Journal of Scientific Computing (2021) 86:3
https://doi.org/10.1007/s10915-020-01381-z

The Active Flux Scheme for Nonlinear Problems

Wasilij Barsukow[1]

Received: 31 October 2019 / Revised: 28 September 2020 / Accepted: 2 December 2020 /
Published online: 21 December 2020

Abstract

The Active Flux scheme is a finite volume scheme with additional point values distributed along the cell boundary. It is third order accurate and does not require a Riemann solver. Instead, given a reconstruction, the initial value problem at the location of the point value is solved. The intercell flux is then obtained from the evolved values along the cell boundary by quadrature. Whereas for linear problems an exact evolution operator is available, for nonlinear problems one needs to resort to approximate evolution operators. This paper presents such approximate operators for nonlinear hyperbolic systems in one dimension and nonlinear scalar equations in multiple spatial dimensions. They are obtained by estimating the wave speeds to sufficient order of accuracy. Additionally, an entropy fix is introduced and a new limiting strategy is proposed. The abilities of the scheme are assessed on a variety of smooth and discontinuous setups.

Keywords Finite volume methods · Active flux · Hyperbolic conservation laws · Limiter

Mathematics Subject Classification 35L65 · 35L45 · 65M08 · 65M25

1 Introduction

Hyperbolic $m \times m$ systems of conservation laws in d spatial dimensions have the form

$$\partial_t q + \nabla \cdot \mathbf{f}(q) = 0 \quad q : \mathbb{R}^+_0 \times \mathbb{R}^d \to \mathbb{R}^m \tag{1.1}$$

The function $\mathbf{f}$ is called the flux. Exact solutions of these equations are unavailable in general, and one needs to resort to numerical methods.

Cell based methods consider the computational domain to be partitioned into cells. A certain number of discrete degrees of freedom are associated with every cell: e.g. finite

The author was supported by the German Academic Exchange Service (DAAD) with funds from the German Federal Ministry of Education and Research (BMBF) and the European Union (FP7-PEOPLE-2013-COFUND – Grant agreement No. 605728), as well as by the Deutsche Forschungsgemeinschaft (DFG) through Project 429491391.

✉ Wasilij Barsukow
wasilij.barsukow@math.uzh.ch

[1] Institute of Mathematics, Zurich University, 8057 Zurich, Switzerland

volume methods store the average of the dependent variable and spectral/Galerkin methods store coefficients of a decomposition in some basis.

In order to evolve the cell average, *finite volume methods* require the knowledge of the flux through the intercell boundary (see Sect. 2.1 for a derivation). This flux cannot generally be approximated by a symmetric average of fluxes associated to the values in the two adjacent cells, because this results in an unstable method. Instead, the fact that hyperbolic PDEs have certain preferred directions of information propagation needs to be reflected in the numerical method. The choice of the numerical flux as an asymmetric average of the neighbouring values is referred to as *upwinding*. It has been suggested in [14] to use an exact short-time solution as a building block in order to find a numerical flux that leads to a stable scheme. First, a piecewise polynomial function is found, such that it is continuous in every cell and its average agrees with the given average. The discontinuities at cell interfaces present so-called Riemann Problems, which then are solved over a time interval that does not allow them to interact. The exact flux at the location of the cell interface then is used as a numerical flux in the finite volume method. To save computation time, an approximate solution of the Riemann Problem can be used, see e.g. [17,18,24] as well as [20,29] for more details. Higher order of accuracy is achieved by widening the stencil [7,30,31].

Galerkin methods represent the numerical solution in e.g. a polynomial basis. Every basis coefficient is then evolved using the weak formulation of (1.1). Again, for hyperbolic equations this requires modification in order to achieve a stable method, one of which is the *discontinuous Galerkin method* ([6], but see also [2]). The basis functions are piecewise polynomial, and in order to deal with the jumps across cell interfaces a Riemann solver is invoked. Higher order of accuracy is achieved by retaining more coefficients of the basis decomposition.

Even in one spatial dimension, conservation laws (1.1) therefore pose a number of challenges to numerical methods. This does not only include the necessity of upwinding. It is also known that continuous solutions do not generally exist for all times, and thus numerical methods need to be designed in such a way that they can capture discontinuities (weak solutions). Weak solutions in one spatial dimension only become unique upon additional conditions (entropy conditions), and numerical methods need to fulfill a discrete counterpart of these conditions (entropy stability, see e.g. [28]). These aspects have been subject of numerous investigations, see e.g. [20] for an introduction.

In multiple spatial dimensions, systems of conservation laws have a rich phenomenology which is absent in the one-dimensional case. In the context of the Euler equations these are vortices (e.g. created by Kelvin-Helmholtz instabilities), multi-dimensional shock interactions, the low Mach number/incompressible limit and many more. The easiest way of extending a one-dimensional numerical method to multiple dimensions is directional splitting, i.e. the problem is replaced by a number of one-dimensional problems. This, however, has been demonstrated to require excessive grid refinement in order to capture truly multi-dimensional features even for systems much simpler than the Euler equations (e.g. [1,4,13,22]). It has been found that numerical methods should reflect essential properties of the solution at discrete level in order to avoid expensive grid refinement. Such methods are called *structure preserving*. So far, modifications of existing schemes have been suggested, but it is largely unexplored how such schemes can be derived from first principles.

The *Active Flux* scheme is a new scheme ([10], an extension of [31]) that combines a finite volume scheme with additional, independently evolved degrees of freedom which are interpreted as point values. These point values are located at cell boundaries and Active Flux thus uses a continuous reconstruction. This is a major difference to finite volume schemes. The Active Flux scheme is nevertheless able to resolve shocks (which are approximated by steep gradients), as can be seen below. The reconstruction is parabolic and therefore Active Flux is third order accurate. It has been shown in [3] for the equations of linear acoustics that the scheme is vorticity preserving without any fix which makes it a good candidate as a structure preserving method for more complicated multi-dimensional problems.

So far, the Active Flux scheme has been studied in great detail for linear equations [3, 10,31]. As explained in Sect. 2, the essential ingredient is an approximate solution operator for the initial value problem which is used to update the pointwise degrees of freedom. For linear equations, the point values can be updated using an exact evolution operator.

For nonlinear problems, an approximate evolution operator is required. By exploiting special properties the Active Flux scheme has been applied to Burgers' equation [8,9,25] and Euler equations [9,12,16,21]. Some of these extensions lose the order of convergence when applied to nonlinear equations. The approximate evolution operator has to be of sufficiently high order, e.g. local linearization as used in [9] is not sufficient to yield an overall third order scheme. In [16], a solution operator based on the Cauchy-Kovalevskaya/Lax-Wendroff procedure has been suggested which can be applied to general nonlinear hyperbolic systems in one spatial dimension and has shown the correct order of convergence when applied to one-dimensional Euler equations in practice. However, both the procedure of [16] itself and the evaluation of the higher order spatial derivatives can be rather complicated. In particular, the derivatives are required at locations where the reconstruction is not differentiable.

The aim of this paper is to provide a simpler solution operator that allows to apply the Active Flux scheme to a large class of hyperbolic conservation laws. The general idea is to keep the structure of a characteristic-based (or in multi-d characteristic-cone-based, see [3]) evolution operator but to estimate carefully the wave speeds—which are not constant in the nonlinear case. This also includes an estimate on whether a shock has occurred by self-steepening. In Sect. 3 such approximate evolution operators are provided for scalar conservation laws in one and several spatial dimensions, and in Sect. 4—for hyperbolic systems of conservation laws in one spatial dimension. This leads to algorithms significantly different from the scalar case. This paper is the first part of a sequence of papers devoted to the application of the Active Flux scheme to nonlinear problems. The case of multi-dimensional systems shall form the content of a forthcoming work. Although the examples presented here are all computed on Cartesian grids, the approximate evolution operators can be immediately applied to unstructured grids. A detailed experimental study concerning unstructured grids, however, is subject of future work.

Section 5 is describing a limiting procedure. As the Active Flux scheme is of higher order, spurious oscillations can appear. The continuous reconstructions employed in the Active Flux scheme do not allow to make immediate use of the same limiting strategies as in the case of usual finite volume schemes. Several limiters for Active Flux have been suggested in the literature: in [16,23] the parabolic reconstruction inside a cell is replaced by several parabolae joined in a continuous and monotone manner. This, however, might add implementational and computational complexity. The same is true for the hyperbolic reconstruction considered in [16] as its parameters cannot be computed analytically. Additionally, discontinuous reconstructions have been considered in [9,11,16] as limiting strategies. However, favorable properties have been deduced from the continuous reconstruction in [1,3], and these discontinuous limiting strategies violate the principle of Active Flux and move it again closer to usual finite volume schemes. This shows the need for a simple limiter that keeps the reconstruction continuous. Here, such a limiter is presented—it is optimally monotone (see Theorem 6 for precise statement) and is at the same time computationally efficient.

The paper thus is organized as follows: Approximate evolution operators are presented in Sect. 3 for scalar conservation laws and in Sect. 4 for systems. Limiting is discussed in

Sect. 5 and numerical examples for problems in one and two spatial dimensions are shown in Sect. 6.

2 The Active Flux Scheme

2.1 Finite volume Scheme

Consider the computational domain to be divided into (polyhedral) computational cells $\mathcal{C} \subset \mathbb{R}^d$ and discretize the time into points t^n, $n \in \mathbb{N}_0$ separated by (not necessarily equal) time steps Δt. Recall that in order to solve (1.1), finite volume schemes use cell averages[1]

$$\bar{q}_{\mathcal{C}} = \frac{1}{|\mathcal{C}|} \int_{\mathcal{C}} \mathrm{d}\mathbf{x}\, q(t, \mathbf{x}) \tag{2.1}$$

as discrete degrees of freedom. The cell average is updated in time using fluxes f_e through cell boundaries (edges e):

$$\frac{\bar{q}^{n+1}_{\mathcal{C}} - \bar{q}^n_{\mathcal{C}}}{\Delta t} + \sum_{e \subset \partial\mathcal{C}} \frac{|e|}{|\mathcal{C}|} f_e = 0 \tag{2.2}$$

Here, $\bar{q}^n_{\mathcal{C}}$ denotes the value of the average at time t^n.

Applying Gauss' law to (1.1), f_e can be given the interpretation of approximating

$$f_e \simeq \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathrm{d}t\, \frac{1}{|e|} \int_e \mathrm{d}x \mathbf{n}_e \cdot \mathbf{f}(q) \tag{2.3}$$

with $\mathbf{n}_e$ the outward normal to edge e.

In the one-dimensional ($d = 1$) case the cells are indexed by a finite subset of the integers. Then $\bar{q}_i$ denotes the averages in cell $\mathcal{C}_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, which is centered around x_i. The size $|C_i|$ of a cell is for simplicity denoted by Δx. Most of the results remain valid when the size of the cells varies smoothly.

2.2 Pointwise Degrees of Freedom

The Active Flux scheme is an extension of the finite volume scheme (2.2) for (1.1). Active flux uses point values $q_{\mathbf{x}}$ located at points $\mathbf{x} \in \mathbb{R}^d$ along the cell boundary as additional discrete degrees of freedom (recall that indices never denote derivatives in this paper). They approximate the value $q(t, \mathbf{x})$. So far, the following choices have been considered in the literature (see also Fig. 1):

- In one spatial dimension the point values are located at cell boundaries $x_{i+\frac{1}{2}}$, $i \in \mathbb{Z}$ and are thus rather denoted by $q_{i+\frac{1}{2}}$.
- In two spatial dimensions so far (in [3,10]) the locations of the pointwise degrees of freedom are chosen to be the endpoints and the midpoints of edges.

Active flux is not a staggered-grid finite volume method. Staggered grids consider offset grids of averages, each for a different variable; Active Flux stores additional point values of all the variables inside every cell. This approach is closer to Lagrange-basis spectral/Galerkin

[1] Boldface symbols are reserved for elements of vector spaces of the same dimension d as the space, if $d \geq 2$. Indices never denote derivatives.

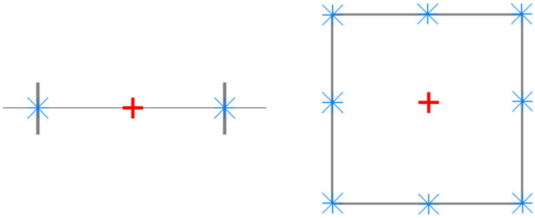

Fig. 1 The degrees of freedom used for Active Flux. Stars indicate the location of point values, and the cross (placed in the center symbolically) refers to the cell average. Left: One spatial dimensions. Right: Two spatial dimensions

methods, with the difference that the cell average is retained as one of the degrees of freedom. A particularity of the Active Flux method is the exclusive distribution of the point values along the cell boundary.

For the evolution of the point values at cell boundaries the Active Flux scheme considers an initial value problem: a reconstruction $q_{\text{recon}}(\mathbf{x})$ plays the role of the initial data, and the evolution in time can be either exact or approximate. This is explained in more detail in the next sections.

Once the time evolution of the point values is known, the numerical flux is obtained using a quadrature of (2.3) in time and along the edge. In order to obtain a third order scheme it is necessary to compute the point values also at half the time step (see [3] for further implementation details). Only the update of the average needs to be conservative, there is no notion of a conservative update for a point value.

2.3 Reconstruction

The reconstruction is interpolating the point values and the average:

$$q_{\text{recon}}(\mathbf{x}) = q_{\mathbf{x}} \quad \forall \text{ locations } \mathbf{x} \text{ of the pointwise degrees of freedom} \tag{2.4}$$

$$\frac{1}{|\mathcal{C}|}\int_{\mathcal{C}} \mathrm{d}\mathbf{x}\, q_{\text{recon}}(\mathbf{x}) = \bar{q}_{\mathcal{C}} \tag{2.5}$$

The reconstruction thus is conservative. The difference to reconstructions in the context of finite volume schemes is the fact that the reconstruction is continuous at the locations of the pointwise degrees of freedom. Additionally, the choices used for the reconstruction so far in the literature were such that the reconstruction is continuous *everywhere* in the computational domain. These particular choices are briefly reviewed next:

- In one spatial dimension, the reconstruction is chosen piecewise parabolic in [31]. This is a natural choice, as (2.4)–(2.5) amount to three conditions in each cell. It reads

$$q_{\text{recon}}(x) = -3(2\bar{q}_i - q_{i-\frac{1}{2}} - q_{i+\frac{1}{2}})\frac{(x-x_i)^2}{\Delta x^2} \tag{2.6}$$

$$+ (q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}})\frac{x-x_i}{\Delta x} + \frac{6\bar{q}_i - q_{i-\frac{1}{2}} - q_{i+\frac{1}{2}}}{4} \qquad x \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \tag{2.7}$$

  The reconstruction is continuous everywhere.
- In two spatial dimensions, in [3,10] the pointwise degrees of freedom are placed at endpoints and at the midpoints of every edge. The reconstruction is chosen always to

reduce to a parabola along any edge and, as a parabola is uniquely defined by three points, the reconstruction is thus continuous across any edge, and thus everywhere.

