymbol{p}} = (\xi_{p_i})_{i=1}^d, p_i = 0, \ldots, N\}$. For a fixed $\boldsymbol{p}$, we denote the product of quadrature weights in each coordinate direction as $\boldsymbol{w}_{\boldsymbol{p}} := \prod_{i=1}^d w_{p_i}$ and the solution point with index suppressed as $\boldsymbol{\xi} := \boldsymbol{\xi}_{\boldsymbol{p}}$. Then, as in Chapter 8, we denote $\boldsymbol{\xi}_i^S$ (Figure 8.1) as projection of $\boldsymbol{\xi}$ to the face $S = L, R$ in the i^{th} direction (8.16). Then, performing quadrature at solution points will give us the following collocation scheme at the fixed $\boldsymbol{\xi} = \boldsymbol{\xi}_{\boldsymbol{p}}$

$$\begin{aligned} &J \frac{\mathrm{d}\boldsymbol{u}_{e,\boldsymbol{p}}^\delta}{\mathrm{d}t} \boldsymbol{w}_{\boldsymbol{p}} + \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}_e^\delta(\boldsymbol{\xi}_{\boldsymbol{p}}) \, \boldsymbol{w}_{\boldsymbol{p}} \\ &\quad + \frac{\boldsymbol{w}_{\boldsymbol{p}}}{w_{p_i}} \sum_{i=1}^{d} ((\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{R,i})^* - \tilde{\boldsymbol{f}}_e^\delta \cdot \hat{\boldsymbol{n}}_{R,i})(\boldsymbol{\xi}_i^R) \, \ell_{p_i}(1) + ((\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{L,i})^* - \tilde{\boldsymbol{f}}_e^\delta \cdot \hat{\boldsymbol{n}}_{L,i})(\boldsymbol{\xi}_i^L) \, \ell_{p_i}(-1) = \boldsymbol{0} \end{aligned}$$

where $(\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{s,i})^*(\boldsymbol{\xi}_i^s)$ and $\tilde{\boldsymbol{f}}_e^\delta \cdot \hat{\boldsymbol{n}}_{s,i}(\xi_i^s)$ denote numerical flux and physical flux at interface solution point $\boldsymbol{\xi}_i^s$. Dividing by $J\boldsymbol{w_p}$ gives

$$\begin{aligned}&\frac{\mathrm{d}\boldsymbol{u}_{e,\boldsymbol{p}}^\delta}{\mathrm{d}t} + \frac{1}{J}\nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}_e^\delta(\boldsymbol{\xi_p}) \\ &\quad + \frac{1}{J}\sum_{i=1}^{d} ((\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{R,i})^* - \tilde{\boldsymbol{f}}_e^\delta \cdot \hat{\boldsymbol{n}}_{R,i})(\boldsymbol{\xi}_i^R)\,\frac{\ell_{p_i}(1)}{w_{p_i}} - ((\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{L,i})^* - \tilde{\boldsymbol{f}}_e^\delta \cdot \hat{\boldsymbol{n}}_{L,i})(\boldsymbol{\xi}_i^L)\,\frac{\ell_{p_i}(-1)}{w_{p_i}} = \boldsymbol{0}\end{aligned} \tag{B.3}$$

The equivalence of FR and DG for choices of different solution points and correction functions has been studied in [94, 127]. We use the following identities whose proofs are based on properties of special polynomials observed in [94] (see Appendix B.2.1) which generalize the proofs of equivalence in [94, 127]

$$\begin{aligned}&\frac{\ell_{p_i}(-1)}{w_{p_i}}, \frac{\ell_{p_i}(1)}{w_{p_i}} \\ &= \begin{cases} -g'_{\mathrm{Radau},L}(\xi_{p_i}),\, g'_{\mathrm{Radau},R}(\xi_{p_i}), & \text{GL solution points and quadrature} \\ -g'_{2,L}(\xi_{p_i}),\, g'_{2,R}(\xi_{p_i}), & \text{GLL solution points and quadrature} \end{cases}\end{aligned} \tag{B.4}$$

The g_{Radau}, g_2 are FR correction functions introduced in [94] and their explicit expressions are (B.9, B.16)[B.1]. By (B.4), we can choose the corrector functions g_L, g_R corresponding to the solution points so that (B.3) can be written as

$$\begin{aligned}&\frac{\mathrm{d}\boldsymbol{u}_{e,\boldsymbol{p}}^\delta}{\mathrm{d}t} + \frac{1}{J}\nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}_e^\delta(\boldsymbol{\xi_p}) \\ &\quad + \frac{1}{J}\sum_{i=1}^{d} ((\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{R,i})^* - \tilde{\boldsymbol{f}}_e^\delta \cdot \hat{\boldsymbol{n}}_{R,i})(\boldsymbol{\xi}_i^R)\, g'_R(\xi_{p_i}) + ((\tilde{\boldsymbol{f}}_e \cdot \hat{\boldsymbol{n}}_{L,i})^* - \tilde{\boldsymbol{f}}_e^\delta \cdot \hat{\boldsymbol{n}}_{L,i})(\boldsymbol{\xi}_i^L)\, g'_L(\xi_{p_i}) = \boldsymbol{0}\end{aligned} \tag{B.5}$$

This is the same explicit form of FR as in (8.22), proving the equivalence between FR and DG schemes.

B.2.1. Corrector function identites

In this section, we prove the following for $0 \le p \le N$

$$\begin{aligned}&\frac{\ell_p(-1)}{w_p}, \frac{\ell_p(1)}{w_p} \\ &= \begin{cases} -g'_{\mathrm{Radau},L}(\xi_p),\, g'_{\mathrm{Radau},R}(\xi_p), & (\xi_p, w_p) \text{ are GL solution, quadrature points} \\ -g'_{2,L}(\xi_p),\, g'_{2,R}(\xi_p), & (\xi_p, w_p) \text{ are GLL solution, quadrature points} \end{cases}\end{aligned} \tag{B.6}$$

B.1. The g_{Radau} and g_2 correction function expressions of (B.9, B.16) are defined for the reference interval $[-1, 1]$ and are thus different from those in (3.19, 3.20) defined for reference interval $[0, 1]$.

We first prove it for Gauss-Legendre solution points, with the Radau correction function. Since degree N Gauss-Legendre solution points are the $N+1$ zeros of the degree $N+1$ Legendre polynomial L_{N+1} where we make the normalization choice $L_{N+1}(1)=1$, the Lagrange polynomials corresponding to Gauss-Legendre points are given by

$$\ell_j(\xi)=\frac{L_{N+1}(\xi)}{(\xi-\xi_j)\,L'_{N+1}(\xi_j)}, \qquad 0\le j\le N \tag{B.7}$$

The quadrature weights are

$$w_j=\frac{2}{(1-\xi_j^2)\,[L'_{N+1}(\xi_j)]^2}, \qquad 0\le j\le N \tag{B.8}$$

The Radau correction functions are

$$g_L(\xi)=R_{N+1,R}(\xi), \qquad g_R(\xi)=g_L(-\xi)=R_{N+1,L}(\xi) \tag{B.9}$$

where $R_{N+1,R}$ is the right Radau polynomial characterized as the polynomial perpendicular to $\mathbb{P}_{N-1}$ and satisfying $R_{N+1,R}(-1)=1$, $R_{N+1,R}(1)=0$. The right, left Radau polynomials are explicitly given by

$$R_{N+1,R}=\frac{(-1)^{N+1}}{2}\,(L_{N+1}-L_N), \qquad R_{N+1,L}=R_{N+1,R}(-\xi)=\frac{1}{2}\,(L_N+L_{N+1}) \tag{B.10}$$

We will also be using the identities (8.5.7) of Hildebrand [91]

$$(1-\xi^2)\,L'_N(\xi) \;=\; -N\xi\,L_N(\xi)+N\,L_{N-1}(\xi) \tag{B.11}$$

$$(1-\xi^2)\,L'_N(\xi) \;=\; (N+1)\,\xi\,L_N(\xi)-(N+1)\,L_{N+1}(\xi) \tag{B.12}$$

Now, using $L_{N+1}(-1)=(-1)^{N+1}$, we get from (B.7, B.8)

$$-\frac{\ell_j(-1)}{w_j}=\frac{1}{2}\,(-1)^N\,(\xi_j-1)\,L'_{N+1}(\xi_j) \tag{B.13}$$

Then, using (B.10, B.11) gives

$$L'_N(\xi_j)=\frac{(N+1)\,\xi_j\,L_N(\xi_j)}{1-\xi_j^2}, \qquad L'_{N+1}(\xi_j)=\frac{(N+1)\,L_N(\xi_j)}{1-\xi_j^2} \tag{B.14}$$

and thus, using (B.13, B.14), Radau correction function (B.9) satisfies

$$g'_L(\xi_j)+\frac{\ell_j(-1)}{w_j}=\frac{(-1)^N}{2}\,(L'_N(\xi_j)-\xi_j\,L'_{N+1}(\xi_j))=0,$$

and we get the claim (B.6) for Radau correction functions. We now prove the claim (B.6) for g_2 correction functions. Since GLL points include ±1, the Lagrange polynomials with GLL points satisfy

$$\frac{\ell_p(-1)}{w_p}=\frac{\delta_{p0}}{w_p}, \qquad \frac{\ell_p(1)}{w_p}=\frac{\delta_{pN}}{w_p}$$

where δ_{kl} is the Dirac delta function. The quadrature weights corresponding to GLL points are given by

$$w_p=\begin{cases}\dfrac{2}{N\,(N+1)}\,\dfrac{1}{[L_N(\xi_p)]^2} & \text{if} \quad 0<p<N\\[2ex] \dfrac{2}{N\,(N+1)} & \text{if} \quad p=0 \;\text{ or }\; p=N\end{cases} \tag{B.15}$$

The g_2 correction functions are given by [94]

$$g_{2,L} = \frac{N}{2N+1} R_{R,N+1} + \frac{N+1}{2N+1} R_{R,N} \tag{B.16}$$

where R_N are Radau polynomials (B.10). In Appendix E of [94], it is proven that $g_{2,L}$ has extremums at all Lobatto points other than the left boundary, where it satisfies by (B.15)

$$g'_{2,L}(-1) = -\frac{1}{2} N (N+1) = -\frac{\ell_0(-1)}{w_0}$$

giving our claim (B.6).

Appendix C

Equivalence with DFR

The direct flux reconstruction method does not require the choice of correction function. Following the ideas of [147], we will prove that the LWFR scheme using Gauss-Legendre points and Radau correction function described in Section 4.2.2 is equivalent to the LWDFR scheme described in Section 4.2.3, by showing that the $\mathsf{b}_L, \mathsf{b}_R, \mathsf{D}_1$ are same for both.

Equivalence of b_L We begin by proving the claim for b_L. For the FR scheme, we have

$$\mathsf{b}_L^{\mathrm{FR}} = \begin{bmatrix} g_L'(\xi_0) \\ \vdots \\ g_L'(\xi_N) \end{bmatrix}$$

where g_L is the Radau correction function and $\{\xi_p, 0 \leq p \leq N\}$ are Gauss-Legendre quadrature points on the interval $[0,1]$. For the DFR scheme, we have

$$\mathsf{b}_L^{\mathrm{FR}} = \begin{bmatrix} \tilde{\ell}_{-1}'(\xi_0) \\ \vdots \\ \tilde{\ell}_{-1}'(\xi_N) \end{bmatrix}$$

where $\tilde{\ell}_p$'s are Lagrange polynomials associated to the points $\{\xi_p, -1 \leq p \leq N+1\}$ where $\xi_{-1} = 0$ and $\xi_{N+1} = 1$. Since the $N+1$ zeros of L_{N+1} are also zeros of $\tilde{\ell}_{-1}$ and $\tilde{\ell}_{-1}(0) = 1$, $\tilde{\ell}_{-1}(1) = 0$, we must have

$$\tilde{\ell}_{-1}(\xi) = (-1)^N (\xi - 1) L_{N+1}(2\xi - 1)$$

To prove our claim, we need to prove

$$\frac{\mathrm{d}}{\mathrm{d}\xi}(g_L - \tilde{\ell}_{-1})(\xi_p) = 0, \qquad p = 0, 1, \ldots, N$$

i.e.,

$$L_N'(2\xi_p - 1) - L_{N+1}(2\xi_p - 1) - (2\xi_p - 1) L_{N+1}'(2\xi_p - 1) = 0, \qquad p = 0, 1, \ldots, N$$

To work in $[-1, 1]$ which is the natural domain of Legendre polynomials, we define the residual $R(\eta) = L_N'(\eta) - L_{N+1}(\eta) - \eta L_{N+1}'(\eta)$ so we have to show

$$R(\eta_p) = 0, \qquad p = 0, 1, \ldots, N$$

where $\eta_p = 2\xi_p - 1$ are the Gauss-Legendre points in $[-1, +1]$. Using the recurrence relations

$$\begin{aligned} (1 - \eta^2) L_{N+1}'(\eta) &= (N+1)[L_N(\eta) - \eta L_{N+1}(\eta)] \\ L_N'(\eta) &= (N+1)[\eta L_N(\eta) - L_{N+1}(\eta)] \end{aligned}$$

we get

$$R(\eta) = -(N+2)\, L_{N+1}(\eta)$$

proving that $R(\eta_p)=0$ for all $p=0,1,\ldots,N$ since these η_p are the zeros of L_{N+1}. Thus, $\mathsf{b}_L^{\mathrm{FR}}=\mathsf{b}_L^{\mathrm{DFR}}$. The claim for right correction follows analogously.

Equivalence of $\mathbf{D_1}$. Writing $\mathsf{b}_L=\mathsf{b}_L^{\mathrm{FR}}=\mathsf{b}_L^{\mathrm{DFR}}$ and $\mathsf{b}_R=\mathsf{b}_R^{\mathrm{FR}}=\mathsf{b}_R^{\mathrm{DFR}}$, proving that the D_1 matrices are same for both schemes is equivalent to showing that

$$\mathsf{D}=\mathsf{D}_1^{\mathrm{DFR}}+\mathsf{b}_L\,\mathsf{V}_L^\top+\mathsf{b}_R\,\mathsf{V}_R^\top$$

where D is the differentiation matrix on Gauss-Legendre points. Further, to show that these two matrices are equal, it is enough to prove that their action on a set of $N+1$ linearly independent column vectors is the same. For this, we consider an arbitrary polynomial $p(\xi)$ of degree less than or equal to N, and let $\mathsf{p}=[p(\xi_0),\cdots,p(\xi_N)]^\top$ and $\mathsf{p}'=[p'(\xi_0),\cdots,p'(\xi_N)]^\top=\mathsf{D}\mathsf{p}$. We have

$$\mathsf{b}_L\mathsf{V}_L^\top\mathsf{p}=\mathsf{b}_L\sum_{p=0}^{N} p(\xi_p)\,\ell_p(0)=\mathsf{b}_L p(0)=p(0)[\tilde{\ell}'_{-1}(\xi_0),\cdots,\tilde{\ell}'_{-1}(\xi_N)]^\top$$

and

$$\mathsf{b}_R\mathsf{V}_R^\top\mathsf{p}=\mathsf{b}_R\sum_{p=0}^{N} p(\xi_p)\,\ell_p(1)=\mathsf{b}_R p(1)=p(1)[\tilde{\ell}'_{N+1}(\xi_0),\cdots,\tilde{\ell}'_{N+1}(\xi_N)]^\top$$

As p is a polynomial of degree less than or equal to N, we can write

$$p(\xi)=\sum_{p=-1}^{N+1} p(\xi_p)\,\tilde{\ell}_p(\xi),\qquad p'(\xi)=\sum_{p=-1}^{N+1} p(\xi_p)\,\tilde{\ell}'_p(\xi)$$

We get

$$\begin{aligned}
&(\mathsf{D}_1^{\mathrm{DFR}}+\mathsf{b}_L\mathsf{V}_L^\top+\mathsf{b}_R\mathsf{V}_R^\top)\mathsf{p}\\
&=\begin{bmatrix} \sum_{q=0}^{N} p(\xi_0)\,\tilde{\ell}'_q(\xi_0) \\ \vdots \\ \sum_{q=0}^{N} p(\xi_N)\,\tilde{\ell}'_q(\xi_N) \end{bmatrix}+\begin{bmatrix} p(0)\,\tilde{\ell}'_{-1}(\xi_0) \\ \vdots \\ p(0)\,\tilde{\ell}'_{-1}(\xi_N) \end{bmatrix}+\begin{bmatrix} p(1)\,\tilde{\ell}'_{N+1}(\xi_0) \\ \vdots \\ p(1)\,\tilde{\ell}'_{N+1}(\xi_N) \end{bmatrix}\\
&=\mathsf{p}'=\mathsf{D}\mathsf{p}
\end{aligned}$$

for all $\mathsf{p}\in\mathbb{R}^{N+1}$, which proves the claim.

Appendix D

Some numerical fluxes

We describe the procedure to compute the numerical flux for systems at one single face $e+\frac{1}{2}$. The numerical flux for LWFR is computed using the trace values of the solution $\boldsymbol{U}_l = \boldsymbol{U}^-_{e+\frac{1}{2}}$, $\boldsymbol{U}_r = \boldsymbol{U}^+_{e+\frac{1}{2}}$ and time average fluxes $\boldsymbol{F}_l = \boldsymbol{F}^-_{e+\frac{1}{2}}$, $\boldsymbol{F}_r = \boldsymbol{F}^+_{e+\frac{1}{2}}$. Here $\boldsymbol{U}_l$, $\boldsymbol{U}_r$ may be the solution values at time t_n for dissipation model D1 or the time average value in case of dissipation model D2. Further, we use the cell average values at time $t = t_n$, $\bar{\boldsymbol{U}}_l = \bar{\boldsymbol{u}}^n_e$, $\bar{\boldsymbol{U}}_r = \bar{\boldsymbol{u}}^n_{e+1}$, to compute the dissipation coefficients. In the following subsections, we described different numerical fluxes which are functions of the quantities: $\bar{\boldsymbol{U}}_l, \bar{\boldsymbol{U}}_r, \boldsymbol{U}_l, \boldsymbol{U}_r, \boldsymbol{F}_l, \boldsymbol{F}_r$.

D.1. Rusanov flux

The Rusanov flux [152] is a local version of the Lax-Friedrichs flux with the wave speed being estimated locally. The flux approximation is given by

$$\boldsymbol{F}(\boldsymbol{U}_l, \boldsymbol{U}_r, \boldsymbol{F}_l, \boldsymbol{F}_r; \bar{\boldsymbol{U}}_l, \bar{\boldsymbol{U}}_r) = \frac{1}{2}\left(\boldsymbol{F}_l + \boldsymbol{F}_r\right) - \frac{1}{2}\lambda\left(\boldsymbol{U}_r - \boldsymbol{U}_l\right)$$

where λ is an estimate of the maximum wave speed in the two states

$$\lambda = \max\left\{\rho(\bar{\boldsymbol{U}}_l), \rho(\bar{\boldsymbol{U}}_r)\right\}$$

and ρ denotes the spectral radius of the flux jacobian, $\boldsymbol{f}'(\boldsymbol{u})$.

D.2. Roe flux

The Roe flux [146] is built on a local linearization of the hyperbolic conservation law and solving the Riemann problem exactly. The Roe flux is given by

$$\boldsymbol{F}(\boldsymbol{U}_l, \boldsymbol{U}_r, \boldsymbol{F}_l, \boldsymbol{F}_r; \bar{\boldsymbol{U}}_l, \bar{\boldsymbol{U}}_r) = \frac{1}{2}\left(\boldsymbol{F}_l + \boldsymbol{F}_r\right) - \frac{1}{2}R\left|\Lambda\right|L\left(\boldsymbol{U}_r - \boldsymbol{U}_l\right)$$

where R, Λ, L are the right eigenvector matrix, diagonal matrix of eigenvalues and left eigenvector matrix corresponding to the flux Jacobian at the face, computed using the Roe average based on cell average values $\bar{\boldsymbol{U}}_l$, $\bar{\boldsymbol{U}}_r$.

D.3. HLL flux

The HLL Riemann solver [89] models the solution of the Riemann problem using only the slowest and fastest waves with an intermediate state. Let the slowest and fastest speeds, denoted by $S_l < S_r$, be assumed to be known. We can determine the intermediate state and flux by writing the jump conditions across the two waves,

$$\boldsymbol{F}_* - \boldsymbol{F}_l = S_l\left(\boldsymbol{U}_* - \boldsymbol{U}_l\right), \qquad \boldsymbol{F}_r - \boldsymbol{F}_* = S_r\left(\boldsymbol{U}_r - \boldsymbol{U}_*\right)$$

whose solution is given by

$$\boldsymbol{U}_* = \frac{S_r \boldsymbol{U}_r - S_l \boldsymbol{U}_l - (\boldsymbol{F}_r - \boldsymbol{F}_l)}{S_r - S_l}, \qquad \boldsymbol{F}_* = \frac{S_r \boldsymbol{F}_l - S_l \boldsymbol{F}_r + S_l S_r (\boldsymbol{U}_r - \boldsymbol{U}_l)}{S_r - S_l}$$

The numerical flux is given by

$$\boldsymbol{F}(\boldsymbol{U}_l, \boldsymbol{U}_r, \boldsymbol{F}_l, \boldsymbol{F}_r; \bar{\boldsymbol{U}}_l, \bar{\boldsymbol{U}}_r) = \begin{cases} \boldsymbol{F}_l, & S_l > 0 \\ \boldsymbol{F}_r, & S_r < 0 \\ \boldsymbol{F}_*, & \text{otherwise} \end{cases}$$

The speeds S_l, S_r are computed using the cell average values $\bar{\boldsymbol{U}}_l$, $\bar{\boldsymbol{U}}_r$ and there are various methods available [71, 23, 181, 83, 179]. In the numerical tests, we use the method from [181] to estimate the slowest and fastest speeds.

