Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2024-0225

Vol. **38**, No. 1, pp. 74-108
July 2025

Graph-Based Methods for Hyperbolic Systems of Conservation Laws Using Discontinuous Space Discretizations

Martin Kronbichler[1], Matthias Maier[2] and Ignacio Tomas[3,*]

[1] *University of Augsburg, Germany and Ruhr University Bochum, Bochum 44801, Germany.*
[2] *Department of Mathematics, Texas A&M University, 3368 TAMU, College Station, TX 77843, USA.*
[3] *Department of Mathematics and Statistics, Texas Tech University, 2500 Broadway, Lubbock, TX 79409, USA.*

Communicated by Chi-Wang Shu

Received 24 September 2024; Accepted (in revised version) 28 February 2025

Abstract. We present a graph-based numerical method for solving hyperbolic systems of conservation laws using discontinuous finite elements. This work fills important gaps in the theory as well as practice of graph-based schemes. In particular, four building blocks required for the implementation of flux-limited graph-based methods are developed and tested: a first-order method with mathematical guarantees of robustness; a high-order method based on the entropy viscosity technique; a procedure to compute local bounds; and a convex limiting scheme. Two important features of the current work are the fact that (i) boundary conditions are incorporated into the mathematical theory as well as the implementation of the scheme. For instance, the first-order version of the scheme satisfies pointwise entropy inequalities including boundary effects for any boundary data that is admissible; (ii) sub-cell limiting is built into the convex limiting framework. This is in contrast to the majority of the existing methodologies that consider a single limiter per cell providing no sub-cell limiting capabilities. From a practical point of view, the implementation of graph-based methods is algebraic, meaning that they operate directly on the stencil of the spatial discretization. In principle, these methods do not need to use or invoke loops on cells or faces of the mesh. Finally, we verify convergence rates on various well-known test problems with differing regularity. We propose a simple test in order to verify the implementation of boundary conditions and their convergence rates.

AMS subject classifications: 35L65, 35Q31, 65M12, 65N30, 65M22, 65M60

Key words: Discontinuous finite elements, graph-based formulation, hyperbolic systems, invariant sets, convex limiting, boundary conditions.

*Corresponding author. *Email addresses:* martin.kronbichler@rub.de (M. Kronbichler), maier@tamu.edu (M. Maier), igtomas@ttu.edu (I. Tomas)

1 Introduction

For the last four decades the field of numerical methods for solving hyperbolic systems of conservation equations has been dominated by a paradigm that is commonly referred to as high-resolution schemes. These are numerical methods in which the order of consistency is automatically adjusted locally (in space) depending on some chosen smoothness criteria; see early references [4, 27, 55, 61]. While a heuristic high-resolution method is a good starting point for practical computations, it is not enough to achieve unconditional robustness of the scheme. Here, we define unconditional robustness as the guarantee that the computed update at a given time step remains admissible and maintains crucial physical invariants, such that the resulting state can be used again as input for the next time step update. Modern approaches for constructing robust high-order schemes are based on the following ingredients [8, 20, 25, 32, 36, 39, 40, 44–47, 50, 52, 56, 57, 61, 63]:

(a) a reference low-order method with mathematically guarantees of robustness,

(b) a formally high-order method that may or may not guarantee any robustness properties,

(c) a post-processing procedure based on either flux or slope limiting techniques that blends the low-order and high-order solutions.

A particular incarnation of such postprocessing technique is the convex limiting technique that establishes mathematical guarantees for maintaining a (local) invariant-set property [20].

First-order graph-based formulations combined with discontinuous spatial discretizations are only tersely described in [25, Section 4.3], leaving the path towards high-performance high-order graph-based methods underdeveloped. Therefore, the first goal of the present paper is to complete the mathematical theory and discuss computational aspects of invariant-set preserving schemes for the case of discontinuous finite elements in comprehensive detail. In particular, we incorporate boundary conditions into the formulation of the scheme, provide proofs of invariant-set preservation and discrete entropy inequalities including the effects of boundary data.

The second goal of this paper is to lay out the elementary building blocks required to construct a robust high-order scheme with convex limiting, using a graph-based discontinuous Galerkin discretization. This requires the development and testing of three components:

(i) a heuristic high-order method,

(ii) a suitable strategy for constructing local bounds and their relaxation in order to prevent degradation to first-order accuracy,

(iii) a convex-limiting procedure that blends the high and low order methods while maintaining an invariant set.

1.1 Background: Graph-based methods

A *graph-based formulation* [14, 42, 51] is a numerical method that operates directly on the *stencil* or *sparsity-graph* of the discretization and its degrees of freedom, bypassing entities such as cells, faces, or bilinear forms of the underlying discretization paradigm. In its simplest incarnation, a graph-based formulation takes a solution vector consisting of states $\mathbf{U}_i^n$ associated with a (collocated) degree of freedom i at time t_n and computes an updated state $\mathbf{U}_i^{n+1}$ for the time t_{n+1} as follows:

$$m_i \frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\tau_n} + \sum_{j \in \mathcal{I}(i)} \mathbb{f}(\mathbf{U}_j^n)\mathbf{c}_{ij} - d_{ij}(\mathbf{U}_j^n - \mathbf{U}_i^n) = \mathbf{0}. \tag{1.1}$$

Here, $\mathcal{I}(i)$ is the stencil or "adjacency list" of the i-th degree of freedom, the expression $\sum_{j\in\mathcal{I}(i)} \mathbb{f}(\mathbf{U}_j^n)\mathbf{c}_{ij}$ is an algebraic representation of the inviscid divergence operator, and $d_{ij}(\mathbf{U}_j^n - \mathbf{U}_i^n)$ represents artificial viscous fluxes. The graph-based formulation used in this paper is introduced in detail in Section 3.

The concept of graph-based methods is quite old – one can argue that its roots lie in finite difference approximations on unstructured grids. A notable modern predecessor of what we call graph-based methods is the *group finite element formulation* of Fletcher [14, 51]. More recently, the concept of graph-based methods has been associated with flux-corrected transport techniques; see [37]. There is a rich record of applications of graph-based (sometimes also called edge-based) methods in computational fluid dynamics; see [42, Chapter 10] for a historical account. From the mathematical point of view, invariant-set preserving methods [24] and convex-limiting [20, 25] techniques lead to increased interest in graph-based formulations. Computationally, the algebraic structure of the scheme is a natural idea in order to manipulate individual degrees of freedom and preserve pointwise stability properties.

1.2 Background: Invariant set preservation and convex limiting

As outlined above, one of the fundamental prerequisites for the development of a robust high-resolution scheme is the availability of a first-order scheme with mathematically guaranteed robustness properties. The graph-based methods described in [24, 25] provide such mathematical assurances, by maintaining the so-called *invariant-set* property [24]. These first-order methods provide a discretization agnostic paradigm that works on arbitrary meshes, arbitrary polynomial degree, and arbitrary space dimension.

The *convex-limiting framework* introduced in [20, 25] provides a limiter technique that works on individual degrees of freedom and maintains the invariant-set property. By construction, the limiter formulation is purely algebraic and is not bound to a specific discretization technique. In the context of finite element discretizations, it does not distinguish between cell-interior degrees of freedom and those located at the cell boundary. Therefore, convex limiting is an instance of sub-cell flux limiting. First-order invariant-set preserving methods, as well as the convex-limiting technique, were particularly well

received by the *discontinuous Galerkin spectral element* community [40, 46, 50] and related sub-cell limiting efforts [36, 39].

1.3 Graph-based discontinuous Galerkin formulations and objectives

The framework of invariant-set preserving methods based on convex limiting have been discussed in detail in the context of *continuous* finite element formulations [7, 17, 18, 43]. The low-order stencil-based invariant-set preserving methods have also been combined effectively with discontinuous Galerkin spectral element methods in [40, 46, 50]. This requires combining two different discretization techniques: the low-order method is described and implemented using a purely algebraic (or *stencil based*) approach, while the high-order method is most frequently described and implemented with a cell-based formulation. Unifying the high and low order methods into a single stencil-based description is very desirable, since it greatly simplifies the analysis, construction, and implementation of the schemes.

Therefore, we introduce and discuss suitable low-order and high-order methods based on the algebraic structure introduced in [25, Section 4.3], using the same stencil, and discuss the convex limiting paradigm adapted to the graph-based discontinuous Galerkin setting. We will corroborate our analytical formulation with a computational validation of convergence rates and qualitative fidelity.

1.4 Related works

Regarding other methods, we start by noting that the dominant body of DG schemes are cell-based, formally high-order methods, potentially supplemented with slope-limiting (e.g. Zhang-Shu limiter [62]) or troubled-cell indicator. These technique do not use a reference first-order scheme that preserves all entropy inequalities and all invariant sets. While this may look advantageous, since they do not compute both a high-order and a low-order solution, they do not have local bounds to enforce to the high-order method. That explains why the current slope-limiting paradigm is mostly limited to positivity preservation ($\rho > 0$ and $e > 0$) and rarely even uses local bounds. A noteworthy exceptions of this approach are the publications [46, 50] which borrow some ideas from [25]. These publications indeed rely on a low-order finite volume subgrid method, while the high-order method is a summation-by-parts method. The work in [46,50] uses high-order and low method with different spatial discretizations, thus having different stencils and different basis functions. One of goals of the present work is to present a simple approach that incorporates some form of sub-cell limiting using a single spatial discretization.

The rest of the paper is organized as follows. Section 2 is dedicated to preliminaries and notation about the spatial discretization and hyperbolic systems of conservation laws. In Section 3 we define a first-order method by extending the mathematical description and analysis of graph-based discontinuous Galerkin methods in terms of invariant-set preservation and entropy inequalities. In Section 4 we discuss the procedures for

reflecting boundaries, supersonic inflows and outflows as well as subsonic inflows and outflows. In Section 5 we develop a convex-limited scheme using a robust first-order method and a high-order method using the entropy-viscosity technique. In Section 6 we provide a series of numerical experiments demonstrating that the convex-limited scheme exhibits expected convergence rates. We also propose a simple test for validating the implementation of outflow and inflow boundary conditions.

2 Preliminaries

We briefly introduce relevant notation, recall the concepts of hyperbolic conservation laws, invariant sets and entropy inequalities from the literature and discuss the finite element setting for the proposed graph-based discontinuous formulations. We loosely follow the notation from [20,25].

2.1 Hyperbolic systems, invariant sets, and entropy inequalities

We are interested in solving partial differential equations of the form

$$\partial_t \boldsymbol{u} + \operatorname{div} \mathbb{f}(\boldsymbol{u}) = \mathbf{0}, \tag{2.1}$$

where $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x},t) \in \mathbb{R}^m$ is the state (here, m is the number of components of the system), $\boldsymbol{x} \in \mathbb{R}^d$ is the spatial coordinate with the space dimension d, and $\mathbb{f}(\boldsymbol{u}) : \mathbb{R}^m \to \mathbb{R}^{m\times d}$ denotes the flux. The divergence of the flux is defined as $[\operatorname{div} \mathbb{f}(\boldsymbol{u})]_i = \sum_{j\in\{1:d\}} \partial_{\boldsymbol{x}_j} [\mathbb{f}(\boldsymbol{u})]_{ij}$. We make the following assumptions, see also [23,24].

Assumption 2.1 (Admissible Set)**.** We assume that there is a convex set $\mathcal{A} \subset \mathbb{R}^m$, called *admissible set*, such that the matrix

$$\partial_x [\mathbb{f}(\boldsymbol{u})\boldsymbol{n}] \in \mathbb{R}^{m\times m} \quad \text{with} \quad x := \boldsymbol{n}\cdot\boldsymbol{x}$$

has real eigenvalues for all $\boldsymbol{n} \in \mathbb{S}^{d-1}$. We assume that the solution of (2.1) is understood as the zero-viscosity limit $\boldsymbol{u} = \lim_{\varepsilon\to 0^+} \boldsymbol{u}^\epsilon$, where $\boldsymbol{u}^\epsilon$ solves the parabolic regularization

$$\partial_t \boldsymbol{u}^\epsilon + \operatorname{div} \mathbb{f}(\boldsymbol{u}^\epsilon) = \epsilon \Delta \boldsymbol{u}^\epsilon. \tag{2.2}$$

Assumption 2.2 (Entropy Inequality)**.** Furthermore, we make the important assumption [23,24] that there exists at least one entropy-flux pair $\{\eta(\boldsymbol{u}), \mathfrak{q}(\boldsymbol{u})\}$ associated to (2.2), with $\eta(\boldsymbol{u}) : \mathbb{R}^m \to \mathbb{R}$ and $\mathfrak{q}(\boldsymbol{u}) : \mathbb{R}^m \to \mathbb{R}^d$ such that

$$\partial_t \eta(\boldsymbol{u}^\epsilon) + \operatorname{div} \mathfrak{q}(\boldsymbol{u}^\epsilon) \le \epsilon \Delta \eta(\boldsymbol{u}^\epsilon) \quad \text{for all} \quad \epsilon > 0,$$

see [16, p. 28], such that the zero-viscosity limit $\boldsymbol{u} = \lim_{\varepsilon\to 0^+} \boldsymbol{u}^\epsilon$ satisfies the entropy dissipation inequality

$$\partial_t \eta(\boldsymbol{u}) + \operatorname{div} \mathfrak{q}(\boldsymbol{u}) \le 0.$$

Assumption 2.3 (Finite Speed of Propagation)**.** Consider the solution $\boldsymbol{u}(\boldsymbol{x},t)$ of the projected Riemann problem

$$\partial_t \boldsymbol{u} + \partial_x(\mathbb{f}(\boldsymbol{u})\cdot\boldsymbol{n}) = 0, \quad \text{where} \quad \mathbf{U}_0 = \begin{cases} \boldsymbol{u}_L, & \text{if} \quad x \le 0, \\ \boldsymbol{u}_R, & \text{if} \quad x > 0, \end{cases} \tag{2.3}$$

and $x := \boldsymbol{x}\cdot\boldsymbol{n}$. Given the solution $\boldsymbol{u}(\boldsymbol{x},t)$ of the Riemann problem (2.3), we assume that there is a maximum wave speed of propagation, denoted as $\lambda_{\max}(\boldsymbol{u}_L,\boldsymbol{u}_R,\boldsymbol{n}) > 0$ such that

$$\boldsymbol{u}(\boldsymbol{x},t) = \boldsymbol{u}_L \quad \text{for} \quad \boldsymbol{x}\cdot\boldsymbol{n} \le \frac{1}{2} \quad \text{and} \quad \boldsymbol{u}(\boldsymbol{x},t) = \boldsymbol{u}_R \quad \text{for} \quad \boldsymbol{x}\cdot\boldsymbol{n} \ge \frac{1}{2}$$

provided that $t\lambda_{\max}(\boldsymbol{u}_L,\boldsymbol{u}_R,\boldsymbol{n}) \le 1/2$.

Assumption 2.4 (Invariant Set)**.** We assume that the Riemann problem (2.3) satisfies an invariant-set property of the form: There exist convex subsets $\mathcal{B} \subset \mathcal{A}$ such that

$$\overline{\boldsymbol{u}}(t) := \int_{-1/2}^{1/2} \boldsymbol{u}(x,t)\,\mathrm{d}x \in \mathcal{B},$$

provided that $\boldsymbol{u}(x,t)$ is the unique entropy solution of the Riemann problem with left state $\boldsymbol{u}_L \in \mathcal{B}$ and right state $\boldsymbol{u}_R \in \mathcal{B}$ and that $t\lambda_{\max}(\boldsymbol{u}_L,\boldsymbol{u}_R,\boldsymbol{n}) \le 1/2$. The precise description of the set $\mathcal{B} \subset \mathbb{R}^m$ depends on the initial data and the hyperbolic system at hand; see Section 2.2 for a detailed summary for the compressible Euler equations.

For additional background on entropy inequalities, parabolic regularization principles and invariant sets we refer the reader to [5, 15, 23, 24, 31] and references therein. A general reference on hyperbolic systems of conservation laws is [16].

