Journal of Computational Physics 520 (2025) 113471

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

On the robustness of high-order upwind summation-by-parts methods for nonlinear conservation laws

Hendrik Ranocha [a,*], Andrew R. Winters [b], Michael Schlottke-Lakemper [c,d,e], Philipp Öffner [a,f], Jan Glaubitz [g], Gregor J. Gassner [h,i]

[a] *Institute of Mathematics, Johannes Gutenberg University Mainz, Staudingerweg 9, 55128 Mainz, Germany*
[b] *Department of Mathematics; Applied Mathematics, Linköping University, Sweden*
[c] *High-Performance Scientific Computing, University of Augsburg, Germany*
[d] *Applied and Computational Mathematics, RWTH Aachen University, Germany*
[e] *High-Performance Computing Center Stuttgart (HLRS), University of Stuttgart, Germany*
[f] *Institute of Mathematics, TU Clausthal, Germany*
[g] *Aeronautics and Astronautics, Massachusetts Institute of Technology, USA*
[h] *Department of Mathematics and Computer Science, University of Cologne, Germany*
[i] *Center for Data and Simulation Science, University of Cologne, Germany*

ARTICLE INFO

MSC:
65M06
65M20
65M70

Keywords:
Summation-by-parts operators
Conservation laws
Finite difference methods
Discontinuous Galerkin methods
Flux vector splitting

ABSTRACT

We use the framework of upwind summation-by-parts (SBP) operators developed by Mattsson (2017, doi:10.1016/j.jcp.2017.01.042) and study different flux vector splittings in this context. To do so, we introduce discontinuous-Galerkin-like interface terms for multi-block upwind SBP methods applied to nonlinear conservation laws. We investigate the behavior of the upwind SBP methods for flux vector splittings of varying complexity on Cartesian as well as unstructured curvilinear multi-block meshes. Moreover, we analyze the local linear/energy stability of these methods following Gassner, Svärd, and Hindenlang (2022, doi:10.1007/s10915-021-01720-8). Finally, we investigate the robustness of upwind SBP methods for challenging examples of shock-free flows of the compressible Euler equations such as a Kelvin-Helmholtz instability and the inviscid Taylor-Green vortex.

1. Introduction

Stability and robustness are crucial properties of numerical methods for conservation laws to obtain reliable simulations, in particular for under-resolved flows. At the same time, high-order methods can be very efficient and fit well to modern hardware. However, it is non-trivial to ensure robustness of high-order methods without destroying their high-order accuracy.

Over the last decade, entropy-based methods have emerged as a popular choice to construct robust high-order methods in a wide range of applications. Built from the seminal work of Tadmor [1,2], high-order extensions have been developed in [3,4]. These flux differencing schemes work well for under-resolved flows, e.g., [5–9]. However, some doubts have been raised recently within the high-order community by Gassner et al. [10]. In their article, the authors demonstrated critical failures of high-order entropy-dissipative methods for a conceptually simple setup of the 1D compressible Euler equations; with constant velocity and pressure,

[*] Corresponding author.
E-mail address: hendrik.ranocha@uni-mainz.de (H. Ranocha).

https://doi.org/10.1016/j.jcp.2024.113471
Received 26 March 2024; Received in revised form 18 September 2024; Accepted 28 September 2024

Available online 3 October 2024

these equations reduce to simple linear advection of the density. Central schemes without any entropy properties perform well in this case but crash for demanding simulations of under-resolved flows such as the inviscid Taylor-Green vortex. In contrast, entropy-stable flux differencing methods work well for the Taylor-Green vortex but fail for the apparently simple advection example.

Failures due to positivity issues can be fixed by adding invariant domain preserving techniques, e.g., [11–14]. However, it is desirable to combine such shock-capturing and invariant domain preserving approaches with a good baseline scheme such that the amount of additional dissipation can be kept low [15]. Thus, we are interested in high-order baseline schemes that come already with some built-in dissipation everywhere, not only at element interfaces as typical in discontinuous Galerkin (DG) methods. At the same time, we would like to avoid having additional parameters in the schemes that need to be tuned manually.

Many high-order methods with some provable stability properties can be obtained in the general framework of summation-by-parts (SBP) operators. SBP operators were originally developed for finite difference methods [16,17]. They are the basis of entropy-stable flux differencing methods by mimicking integration by parts discretely. Many common numerical methods can be formulated using SBP operators, e.g., finite volume methods [18,19], continuous Galerkin methods [20–23], DG methods [24–26], and flux reconstruction methods [27,28]. Further information and background material on SBP operators is collected in the review articles [29,30].

Classical SBP operators can be used to design numerical schemes that are provably stable. Typically, SBP methods are based on central-type discretizations in the interior and weak imposition of boundary data using simultaneous approximation terms (SATs) [31,32] that introduce some dissipation. In a multi-block finite difference or DG setting, such SATs are also used to couple the blocks/elements weakly and introduce additional dissipation — but only at interfaces, not in the interior of the block/elements. To obtain additional dissipation everywhere, artificial dissipation operators can be used [33]. These operators can be combined with a user-chosen amount of dissipation and it may be non-trivial to choose an appropriate amount of dissipation.

Combining classical SBP operators and artificial dissipation can be interpreted as upwinding [34,35]. Mattsson [36] introduced a general definition of upwind SBP operators and constructed a range of schemes with good numerical properties, resulting in a parameter-free combination of central-type SBP operators and artificial dissipation. These upwind SBP operators have been used successfully for a range of applications such as the shallow water equations [37], atmospheric flows [38], and scalar conservation laws [39]. They have also been extended to staggered grids in [40]. Their relations to DG methods have been discussed in [41,42].

To apply upwind SBP operators to nonlinear conservation laws, a flux vector splitting is required [36]. Across the literature [34–39] the numerical testing is predominantly done with Lax-Friedrichs type splittings. For many numerical schemes, such Lax-Friedrichs type splittings are not ideal. As stated by Stiernström et al. [39, Remark 4.2], many other flux vector splittings are available but have not been studied in detail with upwind SBP operators so far. One of the goals of this article is to fill this gap and investigate the impact of different flux vector splittings on robustness for challenging examples on Cartesian and curvilinear meshes.

To do so, we first review upwind SBP operators [36] and classical flux vector splittings [43, Chapter 8] in Section 2. These flux vector splitting methods have been widely developed and used in the last century [44–49] but were abandoned in favor of other techniques due to their significant amount of numerical dissipation [50]. We will see that the combination of flux vector splitting techniques with high-order difference operators does not lead to an excessive amount of artificial dissipation.

Next, we formulate high-order upwind SBP methods for nonlinear problems in Section 3 based on the seminal works of Mattsson and collaborators [34–39]. To enable an investigation across a range of different flux vector splittings in multi-block finite difference methods, we need to introduce appropriate SATs. To do so, we start with a classical upwind SBP formulation and introduce interface terms as in DG methods — using numerical fluxes resulting from the flux vector splitting. We then discuss the relation of this formulation to the construction of global upwind SBP operators as done in [41].

In the final part of Section 3, we consider the upwind SBP methods on unstructured curvilinear multi-block meshes. The formulation in generalized coordinates reveals a subtle interplay between the finite difference operator and the particular flux vector splitting. Moreover, we demonstrate that these subtleties are not an issue for Lax-Friedrichs type splittings; however, they are present for more sophisticated splitting techniques.

Afterwards, we follow Gassner, Svärd, and Hindenlang [10] and analyze the local linear/energy stability properties of upwind SBP methods in Section 4. In particular, we prove local linear/energy stability for Burgers' equation in the setting where Gassner et al. [10] observed stability issues for entropy-stable methods based on classical SBP operators.

In Section 5, we investigate the behavior of upwind SBP methods with different flux vector splittings numerically. We begin with 1D convergence tests, verify the local linear/energy stability results, and then proceed to 2D and 3D simulations of under-resolved flows on Cartesian meshes. In particular, we consider shock-free setups for the compressible Euler equations and study the robustness for two challenging setups: a Kelvin-Helmholtz instability and the inviscid Taylor-Green vortex. We further study the convergence and free-stream preservation properties on unstructured curvilinear meshes with different flux vector splittings. Finally, we summarize our findings and provide an outlook on further research in Section 6.

2. Review of upwind SBP operators and flux vector splitting

Consider a hyperbolic conservation law

$$\partial_t u(t,x) + \partial_x f\Big(u(t,x)\Big) = 0, \qquad t \in (0,T), x \in (x_{\min}, x_{\max}), \tag{2.1}$$

with conserved variable u and flux f in one space dimension, equipped with appropriate initial and boundary conditions. For now, we concentrate on the 1D setting to describe the overall methodologies. Extension of the method to multiple space dimensions is done using a tensor product structure. We delay a detailed discussion of the continuous and discrete formulations in generalized curvilinear coordinates to Section 3.4.

In this section, we review classical flux vector splitting techniques, the basic idea of upwind SBP methods, and collect some useful properties of upwind SBP operators for reference. All these concepts and results are well-known in the literature, but we collect them here to make the article self-contained.

2.1. Flux vector splitting

The classical flux vector splitting approach [43, Chapter 8] to create (semi-)discretizations of the conservation law (2.1) begins with an appropriate splitting of the flux f such that

$$f(u) = f^-(u) + f^+(u), \tag{2.2}$$

where the eigenvalues $\lambda_i^\pm$ of the Jacobians $J^\pm = \partial_u f^\pm$ satisfy

$$\forall i: \quad \lambda_i^- \le 0,\ \lambda_i^+ \ge 0. \tag{2.3}$$

There is a great deal of freedom in the construction of a flux vector splitting (2.2) to create an upwind scheme. The design of $f^-(u)$ and $f^+(u)$ typically relies on the mathematically sound characteristic theory for hyperbolic partial differential equations. Depending on how one treats the different characteristics, for instance separating the convective and pressure components of the compressible Euler equations, leads to a wide variety of flux vector splittings, e.g., [45–47,51,52]. Because the flux vector splitting separates the upwind directions with which solution information propagates, the resulting scheme does not require the (approximate) solution of a Riemann problem. This makes flux vector splitting based algorithms particularly attractive due to their simplicity and ability to approximate shock waves. To demonstrate this simplicity, consider a classical first-order finite volume method of the form

$$\partial_t \boldsymbol{u}_i + \frac{1}{\Delta x}\left(f^{\mathrm{num}}(\boldsymbol{u}_i, \boldsymbol{u}_{i+1}) - f^{\mathrm{num}}(\boldsymbol{u}_{i-1}, \boldsymbol{u}_i)\right) = 0 \tag{2.4}$$

with numerical flux f^{num}. In the flux vector splitting approach, the numerical flux is chosen as

$$f^{\mathrm{num}}(u_l, u_r) = f^+(u_l) + f^-(u_r). \tag{2.5}$$

Thus, the chosen splitting determines the scheme completely.

Example 2.1. The global Lax-Friedrichs splitting requires a global upper bound λ on the possible wave speeds and uses

$$f^\pm(u) = \frac{1}{2}\left(f(u) \pm \lambda u\right). \tag{2.6}$$

This results in the numerical flux

$$f^{\mathrm{num}}(u_l, u_r) = f^+(u_l) + f^-(u_r) = \frac{1}{2}\left(f(u_l) + f(u_r)\right) - \frac{\lambda}{2}(u_r - u_l). \tag{2.7}$$

This splitting has predominantly been used in previous works on upwind SBP operators, e.g., [36,37,39]. ◁

Next, we present some examples for the 1D compressible Euler equations

$$\partial_t \begin{pmatrix} \varrho \\ \varrho v \\ \varrho e \end{pmatrix} + \partial_x \begin{pmatrix} \varrho v \\ \varrho v^2 + p \\ (\varrho e + p) v \end{pmatrix} = 0 \tag{2.8}$$

of an ideal gas with density ϱ, velocity v, total energy density ϱe, and pressure

$$p = (\gamma - 1)\left(\varrho e - \frac{1}{2}\varrho v^2\right), \tag{2.9}$$

where the ratio of specific heats is usually chosen as $\gamma = 1.4$. To the best of our knowledge, the splittings described in the following examples have not been combined with upwind SBP operators in the existing literature [34–39] or only in less detail.

Example 2.2. To describe the Steger-Warming splitting [44], we use the standard notation

$$\lambda_i^\pm = \frac{\lambda_i \pm |\lambda_i|}{2} \tag{2.10}$$

for the positive/negative part of an eigenvalue λ_i. The wave speeds of the 1D Euler equations are

$$\lambda_1 = v - a, \quad \lambda_2 = v, \quad \lambda_3 = v + a, \tag{2.11}$$

where the speed of sound is $a = \sqrt{\gamma p / \varrho}$. Then, the flux splitting of Steger and Warming is given by

$$f^\pm = \frac{\varrho}{2\gamma}\begin{pmatrix} \lambda_1^\pm + 2(\gamma-1)\lambda_2^\pm + \lambda_3^\pm \\ (v-a)\lambda_1^\pm + 2(\gamma-1)v\lambda_2^\pm + (v-a)\lambda_3^\pm \\ (H-va)\lambda_1^\pm + (\gamma-1)v^2\lambda_2^\pm + (H+va)\lambda_3^\pm \end{pmatrix}, \tag{2.12}$$

where $H = (\varrho e + p)/\varrho = v^2/2 + a^2/(\gamma-1)$ is the enthalpy, see also [43, Section 8.4.2]. ◁

Example 2.3. Next, we describe the van Leer-Hänel splitting [45–47] based on a splitting of van Leer with a modification of the energy flux proposed by Hänel et al. and the "p4" splitting of the pressure proposed by Liou and Steffen. First, we introduce the signed Mach number $M = v/a$ and the pressure splitting

$$p^\pm = \frac{1 \pm \gamma M}{2} p. \tag{2.13}$$

The fluxes are given by

$$f^\pm = \pm\frac{\varrho a (M \pm 1)^2}{4}\begin{pmatrix} 1 \\ v \\ H \end{pmatrix} + \begin{pmatrix} 0 \\ p^\pm \\ 0 \end{pmatrix}, \tag{2.14}$$

where $H = (\varrho e + p)/\varrho = v^2/2 + a^2/(\gamma-1)$ is again the enthalpy. ◁

2.2. Upwind SBP operators

In this article, we focus on a collocation setting as in classical finite difference methods. Thus, we consider a grid $\boldsymbol{x} = (\boldsymbol{x}_i)_{i=1}^N$ with nodes $\boldsymbol{x}_i$ and use pointwise approximations such as $\boldsymbol{u}_i = u(\boldsymbol{x}_i)$ and $\boldsymbol{1} = (1, \dots, 1)^T$. We also assume that the grid includes the boundary nodes of the domain, i.e.,

$$\boldsymbol{x}_1 = x_{\min}, \qquad \boldsymbol{x}_N = x_{\max}. \tag{2.15}$$

Then, classical SBP operators are constructed to mimic integration-by-parts, cf. [29,30].

Definition 2.4. An *SBP operator* on the domain $[x_{\min}, x_{\max}]$ consists of a grid $\boldsymbol{x}$, a symmetric and positive definite mass/norm matrix M satisfying $\boldsymbol{1}^T M \boldsymbol{1} = x_{\max} - x_{\min}$, and a consistent derivative operator D such that

$$M D + D^T M = \boldsymbol{t}_R \boldsymbol{t}_R^T - \boldsymbol{t}_L \boldsymbol{t}_L^T, \tag{2.16}$$

where $\boldsymbol{t}_R^T = (0, \dots, 0, 1)$ and $\boldsymbol{t}_L = (1, 0, \dots, 0)^T$. It is called diagonal-norm operator if M is diagonal. ◁

We often identify an SBP operator with the derivative operator D and assume that the remaining parts are clear from the context. Since the boundary nodes are included, (2.16) guarantees that the discrete operators mimic integration-by-parts as

$$\begin{array}{ccc}
\underbrace{\boldsymbol{u}^T M D \boldsymbol{v} + \boldsymbol{u}^T D^T M \boldsymbol{v}} & = & \underbrace{\boldsymbol{u}^T \boldsymbol{t}_R \boldsymbol{t}_R^T \boldsymbol{v} - \boldsymbol{u}^T \boldsymbol{t}_L \boldsymbol{t}_L^T \boldsymbol{v}}, \\
\wr & & \wr \\
\overbrace{\int_{x_{\min}}^{x_{\max}} u\,(\partial_x v) + \int_{x_{\min}}^{x_{\max}} (\partial_x u)\, v} & = & \overbrace{u(x_{\max})v(x_{\max}) - u(x_{\min})v(x_{\min})}.
\end{array} \tag{2.17}$$

Upwind SBP operators were introduced by Mattsson [36]. The basic idea is to introduce two derivative operators $D_\pm$ that mimic integration-by-parts together and are compatible in the sense that their difference is negative semidefinite, which allows to introduce artificial dissipation.

Definition 2.5. An *upwind SBP operator* on the domain $[x_{\min}, x_{\max}]$ consists of a grid $\boldsymbol{x}$, a symmetric and positive definite mass/norm matrix M satisfying $\boldsymbol{1}^T M \boldsymbol{1} = x_{\max} - x_{\min}$, and two consistent derivative operators $D_\pm$ such that

$$M D_+ + D_-^T M = \boldsymbol{t}_R \boldsymbol{t}_R^T - \boldsymbol{t}_L \boldsymbol{t}_L^T, \qquad M(D_+ - D_-) \text{ is negative semidefinite}, \tag{2.18}$$

where again $\boldsymbol{t}_R^T = (0, \dots, 0, 1)$ and $\boldsymbol{t}_L = (1, 0, \dots, 0)^T$. It is called diagonal-norm operator if M is diagonal. ◁

For convenience, we also identify an upwind SBP operator simply with the derivative matrices $D_\pm$. In matrix form, the upwind SBP operators derived by Mattsson [36] are constructed such that D_+ is biased toward the upper-triangular part, i.e., it has more non-zero entries above the diagonal. Similarly, D_- is biased toward the lower-triangular part.

Example 2.6. The second-order accurate upwind operators of [36] are given by

$$D_+ = \frac{1}{\Delta x}\begin{pmatrix} -3 & 5 & -2 & & & & \\ -1/5 & -1 & 8/5 & -2/5 & & & \\ & & -3/2 & 2 & -1/2 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 3/2 & 2 & -1/2 \\ & & & & & -1 & 1 \\ & & & & & -1 & 1 \end{pmatrix}, \tag{2.19}$$

$$D_- = \frac{1}{\Delta x}\begin{pmatrix} -1 & 1 & & & & & & \\ -1 & 1 & & & & & & \\ 1/2 & -2 & 3/2 & & & & & \\ & 1/2 & -2 & 3/2 & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & 1/2 & -2 & 3/2 & & \\ & & & & 2/5 & -8/5 & 1 & 1/5 \\ & & & & & 2 & -5 & 3 \end{pmatrix}, \tag{2.20}$$

and $M = \Delta x \operatorname{diag}(1/4, 5/4, 1, \dots, 1, 5/4, 1/4)$. ◁

Upwind SBP operators are constructed to create provably stable semidiscretizations of linear transport problems as already described in [36]. For completeness and as an example, consider the linear advection equation

$$\begin{aligned} \partial_t u(t,x) + \partial_x u(t,x) = 0, &\qquad t \in (0,T), x \in (x_{\min}, x_{\max}), \\ u(0,x) = u^0(x), &\qquad x \in [x_{\min}, x_{\max}], \\ u(t, x_{\min}) = g_L(t), &\qquad t \in [0,T]. \end{aligned} \tag{2.21}$$

Since the transport happens from left to right, we choose the left-biased upwind operator D_- such that solution information from the correct characteristic direction is used and obtain a stable semidiscretization

$$\partial_t \boldsymbol{u} + D_- \boldsymbol{u} = M^{-1} \boldsymbol{t}_L (g_L - \boldsymbol{t}_L^T \boldsymbol{u}). \tag{2.22}$$

This semidiscretization is globally conservative, since

$$\begin{aligned} \partial_t(\boldsymbol{1}^T M \boldsymbol{u}) &= \boldsymbol{1}^T M \partial_t \boldsymbol{u} = -\boldsymbol{1}^T M D_- \boldsymbol{u} + \boldsymbol{1}^T \boldsymbol{t}_L (g_L - \boldsymbol{t}_L^T \boldsymbol{u}) \\ &= \boldsymbol{1}^T D_+^T M \boldsymbol{u} - \boldsymbol{1}^T (\boldsymbol{t}_R \boldsymbol{t}_R^T - \boldsymbol{t}_L \boldsymbol{t}_L^T) \boldsymbol{u} + \boldsymbol{1}^T \boldsymbol{t}_L (g_L - \boldsymbol{t}_L^T \boldsymbol{u}) = g_L - \boldsymbol{t}_R^T \boldsymbol{u}, \end{aligned} \tag{2.23}$$

where we have used the upwind SBP property (2.18) and consistency of the derivative operator. Furthermore, the semidiscretization (2.22) is energy-stable since

$$\begin{aligned} \partial_t \|\boldsymbol{u}\|_M^2 &= 2\boldsymbol{u}^T M \partial_t \boldsymbol{u} = -2\boldsymbol{u}^T M D_- \boldsymbol{u} + 2\boldsymbol{u}^T \boldsymbol{t}_L (g_L - \boldsymbol{t}_L^T \boldsymbol{u}) \\ &= -\boldsymbol{u}^T M D_- \boldsymbol{u} + \boldsymbol{u}^T D_+^T M \boldsymbol{u} - \boldsymbol{u}^T (\boldsymbol{t}_R \boldsymbol{t}_R^T - \boldsymbol{t}_L \boldsymbol{t}_L^T) \boldsymbol{u} + 2\boldsymbol{u}^T \boldsymbol{t}_L (g_L - \boldsymbol{t}_L^T \boldsymbol{u}) \\ &\le 2(\boldsymbol{t}_L^T \boldsymbol{u}) g_L - (\boldsymbol{t}_L^T \boldsymbol{u})^2 - (\boldsymbol{t}_R^T \boldsymbol{u})^2 = g_L^2 - (\boldsymbol{t}_R^T \boldsymbol{u})^2 - (g_L - \boldsymbol{t}_L^T \boldsymbol{u})^2, \end{aligned} \tag{2.24}$$

mimicking the estimate

$$\partial_t \|u(t)\|_{L^2}^2 = g_L(t)^2 - u(t, x_{\max})^2 \tag{2.25}$$

up to additional artificial dissipation due to the upwind operators and the weak imposition of boundary data.