D.4. HLLC FLUX

We describe the HLLC flux for 1-D Compressible Euler's equations (4.16). The HLLC Riemann solver [180] includes the contact wave by using a three wave model with three wave speeds $S_l < S_* < S_r$ and two intermediate states $\boldsymbol{U}_{*l}$ and $\boldsymbol{U}_{*r}$. The contact wave is the middle wave with speed S_*. The pressure and normal velocity are continuous across the contact wave, i.e.,

$$p_{*l} = p_{*r} = p_*, \qquad u_{*l} = u_{*r} = u_*$$

and the speed of the contact wave coincides with the intermediate velocity $S_* = u_*$. The jump condition across the S_l and S_r wave reads as

$$\boldsymbol{F}_{*\alpha} - \boldsymbol{F}_\alpha = S_\alpha (\boldsymbol{U}_{*\alpha} - \boldsymbol{U}_\alpha), \quad \alpha = l, r$$

In the full form, the jump conditions are given by

$$\begin{bmatrix} \rho_{*\alpha} u_* \\ p_* + \rho_{*\alpha} u_*^2 \\ (E_{*\alpha} + p_*) u_* \end{bmatrix} - S_\alpha \begin{bmatrix} \rho_{*\alpha} \\ \rho_{*\alpha} u_* \\ E_{*\alpha} \end{bmatrix} = \begin{bmatrix} F_\alpha^\rho \\ F_\alpha^m \\ F_\alpha^E \end{bmatrix} - S_\alpha \begin{bmatrix} \rho_\alpha \\ m_\alpha \\ E_\alpha \end{bmatrix}$$

Using this expression we determine the unknown variables ρ_*, u_*, p_* and E_*. From the first jump condition we get

$$\rho_{*\alpha} = \frac{S_\alpha \rho_\alpha - F_\alpha^\rho}{S_\alpha - u_*}$$

From the second equation we write the intermediate pressure

$$p_* = F_\alpha^m - S_\alpha m_\alpha + \rho_{*\alpha} \boldsymbol{u}_* (S_\alpha - u_*) = F_\alpha^m - S_\alpha m_\alpha + u_* (S_\alpha \rho_\alpha - F_\alpha^\rho) \tag{D.1}$$

We get two estimates of pressure p_* from the l, r states, and equating these two values

$$F_l^m - S_l m_l + u_* (S_l \rho_l - F_l^\rho) \;=\; F_r^m - S_r m_r + u_* (S_r \rho_r - F_r^\rho)$$

we obtain the intermediate velocity

$$u_* = \frac{(S_r m_r - F_r^m) - (S_l m_l - F_l^m)}{(S_r \rho_r - F_r^\rho) - (S_l \rho_l - F_l^\rho)}$$

The intermediate pressure can be computed from (D.1) or from the following expression

$$p_* = \frac{(S_r\, m_r - F_r^m)\,(S_l\, \rho_l - F_l^\rho) - (S_l\, m_l - F_l^m)\,(S_r\, \rho_r - F_r^\rho)}{(S_r\, \rho_r - F_r^\rho) - (S_l\, \rho_l - F_l^\rho)}$$

From the last jump condition we obtain

$$E_{*\alpha} = \frac{p_*\, u_* + S_\alpha\, E_\alpha - F_\alpha^E}{S_\alpha - u_*}$$

The flux is now given by

$$\boldsymbol{F}\left(\boldsymbol{U}_l, \boldsymbol{U}_r, \boldsymbol{F}_l, \boldsymbol{F}_r; \bar{\boldsymbol{U}}_l, \bar{\boldsymbol{U}}_r\right) = \begin{cases} \boldsymbol{F}_l, & S_l > 0 \\ \boldsymbol{F}_r, & S_r < 0 \\ \boldsymbol{F}_{*l} = \boldsymbol{F}_l + S_l\,(\boldsymbol{U}_{*l} - \boldsymbol{U}_l), & S_l < 0 < u_* \\ \boldsymbol{F}_{*r} = \boldsymbol{F}_r + S_r\,(\boldsymbol{U}_{*r} - \boldsymbol{U}_r), & u_* < 0 < S_r \end{cases}$$

where the wave speeds S_l and S_r are computed using the cell average values $\bar{\boldsymbol{U}}_l, \bar{\boldsymbol{U}}_r$.

Appendix E

Efficient local differential operators

In our implementation of Lax-Wendroff Flux Reconstruction scheme, we use differentiation matrices for computing polynomial derivatives within an element. For instance, the matrix D defined in (3.5) is used to compute the local derivatives in the approximate Lax-Wendroff procedure (Section 4.2.4) and the matrix D_1 is used for computing the derivatives of the continuous flux (4.9, 4.24). This appendix describes how these derivative operators are applied in a cache blocking way [3] that avoids writing to memory (RAM). The approach is also used in `Trixi.jl` [141] and `PyFR` [195, 3].

We describe the process when dealing with the 1-D system of conservation laws (3.1) with `NVAR` variables solved on a grid of `ncell` cells and polynomial degree N. Let `u` be the solution array of size (`NVAR`, $N+1$, `ncell`) containing `Float64` values. The approximate Lax-Wendroff procedure (Section 4.2.4) and derivative of continuous flux (4.9, 4.24) require us to loop over the `ncell` cells and compute the flux derivative within each cell. A natural approach to compute the flux derivatives will be to compute fluxes at all solution points, store them in an array and apply the differentiation matrix. This is performed in the pseudocode below where `f` is an array of size (`NVAR`, $N+1$).

```
for cell in eachelement(grid) # Cell loop
    for i in eachnode(basis)  # DoF loop
        f[:,i] = flux(u[:,i,cell])
    end
    BLAS.mul(D, f, fder) # fder = D * f
end
```

This issue with this approach is that storing the flux in an array requires writing to memory (RAM). The idea of cache blocking is to compute the flux derivative without writing the flux to memory. This is ensured by computing the flux derivative by summing contributions to flux derivative from all solution points. To be precise, during the loop over solution points, we simply compute the flux at that solution point, compute its contribution to the derivative, and add it to target `fder`. At each solution point, the flux only consists of `NVAR` values of `Float64` type. The `NVAR` is relatively small (3 for 1-D Euler's equations (4.16), 6 for the ten moment problem (6.17)) and the code is set up to ensure that `NVAR` is known at the time of compilation. Thus, the `NVAR` flux values will be stored in the cache.

We now discuss how cache blocking is performed in practice. The implementation will typically be dependent on the programming language and we only describe it for Julia [29]. We first describe how to ensure that `NVAR` is known at the time of compilation.

```
abstract type AbstractEquations{NDIMS, NVAR} end
nvariables(::AbstractEquations{NDIMS,NVAR}) where {NDIMS, NVAR} = NVAR
```

The type `AbstractEquations` contains the number of dimensions `NDIMS` and number of variables `NVAR`. Since types are resolved at the time of compilation, these values will be known to the compiler and can be queried by the function `nvariables`. This `abstract type` is then used in a particular system of equations, like 1-D Euler's equations, as follows.

```
struct Euler1D <: AbstractEquations{1, 3}
    ⋮ # contains information like gas constant gamma
    ⋮
end
```

Thus, any instantiation of `Euler1D` will know `NDIMS=1` and `NVAR=3` so that these values are known at the time of compilation. If `eq` is an instantiation of such a `struct`, the following function makes use of this compile time information[E.1].

```
function mul_add_to_node_vars!(u, factor, u_node, eq, indices)
    for v in 1:nvariables(eq)
        u[v, indices] = u[v, indices] + factor * u_node[v]
    end
    return nothing
end
```

In the slicing notation of `python` and `fortran`, this function performs `u[:,indices] += u[:,indices] + factor * u_node` but with the capability of performing the operation faster as it knows the size of each slice. The final ingredient we will be needing in Julia is an array type that can store data in the cache. The array type we use is `SVector` (static vector) from the Julia package `StaticArrays.jl`. The usage of this package is recommended for arrays with less than 100 entries. To motivate the usage of `StaticArrays.jl`, we first show a pseudocode for application of the D_1 matrix on the flux (4.9, 4.24) in Algorithm E.1.

Algorithm E.1

Cache blocking flux differentiation

```
for cell in eachelement(grid) # Cell loop
    for i in eachnode(basis)  # DoF loop
        u_node = get_node_vars(equations, u, i, cell)
        f_node = flux(u_node)
        for ix in eachnode(basis)
            # Equivalent to fder[:,ix,i,cell] += D1[ix,i] * f_node
            mul_add_to_node_vars!(eq, D1[ix,i], f_node, fder, ix, cell)
        end
    end
end
```

The function `get_node_vars` in Algorithm E.1 loads the `NVAR` variables at a solution point into `u_node` of type `SVector`.

```
@inline function get_node_vars(u, eq, indices)
    SVector(ntuple(@inline(v -> u[v, indices]), nvariables(eq)))
end
```

E.1. Julia uses a just-in-time compiler and we thus do not need to specify the types when defining a function.

The function is made to use the `NVAR` information known at the time of compilation by using the `nvariables` function we described earlier. This leads to an `SVector` that will be stored in the cache. Then, the flux computation can be performed without a write to the memory, returning another `SVector` which will live in cache.

```
@inline function flux(u, eq::Euler1D)
    rho, rho_v1, rho_e = u
    v1 = rho_v1 / rho
    p  = (eq.gamma - 1) * (rho_e - 0.5 * rho_v1 * v1)
    f1 = rho_v1
    f2 = rho_v1 * v1 + p
    f3 = (rho_e + p) * v1
    return SVector(f1, f2, f3)
end
```

Once the flux at a solution point is computed, another loop over solution points is performed in Algorithm E.1 to add its contribution to `fder`. The `mul_add_to_node_vars!` function is used for efficient application of this operation. A minimal working example of a code that involves writing to memory is provided at

https://github.com/Arpit-Babbar/dissertation/blob/main/memory_write.jl

Another code with cache blocking is provided here

https://github.com/Arpit-Babbar/dissertation/blob/main/cache_block.jl

The codes have benchmarking built into them that clearly shows the many factors of improvement obtained by cache blocking. The figure E.1 illustrates cache blocking where $\tilde{f}, \tilde{u}$ denote `f_node`, `u_node` respectively.

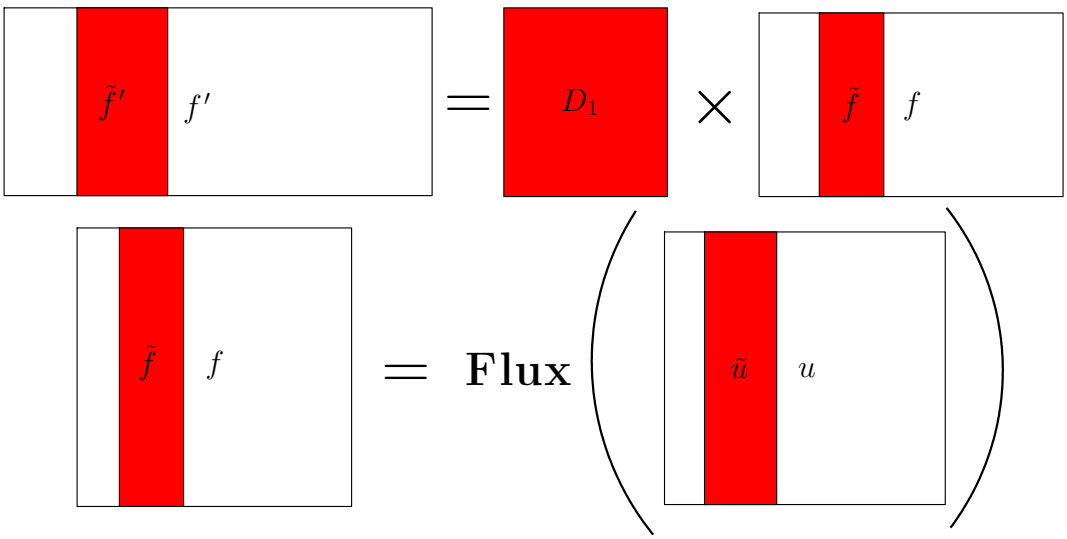


Figure E.1. Cache blocking flux differentiation (illustration from [192]).

Appendix F

Scaling limiter

In Chapters 5, 6, we developed Flux Reconstruction schemes that were admissibility preserving in means (Definition 5.2). In this section, we review the scaling limiter of [205] to use admissibility in means to obtain an admissibility preserving scheme (Definition 5.1).

Consider the solution $\boldsymbol{u}_h^n$ at current time time level n. Within each element, $\boldsymbol{u}_h^n \in \mathbb{P}_N$ and since the scheme is admissibility preserving in means, we assume $\bar{\boldsymbol{u}}_e^n \in \mathcal{U}_{\text{ad}}$ for each element e. We will iteratively correct all admissibility constraints $\{P_k\}_{k=1}^d$ (5.1). For each constraint P_k, we find $\theta_k \in [0,1]$ such that $P_k((1-\theta_k)\,\bar{\boldsymbol{u}}_e^n + \theta_k\,\boldsymbol{u}_h^n) > 0$ at the $N+1$ solution points and replace the polynomial $\boldsymbol{u}_h^n$ with $(1-\theta)\,\bar{\boldsymbol{u}}_e^n + \theta\,\boldsymbol{u}_h^n$. In case of concave P_k, we choose θ_k to be

$$\theta_k = \min\left(\min_{0\le p\le N}\left|\frac{\epsilon_p - P_k(\bar{\boldsymbol{u}}_e^n)}{P_k(\boldsymbol{u}_{e,p}^n) - P_k(\bar{\boldsymbol{u}}_e^n)}\right|, 1\right) \tag{F.1}$$

If P_k is not concave, we solve a nonlinear equation to find the largest $\theta_k \in [0,1]$ satisfying

$$P_k((1-\theta_k)\,\bar{\boldsymbol{u}}_e^n + \theta_k\,\boldsymbol{u}_{e,p}^n) = \epsilon_p, \qquad 0 \le p \le N \tag{F.2}$$

This procedure is performed for all k and the minimum is successively taken, as described in Algorithm F.1.

Algorithm F.1

Scaling limiter

$\theta = 1$
for $k = 1{:}K$ **do**
  $\epsilon_k = \frac{1}{10} P_k(\bar{\boldsymbol{u}}_e^n)$
  Find θ_k by solving (F.2) or by using (F.1) if P_k is concave
  $\theta \leftarrow \min(\theta_k, \theta)$
end for

The idea of choosing θ_k by solving (F.2) is to maintain the formal order of accuracy. In [206, 125], it was shown that (F.2) maintains optimal order of accuracy for Compressible Euler's equations (2.13) and Ten Moment problem (6.17) respectively. In [206, 125], the θ_k in (F.2) was found by solving a quadratic and cubic equation respectively. In this work, we solve (F.2) by using a general iterative solver like Newton-Raphson that can be used any choice of P_k.

Appendix G

Admissibility of MUSCL-Hancock on general grids

G.1. Introduction and notations

In this appendix, we prove Theorem 5.4 regarding admissibility of the MUSCL-Hancock scheme described in Section 5.4 on non-cell centred grids. These grids arise in the subcell based blending scheme of Section 5.3.1 as we demand a conservative scheme. The proof is provided here for general non-cell centred grids like in Figure G.1.

We now mention some notations that will be used in the proof. For the 1-D conservation law (3.1), define $\sigma(\boldsymbol{u}_1, \boldsymbol{u}_2)$ as

$$\sigma(\boldsymbol{u}_1, \boldsymbol{u}_2) = \max\left\{\rho(\boldsymbol{f}'(\boldsymbol{u}_\lambda)) \colon \boldsymbol{u}_\lambda = \lambda\,\boldsymbol{u}_1 + (1-\lambda)\,\boldsymbol{u}_2, \quad 0 \le \lambda \le 1\right\}$$

where $\rho(A)$ denotes the spectral radius of matrix A. For the 2-D hyperbolic conservation law

$$\boldsymbol{u}_t + \boldsymbol{f}_x + \boldsymbol{g}_y = \boldsymbol{0} \tag{G.1}$$

where $(\boldsymbol{f}, \boldsymbol{g})$ are Cartesian components of the flux vector; the wave speed estimates in x, y directions are defined as follows

$$\begin{aligned}
\sigma_x(\boldsymbol{u}_1, \boldsymbol{u}_2) &= \max\left\{\rho(\boldsymbol{f}'(\boldsymbol{u}_\lambda)) \colon \boldsymbol{u}_\lambda = \lambda\,\boldsymbol{u}_1 + (1-\lambda)\,\boldsymbol{u}_2, \quad 0 \le \lambda \le 1\right\} \\
\sigma_y(\boldsymbol{u}_1, \boldsymbol{u}_2) &= \max\left\{\rho(\boldsymbol{g}'(\boldsymbol{u}_\lambda)) \colon \boldsymbol{u}_\lambda = \lambda\,\boldsymbol{u}_1 + (1-\lambda)\,\boldsymbol{u}_2, \quad 0 \le \lambda \le 1\right\}
\end{aligned}$$

We assume that the admissibility set $\mathcal{U}_{\text{ad}}$ of the conservation law is a convex subset of $\mathbb{R}^p$ which can be written as (5.1). The following assumption is made concerning the admissibility of first order finite volume scheme.

Admissibility of first order finite volume scheme. Under the time step restriction

$$\max_p \frac{\Delta t}{\Delta x_p}\,\sigma(\boldsymbol{u}_p^n, \boldsymbol{u}_{p+1}^n) \le 1 \tag{G.2}$$

the first order finite volume method

$$\boldsymbol{u}_p^{n+1} = \boldsymbol{u}_p^n - \frac{\Delta t}{\Delta x_p}\left(\boldsymbol{f}(\boldsymbol{u}_p^n, \boldsymbol{u}_{p+1}^n) - \boldsymbol{f}(\boldsymbol{u}_{p-1}^n, \boldsymbol{u}_p^n)\right)$$

is admissibility preserving, i.e., $\boldsymbol{u}_p^n \in \mathcal{U}_{\text{ad}}$ for all p implies that $\boldsymbol{u}_p^{n+1} \in \mathcal{U}_{\text{ad}}$ for all p.

G.2. Review of MUSCL-Hancock scheme

Here we review the MUSCL-Hancock scheme for general uniform grids that need not be cell-centered (Figure G.1) in the sense that

$$x_{p+\frac{1}{2}} - x_p \neq x_p - x_{p-\frac{1}{2}}, \tag{G.3}$$

for some p where x_p is the solution point in finite volume element $(x_{p-\frac{1}{2}}, x_{p+\frac{1}{2}})$. The grid used in the blending limiter (Figure 5.1) is a special case of (G.3).

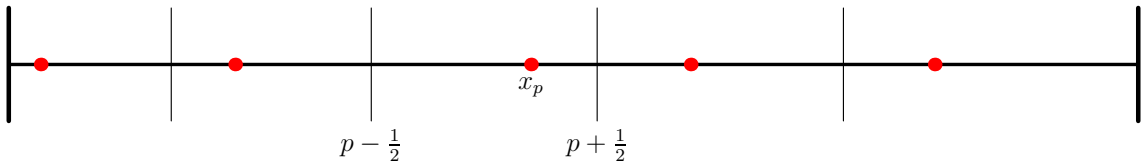


Figure G.1. Non-uniform, non-cell-centered finite volume grid

For the p^{th} finite volume element $(x_{p-\frac{1}{2}}, x_{p+\frac{1}{2}})$, the constant state is denoted $\boldsymbol{u}_p^n$ and the linear approximation will be denoted $\boldsymbol{r}_p^n(x)$. For conservative reconstruction[G.1], the linear reconstruction is given by

$$\boldsymbol{r}_p^n(x) = \boldsymbol{u}_p^n + (x - x_p)\,\boldsymbol{\delta}_p, \qquad x \in \left(x_{p-\frac{1}{2}}, x_{p+\frac{1}{2}}\right)$$

The values on left and right faces will be computed as

$$\boldsymbol{u}_p^{n,-} = \boldsymbol{u}_p^n + (x_{p-\frac{1}{2}} - x_p)\,\boldsymbol{\delta}_p, \qquad \boldsymbol{u}_p^{n,+} = \boldsymbol{u}_p + (x_{p+\frac{1}{2}} - x_p)\,\boldsymbol{\delta}_p \tag{G.4}$$

We use Taylor's expansion to evolve the solution to $t_n + \frac{1}{2}\Delta t$

$$\begin{aligned} \boldsymbol{u}_p^{n+\frac{1}{2},-} &= \boldsymbol{u}_p^{n,-} - \frac{\Delta t}{2\,\Delta x_p}\left(\boldsymbol{f}(\boldsymbol{u}_p^{n,+}) - \boldsymbol{f}(\boldsymbol{u}_p^{n,-})\right) \\ \boldsymbol{u}_p^{n+\frac{1}{2},+} &= \boldsymbol{u}_p^{n,+} - \frac{\Delta t}{2\,\Delta x_p}\left(\boldsymbol{f}(\boldsymbol{u}_p^{n,+}) - \boldsymbol{f}(\boldsymbol{u}_p^{n,-})\right) \end{aligned} \tag{G.5}$$

where $\Delta x_p = x_{p+\frac{1}{2}} - x_{p-\frac{1}{2}}$. The final update is performed by using an approximate Riemann solver on the evolved quantities

$$\boldsymbol{u}_p^{n+1} = \boldsymbol{u}_p^n - \frac{\Delta t}{\Delta x_p}\left(\boldsymbol{f}_{p+\frac{1}{2}}^{n+\frac{1}{2}} - \boldsymbol{f}_{p-\frac{1}{2}}^{n+\frac{1}{2}}\right) \tag{G.6}$$

G.1. The reconstruction uses conservative variables and hence is termed *conservative reconstruction*. It does not satisfy $\frac{1}{\Delta x_p}\int_{x_{p-\frac{1}{2}}}^{x_{p+\frac{1}{2}}} \boldsymbol{r}_p(x)\,\mathrm{d}x = \boldsymbol{u}_p$ as the linear reconstruction $\boldsymbol{r}_p$ is not centered at the mid point.

where

$$\boldsymbol{f}_{p+\frac{1}{2}}^{n+\frac{1}{2}} = \boldsymbol{f}\Big(\boldsymbol{u}_p^{n+\frac{1}{2},+}, \boldsymbol{u}_{p+1}^{n+\frac{1}{2},-}\Big)$$

is some numerical flux function. The key idea of the proof is to write the evolution $\boldsymbol{u}_p^{n+\frac{1}{2},\pm}$ from (G.5) as a convex combination of exact solution of some Riemann problem and the final update $\boldsymbol{u}_p^{n+1}$ from (G.6) as a convex combination of first order finite volume updates on appropriately chosen subcells.