2.4 Evolution of Pointwise Degrees of Freedom

Active flux is a time-explicit method and thus subject to a CFL condition

$$\Delta t < \frac{L_{\min}}{\lambda_{\max}} \tag{2.8}$$

In the following, the time step is chosen based on the maximum value $\lambda_{\max}$ of the characteristic speed at the location of pointwise degrees of freedom. The shortest length $L_{\min}$ in the one-dimensional case is the size of the cell, and in the two-dimensional case half the edge length (as there is a pointwise degree of freedom located at its midpoint).

The update procedure of the Active Flux scheme for the point value is the (exact or approximate) solution of the initial value problem at its location. The initial data are given by the reconstruction. When the Active Flux scheme is applied to linear equations (as in [3,10,31]) an exact evolution is easily available. For nonlinear equations it is necessary to devise approximate evolution operators. This is the topic of Sect. 3 (for scalar nonlinear conservation laws) and Sect. 4 (for systems of conservation laws). Here, only a general statement shall be given that concerns the necessary accuracy of an approximate evolution operator. First, the following result is needed:

Lemma 1 *For $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$, both analytic and $n_1, n_2 \in \mathbb{N}$, assume*

$$f(x, \Delta x) = g(x) + \Delta x^{n_1} g^{(n_2)}(x) + \mathcal{O}((\Delta x)^{n_1+1}) \qquad \forall x \tag{2.9}$$

Then

$$f(x + \Delta x, \Delta x) - f(x, \Delta x) = g(x + \Delta x) - g(x) + \mathcal{O}((\Delta x)^{n_1+1}) \tag{2.10}$$

Proof Expand

$$\begin{aligned} & f(x + \Delta x, \Delta x) - f(x, \Delta x) \\ &\quad = g(x + \Delta x) - g(x) \\ &\qquad + \Delta x^{n_1} g^{(n_2)}(x + \Delta x) - \Delta x^{n_1} g^{(n_2)}(x) + \mathcal{O}((\Delta x)^{n_1+1}) && (2.11) \\ &\quad = g(x + \Delta x) - g(x) \\ &\qquad + \Delta x^{n_1} \Big(g^{(n_2)}(x) + \mathcal{O}(\Delta x) \Big) - \Delta x^{n_1} g^{(n_2)}(x) + \mathcal{O}((\Delta x)^{n_1+1}) && (2.12) \end{aligned}$$

□

Recall that $f \in \Theta(g)$ means that asymptotically $c_1|g| \leq |f| \leq c_2|g|$ for some $c_1, c_2 > 0$.

Theorem 1 *Assume a hyperbolic CFL condition $\Delta x \in \Theta(\Delta t)$ as $\Delta t \to 0$. If the approximate evolution $\tilde{q}(t, x)$ for any $x \in \mathbb{R}$ approximates the exact solution $q(t, x)$ at least as*

$$\tilde{q}(t, x) = q(t, x) + \mathcal{O}(t^3) \tag{2.13}$$

and the quadrature rules used to approximate (2.3) yield the exact value up to an error of $\mathcal{O}(\Delta t^\alpha \Delta x^\beta)$, $\alpha + \beta \geq 3$ then Active Flux formally achieves third order accuracy.

Proof Denote by $T_t[q_0]$ the exact evolution operator applied to initial data q_0 and evolving them to a time t, and by $\tilde{T}_t[q_0]$ its corresponding approximation.

Assume point values of $q(t^n, x)$ to be used in the reconstruction. Then, because the reconstruction is an interpolation, and taking x to be the location of one of the point values (where the interpolation is exact)

$$q^n_{\text{recon}}(x + \delta x) = q(t^n, x) + \mathcal{O}((\delta x)^\alpha \Delta x^\beta) \qquad \text{with } \alpha + \beta \geq 3, \beta \geq 1 \tag{2.14}$$

This statement can also be understood as follows: the interpolation matches the Taylor series

$$q(t^n, x) + \partial_x q(t^n, x)\delta x + \frac{1}{2}\partial^2_x q(t^n, x)(\delta x)^2 + \mathcal{O}((\delta x)^3)$$

of $q(t^n, x + \delta x)$ in δx to sufficiently high powers of δx. At the same time, the derivatives if q that appear as coefficients in this Taylor series are approximated by finite differences, which carry error terms $\mathcal{O}(\Delta x^\beta)$, $\beta > 1$. Because they all use the same point values in the approximation, lower order derivatives are approximated better.

The approximate evolution operator uses initial data from the neighbouring cells at a distance $\mathcal{O}(\Delta t)$ from some fixed x. Therefore

$$\tilde{T}_{\Delta t}[q^n_{\text{recon}}](x) = T_{\Delta t}[q^n_{\text{recon}}](x) + \mathcal{O}(\Delta t^3) \tag{2.15}$$

$$= T_{\Delta t}[q(t^n, \cdot)](x) + \mathcal{O}(\Delta t^\alpha \Delta x^\beta) \qquad \text{with } \alpha + \beta \geq 3 \tag{2.16}$$

For the average update, the numerical flux is obtained using a quadrature of (2.3), such that the numerical flux differs from the exact one by the same error. Thus, using the assumption of a hyperbolic CFL constraint, Lemma 1 implies that the leading errors cancel when the fluxes at $x + \Delta x$ and x are subtracted. One is left with

$$\bar{q}^{n+1}_i = \bar{q}^n_i - \frac{\Delta t}{\Delta x}(f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}) = \bar{q}^n_i + \text{exact flux difference} + \mathcal{O}(\Delta t^4) \tag{2.17}$$

This on total gives a numerical method of third order. □

2.5 Overview of the Algorithm

The overall algorithm of Active Flux is as follows:

1. Given cell averages and point values, compute a reconstruction according to Sect. 2.3.
2. Use the reconstruction as initial data in the update of the point values (Sect. 2.4). Approximate evolution operators for scalar nonlinear problems are discussed in Sect. 3 and for nonlinear systems in one spatial dimensions in Sect. 4.
3. Given the updated point values along the cell interfaces, compute the intercell fluxes via quadrature of (2.3). Here, a space-time Simpson rule is used.
4. Update the cell averages via (2.2).

3 Scalar Nonlinear Equations

Consider the initial value problem for the following scalar (i.e. $m = 1$) conservation law

$$\partial_t q + \nabla \cdot \mathbf{f}(q) = 0 \qquad q : \mathbb{R}^+_0 \times \mathbb{R}^d \to \mathbb{R} \tag{3.1}$$

$$q(0, \mathbf{x}) = q_0(\mathbf{x}) \qquad \mathbf{f} : \mathbb{R} \to \mathbb{R}^d \tag{3.2}$$

Assume the flux function to be smooth and convex.

3.1 Fix-Point Iteration

In the absence of shocks (3.1) can be rewritten as

$$\partial_t q + \mathbf{a}(q) \cdot \nabla q = 0 \tag{3.3}$$

with $\mathbf{a}(q) = \partial_q \mathbf{f}(q)$. The characteristics $\boldsymbol{\xi} : \mathbb{R}_0^+ \rightarrow \mathbb{R}^d$ are straight lines on which the solution is constant. They fulfill

$$\boldsymbol{\xi}'(\cdot) = \mathbf{a}\Big(q(\cdot, \boldsymbol{\xi}(\cdot))\Big) \qquad\qquad \boldsymbol{\xi}(t) = x \tag{3.4}$$

The exact solution is found by evaluating the initial data at the footpoint $\hat{\boldsymbol{\xi}} = \boldsymbol{\xi}(0)$ of the characteristic

$$q(t, \mathbf{x}) = q_0(\hat{\boldsymbol{\xi}}) \tag{3.5}$$

as q remains constant along it. This also allows to write

$$\mathbf{x} = \hat{\boldsymbol{\xi}} + \mathbf{a}(q_0(\hat{\boldsymbol{\xi}}))t \tag{3.6}$$

This equation can be solved for $\hat{\boldsymbol{\xi}}$ efficiently using a fixpoint iteration:

Theorem 2 $\hat{\boldsymbol{\xi}}^{(n)}$*, given recursively by*

$$\hat{\boldsymbol{\xi}}^{(0)} = \mathbf{x} \tag{3.7}$$

$$\hat{\boldsymbol{\xi}}^{(n)} = \mathbf{x} - \mathbf{a}(q_0(\hat{\boldsymbol{\xi}}^{(n-1)}))t \qquad n = 1, 2, \ldots \tag{3.8}$$

for $t \geq 0$, formally approximates $\hat{\boldsymbol{\xi}}$ to n-th order, i.e. $\hat{\boldsymbol{\xi}}^{(n)} = \hat{\boldsymbol{\xi}} + \mathcal{O}(t^{n+1})$.

Proof Define the error $\epsilon^{(n)}\mathbf{d}^{(n)} := \hat{\boldsymbol{\xi}}^{(n)} - \hat{\boldsymbol{\xi}}$ with $\|\mathbf{d}^{(n)}\| = 1, \epsilon^{(n)} \geq 0$ and $\mathbf{A} := \mathbf{a} \circ q_0$. Then

$$\hat{\boldsymbol{\xi}} + \epsilon^{(n)}\mathbf{d}^{(n)} = \hat{\boldsymbol{\xi}}^{(n)} \overset{(3.8)}{=} \mathbf{x} - \mathbf{A}(\hat{\boldsymbol{\xi}} + \epsilon^{(n-1)}\mathbf{d}^{(n-1)})t \tag{3.9}$$

$$= \mathbf{x} - \mathbf{A}(\hat{\boldsymbol{\xi}})t - \sum_{i=1}^{\infty} \boldsymbol{\alpha}_i \cdot (\epsilon^{(n-1)})^i \cdot t \tag{3.10}$$

where $\boldsymbol{\alpha}_i = \frac{1}{i!}\nabla_{\boldsymbol{\xi}}\mathbf{A}\big|_{\hat{\boldsymbol{\xi}}} \cdot \mathbf{d}^{(n-1)}$

$$\epsilon^{(n)} = \|\epsilon^{(n)}\mathbf{d}^{(n)}\| = \left\| \sum_{i=1}^{\infty} \boldsymbol{\alpha}_i \cdot (\epsilon^{(n-1)})^i \cdot t \right\| \tag{3.11}$$

Obviously $\epsilon^{(0)} \in \mathcal{O}(t)$. Then by induction, if $\epsilon^{(n-1)} \in \mathcal{O}(t^n)$, then for $n \geq 1$ and some constant $C \geq 0$

$$\epsilon^{(n)} \leq C \cdot \sum_{i=1}^{\infty} (\epsilon^{(n-1)})^i \cdot t \tag{3.12}$$

$$\epsilon^{(n)} \in \mathcal{O}(t^{n+1}) \tag{3.13}$$

which proves the assertion. □

This iteration seems related to the Picard iteration, but it is exact for linear problems for *any* initial data after *one* step. Therefore the above iteration is even more powerful than a standard Picard iteration.

In view of Theorem 1, an evolution operator for the discrete degree of freedom $q_{\mathbf{x}}$ located at $\mathbf{x}$ is

$$q_{\mathbf{x}}^{n+1} = q_{\text{recon}}(\hat{\boldsymbol{\xi}}^{(2)}) \tag{3.14}$$

and instead of q_0 the reconstruction q_{recon} based on values at time t^n would be used in the fixpoint iteration.

3.2 Comparison to Previous Results

Before turning to questions regarding the possible presence of shocks in the solution, compare this evolution to similar approaches available in the literature. Note that (3.14) estimates the speed of the characteristic as

$$\mathbf{a}(q_0(\hat{\boldsymbol{\xi}}^{(1)})) = \mathbf{a}(q_0(\mathbf{x} - \mathbf{a}(q_0(\mathbf{x}))t)) \tag{3.15}$$

Local linearization would correspond to taking the evolution operator $q_{\text{recon}}(\hat{\boldsymbol{\xi}}^{(1)})$, and thus estimate the characteristic speed simply by $\mathbf{a}(q_0(\mathbf{x}))$. For the special case of Burgers' equation, in [9] it is suggested to estimate the characteristic speed in one spatial dimension by

$$\frac{1}{2}(q_{i+\frac{1}{2}} + q_{i-\frac{1}{2}}) \tag{3.16}$$

However, this approach does not lead to an increase in the order of convergence (as can be shown by direct computation) and thus is not fundamentally superior to local linearization.

In [25] the exact speed of the characteristic for *linear* data is used as an estimate. Linear data in 1d ($\partial_x q_0 = \text{const}$) in (3.6) yield for Burgers' equation ($a(q) = q$)

$$\hat{\xi} = \frac{x - tq_0(x) + t\partial_x q_0(x)x}{1 + t\partial_x q_0(x)} \tag{3.17}$$

Usage of this formula as an evolution operator for the pointwise degrees of freedom requires the evaluation of the derivative at a location where the data are not differentiable. Also, equations with more complicated wave speeds lead to a lot more complicated formulae.

3.3 Modification of the Fixpoint Iteration in Order to Account for Shocks

It is well-known that nonlinear hyperbolic equations develop shocks even when the initial data are smooth. Therefore, when studying the time evolution of the reconstruction, the assumption that no shocks appear cannot always be true. However, the reconstruction is continuous. An initial value problem with continuous data does not develop a shock immediately. The shock can only appear only after a time $t_{\text{s}} > 0$. Whenever the time step happens to be small enough ($\Delta t < t_{\text{s}}$), the reconstruction did not have time to develop a shock and (3.14) is a good estimate.

