Continuously bounds-preserving discontinuous Galerkin methods for hyperbolic conservation laws

T. Dzanic

January 5, 2024

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Continuously bounds-preserving discontinuous Galerkin methods for hyperbolic conservation laws

T. Dzanic

January 5, 2024

Disclaimer

Continuously bounds-preserving discontinuous Galerkin methods for hyperbolic conservation laws

T. Dzanic[a,*]

[a]*Lawrence Livermore National Laboratory, Livermore, CA 94551, United States of America*

ARTICLE INFO

ABSTRACT

For finite element approximations of transport phenomena, it is often necessary to apply a form of limiting to ensure that the discrete solution remains well-behaved and satisfies physical constraints. However, these limiting procedures are typically performed at discrete nodal locations, which is not sufficient to ensure the robustness of the scheme when the solution must be evaluated at arbitrary locations (e.g., for adaptive mesh refinement, remapping in arbitrary Lagrangian–Eulerian solvers, overset meshes, etc.). In this work, a novel limiting approach for discontinuous Galerkin methods is presented which ensures that the solution is *continuously* bounds-preserving (i.e., across the entire solution polynomial) for any arbitrary choice of basis, approximation order, and mesh element type. Through a modified formulation for the constraint functionals, the proposed approach requires only the solution of a single spatial scalar minimization problem per element for which a highly efficient numerical optimization procedure is presented. The efficacy of this approach is shown in numerical experiments by enforcing continuous constraints in high-order unstructured discontinuous Galerkin discretizations of hyperbolic conservation laws, ranging from scalar transport with maximum principle preserving constraints to compressible gas dynamics with positivity-preserving constraints.

1. Introduction

The efficient and accurate computation of complex transport phenomena remains a driving force in the development of high-fidelity numerical discretization techniques. Within this broad application area, the use of high-order schemes has grown in prevalence, primarily due to their ability to obtain higher accuracy per degree of freedom which is advantageous in the context of scale-resolving simulations. Various high-order numerical methods have been routinely applied to simulating transport phenomena, with these schemes typically falling within the overarching classes of finite difference, finite volume, and finite element methods. Of these approaches, finite element methods offer notable benefits as they can achieve arbitrarily high-order accuracy and are straightforward to generalize to unstructured meshes. One particular formulation of finite element methods, discontinuous Galerkin (DG) [1] schemes and variants thereof (e.g., flux reconstruction [2]), offers further advantages as they are typically better suited for approximating transport phenomena and possess a compact data structure that is beneficial for modern massively-parallel high performance computing architectures. As such, these characteristics have resulted in the successful application of DG schemes to a wide variety of problems ranging from fluid dynamics and astrophysics to solid mechanics and bioengineering [3–6].

However, the accuracy and robustness of DG schemes (and finite element methods in general) can become severely degraded when the physical systems in question must abide by strict physical constraints, which is frequently the case in transport-dominated applications (e.g., positivity of density and internal energy in gas dynamics, subluminal velocity in relativistic magnetohydrodynamics, bounded mass fractions in multi-species flows, etc.). Standard numerical implementations generally do not guarantee that these constraints are satisfied by the discrete solution, which may result in inaccurate and physically inconsistent predictions or, more typically, the failure of the scheme altogether. This problem is further exacerbated in more complex applications such as multi-physics modeling, where the strong coupling between various physical models can be particularly sensitive to ill-behaved numerical solutions.

A typical remedy for this problem is to enforce these constraints via some form of *a posteriori* limiting on the discrete solution to ensure that the solution remains *bounds-preserving* (i.e., constraint-satisfying). These methods generally rely on utilizing a secondary less accurate (but more robust) numerical approximation of the governing equations

*Corresponding author
✉ dzanic1@llnl.gov (T. Dzanic)
ORCID(s): 0000-0003-3791-1134 (T. Dzanic)

which is guaranteed to be bounds-preserving. The solutions as predicted by the high-order baseline scheme and the low-order secondary scheme are then combined to yield a bounds-preserving approximation that ostensibly retains the accuracy of the high-order scheme. Within the overarching framework of limiting methods, the quintessential example is the flux-corrected transport method of Boris and Book [7] from which a variety of approaches are derived [8–16]. Alternate limiting formulations which do not rely on a secondary low-order scheme have also been developed by using bounds-preserving estimates derived from the high-order solution [17–24] or through slope-limiting/stencil modification of the underlying solution [25–27].

However, a critical drawback of these limiting approaches is that they are typically performed *discretely*, whereas the solution for the DG approximation is represented (piecewise) continuously. The consequence is that the schemes are only provably bounds-preserving at the discrete spatial locations where the limiting was performed. For many applications, this is not sufficient – it is sometimes necessary to have to evaluate the solution at arbitrary locations. For example, adaptive mesh refinement frequently requires evaluating the solution at a new set of quadrature points which do not coincidence with the quadrature points used for the element. This is similarly the case for coupled multi-physics solvers that rely on independent meshes, where the solution must be frequently transferred between these meshes. In some cases, the location that the solution must be evaluated at may not even be known *a priori*, such as in the remap stage of an arbitrary Lagrangian–Eulerian solver or with dynamic overset meshes. In other cases, the solution may not even be queried in real time, such as when coupling of physical models is done as a post-processing step after the end of the simulation, making it impossible to perform *a posteriori* limiting based on the location where the solution will be evaluated. For such applications, the limited solution may still violate the constraints as it is not bounds-preserving at these arbitrary points, causing the failure of the scheme and/or the breakdown of physical models.

This issue is typically counteracted by changing the basis of the underlying finite element method such that the solution remains continuously bounded (i.e., across the entire solution polynomial). One such *continuously bounds-preserving* approach can be obtained by ensuring that the solution approximation is at most linear, but this effectively limits the scheme to second-order accuracy. Alternatively, one may use bounded shape functions such Bernstein polynomials with appropriate limiting to guarantee that the solution is continuously bounds-preserving even for high-order approximations [9, 28]. However, the use of bounded Bernstein polynomials for the solution basis gives rise to some notable drawbacks, such as numerical difficulties due to the ill-conditioned basis transformation operation and the lack of flexibility of the underlying approximation space which does not easily extend to more complex bases (e.g., divergence-free basis functions, rational basis functions, etc.). Additionally, limiting based on the convex hull property of Bernstein polynomials can introduce excessive numerical dissipation as it generally only provides a *sufficient* degree of limiting, not the *necessary* degree of limiting, even for linear constraints and for solutions which already continuously satisfy constraints. As such, there is potential to significantly decrease the computational cost and increase the accuracy and applicability of DG schemes if the underlying discretization can be modified such as to continuously enforce constraints for general DG discretizations.

In this work, a novel *a posteriori* limiting procedure for DG schemes is presented which ensures that the solution is continuously bounds-preserving for any arbitrary choice of basis, approximation order, and mesh element type. The proposed method extends the limiting approach of Zhang and Shu [17] to continuously enforce general algebraic constraints on the high-order solution. We show that limiting based off of the continuous extrema of the constraint functionals is not sufficient to enforce constraints continuously and propose a modified formulation for constraints which can guarantee a continuously bounds-preserving solution with only a single spatial minimization problem per element. This approach enforces the exact amount of limiting necessary for linear constraints and a sufficient amount of limiting for nonlinear constraints. Furthermore, a highly efficient numerical optimization procedure for solving the required minimization problem is presented, although the non-convexity of the problem does not necessarily mathematically guarantee convergence to the global minimum. The proposed approach is applied to a high-order unstructured DG scheme for hyperbolic conservation laws ranging from scalar transport to compressible gas dynamics.

The remainder of this manuscript is organized as follows. In Section 2, some preliminaries on hyperbolic conservation laws, convex invariants and constraints, and DG schemes are presented. The proposed numerical approach is then introduced in Section 3, followed by numerical implementation details in Section 4 and the results of the numerical experiments in Section 5. Conclusions are then drawn in Section 6.

2. Preliminaries

2.1. Hyperbolic conservation laws and convex constraints
This work pertains to hyperbolic conservation laws of the form

$$\begin{cases} \partial_t \mathbf{u}(\mathbf{x}, t) + \nabla \cdot \mathbf{F}(\mathbf{u}) = 0, & \text{for } (\mathbf{x}, t) \in \Omega \times \mathbb{R}_+, \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), & \text{for } \mathbf{x} \in \Omega, \end{cases} \tag{1}$$

where $\Omega \subset \mathbb{R}^d$ is a d-dimensional spatial domain, $\mathbf{u} \in \mathbb{R}^m$ is a vector-valued solution of m variables, and $\mathbf{F}(\mathbf{u}) \in \mathbb{R}^{m \times d}$ is the associated flux. We assume that there exists a contiguous set of *admissible* solutions to Eq. (1), denoted by $\mathcal{G}$, which possess certain properties that are desirable for numerical approximations of these problems. Without giving a precise definition to what an admissible solution may be, which in many cases remains an open problem, we assume that these admissible solutions must abide by some strict constraints, formalized by the positivity of some set of N_c linear or nonlinear constraint functionals of $\mathbf{u}$ as

$$g_i(\mathbf{u}) \geq 0 \,\, \forall \, i \in \{1, \dots, N_c\}. \tag{2}$$

Furthermore, we assume that this set $\mathcal{G}$ is convex, such that the constraint functionals are quasiconcave (i.e., $\mathcal{G}$ is the upper contour set of quasiconcave functionals). The convexity of the admissible set of solutions is closely related to the invariant region of many hyperbolic conservation laws (e.g., the set of solutions corresponding to the positivity of density and internal energy in gas dynamics is convex), the details of which have been explored by Lax [29], Hoff [30], and Frid [31]. The goal of this work is to ensure that the numerical approximation of these systems satisfies the given constraints stemming from convex invariants of the system. For brevity, only weak inequalities are considered, but this work readily extends to strict inequalities as they are effectively identical from the perspective of finite precision arithmetic.

2.2. Discontinuous Galerkin methods
The numerical method of choice for approximating Eq. (1) is the DG method [1]. In this approach, the domain Ω is partitioned into N elements Ω_k such that $\Omega = \bigcup_N \Omega_k$ and $\Omega_i \cap \Omega_j = \emptyset$ for $i \neq j$. With a slight abuse of notation, the discrete solution $\mathbf{u}(\mathbf{x})$ within each element Ω_k is approximated via a summation of a set of n_s basis functions as

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{n_s} \mathbf{u}_i \phi_i(\mathbf{x}) \subset V_h, \tag{3}$$

where $\phi_i(\mathbf{x})$ are the basis functions, $\mathbf{u}_i$ are their associated coefficients, and V_h is the piece-wise polynomial space spanned by the basis functions. For polynomial approximations, which are the focus of this work, we denote $\mathbb{P}_p$ as the order of the approximation, where p is the maximal order of $\mathbf{u}(\mathbf{x})$.

The DG formulation involves integrating Eq. (1) with respect to a test function $\mathbf{w}(\mathbf{x}) \subset V_h$, which resides in the same finite element space as $\mathbf{u}(\mathbf{x})$, yielding

$$\sum_{k=1}^{N} \left\{ \int_{\Omega_k} \partial_t \mathbf{u} \cdot \mathbf{w} \, dV + \int_{\partial \Omega_k} \hat{\mathbf{F}}(\mathbf{u}^-, \mathbf{u}^+, \mathbf{n}) \cdot \mathbf{w} \, dS - \int_{\Omega_k} \mathbf{F}(\mathbf{u}) \cdot \nabla \mathbf{w} \, dV \right\} = 0. \tag{4}$$

Along the element interfaces $\partial \Omega$, the flux function is replaced by a numerical interface flux $\hat{\mathbf{F}}(\mathbf{u}^-, \mathbf{u}^+, \mathbf{n})$. This interface flux is dependent on the solution within the element of interest (denoted by the $\mathbf{u}^-$), its face-adjacent neighbor (denoted by $\mathbf{u}^+$), and the outward facing normal vector $\mathbf{n}$. Due to the discontinuous nature of the solution across the element interfaces, the flux is generally computed using exact [32] or approximate Riemann solvers such as the approach of Rusanov [33].

The DG scheme possesses a highly useful property in that the evolution of the element-wise mean solution $\bar{\mathbf{u}}$, defined as

$$\bar{\mathbf{u}} = \frac{\int_{\Omega_k} \mathbf{u}(\mathbf{x}) \, d\mathbf{x}}{\int_{\Omega_k} d\mathbf{x}}, \tag{5}$$

is dependent on only the interface flux, i.e.,

$$\partial_t \overline{\mathbf{u}} = - \int_{\partial \Omega_k} \hat{\mathbf{F}}(\mathbf{u}^-, \mathbf{u}^+, \mathbf{n}) \, \mathrm{d}S. \tag{6}$$

This yields a critical property in which the temporal update of the element-wise mean is equivalent to a convex combination of temporal updates of first-order Godunov schemes [34], which, under some relatively minor assumptions on the numerical scheme (e.g., CFL condition, appropriate Riemann solver, strong stability preserving time integration, appropriate quadrature, etc.), ensures that the element-wise mean preserves convex invariants of the system [17]. We remark here that while the focus of this work is on explicit DG schemes for hyperbolic conservation laws without source terms, the proposed method may be directly extended to any system for which the DG scheme preserves invariants of on the element-wise mean, which allows for its application to implicit schemes [35], mixed hyperbolic-parabolic systems [36], and certain hyperbolic systems with source terms which possess these properties on the element-wise mean [37]. This property of the element-wise mean plays a key role in many numerical stabilization methods for high-order DG schemes [17, 19–21, 24, 35, 36, 38], and the reader is referred to the aforementioned series of works stemming from Zhang and Shu [17] for a more in-depth overview on the sufficient conditions for this property. Furthermore, the proposed approach can be combined with any other approach which ensures that the element-wise mean satisfies a given set of physical constraints, including additional limiting/stabilization methods which satisfy the constraints on a set of discrete quadrature points or solution post-processing operations such as coarsening for adaptive mesh refinement.

3. Methodology

In this section, the proposed limiting formulation for enforcing continuous constraints in DG schemes is first introduced. Afterwards, the numerical optimization procedure for performing the limiting and an overview of various possible constraint functionals are then presented.

3.1. Numerical formulation

The underlying limiting method of the proposed approach is the *a posteriori* "squeeze" limiter of Zhang and Shu [17] which was introduced for the purposes of *discretely* enforcing convex constraints in DG schemes. In this approach, the limiter acts to uniformly contract the high-order solution $\mathbf{u}(\mathbf{x})$ towards the element-wise mean $\overline{\mathbf{u}}$, given as

$$\hat{\mathbf{u}}(\mathbf{x}) = (1 - \alpha)\mathbf{u}(\mathbf{x}) + \alpha \overline{\mathbf{u}} = \mathbf{u}(\mathbf{x}) + \alpha \left(\overline{\mathbf{u}} - \mathbf{u}(\mathbf{x}) \right), \tag{7}$$

where $\hat{\mathbf{u}}(\mathbf{x})$ is the limited solution and $\alpha \in [0, 1]$ is a free parameter dictating the degree of limiting. In the discrete form, the limiting procedure ensures that the limited solution is bounds-preserving at some set of discrete nodal locations $\mathbf{x}_i \, \forall \, i \in S$, i.e.,

$$g(\hat{\mathbf{u}}(\mathbf{x}_i)) \geq 0 \, \forall \, i \in S, \tag{8}$$

where S may be the set of quadrature nodes and/or solution nodes for the scheme and g refers to the constraint functionals in Eq. (2). The limiter relies on the ability of DG schemes to preserve convex invariants on the element-wise mean as discussed in Section 2 (i.e., $g(\overline{\mathbf{u}}) \geq 0$), such that there always exists a value of α for which the constraints are satisfied by the limited solution. This limiting approach encompasses a variety of numerical stabilization methods, including slope-limited, maximum principle/positivity-preserving, and entropy-bounded schemes [17, 19–21, 25]. Furthermore, it is highly advantageous for DG schemes as it is locally conservative, does not require a secondary low-order numerical scheme, is trivial to implement on general unstructured meshes, and can retain the high-order accuracy of the underlying DG scheme.