G.3. Primary generalization for proof

For the uniform, cell-centered case, Berthon [26] defined $\boldsymbol{u}_p^{*,\pm}$ to satisfy

$$\frac{1}{2}\,\boldsymbol{u}_p^{n,-} + \boldsymbol{u}_p^{*,\pm} + \frac{1}{2}\,\boldsymbol{u}_p^{n,+} = 2\,\boldsymbol{u}_p^{n,\pm}$$

We generalize it for non-cell centered grids (G.3)

$$\mu_-\boldsymbol{u}_p^{n,-} + \boldsymbol{u}_p^{*,\pm} + \mu_+\boldsymbol{u}_p^{n,+} = 2\,\boldsymbol{u}_p^{n,\pm}$$

where

$$\mu_- = \frac{x_{p+\frac{1}{2}} - x_p}{x_{p+\frac{1}{2}} - x_{p-\frac{1}{2}}}, \qquad \mu_+ = \frac{x_p - x_{p-\frac{1}{2}}}{x_{p+\frac{1}{2}} - x_{p-\frac{1}{2}}} \tag{G.7}$$

This choice was made to keep the natural extension of $\boldsymbol{u}_p^{*,\pm}$ in the conservative reconstruction case:

$$\boldsymbol{u}_p^{*,\pm} = \boldsymbol{u}_p^n + 2\,(x_{p\pm\frac{1}{2}} - x_p)\,\boldsymbol{\delta}_p$$

noting that $\boldsymbol{u}_p^{n,\pm}$ are given by (G.4).

G.4. Proving admissibility

The following lemma about conservation laws will be crucial in the proof.

Lemma G.1. *Consider the 1-D Riemann problem*

$$\begin{aligned} \boldsymbol{u}_t + \boldsymbol{f}(\boldsymbol{u})_x &= \boldsymbol{0} \\ \boldsymbol{u}(x,0) &= \begin{cases} \boldsymbol{u}_l, & x<0 \\ \boldsymbol{u}_r, & x>0 \end{cases} \end{aligned}$$

in $[-h,h]\times[0,\Delta t]$ *where*

$$\frac{\Delta t}{h}\,\sigma(\boldsymbol{u}_l,\boldsymbol{u}_r) \le 1 \tag{G.8}$$

Then, for all $t \le \Delta t$,

$$\int_{-h}^{h} \boldsymbol{u}(x,t)\,dx = h\,(\boldsymbol{u}_l + \boldsymbol{u}_r) - t\,(\boldsymbol{f}(\boldsymbol{u}_r) - \boldsymbol{f}(\boldsymbol{u}_l))$$

Proof. Integrate the conservation law over $(-h,0)\times(0,t)$

$$\begin{aligned}0&=\int_{-h}^{0}\boldsymbol{u}(x,t)\mathrm{d}x-h\,\boldsymbol{u}_l+\int_0^t\left(\boldsymbol{f}(\boldsymbol{u}(0^-,t))-\boldsymbol{f}(\boldsymbol{u}(-h,t))\right)\mathrm{d}t\\&=\int_{-h}^{0}\boldsymbol{u}(x,t)\mathrm{d}x-h\,\boldsymbol{u}_l+t\left(\boldsymbol{f}(\tilde{\boldsymbol{u}}(0^-))-\boldsymbol{f}(\boldsymbol{u}_l)\right)\end{aligned}$$

where, by self-similarity of solution of Riemann problem, $\tilde{\boldsymbol{u}}$ is defined so that $\boldsymbol{u}(x,t)=\tilde{\boldsymbol{u}}(x/t)$ and $\boldsymbol{f}(\boldsymbol{u}(-h,t))=\boldsymbol{f}(\boldsymbol{u}_l)$ is obtained as waves do not reach $x=-h$ due to the time restriction (G.8). Rewriting gives

$$\int_{-h}^{0}\boldsymbol{u}(x,t)\mathrm{d}x=h\,\boldsymbol{u}_l-t(\boldsymbol{f}(\tilde{\boldsymbol{u}}(0^-)-\boldsymbol{f}(\boldsymbol{u}_l))$$

Similarly,

$$\int_0^h\boldsymbol{u}(x,t)\,\mathrm{d}x=h\,\boldsymbol{u}_r-t\left(\boldsymbol{f}(\boldsymbol{u}_r)-\boldsymbol{f}(\tilde{\boldsymbol{u}}(0^+))\right)$$

If $\tilde{\boldsymbol{u}}$ is discontinuous at $x=0$, by Rankine-Hugoniot conditions, we will have a stationary jump at $x/t=0$ and obtain $\boldsymbol{f}(\tilde{\boldsymbol{u}}(0^+))=\boldsymbol{f}(\tilde{\boldsymbol{u}}(0^-))$. The same trivially holds if $\tilde{\boldsymbol{u}}$ is continuous at $x/t=0$. Thus, we can sum the previous two identities to get (G.1). □

We will now give a criterion under which we can prove $\boldsymbol{u}_p^{n+\frac{1}{2},\pm}\in\mathcal{U}_{\mathrm{ad}}$, i.e., the evolution step (G.5) preserves $\mathcal{U}_{\mathrm{ad}}$.

LEMMA G.2. *Define $\mu_\pm$ by (G.7) and pick $\boldsymbol{u}_p^{*,\pm}$ to satisfy*

$$\frac{\mu_-}{2}\boldsymbol{u}_p^{n,-}+\frac{1}{2}\boldsymbol{u}_p^{*,\pm}+\frac{\mu_+}{2}\boldsymbol{u}_p^{n,+}=\boldsymbol{u}_p^{n,\pm}\tag{G.9}$$

Assume $\boldsymbol{u}_p^{n,\pm},\boldsymbol{u}_p^{*,\pm}\in\mathcal{U}_{\mathrm{ad}}$ and the CFL restrictions

$$\max_p\frac{\Delta t}{\mu_-\Delta x_p}\sigma(\boldsymbol{u}_p^{n,-},\boldsymbol{u}_p^{*,\pm})\le 1,\qquad\max_p\frac{\Delta t}{\mu_+\Delta x_p}\sigma(\boldsymbol{u}_p^{*,\pm},\boldsymbol{u}_p^{n,+})\le 1\tag{G.10}$$

are satisfied. Then, $\boldsymbol{u}_p^{n+\frac{1}{2},\pm}$ given by the first step (G.5) of the MUSCL-Hancock scheme is in $\mathcal{U}_{\mathrm{ad}}$.

Proof. We will prove that $\boldsymbol{u}_p^{n+\frac{1}{2},+}\in\mathcal{U}_{\mathrm{ad}}$, and the proof for $\boldsymbol{u}_p^{n+\frac{1}{2},-}$ shall follow similarly. The key idea is to write $\boldsymbol{u}_p^{n+\frac{1}{2},\pm}$ as the exact solution of some Riemann problems. Define $\boldsymbol{u}^h(x,t)\colon(x_{p-\frac{1}{2}},x_{p+\frac{1}{2}})\times(0,\Delta t/2)\to\mathcal{U}_{\mathrm{ad}}$ to be the weak solution of the Cauchy problem with initial data

$$\boldsymbol{u}^h(x,0)=\begin{cases}\boldsymbol{u}_p^{n,-}, & \text{if } x\in(x_{p-\frac{1}{2}},x_{p-1/4})\\ \boldsymbol{u}_p^{*,+}, & \text{if } x\in(x_{p-1/4},x_{p+1/4})\\ \boldsymbol{u}_p^{n,+}, & \text{if } x\in(x_{p+1/4},x_{p+\frac{1}{2}})\end{cases}$$

where

$$x_{p-\frac{1}{4}} = \frac{1}{2}(x_{p-\frac{1}{2}} + \tilde{x}_p), \qquad x_{p+\frac{1}{4}} = \frac{1}{2}(\tilde{x}_p + x_{p+\frac{1}{2}}), \qquad \tilde{x}_p = x_{p-\frac{1}{2}} + \mu_- \Delta x_p$$

Under our time step restrictions (G.10), the solution $\boldsymbol{u}^h$ at time $\frac{\Delta t}{2}$ is made up of non-interacting Riemann problems centered at $x_{p\pm\frac{1}{4}}$, see Figure G.2. We take the projection of $\boldsymbol{u}^h(x, \Delta t/2)$ on piecewise-constant functions

$$\tilde{\boldsymbol{u}}_p^{n+\frac{1}{2},+} := \frac{1}{\Delta x_p} \int_{x_{p-\frac{1}{2}}}^{x_{p+\frac{1}{2}}} \boldsymbol{u}^h\left(x, \frac{\Delta t}{2}\right) \mathrm{d}x$$

Since we assumed that the conservation law preserves $\mathcal{U}_{\text{ad}}$, we get $\tilde{\boldsymbol{u}}_p^{n+\frac{1}{2},+} \in \mathcal{U}_{\text{ad}}$. If we prove $\tilde{\boldsymbol{u}}_p^{n+\frac{1}{2},+} = \boldsymbol{u}_p^{n+\frac{1}{2},+}$, we will have our claim. Applying Lemma G.1 to the two non-interacting Riemann problems, we get

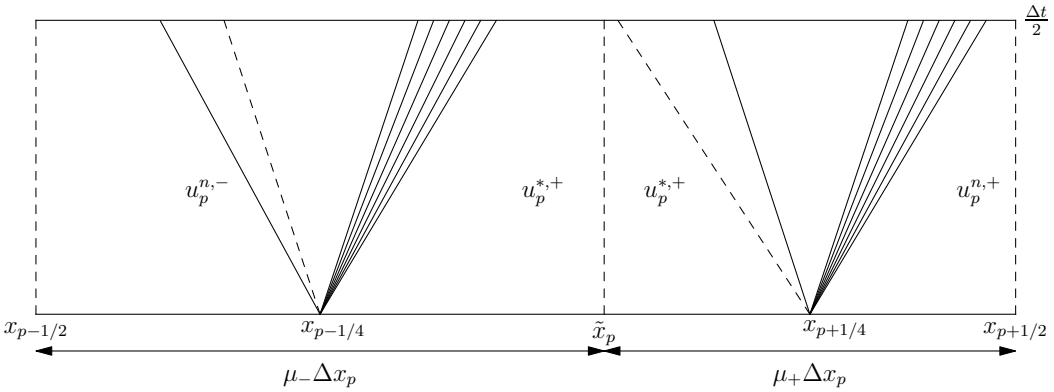


Figure G.2. Two non-interacting Riemann problems

$$\begin{aligned}
\tilde{\boldsymbol{u}}_p^{n+\frac{1}{2},+} &= \frac{1}{\Delta x_p}\left(\int_{x_{p-\frac{1}{2}}}^{\tilde{x}_p} \boldsymbol{u}^h\left(x, \frac{\Delta t}{2}\right) \mathrm{d}x + \int_{\tilde{x}_p}^{x_{p+\frac{1}{2}}} \boldsymbol{u}^h\left(x, \frac{\Delta t}{2}\right) \mathrm{d}x\right) \\
&= \frac{1}{\Delta x_p}\left[\begin{array}{c} \dfrac{\tilde{x}_p - x_{p-\frac{1}{2}}}{2} \boldsymbol{u}_p^{n,-} + \dfrac{\Delta x_p}{2} \boldsymbol{u}_p^{*,+} + \dfrac{x_{p+\frac{1}{2}} - \tilde{x}_p}{2} \boldsymbol{u}_p^{n,+} \\ -\dfrac{\Delta t}{2}\left(\boldsymbol{f}(\boldsymbol{u}_p^{n,+}) - \boldsymbol{f}(\boldsymbol{u}_p^{n,-})\right) \end{array}\right] \\
&= \frac{1}{2}(\mu_- \boldsymbol{u}_p^{n,-} + \boldsymbol{u}_p^{*,+} + \mu_+ \boldsymbol{u}_p^{n,+}) - \frac{\Delta t/2}{\Delta x_p}\left(\boldsymbol{f}(\boldsymbol{u}_p^{n,+}) - \boldsymbol{f}(\boldsymbol{u}_p^{n,-})\right) \\
&= \boldsymbol{u}_p^{n,+} - \frac{\Delta t/2}{\Delta x_p}\left(\boldsymbol{f}(\boldsymbol{u}_p^{n,+}) - \boldsymbol{f}(\boldsymbol{u}_p^{n,-})\right), \quad \text{using (G.9)} \\
&= \boldsymbol{u}_p^{n+\frac{1}{2},+}, \quad \text{by (G.5)}
\end{aligned}$$

This proves our claim. □

Now, we introduce a new variable $\boldsymbol{u}_p^{n+\frac{1}{2},*}$ defined as follows:

$$\mu_- \boldsymbol{u}_p^{n+\frac{1}{2},-} + \boldsymbol{u}_p^{n+\frac{1}{2},*} + \mu_+ \boldsymbol{u}_p^{n+\frac{1}{2},+} = 2\,\boldsymbol{u}_p^n \tag{G.11}$$

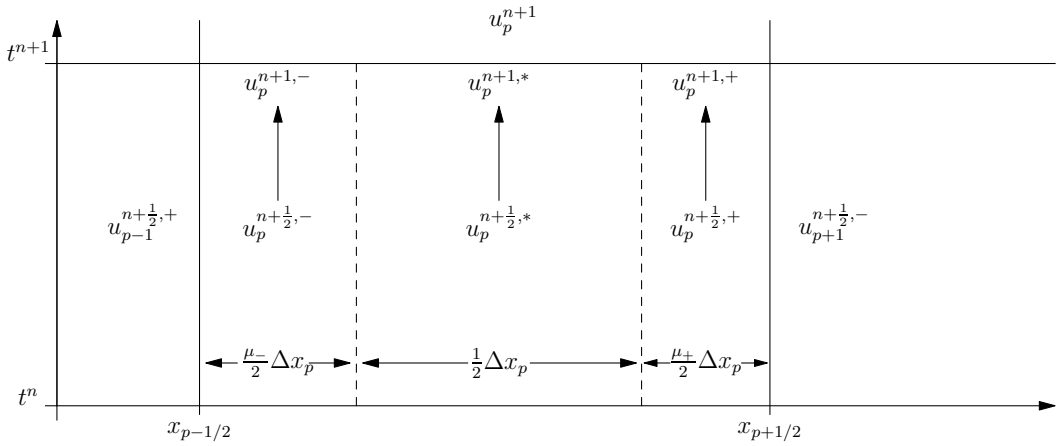


Figure G.3. Finite volume evolution

As illustrated in Figure G.3, we evolve each state according to the associated first order scheme to define the following

$$\begin{aligned}
\boldsymbol{u}_p^{n+1,-} &= \boldsymbol{u}_p^{n+\frac{1}{2},-} - \frac{\Delta t}{\mu_- \Delta x_p/2}\left(\boldsymbol{f}\left(\boldsymbol{u}_p^{n+\frac{1}{2},-}, \boldsymbol{u}_p^{n+\frac{1}{2},*}\right) - \boldsymbol{f}\left(\boldsymbol{u}_{p-1}^{n+\frac{1}{2},+}, \boldsymbol{u}_p^{n+\frac{1}{2},-}\right)\right) \\
\boldsymbol{u}_p^{n+1,*} &= \boldsymbol{u}_p^{n+\frac{1}{2},*} - \frac{\Delta t}{\Delta x_p/2}\left(\boldsymbol{f}\left(\boldsymbol{u}_p^{n+\frac{1}{2},*}, \boldsymbol{u}_p^{n+\frac{1}{2},+}\right) - \boldsymbol{f}\left(\boldsymbol{u}_p^{n+\frac{1}{2},-}, \boldsymbol{u}_p^{n+\frac{1}{2},*}\right)\right) \qquad \text{(G.12)} \\
\boldsymbol{u}_p^{n+1,+} &= \boldsymbol{u}_p^{n+\frac{1}{2},+} - \frac{\Delta t}{\mu_+ \Delta x_p/2}\left(\boldsymbol{f}\left(\boldsymbol{u}_p^{n+\frac{1}{2},+}, \boldsymbol{u}_{p+1}^{n+\frac{1}{2},-}\right) - \boldsymbol{f}\left(\boldsymbol{u}_p^{n+\frac{1}{2},*}, \boldsymbol{u}_p^{n+\frac{1}{2},+}\right)\right)
\end{aligned}$$

Recall that (G.6) is

$$\boldsymbol{u}_p^{n+1} = \boldsymbol{u}_p^n - \frac{\Delta t}{\Delta x_p}\left(\boldsymbol{f}\left(\boldsymbol{u}_p^{n+\frac{1}{2},+}, \boldsymbol{u}_{p+1}^{n+\frac{1}{2},-}\right) - \boldsymbol{f}\left(\boldsymbol{u}_{p-1}^{n+\frac{1}{2},+}, \boldsymbol{u}_p^{n+\frac{1}{2},-}\right)\right)$$

Using (G.11) and (G.12), we get

$$\frac{\mu_-}{2}\boldsymbol{u}_p^{n+1,-} + \frac{1}{2}\boldsymbol{u}_p^{n+1,*} + \frac{\mu_+}{2}\boldsymbol{u}_p^{n+1,+} = \boldsymbol{u}_p^{n+1}$$

Thus, assuming $\boldsymbol{u}_p^{n+\frac{1}{2}\pm}, \boldsymbol{u}_p^{n+\frac{1}{2},*} \in \mathcal{U}_{\text{ad}}$ for all p, and since $\frac{1}{2}\mu_- + \frac{1}{2}\mu_+ = 1$, we get $\boldsymbol{u}_p^{n+1} \in \mathcal{U}_{\text{ad}}$ under the following time step restrictions arising from the assumed time step requirement (G.2) for admissibility of the first order finite volume method

$$\begin{aligned}
&\max_p \frac{\Delta t}{\mu_- \Delta x_p/2}\sigma\left(\boldsymbol{u}_p^{n+\frac{1}{2},-}, \boldsymbol{u}_p^{n+\frac{1}{2},*}\right) \le 1, \quad \max_p \frac{\Delta t}{\Delta x_p/2}\sigma\left(\boldsymbol{u}_p^{n+\frac{1}{2},-}, \boldsymbol{u}_p^{n+\frac{1}{2},*}\right) \le 1 \\
&\max_p \frac{\Delta t}{\mu_- \Delta x_p/2}\sigma\left(\boldsymbol{u}_{p-1}^{n+\frac{1}{2},+}, \boldsymbol{u}_p^{n+\frac{1}{2},-}\right) \le 1, \quad \max_p \frac{\Delta t}{\mu_+ \Delta x_p/2}\sigma\left(\boldsymbol{u}_p^{n+\frac{1}{2},+}, \boldsymbol{u}_{p+1}^{n+\frac{1}{2},-}\right) \le 1 \qquad \text{(G.13)} \\
&\max_p \frac{\Delta t}{\Delta x_p/2}\sigma\left(\boldsymbol{u}_p^{n+\frac{1}{2},*}, \boldsymbol{u}_p^{n+\frac{1}{2},+}\right) \le 1, \quad \max_p \frac{\Delta t}{\mu_+ \Delta x_p/2}\sigma\left(\boldsymbol{u}_p^{n+\frac{1}{2},*}, \boldsymbol{u}_p^{n+\frac{1}{2},+}\right) \le 1
\end{aligned}$$

This can be summarised in the following Lemma.

LEMMA G.3. *Assume that the states* $\left\{\boldsymbol{u}_p^{n+\frac{1}{2},\pm}\right\}_p$, $\left\{\boldsymbol{u}_p^{n+\frac{1}{2},*}\right\}_p$ *belong to* $\mathcal{U}_{\text{ad}}$, *where* $\boldsymbol{u}_p^{n+\frac{1}{2},*}$ *is defined as in* (G.11). *Then, the updated solution* $\boldsymbol{u}_p^{n+1}$ *of MUSCL-Hancock scheme (G.4-G.6) is in* $\mathcal{U}_{\text{ad}}$ *under the CFL conditions* (G.13).

Since Lemma G.2 states that $\boldsymbol{u}_p^{n+\frac{1}{2},\pm} \in \mathcal{U}_{\text{ad}}$ if $\boldsymbol{u}_p^{*,\pm} \in \mathcal{U}_{\text{ad}}$, the only new condition pertains to $\boldsymbol{u}_p^{n+\frac{1}{2},*}$. Our goal now is to understand this condition, and ultimately prove that it follows from the requirement that $\boldsymbol{u}_p^{*,\pm} \in \mathcal{U}_{\text{ad}}$ in case of conservative reconstruction.

Recall that $\boldsymbol{u}_p^{n+\frac{1}{2},*}$ was defined by (G.11); expanding the definition of $\boldsymbol{u}_p^{n+\frac{1}{2},\pm}$ given by (G.5) yields

$$\boldsymbol{u}_p^{n+\frac{1}{2},*} = 2\,\boldsymbol{u}_p^n - (\mu_- \boldsymbol{u}_p^{n,-} + \mu_+ \boldsymbol{u}_p^{n,+}) - \frac{\Delta t}{2\,\Delta x_p}\,(\boldsymbol{f}(\boldsymbol{u}_p^{n,-}) - \boldsymbol{f}(\boldsymbol{u}_p^{n,+})) \tag{G.14}$$

This identity (G.14) will be seen as an evolution update similar to (G.5) with $\boldsymbol{u}_p^{n,+}$ and $\boldsymbol{u}_p^{n,-}$ being swapped and $\boldsymbol{u}_p^n$ replaced with $2\,\boldsymbol{u}_p^n - (\mu_-\,\boldsymbol{u}_p^{n,-} + \mu_+\,\boldsymbol{u}_p^{n,+})$. The admissibility of $\boldsymbol{u}_p^{n+\frac{1}{2},*}$ will be studied by adapting the proof of admissibility for (G.5), accounting for the differences in the case of (G.14). Define $\boldsymbol{u}_p^{*,*}$ so that

$$\frac{\mu_-}{2}\,\boldsymbol{u}_p^{n,-} + \frac{1}{2}\,\boldsymbol{u}_p^{*,*} + \frac{\mu_+}{2}\,\boldsymbol{u}_p^{n,+} = 2\,\boldsymbol{u}_p^n - (\mu_-\,\boldsymbol{u}_p^{n,-} + \mu_+\,\boldsymbol{u}_p^{n,+}) \tag{G.15}$$

i.e.,

$$\boldsymbol{u}_p^{*,*} = 4\,\boldsymbol{u}_p^n - 3\,(\mu_-\,\boldsymbol{u}_p^{n,-} + \mu_+\,\boldsymbol{u}_p^{n,+}) \tag{G.16}$$

The following Lemma extends the proof of Lemma G.2 to obtain conditions for $\boldsymbol{u}_p^{n+\frac{1}{2},*} \in \mathcal{U}_{\text{ad}}$.