This gives an explanation why in certain cases even Riemann problems can be successfully computed with the Active Flux scheme endowed with (3.14). Figure 2 shows such a successful computation of a Riemann problem between values $q_{\text{high}} = 11$ and $q_{\text{low}} = 1$ for Burgers' equation. Recall that the initial data in the cell containing the discontinuity are

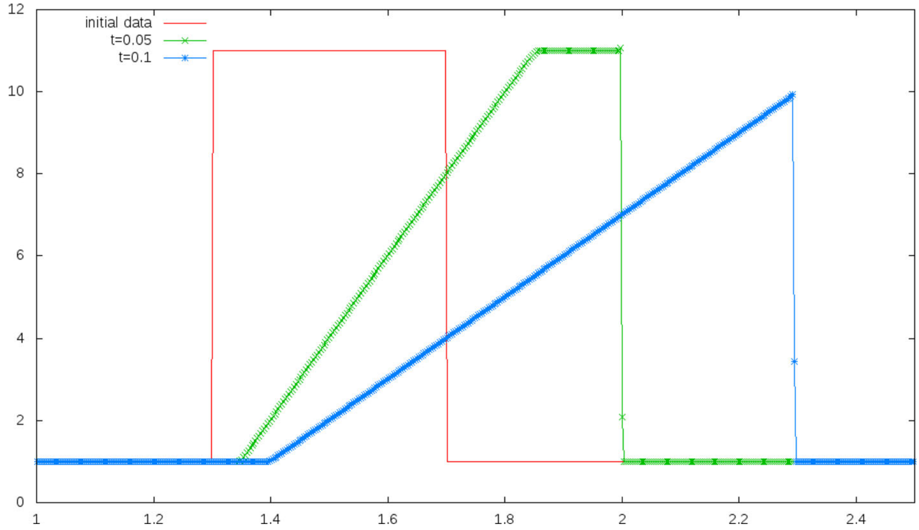

Fig. 2 Two Riemann problems for Burgers' equation solved with the Active Flux scheme using iteration (3.7)–(3.8) with $\Delta x = 3 \cdot 10^{-3}$. Power law limiting (Sect. 5) has been used. Cell averages are shown

still reconstructed continuously. One can estimate its self-steepening time in this situation as $\frac{q_{\text{high}}-q_{\text{low}}}{\Delta x}$. The CFL condition involves the maximum speed q_{high} in this case. Thus, for a Riemann problem with uniformly positive values the time step is always smaller than the estimate of the self-steepening time.

Riemann problems involving both positive and negative values do show artefacts. In [16] it has been shown, that on such Riemann problems for Burgers' equation evolution operators like the one from [25] fail. (3.14) suffers from very similar problems. In [16] it is suggested to revert to a discontinuous reconstruction in this case. However, the failure can be explained by an insufficiently accurate evolution operator rather than tracing it back to continuity of the reconstruction. In order to do this, consider an even simpler Riemann problem for Burgers' equation:

$$q_0(x) = \begin{cases} 1 & x < 0 \\ 0 & x > 0 \end{cases} \tag{3.18}$$

The exact solution is a shock moving at speed $\frac{1}{2}$. However, the evolution operator using (3.14) leaves these data stationary! Indeed, for $x > 0$ the fixpoint iteration is initialized with zero speed. For $x < 0$ the fixpoint iteration converges after one iteration to $\hat{\xi}^{(n)} = \hat{\xi}^{(1)} = x - t$ and $q_0(x - t) = 1\ \forall t$.

Additionally, transonic rarefactions show non-entropic artefacts similar to the ones observed for finite volume schemes with Riemann solvers. Examples of such are shown in Fig. 10.

All these problems are removed by modifying the initialization of the fixpoint iteration (3.14) as follows:

$$\hat{\boldsymbol{\xi}}_\ell^{(0)} = \mathbf{x} + \boldsymbol{\delta}_\ell \qquad \ell = 1, \dots, 2d \tag{3.19}$$

$$\hat{\boldsymbol{\xi}}_\ell^{(n)} = \mathbf{x} - \mathbf{a}(q_0(\hat{\boldsymbol{\xi}}_\ell^{(n-1)}))t \qquad n = 1, 2, \dots \tag{3.20}$$

On two-dimensional Cartesian grids,

$$\boldsymbol{\delta}_1 := \begin{pmatrix} \Delta x \\ 0 \end{pmatrix} \quad \boldsymbol{\delta}_2 := \begin{pmatrix} -\Delta x \\ 0 \end{pmatrix} \quad \boldsymbol{\delta}_3 := \begin{pmatrix} 0 \\ \Delta y \end{pmatrix} \quad \boldsymbol{\delta}_4 := \begin{pmatrix} 0 \\ -\Delta y \end{pmatrix} \tag{3.21}$$

$$L := \operatorname{argmax}_\ell |\mathbf{a}(q_0(\hat{\boldsymbol{\xi}}_\ell^{(1)}))| \tag{3.22}$$

and

$$q_{\mathbf{x}}^{n+1} = q_0(\hat{\boldsymbol{\xi}}_L^{(2)}) \tag{3.23}$$

The reasoning behind this algorithm is the following: Shock formation occurs because of crossing characteristics. Iteration (3.7)–(3.8) converges to both footpoints $\hat{\boldsymbol{\xi}}_+$ and $\hat{\boldsymbol{\xi}}_-$ if its initial estimates $\hat{\boldsymbol{\xi}}_\pm^{(0)}$ are chosen appropriately. The above algorithm (3.19)–(3.23) initializes the iteration with the two locations $x \pm \Delta x$, placed symmetrically around x. In order to find the solution, one thus needs to estimate which of the characteristics will have survived until time t (and not gone into the shock). The choice of (3.23) is to use the value transported by the quicker characteristic. This choice is inspired by the above example (3.18) of a Riemann problem, where it makes information flow into the right direction. Of course in general it remains an approximation.

Note that the order of the approximation is not modified, as the modification affects only the initial step of the iteration and is of the order $\mathcal{O}(\Delta x)$. On smooth solutions, characteristics do not cross, and the two initializations are expected to converge to the same final result.

Despite its simplicity, in experiments the modification has proven itself able to reliably cure both the artificially stationary shocks and the non-entropic features at transonic rarefactions. One thus may speak of the modification as an entropy fix. To actually prove a statement on the discrete entropy is subject of future work. Instead here a number of different test cases are shown: self-steepening and Riemann problems resulting in strong and weak shocks and (transonic) rarefactions (see Sects. 6.1–6.2).

4 Nonlinear Systems

Consider now an $m \times m$ nonlinear hyperbolic system of conservation laws in one spatial dimension:

$$\partial_t q + \partial_x f(q) = 0 \qquad q : \mathbb{R}_0^+ \times \mathbb{R} \to \mathbb{R}^m \tag{4.1}$$

$$f : \mathbb{R}^m \to \mathbb{R}^m \tag{4.2}$$

The Jacobian matrix is denoted by $J(q) := \nabla_q f$. Hyperbolicity guarantees that J has real eigenvalues.

In certain cases a variable change from conservative to characteristic variables $q \mapsto Q$ can be found, such that in the absence of shocks (4.1) can be rewritten as

$$\partial_t Q + \operatorname{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)\partial_x Q = 0 \tag{4.3}$$

with $\lambda_1, \dots, \lambda_m$ the eigenvalues of J. Denote the initial data as $Q_{i,0}(x) = Q_i(0, x)$, $i = 1, \dots, m$.

In the linear case this can be solved by solving an advection equation in every component (see e.g. [9]). In the nonlinear case λ_i is, in general, a function of all the components of Q:

$$\partial_t Q_1 + \lambda_1(Q_1, \dots, Q_m)\partial_x Q_1 = 0 \tag{4.4}$$
$$\partial_t Q_2 + \lambda_2(Q_1, \dots, Q_m)\partial_x Q_2 = 0 \tag{4.5}$$
$$\vdots$$
$$\partial_t Q_m + \lambda_m(Q_1, \dots, Q_m)\partial_x Q_m = 0 \tag{4.6}$$

Therefore, in general the characteristics are curved. This is also a fundamental difference to the nonlinear *scalar* case, where the characteristics remain straight. This is why applying the fixpoint iteration (3.14) to every component of (4.3) does not lead to sufficient order of accuracy (as can be checked by direct computation). A different approximate evolution operator is necessary in the case of systems, which takes into account the curvature of characteristics. Sections 4.1 and 4.2 describe two such approaches. They yield comparable results, but the strategies and resulting algorithms are fundamentally different. In particular, the algorithm in Sect. 4.1 does not assume a transformation to characteristic variables. In view of future extensions, e.g. to multiple spatial dimensions, so far it is not clear which of them would be most suitable. They also differ in the nature of necessary computations. Therefore both are presented here.

Even in case the approximate evolution operator is formulated in characteristic variables, the reconstruction still uses conservative variables (as it requires a cell average). At the locations where the initial data need to be evaluated, a transformation to characteristic variables is performed. After obtaining the result of the approximate evolution operator in characteristic variables, they are transformed back to conservative variables.

4.1 Estimating Curved Characteristics

It can be shown by explicit calculation that a straightforward extension of iteration (3.7)–(3.8) to (4.3) does not allow to prove a statement analogous to Theorem 2—the higher order terms are not correct. This is due to the fact that characteristics are now curved. However, other ways of obtaining a high order estimate can be found. The first is presented in this section, the second—in Sect. 4.2

Consider (4.1) and diagonalize $RJR^{-1} = \Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_m)$. Note that, in general, R and all λ_i depend on q. To make this explicit, write $R(q)$ and $\Lambda(q)$. The equation becomes

$$R(q)\partial_t q + \Lambda(q)R(q)\partial_x q = 0 \tag{4.7}$$

Although for some systems it is possible to find $Q(q)$ such that $\partial_t Q = R(q)\partial_t q$ (as mentioned in the introduction to this section), this is not assumed in the following algorithm.

Denote by $F^{(k)} = R^{-1} \operatorname{diag}(0, \dots, 0, \overset{k}{1}, 0 \dots, 0)R$ the projector associated with the k-th eigenvalue. Then obviously

$$\sum_{k=1}^m F^{(k)} = \mathbb{1} \qquad\qquad \sum_{k=1}^m F^{(k)}\lambda_k = J \tag{4.8}$$

In the following, matrix indices are frequently made explicit, e.g. q_i denotes the i-th component of the vector q, and R_{ij} the element of R found in its i-th row and j-th column. It is also sometimes useful to write $\lambda_i(q_1, \dots, q_m)$ instead of $\lambda_i(q)$.

Theorem 3 *Consider a predictor step* $(i = 1, \dots, m)$

$$q^{(i)}_\beta := \sum_{k,\alpha=1}^m F^{(k)}_{\beta\alpha}(x) q_{\alpha,0}\left(x - t\frac{\lambda_i(x) + \lambda_k(x)}{2}\right) \tag{4.9}$$

where $\lambda_i(x)$ is shorthand for $\lambda_i(q_{1,0}(x), \dots, q_{m,0}(x))$ $\forall i$ and analogously for $F^{(k)}(x)$. Then, with

$$\lambda_i^* := \lambda_i(q^{(i)}) \qquad\qquad R_{ij}^* := R_{ij}(q^{(i)}) \tag{4.10}$$

the approximate solution operator

$$\tilde{q}_\ell(t, x) := \sum_{i=1}^{m} (R^*)^{-1}_{\ell i} \sum_{j=1}^{m} R^*_{ij} q_{j,0}(x - \lambda_i^* t) \tag{4.11}$$

approximates the exact solution of (4.1) *with $J = R^{-1}\mathrm{diag}(\lambda_1, \dots, \lambda_m) R$ as*

$$q_k(t, x) + \mathcal{O}(t^3) \tag{4.12}$$

Note: The inconspicuously looking equation $R^*_{ij} := R_{ij}(q^{(i)})$ is non-trivial. It states that the rows of R^* are evaluated independently, each on a different predictor value.

Proof Wherever the summation is from 1 to m, it is omitted for the sake of readability. Recall

$$\sum_i R^{-1}_{\ell i} \lambda_i R_{ij} = J_{\ell j} \tag{4.13}$$

and note that

$$\partial_t q^{(i)}_\beta \Big|_{t=0} = \sum_{k,\alpha} F^{(k)}_{\beta\alpha}(x) \frac{\lambda_i + \lambda_k}{2} q'_{\alpha,0}(x) \tag{4.14}$$

$$= -\frac{1}{2}\lambda_i q'_{\beta,0}(x) - \frac{1}{2}\sum_\alpha J_{\beta\alpha} q'_{\alpha,0}(x) \tag{4.15}$$

(The prime denotes differentiation with respect to the unique argument.)

In order to compare the leading order terms in the Taylor series, differentiate the approximate evolution operator with respect to time:

$$\partial_t \tilde{q}(t, x) = \sum_{i,j} \partial_t \Big((R^*)^{-1}_{\ell i} R^*_{ij} \Big) q_{j,0}(x - \lambda_i^* t) \tag{4.16}$$

$$- \sum_{i,j} (R^*)^{-1}_{\ell i} R^*_{ij} q'_{j,0}(x - \lambda_i^* t)(\partial_t \lambda_i^* t + \lambda_i^*) \tag{4.17}$$

On the one hand then

$$\partial_t \tilde{q}(t, x)\Big|_{t=0} = \sum_{i,j} \partial_t \Big((R^*)^{-1}_{\ell i} R^*_{ij} \Big)\Big|_{t=0} q_{j,0}(x) - \sum_{i,j} R^{-1}_{\ell i} R_{ij} q'_{j,0}(x)\lambda_i \tag{4.18}$$

$$= -\sum_j J_{\ell j} q'_{j,0}(x) \tag{4.19}$$

Here $\lambda_i^*|_{t=0} = \lambda_i$, $R^*|_{t=0} = R$ was used. On the other hand

$$\partial_t^2 \tilde{q}(t, x)\Big|_{t=0} = \sum_{i,j} \partial_t^2 \Big((R^*)^{-1}_{\ell i} R^*_{ij} \Big)\Big|_{t=0} q_{j,0}(x) - 2\sum_{i,j} \partial_t \Big((R^*)^{-1}_{\ell i} R^*_{ij} \Big)\Big|_{t=0} q'_{j,0}(x)\lambda_i$$
$$+ \sum_{i,j} R^{-1}_{\ell i} R_{ij} q''_{j,0}(x)\lambda_i^2 - 2\sum_{i,j} R^{-1}_{\ell i} R_{ij} q'_{j,0}(x)\partial_t \lambda_i^* \Big|_{t=0}$$
$$= \sum_j (J^2)_{\ell j} q''_{j,0}(x)$$

$$-\sum_j \left(2\sum_i \partial_t \left((R^*)^{-1}_{\ell i} R^*_{ij} \right)\Big|_{t=0} \lambda_i + 2\sum_{i,j} R^{-1}_{\ell i} R_{ij}\partial_t \lambda^*_i \Big|_{t=0} \right) q'_{j,0}(x)$$