The goal of the proposed approach is to efficiently find some value of α such that the limited solution is *continuously* bounds-preserving, i.e.,

$$g(\hat{\mathbf{u}}(\mathbf{x})) \geq 0 \, \forall \, \mathbf{x}. \tag{9}$$

Without loss of generality, we consider only one constraint functional $g(\mathbf{u})$ for some arbitrary element Ω_k in the mesh, but the analysis readily extends to sets of functionals and all elements in the mesh. For brevity, the notation $\forall \, \mathbf{x}$ is used to refer to all $\mathbf{x} \in \Omega_k$. A schematic of the goal of the proposed limiting approach is presented in Fig. 1, showing the high-order solution contracting towards the element-wise mean to where the limited solution resides completely within

the admissible set $\mathcal{G}$. In the ideal case, the limiting is performed such as to minimally modify the high-order solution while continuously enforcing constraints, which effectively corresponds to finding the limiting factor for which the limited solution is tangent to the zero contour of the constraint functional.
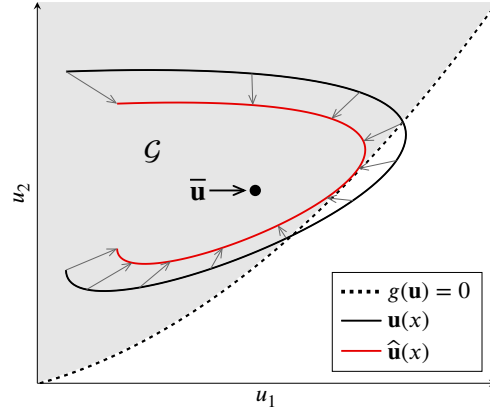

Figure 1: Schematic of the limiting procedure for a vector-valued solution $\mathbf{u}(x) = [u_1(x), u_2(x)]^T$ showcasing the minimal limiting necessary for the limited solution $\hat{\mathbf{u}}(x)$ to be continuously bounds-preserving.

Since the goal is to now enforce constraints continuously via the limiting approach, it is typically necessary to find the continuous extrema of the solution (or some functional of the solution) to see to what extent the solution violates the constraints. Outside of some trivial cases, this requires the use of optimization techniques to find the continuous extrema in each element. However, this optimization problem must be carefully formulated to avoid incurring a substantial computational cost. A natural choice may be to find the minima of the constraint functional and apply limiting based on that nodal location. If we denote g^* as the spatial minimum of the constraint functional, computed as

$$g^* = \min_{\mathbf{x}} g(\mathbf{u}(\mathbf{x})), \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} g(\mathbf{u}(\mathbf{x})), \tag{10}$$

then it can easily be shown from the work of Zhang and Shu [17] that setting α as

$$\alpha = \max\left[0, \frac{g^*}{g^* - g(\overline{\mathbf{u}})}\right] \tag{11}$$

ensures that $g(\hat{\mathbf{u}}(\mathbf{x}^*)) \geq 0$ under the standard assumptions. However, we claim here that limiting based on the minimum of the constraint functional *does not ensure that the limited solution will be continuously bounds-preserving*, i.e., $g(\hat{\mathbf{u}}(\mathbf{x}^*)) \geq 0 \not\Rightarrow g(\hat{\mathbf{u}}(\mathbf{x})) \geq 0 \ \forall \ \mathbf{x}$.

The reason for this difficulty is that the minimum of the constraint functional is not necessarily the point which requires the most limiting. It can be seen that limiting per Eq. (11) is dependent on not only the minimum of the constraint functional but also the value of the constraint functional of the element-wise mean. In fact, α here behaves as the contraction factor necessary to "squeeze" the high-order solution into the admissible set, as shown in Fig. 1. For any given spatial location (i.e., a point on the black phase-space curve in Fig. 1), the limiting operation contracts the solution from this point along the line segment between this point and the element-wise mean. In this view, we can define an inner distance and an outer distance, referring to the length of the line segment residing inside and outside of the set, respectively. As the contraction is a convex combination of both the high-order solution point and the element-wise mean, the necessary level of contraction is not dependent on the outer distance but the *ratio* of the outer and inner distances (i.e., the necessary limiting factor depends on not just the distance of the solution point to the boundary of the admissible set but also the distance of the element-wise mean to the boundary of the admissible set along its respective line segment). If two arbitrary points are taken along the phase-space curve of the high-order solution (with their two associated line segments towards the element-wise mean), the point which resides closer to the admissible set may in fact require more limiting if the element-wise mean is very close to the boundary of the admissible set along its respective line segment.

A simple example of this can be given for a vector-valued solution $\mathbf{u}(x) = [u_1(x), u_2(x)]^T$, where $u_1(x)$ and $u_2(x)$ refer to two components of an arbitrary system of equations, on the domain $\Omega = [-1, 1]$, with the solution and constraint functional taken as

$$u_1(x) = \exp(2x) - 2.8, \tag{12a}$$

$$u_2(x) = \cos(\pi x), \tag{12b}$$

$$g(\mathbf{u}) = 1 - u_1^2 - u_2^2. \tag{12c}$$

The solution profiles and the constraint functional are arbitrarily chosen to illustrate the concept. Fig. 2 shows that for this example system, limiting based on the extremum of the constraint functional does in fact guarantee that the limited solution is bounds-preserving at that point, but it does *not* guarantee that the solution is continuously bounds-preserving as $\exists x : \hat{\mathbf{u}}(x) \notin \mathcal{G}$. In this scenario, the limiting procedure would have to be iterated until the constraints are satisfied continuously which would incur a large computational cost as the optimization procedure would have to be repeated for each iteration.
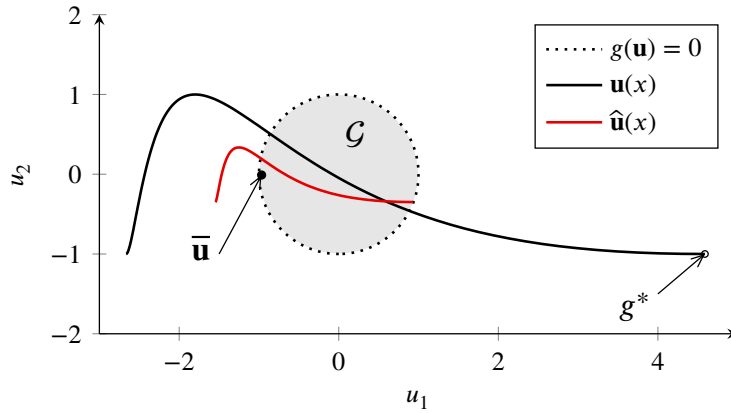

Figure 2: Schematic for an example system and constraints showing that limiting based on the extremum of the constraint functional (g^*) does *not* guarantee that the limited solution is continuously bounds-preserving.

The primary novelty of this work is to introduce a *modified* formulation for the constraint functionals such that the limiting approach can ensure a continuously bounds-preserving solution with only a single spatial minimization problem per element. To this end, we define these modified constraint functionals as

$$h(\mathbf{u}) = \begin{cases} h^+(\mathbf{u}), & \text{if } g(\mathbf{u}) \geq 0, \\ h^-(\mathbf{u}), & \text{else,} \end{cases} \tag{13}$$

where the sub-functionals are defined as

$$h^+(\mathbf{u}) = \frac{g(\mathbf{u})}{g(\overline{\mathbf{u}})} \tag{14}$$

and

$$h^-(\mathbf{u}) = \frac{g(\mathbf{u})}{g(\overline{\mathbf{u}}) - g(\mathbf{u})}. \tag{15}$$

The purpose of these modifications as well as this piecewise formulation will be explained momentarily. We claim that if one instead optimizes over the modified functional, yielding

$$h^* = \min_{\mathbf{x}} h(\mathbf{u}(\mathbf{x})), \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} h(\mathbf{u}(\mathbf{x})), \tag{16}$$

then by setting the limiting factor as

$$\alpha = \max(0, -h^*), \tag{17}$$

the limited solution will be continuously bounds-preserving with only one optimization pass (i.e., only one calculation of the spatial minimization problem). From this formulation, it can be seen that $\alpha \in [0, 1]$ as $g(\overline{\mathbf{u}})$ is strictly non-negative.

The justification for this formulation is as follows. The sub-functional of interest for a bounds-violating solution is $h^-(\mathbf{u})$ as $h(\mathbf{u}) = h^-(\mathbf{u})$ if $g(\mathbf{u}) < 0$. It can be seen that this sub-functional mimics Eq. (11). In fact, optimizing over this functional is effectively optimizing over the limiting factor (or at least a linear approximation thereof) necessary to ensure that the limited solution is bounds-preserving at that given spatial location. This transforms the optimization problem from finding where the constraints are violated the most to finding where the most limiting is required. As will be shown, finding the minimum of $h^-(\mathbf{u})$ and limiting based off of it is sufficient to ensure that the limited solution is continuously bounds-preserving. However, this sub-functional is not necessarily well-behaved due to its rational nature as its value may become undefined — for scalar systems/constraints, this behavior is guaranteed in at least one location per element. To remedy this, $h^-(\mathbf{u})$ is augmented in a piecewise manner with $h^+(\mathbf{u})$ to yield a modified constraint functional $h(\mathbf{u})$ with the following properties:

- Well-defined for all $\mathbf{x}$ to yield numerically stable behavior.

- Of the same sign as $g(\mathbf{u})$ for consistency (i.e., $h(\mathbf{u}) < 0$ for a bounds-violating solution and $h(\mathbf{u}) \geq 0$ for a bounds-preserving solution).

- C^1 continuous with respect to $\mathbf{x}$ to be amenable to optimization algorithms.

These desired properties motivate the definition of $h^+(\mathbf{u})$ in Eq. (14), such that the modified formulation of $h(\mathbf{u})$ in Eq. (13) achieves much more desirable numerical behavior, namely that it and its derivative are always well-defined (if $g(\mathbf{u})$ and its derivative are well-defined).

With this proposed formulation in mind, the proofs of certain key properties of the proposed scheme are now presented. The following proofs rely on two key assumptions, the quasiconcavity of the constraint functional and the bounds-preserving property of the element-wise mean. Furthermore, we assume a strict inequality on the element-wise mean (i.e., $g(\overline{\mathbf{u}}) > 0$) for brevity, with the limiting procedure for the simple edge-case of $g(\overline{\mathbf{u}}) = 0$ shown later.

Theorem 3.1 (Bounds preservation). *The limited solution as given by Eq.* (7) *is continuously bounds-preserving (i.e.,* $g(\widehat{\mathbf{u}}(\mathbf{x})) \geq 0 \ \forall \ \mathbf{x}$) *if the limiting factor is computed as Eq.* (17).

PROOF OF THEOREM 3.1. Per Eq. (17), the limiting factor may either be $\alpha = 0$ or $\alpha = -h^*$. For the former case, $h^* \geq 0$, which gives the equivalency $h^* = h^+(\mathbf{u}(\mathbf{x}^*))$. Given the positivity of the denominator in Eq. (14), it can be seen that

$$g(\mathbf{u}(\mathbf{x})) \geq g(\mathbf{u}(\mathbf{x}^*)) \geq 0 \ \forall \ \mathbf{x}. \tag{18}$$

It then follows that for $\alpha = 0$,

$$g(\widehat{\mathbf{u}}(\mathbf{x})) = g(\mathbf{u}(\mathbf{x})) \geq 0 \ \forall \ \mathbf{x}. \tag{19}$$

For the latter case, $h^* \leq 0$, which gives the equivalency $h^* = h^-(\mathbf{u}(\mathbf{x}^*))$. Therefore,

$$\widehat{\mathbf{u}}(\mathbf{x}) = \mathbf{u}(\mathbf{x}) - \frac{g(\mathbf{u}(\mathbf{x}^*))}{g(\overline{\mathbf{u}}) - g(\mathbf{u}(\mathbf{x}^*))} \left(\overline{\mathbf{u}} - \mathbf{u}(\mathbf{x}) \right) = (1 - \alpha)\mathbf{u}(\mathbf{x}) + \alpha \overline{\mathbf{u}}, \tag{20}$$

where

$$\alpha = -\frac{g(\mathbf{u}(\mathbf{x}^*))}{g(\overline{\mathbf{u}}) - g(\mathbf{u}(\mathbf{x}^*))}. \tag{21}$$

Furthermore, for $h^* \leq 0$, it can be seen that $g(\mathbf{u}(\mathbf{x}^*)) \leq 0$. Due to Jensen's inequality,

$$g(\widehat{\mathbf{u}}(\mathbf{x})) = g \left((1 - \alpha)\mathbf{u}(\mathbf{x}) + \alpha \overline{\mathbf{u}} \right) \geq (1 - \alpha)g(\mathbf{u}(\mathbf{x})) + \alpha g(\overline{\mathbf{u}}) = g(\mathbf{u}(\mathbf{x})) + \alpha \left(g(\overline{\mathbf{u}}) - g(\mathbf{u}(\mathbf{x})) \right). \tag{22}$$

As such, it is sufficient to show that

$$g(\mathbf{u}(\mathbf{x})) + \alpha \left(g(\overline{\mathbf{u}}) - g(\mathbf{u}(\mathbf{x})) \right) = g(\mathbf{u}(\mathbf{x})) - \frac{g(\mathbf{u}(\mathbf{x}^*))}{g(\overline{\mathbf{u}}) - g(\mathbf{u}(\mathbf{x}^*))} \left(g(\overline{\mathbf{u}}) - g(\mathbf{u}(\mathbf{x})) \right) \geq 0 \ \forall \ \mathbf{x}. \tag{23}$$

Note that only the case of $g(\mathbf{u}(\mathbf{x})) < 0$ has to be proven for as $g(\widehat{\mathbf{u}}(\mathbf{x})) \geq 0$ for any $\alpha \in [0,1]$ if $g(\mathbf{u}(\mathbf{x})) \geq 0$ (i.e., any convex combination of bounds-preserving states is bounds-preserving due to the convexity of $\mathcal{G}$). In this case, the quantity $g(\overline{\mathbf{u}}) - g(\mathbf{u}(\mathbf{x}))$ is positive, such that Eq. (23) can be rearranged to

$$\frac{g(\mathbf{u}(\mathbf{x}))}{g(\overline{\mathbf{u}}) - g(\mathbf{u}(\mathbf{x}))} - \frac{g(\mathbf{u}(\mathbf{x}^*))}{g(\overline{\mathbf{u}}) - g(\mathbf{u}(\mathbf{x}^*))} \geq 0 \; \forall \, \mathbf{x}. \tag{24}$$

This is simply the condition

$$h(\mathbf{u}(\mathbf{x})) - h(\mathbf{u}(\mathbf{x}^*)) \geq 0 \; \forall \, \mathbf{x}, \tag{25}$$

which is satisfied as, by definition, $h(\mathbf{u}(\mathbf{x})) \geq h(\mathbf{u}(\mathbf{x}^*))$ for all $\mathbf{x}$. Consequently,

$$g(\widehat{\mathbf{u}}(\mathbf{x})) \geq 0 \; \forall \, \mathbf{x}. \tag{26}$$

Corollary 3.1.1. *From Theorem 3.1, the limiting procedure does not modify the high-order solution $\mathbf{u}(\mathbf{x})$ if the high-order solution is continuously bounds-preserving.*

Theorem 3.2 (Continuity). *The modified constraint functional in Eq. (13) is C^1 continuous with respect to $\mathbf{x}$ if $g(\mathbf{u}(\mathbf{x}))$ is C^1 continuous with respect to $\mathbf{x}$.*

PROOF OF THEOREM 3.2. To show C^1 continuity for the modified constraint functional in Eq. (13), it is sufficient to show that $h^+(\mathbf{u}(\mathbf{x})) \in C^1$ for $g(\mathbf{u}(\mathbf{x})) \geq 0$, $h^-(\mathbf{u}(\mathbf{x})) \in C^1$ for $g(\mathbf{u}(\mathbf{x})) \leq 0$, and $h^{+'}(\mathbf{u}(\mathbf{x})) = h^{-'}(\mathbf{u}(\mathbf{x}))$ for $g(\mathbf{u}(\mathbf{x})) = 0$. Since $g(\overline{\mathbf{u}})$ is constant across an element, $g'(\overline{\mathbf{u}}) = 0$. Therefore,

$$h^{+'}(\mathbf{u}(\mathbf{x})) = \frac{g'(\mathbf{u}(\mathbf{x}))}{g(\overline{\mathbf{u}})} \quad \text{and} \quad h^{-'}(\mathbf{u}(\mathbf{x})) = \frac{g(\overline{\mathbf{u}})g'(\mathbf{u}(\mathbf{x}))}{(g(\overline{\mathbf{u}}) - g(\mathbf{u}(\mathbf{x})))^2}. \tag{27}$$

Due to the positivity of $g(\overline{\mathbf{u}})$, it can be seen that if $g(\mathbf{u}(\mathbf{x})) \in C^1$, then $h^+(\mathbf{u}(\mathbf{x}))$ and $h^-(\mathbf{u}(\mathbf{x}))$ are also C^1 continuous across their respective domains. Furthermore, for $g(\mathbf{u}(\mathbf{x})) = 0$,

$$h^{+'}(\mathbf{u}(\mathbf{x})) = h^{-'}(\mathbf{u}(\mathbf{x})). \tag{28}$$

Lemma 3.3. *The limiting factor required for continuous bounds-preservation (computed by Eq. (17)) is bounded from below by the limiting factor required for discrete bounds-preservation (computed by Eq. (11)).*

With these key properties of the proposed scheme shown, some further details are presented concerning edge cases, higher-order limiting, and multiple constraint functionals. First, the edge case of $g(\overline{\mathbf{u}}) \to 0$ can result in some numerical difficulties, particularly as $g(\mathbf{u}) \to 0$. In this scenario, the limiting factor is explicitly set to $\alpha = 1$ for $g(\overline{\mathbf{u}}) < \epsilon$, where $\epsilon = 10^{-12}$ is some small positive number on the order of machine precision, as it can be seen that $\alpha \to 1$ as $g(\overline{\mathbf{u}}) \to 0$.

