Efficient optimization-based invariant-domain-preserving limiters in solving gas dynamics equations

Chen Liu[a], Dionysis Milesis[b], Chi-Wang Shu[c], Xiangxiong Zhang[d,]

[a] *Department of Mathematical Sciences, University of Arkansas, Fayetteville, Arkansas 72701, USA.*
[b] *Department of Mathematics and Statistics, Boston University, Boston, MA 02215, USA.*
[c] *Division of Applied Mathematics, Brown University, Providence, Rhode Island 02912, USA.*
[d] *Department of Mathematics, Purdue University, West Lafayette, Indiana 47907, USA.*

Abstract

We introduce effective splitting methods for implementing optimization-based limiters to enforce the invariant domain in gas dynamics in high order accurate numerical schemes. The key ingredients include an easy and efficient explicit formulation of the projection onto the invariant domain set, and also proper applications of the classical Douglas-Rachford splitting and its more recent extension Davis-Yin splitting. Such an optimization-based approach can be applied to many numerical schemes to construct high order accurate, globally conservative, and invariant-domain-preserving schemes for compressible flow equations. As a demonstration, we apply it to high order discontinuous Galerkin schemes and test it on demanding benchmarks to validate the robustness and performance of both ℓ^1-norm minimization limiter and ℓ^2-norm minimization limiter.

Keywords: optimization-based limiters, invariant-domain-preserving, positivity-preserving, discontinuous Galerkin, Davis-Yin splitting, Douglas-Rachford splitting

2000 MSC: 65K05, 65K10, 65M60, 90C25

1. Introduction

1.1. Background and motivation

The compressible Euler and Navier-Stokes (NS) equations are fundamental models in gas dynamics, with broad applications in aeronautics, astronautics, and astrophysics. When solving gas dynamics equations, it is necessary for numerical methods to preserve positivity of density and internal energy, not only to produce physically meaningful solutions, but more importantly to achieve nonlinear stability, especially for demanding applications involving low density and pressure such as very high speed shocks and blast waves [21, 40].

The conserved variables in the gas dynamics equations are density, momentum, and total energy denoted by $\rho, \boldsymbol{m}, E$. Then the internal energy can be written as $\rho e = E - \frac{1}{2}\frac{\|\boldsymbol{m}\|^2}{\rho}$, and for ideal gas the pressure is $p = (\gamma - 1)\rho e$ with a constant $\gamma > 1$. Let $\boldsymbol{m}$ be a column vector, then the set of admissible states of positive density and positivity internal energy (or pressure) can be written as

$$G = \left\{ \boldsymbol{U} = [\rho, \boldsymbol{m}^{\mathrm{T}}, E]^{\mathrm{T}} : \ \rho > 0, \quad \rho e(\boldsymbol{U}) = E - \frac{\|\boldsymbol{m}\|^2}{2\rho} > 0 \right\}. \tag{1}$$

Since the function $\rho e(\boldsymbol{U})$ is concave with respect to $\boldsymbol{U}$, by Jensen's inequality, the set G is convex. In the literature, numerical schemes with solutions staying in the convex set G are positivity-preserving schemes

Email addresses: chenl@uark.edu (Chen Liu), dmilesis@bu.edu (Dionysis Milesis), chi-wang_shu@brown.edu (Chi-Wang Shu), zhan1966@purdue.edu (Xiangxiong Zhang)

[12, 46], and such schemes are also called invariant-domain-preserving schemes [18] since well-posed solutions should stay in this invariant domain set G. For numerical implementation, a numerical admissible set or invariant domain G^ε is often considered with a small $\varepsilon > 0$:

$$G^\varepsilon = \left\{ \boldsymbol{U} = [\rho, \boldsymbol{m}^{\mathrm{T}}, E]^{\mathrm{T}} : \ \rho \geq \varepsilon, \ \ \rho e(\boldsymbol{U}) = E - \frac{\|\boldsymbol{m}\|_2^2}{2\rho} \geq \varepsilon \right\}, \tag{2}$$

which is a convex and closed set.

For enforcing bounds or positivity, there are quite a few approaches developed in the past 15 years for high order accurate schemes, most of which use fully explicit time discretizations [45, 20, 24]. For the compressible NS equations, the fully explicit method [44] typically suffers from restrictive time step size constraints like $\Delta t = O(\mathrm{Re}\,\Delta x^2)$, which is suitable only for large Reynolds numbers $\mathrm{Re} \gg 1$. To use larger time step sizes like $\Delta t = O(\Delta x)$, semi-implicit [18, 31] and the fully implicit method [16] can be considered, but extensions of such positivity-preserving schemes to arbitrarily high order accuracy are in general quite difficult. For a more detailed review, see [31, Section 1.2].

For avoiding small time steps and easy extensions to arbitrarily high order accuracy, one approach is the optimization-based method [17, 39, 38, 29]. A minimization is solved to seek minimal modification of the given numerical solution while enforcing the conservation and positivity or bounds as constraints. In this paper, we focus on how to design such an optimization-based method with the invariant domain G^ε as a constraint, which has not been well studied in previous efforts of designing optimization-based positivity-preserving limiters in the literature.

1.2. Optimization-based invariant-domain-preserving limiters

An optimization-based limiter can be applied to any numerical schemes. In this paper, we focus on discontinuous Galerkin (DG) methods as an example, and the generalization to other popular schemes such as finite difference, finite volume, and continuous finite element method is straightforward. For simplicity, we only consider DG on uniform meshes. For reducing computational cost, we only apply optimization-based limiters to post process the cell averages of DG solutions so that cell averages are ensured to stay in the invariant domain G^ε, after which the simple scaling limiter in [46] can be applied to each DG polynomial in each cell to further ensure quadrature point values to be in the invariant domain G^ε. Another alternative is to apply optimization-based limiters to all the quadrature point values of the DG solution, to which the efficient operator splitting methods in this paper can also be used.

Given a DG solution $\boldsymbol{U}_h$, define the piecewise constant function $\overline{\boldsymbol{U}_h}$ as the cell average of $\boldsymbol{U}_h$ in each cell K, that is, $\overline{\boldsymbol{U}_h}|_K = \frac{1}{|K|}\int_K \boldsymbol{U}_h$. If there exists a bad cell K with $\overline{\boldsymbol{U}_h}|_K \notin G^\varepsilon$, then we seek a piecewise constant $\boldsymbol{X}_h$ which minimizes the distance to $\overline{\boldsymbol{U}_h}$ under a chosen norm with the constraints of preserving global conservation and invariant domain:

$$\min_{\boldsymbol{X}_h} \|\boldsymbol{X}_h - \overline{\boldsymbol{U}_h}\| \quad \text{subject to} \quad \int_\Omega \boldsymbol{X}_h = \int_\Omega \boldsymbol{U}_h \ \text{ and } \ \boldsymbol{X}_h|_{K_i} \in G^\varepsilon \ \text{ for any cell } K_i. \tag{3}$$

One can certainly add more constraints to the minimization above for enforcing more desired physical properties. On the other hand, one major computational challenge is how to efficiently solve such a constrained minimization, which is a limiter to be applied to each time step of high order schemes solving a time dependent problem. As will be shown in this paper, the two constraints in (3) can be efficiently handled by splitting methods.

Let $\overline{\boldsymbol{W}_h}$ denote the minimizer to (3). Then after applying the optimization-based limiter (3), the post-processed DG polynomial can be written as

$$\widehat{\boldsymbol{U}}_h = (\boldsymbol{U}_h - \overline{\boldsymbol{U}_h}) + \overline{\boldsymbol{W}_h}, \tag{4}$$

and it preserves global conservation $\int_\Omega \widehat{\boldsymbol{U}}_h = \int_\Omega \boldsymbol{U}_h$ and $\widehat{\boldsymbol{U}}_h$ has cell averages $\overline{\boldsymbol{W}_h}|_{K_i} \in G^\varepsilon$.