2.3. Some useful properties of upwind SBP operators

As described in [36], upwind SBP operators can be interpreted as classical SBP operators plus artificial dissipation in the context of the linear advection equation. Indeed,

$$\begin{aligned} D_+ &= \frac{1}{2}(D_- + D_+) + \frac{1}{2}(D_+ - D_-), \\ D_- &= \frac{1}{2}(D_- + D_+) - \frac{1}{2}(D_+ - D_-). \end{aligned} \tag{2.26}$$

The average of the upwind operators is a classical SBP operator since [36]

$$M(D_- + D_+) + (D_- + D_+)^T M = (M D_+ + D_-^T M) + (M D_- + D_+^T M) = 2(\boldsymbol{t}_R \boldsymbol{t}_R^T - \boldsymbol{t}_L \boldsymbol{t}_L^T). \tag{2.27}$$

The difference of the upwind SBP operators introduces artificial dissipation for the linear advection equation when multiplied by the mass matrix M by construction. Thus, the upwind SBP discretization (2.22) can be written as [36]

$$\partial_t \boldsymbol{u} + \underbrace{\frac{D_- + D_+}{2}\boldsymbol{u}}_{\text{central}} - \underbrace{\frac{D_+ - D_-}{2}\boldsymbol{u}}_{\text{dissipation}} = M^{-1}\boldsymbol{t}_L(g_L - \boldsymbol{t}_L^T\boldsymbol{u}). \tag{2.28}$$

This is the form of a central SBP discretization plus artificial dissipation for linear advection. For general nonlinear problems, we can still use the negative semidefiniteness of the difference of the operators to introduce artificial dissipation, but we need proper upwinding as discussed in Section 3.

2.4. Upwind SBP operators in periodic domains

In periodic domains, we require boundary terms to vanish, resulting in the following definitions [41].

Definition 2.7. A *periodic SBP operator* on the domain $[x_{\min}, x_{\max}]$ consists of a grid $\boldsymbol{x}$, a symmetric and positive definite mass/norm matrix M satisfying $\boldsymbol{1}^T M \boldsymbol{1} = x_{\max} - x_{\min}$, and a consistent derivative operator D such that

$$M D + D^T M = 0. \tag{2.29}$$

It is called diagonal-norm operator if M is diagonal. ◁

Definition 2.8. A *periodic upwind SBP operator* on the domain $[x_{\min}, x_{\max}]$ consists of a grid $\boldsymbol{x}$, a symmetric and positive definite mass/norm matrix M satisfying $\boldsymbol{1}^T M \boldsymbol{1} = x_{\max} - x_{\min}$, and two consistent derivative operators $D_\pm$ such that

$$M D_+ + D_-^T M = 0, \qquad M(D_+ - D_-) \text{ is negative semidefinite.} \tag{2.30}$$

It is called diagonal-norm operator if M is diagonal. ◁

An upwind SBP discretization of the linear advection equation

$$\begin{aligned} \partial_t u(t,x) + \partial_x u(t,x) &= 0, \qquad && t \in (0,T), x \in (x_{\min}, x_{\max}), \\ u(0,x) &= u^0(x), && x \in [x_{\min}, x_{\max}], \end{aligned} \tag{2.31}$$

with periodic boundary conditions is

$$\partial_t \boldsymbol{u} + D_- \boldsymbol{u} = \boldsymbol{0}. \tag{2.32}$$

Following the same steps as in the case of a bounded domain, we see that it is conservative, i.e.,

$$\partial_t(\boldsymbol{1}^T M \boldsymbol{u}) = 0, \tag{2.33}$$

and energy-stable, i.e.,

$$\partial_t \|\boldsymbol{u}\|_M^2 \le 0. \tag{2.34}$$

Example 2.9. The interior stencils of the second-order accurate upwind operators of [36] shown in Example 2.6 yield periodic upwind operators. Specifically, we have

$$D_+ = \frac{1}{\Delta x}\begin{pmatrix} -3/2 & 2 & -1/2 & & & & \\ & -3/2 & 2 & -1/2 & & & \\ & & -3/2 & 2 & -1/2 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 3/2 & 2 & -1/2 \\ -1/2 & & & & & -3/2 & 2 \\ 2 & -1/2 & & & & & -3/2 \end{pmatrix}, \tag{2.35}$$

$$D_- = \frac{1}{\Delta x}\begin{pmatrix} 3/2 & & & & & & 1/2 & -2 \\ -2 & 3/2 & & & & & & 1/2 \\ 1/2 & -2 & 3/2 & & & & & \\ & 1/2 & -2 & 3/2 & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & 1/2 & -2 & 3/2 & & \\ & & & & 1/2 & -2 & 3/2 & \\ & & & & & 1/2 & -2 & 3/2 \end{pmatrix}, \tag{2.36}$$

and $M = \Delta x \operatorname{diag}(1, \dots, 1)$. ◁

3. Formulation of upwind SBP methods for nonlinear problems

Following earlier work on upwind SBP operators [36], we first apply the flux vector splitting (2.2) and rewrite the hyperbolic conservation law in one space dimension (2.1) as

$$\partial_t u + \partial_x f^-(u) + \partial_x f^+(u) = 0. \tag{3.1}$$

Next, we discretize the conservation law in space by using upwind SBP operators as

$$\partial_t \boldsymbol{u} + D_+ \boldsymbol{f}^- + D_- \boldsymbol{f}^+ = \boldsymbol{0}. \tag{3.2}$$

These formulations are well-known in the literature, e.g., [34–37,39]. To couple multiple blocks of upwind SBP operators, we introduce interface terms as in discontinuous Galerkin methods in the following. On each element, we will use the semidiscretzation (3.2) as a baseline and add additional terms to couple the elements weakly at the interfaces. Such a construction has been used for central-type SBP operators in several works, e.g., [24].

After describing and contextualizing the method in one space dimension, we describe and analyze the method in two-dimensional curvilinear coordinates in Section 3.4.

Remark 3.1. The indices ± of the upwind operators and the fluxes do not match. This is due to historical reasons since we want to keep backwards compatibility with both the notation of flux vector splitting methods [43, Chapter 8] and upwind SBP operators as introduced in [36]. ◁

3.1. Local upwind SBP formulation with SATs and numerical fluxes

On each element, we consider a discretization of the form

$$\partial_t \boldsymbol{u} + D_+ \boldsymbol{f}^- + D_- \boldsymbol{f}^+ = \mathbf{SAT} \tag{3.3}$$

with a simultaneous approximation term **SAT**. To motivate the construction of the SAT, we consider the upwind SBP discretization (3.3) with the global Lax-Friedrichs flux vector splitting, resulting in

$$\partial_t \boldsymbol{u} + \frac{1}{2} D_+ (\boldsymbol{f} + \lambda \boldsymbol{u}) + \frac{1}{2} D_- (\boldsymbol{f} - \lambda \boldsymbol{u}) = \mathbf{SAT}. \tag{3.4}$$

We can formulate this in the central SBP plus dissipation form as

$$\partial_t \boldsymbol{u} + \frac{1}{2} (D_- + D_+) \boldsymbol{f} + \frac{\lambda}{2} (D_+ - D_-) \boldsymbol{u} = \mathbf{SAT}. \tag{3.5}$$

The second term is the central SBP discretization and the third term is the artificial dissipation term built into the upwind operators. Thus, we select the standard SAT for a central SBP discretization, i.e.,

$$\mathbf{SAT} = -M^{-1} \boldsymbol{t}_R (f^{\mathrm{num}}_R - \boldsymbol{t}_R^T \boldsymbol{f}) + M^{-1} \boldsymbol{t}_L (f^{\mathrm{num}}_L - \boldsymbol{t}_L^T \boldsymbol{f}), \tag{3.6}$$

where $f^{\mathrm{num}}_{L/R}$ is the numerical flux at the left/right interface of the element. To clarify this notation, we use an upper index to denote the element. Thus, $\boldsymbol{u}^k$ is the numerical solution in element k. Then, the SAT in element k becomes

$$\mathbf{SAT}^k = -M^{-1} \boldsymbol{t}_R \left(f^{\mathrm{num}}(\boldsymbol{u}^k_R, \boldsymbol{u}^{k+1}_L) - \boldsymbol{f}^k_R \right) + M^{-1} \boldsymbol{t}_L \left(f^{\mathrm{num}}(\boldsymbol{u}^{k-1}_R, \boldsymbol{u}^k_L) - \boldsymbol{f}^k_L \right), \tag{3.7}$$

where we have abbreviated the left/right interface value as $\boldsymbol{u}_{L/R} = \boldsymbol{t}_{L/R}^T \boldsymbol{u}$.

There are many classical numerical fluxes that we can use for f^{num}. Next, we use the flux vector splitting to design the numerical fluxes. We demonstrate this procedure first for the right interface. Using the same splitting for the numerical flux and the physical flux yields

$$\begin{aligned} f^{\mathrm{num}}(\boldsymbol{u}^k_R, \boldsymbol{u}^{k+1}_L) - \boldsymbol{f}^k_R &= \left(f^+(\boldsymbol{u}^k_R) + f^-(\boldsymbol{u}^{k+1}_L) \right) - \left(f^+(\boldsymbol{u}^k_R) + f^-(\boldsymbol{u}^k_R) \right) \\ &= f^-(\boldsymbol{u}^{k+1}_L) - f^-(\boldsymbol{u}^k_R). \end{aligned} \tag{3.8}$$

Similarly, we get

$$\begin{aligned} f^{\mathrm{num}}(\boldsymbol{u}^{k-1}_R, \boldsymbol{u}^k_L) - \boldsymbol{f}^k_L &= \left(f^+(\boldsymbol{u}^{k-1}_R) + f^-(\boldsymbol{u}^k_L) \right) - \left(f^+(\boldsymbol{u}^k_L) + f^-(\boldsymbol{u}^k_L) \right) \\ &= f^+(\boldsymbol{u}^{k-1}_R) - f^+(\boldsymbol{u}^k_L), \end{aligned} \tag{3.9}$$

for the left interface. Thus, the SAT becomes

$$\mathbf{SAT}^k = -M^{-1}\boldsymbol{t}_R\left(f^-(\boldsymbol{u}_L^{k+1}) - f^-(\boldsymbol{u}_R^k)\right) + M^{-1}\boldsymbol{t}_L\left(f^+(\boldsymbol{u}_R^{k-1}) - f^+(\boldsymbol{u}_L^k)\right). \tag{3.10}$$

To sum up, we arrive at the upwind SBP discretization

$$\partial_t \boldsymbol{u} + D_+\boldsymbol{f}^- + D_-\boldsymbol{f}^+ = \mathbf{SAT}^k, \tag{3.11}$$

where the simultaneous approximation term can be expressed using (general) numerical fluxes as

$$\mathbf{SAT}^k = -M^{-1}\boldsymbol{t}_R\left(f^{\mathrm{num}}(\boldsymbol{u}_R^k,\boldsymbol{u}_L^{k+1}) - \boldsymbol{f}_R^k\right) + M^{-1}\boldsymbol{t}_L\left(f^{\mathrm{num}}(\boldsymbol{u}_R^{k-1},\boldsymbol{u}_L^k) - \boldsymbol{f}_L^k\right) \tag{3.12}$$

or specifically using the upwind fluxes as

$$\mathbf{SAT}^k = -M^{-1}\boldsymbol{t}_R\left(f^-(\boldsymbol{u}_L^{k+1}) - f^-(\boldsymbol{u}_R^k)\right) + M^{-1}\boldsymbol{t}_L\left(f^+(\boldsymbol{u}_R^{k-1}) - f^+(\boldsymbol{u}_L^k)\right). \tag{3.13}$$

Finally, we can integrate the semidiscretization in time using any suitable time integration scheme, e.g., Runge-Kutta methods.

Remark 3.2. If the corresponding upwind flux is used at interfaces as in (3.13), the final discretization of the hyperbolic conservation law (2.1) is actually agnostic to the flux and does not require the solution to a Riemann problem. All the physics is contained in the particular flux splitting one considers. ◁

3.2. Global upwind SBP formulation

There is another formulation of the method above that is useful as an interpretation. Instead of first introducing an upwind SBP discretization on each element and coupling terms in a second step, we can directly couple the element-local upwind operators to obtain a global upwind operator as described in [41]. Here, we just concentrate on a coupling as in DG methods.

Theorem 3.3 *(Theorem 2.2 of [41]). Consider two upwind SBP operators $D_{\pm,l/r}$ on the grids $\boldsymbol{x}_{l/r}$ with $\boldsymbol{x}_{N_l,l} = \boldsymbol{x}_{1,r}$. Then,*

$$D_+ = \begin{pmatrix} D_{+,l} - M_l^{-1}\boldsymbol{t}_{R,l}\boldsymbol{t}_{R,l}^T & M_l^{-1}\boldsymbol{t}_{R,l}\boldsymbol{t}_{L,r}^T \\ 0 & D_{+,r} \end{pmatrix}, \quad D_- = \begin{pmatrix} D_{-,l} & 0 \\ -M_r^{-1}\boldsymbol{t}_{L,r}\boldsymbol{t}_{R,l}^T & D_{-,r} + M_r^{-1}\boldsymbol{t}_{L,r}\boldsymbol{t}_{L,r}^T \end{pmatrix},$$
$$M = \begin{pmatrix} M_l & 0 \\ 0 & M_r \end{pmatrix}, \tag{3.14}$$

yield upwind SBP operators on the joint grid $\boldsymbol{x} = (\boldsymbol{x}_{1,l},\dots,\boldsymbol{x}_{N_l,l},\boldsymbol{x}_{1,r},\dots,\boldsymbol{x}_{N_r,r})^T$ with $N = N_l + N_r$ nodes. These global operators have the same order of accuracy as the less accurate one of the given local operators.

The global upwind SBP operators described in Theorem 3.3 are obtained by taking upwind numerical fluxes in a DG-type discretization. Indeed, consider the discretization of two elements and their shared interface written using the coupled upwind operators of Theorem 3.3. We have

$$\partial_t\begin{pmatrix}\boldsymbol{u}_l\\ \boldsymbol{u}_r\end{pmatrix} + D_+\begin{pmatrix}\boldsymbol{f}_l^-\\ \boldsymbol{f}_r^-\end{pmatrix} + D_-\begin{pmatrix}\boldsymbol{f}_l^+\\ \boldsymbol{f}_r^+\end{pmatrix} = \mathbf{BTs}, \tag{3.15}$$

where **BTs** collects the surface terms of their non-shared interfaces. For the left element, we get

$$\partial_t\boldsymbol{u}_l + (D_{+,l} - M_l^{-1}\boldsymbol{t}_{R,l}\boldsymbol{t}_{R,l}^T)\boldsymbol{f}_l^- + M_l^{-1}\boldsymbol{t}_{R,l}\boldsymbol{t}_{L,r}^T\boldsymbol{f}_r^- + D_{-,l}\boldsymbol{f}_l^+ = \mathbf{BTs}_l. \tag{3.16}$$

Replacing the index l by the element number k leads to

$$\partial_t\boldsymbol{u}^k + D_+\boldsymbol{f}^- + D_-\boldsymbol{f}^+ = -M^{-1}\boldsymbol{t}_R\left(f^-(\boldsymbol{u}_L^{k+1}) - f^-(\boldsymbol{u}_R^k)\right) + \mathbf{BTs}^k. \tag{3.17}$$

Thus, the interface term is identical to the SAT (3.13) using the upwind numerical flux coming from the flux vector splitting.

In particular, the discontinuous Galerkin spectral element method (DGSEM) with upwind flux for the linear advection equation yields an upwind SBP operator. Indeed, the local operators used on each element with Gauss-Lobatto-Legendre nodes are classical SBP operators [24] and the upwind (Godunov) flux yields exactly the interface coupling described in Theorem 3.3.

Example 3.4. Consider the DGSEM with polynomials of degree $p = 2$ and two elements in the domain $[0,2]$. The corresponding nodes are

$$\boldsymbol{x}_l = (0,1/2,1)^T, \qquad \boldsymbol{x}_r = (1,3/2,2)^T. \tag{3.18}$$

The polynomial derivative matrix D and the mass matrix M on each element with length unity are given by

$$D = \begin{pmatrix} -3 & 4 & -1 \\ -1 & 0 & 1 \\ 1 & -4 & 3 \end{pmatrix}, \qquad M = \begin{pmatrix} 1/6 & & \\ & 2/3 & \\ & & 1/6 \end{pmatrix}. \tag{3.19}$$

These matrices satisfy $M D + D^T M = \operatorname{diag}(-1, 0, 1)$, i.e., the SBP property (2.16). The construction in Theorem 3.3 yields the global operators

$$D_+ = \begin{pmatrix} -3 & 4 & -1 & & & \\ -1 & 0 & 1 & & & \\ 1 & -4 & -3 & 6 & & \\ & & & -3 & 4 & -1 \\ & & & -1 & 0 & 1 \\ & & & 1 & -4 & 3 \end{pmatrix}, \qquad D_- = \begin{pmatrix} -3 & 4 & -1 & & & \\ -1 & 0 & 1 & & & \\ 1 & -4 & 3 & & & \\ & & -6 & 3 & 4 & -1 \\ & & & -1 & 0 & 1 \\ & & & 1 & -4 & 3 \end{pmatrix}, \tag{3.20}$$

and $M = \operatorname{diag}(1/6, 2/3, 1/6, 1/6, 2/3, 1/6)$. These operators satisfy

$$M D_+ + D_-^T M = \operatorname{diag}(-1, 0, 0, 0, 0, 1), \quad M(D_+ - D_-) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3.21}$$

The eigenvalues of the symmetric matrix $M(D_+ - D_-)$ are zero (with multiplicity five) and -2 (with multiplicity one). Thus, it is symmetric and negative semidefinite. Hence, the defining property (2.18) of upwind SBP operators is satisfied. ◁

Remark 3.5. This also holds on periodic domains. Indeed, coupling upwind SBP operators as in Theorem 3.3 on all interfaces results in periodic upwind SBP operators. ◁

3.3. *Classical flux vector splitting using upwind SBP operators*

There are no first-order accurate upwind SBP operators in [36]. However, we can construct such finite difference operators as

$$D_- = \frac{1}{\Delta x} \begin{pmatrix} 0 & 0 & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & -1 & 1 & \\ & & & -2 & 2 \end{pmatrix}, \quad D_+ = \frac{1}{\Delta x} \begin{pmatrix} -2 & 2 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ & & & 0 & 0 \end{pmatrix}, \tag{3.22}$$
$$M = \Delta x \operatorname{diag}(1/2, 1, \dots, 1, 1/2).$$

Indeed,

$$M D_+ + D_-^T M = \operatorname{diag}(-1, 0, \dots, 0, 1) \tag{3.23}$$

and

$$M(D_+ - D_-) = \begin{pmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{pmatrix} \tag{3.24}$$

is negative semidefinite. In fact, (3.24) is the classical finite difference discretization of the Laplacian with homogeneous Neumann boundary conditions. Note that the order of accuracy of $D_\pm$ at one of the boundaries is reduced to zero, in accordance with the general order reduction of SBP operators.

An upwind SBP semidiscretization of a conservation law $\partial_t u + \partial_x f(u) = 0$ is the scheme

$$\partial_t \boldsymbol{u} + D_+ \boldsymbol{f}^- + D_- \boldsymbol{f}^+ = \mathbf{SATs}, \tag{3.25}$$

where **SATs** are boundary terms used to impose the boundary conditions. Applying the upwind SBP operators shown above in such a discretization results in the classical first-order flux vector splitting

$$\begin{aligned}\partial_t u_i &= -\frac{1}{\Delta x}\left(f^*_{i+\frac{1}{2}} - f^*_{i-\frac{1}{2}}\right)\\ &= -\frac{1}{\Delta x}\left((f_i^+ + f_{i+1}^-) - (f_{i-1}^+ + f_i^-)\right)\\ &= -\frac{1}{\Delta x}\left((f_i^+ - f_{i-1}^+) + (f_{i+1}^- - f_i^-)\right)\end{aligned} \tag{3.26}$$

in the interior. Thus, high-order upwind SBP methods can be seen as extensions of the classical first-order flux vector splitting methods.