2.2 The compressible Euler equations

For the compressible Euler equations [16, 58] the conserved state is $\boldsymbol{u} = [\rho,\boldsymbol{m},\mathcal{E}]^\top \in \mathbb{R}^m$ with $m = d+2$ is comprised of the density ρ, momentum $\boldsymbol{m}$, and total energy $\mathcal{E}$. The flux $\mathbb{f}(\boldsymbol{u}) : \mathbb{R}^{d+2} \to \mathbb{R}^{(d+2)\times d}$ is given by

$$\mathbb{f}(\boldsymbol{u}) := \left[\boldsymbol{m}, \rho^{-1}\boldsymbol{m}\otimes\boldsymbol{m} + p\mathbb{I}_d, \frac{\boldsymbol{m}}{\rho}(\mathcal{E}+p)\right]^T, \tag{2.4}$$

where $\mathbb{I}_d$ is the $d\times d$ identity matrix, and p is the pressure. For the sake of simplicity we assume that the system is closed with a polytropic ideal gas equation of state [16,58]. This implies that

$$p = (\gamma-1)\varepsilon(\boldsymbol{u}), \quad \varepsilon(\boldsymbol{u}) := \mathcal{E} - \frac{1}{2}\frac{|\boldsymbol{m}|^2}{\rho}, \tag{2.5}$$

where $\varepsilon(\boldsymbol{u})$ is the internal energy and $\gamma>1$ denotes the ratio of specific heats. The admissible set $\mathcal{A}$ is given by

$$\mathcal{A}=\Big\{[\rho,\boldsymbol{m},\mathcal{E}]^\top\in\mathbb{R}^{d+2}\mid \rho>0 \text{ and } \varepsilon(\boldsymbol{u})>0\Big\}. \tag{2.6}$$

For the case of a polytropic ideal gas equation of state, the Euler equations admit a *mathematical* entropy-flux pair $\{\eta(\boldsymbol{u}),\mathfrak{q}(\boldsymbol{u})\}$ given by

$$\eta(\boldsymbol{u}):=-\rho s(\boldsymbol{u}),\quad \mathfrak{q}(\boldsymbol{u}):=-\boldsymbol{m}s(\boldsymbol{u}),\quad \text{where}\quad s(\boldsymbol{u}):=\log(\rho^{-\gamma}p(\boldsymbol{u})). \tag{2.7}$$

Here, $s(\boldsymbol{u})$ denotes the *specific* entropy. We note that this choice of entropy-flux pair $\{\eta,\mathfrak{q}\}$ is not unique, as there are infinitely many entropy-flux pairs for the Euler equations [26, 33].

Let $\boldsymbol{u}(\boldsymbol{x},t)$ be the unique entropy solution of the Riemann problem (2.3). Then, the Riemann average $\overline{\boldsymbol{u}}(t)$ belongs to the invariant set

$$\mathcal{B}=\big\{\boldsymbol{u}\in\mathcal{A}\mid s(\boldsymbol{u})\geq\min\{s(\boldsymbol{u}_L),s(\boldsymbol{u}_R)\}\big\},$$

provided that the initial data is admissible; i.e., $\boldsymbol{u}_L,\boldsymbol{u}_R\in\mathcal{A}$; and that $t\lambda_{\max}\leq 1/2$; see Assumption 2.3. Note that $\mathcal{B}$ characterizes a minimum principle of the specific entropy s.

2.3 Space discretization

We consider a quadrilateral or hexahedral mesh $\mathcal{T}_h$ and a corresponding nodal, scalar-valued discontinuous finite element space $\mathbb{V}_h$ for each component of the hyperbolic system

$$\mathbb{V}_h=\Big\{v_h(\boldsymbol{x})\in\mathrm{L}^2(\Omega)\mid (v_h\circ\boldsymbol{T}_K)(\widehat{\boldsymbol{x}})\in\mathbb{Q}^k(\widehat{K}),\ \forall K\in\mathcal{T}_h\Big\}.$$

Here, $\boldsymbol{T}_K:\widehat{K}\to K$ denotes a diffeomorphism mapping the unit square or unit cube $\widehat{K}$ to the physical element $K\in\mathcal{T}_h$ and $\mathbb{Q}^k(\widehat{K})$ is the space of bi-, or trilinear Lagrange polynomials of degree k defined on the reference element $\widehat{K}$. That is, the Lagrangian shape functions are defined by enforcing the property $\widehat{\phi}_k(\widehat{\boldsymbol{x}}_j)=\delta_{jk}$ with the Gauß-Lobatto points $\{\widehat{\boldsymbol{x}}_k\}_{k\in\mathcal{N}}$ defined on the reference element. Here, $\mathcal{N}$ is the number of local degrees of freedom on the cell K. The basis functions on the physical element K are then generated using the reference-to-physical map $\boldsymbol{T}_K$: More precisely, for each physical element K, we define shape functions by setting $\phi_{K,i}(\boldsymbol{x}):=\widehat{\phi}_i(\boldsymbol{T}_K^{-1}(\boldsymbol{x}))$ for all $i\in\mathcal{N}$. More detail on the construction and implementation of finite element spaces we refer the reader to [6, 10].

Remark 2.1 (Choice of Basis Functions)**.** We have chosen quadrilateral and hexahedral elements for implementational convenience. Our framework can accommodate the usual set of simplicial elements, tensor-product elements, or even more exotic spatial discretizations, such as rational barycentric coordinates on arbitrary polygons. The only restriction is that the chosen basis $\{\phi_i(\boldsymbol{x})\}$ is interpolatory, has non-negative mass, viz., $\int\phi_i(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\geq 0$, and satisfies the partition of unity property $\sum_{i\in\mathcal{V}}\phi_i(\boldsymbol{x})=1$ for all $\boldsymbol{x}$ in the domain. The precise location of the nodes is of no consequence.

We define $\mathcal{V}=\{i\in\mathbb{N}|1\leq i\leq \dim(\mathbb{V}_h)\}$ as the index set of global, scalar-valued degrees of freedom corresponding to $\mathbb{V}_h$. Similarly, we introduce the set of global shape functions $\{\phi_i(\boldsymbol{x})\}_{i\in\mathcal{V}}$ and the set of collocation points $\{\boldsymbol{x}_i\}_{i\in\mathcal{V}}$. Note that, in the context of nodal discontinuous finite elements, different degrees of freedom can be collocated at the same spatial coordinates. In other words, the situation $\boldsymbol{x}_i=\boldsymbol{x}_j$ with $i\neq j\in\mathcal{V}$ may occur whenever $\boldsymbol{x}_i$ lies on a vertex, edge or face of the mesh. We introduce the index sets

$$\begin{aligned}\mathcal{I}(K)&=\{j\in\mathcal{V}\mid \mathrm{supp}(\phi_j)\cap K\neq\emptyset\},\\ \mathcal{I}(\partial K)&=\{j\in\mathcal{V}\mid \phi_j|_{\partial K}\not\equiv\mathbf{0}\},\\ \mathcal{I}(\partial\Omega)&=\{j\in\mathcal{V}\mid \phi_j|_{\partial\Omega}\not\equiv\mathbf{0}\}.\end{aligned}$$

Note that the set $\mathcal{I}(\partial K)$ also contains indices of shape functions that have no support on K but on a neighboring element of K. We also note that when using finite elements with Gauß-Lobatto points the situation $j\in\mathcal{I}(\partial\Omega)$ can only occur if $\boldsymbol{x}_j$ lies on the boundary $\partial\Omega$. We assume that the basis functions satisfy the following partition of unity property for each element K:

$$\sum_{j\in\mathcal{I}(K)}\phi_j(\boldsymbol{x})=1,\quad \boldsymbol{x}\in K. \tag{2.8}$$

Finally, we introduce some matrices to be used for the algebraic discretization. We define the consistent mass matrix with entries $m_{ij}\in\mathbb{R}$ and lumped mass matrix with entries $m_i\in\mathbb{R}$ as

$$m_{ij}:=\int_K\phi_i\phi_j\mathrm{d}\boldsymbol{x},\quad m_i:=\int_K\phi_i\mathrm{d}\boldsymbol{x}. \tag{2.9}$$

In order to discretize the divergence of the flux, we introduce a vector-valued matrix

$$\mathbf{c}_{ij}:=\begin{cases}\mathbf{c}_{ij}^{K_i}-\mathbf{c}_{ij}^{\partial K_i}, & \text{if}\quad j\in\mathcal{I}(K_i),\\ \mathbf{c}_{ij}^{\partial K_i}, & \text{if}\quad j\in\mathcal{V}\backslash\mathcal{I}(K_i),\end{cases} \tag{2.10}$$

where $K_i\in\mathcal{T}_h$ is the uniquely defined element satisfying $\mathrm{supp}(\phi_i)\cap K_i\neq\emptyset$, and

$$\mathbf{c}_{ij}^K:=\int_K\nabla\phi_j\phi_i\mathrm{d}\boldsymbol{x},\quad \mathbf{c}_{ij}^{\partial K}:=\frac{1}{2}\int_{\partial K}\phi_j\phi_i\boldsymbol{n}_K\mathrm{d}\boldsymbol{s},\quad \mathbf{c}_i^{\partial\Omega}:=\frac{1}{2}\int_{\partial K\cap\partial\Omega}\phi_i\boldsymbol{n}_K\mathrm{d}\boldsymbol{s}, \tag{2.11}$$

where $\boldsymbol{n}_K$ is the outwards pointing normal of the element K. The stencil at the node i is defined as $\mathcal{I}(i)=\{j\in\mathcal{V}\,|\,\mathbf{c}_{ij}\neq 0\}$. From definitions (2.10) and (2.11) it follows that $\mathbf{c}_{ii}=\mathbf{0}$, see Lemma 3.2 for more details.

3 Low-order method

We now introduce a first-order, graph-based method for approximating (2.1). The scheme is based on results previously reported in [24, 25, 44]. A particular novelty of the proposed scheme is the inclusion of boundary conditions in the formulation, which will be

discussed in more detail in Section 4. We introduce the scheme in Section 3.1, discuss conservation in Section 3.2, and prove invariant-set and entropy-dissipation properties in Section 3.3.

3.1 Description of the low-order scheme

Let $\boldsymbol{u}_h^n(\boldsymbol{x})=\sum_{i\in\mathcal{V}}\mathbf{U}_i^n\phi_i(\boldsymbol{x})$ be a given (discontinuous) finite element approximation at time t_n, where the coefficients shall be admissible states $\mathbf{U}_i^n\in\mathcal{A}$. In addition, let $\mathbf{U}_i^{\partial\Omega,n}\in\mathcal{A}$ with $i\in\mathcal{I}(\partial\Omega)$, be a given vector of admissible boundary data for time t_n. We then compute the low order update $\mathbf{U}_i^{\mathrm{L},n+1}$ at time $t_{n+1}=t_n+\tau_n$ as follows:

$$\begin{aligned} m_i\frac{\mathbf{U}_i^{\mathrm{L},n+1}-\mathbf{U}_i^n}{\tau_n}+\sum_{j\in\mathcal{I}(i)}\Big\{\mathbb{f}(\mathbf{U}_j^n)\mathbf{c}_{ij}-d_{ij}^{\mathrm{L},n}(\mathbf{U}_j^n-\mathbf{U}_i^n)\Big\}\\ +\mathbb{f}\big(\mathbf{U}_i^{\partial\Omega,n}\big)\mathbf{c}_i^{\partial\Omega}-d_i^{\partial\Omega,n}\big(\mathbf{U}_i^{\partial\Omega,n}-\mathbf{U}_i^n\big)=\mathbf{0},\quad\text{for}\quad i\in\mathcal{V}. \end{aligned}\tag{3.1}$$

Scheme (3.1) is an algebraic formulation of a discontinuous Galerkin formulation [25] in which the underlying weak formulation is hidden in the matrices m_i and $\mathbf{c}_{ij}$. We note that (3.1) is based on a *central flux* approximation [25] with subsequent interpolation, using the nodal property of the shape functions [30]

$$\begin{aligned} \sum_{j\in\mathcal{I}(i)}\mathbb{f}(\mathbf{U}_j^n)\mathbf{c}_{ij}&=\int_{K_i}\nabla\cdot\Big(\sum_{j\in\mathcal{I}(K_i)}\mathbb{f}(\mathbf{U}_j^n)\phi_j\Big)\phi_i\,\mathrm{d}\boldsymbol{x}\\ &\quad+\frac{1}{2}\int_{\partial K_i}\Big(\sum_{j\in\mathcal{I}(\partial K_i)\setminus\mathcal{I}(K_i)}\mathbb{f}(\mathbf{U}_j^n)\phi_j-\sum_{j\in\mathcal{I}(\partial K_i)\cap\mathcal{I}(K_i)}\mathbb{f}(\mathbf{U}_j^n)\phi_j\Big)\cdot\boldsymbol{n}_{K_i}\phi_i\,\mathrm{d}\boldsymbol{x}\\ &\approx-\int_{K_i}\mathbb{f}(\boldsymbol{u}_h^n)\cdot\nabla\phi_i\,\mathrm{d}\boldsymbol{x}+\int_{\partial K_i}\{\{\mathbb{f}(\boldsymbol{u}_h^n)\}\}_{\partial K_i}\cdot\boldsymbol{n}_{K_i}\phi_i\,\mathrm{d}\boldsymbol{x}, \end{aligned}$$

where $\{\{\mathbb{f}(\boldsymbol{u}_h^n)\}\}_{\partial K_i}$ denotes the average between the two adjacent states. For more details, see also [25].

Scheme (3.1) imposes the boundary data weakly with a jump condition, defining a boundary flux through an outer state with the help of boundary conditions in the usual DG form, see e. g., [3, 28, 30], albeit with a simplified, *diagonal* flux, $\mathbb{f}(\mathbf{U}_i^{\partial\Omega,n})\mathbf{c}_i^{\partial\Omega}$, as detailed in Section 4 below. Since $\mathbf{c}_i=\mathbf{0}$ and $d_i^{\partial\Omega,n}=0$ for $i\in\mathcal{V}\setminus\mathcal{I}(\partial\Omega)$, (3.1) is valid for both boundary and interior degrees of freedom simultaneously. We defer the discussion on how to construct a suitable boundary data vector $\mathbf{U}_i^{\partial\Omega,n}$ to Section 4.

The graph viscosities $d_{ij}^{\mathrm{L},n}, d_i^{\partial\Omega,n}>0$ in (3.1) are computed as follows:

$$\begin{cases} d_{ij}^{\mathrm{L},n}:=|\mathbf{c}_{ij}|_{\ell^2}\lambda_{\max}^+(\mathbf{U}_i^n,\mathbf{U}_j^n,\boldsymbol{n}_{ij}), & \text{where}\quad \boldsymbol{n}_{ij}=\dfrac{\mathbf{c}_{ij}}{|\mathbf{c}_{ij}|_{\ell^2}},\quad\text{for}\quad i\neq j,\\ d_i^{\partial\Omega,n}:=|\mathbf{c}_i^{\partial\Omega}|_{\ell^2}\lambda_{\max}^+(\mathbf{U}_i^n,\mathbf{U}_i^{\partial\Omega,n},\boldsymbol{n}_i), & \text{where}\quad \boldsymbol{n}_i=\dfrac{\mathbf{c}_i^{\partial\Omega}}{|\mathbf{c}_i^{\partial\Omega}|_{\ell^2}}. \end{cases}\tag{3.2}$$

Here, $\lambda^+_{\max}(\mathbf{U}_L,\mathbf{U}_R,\boldsymbol{n}):\mathbb{R}^m\times\mathbb{R}^m\times\mathbb{S}^{d-1}\longrightarrow\mathbb{R}^+$ is an upper bound of the maximum wavespeed of the projected Riemann problem (2.3) [24]. We also introduce $d_{ii}^{\mathrm{L},n}$

$$d_{ii}^{\mathrm{L},n} := -\sum_{j\in\mathcal{I}(i),j\neq i} d_{ij}^{\mathrm{L},n} - d_i^{\partial\Omega,n}. \tag{3.3}$$

This definition plays a role in the computation of the largest admissible time-step size, see (3.11), however, we note that $d_{ii}^{\mathrm{L},n}$ is not needed in order to compute the update $\mathbf{U}_i^{\mathrm{L},n+1}$ with (3.1).