The modification (4) does not destroy the approximation order [29, 27, 28]. In particular, let $\overline{\boldsymbol{U}}$ denote the cell average of the exact solution, and assume the numerical solution has the same integral as the exact

solution, then $\overline{\boldsymbol{U}}$ satisfies both constraints in (3), thus the minimizer $\overline{\boldsymbol{W}_h}$ to (3) satisfies $\|\overline{\boldsymbol{W}_h}-\overline{\boldsymbol{U}_h}\| \le \|\overline{\boldsymbol{U}}-\overline{\boldsymbol{U}_h}\|$, and

$$\|\overline{\boldsymbol{W}_h}-\overline{\boldsymbol{U}}\| \le \|\overline{\boldsymbol{W}_h}-\overline{\boldsymbol{U}_h}\| + \|\overline{\boldsymbol{U}_h}-\overline{\boldsymbol{U}}\| \le 2\|\overline{\boldsymbol{U}_h}-\overline{\boldsymbol{U}}\|. \tag{5}$$

This simple argument shows that the limiter (3) does not destroy the order of accuracy w.r.t. the chosen norm in (3). As will be shown in Theorem 1 in Section 3, we can get a better accuracy result than (5) if L^2-norm is used, i.e., taking $\|\cdot\| = \|\cdot\|_{L^2}$ in (3),

$$\|\overline{\boldsymbol{W}_h}-\overline{\boldsymbol{U}}\|_{L^2} < \|\overline{\boldsymbol{U}_h}-\overline{\boldsymbol{U}}\|_{L^2}. \tag{6}$$

Once the DG polynomial cell averages are in the invariant domain G^ε, the simple scaling limiter in [45, 46] can be applied to further enforce quadrature point values to be in G^ε. In the literature, such a limiter is sometimes called Zhang-Shu limiter, which does not affect accuracy either [44].

1.3. Operator splitting methods in convex optimization

In (3), we still need to specify a norm, and a popular choice is to employ the L^2 norm, i.e. taking $\|\cdot\| = \|\cdot\|_{L^2}$. Meanwhile, the L^1 norm $\|\cdot\| = \|\cdot\|_{L^1}$ has also been often considered in the literature. As will be shown in our numerical tests, L^1 norm limiter may be more desirable for certain problems, such as the astrophysical jet. In the convex optimization literature, the indicator function for a closed convex Λ is defined as:

$$\iota_\Lambda(\boldsymbol{X}) = \begin{cases} 0, & \boldsymbol{X} \in \Lambda, \\ +\infty, & \boldsymbol{X} \notin \Lambda. \end{cases} \tag{7}$$

We remark that $+\infty$ must be used in the definition of the indicator function, instead of a large enough positive number, so that $\iota_\Lambda(\boldsymbol{X})$ can be a proper convex function, see [7, Section 3.2]. With respect to the L^2 and L^1 norms, the optimization problem (3) can be equivalently written as the following two models, respectively,

$$(L^2 \text{ model}): \quad \min_{\boldsymbol{X}_h} \|\boldsymbol{X}_h - \overline{\boldsymbol{U}_h}\|_{L^2}^2 + \iota_{\Lambda_1}(\boldsymbol{X}_h) + \iota_{\Lambda_2}(\boldsymbol{X}_h), \tag{8}$$

$$(L^1 \text{ model}): \quad \min_{\boldsymbol{X}_h} \|\boldsymbol{X}_h - \overline{\boldsymbol{U}_h}\|_{L^1} + \iota_{\Lambda_1}(\boldsymbol{X}_h) + \iota_{\Lambda_2}(\boldsymbol{X}_h), \tag{9}$$

where the two closed convex sets

$$\Lambda_1 = \{\boldsymbol{X}_h : \int_\Omega \boldsymbol{X}_h = \int_\Omega \boldsymbol{U}_h\}, \qquad \Lambda_2 = \{\boldsymbol{X}_h : \boldsymbol{X}_h|_{K_i} \in G^\varepsilon, \ \forall\, i\}, \tag{10}$$

are associated to conservation and invariant-domain-preserving constraints.

For solving (8) and (9), which are a composite of a few convex closed proper functions, the scalability of first order operator splitting methods suits well for large problems. For two operator splitting convex optimization scheme, the most robust splitting is the Douglas-Rachford splitting (DRS), which was first introduced in 1950s for solving heat equations in two dimensions [33, 11]. In 1979, Lions and Mercier extended DRS to composite convex optimization [26]. It has been well known that quite a few popular splitting methods such as the alternating direction method of multipliers (ADMM) and the dual split Bregman method can be exactly equivalent to DRS under proper choices of parameters. See [10, 15] and references therein for the exact equivalent relation. In [35, 36], it was shown that DRS and forward-backward splitting can be integrated, followed by its extension in [6]. Such a method was proven convergent by Davis and Yin [9], and such a three operator splitting is also referred to as Davis-Yin splitting (DYS).

We emphasize that it is critical to design solvers to efficiently solve (3), which is needed in each time step for solving a time-dependent PDE. The DRS in [26] has been shown to be efficient for solving optimization-based limiters for enforcing global conservation and bounds of scalar variables in high order DG schemes solving challenging PDEs including Cahn-Hilliard-Navier-Stokes system [29], compressible Navier-Stokes [27], and highly anisotropic diffusion [28]. In the rest of the paper, we focus on how to construct splitting methods to solve (8) and (9) for invariant-domain-preserving of a vector variable $\boldsymbol{U}$, which is a more difficult problem than the simpler problem of enforcing bounds or positivity for a scalar variable in [29, 27, 28].

Two-operator Douglas-Rachford splitting. Let ∂g and ∂h be the subdifferentials of convex functions g and h. Let I denote an identity operator. The proximal operator with a parameter $\gamma > 0$ for the convex function g is defined as

$$\texttt{prox}_g^\gamma(\boldsymbol{X}) = (\mathrm{I} + \gamma\partial g)^{-1}(\boldsymbol{X}) = \mathrm{argmin}_{\boldsymbol{Z}}\, g(\boldsymbol{Z}) + \frac{1}{2\gamma}\|\boldsymbol{Z} - \boldsymbol{X}\|_2^2, \tag{11}$$

and $\texttt{prox}_h^\gamma$ is defined similarly. To find a minimizer $\boldsymbol{X}^*$ of $g(\boldsymbol{X}) + h(\boldsymbol{X})$, the generalized DRS iteration with a step size $\gamma > 0$ and a relaxation parameter $\lambda \in (0, 2)$ is given by:

$$\text{(DRS)} \quad \begin{cases} \boldsymbol{Y}^{k+1} = \lambda\, \texttt{prox}_g^\gamma(2\boldsymbol{X}^k - \boldsymbol{Y}^k) + \boldsymbol{Y}^k - \lambda \boldsymbol{X}^k, \\ \boldsymbol{X}^{k+1} = \texttt{prox}_h^\gamma(\boldsymbol{Y}^{k+1}), \end{cases} \tag{12}$$

in which $\boldsymbol{Y}$ is only an auxiliary variable, and $\boldsymbol{X}^k$ will converge to the minimizer for any fixed $\gamma > 0$ and $\lambda \in (0, 2)$ if the two functions g and h are convex closed proper, e.g., L^1 function and indicator functions of a closed convex set. If one of the functions is also strongly convex, then (12) also converges for $\lambda = 2$, and the special case $\lambda = 2$ is also called Peaceman-Rachford splitting. For using DRS for solving (8) for enforcing bounds of a scalar variable, in [29] the asymptotic linear convergence of (12) has been analyzed, from which a simple formula for selecting optimal parameters γ and λ is derived, but neither the analysis nor the parameter formula in [29] can be extended to the vector variable case for the invariant domain (2).