Second, the denominator in Eq. (15) effectively acts as a linear approximation of the convex limiting factor necessary to ensure that the limited solution is continuously bounds-preserving, approximating *from above* the solution of

$$g\left((1-\alpha)\mathbf{u}(\mathbf{x}) + \alpha\overline{\mathbf{u}}\right) = 0.$$

For constraint functionals that are linear with respect to the solution variables, the resulting limiting factor is exact, i.e., it is the smallest necessary limiting factor. However, for arbitrary nonlinear quasiconcave constraint functionals, the resulting limiting factor is only sufficient (i.e., the above equality becomes a positive inequality), and there may be a smaller limiting factor for which the limited solution is still continuously bounds-preserving. It is possible to formulate the modified constraint functionals such that the resulting limiting factor is exact for higher-order constraint functionals (e.g., quadratic constraints such as pressure/internal energy in gas dynamics) or to incorporate a nonlinear limiting procedure in the spatial optimization problem to better approximate the minimum necessary limiting factor for arbitrary nonlinear constraints. This may be achieved by replacing the subfunctional $h^-(\mathbf{u})$ with the exact (or sufficiently approximate) solution of the intersection problem,

$$h^-(\mathbf{u}) = -\alpha^* : g\left((1-\alpha^*)\mathbf{u} + \alpha^*\overline{\mathbf{u}}\right) = 0,$$

which enforces an equality on the constraint functional of the limited solution (i.e., a necessary condition) instead of an inequality (i.e., a sufficient condition). However, this is left as a topic of future work. We remark here that this effect is only present for solutions where the nonlinear constraints are violated – per Corollary 3.1.1, for a solution that is already continuously bounds-preserving, no limiting whatsoever is applied.

Finally, the use of multiple constraint functionals provides a choice in how to approach the limiting procedure. The first method is to individually compute α for each of the N_c constraints and take the maximum value. The second method is to sequentially perform the limiting procedure, where for each constraint, the limited solution as computed from the previous constraint is used as the "unlimited" solution for the next constraint. We posit here that the first method will likely yield more performant behavior (i.e., less numerical dissipation) for general nonlinear constraints whereas the two methods would be identical for linear constraints, the reasoning being closely related to the previous discussion on the sufficient versus necessary limiting factor. For nonlinear constraints, computing α by the proposed approach may yield an overprediction in the necessary limiting factor. As such, computing α sequentially may compound this overprediction, causing a higher degree of limiting than computing α individually. However, in some scenarios it may be necessary to compute the limiting sequentially, such as when the convexity of $\mathcal{G}$ or the well-definedness of some constraint functionals rely on other constraints being satisfied. An example of this can be seen in the gas dynamics equations, where the pressure/internal energy functional is quasiconcave only if the density is positive and the entropy functional is typically only well-defined if the density and pressure are positive. In this case, it is necessary to perform limiting sequentially, first on density, then pressure, then entropy.

3.2. Optimization procedure

The proposed limiting approach relies on finding the continuous minimum of the modified constraint functional within the element. For general applications, this minimum does not have a closed-form expression. As such, it must be found numerically using standard optimization techniques. Note that irrespective of the properties of the constraint functional itself, this is generally a global (but bounded) non-convex optimization problem as the solution itself can be and often is spatially non-convex. However, certain properties of the limiting procedure as well as judicious choices for the numerical formulation reduce this complexity.

Bounded optimization. While the optimization problem itself is global, it is bounded to a fixed domain (i.e., the element). As there is no inherent requirement to perform the search for the minimum of the constraint functional in physical space, the optimization can instead be performed in reference space. This greatly simplifies the implementation of bounded optimization schemes as the elements reduce to much simpler (convex) representations in reference space, particularly so for curved elements. The enforcement of search bounds then becomes easier to implement as it is then trivial to evaluate if the search coordinate is within the element bounds (e.g., the reference quadrilateral is simply $[-1, 1]^2$), and bounded optimization methods such as Lagrange multiplier-type approaches can be readily used as the element boundaries in reference space reduce to planar faces with fixed normals. Furthermore, this allows for the user to implement the optimization procedure only for a given reference element without having to consider the intricacies of performing bounded optimization on arbitrarily shaped/curved elements.

Continuous evaluation. Since the optimization problem requires repeatedly evaluating the solution at arbitrary locations, the standard nodal representation of the solution is not well-suited. Instead, the solution may be transformed from its nodal representation to a modal representation, given as

$$\mathbf{u}(\hat{\mathbf{x}}) = \sum_{i=1}^{n_s} \mathbf{u}_i \phi_i(\hat{\mathbf{x}}) = \sum_{i=1}^{n_s} \widetilde{\mathbf{u}}_i \psi_i(\hat{\mathbf{x}}), \tag{29}$$

where $\widetilde{\mathbf{u}}_i$ are a set of modal basis coefficients and $\psi_i(\hat{\mathbf{x}})$ are their respective basis functions. Note that $\hat{\mathbf{x}}$ is used instead of $\mathbf{x}$ to represent that the basis is with respect to the reference space instead of the physical space. This formulation is typically much more amenable to continuous evaluation. In particular, we use the monomial (power) basis to represent the solution, given as

$$\psi_i(\hat{\mathbf{x}}) = \hat{\mathbf{x}}^{\mathbf{q}_i} = \prod_{j=1}^{d} \hat{\mathbf{x}}_j^{\mathbf{q}_{ij}}, \tag{30}$$

which makes it trivial to evaluate the solution anywhere within the element. While the transformation to the monomial basis may be ill-conditioned in some scenarios, this is typically avoided if the solution is not of very high order.

However, in such cases, one may use other bases such as orthogonal polynomials which may possess more robust numerical behavior.

Initialization.   The choice of initial guess for the minimization problem can play a large role in non-convex optimization methods. Even if the optimization landscape has many local minima, convex optimization techniques can still converge to the global minimum if the initial guess on a bounded domain is sampled sufficiently enough such that the initial guess falls within a locally convex region near the global minimum. Fortunately, for DG schemes, the solution is interpolated to the volume and surface quadrature nodes at each temporal step. As such, these quadrature nodes form a suitable set of locations to initially evaluate solution at. In fact, it can be shown through Bézout's theorem that for simpler cases such as linear constraint functionals the number of strict local minima is substantially fewer than the number of quadrature points [39].

With these points in mind, the proposed optimization procedure is presented, first with an introduction and explanation of the various features of the procedure followed by an overview of the overall optimization method. We remark here that since the proof in Theorem 3.1 is contingent on being able to find the global minimum of the modified constraint functional in the element (or at least a lower bound thereof), if one wants to truly *guarantee* that limited solution is continuously bounds-preserving, global non-convex optimization techniques on bounded convex domains should be used. For this, branch-and-bound optimization methods [40] would likely be a good option as the monomial basis allows for simple — albeit somewhat loose — estimates of upper/lower bounds of the solution on convex subdomains which, by the quasiconcavity of $g(\mathbf{u})$, allows for estimates on the lower bound on the constraint functional necessary for branch-and-bound methods. Stricter bounds can be placed by instead using the convex hull property of the Bernstein basis and are often used to provide bounds for nonlinear functionals of the solution on convex subdomains of the element [41]. Alternatively, one may use simulated annealing-type methods [42] as it will be shown later that finding an approximate global minimum instead of a precise local minimum is sufficient.

However, while the computational cost of these bounded global optimization methods is not intractable, particularly so on GPU computing architectures for which the compute-heavy, element-local operations of the methods are well-suited, their algorithmic complexity motivates the use of more straightforward optimization methods. To this end, we instead utilize convex optimization techniques, namely Newton–Raphson and gradient descent methods, and rely on the fact that the solution is sampled sufficiently finely on the element such that it is very likely that one of the sampled points lies within a locally convex region near the global minimum. As this approach does not necessarily ensure that it will find the *global* minimum, it does however lose the mathematical property that the solution is *guaranteed* to be continuously bounds-preserving for general nonlinear constraint functionals, but in the numerical experiments soon to be shown, this approach always yielded a continuously bounds-preserving limited solution.

First, an iterative method must be chosen for optimization. The primary underlying optimization method in this work is the Newton–Raphson scheme, chosen due to its high rate of convergence, with the gradient descent method used as a secondary optimization method under edge cases where the Newton–Raphson scheme is not well-suited. Using the solution interpolated to the surface and volume quadrature nodes, denoted as $\mathbf{x}^q$, the initial guess is taken as

$$\mathbf{x}_0 = \arg\min_{\mathbf{x}_i \in \mathbf{x}^q} h(\mathbf{u}(\mathbf{x}^i)). \tag{31}$$

Each iteration of the Newton–Raphson scheme can be given as

$$\mathbf{x}_n = \mathbf{x}_{n-1} - \mathbf{H}_{n-1}^{-1}\mathbf{J}_{n-1}, \tag{32}$$

where $\mathbf{H}_{n-1}$ and $\mathbf{J}_{n-1}$ represent the Hessian and Jacobian, respectively, of the cost functional $h(\mathbf{u}(\mathbf{x}))$ evaluated at iteration $n-1$. While these quantities may be evaluated analytically using the chain rule, for memory bandwidth-bound computing architectures such as GPUs (which are primarily used in this work), it is generally much more efficient to evaluate them numerically. As such, a centered difference approximation is used, e.g. for a two-dimensional domain, the Jacobian and Hessian can be computed as

$$\mathbf{J}_n = \left[\frac{h(\mathbf{u}(x_n + \delta x, y_n)) - h(\mathbf{u}(x_n - \delta x, y_n))}{2\delta x}, \frac{h(\mathbf{u}(x_n, y_n + \delta x)) - h(\mathbf{u}(x_n, y_n - \delta x))}{2\delta x} \right]^T \tag{33}$$

and

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix}, \tag{34}$$

where

$$\mathbf{H}_{11} = \frac{h(\mathbf{u}(x_n + \delta x, y_n)) - 2h(\mathbf{u}(x_n, y_n)) + h(\mathbf{u}(x_n - \delta x, y_n))}{\delta x^2} \tag{35a}$$

$$\mathbf{H}_{12} = \mathbf{H}_{21} = \frac{h(\mathbf{u}(x_n + \delta x, y_n + \delta x)) - h(\mathbf{u}(x_n - \delta x, y_n + \delta x)) - h(\mathbf{u}(x_n + \delta x, y_n - \delta x)) + h(\mathbf{u}(x_n - \delta x, y_n - \delta x))}{4\delta x^2} \tag{35b}$$

$$\mathbf{H}_{22} = \frac{h(\mathbf{u}(x_n, y_n + \delta x)) - 2h(\mathbf{u}(x_n, y_n)) + h(\mathbf{u}(x_n, y_n - \delta x))}{\delta x^2}. \tag{35c}$$

The step size is taken as $\delta x = 10^{-4}$ to yield a minimum denominator of approximately the square root of the FP64 machine precision error [43]. It will later be shown that the possible convergence issues due to using the approximate Jacobian/Hessian are avoided as it will not necessary to compute the exact minimum.

For the gradient descent method, each iteration is simply computed as

$$\mathbf{x}_n = \mathbf{x}_{n-1} - \beta \frac{\mathbf{J}_{n-1}}{\|\mathbf{J}_{n-1}\|_2}, \tag{36}$$

where β is a free parameter dictating the step size of the iteration. This step size is computed adaptively using back-tracking line search with the initial step length taken as $\beta_0 = 2/(p + 1)$ as this is the approximate average distance between nodal quadrature points in reference space. At each inner iteration of the line search, corresponding to one outer iteration of the gradient descent method, the step size is halved (i.e., $\beta_{i+1} = \beta_i/2$), and the process is repeated until either a maximum of five line search iterations have completed or the Armijo–Goldstein [44] condition is satisfied, i.e.,

$$h(\mathbf{u}(\mathbf{x}_n)) \leq h(\mathbf{u}(\mathbf{x}_{n-1})) - c\beta\|\mathbf{J}_{n-1}\|_2, \tag{37}$$

where the control parameter is taken as $c = 0.5$ as per Armijo [44]. This line search also has the benefit of helping prevent the optimizer from getting stuck in local minima. The gradient descent scheme is used as a fallback for when the Hessian is not invertible or if the functional is not locally convex. Therefore, the Newton–Raphson method is used when both $det(\mathbf{H}) > \epsilon$ and $\mathbf{H}$ is positive semi-definite, where ϵ is the same tolerance used for the edge case of $g(\overline{\mathbf{u}}) \to 0$. Otherwise, the gradient descent method is used.

Next, since the Newton–Raphson and gradient descent methods do not naturally ensure that the spatial location at each iteration will remain within the element, it is necessary to apply some techniques for bounded optimization. The overarching theoretical framework for this approach derives from Lagrange multiplier methods — or, more aptly, the Karush–Kuhn–Tucker conditions — but the simplicity of the bounding domain allows for its straightforward implementation as simple linear projections. We assume here that the reference element is convex and its boundary can be represented by a finite set of straight edges/planar faces. The iterative update $\mathbf{x}_n \to \mathbf{x}_{x+1}$, equivalently represented in terms of a search direction $\Delta\mathbf{x}_n$ as $\mathbf{x}_{x+1} = \mathbf{x}_n + \Delta\mathbf{x}_n$, is modified according to one of three conditions.

1. If $\mathbf{x}_{n+1} \in \Omega_k$, i.e., the iterative update moves strictly within the element, leave $\mathbf{x}_{n+1}$ as is.
2. If $\mathbf{x}_{n+1} \notin \Omega_k$ and $\mathbf{x}_n \notin \partial\Omega_k$, i.e., the iterative update moves from within the element (but not on the boundary) to outside of the element, project the update *distance* onto the element boundary *along the search direction* by solving for $\eta \in [0, 1]$ such that $\mathbf{x}_{n+1} = \mathbf{x}_n + \eta\Delta\mathbf{x}_n \in \partial\Omega_k$.
3. If $\mathbf{x}_{n+1} \notin \Omega_k$ and $\mathbf{x}_n \in \partial\Omega_k$, i.e., the iterative update moves from the element boundary to outside of the element, project the update *direction* onto the element boundary *parallel to the boundary* as $\mathbf{x}_{n+1} = \mathbf{x}_n + (1 - \mathbf{n}) \odot \Delta\mathbf{x}_n \in \partial\Omega_k$, where $\mathbf{n}$ is the normal direction of the particular boundary edge/surface that $\mathbf{x}_n$ is currently on. While this ensures that the iterative update does not intersect the particular boundary edge/surface that $\mathbf{x}_n$ is currently on, it does not ensure that it does not intersect any of the other boundary edges/surfaces. As such, project as per (2) if necessary.

It can be seen here that there are one of two possibilities in this optimization process. The optimizer will either converge towards a local minimum within the element, or the optimizer will converge towards a local minimum along the boundary of the element where the gradient of the constraint functional is parallel to the normal direction of the boundary (i.e., the Lagrange multiplier condition). Due to the fact that the boundary of reference elements can be represented as a small number of straight edges/planar faces, the computations involved in these projections at

most simply reduce to computing the intersection of a line with other lines or planes. For the line search in the gradient descent method, this projection is performed only at the initial step $\Delta \mathbf{x} = -\beta_0 \mathbf{J}$ as every subsequent step is guaranteed to be within the element bounds. Algorithmic implementation details of these projection steps are presented in Appendix A.

Finally, the use of iterative methods requires applying some small convergence tolerance to the optimization process. However, this can cause issues in the limiting method as the bounds-preserving properties shown in Theorem 3.1 are contingent on truly finding the minimum of $h(\mathbf{u})$ as per Eq. (25). If one instead limits based on $h^* + \delta h$, where δh is some positive value smaller than the convergence tolerance, it is possible that the limited solution may violate the bounds on the order of δh. In most cases, this is not acceptable as these bounds must be strictly enforced to ensure the robustness of the numerical scheme, such that one would have to converge to effectively machine precision levels to ensure a continuously bounds-preserving scheme which is computationally intractable. Instead, we propose a final step in the optimization process to alleviate this constraint and allow for reasonable values of the convergence tolerance. Note that by Eq. (25), one does not necessarily need to find the minimum of $h(\mathbf{u})$ per se, simply a lower bound for its minimum. This corresponds to the observation that one may not limit marginally less than what is necessary to ensure a continuously bounds-preserving scheme, but limiting marginally more than necessary is perfectly acceptable. As such, the final step in the optimization approach is to extrapolate a lower bound for $h(\mathbf{u})$ given the information derived from the iterative method. If we denote $h(\mathbf{u}(\mathbf{x}^*))$ as the value of the true minimum of $h(\mathbf{u})$, we instead seek some value $h^{**} \leq h(\mathbf{u}(\mathbf{x}^*))$ that will instead be used for limiting as per Eq. (17).