3.4. Formulation in two-dimensional curvilinear coordinates

The generic conservation law in two dimensions takes the form

$$\partial_t u(t,x,y) + \partial_x f_1\Big(u(t,x,y)\Big) + \partial_y f_2\Big(u(t,x,y)\Big) = 0, \quad t \in (0,T), \quad (x,y) \in \Omega \subset \mathbb{R}^2, \tag{3.27}$$

with conserved variable u and fluxes f_1, f_2 in each coordinate direction, equipped with appropriate initial and boundary conditions. We first subdivide the problem domain Ω into K non-overlapping quadrilateral elements E_k, $k = 1, \dots, K$. In the following, we consider the conservation law (3.27) on an individual element and suppress the index k.

Next, we create a transformation on each element E_k between the computational coordinates $(\xi, \eta) \in E_0$ where $E_0 = [-1,1]^2$ is the reference element and the physical coordinates (x,y) as

$$x = X(\xi,\eta), \quad y = Y(\xi,\eta). \tag{3.28}$$

Typically, this mapping is a linear blending transfinite map between the opposing sides of an element [53,54]. When the element sides are straight, the mapping (3.28) is linear in each coordinate direction. However, if the sides are curved and high-order polynomials are used to approximate the element boundaries, then the mapping (3.28) is a polynomial in each direction. In that case, we represent the mapping as a polynomial of degree N_{geo} in each coordinate direction.

Under this transformation, the conservation law in physical coordinates remains a conservation law in reference coordinates, see, e.g., [38,54]

$$J\,\partial_t u(t,\xi,\eta) + \partial_\xi \tilde{f}_1\Big(u(t,\xi,\eta)\Big) + \partial_\eta \tilde{f}_2\Big(u(t,\xi,\eta)\Big) = 0, \tag{3.29}$$

where the contravariant fluxes $\tilde{f}$ and Jacobian J for the two dimensional transformation are

$$\tilde{f}_1 = Y_\eta f_1 - X_\eta f_2, \quad \tilde{f}_2 = -Y_\xi f_1 + X_\xi f_2, \quad J = Y_\eta X_\xi - Y_\xi X_\eta. \tag{3.30}$$

For convenience we introduce a compact notation for the flux in the contravariant (or normal) direction. The normal direction (but not normalized) vectors in reference space are written as

$$\hat{n}^1 = (Y_\eta, -X_\eta)^T \quad \text{and} \quad \hat{n}^2 = (-Y_\xi, X_\xi)^T, \tag{3.31}$$

where $X_\xi, X_\eta, Y_\xi, Y_\eta$ are the metric terms. So, for example, the first contravariant flux is given by $\tilde{f}_1 = f_1 \hat{n}^1_1 + f_2 \hat{n}^1_2$. Additionally, the metric terms satisfy two metric identities

$$\partial_\xi Y_\eta - \partial_\eta Y_\xi = 0 \quad \text{and} \quad -\partial_\xi X_\eta + \partial_\eta X_\xi = 0 \tag{3.32}$$

that are crucial to guarantee free-stream preservation (FSP) [54–56]. That is, given a flux that is constant in space, its divergence vanishes and the (constant) solution of (3.27) does not change in time. We will revisit the recovery of FSP on the discrete level later in this section.

From the mapped conservation law (3.29) the next step is to perform a flux vector splitting. However, it is important to note that one cannot simply multiply the Cartesian flux vector splittings with the metric terms to create their curvilinear counterparts. This would lead to inconsistencies with respect to the directionality of the waves in the considered splitting. Instead, one follows a procedure of rotation into a generalized coordinate's normal direction, performing the flux vector splitting, and back-rotating the result, see [57,58] for complete details. This process guarantees that the flux vector splittings satisfy the following relationship in each contravariant direction

$$\tilde{f}_i(u) = f_1(u)\hat{n}^i_1 + f_2(u)\hat{n}^i_2 = f^+(u;\hat{n}^i) + f^-(u;\hat{n}^i) = \tilde{f}^+_i(u) + \tilde{f}^-_i(u), \quad i = 1,2. \tag{3.33}$$

We introduce the notation $f^\pm(u;\hat{n}^i)$ to highlight that the normal direction components can no longer be factored out of the flux vector splitting and different flux components may depend on the normal direction in different ways. To clarify the form of the flux vector splittings in generalized coordinate directions we highlight three examples for the compressible Euler equations.

Example 3.6. The local Lax-Friedrichs splitting in the contravariant directions uses a local estimate for the largest value of λ for the possible wave speeds and has the form

$$\tilde{f}_i^{\pm} = f^{\pm}(u;\hat{n}^i) = \frac{1}{2}\left(\tilde{f}_i(u) \pm \lambda u\right), \quad i = 1,2, \tag{3.34}$$

where $u = (\varrho, \varrho v_1, \varrho v_2, \varrho e)^T$ and $\lambda = \sqrt{v_1^2 + v_2^2} + a$. As discussed above, (3.34) has a linear dependency on the mapping terms; however they cannot be factored out to separate the Cartesian splitting from the normal directions as was the case for the complete physical flux (3.30). ◁

Example 3.7. We describe an improved variant of the Steger-Warming splitting (Example 2.2) for generalized coordinates due to Drikakis and Tsangaris [59]. We, again, use the standard notation for the positive/negative part of an eigenvalue λ_i with

$$\lambda_i^{\pm} = \frac{\lambda_i + |\lambda_i|}{2}. \tag{3.35}$$

The wave speeds in the normal direction $\hat{n}^i$ used by this splitting are

$$\tilde{\lambda}_1 = v_1\hat{n}_1^i + v_2\hat{n}_2^i - a, \qquad \tilde{\lambda}_2 = v_1\hat{n}_1^i + v_2\hat{n}_2^i + a, \tag{3.36}$$

with the sound speed $a = \sqrt{\gamma p/\varrho}$. The flux vector splitting of Drikakis and Tsangaris is given by

$$\tilde{f}_i^{\pm} = f^{\pm}(u;\hat{n}^i) = \frac{\varrho}{2}\begin{pmatrix} \tilde{\lambda}_1^{\pm} + \tilde{\lambda}_2^{\pm} \\ (\tilde{\lambda}_1^{\pm} + \tilde{\lambda}_2^{\pm})v_1 + \frac{a\hat{n}_1^i}{\gamma}(\tilde{\lambda}_2^{\pm} - \tilde{\lambda}_1^{\pm}) \\ (\tilde{\lambda}_1^{\pm} + \tilde{\lambda}_2^{\pm})v_2 + \frac{a\hat{n}_2^i}{\gamma}(\tilde{\lambda}_2^{\pm} - \tilde{\lambda}_1^{\pm}) \\ (\tilde{\lambda}_1^{\pm} + \tilde{\lambda}_2^{\pm})H \end{pmatrix}, \tag{3.37}$$

where $H = (\varrho e + p)/\varrho = v^2/2 + a^2/(\gamma - 1)$ is again the enthalpy. ◁

Example 3.8. As a last example, we describe the van Leer-Hänel splitting [45–47] rotated into a contravariant normal direction [57]. We introduce the signed Mach number in the normal direction

$$\tilde{M} = \frac{v_1\hat{n}_1^i + v_2\hat{n}_2^i}{a}, \quad i = 1,2, \tag{3.38}$$

and the pressure splitting

$$p^{\pm} = \frac{1 \pm \gamma\tilde{M}}{2}p. \tag{3.39}$$

The flux splittings are then given by

$$\tilde{f}_i^{\pm} = f^{\pm}(u;\hat{n}^i) = \pm\frac{\varrho a(\tilde{M} \pm 1)^2}{4}\begin{pmatrix} 1 \\ v_1 \\ v_2 \\ H \end{pmatrix} + \begin{pmatrix} 0 \\ \hat{n}_1^i p^{\pm} \\ \hat{n}_2^i p^{\pm} \\ 0 \end{pmatrix}, \tag{3.40}$$

where H is the enthalpy. ◁

Remark 3.9. Notice, as the flux vector splittings become more sophisticated their dependency on the normal direction (and in turn the metric terms) increase in complexity as well. That is, the local Lax-Friedrichs splitting (3.34) is linear in the metric terms, the Drikakis-Tsangaris splitting (3.37) is linear in the advective components and quadratic in the metric terms for the pressure splitting, and the van Leer-Hänel splitting (3.40) is quadratic in the metric terms in all components. ◁

With appropriate flux vector splittings that satisfy (3.33) in hand, we split the contravariant fluxes in mapped conservation law on each element (3.29) to have

$$J\partial_t u + \partial_\xi \tilde{f}_1^+ + \partial_\xi \tilde{f}_1^- + \partial_\eta \tilde{f}_2^+ + \partial_\eta \tilde{f}_2^- = 0. \tag{3.41}$$

Just as in Section 3.1, we discretize the mapped conservation law in space with upwind SBP operators and couple the element to its neighbor elements with appropriate **SATs**. These **SATs** have the same form as given in (3.12) along each of the four element interfaces. As before, if the same splitting in the normal direction is used for the numerical flux as the physical flux, then we recover analogous statements to (3.8) and (3.9). We then have a generic statement of a **SAT** in the normal direction on an interface in element k

$$\widetilde{\mathbf{SAT}}^k = -M^{-1}\boldsymbol{t}_R\left(f^{\mathrm{num}}(\boldsymbol{u}_R^k, \boldsymbol{u}_L^{k+1}; \hat{n}^i) - \tilde{\boldsymbol{f}}_R^k\right) + M^{-1}\boldsymbol{t}_L\left(f^{\mathrm{num}}(\boldsymbol{u}_R^{k-1}, \boldsymbol{u}_L^k; \hat{n}^i) - \tilde{\boldsymbol{f}}_L^k\right). \tag{3.42}$$

The resulting upwind SBP discretization on element k takes the form

$$\boldsymbol{J}\,\partial_t \boldsymbol{u} + D_-\tilde{\boldsymbol{f}}_1^+ + D_+\tilde{\boldsymbol{f}}_1^- + \tilde{\boldsymbol{f}}_2^+ D_-^T + \tilde{\boldsymbol{f}}_2^- D_+^T = \widetilde{\mathbf{SATs}}^k. \tag{3.43}$$

Compared to the one-dimensional case described in Section 3.1, we use a tensor product structure for the mapped conservation law (3.29) on each quadrilateral element. Thus, we can think of the discrete solution $\boldsymbol{u}$ as a two-dimensional array of size $N_\xi \times N_\eta$, where N_ξ and N_η are the number of grid points in the ξ and η directions, respectively. At each grid point with index (i,j), $\boldsymbol{u}_{i,j}$ is the vector of conserved variables at this point. With this data layout, multiplying the mapped fluxes $\tilde{\boldsymbol{f}}_1^\pm$ from the left by the upwind SBP operator $D_\mp$ approximates the derivative in the ξ-direction. Similarly, multiplying the mapped fluxes $\tilde{\boldsymbol{f}}_2^\pm$ from the right by the upwind SBP operator $D_\mp^T$ approximates the derivative in the η-direction. Due to the tensor product structure for the mapped conservation law (3.29), the $\widetilde{\mathbf{SAT}}$ (3.42) in the normal directions are computed in a similar way as in the one-dimensional case.

The final component to fully describe the upwind SBP method on curvilinear domains (3.43) is to discuss how the metric terms are approximated. By design, from [36], the upwind SBP operators $D_\pm$, as well as the central SBP operator $(D_+ + D_-)/2$ they generate, have p^{th} order accurate interior stencils and $p/2$ order accurate boundary stencils. This boundary closure means that any of the three available differencing operators can differentiate polynomials up to degree $p/2$ exactly. For instance, one available upwind SBP operator is the fourth-order interior, second-order boundary closure, denoted 4-2, operator where $D_\pm$ or $D = (D_+ + D_-)/2$ can differentiate up to quadratic polynomials exactly.

Because all available upwind SBP operators of a given order have the same boundary closure accuracy, we use the central operator $D = (D_+ + D_-)/2$ to compute the metric terms by directly differentiating the mapping $X(\xi,\eta)$ from (3.28), i.e.,

$$X_\xi \approx D\boldsymbol{X} = \boldsymbol{X}_{\boldsymbol{\xi}}, \quad X_\eta \approx \boldsymbol{X}D^T = \boldsymbol{X}_{\boldsymbol{\eta}}, \quad Y_\xi \approx D\boldsymbol{Y} = \boldsymbol{Y}_{\boldsymbol{\xi}}, \quad Y_\eta \approx \boldsymbol{Y}D^T = \boldsymbol{Y}_{\boldsymbol{\eta}}. \tag{3.44}$$

We note, depending on the strategy used to compute the discrete metric terms, one may or may not recover a discrete equivalent of the metric identities (3.32). That is, it is possible to lose discrete FSP [54–56]. Applying the approximation strategy from (3.44), we examine the discrete version of the metric identities (3.32) to find

$$D\boldsymbol{Y}_{\boldsymbol{\eta}} - \boldsymbol{Y}_{\boldsymbol{\xi}}D^T = D\boldsymbol{Y}D^T - D\boldsymbol{Y}D^T = \boldsymbol{0} \quad \text{and} \quad -D\boldsymbol{X}_{\boldsymbol{\eta}} + \boldsymbol{X}_{\boldsymbol{\xi}}D^T = -D\boldsymbol{X}D^T + D\boldsymbol{X}D^T = \boldsymbol{0}. \tag{3.45}$$

Thus, the metric identities hold discretely as has been shown previously for finite difference methods in two-dimensional curvilinear coordinates, e.g., [55,56,60]. Moreover, the result (3.45) actually holds independently of the boundary closure accuracy or the polynomial degree N_{geo} of the mapping (3.28). The result that a central SBP finite difference method is free-stream preserving, via the discrete metric identities (3.45), is directly related to the fact that the contravariant fluxes (3.30) have a linear dependency on the metric terms. However, as discussed in Remark 3.9, this is not always the case for a splitting in curvilinear coordinates.

For more sophisticated splittings, like that of van Leer-Hänel in Example 3.8, the issue of FSP becomes more subtle. There is a delicate interplay between the dependency a given splitting has with respect to the metric terms, the boundary closure order of the upwind SBP operator, and the polynomial degree N_{geo} of the mapping. We collect the implications of this interplay into the following theorem.

Theorem 3.10 *(FSP for the curvilinear upwind SBP method). Consider a flux vector splitting that has a maximum dependency on the metric terms of degree m, a set of upwind SBP operators with p^{th} order interior stencils, and mappings $X(\xi,\eta)$ and $Y(\xi,\eta)$ with polynomial degree of N_{geo} in each coordinate direction. The curvilinear upwind SBP method* (3.43) *is free-stream preserving when either*

1. *$m = 1$, i.e., there is a linear dependence on the metric terms, or*
2. *$m > 1$ and the polynomial degree of the mapping satisfies*

$$N_{\text{geo}} \le \frac{p}{2m}. \tag{3.46}$$

That is, the boundary closure can exactly differentiate polynomials up to degree $mN_{\text{geo}} \le p/2$.

Proof. Assume we have a constant solution u_∞ for the conservation law (3.27). The physical variable terms in the flux vector are then also constants.

Part 1$(m = 1)$: By construction, the metric terms in the approximation satisfy the discrete metric identities (3.45). Therefore, the flux splitting terms with a linear dependence on the metric terms vanish from the same reasoning that the standard, non-split curvilinear flux formulation vanishes.

Part 2$(m > 1)$: To guarantee that the curvilinear divergence of the upwind SBP scheme (3.43) vanishes, it is sufficient if the following terms individually vanish

$$D_-\boldsymbol{Y}_{\boldsymbol{\eta}}{}^m - \boldsymbol{Y}_{\boldsymbol{\xi}}{}^m D_-^T, \quad D_+\boldsymbol{Y}_{\boldsymbol{\eta}}{}^m - \boldsymbol{Y}_{\boldsymbol{\xi}}{}^m D_+^T, \quad -D_-\boldsymbol{X}_{\boldsymbol{\eta}}{}^m + \boldsymbol{X}_{\boldsymbol{\xi}}{}^m D_-^T, \quad -D_+\boldsymbol{X}_{\boldsymbol{\eta}}{}^m + \boldsymbol{X}_{\boldsymbol{\xi}}{}^m D_+^T. \tag{3.47}$$

The four terms above are similar to the discrete metric identities, but the metric terms are now polynomials of higher degree. To clarify their appearance, consider the van Leer-Hänel splitting (3.40) of Example 3.8. The signed Mach number $\tilde{M}$ from (3.38) depends linearly on the rotated normal direction $\hat{n}$, i.e., linearly on the metric terms. Thus, the split pressure $p^\pm$ from (3.39) depends linearly on the metric terms as well. Hence, the van-Leer-Hänel splitting (3.40) has a quadratic dependency on the metric terms because the advective components are scaled with $(\tilde{M} \pm 1)^2$ and the pressure components have the form $\hat{n}p^\pm$. As such, for a constant solution u_∞, the fluxes for the van-Leer-Hänel splitting are constant physical variables multiplied with expressions that depend quadratically on the (non-constant) metric terms, i.e., $m = 2$.

Consider the first term in (3.47), i.e., $D_- \boldsymbol{Y_\eta}^m - \boldsymbol{Y_\xi}^m D_-^T$. From the constraint on the polynomial degree of the mapping (3.46) we know that the boundary closure order of the upwind SBP operators can exactly differentiate polynomials up to degree $mN_{\text{geo}} \le p/2$. The discrete metric term $\boldsymbol{Y_\eta} = \boldsymbol{Y} D^T$ is a polynomial of degree N_{geo} in the ξ-direction and degree $N_{\text{geo}} - 1$ in the η-direction and the discrete metric term $\boldsymbol{Y_\xi} = D\boldsymbol{Y}$ is a polynomial of degree $N_{\text{geo}} - 1$ in the ξ-direction and degree N_{geo} in the η-direction. Taking these metric terms to the power m means that $\boldsymbol{Y_\eta}^m$ is a polynomial of degree mN_{geo} in the ξ-direction and degree $m(N_{\text{geo}} - 1)$ in the η-direction and $\boldsymbol{Y_\xi}^m$ is a polynomial of degree $m(N_{\text{geo}} - 1)$ in the ξ-direction and degree mN_{geo} in the η-direction. By design, the upwind SBP derivative operators $D_\pm$ and $D = (D_+ + D_-)/2$ all have the same boundary order closure and can differentiate polynomials up to degree $p/2$ exactly. Therefore, under the constraint (3.46) the D_- operator can exactly differentiate the term $\boldsymbol{Y_\eta}^m$ in the ξ- direction as it is a polynomial of degree mN_{geo}. Similarly, the D_- operator can exactly differentiate the term $\boldsymbol{Y_\xi}^m$ in the η- direction as it is also a polynomial of degree mN_{geo}. Thus,

$$D_- \boldsymbol{Y_\eta}^m - \boldsymbol{Y_\xi}^m D_-^T = \boldsymbol{0}, \tag{3.48}$$

due to the exactness of polynomial differentiation of the boundary closure. The remaining three terms from (3.47) individually vanish from a similar argument. Therefore, the curvilinear upwind SBP method is FSP. □

The result of Theorem 3.10 is two-fold. If the dependency of the curvilinear flux vector splitting on the metric terms remains linear, i.e. $m = 1$, then the curvilinear upwind SBP method (3.43) retains discrete FSP regardless of the boundary closure order and polynomial degree of the mapping N_{geo}. This is the case for the local Lax-Friedrichs splitting in Example 3.6 and, as shown in Section 5.5, this splitting retains discrete FSP for all considered upwind SBP operators and meshes. These results agree with the previous work of Rydin et al. [38], where the global Lax-Friedrichs splitting was used and there were no reported spurious wave artifacts due to curved elements. However, if the curvilinear flux vector splitting has a higher degree polynomial dependence on the metric terms it places a cap on the polynomial degree of curvilinear elements. This is the case for the Drikakis-Tsangaris and van Leer-Hänel splittings of the compressible Euler equations given in Examples 3.7 and 3.8, respectively. For both curvilinear splittings, the maximum dependency on the metric terms is quadratic, so $m = 2$. Thus, upwind SBP operators with a boundary closure order of two are restricted to unstructured bi-linear element meshes and operators with a boundary closure order of four are restricted to at most $N_{\text{geo}} = 2$ or quadratic polynomial boundaries. If curvilinear meshes are constructed with boundaries beyond these values for N_{geo}, then the method is not FSP. We numerically examine the FSP properties of the high-order, curvilinear upwind SBP method (3.43) with different splittings, boundary closures, and mesh polynomial degrees in Section 5.5.