Three-operator Davis-Yin splitting. For solving a problem of the form $\min_{\boldsymbol{X}} f(\boldsymbol{X}) + g(\boldsymbol{X}) + h(\boldsymbol{X})$, DYS in [9] is given by:

$$\text{(DYS)} \quad \begin{cases} \boldsymbol{X}^{k+1/2} = \texttt{prox}_g^\gamma(\boldsymbol{Z}^k), \\ \boldsymbol{X}^{k+1} = \texttt{prox}_f^\gamma\big(2\boldsymbol{X}^{k+1/2} - \boldsymbol{Z}^k - \gamma\nabla h(\boldsymbol{X}^{k+1/2})\big), \\ \boldsymbol{Z}^{k+1} = \boldsymbol{Z}^k + \boldsymbol{X}^{k+1} - \boldsymbol{X}^{k+1/2}. \end{cases} \tag{13}$$

For proper closed convex functions f, g, and h, where ∇h is Lipschitz continuous with the Lipschitz constant L, iteration (13) converges for any constant step size $\gamma \in (0, 2/L)$. We remark that three-block ADMM methods can also be used here, but in practice DYS $\gamma = \frac{1}{L}$ performs much better than all other alternatives, see a numerical comparison for solving (8) for enforcing bounds of a scalar variable in [1].

1.4. The main results of this paper

For solving (8), both DRS (12) and DYS (13) can be used, if all the proximal operators are available. For the indicator function ι_Ω for a closed convex set Ω, its proximal operator is simply the Euclidean projection to the set Ω. To implement DRS and DYS, the only nontrivial operator is the projection onto the invariant domain. Even though the definition of the set G^ε in (2) seems simple, its projection formula is no longer straightforward. The first key result of this paper is to derive an efficient explicit projection formula for convex set G^ε in (2), with which both DRS (12) and DYS (13) can be easily implemented for solving (8). We are able to derive the projection formula as a cubic root. For the ease of presentation, we list the projection algorithms in Appendix B.2 and Appendix C.2.

For solving (9), neither DRS (12) nor DYS (13) can be directly used. As will be shown in this paper, (8) can be regarded as a subproblem in DRS solving (9), and this subproblem can be solved efficiently by DYS (13). In other words, we propose to use DRS nested with DYS for solving (9), which is an efficient way for solving L^1-norm optimization-based invariant-domain-preserving limiter.

Our numerical results indicate that the L^1-norm limiter (9) is not better than the L^2-norm limiter (8) in terms of sparsity of the minimizer, unlike ℓ^1-norm minimizations in many other applications. Moreover, the L^2-norm limiter (8) is cheaper to solve, and it improves the accuracy of the DG solution, as shown in Theorem 1 in Section 3. On the other hand, as will be shown in our numerical tests, for certain important time dependent problems, the L^1-norm limiter (9) may be more desirable. For instance, as shown in Figure 5 for the high speed astrophyscial jet problem, the optimization-based limiter is triggered less during the time evolution if using L^1-norm limiter (9), compared to using L^2-norm limiter (8).

1.5. Related work

In the past 15 years, there have been active research on constructing invariant-domain-preserving limiters for high order schemes solving hyperbolic and related systems. One popular approach introduced by Zhang and Shu [45, 46] is to only modify the approximation polynomial by a simple scaling limiter to eliminate overshoots and undershoots at certain quadrature points, with which DG and finite volume type schemes using strong stability preserving (SSP) time discretizations can ensure the cell averages to be invariant-domain-preserving. See also [44, Appendix C] for the accuracy proof of this limiter. Another widely used approach is the flux corrected transport (FCT) type method [43], which is to modify the high order numerical flux by a convex combination with a low order flux [20, 19, 23, 22]. The FCT is flexible and applicable to a broad range of spatial discretizations. Recent developments include convex limiting and subcell convex limiting [22, 32, 37, 25], etc. It is in general difficult to establish rigorous accuracy proof for FCT type methods, though the accuracy for the 1D linear scalar case can be proven [41]. All these approaches rely on existence of a low order invariant-domain-preserving numerical flux, which is however not always available for complicated systems, e.g., higher order PDEs like phase-field equations [29].

As a more flexible alternative to the traditional methods above, optimization-based limiters like (3) can be considered, and it is mostly studied for a scalar variable, for which the admissible set $G = [m, M]$ means bound-preserving, or $G = [0, +\infty]$ means positivity-preserving. The same or similar optimization-based limiters for scalar quantities have been considered under different contexts, e.g., [17, 3, 5, 42, 2, 34]. See also [39, 38] for different optimization-based approaches for enforcing bounds. In [29, 27, 28], the optimization-based limiter was considered for some challenging equations and systems but the limiter was only applied to scalar variables.

To the best of our knowledge, all these existing optimization-based approaches consider the treatment of a scalar variable, rather than directly tackling the invariant domain set G^ε defined in (2). On the other hand, if $\boldsymbol{X}$ is a scalar variable in (3), it has been proven in [5] that the unique minimizer to the L^2 minimization (8) is one of the minimizers to the L^1 minimization (8), and an explicit cheap construction to one of the L^1 minimizers to (9) was given in [5], which will be reviewed in Section 2. In other words, for a scalar variable $\boldsymbol{X}$ in L^1 minimization (9), the best solution is the explicit construction of one particular minimizer in [5], which is however not a minimizer to (8), and one gets the accuracy justification in (5) under L^1-norm. For using L^2 minimization (8) for a scalar variable $\boldsymbol{X}$, in the literature there are many efficient solvers that were used, for which however the computational complexity is usually not quantified. For solving L^2 minimization (8) for a scalar variable $\boldsymbol{X}$, when using the optimal parameters given in [29], DRS has a complexity around $80N$, if the bad cell ratio is small, which is the case for many good schemes solving many practical problems.

Among all the efficient solvers considered and used for (8) for a scalar variable, only the operator splitting methods can be easily extended to enforcing the invariant domain for a vector variable. For example, it was shown in [4] that (8) for scalar variables can be efficiently solved by the solver in [8], which was designed for enforcing bounds rather than an invariant domain like (2).

1.6. Contributions and organization of the paper

In this paper, we provide a simplified explicit formula for projecting a point onto the admissible set (2) for gas dynamics equations such as compressible Euler and NS equations. We further use this projection to construct efficient operator splitting schemes for solving optimization-based post-processing limiters (8) and (9). To the best of our knowledge, this is the first study on constructing optimization-based limiters for directly enforcing the invariant domain (2), and this is also the first time to use in this context a three-operator splitting scheme like DYS, which performs quite well numerically for solving L^2-norm limiter (8). The proposed invariant-domain-preserving limiter also works in enforcing positivity of cell averages in DG methods, even if the time discretization is not SSP type, as will be shown in our numerical tests.

The rest of this paper is organized as follows. In Section 2, we review some existing results then explain how to design splitting methods for solving the L^2 and L^1 optimization-based limiter for scalar variables. In Section 3, we describe the designed splitting methods for optimization-based limiters for the enforcing invariant domain. DYS method is used for solving the L^2-norm invariant-domain-preserving limiter, and DRS nested with DYS can be used for the L^1-norm invariant-domain-preserving limiter. Both limiters

involve the projection to the invariant domain, and formulae are given in the Appendix. Numerical tests are shown in Section 4. Concluding remarks are given in Section 5.

2. Bound-preserving limiters for scalar conservation laws

In this section, we first review the result in [5], then discuss splitting methods for preserving bounds of scalar variable, i.e., the invariant domain is $G = [m, M]$, where m and M are the lower and upper bounds satisfied by the exact or physical solutions.

2.1. The problem setup and existsing results for the scalar case

For a scalar-valued function u in solving a scalar equation, a DG scheme may violate the bound-preserving property with cell averages out of the interval $G = [m, M]$, which may occur when using a high order scheme with a large time step size or a high order accurate implicit scheme. The L^2 model (8) enforces global conservation and bounds without compromising high order accuracy, see [28].