The term in brackets can now be expanded using the definitions of R^* and λ^*:

$$\begin{aligned}
&2\sum_{i,j} \partial_t \left((R^*)^{-1}_{\ell i} R^*_{ij} \right) \lambda_i \Big|_{t=0} + 2\sum_i R^{-1}_{\ell i} R_{ij}\partial_t \lambda^*_i \Big|_{t=0} \\
&= 2\sum_i \partial_t (R^*)^{-1}_{\ell i}\Big|_{t=0} R_{ij}\lambda_i + 2\sum_i R^{-1}_{\ell i}\partial_t R^*_{ij}\Big|_{t=0}\lambda_i + 2\sum_i R^{-1}_{\ell i} R_{ij}\partial_t \lambda^*_i \Big|_{t=0}
\end{aligned}$$

and using $\partial_t R^{-1} = -R^{-1}(\partial_t R)R^{-1}$, which follows from $\partial_t(R^{-1}R) = \partial_t \mathbb{1} = 0$:

$$= -2\sum_{i,h,s} R^{-1}_{\ell h}\partial_t R^*_{hs}\Big|_{t=0} R^{-1}_{si} R_{ij}\lambda_i + 2\sum_i R^{-1}_{\ell i}\partial_t R^*_{ij}\Big|_{t=0}\lambda_i + 2\sum_i R^{-1}_{\ell i} R_{ij}\partial_t \lambda^*_i \Big|_{t=0}$$

and with (4.15)

$$\begin{aligned}
&= \sum_\beta \left(\sum_{i,h,s} R^{-1}_{\ell h} \frac{\partial R_{hs}}{\partial q_\beta}\lambda_h R^{-1}_{si} R_{ij}\lambda_i - \sum_i R^{-1}_{\ell i}\frac{\partial R_{ij}}{\partial q_\beta}\lambda_i^2 - \sum_i R^{-1}_{\ell i} R_{ij}\frac{\partial \lambda_i}{\partial q_\beta}\lambda_i \right) q'_{\beta,0}(x) \\
&\quad + \sum_{\alpha,\beta} \left(\sum_{i,h,s} R^{-1}_{\ell h}\frac{\partial R_{hs}}{\partial q_\beta} R^{-1}_{si}\lambda_i R_{ij} - \sum_i R^{-1}_{\ell i}\lambda_i \frac{\partial R_{ij}}{\partial q_\beta} - \sum_i R^{-1}_{\ell i}\frac{\partial \lambda_i}{\partial q_\beta} R_{ij} \right) J_{\beta\alpha} q'_{\alpha,0}(x) \\
&= \sum_\beta \left(-\sum_{i,s} J_{\ell s}\frac{\partial R^{-1}_{si}}{\partial q_\beta}\lambda_i R_{ij} - \sum_{h,s} J_{\ell s} R^{-1}_{sh}\lambda_h \frac{\partial R_{hj}}{\partial q_\beta} - \sum_{h,s} J_{\ell s} R^{-1}_{sh}\frac{\partial \lambda_h}{\partial q_\beta} R_{hj} \right) q'_{\beta,0}(x) \\
&\quad - \sum_{\alpha,\beta} \frac{\partial J_{\ell j}}{\partial q_\beta} J_{\beta\alpha} q'_{\alpha,0}(x) \\
&= -\sum_{\beta,s} J_{\ell s}\frac{\partial J_{sj}}{\partial q_\beta} q'_{\beta,0}(x) - \sum_{\alpha,\beta}\frac{\partial J_{\ell j}}{\partial q_\beta} J_{\beta\alpha} q'_{\alpha,0}(x)
\end{aligned}$$

where $\frac{\partial R}{\partial q_\beta}R^{-1} = -R\frac{\partial R^{-1}}{\partial q_\beta}$, (4.13) and

$$\sum_{h,s} R_{hs} R^{-1}_{si}\lambda_h = \lambda_i \tag{4.20}$$

was used.

On the other hand, by performing the Cauchy-Kovalevskaya/Lax-Wendroff procedure on the PDE,

$$\begin{aligned}
\partial_t q_\ell &= -\sum_h J_{\ell h}\partial_x q_h \\
\partial_t^2 q_\ell &= -\sum_h \partial_t J_{\ell h}\partial_x q_h - \sum_h J_{\ell h}\partial_x\partial_t q_h \\
&= -\sum_{h,\beta}\frac{\partial J_{\ell h}}{\partial q_\beta}\partial_t q_\beta \partial_x q_h + \sum_{h,j} J_{\ell h}\partial_x \left(J_{hj}\partial_x q_j \right)
\end{aligned}$$

$$= \sum_{j,\alpha,\beta} \frac{\partial J_{\ell j}}{\partial q_\beta} J_{\beta\alpha} \partial_x q_\alpha \partial_x q_j + \sum_{h,j,\beta} J_{\ell h} \frac{\partial J_{hj}}{\partial q_\beta} \partial_x q_\beta \partial_x q_j + \sum_j (J^2)_{\ell j} \partial_x^2 q_j$$

which proves the assertion. □

It makes sense to express the matrices R in variables which make the computation simple. In the numerical examples for the full Euler equations, (ρ, v, p) are used:

$$\lambda_+ = v + c \qquad \lambda_0 = v \qquad \lambda_- = v - c \tag{4.21}$$

The transformation matrix R (from (ρ, v, p) to the eigenspace $(+, 0, -)$) reads

$$R_{+,\cdot} = (0, 1, \frac{c}{\gamma p}) \tag{4.22}$$

$$R_{0,\cdot} = (-\gamma p \rho^{-\gamma-1}, 0, \rho^{-\gamma}) \tag{4.23}$$

$$R_{-,\cdot} = (0, -1, \frac{c}{\gamma p}) \tag{4.24}$$

This gives

$$F^{(+)} = \begin{pmatrix} 0 & \frac{\rho}{2c} & \frac{\rho}{2\gamma p} \\ 0 & \frac{1}{2} & \frac{c}{2\gamma p} \\ 0 & \frac{\gamma p}{2c} & \frac{1}{2} \end{pmatrix} \quad F^{(0)} = \begin{pmatrix} 1 & 0 & -\frac{\rho}{\gamma p} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad F^{(-)} = \begin{pmatrix} 0 & -\frac{\rho}{2c} & \frac{\rho}{2\gamma p} \\ 0 & \frac{1}{2} & -\frac{c}{2\gamma p} \\ 0 & -\frac{\gamma p}{2c} & \frac{1}{2} \end{pmatrix} \tag{4.25}$$

Lemma 2 *When expressing R in the variables ρ,v and p for the Euler equations, the approximate evolution operator (4.11) is exact on contact waves ($p = \text{const}$, $v = \text{const}$).*

Proof Assume $p = \text{const}$, $v = \text{const}$ uniformly. Then

$$\begin{aligned} q^{(i)} = &\begin{pmatrix} 0 & \frac{\rho(x)}{2c(x)} & \frac{\rho(x)}{2\gamma p} \\ 0 & \frac{1}{2} & \frac{c(x)}{2\gamma p} \\ 0 & \frac{\gamma p}{2c(x)} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} \rho\left(x - t\frac{\lambda_i(x)+\lambda_+(x)}{2}\right) \\ v \\ p \end{pmatrix} \\ &+ \begin{pmatrix} 1 & 0 & -\frac{\rho(x)}{\gamma p} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \rho\left(x - t\frac{\lambda_i(x)+\lambda_0(x)}{2}\right) \\ v \\ p \end{pmatrix} \\ &+ \begin{pmatrix} 0 & -\frac{\rho(x)}{2c(x)} & \frac{\rho(x)}{2\gamma p} \\ 0 & \frac{1}{2} & -\frac{c(x)}{2\gamma p} \\ 0 & -\frac{\gamma p}{2c(x)} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} \rho\left(x - t\frac{\lambda_i(x)+\lambda_-(x)}{2}\right) \\ v \\ p \end{pmatrix} && (4.26) \\ = &\begin{pmatrix} \rho^{(i)} \\ v \\ p \end{pmatrix} && (4.27) \end{aligned}$$

with $\rho^{(i)} := \rho\left(x - t\frac{\lambda_i(x)+\lambda_0(x)}{2}\right)$, and the index is taken from the symbolic set $i \in \{+, -, 0\}$.

$$\lambda_{\pm}^{*} = v \pm c^{(\pm)} \qquad \lambda_{0}^{*} = v \tag{4.28}$$

$$R^{*} = \begin{pmatrix} 0 & 1 & \frac{c^{(+)}}{\gamma p} \\ -\gamma p(\rho^{(0)})^{-\gamma-1} & 0 & (\rho^{(0)})^{-\gamma} \\ 0 & -1 & \frac{c^{(-)}}{\gamma p} \end{pmatrix} \tag{4.29}$$

$$(R^{*})^{-1} = \begin{pmatrix} \frac{\rho^{(0)}}{c^{(-)}+c^{(+)}} & -\frac{(\rho^{(0)})^{1+\gamma}}{\gamma p} & \frac{\rho^{(0)}}{c^{(-)}+c^{(+)}} \\ \frac{c^{(-)}}{c^{(-)}+c^{(+)}} & 0 & -\frac{c^{(+)}}{c^{(-)}+c^{(+)}} \\ \frac{\gamma p}{c^{(-)}+c^{(+)}} & 0 & \frac{\gamma p}{c^{(-)}+c^{(+)}} \end{pmatrix} \tag{4.30}$$

$$\begin{aligned} \tilde{q}(t,x) = & \begin{pmatrix} \frac{\rho^{(0)}}{c^{(-)}+c^{(+)}} \\ \frac{c^{(-)}}{c^{(-)}+c^{(+)}} \\ \frac{\gamma p}{c^{(-)}+c^{(+)}} \end{pmatrix} (0, 1, \frac{c^{(+)}}{\gamma p}) \begin{pmatrix} \rho(x-\lambda_{+}^{*}t) \\ v \\ p \end{pmatrix} \\ & + \begin{pmatrix} -\frac{(\rho^{(0)})^{1+\gamma}}{\gamma p} \\ 0 \\ 0 \end{pmatrix} (-\gamma p(\rho^{(0)})^{-\gamma-1}, 0, (\rho^{(0)})^{-\gamma}) \begin{pmatrix} \rho(x-\lambda_{0}^{*}t) \\ v \\ p \end{pmatrix} \\ & + \begin{pmatrix} \frac{\rho^{(0)}}{c^{(-)}+c^{(+)}} \\ -\frac{c^{(+)}}{c^{(-)}+c^{(+)}} \\ \frac{\gamma p}{c^{(-)}+c^{(+)}} \end{pmatrix} (0, -1, \frac{c^{(-)}}{\gamma p}) \begin{pmatrix} \rho(x-\lambda_{-}^{*}t) \\ v \\ p \end{pmatrix} \end{aligned} \tag{4.31}$$

$$= \begin{pmatrix} 0 \\ v \\ 0 \end{pmatrix} + \begin{pmatrix} \frac{\rho^{(0)}}{\gamma} \\ 0 \\ p \end{pmatrix} + \begin{pmatrix} \rho(x-\lambda_{0}^{*}t) - \frac{\rho^{(0)}}{\gamma} \\ 0 \\ 0 \end{pmatrix} \tag{4.32}$$

$$= \begin{pmatrix} \rho(x-vt) \\ v \\ p \end{pmatrix} \tag{4.33}$$

This completes the proof. □

Corollary 1 *For (4.3), consider a predictor step ($i = 1, \ldots, m$)*

$$\hat{\xi}_{ij}^{*} = x - t\frac{\lambda_i(x)+\lambda_j(x)}{2} \tag{4.34}$$

where the abbreviation $\lambda_i(x) \equiv \lambda_i(Q_{1,0}(x), \ldots, Q_{m,0}(x))$ has been used.

Then, with

$$\hat{\xi}_i = x - t\lambda_i(Q_{1,0}(\hat{\xi}_{i1}^{*}), \ldots, Q_{m,0}(\hat{\xi}_{im}^{*})) \tag{4.35}$$

the approximate solution operator $Q_{i,0}(\hat{\xi}_i)$ approximates the exact solution as

$$Q_{i,0}(\hat{\xi}_i) = Q_i(t,x) + \mathcal{O}(t^3) \tag{4.36}$$

Numerical examples are shown in Sects. 6.3–6.4.

4.2 Runge–Kutta Scheme

Recall the second order Runge Kutta method for an ordinary differential equation

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \lambda(t,x) \tag{4.37}$$

$$x^* = x(0) + \alpha t \lambda(0, x(0)) \tag{4.38}$$

$$\begin{aligned} x(t) &= x(0) + t\left(1 - \frac{1}{2\alpha}\right)\lambda(0, x(0)) \\ &\quad + t\frac{1}{2\alpha}\lambda(\alpha t, x^*) + \mathcal{O}(t^3) \end{aligned} \tag{4.39}$$

For $\alpha = \frac{1}{2}$ this can be simplified to the midpoint method

$$x(t) = x(0) + t\lambda\left(\frac{1}{2}t, x(0) + \frac{1}{2}t\lambda(0, x(0))\right) + \mathcal{O}(t^3) \tag{4.40}$$

For simplicity of presentation, consider $m = 2$. The Runge-Kutta integration can be applied to the characteristic relations

$$\xi_1' = \lambda_1(Q_1(t, \xi_1), Q_2(t, \xi_1)) \tag{4.41}$$

$$\xi_2' = \lambda_2(Q_1(t, \xi_2), Q_2(t, \xi_2)) \tag{4.42}$$

that govern the time evolution of the characteristic curves $\xi_i : \mathbb{R}_0^+ \to \mathbb{R}$, $i = 1, 2$.

Theorem 4 *Consider the predictor step*

$$\hat{\xi}_1^* := x - \alpha t \lambda_1(Q_{1,0}(x), Q_{2,0}(x)) \tag{4.43}$$

$$\hat{\xi}_2^* := x - \alpha t \lambda_2(Q_{1,0}(x), Q_{2,0}(x)) \tag{4.44}$$

Then define

$$\lambda_1^* := \lambda_1\Big(Q_{1,0}(\hat{\xi}_1^* - \alpha t \lambda_1(Q_{1,0}(\hat{\xi}_1^*), Q_{2,0}(\hat{\xi}_1^*))), \tag{4.45}$$

$$Q_{2,0}(\hat{\xi}_1^* - \alpha t \lambda_2(Q_{1,0}(\hat{\xi}_1^*), Q_{2,0}(\hat{\xi}_1^*)))\Big) \tag{4.46}$$

$$\lambda_2^* := \lambda_2\Big(Q_{1,0}(\hat{\xi}_2^* - \alpha t \lambda_1(Q_{1,0}(\hat{\xi}_2^*), Q_{2,0}(\hat{\xi}_2^*))), \tag{4.47}$$

$$Q_{2,0}(\hat{\xi}_2^* - \alpha t \lambda_2(Q_{1,0}(\hat{\xi}_2^*), Q_{2,0}(\hat{\xi}_2^*)))\Big) \tag{4.48}$$

and

$$\hat{\xi}_1 := x - t\left(1 - \frac{1}{2\alpha}\right)\lambda_1(Q_{1,0}(x), Q_{2,0}(x)) - t\frac{1}{2\alpha}\lambda_1^* \tag{4.49}$$

$$\hat{\xi}_2 := x - t\left(1 - \frac{1}{2\alpha}\right)\lambda_2(Q_{1,0}(x), Q_{2,0}(x)) - t\frac{1}{2\alpha}\lambda_2^* \tag{4.50}$$

Now, $Q_{1,0}(\hat{\xi}_1)$, $Q_{2,0}(\hat{\xi}_2)$ *approximates* $Q_1(t, x)$, $Q_2(t, x)$ *with an error* $\mathcal{O}(t^3)$ *for any* $\alpha \in (0, 1]$.