Remark 3.11. As a word of caution, one must take care when adapting a flux vector splitting into the curvilinear high-order upwind SBP context. For instance, the pressure for the van Leer-Hänel splitting (3.39) considered herein is linear with respect to the signed Mach number in the normal direction. This means that the pressure term is quadratic in the metric terms as the pressure splitting is multiplied with the components of the normal direction vector $\hat{n}^i$. Other pressure splittings are proposed by Liou and Steffen [47], e.g.,

$$p^\pm = \frac{1}{4}(\tilde{M} \pm 1)^2 (2 \mp \tilde{M}) p, \tag{3.49}$$

which is cubic in the signed Mach number and, in turn, cubic with respect to the metric terms. This means that overall, the van Leer-Hänel splitting with the above pressure splitting has $m = 4$, as the pressure components are further multiplied with $\hat{n}$. Thus, bi-linear element meshes with $N_{\text{geo}} = 1$ require at least fourth order boundary closures to guarantee FSP of the upwind SBP method. ◁

Remark 3.12. The constraint on the boundary polynomial degree (3.46) is similar to the constraint found by Kopriva [54, Theorem 4] for three-dimensional cross-product discrete metric terms. ◁

Remark 3.13. The conditions given in Theorem 3.10 are sufficient to guarantee discrete FSP. They may not be necessary in all cases. However, the numerical results in Section 5.5 suggest that the result is sharp for the considered flux vector splittings, nodal polynomial approximation of curved boundaries, and tensor product extension to curvilinear coordinates. ◁

4. Analysis of local linear/energy stability

From the review and discussion of upwind SBP methods for nonlinear conservation laws, we now turn to one goal of this article: the analysis of local linear/energy stability properties. We follow [10] and consider local linear/energy stability for Burgers' equation

$$\partial_t u(t,x) + \partial_x \frac{u(t,x)^2}{2} = 0 \tag{4.1}$$

with periodic boundary conditions. Thus, we linearize the equation around a baseflow $\widetilde{u}$, write $u = \widetilde{u} + v$, and get

$$\partial_t v + \partial_x(\widetilde{u} v) = 0. \tag{4.2}$$

This is a linear advection equation for the perturbation v with variable coefficient $\widetilde{u}$. For a positive baseflow $\widetilde{u} > 0$, the spatial operator has an imaginary spectrum since it is skew-symmetric with respect to the weighted L^2 inner product $(v, w) \mapsto \int \widetilde{u} v w$ [10,61].

However, Gassner et al. [10] observed that high-order semidiscretizations conserving/dissipating the L^2 entropy $\int u^2$ lead to a linearized operator having some eigenvalues with a significantly positive real part for a non-constant baseflow $\widetilde{u}$. They discussed this in the context of local linear/energy stability and issues of discretizations for under-resolved flows, see also [62].

From the results and discussion in [10], it appears to be desirable that a semidiscretization mimics the property of a linearization having eigenvalues with non-positive real part for all positive baseflows $\widetilde{u}$. Before studying the upwind SBP method specifically for Burgers' equation, we concentrate on constant baseflows in a general setting.

Consider a scalar conservation law $\partial_t u + \partial_x f(u) = 0$ with periodic boundary conditions. We first consider entropy-conservative flux differencing schemes of the form

$$\partial_t \boldsymbol{u}_i + \sum_j 2 D_{ij} f^{\mathrm{vol}}(\boldsymbol{u}_i, \boldsymbol{u}_j) = 0, \tag{4.3}$$

where D is a periodic SBP operator and f^{vol} is an entropy-conservative numerical flux in the sense of Tadmor [1,2], i.e., it satisfies

$$\forall u_l, u_r: \left(w(u_r) - w(u_l) \right) \cdot f^{\mathrm{vol}}(u_l, u_r) = \psi(u_r) - \psi(u_l), \tag{4.4}$$

where $w(u) = U'(u)$ are the entropy variables and ψ is the flux potential associated to a convex entropy U. These methods have been introduced in [3,4]; see also [63,64].

Theorem 4.1. *Entropy-conservative semidiscretizations using flux differencing in periodic domains are linearly/energy stable around constant states; in particular, their Jacobian has a purely imaginary spectrum.*

Proof. Here, we use the notation of [65, Theorem 2.1], i.e., $Q = M D$ and $F_{ij} = f^{\mathrm{vol}}(\boldsymbol{u}_i, \boldsymbol{u}_j)$. The Jacobian (multiplied by the negative mass matrix) is

$$J = 2(Q \circ F_y) - \operatorname{diag}(1^T (2 Q \circ F_y)), \tag{4.5}$$

where Q is skew-symmetric (due to the periodic boundary conditions) and $\circ$ denotes the Hadamard (pointwise) product of two matrices. The matrix F_y is given by the entries $\partial_2 f^{\mathrm{vol}}(\boldsymbol{u}_i, \boldsymbol{u}_j)$, i.e., the derivatives of the numerical flux f^{vol} with respect to the second argument evaluated at the states $\boldsymbol{u}_i, \boldsymbol{u}_j$. Since the derivative F_y is evaluated at a constant state $\widetilde{u}$, all of its components are the same. In particular, $\forall i, j: (F_y)_{ij} = \frac{1}{2} f'(u)$ [66, Lemma 3.1]. Thus,

$$J = 2(Q \circ F_y) - \operatorname{diag}(1^T (2 Q \circ F_y)) = f'(u) \left(Q - \operatorname{diag}(1^T Q) \right) = f'(u) Q, \tag{4.6}$$

where we used the SBP property. Hence, J is skew-symmetric and has a purely imaginary spectrum. □

Next, we consider a central SBP discretization of the form

$$\partial_t \boldsymbol{u} + D \boldsymbol{f} = \boldsymbol{0}, \tag{4.7}$$

where D is a periodic SBP operator. As observed numerically in [10], this leads to a purely imaginary spectrum of the linearization.

Theorem 4.2. *Central nodal diagonal-norm SBP semidiscretizations of conservation laws in periodic domains are linearly stable around states with positive speed $f'(\widetilde{u}) > 0$; in particular, the Jacobian has a purely imaginary spectrum.*

Proof. The Jacobian (multiplied by the negative mass matrix) is

$$J = Q \operatorname{diag}(\boldsymbol{f}'). \tag{4.8}$$

Thus, it is skew-symmetric w.r.t. the inner product weighted by $\boldsymbol{f}' > 0$. □

Next, we consider fully upwind SBP methods.

Theorem 4.3. *Consider a possibly spatially varying baseflow $\widetilde{u}$ with positive speed $f'(\widetilde{u}) > 0$ everywhere. Upwind nodal diagonal-norm SBP semidiscretizations of the form $\partial_t \boldsymbol{u} + D_- \boldsymbol{f} = \boldsymbol{0}$ in periodic domains are linearly stable; in particular, the Jacobian has a spectrum in the left half of the complex plane.*

Proof. The Jacobian of the semidiscretization $-D_- \boldsymbol{f}$ is

$$J = -D_- \operatorname{diag}(\boldsymbol{f}'). \tag{4.9}$$

The spectrum of this operator must be in the left half of the complex plane, since for each (possibly complex-valued) vector $\boldsymbol{v}$

$$\begin{aligned}
2\operatorname{Re}\langle \boldsymbol{v}, J\boldsymbol{v}\rangle_{\operatorname{diag}(\boldsymbol{f}')M} &= \langle \boldsymbol{v}, J\boldsymbol{v}\rangle_{\operatorname{diag}(\boldsymbol{f}')M} + \langle J\boldsymbol{v}, \boldsymbol{v}\rangle_{\operatorname{diag}(\boldsymbol{f}')M} \\
&= -\boldsymbol{v}^* \operatorname{diag}(\boldsymbol{f}') M D_- \operatorname{diag}(\boldsymbol{f}')\boldsymbol{v} - \boldsymbol{v}^* \operatorname{diag}(\boldsymbol{f}') D_-^T M \operatorname{diag}(\boldsymbol{f}')\boldsymbol{v} \\
&= \boldsymbol{v}^* \operatorname{diag}(\boldsymbol{f}')(-D_-^T M - M D_-) \operatorname{diag}(\boldsymbol{f}')\boldsymbol{v} \\
&= \boldsymbol{v}^* \operatorname{diag}(\boldsymbol{f}')(M D_+ - M D_-) \operatorname{diag}(\boldsymbol{f}')\boldsymbol{v} \le 0,
\end{aligned} \tag{4.10}$$

where we used the SBP property and negative semidefiniteness for periodic upwind operators (2.30) in the last two steps. □

Remark 4.4. The assumption of a positive speed $f'(\widetilde{u}) > 0$ is equivalent to a positive baseflow $\widetilde{u}$ for Burgers' equation. While this is a strong assumption, it is exactly the situation investigated in [10] where local linear/energy stability fails for entropy-stable methods. ◁

We get similar results for splittings such as the Lax-Friedrichs splitting, at least for Burgers' equation.

Theorem 4.5. *Upwind nodal diagonal-norm SBP semidiscretizations of Burgers' equation with local Lax-Friedrichs flux splitting in periodic domains are linearly stable around positive states. In particular, the Jacobian has a spectrum in the left half of the complex plane.*

Proof. The flux splitting is

$$\frac{u^2}{2} = \frac{1}{2}\left(\frac{u^2}{2} + |u|u\right) + \frac{1}{2}\left(\frac{u^2}{2} - |u|u\right). \tag{4.11}$$

For positive $\boldsymbol{u}$, the semidiscretization is

$$\partial_t \boldsymbol{u} = -\frac{3}{4} D_- \boldsymbol{u}^2 + \frac{1}{4} D_+ \boldsymbol{u}^2. \tag{4.12}$$

The Jacobian of the right-hand side is

$$J = -\frac{3}{4} D_- \operatorname{diag}((\boldsymbol{u}^2)') + \frac{1}{4} D_+ \operatorname{diag}((\boldsymbol{u}^2)'). \tag{4.13}$$

As in Theorem 4.3, we can show that this Jacobian has a spectrum in the left half of the complex plane, since for all (complex) grid vectors $\boldsymbol{v}$

$$\begin{aligned}
2\operatorname{Re}\langle \boldsymbol{v}, J\boldsymbol{v}\rangle_{\operatorname{diag}((\boldsymbol{u}^2)')M} &= -\frac{3}{2}\boldsymbol{v}^* \operatorname{diag}((\boldsymbol{u}^2)') M D_- \operatorname{diag}((\boldsymbol{u}^2)')\boldsymbol{v} + \frac{1}{2}\boldsymbol{v}^* \operatorname{diag}((\boldsymbol{u}^2)') M D_+ \operatorname{diag}((\boldsymbol{u}^2)')\boldsymbol{v} \\
&\quad - \frac{3}{2}\boldsymbol{v}^* \operatorname{diag}((\boldsymbol{u}^2)') D_-^T M \operatorname{diag}((\boldsymbol{u}^2)')\boldsymbol{v} + \frac{1}{2}\boldsymbol{v}^* \operatorname{diag}((\boldsymbol{u}^2)') D_+^T M \operatorname{diag}((\boldsymbol{u}^2)')\boldsymbol{v} \\
&= \frac{3}{2}\boldsymbol{v}^* \operatorname{diag}((\boldsymbol{u}^2)') \left(-M D_- - D_-^T M\right) \operatorname{diag}((\boldsymbol{u}^2)')\boldsymbol{v} \\
&\quad + \frac{1}{2}\boldsymbol{v}^* \operatorname{diag}((\boldsymbol{u}^2)') \left(M D_+ + D_+^T M\right) \operatorname{diag}((\boldsymbol{u}^2)')\boldsymbol{v}
\end{aligned} \tag{4.14}$$

Both matrices in brackets are negative semidefinite, since the upwind SBP properties guarantee that

$$-M D_- - D_-^T M = M D_+ - M D_- = M D_+ + D_+^T M \tag{4.15}$$

is negative semidefinite. □

Remark 4.6. The proof of Theorem 4.5 holds for scalar conservation laws with homogeneous flux $f(su) = s^\alpha f(u)$. In this case, the flux splitting with positive speeds is

$$f_+ = \frac{1}{2}\left(f + f_u u\right) = \frac{1+\alpha}{2} f, \quad f_- = \frac{1}{2}\left(f - f_u u\right) = \frac{1-\alpha}{2} f, \tag{4.16}$$

due to Eulers' theorem. ◁

4.1. A special choice of entropy

We relate standard central schemes $\partial_t \boldsymbol{u} + D\boldsymbol{f} = \boldsymbol{0}$ to entropy-conservative schemes with a special choice of entropy function. Consider a scalar conservation law $\partial_t u + \partial_x f(u) = 0$ with positive wave speeds $f'(u) > 0$. In this case, the primitive $U(u) = \int^u f(y)\,\mathrm{d}y$ of the flux is a convex entropy with entropy flux $F = f^2/2$, cf. [67]. The associated entropy variable $w = U' = f$ is the flux itself and a smooth solution yields

$$\partial_t U + \partial_x F = \partial_t U + \partial_x \frac{f^2}{2} = f \cdot (\partial_t u + \partial_x f) = 0. \tag{4.17}$$

Thus, the entropy-conservative numerical flux of Tadmor is given by the central flux since

$$(f_r - f_l)\frac{f_r + f_l}{2} = \frac{f_r^2}{2} - \frac{f_l^2}{2}, \tag{4.18}$$

where we have used the entropy flux potential

$$\psi = w \cdot f - F = f^2 - \frac{f^2}{2} = \frac{f^2}{2}. \tag{4.19}$$

Finally, flux differencing methods using the central numerical flux are equivalent to the central discretization $\partial_t \boldsymbol{u} + D\boldsymbol{f} = \boldsymbol{0}$. Thus, nonlinear entropy stability and local linear/energy stability can be combined in this very special situation.

Remark 4.7. The dissipation introduced by upwind SBP operators is compatible with the structure of the local linear/energy stability estimate. In particular, the dissipation introduced compared to a central scheme dissipates the entropy $U(u) = \int^u f(y)\,\mathrm{d}y$, since

$$\boldsymbol{f}^T M(-D_-\boldsymbol{f}) = -\boldsymbol{f}^T M\frac{D_+ + D_-}{2}\boldsymbol{f} + \boldsymbol{f}^T M\frac{D_+ - D_-}{2}\boldsymbol{f} \le -\boldsymbol{f}^T M\frac{D_+ + D_-}{2}\boldsymbol{f}. \tag{4.20}$$

The central scheme with operator $(D_+ + D_-)/2$ conserves this entropy, so the total upwind scheme is entropy-dissipative. ◁

4.2. Discussion

Following [10], three desirable properties of numerical methods are i) nonlinear entropy stability, ii) local linear/energy stability, and iii) high-order accuracy. There have been substantial discussions in the high-order community about these properties. Clearly, central-type schemes can be high-order accurate and just satisfy local linear/energy stability without any dissipation. We have shown that upwind SBP methods can have the same properties while coming with some built-in dissipation. However, it is unclear whether they have some nonlinear/entropy stability properties besides the special, academic choice of entropy in the previous subsection. Clearly, first-order methods such as Godunov's method can have both nonlinear entropy and local linear/energy stability properties. It is an open question whether higher-order methods can have both of these stability properties as well.

5. Numerical experiments

We use the Julia programming language [68] for the numerical experiments. Time integration is performed using Runge-Kutta methods implemented in OrdinaryDiffEq.jl [69] (specific choices of the Runge-Kutta methods are stated below). The spatial discretizations are available in Trixi.jl [70,71]. All numerical experiments presented in this section use diagonal-norm upwind SBP operators of [36] available from SummationByPartsOperators.jl [72] (unless stated otherwise). Some of the unstructured curvilinear quadrilateral meshes were constructed with HOHQMesh.jl.[1] We use Plots.jl [73] and ParaView [74] to visualize the results. All source code required to reproduce the numerical experiments is available online in our reproducibility repository [75].

5.1. Convergence experiments with linear advection

First, we consider the linear advection equation

$$\begin{aligned} \partial_t u(t,x) + \partial_x u(t,x) &= 0, \qquad && t \in (0,5), x \in (-1,1), \\ u(0,x) &= \sin(\pi x), && x \in [-1,1], \end{aligned} \tag{5.1}$$

with periodic boundary conditions. We use the classical Lax-Friedrichs flux vector splitting with $\lambda = 1$, i.e.,

$$f^-(u) = 0, \quad f^+(u) = u. \tag{5.2}$$

We use the fourth-order accurate Runge-Kutta method of [76] with error-based step size control and a sufficiently small tolerance to integrate the semidiscretizations in time. We measure the discrete L^2 error using the quadrature rule induced by the mass matrix M. Results of these convergence experiments, including the experimental order of convergence (EOC), are shown in Tables 1–4.

When used in DG refinement mode, i.e., increasing the number of elements while keeping the number of nodes per element constant, the methods with an interior order of accuracy p converge with an EOC of $\lfloor p/2 + 1 \rfloor$. When used in FD refinement mode, i.e., increasing the number of nodes per element while keeping the number of elements constant, the methods with an interior order of accuracy p converge with an EOC of roughly $\max(p, \lfloor p/2 + 1 \rfloor + 1/2)$.

5.2. Convergence experiments with the compressible Euler equations

Next, we investigate the experimental order of convergence for the upwind SBP framework with different flux vector splittings in one and two space dimensions. The one-dimensional results are presented in Section 5.2.1 and the two-dimensional results on unstructured curvilinear meshes are given in Section 5.2.2.

[1] https://github.com/trixi-framework/HOHQMesh.jl.

Table 1
Convergence results using upwind SBP discretizations of the linear advection equation with Lax-Friedrichs splitting, K elements, N nodes per element, and an interior order of accuracy 2.

K	N	L^2 error	EOC	K	N	L^2 error	EOC
1	20	3.46×10^{-1}		4	10	9.32×10^{-2}	
2	20	9.24×10^{-2}	1.91	4	20	2.33×10^{-2}	2.00
4	20	2.33×10^{-2}	1.99	4	40	5.77×10^{-3}	2.01
8	20	5.83×10^{-3}	2.00	4	80	1.44×10^{-3}	2.01
16	20	1.46×10^{-3}	2.00	4	160	3.58×10^{-4}	2.00
32	20	3.64×10^{-4}	2.00	4	320	8.94×10^{-5}	2.00
64	20	9.11×10^{-5}	2.00	4	640	2.23×10^{-5}	2.00
128	20	2.28×10^{-5}	2.00	4	1280	5.58×10^{-6}	2.00

Table 2
Convergence results using upwind SBP discretizations of the linear advection equation with Lax-Friedrichs splitting, K elements, N nodes per element, and an interior order of accuracy 3.

K	N	L^2 error	EOC	K	N	L^2 error	EOC
1	20	3.40×10^{-2}		4	10	7.98×10^{-3}	
2	20	4.93×10^{-3}	2.78	4	20	8.73×10^{-4}	3.19
4	20	8.73×10^{-4}	2.50	4	40	1.05×10^{-4}	3.06
8	20	1.87×10^{-4}	2.22	4	80	1.34×10^{-5}	2.96
16	20	4.47×10^{-5}	2.07	4	160	1.83×10^{-6}	2.87
32	20	1.11×10^{-5}	2.01	4	320	2.68×10^{-7}	2.77
64	20	2.76×10^{-6}	2.00	4	640	4.20×10^{-8}	2.68
128	20	6.90×10^{-7}	2.00	4	1280	6.91×10^{-9}	2.60

Table 3
Convergence results using upwind SBP discretizations of the linear advection equation with Lax-Friedrichs splitting, K elements, N nodes per element, and an interior order of accuracy 4.

K	N	L^2 error	EOC	K	N	L^2 error	EOC
1	20	5.03×10^{-3}		4	10	3.86×10^{-4}	
2	20	3.96×10^{-4}	3.67	4	20	3.19×10^{-5}	3.60
4	20	3.19×10^{-5}	3.63	4	40	2.57×10^{-6}	3.63
8	20	3.66×10^{-6}	3.12	4	80	2.10×10^{-7}	3.61
16	20	4.51×10^{-7}	3.02	4	160	1.77×10^{-8}	3.57
32	20	5.57×10^{-8}	3.02	4	320	1.51×10^{-9}	3.55
64	20	6.96×10^{-9}	3.00	4	640	1.30×10^{-10}	3.54
128	20	8.73×10^{-10}	3.00	4	1280	1.17×10^{-11}	3.48

Table 4
Convergence results using upwind SBP discretizations of the linear advection equation with Lax-Friedrichs splitting, K elements, N nodes per element, and an interior order of accuracy 5.

K	N	L^2 error	EOC	K	N	L^2 error	EOC
1	20	3.49×10^{-3}		4	10	5.17×10^{-4}	
2	20	3.77×10^{-4}	3.21	4	20	3.31×10^{-5}	3.97
4	20	3.31×10^{-5}	3.51	4	40	2.64×10^{-6}	3.65
8	20	4.09×10^{-6}	3.01	4	80	2.23×10^{-7}	3.56
16	20	5.13×10^{-7}	3.00	4	160	1.93×10^{-8}	3.53
32	20	6.43×10^{-8}	3.00	4	320	1.69×10^{-9}	3.51
64	20	8.02×10^{-9}	3.00	4	640	1.51×10^{-10}	3.48
128	20	1.01×10^{-9}	2.99	4	1280	1.34×10^{-11}	3.49

5.2.1. One spatial dimension

Consider the 1D compressible Euler equations

$$\partial_t \begin{pmatrix} \varrho \\ \varrho v \\ \varrho e \end{pmatrix} + \partial_x \begin{pmatrix} \varrho v \\ \varrho v^2 + p \\ (\varrho e + p) v \end{pmatrix} = 0 \tag{5.3}$$

of an ideal gas with density ϱ, velocity v, total energy density ϱe, and pressure

Table 5
Convergence results using upwind SBP discretizations of the compressible Euler equations with K elements, N nodes per element, and an interior order of accuracy 2.