We introduce a vector $\boldsymbol{u} \in \mathbb{R}^N$ to represent cell averages of the scalar-valued DG polynomial solution U_h, that is, the i-th entry of $\boldsymbol{u}$ equals $\overline{U_h}|_{K_i} = \frac{1}{|K_i|}\int_{K_i} U_h$. Define a matrix $\mathbf{A} = [1\ 1\ \cdots\ 1] \in \mathbb{R}^{1\times N}$ and $b = \mathbf{A}\boldsymbol{u}$. Then, the limiter (8) is equivalent to the following problem with a parameter $\alpha > 0$ in matrix-vector form:

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^N} \frac{1}{2\alpha}\|\boldsymbol{x}-\boldsymbol{u}\|_2^2 + \iota_{\Lambda_1}(\boldsymbol{x}) + \iota_{\Lambda_2}(\boldsymbol{x}), \\ &\text{where}\ \ \Lambda_1 = \{\boldsymbol{x}\in\mathbb{R}^N : \mathbf{A}\boldsymbol{x} = b\}, \\ &\text{and}\ \ \Lambda_2 = \{\boldsymbol{x}\in\mathbb{R}^N : x_i \in [m,M],\ \forall i = 1,\cdots,N\}. \end{aligned} \tag{14}$$

Notice that we have a freedom to choose a different $\alpha > 0$, which may affect performance of solvers for (14). Similarly, the L^1 model (9) for a scalar variable with $G = [m, M]$ is equivalent to

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^N} \|\boldsymbol{x}-\boldsymbol{u}\|_1 + \iota_{\Lambda_1}(\boldsymbol{x}) + \iota_{\Lambda_2}(\boldsymbol{x}), \\ &\text{where}\ \ \Lambda_1 = \{\boldsymbol{x}\in\mathbb{R}^N : \mathbf{A}\boldsymbol{x} = b\} \\ &\text{and}\ \ \Lambda_2 = \{\boldsymbol{x}\in\mathbb{R}^N : x_i \in [m,M],\ \forall i = 1,\cdots,N\}. \end{aligned} \tag{15}$$

By [5, Proposition 4.1], the unique minimizer to the strongly convex ℓ^2 problem (14) also minimizes the ℓ^1 problem (15), which however has multiple minimizers. Moreover, it is proven in [5] that one particular ℓ^1 minimizer can be constructed explicitly by `ClipAndAssuredSum` ([5, Algorithm 3.1]):

$$\texttt{clip}(\boldsymbol{u}) = \min\left(\max\left(\boldsymbol{u}, m\right), M\right), \tag{16a}$$

$$\boldsymbol{d} = \begin{cases} \frac{M\mathbf{1}-\texttt{clip}(\boldsymbol{u})}{MN-\mathbf{1}\cdot\texttt{clip}(\boldsymbol{u})}, & \text{if } \mathbf{1}\cdot\texttt{clip}(\boldsymbol{u}) - b < 0, \\ \mathbf{0}, & \text{if } \mathbf{1}\cdot\texttt{clip}(\boldsymbol{u}) - b = 0, \\ \frac{\texttt{clip}(\boldsymbol{u})-m\mathbf{1}}{\mathbf{1}\cdot\texttt{clip}(\boldsymbol{u})-mN}, & \text{if } \mathbf{1}\cdot\texttt{clip}(\boldsymbol{u}) - b > 0, \end{cases} \tag{16b}$$

$$\boldsymbol{x}^* = \texttt{clip}(\boldsymbol{u}) - (\mathbf{1}\cdot\texttt{clip}(\boldsymbol{u}) - b)\boldsymbol{d}. \tag{16c}$$

If one wants an ℓ^1 minimizer, the `ClipAndAssuredSum` is the best solution because of its low cost. On the other hand, the $\boldsymbol{x}^*$ in (16) is not necessarily an ℓ^2 minimizer. Moreover, it seems quite difficult to extend the proof and results in [5] to solving (9) in the context of the invariant domain (2).

We emphasize that the minimizers to (15) are usually not sparse, unlike many other ℓ^1 minimization problems in other applications. In terms of sparsity, the minimizers to ℓ^1 minimization (15) are often only marginally better than the minimizer to ℓ^2 minimization (14).

Remark 1. *The objective function of the ℓ^1 minimization problem* (15) *is convex but not strongly convex, which implies the non-uniqueness of the minimizer. To see this, let us consider the following simple example with four variables:*

$$\min_{x_1,x_2,x_3,x_4} |x_1 - 1| + |x_2 - 1| + |x_3 - 2| + |x_4 - 2.1| \tag{17a}$$

$$\text{subject to } x_1 + x_2 + x_3 + x_4 = 6.1 \text{ and } x_i \in [1, 2] \text{ for } i = 1, \cdots, 4. \tag{17b}$$

The vector $[x_1, x_2, x_3, x_4]^{\mathrm{T}} = [1, 1.1, 2, 2]^{\mathrm{T}}$ is a minimizer, which yields the objective function value 0.2. And $[1.05, 1.05, 2, 2]^{\mathrm{T}}$ achieves the same objective value, i.e., it is also a minimizer. When interpreting the values $[1, 1, 2, 2.1]^{\mathrm{T}}$ in (17a) *as cell averages, the first minimizer $[1, 1.1, 2, 2]^{\mathrm{T}}$ modifies fewer cells (two cells) than the second minimizer $[1.05, 1.05, 2, 2]^{\mathrm{T}}$, which modifies three. In this simple example* (17)*, both* `ClipAndAssuredSum` *and splitting methods listed in this section produce the second minimizer, which is not the "sparsest solution" in the sense of modifying the fewest number of cells.*

2.2. Davis-Yin splitting for ℓ^2 minimization

Both sets Λ_1 and Λ_2 in (14) are convex and closed, which implies the objective function of (14) is a proper closed strongly convex function and thus has a unique minimizer. In [28], DRS (12) was used to solve (14) with optimal parameters. Here we provide an easier alternative approach based on DYS method, which serves as an important component in constructing the splitting method for L^1 model in this paper. To apply DYS (13), we choose

$$f(\boldsymbol{x}) = \iota_{\Lambda_1}(\boldsymbol{x}), \quad g(\boldsymbol{x}) = \iota_{\Lambda_2}(\boldsymbol{x}), \quad \text{and} \quad h(\boldsymbol{x}) = \frac{1}{2\alpha}\|\boldsymbol{x} - \boldsymbol{u}\|_2^2. \tag{18}$$

Let $\mathbf{A}^+ = \mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{A}^{\mathrm{T}})^{-1}$ denote the pseudo inverse of the matrix $\mathbf{A}$. The proximal operator for the function f defined in (18) is given by

$$\mathtt{prox}_f^{\gamma}(\boldsymbol{x}) = \mathbf{A}^+(b - \mathbf{A}\boldsymbol{x}) + \boldsymbol{x}. \tag{19}$$

The proximal operator of an indicator of a set is the Euclidean projection onto that set. Thus, the $\mathtt{prox}_g^{\gamma}$ projects each point x_i, $i = 1, \cdots, N$, onto the admissible set $[m, M]$, i.e., it is the cut-off operation:

$$[\mathtt{prox}_g^{\gamma}(\boldsymbol{x})]_i = \min(\max(x_i, m), M), \quad \forall i = 1, \cdots, N. \tag{20}$$

Here, the subscript i denotes the i-th component in corresponding vector. In addition, we have

$$\nabla h = \frac{1}{\alpha}(\boldsymbol{x} - \boldsymbol{u}) \quad \text{and} \quad \mathtt{prox}_h^{\gamma}(\boldsymbol{x}) = \frac{\alpha}{\gamma + \alpha}\boldsymbol{x} + \frac{\gamma}{\gamma + \alpha}\boldsymbol{u} \tag{21}$$

Substituting (19)-(21) into (13) and taking the step size $\gamma = \frac{1}{L} = \alpha$, where $L = \frac{1}{\alpha}$ is the Lipschitz constant of ∇h, we obtain:

$$\boldsymbol{X}^{k+1/2} = \min(\max(\boldsymbol{Z}^k, m), M), \tag{22a}$$

$$\boldsymbol{Y}^{k+1} = \boldsymbol{X}^{k+1/2} - \boldsymbol{Z}^k + \boldsymbol{u}, \tag{22b}$$

$$\boldsymbol{X}^{k+1} = \mathbf{A}^+(b - \mathbf{A}\boldsymbol{Y}^{k+1}) + \boldsymbol{Y}^{k+1}, \tag{22c}$$

$$\boldsymbol{Z}^{k+1} = \boldsymbol{Z}^k + \boldsymbol{X}^{k+1} - \boldsymbol{X}^{k+1/2}. \tag{22d}$$

In (22a), the min and max denote entrywise operation. We initialize the algorithm above by setting $\boldsymbol{Z}^0 = \boldsymbol{u}$. To terminate the iteration, we can choose a small positive tolerance ϵ near machine zero and employ the stopping criterion $\|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}^k\|_{2h} < \epsilon$, where norm $\|\cdot\|_{2h} = h^{d/2}\|\cdot\|_2$.