For this, we utilize some convergence properties of Newton–Raphson/backtracking gradient descent methods, where we seek some correction $\Delta h < 0$, such that

$$h^n - \Delta h \leq h^*. \tag{38}$$

If we assume that the optimization process after n iterations is within some locally convex region of $h(\mathbf{u}(\mathbf{x}))$ in the interior (or convex subregion on the face/edge), then the magnitude of the Jacobian is strictly decreasing across the line segment $[\mathbf{x}^n, \mathbf{x}^*]$, with $\mathbf{J}(\mathbf{x}^*) = \mathbf{0}$, i.e.,

$$\|\mathbf{J}^n\|_2 \geq \max_{x \in [x^n, x^*]} \|\mathbf{J}(\mathbf{x})\|_2. \tag{39}$$

Note here that if the $\mathbf{x}^n$ resides on the face/edge of the element, the Jacobian here refers to the projected Jacobian as per step (3) above. Furthermore, error estimates for the chosen optimization algorithms under the given assumptions yield [45, 46]

$$\left\|\mathbf{x}^* - \mathbf{x}^n\right\|_2 \leq \left\|\mathbf{x}^n - \mathbf{x}^{n-1}\right\|_2 = \left\|\Delta \mathbf{x}_{n-1}\right\|_2. \tag{40}$$

The combination of these two properties yields the relation

$$\left|h^n - h^*\right| \leq \|\mathbf{J}^n\|_2 \left\|\Delta \mathbf{x}_{n-1}\right\|_2. \tag{41}$$

Therefore, we set

$$\Delta h = \|\mathbf{J}^n\|_2 \left\|\Delta \mathbf{x}_{n-1}\right\|_2 \tag{42}$$

and

$$h^{**} = \max\left(-1, h^n - \Delta h\right), \tag{43}$$

which satisfies the condition $h^{**} \leq h(\mathbf{u}(\mathbf{x}^*))$. A maximum is taken to ensure that $\alpha \in [0, 1]$. A simple schematic representation of this tolerance correction is shown in Fig. 3.

With the proposed approach introduced, algorithmic implementation details are presented in Appendix A, including an overview of the limiting approach and the bounded optimization process. For brevity, we present the bounded optimization approach in terms of a single boundary face/normal vector, but the approach readily extends to the multiple faces required for multi-dimensional elements.

3.3. Constraints

For generality, the proposed approach was introduced with respect to an arbitrary quasiconcave functional $g(\mathbf{u})$. Here, we present some examples of common constraints encountered in the simulation of convection dominated flows,
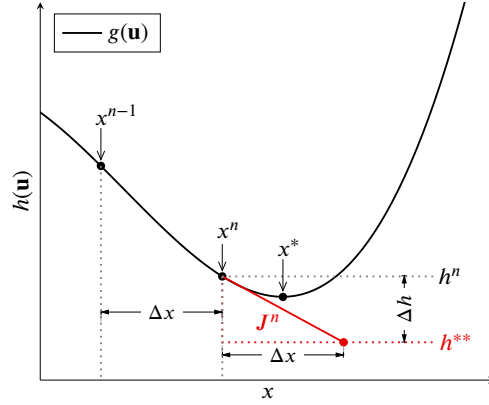

Figure 3: Schematic of the tolerance correction Δh computed at the end of the optimization process.

with more in-depth details of constraints relating to particular systems later presented in Section 5. The first example is a minimum principle, written with respect to some functional of the solution $\mu(\mathbf{u})$ and some minimum bound c as

$$\mu(\mathbf{u}) \geq c. \tag{44}$$

It is common to enforce positivity of some functional of the solution (e.g., density and pressure in gas dynamics), which can be expressed by setting $c = 0$. Alternatively, a more general minimal principle can be enforced by setting $c = \mu_{\min}$, where $\mu_{\min}$ is some minimum bound on the functional (e.g., entropy). This can be enforced by simply setting the constraint functional as

$$g(\mathbf{u}) = \mu(\mathbf{u}) - c. \tag{45}$$

In other cases, the chosen functional must be bounded from above as well as from below, i.e.,

$$\mu(\mathbf{u}) \in [a, b], \tag{46}$$

where a and b ($b > a$) are some minimum/maximum bounds, respectively. These bounds are common in the simulation of scalar systems which obey a strict minimum/maximum principle and can be enforced through a pair of constraint functionals as

$$g_1(\mathbf{u}) = \mu(\mathbf{u}) - a, \quad g_2(\mathbf{u}) = b - \mu(\mathbf{u}). \tag{47}$$

Depending on the constraints and system in question, the order which these constraints are enforced may be of importance (as discussed in Section 3.1).

As the constraints are enforced independently on an element-wise basis, the proposed scheme does not differentiate whether the given bounds (e.g., the values of a, b, and c) are global or local. In some cases, the bounds may depend on the values of the solution in a local domain of influence [20], or they may be fixed throughout the entire domain. In this work, we consider only constraints with global bounds, but the extension to arbitrary local bounds is trivial as long as the bounds are constant across the element.

4. Implementation

The proposed approach was implemented using the DG scheme recovered by the flux reconstruction (FR) method of Huynh [2] within PyFR [47], a high-order unstructured FR solver that can target massively-parallel CPU and GPU computing architectures. Experiments were performed on scalar transport equations and the gas dynamics equations in one and two dimensions on both CPUs and GPUs. At the interfaces, upwinding was performed for scalar equations whereas the Rusanov [33] Riemann solver with the Davis [48] wavespeed estimate was used for the gas dynamics equations. The FR solution/flux points, corresponding to collocated solution/quadrature nodes in DG, were distributed along the Gauss–Lobatto points for tensor-product elements and α-optimized points [1] for simplex elements. We note here that the any set of solution nodes (within reason) can be used without affecting the bounds-preserving properties of the proposed scheme as the approach can enforce constraints across the entire solution polynomial.

Temporal integration was performed using a three-stage, third-order strong stability preserving Runge–Kutta scheme (SSP-RK3 [49]). Limiting was applied after each substage of the SSP-RK3 scheme using three iterations of the Newton–Raphson/gradient descent method. Additionally, limiting was applied at the initial time $t = 0$ after interpolation to the solution nodes to ensure that the solution was initially continuously bounds-preserving. In some experiments, comparisons were performed between the proposed continuously bounds-preserving approach and a standard discrete limiting approach, which coincides to limiting based on the solution value at the discrete solution nodes. This discretely bounds-preserving approach could be recovered by simply taking zero iterations of the optimization problem as the initial guess is taken over the discrete solution nodes.

To evaluate the bounds-preserving properties of the scheme in a continuous sense, the constraints were sampled at a large number of points as a post-processing operation to yield a sufficiently accurate approximation of the continuous minima of the constraint functionals. In this work, we utilize 100 equispaced nodes per spatial dimension. Depending on the experiment, this evaluation was either performed at a fixed time (i.e., a spatial minimum) or continuously in time (i.e., a spatio-temporal minimum). The latter was approximated by evaluating the spatial over-sampling at each discrete temporal integration step. For problems with multiple constraint functionals, the minimum of the respective minima of the constraint functionals was used.

As the proposed approach only enforces given constraints on the solution, it may not guarantee that the solution will be well-behaved around strong discontinuities if the constraints are not restrictive enough. For robustness around discontinuities, it is generally necessary to combine bounds-preserving limiters with additional numerical stabilization methods (e.g., slope limiting [19]) or enforce more restrictive constraints (e.g., minimum entropy constraints [20, 21]). In this work, we utilize a simple subcell-type slope limiter based on a low-order subcell approximation used in the works of Pazner [10], Rueda-Ramírez et al. [12], and Lin et al. [11], the details of which are given in Appendix B. We note here that the choice of numerical stabilization method for discontinuity capturing is orthogonal to the purpose of bounds-preserving limiters (see Zhang and Shu [19]). As long as it does not affect the assumptions that the proposed scheme is built upon (e.g., conservation, bounds-preservation on the element-wise mean, etc.), the combination with slope limiting is largely irrelevant with regards to the benefits and novelties of the proposed scheme, and we do not make any claims about the efficacy of the chosen slope limiter in comparison to other approaches. One may instead use one of a variety of other stabilization approaches of varying degrees of robustness and accuracy.

5. Results

5.1. Linear advection

The proposed approach was first applied to scalar linear transport problems through the linear advection equation, given in conservative form as

$$\partial_t u + \nabla \cdot (\mathbf{c}(\mathbf{x})u) = 0, \tag{48}$$

where $\mathbf{c}(\mathbf{x})$ is some constant or spatially-varying advection field. As with many scalar systems, entropy solutions of the linear advection equation satisfy a maximum principle, i.e.,

$$u(\mathbf{x}, t) \in \left[u_{0,\min}, u_{0,\max}\right] \quad \forall\, \mathbf{x}, t, \tag{49}$$

where

$$u_{0,\min} = \min_{\mathbf{x}} u_0(\mathbf{x}) \quad \text{and} \quad u_{0,\max} = \max_{\mathbf{x}} u_0(\mathbf{x}). \tag{50}$$

This was simply enforced in terms of a pair of constraint functionals,

$$g_1(u) = u - u_{0,\min} \quad \text{and} \quad g_2(u) = u_{0,\max} - u. \tag{51}$$

5.1.1. Advecting waveforms

The bounds-preserving and error convergence properties of the proposed approach were first evaluated through the advecting waveforms problem. The domain is taken as $\Omega = [0, 1]$ with periodic boundary conditions, and the initial conditions are given as

$$u_0(x) = \begin{cases} \exp\left(-300(2x - 0.3)^2\right), & \text{if } |2x - 0.3| \leq 0.25, \\ 1, & \text{if } |2x - 0.9| \leq 0.2, \\ \sqrt{1 - \left(\frac{2x - 1.6}{0.2}\right)^2}, & \text{if } |2x - 1.6| \leq 0.2, \\ 0, & \text{else.} \end{cases} \tag{52}$$

These conditions yield waveforms with varying degrees of continuity, which allows for the evaluation of the approach for both smooth and discontinuous solutions. The advection velocity was simply set as $c = 1$, and global maximum principle bounds were enforced, i.e., $u \in [0, 1]$.

The solutions profiles after one advection cycle ($t = 1$) through the domain as computed by a $\mathbb{P}_2$ and $\mathbb{P}_4$ scheme with varying mesh resolution are shown in Fig. 4. It can be seen that the approach shows clear qualitative convergence to the analytic solution with increasing resolution, both with the $\mathbb{P}_2$ and $\mathbb{P}_4$ approximation. Furthermore, the enforcement of the global maximum principle bounds ensured that the solution remained well-behaved in the vicinity of discontinuities.
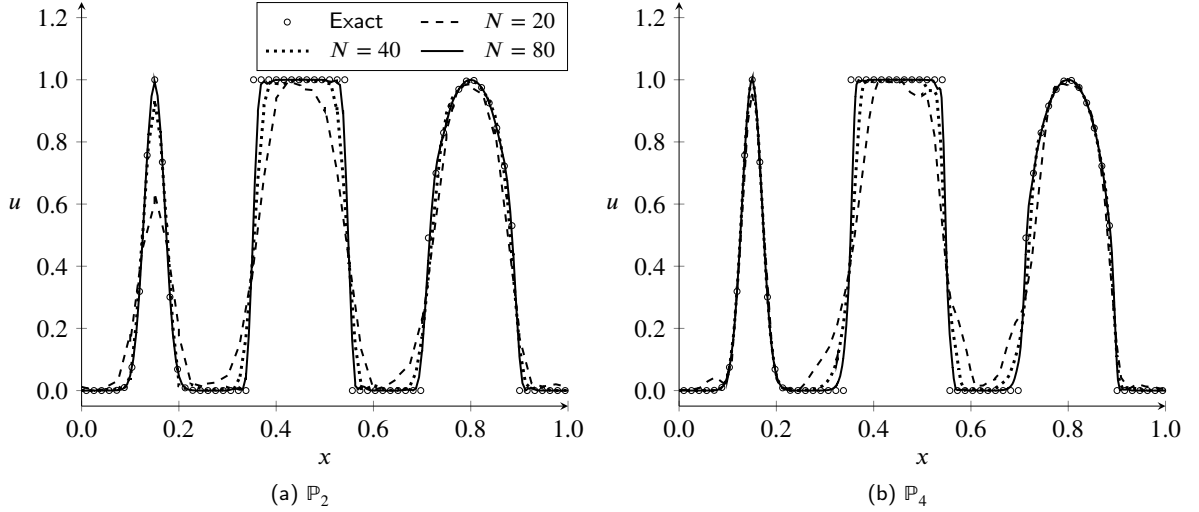

(a) $\mathbb{P}_2$ (b) $\mathbb{P}_4$

Figure 4: Solution profiles for the advecting waveforms problem at $t = 1$ computed using a $\mathbb{P}_2$ (left) and $\mathbb{P}_4$ (right) approximation at varying mesh resolution with *continuously* bounds-preserving limiting and global max principle bounds ($u \in [0, 1]$).

A more quantitative evaluation of the proposed approach was performed by analyzing the error convergence rates. The L^1 norm of the solution error after one advection cycle was computed via quadrature at the solution nodes at varying approximation orders using both the proposed continuous limiting approach and a discrete limiting approach, the results of which are shown in Table 1 and Table 2, respectively. Both approaches showed very similar error behavior, with nearly identical convergence rates at all approximation orders. Furthermore, the magnitude of the error was also similar, with the continuous limiting approach showing marginally less error than the discrete limiting approach at all approximation orders and mesh resolution.

N	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_4$	$\mathbb{P}_5$
20	7.52×10^{-2}	7.63×10^{-2}	7.77×10^{-2}	7.69×10^{-2}
40	3.50×10^{-2}	3.24×10^{-2}	3.39×10^{-2}	3.59×10^{-2}
60	2.16×10^{-2}	2.03×10^{-2}	2.16×10^{-2}	2.30×10^{-2}
80	1.55×10^{-2}	1.48×10^{-2}	1.57×10^{-2}	1.68×10^{-2}
100	1.22×10^{-2}	1.16×10^{-2}	1.23×10^{-2}	1.32×10^{-2}
120	1.01×10^{-2}	9.52×10^{-3}	1.01×10^{-2}	1.09×10^{-2}
RoC	1.13	1.16	1.14	1.09

Table 1: Convergence in the L^1 norm for the advecting waveforms problem at varying approximation order with *continuously* bounds-preserving limiting and global max principle bounds ($u \in [0, 1]$).

The bounds-preserving properties of the proposed continuous limiting approach was then verified by evaluating the spatio-temporal minimum of the constraint functionals over the domain and simulation time, i.e., $\min \left(g_1(u), g_2(u) \right)$

N	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_4$	$\mathbb{P}_5$
20	8.68×10^{-2}	8.35×10^{-2}	8.40×10^{-2}	8.34×10^{-2}
40	4.05×10^{-2}	3.66×10^{-2}	3.69×10^{-2}	3.91×10^{-2}
60	2.52×10^{-2}	2.31×10^{-2}	2.36×10^{-2}	2.50×10^{-2}
80	1.81×10^{-2}	1.69×10^{-2}	1.72×10^{-2}	1.83×10^{-2}
100	1.43×10^{-2}	1.33×10^{-2}	1.35×10^{-2}	1.44×10^{-2}
120	1.18×10^{-2}	1.10×10^{-2}	1.11×10^{-2}	1.19×10^{-2}
RoC	1.12	1.13	1.13	1.09

Table 2: Convergence in the L^1 norm for the advecting waveforms problem at varying approximation order with *discretely* bounds-preserving limiting and global max principle bounds ($u \in [0, 1]$).

sampled at 100 points per element at each temporal integration step. These minima are tabulated in Table 3 for the continuous limiting approach. It can be seen that the proposed approach effectively enforces constraints *continuously* down to machine precision levels, even with coarse mesh resolution. In comparison, the results of the discrete limiting approach, tabulated in Table 4, showed significant violations of the bounds which did not reduce with increasing resolution. In fact, this bounds violation stayed constant with increasing resolution, which can be attributed to the initial interpolation error of a discontinuous field onto a piece-wise continuous solution space. As such, even with highly-resolved meshes, discrete limiting approaches would likely be inappropriate for situations where constraints must be enforced continuously whereas the proposed approach would be effective regardless of the resolution.

N	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_4$	$\mathbb{P}_5$
20	0.0	-1.33×10^{-15}	0.0	-4.44×10^{-16}
40	0.0	-1.33×10^{-15}	0.0	-4.44×10^{-16}
60	-5.55×10^{-17}	-1.33×10^{-15}	0.0	-4.44×10^{-16}
80	0.0	-1.33×10^{-15}	0.0	-4.44×10^{-16}
100	0.0	-1.33×10^{-15}	0.0	-4.44×10^{-16}
120	0.0	-1.33×10^{-15}	0.0	-4.44×10^{-16}

Table 3: Spatio-temporal minimum constraint functional value encountered during simulation for the advecting waveforms problem at varying approximation order and mesh resolution with *continuously* bounds-preserving limiting and global max principle bounds ($u \in [0, 1]$).