(a) van Leer-Hänel spl. [45–47].

K	N	L^2 error	EOC
1	20	1.01×10^{-2}	
2	20	2.94×10^{-3}	1.78
4	20	7.55×10^{-4}	1.96
8	20	1.91×10^{-4}	1.98
16	20	4.79×10^{-5}	2.00
32	20	1.19×10^{-5}	2.01
64	20	2.98×10^{-6}	2.00
128	20	7.45×10^{-7}	2.00

(b) Steger-Warming splitting [44].

K	N	L^2 error	EOC
1	20	1.02×10^{-2}	
2	20	2.95×10^{-3}	1.79
4	20	7.59×10^{-4}	1.96
8	20	1.92×10^{-4}	1.99
16	20	4.79×10^{-5}	2.00
32	20	1.19×10^{-5}	2.01
64	20	2.98×10^{-6}	2.00
128	20	7.46×10^{-7}	2.00

(c) Steger-Warming splitting [44].

K	N	L^2 error	EOC
4	10	2.97×10^{-3}	
4	20	7.59×10^{-4}	1.97
4	40	1.89×10^{-4}	2.01
4	80	4.69×10^{-5}	2.01
4	160	1.17×10^{-5}	2.01
4	320	2.92×10^{-6}	2.00
4	640	7.29×10^{-7}	2.00
4	1280	1.82×10^{-7}	2.00

Table 6
Convergence results using upwind SBP discretizations of the compressible Euler equations with K elements, N nodes per element, and an interior order of accuracy 3.

(a) van Leer-Hänel spl. [45–47].

K	N	L^2 error	EOC
1	20	1.14×10^{-3}	
2	20	1.75×10^{-4}	2.70
4	20	4.41×10^{-5}	1.99
8	20	1.21×10^{-5}	1.86
16	20	3.16×10^{-6}	1.94
32	20	5.96×10^{-7}	2.41
64	20	1.43×10^{-7}	2.06
128	20	3.43×10^{-8}	2.06

(b) Steger-Warming splitting [44].

K	N	L^2 error	EOC
1	20	1.18×10^{-3}	
2	20	1.83×10^{-4}	2.69
4	20	4.51×10^{-5}	2.02
8	20	1.21×10^{-5}	1.90
16	20	3.05×10^{-6}	1.98
32	20	5.59×10^{-7}	2.45
64	20	1.35×10^{-7}	2.05
128	20	3.22×10^{-8}	2.07

(c) Steger-Warming splitting [44].

K	N	L^2 error	EOC
4	10	2.97×10^{-4}	
4	20	4.51×10^{-5}	2.72
4	40	7.56×10^{-6}	2.58
4	80	1.16×10^{-6}	2.70
4	160	1.78×10^{-7}	2.70
4	320	2.81×10^{-8}	2.66
4	640	4.59×10^{-9}	2.62
4	1280	7.69×10^{-10}	2.58

Table 7
Convergence results using upwind SBP discretizations of the compressible Euler equations with K elements, N nodes per element, and an interior order of accuracy 4.

(a) van Leer-Hänel spl. [45–47].

K	N	L^2 error	EOC
1	20	3.01×10^{-4}	
2	20	3.51×10^{-5}	3.10
4	20	4.14×10^{-6}	3.08
8	20	5.84×10^{-7}	2.83
16	20	6.60×10^{-8}	3.14
32	20	6.68×10^{-9}	3.31
64	20	8.24×10^{-10}	3.02
128	20	9.63×10^{-11}	3.10

(b) Steger-Warming splitting [44].

K	N	L^2 error	EOC
1	20	2.30×10^{-4}	
2	20	2.71×10^{-5}	3.09
4	20	4.21×10^{-6}	2.69
8	20	5.88×10^{-7}	2.84
16	20	6.50×10^{-8}	3.18
32	20	6.53×10^{-9}	3.32
64	20	7.94×10^{-10}	3.04
128	20	9.17×10^{-11}	3.11

(c) Steger-Warming splitting [44].

K	N	L^2 error	EOC
4	10	3.72×10^{-5}	
4	20	4.21×10^{-6}	3.14
4	40	3.66×10^{-7}	3.52
4	80	2.66×10^{-8}	3.78
4	160	1.95×10^{-9}	3.77
4	320	1.60×10^{-10}	3.61
4	640	1.37×10^{-11}	3.54
4	1280	2.05×10^{-12}	2.74

$$p = (\gamma - 1)\left(\varrho e - \frac{1}{2}\varrho v^2\right), \tag{5.4}$$

where the ratio of specific heats is chosen as $\gamma = 1.4$. We add a source term to create the manufactured solution

$$\varrho(t,x) = h(t,x), \quad v(t,x) = 1, \quad \varrho e(t,x) = h(t,x)^2, \tag{5.5}$$

with

$$h(t,x) = 2 + 0.1\sin\Big(\pi(x-t)\Big) \tag{5.6}$$

for $t \in [0,2]$ and $x \in [0,2]$. We use the flux vector splittings introduced in Examples 2.2 and 2.3.

The convergence results for the compressible Euler equations shown in Tables 5–8 confirm that the behavior of the experimental order of convergence observed earlier for the linear advection equation remains the same for a nonlinear hyperbolic system.

5.2.2. Two spatial dimensions

Next, consider the 2D compressible Euler equations

$$\partial_t \begin{pmatrix} \varrho \\ \varrho v_1 \\ \varrho v_2 \\ \varrho e \end{pmatrix} + \partial_x \begin{pmatrix} \varrho v_1 \\ \varrho v_1^2 + p \\ \varrho v_1 v_2 \\ (\varrho e + p) v_1 \end{pmatrix} + \partial_y \begin{pmatrix} \varrho v_2 \\ \varrho v_1 v_2 \\ \varrho v_2^2 + p \\ (\varrho e + p) v_2 \end{pmatrix} = 0 \tag{5.7}$$

Table 8
Convergence results using upwind SBP discretizations of the compressible Euler equations with K elements, N nodes per element, and an interior order of accuracy 5.

(a) van Leer-Hänel spl. [45–47].

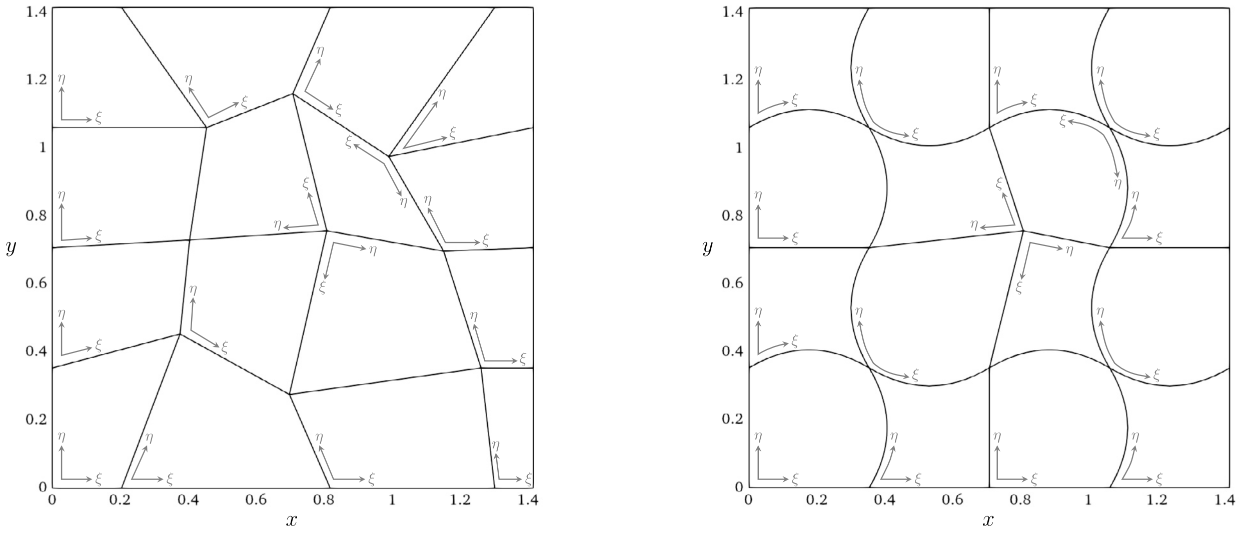
K	N	L^2 error	EOC
1	20	2.05×10^{-4}	
2	20	3.29×10^{-5}	2.64
4	20	3.74×10^{-6}	3.14
8	20	5.15×10^{-7}	2.86
16	20	5.99×10^{-8}	3.10
32	20	6.97×10^{-9}	3.10
64	20	8.67×10^{-10}	3.01
128	20	1.05×10^{-10}	3.04

(b) Steger-Warming splitting [44].

K	N	L^2 error	EOC
1	20	1.17×10^{-4}	
2	20	2.44×10^{-5}	2.26
4	20	4.13×10^{-6}	2.56
8	20	5.20×10^{-7}	2.99
16	20	5.85×10^{-8}	3.15
32	20	6.78×10^{-9}	3.11
64	20	8.38×10^{-10}	3.02
128	20	1.02×10^{-10}	3.04

(c) Steger-Warming splitting [44].

K	N	L^2 error	EOC
4	10	4.20×10^{-5}	
4	20	4.13×10^{-6}	3.35
4	40	3.28×10^{-7}	3.66
4	80	2.53×10^{-8}	3.69
4	160	2.05×10^{-9}	3.63
4	320	1.75×10^{-10}	3.55
4	640	1.52×10^{-11}	3.52
4	1280	1.63×10^{-12}	3.23



(a) Mesh with bi-linear elements.

(b) Mesh with (possibly) quadratic elements.

Fig. 1. Non-overlapping quadrilateral meshes used for the convergence testing on unstructured meshes. The local coordinate axes on each element denoted with ξ and η demonstrate that several elements have flipped local coordinate systems with respect to their neighbor elements.

of an ideal gas with density ϱ, velociteis v_1, v_2, total energy density ϱe, and pressure

$$p = (\gamma - 1)\left(\varrho e - \frac{1}{2}\varrho(v_1^2 + v_2^2)\right), \tag{5.8}$$

where the ratio of specific heats is chosen as $\gamma = 1.4$. We add a source term to create the manufactured solution

$$\varrho(t,x) = h(t,x), \quad v_1(t,x) = v_2(t,x) = 1, \quad \varrho e(t,x) = h(t,x)^2, \tag{5.9}$$

with

$$h(t,x) = 2 + 0.1 \sin\left(\sqrt{2}\pi(x-t)\right) \tag{5.10}$$

for $t \in [0,2]$, $x \in [0,\sqrt{2}]^2$, and periodic boundary conditions. The full expressions of the source terms and all code required to reproduce these experiments is available in our reproducibility repository [75].

We subdivide the domain $[0,\sqrt{2}]^2$ with 16 non-overlapping quadrilateral elements. For these convergence tests we consider two unstructured meshes, one with only bi-linear elements and the other containing internal element boundaries approximated with quadratic polynomials. Moreover, we design these meshes such that several neighboring elements have flipped local coordinate systems, as is possible in unstructured mesh computations. Even so, the domain discretized with either mesh remains periodic as required by the manufactured solution setup. The two meshes used for the convergence testing are given in Fig. 1.

We use the manufactured solution described above to compute the experimental order of convergence for the Lax-Friedrichs, Drikakis-Tsangaris, and van Leer-Hänel splittings on both meshes given in Fig. 1. In particular, we use the bi-linear unstructured mesh from Fig. 1(a) to test convergence of the 4-2 and 6-3 upwind SBP operators and the quadratic unstructured mesh shown in Fig. 1(b) to test the convergence of the 8-4 upwind SBP operator.

Table 9
Convergence results using curvilinear upwind SBP discretizations for the compressible Euler equations with K elements, N nodes per element, and an interior order of accuracy 4 on the unstructured bi-linear mesh shown in Fig. 1(a).

(a) local Lax-Friedrichs spl. Ex. 3.6.

K	N	L^2 error	EOC
16	17	1.92×10^{-4}	
16	34	1.60×10^{-5}	3.58
16	68	1.46×10^{-6}	3.45
16	136	1.53×10^{-7}	3.26
16	272	1.66×10^{-8}	3.20

(b) Drikakis-Tsangaris splitting [59].

K	N	L^2 error	EOC
16	17	1.35×10^{-4}	
16	34	1.18×10^{-5}	3.52
16	68	1.14×10^{-6}	3.36
16	136	1.18×10^{-7}	3.27
16	272	1.30×10^{-8}	3.18

(c) van Leer-Hänel splitting [57].

K	N	L^2 error	EOC
16	17	9.21×10^{-5}	
16	34	8.07×10^{-6}	3.51
16	68	8.15×10^{-7}	3.31
16	136	8.19×10^{-8}	3.31
16	272	8.83×10^{-9}	3.21

Table 10
Convergence results using curvilinear upwind SBP discretizations for the compressible Euler equations with K elements, N nodes per element, and an interior order of accuracy 6 on the unstructured bi-linear mesh shown in Fig. 1(a).

(a) local Lax-Friedrichs spl. Ex. 3.6.

K	N	L^2 error	EOC
16	17	1.95×10^{-5}	
16	34	9.26×10^{-7}	4.40
16	68	4.71×10^{-8}	4.30
16	136	2.46×10^{-9}	4.26
16	272	1.32×10^{-10}	4.22

(b) Drikakis-Tsangaris splitting [59].

K	N	L^2 error	EOC
16	17	2.15×10^{-5}	
16	34	1.04×10^{-6}	4.37
16	68	5.75×10^{-8}	4.17
16	136	3.06×10^{-9}	4.23
16	272	1.65×10^{-10}	4.21

(c) van Leer-Hänel splitting [57].

K	N	L^2 error	EOC
16	17	2.33×10^{-5}	
16	34	1.15×10^{-6}	4.34
16	68	6.74×10^{-8}	4.09
16	136	3.70×10^{-9}	4.19
16	272	2.06×10^{-10}	4.17

Table 11
Convergence results using curvilinear upwind SBP discretizations for the compressible Euler equations with K elements, N nodes per element, and an interior order of accuracy 8 on the unstructured quadratic curvilinear mesh shown in Fig. 1(b).

(a) local Lax-Friedrichs spl. Ex. 3.6.

K	N	L^2 error	EOC
16	17	1.71×10^{-5}	
16	34	4.16×10^{-7}	5.36
16	68	1.17×10^{-8}	5.15
16	136	3.64×10^{-10}	5.01
16	272	1.15×10^{-11}	4.99

(b) Drikakis-Tsangaris splitting [59].

K	N	L^2 error	EOC
16	17	1.71×10^{-5}	
16	34	4.40×10^{-7}	5.28
16	68	1.32×10^{-8}	5.06
16	136	4.33×10^{-10}	4.93
16	272	1.33×10^{-11}	5.02

(c) van Leer-Hänel splitting [57].

K	N	L^2 error	EOC
16	17	1.66×10^{-5}	
16	34	4.51×10^{-7}	5.20
16	68	1.42×10^{-8}	4.99
16	136	4.84×10^{-10}	4.88
16	272	1.52×10^{-11}	4.99

The convergence results for the compressible Euler equations shown in Tables 9–11 confirm the behavior of the experimental order of convergence observed for the earlier one-dimensional convergence tests.

5.3. Spectral analysis

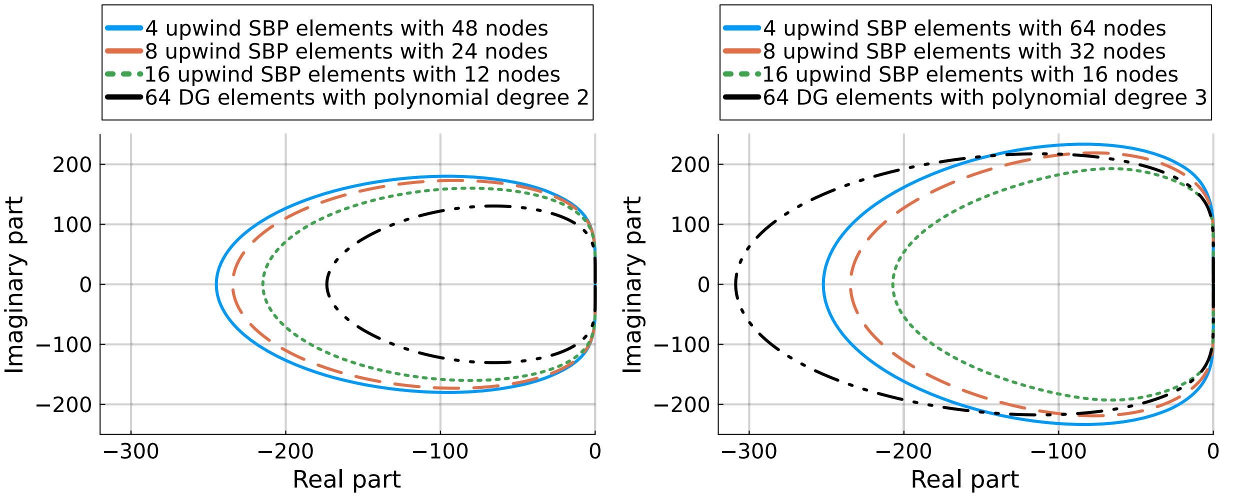
We consider the same linear advection setup with periodic boundary conditions as in Section 5.1 and compute the spectra of the semidiscretizations. The results visualized in Fig. 2 demonstrate the linear stability of the upwind discretizations, since the spectra are contained in the left half of the complex plane and the maximum real part is zero up to machine precision. Furthermore, they indicate that the stiffness of the methods (as measured by the largest eigenvalue by magnitude) increases when reducing the number of elements and increasing the number of nodes per element such that the total number of degrees of freedom (DOFs) is constant.

In general, the spectra are comparable to the spectra obtained by the classical nodal DGSEM method on Gauss-Lobatto-Legendre nodes. The spectra shown in Fig. 2 suggest that the upwind SBP methods with an interior order of accuracy 4 are stiffer than DGSEM with a polynomial degree of 2; the situation is reversed for upwind SBP methods with an interior of accuracy 6 and DGSEM with a polynomial degree of 3.

5.4. Local linear/energy stability

Next, we verify the local linear/energy stability properties discussed in Section 4 numerically. For this, we discretize Burgers' equation in the domain $(-1, 1)$ with periodic boundary conditions using upwind SBP methods with a fully upwind discretization using only D_-. To stress the methods, we consider a completely under-resolved case by computing the Jacobian at a random non-negative state using automatic/algorithmic differentiation via ForwardDiff.jl [77].

The results are shown in Table 12. Clearly, the maximal positive real part of the spectrum is around machine precision for 64 bit floating point numbers in all cases.



(a) Upwind SBP methods with interior order of accuracy 4 and classical DGSEM with polynomial degree $p = 2$.

(b) Upwind SBP methods with interior order of accuracy 6 and classical DGSEM with polynomial degree $p = 3$.

Fig. 2. Spectra of semidiscretizations of the 1D linear scalar advection equation with periodic boundary conditions. The maximum real part of all eigenvalues is around machine precision. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 12
Maximal real part of the spectrum of upwind SBP discretizations of Burgers' equation with different interior order of accuracy p, K elements, N nodes per element, and a purely upwind discretization using only D_-.

p	K	N	$\max \operatorname{Re} \sigma$
2	1	13	2.93×10^{-16}
2	1	14	-5.58×10^{-16}
2	2	13	2.62×10^{-15}
2	2	14	1.89×10^{-15}
3	1	13	3.08×10^{-17}
3	1	14	4.76×10^{-16}
3	2	13	5.51×10^{-16}
3	2	14	-2.78×10^{-16}
4	1	13	-4.67×10^{-16}
4	1	14	2.29×10^{-16}
4	2	13	-6.84×10^{-17}
4	2	14	3.39×10^{-16}
5	1	13	-1.67×10^{-16}
5	1	14	1.77×10^{-17}
5	2	13	2.11×10^{-16}
5	2	14	4.03×10^{-16}
6	1	13	-2.08×10^{-16}
6	1	14	-2.67×10^{-16}
6	2	13	-2.40×10^{-16}
6	2	14	-3.08×10^{-16}
7	1	13	1.65×10^{-16}
7	1	14	2.72×10^{-16}
7	2	13	4.33×10^{-16}
7	2	14	1.23×10^{-16}

5.5. Free-stream preservation on unstructured meshes

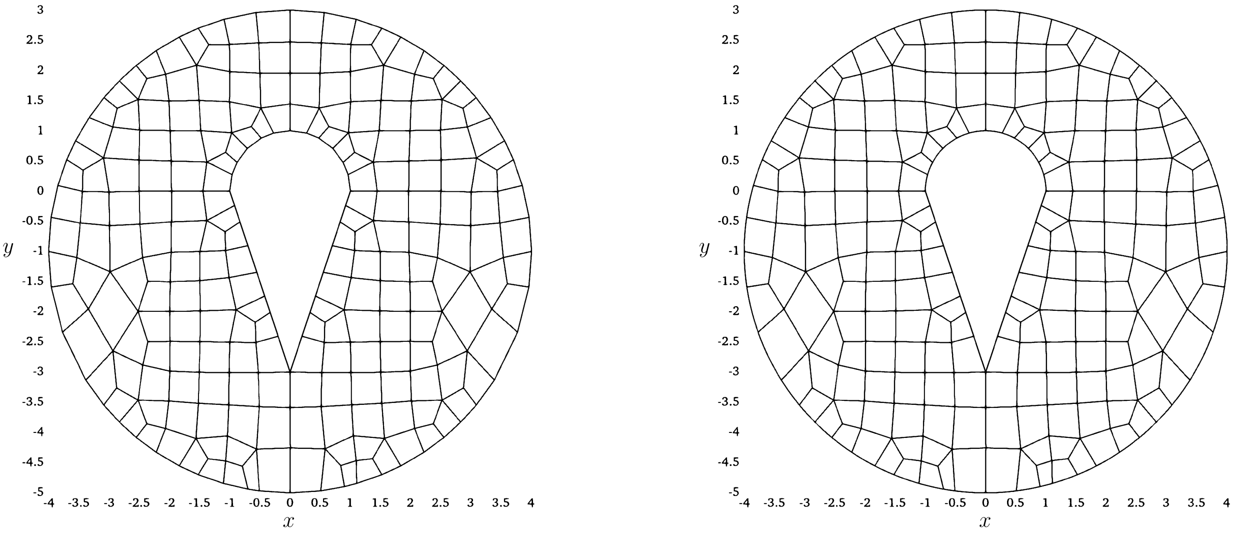
In this section, we present numerical evidence for the proof of free-stream preservation for the upwind SBP framework presented in Theorem 3.10. This theorem found that more complicated splittings, like the Drikakis-Tsangaris, for the upwind method in curvilinear coordinates are only guaranteed to be FSP under a particular interplay between the boundary (or interface) polynomial degree of an unstructured curvilinear mesh, the particular flux vector splitting, and the boundary closure order of the upwind SBP operator. The analysis in Section 3.4 also demonstrated that FSP is easily obtained provided the splitting technique remains linear as a function of the metrics terms, as was the case for the local Lax-Friedrichs splitting.