A practical advantage of the DYS (22) is that it does not require any parameter tuning. Although an optimal parameter selection guideline exists for DRS in the scalar cases [29], no established principle is

currently known for choosing parameters in a DRS when applied to an invariant domain (2). In such cases, a parameter-free DYS is preferable. In practice, DYS with the step size $\gamma = \frac{1}{L}$ is empirically quite efficient, often converges to machine zero accuracy within a few iterations. See a numerical comparison of DYS with other similar methods for solving (14) in [1]. Thus, for compressible Euler equations, we propose using the DYS method to solve the L^2 minimization problem (8), as will be shown in Section 3.

2.3. Douglas-Rachford splitting for ℓ^1 minimization only for scalar variables

We first consider the DRS method (12) to solve ℓ^1 minimization (15), with functions g and h chosen as follows:

$$g(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{u}\|_1 + \iota_{\Lambda_2}(\boldsymbol{x}) \quad \text{and} \quad h(\boldsymbol{x}) = \iota_{\Lambda_1}(\boldsymbol{x}). \tag{23}$$

The proximal of h above is the same as (19). As shown in Appendix A, the proximal operator of g is,

$$[\texttt{prox}_g^\gamma(\boldsymbol{x})]_i = \max\{\min\{u_i + \mathrm{S}_\gamma(x_i - u_i), M\}, m\}, \tag{24a}$$

$$\mathrm{S}_\gamma(a) = \mathrm{sgn}(a)\max\{|a| - \gamma, 0\}. \tag{24b}$$

The DRS method can easily be written out with these explicit formulae of proximal operators, and it converges with any positive step size $\gamma > 0$ and any relaxation parameter $\lambda \in (0, 2)$.

However, when extending from $G = [m, M]$ to (2) for systems of conservation laws such as the Euler equations, the formula above and its derivation no longer hold, and there is no simple analytical expression for the proximal operator when combining the ℓ^1 norm with an invariant-domain-preserving constraint (2).

2.4. Douglas-Rachford splitting combined with Davis-Yin splitting for for ℓ^1 minimization

We now consider the DRS method (12) to solve ℓ^1 minimization (15), with functions g and h chosen as follows:

$$g(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{u}\|_1 \quad \text{and} \quad h(\boldsymbol{x}) = \iota_{\Lambda_1}(\boldsymbol{x}) + \iota_{\Lambda_2}(\boldsymbol{x}). \tag{25}$$

Define a shrinkage operator $\mathrm{S}_\gamma(x) = \mathrm{sgn}(x)\max\{|x| - \gamma, 0\}$. Associated with the function g above, the proximal is given by

$$[\texttt{prox}_g^\gamma(\boldsymbol{x})]_i = u_i + \mathrm{S}_\gamma(x_i - u_i) = u_i + \mathrm{sgn}(x_i - u_i)\max\{|x_i - u_i| - \gamma, 0\}, \tag{26}$$

where the subscript i denotes the i-th component in corresponding vector. The proximal of the function h in (25) lacks a tractable analytical form. To overcome this difficulty, we design a nested approach, where the inner DYS iteration computes $\texttt{prox}_h^\gamma$ numerically. Notice that we have the following relation by definition of a proximal operator,

$$\texttt{prox}_h^\gamma(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{z}\in\mathbb{R}^N}\ \frac{1}{2\gamma}\|\boldsymbol{z} - \boldsymbol{x}\|_2^2 + \iota_{\Lambda_1}(\boldsymbol{z}) + \iota_{\Lambda_2}(\boldsymbol{z}), \tag{27}$$

which is nothing but the problem (14). Therefore, to implement the proximal operator (27), it can be solved numerically by applying DYS (22). After obtaining $\texttt{prox}_h^\gamma$, we use the DRS (12) to compute a minimizer of (15). We initialize the algorithm above by setting $\boldsymbol{Y}^0 = \boldsymbol{u}$ and $\boldsymbol{X}^0 = \texttt{prox}_g^\gamma(\boldsymbol{Y}^0)$. To terminate the iteration, we choose a small positive tolerance ϵ and employ the stopping criterion $\|\boldsymbol{Y}^{k+1} - \boldsymbol{Y}^k\|_{2h} < \epsilon$.

In other words, this is a nested method with DRS as an outer loop and DYS as the inner loop for the subproblem (27). This approach can be easily extended to the invariant domain (2).

3. An efficient implementation of invariant-domain-preserving limiters by splitting methods

As an example, consider solving compressible Euler equations in a bounded spatial domain $\Omega \subset \mathbb{R}^d$ over the time interval $t \in [0, T]$. Let $\boldsymbol{U} = [\rho, \boldsymbol{m}^{\mathrm{T}}, E]^{\mathrm{T}}$ denote the conservative variables for density, momentum, and total energy. The compressible Euler equations are

$$\partial_t \boldsymbol{U} + \nabla \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}) = \boldsymbol{0}, \quad \text{where} \quad \boldsymbol{F}^{\mathrm{a}} = \begin{bmatrix} \boldsymbol{m} \\ \frac{1}{\rho}\boldsymbol{m} \otimes \boldsymbol{m} + p\mathbf{I} \\ (E+p)\frac{\boldsymbol{m}}{\rho} \end{bmatrix}. \tag{28}$$

The total energy is $E = \rho e + \frac{\|\boldsymbol{m}\|_2^2}{2\rho}$, where e denotes the internal energy. For the ideal gas, the equation of state is $p = (\gamma - 1)\rho e$, where $\gamma > 1$ is a constant, e.g., $\gamma = 1.4$ for air. When vacuum does not occur, a physically meaningful solution to (28) should have positive density and positive internal energy, namely it should stay in the admissible set G. We will discuss optimization-based limiters for enforcing a numerical invariant domain G^ε in (2).

3.1. An ℓ^2 minimization invariant-domain-preserving limiter for cell averages

High order explicit schemes with large time steps, as well as implicit schemes, may produce negative cell averages. We propose the optimization-based approach that incorporates preserving global conservation and invariant domain as constraints to postprocess the cell averages in DG solution.

In limiter (8), the vector-valued piecewise constant polynomial $\boldsymbol{\mathcal{X}}_h$ minimizes the L^2 distance to the cell average of the DG polynomial $\boldsymbol{\mathcal{U}}_h = [\rho_h, \boldsymbol{m}_h{}^{\mathrm{T}}, E_h]^{\mathrm{T}}$. We introduce a matrix $\overline{\mathbf{U}} \in \mathbb{R}^{N\times(2+d)}$ to represent cell averages of the DG polynomial $\boldsymbol{\mathcal{U}}_h$, that is, the i-th row of $\overline{\mathbf{U}}$ equals

$$\overline{\boldsymbol{\mathcal{U}}_h}^{\mathrm{T}}|_{K_i} = \left[\frac{1}{|K_i|}\int_{K_i} \rho_h \quad \frac{1}{|K_i|}\int_{K_i} \boldsymbol{m}_h{}^{\mathrm{T}} \quad \frac{1}{|K_i|}\int_{K_i} E_h \right]. \tag{29}$$

Then, model (8) is equivalent to the following unconstrained minimization problem in matrix-vector form:

$$\begin{aligned} &\min_{\mathbf{X}\in\mathbb{R}^{N\times(2+d)}} \frac{1}{2\alpha}\|\mathbf{X} - \overline{\mathbf{U}}\|_F^2 + \iota_{\Lambda_1}(\mathbf{X}) + \iota_{\Lambda_2}(\mathbf{X}), \\ &\quad \text{where} \quad \Lambda_1 = \{\mathbf{X}\colon\ \mathbf{A}\mathbf{X} = \boldsymbol{b}^{\mathrm{T}}\} \\ &\qquad \text{and} \quad \Lambda_2 = \{\mathbf{X}\colon\ \text{the } i\text{-th row } \mathbf{X}_i{}^{\mathrm{T}} \in G^\varepsilon,\ \forall i\}. \end{aligned} \tag{30}$$

Here, $\|\cdot\|_F$ is the matrix Frobenius norm. On a uniform mesh, we have $\mathbf{A} = [1, 1, \cdots, 1] \in \mathbb{R}^{1\times N}$ and $\boldsymbol{b}^{\mathrm{T}} = \mathbf{A}\overline{\mathbf{U}}$. The closed convex sets Λ_1 and Λ_2 represent global conservation and invariant-domain-preserving constraints, respectively. The functions ι_{Λ_1} and ι_{Λ_2} are convex, indicating that (30) is a strongly convex minimization problem. Therefore, the solution of (30) is unique.