Proof Apply the Runge-Kutta scheme (4.38)–(4.39) to the *backward* time evolution of (4.41)–(4.42) with $\xi_i(t) = x$, $i = 1, 2$. The equation to solve here is

$$\begin{pmatrix} \xi_1' \\ \xi_2' \end{pmatrix} = \begin{pmatrix} \lambda_1(Q_1(t, \xi_1), Q_2(t, \xi_1)) \\ \lambda_2(Q_1(t, \xi_2), Q_2(t, \xi_2)) \end{pmatrix} \tag{4.51}$$

The predictor step (4.38) reads

$$\begin{pmatrix} \xi_1(t) \\ \xi_2(t) \end{pmatrix} - \begin{pmatrix} \xi_1(t - \alpha t) \\ \xi_2(t - \alpha t) \end{pmatrix} = \alpha t \begin{pmatrix} \lambda_1(Q_1(0, \xi_1(t)), Q_2(0, \xi_1(t))) \\ \lambda_2(Q_1(0, \xi_2(t)), Q_2(0, \xi_2(t))) \end{pmatrix} \tag{4.52}$$

which gives (4.43)–(4.44) defining $\xi_i(t(1-\alpha)) =: \hat{\xi}_i^*, i = 1, 2$. Now in (4.39) the speeds at time $t(1-\alpha)$ are used. One thus needs an estimate of $Q_i, i = 1, 2$ at time $t(1-\alpha)$. Q_i is constant along the i-th characteristic. Thus, for any location ξ

$$Q_i(t(1-\alpha), \xi) = Q_i\Big(0, \xi - \alpha t \lambda_i(Q_1(0, \xi), Q_2(0, \xi))\Big) \tag{4.53}$$

is an estimate of the solution, which yields (4.46)–(4.48). □

Note: The extension to any m is obtained analogously.

By analogy with the modified fixpoint iteration (3.19)–(3.20) the following modification is suggested for the case of systems: Instead of (4.43)–(4.44), compute ($\ell = 1, 2$)

$$\hat{\xi}^*_{1,\ell} := x - \alpha t \lambda_1(Q_{1,0}(x + \delta_\ell), Q_{2,0}(x + \delta_\ell)) \tag{4.54}$$

$$\hat{\xi}^*_{2,\ell} := x - \alpha t \lambda_2(Q_{1,0}(x + \delta_\ell), Q_{2,0}(x + \delta_\ell)) \tag{4.55}$$

$$\delta_1 = \Delta x \; \delta_2 = -\Delta x \tag{4.56}$$

For each ℓ, proceed with the algorithm to obtain $\hat{\xi}_{1,\ell}, \hat{\xi}_{2,\ell}$. Choose

$$\hat{\xi}_1 = \begin{cases} \hat{\xi}_{1,1} & \text{if } |\hat{\xi}_{1,1} - x| > |\hat{\xi}_{1,2} - x| \\ \hat{\xi}_{1,2} & \text{else} \end{cases} \qquad \hat{\xi}_2 = \begin{cases} \hat{\xi}_{2,1} & \text{if } |\hat{\xi}_{2,1} - x| > |\hat{\xi}_{2,2} - x| \\ \hat{\xi}_{2,2} & \text{else} \end{cases} \tag{4.57}$$

Define the approximate solution operator by $(Q_{1,0}(\hat{\xi}_1), Q_{2,0}(\hat{\xi}_2))$.

Numerical examples are shown in Sects. 6.3–6.4.

5 Limiting in One Spatial Dimension

The Active Flux scheme uses a conservative reconstruction in order to evolve the pointwise degrees of freedom. It has to fulfill several conditions (i.e. (2.4)–(2.5)). In one spatial dimension conservation and interpolation of the two point values at cell boundaries amount to three conditions. The natural choice therefore is a parabola (e.g. in [31]). However, polynomials in general do not fulfill a maximum principle: The maximum of the reconstruction $q_{\text{recon}}(x)$ in cell i can exceed $\max(\bar{q}_i, q_{i+\frac{1}{2}}, q_{i-\frac{1}{2}})$. To correct this (whenever possible) is the objective of the limiting procedure introduced below.

The starting point is an analysis of the failure of the parabolic reconstruction to be monotone. Assume in the following that $q_{i-\frac{1}{2}} < q_{i+\frac{1}{2}}$; for the opposite situation analogous statements are true.

Theorem 5 (i) *If* $q_{i-\frac{1}{2}} < \bar{q}_i < q_{i+\frac{1}{2}}$ *then there exists a monotone continuous function satisfying* (2.4)–(2.5).

(ii) *With* $r := \frac{q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}}{3} > 0$, *if* $\bar{q}_i > q_{i+\frac{1}{2}} - r$ *or* $\bar{q}_i < q_{i-\frac{1}{2}} + r$ *then the parabolic reconstruction* (2.7) *is not monotone.*

Proof (i) E.g. a piecewise linear function can easily be constructed to fulfill the conditions.

(ii) The reconstruction (2.7) has a maximum inside $[-\frac{\Delta x}{2}, \frac{\Delta x}{2}]$ if the average is too close to the point values. The maximum is located at $\frac{\Delta x(q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}})}{12\bar{q}_i - 6(q_{i-\frac{1}{2}} + q_{i+\frac{1}{2}})}$. Equating this to $\pm\frac{\Delta x}{2}$ yields the bounds.

□

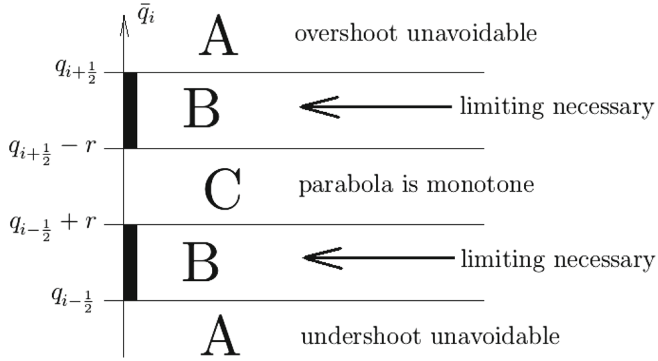

Fig. 3 Values of $\bar{q}_i$ for which limiting is possible and necessary

Thus, there are three possible cases, which are shown in Fig. 3.

A. $\bar{q}_i > q_{i+\frac{1}{2}}$ or $\bar{q}_i < q_{i-\frac{1}{2}}$: no continuous monotone reconstruction exists: use the parabolic reconstruction.

B. $q_{i-\frac{1}{2}} < \bar{q}_i < q_{i-\frac{1}{2}} + r$ or $q_{i+\frac{1}{2}} - r < \bar{q}_i < q_{i+\frac{1}{2}}$: correction is needed.

C. $q_{i-\frac{1}{2}} + r \le \bar{q}_i \le q_{i+\frac{1}{2}} - r$: the parabolic reconstruction is monotone and no limiting needed.

The following function corrects the failure of the parabolic reconstruction:

Theorem 6 (Power law limiting) *Under the conditions of Theorem 5(ii) the function*

$$p_N(x) = q_{i-\frac{1}{2}} + (q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}})\left(\frac{x - x_i + \Delta x/2}{\Delta x}\right)^N \qquad x_i - \frac{\Delta x}{2} < x < x_i + \frac{\Delta x}{2} \tag{5.1}$$

with $N = \dfrac{q_{i+\frac{1}{2}} - \bar{q}_i}{\bar{q}_i - q_{i-\frac{1}{2}}}$ *fulfills* (2.4)–(2.5) *and is monotone.*

Proof Monotonicity and (2.4) are obvious. For (2.5) compute

$$\frac{1}{\Delta x}\int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} \mathrm{d}x\, p_N(x) = q_{i-\frac{1}{2}} + (q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}})\frac{1}{\Delta x^{N+1}}\int_0^{\Delta x} \mathrm{d}x\, x^N \tag{5.2}$$

$$= q_{i-\frac{1}{2}} + (q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}})\frac{1}{N+1} = \bar{q}_i \tag{5.3}$$

□

Thus, using the limiter amounts to replacing the parabolic reconstruction by (5.1) in the region B:

$$q_{\text{recon}}(x) = \begin{cases} p_N(x) & (q_{i-\frac{1}{2}} < \bar{q}_i < q_{i-\frac{1}{2}} + r) \text{ or } (q_{i+\frac{1}{2}} - r < \bar{q}_i < q_{i+\frac{1}{2}}) \\ \text{parabolic (2.7)} & \text{else} \end{cases} \tag{5.4}$$

The effect is illustrated in Fig. 4.

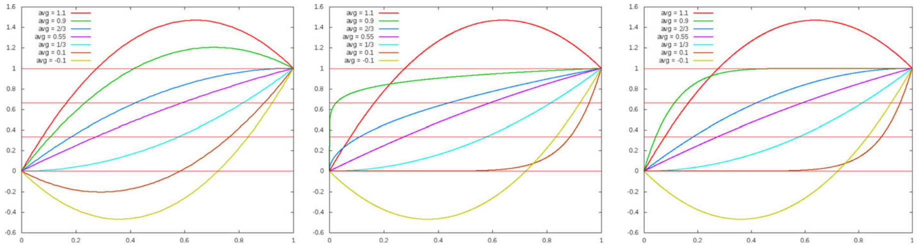

Fig. 4 Example of reconstructions with $\Delta x = 1$, $x_i = 0.5$. The data are $q_{i-\frac{1}{2}} = 0$, $q_{i+\frac{1}{2}} = 1$ and different averages: $\bar{q}_i \in \{1.1, 0.9, \frac{2}{3}, 0.55, \frac{1}{3}, 0.1, -0.1\}$. *Left*: Parabolic reconstruction (2.7). It is only monotone for $\frac{1}{3} \le \bar{q}_i \le \frac{2}{3}$. *Center*: Limiting applied. Now the reconstruction is monotone for $q_{i-\frac{1}{2}} \le \bar{q}_i \le q_{i+\frac{1}{2}}$. Outside this range no monotone function can be found, and parabolic reconstruction is still used. *Right*: A version of the limiting symmetric with respect to $N \mapsto \frac{1}{N}$ according to (5.5)

Notes:

(i) If $\bar{q}_i > q_{i-\frac{1}{2}} + r$ then $N < 2$. Therefore inside the region C ($q_{i-\frac{1}{2}} + r < \bar{q}_i < q_{i+\frac{1}{2}} - r$) where the parabolic reconstruction (2.7) is monotone, the power law would have a formal approximation order less than the parabola.

(ii) Usage of the parabolic reconstruction whenever an overshoot (undershoot) is unavoidable (region A) is making the limiter affect maxima (minima) as little as possible, and thus to avoid clipping. In practice, limiting is discarded if it would imply $\max(N, \frac{1}{N}) > 50$.

(iii) As can be seen from Fig. 4 (center), the curves for N and $1/N$ are not symmetric. One thus can consider

$$q_{\text{recon}}(x) = \begin{cases} p_N(x) & \text{if } q_{i-\frac{1}{2}} < \bar{q}_i < q_{i-\frac{1}{2}} + r \\ q_{i+\frac{1}{2}} - (q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}) \left(\frac{\Delta x/2 - x + x_i}{\Delta x} \right)^{1/N} & \text{if } q_{i+\frac{1}{2}} - r < \bar{q}_i < q_{i+\frac{1}{2}} \\ \text{parabolic (2.7)} & \text{else} \end{cases} \tag{5.5}$$

as a limiting strategy instead of (5.4). The result is shown in Fig. 4 (right).

(iv) For $q_{i-\frac{1}{2}} > q_{i+\frac{1}{2}}$ the parabola is monotone if $\bar{q}_i$ fulfills

$$q_{i+\frac{1}{2}} - \frac{q_{i-\frac{1}{2}} - q_{i+\frac{1}{2}}}{3} \le \bar{q}_i \le q_{i-\frac{1}{2}} - \frac{q_{i-\frac{1}{2}} - q_{i+\frac{1}{2}}}{3} \tag{5.6}$$

but the formula (5.1) remains unchanged.

In Fig. 5 the effect of limiting is shown for linear advection. The initial data are compared to the numerical solution on a periodic grid after one revolution.

The approximate evolution operators in Sects. 3 and 4 approximate the solution at time t by evaluating the initial data at some particular location. The above limiting procedure thus guarantees that the point value update satisfies the maximum principle in all cases when a monotone reconstruction is at all possible. This does not mean, however, that the full numerical solution will be free of spurious oscillations. The finite volume update of the average might still lead to the appearance of oscillations. In practice, for linear advection one does observe oscillations for large CFL numbers, but they are much smaller than without limiting. A limiting of the finite volume step for Active Flux has not yet been considered in the literature and is subject of future work.

