Journal of Scientific Computing (2023) 97:13
https://doi.org/10.1007/s10915-023-02322-2

Applications of Limiters, Neural Networks and Polynomial Annihilation in Higher-Order FD/FV Schemes

Dorian Hillebrand[1] · **Simon-Christian Klein**[1] · **Philipp Öffner**[2]

Received: 17 August 2022 / Revised: 8 August 2023 / Accepted: 10 August 2023 /
Published online: 7 September 2023

Abstract
The construction of high-order structure-preserving numerical schemes to solve hyperbolic conservation laws has attracted a lot of attention in the last decades and various different ansatzes exist. In this paper, we compare several completely different approaches, i.e. deep neural networks, limiters and the application of polynomial annihilation to construct high-order accurate shock capturing finite difference/volume (FD/FV) schemes. We further analyze their analytical and numerical properties. We demonstrate that all techniques can be used and yield highly efficient FD/FV methods but also come with some additional drawbacks which we point out. Our investigation of the different strategies should lead to a better understanding of those techniques and can be transferred to other numerical methods as well which use similar ideas.

Keywords Hyperbolic conservation laws · Structure-preserving · Finite difference/volume · Machine learning · Polynomial annihilation · Limiters

Mathematics Subject Classification 65M08 · 65M06

1 Introduction

Hyperbolic conservation laws play a fundamental role within mathematical models for various physical processes, including fluid mechanics, electromagnetism and wave phenomena. However, since especially nonlinear conservation laws cannot be solved analytically, numerical methods have to be applied. Starting already in 1950 with first-order finite difference

✉ Philipp Öffner
poeffner@uni-mainz.de

Dorian Hillebrand
d.hillebrand@tu-braunschweig.de

Simon-Christian Klein
simon-christian.klein@tu-braunschweig.de

[1] Institute of Mathematics, Technical University Brunswick, Brunswick, Germany

[2] Institute of Mathematics, Johannes Gutenberg University, Mainz, Germany

methods (FD), the development has dramatically increased over the last decades including finite volume (FV) and finite element (FE) ansatzes [6, 19, 53]. To use modern computer power efficiently, high-order methods are nowadays constructed and are used to obtain accurate solutions in a fast way. However, the drawback of high-order methods is that they suffer from stability issues, in particular after the development of discontinuities which is a natural feature of hyperbolic conservation laws/balance laws. Here, first-order methods are favourable since their natural amount of high dissipation results in robust methods. In addition, many first-order methods have also the property that they preserve other physical constraints like the positivity of density or pressure in the context of the Euler equations of gas dynamics. In contrast, high-order approaches need additional techniques like positivity preserving limiters, etc. [67]. Due to those reasons, researchers have combined low-order methods with high-order approaches to obtain schemes with favourable properties as applied already in [30]. The high-order accuracy of the method in smooth regions is kept, while also the excellent stability conditions and the preservation of physical constraints of the low-order methods near the discontinuities remain. Techniques in such context are e.g. Multi-dimensional Optimal Order Detection (MOOD) [9, 13], subcell FV methods [31, 60] or limiting [26, 39, 40] strategies to name some. In the last two approaches, mostly free parameters are selected/determined which mark the problematic cells where the discontinuity may live. Here, the low-order method is used whereas, in the unmarked cells, the high-order scheme still remains. To select those parameters, one uses either shock sensors [44, 47] or constraints on physical quantities (entropy inequality, the positivity of density and pressure, etc.). As an alternative to those classical ansatzes, the application of machine learning (ML) techniques as shock sensors and to control oscillations have recently driven a lot of attention [7, 10, 18, 66]. ML can be used for function approximation, classification and regression [15]. In this manuscript, we will extend those investigations in various ways.

In [37], the author has proposed a simple blending scheme that combines a high-order entropy conservative numerical flux with the low-order Godunov-type flux in a convex combination. The convex parameter is selected by a predictor step automatically to enforce that the underlying method satisfies the Dafermos entropy condition numerically. We focus on this scheme and extend the investigation from [37] in various ways. First, we propose a second blending stage to enforce the preservation of other physical constraints, e.g. the positivity of density and pressure. Further, we investigate the application of forward neural networks (NN) to specify the convex parameter. As the last approach, we apply polynomial annihilation (PA) operators described in [25]. Our investigation of the different limiting strategies should lead to a better understanding of those techniques and can be transferred to alternative approaches based on similar ideas. Finally, all of our extensions will lead to highly efficient numerical methods for solving hyperbolic conservation laws. The rest of the paper is organized as follows:

In Sect. 2, we present the one-dimensional blending scheme from [37], introduce the notation and repeat its basic properties. We further demonstrate that a fully discrete cell entropy inequality will be satisfied under certain constraints on the blending parameter. In Sect. 3, we specify the parameter selection not only taking the entropy condition into account but also other physical constraints. Here, we concentrate on the Euler equation of gas dynamics and demand the positivity of density and pressure. In Sect. 4, we repeat forward NN and how we apply them to determine the convex parameter in the extended blending scheme to obtain a highly efficient numerical scheme. In Sect. 5, the polynomial annihilation operators are finally explained and how they are used in our framework to select the blending parameter. In Sect. 6, we test all presented methods and limiting strategies and compare the results with each other. Here, we focus on the most common benchmark test

cases. We discuss the advantages and disadvantages of all the presented methods and give finally a summary with a conclusion.

2 Numerical Method for Hyperbolic Conservation Laws

2.1 Notation

We are interested in solving hyperbolic conservation laws

$$\partial_t \mathbf{u}(x,t) + \partial_x f(\mathbf{u}(x,t)) = 0, \quad x \in \Omega \subset \mathbb{R}, t > 0, \tag{1}$$

where $\mathbf{u} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^m$ are the conserved variables and f is the flux function. In this manuscript, we restrict ourselves to the one-dimensional setting for simplicity. In the case of a scalar equation, we use u instead of $\mathbf{u}$. Equation (1) will be later equipped with suitable boundary and initial conditions. Since hyperbolic conservation laws may develop discontinuities even for smooth initial data, weak solutions are considered but they are not necessarily unique. Motivated by physics, one narrows down the number of possible solutions by demanding the entropy inequality

$$\partial_t U(\mathbf{u}) + \operatorname{div} F(\mathbf{u}) \leq 0 \tag{2}$$

with convex entropy U and entropy flux F for admissible solutions. We are working in the framework of FD/FV methods, therefore different kinds of numerical fluxes are used in the paper. Please note that our methodology can be interpreted in either a finite difference (FD) or finite volume (FV) framework, depending on how the data is initialized and evaluated. However, to avoid any confusion, we will only use FV in this paper, as our schemes are based on the preliminary work introduced in [37], where the author presented his scheme in the context of FV and we will follow his notation. We denote a general numerical flux of f with f^{num}. It has two or more arguments in the following, i.e. $f^{\text{num}}(\mathbf{u}_{k-p+1}, \dots, \mathbf{u}_{k+p})$. If we apply an entropy stable flux, e.g. the Godunov flux, we denote this numerical flux by $g : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$. Using g in a classical FV methodology results in a low (first) order method. Contrary, $h : \mathbb{R}^{m \times 2p} \to \mathbb{R}^m$ denotes an entropy conservative and high-order accurate numerical flux, cf. [42, 63]. Please be aware that g and h even without the superscript " num " denote always in this paper numerical fluxes. The entropy-entropy flux pairs (U, F) are designated using uppercase letters and the notation of numerical entropy fluxes are the same as above. The numerical entropy flux $G : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is associated with a dissipative numerical flux g, also h. We use further the standard abbreviation, i.e. $g(\mathbf{u}(x_k,t), \mathbf{u}(x_{k+1},t)) = g(\mathbf{u}_k(t), \mathbf{u}_{k+1}(t)) = g_{k+\frac{1}{2}}(t)$ generalizing x_k as a way of referring to the center of cell k and $x_{k+\frac{1}{2}}$ to the right cell boundary, cf. Fig. 1.

The same procedure is used for grid points in time in the fully discrete setting, i.e. $g(\mathbf{u}(x_k,t_n), \mathbf{u}(x_{k+1},t_n)) = g(\mathbf{u}_k^n, \mathbf{u}_{k+1}^n) = g_{k+\frac{1}{2}}^n$. Please note that a $2p$ point numerical flux at position $k+\frac{1}{2}$ uses the points $\mathbf{u}_{k-p+1}, \dots, \mathbf{u}_{k+p}$, e.g. for $p=2$ we have $h(\mathbf{u}_{k-1}^n, \mathbf{u}_k^n, \mathbf{u}_{k+1}^n, \mathbf{u}_{k+2}^n) = h_{k+\frac{1}{2}}^n$. One can express the combination of numerical fluxes in a convex manner as:

$$f_{\alpha_{k+\frac{1}{2}}}^n = \alpha_{k+\frac{1}{2}} g_{k+\frac{1}{2}}^n + \left(1 - \alpha_{k+\frac{1}{2}}\right) h_{k+\frac{1}{2}}^n.$$

Working with reconstruction-free FV methods, the numerical solution in the cell is constant in space at a certain time, in short form i.e. $f_k^n = f(\mathbf{u}_k^n) = f(\mathbf{u}(x_k,t_n))$ for instance.

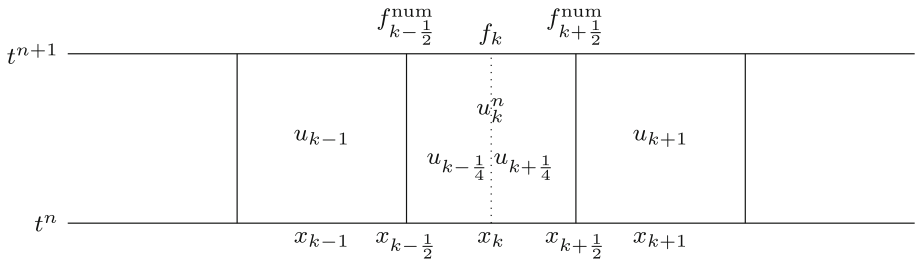

Fig. 1 The subdivision of a cell in space, initialized with the mean value of the old cell

Sometimes cells are cut in half at position x_k as described in Fig. 1. Therefore there exist cell interfaces at $x_{k-1}, x_{k-\frac{1}{2}}, x_k, x_{k+\frac{1}{2}}$ and x_{k+1} in this case. The middle points are $x_{k-\frac{3}{4}}, x_{k-\frac{1}{4}}, x_{k+\frac{1}{4}}, x_{k+\frac{3}{4}}$. These subcells are initialized with the value of the old cell. We use a uniform mesh with cell length $\Delta x = x_{k+\frac{1}{2}} - x_{k-\frac{1}{2}}$ and constant length time-steps $\Delta t = t^{n+1} - t^n$. The mesh ratio is defined by $\lambda = \frac{\Delta t}{\Delta x}$.
As mentioned above, to select the physical meaningful solution (2) has to be fulfilled. In terms of our numerical approximation, the determined solution has been constructed to imitate (2) semi-discretely or discretely, i.e. in the context of first-order FV/FD this means

$$\frac{U_k^{n+1} - U_k^n}{\Delta t} + \frac{G^n_{k+\frac{1}{2}} - G^n_{k-\frac{1}{2}}}{\Delta x} \leq 0$$

for an entropy stable numerical flux g with entropy flux G while the high-order method satisfies

$$\frac{\mathrm{d}U(u_k(t))}{\mathrm{d}t} \leq \frac{H^n_{k-\frac{1}{2}} - H^n_{k+\frac{1}{2}}}{\Delta x}.$$

If the approximated solution satisfies for all entropy pairs the corresponding inequalities, we call the scheme entropy stable and entropy dissipative if it is only fulfilled for one specific entropy pair. In the last years, many researchers have worked on the construction of entropy conservative and dissipative schemes based either on FD, FV or FE ansatzes, cf. [1, 4, 5, 11, 12, 22–24, 40, 45, 51, 52]. Here, the entropy condition is fulfilled locally.

2.2 FV Method

To explain our blending scheme, we start with the classical FV method. An FV method results from integrating the conservation law over a rectangle $\left[x_{k-\frac{1}{2}}, x_{k+\frac{1}{2}}\right] \times \left[t^n, t^{n+1}\right]$

$$\begin{aligned} u_k^{n+1} &= \int_{x_{k-\frac{1}{2}}}^{x_{k+\frac{1}{2}}} \frac{\mathbf{u}(x, t^{n+1})}{\Delta x} \mathrm{d}x \\ &= \int_{x_{k+\frac{1}{2}}}^{x_{k+\frac{1}{2}}} \frac{\mathbf{u}(x, t^n)}{\Delta x} \mathrm{d}x + \frac{1}{\Delta x} \int_{t^n}^{t^{n+1}} f\left(\mathbf{u}\left(x_{k-\frac{1}{2}}, \tau\right)\right) - f\left(\mathbf{u}\left(x_{k+\frac{1}{2}}, \tau\right)\right) \mathrm{d}\tau \qquad (3) \\ &\approx \mathbf{u}_k^n + \frac{\Delta t}{\Delta x}\left(f^{n,\text{num}}_{k-\frac{1}{2}} - f^{n,\text{num}}_{k+\frac{1}{2}}\right). \end{aligned}$$

Taking the limit $\lim_{\Delta t \to 0} \frac{1}{\Delta t}$ in (3) results in a system of ordinary differential equations (ODEs) which can be solved using e.g. Runge–Kutta (RK) schemes [58, 59]. Here, one splits between the space and time discretization also referred to as the method of lines ansatz (MOL). If only the PDE is discretized in space, we call the scheme in semi-discrete form. A different approach is based on the assumption that a numerical flux for timesteps $\Delta t = t^{n+1} - t^n$ could be devised based on knowledge of the conservation law and the local time evolution of the solution. The Cauchy Kowaleskaya expansion follows this line of thought, utilized by [28] to provide a high-order time-stepping method. The drawback of the Cauchy Kowaleskaya approach is that it typically results in lengthy calculations, complex implementations and/or implicit methods where nonlinear solvers are needed. However, we distinguish between the semi-discrete and the fully discrete schemes in the following sections. In (3) the coupling between neighbouring cells has been done via numerical fluxes $f^{n,\text{num}}_{k-\frac{1}{2}}$ to ensure the conservation property. A vast amount of numerical fluxes is known in the literature [29, 35, 41, 50, 54] and even selecting a flux is a nontrivial task [50]. Some fluxes, like the Godunov, Lax–Friedrichs, Roe and HLL fluxes, that can be interpreted by exact or approximate Riemann problem solutions, are meant to approximate the flux through some cell boundary over time Δt, i.e. being the mean value of the flux over this period. Numerical fluxes that have only a semidiscrete interpretation need some sort of high-order time integration method, and we use the family of strong stability preserving Runge–Kutta (SSPRK) methods for time integration [58]. To describe the method, we follow [37] where the considered blending FV scheme has been proposed. The method fulfills Dafermos' entropy condition [16]:

Definition 1 (*Dafermos' Criteria*) Let $\mathbf{u}$ be a weak solution of (1) and U an entropy. The total entropy in the domain Ω is given by

$$E_{\mathbf{u}}(t) = \int_{\Omega} U(\mathbf{u}(x,t))\mathrm{d}x.$$

A Dafermos entropy solution $\mathbf{u}$ is a weak solution that satisfies

$$\forall t > 0: \quad \partial_t E_{\mathbf{u}}(t) \leq \partial_t E_{\tilde{\mathbf{u}}}(t) \tag{4}$$

compared to all other weak solutions $\tilde{\mathbf{u}}$ of the conservation law (1). In essence, the entropy of the selected solution decreases faster than the entropy of all other solutions.