N	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_4$	$\mathbb{P}_5$
20	-1.25×10^{-1}	-1.31×10^{-1}	-1.32×10^{-1}	-1.32×10^{-1}
40	-1.25×10^{-1}	-1.31×10^{-1}	-1.32×10^{-1}	-1.32×10^{-1}
60	-1.25×10^{-1}	-1.31×10^{-1}	-1.32×10^{-1}	-1.32×10^{-1}
80	-1.25×10^{-1}	-1.31×10^{-1}	-1.32×10^{-1}	-1.32×10^{-1}
100	-1.25×10^{-1}	-1.31×10^{-1}	-1.32×10^{-1}	-1.32×10^{-1}
120	-1.25×10^{-1}	-1.31×10^{-1}	-1.32×10^{-1}	-1.32×10^{-1}

Table 4: Spatio-temporal minimum constraint functional value encountered during simulation for the advecting waveforms problem at varying approximation order and mesh resolution with *discretely* bounds-preserving limiting and global max principle bounds ($u \in [0, 1]$).

5.1.2. Solid body rotation

The extension to two-dimensional problems and unstructured meshes was then performed through the solid body rotation problem of LeVeque [50]. In this problem, the domain is set as $\Omega = [0, 1]^2$ with periodic boundary conditions,

---

and the initial conditions are given

$$u_0(x) = \begin{cases} 1, & \text{if } (x-0.5)^2 + (y-0.75)^2 \leq 0.15^2 \\ & \text{and } x, y \notin [0.475, 0.525] \times [0.6, 0.85], \\ 0.25\left(1 + \cos\left(\frac{\pi}{0.15}\sqrt{(x-0.25)^2 + (y-0.5)^2}\right)\right), & \text{if } (x-0.25)^2 + (y-0.5)^2 \leq 0.15^2, \\ 1 - \sqrt{(x-0.5)^2 + (y-0.25)^2}/0.15, & \text{if } (x-0.5)^2 + (y-0.25)^2 \leq 0.15^2, \\ 0, & \text{else.} \end{cases} \quad (53)$$

The advection field is set as

$$c(\mathbf{x}) = [-2\pi(y-0.5),\ 2\pi(x-0.5)]^T, \quad (54)$$

which results in counter-clockwise rotation of constant angular velocity about the center of the domain with a unit rotation period. Similarly to the advecting waveforms problem, the initial conditions yield solution profiles of varying degrees of continuity, including a C^∞ cosinusoidal hump, a C^0 sharp cone, and a discontinuous notched cylinder. These profiles can present a challenge to high-resolution numerical schemes, particularly the notched cylinder which is sensitive to excessive diffusion in numerical stabilization methods.

The problem was solved using a $\mathbb{P}_2$ approximation with global max principle bounds ($u \in [0, 1]$) using both structured and unstructured meshes. The solution contours after one rotation period as computed on structured quadrilateral meshes of varying resolution are shown in Fig. 5. It can be seen that for linear problems, the imposition of global max principle bounds effectively stabilized the solution around discontinuities. With increasing resolution, the numerical diffusion around the notched cylinder decreased, such that the initial profile was well-recovered at the highest resolution. The minimum value of the constraint functionals at the final time, sampled at 100^2 points per element, was $g_{min} = -2.5 \cdot 10^{-18}, -1.6 \cdot 10^{-18}$, and $-1.4 \cdot 10^{-19}$ for the $N = 32^2, 64^2$, and 128^2 mesh, respectively. The proposed limiting approach could ensure continuous boundedness to machine precision levels.
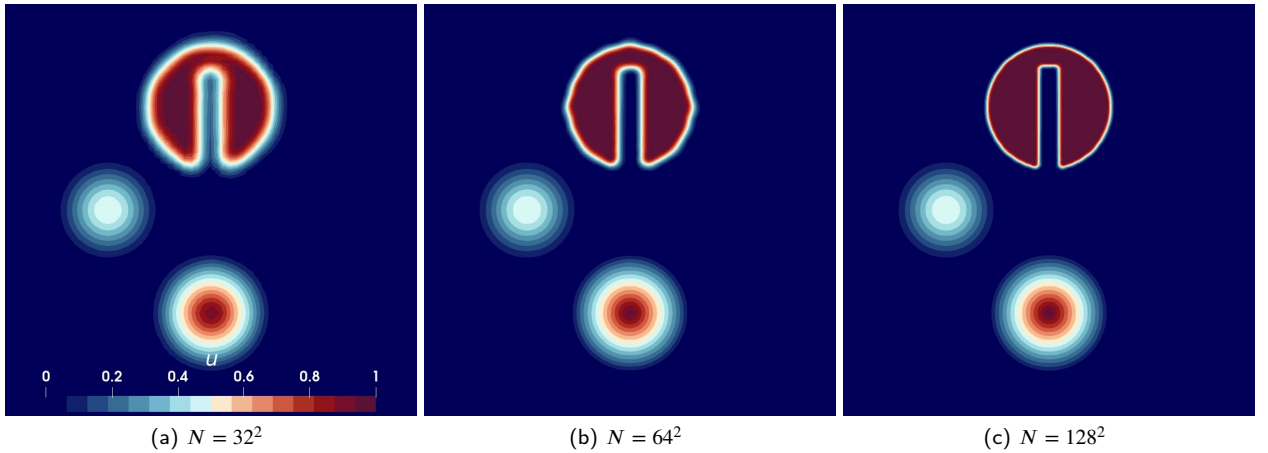

(a) $N = 32^2$ (b) $N = 64^2$ (c) $N = 128^2$

Figure 5: Solution contours for the solid body rotation problem at $t = 1$ computed using a $\mathbb{P}_2$ approximation with *continuously* bounds-preserving limiting and global max principle bounds ($u \in [0, 1]$) on a structured quadrilateral mesh with varying mesh resolution.

To verify the proposed approach for unstructured meshes, the solid body rotation problem was similarly computed on triangular meshes of equivalent resolution. The meshes were generated by uniformly triangulating the previous quadrilateral meshes, such that the number of triangular elements was double the number of quadrilateral elements. The solution contours after one rotation period as computed on the unstructured triangular meshes of varying resolution are shown in Fig. 6. It can be seen that nearly identical results are recovered with the unstructured meshes, indicating the proposed approach is nearly independent of the mesh discretization. Similarly as with the quadrilateral meshes, the minimum value of the constraint functional at the final time, sampled at 100^2 points per element, was $g_{min} = -4 \cdot 10^{-19}$, $-1.8 \cdot 10^{-19}$, and $-2.5 \cdot 10^{-18}$ for the $N = 2 \cdot 32^2, 2 \cdot 64^2$, and $2 \cdot 128^2$ mesh, respectively.
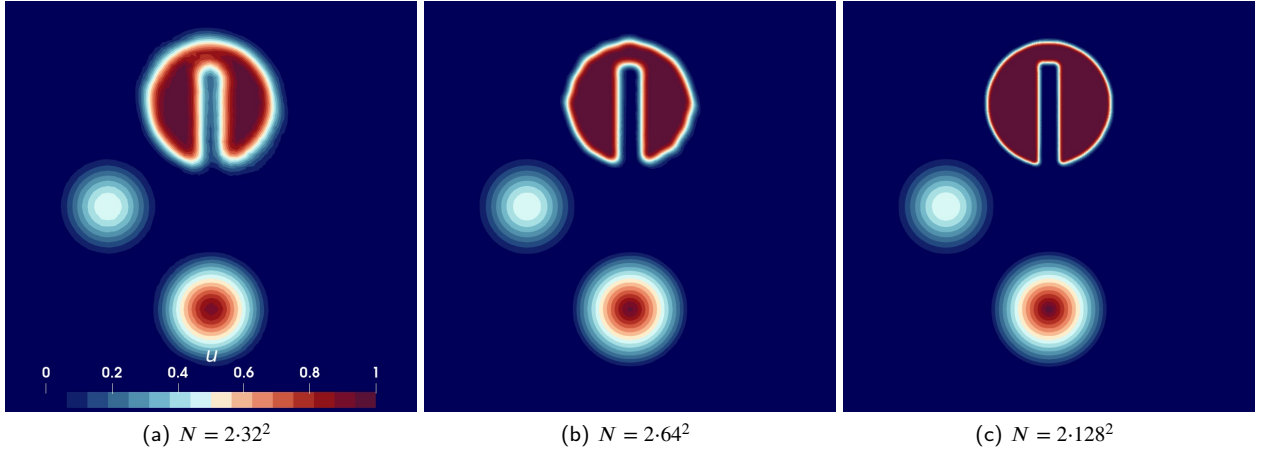
(a) $N = 2 \cdot 32^2$ (b) $N = 2 \cdot 64^2$ (c) $N = 2 \cdot 128^2$

Figure 6: Solution contours for the solid body rotation problem at $t = 1$ computed using a $\mathbb{P}_2$ approximation with *continuously* bounds-preserving limiting and global max principle bounds ($u \in [0, 1]$) on an unstructured triangular mesh with varying mesh resolution.

An additional comparison between the structured and unstructured meshes was compared by observing the cross-sectional profiles of the notched cylinder after one rotation period. The solution profile on the cross-section $y = 0.75$ as computed by both mesh types is shown in Fig. 7. Both mesh types resulted in effectively identical solution profiles, not just in terms of stability around discontinuities but also dissipation of the solution features. These results further showcase the generalizability of the proposed approach to unstructured meshes.
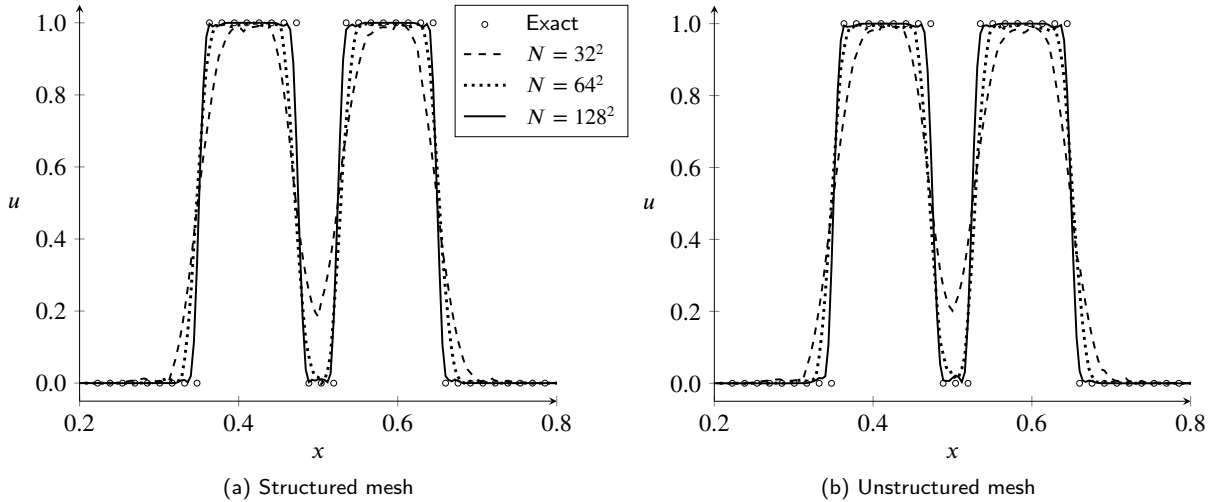

(a) Structured mesh   (b) Unstructured mesh

Figure 7: Solution profile on the cross-section $y = 0.75$ for the solid body rotation problem at $t = 1$ computed using a $\mathbb{P}_2$ approximation with *continuously* bounds-preserving limiting and global max principle bounds ($u \in [0, 1]$). Solution showed for structured quadrilateral meshes (left) and unstructured triangular meshes (right) with varying mesh resolution.

An evaluation of the computational cost of the proposed approach was also performed for the two-dimensional experiments. The relative computational cost of the limiting was calculated by observing the elapsed wall clock time of the the structured $N = 128^2$ experiment for a varying number of optimization iterations (n_{iters}). This cost was normalized by the computational cost of the discrete limiting approach, with a value of one corresponding to no additional computational cost. Comparisons were performed on one NVIDIA V100 GPU. The normalized cost is shown in Fig. 8 for 1, 3, 10, 50, and 100 optimization iterations, with the standard number of iterations used in this

work ($n_{\text{iters}} = 3$) highlighted in red. It can be seen that the cost of the proposed approach is actually quite minor, with 1 iteration requiring only an 11% increase in the computational cost, largely due to an overhead cost associated with transforming the solution basis into its modal form, and 3 iterations requiring only an 18% increase. It was not until the iteration was increased on the order of $n_{\text{iters}} = 30$ that the cost of the optimization approach became more than the underlying solver itself, with $n_{\text{iters}} = 50$ causing a 154% cost increase and $n_{\text{iters}} = 100$ causing a 296% increase. However, this is many more iterations than what is used in this work, which does not appreciably impact the overall computational cost. In fact, this indicates that the further computational effort can be expended without notably degrading the performance of the underlying solver, such as for sampling even more initial points for optimization or performing multiple optimization passes using separate initial guesses.
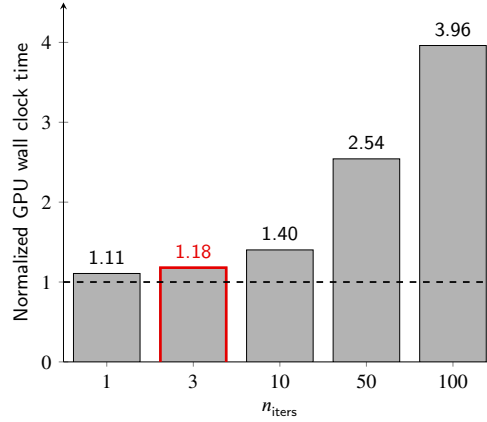

Figure 8: Computational cost of the continuously bounds-preserving limiting approach (normalized by the cost of the discrete limiting approach) with respect to the number of optimization iterations for the solid body rotation problem computed with a $\mathbb{P}_2$ approximation and $N = 128^2$ elements. Number of iterations used in this work $n_{\text{iters}} = 3$ highlighted in red. Unit value (dashed line) denotes no additional cost.

5.2. Burgers' equation

The proposed approach was extended to nonlinear governing equations through Burgers' equation, given in the form of a hyperbolic conservation law as

$$\partial_t u + \partial_x \left(\frac{1}{2} u^2 \right) = 0. \tag{55}$$

Similarly to the advection equation, the entropy solution of Burgers' equation satisfies a maximum principle. Due to the nonlinear nature of the equation, further numerical stabilization (e.g., slope limiting, entropy constraints) may be required to achieve a well-behaved solution as a global maximum principle may not be restrictive enough. However, as the purpose of this experiment is simply to compare the bounds-preserving properties of the proposed approach to other methods, we only enforce global max principle constraints and observe the behavior of the solution near constraint boundaries.

5.2.1. Compression/expansion wave

We consider the problem of a compression and expansion wave around a static point. The problem is solved on the periodic domain $\Omega = [0, 1]$ with the initial conditions

$$u_0(x) = \max \left(1 - 2x, 2x - 1 \right). \tag{56}$$

These initial conditions consist of a compression front decelerating to zero velocity followed by an expansion front, which results in a shock forming around the static point. The problem was solved with a $\mathbb{P}_3$ and $\mathbb{P}_6$ approximation with $N = 24$ elements using global max principle bounds ($u \in [0, 1]$). A comparison of the solution as computed using no limiting, discrete limiting, and the proposed continuous limiting method is shown in Fig. 9 at shock formation time ($t = 0.5$). As the constraints imposed are quite relaxed, the three approaches showed similar qualitative behavior, with the continuous limiting approach showing a marginally better behaved solution around the discontinuity, particularly

at higher order. However, the efficacy of the proposed approach became much more pronounced when observing the continuous solution near the shock at a closer scale.
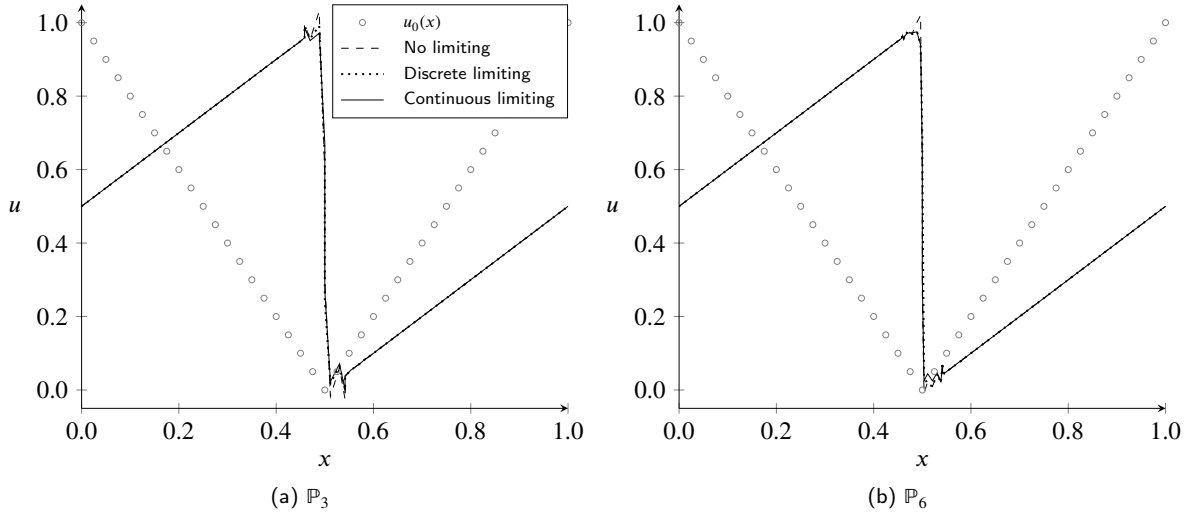

Figure 9: Solution profiles for the Burgers' equation compression/expansion wave problem at $t = 0.5$ computed using a $\mathbb{P}_3$ (left) and $\mathbb{P}_6$ (right) approximation and $N = 24$ elements with various limiting approaches and global max principle bounds ($u \in [0, 1]$).