Lemma G.4. *Assume that $\boldsymbol{u}_p^n \in \mathcal{U}_{\text{ad}}$ for all p. Consider the reconstructions $\boldsymbol{u}_p^{n,\pm}$ and the $\boldsymbol{u}_p^{*,*}$ defined in (G.15). Assume $\boldsymbol{u}_p^{n,\pm}, \boldsymbol{u}_p^{*,*} \in \mathcal{U}_{\text{ad}}$ and the time step restrictions*

$$\max_p \frac{\Delta t}{\mu_-\,\Delta x_p}\,\sigma(\boldsymbol{u}_p^{*,*}, \boldsymbol{u}_p^{n,-}) \le 1, \qquad \max_p \frac{\Delta t}{\mu_+\,\Delta x_p}\,\sigma(\boldsymbol{u}_p^{n,+}, \boldsymbol{u}_p^{*,*}) \le 1 \tag{G.17}$$

Then $\boldsymbol{u}_p^{n+\frac{1}{2},*} \in \mathcal{U}_{\text{ad}}$.

Proof. We will use the identity which follows from (G.14, G.15)

$$\boldsymbol{u}_p^{n+\frac{1}{2},*} = \frac{\mu_-\,\boldsymbol{u}_p^{n,-} + \boldsymbol{u}_p^{*,*} + \mu_+\,\boldsymbol{u}_p^{n,+}}{2} - \frac{\Delta t}{2\,\Delta x_p}\,(\boldsymbol{f}(\boldsymbol{u}_p^{n,-}) - \boldsymbol{f}(\boldsymbol{u}_p^{n,+})) \tag{G.18}$$

to fall back to previous case of Lemma G.2.

Define $\boldsymbol{u}^h(x,t)\colon (x_{p-\frac{1}{2}}, x_{p+\frac{1}{2}}) \times (0, \Delta t/2) \to \mathcal{U}_{\text{ad}}$ to be the weak solution of the Cauchy problem with initial data

$$\boldsymbol{u}^h(x,0) = \begin{cases} \boldsymbol{u}_p^{n,+}, & \text{if } x \in (x_{p-\frac{1}{2}}, x_{p-1/4}) \\ \boldsymbol{u}_p^{*,*}, & \text{if } x \in (x_{p-\frac{1}{4}}, x_{p+1/4}) \\ \boldsymbol{u}_p^{n,-}, & \text{if } x \in (x_{p+\frac{1}{4}}, x_{p+\frac{1}{2}}) \end{cases}$$

where

$$x_{p-\frac{1}{4}} = \frac{1}{2}\,(x_{p-\frac{1}{2}} + x_p), \qquad x_{p+\frac{1}{4}} = \frac{1}{2}\,(x_p + x_{p+\frac{1}{2}})$$

Note that we have already accounted for the swapped $\boldsymbol{u}_p^{n,-}$ and $\boldsymbol{u}_p^{n,+}$ while defining this initial condition, see Figure G.4.

Under the assumed CFL conditions (G.17), the solution $\boldsymbol{u}^h$ at time $\frac{\Delta t}{2}$ is made up of non-interacting Riemann problems centered at $x_{p\pm\frac{1}{4}}$. Take the projection of $\boldsymbol{u}^h(x, t/2)$ on piecewise-constant functions

$$\tilde{\boldsymbol{u}}_p^{n+\frac{1}{2},*} := \frac{1}{\Delta x_p} \int_{x_{p-\frac{1}{2}}}^{x_{p+\frac{1}{2}}} \boldsymbol{u}^h\left(x, \frac{\Delta t}{2}\right) \mathrm{d}x \in \mathcal{U}_{\mathrm{ad}}$$

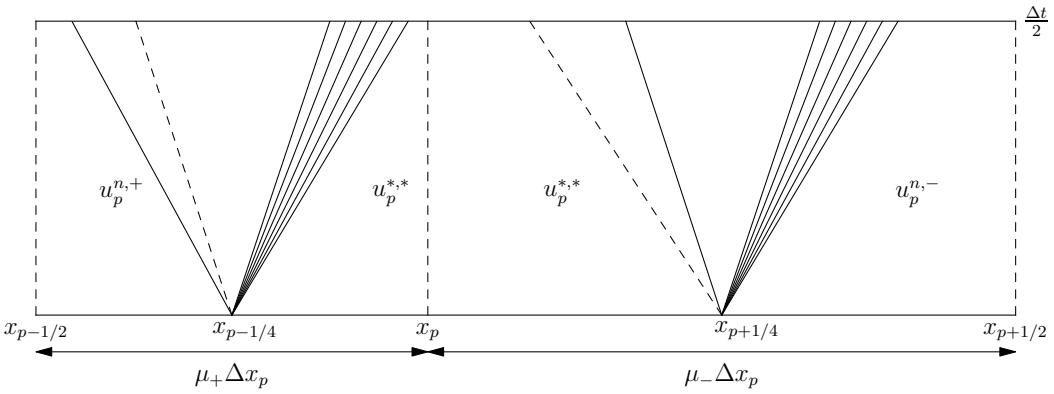


Figure G.4. Two non-interacting Riemann problems

As in Lemma G.2, we will show $\boldsymbol{u}_p^{n+\frac{1}{2},*} \in \mathcal{U}_{\mathrm{ad}}$ by showing $\boldsymbol{u}_p^{n+\frac{1}{2},*} = \tilde{\boldsymbol{u}}_p^{n+\frac{1}{2},*}$. Applying Lemma G.1 to the two non-interacting Riemann problems, we get

$$\begin{aligned}
\tilde{\boldsymbol{u}}_p^{n+\frac{1}{2},*} &= \frac{1}{\Delta x_p}\left(\int_{x_{p-\frac{1}{2}}}^{x_p} \boldsymbol{u}^h\left(x, \frac{\Delta t}{2}\right) \mathrm{d}x + \int_{x_p}^{x_{p+\frac{1}{2}}} \boldsymbol{u}^h\left(x, \frac{\Delta t}{2}\right) \mathrm{d}x\right) \\
&= \frac{1}{\Delta x_p}\left[\begin{array}{c} \dfrac{x_p - x_{p-\frac{1}{2}}}{2} \boldsymbol{u}_p^{n,+} + \dfrac{\Delta x_p}{2} \boldsymbol{u}_p^{*,*} + \dfrac{x_{p+\frac{1}{2}} - x_p}{2} \boldsymbol{u}_p^{n,-} \\ -\dfrac{\Delta t}{2}\left(\boldsymbol{f}(\boldsymbol{u}_p^{n,-}) - \boldsymbol{f}(\boldsymbol{u}_p^{n,+})\right) \end{array}\right] \\
&= \frac{1}{2}\left(\mu_+ \boldsymbol{u}_p^{n,+} + \boldsymbol{u}_p^{*,*} + \mu_- \boldsymbol{u}_p^{n,-}\right) - \frac{\Delta t/2}{\Delta x_p}\left(\boldsymbol{f}(\boldsymbol{u}_p^{n,-}) - \boldsymbol{f}(\boldsymbol{u}_p^{n,+})\right) \\
&= \boldsymbol{u}_p^{n+\frac{1}{2},*}, \quad \text{by (G.18)}
\end{aligned}$$

This proves our claim. □

For conservative reconstruction,

$$\mu_- \boldsymbol{u}_p^{n,-} + \mu_+ \boldsymbol{u}_p^{n,+} = \boldsymbol{u}_p^n$$

and thus by (G.16), $\boldsymbol{u}_p^{*,*} = \boldsymbol{u}_p^n$. The previous lemma can thus be specialized as follows.

LEMMA G.5. *Assume that $\boldsymbol{u}_p^n \in \mathcal{U}_{\mathrm{ad}}$ and $\boldsymbol{u}_p^{n,\pm} \in \mathcal{U}_{\mathrm{ad}}$ for all p with conservative reconstruction. Also assume the CFL restrictions*

$$\max_p \frac{\Delta t}{\mu_- \Delta x_p} \sigma(\boldsymbol{u}_p^n, \boldsymbol{u}_p^{n,-}) \le 1, \qquad \max_p \frac{\Delta t}{\mu_+ \Delta x_p} \sigma(\boldsymbol{u}_p^{n,+}, \boldsymbol{u}_p^n) \le 1 \tag{G.19}$$

where $\mu^{\pm}$ *are defined in* (G.7). *Then,* $\boldsymbol{u}_p^{n+\frac{1}{2},*}$ *defined in* (G.11) *is in* $\mathcal{U}_{\text{ad}}$.

Combining Lemmas G.2, G.3, G.5, we obtain the final criterion for admissibility preservation of MUSCL-Hancock with conservative reconstruction in the following Theorem G.6.

Theorem G.6. *Let* $\boldsymbol{u}_p^n \in \mathcal{U}_{\text{ad}}$ *for all* p *and* $\boldsymbol{u}_p^{n,\pm}$ *be the conservative reconstructions defined as*

$$\boldsymbol{u}_p^{n,+} = \boldsymbol{u}_p^n + (x_{p+\frac{1}{2}} - x_p)\,\boldsymbol{\delta}_p, \qquad \boldsymbol{u}_p^{n,-} = \boldsymbol{u}_p^n + (x_{p-\frac{1}{2}} - x_p)\,\boldsymbol{\delta}_p$$

so that $\boldsymbol{u}_p^{*,\pm}$ *defined in* (G.9) *is also given by*

$$\boldsymbol{u}_p^{*,\pm} = \boldsymbol{u}_p^n + 2\,(x_{p\pm\frac{1}{2}} - x_p)\,\boldsymbol{\delta}_p \tag{G.20}$$

Assume that the slope $\boldsymbol{\delta}_p$ *is chosen such that* $\boldsymbol{u}_p^{*,\pm} \in \mathcal{U}_{\text{ad}}$ *and the CFL restrictions (G.10, G.13, G.19) hold. Then, the updated solution* $\boldsymbol{u}_p^{n+1}$, *defined by MUSCL-Hancock scheme* (G.6) *is in* $\mathcal{U}_{\text{ad}}$.

Proof. Once we obtain $\boldsymbol{u}_p^{n,\pm} \in \mathcal{U}_{\text{ad}}$, the claim follows from Lemmas G.2-G.5. To prove that $\boldsymbol{u}_p^{n,\pm}$ is indeed in $\mathcal{U}_{\text{ad}}$, we make the straight forward observation that

$$\boldsymbol{u}_p^{n,\pm} = \frac{1}{2}\,\boldsymbol{u}_p^{*,\pm} + \frac{1}{2}\,\boldsymbol{u}_p^n$$

Since $\boldsymbol{u}_p^{*,\pm}$ and $\boldsymbol{u}_p^n$ are in $\mathcal{U}_{\text{ad}}$, the proof is completed by the convex property of $\mathcal{U}_{\text{ad}}$. □

Remark G.7. The strictest time step restriction for admissibility of the MUSCL-Hancock scheme is imposed by (G.13). Thus, we can find the CFL coefficient for grid used by subcell-based blending scheme (5.8) by minimizing the denominator in (G.13) which is given by

$$\frac{1}{2} \min_{p=0,\dots,N} \left(\xi_p - \sum_{k=0}^{p-1} w_k \right) w_p = \frac{1}{2}\,\xi_0\, w_0$$

where ξ_0, w_0 are the first Gauss-Legendre quadrature point (3.2) and weight in $[0,1]$. This coefficient is less than half of the optimal CFL coefficient that arises from Fourier stability analysis of the LWFR scheme with D2 dissipation, see Table 4.1.

G.5. Non-conservative reconstruction

To maintain the simple admissibility criterion (Theorem G.6), we have restricted ourselves to conservative reconstruction in this work. In this section, we explain the complexities that will arise in enforcing admissibility if we perform reconstruction with non-conservative variables $\boldsymbol{v}$ defined by the change of variables formula

$$\boldsymbol{v} = \kappa(\boldsymbol{u})$$

The linear approximation is given by

$$\boldsymbol{r}_p^n(x) = \boldsymbol{v}_p^n + (x - x_p)\,\boldsymbol{\delta}_p, \quad x \in [x_{p-\frac{1}{2}}, x_{p+\frac{1}{2}}]$$

and thus the trace values are

$$\boldsymbol{v}_p^{n,\pm} = \boldsymbol{v}_p^n + (x_{p\pm\frac{1}{2}} - x_p)\,\boldsymbol{\delta}_p$$

Since the arguments of proof of admissibility depend on constraints on the conservative variables, we have to take the inverse map on our reconstructions. For example, conservative variables at the face are obtained as

$$\boldsymbol{u}_p^{n,\pm} = \kappa^{-1}(\boldsymbol{v}_p^{n,\pm}) \tag{G.21}$$

Due to the non-linearity of the map κ, unlike the conservative case, we have

$$\mu_-\,\boldsymbol{u}_p^{n,-} + \mu_+\,\boldsymbol{u}_p^{n,+} \neq \boldsymbol{u}_p^n$$

which is why several reductions of admissibility constraints will fail. The admissibility criteria for non-conservative reconstruction is stated in Theorem G.8.

THEOREM G.8. *Assume that $\boldsymbol{u}_p^n \in \mathcal{U}_{\text{ad}}$ for all p. Consider $\boldsymbol{u}_p^{n,\pm}$ defined in (G.21), $\boldsymbol{u}_p^{*,\pm}$ defined in (G.9) and $\boldsymbol{u}_p^{*,*}$ defined so that*

$$\frac{\mu_-}{2}\,\boldsymbol{u}_p^{n,-} + \frac{1}{2}\,\boldsymbol{u}_p^{*,*} + \frac{\mu_+}{2}\,\boldsymbol{u}_p^{n,+} = 2\,\boldsymbol{u}_p^n - (\mu_-\,\boldsymbol{u}_p^{n,-} + \mu_+\,\boldsymbol{u}_p^{n,+})$$

Assume that the slope $\boldsymbol{\delta}_p$ is chosen so that $\boldsymbol{u}_p^{n,\pm}, \boldsymbol{u}_p^{*,\pm}, \boldsymbol{u}_p^{*,*} \in \mathcal{U}_{\text{ad}}$ and that the CFL restrictions (G.10, G.13, G.17) are satisfied. Then the updated solution $\boldsymbol{u}_p^{n+1}$ of MUSCL-Hancock scheme (G.6) is in $\mathcal{U}_{\text{ad}}$.

Remark G.9. In the case of conservative reconstruction, we propose a simple and problem independent slope limiter to enforce the conditions for Theorem G.6 in Section 5.4.1. However, such a procedure cannot be used for nonconservative reconstruction because the slope has a nonlinear relation with the conservative variables (G.21). Thus, a problem dependent procedure similar to Section 5 of [26] will have to be developed for the non-cell centred grids.

G.6. MUSCL-HANCOCK SCHEME IN 2-D

Consider the 2-D hyperbolic conservation law (G.1) with fluxes $\boldsymbol{f}, \boldsymbol{g}$. For simplicity, assume that the reconstruction is performed on conservative variables. Thus, the linear reconstructions are given by

$$\boldsymbol{r}_{pq}^n(x, y) = \boldsymbol{u}_{pq}^n + (x - x_p)\,\boldsymbol{\delta}_p^x + (y - y_q)\,\boldsymbol{\delta}_q^y$$

and the approximations at the face $\boldsymbol{u}^{n,+x}, \boldsymbol{u}^{n,-x}, \boldsymbol{u}^{n,+y}, \boldsymbol{u}^{n,-y}$ are

$$\begin{aligned} \boldsymbol{u}_{pq}^{n,\pm x} &= r_{pq}^n(x_{p\pm\frac{1}{2}}, y_q) = \boldsymbol{u}_{pq}^n + (x_{p\pm\frac{1}{2}} - x_p)\,\boldsymbol{\delta}_p^x \\ \boldsymbol{u}_{pq}^{n,\pm y} &= r_{pq}^n(x_p, y_{q\pm\frac{1}{2}}) = \boldsymbol{u}_{pq}^n + (y_{q\pm\frac{1}{2}} - y_q)\,\boldsymbol{\delta}_q^y \end{aligned} \tag{G.22}$$

and the derivative approximations are given by

$$\partial_x \boldsymbol{f}_{pq} := \frac{1}{\Delta x_p}\left(\boldsymbol{f}(\boldsymbol{u}_{pq}^{n,+x}) - \boldsymbol{f}(\boldsymbol{u}_{pq}^{n,-x})\right), \qquad \partial_y \boldsymbol{g}_{pq} := \frac{1}{\Delta y_q}\left(\boldsymbol{g}(\boldsymbol{u}_{pq}^{n,+y}) - \boldsymbol{g}(\boldsymbol{u}_{pq}^{n,-y})\right)$$

$$\partial_t \boldsymbol{u}_{pq}^{n} := -\partial_x \boldsymbol{f}_{pq} - \partial_y \boldsymbol{g}_{pq}$$

The evolutions to time level $n+\frac{1}{2}$ are given by

$$\boldsymbol{u}_{pq}^{n+\frac{1}{2},\pm x} = \boldsymbol{u}_{pq}^{n,\pm x} + \frac{\Delta t}{2}\,\partial_t \boldsymbol{u}_{pq}^{n}, \qquad \boldsymbol{u}_{pq}^{n+\frac{1}{2},\pm y} = \boldsymbol{u}_{pq}^{n,\pm y} + \frac{\Delta t}{2}\,\partial_t \boldsymbol{u}_{pq}^{n} \tag{G.23}$$

and then the final update is performed as

$$\boldsymbol{u}_{pq}^{n+1} = \boldsymbol{u}_{pq}^{n} - \frac{\Delta t}{\Delta x_p}\left(\boldsymbol{f}_{p+\frac{1}{2},q}^{n+\frac{1}{2}} - \boldsymbol{f}_{p-\frac{1}{2},q}^{n+\frac{1}{2}}\right) - \frac{\Delta t}{\Delta y_q}\left(\boldsymbol{g}_{p,q+\frac{1}{2}}^{n+\frac{1}{2}} - \boldsymbol{g}_{p,q-\frac{1}{2}}^{n+\frac{1}{2}}\right) \tag{G.24}$$

where the numerical fluxes are computed as

$$\boldsymbol{f}_{p+\frac{1}{2},q}^{n+\frac{1}{2}} = \boldsymbol{f}(\boldsymbol{u}_{pq}^{n+\frac{1}{2},+x}, \boldsymbol{u}_{p+1,q}^{n+\frac{1}{2},-x}), \qquad \boldsymbol{g}_{p,q+\frac{1}{2}}^{n+\frac{1}{2}} = \boldsymbol{g}(\boldsymbol{u}_{pq}^{n+\frac{1}{2},+y}, \boldsymbol{u}_{p,q+1}^{n+\frac{1}{2},-y})$$

G.6.1. First evolution step

As in 1-D, define $\boldsymbol{u}_{pq}^{*,\pm x}, \boldsymbol{u}_{pq}^{*,\pm y}$ so that

$$\begin{aligned} \mu_{+x}\,\boldsymbol{u}^{n,+x} + \boldsymbol{u}_{pq}^{*,\pm x} + \mu_{-x}\,\boldsymbol{u}_{pq}^{n,-x} &= 2\,\boldsymbol{u}_{pq}^{n,\pm x} \\ \mu_{+y}\,\boldsymbol{u}_{pq}^{n,+y} + \boldsymbol{u}_{pq}^{*,\pm y} + \mu_{-y}\,\boldsymbol{u}_{pq}^{n,-y} &= 2\,\boldsymbol{u}_{pq}^{n,\pm y} \end{aligned} \tag{G.25}$$

where

$$\begin{aligned} \mu_{+x} &= \frac{x_p - x_{p-\frac{1}{2}}}{x_{p+\frac{1}{2}} - x_{p-\frac{1}{2}}}, \qquad \mu_{-x} = \frac{x_{p+\frac{1}{2}} - x_p}{x_{p+\frac{1}{2}} - x_{p-\frac{1}{2}}} \\ \mu_{+y} &= \frac{y_q - y_{q-\frac{1}{2}}}{y_{q+\frac{1}{2}} - y_{q-\frac{1}{2}}}, \qquad \mu_{-y} = \frac{y_{q+\frac{1}{2}} - y_q}{y_{q+\frac{1}{2}} - y_{q-\frac{1}{2}}} \end{aligned} \tag{G.26}$$

Since we assume conservative reconstruction

$$\mu_{+x}\,\boldsymbol{u}_{pq}^{n,+x} + \mu_{-x}\,\boldsymbol{u}_{pq}^{n,-x} = \mu_{+y}\,\boldsymbol{u}_{pq}^{n,+y} + \mu_{-y}\,\boldsymbol{u}_{pq}^{n,-y} = \boldsymbol{u}_{pq}^{n}$$

Thus, we have

$$\boldsymbol{u}_{pq}^{*,\pm x} = \boldsymbol{u}_{pq} + 2\,(x_{p\pm\frac{1}{2}} - x_p)\,\boldsymbol{\delta}_p^x, \qquad \boldsymbol{u}_{pq}^{*,\pm y} = \boldsymbol{u}_{pq} + 2\,(y_{q\pm\frac{1}{2}} - x_q)\,\boldsymbol{\delta}_q^y$$

We will particularly discuss admissibility of the updates

$$\boldsymbol{u}_{pq}^{n+\frac{1}{2},+x} = \boldsymbol{u}_{pq}^{n,+x} - \frac{\Delta t/2}{\Delta x_p}\left(\boldsymbol{f}(\boldsymbol{u}_{pq}^{n,+x}) - \boldsymbol{f}(\boldsymbol{u}_{pq}^{n,-x})\right) - \frac{\Delta t/2}{\Delta y_q}\left(\boldsymbol{g}(\boldsymbol{u}_{pq}^{n,+y}) - \boldsymbol{g}(\boldsymbol{u}_{pq}^{n,-y})\right) \tag{G.27}$$

Admissibility of the other three updates $\boldsymbol{u}_{pq}^{n+\frac{1}{2},-x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},\pm y}$ will follow similarly. For some k_x, k_y chosen such that $k_x + k_y = 1$, we write (G.27) as

$$\boldsymbol{u}_{pq}^{n+\frac{1}{2},+x} = k_x\,\boldsymbol{\theta}_{pq}^{+x} + k_y\,\boldsymbol{\theta}_{pq}^{+y}$$

where

$$\boldsymbol{\theta}_{pq}^{+x} := \boldsymbol{u}_{pq}^{n,+x} - \frac{\Delta t/2}{k_x \Delta x_p} \left(\boldsymbol{f}(\boldsymbol{u}_{pq}^{n,+x}) - \boldsymbol{f}(\boldsymbol{u}_{pq}^{n,-x}) \right) \tag{G.28}$$

and

$$\boldsymbol{\theta}_{pq}^{+y} := \boldsymbol{u}_{pq}^{n,+x} - \frac{\Delta t/2}{k_y \Delta y_q} \left(\boldsymbol{g}(\boldsymbol{u}_{pq}^{n,+y}) - \boldsymbol{g}(\boldsymbol{u}_{pq}^{n,-y}) \right) \tag{G.29}$$

We will choose the slopes $\boldsymbol{\delta}_p^x, \boldsymbol{\delta}_q^y$ and time step Δt so that $\boldsymbol{\theta}_{pq}^{+x}, \boldsymbol{\theta}_{pq}^{+y} \in \mathcal{U}_{\text{ad}}$. Then, we can take convex combinations of the two terms to obtain admissibility of $\boldsymbol{u}_{pq}^{n+\frac{1}{2},+x}$.