Definition 2 The blending scheme is based on the FV approach in a conservative form. Instead of using classical numerical fluxes in (3), a convex combination between a classical Godunov-type flux and a high-order entropy conservative flux is used instead. The combined flux, called GT-flux, is given by

$$f^n_{\alpha_{k+\frac{1}{2}}} := \alpha_{k+\frac{1}{2}}\, g\left(\mathbf{u}^n_k, \mathbf{u}^n_{k+1}\right) + \left(1 - \alpha_{k+\frac{1}{2}}\right) h\left(\mathbf{u}^n_k, \mathbf{u}^n_{k+1}\right), \tag{5}$$

where $\alpha_{k+\frac{1}{2}} \in [0,1]$ is the convex parameter.

Definition 3 While the h flux is only second-order accurate, a high-order extension can be constructed using a linear combination [42]. The corresponding linear combination of fluxes of order $2p$ is given by

$$\begin{aligned} f^n_{\alpha_{k+\frac{1}{2}}} := {} & c_1^p \left(\alpha_{k+\frac{1}{2}}\, g\left(\mathbf{u}^n_k, \mathbf{u}^n_{k+1}\right) + \left(1 - \alpha_{k+\frac{1}{2}}\right) h\left(\mathbf{u}^n_k, \mathbf{u}^n_{k+1}\right)\right) \\ & + \sum_{r=2}^{p} c_r^p \left(h\left(\mathbf{u}^n_{k-r}, \mathbf{u}^n_{k+1}\right) + \ldots + h\left(\mathbf{u}^n_{k-1}, \mathbf{u}^n_{k+r}\right)\right) \end{aligned} \tag{6}$$

Example 1 To give a concrete example, using the explicit Euler method for the time, the scheme is given by

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \frac{\Delta t}{\Delta x}\left(f^n_{\alpha_{k+\frac{1}{2}}} - f^n_{\alpha_{k-\frac{1}{2}}}\right). \tag{7}$$

To obtain higher order in time, RK methods can be used instead.

The properties of the scheme highly depend on the selected fluxes and the convex parameter $\alpha_{k+\frac{1}{2}}$. The value of $\alpha_{k+\frac{1}{2}} = \alpha(\mathbf{u}_{k-p+1}, \ldots, \mathbf{u}_{k+p})$ itself depends on $\mathbf{u}_i$ which takes the high-order stencil into account. Before we describe how α has to be selected to ensure that our scheme fulfils additionally Dafermos' criteria (4), we want to summarize the following basic properties of the scheme and the numerical fluxes:

- The GT-flux, the blending of Godunov's and Tadmor's flux, is consistent and local Lipschitz continuous [37, Lemma 1].
- The GT-flux with Tadmor's entropy conservative flux or the high-order modification from [42] satisfies as well the semidiscrete cell entropy inequality locally for the selected entropy pair used in the construction of the flux for all $\alpha \in (0, 1]$ [37, Theorem 1].
- Due to the conservation form of (7) and the convex combination of the flux, the scheme is locally conservative and the Lax–Wendroff theorem is valid due to the applications of the results from [57].

As we mentioned before, the parameter selection of α is essential for the properties of the underlying method and we repeat from [37] the following definition where also some motivation can be found:

Definition 4 We call $\alpha : \mathbb{R}^{2p\times m} \to [0, 1]$ an entropy inequality predictor with a $(2p)$ point stencil if

$$\lim_{\Delta x\to 0} \alpha(\mathbf{u}_{k-p+1}, \ldots, \mathbf{u}_{k+p}) = \begin{cases} 0 & \exists x \in [x_k - (p-1)\Delta x, x_k + p\Delta x] : \frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} < 0 \\ 1 & \forall x \in [x_k - (p-1)\Delta x, x_k + p\Delta x] : \frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0 \end{cases}$$

holds for the complete stencil. We will call the entropy inequality predictor slope limited if

$$|\alpha_k - \alpha_{k+1}| < M \quad \text{with} \quad \alpha_k = \alpha(\mathbf{u}_{k-p+1}, \ldots, \mathbf{u}_{k+p})$$

holds for some $M < 1$ and all i.

In [37], a slope entropy inequality predictor was constructed starting from a Godunov-type flux and demonstrated that it is slope limited. The predictor is given by $\alpha^n = H_{sm}\left(\frac{\frac{s_k^n}{s_{ref}} - a}{b}\right) \circledast \hat{h}$, where s_k^n is the entropy dissipative rate from the classical Godunov scheme

$$s_k^n(t) = \frac{G\left(\mathbf{u}_{k+1}^n, \mathbf{u}_k^n\right) - G\left(\mathbf{u}_k^n, \mathbf{u}_{k-1}^n\right)}{\Delta x} + \frac{U\left(\mathbf{u}_k^{n+1}\right) - U\left(\mathbf{u}_k^n\right)}{\Delta t}.$$

s_{ref} its minimum value, $\hat{h}$ the cut hat function ($h(x) = \max(0, \min(1, 2x + 2, -2x + 2))$, $H \in C^2$ the smooth step function

$$H_{sm}(x) = \begin{cases} 0 & x \leq 0 \\ 6x^5 - 15x^4 + 10x^3 & 0 \leq x \leq 1 \\ 1 & 1 \leq x, \end{cases}$$

and $\circledast$ denotes the discrete sup-mollification

$$(f \circledast g)|_{[i/n,(i+1)/n]} = \max_{j \in \{0,\dots,n-1\}} f_j g_{i-j} \text{ for } i = 0, \dots, n-1$$

and step functions f, g.

Remark 1 Instead of working with the classical Godunov flux in (2), we use approximated Riemann solvers. In [37], the local Lax–Friedrich flux (LLF) (Rusanov) has been used and shows good results. Due to that, we apply always the LLF flux in the numerical Sect. 6 to obtain a more efficient method. Finally, via a tensor-structure ansatz, an extension of the approach to two or three dimensions is straightforward and all of the results transfer.

As demonstrated in [37], the formal order of the scheme depends on the used high-order flux and the behaviour of α. If the high-order flux is of order $2p$ and α tends for a smooth solution to zero (respectively with order $2p$ or higher), the hybrid scheme has order $2p$.

2.3 Local Entropy Inequality

In the following subsection, we extend the investigation of [37]. We demonstrate that a hybrid scheme constructed with a discrete entropy dissipative flux g and any consistent flux h satisfies a fully discrete entropy inequality locally under certain restrictions on α. Inside the definitions, we refer again to Fig. 1 for the nomenclature. Let

$$p_k^n := \frac{H\left(\mathbf{u}_{k-p+1}, \dots, \mathbf{u}_{k+p}\right) - H\left(\mathbf{u}_{k-p}, \dots, \mathbf{u}_{k+p-1}\right)}{\Delta x} + \frac{U\left(\mathbf{u}_k^{n+1}\right) - U\left(\mathbf{u}_k^n\right)}{\Delta t}$$

be the entropy production of our high-order scheme on cell k. We may divide cell k into two subcells centerd around $x_{k-\frac{1}{4}}$ and $x_{k+\frac{1}{4}}$ and can now define the entropy production inside these subcells via

$$s_{k+\frac{1}{4}}^n := \frac{G\left(\mathbf{u}_{k+1}^n, \mathbf{u}_k^n\right) - F(\mathbf{u}_k^n)}{\frac{\Delta x}{2}} + \frac{U\left(\mathbf{u}_{k+\frac{1}{4}}^{n+1}\right) - U\left(\mathbf{u}_{k+\frac{1}{4}}^n\right)}{\Delta t}$$

$$p_{k+\frac{1}{4}}^n := \frac{H\left(\mathbf{u}_{k-p+1}, \dots, \mathbf{u}_{k+p}\right) - F(\mathbf{u}_k^n)}{\frac{\Delta x}{2}} + \frac{U\left(\mathbf{u}_{k+\frac{1}{4}}^{n+1}\right) - U\left(\mathbf{u}_{k+\frac{1}{4}}^n\right)}{\Delta t}$$

We can now define **Condition** F:

Definition 5 The parameter α is said to satisfy **Condition** F for cell k if

$$\alpha_{k+\frac{1}{2}} s_{k+\frac{1}{4}}^n + \left(1 - \alpha_{k+\frac{1}{2}}\right) p_{k+\frac{1}{4}}^n \leq 0 \text{ and } \alpha_{k-\frac{1}{2}} s_{k-\frac{1}{4}}^n + \left(1 - \alpha_{k-\frac{1}{2}}\right) p_{k-\frac{1}{4}}^n \leq 0$$

holds for the left and right interfaces.

We can prove:

Lemma 1 ***Condition*** *F is fulfilled for cell k if one of the following conditions is satisfied on each interface, i.e. for $k+\frac{1}{4}$ and $k-\frac{1}{4}$.*

1. *It holds $s^n_{k+\frac{1}{4}} \leq 0$ and $p^n_{k+\frac{1}{4}} \leq 0$, $\alpha \in [0,1]$ is arbitrary.*
2. *It holds $s^n_{k+\frac{1}{4}} \leq 0$ and $p^n_{k+\frac{1}{4}} > 0$ and $\alpha \geq \frac{p^n_{k+\frac{1}{4}}}{p^n_{k+\frac{1}{4}} - s^n_{k+\frac{1}{4}}}$.*

Proof The first condition is obvious. For the second one the following calculation

$$\alpha s + (1-\alpha)p \leq 0 \iff \alpha(s-p) + p \leq 0 \iff \alpha \geq \frac{p}{p-s} \geq 0$$

with suppressed indices shows the result. □

If one can guarantee that $s^n_{k+\frac{1}{4}} < 0$, we can calculate a lower bound on α to enforce **Condition** F and we can prove the following theorem which combines ideas of [62] and [37]:

Theorem 1 *We consider the hybrid scheme*

$$\mathbf{u}^{n+1}_k = \mathbf{u}^n_k + \lambda \left(f^n_{\alpha_{k-\frac{1}{2}}} - f^n_{\alpha_{k+\frac{1}{2}}} \right) \tag{8}$$

with numerical flux $f^n_{\alpha_{k+\frac{1}{2}}} = \alpha_{k+\frac{1}{2}} g^n_{k+\frac{1}{2}} + \left(1-\alpha_{k+\frac{1}{2}}\right) h^n_{k+\frac{1}{2}}$. If $\alpha_{k+\frac{1}{2}}$ fulfils Condition F for both cell boundaries and the CFL restriction is half that of the minimum of either flux, the scheme (8) satisfies a discrete cell entropy inequality with the numerical entropy flux $F^{\text{num}}(\mathbf{u}_{k-p+1}, \ldots, \mathbf{u}_{k+p}) = \alpha_{k+\frac{1}{2}} G(\mathbf{u}_k, \mathbf{u}_{k+1}) + \left(1-\alpha_{k+\frac{1}{2}}\right) H(\mathbf{u}_{k-p+1}, \ldots, \mathbf{u}_{k+p})$