The low order scheme (3.1)-(3.2) is a first-order invariant-set preserving approximation in the spirit of [24]. To this end we rigorously establish conservation properties in Section 3.2 and derive a *bar state* characterization that in turn implies an invariant-set property and discrete entropy inequalities (see Lemma 3.5). We have summarized an adaptation of (3.1)-(3.2) for the prescription of boundary conditions in the context of *continuous* finite element discretizations in Appendix B.

3.2 Conservation properties

In order to establish conservation properties of the scheme we make use of some auxiliary results.

Lemma 3.1 (Partition of Unity Property)**.** *It holds that*

$$\sum_{j\in\mathcal{I}(i)} \mathbf{c}_{ij} + \mathbf{c}_i^{\partial\Omega} = \mathbf{0}, \quad \textit{for all} \quad i\in\mathcal{V}. \tag{3.4}$$

Proof. Let $i\in\mathcal{V}$ be arbitrary and let K_i denote the cell with $i\in\mathcal{I}(K)$. Using the definitions (2.10) and (2.11), as well as property (2.8), we can compute

$$\begin{aligned}
\sum_{j\in\mathcal{I}(i)} \mathbf{c}_{ij} + \mathbf{c}_i^{\partial\Omega} &= \sum_{j\in\mathcal{I}(K_i)} \left(\mathbf{c}_{ij}^{K_i} - \mathbf{c}_{ij}^{\partial K_i}\right) + \sum_{j\in\mathcal{I}(i)\setminus\mathcal{I}(K_i)} \mathbf{c}_{ij}^{\partial K_i} + \mathbf{c}_i^{\partial\Omega} \\
&= \int_{K_i} \nabla\Big(\sum_{j\in\mathcal{I}(K_i)} \phi_j\Big)\phi_i \,\mathrm{d}\boldsymbol{x} - \frac{1}{2}\int_{\partial K_i}\Big(\sum_{j\in\mathcal{I}(K_i)}\phi_j\Big)\phi_i \boldsymbol{n}_{K_i}\,\mathrm{d}\boldsymbol{s} \\
&\quad + \frac{1}{2}\int_{\partial K_i\setminus\partial\Omega}\Big(\sum_{j\in\mathcal{I}(i)\setminus\mathcal{I}(K_i)}\phi_j\Big)\phi_i\boldsymbol{n}_{K_i}\,\mathrm{d}\boldsymbol{s} + \frac{1}{2}\int_{\partial K_i\cap\partial\Omega}\phi_i\boldsymbol{n}_{K_i}\,\mathrm{d}\boldsymbol{s} \\
&= -\frac{1}{2}\int_{\partial K_i}\phi_i\boldsymbol{n}_{K_i}\,\mathrm{d}\boldsymbol{s} + \frac{1}{2}\int_{\partial K_i\setminus\partial\Omega}\phi_i\boldsymbol{n}_{K_i}\,\mathrm{d}\boldsymbol{s} + \frac{1}{2}\int_{\partial K_i\cap\partial\Omega}\phi_i\boldsymbol{n}_{K_i}\,\mathrm{d}\boldsymbol{s} = \mathbf{0}.
\end{aligned}$$

The proof is complete. □

Lemma 3.2. *The matrix $\mathbf{c}_{ij}$ is skew-symmetric*

$$\mathbf{c}_{ij} = -\mathbf{c}_{ji} \quad \textit{for all} \quad i,j\in\mathcal{V}. \tag{3.5}$$

Proof. The statement is an immediate consequence of definition (2.10) and the fact that integration by parts shows

$$\mathbf{c}_{ij}^{K_i} - \mathbf{c}_{ij}^{\partial K_i} = -\mathbf{c}_{ji}^{\partial K_i} + \mathbf{c}_{ji}^{K_i} = -\left(\mathbf{c}_{ji}^{\partial K_j} - \mathbf{c}_{ji}^{K_j}\right)$$

for $K_i = K_j$; or that $\mathbf{c}_{ij}^{\partial K_i} = \mathbf{c}_{ji}^{\partial K_i} = -\mathbf{c}_{ji}^{\partial K_j}$ whenever $K_i \neq K_j$ due to the fact that $\boldsymbol{n}_{K_i} = -\boldsymbol{n}_{K_j}$ on $\partial K_i \cap \partial K_j$. □

Conservation equation (2.1) implies that the following flux-balance of the state $\mathbf{U}_i$ between time t^n and t^{n+1} holds true:

$$\int_\Omega \boldsymbol{u}(\boldsymbol{x},t^{n+1})\,\mathrm{d}\boldsymbol{x} + \int_{t^n}^{t^{n+1}} \int_{\partial\Omega} \mathbb{f}(\boldsymbol{u}(\boldsymbol{x},s))\boldsymbol{n}_K \mathrm{d}\boldsymbol{s}\mathrm{d}s = \int_\Omega \boldsymbol{u}(\boldsymbol{x},t^n)\,\mathrm{d}\boldsymbol{x}.$$

We now show that the scheme (3.1) satisfies a discrete counterpart of such a flux balance. We start by deriving an explicit *skew-symmetric* counterpart of scheme (3.1):

Lemma 3.3 (Skew Symmetric Form)**.** *Using property* (3.4) *scheme* (3.1) *can be written equivalently as*

$$m_i\left(\mathbf{U}_i^{\mathrm{L},n+1} - \mathbf{U}_i^n\right) + \sum_{j\in\mathcal{I}(i)} \mathbf{F}_{ij}^{\mathrm{L}} + \mathbf{F}_i^{\partial\Omega,\mathrm{L}} = \mathbf{0} \tag{3.6}$$

with the fluxes

$$\begin{aligned}
\mathbf{F}_{ij}^{\mathrm{L}} &:= \tau_n\left(\mathbb{f}(\mathbf{U}_j^n) + \mathbb{f}(\mathbf{U}_i^n)\right)\mathbf{c}_{ij} - \tau_n d_{ij}^{\mathrm{L},n}\left(\mathbf{U}_j^n - \mathbf{U}_i^n\right),\\
\mathbf{F}_i^{\partial\Omega,\mathrm{L}} &:= \tau_n\left(\mathbb{f}(\mathbf{U}_i^{\partial\Omega,n}) + \mathbb{f}(\mathbf{U}_i^n)\right)\mathbf{c}_i^{\partial\Omega} - \tau_n d_i^{\partial\Omega,n}\left(\mathbf{U}_i^{\partial\Omega,n} - \mathbf{U}_i^n\right),
\end{aligned}$$

where the vectors $\mathbf{F}_{ij}^{\mathrm{L}} \in \mathbb{R}^m$ are skew symmetric, i.e., $\mathbf{F}_{ij}^{\mathrm{L}} = -\mathbf{F}_{ji}^{\mathrm{L}}$. Note that $\mathbf{F}_i^{\partial\Omega,\mathrm{L}}$ is a boundary flux.

Summing up (3.6) over the index i and using the skew symmetry of $\mathbf{F}_{ij}^{\mathrm{L}}$ leads to the following corollary.

Corollary 3.1 (Total Balance)**.** *Scheme* (3.1) *satisfies the balance equation*

$$\sum_{i\in\mathcal{V}} m_i \frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\tau_n} + \sum_{i\in\mathcal{I}(\partial\Omega)} \mathbf{F}_i^{\partial\Omega,\mathrm{L}} = 0 \tag{3.7}$$

for all $i \in \mathcal{V}$. Using definitions (2.9) *and* (2.11) *one can write* (3.7) *equivalently as*

$$\begin{aligned}
&\int_\Omega \boldsymbol{u}_h^{n+1}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \frac{\tau_n}{2}\int_{\partial\Omega} \sum_{i\in\mathcal{I}(\partial\Omega)} \phi_i\left(\mathbb{f}(\mathbf{U}_i^{\partial\Omega,n}) + \mathbb{f}(\mathbf{U}_i^n)\right)\boldsymbol{n}_K \mathrm{d}\boldsymbol{s}\\
&- \sum_{i\in\mathcal{I}(\partial\Omega)} d_i^{\partial\Omega,n}\left(\mathbf{U}_i^{\partial\Omega,n} - \mathbf{U}_i^n\right) = \int_\Omega \boldsymbol{u}_h^n(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.
\end{aligned}$$

Remark 3.1. $\mathbf{F}_i^{\partial\Omega,\mathrm{L}}$ can be viewed as a central flux between the boundary state $\mathbf{U}_i^n$ and a *ghost node* with state $\mathbf{U}_i^{\partial\Omega,n}$ in the usual discontinuous Galerkin framework.

3.3 Invariant-set preservation

We now focus on stability properties of the low-order scheme. In the spirit of [24, 25], we rewrite (3.1) as a convex combination of *bar states*. These are states formed by an algebraic combination of interacting degrees of freedom that, under a suitable CFL condition, correspond to the spatial average of an associated one dimensional Riemann problem. However, in contrast to the discussion found in the references above, we end up with an additional set of bar states that depend on the boundary data. We start with the following algebraic identity.

Lemma 3.4 (Convex Reformulation)**.** *The update procedure* (3.1) *can equivalently be written as follows:*

$$\mathbf{U}_i^{\mathrm{L},n+1} = \left(1+\frac{2\tau_n d_{ii}^{\mathrm{L},n}}{m_i}\right)\mathbf{U}_i^n + \frac{2\tau_n d_i^{\partial\Omega,n}}{m_i}\overline{\mathbf{U}}_i^{\partial\Omega,n} + \sum_{j\in\mathcal{I}(i)\setminus\{i\}}\frac{2\tau_n d_{ij}^{\mathrm{L},n}}{m_i}\overline{\mathbf{U}}_{ij}^n, \tag{3.8}$$

where $\overline{\mathbf{U}}_{ij}^n$ *and* $\overline{\mathbf{U}}_i^{\partial\Omega,n}$ *are the bar states defined by*

$$\overline{\mathbf{U}}_{ij}^n = \frac{1}{2}\left(\mathbf{U}_j^n+\mathbf{U}_i^n\right) - \frac{|\mathbf{c}_{ij}|}{2d_{ij}^{\mathrm{L}}}\left(\mathbb{f}(\mathbf{U}_j^n)-\mathbb{f}(\mathbf{U}_i^n)\right)\boldsymbol{n}_{ij}, \tag{3.9}$$

$$\overline{\mathbf{U}}_i^{\partial\Omega,n} = \frac{1}{2}\left(\mathbf{U}_i^{\partial\Omega,n}+\mathbf{U}_i^n\right) - \frac{|\mathbf{c}_i^{\partial\Omega}|}{2d_i^{\partial\Omega,n}}\left(\mathbb{f}\left(\mathbf{U}_i^{\partial\Omega,n}\right)-\mathbb{f}\left(\mathbf{U}_i^n\right)\right)\boldsymbol{n}_i. \tag{3.10}$$

Note that $\overline{\mathbf{U}}_i^{\partial\Omega,n}$ *depends on the boundary data* $\mathbf{U}_i^{\partial\Omega,n}$*.*

Proof. Similarly to the derivation of the skew symmetric form (3.6), we can use identity (3.4) to rewrite (3.1) as follows:

$$\begin{aligned} m_i\mathbf{U}_i^{n+1} = m_i\mathbf{U}_i^n &- \tau_n\sum_{j\in\mathcal{I}(i)}\left(\mathbb{f}(\mathbf{U}_j^n)-\mathbb{f}(\mathbf{U}_i^n)\right)\mathbf{c}_{ij} - d_{ij}^{\mathrm{L},n}\left(\mathbf{U}_j^n-\mathbf{U}_i^n\right) \\ &- \tau_n\left(\mathbb{f}\left(\mathbf{U}_i^{\partial\Omega,n}\right)-\mathbb{f}\left(\mathbf{U}_i^n\right)\right)\mathbf{c}_i^{\partial\Omega} - d_i^{\partial\Omega,n}\left(\mathbf{U}_i^{\partial\Omega,n}-\mathbf{U}_i^n\right). \end{aligned}$$

We now add and subtract $2\tau_n\sum_{j\in\mathcal{I}(i)\setminus\{i\}} d_{ij}^{\mathrm{L},n}\mathbf{U}_i^n$ and $2\tau_n d_i^{\partial\Omega,n}\mathbf{U}_i^{\partial\Omega,n}$

$$\begin{aligned} m_i\mathbf{U}_i^{n+1} = &\left(m_i - 2\tau_n d_i^{\partial\Omega,n} - \sum_{j\in\mathcal{I}(i)\setminus\{i\}}2\tau_n d_{ij}^{\mathrm{L},n}\right)\mathbf{U}_i^n \\ &- \tau_n\sum_{j\in\mathcal{I}(i)\setminus\{i\}}\left(\mathbb{f}(\mathbf{U}_j^n)-\mathbb{f}(\mathbf{U}_i^n)\right)\mathbf{c}_{ij} - d_{ij}^{\mathrm{L},n}\left(\mathbf{U}_j^n+\mathbf{U}_i^n\right) \\ &- \tau_n\left(\mathbb{f}\left(\mathbf{U}_i^{\partial\Omega,n}\right)-\mathbb{f}\left(\mathbf{U}_i^n\right)\right)\mathbf{c}_i^{\partial\Omega} - d_i^{\partial\Omega,n}\left(\mathbf{U}_i^{\partial\Omega,n}+\mathbf{U}_i^n\right). \end{aligned}$$

The result now follows readily dividing both sides of the equality by m_i and using definition (3.3). □

Lemma 3.5 (Pointwise Entropy Inequality)**.** *Let* $\{\eta,\mathfrak{q}\}$ *be* any *entropy-flux pair of the hyperbolic system* $\partial_t \boldsymbol{u}+div\mathbb{f}(\boldsymbol{u})=\mathbf{0}$ [23, 24]. *Assume that the update* (3.1) *is performed with a time step size* τ_n *satisfying the following CFL condition:*

$$-\tau_n \frac{2d_{ii}^{\mathrm{L},n}}{m_i} \leq 1 \quad \textit{for all} \quad i\in\mathcal{V}. \tag{3.11}$$

Then the update $\mathbf{U}_i^{n+1}$ *satisfies the following pointwise entropy inequality:*

$$\begin{aligned} m_i\frac{\eta(\mathbf{U}_i^{n+1})-\eta(\mathbf{U}_i^n)}{\tau_n}+\sum_{j\in\mathcal{I}(i)}\mathfrak{q}(\mathbf{U}_j^n)\cdot\mathbf{c}_{ij}-d_{ij}^{\mathrm{L},n}\big(\eta(\mathbf{U}_j^n)-\eta(\mathbf{U}_i^n)\big)\\ +\mathfrak{q}\big(\mathbf{U}_i^{\partial\Omega,n}\big)\cdot\mathbf{c}_i^{\partial\Omega}-d_i^{\partial\Omega,n}\Big(\eta\big(\mathbf{U}_i^{\partial\Omega,n}\big)-\eta\big(\mathbf{U}_i^n\big)\Big)\leq 0 \end{aligned} \tag{3.12}$$

for all $i\in\mathcal{V}$.