Remark G.10. The choice of k_x, k_y will not influence the slope restriction, but only the time step restriction required to obtain admissibility. In this work, for Cartesian meshes, we compute the time step size using (4.30) with CFL number dictated by Fourier stability analysis (Table 4.1). With this restriction, we observe admissibility preservation in all our numerical experiments even with the trivial choice of $k_x = k_y = 1/2$. Thus, we do not study the choice of k_x, k_y in this work. However, in a similar context, [205] proposed the choice of

$$k_x = \frac{a_x/\Delta x_p}{a_x/\Delta x_p + a_y/\Delta y_q}, \qquad k_y = \frac{a_y/\Delta y_q}{a_x/\Delta x_p + a_y/\Delta y_q} \tag{G.30}$$

where

$$a_x = \sigma_x(\boldsymbol{u}_{pq}^{n,-x}, \boldsymbol{u}_{pq}^{n,+x}), \qquad a_y = \sigma_y(\boldsymbol{u}_{pq}^{n,-y}, \boldsymbol{u}_{pq}^{n,+y})$$

In [56], it was shown that the time step restriction imposed by the above decomposition is suboptimal and optimal decompositions were proposed.

After choosing k_x, k_y (Remark G.10), following the 1-D procedures from Section G.4, the slopes $\boldsymbol{\delta}_p^x, \boldsymbol{\delta}_q^y$ will be limited to enforce admissibility of $\boldsymbol{\theta}_{pq}^{+x}, \boldsymbol{\theta}_{pq}^{+y}$ (G.28, G.29). The admissibility preservation of $\boldsymbol{\theta}_{pq}^{+x}$ (G.28) follows directly from the arguments used in Lemma G.2, enforcing slope restriction so that $\boldsymbol{u}_{pq}^{n,\pm x}$ and $\boldsymbol{u}_{pq}^{*,+x}$ are admissible, and appropriate time step restrictions. For admissibility of $\boldsymbol{\theta}_{pq}^{+y}$ (G.29), we define $\boldsymbol{u}_{pq}^{*,+xy}$ so that

$$\mu_{+y} \boldsymbol{u}_{pq}^{n,+y} + \boldsymbol{u}_{pq}^{*,+xy} + \mu_{-y} \boldsymbol{u}_{pq}^{n,-y} = 2\, \boldsymbol{u}_{pq}^{n,+x}$$

Thus, the proof of Lemma G.2 shall apply as in 1-D under the assumption of admissibility of $\boldsymbol{u}_{pq}^{n,\pm y}, \boldsymbol{u}_{pq}^{*,+xy}$ and some CFL conditions. Thus, we will have admissibility of $\boldsymbol{\theta}_{pq}^{+y} \in \mathcal{U}_{\text{ad}}$. We obtain further simplifications because of conservative reconstructions

$$\boldsymbol{u}_{pq}^{*,+xy} = \boldsymbol{u}_{pq}^{*,+x}$$

and thus the slope limiting for enforcing admissibility of $\boldsymbol{u}_{pq}^{*,+x}$ will suffice. We note the precise slope and CFL restrictions are in Lemma G.11.

LEMMA G.11. *For $\mu_{\pm x}, \mu_{\pm y}$ defined in (G.26), $\boldsymbol{u}_{pq}^{n,\pm x}, \boldsymbol{u}_{pq}^{n,\pm y}$ reconstructed in (G.22), $\boldsymbol{u}_{pq}^{*,\pm x}, \boldsymbol{u}_{pq}^{*,\pm y}$ picked as in (G.25), assume*

$$\boldsymbol{u}_{pq}^{n,\pm x}, \boldsymbol{u}_{pq}^{n,\pm y}, \boldsymbol{u}_{pq}^{*,\pm x}, \boldsymbol{u}_{pq}^{*,\pm y} \in \mathcal{U}_{\text{ad}}$$

and the CFL restrictions

$$\begin{aligned}
&\max_{p,q} \frac{\lambda_{x_p}}{\mu_{-x}} \sigma_x(\boldsymbol{u}_{pq}^{n,-x}, \boldsymbol{u}_{pq}^{*,\pm x}) \le 1, \qquad &&\max_{p,q} \frac{\lambda_{x_p}}{\mu_{+x}} \sigma_x(\boldsymbol{u}_{pq}^{*,\pm x}, \boldsymbol{u}^{n,+x}) \le 1 \\
&\max_{p,q} \frac{\lambda_{y_q}}{\mu_{-y}} \sigma_y(\boldsymbol{u}_{pq}^{n,-y}, \boldsymbol{u}_{pq}^{*,\pm x}) \le 1, \qquad &&\max_{p,q} \frac{\lambda_{y_q}}{\mu_{+y}} \sigma_y(\boldsymbol{u}_{pq}^{*,\pm x}, \boldsymbol{u}^{n,+y}) \le 1 \\
&\max_{p,q} \frac{\lambda_{y_q}}{\mu_{-y}} \sigma_y(\boldsymbol{u}_{pq}^{n,-y}, \boldsymbol{u}_{pq}^{*,\pm y}) \le 1, \qquad &&\max_{p,q} \frac{\lambda_{y_q}}{\mu_{+y}} \sigma_y(\boldsymbol{u}_{pq}^{*,\pm y}, \boldsymbol{u}_{pq}^{n,+y}) \le 1 \\
&\max_{p,q} \frac{\lambda_{x_p}}{\mu_{-x}} \sigma_x(\boldsymbol{u}_{pq}^{n,-x}, \boldsymbol{u}_{pq}^{*,\pm y}) \le 1, \qquad &&\max_{p,q} \frac{\lambda_{x_p}}{\mu_{+x}} \sigma_x(\boldsymbol{u}_{pq}^{*,\pm y}, \boldsymbol{u}_{pq}^{n,+x}) \le 1
\end{aligned} \tag{G.31}$$

where $\lambda_{x_p} = \frac{\Delta t}{k_x \Delta x_p}, \lambda_{y_q} = \frac{\Delta t}{k_y \Delta y_q}$ *for all* p, q *and* $k_x + k_y = 1$. *Then, the updates* $\boldsymbol{u}_{pq}^{n+\frac{1}{2},\pm x}$, $\boldsymbol{u}_{pq}^{n+\frac{1}{2},\pm y}$ (G.27) *of the first step of 2-D MUSCL-Hancock scheme are admissible.*

G.6.2. Finite volume step

The final update is given by

$$\boldsymbol{u}_{pq}^{n+1} = \boldsymbol{u}_{pq}^{n} - \frac{\Delta t}{\Delta x_p} (\boldsymbol{f}_{p+\frac{1}{2},q}^{n+\frac{1}{2}} - \boldsymbol{f}_{p-\frac{1}{2},q}^{n+\frac{1}{2}}) - \frac{\Delta t}{\Delta y_q} (\boldsymbol{g}_{p,q+\frac{1}{2}}^{n+\frac{1}{2}} - \boldsymbol{g}_{p,q-\frac{1}{2}}^{n+\frac{1}{2}}) \tag{G.32}$$

where the numerical fluxes are computed as

$$\boldsymbol{f}_{p+\frac{1}{2},q}^{n+\frac{1}{2}} = \boldsymbol{f}\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},+x}, \boldsymbol{u}_{p+1,q}^{n+\frac{1}{2},-x}\Big), \qquad \boldsymbol{g}_{p,q+\frac{1}{2}}^{n+\frac{1}{2}} = \boldsymbol{g}\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},+y}, \boldsymbol{u}_{p,q+1}^{n+\frac{1}{2},-y}\Big)$$

As in the previous step, the expression (G.32) is split into a convex combination

$$\boldsymbol{u}_{pq}^{n+1} = k_x \boldsymbol{\zeta}_{pq}^{x} + k_y \boldsymbol{\zeta}_{pq}^{y}$$

where

$$\boldsymbol{\zeta}_{pq}^{x} := \boldsymbol{u}_{pq}^{n} - \frac{\Delta t}{k_x \Delta x_p} (\boldsymbol{f}_{p+\frac{1}{2},q}^{n+\frac{1}{2}} - \boldsymbol{f}_{p-\frac{1}{2},q}^{n+\frac{1}{2}}), \qquad \boldsymbol{\zeta}_{pq}^{y} := \boldsymbol{u}_{pq}^{n} - \frac{\Delta t}{k_y \Delta y_q} (\boldsymbol{g}_{p,q+\frac{1}{2}}^{n+\frac{1}{2}} - \boldsymbol{g}_{p,q-\frac{1}{2}}^{n+\frac{1}{2}})$$

for some $k_x, k_y \ge 0$ with $k_x + k_y = 1$. The admissibility of $\boldsymbol{\zeta}_{pq}^{x}$ and $\boldsymbol{\zeta}_{pq}^{y}$ will imply the admissibility of $\boldsymbol{u}^{n+1}$. The admissibility of $\boldsymbol{\zeta}_{pq}^{x}, \boldsymbol{\zeta}_{pq}^{y}$ will follow exactly as from the procedure in 1-D (Lemma G.3) with appropriate time step restrictions and assumption of admissibility of terms $\boldsymbol{u}_{pq}^{n+\frac{1}{2},\pm x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},\pm y}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},*y}$ for $\boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},*y}$ defined as

$$\begin{aligned}
\mu_{-x} \boldsymbol{u}_{pq}^{n+\frac{1}{2},-x} + \boldsymbol{u}_{pq}^{n+\frac{1}{2},*x} + \mu_{+x} \boldsymbol{u}_{pq}^{n+\frac{1}{2},+x} &= 2\, \boldsymbol{u}_{pq}^{n} \\
\mu_{-y} \boldsymbol{u}_{pq}^{n+\frac{1}{2},-y} + \boldsymbol{u}_{pq}^{n+\frac{1}{2},*y} + \mu_{+y} \boldsymbol{u}_{pq}^{n+\frac{1}{2},+y} &= 2\, \boldsymbol{u}_{pq}^{n}
\end{aligned}$$

The precise CFL restrictions and admissibility constraints are in the following Lemma G.12.

LEMMA G.12. *Assume that the states* $\left\{\boldsymbol{u}_{pq}^{n+\frac{1}{2},\pm x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},\pm y}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},*y}\right\}_{p,q}$ *belong to* $\mathcal{U}_{\mathrm{ad}}$*, where* $\boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},*y}$ *are defined as in* (G.32)*. Then, the updated solution* $\boldsymbol{u}_{pq}^{n+1}$ *of MUSCL-Hancock scheme is in* $\mathcal{U}_{\mathrm{ad}}$ *under the CFL conditions*

$$
\begin{aligned}
&\frac{2\,\lambda_{x_p}}{\mu_{-x}}\,\sigma_x\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},-x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}\Big) \le 1, \quad 2\,\lambda_{x_p}\,\sigma_x\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},+x}\Big) \le 1\\
&\frac{2\,\lambda_{x_p}}{\mu_{+x}}\,\sigma_x\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},+x}, \boldsymbol{u}_{p+1,q}^{n+\frac{1}{2},-x}\Big) \le 1, \quad \frac{2\,\lambda_{x_p}}{\mu_{-x}}\,\sigma_x\Big(\boldsymbol{u}_{p-1,q}^{n+\frac{1}{2},+x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},-x}\Big) \le 1\\
&2\,\lambda_{x_p}\,\sigma_x\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},-x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}\Big) \le 1, \quad \frac{2\,\lambda_{x_p}}{\mu_{+x}}\,\sigma_x\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},+x}\Big) \le 1\\
&\frac{2\,\lambda_{x_p}}{\mu_{-x}}\,\sigma_x\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},-x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}\Big) \le 1, \quad 2\,\lambda_{x_p}\,\sigma_x\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},+x}\Big) \le 1\\
&\frac{2\,\lambda_{x_p}}{\mu_{+x}}\,\sigma_x\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},+x}, \boldsymbol{u}_{p+1,q}^{n+\frac{1}{2},-x}\Big) \le 1, \quad \frac{2\,\lambda_{x_p}}{\mu_{-x}}\,\sigma_x\Big(\boldsymbol{u}_{p-1,q}^{n+\frac{1}{2},+x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},-x}\Big) \le 1\\
&2\,\lambda_{x_p}\,\sigma_x\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},-x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}\Big) \le 1, \quad \frac{2\,\lambda_{x_p}}{\mu_{+x}}\,\sigma_x\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},+x}\Big) \le 1\\
&\frac{2\,\lambda_{y_q}}{\mu_{-y}}\,\sigma_y\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},-y}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},*y}\Big) \le 1, \quad 2\,\lambda_{y_q}\,\sigma_y\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},*y}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},+y}\Big) \le 1\\
&\frac{2\,\lambda_{y_q}}{\mu_{+y}}\,\sigma_y\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},+y}, \boldsymbol{u}_{p,q+1}^{n+\frac{1}{2},-y}\Big) \le 1, \quad \frac{2\,\lambda_{y_q}}{\mu_{-y}}\,\sigma_y\Big(\boldsymbol{u}_{p,q-1}^{n+\frac{1}{2},+y}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},-y}\Big) \le 1\\
&2\,\lambda_{y_q}\,\sigma_y\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},-y}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},*y}\Big) \le 1, \quad \frac{2\,\lambda_{y_q}}{\mu_{+y}}\,\sigma_y\Big(\boldsymbol{u}_{pq}^{n+\frac{1}{2},*y}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},+y}\Big) \le 1
\end{aligned}
\tag{G.33}
$$

where $\lambda_{x_p} = \frac{\Delta t}{k_x\,\Delta x_p}, \lambda_{y_q} = \frac{\Delta t}{k_y\,\Delta y_q}$ *for all* p, q.

As in 1-D, we now show that admissibility of $\boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},*y}$ can also be reduced to admissibility of $\boldsymbol{u}_{pq}^{*,\pm x}, \boldsymbol{u}_{pq}^{*,\pm y}$, similar to Lemma G.5. Expanding the definition of $\boldsymbol{u}_{pq}^{n+\frac{1}{2},*y}$ gives us

$$
\begin{aligned}
\boldsymbol{u}_{pq}^{n+\frac{1}{2},*y} \;=\; & 2\,\boldsymbol{u}_{pq}^{n} - (\mu_{-y}\,\boldsymbol{u}_{pq}^{n,-y} + \mu_{+y}\,\boldsymbol{u}_{pq}^{n,+y}) - \frac{\Delta t}{\Delta x_p}\,(\boldsymbol{f}(\boldsymbol{u}_{pq}^{n,-x}) - \boldsymbol{f}(\boldsymbol{u}_{pq}^{n,+x}))\\
& - \frac{\Delta t}{\Delta y_q}\,(\boldsymbol{g}(\boldsymbol{u}_{pq}^{n,-y}) - \boldsymbol{g}(\boldsymbol{u}_{pq}^{n,+y}))
\end{aligned}
$$

If we obtain the admissibility of

$$
\boldsymbol{\eta}_{pq}^{*yx} := 2\,\boldsymbol{u}_{pq}^{n} - (\mu_{-y}\,\boldsymbol{u}_{pq}^{n,-y} + \mu_{+y}\,\boldsymbol{u}_{pq}^{n,+y}) - \frac{\Delta t}{k_x\,\Delta x_p}\,(\boldsymbol{f}(\boldsymbol{u}_{pq}^{n,-x}) - \boldsymbol{f}(\boldsymbol{u}_{pq}^{n,+x})) \tag{G.34}
$$

and

$$
\boldsymbol{\eta}_{pq}^{*yy} := 2\,\boldsymbol{u}_{pq}^{n} - (\mu_{-y}\,\boldsymbol{u}_{pq}^{n,-y} + \mu_{+y}\,\boldsymbol{u}_{pq}^{n,+y}) - \frac{\Delta t}{k_y\,\Delta y_q}\,(\boldsymbol{g}(\boldsymbol{u}_{pq}^{n,-y}) - \boldsymbol{g}(\boldsymbol{u}_{pq}^{n,+y})) \tag{G.35}
$$

for some $k_x, k_y \in [0,1]$ with $k_x + k_y = 1$, then the admissibility of $\boldsymbol{u}_{pq}^{n+\frac{1}{2},*y}$ follows as we can write it as a convex combination

$$
\boldsymbol{u}_{pq}^{n+\frac{1}{2},*y} = k_x\,\boldsymbol{\eta}_{pq}^{*yx} + k_y\,\boldsymbol{\eta}_{pq}^{*yx}
$$

and obtain the admissibility of $\boldsymbol{u}_{pq}^{n+\frac{1}{2},*y}$. Thus, we need to limit the slope so that (G.34, G.35) are admissible. To that end, define $\boldsymbol{u}_{pq}^{**yx}, \boldsymbol{u}_{pq}^{**yy}$ to satisfy

$$\mu_{-x}\,\boldsymbol{u}_{pq}^{n,-x} + \boldsymbol{u}_{pq}^{**yx} + \mu_{+x}\,\boldsymbol{u}_{pq}^{n,+x} = 2\,(2\,\boldsymbol{u}_{pq}^{n} - (\mu_{-y}\,\boldsymbol{u}_{pq}^{n,-y} + \mu_{+y}\,\boldsymbol{u}_{pq}^{n,+y}))$$
$$\mu_{-y}\,\boldsymbol{u}_{pq}^{n,-y} + \boldsymbol{u}_{pq}^{**yy} + \mu_{+y}\,\boldsymbol{u}_{pq}^{n,+y} = 2\,(2\,\boldsymbol{u}_{pq}^{n} - (\mu_{-y}\,\boldsymbol{u}_{pq}^{n,-y} + \mu_{+y}\,\boldsymbol{u}_{pq}^{n,+y}))$$

respectively. Consequently,

$$\boldsymbol{\eta}_{pq}^{*yx} = \frac{1}{2}\,(\mu_{-x}\,\boldsymbol{u}_{pq}^{n,-x} + \boldsymbol{u}_{pq}^{**yx} + \mu_{+x}\,\boldsymbol{u}_{pq}^{n,+x}) - \frac{\Delta t}{k_x\,\Delta x_p}\,(\boldsymbol{f}(\boldsymbol{u}_{pq}^{n,-x}) - \boldsymbol{f}(\boldsymbol{u}_{pq}^{n,+x}))$$
$$\boldsymbol{\eta}_{pq}^{*yy} = \frac{1}{2}\,(\mu_{-y}\,\boldsymbol{u}_{pq}^{n,-y} + \boldsymbol{u}_{pq}^{**yy} + \mu_{+y}\,\boldsymbol{u}_{pq}^{n,+y}) - \frac{\Delta t}{k_y\,\Delta y_q}\,(\boldsymbol{g}(\boldsymbol{u}_{pq}^{n,-y}) - \boldsymbol{g}(\boldsymbol{u}_{pq}^{n,+y}))$$

Then, assuming the admissibility of $\boldsymbol{u}_{pq}^{**yx}, \boldsymbol{u}_{pq}^{**yy}$ and proceeding as in the proof of Lemma G.4, we can ensure that $\boldsymbol{\eta}_{pq}^{*yx}, \boldsymbol{\eta}_{pq}^{*yy} \in \mathcal{U}_{\mathrm{ad}}$ and thus $\boldsymbol{u}_{pq}^{n+\frac{1}{2},*y} \in \mathcal{U}_{\mathrm{ad}}$. Furthermore, since the reconstruction is conservative

$$\mu_{-y}\,\boldsymbol{u}_{pq}^{n,-y} + \mu_{+y}\,\boldsymbol{u}_{pq}^{n,+y} = \mu_{-x}\,\boldsymbol{u}_{pq}^{n,-x} + \mu_{+x}\,\boldsymbol{u}_{pq}^{n,+x} = \boldsymbol{u}_{pq}^{n}$$

Thus, admissibility of $\boldsymbol{u}_{pq}^{**yx}, \boldsymbol{u}_{pq}^{**yy}$ is obtained as

$$\boldsymbol{u}_{pq}^{**yx} = \boldsymbol{u}_{pq}^{**yy} = \boldsymbol{u}_{pq}^{n}$$

The arguments for admissibility of $\boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}$ are similar. The admissibility criteria of $\boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}, \boldsymbol{u}_{pq}^{n+\frac{1}{2},*y}$ are summarised in the following lemma.

Lemma G.13. *Assume that $\boldsymbol{u}_{pq}^{n} \in \mathcal{U}_{\mathrm{ad}}$ and $\boldsymbol{u}_{pq}^{n,\pm x}, \boldsymbol{u}_{pq}^{n,\pm y} \in \mathcal{U}_{\mathrm{ad}}$ for all p, q with conservative reconstruction. Also assume the CFL restrictions*

$$\begin{aligned} &\max_{p,q} \frac{\lambda_{x_p}}{\mu_{-x}}\,\sigma_x(\boldsymbol{u}_{pq}^{n}, \boldsymbol{u}_{pq}^{n,-x}) \le 1, \qquad &&\max_{p,q} \frac{\lambda_{x_p}}{\mu_{+x}}\,\sigma_x(\boldsymbol{u}_{pq}^{n,+x}, \boldsymbol{u}_{pq}^{n}) \le 1 \\ &\max_{p,q} \frac{\lambda_{y_q}}{\mu_{-y}}\,\sigma_y(\boldsymbol{u}_{pq}^{n}, \boldsymbol{u}_{pq}^{n,-y}) \le 1, \qquad &&\max_{p,q} \frac{\lambda_{y_q}}{\mu_{+y}}\,\sigma_y(\boldsymbol{u}_{pq}^{n,+y}, \boldsymbol{u}_{pq}^{n}) \le 1 \end{aligned} \tag{G.36}$$

where $\lambda_{x_p} = \frac{\Delta t}{k_x\,\Delta x_p}$, $\lambda_{y_q} = \frac{\Delta t}{k_y\,\Delta y_q}$ and $\mu_{\pm x}, \mu_{\pm y}$ are defined in (G.26). Then, $\boldsymbol{u}_{pq}^{n+\frac{1}{2},*x}$, $\boldsymbol{u}_{pq}^{n+\frac{1}{2},*y}$ defined in (G.32) are in $\mathcal{U}_{\mathrm{ad}}$.