Proof We first state that the cell mean u^{n+1}_k can be written as the average value

$$\begin{aligned}
\mathbf{u}^{n+1}_k &= \mathbf{u}^n_k + \lambda \left(f^{n,\text{num}}_{\alpha_{k-\frac{1}{2}}} - f^{n,\text{num}}_{\alpha_{k+\frac{1}{2}}} \right) \\
&= \frac{\mathbf{u}^n_k + 2\lambda \left(f^{n,\text{num}}_{\alpha_{k-\frac{1}{2}}} - f\left(\mathbf{u}^n_k\right) \right) + \mathbf{u}^n_k + 2\lambda \left(f\left(\mathbf{u}^n_k\right) - f^{n,\text{num}}_{\alpha_{k+\frac{1}{2}}} \right)}{2} \\
&= \frac{\alpha_{k-\frac{1}{2}} \left(\mathbf{u}^n_k + 2\lambda \left(g^n_{k-\frac{1}{2}} - f\left(\mathbf{u}^n_k\right) \right) \right) + \left(1-\alpha_{k-\frac{1}{2}}\right) \left(\mathbf{u}^n_k + 2\lambda \left(h^n_{k-\frac{1}{2}} - f\left(\mathbf{u}^n_k\right) \right) \right)}{2} \\
&\quad + \frac{\alpha_{k+\frac{1}{2}} \left(\mathbf{u}^n_k + 2\lambda \left(f\left(\mathbf{u}^n_k\right) - g^n_{k+\frac{1}{2}} \right) \right) + \left(1-\alpha_{k+\frac{1}{2}}\right) \left(\mathbf{u}^n_k + 2\lambda \left(f\left(\mathbf{u}^n_k\right) - h^n_{k+\frac{1}{2}} \right) \right)}{2} \\
&= \frac{\mathbf{u}^{n+1}_{k-\frac{1}{4}} + \mathbf{u}^{n+1}_{k+\frac{1}{4}}}{2}
\end{aligned}$$

of two schemes. Finally, we can conclude that the entropy of cell k satisfies

$$
\begin{aligned}
U(\mathbf{u}_k^{n+1}) - U(\mathbf{u}_k^n) + \lambda(F^n_{\alpha_{k+\frac{1}{2}}} - F^n_{\alpha_{k-\frac{1}{2}}}) &\leq \frac{U\left(\mathbf{u}^{n+1}_{k-\frac{1}{4}}\right) + U\left(\mathbf{u}^{n+1}_{k+\frac{1}{4}}\right)}{2} \\
&- U\left(\mathbf{u}_k^n\right) + \lambda\left(F^n_{\alpha_{k+\frac{1}{2}}} - F^n_{\alpha_{k-\frac{1}{2}}}\right) \\
&= \frac{U\left(\mathbf{u}^{n+1}_{k-\frac{1}{4}}\right) - U\left(\mathbf{u}_k^n\right) + 2\lambda\left(F_k^n - F^n_{\alpha_{k-\frac{1}{2}}}\right)}{2} + \frac{U\left(\mathbf{u}^{n+1}_{k+\frac{1}{4}}\right) - U\left(\mathbf{u}_k^n\right) + 2\lambda\left(F^n_{\alpha_{k+\frac{1}{2}}} - F_k^n\right)}{2} \\
&\leq \frac{\alpha_{k-\frac{1}{2}}}{2}\left(U\left(\mathbf{u}_k^n + 2\lambda\left(g^n_{k-\frac{1}{2}} - f\left(\mathbf{u}_k^n\right)\right)\right) - U(\mathbf{u}_k^n) + 2\lambda\left(F_k^n - G^n_{k-\frac{1}{2}}\right)\right) \\
&+ \frac{1-\alpha_{k-\frac{1}{2}}}{2}\left(U\left(\mathbf{u}_k^n + 2\lambda\left(h^n_{k-\frac{1}{2}} - f\left(\mathbf{u}_k^n\right)\right)\right) - U(\mathbf{u}_k^n) + 2\lambda\left(F_k^n - H^n_{k-\frac{1}{2}}\right)\right) \\
&+ \frac{\alpha_{k+\frac{1}{2}}}{2}\left(U\left(\mathbf{u}_k^n + 2\lambda\left(f\left(\mathbf{u}_k^n\right) - g^n_{k+\frac{1}{2}}\right)\right) - U(\mathbf{u}_k^n) + 2\lambda\left(G^n_{k+\frac{1}{2}} - F_k^n\right)\right) \\
&+ \frac{1-\alpha_{k+\frac{1}{2}}}{2}\left(U\left(\mathbf{u}_k^n + 2\lambda\left(f\left(\mathbf{u}_k^n\right) - h^n_{k+\frac{1}{2}}\right)\right) - U(\mathbf{u}_k^n) + 2\lambda\left(H^n_{k+\frac{1}{2}} - F_k^n\right)\right) \\
&= \frac{\alpha_{k-\frac{1}{2}}}{2}s^n_{k-\frac{1}{4}} + \frac{1-\alpha_{k-\frac{1}{2}}}{2}p^n_{k-\frac{1}{4}} + \frac{\alpha_{k+\frac{1}{2}}}{2}s^n_{k+\frac{1}{4}} + \frac{1-\alpha_{k+\frac{1}{2}}}{2}p^n_{k+\frac{1}{4}} \leq 0,
\end{aligned}
$$

because U is convex. □

If one can enforce **Condition** F, we obtain a fully discrete entropy dissipative scheme by choosing an appropriate α. Note that the bound is sufficient but not necessary. We will now focus on the Euler equation of gas dynamics. There, we can apply the same technique to enforce also the positivity of pressure and density. Note that the above proof works for any combination of a discrete entropy stable flux and another flux. There is no need for two-point fluxes and we can use a high-order flux for h.

3 Positivity of Pressure and Density for the Euler Equations

Instead of focusing on the entropy inequality, we can apply the same mechanism to enforce positivity of pressure (internal energy) and/or density for numerical solutions of the Euler equations of gas dynamics [27]:

$$
\mathbf{u} = (\rho, \rho v, E)^T, \quad f(\rho, \rho v, E) = \begin{bmatrix} \rho v \\ \rho v^2 + p \\ v(E+p) \end{bmatrix}, \quad p = (\gamma - 1)\left(E - \frac{1}{2}\rho v^2\right), \tag{9}
$$

where ρ denotes the density, v the velocity, E the total energy, p the pressure and $\gamma > 1$ the adiabatic constant. This system is equipped with the following entropy-entropy flux pair

$$
U(\rho, \rho v, E) = -\rho S \quad F(\rho, \rho v, E) = -\rho v S \quad S = \ln(p\rho^{-\gamma}). \tag{10}
$$

One can define the set of admissible states including density, momentum and total energy. This set is convex under standard assumptions on the thermodynamics variables. Due to this, the pressure is obviously a concave function. We are interested in preserving the positivity of the

pressure and density. Therefore, we can equally enforce the negativity of the negative pressure and negative density. Indeed, this task is equivalent to enforce upper bounds on convex functionals. It is well-known [48, 67] that Godunov and (local) Lax–Friedrichs schemes are positivity preserving under a suitable CFL number. In the following g will stand for the flux of a positivity preserving dissipative scheme and our convex functionals that should be enforced are denoted by $c_1(u) = -p(u)$ and $c_2 = -\rho(u)$. The counterparts of **Condition** F are **Condition** ρ and **Condition** P.

Definition 6 The parameter α for cell k is said to satisfy

- **Condition P** if

$$c_1\left(\mathbf{u}_k^n + 2\lambda\left(f^{n,\text{num}}_{\alpha_{k-\frac{1}{2}}} + f(\mathbf{u}_k^n)\right)\right) \le 0, \quad c_1\left(\mathbf{u}_k^n + 2\lambda\left(f_k^n - f^{n,\text{num}}_{\alpha_{k+\frac{1}{2}}}\right)\right) \le 0.$$

- **Condition** ρ if

$$c_2\left(\mathbf{u}_k^n + 2\lambda\left(f^{n,\text{num}}_{\alpha_{k-\frac{1}{2}}} + f(\mathbf{u}_k^n)\right)\right) \le 0, \quad c_2\left(\mathbf{u}_k^n + 2\lambda\left(f_k^n - f^{n,\text{num}}_{\alpha_{k+\frac{1}{2}}}\right)\right) \le 0.$$

We can derive similar conditions to ensure the positivity of pressure and density using the technique from Subsection 2.3. We demonstrate it here for the pressure (using **Condition P**). The same steps lead also to a condition for the density. First, we obtain an equivalent lemma to Lemma 1:

Lemma 2 ***Condition P*** *is fulfilled if one of the following conditions is satisfied for* $k + \frac{1}{4}$ *and* $k - \frac{1}{4}$*:*

1. *It holds* $c_1\left(\mathbf{u}_k^n + 2\lambda\left(h^n_{k+\frac{1}{2}} - f_k^n\right)\right) \le 0$ *and* $\alpha \in [0, 1]$ *is arbitrary.*
2. *It holds* $c_1\left(\mathbf{u}_k^n + 2\lambda\left(h^n_{k+\frac{1}{2}} - f_k^n\right)\right) > 0$ *and*

$$\alpha \ge \frac{c_1\left(\mathbf{u}_k^n + 2\lambda\left(h^n_{k+\frac{1}{2}} - f_k^n\right)\right)}{c_1\left(\mathbf{u}_k^n + 2\lambda\left(h^n_{k+\frac{1}{2}} - f_k^n\right)\right) - c_1\left(\mathbf{u}_k^n + 2\lambda\left(g^n_{k+\frac{1}{2}} - f_k^n\right)\right)}$$

.

Proof A dissipative scheme implies $c_1\left(\mathbf{u}_k^n + 2\lambda\left(g^n_{k-\frac{1}{2}} - f_k^n\right)\right) \le 0$. We get

$$\begin{aligned} &c_1\left(\mathbf{u}_k^n + 2\lambda(f^{n,\text{num}}_{\alpha_{k-\frac{1}{2}}} - f_k^n)\right) \\ =&c_1\left(\alpha\left(\mathbf{u}_k^n + 2\lambda\left(g^n_{k-\frac{1}{2}} - f_k^n\right)\right) + (1-\alpha)\left(\mathbf{u}_k^n + 2\lambda\left(h^n_{k-\frac{1}{2}} - f_k^n\right)\right)\right) \qquad (11)\\ \le&\alpha c_1\left(\mathbf{u}_k^n + 2\lambda\left(g^n_{k-\frac{1}{2}} - f_k^n\right)\right) + (1-\alpha)c_1\left(\mathbf{u}_k^n + 2\lambda\left(h^n_{k-\frac{1}{2}} - f_k^n\right)\right) \le 0 \end{aligned}$$

due to the convex combination. We obtain the same for $\alpha_{k+\frac{1}{2}}$. The second part follows directly from (11). □

Lemma 3 *Under* ***Condition P****, we get* $p\left(u_k^n + \lambda\left(f^{n,\text{num}}_{\alpha_{k-\frac{1}{2}}} - f^{n,\text{num}}_{\alpha_{k+\frac{1}{2}}}\right)\right) \ge 0.$

Proof Due to the convexity, we obtain

$$\begin{aligned}
-p\left(u_k^{n+1}\right) &= c_1\left(\mathbf{u}_k^{n+1}\right) = c_1\left(\mathbf{u}_k^n + \lambda\left(f^{n,\mathrm{num}}_{\alpha_{k-\frac{1}{2}}} - f^{n,\mathrm{num}}_{\alpha_{k+\frac{1}{2}}}\right)\right) \\
&= c_1\left(\frac{\mathbf{u}_k^n + 2\lambda\left(f^{n,\mathrm{num}}_{\alpha_{k-\frac{1}{2}}} - f_k^n\right) + \mathbf{u}_k^n + 2\lambda\left(f_k^n - f^{n,\mathrm{num}}_{\alpha_{k+\frac{1}{2}}}\right)}{2}\right) \\
&\leq \frac{1}{2}\left(c_1\left(\mathbf{u}_k^n + 2\lambda\left(f^{n,\mathrm{num}}_{\alpha-\frac{1}{2}} - f_k^n\right)\right) + c_1\left(\mathbf{u}_k^n + 2\lambda\left(f_k^n - f^{n,\mathrm{num}}_{\alpha+\frac{1}{2}}\right)\right)\right) \\
&\leq 0.
\end{aligned}$$

□

As mentioned above, we obtain similar results for the density, actually for every convex functional that is bounded by the low order scheme.

Finally, we like to mark that conditions on the physical constraints are not new and used in many different approaches, cf. [40, 55].

4 Limiting via Neural Networks

4.1 Basics of Feedforward Networks

In this section, we explain how we select our numerical flux using feed-forward neural networks (FNN). The network is used to determine the local indicator α which steers our convex combination inside the numerical flux and to determine if the high-order or low-order part of the scheme is used at a certain point. Further, it is clear if the solution is not entropy conservative, it is also not continuous. It is a generic example of a high-dimensional function interpolation. We further assume that the parameter depends continuously on the input space and then the theoretical foundation for our approach is based on the following result[1] from [15]:

Theorem 2 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a sigmoidal function. Then the finite sum of the form $\mathcal{A} \circ G(x) = \sum_{j=1}^{N} \alpha_j \sigma(\langle y_j, x\rangle + b_j)$ are dense in $(C(I_n), \|\cdot\|_\infty)$.*

This theorem motivates the usage of FNN to approximate any function

$$\mathcal{C} \subset C(\mathbb{R}^n, \mathbb{R}). \tag{12}$$

To explain the approach, we give a short presentation of the general theory of neural networks (NN). Our FNN is on a particular example and it is set up in a sequence of layers containing a certain amount of neurons (computing units). The first layer (input/source layer) is handling the input data/signal to the network. The output layer (last layer) uses the information from the NN and build output data, e.g. function expressions which are used in the following, e.g. α in our case. Hidden layers are laid in between where all calculations are done. A FNN with depth K contains $K-1$ hidden layers and one output layer. What happens in the network is the following operation: For an input signal $\mathbf{X} \in \mathbb{R}^n$, we have the output:

$$\tilde{\mathbf{Y}} = \mathcal{F} \circ G_k \circ \mathcal{A} \circ G_{k-1}\mathcal{A} \circ G_{k-2} \circ \cdots \circ G_1(\mathbf{X}), \tag{13}$$

[1] We refer also to [49] for more intricate results on the approximation properties of neural networks.

where G_k denotes the affine transformation of the $k-$layer on a vector $\mathbf{Z} \in \mathbb{R}^{N_{k-1}}$ with

$$G_k(\mathbf{Z}) = \mathbf{W}_k\mathbf{Z} + \mathbf{b}_k, \qquad \mathbf{W}_k \in \mathbb{R}^{N_k \times N_{k-1}}, \quad \mathbf{b}_k \in \mathbb{R}^{N_k}. \tag{14}$$

$\mathbf{W}_k$ are the weights matrices and $\mathbf{b}_k$ are the bias vectors. Both contain trainable parameters. Further, in (13), $\mathcal{A}$s are non-linear activation functions and $\mathcal{F}$ is a non-linear output function that transforms the output data into a suitable form. There are a number of different activation functions for different problems, cf. [20] for a survey and an overview. In our manuscript, we restrict ourselves to the currently popular **Exponential Linear Units** (ELU) function [14]

$$\mathrm{ELU}(t) = \begin{cases} x, & x > 0, \\ \gamma(\exp(x) - 1), & \text{else.} \end{cases} \tag{15}$$

We set $\gamma \equiv 1$ in our numerical simulations.

To approximate finally (12) with our network (13), we must train the parameters using our training data. Therefore, we first create a set of training data with N_T samples

$$T = \{(\mathbf{X}_i, \mathbf{Y}_i) : \mathbf{Y}_i = \mathcal{C}(\mathbf{X}_i) \forall i = 1, \ldots, N_T\}.$$

Then, we define a suitable cost/loss function that measures the discrepancy between the actual result vector $\mathbf{Y}$ and the predicted result vector $\tilde{\mathbf{Y}}$. We apply always the mean square error $L(Y, \tilde{Y}) = \frac{\sum_{i=1}^{N_T}(Y_i - \tilde{Y}_i)^2}{N_T}$, as loss function. To train the network, we minimize the loss function concerning the parameters $\{\mathbf{W}_k, \mathbf{b}_k\}_k$ over the set of training data. For the minimization process, we use an iterative optimization algorithm in our case the ADAM minimizer [36].