The limiter (30) improves the accuracy of the DG solution in the following sense:

Theorem 1. *Let $\mathbf{X}^*$ denote the minimizer of* (30) *and let $\overline{\mathbf{U}^{\mathrm{exact}}} \in \mathbb{R}^{N\times(2+d)}$ be a matrix storing the cell averages of an exact solution. If the DG solution has the same integrals over the whole domain as the exact solution, and the DG cell averages violate the invariant domain (i.e., $\overline{\mathbf{U}} \notin \Lambda_2$), then the minimizer $\mathbf{X}^*$ improves the accuracy:*

$$\|\mathbf{X}^* - \overline{\mathbf{U}^{\mathrm{exact}}}\|_F < \|\overline{\mathbf{U}} - \overline{\mathbf{U}^{\mathrm{exact}}}\|_F. \tag{31}$$

Proof. For two matrices $\mathbf{B} = [b]_{ij}$ and $\mathbf{C} = [c]_{ij}$ of same sizes, define the inner product $\mathbf{B} : \mathbf{C} = \sum_{ij} b_{ij}c_{ij}$. The sets Λ_1 and Λ_2 are convex and closed, thus $\Lambda_1 \cap \Lambda_2$ is a convex closed set. Thus, $\overline{\mathbf{U}^{\mathrm{exact}}}$ and $\mathbf{X}^*$ belongs to $\Lambda_1 \cap \Lambda_2$, which implies

$$\eta\overline{\mathbf{U}^{\mathrm{exact}}} + (1-\eta)\mathbf{X}^* \in \Lambda_1 \cap \Lambda_2, \quad \forall \eta \in [0, 1]. \tag{32}$$

Define

$$\phi(\eta) := \|\overline{\mathbf{U}} - (\eta\overline{\mathbf{U}^{\text{exact}}} + (1-\eta)\mathbf{X}^*)\|_F^2 = \eta^2\|\overline{\mathbf{U}^{\text{exact}}} - \mathbf{X}^*\|_F^2 - 2\eta(\overline{\mathbf{U}} - \mathbf{X}^*) : (\overline{\mathbf{U}^{\text{exact}}} - \mathbf{X}^*) + \|\overline{\mathbf{U}} - \mathbf{X}^*\|_F^2. \quad (33)$$

If $\|\overline{\mathbf{U}^{\text{exact}}} - \mathbf{X}^*\|_F = 0$, then $\|\overline{\mathbf{U}^{\text{exact}}} - \mathbf{X}^*\|_F = 0 < \|\overline{\mathbf{U}^{\text{exact}}} - \overline{\mathbf{U}}\|_F$, because $\overline{\mathbf{U}^{\text{exact}}} \neq \overline{\mathbf{U}}$ due to $\overline{\mathbf{U}} \notin \Lambda_2$. Otherwise, it is obvious that $\phi(\eta)$ is a quadratic function with respect to η. From (30), we know $\mathbf{X}^*$ minimizes $\|\overline{\mathbf{U}} - \mathbf{X}\|_F^2$ for all $\mathbf{X} \in \Lambda_1 \cap \Lambda_2$. Thus, $\phi(\eta)$ achieves its minimum at $\eta = 0$, which implies $(\mathbf{X}^* - \overline{\mathbf{U}}) : (\overline{\mathbf{U}^{\text{exact}}} - \mathbf{X}^*) \geq 0$. Therefore, we have

$$\begin{aligned}\|\overline{\mathbf{U}^{\text{exact}}} - \overline{\mathbf{U}}\|_F^2 &= \|\overline{\mathbf{U}^{\text{exact}}} - \mathbf{X}^* + \mathbf{X}^* - \overline{\mathbf{U}}\|_F^2 \\ &= \|\overline{\mathbf{U}^{\text{exact}}} - \mathbf{X}^*\|_F^2 + 2(\mathbf{X}^* - \overline{\mathbf{U}}) : (\overline{\mathbf{U}^{\text{exact}}} - \mathbf{X}^*) + \|\mathbf{X}^* - \overline{\mathbf{U}}\|_F^2 > \|\overline{\mathbf{U}^{\text{exact}}} - \mathbf{X}^*\|_F^2. \quad (34)\end{aligned}$$

□

Finally, we present an equivalent reformulation of problem (30) to facilitate the design of splitting methods. Partition columns in the matrix $\mathbf{X} = [\boldsymbol{\rho}, \boldsymbol{m}_1, \cdots, \boldsymbol{m}_d, \boldsymbol{E}]$ such that, the first column $\boldsymbol{\rho}$ contains the cell averages of density, the next d columns $\boldsymbol{m}_1$ to $\boldsymbol{m}_d$ represent the cell averages of the momentum components, and the last column $\boldsymbol{E}$ corresponds to the cell averages of total energy. Similarly, we define the matrix $\overline{\mathbf{U}} = [\boldsymbol{u}, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_d, \boldsymbol{w}]$ and let $\boldsymbol{b}^{\mathrm{T}} = [b_\rho, b_{m_1}, \cdots, b_{m_d}, b_E]$. Then, the minimization problem (30) is equivalent to

$$\begin{aligned}&\min_{\mathbf{X}} \frac{1}{2\alpha}\Big(\|\boldsymbol{\rho} - \boldsymbol{u}\|_2^2 + \sum_{i=1}^{d}\|\boldsymbol{m}_i - \boldsymbol{v}_i\|_2^2 + \|\boldsymbol{E} - \boldsymbol{w}\|_2^2\Big) + \iota_{\Lambda_1}(\boldsymbol{\rho}, \boldsymbol{m}_1, \cdots, \boldsymbol{m}_d, \boldsymbol{E}) + \iota_{\Lambda_2}(\boldsymbol{\rho}, \boldsymbol{m}_1, \cdots, \boldsymbol{m}_d, \boldsymbol{E}), \\ &\text{where } \Lambda_1 = \{\mathbf{X}\colon \mathbf{A}\boldsymbol{\rho} = b_\rho,\ \mathbf{A}\boldsymbol{m}_1 = b_{m_1},\ \cdots,\ \mathbf{A}\boldsymbol{m}_d = b_{m_d},\ \mathbf{A}\boldsymbol{E} = b_E\} \quad (35) \\ &\quad \text{and } \Lambda_2 = \{\mathbf{X}\colon [\rho_i, m_{1i}, \cdots, m_{di}, E_i]^{\mathrm{T}} \in G^\varepsilon,\ \forall i\}.\end{aligned}$$

Remark 2. *We emphasize that the work in [29, 27, 28] focuses solely on a scalar variable and is therefore applicable only to compressible NS equations with Strang splitting [31, 27], but not for Euler equations. The proposed limiter in this section is much more general than the method in [27]. In contrast, the invariant-domain-preserving limiter* (30) *directly preserves the convex invariant domain thus is more broadly applicable in compressible flow simulations.*

3.2. Splitting methods for ℓ^2 minimization limiter

An efficient implementation of splitting methods to solve (35) requires computing proximal operators easily and efficiently. We split the objective function in a manner that facilitates the derivation of explicit formulae.