Combining Lemmas G.11, G.12, G.13, we will have the 2-D result corresponding to Theorem G.6 with the same proof.

Theorem G.14. *Let $\boldsymbol{u}_{pq}^{n} \in \mathcal{U}_{\mathrm{ad}}$ for all p, q and $\boldsymbol{u}_{pq}^{n,\pm x}, \boldsymbol{u}_{pq}^{n,\pm y}$ be the conservative reconstructions defined as*

$$\boldsymbol{u}_{pq}^{n,\pm x} = \boldsymbol{u}_{pq}^{n} + (x_{p\pm\frac{1}{2}} - x_p)\,\boldsymbol{\delta}_p^x, \qquad \boldsymbol{u}_{pq}^{n,\pm y} = \boldsymbol{u}_{pq}^{n} + (y_{q\pm\frac{1}{2}} - y_q)\,\boldsymbol{\delta}_q^y$$

so that $\boldsymbol{u}_{pq}^{*,\pm x}, \boldsymbol{u}_{pq}^{*,\pm y}$ (G.25) are given by

$$\boldsymbol{u}_{pq}^{*,\pm x} = \boldsymbol{u}_{pq}^{n} + 2\,(x_{p\pm\frac{1}{2}} - x_p)\,\boldsymbol{\delta}_p^x, \qquad \boldsymbol{u}_{pq}^{*,\pm y} = \boldsymbol{u}_{pq}^{n} + 2\,(y_{q\pm\frac{1}{2}} - y_q)\,\boldsymbol{\delta}_q^y$$

Assume that the slopes $\boldsymbol{\delta}_p^x, \boldsymbol{\delta}_q^y$ are chosen to satisfy $\boldsymbol{u}_{pq}^{*,\pm x}, \boldsymbol{u}_{pq}^{*,\pm y} \in \mathcal{U}_{\mathrm{ad}}$ for all p, q and that the CFL restrictions (G.31, G.33, G.36) are satisfied. Then the updated solution $\boldsymbol{u}_{pq}^{n+1}$ of MUSCL-Hancock procedure is in $\mathcal{U}_{\mathrm{ad}}$.

Appendix H

Limiting numerical flux in 2-D

Consider the 2-D hyperbolic conservation law (G.1). Following Section 4.9, the Lax-Wendroff update is

$$(\boldsymbol{u}_{pq}^{e})^{n+1} = (\boldsymbol{u}_{pq}^{e})^{n} - \Delta t \left[\frac{1}{\Delta x_e} \frac{\partial \boldsymbol{F}_h^{\mathbf{e}}}{\partial \xi}(\xi_p, \xi_q) + \frac{1}{\Delta y_e} \frac{\partial \boldsymbol{G}_h^{\mathbf{e}}}{\partial \eta}(\xi_p, \xi_q) \right], \qquad 0 \leq p, q \leq N$$

where $\boldsymbol{F}_h^{\mathbf{e}}, \boldsymbol{G}_h^{\mathbf{e}}$ are continuous time-averaged fluxes (4.8) in the x, y directions for the grid element $\mathbf{e}=(e_x, e_y)$. Since the 2-D scheme is formed by taking a tensor product of the 1-D scheme, Theorem 5.5 applies, i.e., the scheme will be admissibility preserving in means (Definition 5.2) if we choose the blended numerical flux such that the lower order updates are admissible at solution points adjacent to the interfaces. Thus, we now explain the process of constructing the numerical flux where, to minimize storage requirements and memory reads, we will perform the correction within the interface loop where only one of x or y flux will be available in one iteration. Thus theoretical justification for the algorithm comes from breaking 2-D lower order updates into 1-D convex combinations. The general structure of the LWFR Algorithm 5.2 will remain the same. Here, we justify Algorithm H.1 for construction of blended x flux with knowledge of only the x flux. The algorithm for blended y fluxes will be analogous.

We consider the calculation of the blended numerical flux for a corner solution point of the element, as this situation differs from 1-D, due to the fact that a corner solution point is adjacent to both x and y interfaces. In particular, we consider the bottom-left corner point $\mathbf{0} = (0, 0)$ and show that the procedure in Algorithm H.1 ensures admissibility at such points. The same justification applies to other corner and non-corner points. For the element $\mathbf{e}=(e_x, e_y)$, denoting interfaces along x, y directions as $(e_x \pm \frac{1}{2}, e_y)$, $(e_x, e_y \pm \frac{1}{2})$, we consider the update at the bottom left corner $\mathbf{0} = (0, 0)$, suppressing the local solution point index $p = 0$ or $q = 0$ when considering the FR

interface fluxes. The lower order update is given by

$$\hat{\boldsymbol{u}}_{\mathbf{0}}^{n+1} = \boldsymbol{u}_{e,\mathbf{0}}^{n} - \frac{\Delta t}{\Delta x_e \, w_0} \left(\boldsymbol{f}^{\mathbf{e}}_{(\frac{1}{2},0)} - \hat{\boldsymbol{F}}_{(e_x - \frac{1}{2}, e_y)} \right) - \frac{\Delta t}{\Delta y_e \, w_0} \left(\boldsymbol{g}^{\mathbf{e}}_{(0,\frac{1}{2})} - \hat{\boldsymbol{G}}_{(e_x, e_y - \frac{1}{2})} \right)$$

where $\hat{\boldsymbol{F}}_{(e_x-\frac{1}{2},e_y)}, \hat{\boldsymbol{G}}_{(e_x,e_y-\frac{1}{2})}$ are heuristically guessed candidates for the blended numerical flux (5.11). Pick $k_x, k_y > 0$ such that $k_x + k_y = 1$ and

$$\begin{aligned} \hat{\boldsymbol{u}}_x^{\text{low},n+1} &= (\boldsymbol{u}^{\mathbf{e}}_{\mathbf{0}})^n - \frac{\Delta t}{k_x \, \Delta x_e \, w_0} \left(\boldsymbol{f}^{\mathbf{e}}_{(\frac{1}{2},0)} - \boldsymbol{f}_{(e_x - \frac{1}{2}, e_y)} \right) \\ \hat{\boldsymbol{u}}_y^{\text{low},n+1} &= (\boldsymbol{u}^{\mathbf{e}}_{\mathbf{0}})^n - \frac{\Delta t}{k_y \, \Delta y_e \, w_0} \left(\boldsymbol{g}^{\mathbf{e}}_{(0,\frac{1}{2})} - \boldsymbol{g}_{(e_x, e_y - \frac{1}{2})} \right) \end{aligned} \tag{H.1}$$

satisfy

$$\hat{\boldsymbol{u}}_x^{\text{low},n+1}, \hat{\boldsymbol{u}}_y^{\text{low},n+1} \in \mathcal{U}_{\text{ad}} \tag{H.2}$$

Such k_x, k_y exist because the lower order scheme with lower order flux at element interfaces is admissibility preserving. The choice of k_x, k_y should be made so that (H.2) is satisfied with the least time step restriction, but we have found the Fourier stability restriction imposed by (4.30) to be sufficient even with the most trivial choice of $k_x = k_y = \frac{1}{2}$. The discussion of literature for the optimal choice of k_x, k_y is the same as the one made for the 2-D MUSCL Hancock scheme (G.30) and is not repeated here. After the choice of k_x, k_y is made, if we repeat the same procedure as in the 1-D case, we can perform slope limiting to find $\boldsymbol{F}_{e_x-\frac{1}{2},e_y}$, $\boldsymbol{F}_{e_x,e_y-\frac{1}{2}}$ such that

$$\begin{aligned} \hat{\boldsymbol{u}}_x^{n+1} &= \boldsymbol{u}_{e,\mathbf{0}}^{n} - \frac{\Delta t}{k_x \, \Delta x_e \, w_0} \left(\boldsymbol{f}^{\mathbf{e}}_{(\frac{1}{2},0)} - \boldsymbol{F}_{(e_x - \frac{1}{2}, e_y)} \right) \\ \hat{\boldsymbol{u}}_y^{n+1} &= \boldsymbol{u}_{e,\mathbf{0}}^{n} - \frac{\Delta t}{k_y \, \Delta y_e \, w_0} \left(\boldsymbol{g}^{\mathbf{e}}_{(0,\frac{1}{2})}, - \boldsymbol{G}_{(e_x, e_y - \frac{1}{2})} \right) \end{aligned}$$

are also in the admissible region. Then, we will get

$$k_x \, \hat{\boldsymbol{u}}_x^{n+1} + k_y \, \hat{\boldsymbol{u}}_y^{n+1} = \hat{\boldsymbol{u}}_{\mathbf{0}}^{n+1} \tag{H.3}$$

We now justify Algorithm H.1 as follows. Algorithm H.1 corrects the numerical fluxes during the loop over x interfaces to enforce admissibility of $\hat{\boldsymbol{u}}_x^{n+1}$ (H.2) at all solution points neighbouring x interfaces including the corner solution points, and the analogous algorithm for y interfaces will ensure admissibility of $\hat{\boldsymbol{u}}_y^{n+1}$ (H.2) at all solution points neighbouring y interfaces including the corner points. At the end of the loop over interfaces, (H.3) will ensure that lower order updates at all solutions points neighbouring interfaces are admissible and Algorithm 5.2 will be an admissibility preserving Lax-Wendroff scheme for 2-D if we compute the blended numerical fluxes $\boldsymbol{F}_{(e_x+\frac{1}{2},e_y)}$, $\boldsymbol{F}_{(e_x,e_y+\frac{1}{2})}$ using Algorithm H.1 and its counterpart in the y direction.

Algorithm H.1

Computation of blended flux $\boldsymbol{F}_{e_x+\frac{1}{2},e_y,q}$ where $(e_x+\frac{1}{2},e_y)$ are the interface indices and $q\in\{0,\dots,N\}$ is the solution point index on the interface

Input: $\boldsymbol{F}^{\mathrm{LW}}_{e_x+\frac{1}{2},e_y,q}$, $\boldsymbol{f}_{e_x+\frac{1}{2},e_y,q}$, $\boldsymbol{f}^{e_x+1,e_y}_{\frac{1}{2},q}$, $\boldsymbol{f}^{\mathbf{e}}_{N-\frac{1}{2},q}$, $\boldsymbol{u}^{e_x+1,e_y}_{(0,q)}$, $\boldsymbol{u}^{\mathbf{e}}_{(0,q)}$, $\alpha_{\mathbf{e}}$, α_{e_x+1,e_y}, $k_x^{e_x,e_y}$, $k_x^{e_x+1,e_y}$

Output: $\boldsymbol{F}_{e_x+\frac{1}{2},e_y,q}$

$\bar{\alpha}=\dfrac{\alpha_{e_x,e_y}+\alpha_{e_x+1,e_y}}{2}$

$k_x^0, k_x^N = k_x^{e_x,e_y}, k_x^{e_x+1,e_y}$ ▷For ease of writing

$\boldsymbol{F}_{e_x+\frac{1}{2},e_y,q}\leftarrow(1-\bar{\alpha})\,\boldsymbol{F}^{\mathrm{LW}}_{e_x+\frac{1}{2},e_y,q}+\bar{\alpha}\,\boldsymbol{f}_{e_x+\frac{1}{2},e_y,q}$ ▷Heuristic guess to control oscillations

▷FV updates with guessed $\boldsymbol{F}_{e_x+\frac{1}{2},e_y,q}$

$\hat{\boldsymbol{u}}_0^{n+1}\leftarrow(\boldsymbol{u}_{0,q}^{e_x+1,e_y})^n-\dfrac{\Delta t}{k_x^0\,w_0\,\Delta x_{e+1}}\,(\boldsymbol{f}^{e_x+1,e_y}_{\frac{1}{2},q}-\boldsymbol{F}_{e_x+\frac{1}{2},e_y,q})$

$\hat{\boldsymbol{u}}_N^{n+1}\leftarrow(\boldsymbol{u}_{N,q}^{e_x,e_y})^n-\dfrac{\Delta t}{k_x^N\,w_N\,\Delta x_e}\,(\boldsymbol{F}_{e_x+\frac{1}{2},e_y,q}-\boldsymbol{f}^{\mathbf{e}}_{(N-\frac{1}{2},q)})$

▷FV inner updates with $\boldsymbol{f}_{e_x+\frac{1}{2},e_y,q}$

$\hat{\boldsymbol{u}}_0^{\mathrm{low},n+1}=(\boldsymbol{u}_{0,q}^{e_x+1,e_y})^n-\dfrac{\Delta t}{k_x^0\,w_0\,\Delta x_{e+1}}\,(\boldsymbol{f}^{e_x+1,e_y}_{\frac{1}{2},q}-\boldsymbol{f}_{e_x+\frac{1}{2},e_y,q})$

$\hat{\boldsymbol{u}}_N^{\mathrm{low},n+1}=(\boldsymbol{u}_{N,q}^{e_x,e_y})^n-\dfrac{\Delta t}{k_x^N\,w_N\,\Delta x_e}\,(\boldsymbol{f}_{e_x+\frac{1}{2},e_y,q}-\boldsymbol{f}^{\mathbf{e}}_{N-\frac{1}{2},q})$

▷Correct $\boldsymbol{F}_{e_x+\frac{1}{2},e_y,q}$ for K admissibility constraints

for $k=1{:}\,K$ **do**

$\epsilon_0,\epsilon_N\leftarrow\frac{1}{10}\,P_k(\hat{\boldsymbol{u}}_0^{\mathrm{low},n+1}),\frac{1}{10}\,P_k(\hat{\boldsymbol{u}}_N^{\mathrm{low},n+1})$

$\theta\leftarrow\min\left(\min_{p=0,N}\left|\dfrac{\epsilon_p-P_k(\hat{\boldsymbol{u}}_p^{n+1})}{P_k(\hat{\boldsymbol{u}}_p^{\mathrm{low},n+1})-P_k(\hat{\boldsymbol{u}}_p^{n+1})}\right|,1\right)$

$\boldsymbol{F}_{e_x+\frac{1}{2},e_y,q}\leftarrow\theta\,\boldsymbol{F}_{e_x+\frac{1}{2},e_y,q}+(1-\theta)\,\boldsymbol{f}_{e_x+\frac{1}{2},e_y,q}$

▷FV inner updates with new $\boldsymbol{F}_{e_x+\frac{1}{2},e_y,q}$

$\hat{\boldsymbol{u}}_0^{n+1}\leftarrow(\boldsymbol{u}_{0,q}^{e_x+1,e_y})^n-\dfrac{\Delta t}{k_x^0\,w_0\,\Delta x_{e+1}}\,(\boldsymbol{f}^{e_x+1,e_y}_{\frac{1}{2},q}-\boldsymbol{F}_{e_x+\frac{1}{2},e_y,q})$

$\hat{\boldsymbol{u}}_N^{n+1}\leftarrow(\boldsymbol{u}_{N,q}^{e_x,e_y})^n-\dfrac{\Delta t}{k_x^N\,w_N\,\Delta x_e}\,(\boldsymbol{F}_{e_x+\frac{1}{2},e_y,q}-\boldsymbol{f}^{\mathbf{e}}_{(N-\frac{1}{2},q)})$

end

Appendix I

Formal accuracy of multi-derivative RK

We consider the system of time dependent equations

$$\boldsymbol{u}_t = \boldsymbol{L}(\boldsymbol{u})$$

which relates to the hyperbolic conservation law (3.1) by formally setting $\boldsymbol{L} = -\boldsymbol{f}(\boldsymbol{u})_x$. Consider a two stage method of the following form.

$$\begin{aligned} \boldsymbol{u}^* &= \boldsymbol{u}^n + \Delta t\, a_{21}\, \boldsymbol{L}(\boldsymbol{u}^n) + \Delta t^2\, \hat{a}_{21}\, \boldsymbol{L}_t(\boldsymbol{u}^n) \\ \boldsymbol{u}^{n+1} &= \boldsymbol{u}^n + \Delta t\, (b_1\, \boldsymbol{L}(\boldsymbol{u}^n) + b_2\, \boldsymbol{L}(\boldsymbol{u}^*)) + \Delta t^2\, (\hat{b}_1\, \partial_t \boldsymbol{L} + \hat{b}_2\, \partial_t \boldsymbol{L}(\boldsymbol{u}^*)) \end{aligned} \tag{I.1}$$

Further note that, we use the Approximate Lax-Wendroff (Section 7.2.3) to approximate $\partial_t \boldsymbol{L}(\boldsymbol{u}^n), \partial_t \boldsymbol{L}(\boldsymbol{u}^*)$ to $O(\Delta t^3)$ accuracy and thus we perform an error analysis of an evolution performed as

$$\boldsymbol{u}^* = \boldsymbol{u}^n + \Delta t\, a_{21}\, \boldsymbol{L}(\boldsymbol{u}^n) + \Delta t^2\, \hat{a}_{21}\, \boldsymbol{L}_t(\boldsymbol{u}^n) + O(\Delta t^5)$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \Delta t\, (b_1\, \boldsymbol{L}(\boldsymbol{u}^n) + b_2\, \boldsymbol{L}(\boldsymbol{u}^*)) + \Delta t^2\, (\hat{b}_1\, \partial_t \boldsymbol{L} + \hat{b}_2\, \partial_t \boldsymbol{L}(\boldsymbol{u}^*)) + O(\Delta t^5) \tag{I.2}$$

Now, note that

$$\begin{aligned} \boldsymbol{u}_{tt} = \boldsymbol{L}_{\boldsymbol{u}}\, \boldsymbol{u}_t = \boldsymbol{L}_{\boldsymbol{u}}\, \boldsymbol{L}, \qquad \boldsymbol{u}_{ttt} = \boldsymbol{L}_{\boldsymbol{uu}}\, \boldsymbol{u}_t^2 + \boldsymbol{L}_{\boldsymbol{u}}\, \boldsymbol{u}_{tt} = \boldsymbol{L}_{\boldsymbol{uu}}\, \boldsymbol{L}^2 + \boldsymbol{L}_{\boldsymbol{u}}^2\, \boldsymbol{L} \\ \boldsymbol{u}_{tttt} = \boldsymbol{L}_{\boldsymbol{uuu}}\, \boldsymbol{u}_t^3 + 3\, \boldsymbol{L}_{\boldsymbol{uu}}\, \boldsymbol{u}_t\, \boldsymbol{u}_{tt} + \boldsymbol{L}_{\boldsymbol{u}}\, \boldsymbol{u}_{ttt} = \boldsymbol{L}_{\boldsymbol{uuu}}\, \boldsymbol{L}^3 + 4\, \boldsymbol{L}_{\boldsymbol{uu}}\, \boldsymbol{L}_{\boldsymbol{u}}\, \boldsymbol{L}^2 + \boldsymbol{L}_{\boldsymbol{u}}^3\, \boldsymbol{L} \end{aligned} \tag{I.3}$$

Starting from $\boldsymbol{u} = \boldsymbol{u}^n$, the exact solution satisfies

$$\boldsymbol{u}^{n+1} = \boldsymbol{u} + \Delta t\, \boldsymbol{u}_t + \frac{\Delta t^2}{2}\, \boldsymbol{u}_{tt} + \frac{\Delta t^3}{6}\, \boldsymbol{u}_{ttt} + \frac{\Delta t^4}{24}\, \boldsymbol{u}_{tttt} + O(\Delta t^5) \tag{I.4}$$

We note the following identities

$$\begin{gathered} \partial_t\, \boldsymbol{L}(\boldsymbol{u}) = \boldsymbol{L}_{\boldsymbol{u}}\, \boldsymbol{u}_t \\ \boldsymbol{u}^* = \boldsymbol{u} + \Delta t\, a_{21}\, \boldsymbol{L} + \Delta t^2\, \hat{a}_{21}\, \boldsymbol{L}_{\boldsymbol{u}}\, \boldsymbol{L} + O(\Delta t^5) \\ \boldsymbol{L}(\boldsymbol{u}^*) = \boldsymbol{L} + \boldsymbol{L}_{\boldsymbol{u}}(\boldsymbol{u}^* - \boldsymbol{u}) + \frac{1}{2}\, \boldsymbol{L}_{\boldsymbol{uu}}\, (\boldsymbol{u}^* - \boldsymbol{u})^2 + \frac{1}{6}\, \boldsymbol{L}_{\boldsymbol{uuu}}\, (\boldsymbol{u}^* - \boldsymbol{u})^3 + O(\Delta t^4) \\ \boldsymbol{L}_{\boldsymbol{u}}(\boldsymbol{u}^*) = \boldsymbol{L}_{\boldsymbol{u}} + \boldsymbol{L}_{\boldsymbol{uu}}\, (\boldsymbol{u}^* - \boldsymbol{u}) + \frac{1}{2}\, \boldsymbol{L}_{\boldsymbol{uuu}}\, (\boldsymbol{u}^* - \boldsymbol{u})^2 + O(\Delta t^3) \\ \partial_t\, \boldsymbol{L}(\boldsymbol{u}^*) = \boldsymbol{L}_{\boldsymbol{u}}(\boldsymbol{u}^*)\, \boldsymbol{L}(\boldsymbol{u}^*) \end{gathered}$$

Now we will substitute these four equations into (I.2) and use (I.3) to obtain the update equation in terms of temporal derivatives on $\boldsymbol{u}$. Then, we compare with the Taylor's expansion of $\boldsymbol{u}$ (I.4) to get conditions for the respective orders of accuracy
First order:

$$b_1 + b_2 = 1 \tag{I.5}$$

Second order:

$$b_2\, a_{21} + \hat{b}_1 + \hat{b}_2 = \frac{1}{2} \tag{I.6}$$

Third order:

$$b_2\, a_{21}^2 + 2\, \hat{b}_2\, a_{21} = \frac{1}{3} \tag{I.7}$$

$$b_2\, \hat{a}_{21} + \hat{b}_2\, a_{21} = \frac{1}{6} \tag{I.8}$$

Fourth order:

$$b_2\, a_{21}^3 + 3\, \hat{b}_2\, a_{21}^2 = \frac{1}{4} \tag{I.9}$$

$$b_2\, a_{21}\, \hat{a}_{21} + \hat{b}_2\, a_{21}^2 + \hat{b}_2\, \hat{a}_{21} = \frac{1}{8} \tag{I.10}$$

$$\hat{b}_2\, a_{21}^2 = \frac{1}{12} \tag{I.11}$$

$$\hat{b}_2\, \hat{a}_{21} = \frac{1}{24} \tag{I.12}$$

From (I.11), (I.12) we get

$$\hat{a}_{21} = \frac{1}{2}\, a_{21}^2 \tag{I.13}$$

We then see that equations (I.7), (I.8) become identical, and equations (I.9), (I.10) become identical. Simplifying the above equations, we get five equations for the five unknown coefficients.

$$b_1 + b_2 = 1 \tag{I.14}$$

$$b_2\, a_{21} + \hat{b}_1 + \hat{b}_2 = \frac{1}{2} \tag{I.15}$$

$$b_2\, a_{21}^2 + 2\, \hat{b}_2\, a_{21} = \frac{1}{3} \tag{I.16}$$

$$b_2\, a_{21}^3 + 3\, \hat{b}_2\, a_{21}^2 = \frac{1}{4} \tag{I.17}$$

$$\hat{b}_2\, a_{21}^2 = \frac{1}{12} \tag{I.18}$$

Using (I.18) in (I.17) we get

$$b_2\, a_{21}^3 = 0$$

The solution $a_{21} = 0$ does not satisfy (I.16), (I.17), hence let us choose

$$b_2 = 0$$

Then we get the unique solution for the coefficients

$$b_1 = 1, \quad b_2 = 0, \quad \hat{b}_1 = \frac{1}{6}, \quad \hat{b}_2 = \frac{1}{3}, \quad a_{21} = \frac{1}{2}, \quad \hat{a}_{21} = \frac{1}{8}$$

These coefficients do give the scheme (7.5) for which the two stage method is fourth order accurate.