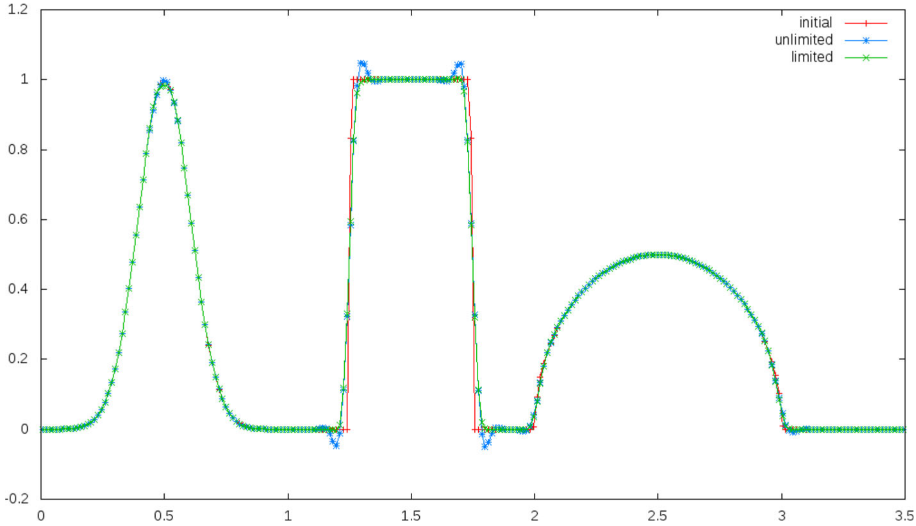

Fig. 5 Effect of limiting for linear advection with speed 1.0. Here, $\Delta x = 3.5/250$, the CFL number is chosen 0.45 and the curves show cell averages of the initial data and of the solution at $t = 3.5$, i.e. after one revolution with periodic boundaries. The blue curve shows the solution without limiting, and the green one with limiting

For systems, limiting is applied to conservative quantities individually. Numerical examples of limiting for nonlinear problems are shown in Sect. 6.

6 Numerical Examples

The Active Flux scheme endowed with the approximate solution operators of the above section is now applied to several problems in order to assess its abilities experimentally. First, scalar equations and systems in one spatial dimension are considered (Sects. 6.1–6.4); in Sect. 6.6 the scheme is applied to multi-dimensional scalar conservation laws.

The CFL condition is applied using the maximum absolute value of the characteristic speed (eigenvalue of the Jacobian in case of systems) evaluated at the point values (edge midpoint values in the multi-dimensional case).

Third order accuracy in time requires the computation of the point values at both the full and the half time step. Simpson's rule in time then is used to compute the fluxes necessary for the cell averages update (2.2).

6.1 Burgers' Equation

Here, Burgers' equation

$$\partial_t q + \partial_x \left(\frac{q^2}{2} \right) = 0 \tag{6.1}$$

is solved with the Active Flux scheme. In all cases the approximate evolution operator (3.23) is used with $a(q) = q$. Additionally, Sect. 6.1.3 shows the artefacts appearing upon usage of the simple, unmodified fixpoint iteration (3.14), and their absence when using (3.23).

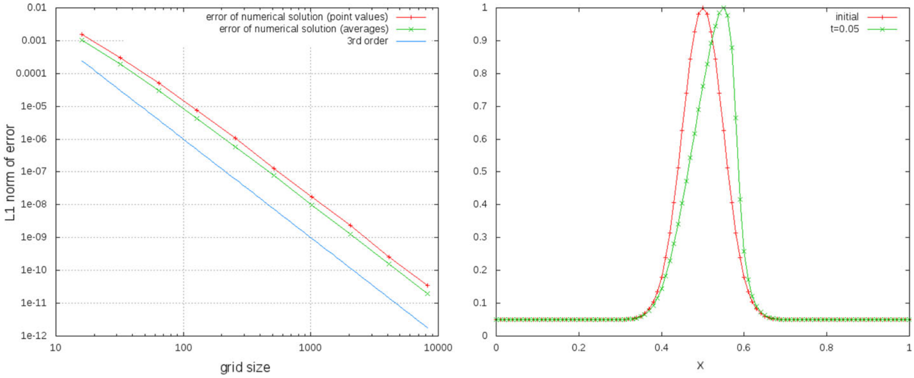

Fig. 6 Burgers' equation solved with Active Flux using the modified fixpoint iteration (3.23). Left: Third order convergence of the numerical solution on both point values and averages. Right: Setup and numerical solution for $\Delta x = 1/100$. The error is evaluated at $t = 0.05$. The CFL number is 0.45

6.1.1 Convergence Study

In order to assess the experimental convergence rate, Gaussian initial data are evolved on grids of different refinement. The simulation is stopped before a shock occurs. Figure 6 (right) shows the setup and the solution at final time on a grid with $\Delta x = 1/100$. Figure 6 (left) shows the ℓ_1 norm of the error of both the point values and the averages at $t = 0.05$. The reference solution is obtained by evolving a piecewise linear approximation of the data exactly on a grid 30 times finer. For the error of the averages, Simpson's rule is used to compute cell averages of the reference solution. Periodic boundaries are used, though the Gaussian decays sufficiently quickly to a constant towards the boundaries. Limiting is not used.

6.1.2 Self-steepening

Figures 7 and 8 show continuous initial data which self-steepen into a shock. In both cases the arising shock connects a positive and a negative value—a situation in which the simple fixpoint iteration (3.14) is known to fail. Figure 7 (left) shows such a failure: the shock is stationary, which violates the Rankine-Hugoniot condition (this problem has been first reported in [16]). Figure 7 (right) demonstrates that the modified iteration (3.23) yields the correct shock speed 0.75. Moreover, in Fig. 9 it is shown that the simple fixpoint iteration (3.14) produces a spurious oscillation when a shock appears. This is not the usual oscillation due to the high order of the method, as it is removed upon usage of the modified fixpoint iteration (3.23), even if limiting is not applied. The reason for the appearance of the oscillation is related to the stationary-shock artefact of Fig. 7 (left). As is emphasized in [16], stationary point values imply fluxes which do not change in time. At the location of the shock, the two fluxes of the cell are different and the average keeps growing. This pile-up is observed as an oscillation in Fig. 9, although the non-zero value on the left side of the shock is enough to make it travel at the right speed.

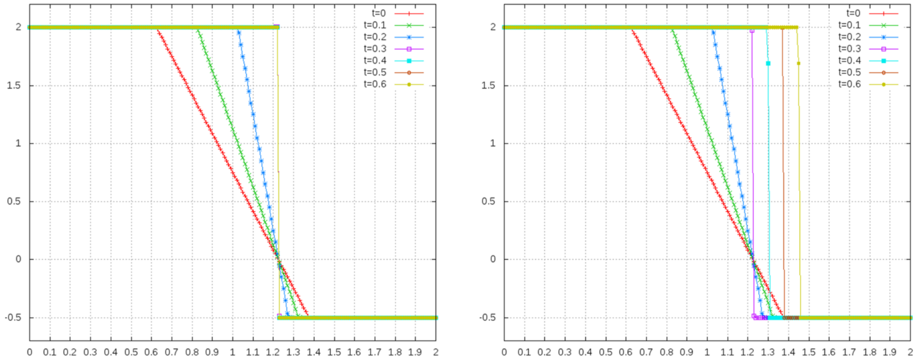

Fig. 7 Burgers' equation evolved with the Active Flux scheme. The initial data self-steepen and a shock forms at $t = 0.3$. Point values are shown at times $t \in \{0, 0.1, \dots, 0.6\}$. The solution is computed on a much larger grid such that the boundaries are of no influence. $\Delta x = 1/100$ and the CFL number is 0.9. Limiting procedure according to Theorem 6/Eq. (5.4) is used. Left: Usage of fixpoint iteration (3.14) yields zero shock speed (the corresponding lines are on top of each other). Right: The modified iteration (3.23) yields the correct shock speed

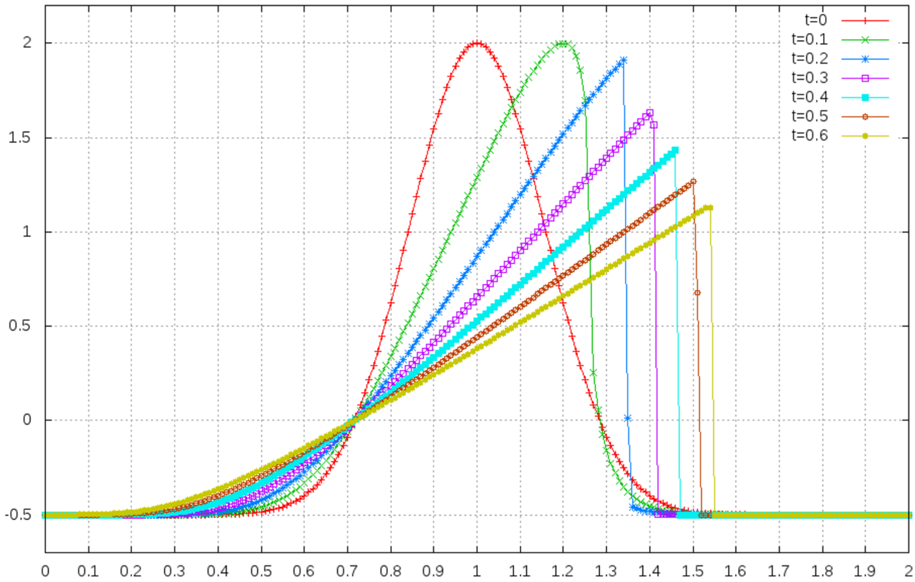

Fig. 8 Burgers' equation evolved with the Active Flux scheme. Point values are shown at times $t \in \{0, 0.1, \dots, 0.6\}$. The solution is computed on a much larger grid such that the boundaries are of no influence. $\Delta x = 1/100$ and the CFL number is 0.9. Limiting procedure according to Theorem 6/Eq. (5.4) is used

6.1.3 Riemann Problems

Finally, a number of Riemann problems are used in order to assess the properties of the suggested modification of the fixpoint iteration. Figure 10 shows an initial setup chosen to include many interesting cases at once, including strong and weak shocks and rarefactions, and shocks and rarefactions where the discontinuity crosses 0. These latter are known to be

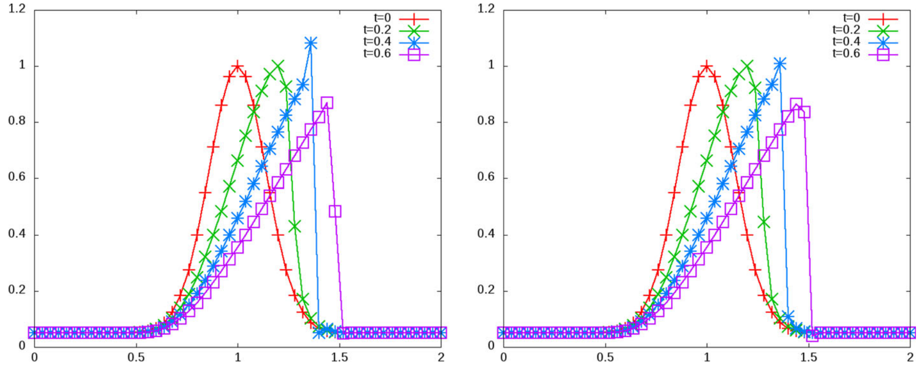

Fig. 9 Burgers' equation evolved with the Active Flux scheme. Point values are shown at times $t \in \{0, 0.2, 0.4, 0.6\}$. $\Delta x = 1/25$ and the CFL number is 0.45. No limiting is used. Left: Simple fixpoint iteration (3.14) is producing a spurious oscillation. Right: Modified fixpoint iteration (3.23) yields much better results, even without limiting

the most problematic. On the right also some of the constant states are 0. The boundaries are periodic.

In the same Figure, a solution at time $t = 0.1$ is shown which is using the simple, unmodified fixpoint iteration (3.14). One observes that only uniformly positive or uniformly negative shocks are evolved correctly. Some of the rarefactions are correct, some have nonentropic features.

For comparison, Fig. 11 shows the evolution of the same setup using the modified fixpoint iteration (3.23). Now all the shocks have the correct speed, and the rarefactions do not contain additional non-entropic shocks. The strong rarefaction with values symmetric around 0 seems particularly difficult to capture. For large CFL numbers, little artefacts have been observed, which, however, vanish upon grid refinement.

To the author's knowledge the performance of Active Flux for nonlinear problems has never been studied on transonic rarefactions or strong shocks in the literature.

6.2 Other Scalar Equations

Consider the following scalar conservation law

$$\partial_t q + \partial_x \left(\frac{q^4}{4} \right) = 0 \tag{6.2}$$

The speed of a shock joining q_L and q_R is

$$s = \frac{q_\mathrm{L}^3 + q_\mathrm{R} q_\mathrm{L}^2 + q_\mathrm{R}^2 q_\mathrm{L} + q_\mathrm{R}^3}{4} \tag{6.3}$$

E.g. a shock joining 1 and -5 is moving at speed -26.

The self-similar rarefaction is given by $\sqrt[3]{\frac{x}{t}}$. Figure 12 shows again a selection of shocks and rarefaction together with the numerical and the exact solution. Usage of the modified fixpoint iteration (3.23) allows to resolve all the shocks and rarefactions correctly.