Proof. Proving inequality (3.12) relies on the convex reformulation (3.8), the convexity of the entropy η and corresponding inequalities that hold true for the bar states defined in (3.9) and (3.10)

$$\begin{aligned} \eta\big(\overline{\mathbf{U}}_{ij}^n\big)&\leq\frac{1}{2}\big(\eta(\mathbf{U}_i^n)+\eta(\mathbf{U}_j^n)\big)-\frac{|\mathbf{c}_{ij}|_{\ell^2}}{2d_{ij}^{\mathrm{L}}}\big(\mathfrak{q}(\mathbf{U}_j^n)-\mathfrak{q}(\mathbf{U}_i^n)\big)\boldsymbol{n}_{ij},\\ \eta\big(\overline{\mathbf{U}}_i^{\partial\Omega,n}\big)&\leq\frac{1}{2}\Big(\eta\big(\mathbf{U}_i^{\partial\Omega,n}\big)+\eta\big(\mathbf{U}_i^n\big)\Big)-\frac{|\mathbf{c}_i^{\partial\Omega}|_{\ell^2}}{2d_i^{\partial\Omega,n}}\Big(\mathfrak{q}\big(\mathbf{U}_i^{\partial\Omega,n}\big)-\mathfrak{q}\big(\mathbf{U}_i^n\big)\Big)\boldsymbol{n}_i. \end{aligned}$$

Such inequalities hold true provided that $d_{ij}^{\mathrm{L},n}$ and $d_i^{\partial\Omega,n}$ are chosen large enough so that the bar states represent an *average value* over the *Riemann fan* [24]. In particular, the choice (3.2) is sufficient [24]. For further details we refer to the detailed discussion found in [24, Theorem 4.7] and [25, Theorem 3.8]. □

Remark 3.2 (Global Entropy Inequality)**.** Similarly to the procedure in Section 3.2 that establishes global conservation (see Lemma 3.3 and Corollary 3.1), we can rewrite inequality (3.12) in skew symmetric form and sum up. This leads to a global entropy inequality

$$\sum_{i\in\mathcal{V}} m_i\eta\big(\mathbf{U}_i^{n+1}\big)+\tau_n\sum_{i\in\mathcal{I}(\partial\Omega)}\mathbf{Q}_i^{\partial\Omega,\mathrm{L}}\leq\sum_{i\in\mathcal{V}} m_i\eta(\mathbf{U}_i^n) \tag{3.13}$$

with (viscous) boundary fluxes

$$\mathbf{Q}_i^{\partial\Omega,\mathrm{L}}:=\Big(\mathfrak{q}\big(\mathbf{U}_i^{\partial\Omega,n}\big)+\mathfrak{q}\big(\mathbf{U}_i^n\big)\Big)\cdot\mathbf{c}_i^{\partial\Omega}-d_i^{\partial\Omega,n}\Big(\eta\big(\mathbf{U}_i^{\partial\Omega,n}\big)-\eta\big(\mathbf{U}_i^n\big)\Big).$$

This is nothing else than the discrete counterpart of a global entropy inequality satisfied by the entropy-flux pair $\{\eta,\mathfrak{q}\}$.

Remark 3.3 (Time Step Size Restriction)**.** CFL condition (3.11) determines the largest time step size that can be used for an individual update step. For example, in practical implementations it is convenient to select a (user specified) constant $0 < \mathrm{Cr} \leq 1$ and then compute a time step size as follows:

$$\tau_n = \mathrm{Cr} \cdot \min_{i \in \mathcal{V}} \left(-\frac{m_i}{2 d_{ii}^{\mathrm{L},n}} \right). \tag{3.14}$$

Here, m_i decreases and $d_{ii}^{L,n}$ grows with increasing polynomial degree k leading to a stricter time step size restriction.

Lemma 3.6 (Invariant Set Property)**.** *Under the stated CFL condition* (3.11) *and assuming that the provided boundary data* $\mathbf{U}_i^{\partial\Omega,n}$ *is in the invariant set* $\mathcal{B}$ *for all* $i \in \mathcal{I}(\partial\Omega)$*, then the update* $\mathbf{U}_i^{n+1}$ *computed by* (3.1) *and* (3.2) *will satisfy* $\mathbf{U}_i^{n+1} \in \mathcal{B}$ *as well.*

Proof. The statement is a direct consequence of the fact that (3.8) expresses $\mathbf{U}_i^{n+1}$ as a convex combination of bar states, that in turn are located in the invariant set $\mathcal{B}$ provided that the CFL condition (3.11) holds; see Lemma 3.5. □

4 Boundary conditions

We now discuss how to construct the boundary data vector $\mathbf{U}_i^{\partial\Omega,n} \in \mathbb{R}^m$, $i \in \mathcal{I}(\partial\Omega)$ for different types of boundary conditions.

The construction of the boundary data vector $\mathbf{U}_i^{\partial\Omega,n}$ follows well established procedures, we refer to e. g., [3, 9, 28–30]. For the sake of completeness we briefly summarize our approach based on [18].

4.1 Construction of boundary data $\mathbf{U}_i^{\partial\Omega,n}$

We distinguish Dirichlet boundary conditions, slip boundary conditions, supersonic and subsonic in- and outflow.

Dirichlet boundary conditions. For Dirichlet boundaries we simply set $\mathbf{U}_i^{\partial\Omega,n} \in \mathcal{A}$ to the desired boundary data at position $\boldsymbol{x}_i$ for time t_n.

Slip boundary conditions. We impose slip boundary conditions for a boundary state $\mathbf{U}_i^n = [\rho_i^n, \boldsymbol{m}_i^n, \mathcal{E}_i^n]^\top$ at a boundary collocation point $\boldsymbol{x}_i$ by setting

$$\mathbf{U}_i^{\partial\Omega,n} := [\rho_i^n, \boldsymbol{m}_i^{\partial\Omega,n}, \mathcal{E}_i^n]^\top, \quad \text{where} \quad \boldsymbol{m}_i^{\partial\Omega,n} := \boldsymbol{m}_i^n - 2(\boldsymbol{m}_i^n \cdot \boldsymbol{n}_i)\boldsymbol{n}_i, \tag{4.1}$$

and where we recall the definition $\boldsymbol{n}_i = \mathbf{c}_i^{\partial\Omega} / |\mathbf{c}_i^{\partial\Omega}|_{\ell^2}$ for a boundary collocation point $\boldsymbol{x}_i$. This implies that $\boldsymbol{m}_i^{\partial,n}$ and $\boldsymbol{m}_i^n$ have opposite normal components but the same tangential projection with respect to the normal $\boldsymbol{n}_i$. The boundary flux $\mathbf{F}_i^{\partial\Omega,\mathrm{L}}$ consequently only

affects the balance of the normal component of the momentum and leaves all other components unaffected.

Supersonic in- and outflow. In order to impose supersonic in- and outflow at portions of the boundary, we proceed as follows. Given a state $\mathbf{U}_i^n = [\rho_i^n, \boldsymbol{m}_i^n, \mathcal{E}_i^n]^\top$ at a boundary collocation point $\boldsymbol{x}_i$ with velocity $\boldsymbol{v}_i^n$, local speed of sound $a_i(\mathbf{U}_i^n)$, and corresponding Dirichlet data $\mathbf{U}_i^{\mathrm{d},n}$ for the inflow, we set

$$\mathbf{U}_i^{\partial\Omega,n} = \begin{cases} \mathbf{U}_i^{\mathrm{d},n}, & \text{if} \quad \boldsymbol{v}_i^n \cdot \boldsymbol{n}_i < -a_i, \\ \mathbf{U}_i^n, & \text{if} \quad \boldsymbol{v}_i^n \cdot \boldsymbol{n}_i \geq a_i. \end{cases} \tag{4.2}$$

Subsonic in-flow and outflow. For the subsonic case, we need to distinguish in- and outgoing characteristics. To impose conditions only on the ingoing ones, we construct the boundary data vector $\mathbf{U}_i^{\partial\Omega,n}$ by blending together the current state $\mathbf{U}_i^n$ and the given Dirichlet data $\mathbf{U}_i^{\mathrm{d},n}$ [9,18,29]. To this end, we briefly review the approach described in [18] for the case of our discontinuous formulation.

Given a state $\mathbf{U} = [\rho, \boldsymbol{m}, \mathcal{E}]^\top$ and a unit vector $\boldsymbol{n}$ we introduce the following set of characteristic variables $\mathcal{R}_k(\mathbf{U}, \boldsymbol{n})$ and characteristic speeds $\lambda_k(\mathbf{U}, \boldsymbol{n})$, $k = 1, \ldots, 4$:

$$\begin{cases} \mathcal{R}(\mathbf{U}, \boldsymbol{n}) := \left\{ v_{\boldsymbol{n}} - \dfrac{2a}{\gamma - 1}, \ \dfrac{p(\mathbf{U})}{\rho^\gamma}, \ \boldsymbol{v} - (\boldsymbol{v} \cdot \boldsymbol{n})\boldsymbol{n}, \ v_{\boldsymbol{n}} + \dfrac{2a}{\gamma - 1} \right\}, \\ \lambda(\mathbf{U}, \boldsymbol{n}) := \{ v_{\boldsymbol{n}} - a, \ v_{\boldsymbol{n}}, \ v_{\boldsymbol{n}}, \ v_{\boldsymbol{n}} + a \}. \end{cases} \tag{4.3}$$

Here, $v_{\boldsymbol{n}} = \boldsymbol{v} \cdot \boldsymbol{n}$ and $a = \sqrt{\gamma p / \rho}$ is the local speed of sound. The strategy now consists of constructing the boundary data satisfying: $\mathbf{U}_i^{\partial\Omega,n}$ with $\mathcal{R}_i(\mathbf{U}_i^{\partial\Omega,n}, \boldsymbol{n}) = \mathcal{R}_i(\mathbf{U}_i^{\mathrm{d},n}, \boldsymbol{n})$ if $\lambda_i(\mathbf{U}_i^n, \boldsymbol{n}) \leq 0$ (incoming characteristics), and $\mathcal{R}_i(\mathbf{U}_i^{\partial\Omega,n}, \boldsymbol{n}) = \mathcal{R}_i(\mathbf{U}_i^n, \boldsymbol{n})$ otherwise. Constructing an admissible state $\mathbf{U}_i^{\partial\Omega,n}$ satisfying such constraint is always possible; see [18, § 4.3.2]. In addition, the resulting state $\mathbf{U}_i^{\partial\Omega,n}$ is always admissible provided $\mathbf{U}_i^{\mathrm{d}}$ is admissible, and the construction coincides with (4.2) for the case of supersonic in- and outflow.

4.2 Multi-valued boundary conditions

As opposed to classical implementations of discontinuous Galerkin schemes via face integrals, the algebraic approach implies that more than one boundary condition applies to a given boundary collocation point $\boldsymbol{x}_i$, for instance

– For a typical channel flow setup, a small subset of the boundary collocation points $\boldsymbol{x}_i$ (those at corners) lie between the slip boundaries at the top and bottom as well as the inflow and outflow boundaries at the left and right, respectively.

– Another example is given by two parts of the boundary with slip boundary conditions that meet at an angle, say 90°. Here, slip boundary conditions for both normals should rather be enforced instead of a single slip boundary condition with a combined normal (of 45°).

In order to treat such boundary states, we first partition the boundary $\partial\Omega$ into all $\mathcal{K}$ disjoint components $\partial\Omega_k$, where we either apply a different boundary condition, or where portions of the boundary meet with a large angle. Then we split $\mathbf{c}_i^{\partial\Omega}$ accordingly

$$\partial\Omega = \bigcup_{k\in[1:\mathcal{K}]} \partial\Omega_k, \quad \mathbf{c}_i^{\partial\Omega} = \sum_{k\in[1:\mathcal{K}]} \mathbf{c}_i^{\partial\Omega,k}, \quad \text{with} \quad \mathbf{c}_i^{\partial\Omega,k} := \frac{1}{2}\int_{\partial K\cap\partial\Omega_k} \phi_i \boldsymbol{n}_K \mathrm{d}s.$$

Finally, scheme (3.1) takes the form

$$\begin{aligned} m_i \frac{\mathbf{U}_i^{\mathrm{L},n+1}-\mathbf{U}_i^n}{\tau_n} &+ \sum_{j\in\mathcal{I}(i)} \left\{ \mathbb{f}(\mathbf{U}_j^n)\mathbf{c}_{ij} - d_{ij}^{\mathrm{L},n}(\mathbf{U}_j^n-\mathbf{U}_i^n)\right\} \\ &+ \sum_{k\in[1:\mathcal{K}]} \left\{ \mathbb{f}\big(\mathbf{U}_i^{\partial\Omega,k,n}\big)\mathbf{c}_i^{\partial\Omega,k} - d_i^{\partial\Omega,k,n}\big(\mathbf{U}_i^{\partial\Omega,k,n}-\mathbf{U}_i^n\big)\right\} = \mathbf{0}, \quad \text{for} \quad i\in\mathcal{V}, \end{aligned} \tag{4.4}$$

where $\mathbf{U}_i^{\partial\Omega,k,n}$ is a modified vector of appropriately chosen boundary data, and the modified graph viscosity $d_i^{\partial\Omega,k,n}$ is computed as follows:

$$d_i^{\partial\Omega,k,n} := \big|\mathbf{c}_i^{\partial\Omega,k}\big|_{\ell^2} \lambda_{\max}^+\big(\mathbf{U}_i^{\partial\Omega,k,n}, \mathbf{U}_i^n, \boldsymbol{n}_i^k\big), \quad \text{where} \quad \boldsymbol{n}_i^k = \frac{\mathbf{c}_i^{\partial\Omega,k}}{\big|\mathbf{c}_i^{\partial\Omega,k}\big|_{\ell^2}}. \tag{4.5}$$

5 High-order method and convex limiting

Following the same approach as discussed in [20,25], we now introduce a formally high-order method by using the consistent mass matrix and introducing a *high-order graph-viscosity* d_{ij}^{H},

$$\begin{aligned} \sum_{j\in\mathcal{I}(i)} m_{ij} \frac{\mathbf{U}_j^{\mathrm{H},n+1}-\mathbf{U}_j^n}{\tau_n} &+ \sum_{j\in\mathcal{I}(i)} \left\{ \mathbb{f}(\mathbf{U}_j^n)\mathbf{c}_{ij} - d_{ij}^{\mathrm{H},n}(\mathbf{U}_j^n-\mathbf{U}_i^n)\right\} \\ &+ \mathbb{f}\big(\mathbf{U}_i^{\partial\Omega,n}\big)\mathbf{c}_i^{\partial\Omega} - d_i^{\partial\Omega,\mathrm{H},n}\big(\mathbf{U}_i^{\partial\Omega,n}-\mathbf{U}_i^n\big) = \mathbf{0}, \quad \text{for} \quad i\in\mathcal{V}. \end{aligned} \tag{5.1}$$

Here, $m_{ij} = \int_K \phi_i\phi_j \mathrm{d}\boldsymbol{x}$ denotes the consistent mass matrix. A considerable body of stabilization techniques have been developed over the years, supplying ideas that could be adapted to the computation of high-order graph viscosities $d_{ij}^{\mathrm{H},n}$. Among these methods we mention entropy-viscosity [22], smoothness sensors [47], and semi-discrete entropy-stable flux constructions [12,13]; see also [45] for a comprehensive review of approaches.

All of these methods have in common that they try to ensure that $d_{ij}^{\mathrm{H},n} \approx d_{ij}^{\mathrm{L},n}$ near shocks and discontinuities, while forcing $d_{ij}^{\mathrm{H},n} \approx 0$ in smooth regions of the solution.

The development of high-order methods for discontinuous spatial discretizations requires some attention to the minimal amount of viscosity. Without enough viscosity between the element interfaces, the method might not even be stable for smooth solutions. Therefore, we will first present a high-order viscosity $d_{ij}^{\min,n}$ such that if used in scheme (5.1):

(i) it results in a stable approximation of smooth solutions on structured and unstructured meshes;

(ii) we observe optimal convergence rates for smooth problems. We will call such a viscosity $d_{ij}^{\min,n}$ a *minimally stabilizing viscosity*.

This viscosity will not have any shock-capturing capability and might not be best choice of viscosity for non-smooth problems. The sole purpose of such a viscosity is to define a minimal amount of viscosity that the method should always have.

On the other hand, we would like to adapt the entropy viscosity methodology described in [21,22] to the context of graph-based methods using discontinuous spatial discretizations. We will denote the entropy viscosity as $d_{ij}^{\mathrm{ev},n}$. Such viscosity should provide the required shock capturing capabilities. Ultimately, we set

$$d_{ij}^{\mathrm{H},n} := \max\left\{ d_{ij}^{\min,n}, d_{ij}^{\mathrm{ev},n} \right\} \tag{5.2}$$

to guarantee that the high-order scheme (5.1) possesses enough viscosity to deliver stable solutions and optimal convergence rates in the context of smooth solutions as well as shock-capturing capabilities in the context of non-smooth problems.

In Section 5.1 we define the minimally stabilizing viscosity $d_{ij}^{\min}$ while in Section 5.2 we describe the entropy viscosity $d_{ij}^{\mathrm{ev},n}$. In Section 5.3 we describe the convex limiting procedure to ensure that the blended method satisfies *local* bounds at every collocation point.