Remark 2 (Overfitting and Dropout Layer) As mentioned *inter alia* in [18], the training set has to be selected quite carefully to avoid over-fitting. In such a case, the network performs poorly on general data since it is highly optimized for the training set. To avoid this problem, a regularization technique is used. A popular regularization strategy is using a drop-out layer [61]. During each optimization update step in the training phase of the network, a dropout layer in front of the k-th layer randomly sets a predefined fraction of the components of the intermediate vector computed by the k-th layer to zero. The advantages of this technique are that the training is not biased towards a specific network architecture, additional stochasticity is injected into the optimization process to avoid getting trapped in local optima, and a sparsity structure is introduced into the network structure.

4.2 Data Driven Scheme for Conservation Laws

Our method using neuronal nets is based on the following approach. We use neuronal nets as building blocks to approximate unknown real maps in the following recipe:

1. Select a random set of initial conditions $\mathcal{I} = \{u_1, u_2, \ldots, u_N\}$ of Riemann problems.
2. Calculate high quality numerical solutions v to this set $\mathcal{I}$.
3. Determine projections u of these solutions v to a low resolution finite volume mesh.
4. Calculate the flux of v over the given mesh boundaries and in a suitable **time interval** to high accuracy.
5. Infer suitable values for the convex combination parameter α **for the high order extension** (6).
6. Use this database to train a NN as a predictor for the unknown map $\alpha(\mathbf{u}, \Delta t)$.

The high-quality numerical solution v was calculated using classic FV methods on fine grids. The projection of these solutions to a low-resolution mesh is given by

$$\mathbf{u}_k = \frac{1}{\Delta x}\int_{x_{k-\frac{1}{2}}}^{x_{k+\frac{1}{2}}} v(x,t)\mathrm{d}x \quad \text{with} \quad v(x,t) = \sum_k v_k(t)\chi_{\omega_k}(x).$$

Here, ω_k denotes the cell k of the fine grid and v_k the mean value of the solution as approximated by an FV method. The calculation of an accurate numerical flux approximation $f^{n,\text{precise}}$ at the interfaces of the coarse grid is based on numerical quadrature in time, i.e.

$$f^{n,\text{precise}}_{k+\frac{1}{2}} = \mathrm{I}^{t^{n+1}}_{t^n} g\left(v\left(x^-_{k+\frac{1}{2}},\cdot\right), v\left(x^+_{k+\frac{1}{2}},\cdot\right)\right) \approx \int_{t^n}^{t^{n+1}} f(v(x,t))\mathrm{d}t.$$

In our numerical tests, we use low-order quadrature methods as we are especially interested in flux values for non-smooth $\mathbf{u}$. Therefore, there is no need for high-order quadrature rules. Our next problem consists of finding a suitable and well-defined $\alpha_{k+\frac{1}{2}}$ that satisfies $f^{n,\text{neural}}_{\alpha_{k+\frac{1}{2}}} \approx f^{n,\text{precise}}_{k+\frac{1}{2}}$. We define

$$f^{n,\text{neural}}_{\alpha_{k+\frac{1}{2}}} := \min_{f\in\text{conv}\left(h^n_{k+\frac{1}{2}}, g^n_{k+\frac{1}{2}}\right)} \left\| f - f^{n,\text{precise}}_{k+\frac{1}{2}} \right\|_2 = \mathrm{P}_{\text{conv}\left(h^n_{k+\frac{1}{2}}, g^n_{k+\frac{1}{2}}\right)} f^{n,\text{precise}}$$

of the target value of the neural network GT flux as the solution of a constrained optimization problem. It is the projection (denoted by P) of the flux to the convex hull of the dissipative low-order and non-dissipative high-order fluxes. This formulation is usable in scalar conservation laws as well as for systems[2]. Since the domain is convex as well as the objects, the above minimization problem has a unique solution. However, the situation is worse if we focus on

$$\alpha = \arg\min_{\tilde{\alpha}\in[0,1]} \left\| f^{n,\text{num}}_{\tilde{\alpha}} - f^{n,\text{precise}}_{k+\frac{1}{2}} \right\|_2.$$

instead. Obviously, for $g = h$ which occurs $u = \text{const.}$, we do have not a unique solution. We make use of the following ansatz

$$\alpha = \max\left(\arg\min_{\tilde{\alpha}\in[0,1]} \left\| f^{n,\text{num}}_{\tilde{\alpha}} - f^{n,\text{precise}} \right\|_2\right)$$

to select the most dissipative value of α in the degenerate case. The numerical solution using the 2-norm is based on the application of the Penrose inverse $b = f^{n,\text{precise}} - g,\ A = h - g,\ \beta = \min(1, \max(0, A^\dagger b)),\ \alpha = 1 - \beta$.

The affine-linear map

$$M_{k+\frac{1}{2}} : \mathbb{R} \to \mathbb{R}^m, \beta \mapsto \beta h_{k+\frac{1}{2}} + (1-\beta)g_{k+\frac{1}{2}} = g_{k+\frac{1}{2}} + \alpha(h_{k+\frac{1}{2}} - g_{k+\frac{1}{2}}) = w + A\alpha$$

can be expressed in the standard basis using the matrix $A_{k+\frac{1}{2}} = h_{k+\frac{1}{2}} - g_{k+\frac{1}{2}}$ and the support vector $w_{k+\frac{1}{2}} = g_{k+\frac{1}{2}}$. The value β controls an affine combination, where $\tilde{\beta} = 1 - \alpha$ yields the identical value as before using the blending scheme. We finally get

$$\arg\min \left\| w + A\beta - f^{n,precise} \right\|_2 = \arg\min \left\| A\beta - \underbrace{(f^{n,precise} - w)}_{b} \right\|_2 = A^\dagger b$$

[2] A different norm or a different convex functional could be also used instead. Such investigations will be left for future research.

for the projection of $f^{n,precise}$ onto the subspace ran M. As the Penrose inverse is not only the least squares but also the least norm solution. It has also the smallest absolute value, i.e. $\beta = 0$ in the case that A is degenerate. The distinction between α and β was made to enforce $\alpha = 1$ for degenerate A. We are interested in the projection of $f^{n,precise}$ onto $M_{k+\frac{1}{2}}([0, 1])$. If the unconstrained minimizer lies outside of the image of $[0, 1]$ under M, the constrained minimizer must be on one of the edges and in fact, the edge lying nearer to the unconstrained minimizer. This yields the given formula for the minimizer.

5 Polynomial Annihilation Based Scheme

In this chapter, we want to propose another possibility to approximate the blending parameter α. We apply polynomial annihilation (PA) operators. First, we explain their construction in one spatial dimension. These operators approximate the jump function of a given sensing variable. We use them to select α.

5.1 Polynomial Annihilation-Basic Framework

The general idea of PA operators proposed in [8] is to approximate the jump function

$$[s](x) = s(x^+) - s(x^-) \tag{16}$$

for a given $s : \Omega \to \mathbb{R}$ called sensing variable. We want to construct an operator $L_m[s](\xi)$ approximating $[s](\xi)$ with m-th order of accuracy. For a given $\xi \in \Omega$, we first choose a stencil of $m + 1$ grid points around ξ. It is $S_\xi = (x_k, \ldots, x_{k+m})$ with $x_k \le \xi \le x_{k+m}$. In the next step, the annihilation coefficients c_j are defined implicitly by

$$\sum_{x_j \in S_\xi} c_j(\xi) p_l(x_j) = p_l^{(m)}(\xi). \tag{17}$$

Here, $\{p_l\}_{l=0}^m$ is any selected basis of the space of polynomials with degree $\le m$.

Finally, a normalization factor q_m is calculated by $q_m = \sum_{x_j \in S_\xi^+} c_j(\xi)$, with $S_\xi^+ = \{x_j \in S_\xi | x_j \ge \xi\}$. For a fixed choice of S_ξ, q_m is constant Finally, we can define the PA operator of order m by

$$L_m[s](\xi) := \frac{1}{q_m} \sum_{x_j \in S_\xi} c_j(\xi) s(x_j). \tag{18}$$

In [8], it was shown that

$$L_m[s](\xi) = \begin{cases} [s](\tilde{x}) + \mathcal{O}\left(\tilde{h}(\xi)\right), & \text{if } x_{j-1} \le \xi, \tilde{x} \le x_j, \\ \mathcal{O}\left((\tilde{h}(\xi))^{\min(m,l)}\right), & \text{if } s \in C^l([x_k, x_{k+m}]), \end{cases}$$

where $\tilde{x}$ denotes a jump discontinuity of s and $\tilde{h}(\xi) := \max\{|x_i - x_{i-1}| \mid x_i, x_{i-1} \in S_\xi\}$. To demonstrate the behaviour of (18), we give the following example from [8]:

Example 2 We are considering the function

$$f(x) = \begin{cases} \cos(3\pi x), & -1 \le x < 0, \\ \frac{2}{1+3\exp(-50x+25)} - 1, & 0 < x \le 1. \end{cases} \tag{19}$$

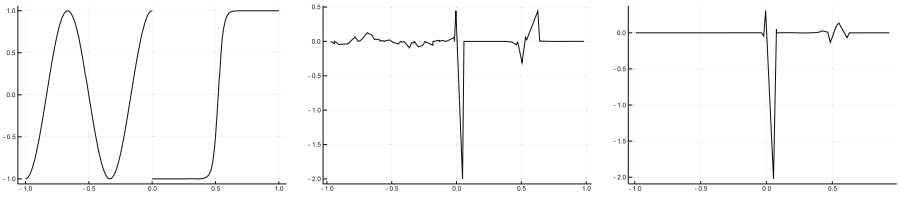

Fig. 2 Reference function f an two approximations of the corresponding jump function with $m = 3$ and $m = 5$

and visualize the PA operator $L_m[f]$ using 100 randomly chosen points. It is supposed to approximate the corresponding jump function

$$[f](x) \approx \begin{cases} -2, & x = 0, \\ 0, & \text{else.} \end{cases} \tag{20}$$

In Fig. 2, the function f (left picture) and the PA operators with order $m = 3$ (middle picture) and $m = 5$ (right picture) are presented. It can be recognized that the jump is approximated fine. However, we have an overshoot in both cases around $x = 0.5$.

5.2 Scheme Based on Polynomial Annihilation

Based on the above-presented framework, we construct the convex parameter α. In regions with a smooth solution, α should be zero whereas in cells with a discontinuity, α should be one. This can be achieved by PA operators which are not constructed to give the location of a discontinuity but to approximate the height of the jump at that location. Hence, we need to normalize the operator by a factor approximating the height of a typical jump, i.e. $\frac{1}{z} L_{2q}[s]$ with a normalization factor $z \approx [s](\tilde{x})$. Here, the PA operator is used on the $2q$-point stencil $(x_{k-q+1}, \ldots, x_{k+q})$ and the corresponding mean values $(u^n_{k-q+1}, \ldots, u^n_{k+q})$ for a given n. This normalization factor z is also provided by a PA operator. Therefore, we apply L_{2q} to the idealized values $(u^n_{\max}, \ldots, u^n_{\max}, u^n_{\min}, \ldots, u^n_{\min})$ based on the same $2q$-point stencil with $u^n_{\max} = \max\{u^n_{k-q+1}, \ldots, u^n_{k+q}\}$, $u^n_{\min} = \min\{u^n_{k-q+1}, \ldots, u^n_{k+q}\}$. Using this normalization factor, the natural selection of α is $\alpha = \frac{L_{2q}[u]}{z}$. However, this choice does not fulfil the before mentioned recommended property since the normalization gives a much more accurate approximation of the jump height. By using $L_{2q}[u]$ on $(u^n_{k-q+1}, \ldots, u^n_{k+q})$ instead, we obtain an approximation of the jump function with a lower total height. Another occurring problem is that the normalization factor z vanishes. It is equal to zero if $u^n_{\max} = u^n_{\min}$. A possible solution for both issues can be obtained by simple regularization. We choose $\alpha^n = \frac{c_1 L_{2q}[u]}{z + c_2}$, with $c_2 > 0$. Our experiments will show that $c_1 = 10$ is an appropriate choice to compensate for the difference between the accuracies of the approximations. The regularization is picked as $c_2 = \|\mathbf{u}\|_1 / \mu(\Omega)$ with discrete L^1-norm $\|\mathbf{u}^n\|_1 = \sum_{i=1}^N \frac{|u^n_i|}{N} \mu(\Omega)$. In the numerical section, we select PA operators using $q = 4$ and in the system case, we determine the value of α^n by the maximum of the separately calculated values of each conserved quantity. Finally, we apply a sup-mollification to define the predictor $\tilde{\alpha}^n := \alpha^n \circledast \max\left\{1 - \frac{1}{3} \left\| \frac{x}{\Delta x} \right\|, 0\right\}$. The final step is motivated by the fact that the PA operator may introduce overshoots even in smooth regions, as demonstrated already in Fig. 2. By applying mollification, we observed a reduction in their impact and achieved more precise results.

6 Numerical Experiments

In the following part, we determine the blending parameter by the techniques described in Sect. 2.3–5. We investigate and compare the different methods and focus especially on the following questions:

- Which order of accuracy can we expect from our schemes?
- Are the schemes able to capture strong shocks and are they oscillation free?
- Do we obtain the structure-preserving properties?
- Is there a most efficient technique which should be applied?