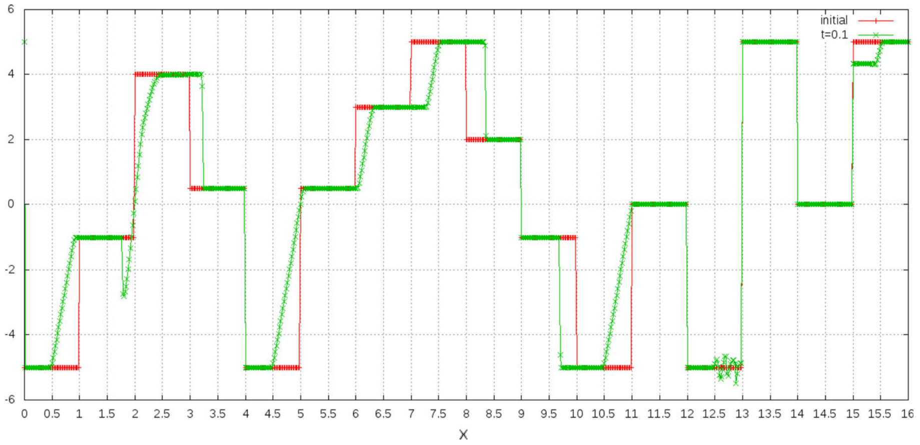

Fig. 10 Burgers' equation evolved with the Active Flux scheme using the simple fixpoint iteration (3.14): wrong shock speeds and artefacts in transonic rarefactions are visible. The initial data are piecewise constant. Point values of the solution are shown at time $t = 0.1$. The boundaries are periodic. $\Delta x = 2/100$ and the CFL number is 0.45. Limiting procedure according to Theorem 6/Eq. (5.4) is used

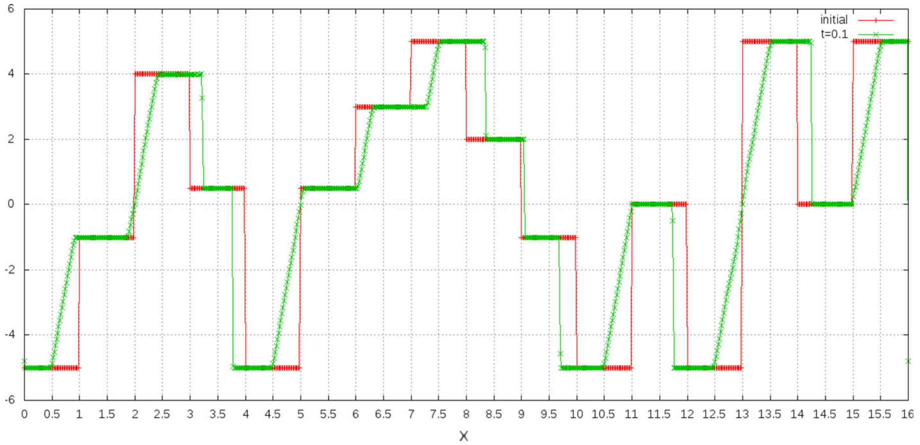

Fig. 11 Burgers' equation evolved with the Active Flux scheme using the modified fixpoint iteration (3.23). All the shock speeds are correct and the rarefactions are not distorted by non-entropic shocks. Point values of the solution are shown at time $t = 0.1$. Setup as in Fig. 10

6.3 *p*-system

The p-system

$$\partial_t \rho + \partial_x v = 0 \tag{6.4}$$

$$\partial_t v + \partial_x p(\rho) = 0 \qquad\qquad p(\rho) = \rho^\gamma \tag{6.5}$$

in case of smooth solutions can be rewritten in the form (4.3) as

$$\partial_t \left(\frac{2\rho c}{\gamma + 1} \pm v \right) \pm c \partial_x \left(\frac{2\rho c}{\gamma + 1} \pm v \right) = 0 \tag{6.6}$$

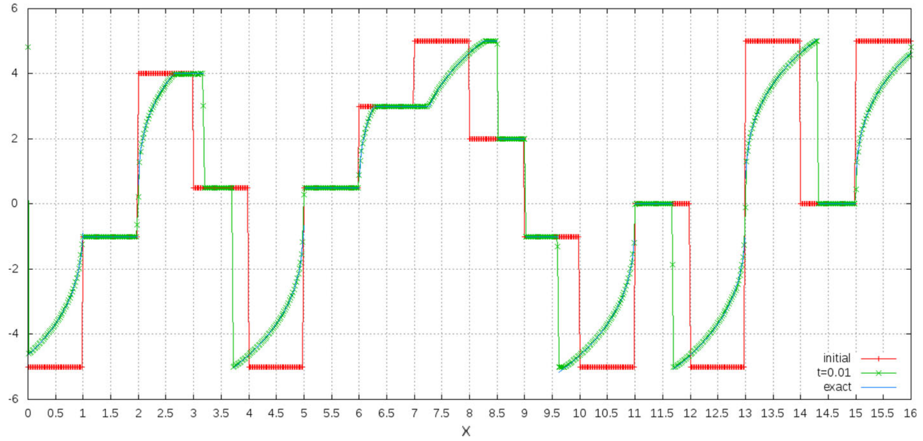

Fig. 12 Equation (6.2) evolved with the Active Flux scheme using the modified fixpoint iteration (3.23). All the shock speeds and rarefactions are correctly resolved. The solid line shows the exact solution. The initial data are piecewise constant. Point values of the solution are shown at time $t = 0.01$. The boundaries are periodic. $\Delta x = 2/100$ and the CFL number is 0.45. Limiting procedure according to Theorem 6/Eq. (5.4) is used

with $c = \sqrt{p'(\rho)}$. In the following, $\gamma = 1.4$ is used.

As the eigenvalues cannot switch sign, and never vanish, Active Flux solving the p-system seems less prone to artefacts.

6.3.1 Convergence Study

Figure 13 (left) demonstrates the correct order of convergence for both algorithms of Sects. 4.1 and 4.2. The initial setup (shown in Fig. 13 (right)) is a Gaussian in the density and $v \equiv 0$. During the evolution two waves are forming which self-steepen. At the final time $t = 0.2$ they have not formed shocks, though. The reference solution is obtained by evolving the problem on a highly resolved grid of $16{,}384 = 2^{14}$ cells with the algorithm of Sect. 4.2.

6.3.2 Riemann Problem

Consider the following Riemann problem

$$v = \begin{cases} \rho = 2, v = 1 & 0.3 < x < 0.7 \\ \rho = 0.1, v = -0.5 & \text{else} \end{cases} \tag{6.7}$$

The numerical solution is shown in Fig. 14.

6.4 Isentropic Euler Equations

Consider the isentropic Euler equations

$$\partial_t \rho + \partial_x(\rho v) = 0 \tag{6.8}$$

$$\partial_t(\rho v) + \partial_x(\rho v^2 + p(\rho)) = 0 \qquad p(\rho) = K\rho^\gamma \tag{6.9}$$

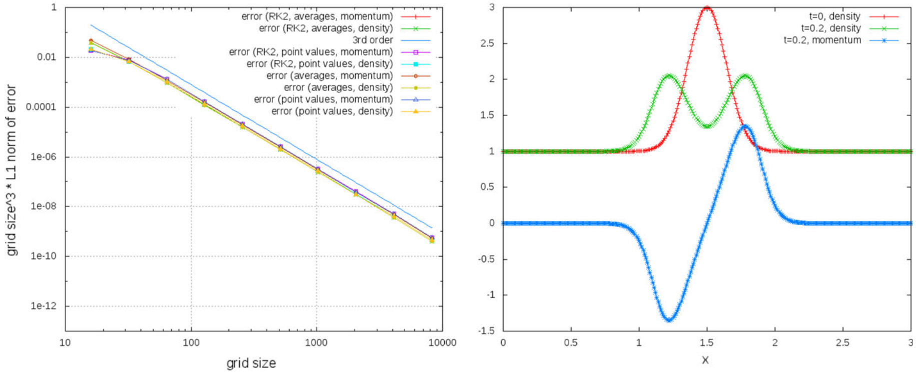

Fig. 13 The p-system evolved with the Active Flux scheme. Left: Third order convergence of the numerical solution on both point values and averages, for momentum v and density ρ, using both the algorithm from Sect. 4.1 and from Sect. 4.2 (the latter marked RK2). The lines showing the results for different schemes and quantities virtually lie on top of each other indicating comparable error. Right: Setup and numerical solution for $\Delta x = 1/100$ showing point values. No limiting used

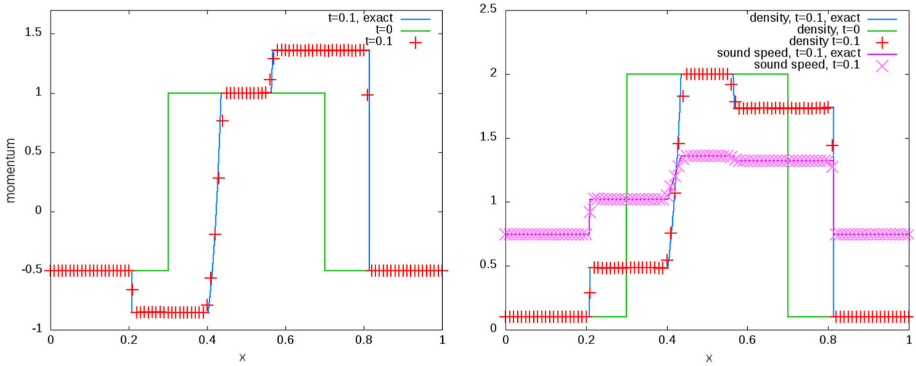

Fig. 14 The p-system evolved with the Active Flux scheme using the iteration from Sect. 4.2 (no difference was apparent upon using the iteration from 4.1). Here, $\Delta x = 1/100$ and CFL $= 0.45$; the limiting function (5.5) is applied to the conserved quantities. Periodic boundaries are used and the curves show the point values at time $t = 0.1$. Left: Momentum v. Right: Density ρ and speed of sound c

On smooth solutions this system is equivalent to

$$\partial_t \left(v \pm \frac{2c}{\gamma - 1} \right) + (v \pm c) \partial_x \left(v \pm \frac{2c}{\gamma - 1} \right) = 0 \tag{6.10}$$

with $c = \sqrt{\gamma p(\rho)/\rho}$. In the following, $K = 1$, $\gamma = 1.4$ are used.

6.4.1 Convergence Study

Figure 15 shows the setup of a Gaussian initial density and no velocity. Its evolution (before a shock forms) is used to study the convergence of the method. The reference solution is the setup solved with the iteration from Sect. 4.2 on a grid of 16,384 cells. Third order is confirmed experimentally.

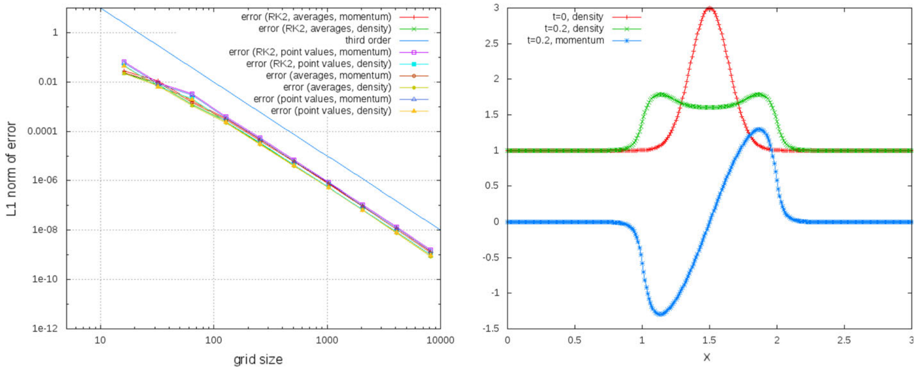

Fig. 15 Isentropic Euler equations solved with the Active Flux scheme. Left: Third order convergence of the numerical solution on both point values and averages, for momentum ρv and density ρ, using both the algorithm from Sect. 4.1 and from Sect. 4.2 (the latter marked RK2). The lines showing the results for different schemes and quantities virtually lie on top of each other indicating comparable error. Right: Setup and numerical solution for $\Delta x = 1/100$ showing point values. No limiting used

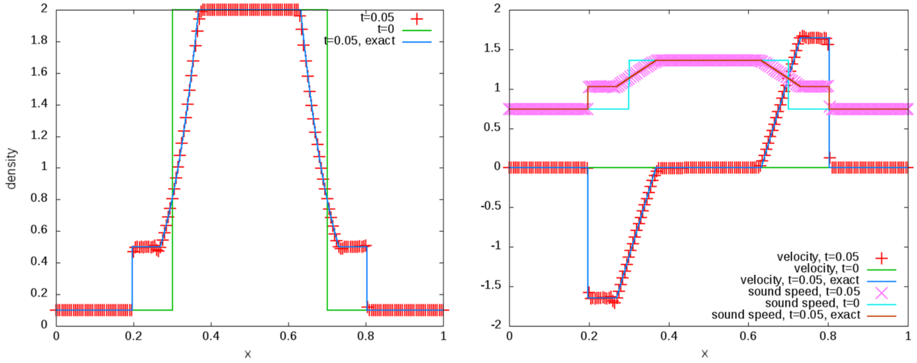

Fig. 16 Isentropic Euler equations solved with the Active Flux scheme using iteration from Sect. 4.2 and the fix (4.54)–(4.57). Here, $\Delta x = 1/200$ and CFL $= 0.45$; the limiting function (5.4) is applied to the conserved quantities. Periodic boundaries are used and the curves show the point values at time $t = 0.05$. Left: Density ρ. Right: Velocity v, sound speed c

6.4.2 Riemann Problems

The scheme is now applied to Riemann problems. Here it becomes important to use the modification (4.54)–(4.57). Figure 16 shows a setup with a transonic rarefaction, accurately resolved, and Fig. 17 shows a strong shock and a double rarefaction. The two tests cover the cases of the two eigenvalues of the system having different and same sign. The double shock case shows little artefacts which vanish upon refining the grid.

6.5 Full Euler Equations

The full Euler equations

$$\partial_t \rho + \partial_x(\rho v) = 0 \tag{6.11}$$

$$\partial_t(\rho v) + \partial_x(\rho v^2 + p) = 0 \tag{6.12}$$

$$\partial_t e + \partial_x(v(e + p)) = 0 \tag{6.13}$$

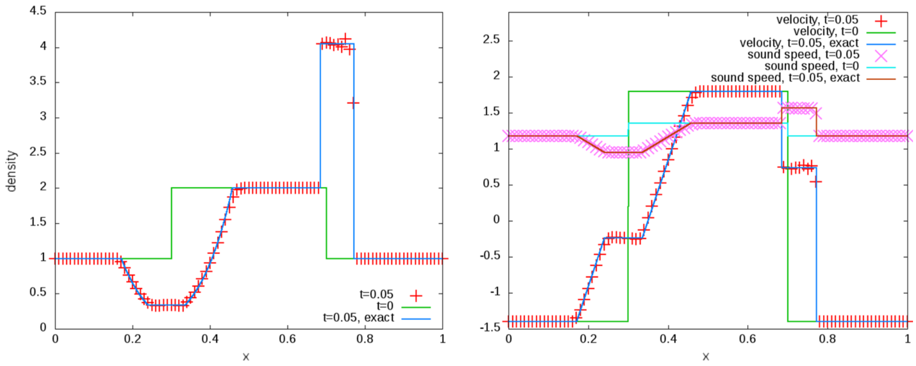

Fig. 17 Isentropic Euler equations solved with the Active Flux scheme. Apart from the values for the states of the Riemann problem ($\rho \in \{1, 2\}$, $v \in \{-1.4, 1.8\}$), solution method and plot as in Fig. 16

with the ideal equation of state

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho v^2 \qquad \gamma > 1 \tag{6.14}$$

form a hyperbolic system of conservation laws, but do not admit characteristic variables. Thus, the most general solution algorithm (4.11) is required. In the following, $\gamma = 1.4$ is used.

6.5.1 Convergence Study

To demonstrate the convergence of the method, the test from [16] is run:

$$\rho_0(x) = p_0(x) = 1 + \frac{1}{2}e^{-80(x-0.5)^2} \quad v_0(x) = 0 \tag{6.15}$$

The numerical results are compared at $t = 0.25$ to a reference solution obtained on a grid of 2048 points. In Fig. 18, one observes third order convergence, in agreement with the theoretical expectation.

6.5.2 Riemann Problem

To assess the performance of the numerical method on discontinuous problems, two Riemann problems are computed: the Sod shock tube ([27], Fig. 19, *left*) and the Lax shock tube ([19], Fig. 19, *right*). One observes the good perfromance of Active Flux even on these discontinuous setups.