List of publications

1. Arpit Babbar, Sudarshan Kumar Kenettinkara, Praveen Chandrashekar (2022) [18]: Lax-Wendroff flux reconstruction method for hyperbolic conservation laws, Journal of Computational Physics Volume 467, 15 October 2022, 11142. https://doi.org/10.1016/j.jcp.2022.111423, https://arxiv.org/abs/2207.02954
2. Arpit Babbar, Sudarshan Kumar Kenettinkara, and Praveen Chandrashekar [19]. Admissibility preserving subcell limiter for lax-wendroff flux reconstruction, *Journal of Scientific Computing*, Volume 99, 2024. https://doi.org/10.1007/s10915-024-02482-9, https://doi.org/10.48550/arXiv.2305.10781
3. Arpit Babbar, Praveen Chandrashekar (2024) [13]: Lax-Wendroff Flux Reconstruction on adaptive curvilinear meshes with error based time stepping for hyperbolic conservation laws. https://arxiv.org/abs/2402.11926
4. Arpit Babbar, Praveen Chandrashekar (2024) [9]: Equivalence of ADER and Lax-Wendroff in DG / FR framework for linear problems, https://arxiv.org/abs/2402.18937
5. Arpit Babbar, Praveen Chandrashekar (2024) [11]: Generalized framework for admissibility preserving Lax-Wendroff Flux Reconstruction for hyperbolic conservation laws with source terms. https://arxiv.org/abs/2402.01442
6. Arpit Babbar, Praveen Chandrashekar (2024) [12]: Lax-Wendroff Flux Reconstruction for advection-diffusion equations with error-based time stepping. https://arxiv.org/abs/2402.12669
7. Arpit Babbar, Praveen Chandrashekar (2024) [14]: Multi-Derivative Runge-Kutta Flux Reconstruction for hyperbolic conservation laws. https://arxiv.org/abs/2403.02141

Bibliography

[1] Yoshiaki Abe, Takanori Haga, Taku Nonomura, and Kozo Fujii. On the freestream preservation of high-order conservative Flux-Reconstruction schemes. *Journal of Computational Physics*, 281:28–54, 2015.

[2] Rémi Abgrall, Elise le Mélédo, Philipp Öffner, and Hendrik Ranocha. Error boundedness of correction procedure via Reconstruction/Flux Reconstruction and the connection to residual distribution schemes. In Alberto Bressan, Marta Lewicka, Dehua Wang, and Yuxi Zheng, editors, *Hyperbolic problems: Theory, numerics, applications*, volume 10 of *AIMS on applied mathematics*, pages 215–222. Springfield, 2020. American Institute of Mathematical Sciences.

[3] Semih Akkurt, Freddie Witherden, and Peter Vincent. Cache blocking strategies applied to Flux Reconstruction. *Computer Physics Communications*, 271:108193, 2022.

[4] Ames Resarch Staff - National Advisory Committee for Aeronautics. Report 1135 - equations, tables and charts for compressible flow. 1951.

[5] Kartikey Asthana and Antony Jameson. High-Order Flux Reconstruction Schemes with Minimal Dispersion and Dissipation. *Journal of Scientific Computing*, 62(3):913–944, mar 2015.

[6] Norbert Attig, Paul Gibbon, and Th Lippert. Trends in supercomputing: the European path to exascale. *Computer Physics Communications*, 182(9):2041–2046, 2011.

[7] Arpit Babbar. ADER and Lax-Wendroff schemes in Flux Reconstruction framework. https://github.com/Arpit-Babbar/ADER_FR/tree/ader, 2024.

[8] Arpit Babbar and Praveen Chandrashekar. Admissibility preservation for Ten moment problem with Tenkai.jl. https://github.com/Arpit-Babbar/tenkai-icosahom2023, 2024.

[9] Arpit Babbar and Praveen Chandrashekar. Equivalence of ADER and Lax-Wendroff in DG / FR framework for linear problems. 2024.

[10] Arpit Babbar and Praveen Chandrashekar. Extension of LWFR to adaptive, curved meshes with error based time stepping. https://github.com/Arpit-Babbar/JCP2024, 2024.

[11] Arpit Babbar and Praveen Chandrashekar. Generalized framework for admissibility preserving Lax-Wendroff Flux Reconstruction for hyperbolic conservation laws with source terms. 2024.

[12] Arpit Babbar and Praveen Chandrashekar. Lax-Wendroff Flux Reconstruction for advection-diffusion equations with error-based time stepping. 2024.

[13] Arpit Babbar and Praveen Chandrashekar. Lax-Wendroff Flux Reconstruction on adaptive curvilinear meshes with error based time stepping for hyperbolic conservation laws. 2024.

[14] Arpit Babbar and Praveen Chandrashekar. Multi-Derivative Runge-Kutta Flux Reconstruction for hyperbolic conservation laws. 2024.

[15] Arpit Babbar and Praveen Chandrashekar. Solving parabolic equations using TrixiLW.jl. https://github.com/Arpit-Babbar/NavierStokesLWFR_ICOSAHOM2023, 2024.

[16] Arpit Babbar, Praveen Chandrashekar, and Sudarshan Kumar Kenettinkara. Admissibility preserving subcell based blending limiter with Tenkai.jl. https://github.com/Arpit-Babbar/jsc2023, 2023.

[17] Arpit Babbar, Praveen Chandrashekar, and Sudarshan Kumar Kenettinkara. Tenkai.jl: Temporal discretizations of high-order PDE solvers. https://github.com/Arpit-Babbar/Tenkai.jl, 2023.

[18] Arpit Babbar, Sudarshan Kumar Kenettinkara, and Praveen Chandrashekar. Lax-wendroff Flux Reconstruction method for hyperbolic conservation laws. *Journal of Computational Physics*, page 111423, 2022.

[19] Arpit Babbar, Sudarshan Kumar Kenettinkara, and Praveen Chandrashekar. Admissibility preserving subcell limiter for Lax-Wendroff Flux Reconstruction. 2023.

[20] D. Balsara, C. Altmann, C.D. Munz, and M. Dumbser. A sub-cell based indicator for troubled zones in RKDG schemes and a novel class of hybrid RKDG+HWENO schemes. *J. Comp. Phys.*, 226:586–620, 2007.

[21] Dinshaw S. Balsara, Tobias Rumpf, Michael Dumbser, and Claus-Dieter Munz. Efficient, high accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics. *Journal of Computational Physics*, 228(7):2480–2516, apr 2009.

[22] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of Computational Physics*, 131(2):267–279, 1997.

[23] P. Batten, N. Clarke, C. Lambert, and D. M. Causon. On the Choice of Wavespeeds for the HLLC Riemann Solver. *SIAM Journal on Scientific Computing*, 18(6):1553–1570, nov 1997.

[24] Matania Ben-Artzi, Jiequan Li, and Gerald Warnecke. A direct Eulerian GRP scheme for compressible fluid flows. *Journal of Computational Physics*, 218(1):19–43, oct 2006.

[25] Christophe Berthon. Numerical approximations of the 10-moment gaussian closure. *Mathematics of Computation*, 75(256):1809–1831, jun 2006.

[26] Christophe Berthon. Why the MUSCL–Hancock scheme is L1-stable. *Numerische Mathematik*, 104(1):27–46, jun 2006.

[27] Christophe Berthon, Bruno Dubroca, and Afeintou Sangam. An entropy preserving relaxation scheme for ten-moments equations with source terms. *Communications in Mathematical Sciences*, 13(8):2119–2154, 2015.

[28] M. Berzins. Temporal error control for convection-dominated equations in two space dimensions. *SIAM Journal on Scientific Computing*, 16(3):558–580, 1995.

[29] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, 59(1):65–98, jan 2017.

[30] Biswarup Biswas, Harish Kumar, and Anshu Yadav. Entropy stable discontinuous Galerkin methods for ten-moment gaussian closure equations. *Journal of Computational Physics*, 431:110148, 2021.

[31] Rupak Biswas, Karen D. Devine, and Joseph E. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Applied Numerical Mathematics*, 14(1):255–283, 1994.

[32] P. Bogacki and L.F. Shampine. A 3(2) pair of runge - kutta formulas. *Applied Mathematics Letters*, 2(4):321–325, 1989.

[33] A. Burbeau, P. Sagaut, and Ch.-H. Bruneau. A problem-independent limiter for high-order Runge–Kutta discontinuous Galerkin methods. *Journal of Computational Physics*, 169(1):111–150, 2001.

[34] Raimund Bürger, Sudarshan Kumar Kenettinkara, and David Zorío. Approximate Lax–Wendroff discontinuous Galerkin methods for hyperbolic conservation laws. *Computers & Mathematics with Applications*, 74(6):1288–1310, sep 2017.

[35] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Ltd, Chichester, UK, jul 2016.

[36] Canaero. 5th international workshop on high-order CFD methods. 2017.

[37] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods: Fundamentals in Single Domains*. Scientific Computation. Springer Berlin Heidelberg, 2007.

[38] Mark H. Carpenter, David Gottlieb, Saul Abarbanel, and Wai-Sun Don. The Theoretical Accuracy of Runge–Kutta Time Discretizations for the Initial Boundary Value Problem: A Study of the Boundary Error. *SIAM Journal on Scientific Computing*, 16(6):1241–1252, nov 1995.

[39] H. Carrillo, E. Macca, C. Parés, G. Russo, and D. Zorío. An order-adaptive compact approximation Taylor method for systems of conservation laws. *Journal of Computational Physics*, 438:110358, aug 2021.

[40] H. Carrillo, C. Parés, and D. Zorío. Lax-Wendroff Approximate Taylor Methods with Fast and Optimized Weighted Essentially Non-oscillatory Reconstructions. *Journal of Scientific Computing*, 86(1):15, jan 2021.

[41] E. Casoni, J. Peraire, and A. Huerta. One-dimensional shock-capturing for high-order discontinuous Galerkin methods. *Int. J. Numer. Meth. Fluids*, 71:737–755, 2013.

[42] C.E. Castro and E.F. Toro. Solvers for the high-order Riemann problem for hyperbolic balance laws. *Journal of Computational Physics*, 227(4):2481–2513, feb 2008.

[43] Zheng Chen, Hongying Huang, and Jue Yan. Third order maximum-principle-satisfying direct discontinuous Galerkin methods for time dependent convection diffusion equations on unstructured triangular meshes. *Journal of Computational Physics*, 308:198–217, 2016.

[44] Kyu Y. Choe and Keith A. Holsapple. The Taylor-Galerkin discontinuous finite element method—An explicit scheme for nonlinear hyperbolic conservation laws. *Finite Elements in Analysis and Design*, 10(3):243–265, dec 1991.

[45] Kyu Y. Choe and Keith A. Holsapple. The discontinuous finite element method with the Taylor-Galerkin approach for nonlinear hyperbolic conservation laws. *Computer Methods in Applied Mechanics and Engineering*, 95(2):141–167, mar 1992.

[46] Alexander Cicchino, David C. Del Rey Fernández, Siva Nadarajah, Jesse Chan, and Mark H. Carpenter. Provably stable Flux Reconstruction high-order methods on curvilinear elements. *Journal of Computational Physics*, 463:111259, 2022.

[47] Alexander Cicchino, Siva Nadarajah, and David C. Del Rey Fernández. Nonlinearly stable Flux Reconstruction high-order methods in split form. *Journal of Computational Physics*, 458:111094, 2022.

[48] S. Clain, S. Diot, and R. Loubère. A high-order finite volume method for hyperbolic systems: Multi-dimensional Optimal Order Detection (MOOD). *J. Comp. Phys.*, 230:4028–4050, 2011.

[49] Bernardo Cockburn, George E. Karniadakis, Chi-Wang Shu, M. Griebel, D. E. Keyes, R. M. Nieminen, D. Roose, and T. Schlick, editors. *Discontinuous Galerkin Methods: Theory, Computation and Applications*, volume 11 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[50] Bernardo Cockburn, San-Yih Lin, and Chi-Wang Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *Journal of Computational Physics*, 84(1):90–113, sep 1989.

[51] Bernardo Cockburn and Chi-Wang Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. *Mathematics of Computation*, 52(186):411–411, may 1989.

[52] Bernardo Cockburn and Chi-Wang Shu. The runge-kutta local projection P^1-discontinuous-Galerkin finite element method for scalar conservation laws. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 25(3):337–361, 1991.

[53] Bernardo Cockburn and Chi-Wang Shu. The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, dec 1998.

[54] Phillip Colella. Multidimensional upwind methods for hyperbolic conservation laws. *Journal of Computational Physics*, 87(1):171–200, 1990.

[55] Phillip Colella and Paul R Woodward. The piecewise parabolic method (ppm) for gas-dynamical simulations. *Journal of Computational Physics*, 54(1):174–201, 1984.

[56] Shumo Cui, Shengrong Ding, and Kailiang Wu. Is the classic convex decomposition optimal for bound-preserving schemes in multiple dimensions? *Journal of Computational Physics*, 476:111882, 2023.

[57] D. De Grazia, G. Mengaldo, D. Moxey, P. E. Vincent, and S. J. Sherwin. Connections between the discontinuous Galerkin method and high-order Flux Reconstruction schemes. 75(12):860–877, 2014.

[58] J. N. de la Rosa and C. D. Munz. Hybrid DG/FV schemes for magnetohydrodynamics and relativistichydrodynamics. *Comp. Phys. Commun.*, 222:113–135, 2018.

[59] 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 and Fluids*, 64:43–63, 2012.

[60] S. Diot, R. Loubère, and S. Clain. The MOOD method in the three-dimensional case: very-high-order finite volume method for hyperbolic systems. *Int. J. Numer. Meth. Fluids*, 73:362–392, 2013.

[61] M. Dumbser and R. Loubère. A simple robust and accurate a posteriorisub-cell finite volume limiter for the discontinuousGalerkin method on unstructured meshes. *J. Comp. Phys.*, 319:163–199, 2016.

[62] M. Dumbser, O. Zanotti, R. Loubère, and S. Diot. A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *J. Comp. Phys.*, 278:47–75, 2014.

[63] Michael Dumbser, Dinshaw S. Balsara, Eleuterio F. Toro, and Claus-Dieter Munz. A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227(18):8209–8253, sep 2008.

[64] Michael Dumbser, Cedric Enaux, and Eleuterio F. Toro. Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws. *Journal of Computational Physics*, 227(8):3971–4001, apr 2008.

[65] Michael Dumbser, Francesco Fambri, Maurizio Tavelli, Michael Bader, and Tobias Weinzierl. Efficient Implementation of ADER Discontinuous Galerkin Schemes for a Scalable Hyperbolic PDE Engine. *Axioms*, 7(3):63, sep 2018.

[66] Michael Dumbser, Martin Käser, Vladimir A. Titarev, and Eleuterio F. Toro. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *Journal of Computational Physics*, 226(1):204–243, sep 2007.

[67] Michael Dumbser and Claus-Dieter Munz. Building Blocks for Arbitrary High Order Discontinuous Galerkin Schemes. *Journal of Scientific Computing*, 27(1-3):215–230, jun 2006.

[68] Michael Dumbser and Olindo Zanotti. Very high order PNPM schemes on unstructured meshes for the resistive relativistic MHD equations. *Journal of Computational Physics*, 228(18):6991–7006, oct 2009.

[69] T. Dzanic and F.D. Witherden. Positivity-preserving entropy-based adaptive filtering for discontinuous spectral element methods. *Journal of Computational Physics*, 468:111501, 2022.

[70] Thomas D. Economon, Francisco Palacios, Sean R. Copeland, Trent W. Lukaczyk, and Juan J. Alonso. Su2: an open-source suite for multiphysics simulation and design. *AIAA Journal*, 54(3):828–846, 2016.

[71] Bernd Einfeldt. On Godunov-Type Methods for Gas Dynamics. *SIAM Journal on Numerical Analysis*, 25(2):294–318, apr 1988.

[72] Truman Ellis, Jesse Chan, and Leszek Demkowicz. *Robust DPG Methods for Transient Convection-Diffusion*, pages 179–203. Springer International Publishing, 2016.

[73] Ashley F Emery. An evaluation of several differencing methods for inviscid fluid flow problems. *Journal of Computational Physics*, 2(3):306–331, 1968.

[74] Björn Engquist and Stanley Osher. One-sided difference approximations for nonlinear conservation laws. *Mathematics of Computation*, 36(154):321–351, 1981.

[75] Francesco Fambri, Michael Dumbser, and Olindo Zanotti. Space–time adaptive ader-dg schemes for dissipative flows: compressible Navier–Stokes and resistive MHD equations. *Computer Physics Communications*, 220:297–318, 2017.

[76] Gregor Gassner, Michael Dumbser, Florian Hindenlang, and Claus-Dieter Munz. Explicit one-step time discretizations for discontinuous Galerkin and finite volume schemes based on local predictors. *Journal of Computational Physics*, 230(11):4232–4247, may 2011.

[77] Gregor J. Gassner and Andrea D. Beck. On the accuracy of high-order discretizations for underresolved turbulence simulations. *Theoretical and Computational Fluid Dynamics*, 27(3–4):221–237, jan 2012.

[78] Gregor J. Gassner, Andrew R. Winters, Florian J. Hindenlang, and David A. Kopriva. The br1 scheme is stable for the compressible Navier–Stokes equations. *Journal of Scientific Computing*, 77(1):154–200, 2018. Citation Key: Gassner2018.

[79] U Ghia, K.N Ghia, and C.T Shin. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of Computational Physics*, 48(3):387–411, 1982.

[80] James Glimm, Christian Klingenberg, Oliver McBryan, Bradley Plohr, David Sharp, and Sara Yaniv. Front tracking and two-dimensional Riemann problems. *Advances in Applied Mathematics*, 6(3):259–290, 1985.

[81] Sergei K. Godunov and I. Bohachevsky. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematičeskij sbornik*, 47(89)(3):271–306, 1959.

[82] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. Strong Stability-Preserving High-Order Time Discretization Methods. *SIAM Review*, 43(1):89–112, jan 2001.

[83] Jean-Luc Guermond and Bojan Popov. Fast estimation from above of the maximum wave speed in the Riemann problem for the Euler equations. *Journal of Computational Physics*, 321(C):908–926, sep 2016.

[84] Wei Guo, Jing-Mei Qiu, and Jianxian Qiu. A New Lax–Wendroff Discontinuous Galerkin Method with Superconvergence. *Journal of Scientific Computing*, 65(1):299–326, oct 2015.

[85] Pierson T. Guthrey and James A. Rossmanith. The Regionally Implicit Discontinuous Galerkin Method: Improving the Stability of DG-FEM. *SIAM Journal on Numerical Analysis*, 57(3):1263–1288, jan 2019.

[86] Youngsoo Ha, Carl Gardner, Anne Gelb, and Chi Wang Shu. Numerical simulation of high mach number astrophysical jets with radiative cooling. *Journal of Scientific Computing*, 24(1):597–612, jul 2005.

[87] Ee Han, Jiequan Li, and Huazhong Tang. An adaptive GRP scheme for compressible fluid flows. *Journal of Computational Physics*, 229(5):1448–1466, mar 2010.

[88] Ami Harten and Stanley Osher. Uniformly high-order accurate nonoscillatory schemes. i. *SIAM Journal on Numerical Analysis*, 24(2):279–309, 1987.

[89] Amiram Harten, Peter D. Lax, and Bram van Leer. On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws. *SIAM Review*, 25(1):35–61, jan 1983.

[90] Sebastian Hennemann, Andrés M. Rueda-Ramírez, Florian J. Hindenlang, and Gregor J. Gassner. A provably entropy stable subcell shock capturing approach for high order split form dg for the compressible Euler equations. *Journal of Computational Physics*, 426:109935, 2021.

[91] Francis Begnaud Hildebrand. *Introduction to numerical analysis*. 1973.

[92] Charles Hirsch. *Numerical Computation of Internal and External Flows, Volume 2: Computational Methods for Inviscid and Viscous Flows*. Wiley, 1990.

[93] A. Huerta, E. Casoni, and J. Peraire. A simple shock-capturing technique for high-order discontinuous Galerkin methods. *Int. J. Numer. Meth. Fluids*, 69:1614–1632, 2012.

[94] H. T. Huynh. A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods. Miami, FL, jun 2007. AIAA.

[95] Haran Jackson. On the eigenvalues of the ADER-WENO galerkin predictor. *Journal of Computational Physics*, 333:409–413, mar 2017.

[96] Rasha Al Jahdali, Radouan Boukharfane, Lisandro Dalcin, and Matteo Parsani. *Optimized Explicit Runge-Kutta Schemes for Entropy Stable Discontinuous Collocated Methods Applied to the Euler and Navier–Stokes equations*, page 0.

[97] A. Jameson, P. E. Vincent, and P. Castonguay. On the Non-linear Stability of Flux Reconstruction Schemes. *Journal of Scientific Computing*, 50(2):434–445, feb 2012.