We reiterate that on a Cartesian box mesh, where all metric terms are constants proportional to fixed values of Δx or Δy, there is no issue with FSP. It is only when we move the approximation into generalized curvilinear coordinates that one must take care of the mapping, the metric terms, and their approximation strategy. The importance, and subtleties, of the discrete approximation of the metric terms has been known for decades across different computational fluid dynamics communities, e.g., [54–56].

Setting up an FSP test is straightforward and, at a glance, fairly innocuous. A constant solution should remain constant (up to machine precision) for all time as indicated by the governing equations (3.27) with appropriate boundary conditions. For the test herein we consider the compressible Euler equations in two space dimensions (5.7). Given the free-stream solution state of the conservative variables

$$u_\infty = \begin{pmatrix} \varrho_\infty \\ (\varrho v_1)_\infty \\ (\varrho v_2)_\infty \\ (\varrho e)_\infty \end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.1 \\ -0.2 \\ 10.0 \end{pmatrix}, \tag{5.11}$$

the fluxes are all constant and their divergence vanishes on the continuous level. However, in the discrete setting this is (potentially) not always the case.



(a) Mesh with linear boundary polynomials.

(b) Mesh with quadratic boundary polynomials.

Fig. 3. Non-overlapping quadrilateral meshes used for the free-stream preservation testing.

Table 13
Free-stream preservation error at final time 10 for the local Lax-Friedrichs splitting on two mesh types with 17 nodes in each spatial direction. As expected from the results in Section 3.4, the local Lax-Friedrichs splitting is FSP across all configurations.

interior order	2	3	4	5	6	7	8	9
bi-linear mesh	7.55×10^{-14}	1.16×10^{-13}	6.28×10^{-14}	6.99×10^{-15}	1.52×10^{-13}	1.16×10^{-13}	4.35×10^{-14}	3.89×10^{-14}
quadratic mesh	2.38×10^{-14}	4.72×10^{-14}	8.41×10^{-14}	5.48×10^{-14}	1.22×10^{-13}	1.47×10^{-14}	3.83×10^{-14}	4.18×10^{-14}

Table 14
Free-stream preservation error at final time 10 for the Drikakis-Tsangaris splitting on two mesh types with 17 nodes in each spatial direction. As shown from the result of Theorem 3.10, this splitting is FSP provided the boundary closure is accurate enough to exactly differentiate polynomials of degree $2N_{\text{geo}}$.

interior order	2	3	4	5	6	7	8	9
bi-linear mesh	2.04×10^{-6}	9.13×10^{-7}	3.34×10^{-14}	1.77×10^{-14}	3.34×10^{-14}	1.24×10^{-14}	8.20×10^{-14}	2.05×10^{-14}
quadratic mesh	2.08×10^{-6}	9.32×10^{-7}	5.74×10^{-9}	2.75×10^{-9}	7.09×10^{-11}	3.33×10^{-11}	2.18×10^{-14}	1.55×10^{-14}

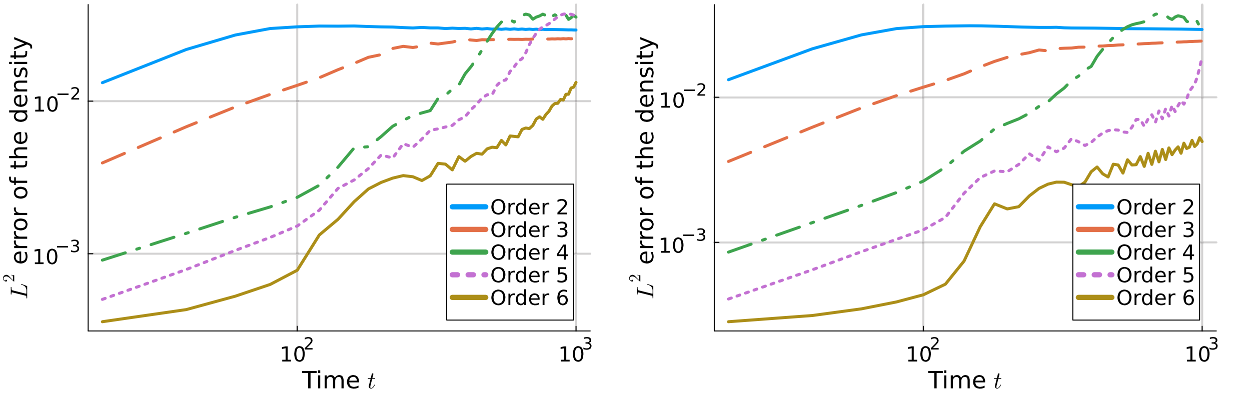
Table 15
Free-stream preservation error at final time 10 for the van Leer-Hänel splitting on two mesh types with 17 nodes in each spatial direction. As shown from the result of Theorem 3.10, this splitting is FSP provided the boundary closure is accurate enough to exactly differentiate polynomials of degree $2N_{\text{geo}}$.

interior order	2	3	4	5	6	7	8	9
bi-linear mesh	3.32×10^{-6}	1.46×10^{-6}	4.78×10^{-14}	5.26×10^{-15}	3.36×10^{-14}	1.02×10^{-14}	2.66×10^{-14}	1.95×10^{-14}
quadratic mesh	3.40×10^{-6}	1.49×10^{-6}	8.78×10^{-9}	4.20×10^{-9}	1.06×10^{-10}	4.98×10^{-11}	1.62×10^{-14}	5.43×10^{-14}

For the tests in this section we consider a domain with a circular outer boundary and an interior boundary composed of two straight lines and a semicircle. This domain is then divided into 204 non-overlapping quadrilateral elements. We create two versions of the mesh presented in Fig. 3: one composed only of bi-linear elements and the other with bi-linear elements in the interior and boundaries approximated with quadratic polynomials.

Whether or not a particular upwind SBP operator is FSP depends upon the splitting, the boundary closure accuracy, and the polynomial degree of the curvilinear boundary approximations in the mesh. We use the two meshes shown in Fig. 3 to examine the theoretical finding of Theorem 3.10 for different combinations of the upwind SBP operator, the flux vector splitting in generalized coordinates, and the polynomial degree of the mesh. For the FSP testing we fix the spatial resolution to be 17 nodes in each spatial direction and integrate the constant solution initial condition (5.11) up to a final time of 10. We present results of the FSP test in Tables 13–15 where we vary the curvilinear splitting and consider the upwind SBP operators provided by Mattsson [36] from interior order 2 up to interior order 9.

As anticipated from the discussion in Section 3.4 the local Lax-Friedrichs splitting maintains FSP for both meshes and all operator orders. The results also support the conclusion of Theorem 3.10 for the more complicated Drikakis-Tsangaris and van Leer-Hänel



(a) 4^2 blocks with 16^2 nodes each using the operators of [36] with different interior order of accuracy.

(b) A single block with 64^2 nodes and fully periodic operators using only the interior stencils of [36].

Fig. 4. Discrete L^2 of the density for long-time simulations of the isentropic vortex for the 2D compressible Euler equations.

splittings. Both of these splittings have a maximum quadratic dependence on the metric terms. We see that the operators with interior accuracy 2 and 3 (and boundary closure accuracy 1) are not FSP for either the Drikakis-Tsangaris nor van Lerr-Hänel splittings due to the lack of accuracy in the boundary closures. The numerical results show that FSP is maintained on the bi-linear test mesh for all operator orders above interior order 4 whereas FSP is only maintained on the quadratic test mesh for the 8-4 and 9-4 operators. All upwind SBP operator with interior order less than eight lack the required boundary closure accuracy to guarantee FSP on a quadratic mesh.

5.6. *Isentropic vortex*

We consider the classical isentropic vortex test case of [78] with initial conditions

$$\begin{aligned} T &= T_0 - \frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2}\exp\Big(1-r^2\Big), \quad \varrho = \varrho_0 (T/T_0)^{1/(\gamma-1)}, \\ v &= v_0 + \frac{\varepsilon}{2\pi}\exp\Big((1-r^2)/2\Big)(-x_2, x_1)^T, \end{aligned} \tag{5.12}$$

where $\varepsilon = 10$ is the vortex strength, r is the distance from the origin, $T = p/\varrho$ the temperature, $\varrho_0 = 1$ the background density, $v_0 = (1,1)^T$ the background velocity, $p_0 = 10$ the background pressure, $\gamma = 1.4$, and $T_0 = p_0/\varrho_0$ the background temperature. The domain $[-5,5]^2$ is equipped with periodic boundary conditions.

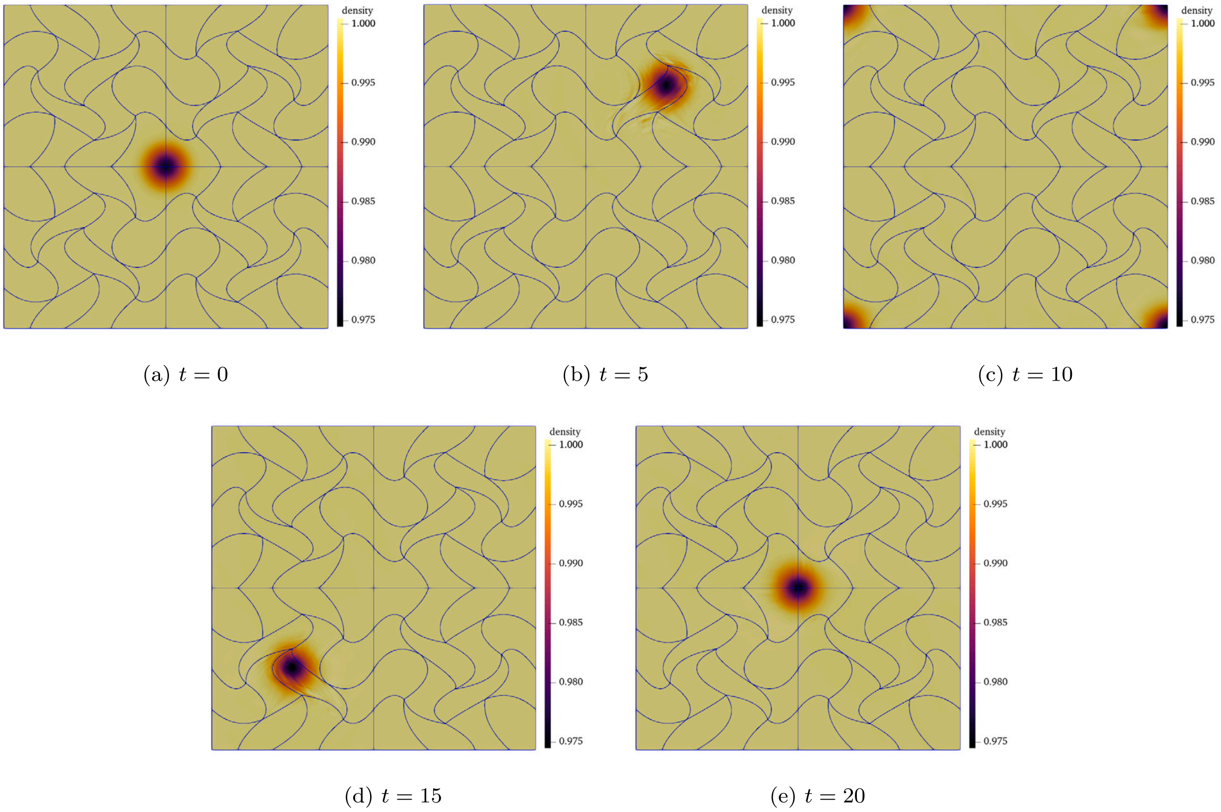
Following [9], we use this setup to demonstrate the long-time stability of the methods. We use the same time integration method and approach to compute the discrete L^2 error of the density as in Section 5.2. As shown in Fig. 4, the upwind methods remain stable and are able to run the simulations successfully for long times.

To demonstrate the robustness of the upwind methods on curvilinear meshes we, again, consider the isentropic vortex (5.12) with $\varepsilon = 5$ and $p_0 = 25$. We take $\Omega = [-10,10]^2$ and subdivide the domain with eight elements in each spatial direction for a total of 64 elements. The Cartesian domain of $\Omega = [-10,10]^2$ is then heavily warped with a strategy adapted from [79,80] where

$$\begin{aligned} y &= \eta + \frac{L_y}{8}\cos\left(\frac{3\pi}{2}\left(\frac{2\xi - L_x}{L_x}\right)\right)\cos\left[\frac{\pi}{2}\left(\frac{2\eta - L_y}{L_y}\right)\right], \\ x &= \xi + \frac{L_x}{8}\cos\left(\frac{\pi}{2}\left(\frac{2\xi - L_x}{L_x}\right)\right)\cos\left[2\pi\left(\frac{2y - L_y}{L_y}\right)\right], \end{aligned} \tag{5.13}$$

with $L_x = L_y = 10$. All domain boundaries remain periodic under this mapping. The resulting mesh, given as the overlay of curvilinear quadrilaterals in Fig. 5, is extremely distorted with many elements that approach degeneracy. That is, several elements in the mesh are close to having an internal angle near 180 degrees that renders the transfinite interpolation procedure to create the element mapping ill-conditioned. This extreme warping to a “poor” quality mesh was purposely done to demonstrate that the upwind methods remain robust for the isentropic vortex test case even in this extreme problem setup.

We approximate the internal curved boundaries with quadratic polynomials. On each element we take 17 nodes in each spatial direction and use the upwind SBP operator that is 8th order in the interior with 4th order boundary closures. We present in Fig. 5 the results at different times between $t = 0$ and $t = 20$ using the curvilinear van Leer-Hänel splitting from Example 3.8. Although there are some grid artifacts as the vortex passes through extremely distorted elements, the method maintains the shape of the vortex well. In Fig. 6 we show the L^2 density error for a long time simulation for the curvilinear local Lax-Friedrichs, Drikakis-Tsangaris, and van



(a) $t = 0$

(b) $t = 5$

(c) $t = 10$

(d) $t = 15$

(e) $t = 20$

Fig. 5. Isentropic vortex evolution up to final time $t = 20$ on a heavily distorted quadrilateral mesh of 64 elements. All curvilinear interior boundaries of the warping (5.13) are approximated with quadratic polynomials. This result used the curvilinear van Leer-Hänel splitting, Example 3.8, with 17 nodes in each spatial direction and the 8th order interior, 4th order boundary accurate upwind SBP operators.

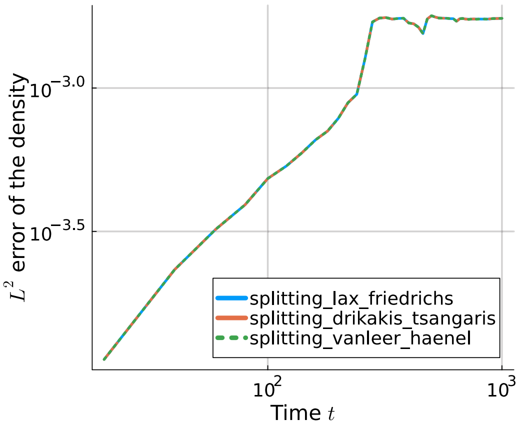


Fig. 6. Discrete L^2 of the density for long-time simulation of three curvilinear splittings applied to the isentropic vortex for the 2D compressible Euler equations. Each run used the same heavily distorted quadrilateral mesh.

Leer-Hänel splittings. All three splittings of this heavily distorted curvilinear mesh remain stable. Because the test case configuration is well-resolved, any differences in density errors between the three splitting techniques are unnoticeable in the eyeball norm.

5.7. *Kelvin-Helmholtz instability*

Next, we move beyond well-resolved test configurations and use a Kelvin-Helmholtz instability setup for the 2D compressible Euler equations of an ideal fluid to further test the robustness of the methods and their different splittings in various under-resolved regimes in detail. Specifically, we use the same setup as in [7], i.e., the initial condition

Table 16
Final times of numerical simulations of the Kelvin-Helmholtz instability with K elements using 16 nodes per coordinate direction for the upwind SBP methods. Final times less than 15 indicate that the simulation crashed.

(a) Upwind SBP, van Leer-Hänel splitting [45–47].

K	interior order of accuracy					
	2	3	4	5	6	7
1	15.0	15.0	15.0	15.0	15.0	15.0
4	15.0	15.0	15.0	15.0	15.0	15.0
16	15.0	15.0	15.0	15.0	4.72	3.97
64	15.0	15.0	4.53	3.86	4.17	3.36
256	15.0	15.0	5.80	3.70	3.66	3.68

(b) Upwind SBP, Steger-Warming splitting [44].

K	interior order of accuracy					
	2	3	4	5	6	7
1	15.0	15.0	15.0	15.0	15.0	15.0
4	15.0	15.0	15.0	15.0	15.0	15.0
16	15.0	15.0	15.0	15.0	4.87	3.88
64	15.0	15.0	4.55	3.85	4.13	4.07
256	15.0	15.0	5.80	3.69	3.66	3.67

(c) Flux differencing DGSEM, flux of Ranocha [62,66,83].

K	polynomial degree					
	2	3	4	5	6	7
16	15.0	4.46	2.47	3.01	2.80	3.59
64	4.68	1.53	4.04	3.70	4.10	3.56
256	4.81	3.77	4.44	3.74	3.37	3.64
1024	4.12	3.66	4.27	3.54	3.66	3.56

(d) Flux differencing DGSEM, flux of Shima et al. [84].

K	polynomial degree					
	2	3	4	5	6	7
16	15.0	2.73	1.81	2.42	1.86	2.27
64	2.92	1.38	3.05	3.07	1.82	2.02
256	3.25	2.82	3.29	2.82	2.84	2.96
1024	3.03	2.88	3.36	2.91	3.08	3.25

Table 17
Final times of numerical simulations of the Kelvin-Helmholtz instability with K elements using $256/\sqrt{K}$ nodes per coordinate direction so that the total number of DOFs stays fixed at 65 536. Final times less than 15 indicate that the simulation crashed.

(a) Upwind SBP, van Leer-Hänel splitting [45–47].

K	interior order of accuracy					
	2	3	4	5	6	7
1	15.0	15.0	15.0	15.0	5.83	4.73
4	15.0	15.0	6.35	15.0	5.83	4.72
16	15.0	15.0	15.0	15.0	5.18	4.00
64	15.0	15.0	15.0	5.80	4.37	3.99
256	15.0	15.0	5.80	3.70	3.66	3.68

(b) Upwind SBP, Steger-Warming splitting [44].

K	interior order of accuracy					
	2	3	4	5	6	7
1	15.0	15.0	15.0	15.0	5.86	4.80
4	15.0	15.0	15.0	15.0	15.0	4.79
16	15.0	15.0	15.0	15.0	5.43	4.03
64	15.0	15.0	15.0	5.68	4.36	4.02
256	15.0	15.0	5.80	3.69	3.66	3.67

$$\varrho = \frac{1}{2} + \frac{3}{4} B(x, y), \quad p = 1, \quad v_1 = \frac{1}{2}\Big(B(x, y) - 1\Big), \quad v_2 = \frac{1}{10}\sin(2\pi x), \tag{5.14}$$

where $B(x, y)$ is the smoothed approximation

$$B(x, y) = \tanh(15y + 7.5) - \tanh(15y - 7.5) \tag{5.15}$$

to a discontinuous step function. The domain is $[-1, 1]^2$ with time interval $[0, 15]$. We integrate the semidiscretizations in time with the third-order, four-stage SSP method of [81] with embedded method of [82] and error-based step size controller developed in [76] with absolute and relative tolerances chosen as 10^{-6}.