- To apply DRS method (12), we choose

$$g(\mathbf{X}) = \frac{1}{2\alpha}\Big(\|\boldsymbol{\rho} - \boldsymbol{u}\|_2^2 + \sum_{i=1}^{d}\|\boldsymbol{m}_i - \boldsymbol{v}_i\|_2^2 + \|\boldsymbol{E} - \boldsymbol{w}\|_2^2\Big) + \iota_{\Lambda_1}(\boldsymbol{\rho}, \boldsymbol{m}_1, \cdots, \boldsymbol{m}_d, \boldsymbol{E}), \quad (36a)$$

$$h(\mathbf{X}) = \iota_{\Lambda_2}(\boldsymbol{\rho}, \boldsymbol{m}_1, \cdots, \boldsymbol{m}_d, \boldsymbol{E}). \quad (36b)$$

- To apply DYS method (13), we choose

$$f(\mathbf{X}) = \iota_{\Lambda_1}(\boldsymbol{\rho}, \boldsymbol{m}_1, \cdots, \boldsymbol{m}_d, \boldsymbol{E}), \quad (37a)$$

$$g(\mathbf{X}) = \iota_{\Lambda_2}(\boldsymbol{\rho}, \boldsymbol{m}_1, \cdots, \boldsymbol{m}_d, \boldsymbol{E}), \quad (37b)$$

$$h(\mathbf{X}) = \frac{1}{2\alpha}\Big(\|\boldsymbol{\rho} - \boldsymbol{u}\|_2^2 + \sum_{i=1}^{d}\|\boldsymbol{m}_i - \boldsymbol{v}_i\|_2^2 + \|\boldsymbol{E} - \boldsymbol{w}\|_2^2\Big). \quad (37c)$$

Let us derive the proximal associated with function g in (36a). The proximal associated with function f in (37a) can be derived similarly or simply set $\alpha = +\infty$. By definition of the proximal operator, for any given $\mathbf{X} = [\boldsymbol{\rho}, \boldsymbol{m}_1, \cdots, \boldsymbol{m}_d, \boldsymbol{E}]$, we need to find $\mathbf{Z} = [\boldsymbol{\xi}, \boldsymbol{\eta}_1, \cdots, \boldsymbol{\eta}_d, \boldsymbol{\zeta}]$ to minimize the following function:

$$\frac{\gamma}{2\alpha}\Big(\|\boldsymbol{\xi}-\boldsymbol{u}\|_2^2+\sum_{i=1}^{d}\|\boldsymbol{\eta}_i-\boldsymbol{v}_i\|_2^2+\|\boldsymbol{\zeta}-\boldsymbol{w}\|_2^2\Big)+\gamma\iota_{\Lambda_1}(\boldsymbol{\xi},\boldsymbol{\eta}_1,\cdots,\boldsymbol{\eta}_d,\boldsymbol{\zeta}) \\ +\frac{1}{2}\|\boldsymbol{\xi}-\boldsymbol{\rho}\|_2^2+\frac{1}{2}\sum_{i=1}^{d}\|\boldsymbol{\eta}_i-\boldsymbol{m}_i\|_2^2+\frac{1}{2}\|\boldsymbol{\zeta}-\boldsymbol{E}\|_2^2. \quad (38)$$

By definition of the indicator function ι_{Λ_1}, minimizing (38) is equivalent to solving the following decoupled constraint optimization problems:

$$\min_{\boldsymbol{\xi}\in\mathbb{R}^N}\frac{\gamma}{2\alpha}\|\boldsymbol{\xi}-\boldsymbol{u}\|_2^2+\frac{1}{2}\|\boldsymbol{\xi}-\boldsymbol{\rho}\|_2^2 \quad \text{subject to} \quad \mathbf{A}\boldsymbol{\xi}=b_\rho, \quad (39a)$$

$$\min_{\boldsymbol{\eta}_i\in\mathbb{R}^N}\frac{\gamma}{2\alpha}\|\boldsymbol{\eta}_i-\boldsymbol{v}_i\|_2^2+\frac{1}{2}\|\boldsymbol{\eta}_i-\boldsymbol{m}_i\|_2^2 \quad \text{subject to} \quad \mathbf{A}\boldsymbol{\eta}_i=b_{m_i}, \quad \text{for } i=1,\cdots,d, \quad (39b)$$

$$\min_{\boldsymbol{\zeta}\in\mathbb{R}^N}\frac{\gamma}{2\alpha}\|\boldsymbol{\zeta}-\boldsymbol{w}\|_2^2+\frac{1}{2}\|\boldsymbol{\zeta}-\boldsymbol{E}\|_2^2 \quad \text{subject to} \quad \mathbf{A}\boldsymbol{\zeta}=b_E. \quad (39c)$$

Let us start with (39a). Define a Lagrange multiplier as follows:

$$\mathcal{L}=\frac{\gamma}{2\alpha}\|\boldsymbol{\xi}-\boldsymbol{u}\|_2^2+\frac{1}{2}\|\boldsymbol{\xi}-\boldsymbol{\rho}\|_2^2+\lambda(\mathbf{A}\boldsymbol{\xi}-b_\rho). \quad (40)$$

Notice, the matrix $\mathbf{A}=[1\ 1\ \cdots\ 1]\in\mathbb{R}^{1\times N}$. Take partial derivatives, we get

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\xi}}=\Big(\frac{\gamma}{\alpha}+1\Big)\boldsymbol{\xi}-\frac{\gamma}{\alpha}\boldsymbol{u}-\boldsymbol{\rho}+\lambda\mathbf{A}^{\mathrm{T}}=\mathbf{0}, \quad (41a)$$

$$\frac{\partial\mathcal{L}}{\partial\lambda}=\mathbf{A}\boldsymbol{\xi}-b_\rho=0. \quad (41b)$$

Left multiply $\mathbf{A}$ on both side of (41a), from the constraints $\mathbf{A}\boldsymbol{\xi}=b_\rho$ and $\mathbf{A}\boldsymbol{u}=b_\rho$, we have

$$b_\rho-\mathbf{A}\boldsymbol{\rho}+\lambda\mathbf{A}\mathbf{A}^{\mathrm{T}}=0 \quad\Rightarrow\quad \lambda=(\mathbf{A}\mathbf{A}^{\mathrm{T}})^{-1}(\mathbf{A}\boldsymbol{\rho}-b_\rho). \quad (42)$$

Substituting the expression for λ above into (41a) and recalling the pseudo inverse of the matrix $\mathbf{A}$ is given by $\mathbf{A}^+=\mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{A}^{\mathrm{T}})^{-1}$. We obtain

$$\boldsymbol{\xi}=\frac{\alpha}{\gamma+\alpha}\big(\mathbf{A}^+(b_\rho-\mathbf{A}\boldsymbol{\rho})+\boldsymbol{\rho}\big)+\frac{\gamma}{\gamma+\alpha}\boldsymbol{u}. \quad (43)$$

The (39b) and (39c) can be addressed similarly. Therefore, associated with the function g in (36a), the proximal $\mathtt{prox}_g^\gamma$ maps

$$\boldsymbol{\rho} \quad\to\quad \frac{\alpha}{\gamma+\alpha}\big(\mathbf{A}^+(b_\rho-\mathbf{A}\boldsymbol{\rho})+\boldsymbol{\rho}\big)+\frac{\gamma}{\gamma+\alpha}\boldsymbol{u}, \quad (44a)$$

$$\boldsymbol{m}_i \quad\to\quad \frac{\alpha}{\gamma+\alpha}\big(\mathbf{A}^+(b_{m_i}-\mathbf{A}\boldsymbol{m}_i)+\boldsymbol{m}_i\big)+\frac{\gamma}{\gamma+\alpha}\boldsymbol{v}_i, \quad \text{for } i=1,\cdots,d, \quad (44b)$$

$$\boldsymbol{E} \quad\to\quad \frac{\alpha}{\gamma+\alpha}\big(\mathbf{A}^+(b_E-\mathbf{A}\boldsymbol{E})+\boldsymbol{E}\big)+\frac{\gamma}{\gamma+\alpha}\boldsymbol{w}. \quad (44c)$$

Notice, the proximal of an indicator of a set is the Euclidean projection onto that set. To this end, we need the projection point onto the numerical admissible set G^ε, i.e., computing the proximal for h in (36b) an g in (37b). Such a projection is by no means trivial. For the ease of presentation, the projection formula to G^ε and its derivation are given in the Appendix.