[98] Guang-Shan Jiang and Chi-Wang Shu. Efficient Implementation of Weighted ENO Schemes. *Journal of Computational Physics*, 126(1):202–228, jun 1996.

[99] Yan Jiang, Chi-Wang Shu, and Mengping Zhang. An Alternative Formulation of Finite Difference Weighted ENO Schemes with Lax–Wendroff Time Discretization for Conservation Laws. *SIAM Journal on Scientific Computing*, 35(2):0, jan 2013.

[100] Martin Käser and Armin Iske. ADER schemes on adaptive triangular meshes for scalar conservation laws. *Journal of Computational Physics*, 205(2):486–508, may 2005.

[101] David I. Ketcheson, Mikael Mortensen, Matteo Parsani, and Nathanael Schilling. More efficient time integration for fourier pseudospectral dns of incompressible turbulence. *International Journal for Numerical Methods in Fluids*, 92(2):79–93, 2020.

[102] Andreas Klöckner, Tim Warburton, and Jan Hesthaven. Viscous shock capturing in a time-explicit discontinuous Galerkin method. *Mathematical Modelling of Natural Phenomena*, 6:0, 02 2011.

[103] David Kopriva. *Implementing Spectral Methods for Partial Differential Equations*. 01 2009.

[104] David A Kopriva, Florian J Hindenlang, Thomas Bolemann, and Gregor J Gassner. Free-stream preservation for curved geometrically non-conforming discontinuous Galerkin spectral elements. *J. Sci. Comput.*, 79(3):1389–1408, jun 2019.

[105] David A. Kopriva. Metric identities and the discontinuous spectral element method on curvilinear meshes. *Journal of Scientific Computing*, 26(3):301–327, 2006.

[106] David A. Kopriva and John H. Kolias. A conservative staggered-grid chebyshev multidomain method for compressible flows. *Journal of Computational Physics*, 125(1):244–261, 1996.

[107] David A. Kopriva, Stephen L. Woodruff, and M. Y. Hussaini. Computation of electromagnetic scattering with a non-conforming discontinuous spectral element method. *International Journal for Numerical Methods in Engineering*, 53(1):105–122, 2002.

[108] David Kopriva and Gregor Gassner. On the quadrature and weak form choices in collocation type discontinuous Galerkin spectral element methods. *J. Sci. Comput.*, 44:136–155, 08 2010.

[109] Jeremy E. Kozdon and Lucas C. Wilcox. An energy stable approach for discretizing hyperbolic equations with nonconforming discontinuous Galerkin methods. *Journal of Scientific Computing*, 76(3):1742–1784, mar 2018.

[110] Lilia Krivodonova. Limiters for high-order discontinuous Galerkin methods. *Journal of Computational Physics*, 226(1):879–896, 2007.

[111] Peter D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Communications on Pure and Applied Mathematics*, 7(1):159–193, feb 1954.

[112] Peter D. Lax and Xu-Dong Liu. Solution of two-dimensional Riemann problems of gas dynamics by positive schemes. *SIAM Journal on Scientific Computing*, 19(2):319–340, 1998.

[113] Peter Lax and Burton Wendroff. Systems of conservation laws. *Communications on Pure and Applied Mathematics*, 13(2):217–237, may 1960.

[114] Youngjun Lee and Dongwook Lee. A single-step third-order temporal discretization with Jacobian-free and Hessian-free formulations for finite difference methods. *Journal of Computational Physics*, 427:110063, feb 2021.

[115] Philippe G. LeFloch. *Hyperbolic Systems of Conservation Laws*. Birkhäuser Basel, 2002.

[116] Randall J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser Basel, 1992.

[117] Randall J. LeVeque. High-Resolution Conservative Algorithms for Advection in Incompressible Flow. *SIAM Journal on Numerical Analysis*, 33(2):627–665, apr 1996.

[118] C. David Levermore. Moment closure hierarchies for kinetic theories. *Journal of Statistical Physics*, 83(5–6):1021–1065, 1996.

[119] Jiequan Li and Zhifang Du. A two-stage fourth order time-accurate discretization for Lax–Wendroff type flow solvers i. hyperbolic conservation laws. *SIAM Journal on Scientific Computing*, 38(5):0, 2016.

[120] Rainald Löhner. An adaptive finite element scheme for transient problems in cfd. *Computer Methods in Applied Mechanics and Engineering*, 61(3):323–338, 1987.

[121] Manuel R. López, Abhishek Sheshadri, Jonathan R. Bull, Thomas D. Economon, Joshua Romero, Jerry E. Watkins, David M. Williams, Francisco Palacios, Antony Jameson, and David E. Manosalvas. Verification and Validation of HiFiLES: a High-Order LES unstructured solver on multi-GPU platforms. In *32nd AIAA Applied Aerodynamics Conference*. Atlanta, GA, jun 2014. American Institute of Aeronautics and Astronautics.

[122] Shuai Lou, Chao Yan, Li-Bin Ma, and Zhen-Hua Jiang. The Flux Reconstruction Method with Lax–Wendroff Type Temporal Discretization for Hyperbolic Conservation Laws. *Journal of Scientific Computing*, 82(2):42, feb 2020.

[123] Jianfang Lu, Yong Liu, and Chi-Wang Shu. An oscillation-free discontinuous Galerkin method for scalar hyperbolic conservation laws. *SIAM Journal on Numerical Analysis*, 59(3):1299–1324, 2021.

[124] Asha Kumari Meena and Harish Kumar. Robust MUSCL Schemes for Ten-Moment Gaussian Closure Equations with Source Terms. *International Journal on Finite Volumes*, Oct 2017.

[125] Asha Kumari Meena, Harish Kumar, and Praveen Chandrashekar. Positivity-preserving high-order discontinuous Galerkin schemes for ten-moment gaussian closure equations. *Journal of Computational Physics*, 339(Supplement C):370–395, jun 2017.

[126] Asha Kumari Meena, Rakesh Kumar, and Praveen Chandrashekar. Positivity-preserving finite difference WENO scheme for ten-moment equations with source term. *Journal of Scientific*

Computing, 82(1), jan 2020.

[127] G. Mengaldo, D. De Grazia, P. E. Vincent, and S. J. Sherwin. On the connections between discontinuous Galerkin and Flux Reconstruction schemes: extension to curvilinear meshes. *Journal of Scientific Computing*, 67(3):1272–1292, oct 2015.

[128] Scott A. Moe, James A. Rossmanith, and David C. Seal. Positivity-preserving discontinuous Galerkin methods with lax–wendroff time discretizations. *Journal of Scientific Computing*, 71:44–70, 2017.

[129] Gino I. Montecinos and Dinshaw S. Balsara. A simplified Cauchy-Kowalewskaya procedure for the local implicit solution of generalized Riemann problems of hyperbolic balance laws. *Computers & Fluids*, 202:104490, apr 2020.

[130] N. Obrechkoff. *Neue Quadraturformeln*. Abhandlungen der Preussischen Akademie der Wissenschaften. Math.-naturw. Klasse. Akad. d. Wissenschaften, 1940.

[131] Philipp Öffner and Hendrik Ranocha. Error Boundedness of Discontinuous Galerkin Methods with Variable Coefficients. *Journal of Scientific Computing*, 79(3):1572–1607, jun 2019.

[132] Liang Pan, Jiequan Li, and Kun Xu. A few benchmark test cases for higher-order Euler solvers. *Numerical Mathematics: Theory, Methods and Applications*, 10:0, 09 2016.

[133] Will Pazner. Sparse invariant domain preserving discontinuous Galerkin methods with subcell convex limiting. *Computer Methods in Applied Mechanics and Engineering*, 382:113876, 2021.

[134] Per-Olof Persson and Jaime Peraire. Sub-Cell Shock Capturing for Discontinuous Galerkin Methods. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics, jan 2006.

[135] J. Qiu and C.-W. Shu. Runge Kutta discontinuous Galerkin method using WENO limiters. *SIAM J. Sci. Comput.*, 26:907–929, 2005.

[136] Jianxian Qiu. A Numerical Comparison of the Lax–Wendroff Discontinuous Galerkin Method Based on Different Numerical Fluxes. *Journal of Scientific Computing*, 30(3):345–367, mar 2007.

[137] Jianxian Qiu, Michael Dumbser, and Chi-Wang Shu. The discontinuous Galerkin method with Lax–Wendroff type time discretizations. *Computer Methods in Applied Mechanics and Engineering*, 194(42-44):4528–4543, oct 2005.

[138] Jianxian Qiu and Chi-Wang Shu. Finite Difference WENO Schemes with Lax–Wendroff-Type Time Discretizations. *SIAM Journal on Scientific Computing*, 24(6):2185–2198, jan 2003.

[139] Christopher Rackauckas and Qing Nie. DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia. *Journal of Open Research Software*, 5(1):15, may 2017.

[140] Hendrik Ranocha, Lisandro Dalcin, Matteo Parsani, and David I. Ketcheson. Optimized runge-kutta methods with automatic step size control for compressible computational fluid dynamics. *Communications on Applied Mathematics and Computation*, 4(4):1191–1228, nov 2021.

[141] Hendrik Ranocha, Michael Schlottke-Lakemper, Andrew Ross Winters, Erik Faulhaber, Jesse Chan, and Gregor Gassner. Adaptive numerical simulations with Trixi.jl: A case study of Julia for scientific computing. *Proceedings of the JuliaCon Conferences*, 1(1):77, 2022.

[142] Hendrik Ranocha, Andrew Winters, Guillermo Castro, Lisandro Dalcin, Michael Schlottke-Lakemper, Gregor Gassner, and Matteo Parsani. On error-based step size control for discontinuous Galerkin methods for compressible fluid dynamics. *Communications on Applied Mathematics and Computation*, page 0, 05 2023.

[143] Deep Ray. Entropy-stable finite difference and finite volume schemes for compressible flows. 2017. PhD Thesis, Tata Institute of Fundamental Research - Centre for Applicable Mathematics.

[144] Deep Ray and Praveen Chandrashekar. An entropy stable finite volume scheme for the two dimensional Navier–Stokes equations on triangular grids. *Applied Mathematics and Computation*, 314:257–286, 2017.

[145] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. In *National topical meeting on mathematical models and computational techniques for analysis of nuclear systems*. Ann Arbor, Michigan, oct 1973. Los Alamos Scientific Lab., N.Mex. (USA).

[146] Philip L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, oct 1981.

[147] J. Romero, K. Asthana, and A. Jameson. A Simplified Formulation of the Flux Reconstruction Method. *Journal of Scientific Computing*, 67(1):351–374, apr 2016.

[148] Andr'es M. Rueda-Ram'irez, Sebastian Hennemann, Florian Hindenlang, Andrew R. Winters, and Gregor J. Gassner. An entropy stable nodal discontinuous Galerkin method for the resistive MHD equations. part II: subcell finite volume shock capturing. *J. Comput. Phys.*, 444:110580, 2020.

[149] Andrés M. Rueda-Ramírez, Sebastian Hennemann, Florian J. Hindenlang, Andrew R. Winters, and Gregor J. Gassner. An entropy stable nodal discontinuous Galerkin method for the resistive mhd equations. part ii: subcell finite volume shock capturing. *Journal of Computational Physics*, 444:110580, 2021.

[150] Andrés M. Rueda-Ramírez, Will Pazner, and Gregor J. Gassner. Subcell limiting strategies for discontinuous Galerkin spectral element methods. *Computers & Fluids*, 247:105627, 2022.

[151] A Rueda-Ramrez and G Gassner. A subcell finite volume positivity-preserving limiter for DGSEM discretizations of the Euler equations. In *14th WCCM-ECCOMAS Congress*. CIMNE, 2021.

[152] V.V Rusanov. The calculation of the interaction of non-stationary shock waves and obstacles. *USSR Computational Mathematics and Mathematical Physics*, 1(2):304–320, jan 1962.

[153] Steven J. Ruuth and R. Spiteri. Two Barriers on Strong-Stability-Preserving Time Discretization Methods. *J. Sci. Comput.*, 2002.

[154] A. Safjan and J.T. Oden. High-Order Taylor-Galerkin Methods for Linear Hyperbolic Systems. *Journal of Computational Physics*, 120(2):206–230, sep 1995.

[155] Kevin Schaal, Andreas Bauer, Praveen Chandrashekar, Rüdiger Pakmor, Christian Klingenberg, and Volker Springel. Astrophysical hydrodynamics with a high-order discontinuous Galerkin scheme and adaptive mesh refinement. *Monthly Notices of the Royal Astronomical Society*, 453(4):4278–4300, 09 2015.

[156] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher. *Benchmark Computations of Laminar Flow Around a Cylinder*, pages 547–566. Vieweg+Teubner Verlag, Wiesbaden, 1996.

[157] Michael Schlottke-Lakemper, Gregor J Gassner, Hendrik Ranocha, Andrew R Winters, and Jesse Chan. Trixi.jl: Adaptive high-order numerical simulations of hyperbolic PDEs in Julia. https://github.com/trixi-framework/Trixi.jl, 09 2021.

[158] Michael Schlottke-Lakemper, Andrew R Winters, Hendrik Ranocha, and Gregor J Gassner. A purely hyperbolic discontinuous Galerkin approach for self-gravitating gas dynamics. *Journal of Computational Physics*, 442:110467, 06 2021.

[159] David Seal, Yaman Güçlü, and Andrew Christlieb. High-order multiderivative time integrators for hyperbolic conservation laws. *Journal of Scientific Computing*, 60:0, 04 2013.

[160] L.I. SEDOV. Chapter iv - one-dimensional unsteady motion of a gas. In L.I. SEDOV, editor, *Similarity and Dimensional Methods in Mechanics*, pages 146–304. Academic Press, 1959.

[161] Jing Shi, Yong-Tao Zhang, and Chi-Wang Shu. Resolution of high order WENO schemes for complicated flow structures. *Journal of Computational Physics*, 186:690–696, 04 2003.

[162] Chi-Wang Shu. Total-variation-diminishing time discretizations. *SIAM Journal on Scientific and Statistical Computing*, 9(6):1073–1084, 1988.

[163] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *Journal of Computational Physics*, 83(1):32–78, jul 1989.

[164] Gary A Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27(1):1–31, apr 1978.

[165] M. Sonntag and C. D. Munz. Shock capturing for discontinuous Galerkin methods using finite volume subcells. In *Finite Volumes for Complex Applications VII*, pages 945–953. Springer, 2014.

[166] Seth C. Spiegel, James R. DeBonis, and H.T. Huynh. Overview of the NASA Glenn Flux Reconstruction Based High-Order Unstructured Grid Code. In *54th AIAA Aerospace Sciences Meeting*. San Diego, California, USA, jan 2016. American Institute of Aeronautics and Astronautics.

[167] Raymond J. Spiteri and Steven J. Ruuth. A New Class of Optimal High-Order Strong-Stability-Preserving Time Discretization Methods. *SIAM Journal on Numerical Analysis*, 40(2):469–491, jan 2002.

[168] Volker Springel. E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *Monthly Notices of the Royal Astronomical Society*, 401(2):791–851, 01 2010.

[169] ASCAC Subcommittee et al. Top ten exascale research challenges. *US Department Of Energy Report*, 2014.

[170] Zheng Sun and Chi-Wang Shu. Stability analysis and error estimates of Lax–Wendroff discontinuous Galerkin methods for linear conservation laws. *ESAIM: Mathematical Modelling and Numerical Analysis*, 51(3):1063–1087, may 2017.

[171] R. C. Swanson and S. Langer. Steady-state laminar flow solutions for NACA0012 airfoil. *Computers & Fluids*, 126(Supplement C):102–128, mar 2016.

[172] B Tabarrok and Jichao Su. Semi-implicit Taylor—Galerkin finite element methods for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 117(3-4):391–410, aug 1994.

[173] K. Takayama and O. Inoue. Shock wave diffraction over a 90 degree sharp corner - posters presented at 18th ISSW. *Shock Waves*, 1(4):301–312, dec 1991.

[174] Huazhong Tang and Tiegang Liu. A note on the conservative schemes for the Euler equations. *Journal of Computational Physics*, 218(2):451–459, 2006.

[175] V. A. Titarev and E. F. Toro. ADER: Arbitrary High Order Godunov Approach. *Journal of Scientific Computing*, 17(1/4):609–618, 2002.

[176] V.A. Titarev and E.F. Toro. Finite-volume WENO schemes for three-dimensional conservation laws. *Journal of Computational Physics*, 201(1):238–260, 2004.

[177] V.A. Titarev and E.F. Toro. ADER schemes for three-dimensional non-linear hyperbolic systems. *Journal of Computational Physics*, 204(2):715–736, apr 2005.

[178] E. F. Toro, R. C. Millington, and L. A. M. Nejad. Towards Very High Order Godunov Schemes. In E. F. Toro, editor, *Godunov Methods*, pages 907–940. Springer US, New York, NY, 2001.

[179] E. F. Toro, L. O. Müller, and A. Siviglia. Bounds for Wave Speeds in the Riemann Problem: Direct Theoretical Estimates. *Computers & Fluids*, 209:104640, sep 2020.

[180] E. F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4(1):25–34, jul 1994.

[181] Eleuterio F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[182] W. Trojak and F. D. Witherden. A new family of weighted one-parameter Flux Reconstruction schemes. *Computers & Fluids*, 222:104918, may 2021.

[183] Ch. Tsitouras. Runge–Kutta pairs of order 5(4) satisfying only the first column simplifying assumption. *Computers & Mathematics with Applications*, 62(2):770–775, jul 2011.

[184] Bram Van Leer. Towards the ultimate conservative difference scheme. iv. a new approach to numerical convection. *Journal of Computational Physics*, 23(3):276–299, 1977.

[185] Bram van Leer. On the relation between the upwind-differencing schemes of godunov, engquist–osher and roe. *SIAM Journal on Scientific and Statistical Computing*, 5:1–20, 03 1984.

[186] Ray Vandenhoeck and Andrea Lani. Implicit high-order Flux Reconstruction solver for high-speed compressible flows. *Computer Physics Communications*, 242:1–24, sep 2019.

[187] B.C. Vermeire and P.E. Vincent. On the behaviour of fully-discrete Flux Reconstruction schemes. *Computer Methods in Applied Mechanics and Engineering*, 315:1053–1079, mar 2017.

[188] François Vilar. A posteriori correction of high-order discontinuous Galerkin scheme through subcell finite volume formulation and Flux Reconstruction. *Journal of Computational Physics*, 387:245–279, jun 2019.

[189] P. E. Vincent, P. Castonguay, and A. Jameson. A New Class of High-Order Energy Stable Flux Reconstruction Schemes. *Journal of Scientific Computing*, 47(1):50–72, apr 2011.

[190] P. E. Vincent, P. Castonguay, and A. Jameson. Insights from von Neumann analysis of high-order Flux Reconstruction schemes. *Journal of Computational Physics*, 230(22):8134–8154, sep 2011.

[191] P. E. Vincent, A. M. Farrington, F. D. Witherden, and A. Jameson. An extended range of stable-symmetric-conservative Flux Reconstruction correction functions. *Computer Methods in Applied Mechanics and Engineering*, 296:248–272, nov 2015.

[192] Peter Vincent. Pyfr: latest developments and future roadmap. May 2022.

[193] Peter Vincent, Freddie Witherden, Brian Vermeire, Jin Seok Park, and Arvind Iyer. Towards Green Aviation with Python at Petascale. In *SC16: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–11. Salt Lake City, UT, nov 2016. IEEE.

[194] J. Ware and M. Berzins. Adaptive finite volume methods for time-dependent p.d.e.s. In Ivo Babuska, William D. Henshaw, Joseph E. Oliger, Joseph E. Flaherty, John E. Hopcroft, and Tayfun Tezduyar, editors, *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*, pages 417–430. New York, NY, 1995. Springer New York.

[195] F.D. Witherden, A.M. Farrington, and P.E. Vincent. Pyfr: an open source framework for solving advection–diffusion type problems on streaming architectures using the Flux Reconstruction approach. *Computer Physics Communications*, 185(11):3028–3040, 2014.

[196] F.D. Witherden and P.E. Vincent. On nodal point sets for Flux Reconstruction. *Journal of Computational and Applied Mathematics*, 381:113014, jan 2021.

[197] Paul Woodward and Phillip Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54(1):115–173, apr 1984.

[198] Yuan Xu, Qiang Zhang, Chi-wang Shu, and Haijin Wang. The L^2-norm Stability Analysis of Runge–Kutta Discontinuous Galerkin Methods for Linear Hyperbolic Equations. *SIAM Journal on Numerical Analysis*, 57(4):1574–1601, jan 2019.

[199] Ziyao Xu and Chi-Wang Shu. Third order maximum-principle-satisfying and positivity-preserving Lax-Wendroff discontinuous Galerkin methods for hyperbolic conservation laws. *Journal of Computational Physics*, 470:111591, 2022.

[200] H.C Yee, N.D Sandham, and M.J Djomehri. Low-Dissipative High-Order Shock-Capturing Methods Using Characteristic-Based Filters. *Journal of Computational Physics*, 150(1):199–238, mar 1999.

[201] Sung-Kie Youn and Sang-Hoon Park. A new direct higher-order Taylor-Galerkin finite element method. *Computers & Structures*, 56(4):651–656, aug 1995.

[202] 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(3):3010–3029, 07 2015.

[203] Tong Zhang and Yuxi Zheng. Conjecture on the structure of solutions of the Riemann problem for two-dimensional gas dynamics systems. *SIAM Journal on Mathematical Analysis*, 21:593–630, 1990.

[204] Tong Zhang and Yuxi Zheng. Exact spiral solutions of the two-dimensional Euler equations. *Discrete and Continuous Dynamical Systems*, 3(1):117–133, 1997.

[205] Xiangxiong Zhang and Chi-Wang Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *Journal of Computational Physics*, 229(9):3091–3120, may 2010.

[206] Xiangxiong Zhang and Chi-Wang Shu. On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. *Journal of Computational Physics*, 229(23):8918–8934, 2010.

[207] Xiangxiong Zhang and Chi-Wang Shu. Positivity-preserving high order finite difference WENO schemes for compressible Euler equations. *Journal of Computational Physics*, 231(5):2245–2258, 2012.

[208] D. Zorío, A. Baeza, and P. Mulet. An Approximate Lax–Wendroff-Type Procedure for High Order Accurate Schemes for Hyperbolic Conservation Laws. *Journal of Scientific Computing*, 71(1):246–273, apr 2017.