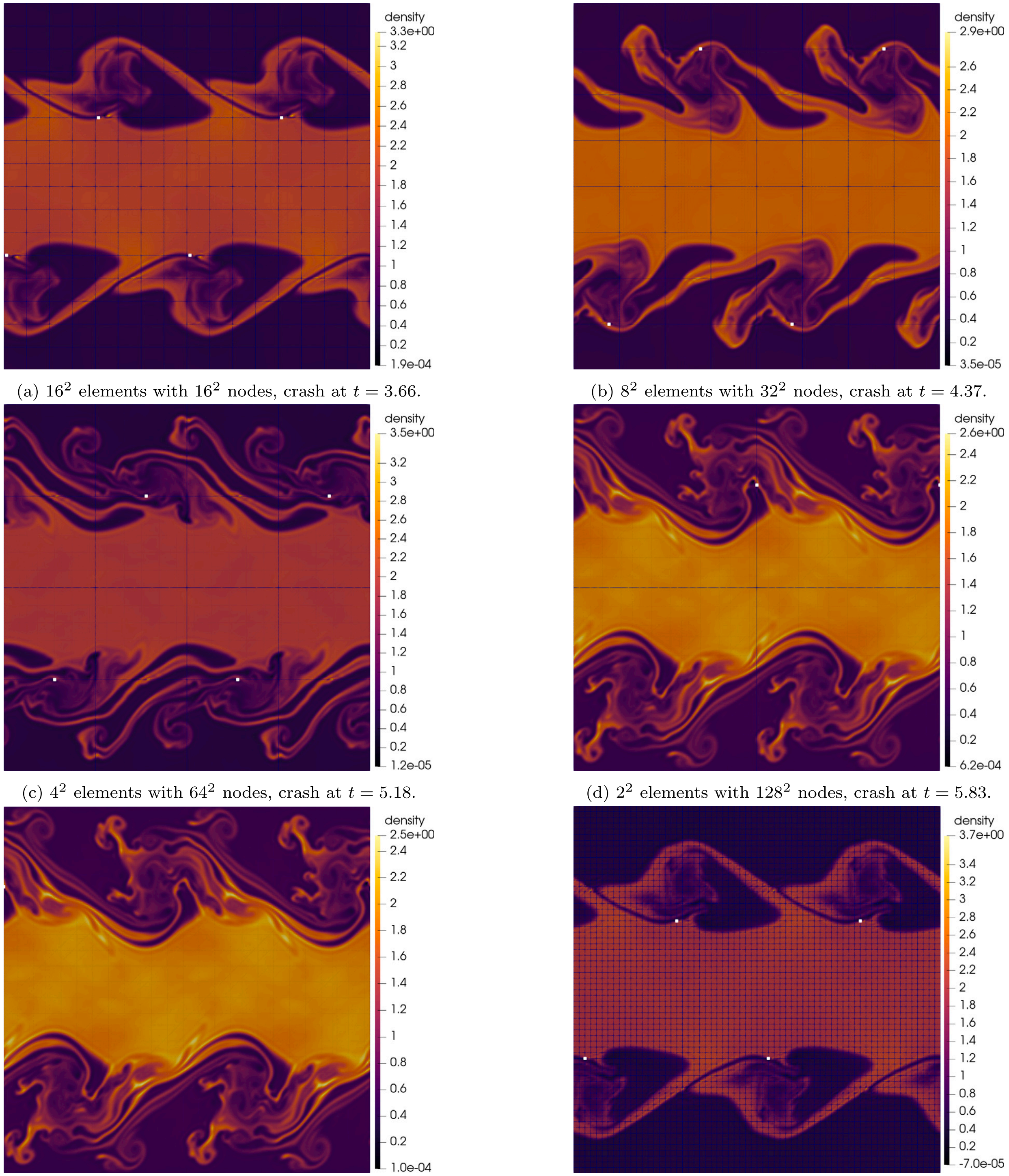
We use two types of semidiscretizations: i) the upwind SBP methods described in this article and ii) flux differencing DGSEM with different volume fluxes and a local Lax-Friedrichs (Rusanov) surface flux. We refer to [64] for a description of this DGSEM variant and its efficient implementation and to [85] for convergence results.

The final times of the simulations are summarized in Table 16. First, it is interesting to observe that all upwind SBP methods with few numbers of elements $K \in \{1, 4\}$ completed the simulation successfully. The same is true for low interior orders of accuracy $\in \{2, 3\}$. However, setups with more elements and higher orders of accuracy became unstable and crashed before $t = 5$.

The same general trend can be observed for flux differencing DGSEM, where nearly all setups became unstable and crashed. It is particularly interesting that the simulations with only a few elements remained stable, even if their total number of DOFs is comparable to DGSEM setups. For example, the upwind SBP methods with $K = 4$ elements and 16 nodes per coordinate direction have $K \cdot 16^2 = 1024$ DOFs, the same amount as the DGSEM with $K = 64$ elements and a polynomial degree $p = 3$.

To further investigate this behavior, we ran additional simulations with upwind SBP operators. Here, we choose the number of nodes such that the total number of DOFs remains constant. The resulting final simulation times are shown in Table 17. As in the case of a constant number of nodes per element investigated before, increasing the number of elements makes the upwind FD methods less robust. The only exceptions are again the low-order methods with an interior order of accuracy two and three (both resulting in an experimental order of convergence of two under mesh refinement by increasing the number of elements).

Fig. 7 shows the numerical solutions corresponding to the constant DOF setup of Table 17 at the time the simulations crashed. As usual, we plot the density of the numerical solutions to allow a comparison with other publications. The white spots mark the points where the pressure is negative (for the upwind SBP methods) or where the density is negative (for the DGSEM). Clearly, the problematic nodes are always located at interfaces between elements. This appears to be correlated with the earlier crash times of simulations with more elements and fewer nodes per element. Note that the periodic boundary conditions are enforced weakly. Thus,



(a) 16^2 elements with 16^2 nodes, crash at $t = 3.66$.

(b) 8^2 elements with 32^2 nodes, crash at $t = 4.37$.

(c) 4^2 elements with 64^2 nodes, crash at $t = 5.18$.

(d) 2^2 elements with 128^2 nodes, crash at $t = 5.83$.

(e) 1 element with 256^2 nodes, crash at $t = 5.83$.

(f) DGSEM, 64^2 elements with $p = 3$, crash at $t = 3.66$.

Fig. 7. Visualization of numerical solutions when the simulations of the Kelvin-Helmholtz instability crashed. All simulations use the same number of DOFs — with varying numbers of elements/nodes for the upwind SBP methods with an interior order of accuracy 6 using the van Leer-Hänel splitting [45–47]. For comparison, results obtained by entropy-stable flux differencing DGSEM are also shown. The white spots mark points where the pressure (upwind SBP) or the density (DGSEM) is negative.

Table 18
Final times of numerical simulations of the Kelvin-Helmholtz instability with a single element using purely periodic upwind methods, i.e., only the interior stencils of the corresponding upwind SBP operators. Final times less than 15 indicate that the simulation crashed.

(a) Van Leer-Hänel splitting [45–47].

#nodes	interior order of accuracy					
	2	3	4	5	6	7
256	15.0	15.0	15.0	15.0	15.0	15.0
1024	15.0	15.0	15.0	15.0	15.0	15.0
4096	15.0	15.0	15.0	15.0	15.0	15.0
16 384	15.0	15.0	15.0	15.0	15.0	15.0
65 536	15.0	15.0	15.0	15.0	15.0	4.77

(b) Steger-Warming splitting [44].

#nodes	interior order of accuracy					
	2	3	4	5	6	7
256	15.0	15.0	15.0	15.0	15.0	15.0
1024	15.0	15.0	15.0	15.0	15.0	15.0
4096	15.0	15.0	15.0	15.0	15.0	15.0
16 384	15.0	15.0	15.0	15.0	15.0	15.0
65 536	15.0	15.0	15.0	15.0	15.0	15.0

Table 19
Final times of numerical simulations of the inviscid Taylor-Green vortex with $\mathrm{Ma} = 0.1$, K elements, and upwind SBP methods using the Steger-Warming splitting [44]. Final times less than 20 indicate that the simulation crashed.

(a) Constant number of nodes per direction ($= 16$).

K	interior order of accuracy					
	2	3	4	5	6	7
1	20.0	20.0	20.0	20.0	20.0	20.0
8	20.0	20.0	20.0	20.0	20.0	13.5
64	20.0	20.0	20.0	20.0	20.0	6.12

(b) Constant number of DOFs ($= 262\,144$).

K	interior order of accuracy					
	2	3	4	5	6	7
1	20.0	20.0	20.0	20.0	20.0	6.1
4	20.0	20.0	20.0	20.0	20.0	15.0
16	20.0	20.0	20.0	20.0	20.0	6.12

Table 20
Final times of numerical simulations of the inviscid Taylor-Green vortex with $\mathrm{Ma} = 0.4$, K elements, and upwind SBP methods using the Steger-Warming splitting [44] with a constant number of nodes per direction ($= 16$). Final times less than 20 indicate that the simulation crashed.

K	interior order of accuracy					
	2	3	4	5	6	7
1	20.0	20.0	20.0	20.0	5.79	5.31
8	20.0	20.0	20.0	4.79	4.00	3.89
64	20.0	20.0	8.40	5.70	4.42	4.18

even the setup with a single element has internal interfaces — and the negative pressure occurs exactly at one of these boundary points (top left quadrant).

To investigate this claim, we considered purely periodic upwind methods using only the interior coefficients of the upwind SBP operators. As shown in Table 18, this version is much more robust. Indeed, all of the simulations ran successfully except the van Leer-Hänel splitting with an interior order of accuracy 7 and $256^2 = 65\,536$ nodes in total.

5.8. Inviscid Taylor-Green vortex

Next, we consider the classical inviscid Taylor-Green vortex for the 3D compressible Euler equations of an ideal gas following [5]. Specifically, we consider the initial conditions

$$\begin{aligned} &\varrho = 1,\ v_1 = \sin(x_1)\cos(x_2)\cos(x_3),\ v_2 = -\cos(x_1)\sin(x_2)\cos(x_3),\ v_3 = 0, \\ p = {}&\frac{\varrho^0}{\mathrm{Ma}^2\gamma} + \varrho^0 \frac{\cos(2x_1)\cos(2x_3) + 2\cos(2x_2) + 2\cos(2x_1) + \cos(2x_2)\cos(2x_3)}{16}, \end{aligned} \tag{5.16}$$

where $\mathrm{Ma} = 0.1$ is the Mach number. We consider the domain $[-\pi, \pi]^3$ with periodic boundary conditions and a time interval $[0, 20]$. We integrate the semidiscretizations in time with the third-order, four-stage SSP method of [81] with embedded method of [82] and error-based step size controller developed in [76] with absolute and relative tolerance chosen as 10^{-6}.

The final times of simulations using upwind SBP operators are shown in Table 19. For 16 nodes per coordinate direction, only the method with an interior order of accuracy 7 crashed — all lower-order methods completed the full simulation.

Next, we repeat the numerical robustness experiments with an increased Mach number $\mathrm{Ma} = 0.4$. The results are shown in Table 20. The increased Mach number introduces more compressibility effects, testing the robustness of the numerical methods in another regime. For the upwind SBP methods considered here, this leads to a reduced numerical robustness. Indeed, only the upwind SBP methods with an interior order of accuracy two and three complete all simulations. The higher-order methods crash for increased resolution. These results are comparable to the robustness results we observed for the Kelvin-Helmholtz instability in Section 5.7.

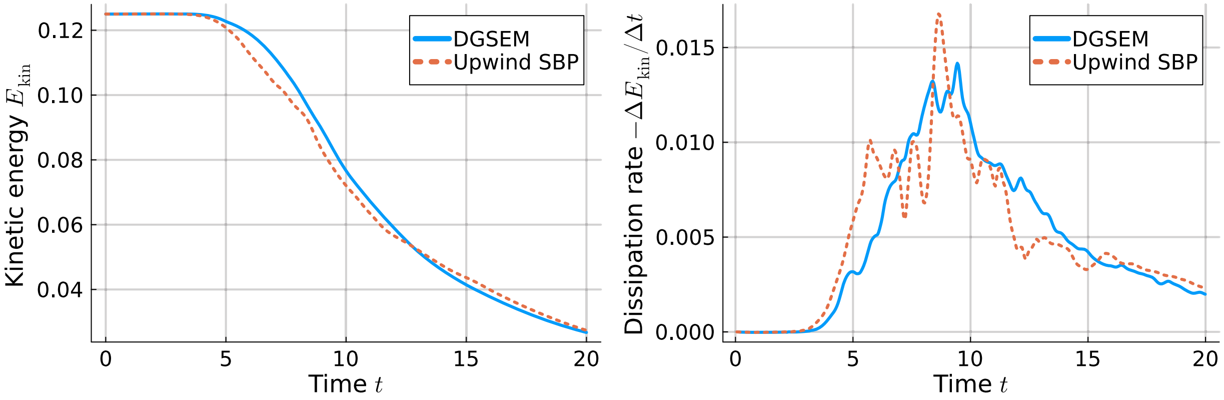
Table 21
Final times of numerical simulations of the inviscid Taylor-Green vortex with a single element and periodic upwind SBP methods using the Steger-Warming splitting [44]. Final times less than 20 indicate that the simulation crashed.

(a) Mach number $\mathrm{Ma} = 0.1$.

#nodes	interior order of accuracy					
	2	3	4	5	6	7
4096	20.0	20.0	20.0	20.0	20.0	20.0
32 768	20.0	20.0	20.0	20.0	20.0	20.0
262 144	20.0	20.0	20.0	20.0	20.0	20.0

(b) Mach number $\mathrm{Ma} = 0.4$.

#nodes	interior order of accuracy					
	2	3	4	5	6	7
4096	20.0	20.0	20.0	20.0	20.0	20.0
32 768	20.0	20.0	20.0	20.0	20.0	20.0
262 144	20.0	20.0	20.0	20.0	20.0	20.0



(a) Discrete kinetic energy.

(b) Discrete kinetic energy dissipation rate.

Fig. 8. Discrete kinetic energy and its dissipation rate for the inviscid Taylor-Green vortex. We compare the results of flux differencing DGSEM with upwind SBP methods. The DGSEM scheme uses 16 elements per coordinate direction, the entropy-conservative flux of Ranocha [62,66,83] in the volume, and the local Lax-Friedrichs (Rusanov) flux at interfaces. The upwind SBP method uses 4 elements and 16 nodes per coordinate direction, the Steger-Warming splitting [44], and the operators of Mattsson [36] with an interior order of accuracy 6. Thus, both simulations use 262 144 DOFs.

Table 21 shows results obtained by fully periodic upwind SBP methods using only the interior coefficients of the upwind SBP operators. As for the Kelvin-Helmholtz instability considered before, this version is much more robust — all of the simulations run successfully to the final time.

Next, we follow the approach of [5] to compute the kinetic energy dissipation rate for the Mach number $\mathrm{Ma} = 0.1$. Specifically, we compute the discrete version of the total kinetic energy

$$E_{\mathrm{kin}}(t) = \int \frac{1}{2}\varrho(t,x)v(t,x)^2 \,\mathrm{d}x \tag{5.17}$$

using the quadrature rule associated with the SBP mass matrix M every 10 accepted time steps. Then, we use central finite differences to compute the discrete kinetic energy dissipation rate $-\Delta E_{\mathrm{kin}}/\Delta t$ approximating $-\,\mathrm{d}E_{\mathrm{kin}}/\,\mathrm{d}t$. The results are visualized in Fig. 8. The qualitative behavior of the kinetic energy and its dissipation rate is the same for the flux differencing DGSEM method and the upwind SBP method. The upwind SBP method tends to begin dissipating the kinetic energy earlier than the DGSEM and shows a dissipation rate that is a bit more oscillatory. The results of the upwind SBP method do not change if we use half the number of elements but double the number of nodes per element (not shown in the plots). The results of the flux differencing DGSEM simulation match the results of [5] for the same polynomial degree and resolution (up to the smaller final time $T = 14$ used there).

We also measured the execution time of the upwind SBP method and the flux differencing DGSEM used in this example on a MacBook with an Apple M2 CPU. The total time spent in the ODE right-hand side computation to simulate the Taylor-Green vortex in the time interval $[0, 1]$ on a single thread without any parallelism is roughly 12.34(1) s for the flux differencing DGSEM and 13.73(11) s for the upwind SBP method (results of five runs, average and standard deviation, same setup as used in Section 5.8). Please note that this compares a highly tuned implementation of the DGSEM using SIMD instructions as described in [64] with a first implementation of the upwind SBP methods in a research code. Thus, we conclude that both methods are of comparable efficiency.

6. Summary and conclusions

We have discussed high-order upwind SBP methods for nonlinear conservation laws. Introduced by Mattsson in [36], these methods combine central-type classical SBP operators with artificial dissipation and need a flux vector splitting for nonlinear conservation laws. Lax-Friedrichs type splittings have been predominantly considered in the literature [34–37,39]. To combine upwind SBP operators with multiple other flux vector splittings, we have described a general way to design SATs as in discontinuous Galerkin methods using numerical fluxes resulting from the chosen splitting in Section 3. Further, we discussed how to extend splittings other than those of Lax-Friedrichs type into the high-order upwind SBP framework of Mattsson on unstructured curvilinear meshes. Through this analysis we found an interplay between the dependency of said splittings, like the van Leer-Hänel, on the metric terms and the boundary closure accuracy of the upwind SBP operator. Only under specific conditions on the mapping, the metric terms, and the boundary closure could the resulting method retain the important free-stream preservation property in generalized coordinates. We have proven the local linear/energy stability of upwind SBP methods for Burgers' equation in Section 4. This kind of stability property is not the classical stability property of a numerical method applied to a linearized problem, but a property of the linearization thereof applied to a nonlinear problem. Since linearization and application of a high-order method for conservation laws do not commute in general, it is nontrivial to satisfy stability properties such as entropy stability for the nonlinear problem and local linear/energy stability at the same time. In particular, we are not aware of any numerical method that has all three desirable properties i) nonlinear entropy stability, ii) local linear/energy stability, and iii) high-order accuracy. Methods based on classical SBP operators can be designed to be high-order accurate and entropy-stable but lack local linear/energy stability [10]. We have complemented these results by proving that high-order upwind SBP methods satisfy local linear/energy stability. We have also discussed the relation to a very special case of entropy stability. While this case is only an academic example, we hope that it may lead the community in a way to solve the entropy/linear stability issue.

We have applied upwind SBP methods with several flux vector splittings in Section 5. The robustness and computational efficiency of the upwind SBP methods for nonlinear conservation laws are roughly comparable to highly tuned flux differencing discontinuous Galerkin spectral element methods, as demonstrated for several examples of compressible fluid flows and under-resolved simulations. The numerical tests demonstrated that the upwind SBP methods remained high-order accurate on unstructured curvilinear domains and free-stream preservation was retained provided any curved boundaries were approximated with an appropriate polynomial order dictated by the boundary closure accuracy of a given SBP operator. These validation tests were performed on well-resolved simulation setups. For under-resolved simulations, we have shown that results for a classical inviscid Taylor-Green vortex are promising, but more challenging tests such as a Kelvin-Helmholtz instability show that upwind SBP methods do not fix all high-order robustness issues for shock-free flows. In particular, some robustness (positivity) issues manifest mainly at interfaces and corners. Thus, upwind SBP methods are roughly comparable to other modern stabilizations for high-order schemes.

CRediT authorship contribution statement

Hendrik Ranocha: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Andrew R. Winters:** Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Conceptualization. **Michael Schlottke-Lakemper:** Writing – review & editing, Visualization, Validation, Software, Methodology, Data curation, Conceptualization. **Philipp Öffner:** Writing – review & editing, Methodology, Conceptualization. **Jan Glaubitz:** Writing – review & editing, Methodology, Conceptualization. **Gregor J. Gassner:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All code is available online in our reproducibility repository [75] available at https://zenodo.org/doi/10.5281/zenodo.10200102 and https://github.com/trixi-framework/paper-2023-upwind.

Acknowledgements

HR was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation, project numbers 513301895 and 528753982 as well as within the DFG priority program SPP 2410 with project number 526031774) and the Daimler und Benz Stiftung (Daimler and Benz foundation, project number 32-10/22). ARW was funded through Vetenskapsrådet, Sweden grant agreement 2020-03642 VR. MSL received funding through the DFG research unit FOR 5409 "Structure-Preserving Numerical Methods for Bulk- and Interface Coupling of Heterogeneous Models (SNuBIC)" (project number 463312734), as well as through a DFG individual grant (project number 528753982). PÖ was supported by the DFG within the priority research program SPP 2410, project OE 661/5-1 (525866748) and under the personal grant 520756621 (OE 661/4-1). JG was supported by the US DOD (ONR MURI) grant #N00014-20-1-2595. GG acknowledges funding through the Klaus-Tschira Stiftung via the project "HiFiLab" (grant number 00.014.2021) and received funding through the DFG research unit FOR 5409 "SNUBIC" and through the BMBF funded project "ADAPTEX" (grant number 16ME0668K).