A comparison of the proposed approach to the discrete limiting method and the unlimited numerical scheme is shown in the enlarged views in Fig. 10 and Fig. 11 for the $\mathbb{P}_3$ and $\mathbb{P}_6$ approximation, respectively. To highlight the effect of enforcing continuous constraints, the polynomial solution is plotted with the discrete solution nodes (i.e., the nodes where discrete limiting would be performed) shown as individual markers. For the $\mathbb{P}_3$ approximation, the unlimited numerical scheme violated the global maximum principle bounds, shown by the dotted lines, both in terms of the discrete solution nodes and the continuous solution polynomial. When discrete limiting was applied, it can be seen that the limiting approach ensured that discrete solution nodes resided strictly within the bounds, but the solution polynomial itself showed overshoots which violated the bounds. As such, if any applications required evaluating the solution at different nodal locations, it is likely that the solution would not satisfy the given constraints. In contrast, with the proposed continuous limiting approach, the *entire* solution polynomial resided within the given bounds, which highlights the ability of the approach to continuously enforce constraints.

A similar analysis was performed with the $\mathbb{P}_6$ approximation, shown in Fig. 11. Due to the higher-order polynomial approximation, the resulting solution polynomial in the vicinity of the discontinuity was significantly more complex, showing more local extrema. As with the $\mathbb{P}_3$ experiment, the $\mathbb{P}_6$ results showed that the unlimited scheme and the discrete limiting method both violated the bounds across the solution polynomial, this time with undershoots instead of overshoots in the discrete limiting method. However, the proposed approach still ensured that the entire solution polynomial resided within the bounds, even with the high polynomial order causing a highly-oscillatory optimization landscape. These results particularly highlight the differences between the proposed continuous limiting method and simple discrete limiting, showcasing the efficacy of the approach for applications where continuous boundedness may be required.

5.3. Euler equations

The robustness of the approach was then evaluated for hyperbolic systems with nonlinear constraints through the Euler equations, given in terms of Eq. (1) as

$$\mathbf{u} = \begin{bmatrix} \rho \\ \mathbf{m} \\ E \end{bmatrix} \quad \text{and} \quad \mathbf{F}(\mathbf{u}) = \begin{bmatrix} \mathbf{m}^T \\ \mathbf{m} \otimes \mathbf{v} + P\mathbf{I} \\ (E + P)\mathbf{v}^T \end{bmatrix}, \tag{57}$$

(a) No limiting

(b) Discrete limiting
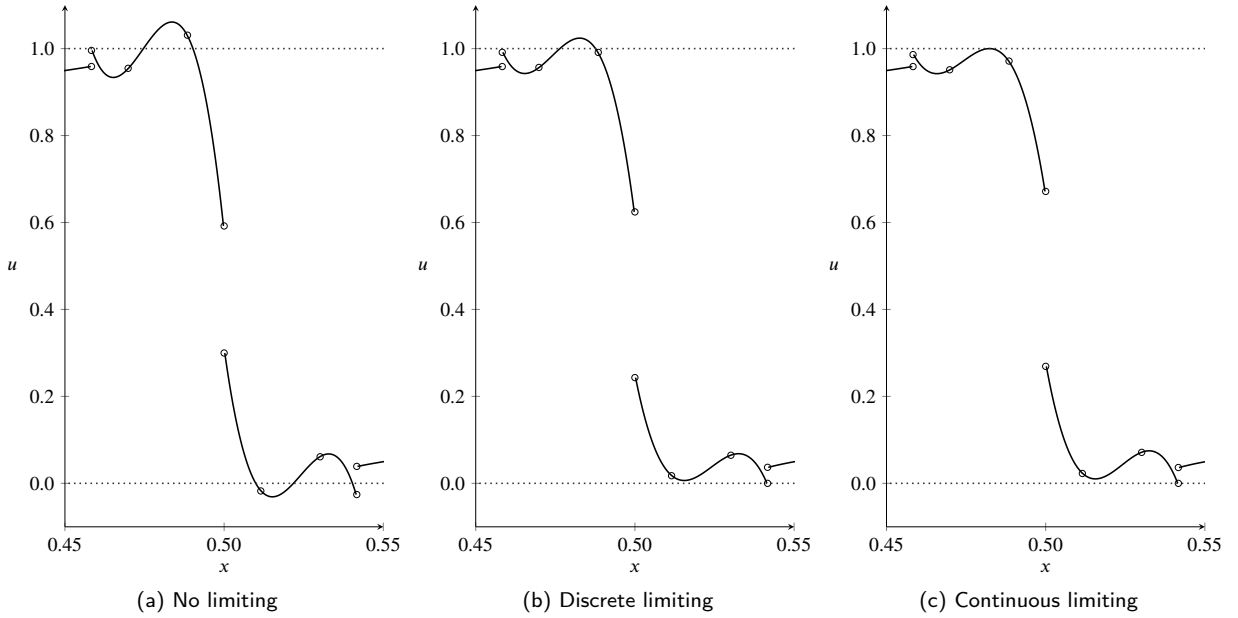
(c) Continuous limiting

Figure 10: Enlarged view of the solution profiles near the shock front for the Burgers' equation compression/expansion wave problem at $t = 0.5$ computed using a $\mathbb{P}_3$ approximation and $N = 24$ elements with various limiting approaches and global max principle bounds. Circle markers denote discrete solution nodes. Bounds (isocontour of $g_1(u) = g_2(u) = 0$) shown by dotted lines.
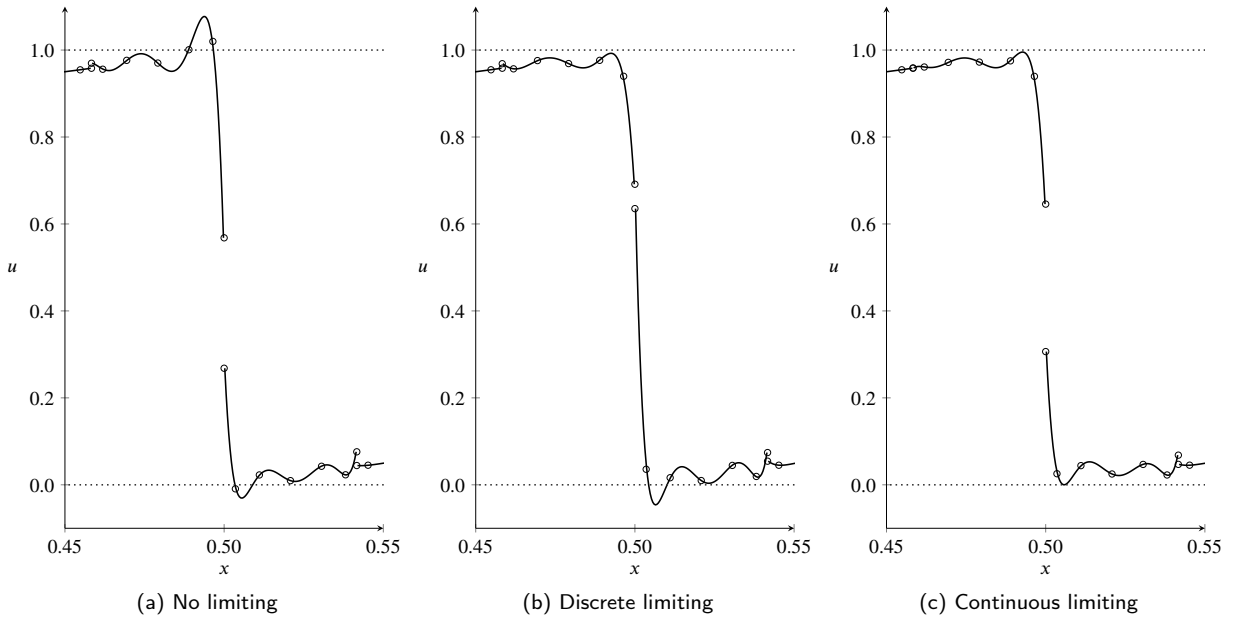

(a) No limiting

(b) Discrete limiting

(c) Continuous limiting

Figure 11: Enlarged view of the solution profiles near the shock front for the Burgers' equation compression/expansion wave problem at $t = 0.5$ computed using a $\mathbb{P}_6$ approximation and $N = 24$ elements with various limiting approaches and global max principle bounds. Circle markers denote discrete solution nodes. Bounds (isocontour of $g_1(u) = g_2(u) = 0$) shown by dotted lines.

where ρ is the density, $\mathbf{m}$ is the momentum, and E is the total energy. The symbol $\mathbf{I}$ denotes the identity matrix in $\mathbb{R}^{d \times d}$, $\mathbf{v} = \mathbf{m}/\rho$ denotes the velocity, and P denotes the pressure, computed as

$$P = (\gamma - 1)\rho e = (\gamma - 1)\left(E - \frac{1}{2}\mathbf{m}\cdot\mathbf{m}/\rho\right), \tag{58}$$

where γ is the specific heat ratio and e is the specific internal energy. Unless otherwise stated, the specific heat ratio is set as $\gamma = 1.4$. For brevity, we denote use $\mathbf{Q} = [\rho, \mathbf{v}, P]^T$ to represent a vector of primitive variables.

Entropy solutions of the Euler equations reside within an invariant set corresponding to solutions with positive density and pressure (i.e., $\mathcal{G} = \{\mathbf{u} \mid \rho \geq 0, P \geq 0\}$) which correspond to physical constraints in gas dynamics. As such, bounds can be imposed in terms of a linear density constraint and a nonlinear (quadratic) pressure constraint as

$$g_1(\mathbf{u}) = \rho - \rho_{\min} \quad \text{and} \quad g_2(\mathbf{u}) = P - P_{\min}, \tag{59}$$

with the limiting applied sequentially. While physically the density and pressure may be zero, it is typically necessary to impose some minimum value to avoid numerical issues with the Riemann solver. Therefore, we impose a minimum density/pressure bound as

$$\rho_{\min} = P_{\min} = 10^{-11}.$$

5.3.1. Near-vacuum convecting density pulse

To verify that imposing positivity constraints via the proposed approach retains high-order accuracy at smooth extrema, we consider the problem of a convecting density pulse in a near-vacuum state. The domain is set as $\Omega = [-0.5, 0.5]$ with periodic boundary conditions, and the initial conditions are given

$$\mathbf{Q}_0(x) = \begin{bmatrix} \exp\left(-200x^2\right) + 2\rho_{\min} \\ 1 \\ 2P_{\min} \end{bmatrix}. \tag{60}$$

The problem consists of a smooth Gaussian density wave, shown in Fig. 12, convecting uniformly through the domain. The ambient density and pressure are set to $2 \cdot 10^{-11}$ which is very close to the minimum density/pressure bound. At these near-vacuum conditions, an unlimited scheme would quickly diverge as the density and pressure attain negative values, such that it is necessary to utilize positivity-preserving schemes. Furthermore, as the exact solution is only marginally within the admissible set, continuous limiting methods which only provide a *sufficient* degree of limiting (e.g., limiting based on Bernstein polynomials) can degrade the high-order accuracy of the underlying numerical scheme as they may introduce numerical dissipation in regions where it is not necessary.
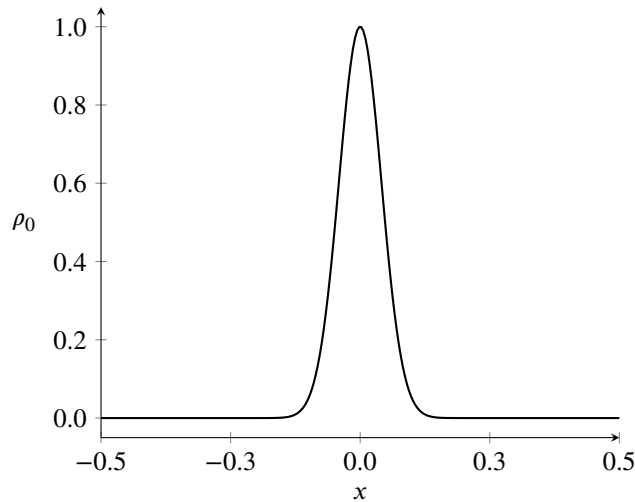

Figure 12: Initial density field for the near-vacuum convecting density pulse.

In Zhang and Shu [19], it was shown that the *discrete* positivity-preserving limiter retains the high-order accuracy of the underlying DG approach. To verify that the proposed continuous limiting approach also exhibits this behavior, the L^∞ norm of the density error was evaluated after one pass-through of the domain ($t = 1$). The convergence of this error is shown in Table 5 for varying approximation orders and mesh resolution. It can be seen that for the given positivity-preserving constraints, the proposed approach also retains the high-order $p + 1$ convergence rates of the underlying scheme, even at smooth extrema, while effectively stabilizing the solution at near-vacuum conditions. This can largely be attributed to the property of the proposed approach in which no limiting is applied to the high-order solution locally if it is already continuously bounds-preserving (Corollary 3.1.1).

N	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_4$	$\mathbb{P}_5$
5	1.36×10^{-1}	7.01×10^{-2}	5.40×10^{-2}	7.41×10^{-2}
10	1.62×10^{-2}	5.47×10^{-3}	2.40×10^{-3}	1.90×10^{-4}
15	9.89×10^{-3}	1.24×10^{-3}	5.12×10^{-5}	1.59×10^{-5}
20	3.51×10^{-3}	1.85×10^{-4}	1.27×10^{-5}	2.03×10^{-6}
25	1.41×10^{-3}	5.62×10^{-5}	4.17×10^{-6}	9.41×10^{-7}
30	7.00×10^{-4}	2.50×10^{-5}	1.72×10^{-6}	5.34×10^{-7}
35	3.55×10^{-4}	1.30×10^{-5}	9.21×10^{-7}	3.30×10^{-7}
40	2.10×10^{-4}	7.27×10^{-6}	5.52×10^{-7}	2.21×10^{-7}
RoC	3.06	4.56	5.76	6.04

Table 5: Convergence of the density error in the L^∞ norm at $t = 1$ for the near-vacuum convecting density pulse at varying approximation order with *continuously* bounds-preserving limiting and positivity-preserving bounds.

5.3.2. Leblanc shock tube

The extension to strong shocks and extreme flow conditions was performed through simulation of the Leblanc shock tube, a notoriously difficult test case. The domain is taken as $\Omega = [0, 9]$ with Dirichlet boundary conditions, and the initial conditions are given as

$$\mathbf{Q}_0(x) = \begin{cases} \mathbf{Q}_L & \text{if } x \leq 3, \\ \mathbf{Q}_R, & \text{else,} \end{cases} \tag{61}$$

where

$$\mathbf{Q}_L = \begin{bmatrix} 1 \\ 0 \\ (\gamma - 1)10^{-1} \end{bmatrix} \quad \text{and} \quad \mathbf{Q}_R = \begin{bmatrix} 10^{-3} \\ 0 \\ (\gamma - 1)10^{-10} \end{bmatrix}. \tag{62}$$

The specific heat ratio is set as $\gamma = 5/3$. Due to extreme pressure ratios on the order of 10^9, the problem is extremely challenging for numerical schemes, and a more in-depth description of the numerical difficulties of this case is presented in Liu et al. [51]. Among them is the tendency for numerical schemes to overpredict the specific internal energy near the contact discontinuity, such that the speed of the shock front is overestimated. This overestimation in the shock speed typically requires an extremely fine resolution to mitigate. Since the purpose of this experiment was to evaluate the bounds-preserving capabilities of the proposed approach, more reasonable resolution levels were used, such that minor overpredictions in the shock speed are to be expected.