We test our schemes on the Euler equations of gas dynamics (9). To analyze the accuracy of the schemes, we consider the smoothly connected density variation from [37]. For the more advanced simulations, we concentrate on some well-known benchmark problems from literature [58, 59, 64]. We consider in detail: Sod's shock tube, the second shock tube problem, 123-problem, the Woodward–Colella blast wave and the Shu–Osher test case.
We compare different hybrid schemes with each other. Especially, the selection of α is essential. In preliminary experiments, we have recognized that the constraints on pressure and density as developed in Sect. 3 have the lowest effect on the blending parameter compared to the other choices, e.g. cell entropy, Dafermos entropy condition, NN or PA operators. However, to ensure the positivity of pressure and density from **Condition** P and **Condition** ρ, we use α_ρ and α_P as lower bounds. We set $\alpha = \max(\alpha_\rho, \alpha_p, \alpha_i)$ where α_i is calculated by one of the above mentioned techniques. Inside the schemes, we use for the high-order fluxes a fourth-order entropy conservative flux with SSPRK(3,3) if nothing else is said. The low order flux in (6) is the local Lax–Friedrichs flux. We consider the following schemes:

1. A data-driven scheme denoted by DDLFT: α is determined using FNN. We use $\alpha = \max(\alpha_\rho, \alpha_p, \alpha_{\mathrm{DD}})$ where α_{DD} is the output of our FNN.
2. A polynomial annihilation based scheme called PALFT: α is determined through the technique described in Sect. 5. Once more, it is $\alpha = \max(\alpha_\rho, \alpha_p, \alpha_{\mathrm{PA}})$.
3. A cell entropy dissipative scheme (DELFT): α is determined through the technique described in Sect. 2.3. Unluckily in our numerical simulations, we have realized that by the selection of fluxes and time integration, we obtain always an order reduction. One can possibly avoid and circumvent this using additional techniques like additional shock detectors and FV subcell limiting strategies, cf. [46], etc. but this is not part of the current paper where we stress also the drawbacks out. We adapted the method (fluxes, time-integration) and the scheme uses an SSPRK(2,2) Predictor-Corrector time integration, written as flux, and two-point fluxes, cf. Appendix 8. This setting gives us a second-order scheme which demonstrates the promising results in our test cases. It's worth noting that the observed order reduction is not surprising since it was already explained in [56] that enforcing a local entropy inequality yields such behaviour. Our scheme is consistent with the method described in [56], as it also requires an extended stencil. We get $\alpha = \max(\alpha_\rho, \alpha_p, \alpha_\eta)$. This α is sup-mollified using a hat function with a radius of 2 cells.
4. The Dafermos hybrid scheme denoted by DALFT: α is determined through the technique described in (4). The value of α is taken as the maximum $\alpha = \max(\alpha_\rho, \alpha_p, \alpha_{\mathrm{Daf}})$.

Before comparing our schemes, we explain how we generate our training data for DDLFT.

Table 1 Used network structure with drop-out rate 0.2 during training

Layer	Input	2	3	4	5	Output
Activation	ELU	ELU	ELU	ELU	ELU	$x \to x$
Number of Neurons	40	80	80	80	80	1

Calculation of Training Data

As explained before, the training data for the NN was taken out of the simulation using an ENO scheme. Special care had to be taken to select initial data that leads to simulations where a representative amount of features of typical solutions to the Euler equations are visible. We decided therefore to use

$$u_0(x) = \begin{cases} u_1 & x \leq 2.5 \\ u_2 & x \leq 5 \\ u_3 & x \leq 7.5 \\ u_4 & x \leq 10.0 \end{cases}$$

as initial condition on the interval [0, 10] with periodic boundary conditions. The four constant states u_i between the three resulting Riemann problems were randomly selected as

$$\rho_i = A_\rho r_{i,1} + \varepsilon_\rho, \quad v_i = 2A_v \left(r_{i,2} - \frac{1}{2} \right), \quad p_i = A_p r_{i,3} + \varepsilon_p.$$

Here, $r_{i,j}$ denotes a random number in the interval [0, 1]. The constants A_ρ, A_v, A_p control the upper bounds of the selected random values and are chosen as $A_\rho = A_v = A_p = 4.0$. To ease the solution of these initial conditions the parameters $\varepsilon_\rho = \varepsilon_p = \frac{1}{100}$ result in strictly positive initial density and pressure. 100 of these initial conditions were solved up to $T = 2.5$ by an ENO scheme with 4000 points. The high-fidelity solution was subsequently sampled at 400-time slices within the interval [0, 2.5] and utilized as outlined in Sect. 4.2. This procedure, using 200 cells on the coarse grid, 3 conserved variables and 400 time slices results in roughly 244 MB training data. Please note that when used for training enough consecutive cells from this data pile are presented to the network, i.e. in total $8 \cdot 10^6$ samples are available for training. Finally note that even if we start with different Riemann problems our training data will contain as well purely smooth data at some time slices.

Layout and Training of the Network

We use a neural network built out of six layers whose dimensions are given in Table 1. In all, but the last layer, the ELU activation function is applied. The inputs are the values of the conserved variables and the pressure of five cells left and right to the cell boundary where α has to be determined. Our network for the prediction of α was trained using the ADAM optimizer [36] with parameters scheduled as given in Table 2. We use the Flux library in Julia to train our network [32, 33]. The resulting loss curve is printed in Fig. 3. The training took circa 20 minutes on 8 cores of an AMD Ryzen Threadripper 5900X at 3.7 GHz.

Table 2 Overview of used training parameters

Section	1	2	3	4	5	6	7
Epochs	25	25	25	25	25	25	25
Batchsize	32	256	1024	4096	4096	4096	4096
Stepsize	0.001	0.001	0.001	0.001	0.0001	0.00001	0.000001

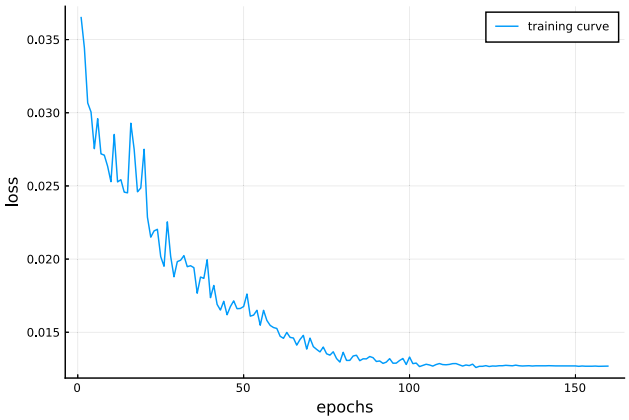

Fig. 3 Training loss of NN

6.1 Numerical Experiments

For the benchmark problems 6.1.2–6.1.4, we use always 100 cells on the interval $[0, 10]$. For the rest, we use either 400 or 800 cells. For simplicity, the CFL number is set to 0.5 for all test cases. If nothing is said about the boundary conditions, we use inflow–outflow conditions. The reference solutions are always calculated using ENO2 with 10000 cells. We use the code from [37] for the Dafermos scheme whereas the other schemes can be found in the corresponding repository.[3] We give only the numerical results for the density profiles for simplicity. The other profiles show similar behaviours.

6.1.1 Smooth Density Variation

To determine the experimental order of convergence, we simulate the smooth transport of a density variation under pressure equilibrium used in [37] up to $T = 1.5$. The initial condition is given by

$$\begin{aligned} \rho_0(x,0) &= 3.857153 + \varepsilon(x)\sin(2x), \quad v_0(x,0) = 2.0, \quad p_0(x,0) = 10.33333, \\ \varepsilon(x) &= \mathrm{e}^{(x-3)^2}. \end{aligned} \tag{21}$$

and periodic boundary conditions are considered. The analytical solution for this test problem is

$$\rho(x,t) = 3.857153 + \varepsilon(x-2t)\sin(2x-4t), \quad v(x,t) = 2.0, \quad p(x,t) = 10.33333.$$

[3] https://github.com/simonius/ddsolver.

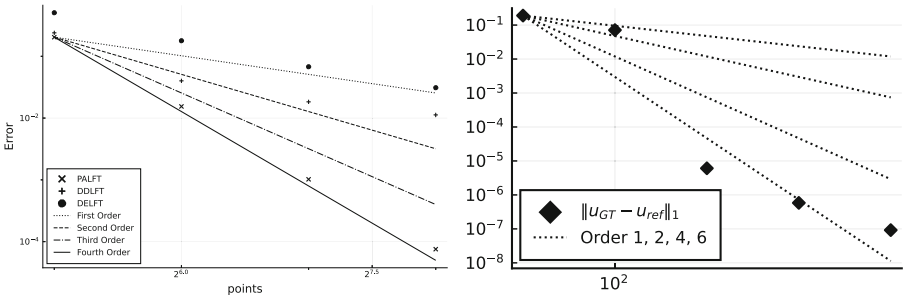

Fig. 4 Convergence plots for (21) (left schemes: DDLFT, PALFT, DELFT) (right: Dafermos scheme from [37])

To obtain the optimal order of accuracy, we use in this test case for the time integration the SSPRK(10,4) method (4th-order, strong-stability preserving Runge–Kutta methods with 10 stages) in the high-order flux. The L^1-errors of the schemes are shown in Fig. 4. The PALFT scheme converges with fourth-order accuracy in this specific test. Given the ability to construct entropy conservative fluxes up to any desired order and the capability to create polynomial annihilation operators that eliminate smooth solution components up to any desired order, it can be proven that PALFT schemes are also theoretically capable of achieving arbitrary high-order of accuracy. Nevertheless, this falls outside the scope of our investigation. We further see a slide decrease in order for the DDLFT for fine grids. This is due to the fact that the NN can not keep up with the DDLFT scheme itself on fine grids, i.e. for a smooth solution is $\alpha_{k+\frac{1}{2}} = \mathcal{O}\left((\Delta x)^3\right)$ not satisfied.[4] The entropy dissipative scheme converges with the second order of accuracy as expected. The convergence plot in Fig. 4 (right) for the Dafermos scheme seems a little bit surprising. However, the mollification process is not adapted in all of our test cases and we have a big jump in the accuracy when it is working more adequately. Then, we obtain also the fourth order of accuracy. For a more detailed description of the Dafermos entropy scheme, we refer to [37] where formally high-order of accuracy was analytically and numerically proven.

6.1.2 Sod's Shock Tube

The first benchmark is the SOD test problem [64, Problem I, Section 4.3.3]. It is a very mild test and its solution contains a left rarefaction, a contact discontinuity and a right shock. The initial conditions are given by

$$\rho_0(x,0) = \begin{cases} 1, \\ 0.125, \end{cases} \qquad v_0(x,0) = \begin{cases} 0, \\ 0, \end{cases} \qquad p_0(x,0) = \begin{cases} 1.0, & x < 5, \\ 0.1, & x \geq 5. \end{cases}$$

We run the simulation until $T = 2.0$ and the results for density can be seen in Fig. 5.

All schemes clearly produce correct predictions without unphysical shocks. Sadly oscillations are visible in the solution calculated by the data-driven scheme, see for example Fig. 5. These problems are nearly invisible in the PALFT scheme. A further surprising result is the ability of the DDLFT and PALFT to produce sharp transitions of shocks even without further

[4] This condition is necessary to ensure that the contribution of the low-order flux inside the convex combination (5) does not pollute the high-order accuracy of the entropy conservative method.

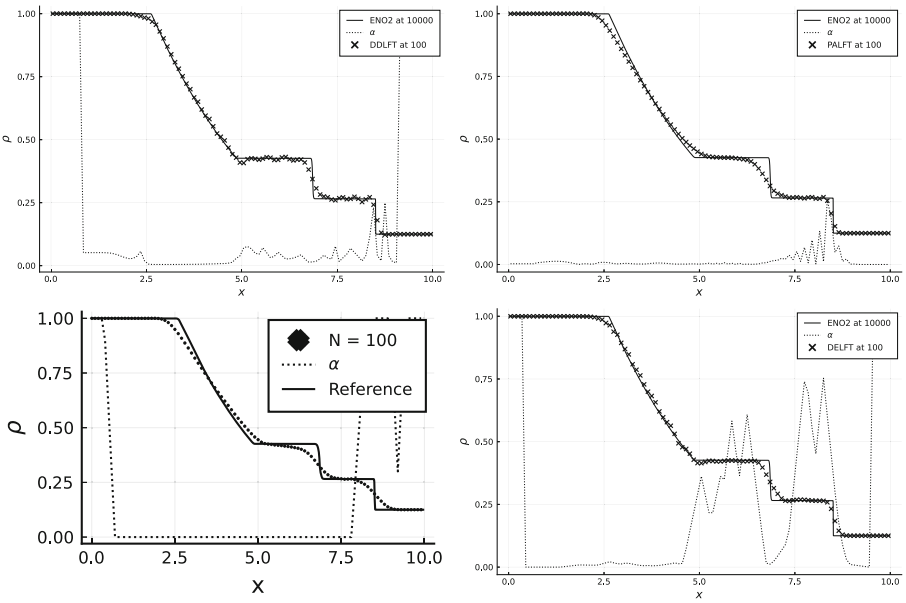

Fig. 5 Density for the first shock tube at $T = 2.0$ calculated using 100 cells. Data-driven scheme (up left), polynomial annihilation-based scheme (upright), Dafermos criterion scheme from [37](down left), discrete entropy stable scheme (down right)

tuning of parameters. The dotted lines in the figure give also the α coefficients in the convex combination of our blending schemes. Furthermore, we may cancel out the oscillation by additionally demanding the Dafermos criterion (4). Finally, we realize that the αs only distinguish essentially from zero around the shock for PALFT as the rest of the solution is smooth. For DDLFT, the lower-order method is also activated in smooth regions (i.e. $\alpha > 0$). These effects counter some of our intuition, as the transitions for the data-driven schemes are sharper than in the PALFT scheme. This could be some effect produced by the desire of the NN to produce the flux with minimum L^2 distance to the exact flux - and it is not clear that this is the least dissipative flux. Further, we like to point out that the DDLFT resolve the contact discontinuity at $x = 7$ best where the other schemes smear it. The DALFT scheme is performing here worst compare to the other. The DALFT scheme further smears all profiles. However, this is done already at the beginning of the calculation as can be seen in Fig. 5 at $T = 2$ $\alpha = 0$ except around the shock. The smearing comes from the beginning of the calculation where the low-order scheme is mostly used. Finally, the DELFT gives numerical approximations in between the DDLFT and PALFT without oscillations, cf. Fig. 5 (down right). Finally, in Fig. 6 the value of the blending parameter α during the simulation over time and cells is given for the different schemes. All schemes turn on the entropy dissipation on the right moving shock wave, while the contact discontinuity also triggers some dissipation in the DELFT scheme in time. It is important to emphasize that the PALFT scheme exhibits more smoothing of the contact discontinuity and rarefaction wave compared to other schemes, but it provides more accurate resolution of the shock. This behavior can be possible explained by examining Fig. 6, which shows that the low-order scheme is initially activated with greater diffusion due to a non-zero α value. This is also a possible explanation for these inaccuracies.