5.1 Minimally stabilizing high-order viscosity

The usual Lax-Friedrichs flux of the form $\int_F \lambda [[u_h]] \phi_i \mathrm{d}s$, where λ is an estimate on the maximum wavespeed between cell interfaces, leads to the usual optimal convergence rate $\|u-u_h\|_{L^2(\Omega)} \leq \mathcal{O}(h^{k+1/2})$. However, it can be observed experimentally that the behavior of the Lax-Friedrichs flux is suboptimal for the case of even polynomial degree in the L^1-norm. For instance, for $k=2$ when solving the isentropic vortex [40,60,62] a rate of $\mathcal{O}(h^{2.75})$ can be observed instead of the expected $\mathcal{O}(h^{3.0})$, see Remark 5.1. The phenomenon of observing suboptimal convergence rates for even polynomial degrees in the L^1 norm has been well known for a while by practitioners but it is rarely commented on

in the literature. Following an argument in [54] we use a different scaling for the cases of odd and even polynomial degrees in order to define a minimally stabilizing high-order viscosity that recovers the optimal rate $\mathcal{O}(h^{k+1})$ in the L^1-norm when k is even.

Remark 5.1. We note that such a degradation in convergence rates typically manifests only after a sufficiently large number of mesh refinements has been reached. Therefore, in our numerical results (reported in Section 6) a sufficiently large number of mesh refinements is used to ensure that we have reached an *asymptotic regime*.

In light of the discussion above, we set $d_{ij}^{\min} := d_{ij}^{\mathrm{L},n}$ if $\boldsymbol{x}_i = \boldsymbol{x}_j$, for the case of *odd* polynomial degree. This choice is roughly equivalent the interfacial Lax-Friedrichs flux. However, for *even* polynomial degree we adapt the idea outlined in [54] and set $d_{ij}^{\min} = \mathcal{O}(h^{1/2}) d_{ij}^{\mathrm{L},n}$ if $\boldsymbol{x}_i = \boldsymbol{x}_j$. However, we want to avoid introducing a length scaling h into the high-order viscosity, therefore we set

$$d_{ij}^{\min} := \begin{cases} c_k \widehat{h}_{ij}^{p_k} d_{ij}^{\mathrm{L},n}, & \text{if} \quad \boldsymbol{x}_i = \boldsymbol{x}_j, \\ 0, & \text{otherwise}, \end{cases} \tag{5.3}$$

where

$$\widehat{h}_{ij} := \left(\frac{1}{2} \frac{m_i + m_j}{|\Omega|} \right)^{\frac{1}{d}}$$

is a dimensionless mesh size, and the constants c_k and p_k are set to $c_k = 1$, and

$$p_k = \begin{cases} \dfrac{1}{2}, & \text{if} \quad k \text{ is even}, \\ 0, & \text{if} \quad k \text{ is odd}. \end{cases}$$

The graph viscosity $d_{ij}^{\min}$ is symmetric by construction. The choice of coefficient p_k follows from the theoretical and computational discussion outlined in [54]. Extensive numerical tests indicate that the power $p = 1/2$ for the case of even polynomial degree is indeed optimal; it maintains a stable approximation of smooth solutions on unstructured meshes, as well as optimal convergence rates in the L^1 norm. On the other hand, the non-dimensional constant $c_k > 0$ can be chosen more freely. For the sake of reproducibility we simply report our particular choice for the parameters used in our numerical results (Section 6), which we found to be reasonable for a large number of test cases.

Remark 5.2 (Superconvergence)**.** We note that discontinuous spatial discretizations of even polynomial degree (without stabilization) are superconvergent for the case of linear conservation equations on uniform meshes. For instance, non-stabilized discontinuous $\mathbb{Q}^2$ spatial discretizations have been shown to be fourth order accurate for smooth linear problems on uniform meshes with periodic boundary conditions, see [1]. In view of these theoretical results, it seems tempting to simply set $d_{ij}^{\min} = 0$ as a minimal viscosity choice. However, we have observed numerically that these theoretical results for linear conservation equations do not necessarily translate to the case of non-linear hyperbolic systems, nor to general unstructured hexahedral meshes.

5.2 Entropy viscosity

The entropy viscosity commutator has been introduced in [22]. Here, we summarize a variant discussed in [21]. Consider the generalized Harten entropy $f(s(\boldsymbol{u}))$, where $s(\boldsymbol{u})$ is the specific entropy (2.7), and f is any function satisfying the constraints

$$f'(s)>0, \quad f'(s)c_p^{-1}-f''(s)>0.$$

Here $c_p=\theta\partial s/\partial\theta$ is the specific heat at constant pressure, see [23]. For any admissible state $\boldsymbol{u}=[\rho,\boldsymbol{m},\mathcal{E}]^\top\in\mathbb{R}^{d+2}$ we adopt the shorthand notation $f(\boldsymbol{u}):=f(s(\boldsymbol{u}))$. We then define a *shifted* generalized mathematical entropy $\Phi_i^n(\boldsymbol{u})$ and a corresponding entropy-flux $\mathfrak{q}_i^n(\boldsymbol{u})$

$$\Phi_i^n(\boldsymbol{u})=\rho[f(\boldsymbol{u})-f(\mathbf{U}_i^n)], \quad \mathfrak{q}_i^n(\boldsymbol{u})=\boldsymbol{m}[f(\boldsymbol{u})-f(\boldsymbol{u}_i^n)].$$

Let $\nabla_{\boldsymbol{u}}\Phi_i^n(\boldsymbol{u})\in\mathbb{R}^{d+2}$ denote the gradient of $\Phi_i^n(\boldsymbol{u})$ with respect to the state $\boldsymbol{u}$ and set

$$\begin{aligned}
R_i &:= \sum_{j\in\mathcal{I}(i)}[\mathfrak{q}_i(\mathbf{U}_j^n)-(\nabla_{\boldsymbol{u}}\Phi_i^n)^\top\mathbb{f}(\mathbf{U}_j^n)]\cdot\mathbf{c}_{ij},\\
D_i &:= \left|\sum_{j\in\mathcal{I}(i)}\mathfrak{q}(\mathbf{U}_j^n)\cdot\mathbf{c}_{ij}\right|+\sum_{k\in m}|[\nabla_{\boldsymbol{u}}\Phi_i]_k||\mathbb{f}_k(\mathbf{U}_j^n)\mathbf{c}_{ij}|,
\end{aligned}$$

where $[\nabla_{\boldsymbol{u}}\Phi_i]_k$ is the k-th component of $\nabla_{\boldsymbol{u}}\Phi_i$ and $\mathbb{f}_k(\mathbf{U}_j^n)\in\mathbb{R}^{1\times d}$ denotes the k-th row of the flux $\mathbb{f}(\mathbf{U}_j^n)\in\mathbb{R}^{(d+2)\times d}$. We define the normalized entropy-viscosity residual N_i and entropy viscosity d_{ij}^{ev} as

$$N_i:=\frac{R_i}{D_i} \quad\text{and}\quad d_{ij}^{\mathrm{ev},n}:=d_{ij}^{\mathrm{L}}\min\left\{c_{\mathrm{ev}}\max\{|N_i|,|N_j|\},1\right\},$$

where c_{ev} is a constant that will in general depend on the polynomial degree. From numerical explorations we have chosen to use $c_{\mathrm{ev}}=1,0.5,0.25$ for the polynomial degrees $k=1,2,3$, respectively.

5.3 Convex limiting: Algebraic reformulation

In analogy to Lemma 3.3, we rewrite the high-order scheme (5.1) as follows:

$$m_i\left(\mathbf{U}_i^{\mathrm{H},n+1}-\mathbf{U}_i^n\right)+\sum_{j\in\mathcal{I}(i)}\mathbf{F}_{ij}^{\mathrm{H}}+\mathbf{F}_i^{\partial\Omega,\mathrm{H}}=\mathbf{0}, \tag{5.4}$$

where the algebraic fluxes $\mathbf{F}_{ij}^{\mathrm{H}}$ are given by

$$\begin{aligned}
\mathbf{F}_{ij}^{\mathrm{H}} &:= \tau_n\left(\mathbb{f}(\mathbf{U}_j^n)+\mathbb{f}(\mathbf{U}_i^n)\right)\mathbf{c}_{ij}-\tau_n d_{ij}^{\mathrm{H},n}\left(\mathbf{U}_j^n-\mathbf{U}_i^n\right)\\
&\quad+(m_{ij}-\delta_{ij}m_i)\left(\mathbf{U}_j^{\mathrm{H},n+1}-\mathbf{U}_j^n-\mathbf{U}_i^{\mathrm{H},n+1}+\mathbf{U}_i^n\right),\\
\mathbf{F}_i^{\partial\Omega,\mathrm{H}} &:= \tau_n\left(\mathbb{f}\left(\mathbf{U}_i^{\partial\Omega,n}\right)+\mathbb{f}\left(\mathbf{U}_i^n\right)\right)\mathbf{c}_i^{\partial\Omega}-\tau_n d_{ij}^{\partial\Omega,\mathrm{H},n}\left(\mathbf{U}_i^{\partial\Omega,n}-\mathbf{U}_i^n\right).
\end{aligned}$$

Here, we have used the fact that $\sum_{j\in\mathcal{I}(i)}(m_{ij}-\delta_{ij}m_i)=0$, which is a well known technique [25, 38] for absorbing the consistent mass matrix m_{ij} into the fluxes. We note that the high-order algebraic fluxes are skew symmetric, $\mathbf{F}_{ij}^{\mathrm{H}}=-\mathbf{F}_{ji}^{\mathrm{H}}$. Furthermore, subtracting (3.6) from (5.4), after some reorganization we obtain

$$m_i\mathbf{U}_i^{\mathrm{H},n+1}=m_i\mathbf{U}_i^{\mathrm{L},n+1}+\sum_{j\in\mathcal{I}(i)}\mathbf{A}_{ij}+\mathbf{A}_i^{\partial\Omega}, \tag{5.5}$$

where $\mathbf{A}_{ij}:=\mathbf{F}_{ij}^{\mathrm{L}}-\mathbf{F}_{ij}^{\mathrm{H}}$ and $\mathbf{A}_i^{\partial\Omega}:=\mathbf{F}_i^{\partial\Omega,\mathrm{L}}-\mathbf{F}_i^{\partial\Omega,\mathrm{H}}$, with $\mathbf{A}_{ij}$ skew symmetric, i.e. $\mathbf{A}_{ij}=-\mathbf{A}_{ij}$. Eq. (5.5) now serves as a starting point for the convex limiting technique. We compute the new, blended update $\mathbf{U}_i^{n+1}$ by setting

$$m_i\mathbf{U}_i^{n+1}=m_i\mathbf{U}_i^{\mathrm{L},n+1}+\sum_{j\in\mathcal{I}(i)}\ell_{ij}^n\mathbf{A}_{ij}+\ell_i^{\partial\Omega,n}\mathbf{A}_i^{\partial\Omega}, \tag{5.6}$$

where $\ell_{ij}^n=\ell_{ji}^n\in[0,1]$ and $\ell_i^{\partial\Omega,n}\in[0,1]$ are limiter coefficients. From (5.6) and (5.5), it is evident that $\ell_{ij},\ell_i^{\partial\Omega}=0$ recovers the low-order scheme and, conversely, $\ell_{ij},\ell_i^{\partial\Omega}=1$ the high-order scheme. The goal is thus to choose the limiter coefficients as large as possible while maintaining a pointwise invariant-set property (in the spirit of Lemma 3.6), i.e., $\mathbf{U}_i^{n+1}\in\mathcal{A}$.

Remark 5.3 (First-order Scheme and High-order Polynomials)**.** In this work, we use the same finite element basis and stencil for both the high-order and low-order methods. However, we note that several authors [19, 41, 46] have explored the argument that the first-order method degrades their accuracy with increasing polynomial degree k. Therefore, the first-order scheme should be computed using a subgrid of low-order polynomial degree, usual $k=0$ or $k=1$. While it is indeed true that the first-order method degrades its accuracy with increasing polynomial degree, the importance of such a degradation is not substantial for modest polynomial degree, as illustrated by Table 5. For instance, in the table it can be observed that the error of the $\mathbb{Q}_2$ first-order scheme is $1.6\times$ the error of the $\mathbb{Q}_1$ first-order scheme. Similarly, the error of the $\mathbb{Q}_3$ first-order scheme is $2.3\times$ larger than the error of the corresponding $\mathbb{Q}_1$ method using the same total number of DOFs. We believe that this higher error pre-factor is acceptable: in return we obtain a simpler method with a more straight-forward code, while avoiding all the complexity associated to have a low-order method defined in a subgrid. Of course, for very high order polynomial degrees, say $k\geq 5$, this sentiment might not hold true.

5.4 Convex limiting: Local bounds and line search

We want to enforce *local* bounds on the density ρ and the specific entropy $s(\boldsymbol{u})$ given by (2.7). However, the logarithm in (2.7) makes the specific entropy a rather cumbersome quantity to work with directly. Following our previous work [18, 20, 25], we use the

rescaled quantity

$$\widetilde{s}(\boldsymbol{u}) = \rho^{-\gamma}\varepsilon(\boldsymbol{u}) = \frac{1}{\gamma-1}\exp(s(\boldsymbol{u}))$$

instead. Since it is a monotonic rescaling, enforcing a minimum bound on $\widetilde{s}(\boldsymbol{u})$ will also enforce a minimum bound on $s(\boldsymbol{u})$. For each node $i \in \mathcal{V}$, we construct local bounds $\rho_i^{\min}$, $\rho_i^{\max}$, $\widetilde{s}_i^{\min}$, and construct limiter coefficients ℓ_{ij}^n, $\ell_i^{\partial\Omega,n}$, such that the final update $\mathbf{U}_i^{n+1} = [\rho_i^{n+1}, \boldsymbol{m}_i^{n+1}, \mathcal{E}_i^{n+1}]$ given by (5.6), satisfies bounds

$$\rho_i^{\min} \le \rho_i^{n+1} \le \rho_i^{\max}, \quad \widetilde{s}\big(\mathbf{U}_i^{n+1}\big) \ge \widetilde{s}_i^{\min}.$$

In this manuscript we use the following local bounds:

$$\begin{cases} \rho_i^{\min} := r_h^- \min\Big\{\rho(\overline{\mathbf{U}}_i^{\partial\Omega,n}), \min_{j\in\mathcal{I}(i)}\min_{k\in\mathcal{I}(j)}\rho_k^n, \min_{j\in\mathcal{I}(i)}\min_{k\in\mathcal{I}(j)}\rho(\overline{\mathbf{U}}_{jk}^n)\Big\}, \\ \rho_i^{\max} := r_h^+ \max\Big\{\rho(\overline{\mathbf{U}}_i^{\partial\Omega,n}), \max_{j\in\mathcal{I}(i)}\max_{k\in\mathcal{I}(j)}\rho_k^n, \max_{j\in\mathcal{I}(i)}\max_{k\in\mathcal{I}(j)}\rho(\overline{\mathbf{U}}_{jk}^n)\Big\}, \\ \widetilde{s}_i^{\min} := r_h^- \min\Big\{\widetilde{s}(\overline{\mathbf{U}}_i^{\partial\Omega,n}), \min_{j\in\mathcal{I}(i)}\min_{k\in\mathcal{I}(j)}\widetilde{s}(\mathbf{U}_k^n), \min_{j\in\mathcal{I}(i)}\min_{k\in\mathcal{I}(j)}\widetilde{s}(\overline{\mathbf{U}}_{jk}^n)\Big\}, \end{cases} \tag{5.7}$$

where the bar states $\overline{\mathbf{U}}_{jk}^n$ and $\overline{\mathbf{U}}_i^{\partial\Omega,n}$ are defined in (3.9) and (3.10) respectively. We note that the use of the bar states in (5.7) is owed to equation (3.8): the low-order update is a convex combination of the bar states; see also the discussion in [20, Section 4.1] and [25, Lemma 7.15]. The relaxation coefficients $r_h^\pm$ are defined as follows:

$$r_h^- := 1 - c_r \widehat{h}_i^{p_r}, \quad r_h^+ := 1 + c_r \widehat{h}_i^{p_r},$$

where $\widehat{h}_i = (m_i/|\Omega|)^{1/d}$ is a dimensionless mesh size. For the numerical tests in Section 6 we use the constants

$$c_r = 4.0, \quad p_r = 1.5 \tag{5.8}$$

for all polynomial degrees throughout.