The problem was computed using a $\mathbb{P}_3$ scheme with $N = 1200$ elements augmented with a subcell slope limiter. For this case, simply utilizing the subcell slope limiter without positivity-preserving limiting still quickly resulted in the divergence of the numerical scheme. A comparison of the discrete limiting method with the proposed continuous limiting method is shown in the density, velocity, pressure, and specific internal energy profiles at $t = 6$ in Fig. 13. Both approaches quantitatively achieved similar result, with the only visually discernible difference being an underprediction in the pressure field around the shock front by the discrete limiting method.

The distinction in the proposed continuous limiting approach became evident when observing the solution polynomial in the vicinity of the shock. An enlarged view of the pressure profiles near the shock front is shown in Fig. 14 for both the discrete limiting and continuous limiting approach, with the discrete solution nodes shown by markers and the
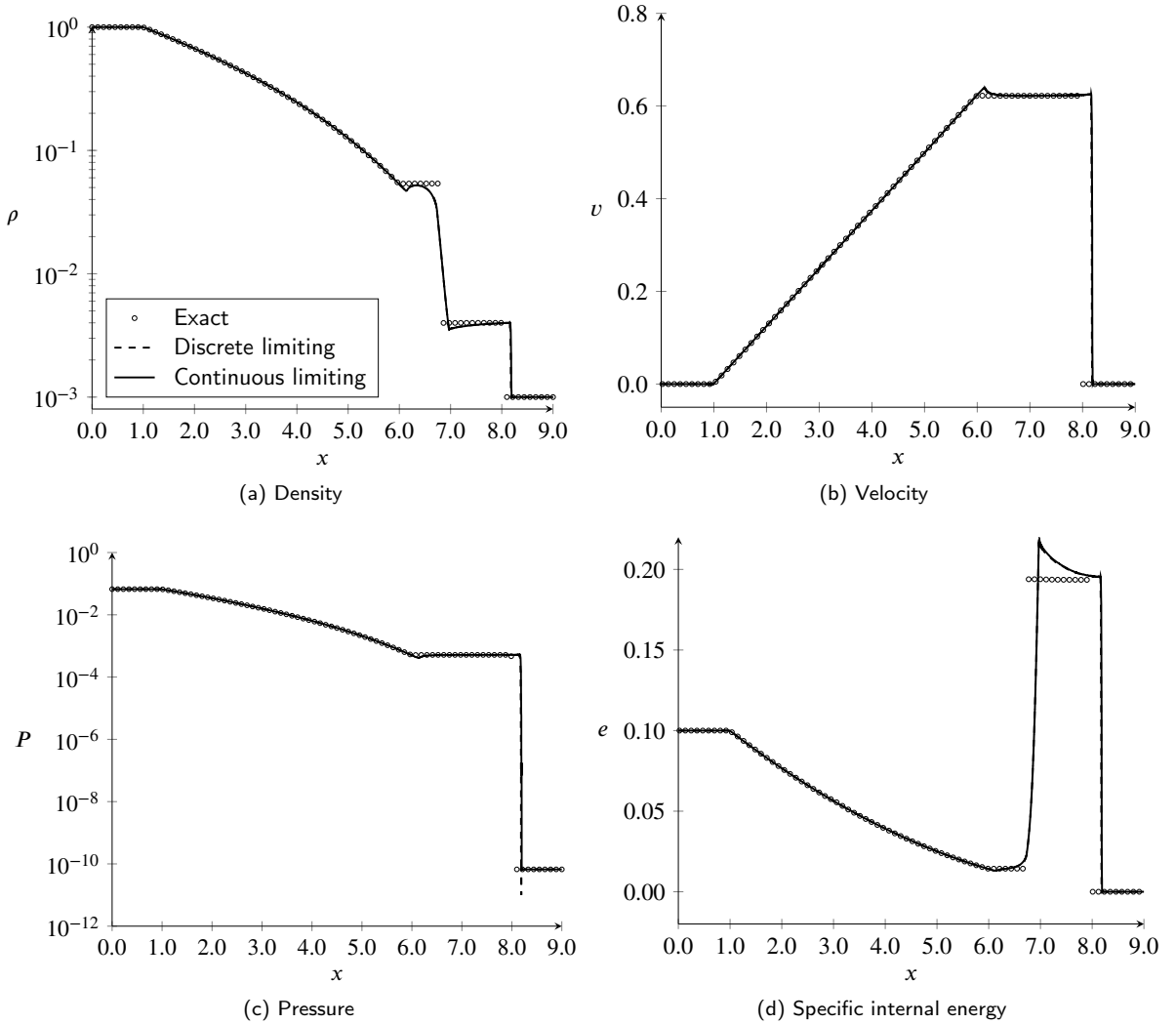
Figure 13: Profiles of density (top left), velocity (top right), pressure (bottom left), and specific internal energy (bottom right) for the Leblanc shock tube problem at $t = 6$ computed with a $\mathbb{P}_3$ approximation and $N = 1200$ elements. Results shown for positivity-preserving bounds with continuously/discretely bounds-preserving limiting augmented with a subcell slope limiter.

full solution polynomial shown by lines. It can be seen that while the discrete limiting method ensured that the pressure at the discrete solution nodes was positive, it did not prevent undershoots in the polynomial approximation from causing negative pressure values. In fact, the minimum continuous pressure for the discrete limiting method at $t = 6$, evaluated at 100 points per element, was $-3.2 \cdot 10^{-6}$. In contrast, the continuous limiting approach showed a positive pressure profile across the entire solution polynomial, even across the strong shock front. The minimum continuous pressure for the proposed continuous limiting approach at $t = 6$ was $6.6 \cdot 10^{-11}$. This value aligns with the minimum initial pressure (and the minimum pressure in the exact solution) which indicates that the additional dissipation in the proposed approach is better suited at mitigating spurious undershoots even at the discrete level.

5.3.3. Sedov blast

For the final numerical experiment and a validation of the proposed approach for two-dimensional gas dynamics problems, the Sedov blast wave [52] was considered. In this problem, an energy point-source in an ambient gas drives an outward-running radial explosion. The problem setup is taken identically to the work of Maire [53], although the
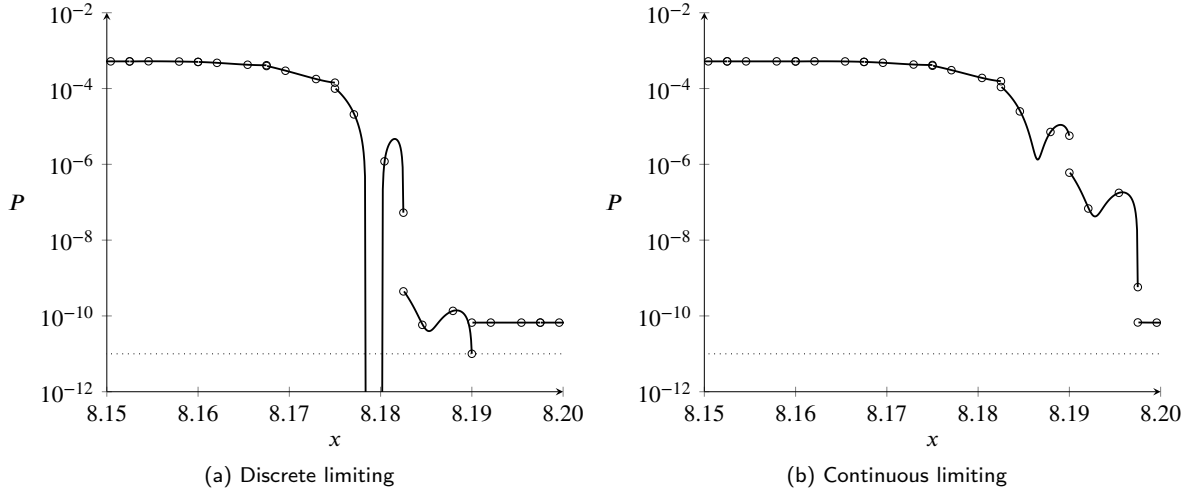
Figure 14: Enlarged view of the pressure profiles near the shock front for the Leblanc shock tube problem at $t = 6$ computed with a $\mathbb{P}_3$ approximation and $N = 1200$ elements. Results shown for positivity-preserving bounds with continuously/discretely bounds-preserving limiting augmented with a subcell slope limiter. Circle markers denote discrete solution nodes. Minimum pressure bound (isocontour of $g_2(\mathbf{u}) = P_{\min}$) shown by dotted line.

entire domain is simulated instead of a single quadrant. The domain is set as $\Omega = [-1.2, 1.2]^2$ with periodic boundary conditions. Due to the finite rate of propagation of the shock wave, the boundary conditions do not impact the solution over the given simulation time. The initial conditions consist of an ambient gas with a constant density $\rho_0 = 1$ and velocity $\mathbf{v} = \mathbf{0}$. Furthermore, the pressure in the domain is set to an ambient value of $P_a = 10^{-6}$ everywhere except in the center-most element (i.e., the element centered at $x = y = 0$), where an overpressure value is set in terms of the internal energy e_0 and cell volume V_0 as

$$P_0 = 4(\gamma - 1)\rho_0 e_0 / V_0. \tag{63}$$

The element volume is simply computed as $V_0 = \Delta x \Delta y = 2.4^2/N$, and the initial specific internal energy is set as $\epsilon_0 = 0.244816$ which yields a shock front with a maximum density of $\rho = 6$ and a radius of 1 at $t = 1$. The factor of 4 in the initial pressure is used to account for the fact that the entire domain is being simulated instead of one quadrant.

The problem was computed with a $\mathbb{P}_2$ approximation and $N = 257^2$ elements using the proposed continuously bounds-preserving limiter augmented with a subcell slope limiter. Like with the Leblanc shock tube case, simply using the subcell slope limiter without enforcing positivity constraints resulted in the divergence of the numerical scheme. The contours of density, radial velocity, and pressure at $t = 1$ are shown in Fig. 15. The typical blast wave structure was observed, with a sharp shock front in the density and pressure field and a smoothly varying velocity field in the interior. Minor mesh imprinting was observed in the velocity field. Additionally, the solution profiles on the positive diagonal cross-section $x = y$ ($x, y \geq 0$) are shown in Fig. 16 in comparison to the analytic solution. Good agreement with the analytic solution was obtained with the proposed approach. To evaluate the continuously bounds-preserving property of the proposed scheme, the minimum density and pressure in the solution at $t = 1$ were calculated across 100^2 points per element. At the final time, the minimum density was $1.5 \cdot 10^{-3}$ and the minimum pressure was $5.9 \cdot 10^{-11}$, well above the minimum density/pressure bounds imposed.
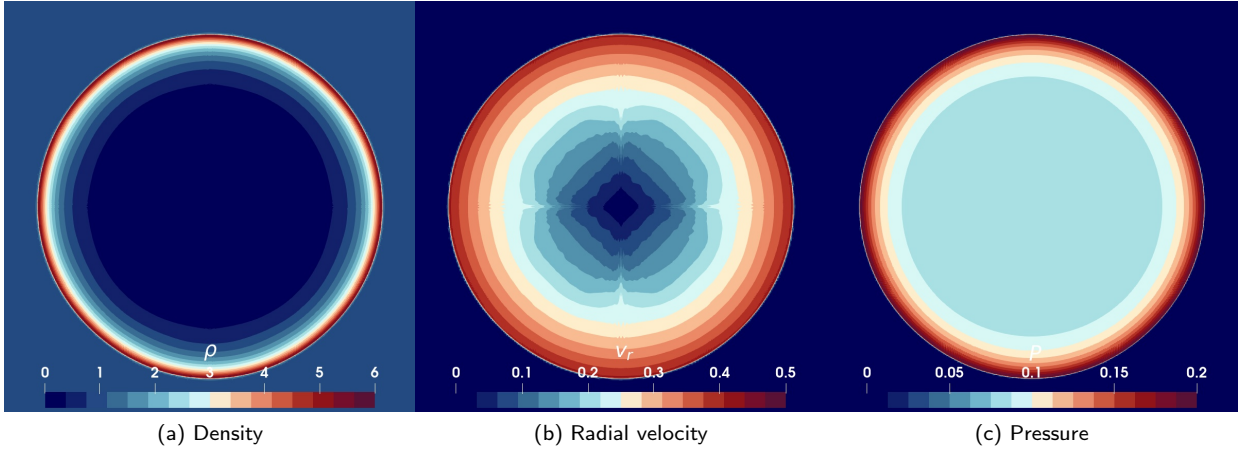
Figure 15: Contours of density (left), radial velocity (middle), and pressure (right) for the two-dimensional Sedov blast problem at $t = 1$ computed with a $\mathbb{P}_2$ approximation, $N = 257^2$ elements, positivity-preserving bounds, and continuously bounds-preserving limiting augmented with a subcell slope limiter.
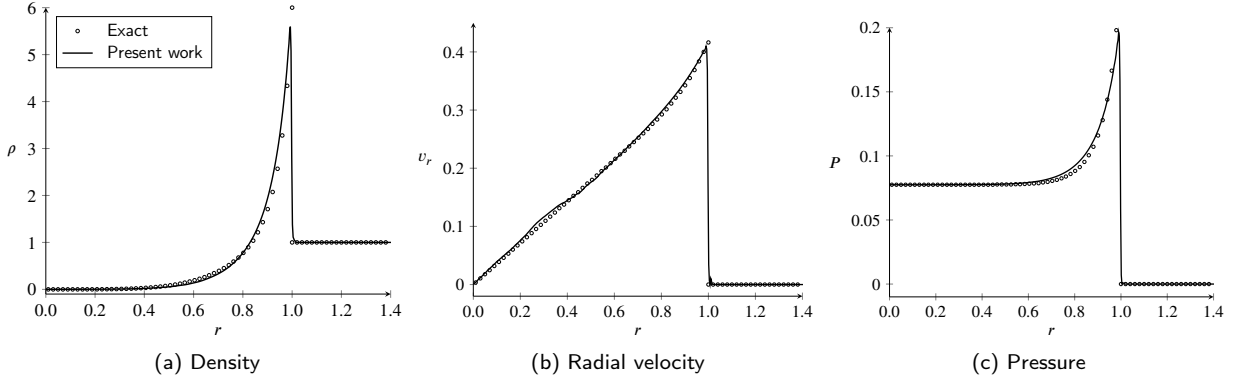

Figure 16: Cross-sectional profiles of density (left), radial velocity (middle), and pressure (right) for the two-dimensional Sedov blast problem at $t = 1$ along the positive $x = y$ diagonal computed with a $\mathbb{P}_2$ approximation, $N = 257^2$ elements, positivity-preserving bounds, and continuously bounds-preserving limiting augmented with a subcell slope limiter.

6. Conclusion

In this work, we presented a novel limiting approach for ensuring that DG approximations of hyperbolic conservation laws can enforce constraints continuously across the entire solution polynomial. The proposed method relied on introducing a novel modification to the constraint functionals which allow for applying bounded optimization techniques to find a limiting factor that ensures the limited solution is continuously bounds-preserving. An efficient optimization approach was then introduced for finding this limiting factor. The proposed continuously bounds-preserving limiting method was applied to a variety of hyperbolic conservation laws on structured and unstructured meshes, ranging from linear scalar transport to compressible gas dynamics with strong shocks, showcasing the ability of the approach in ensuring constraints are satisfied continuously.

The proposed approach has the potential for applications outside of the ones shown in this work as well as further possible developments. Other applications include moving/deforming meshes such as in arbitrary Lagrangian–Eulerian codes and enforcing constraints outside of the element for reconstruction-based methods. Improvements can also be made in the optimization approach to ensure that the given solution is the true global minimum, such as combining the optimization approach with a bounding method [41] on the constraint functional or even the Hessian itself. Further improvements in the accuracy can be made by utilizing higher-order approximations of the limiting factor (or

iterative limiting procedures) for nonlinear constraints as well as modifying the formulation to allow for enforcing bounds which vary across the element.

Acknowledgements

References

[1] Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods*. Springer New York, 2008. doi: 10.1007/978-0-387-72067-8.

[2] H. T. Huynh. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. In *18th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2007. doi: 10.2514/6.2007-4079.

[3] Christopher Cox, Mohammad Reza Najjari, and Michael W. Plesniak. Three-dimensional vortical structures and wall shear stress in a curved artery model. *Physics of Fluids*, 31(12), December 2019. doi: 10.1063/1.5124876.

[4] Johannes Markert. *Discontinuous Galerkin Spectral Element Methods for Astrophysical Flows in Multi-physics Applications*. PhD thesis, Universität zu Köln, 2022.

[5] Jianfeng Zhu, Yong-Tao Zhang, Stuart A. Newman, and Mark Alber. Application of discontinuous Galerkin methods for reaction-diffusion systems in developmental biology. *Journal of Scientific Computing*, 40(1-3):391–418, July 2008. doi: 10.1007/s10915-008-9218-4.

[6] Vinh Phu Nguyen. Discontinuous Galerkin/extrinsic cohesive zone modeling: Implementation caveats and applications in computational fracture mechanics. *Engineering Fracture Mechanics*, 128:37–68, September 2014. doi: 10.1016/j.engfracmech.2014.07.003.