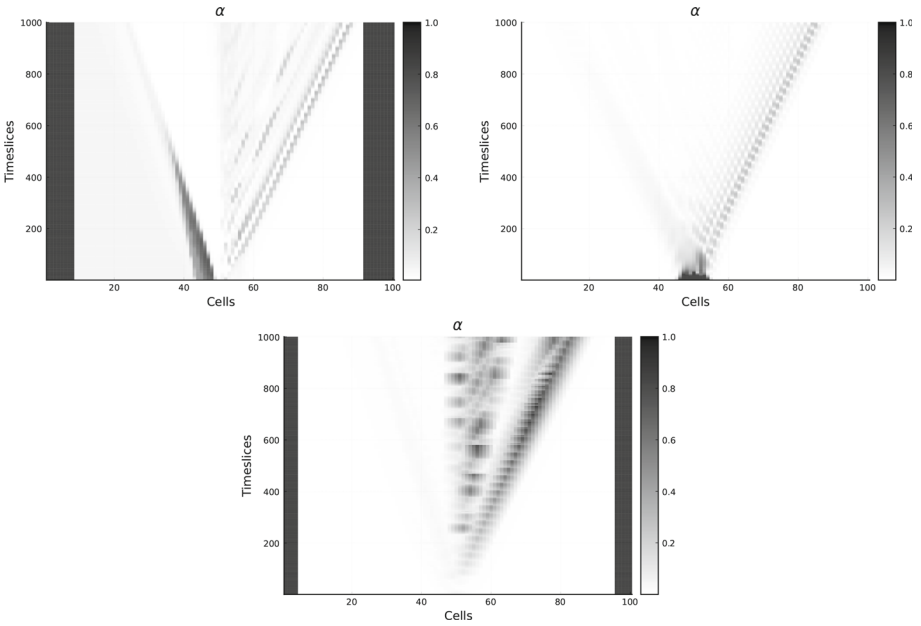

Fig. 6 Value of the blending parameter α during the simulation over time for the three schemes presented in this publication (Data-driven in the upper left, polynomial annihilation upper right and discretely entropy dissipative lower centre). All schemes turn on the entropy dissipation on the right moving shock wave, while the contact discontinuity also triggers some dissipation in the discrete entropy dissipative scheme. This is astonishing as a contact discontinuity does not dissipate entropy in the exact solution

Later only the shock is detected in the space-time scale. On the other side the entropy dissipative scheme starts with nearly every α value at zero but as longer the simulation progresses two cones with non-zero α values are developing and further transported: one for the shock and one for the contact discontinuities. It should be noted that our investigation of **Condition** F is sufficient but not necessary, indicating that we are over-activating the low-order scheme, which is also activated at local maxima at the contact discontinuity.

6.1.3 Second Shock Tube Problem

The second shock tube problem is given by the initial conditions

$$\rho_0(x,0) = \begin{cases} 0.445, \\ 0.5, \end{cases} \qquad v_0(x,0) = \begin{cases} 0.698, \\ 0, \end{cases} \qquad p_0(x,0) = \begin{cases} 3.528, & x < 5.0, \\ 0.571, & x \geq 5.0. \end{cases}$$

We run the simulations to $T = 1.3$ and the results for ρ are presented in Fig. 7. The second benchmark demonstrates similar behaviour as before. The DDLFT scheme performs quite well but small oscillations can be seen in Fig. 7 (left above picture) which and we assume that these could be cancelled out by further tuning the network and/or more (specific) training data. The PA performs as well good. The top plateau only displays a single oscillation point, and the contact discontinuity is less smeared than it is in the DALFT scheme, but it is still more smeared than in the DELFT scheme due to the early activation of $\alpha \neq 0$ (not shown here). The DELFT scheme provides results that lie between those of the DDLFT and PALFT schemes, and it yields good results. However, it is evident that the low-order part of the

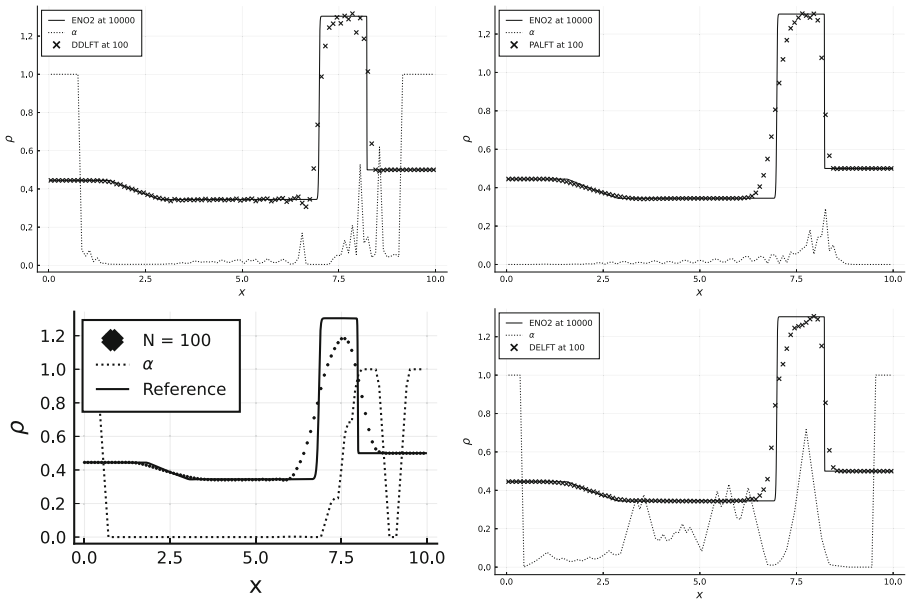

Fig. 7 Density of the second shock tube problem at $t = 1.3$

scheme is activated more frequently, even in smooth regions, when compared to the other schemes considered. This confirms that **Condition** F overestimates the α value.

6.1.4 123-Problem

Our next test case is the 123 problem [64, Problem II, Section 4.3.3]. The initial condition is given by

$$\rho_0(x,0) = \begin{cases} 1.0, \\ 1.0, \end{cases} \quad v_0(x,0) = \begin{cases} -2.0, \\ 2.0, \end{cases} \quad p_0(x,0) = \begin{cases} 0.4, & x < 5.0, \\ 0.4, & x \geq 5.0. \end{cases}$$

The solution contains two strong rarefactions and a trivial stationary contact discontinuity; the pressure p is very small (close to vacuum) and this can lead to difficulties. The results are shown in Fig. 8. These results are the only instance where our positivity preserving lower bounds on α has been activated but only for the DDLFT. It was different from zero. It should be pointed out that for the DD scheme, the approach using the positivity limiters inside α is necessary. Without them, we would obtain unphysical negative pressure and density. This underlines the ability of the data-driven scheme to combine bounds on α for positivity derived by hand and the educated guess of an optimal α by an FNN. The results look promising and as before the DDLFT is more accurate than the PALFT scheme. We reach nearly zero in the DDLFT scheme. It should be pointed out that α, in this case, is also not symmetric around $x = 5$. The rest of the schemes do not need this additional requirement of the limiters. We further recognize that the PALFT scheme is much too dissipative where the DALFT scheme is in between the DALFT and PALFT schemes. Again the reason is the starting point of the simulations as before. The DELFT scheme performs again quite well.

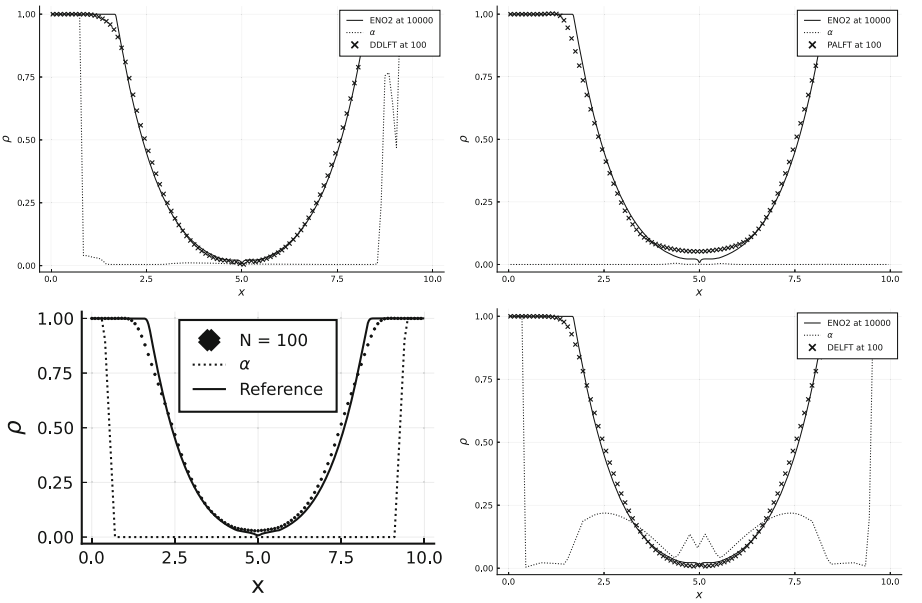

Fig. 8 Density and pressure for the 123 problem $t = 1.2$ calculated using 100 cells

6.1.5 Woodward–Colella Blast Wave

As a more complex test case the Woodward Colella blast wave problem is considered as proposed in [65]. The solution contains a collision of two shock waves. We have reflecting wall boundary conditions. These boundary conditions are implemented in our scheme using ghost cells u_0 and u_{N+1} placed outside of the domain, with the following values

$$u_0 = \begin{pmatrix} \rho_{lg} \\ \rho_{lg} v_{lg} \\ E_{lg} \end{pmatrix}, \quad , u_1 = \begin{pmatrix} \rho_l \\ \rho_l v_l \\ E_l \end{pmatrix}, \quad u_N = \begin{pmatrix} \rho_r \\ \rho_r v_r \\ E_r \end{pmatrix}, \qquad u_{N+1} = \begin{pmatrix} \rho_{rg} \\ \rho_{rg} v_{rg} \\ E_{rg} \end{pmatrix}$$

A solid boundary is implemented now by setting

$$\rho_{lg} = \rho_l, \quad v_{lg} = -v_l, \quad p_{lg} = p_l$$

for the left and

$$\rho_{rg} = \rho_r, \quad v_{rg} = -v_r, \quad p_{rg} = p_r$$

for the right boundary. For high-order schemes with wider stencils more ghost cells are added via symmetry. The initial data is given by the following three initial states

$$\rho_0(x,0) = \begin{cases} 1.0 \\ 1.0 \\ 1.0 \end{cases} \quad v_0(x,0) = \begin{cases} 0.0 \\ 0.0 \\ 0.0 \end{cases} \quad p_0(x,0) = \begin{cases} 10^3 & x \le 1.0 \\ 10^{-2} & x \le 9.0 \\ 10^2 & x < 10 \end{cases}.$$

This test case is significantly more demanding than the test cases before, as the interaction of two shocks, one moving from the left to the right, and one moving from the right to the

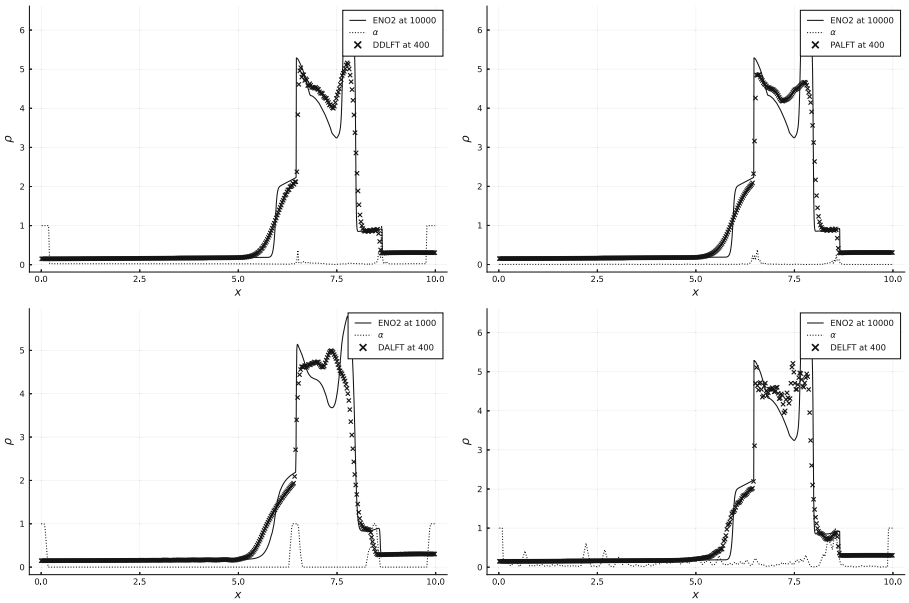

Fig. 9 Density profiles for the Woodward–Colella blast wave test case at $t = 0.38$

left part of the domain, has to be calculated. Therefore, we increase the number of cells and use $N = 400$. Again, the different density profiles can be seen in Fig. 9.

We like to point out that both the DDLFT and PALFT schemes give good results (above row) whereas the numerical solution using the DELFT scheme contains some oscillations in between the shock. The DALFT scheme is too dissipative to catch both shock phases. Here, it is also surprising that the solution of DDLFT scheme does not show any oscillation different from the shock tube test cases.