The constants have been chosen with a quick parametric study such that we observe expected convergence rates for the numerical tests summarized in Section 6. The relaxation coefficients (5.8) are necessary to recover optimal convergence rates because a strict enforcement of the *local* minimum principle on the specific entropy would result in a first order scheme; see [20,34]. We note that this relaxation has no consequence on the robustness of the scheme: As long as the initial data is admissible, the update will result again in an admissible state.

A tempting alternative to the relaxation of the local bounds (5.7) is to dispense with using local bounds altogether and replacing them with a single set of global bounds, $\{\rho_{\min}^{\text{global}}, \rho_{\max}^{\text{global}}, \widetilde{s}_{\min}^{\text{global}}\}$ with $\rho_{\min}^{\text{global}} > 0$ and $\widetilde{s}_{\min}^{\text{global}} > 0$, for all degrees of freedom. While positivity preserving, such a limiter strategy – at least from our experience – lacks control on over- and undershoots and leads to unsatisfying numerical results for benchmark configurations.

The limiter coefficients ℓ_{ij}^n, $\ell_i^{\partial\Omega,n}$ are now constructed with the help of one dimensional line searches [20]. For this, we first rewrite (5.7) in terms of a convex set,

$$\mathcal{B}_i = \left\{ \mathbf{U} = [\rho, \boldsymbol{m}, \mathcal{E}]^\top \in \mathbb{R}^{d+2} \;\middle|\; \rho_i^{\min} \le \rho \le \rho_i^{\max}, \widetilde{s}(\mathbf{U}) \ge \widetilde{s}_i^{\min} \right\}, \tag{5.9}$$

and rewrite (5.6) as follows:

$$\mathbf{U}_i^{n+1} = \sum_{j \in \mathcal{I}(i)} \kappa_i \left(\mathbf{U}_i^{\mathrm{L},n+1} + \ell_{ij}^n \mathbf{P}_{ij} \right) + \kappa_i \left(\mathbf{U}_i^{\mathrm{L},n+1} + \ell_i^{\partial\Omega,n} \mathbf{P}_i^{\partial\Omega} \right) \tag{5.10}$$

with $\mathbf{P}_{ij} := \mathbf{A}_{ij} / \kappa_i m_i$, $\mathbf{P}_i^{\partial\Omega} = \mathbf{A}_i^{\partial\Omega} / \kappa_i m_i$, and $\kappa_i := (\mathrm{card}\mathcal{I}(i) + 1)^{-1}$.

Eq. (5.10) describes the update $\mathbf{U}_i^{n+1}$ as a convex combination of limited, unidirectional updates $\mathbf{U}_i^{\mathrm{L},n+1} + \ell_{ij}^n \mathbf{P}_{ij}$. This allows us to reduce the construction of ℓ_{ij}^n and $\ell_i^{\partial\Omega,n}$ to solving one dimensional line searches [20, Lemma 4.3].

Lemma 5.1. *Assume that $\mathbf{U}_i^{\mathrm{L},n+1} + \ell_{ij} \mathbf{P}_{ij} \in \mathcal{B}_i$ for all $j \in \mathcal{I}(i)$ and $\mathbf{U}_i^{\mathrm{L},n+1} + \ell_i^{\partial\Omega} \mathbf{P}_i^{\partial\Omega} \in \mathcal{B}_i$. Then, $\mathbf{U}_i^{n+1}$ as defined by* (5.10)*, belongs to the set $\mathcal{B}_i$ as well. If l_{ij} are symmetric, meaning $\ell_{ij} = \ell_{ji}$, then the convex limited high-order update $\mathbf{U}_i^{n+1}$ is also conservative.*

In summary, the limiter coefficients are chosen such that

$$\mathbf{U}_i^{\mathrm{L},n+1} + \ell_{ij} \mathbf{P}_{ij} \in \mathcal{B}_i \quad \text{for all} \quad j \in \mathcal{I}(i), \quad \text{and} \quad \mathbf{U}_i^{\mathrm{L},n+1} + \ell_i^{\partial\Omega} \mathbf{P}_i^{\partial\Omega} \in \mathcal{B}_i,$$

which in turn implies that $\mathbf{U}_i^{n+1} \in \mathcal{B}_i$.

6 High-performance implementation and computational results

We now outline a high-performance implementation of the numerical scheme in the hydrodynamic solver framework `ryujin` [18, 43]. The code supports discontinuous and continuous finite elements on quadrangular meshes for the spatial approximation and is built upon the `deal.II` finite element library [2]. We conclude the section by discussing a number of validation and benchmark results.

6.1 Implementation

Due to the graph-based construction of the method, the implementation of the proposed discontinuous Galerkin scheme can be realized in analogy to the scheme described for continuous elements in [43], and can be applied to arbitrarily unstructured meshes including local adaptive refinement. The implementation computes the necessary information row-by-row by a (parallel) loop over the index range of variable i, with single-instruction multiple-data (SIMD) vectorization across several rows to ensure a high utilization of data-level parallelism. The kernels are written to balance data access and computations for optimal performance on modern CPU-based high-performance architecture, performing the following main steps:

- For the computation of the low-order update, the flux $\mathbb{f}(\boldsymbol{u}_h^n)$ is evaluated point-wise and the graph viscosity (3.2) is computed with a point-wise Riemann solver. For data locality reasons, most of the factors for the high-order viscosity are also computed in the necessary sweep over all mesh nodes.

- The high-order update and convex-limiting steps (5.10) involve combinations of the fluxes along the i and j indices as well as the high-order viscosity (5.2), together with the evaluation of the limiter coefficients ℓ_{ij}^n.

- In order to obtain converged results of the one-dimensional line searches, the limiter step is executed twice, necessitating two sweeps through all nodes.

As described in [43], the computational cost of the above steps is not only dominated by the actual flux computations, and the elevated number of around 14-20 divisions per non-zero entry (i,j) of the stencil, but also by four transcendental power functions per non-zero (using Padé-type approximations) that play a crucial role. Furthermore, the cost for indirect addressing into generic sparse matrix data structures along the index j are also relatively high. Overall, the proposed scheme yields an arithmetic cost proportional to $\mathcal{O}((k+1)^d)$ operations per degree of freedom. This is a substantial cost when compared to state-of-the-art cell-based implementations of discontinuous Galerkin methods, where modern implementations typically utilize on-the-fly evaluation of the underlying finite-element integrals using sum-factorization techniques for $\mathcal{O}(k+1)$ complexity per degree of freedom [11,35], or related properties deduced by spectral polynomial bases and one-dimensional differentiation operations [49]. The cost has to be contrasted against the mathematically proven robust realization proposed here.

Note that the chosen implementation does not utilize the additional structure provided by the element-wise basis functions, which could allow to fuse some of the indirect addressing and additional computations for the unknowns inside a finite element cell, compared to the abstract row-by-row processing of our approach. However, as the computational part with expensive transcendental functions is the more restrictive bottleneck on current architectures [43], which is addressed in ongoing research.

6.2 Validation tests

We now verify the proposed method for three different solution regimes:

(i) a smooth analytic solution given by the isentropic vortex [59],

(ii) a semi-smooth solution (continuous with second derivatives of bounded variation) given by a single rarefaction wave,

(iii) the discontinuous solution of the LeBlanc shock tube that has large pressure and density jumps.

For all three cases analytic expressions for the solution can be found in [20], specifically we use [20, Eq. 5.3], [20, Table 2], and [20, Table 4] with the same choices for computational domains and parameters. As a figure of merit we introduce a consolidated error norm

$$L^p\text{-error} := \frac{\|\rho_h - \rho\|_{L^p(\Omega)}}{\|\rho\|_{L^p(\Omega)}} + \frac{\|\boldsymbol{m}_h - \boldsymbol{m}\|_{L^p(\Omega)}}{\|\boldsymbol{m}\|_{L^p(\Omega)}} + \frac{\|\mathcal{E}_h - \mathcal{E}\|_{L^p(\Omega)}}{\|\mathcal{E}\|_{L^p(\Omega)}},$$

for $p = 1,2,\infty$ evaluated at the final time. We enforce Dirichlet boundary conditions throughout by setting the boundary data $\mathbf{U}_i^{\partial\Omega,n}$ to the exact (time-dependent) solution. We describe each test with more details in the following bullets:

Isentropic vortex (test case (i)). We set the computational domain for the smooth test case (isentropic vortex) to the square $[-5,5]^2$; see [20, Eq. 5.3]. For the sake of completeness we repeat the formulas of the exact analytical solution [59]

$$\rho(\boldsymbol{x},t) = (\rho_\infty + \delta\rho(\boldsymbol{x},t))^{\frac{1}{\gamma-1}}, \quad \boldsymbol{v} = \boldsymbol{v}_\infty + \delta\boldsymbol{v}, \quad p(\boldsymbol{x},t) = \rho(\boldsymbol{x},t)^\gamma,$$
$$\delta\boldsymbol{v}(\boldsymbol{x},t) = \frac{\beta}{2\pi} e^{1-r^2} [-\overline{\boldsymbol{x}}_2, \overline{\boldsymbol{x}}_1]^\top, \quad \delta\rho(\boldsymbol{x},t) = -\frac{(\gamma-1)\beta^2}{8\gamma\pi^2} e^{1-r^2},$$

$\overline{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_0 - t\boldsymbol{v}_\infty$, $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2]^\top$ are the space coordinates, $\boldsymbol{x}_0 = [\boldsymbol{x}_{10}, \boldsymbol{x}_{20}]$ is the initial position of the vortex, and $r = |\overline{\boldsymbol{x}}|_{\ell^2}$. For all our tests: $\rho_\infty = 1$, $\boldsymbol{v}_\infty = [1,1]^\top$, $\boldsymbol{x}_0 = [-1,-1]$, $\gamma = 5/3$, and $\beta = 5.0$. The initial time is $t_0 = 0$ and final time is $t_F = 2$. We discretize the domain with uniform grids with n_e elements per edge, where $n_e = n_k \cdot 2^r$ with $n_k = 24,16,12$ for the cases polynomial degrees $k = 1,2,3$, respectively. We now create a series of increasingly refined meshes by varying r from 0 to 5. The values of n_k are chosen such that each refinement level r has the same number of degrees of freedom for all polynomial degrees $k = 1,2,3$. For time integration we use SSPRK-54 throughout, a fourth order strong stability preserving Runge Kutta method [53]. Computational results are summarized in Table 1 on page 102. Classical error analysis for linear advection problems indicate that the expected rate in the $L^2(\Omega)$-norm is of order $\mathcal{O}(h^{k+1/2})$. In general, we observe expected convergence rates for all reported test cases. A notable exception is a slight reduction of convergence rates in the $L^1(\Omega)$ and $L^\infty(\Omega)$ norms for polynomial degree $k = 3$.

Rarefaction wave (test case (ii)). Similarly, for the rarefaction test case (ii) we split the unit interval $[0,1]$ into $n_e = n_k \cdot 2^r$ uniform subintervals with $n_k = 60,40,30$ for $k = 1,2,3$, respectively, and by varying r from 0 to 7. Regarding time-integration, we use the SSPRK3 scheme for all polynomial degrees. We note that the error for the rarefaction wave is dominated by the fact that the initial data is non-differentiable in $\boldsymbol{x} = 0.2$. This test also has the added difficulty that there is a sonic point at $\boldsymbol{x} = 0.2$: numerical methods without enough artificial viscosity will not produce an entropic solution. Error estimates from polynomial interpolation suggest a limit of $\mathcal{O}(h^2)$ for the convergence rate in the L^1-norm. However,

we are not aware of any scheme capable of delivering second order rates for the rarefaction wave test. For instance, finite volume methods with piecewise linear reconstructions deliver rates $\mathcal{O}(h^p)$ with $p\in[1.333,1.50]$, see [48]; semi-discretely entropy-stable methods yield $p\leq 1.50$ regardless of the polynomial degree, see [40]; first-order continuous finite elements with using entropy-viscosity and convex limiting achieve $p\in[1.60,1.65]$, see [21]. Our results are reported in Table 2 on page 102. We observe a convergence order $\mathcal{O}(h^p)$ in the L^1-norm with average $p\approx 1.70$, 1.60, and 1.63 for polynomial degrees $k=1$, 2, and 3, respectively.

Leblanc shock tube (test case (iii)). Results for the LeBlanc shocktube are summarized in Table 3 page 103. We observe the expected linear convergence rate for all polynomial degrees $k=1$, 2, 3. From mathematical approximation theory, it is well known that high order polynomial degrees offer no advantage when approximating discontinuous problems. In this sense, the numerical results in Table 3 for the LeBlanc shocktube test are optimal. Note that, in Table 3, the exact same number of global degrees of freedom is used on each refinement level for $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_3$ elements. Comparing the obtained L^1-error for all three cases we tend to conclude that $\mathbb{Q}_1$ elements are the optimal choice – at least for the case of discontinuous solutions with strong shocks. For the same number of global degrees of freedom they offer the smallest L^1-error while having low computational complexity, and comparatively large time-step sizes.

Remark 6.1 (Verification of Boundary Conditions with Isentropic Vortex)**.** In order to test our implementation of boundary conditions, in Section 6.3 we modify the isentropic vortex benchmark. We increase the final time t_F so that the simulation ends with the center of the isentropic situated exactly above the top right corner of the computational domain. By doing so, the correct treatment of boundary data is essential for recovering optimal convergence rates.

6.3 Accuracy of boundary condition enforcement

We now briefly evaluate the performance of the boundary condition enforcement in (3.1) and (5.1) by repeating the smooth isentropic vortex test, case (i), with a modified final time t_F for the convex-limited method and bilinear finite elements ($k=1$). Specifically, we choose a final time of $t_F=6/M$ for increasing choices of (directional) vortex speed $M=1$, 1.5, 2.0, 2.5. Here, $M=1$ implies that the center of the vortex is moving exactly with the speed of sound a in x and in y direction individually. We note that for $M=1$ and $M=1.5$ a significant portion of the top and right edges (outflow boundaries) will be subsonic. With the choice of final time $t_F=6/M$, the center of the vortex is located exactly on top of the top right corner at final time t_F. As a rigorous figure of merit we examine convergence rates in the L^1-norm. Four different strategies are tested for the construction of boundary data $\mathbf{U}_i^{\partial\Omega,n}$

(a) exact Dirichlet data by setting $\mathbf{U}_i^{\partial\Omega,n} := \mathbf{U}_i^{\mathrm{sol},n}$ on the entirety of the boundary, where $\mathbf{U}_i^{\mathrm{sol},n}$ is the analytical solution,

(b) sub/super-sonic boundary conditions with exact data: $\mathbf{U}_i^{\mathrm{d},n} = \mathbf{U}_i^{\mathrm{sol},n}$,

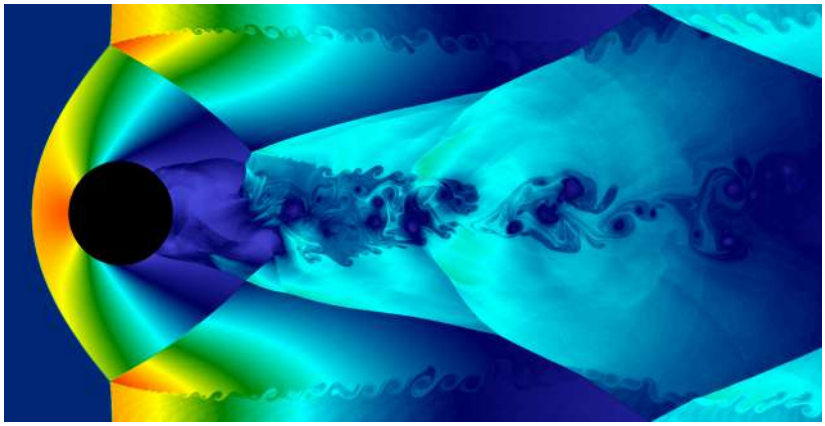
(c) sub/super-sonic boundary conditions with a far-field state: $\mathbf{U}_i^{\mathrm{d},n} = \mathbf{U}^{\mathrm{far},n}$,

(d) sub/super-sonic boundary conditions with the old state: we set $\mathbf{U}_i^{\mathrm{d},n} = \mathbf{U}_i^n$.