[7] Jay P. Boris and David L. Book. Flux-corrected transport. *Journal of Computational Physics*, 135(2):172–186, August 1997. doi: 10.1006/jcph.1997.5700.

[8] Jean-Luc Guermond and Bojan Popov. Invariant domains and first-order continuous finite element approximation for hyperbolic systems. *SIAM Journal on Numerical Analysis*, 54(4):2466–2489, January 2016. doi: 10.1137/16m1074291.

[9] R. Anderson, V. Dobrev, Tz. Kolev, D. Kuzmin, M. Quezada de Luna, R. Rieben, and V. Tomov. High-order local maximum principle preserving (MPP) discontinuous Galerkin finite element method for the transport equation. *Journal of Computational Physics*, 334:102–124, April 2017. doi: 10.1016/j.jcp.2016.12.031.

[10] Will Pazner. Sparse invariant domain preserving discontinuous Galerkin methods with subcell convex limiting. *Computer Methods in Applied Mechanics and Engineering*, 382:113876, August 2021. doi: 10.1016/j.cma.2021.113876.

[11] Yimin Lin, Jesse Chan, and Ignacio Tomas. A positivity preserving strategy for entropy stable discontinuous Galerkin discretizations of the compressible Euler and Navier-Stokes equations. *Journal of Computational Physics*, 475:111850, February 2023. doi: 10.1016/j.jcp.2022.111850.

[12] Andrés M. Rueda-Ramírez, Will Pazner, and Gregor J. Gassner. Subcell limiting strategies for discontinuous Galerkin spectral element methods. *Computers & Fluids*, 247:105627, October 2022. doi: 10.1016/j.compfluid.2022.105627.

[13] Ahmad Peyvan, Khemraj Shukla, Jesse Chan, and George Karniadakis. High-order methods for hypersonic flows with strong shocks and real chemistry. *Journal of Computational Physics*, 490:112310, October 2023. doi: 10.1016/j.jcp.2023.112310.

[14] Dmitri Kuzmin. Monolithic convex limiting for continuous finite element discretizations of hyperbolic conservation laws. *Computer Methods in Applied Mechanics and Engineering*, 361:112804, April 2020. doi: 10.1016/j.cma.2019.112804.

[15] Michael Dumbser and Raphaël Loubère. A simple robust and accurate a posteriori sub-cell finite volume limiter for the discontinuous Galerkin method on unstructured meshes. *Journal of Computational Physics*, 319:163–199, August 2016. doi: 10.1016/j.jcp.2016.05.002.

[16] Yimin Lin and Jesse Chan. High order entropy stable discontinuous Galerkin spectral element methods through subcell limiting. *Journal of Computational Physics*, 498:112677, February 2024. doi: 10.1016/j.jcp.2023.112677.

[17] Xiangxiong Zhang and Chi-Wang Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *Journal of Computational Physics*, 229(9):3091–3120, May 2010. doi: 10.1016/j.jcp.2009.12.030.

[18] Xiangxiong Zhang, Yinhua Xia, and Chi-Wang Shu. Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes. *Journal of Scientific Computing*, 50(1):29–62, February 2011. doi: 10.1007/s10915-011-9472-8.

[19] Xiangxiong Zhang and Chi-Wang Shu. On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. *Journal of Computational Physics*, 229(23):8917–8934, November 2011. doi: 10.1016/j.jcp.2010.08.016.

[20] T. Dzanic and F.D. Witherden. Positivity-preserving entropy-based adaptive filtering for discontinuous spectral element methods. *Journal of Computational Physics*, 468:111501, November 2022. doi: 10.1016/j.jcp.2022.111501.

[21] Eric J. Ching, Ryan F. Johnson, Sarah Burrows, Jacklyn P. Higgs, and Andrew D. Kercher. Positivity-preserving and entropy-bounded discontinuous Galerkin method for the chemically reacting, compressible Navier-Stokes equations. In *AIAA SCITECH 2023 Forum*. American Institute of Aeronautics and Astronautics, January 2023. doi: 10.2514/6.2023-0664.

[22] T. Dzanic and F.D. Witherden. Positivity-preserving entropy filtering for the ideal magnetohydrodynamics equations. *Computers & Fluids*, 266:106056, November 2023. doi: 10.1016/j.compfluid.2023.106056.

[23] Aditya K. Pandare, Jacob Waltz, Weizhao Li, Hong Luo, and Jozsef Bakosi. On the design of stable, consistent, and conservative high-order methods for multi-material hydrodynamics. *Journal of Computational Physics*, 490:112313, October 2023. doi: 10.1016/j.jcp.2023.112313.

[24] Fan Zhang and Jian Cheng. Analysis on physical-constraint-preserving high-order discontinuous Galerkin method for solving Kapila's five-equation model. *Journal of Computational Physics*, 492:112417, November 2023. doi: 10.1016/j.jcp.2023.112417.

[25] Timothy Barth and Dennis Jespersen. The design and application of upwind schemes on unstructured meshes. In *27th Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, January 1989. doi: 10.2514/6.1989-366.

[26] Bernardo Cockburn, San-Yih Lin, and Chi-Wang Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *Journal of Computational Physics*, 84(1):90–113, September 1989. doi: 10.1016/0021-9991(89)90183-6.

[27] Jianxian Qiu and Chi-Wang Shu. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one-dimensional case. *Journal of Computational Physics*, 193(1):115–135, January 2004. doi: 10.1016/j.jcp.2003.07.026.

[28] Jan Glaubitz. Shock capturing by Bernstein polynomials for scalar conservation laws. *Applied Mathematics and Computation*, 363:124593, December 2019. doi: 10.1016/j.amc.2019.124593.

[29] Peter D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Communications on Pure and Applied Mathematics*, 7(1):159–193, February 1954. doi: 10.1002/cpa.3160070112.

[30] David Hoff. Invariant regions for systems of conservation laws. *Transactions of the American Mathematical Society*, 289(2):591–610, 1985.

[31] Hermano Frid. Maps of convex sets and invariant regions for finite-difference systems of conservation laws. *Archive for Rational Mechanics and Analysis*, 160(3):245–269, November 2001. doi: 10.1007/s002050100166.

[32] Eleuterio F. Toro. The Riemann problem for the Euler equations. In *Riemann Solvers and Numerical Methods for Fluid Dynamics*, chapter 4, pages 115–157. Springer Berlin Heidelberg, 1997. doi: 10.1007/978-3-662-03490-3_4.

[33] V.V Rusanov. The calculation of the interaction of non-stationary shock waves and obstacles. *USSR Computational Mathematics and Mathematical Physics*, 1(2):304–320, January 1962. doi: 10.1016/0041-5553(62)90062-9.

[34] Sergei Konstantinovich Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3):271–306, 1959.

[35] Tong Qin and Chi-Wang Shu. Implicit positivity-preserving high-order discontinuous Galerkin methods for conservation laws. *SIAM Journal on Scientific Computing*, 40(1):A81–A107, January 2018. doi: 10.1137/17m112436x.

[36] Xiangxiong Zhang. On positivity-preserving high order discontinuous Galerkin schemes for compressible Navier–Stokes equations. *Journal of Computational Physics*, 328:301–343, January 2017. doi: 10.1016/j.jcp.2016.10.002.

[37] Xiangxiong Zhang and Chi-Wang Shu. Positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations with source terms. *Journal of Computational Physics*, 230(4):1238–1248, February 2011. doi: 10.1016/j.jcp.2010.10.036.

[38] T. Dzanic, F.D. Witherden, and L. Martinelli. A positivity-preserving and conservative high-order flux reconstruction method for the polyatomic Boltzmann–BGK equation. *Journal of Computational Physics*, 486:112146, August 2023. doi: 10.1016/j.jcp.2023.112146.

[39] Alan Durfee, Nathan Kronefeld, Heidi Munson, Jeff Roy, and Ina Westby. Counting critical points of real polynomials in two variables. *The American Mathematical Monthly*, 100(3):255, March 1993. doi: 10.2307/2324459.

[40] E. L. Lawler and D. E. Wood. Branch-and-bound methods: A survey. *Operations Research*, 14(4):699–719, August 1966. doi: 10.1287/opre.14.4.699.

[41] A. Johnen, J.-F. Remacle, and C. Geuzaine. Geometrical validity of curvilinear finite elements. *Journal of Computational Physics*, 233:359–372, January 2013. doi: 10.1016/j.jcp.2012.08.051.

[42] Dimitris Bertsimas and John Tsitsiklis. Simulated annealing. *Statistical Science*, 8(1), February 1993. doi: 10.1214/ss/1177011077.

[43] J E Dennis, Jr and Robert B Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Society for Industrial and Applied Mathematics, January 1996.

[44] Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, January 1966. doi: 10.2140/pjm.1966.16.1.

[45] J. J. More and D. C. Sorensen. *Newton's method*. February 1982. doi: 10.2172/5326201.

[46] J. C. Dunn. Convergence rates for conditional gradient sequences generated by implicit step length rules. *SIAM Journal on Control and Optimization*, 18(5):473–487, September 1980. doi: 10.1137/0318035.

[47] F.D. Witherden, A.M. Farrington, and P.E. Vincent. PyFR: An open source framework for solving advection–diffusion type problems on streaming architectures using the flux reconstruction approach. *Computer Physics Communications*, 185(11):3028–3040, November 2014. doi: 10.1016/j.cpc.2014.07.011.

[48] S. F. Davis. Simplified second-order Godunov-type methods. *SIAM Journal on Scientific and Statistical Computing*, 9(3):445–473, May 1988. doi: 10.1137/0909030.

[49] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43(1):89–112, January 2001. doi: 10.1137/s003614450036757x.

[50] Randall J. LeVeque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis*, 33(2):627–665, April 1996. doi: 10.1137/0733033.

[51] Wei Liu, Juan Cheng, and Chi-Wang Shu. High order conservative Lagrangian schemes with Lax–Wendroff type time discretization for the compressible Euler equations. *Journal of Computational Physics*, 228(23):8872–8891, December 2009. doi: 10.1016/j.jcp.2009.09.001.

[52] L I Sedov. *Similarity and dimensional methods in mechanics, tenth edition*. CRC Press, 10 edition, September 1993.

[53] Pierre-Henri Maire. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *Journal of Computational Physics*, 228(7):2391–2425, April 2009. doi: 10.1016/j.jcp.2008.12.007.

[54] Michael Dumbser, Olindo Zanotti, Raphaël Loubère, and Steven Diot. A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *Journal of Computational Physics*, 278:47–75, December 2014. doi: 10.1016/j.jcp.2014.08.009.

A. Algorithmic implementation

Algorithm 1 Algorithm for projecting optimizer step to element bounds

---

INPUT: $\mathbf{x}_1$, $\Delta\mathbf{x}$, $\mathbf{n}$ // Initial spatial location, optimizer step, and boundary normal vector
OUTPUT: $\Delta\mathbf{x}'$ // Projected optimizer step

Project($\mathbf{x}_1$, $\Delta\mathbf{x}$, $\mathbf{n}$) :
    // Compute next candidate spatial location.
    $\mathbf{x}_2 = \mathbf{x}_1 + \Delta\mathbf{x}$
    // Apply projection if next spatial location is out of element bounds.
    if $\mathbf{x}_2 \notin \Omega_k$ **then**
        // If initial spatial location is on element boundary, project search direction along boundary.
        // Else project along search direction to boundary.
        if $\mathbf{x}_1 \in \partial\Omega_k$ **then**
            $\Delta\mathbf{x}' = (1 - \mathbf{n}) \odot \Delta\mathbf{x}$
        else
            $\Delta\mathbf{x}' = \eta\Delta\mathbf{x}$ s.t. $\mathbf{x}_1 + \eta\Delta\mathbf{x} \in \partial\Omega_k$
        end if
    else
        $\Delta\mathbf{x}' = \Delta\mathbf{x}$
    end if

Algorithm 2 Optimization and limiting algorithm

INPUT: $\mathbf{u}$, $\mathbf{n}$ // Solution, element boundary normal vector
OUTPUT: $\mathbf{u}$ // Limited solution

Limit($\mathbf{u}$, $\mathbf{n}$) :
  $\mathbf{u} \leftarrow$ Convert nodal basis to monomial basis
  $\mathbf{x}_{\min}, h_{\min} \leftarrow$ Compute discrete minimum of h across solution/quadrature nodes
  // Set minimum/initial values
  $h^* = h_{\min}$
  $\Delta \mathbf{x} = 0$
  // Iterate optimizer.
  for $n \in \{1, ..., n_{\text{iters}}\}$ **do**
    $\mathbf{J}, \mathbf{H} \leftarrow$ Compute $\mathbf{J}$ and $\mathbf{H}$ as per Eq. (33) and Eq. (34)
    // If Hessian is positive semi-definite, compute projected Newton–Raphson step.
    // Else compute projected gradient descent step with backtracking line search.
    if $det(\mathbf{H}) > \epsilon$ and $\mathbf{H}$ is PSD **then**
      $\Delta \mathbf{x} = \mathbf{H}^{-1} \mathbf{J}$
      $\Delta \mathbf{x} = \text{Project}(\mathbf{x}^n_{\min}, \Delta \mathbf{x}, \mathbf{n})$ // Compute projection (if necessary) as per Algorithm 1.
    else
      $\beta = \beta_0$
      $\Delta \mathbf{x} = -\beta \mathbf{J}$
      $\Delta \mathbf{x} = \text{Project}(\mathbf{x}^n_{\min}, \Delta \mathbf{x}, \mathbf{n})$ // Compute projection (if necessary) as per Algorithm 1.
      // Iterate backtracking line search.
      for $i \in \{1, ..., 5\}$ **do**
        $\beta = 0.5\beta$
        $\Delta \mathbf{x} = -\beta \mathbf{J}$
        if $h(\mathbf{u}(\mathbf{x}_{n-1} + \Delta x)) \leq h_{\min} - 0.5\beta \|\mathbf{J}\|_2$ **then**
          break
        end if
      end for
    end if
    // Set next spatial location.
    $\mathbf{x}^n_{\min} = \mathbf{x}^{n-1}_{\min} + \Delta \mathbf{x}$
    $h_{\min} = h(\mathbf{u}(\mathbf{x}^n_{\min}))$
    $h^* = \min(h^*, h_{\min})$
  end for
  // Compute and apply tolerance correction.
  $\Delta h = \|\mathbf{J}\|_2 \|\Delta \mathbf{x}\|_2$
  $h^* = \max(-1, h^* - \Delta h)$
  // Compute limiting factor and limit solution.
  $\alpha = \max(0, -h^*)$
  $\mathbf{u} = (1 - \alpha)\mathbf{u} + \alpha \overline{\mathbf{u}}$

B. Subcell slope limiter

The slope limiter relies on formulating a high-order (unlimited) approximation, a low-order (slope-limited) approximation, and an indicator metric for switching between the two. The high-order candidate approximation, denoted by $\mathbf{u}_H$, is simply the temporal update from the underlying high-order DG scheme. The low-order candidate approximation, denoted by $\mathbf{u}_L$, is computed using a subcell approximation as described by Rueda-Ramírez et al. [12], Section 2.2. The temporal update is then performed by switching between the candidate solutions based on an *a posteriori*

---

indicator metric $\pi(\mathbf{u}_L, \mathbf{u}_H)$ as

$$\mathbf{u}^{n+1} = \begin{cases} \mathbf{u}_L^{n+1}, & \text{if } \pi(\mathbf{u}_L^{n+1}, \mathbf{u}_H^{n+1}) = 1, \\ \mathbf{u}_H^{n+1}, & \text{else.} \end{cases} \tag{64}$$

The indicator used in this work is a discrete maximum principle indicator similar to the multi-dimensional optimal order detection (MOOD) indicator in the work of Dumbser et al. [54] and Dumbser and Loubère [15]. Given some indicator variable of the solution $\mu(\mathbf{u})$, the indicator metric is computed as

$$\pi(\mathbf{u}_L, \mathbf{u}_H) = \begin{cases} 0, & \text{if } \mu_{\min} \leq \mu(\mathbf{u}_{H,i}) \leq \mu_{\max} \ \forall \ i \in S, \\ 1, & \text{else,} \end{cases} \tag{65}$$

where $S = \{1, ..., n_s\}$ represents the discrete solution nodes. The bounds $\mu_{\min}$ and $\mu_{\min}$ are computed through a relaxed discrete maximum principle on the low-order solution, i.e.,

$$\mu_{\min} = (1 - \epsilon_D) \left(\min_{i \in S} \mu(\mathbf{u}_{L,i}) \right) \tag{66}$$

and

$$\mu_{\max} = (1 + \epsilon_D) \left(\max_{i \in S} \mu(\mathbf{u}_{L,i}) \right). \tag{67}$$

ϵ_D represents a relaxation coefficient and is taken as $\epsilon_D = 10^{-2}$ in this work. For the gas dynamics equations, the density field is used for the indicator variable μ.