6.1.6 Shu–Osher

The initial conditions of the Shu–Osther test are given by

$$\rho_0(x,0) = \begin{cases} 3.857153 \\ 1 + \varepsilon \sin(5x) \end{cases} \qquad v_0(x,0) = \begin{cases} 2.629 \\ 0 \end{cases} \qquad p_0(x,0) = \begin{cases} 10.333 & x < 1 \\ 1 & x \geq 1 \end{cases}$$

in the domain $\Omega = [0, 10]$. The parameter ε was set to the canonical value of 0.2 and the adiabatic exponent was set to $\gamma = \frac{7}{5}$ for an ideal gas. The density profiles are printed in Fig. 10 for different amounts of cells $N = 400, 800$. All numerical solutions are describing the reference solution. All schemes are able to resolve the strong shocks without nonphysical oscillations. Oscillations also do not appear in the wake of the shock. The amount of points needed for the transition is small and the wave structure trailing the shock is resolved accurately. Further, we recognize the best convergence inside the different schemes for the PALFT scheme, where increasing the number of points in the DDLFT scheme has less influence on the resolution. The same can be seen for the DELFT scheme. The approximated solution of the DALFT scheme behaves nicely since the shock sensor is optimized for this test problem as described in [37].

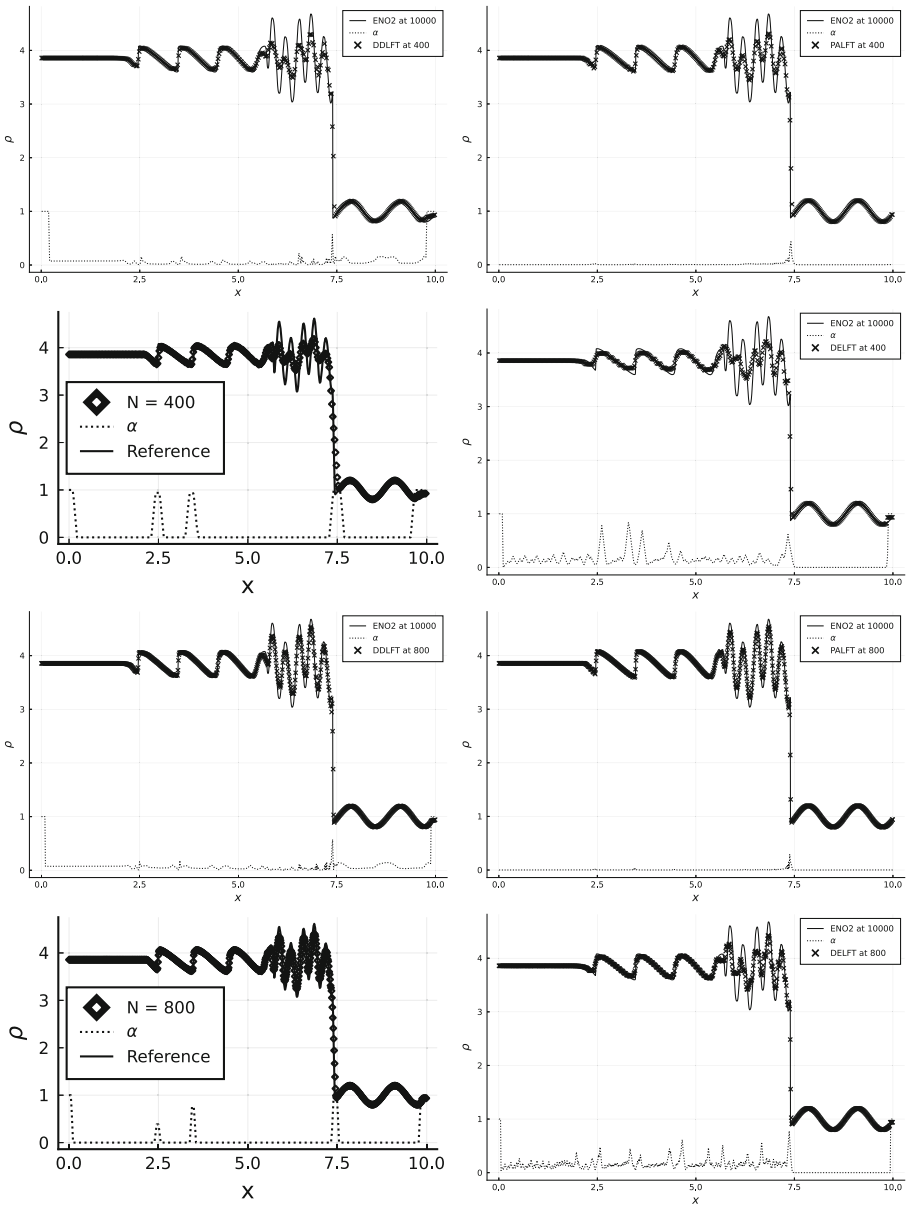

Fig. 10 Density profiles for the Shu–Osher test case number 8 at $t = 1.8$

Remark 3 (Computational costs (CPU)) In our simulations, we have recognized that the data-driven scheme is much more expensive in computational costs compared to the other three schemes which have roughly the same costs. However, we like to stress that our implementations are not optimized and a detailed efficiency analysis would also be desirable. In such an investigation, one should also consider that the data-driven scheme needs less memory to obtain the same resolution quality as the other schemes. Therefore, the data-driven scheme

is particularly suitable for GPU calculations. In an efficiency analysis, this has to be taken as well into account.

7 Summary

In this work, we compared different ways to increase the resolution of high-order finite volume/finite difference schemes for hyperbolic conservation laws, in particular if discontinuities appear. After giving an introduction and an overview over the underlying numerical flux based on a convex combination, some physical constraints were concerned. To be more specific, we gave conditions that assured the cell entropy inequality and/or the positivity of pressure and density of the numerical solution of the Euler equations. A second possibility was further constructed using a feedforward neural network. Here, the network was trained by data which were calculated by a reference scheme. We provided afterwards a choice of the convex parameter based on polynomial annihilation operators after giving a brief introduction to their basic framework. In a last step, the resulting schemes were tested and compared by numerical experiments on the Euler equations. Here, we consider several well-known benchmark problems. All schemes are combined with the ansatz to keep density and pressure positive. This was especially important for the data-driven scheme since it would violate this condition and the algorithm would break down (123-problem). Besides this fact, we further could conclude that the DDLFT scheme shows promising results in all numerical experiments. We obtain good approximations especially for the blast-wave test case but also for the shock-tube problems. The PALFT scheme demonstrates good results as well for most of the cases and seems best on most of them. However, for the 123-problem it was to much dissipative compared to the other schemes. The cell-entropy scheme had the disadvantage that only second order of accuracy could be reached for smooth problems if not additional techniques are applied. For instance, an additional shock detector can be used as a preliminary step. This technique was applied for example in [46] in the DG framework together with convex limiting.[5] Besides this fact, it demonstrates quite well and yield oscillation free numerical approximations except for the blast wave. The DALFT was quite dissipative in most of the cases as the α is selected to calculate the most dissipative approximate solution imaginable. However, by selection the mollification process more suitable like in the Shu–Osher test case, one obtain as well promising numerical solutions. As a conclusion, all techniques can be used and the resulting FD/FV schemes are capable to handle strong shocks and are often oscillation free. The approach can be extended straightforwardly to two-dimensional (or multi-dimensional) problems using a tensor structure strategy in FD (or structured quad grids in FV). However, when focusing on unstructured grids, additional techniques must be developed. This includes the selection of stencils, presentation of data, and their utilization, which are all essential. In our future work, we plan to continue our investigation in this direction. Further, it should be noted that to handle complex geometries with a tensor grid, we can also adopt the approach described in [17]. Extensions to multiphase flows are as well planned. Finally, our high-order FD/FV blending schemes can be also the

[5] We refer also to the work [38] for a possible explanation about the decrease of accuracy using the convex limiting strategy with enforcing entropy stability.

starting point of a convergence analysis for the Euler equations via dissipative weak solutions [3, 21, 43] which is already work in progress.

Fundings Open Access funding enabled and organized by Projekt DEAL. This work was partially supported by the German Science Foundation (DFG) under Grant SO 363/15-1 (Hillebrand), Grant SO 363/14-1 (Klein) and the Gutenberg Research College, JGU Mainz (Öffner).

Data Availability Parts of the datasets generated during and/or analysed during the current study are available in the repository https://github.com/simonius/ddsolver. Further parts are not public since they will be extended but are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

For the fully-discrete entropy correction scheme, the high-order time integration is based on a reinterpretation of predictor-corrector time integration [34, p. 386] as a numerical quadrature of the numerical flux over a cell boundary. We can proof the following theorem:

Theorem 3 (Predictor–Corrector–Fluxes) *Let* $f^{\text{num}}(\mathbf{u}_k, \mathbf{u}_{k+1})$ *be a numerical flux and* $\mathbf{u}_k(t)$ *on* $[t, t + \Delta t]$ *be the exact solution of the scheme*

$$\frac{\mathrm{d}\mathbf{u}_k(t)}{\mathrm{d}t} + \frac{f^{\text{num}}\left(\mathbf{u}_k(t), \mathbf{u}_{k+1}(t)\right) - f^{\text{num}}\left(\mathbf{u}_{k-1}(t), \mathbf{u}_k(t)\right)}{\Delta x} = 0$$

with uniform cell size Δx*. Then, the 4-point numerical flux* $f^{\text{num}}(\mathbf{u}_{k-1}, \mathbf{u}_k, \mathbf{u}_{k+1}, \mathbf{u}_{k+2})$ *defined as*

$$\begin{aligned}
\mathbf{u}_k^1 &= \mathbf{u}_k + \lambda\left(f^{\text{num}}(\mathbf{u}_{k-1}, \mathbf{u}_k) - f^{\text{num}}(\mathbf{u}_k, \mathbf{u}_{k+1})\right), \\
\mathbf{u}_{k+1}^1 &= \mathbf{u}_{k+1} + \lambda\left(f^{\text{num}}(\mathbf{u}_k, \mathbf{u}_{k+1}) - f^{\text{num}}(\mathbf{u}_{k+1}, \mathbf{u}_{k+1})\right), \\
f^{\text{num}}\left(\mathbf{u}_{k-1}, \mathbf{u}_k, \mathbf{u}_{k+1}, \mathbf{u}_{k+2}\right) &= \frac{f^{\text{num}}(\mathbf{u}_k, \mathbf{u}_{k+1}) + f^{\text{num}}\left(\mathbf{u}_k^1, \mathbf{u}_{k+1}^1\right)}{2}
\end{aligned}$$

is a second-order[6] *accurate approximation of* $\frac{1}{\Delta t}\int_t^{t+\Delta t} f^{\text{num}}(\mathbf{u}_k(\tau), \mathbf{u}_{k+1}(\tau))\mathrm{d}\tau$*, i.e.*

$$\left\| f^{\text{num}}\left(\mathbf{u}_{k-1}, \mathbf{u}_k, \mathbf{u}_{k+1}, \mathbf{u}_{k+2}\right) - \frac{1}{\Delta t}\int_t^{t+\Delta t} f^{\text{num}}\left(\mathbf{u}_k(\tau), \mathbf{u}_{k+1}(\tau)\right)\mathrm{d}\tau \right\| = \mathcal{O}(\Delta t)^2.$$

Proof We begin by stating that the intermediate values $\mathbf{u}_k^1, \mathbf{u}_{k+1}^1$ are first-order accurate, i.e.

$$\mathbf{u}_k^1 = \mathbf{u}_k(t + \Delta t) + \mathcal{O}\left((\Delta t)^2\right) \quad \mathbf{u}_{k+1}^1 = u_{k+1}(t + \Delta t) + \mathcal{O}\left((\Delta t)^2\right)$$

[6] Please note that the term p order accurate was coined so that integration via a p order quadrature rule leads to a p order accurate approximation.

due to the explicit Euler method. Calculation of the flux between cell $\mathbf{u}_k$ and $\mathbf{u}_{k+1}$ over time Δt via the trapezoid rule I (second-order) and the exact solution $\mathbf{u}_k(t)$ is second-order accurate, i.e.

$$\begin{aligned}\mathrm{I}\left[f^{\mathrm{num}}(\mathbf{u}_k(\cdot),u_{k+1}(\cdot))\right] &= \frac{\Delta t}{2}\left(f^{\mathrm{num}}\left(\mathbf{u}_k(t),\mathbf{u}_{k+1}(t)\right)+f^{\mathrm{num}}\left(\mathbf{u}_k(t+\Delta t),\mathbf{u}_{k+1}(t+\Delta t)\right)\right)\\ &= \int_t^{t+\Delta t} f^{\mathrm{num}}\left(u_k(\tau),u_{k+1}(\tau)\right)\mathrm{d}\tau+\mathcal{O}\left((\Delta t)^3\right).\end{aligned}$$

Due to the Lipschitz continuity of f^{num}, we have

$$\begin{aligned}&\left\| f^{\mathrm{num}}\left(\mathbf{u}_l(t+\Delta t),\mathbf{u}_r(t+\Delta t)\right)-f^{\mathrm{num}}\left(\mathbf{u}_l^1,\mathbf{u}_r^1\right)\right\|\\ &\le L_f\left(\left\|\mathbf{u}_l(t+\Delta t)-\mathbf{u}_l^1\right\|+\left\|\mathbf{u}_r(t+\Delta t)-\mathbf{u}_r^1\right\|\right),\end{aligned}$$

where $\mathbf{u}_l$ and $\mathbf{u}_r$ denote the left and right value at some generic interface. Due to the accuracy order of $\mathbf{u}_k^1$ and $\mathbf{u}_{k+1}^1$, it follows

$$\left\| f^{\mathrm{num}}\left(\mathbf{u}_k(t+\Delta t),\mathbf{u}_{k+1}(t+\Delta t)\right)-f^{\mathrm{num}}\left(\mathbf{u}_k^1,\mathbf{u}_{k+1}^1\right)\right\|=\mathcal{O}\left(\Delta t^2\right).$$

The combination of these three statements yields that the numerical quadrature of the flux calculated using the approximate values $\mathbf{u}_k^1,\mathbf{u}_{k+1}^1$

$$\begin{aligned}\Delta t f^{\mathrm{num}} &= \frac{\Delta t}{2}\left(f^{\mathrm{num}}\left(\mathbf{u}_k,\mathbf{u}_{k+1}\right)+f^{\mathrm{num}}\left(\mathbf{u}_k^1,\mathbf{u}_{k+1}^1\right)\right)\\ &= \frac{\Delta t}{2}\left(f^{\mathrm{num}}\left(\mathbf{u}_k,\mathbf{u}_{k+1}\right)+f^{\mathrm{num}}\left(\mathbf{u}_k(t+\Delta t),\mathbf{u}_{k+1}(t+\Delta t)\right)+\mathcal{O}(\Delta t)^2\right)\\ &= \mathrm{I}\left[f^{\mathrm{num}}(\mathbf{u}_k(\cdot),\mathbf{u}_{k+1}(\cdot))\right]+\mathcal{O}(\Delta t)^3\\ &= \int_t^{t+\Delta t} f^{\mathrm{num}}\left(\mathbf{u}_k(\tau),\mathbf{u}_{k+1}(\tau)\right)\mathrm{d}\tau+\mathcal{O}\left(\Delta t^3\right)\end{aligned}$$

is a second-order exact approximation and dividing by Δt induces the result. □

The above numerical flux $f^{\mathrm{num}}\left(\mathbf{u}_{k-1},\mathbf{u}_k,\mathbf{u}_{k+1},\mathbf{u}_{k+2}\right)$ could be also interpreted as the flux over the given cell boundary if the semidiscrete scheme is used together with the strong stability preserving (SSP) RK(2,2) method which is equivalent to the deferred correction method of order 2 [2]. However, higher-order quadrature rules can also be applied in this context.