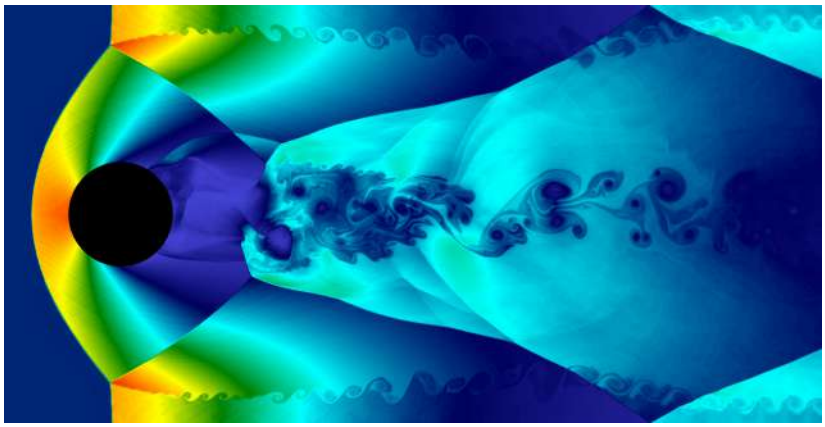
Strategies (a) and (b) are intended to evaluate the formal consistency of the method when the exact boundary data is available. On the other hand, strategies (c) and (d) are meant to evaluate the performance of the method when exact boundary data is not available. For strategies (c) and (d) no rates can be expected as we evaluate the error up to the boundary. The behavior of strategies (c) and (d) are of particular interest in the context of channel flows and transonic exterior aerodynamics. For such applications exact boundary data is unavailable and the specific subsonic or supersonic nature is unknown as well. The numerical results for the four strategies are summarized in Table 4 for the case of $\mathbb{Q}_1$ spatial discretization. We see that strategies (a) and (b) deliver the proper convergence rates. On the other hand, even though no rates should be expected for the case of strategies (c) and (d), we still observe proper convergence rates once the regime becomes fully supersonic ($M=2$ and $M=2.5$). This indicates that the sub/super-sonic boundary condition approach is indeed capable of selecting the boundary-data from the proper upwind direction.

6.4 High fidelity simulation: Mach 3 flow past a cylinder

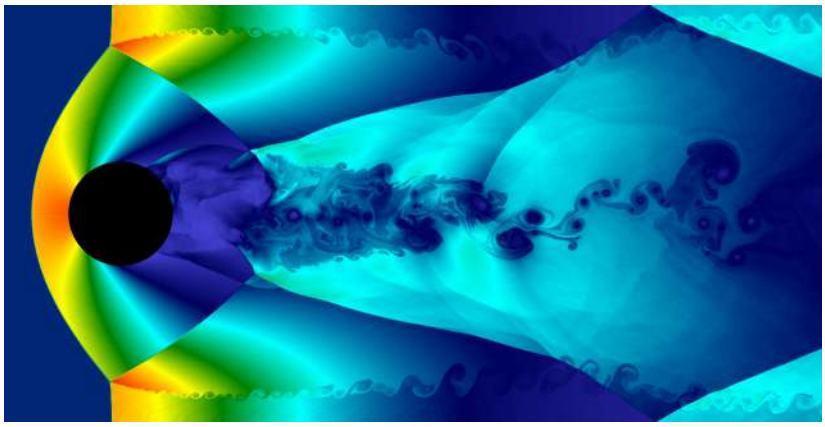
We now present numerical results for a 2D benchmark configuration consisting of a Mach 3 flow past a cylinder with radius 0.25 is centered along $(0.6,0,z)$. The computational domain is $\Omega=[0,4]\times[-1,1]$ and is equipped with Dirichlet boundary conditions on the left of the domain, slip boundary conditions on the cylinder and the top and bottom of the domain, and do nothing boundary outflow conditions on the right side. The initial flow configuration is that of a uniform flow at Mach 3 [20]. The computational domain is meshed with an unstructured quadrilateral coarse mesh. A higher resolution is obtained by subdividing every quadrilateral into 4 children an fixed number of times and adjusting newly generated nodes on the cylinder boundary to lie on the curved surface. Fig. 1 shows a temporal snapshot at time $t=4.0$. The computations where performed with a mesh consisting of 9.4M quadrilaterals corresponding to 9.4M degrees of freedom per component for $\mathbb{Q}^1$, and with a mesh consisting of 2.4M quadrilaterals for $\mathbb{Q}^2$ and $\mathbb{Q}^3$, corresponding to 5.3M ($\mathbb{Q}^2$) and 9.4M ($\mathbb{Q}^3$) degrees of freedom per component. We observe qualitatively that all spatial discretizations lead to a comparable results with well captured (unstable) contact discontinuities emerging from primary and secondary triple points.



(a) $\mathbb{Q}^1$, 9.4M DOFs per component



(b) $\mathbb{Q}^2$, 5.3M DOFs per component



(c) $\mathbb{Q}^3$, 9.4M DOFs per component

Figure 1: Temporal snapshot at time $t=4.0$ of the density profiles of a supersonic Mach 3 flow past a cylinder. Computed for increasing polynomial degree: (a) $\mathbb{Q}^1$, (b) $\mathbb{Q}^2$, (c) $\mathbb{Q}^3$. The density is visualized on a rainbow colormap to highlight discontinuities.

7 Conclusion and outlook

We have introduced a graph-based discontinuous Galerkin method for solving hyperbolic systems of conservation laws. The method has three main ingredients: a first-order scheme, a high-order scheme (based on the entropy-viscosity technique), and a convex-limiting procedure that blends the high and low order schemes. The first-order update satisfies both the invariant-domain property as well as a pointwise discrete entropy inequality for any entropy of the system. The resulting convex-limited scheme preserves the invariant set using relaxed local bounds.

A notable feature of the method is the direct incorporation of boundary conditions. The state at each node is guaranteed to be admissible provided the boundary data supplied to the scheme is admissible. For the case of the first-order method, this allows to prove invariant-set preservation as well as local entropy inequalities *including* the effect of boundary contributions. For the high-order and convex-limited scheme, we have tested the implementation of boundary conditions using the isentropic vortex test with sufficiently large final time, allowing interaction of the vortex with the boundary. If the boundary data is the exact analytical solution, the method delivers optimal convergence rates. On the other hand, if the boundary data consists of the far-field state or the data from the previous time step, the implementation is convergent in the fully supersonic regime.

The convex-limited scheme has been evaluated with a number of numerical tests ranging from a smooth analytic solution to a discontinuous one, observing expected convergence rates. The discontinuous test has verified robustness of our scheme and first-order convergence in the shock-hydrodynamics regime. Consistent with approximation theory, for the same number of global degrees of freedom, the lowest order $\mathbb{Q}_1$ ansatz offers the smallest L^1-error while having the lowest computational complexity, and comparatively large time-step sizes. Finally, the semi-smooth rarefaction test has verified a rather subtle aspect of high resolution methods, which is the ability to produce better than first-order rates for solutions that are continuous with second derivatives of bounded variation. Finally, we verified that the proposed method is suitable for high fidelity simulations with a 2D benchmark configuration of a Mach 3 flow past a cylinder.

Acknowledgments

This material is based upon work supported in part by the National Science Foundation (Grants DMS-2045636 (MM), DMS-2409841 (IT)), by the Air Force Office of Scientific Research, USAF (Grant/Contract number FA9550-23-1-0007 (MM)), and by the German Ministry of Education and Research through project "PDExa: Optimized software methods for solving partial differential equations on exascale supercomputers" (Grant 16ME0637K (MK)).

Appendix A Convergence tables

Table 1: Convex-limited scheme: (i) isentropic vortex test. Error delivered by scheme described by (5.6); see Section 5. We consider the cases of $\mathbb{Q}^k$ spatial discretizations for $k=1,2,3$.

L^1-error						
#DOFs	Q^1	rate	Q^2	rate	Q^3	rate
2304	2.255×10^{-2}		9.787×10^{-3}		3.354×10^{-3}	
9216	7.281×10^{-3}	1.63	1.149×10^{-3}	3.09	2.568×10^{-4}	3.71
36864	2.063×10^{-3}	1.82	1.562×10^{-4}	2.88	1.706×10^{-5}	3.91
147456	5.487×10^{-4}	1.91	2.012×10^{-5}	2.96	1.145×10^{-6}	3.9
589824	1.415×10^{-4}	1.96	2.539×10^{-6}	2.99	7.789×10^{-8}	3.88
2359296	3.594×10^{-5}	1.98	3.193×10^{-7}	2.99	5.496×10^{-9}	3.82
L^2-error						
#DOFs	Q^1	rate	Q^2	rate	Q^3	rate
2304	5.523×10^{-2}		2.082×10^{-2}		8.588×10^{-3}	
9216	1.834×10^{-2}	1.59	2.851×10^{-3}	2.87	6.845×10^{-4}	3.65
36864	5.338×10^{-3}	1.78	4.298×10^{-4}	2.73	4.530×10^{-5}	3.92
147456	1.455×10^{-3}	1.88	6.097×10^{-5}	2.82	3.280×10^{-6}	3.79
589824	3.822×10^{-4}	1.93	8.428×10^{-6}	2.85	2.390×10^{-7}	3.78
2359296	9.825×10^{-5}	1.96	1.178×10^{-6}	2.84	1.922×10^{-8}	3.64
L^∞-error						
#DOFs	Q^1	rate	Q^2	rate	Q^3	rate
2304	3.467×10^{-1}	–	1.446×10^{-1}	–	1.028×10^{-1}	–
9216	1.298×10^{-1}	1.42	3.532×10^{-2}	2.03	6.514×10^{-3}	3.98
36864	4.697×10^{-2}	1.47	9.308×10^{-3}	1.92	5.469×10^{-4}	3.57
147456	1.695×10^{-2}	1.47	1.826×10^{-3}	2.35	6.232×10^{-5}	3.13
589824	5.613×10^{-3}	1.59	3.715×10^{-4}	2.30	5.252×10^{-6}	3.57
2359296	1.750×10^{-3}	1.68	7.391×10^{-5}	2.33	5.763×10^{-7}	3.19

Table 2: Convex-limited scheme: (ii) rarefaction test. Convergence rates for scheme (5.6). We consider the cases of $\mathbb{Q}^k$ spatial discretizations for $k=1,2,3$. The number of elements for each case $k=1,2,3$ has been chosen in a way that the number of degrees of freedom of each refinement case match exactly. Note that the best expected rate for this test is $\mathcal{O}(h^2)$ for all polynomial degrees.

L^1-error						
#DOFs	Q^1	rate	Q^2	rate	Q^3	rate
120	1.772×10^{-3}	–	1.068×10^{-3}	–	3.001×10^{-4}	–
240	5.174×10^{-4}	1.78	2.813×10^{-4}	1.92	1.029×10^{-4}	1.54
480	1.564×10^{-4}	1.73	1.121×10^{-4}	1.33	3.455×10^{-5}	1.57
960	4.628×10^{-5}	1.76	3.542×10^{-5}	1.66	1.158×10^{-5}	1.58
1920	1.301×10^{-5}	1.83	1.269×10^{-5}	1.48	3.946×10^{-6}	1.55
3840	4.204×10^{-6}	1.63	3.863×10^{-6}	1.72	1.049×10^{-6}	1.91
7680	1.365×10^{-6}	1.62	1.423×10^{-6}	1.44	3.573×10^{-7}	1.55
15360	4.516×10^{-7}	1.6	4.335×10^{-7}	1.71	1.070×10^{-7}	1.74

Table 3: Convex-limited scheme: (iii) LeBlanc test. Error delivered by scheme described by (5.6); see Section 5. We consider the cases of $\mathbb{Q}^k$ spatial discretizations for $k=1,2,3$.

L^1-error						
#DOFs	Q^1	rate	Q^2	rate	Q^3	rate
120	1.362×10^{-1}	–	1.256×10^{-1}	–	1.158×10^{-1}	–
240	9.516×10^{-2}	0.52	6.863×10^{-2}	0.87	5.957×10^{-2}	0.96
480	5.519×10^{-2}	0.79	3.840×10^{-2}	0.84	3.995×10^{-2}	0.58
960	3.195×10^{-2}	0.79	2.903×10^{-2}	0.4	2.052×10^{-2}	0.96
1920	1.639×10^{-2}	0.96	1.441×10^{-2}	1.01	1.214×10^{-2}	0.76
3840	8.704×10^{-3}	0.91	8.494×10^{-3}	0.76	6.559×10^{-3}	0.89
7680	4.324×10^{-3}	1.01	4.730×10^{-3}	0.84	4.345×10^{-3}	0.59
15360	2.290×10^{-3}	0.92	2.626×10^{-3}	0.85	2.296×10^{-3}	0.92

Appendix B Implementation of boundary conditions: Adaptation to continuous finite elements

The scheme (3.1)-(3.2) and the mathematical theory developed in this manuscript is entirely valid for the case of continuous finite elements. However, slight changes are required in definitions of (2.10)-(2.11). Here provide the required mathematical statements but avoid doing the proofs since they are just adaptations of the proof already presented for the discontinuous case in Section 3.

In the context of continuous finite elements we have to use the following definitions:

$$\mathbf{c}_{ij} := \int_\Omega \nabla\phi_j\phi_i \mathrm{d}\boldsymbol{x} - \frac{1}{2}\int_{\partial\Omega}\phi_j\phi_i\boldsymbol{n}_{\partial\Omega}\mathrm{d}\boldsymbol{s}, \quad \mathbf{c}_i^{\partial\Omega} = \frac{1}{2}\int_{\partial\Omega}\phi_i\boldsymbol{n}_{\partial\Omega}\mathrm{d}\boldsymbol{s}. \tag{B.1}$$

Note that the face integral $(1/2)\int_{\partial\Omega}\phi_j\phi_i\boldsymbol{n}_{\partial\Omega}\mathrm{d}\boldsymbol{s}$ can only be nonzero if both ϕ_j and ϕ_i have support on the boundary.

Proposition B.1. The vectors $\mathbf{c}_{ij}$ as defined in (B.1) satisfy the usual skew-symmetry property $\mathbf{c}_{ij} = -\mathbf{c}_{ji}$ for all $i,j\in\mathcal{V}$.

Proof. The proof follows by integration by parts arguments. However, in this case, since the shape functions $\{\phi_i\}_{i\in\mathcal{V}}$ are compactly supported and weakly differentiable in Ω, we do not need to use integration by parts on each element K, but rather integration by parts in entire domain Ω, see also [25]. □

Proposition B.2 (Partition of Unity Properties)**.** The vectors $\mathbf{c}_{ij}$ and $\mathbf{c}_i^{\partial\Omega}$ as defined in (B.1) satisfy the partition of unity property $\sum_{j\in\mathcal{I}(i)}\mathbf{c}_{ij}+\mathbf{c}_i^{\partial\Omega}=\mathbf{0}$.

The proof follows using arguments similar to those of Lemma 3.1.

Lemma B.1 (Total Balance)**.** *The scheme* (3.1) *with* $\mathbf{c}_{ij}$ *and* $\mathbf{c}_i^{\partial\Omega}$ *as defined in* (B.1) *satisfies the flux-balance* (3.7).

The proof of this lemma is omitted since it is identical to the proof of Lemma 3.1.

Table 4: Boundary condition validation: modified isentropic vortex test. We use a modified isentropic vortex test to verify the accuracy of the boundary value enforcement outlined in Section 4. We test four different strategies for the construction of boundary data: (a) exact Dirichlet data, (b) sub/super-sonic boundaries with exact analytical data, (c) sub/super-sonic boundaries with far-field state; and (d) sub/super-sonic boundaries using data from the previous time-step; see Section 6.3. The test is repeated for different Mach numbers, $M=1$, 1.5, 2.0, and 2.5. In every case we use $\mathbb{Q}_1$ spatial discretization. We note that for $M=1$ and $M=1.5$ a significant portion of the outflow boundary (top and right edges of the domain) is subsonic. Strategies (a) and (b) deliver proper convergence rates. Strategies (c) and (d) are only convergent for sufficiently large Mach numbers.