Some of the computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

References

[1] E. Tadmor, The numerical viscosity of entropy stable schemes for systems of conservation laws. I, Math. Comput. 49 (1987) 91–103, https://doi.org/10.1090/S0025-5718-1987-0890255-3.

[2] E. Tadmor, Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems, Acta Numer. 12 (2003) 451–512, https://doi.org/10.1017/S0962492902000156.

[3] P.G. LeFloch, J.-M. Mercier, C. Rohde, Fully discrete, entropy conservative schemes of arbitrary order, SIAM J. Numer. Anal. 40 (2002) 1968–1992, https://doi.org/10.1137/S003614290240069X.

[4] T.C. Fisher, M.H. Carpenter, High-order entropy stable finite difference schemes for nonlinear conservation laws: finite domains, J. Comput. Phys. 252 (2013) 518–557, https://doi.org/10.1016/j.jcp.2013.06.014.

[5] G.J. Gassner, A.R. Winters, D.A. Kopriva, Split form nodal discontinuous Galerkin schemes with summation-by-parts property for the compressible Euler equations, J. Comput. Phys. 327 (2016) 39–66, https://doi.org/10.1016/j.jcp.2016.09.013.

[6] D. Rojas, R. Boukharfane, L. Dalcin, D.C.D.R. Fernández, H. Ranocha, D.E. Keyes, M. Parsani, On the robustness and performance of entropy stable discontinuous collocation methods, J. Comput. Phys. 426 (2021) 109891, https://doi.org/10.1016/j.jcp.2020.109891, arXiv:1911.10966.

[7] J. Chan, H. Ranocha, A.M. Rueda-Ramirez, G.J. Gassner, T. Warburton, On the entropy projection and the robustness of high order entropy stable discontinuous Galerkin schemes for under-resolved flows, Front. Phys. 10 (2022), https://doi.org/10.3389/fphy.2022.898028, arXiv:2203.10238.

[8] B.F. Klose, G.B. Jacobs, D.A. Kopriva, Assessing standard and kinetic energy conserving volume fluxes in discontinuous Galerkin formulations for marginally resolved Navier-Stokes flows, Comput. Fluids 205 (2020) 104557, https://doi.org/10.1016/j.compfluid.2020.104557.

[9] B. Sjögreen, H. Yee, High order entropy conservative central schemes for wide ranges of compressible gas dynamics and MHD flows, J. Comput. Phys. 364 (2018) 153–185, https://doi.org/10.1016/j.jcp.2018.02.003.

[10] G.J. Gassner, M. Svärd, F.J. Hindenlang, Stability issues of entropy-stable and/or split-form high-order schemes, J. Sci. Comput. 90 (2022) 1–36, https://doi.org/10.1007/s10915-021-01720-8, arXiv:2007.09026.

[11] A.M. Rueda-Ramirez, B. Bolm, D. Kuzmin, G.J. Gassner, Monolithic convex limiting for Legendre-Gauss-Lobatto discontinuous Galerkin spectral element methods, Commun. Appl. Math. Comput. 6 (2024) 1860–1898, https://doi.org/10.1007/s42967-023-00321-6.

[12] W. Pazner, Sparse invariant domain preserving discontinuous Galerkin methods with subcell convex limiting, Comput. Methods Appl. Mech. Eng. 382 (2021) 113876, https://doi.org/10.1016/j.cma.2021.113876.

[13] M. Maier, M. Kronbichler, Efficient parallel 3D computation of the compressible Euler equations with an invariant-domain preserving second-order finite-element scheme, ACM Trans. Parallel Comput. 8 (2021) 1–30, https://doi.org/10.1145/3470637.

[14] J.-L. Guermond, M. Kronbichler, M. Maier, B. Popov, I. Tomas, On the implementation of a robust and efficient finite element-based parallel solver for the compressible Navier-Stokes equations, Comput. Methods Appl. Mech. Eng. 389 (2022) 114250, https://doi.org/10.1016/j.cma.2021.114250, arXiv:2106.02159.

[15] A.M. Rueda-Ramirez, G.J. Gassner, A subcell finite volume positivity-preserving limiter for DGSEM discretizations of the Euler equations, https://doi.org/10.48550/arXiv.2102.06017, arXiv:2102.06017, 2021.

[16] H.-O. Kreiss, G. Scherer, Finite element and finite difference methods for hyperbolic partial differential equations, in: C. de Boor (Ed.), Mathematical Aspects of Finite Elements in Partial Differential Equations, Academic Press, New York, 1974, pp. 195–212.

[17] B. Strand, Summation by parts for finite difference approximations for d/dx, J. Comput. Phys. 110 (1994) 47–67, https://doi.org/10.1006/jcph.1994.1005.

[18] J. Nordström, M. Björck, Finite volume approximations and strict stability for hyperbolic problems, Appl. Numer. Math. 38 (2001) 237–255, https://doi.org/10.1016/S0168-9274(01)00027-7.

[19] J. Nordström, K. Forsberg, C. Adamsson, P. Eliasson, Finite volume methods, unstructured meshes and strict stability for hyperbolic problems, Appl. Numer. Math. 45 (2003) 453–473, https://doi.org/10.1016/S0168-9274(02)00239-8.

[20] J.E. Hicken, D.C.D.R. Fernández, D.W. Zingg, Multidimensional summation-by-parts operators: general theory and application to simplex elements, SIAM J. Sci. Comput. 38 (2016) A1935–A1958, https://doi.org/10.1137/15M1038360.

[21] J.E. Hicken, Entropy-stable, high-order summation-by-parts discretizations without interface penalties, J. Sci. Comput. 82 (2020) 50, https://doi.org/10.1007/s10915-020-01154-8.

[22] R. Abgrall, J. Nordström, P. Öffner, S. Tokareva, Analysis of the SBP-SAT stabilization for finite element methods part I: linear problems, J. Sci. Comput. 85 (2020) 1–29, https://doi.org/10.1007/s10915-020-01349-z, arXiv:1912.08108.

[23] R. Abgrall, J. Nordström, P. Öffner, S. Tokareva, Analysis of the SBP-SAT stabilization for finite element methods. II: entropy stability, Commun. Appl. Math. Comput. 5 (2023) 573–595, https://doi.org/10.1007/s42967-020-00086-2.

[24] G.J. Gassner, A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods, SIAM J. Sci. Comput. 35 (2013) A1233–A1253, https://doi.org/10.1137/120890144.

[25] M.H. Carpenter, T.C. Fisher, E.J. Nielsen, S.H. Frankel, Entropy stable spectral collocation schemes for the Navier-Stokes equations: discontinuous interfaces, SIAM J. Sci. Comput. 36 (2014) B835–B867, https://doi.org/10.1137/130932193.

[26] J. Chan, On discretely entropy conservative and entropy stable discontinuous Galerkin methods, J. Comput. Phys. 362 (2018) 346–374, https://doi.org/10.1016/j.jcp.2018.02.033.

[27] H.T. Huynh, A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods, in: 18th AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, 2007, p. 4079.

[28] H. Ranocha, P. Öffner, T. Sonar, Summation-by-parts operators for correction procedure via reconstruction, J. Comput. Phys. 311 (2016) 299–328, https://doi.org/10.1016/j.jcp.2016.02.009, arXiv:1511.02052.

[29] M. Svärd, J. Nordström, Review of summation-by-parts schemes for initial-boundary-value problems, J. Comput. Phys. 268 (2014) 17–38, https://doi.org/10.1016/j.jcp.2014.02.031.

[30] D.C.D.R. Fernández, J.E. Hicken, D.W. Zingg, Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations, Comput. Fluids 95 (2014) 171–196, https://doi.org/10.1016/j.compfluid.2014.02.016.

[31] M.H. Carpenter, D. Gottlieb, S. Abarbanel, Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes, J. Comput. Phys. 111 (1994) 220–236, https://doi.org/10.1006/jcph.1994.1057.

[32] M.H. Carpenter, J. Nordström, D. Gottlieb, A stable and conservative interface treatment of arbitrary spatial accuracy, J. Comput. Phys. 148 (1999) 341–365, https://doi.org/10.1006/jcph.1998.6114.

[33] K. Mattsson, M. Svärd, J. Nordström, Stable and accurate artificial dissipation, J. Sci. Comput. 21 (2004) 57–79, https://doi.org/10.1023/B:JOMP.0000027955.75872.3f.

[34] M. Svärd, K. Mattsson, J. Nordström, Steady-state computations using summation-by-parts operators, J. Sci. Comput. 24 (2005) 79–95, https://doi.org/10.1007/s10915-004-4788-2.

[35] K. Mattsson, M. Svärd, M. Carpenter, J. Nordström, High-order accurate computations for unsteady aerodynamics, Comput. Fluids 36 (2007) 636–649, https://doi.org/10.1016/j.compfluid.2006.02.004.

[36] K. Mattsson, Diagonal-norm upwind SBP operators, J. Comput. Phys. 335 (2017) 283–310, https://doi.org/10.1016/j.jcp.2017.01.042.

[37] L. Lundgren, K. Mattsson, An efficient finite difference method for the shallow water equations, J. Comput. Phys. 422 (2020) 109784, https://doi.org/10.1016/j.jcp.2020.109784.

[38] Y. Rydin, K. Mattsson, J. Werpers, High-fidelity sound propagation in a varying 3D atmosphere, J. Sci. Comput. 77 (2018) 1278–1302, https://doi.org/10.1007/s10915-018-0751-5.

[39] V. Stiernström, L. Lundgren, M. Nazarov, K. Mattsson, A residual-based artificial viscosity finite difference method for scalar conservation laws, J. Comput. Phys. 430 (2021) 110100, https://doi.org/10.1016/j.jcp.2020.110100.

[40] K. Mattsson, O. O'Reilly, Compatible diagonal-norm staggered and upwind SBP operators, J. Comput. Phys. 352 (2018) 52–75, https://doi.org/10.1016/j.jcp.2017.09.044.

[41] H. Ranocha, D. Mitsotakis, D.I. Ketcheson, A broad class of conservative numerical methods for dispersive wave equations, Commun. Comput. Phys. 29 (2021) 979–1029, https://doi.org/10.4208/cicp.OA-2020-0119, arXiv:2006.14802.

[42] S. Ortleb, On the stability of IMEX upwind gSBP schemes for 1D linear advection-diffusion equations, Commun. Appl. Math. Comput. (2023) 1–30, https://doi.org/10.1007/s42967-023-00296-4.

[43] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction, Springer, Berlin Heidelberg, 2009.

[44] J.L. Steger, R. Warming, Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite Difference Methods, Technical Memorandum NASA/TM-78605, NASA, NASA Ames Research, Center, Moffett Field, CA, United States, 1979.

[45] B. van Leer, Flux-vector splitting for the Euler equations, in: E. Krause (Ed.), Eighth International Conference on Numerical Methods in Fluid Dynamics, in: Lecture Notes in Physics, vol. 170, Springer, Berlin, Heidelberg, 1982, pp. 507–512.

[46] D. Hänel, R. Schwane, G. Seider, On the accuracy of upwind schemes for the solution of the Navier-Stokes equations, in: 8th Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, 1987, p. 1105.

[47] M.-S. Liou, C.J. Steffen Jr, High-Order Polynomial Expansions (HOPE) for flux-vector splitting, Technical Memorandum NASA/TM-104452, NASA, NASA Lewis Research Center, Cleveland, OH, United States, 1991.

[48] W. Coirier, B. van Leer, Numerical flux formulas for the Euler and Navier-Stokes equations. II. Progress in flux-vector splitting, in: 10th Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, 1991, p. 1566.

[49] P.G. Buning, J.L. Steger, Solution of the two-dimensional Euler equations with generalized coordinate transformation using flux vector splitting, in: 3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, American Institute of Aeronautics and Astronautics, 1982, p. 971.

[50] B. van Leer, Flux-vector splitting for the 1990s, in: NASA, Lewis Research Center, Computational Fluid Dynamics Symposium on Aeropropulsion, 1991, pp. 203–214.

[51] M.-S. Liou, A sequel to AUSM: AUSM+, J. Comput. Phys. 129 (1996) 364–382, https://doi.org/10.1006/jcph.1996.0256.

[52] G.-C. Zha, E. Bilgen, Numerical solutions of Euler equations by using a new flux vector splitting scheme, Int. J. Numer. Methods Fluids 17 (1993) 115–144, https://doi.org/10.1002/fld.1650170203.

[53] W.J. Gordon, C.A. Hall, Construction of curvilinear co-ordinate systems and applications to mesh generation, Int. J. Numer. Methods Eng. 7 (1973) 461–477, https://doi.org/10.1002/nme.1620070405.

[54] D.A. Kopriva, Metric identities and the discontinuous spectral element method on curvilinear meshes, J. Sci. Comput. 26 (2006) 301–327, https://doi.org/10.1007/s10915-005-9070-8.

[55] M. Vinokur, H. Yee, Extension of efficient low dissipation high order schemes for 3-d curvilinear moving grids, in: Frontiers of Computational Fluid Dynamics 2002, World Scientific, 2002, pp. 129–164.

[56] M.R. Visbal, D.V. Gaitonde, High-order-accurate methods for complex unsteady subsonic flows, AIAA J. 37 (1999) 1231–1239, https://doi.org/10.2514/2.591.

[57] W.K. Anderson, J.L. Thomas, B. Van Leer, Comparison of finite volume flux vector splittings for the Euler equations, AIAA J. 24 (1986) 1453–1460, https://doi.org/10.2514/3.9465.

[58] A. Bölcs, T.H. Fransson, M.F. Platzer, Numerical simulation of inviscid transonic flow through nozzles with fluctuating back pressure, J. Turbomach. 111 (1989) 169–180, https://doi.org/10.1115/1.3262253.

[59] D. Drikakis, S. Tsangaris, On the solution of the compressible Navier-Stokes equations using improved flux vector splitting methods, Appl. Math. Model. 17 (1993) 282–297, https://doi.org/10.1016/0307-904X(93)90054-K.

[60] O. Ålund, J. Nordström, Encapsulated high order difference operators on curvilinear non-conforming grids, J. Comput. Phys. 385 (2019) 209–224, https://doi.org/10.1016/j.jcp.2019.02.007.

[61] J. Manzanero, G. Rubio, E. Ferrer, E. Valero, D.A. Kopriva, Insights on aliasing driven instabilities for advection equations with application to Gauss-Lobatto discontinuous Galerkin methods, J. Sci. Comput. (2017), https://doi.org/10.1007/s10915-017-0585-6, arXiv:1705.01503.

[62] H. Ranocha, G.J. Gassner, Preventing pressure oscillations does not fix local linear stability issues of entropy-based split-form high-order schemes, Commun. Appl. Math. Comput. (2021), https://doi.org/10.1007/s42967-021-00148-z, arXiv:2009.13139.

[63] T. Chen, C.-W. Shu, Review of entropy stable discontinuous Galerkin methods for systems of conservation laws on unstructured simplex meshes, CSIAM Trans. Appl. Math. 1 (2020) 1–52, https://doi.org/10.4208/csiam-am.2020-0003.

[64] H. Ranocha, M. Schlottke-Lakemper, J. Chan, A.M. Rueda-Ramirez, A.R. Winters, F. Hindenlang, G.J. Gassner, Efficient implementation of modern entropy stable and kinetic energy preserving discontinuous Galerkin methods for conservation laws, ACM Trans. Math. Softw. (2023), https://doi.org/10.1145/3625559, arXiv:2112.10517.

[65] J. Chan, C.G. Taylor, Efficient computation of Jacobian matrices for entropy stable summation-by-parts schemes, J. Comput. Phys. 448 (2022) 110701, https://doi.org/10.1016/j.jcp.2021.110701, arXiv:2006.07504.

[66] H. Ranocha, Comparison of some entropy conservative numerical fluxes for the Euler equations, J. Sci. Comput. 76 (2018) 216–242, https://doi.org/10.1007/s10915-017-0618-1, arXiv:1701.02264.

[67] E. Tadmor, Entropy functions for symmetric systems of conservation laws, J. Math. Anal. Appl. 122 (1987) 355–359, https://doi.org/10.1016/0022-247X(87)90265-4.

[68] J. Bezanson, A. Edelman, S. Karpinski, V.B. Shah, Julia: a fresh approach to numerical computing, SIAM Rev. 59 (2017) 65–98, https://doi.org/10.1137/141000671, arXiv:1411.1607.

[69] C. Rackauckas, Q. Nie, DifferentialEquations.jl – a performant and feature-rich ecosystem for solving differential equations in Julia, J. Open Res. Softw. 5 (2017) 15, https://doi.org/10.5334/jors.151.

[70] H. Ranocha, M. Schlottke-Lakemper, A.R. Winters, E. Faulhaber, J. Chan, G.J. Gassner, Adaptive numerical simulations with Trixi.jl: a case study of Julia for scientific computing, in: Proceedings of the JuliaCon Conferences, vol. 1, 2022, p. 77, arXiv:2108.06476.

[71] M. Schlottke-Lakemper, A.R. Winters, H. Ranocha, G.J. Gassner, A purely hyperbolic discontinuous Galerkin approach for self-gravitating gas dynamics, J. Comput. Phys. 442 (2021) 110467, https://doi.org/10.1016/j.jcp.2021.110467, arXiv:2008.10593.

[72] H. Ranocha, SummationByPartsOperators.jl: a Julia library of provably stable semidiscretization techniques with mimetic properties, J. Open Sour. Softw. 6 (2021) 3454, https://doi.org/10.21105/joss.03454, https://github.com/ranocha/SummationByPartsOperators.jl.

[73] S. Christ, D. Schwabeneder, C. Rackauckas, M.K. Borregaard, T. Breloff, Plots.jl — a user extendable plotting API for the Julia programming language, J. Open Res. Softw. (2023), https://doi.org/10.5334/jors.431, arXiv:2204.08775.

[74] J. Ahrens, B. Geveci, C. Law, ParaView: an end-user tool for large-data visualization, in: The Visualization Handbook, Elsevier, 2005, pp. 717–731.

[75] H. Ranocha, A.R. Winters, M. Schlottke-Lakemper, P. Öffner, J. Glaubitz, G.J. Gassner, Reproducibility repository for "On the robustness of high-order upwind summation-by-parts methods for nonlinear conservation laws", https://github.com/trixi-framework/paper-2023-upwind, 2023, https://doi.org/10.5281/zenodo.10200102.

[76] H. Ranocha, L. Dalcin, M. Parsani, D.I. Ketcheson, Optimized Runge-Kutta methods with automatic step size control for compressible computational fluid dynamics, Commun. Appl. Math. Comput. 4 (2021) 1191–1228, https://doi.org/10.1007/s42967-021-00159-w, arXiv:2104.06836.

[77] J. Revels, M. Lubin, T. Papamarkou, Forward-mode automatic differentiation in Julia, https://doi.org/10.48550/arXiv.1607.07892, arXiv:1607.07892, 2016.

[78] C.-W. Shu, Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws, Final Report NASA/CR-97-206253, NASA, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton VA United States, 1997.

[79] S. Hennemann, A.M. Rueda-Ramirez, F.J. Hindenlang, G.J. Gassner, A provably entropy stable subcell shock capturing approach for high order split form DG for the compressible Euler equations, J. Comput. Phys. 426 (2021) 109935, https://doi.org/10.1016/j.jcp.2020.109935.

[80] J. Chan, D.C. Del Rey Fernández, M.H. Carpenter, Efficient entropy stable Gauss collocation methods, SIAM J. Sci. Comput. 41 (2019) A2938–A2966, https://doi.org/10.1137/18M1209234.

[81] J.F.B.M. Kraaijevanger, Contractivity of Runge-Kutta methods, BIT Numer. Math. 31 (1991) 482–528, https://doi.org/10.1007/BF01933264.

[82] S. Conde, I. Fekete, J.N. Shadid, Embedded pairs for optimal explicit strong stability preserving Runge-Kutta methods, J. Comput. Appl. Math. 412 (2022) 114325, https://doi.org/10.1016/j.cam.2022.114325, arXiv:1806.08693.

[83] H. Ranocha, Entropy conserving and kinetic energy preserving numerical methods for the Euler equations using summation-by-parts operators, in: S.J. Sherwin, D. Moxey, J. Peiró, P.E. Vincent, C. Schwab (Eds.), Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018, in: Lecture Notes in Computational Science and Engineering, vol. 134, Springer, Cham, 2020, pp. 525–535.

[84] N. Shima, Y. Kuya, Y. Tamaki, S. Kawai, Preventing spurious pressure oscillations in split convective form discretization for compressible flows, J. Comput. Phys. 427 (2021) 110060, https://doi.org/10.1016/j.jcp.2020.110060.

[85] M. Lukáčová-Medvid'ová, P. Öffner, Convergence of discontinuous Galerkin schemes for the Euler equations via dissipative weak solutions, Appl. Math. Comput. 436 (2023) 22, https://doi.org/10.1016/j.amc.2022.127508, id/No 127508.