References

1. Abgrall, R.: A general framework to construct schemes satisfying additional conservation relations. Application to entropy conservative and entropy dissipative schemes. J. Comput. Phys. **372**, 640–666 (2018)
2. Abgrall, R., Le Mélédo, É., Öffner, P., Torlo, D.: Relaxation deferred correction methods and their applications to residual distribution schemes. SMAI J. Comput. Math. **8**, 125–160 (2022). https://doi.org/10.5802/smai-jcm.82
3. Abgrall, R., Lukácova-Medvid'ová, M., Öffner, P.: On the convergence of residual distribution schemes for the compressible Euler equations via dissipative weak solutions. M3AS: Mathematical Models and Methods in Applied Sciences (2023)
4. Abgrall, R., Nordström, J., Öffner, P., Tokareva, S.: Analysis of the SBP-SAT stabilization for finite element methods part II: entropy stability. Commun. Appl. Math. Comput. 1–23 (2021)
5. Abgrall, R., Öffner, P., Ranocha, H.: Reinterpretation and extension of entropy correction terms for residual distribution and discontinuous Galerkin schemes: application to structure preserving discretization. J. Comput. Phys. **453**, 24 (2022). https://doi.org/10.1016/j.jcp.2022.110955.Id/No110955

6. Abgrall, R., Shu, C.W.: Handbook of Numerical Methods for Hyperbolic Problems: Applied and Modern Issues, vol. 18. Elsevier, Amsterdam (2017)
7. Abgrall, R., Veiga, M.H.: Neural network-based limiter with transfer learning. Commun. Appl. Math. Comput. 1–41 (2020)
8. Archibald, R., Gelb, A., Yoon, J.: Polynomial fitting for edge detection in irregularly sampled signals and images. SIAM J. Numer. Anal. **43**(1), 259–279 (2005)
9. Bacigaluppi, P., Abgrall, R., Tokareva, S.: "A posteriori" limited high order and robust schemes for transient simulations of fluid flows in gas dynamics. J. Comput. Phys. **476**, 34 (2023). https://doi.org/10.1016/j.jcp.2022.111898.Id/No11189
10. Beck, A.D., Zeifang, J., Schwarz, A., Flad, D.G.: A neural network based shock detection and localization approach for discontinuous Galerkin methods. J. Comput. Phys. **423**, 109824 (2020)
11. Chan, J.: On discretely entropy conservative and entropy stable discontinuous Galerkin methods. J. Comput. Phys. **362**, 346–374 (2018)
12. Chen, T., Shu, C.W.: Review of entropy stable discontinuous Galerkin methods for systems of conservation laws on unstructured simplex meshes. CSIAM Trans. Appl. Math. **1**, 1–52 (2020)
13. Clain, S., Diot, S., Loubère, R.: A high-order finite volume method for systems of conservation laws-multi-dimensional optimal order detection (MOOD). J. Comput. Phys. **230**(10), 4028–4050 (2011). https://doi.org/10.1016/j.jcp.2011.02.026
14. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289 (2015)
15. Cybenko, G.: Approximation by superpositions of a sigmoidal function. Math. Control Signals Systems **2**, 303–314 (1989)
16. Dafermos, C.M.: The entropy rate admissibility criterion for solutions of hyperbolic conservation laws. J. Differ. Equ. **14**(2), 202–212 (1973)
17. DeZeeuw, D., Powell, K.G.: An adaptively refined cartesian mesh solver for the Euler equations. J. Comput. Phys. **104**(1), 56–68 (1993). https://doi.org/10.1006/jcph.1993.1007
18. Discacciati, N., Hesthaven, J.S., Ray, D.: Controlling oscillations in high-order discontinuous Galerkin schemes using artificial viscosity tuned by neural networks. J. Comput. Phys. **409**, 109304 (2020)
19. Du, Q., Glowinski, R., Hintermüller, M., Suli, E.: Handbook of Numerical Methods for Hyperbolic Problems: Basic and Fundamental Issues. Elsevier, Amsterdam (2016)
20. Dubey, S.R., Singh, S.K., Chaudhuri, B.B.: Activation functions in deep learning: A comprehensive survey and benchmark. Neurocomputing (2022)
21. Feireisl, E., Lukáčová-Medvid'ová, M., Mizerová, H.: Convergence of finite volume schemes for the Euler equations via dissipative measure-valued solutions. Found. Comput. Math. **20**(4), 923–966 (2020). https://doi.org/10.1007/s10208-019-09433-z
22. Fisher, T.C., Carpenter, M.H., Nordström, J., Yamaleev, N.K., Swanson, C.: Discretely conservative finite-difference formulations for nonlinear conservation laws in split form: theory and boundary conditions. J. Comput. Phys. **234**, 353–375 (2013)
23. Fjordholm, U.S., Mishra, S., Tadmor, E.: Arbitrarily high-order accurate entropy stable essentially nonoscillatory schemes for systems of conservation laws. SIAM J. Numer. Anal. **50**(2), 544–573 (2012). https://doi.org/10.1137/110836961
24. Gassner, G.J., Winters, A.R., Kopriva, D.A.: Split form nodal discontinuous Galerkin schemes with summation-by-parts property for the compressible Euler equations. J. Comput. Phys. **327**, 39–66 (2016). https://doi.org/10.1016/j.jcp.2016.09.013
25. Glaubitz, J., Gelb, A.: High order edge sensors with l^1 regularization for enhanced discontinuous Galerkin methods. SIAM J. Sci. Comput. **41**(2), A1304–A1330 (2019)
26. Guermond, J.L., Popov, B., Tomas, I.: Invariant domain preserving discretization-independent schemes and convex limiting for hyperbolic systems. Comput. Methods Appl. Mech. Eng. **347**, 143–175 (2019)
27. Harten, A.: On the symmetric form of systems of conservation laws with entropy. J. Comput. Phys. **49**, 151–164 (1983)
28. Harten, A., Enquist, B., Osher, S., Chakravarthy, S.R.: Uniformly high order accurate essentially non-oscillatory schemes III. J. Comput. Phys. **71**, 231–303 (1987)
29. Harten, A., Lax, P.D., van Leer, B.: On upstream differencing and Godunov type schemes for hyperbolic conservation laws. SIAM Rev. **25**, 35–61 (1983)
30. Harten, A., Zwas, G.: Self-adjusting hybrid schemes for shock computations. J. Comput. Phys. **9**, 568–583 (1972). https://doi.org/10.1016/0021-9991(72)90012-5
31. Hennemann, S., Rueda-Ramírez, A.M., Hindenlang, F.J., Gassner, G.J.: A provably entropy stable subcell shock capturing approach for high order split form dg for the compressible Euler equations. J. Comput. Phys. **426**, 109935 (2021)

32. Innes, M.: Flux: elegant machine learning with Julia. J. Open Sour. Softw. (2018). https://doi.org/10.21105/joss.00602
33. Innes, M., Saba, E., Fischer, K., Gandhi, D., Rudilosso, M.C., Joy, N.M., Karmali, T., Pal, A., Shah, V.: Fashionable modelling with flux. CoRR arXiv:1811.01457 (2018)
34. Isaacson, E., Keller, H.B.: Analysis of Numerical Methods. Wiley, New York (1966)
35. Ismail, F., Roe, P.L.: Affordable, entropy-consistent flux functions II: entropy production at shocks. J. Comput. Phys. **228**, 5410–5436 (2009)
36. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2017)
37. Klein, S.C.: Using the Dafermos entropy rate criterion in numerical schemes. BIT Numer. Math. **62**, 1673–1701 (2022)
38. Kuzmin, D.: Algebraic Flux Correction I, pp. 145–192. Springer, Dordrecht (2012)
39. Kuzmin, D.: Monolithic convex limiting for continuous finite element discretizations of hyperbolic conservation laws. Comput. Methods Appl. Mech. Eng. **361**, 112804 (2020)
40. Kuzmin, D., Hajduk, H., Rupp, A.: Limiter-based entropy stabilization of semi-discrete and fully discrete schemes for nonlinear hyperbolic problems. Comput. Methods Appl. Mech. Eng. **389**, 28 (2022). https://doi.org/10.1016/j.cma.2021.114428.Id/No114428
41. Lax, P.D.: Shock waves and entropy. Contrib. Nonlinear Funct. Anal. 603–634 (1971)
42. LeFloch, P.G., Mercier, J.M., Rohde, C.: Fully discrete, entropy conservative schemes of arbitrary order. SIAM J. Numer. Anal. **40**(5), 1968–1992 (2002). https://doi.org/10.1137/S003614290240069X
43. Lukáčová-Medvid'ová, M., Öffner, P.: Convergence of discontinuous Galerkin schemes for the Euler equations via dissipative weak solutions. Appl. Math. Comput. **436**, 22 (2023). https://doi.org/10.1016/j.amc.2022.127508.Id/No127508
44. Öffner, P.: Zweidimensionale klassische und diskrete orthogonale polynome und ihre anwendung auf spektrale methoden zur lösung von hyperbolischen erhaltungsgleichungen. Ph.D. thesis (2015)
45. Öffner, P., Glaubitz, J., Ranocha, H.: Stability of correction procedure via reconstruction with summation-by-parts operators for Burgers' equation using a polynomial chaos approach. ESAIM Math. Model. Numer. Anal. **52**(6), 2215–2245 (2018). https://doi.org/10.1051/m2an/2018072
46. Pazner, W.: Sparse invariant domain preserving discontinuous Galerkin methods with subcell convex limiting. Comput. Methods Appl. Mech. Eng. **382**, 28 (2021). https://doi.org/10.1016/j.cma.2021.113876.Id/No113876
47. Persson, P.O., Peraire, J.: Sub-cell shock capturing for discontinuous Galerkin methods. In: 44th AIAA Aerospace Sciences Meeting and Exhibit, p. 112 (2006)
48. Perthame, B., Shu, C.W.: On positivity preserving finite volume schemes for Euler equations. Numer. Math. **73**(1), 119–130 (1996). https://doi.org/10.1007/s002110050187
49. Pinkus, A.: Approximation theory of the MLP model in neural networks. In: Acta Numerica, vol. 8, pp. 143–195. Cambridge University Press, Cambridge (1999)
50. Ranocha, H.: Comparison of some entropy conservative numerical fluxes for the Euler equations. J. Sci. Comput. **76**(1), 216–242 (2018)
51. Ranocha, H., Öffner, P., Sonar, T.: Summation-by-parts operators for correction procedure via reconstruction. J. Comput. Phys. **311**, 299–328 (2016). https://doi.org/10.1016/j.jcp.2016.02.009
52. Ranocha, H., Sayyari, M., Dalcin, L., Parsani, M., Ketcheson, D.I.: Relaxation Runge–Kutta methods: fully discrete explicit entropy-stable schemes for the compressible Euler and Navier-Stokes equations. SIAM J. Sci. Comput. **42**(2), A612–A638 (2020). https://doi.org/10.1137/19M1263480
53. Richtmyer, R.D., Morton, K.W.: Difference Methods for Initial-Value Problems. Malabar (1994)
54. Roe, P.L.: Approximate Riemann solvers, parameter vectors and difference schemes. J. Comput. Phys. **43**, 357–372 (1981)
55. Rueda-Ramírez, A.M., Pazner, W., Gassner, G.J.: Subcell limiting strategies for discontinuous Galerkin spectral element methods. arXiv preprint arXiv:2202.00576 (2022)
56. Schonbek, M.E.: Second-order conservative schemes and the entropy condition. Math. Comput. **44**, 31–38 (1985). https://doi.org/10.2307/2007790
57. Shi, C., Shu, C.W.: On local conservation of numerical methods for conservation laws. Comput. Fluids **169**, 3–9 (2018)
58. Shu, C.W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys. **77**, 439–471 (1988)
59. Shu, C.W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing Schemesii. J. Comput. Phys. **83**, 439–471 (1989)
60. Sonntag, M., Munz, C.D.: Shock capturing for discontinuous Galerkin methods using finite volume subcells. In: Finite Volumes for Complex Applications VII-Elliptic, Parabolic and Hyperbolic Problems, pp. 945–953. Springer (2014)

61. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
62. Tadmor, E.: Numerical viscsity and the entropy condition for conservative difference schemes. Math. Comput. **43**, 369–381 (1984)
63. Tadmor, E.: The numerical viscosity of entropy stable schemes for systems of conservation laws. I. Math. Comput. **49**(179), 91–103 (1987)
64. Toro, E.F.: Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction. Springer, Berlin (2009). https://doi.org/10.1007/b79761
65. Woodward, P., Colella, P.: The numerical simulation of two-dimensional fluid flow with strong shocks. J. Comput. Phys. **54**, 115–173 (1984). https://doi.org/10.1016/0021-9991(84)90142-6
66. Zeifang, J., Beck, A.: A data-driven high order sub-cell artificial viscosity for the discontinuous Galerkin spectral element method. J. Comput. Phys. 110475 (2021)
67. Zhang, X., Shu, C.W.: Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments. Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. **467**(2134), 2752–2776 (2011). https://doi.org/10.1098/rspa.2011.0153