6.5.3 Interaction Between a Shock and a Sound Wave

Finally, to demonstrate the performance of the algorithm on a more challenging setup, the Shu-Osher test [26] is shown in Fig. 20. One observes that due to its high order, Active Flux is able to capture the details of the interaction on poorly resolved grids without difficulty (compare e.g. to [5]).

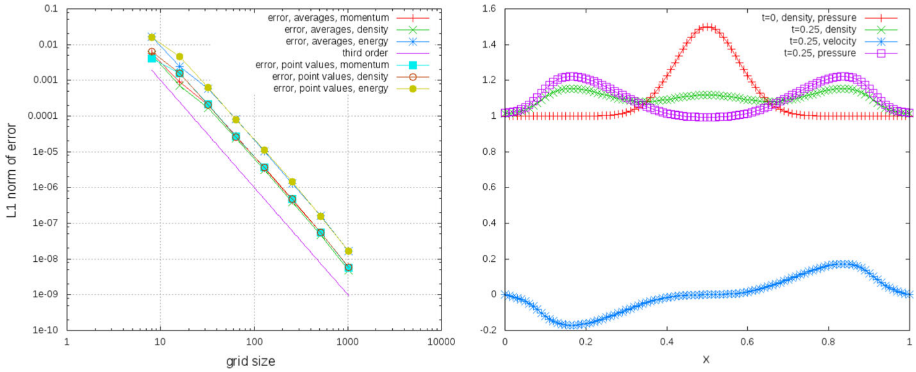

Fig. 18 Full Euler equations solved with the Active Flux scheme using the approximate evolution operator (4.11). Left: Third order convergence of the numerical solution on both point values and averages, for momentum ρv, density ρ and energy e. The lines virtually lie on top of each other indicating comparable error. A CFL number of 0.7 is used. Right: Setup and numerical solution for $\Delta x = 1/100$ showing point values. No limiting used

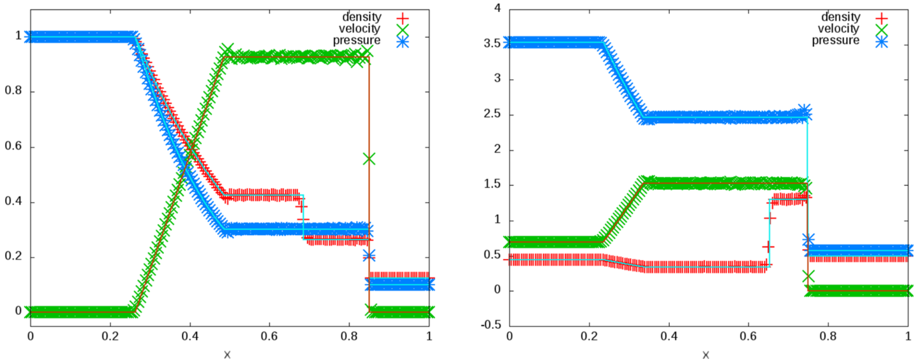

Fig. 19 Riemann problem setups for the full Euler equations. The approximate solution operator (4.11) with limiting is used on a grid with $\Delta x = 1/200$ and with a CFL number of 0.7. Point values are shown at $t = 0.1$. Left: Sod's test problem [27]. Right: Lax's test problem [19]. Solid lines show the exact solution

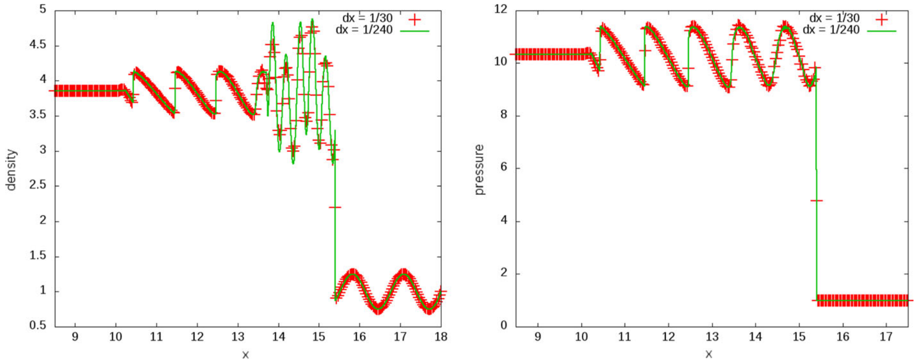

Fig. 20 The Active Flux method is used to solve the Shu-Osher test [26]. The approximate solution operator (4.11) with limiting is used with a CFL number of 0.7 and on grids with $\Delta x = 1/30$ (crosses) and 1/240 (solid line). Point values are shown at $t = 0.18$. Left: Density. Right: Pressure

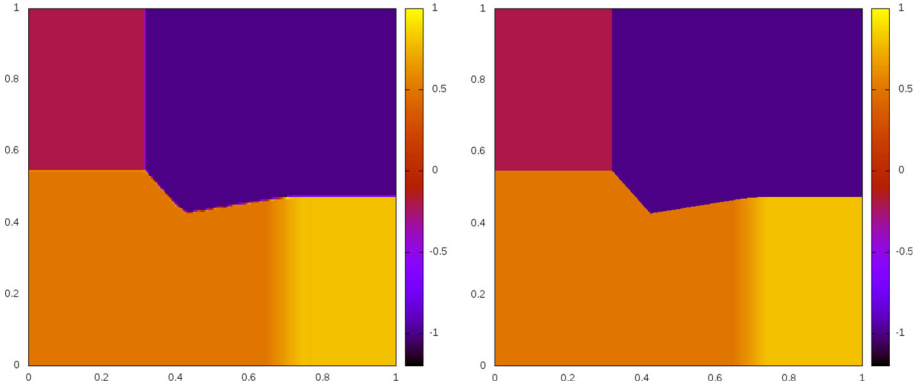

Fig. 21 The multi-dimensional Burgers' equation (6.16) solved with the Active Flux scheme using the fixpoint iteration (3.23). Here, $\Delta x = \Delta y = 1/200$ and CFL $= 0.9$. The solution has been computed on a grid of double size to avoid the influence of the boundaries. Left: Cell averages of the solution at time $t = 0.3$. Right: Exact solution following [15]

6.6 Multi-dimensional Scalar Equations

As a last test case, consider the multi-dimensional Burgers' equation

$$\partial_t q + \partial_x \left(\frac{q^2}{2}\right) + \partial_y \left(\frac{q^2}{2}\right) = 0 \tag{6.16}$$

and a 4-quadrant Riemann problem setup as follows:

$$q_0 = \begin{cases} -1 & \text{NE} \\ -0.2 & \text{NW} \\ 0.5 & \text{SW} \\ 0.8 & \text{SE} \end{cases} \tag{6.17}$$

Figure 21 shows the solution at $t = 0.3$ using Active Flux (with fixpoint iteration (3.23)) along with the exact solution taken from [15] (p. 4258). No limiting is used.

7 Conclusion and Outlook

The Active Flux scheme is a finite volume scheme with additional pointwise degrees of freedom located at the cell boundary. It involves a continuous reconstruction and thus does not make use of Riemann solvers. Instead, an evolution operator for the pointwise degrees of freedom is required. Once their evolution is obtained, the update of the cell average follows the usual finite volume/Godunov scheme idea; the intercell flux is obtained by evaluating the flux function on the point values along the boundary (and using quadrature). The Active Flux scheme has initially [10,31] been employing exact evolution operators, because the problems onto which the scheme was applied admitted closed form solution operators.

In particular it has been shown in [3] that for linear acoustics the Active Flux method is low Mach number compliant and stationarity preserving without the need for any fix. This makes Active Flux an interesting candidate for a class of methods, which are structure preserving by construction. Inspired by the finding for linear acoustics, this paper serves as a stepping stone towards deriving structure preserving Active Flux methods for multi-dimensional nonlinear systems.

Upon an extension of the Active Flux method to nonlinear problems usage of exact evolution operators cannot be maintained. Approximate evolution operators suggested so far in the literature either were reducing the order of the scheme or involved complicated expressions such as the Lax-Wendroff expansion and subsequent solution of Riemann problems (ADER). This paper shows how approximate evolution operators can be found which are not costly and allow to maintain third order of accuracy. The cases considered here are nonlinear scalar conservation laws in one and two spatial dimensions as well as nonlinear hyperbolic systems of conservation laws in one spatial dimension.

As Active Flux is very different from standard finite volume or Galerkin methods, many aspects need to be reconsidered, and many questions well-studied for other methods were still open. Continuous reconstruction might raise doubts about the applicability of Active Flux to problems involving shock formation. It is found that in certain setups too simple an evolution operator fails to correctly recognize the self-steepening. This then leads to artefacts that resemble entropy glitches encountered with certain finite volume schemes. As a cure, in this paper a simple strategy is presented which allows to take into account the fact that characteristics may cross. This "entropy fix" is found to lead to accurate numerical evolutions without artefacts. By means of numerical examples it is shown that Active Flux, for example, is able to accurately solve Riemann problems for one-dimensional systems of nonlinear conservation laws, such as the Euler equations.

As a numerical method of higher order is prone to overshoots around discontinuities, a limiting procedure needs to be in place. Here, a simple limiting is suggested which modifies the reconstruction whenever an avoidable overshoot/undershoot is recognized. Whereas the Active Flux limiters available in the literature either require joining several polynomials inside the cell or re-introduce discontinuities, the suggested monotone reconstruction is simple to compute and retains a continuous reconstruction.

The correct approximation of the entropy solution and limiting in one spatial dimension may not outperform currently available methods of third and higher order. However, all these are necessary ingredients for an extension to multiple spatial dimensions that so far were open, or at least insufficiently studied for the Active Flux method.

Future work shall be devoted to multi-dimensional hyperbolic systems. The approximate evolution operators presented here shall be extended to multiple spatial dimensions and thus combined with the favorable properties of Active Flux in multiple dimensions. This hopefully will pave the way towards a powerful structure-preserving method for multi-dimensional systems of conservation laws.

Funding Open Access funding provided by University of Zurich

References

1. Barsukow, W.: Stationarity preserving schemes for multi-dimensional linear systems. Math. Comput. **88**(318), 1621–1645 (2019)
2. Brooks, A.N., Hughes, T.J.R.: Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. Comput. Methods Appl. Mech. Eng. **32**(1–3), 199–259 (1982)
3. Barsukow, W., Hohm, J., Klingenberg, C., Roe, P.L.: The active flux scheme on Cartesian grids and its low Mach number limit. J. Sci. Comput. **81**(1), 594–622 (2019)
4. Barsukow, W., Klingenberg, C.: Exact solution and a truly multidimensional Godunov scheme for the acoustic equations (2020). Submitted, preprint available as arXiv:2004.04217
5. Cockburn, B., Lin, S.-Y., Shu, C.-W.: TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems. J. Comput. Phys. **84**(1), 90–113 (1989)
6. Cockburn, B., Shu, C.-W.: The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. J. Comput. Phys. **141**(2), 199–224 (1998)
7. Colella, P., Woodward, P.R.: The piecewise parabolic method (PPM) for gas-dynamical simulations. J. Comput. Phys. **54**(1), 174–201 (1984)
8. Eymann, T.A., Roe, P.L.: Active flux schemes. In: 49th AIAA Aerospace Science Meeting (2011)
9. Eymann, T.A., Roe, P.L.: Active flux schemes for systems. In: 20th AIAA Computational Fluid Dynamics Conference (2011)
10. Eymann, T.A., Roe, P.L.: Multidimensional active flux schemes. In: *21st AIAA Computational Fluid Dynamics Conference* (2013)
11. Eymann, T.A.: Active flux schemes. Ph.D. thesis, University of Michigan, Dissertation (2013)
12. Fan, D.: On the acoustic component of active flux schemes for nonlinear hyperbolic conservation laws. Ph.D. thesis, University of Michigan, Dissertation (2017)
13. Guillard, H., Murrone, A.: On the behavior of upwind schemes in the low Mach number limit: II. Godunov type schemes. Comput. Fluids **33**(4), 655–675 (2004)
14. Godunov, S.K.: A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. Matematicheskii Sbornik **89**(3), 271–306 (1959)
15. Guermond, J.-L., Pasquetti, R., Popov, B.: Entropy viscosity method for nonlinear conservation laws. J. Comput. Phys. **230**(11), 4248–4267 (2011)
16. Helzel, C., Kerkmann, D., Scandurra, L.: A new ADER method inspired by the active flux method. J. Sci. Comput. **80**(3), 1463–1497 (2019)
17. Harten, A., Lax, P.D., van Leer, B.: On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. SIAM Rev. **25**(1), 35–61 (1983)
18. Jin, S., Xin, Z.: The relaxation schemes for systems of conservation laws in arbitrary space dimensions. Commun. Pure Appl. Math. **48**(3), 235–276 (1995)
19. Lax, P.D.: Weak solutions of nonlinear hyperbolic equations and their numerical computation. Commun. Pure Appl. Math. **7**(1), 159–193 (1954)
20. LeVeque, R.: Finite Volume Methods for Hyperbolic Problems, vol. 31. Cambridge University Press, Cambridge (2002)
21. Maeng, J.: On the advective component of active flux schemes for nonlinear hyperbolic conservation laws. Ph.D. thesis, University of Michigan, Dissertation (2017)
22. Morton, K.W., Roe, P.L.: Vorticity-preserving Lax-Wendroff-type schemes for the system wave equation. SIAM J. Sci. Comput. **23**(1), 170–192 (2001)
23. Roe, P.L., Lung, T., Maeng, J.: New approaches to limiting. In: 22nd AIAA Computational Fluid Dynamics Conference, p. 2913 (2015)
24. Roe, P.L.: Approximate Riemann solvers, parameter vectors, and difference schemes. J. Comput. Phys. **43**(2), 357–372 (1981)
25. Roe, P.: Is discontinuous reconstruction really a good idea? J. Sci. Comput. **73**(2–3), 1094–1114 (2017)
26. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. In: Upwind and High-Resolution Schemes, pp. 328–374. Springer (1989)
27. Sod, G.A.: A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. J. Comput. Phys. **27**(1), 1–31 (1978)
28. Tadmor, E.: Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems. Acta Numer. **12**, 451–512 (2003)
29. Toro, E.F.: Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction. Springer, Berlin (2009)

30. Titarev, V.A., Toro, E.F.: ADER: arbitrary high order Godunov approach. J. Sci. Comput. **17**(1–4), 609–618 (2002)
31. Van Leer, B.: Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. J. Comput. Phys. **23**(3), 276–299 (1977)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.