Strategy (a): exact Dirichlet data								
#DOFs	$M=1.0$	rate	$M=1.5$	rate	$M=2.0$	rate	$M=2.5$	rate
2304	4.37E-03	-	3.05E-03	-	3.05E-03	-	2.79E-03	-
9216	8.63E-04	2.34	6.26E-04	2.28	6.13E-04	2.31	5.68E-04	2.29
36864	1.78E-04	2.27	1.34E-04	2.21	1.26E-04	2.28	1.15E-04	2.29
147456	4.03E-05	2.14	3.15E-05	2.09	2.85E-05	2.14	2.60E-05	2.15
589824	9.61E-06	2.06	7.63E-06	2.04	6.79E-06	2.07	6.18E-06	2.07
2359296	2.45E-06	1.97	1.87E-06	2.02	1.65E-06	2.03	1.50E-06	2.03
Strategy (b): sub/super-sonic boundary conditions with exact Dirichlet data								
#DOFs	$M=1.0$	rate	$M=1.5$	rate	$M=2.0$	rate	$M=2.5$	rate
2304	4.05E-03	-	2.98E-03	-	2.41E-03	-	1.99E-03	-
9216	8.36E-04	2.27	6.19E-04	2.270	5.15E-04	2.22	4.29E-04	2.21
36864	1.79E-04	2.22	1.33E-04	2.210	1.13E-04	2.18	9.88E-05	2.12
147456	4.12E-05	2.12	3.13E-05	2.094	2.67E-05	2.08	2.37E-05	2.05
589824	9.92E-06	2.05	7.60E-06	2.044	6.52E-06	2.03	5.85E-06	2.02
2359296	2.54E-06	1.96	1.87E-06	2.024	1.60E-06	2.01	1.45E-06	2.01
Strategy (c): sub/super-sonic boundary conditions with far-field approximation								
#DOFs	$M=1.0$	rate	$M=1.5$	rate	$M=2.0$	rate	$M=2.5$	rate
2304	1.36E-02	-	4.09E-03	-	2.41E-03	-	1.99E-03	-
9216	1.07E-02	0.35	1.70E-03	1.26	5.15E-04	2.22	4.29E-04	2.21
36864	9.65E-03	0.14	1.14E-03	0.56	1.14E-04	2.17	9.88E-05	2.12
147456	9.20E-03	0.06	1.01E-03	0.17	2.73E-05	2.05	2.37E-05	2.05
589824	8.98E-03	0.03	9.31E-04	0.12	6.86E-06	1.99	5.85E-06	2.02
2359296	8.86E-03	0.02	8.72E-04	0.09	1.79E-06	1.93	1.45E-06	2.01
Strategy (d): sub/super-sonic boundary conditions with current state								
#DOFs	$M=1.0$	rate	$M=1.5$	rate	$M=2.0$	rate	$M=2.5$	rate
2304	6.12E-03	-	3.02E-03	-	2.41E-03	-	2.03E-03	-
9216	3.99E-03	0.61	6.58E-04	2.20	5.08E-04	2.24	4.33E-04	2.22
36864	3.59E-03	0.15	1.76E-04	1.90	1.13E-04	2.16	9.88E-05	2.13
147456	1.05E-02	-1.56	7.09E-05	1.31	2.67E-05	2.08	2.37E-05	2.05
589824	3.97E-03	1.41	4.45E-05	0.67	6.52E-06	2.03	5.85E-06	2.02
2359296	4.66E-03	-0.22	3.64E-05	0.29	1.60E-06	2.01	1.45E-06	2.01

Table 5: Error of the first-order method with respect polynomial degree. This tables illustrates the growth of the L^1-error of the first-order scheme (3.1)-(3.2) as the polynomial degree grows for the isentropic vortex problem. For every polynomial degree the error halves with each mesh refinement (as expected from a first-order scheme). However, the error constant, or "error pre-factor", grows with the polynomial degree.

#DOFs	$\mathbb{Q}_1$	$\mathbb{Q}_2$	$\mathbb{Q}_3$
2304	1.2799e-01	1.5639e-01	1.7400e-01
9216	8.5427e-02	1.1676e-01	1.4080e-01
36864	5.1002e-02	7.6054e-02	9.8985e-02
147456	2.8281e-02	4.4705e-02	6.1502e-02
589824	1.4981e-02	2.4553e-02	3.5023e-02

References

[1] M. Ainswort, Dispersive behaviour of high order finite element schemes for the one-way wave equation, *J. Comput. Phys.*, 259:1–10, 2014.

[2] D. Arndt, W. Bangerth, M. Bergbauer, M. Feder, M. Fehling, J. Heinz, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, B. Turcksin, D. Wells, and S. Zampini, *The `deal.II` library, version 9.5*, *J. Numer. Math.*, 31:231–246, 2023.

[3] F. Bassi and S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, *J. Comput. Phys.*, 138:251–285, 1997.

[4] J. P. Boris and D. L. Book, Flux-corrected transport, *J. Comput. Phys.*, 135:172–186, 1997.

[5] K. N. Chueh, C. C. Conley, and J. A. Smoller, Positively invariant regions for systems of nonlinear diffusion equations, *Indiana Univ. Math. J.*, 26:373–392, 1977.

[6] P. G. Ciarlet, The finite element method for elliptic problems, in: *Studies in Mathematics and its Applications*, Vol. 4, North-Holland Publishing Co., 1978.

[7] B. Clayton, J.-L. Guermond, M. Maier, B. Popov, and E. J. Tovar, Robust second-order approximation of the compressible Euler equations with an arbitrary equation of state, *J. Comput. Phys.*, 478:111926, 2023.

[8] B. Cockburn and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework, *Math. Comp.*, 52:411–435, 1989.

[9] L. Demkowicz, J. T. Oden, and W. Rachowicz, A new finite element method for solving compressible Navier-Stokes equations based on an operator splitting method and *h-p* adaptivity, *Comput. Methods Appl. Mech. Engrg.*, 84:275–326, 1990.

[10] A. Ern and J.-L. Guermond, *Theory and Practice of Finite Elements*, in: *Applied Mathematical Sciences*, Vol. 159, Springer-Verlag, 2004.

[11] N. Fehn, W. A. Wall, and M. Kronbichler, A matrix-free high-order discontinuous Galerkin compressible Navier-Stokes solver: A performance comparison of compressible and incompressible formulations for turbulent incompressible flows, *Internat. J. Numer. Methods Fluids*, 89:71–102, 2019.

[12] T. C. Fisher and M. H. Carpenter, High-order entropy stable finite difference schemes for nonlinear conservation laws: Finite domains, *J. Comput. Phys.*, 252:518–557, 2013.

[13] U. S. Fjordholm, S. Mishra, and E. Tadmor, Arbitrarily high-order accurate entropy stable essentially nonoscillatory schemes for systems of conservation laws, *SIAM J. Numer. Anal.*, 50:544–573, 2012.

[14] C. A. J. Fletcher, The group finite element formulation, *Comput. Methods Appl. Mech. Engrg.*, 37:225–244, 1983.
[15] H. Frid, Maps of convex sets and invariant regions for finite-difference systems of conservation laws, *Arch. Ration. Mech. Anal.*, 160:245–269, 2001.
[16] E. Godlewski and P.-A. Raviart, Hyperbolic systems of conservation laws, in: *Mathématiques & Applications*, Vol. 3/4, Ellipses, 1991.
[17] J.-L. Guermond, C. Kees, B. Popov, and E. Tovar, Well-balanced second-order convex limiting technique for solving the Serre-Green-Naghdi equations, *Water Waves*, 4:409–445, 2022.
[18] J.-L. Guermond, M. Kronbichler, M. Maier, B. Popov, and 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. Engrg.*, 389:114250, 2022.
[19] J.-L. Guermond, M. Nazarov, and B. Popov, Finite element-based invariant-domain preserving approximation of hyperbolic systems: Beyond second-order accuracy in space, *Comput. Methods Appl. Mech. Engrg.*, 418:116470, 2024.
[20] J.-L. Guermond, M. Nazarov, B. Popov, and I. Tomas, Second-order invariant domain preserving approximation of the Euler equations using convex limiting, *SIAM J. Sci. Comput.*, 40:A3211–A3239, 2018.
[21] J.-L. Guermond, M. Nazarov, B. Popov, and Y. Yang, A second-order maximum principle preserving Lagrange finite element technique for nonlinear scalar conservation equations, *SIAM J. Numer. Anal.*, 52:2163–2182, 2014.
[22] J.-L. Guermond, R. Pasquetti, and B. Popov, Entropy viscosity method for nonlinear conservation laws, *J. Comput. Phys.*, 230:4248–4267, 2011.
[23] J.-L. Guermond and B. Popov, Viscous regularization of the Euler equations and entropy principles, *SIAM J. Appl. Math.*, 74:284–305, 2014.
[24] J.-L. Guermond and B. Popov, Invariant domains and first-order continuous finite element approximation for hyperbolic systems, *SIAM J. Numer. Anal.*, 54:2466–2489, 2016.
[25] J.-L. Guermond, B. Popov, and I. Tomas, Invariant domain preserving discretization-independent schemes and convex limiting for hyperbolic systems, *Comput. Methods Appl. Mech. Engrg.*, 347:143–175, 2019.
[26] A. Harten, P. D. Lax, C. D. Levermore, and W. J. Morokoff, Convex entropies and hyperbolicity for general Euler equations, *SIAM J. Numer. Anal.*, 35:2117–2127, 1998.
[27] A. Harten and G. Zwas, Self-adjusting hybrid schemes for shock computations, *J. Comput. Phys.*, 9:568–583, 1972.
[28] R. Hartmann and P. Houston, Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations, *J. Comput. Phys.*, 183:508–532, 2002.
[29] G. W. Hedstrom, Nonreflecting boundary conditions for nonlinear hyperbolic systems, *J. Comput. Phys.*, 30:222–237, 1979.
[30] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, in: *Texts in Applied Mathematics*, Springer, 2008.
[31] D. Hoff, Invariant regions for systems of conservation laws, *Trans. Amer. Math. Soc.*, 289:591–610, 1985.
[32] A. Huerta, E. Casoni, and J. Peraire, A simple shock-capturing technique for high-order discontinuous Galerkin methods, *Internat. J. Numer. Methods Fluids*, 69:1614–1632, 2012.
[33] T. J. R. Hughes, L. P. Franca, and M. Mallet, A new finite element formulation for computational fluid dynamics. I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics, *Comput. Methods Appl. Mech. Engrg.*, 54:223–234, 1986.

[34] B. Khobalatte and B. Perthame, Maximum principle on the entropy and second-order kinetic schemes, *Math. Comp.*, 62:119–131, 1994.
[35] M. Kronbichler and K. Kormann, Fast matrix-free evaluation of discontinuous Galerkin finite element operators, *ACM Trans.Math. Softw.*, 45:29, 2019.
[36] D. Kuzmin, Monolithic convex limiting for continuous finite element discretizations of hyperbolic conservation laws, *Comput. Methods Appl. Mech. Engrg.*, 361:112804, 2020.
[37] D. Kuzmin and R. Löhner, and S. Turek, *Flux-Corrected Transport: Principles, Algorithms, and Applications*, Springer, 2012.
[38] D. Kuzmin, R. Löhner, and S. Turek, *Flux-Corrected Transport: Principles, Algorithms, and Applications*, in: *Scientific Computation*, Springer, 2012.
[39] D. Kuzmin and M. Quezada de Luna, Subcell flux limiting for high-order Bernstein finite element discretizations of scalar hyperbolic conservation laws, *J. Comput. Phys.*, 411:109411, 2020.
[40] Y. Lin, J. Chan, and I. Tomas, A positivity preserving strategy for entropy stable discontinuous Galerkin discretizations of the compressible Euler and Navier-Stokes equations, *J. Comput. Phys.*, 475:111850, 2023.
[41] C. Lohmann, D. Kuzmin, J. N. Shadid, and S. Mabuza, Flux-corrected transport algorithms for continuous Galerkin methods based on high order Bernstein finite elements, *J. Comput. Phys.*, 344:151–186, 2017.
[42] R. Löhner, *Applied Computational Fluid Dynamics Techniques: An Introduction Based on Finite Element Methods*, John Wiley & Sons, 2008.
[43] M. Maier and 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.*, 8:16, 2021.
[44] M. Maier, J. N. Shadid, and I. Tomas, Structure-preserving finite-element schemes for the Euler-Poisson equations, *Commun. Comput. Phys.*, 33:647–691, 2023.
[45] C. Michoski, C. Dawson, E. J. Kubatko, D. Wirasaet, S. Brus, and J. J. Westerink, A comparison of artificial viscosity, limiters, and filters, for high order discontinuous Galerkin solutions in nonlinear settings, *J. Sci. Comput.*, 66:406–434, 2016.
[46] W. Pazner, Sparse invariant domain preserving discontinuous Galerkin methods with subcell convex limiting, *Comput. Methods Appl. Mech. Engrg.*, 382:113876, 2021.
[47] P.-O. Persson and J. Peraire, Sub-cell shock capturing for discontinuous Galerkin methods, in: 44th AIAA Aerospace Sciences Meeting and Exhibit, 112, 2006.
[48] B. Popov and Y. Hua, Invariant domain preserving central schemes for nonlinear hyperbolic systems, *Commun. Math. Sci.*, 19:529–556, 2021.
[49] H. Ranocha, M. Schlottke-Lakemper, J. Chan, A. M. Rueda-Ramírez, A. R. Winters, F. Hindenlang, and G. J. Gassner, Efficient implementation of modern entropy stable and kinetic energy preserving discontinuous Galerkin methods for conservation laws, *ACM Trans.Math. Softw.*, 49:37, 2023.
[50] A. M. Rueda-Ramírez, W. Pazner, and G. J. Gassner, Subcell limiting strategies for discontinuous Galerkin spectral element methods, *Comput. & Fluids*, 247:105627, 2022.
[51] V. Selmin, The node-centred finite volume approach: bridge between finite differences and finite elements, *Comput. Methods Appl. Mech. Engrg.*, 102:107–138, 1993.
[52] C.-W. Shu, TVB uniformly high-order schemes for conservation laws, *Math. Comp.*, 49:105–121, 1987.
[53] R. J. Spiteri and S. J. Ruuth, A new class of optimal high-order strong-stability-preserving time discretization methods, *SIAM J. Numer. Anal.*, 40:469–491, 2002.

[54] Z. Sun and Y. Xing, On generalized Gauss-Radau projections and optimal error estimates of upwind-biased DG methods for the linear advection equation on special simplex meshes, *J. Sci. Comput.*, 95:40, 2023.
[55] P. K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws, *SIAM J. Numer. Anal.*, 21:995–1011, 1984.
[56] T. E. Tezduyar and M. Senga, Stabilization and shock-capturing parameters in SUPG formulation of compressible flows, *Comput. Methods Appl. Mech. Engrg.*, 195:1621–1632, 2006.
[57] I. Tomas, J. Shadid, M. Maier, and A. Salgado, Asymptotic preserving methods for fluid electron-fluid models in the large magnetic field limit with mathematically guaranteed properties (Final Report), 2022. doi: https://doi.org/10.2172/1872178.
[58] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, Springer-Verlag, 2009.
[59] H. C. Yee, N. D. Sandham, and M. J. Djomehri, Low-dissipative high-order shock-capturing methods using characteristic-based filters, J. Comput. Phys., 150:199–238, 1999.
[60] J. Yu and J. S. Hesthaven, A study of several artificial viscosity models within the discontinuous Galerkin framework, *Commun. Comput. Phys.*, 27:1309–1343, 2020.
[61] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comput. Phys.*, 31:335–362, 1979.
[62] X. Zhang and C.-W. Shu, On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes, *J. Comput. Phys.*, 229:8918–8934, 2010.
[63] V. Zingan, J.-L. Guermond, J. Morel, and B. Popov, Implementation of the entropy viscosity method with the discontinuous Galerkin method, *Comput. Methods Appl. Mech. Engrg.*, 253:479–490, 2013.