3.3. An ℓ^1 invariant-domain-preserving cell average limiter

Consider cell averages $\overline{\mathbf{U}} = [\boldsymbol{u}, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_d, \boldsymbol{w}]$ of a DG polynomial. For compressible Euler equations, the L^1 limiter (9) is equivalent to the following non-constraint optimization problem in matrix-vector form:

$$\begin{aligned} &\min_{\mathbf{X}} \|\boldsymbol{\rho} - \boldsymbol{u}\|_1 + \sum_{i=1}^{d} \|\boldsymbol{m}_i - \boldsymbol{v}_i\|_1 + \|\boldsymbol{E} - \boldsymbol{w}\|_1 + \iota_{\Lambda_1}(\boldsymbol{\rho}, \boldsymbol{m}_1, \cdots, \boldsymbol{m}_d, \boldsymbol{E}) + \iota_{\Lambda_2}(\boldsymbol{\rho}, \boldsymbol{m}_1, \cdots, \boldsymbol{m}_d, \boldsymbol{E}), \\ &\text{where } \Lambda_1 = \{\mathbf{X}\colon \mathbf{A}\boldsymbol{\rho} = b_\rho,\ \mathbf{A}\boldsymbol{m}_1 = b_{m_1},\ \cdots,\ \mathbf{A}\boldsymbol{m}_d = b_{m_d},\ \mathbf{A}\boldsymbol{E} = b_E\} \\ &\quad \text{and } \Lambda_2 = \{\mathbf{X}\colon [\rho_i, m_{1i}, \cdots, m_{di}, E_i]^{\mathrm{T}} \in G^{\varepsilon},\ \forall i\}. \end{aligned} \tag{45}$$

The objective function of (45) is convex but not strongly convex, which implies the non-uniqueness of the minimizer. We consider the DRS method (12) to solve above minimization problem, where functions g and h are chosen as follows:

$$g(\mathbf{X}) = \|\boldsymbol{\rho} - \boldsymbol{u}\|_1 + \sum_{i=1}^{d} \|\boldsymbol{m}_i - \boldsymbol{v}_i\|_1 + \|\boldsymbol{E} - \boldsymbol{w}\|_1 \tag{46a}$$

$$h(\mathbf{X}) = \iota_{\Lambda_1}(\boldsymbol{\rho}, \boldsymbol{m}_1, \cdots, \boldsymbol{m}_d, \boldsymbol{E}) + \iota_{\Lambda_2}(\boldsymbol{\rho}, \boldsymbol{m}_1, \cdots, \boldsymbol{m}_d, \boldsymbol{E}). \tag{46b}$$

Let us derive the proximal associated with function g in (46a). By definition of the proximal operator, for any given $\mathbf{X} = [\boldsymbol{\rho}, \boldsymbol{m}_1, \cdots, \boldsymbol{m}_d, \boldsymbol{E}]$, we need to find $\mathbf{Z} = [\boldsymbol{\xi}, \boldsymbol{\eta}_1, \cdots, \boldsymbol{\eta}_d, \boldsymbol{\zeta}]$ to minimize the following function:

$$\gamma\Big(\|\boldsymbol{\xi} - \boldsymbol{u}\|_1 + \sum_{i=1}^{d} \|\boldsymbol{\eta}_i - \boldsymbol{v}_i\|_1 + \|\boldsymbol{\zeta} - \boldsymbol{w}\|_1\Big) + \frac{1}{2}\Big(\|\boldsymbol{\xi} - \boldsymbol{\rho}\|_2^2 + \sum_{i=1}^{d} \|\boldsymbol{\eta}_i - \boldsymbol{m}_i\|_2^2 + \|\boldsymbol{\zeta} - \boldsymbol{E}\|_2^2\Big). \tag{47}$$

Minimizing the function above is equivalent to solving the following decoupled optimization problems:

$$\min_{\boldsymbol{\xi} \in \mathbb{R}^N} \gamma\|\boldsymbol{\xi} - \boldsymbol{u}\|_1 + \frac{1}{2}\|\boldsymbol{\xi} - \boldsymbol{\rho}\|_2^2, \tag{48a}$$

$$\min_{\boldsymbol{\eta}_i \in \mathbb{R}^N} \gamma\|\boldsymbol{\eta}_i - \boldsymbol{v}_i\|_1 + \frac{1}{2}\|\boldsymbol{\eta}_i - \boldsymbol{m}_i\|_2^2, \quad \text{for } i = 1, \cdots, d, \tag{48b}$$

$$\min_{\boldsymbol{\zeta} \in \mathbb{R}^N} \gamma\|\boldsymbol{\zeta} - \boldsymbol{w}\|_1 + \frac{1}{2}\|\boldsymbol{\zeta} - \boldsymbol{E}\|_2^2, \tag{48c}$$

for which solutions were established in Section 2, see (25) and (26). Recall the shrinkage operator $\mathrm{S}_\gamma(x) = \mathrm{sgn}(x)\max\{|x| - \gamma, 0\}$. Associated with the function g in (46a), the proximal $\mathtt{prox}_g^\gamma$ maps

$$\rho_i \to u_i + \mathrm{S}_\gamma(\rho_i - u_i), \tag{49a}$$

$$m_{1i} \to v_i + \mathrm{S}_\gamma(m_{1i} - v_i), \qquad \cdots, \qquad m_{di} \to v_i + \mathrm{S}_\gamma(m_{1i} - v_i), \tag{49b}$$

$$E_i \to w_i + \mathrm{S}_\gamma(E_i - w_i), \tag{49c}$$

where the subscript i denotes the i-th component in corresponding vector. The proximal of the function h in (48b) lacks a tractable analytical form. To overcome this difficulty, we design a nested approach, where the inner DYS iteration computes $\mathtt{prox}_h^\gamma$ numerically. By definition of a proximal operator, the $\mathtt{prox}_h^\gamma$ gives the solution to the following minimization problem: given $\mathbf{X} \in \mathbb{R}^{N\times(2+d)}$, find $\mathbf{Z} \in \mathbb{R}^{N\times(2+d)}$ that

$$\min_{\mathbf{Z} \in \mathbb{R}^N} \frac{1}{2\gamma}\|\mathbf{Z} - \mathbf{X}\|_F^2 + \iota_{\Lambda_1}(\mathbf{Z}) + \iota_{\Lambda_2}(\mathbf{Z}). \tag{50}$$

Notice, problem (50) is exactly (30) with parameter $\alpha = \gamma$. Therefore, it can be solved by applying DYS method. After obtaining $\mathtt{prox}_h^\gamma$, we utilize the DRS (12) to compute a minimizer of (45). To start the algorithm, we set $\boldsymbol{Y}^0 = [\boldsymbol{u}, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_d, \boldsymbol{w}]$ and $\boldsymbol{X}^0 = \mathtt{prox}_h^\gamma(\boldsymbol{Y}^0)$. To terminate the iteration, we choose the stopping criterion $\|\boldsymbol{Y}^{k+1} - \boldsymbol{Y}^k\|_{2h} < \epsilon$ with a small positive tolerance ϵ.

4. Numerical experiments

In this section, we show results of applying the optimization-based limiters to a series of representative tests. These include studies of using one-dimensional manufactured data and two-dimensional simulations of demanding gas dynamic benchmarks for testing robustness of positivity-preserving schemes.

For solving ℓ^2-norm minimizations (14) and (35) in all the numerical tests in this paper, we use DYS with a fixed step size $\gamma = \frac{1}{L}$, where the Lipschitz constant is $L = \frac{1}{\alpha}$,

For the ℓ^1 minimizations (15) and (45), we use DRS. In each iteration of DRS, the proximal operator for summation of two indicator functions, such as (50), is numerically solved by DYS. The step size in the DYS is still $\frac{1}{L}$ where L is the Lipschitz constant for the ℓ^2 term. The parameters in DRS are taken as follows. We use a relaxation parameter $\lambda = 1$ and a step size $\gamma > 0$ obtained from tuning on a few data sets. For instance, for using the optimization based ℓ^1-norm limiter in each time step in a time dependent problem, we first run DRS with a different choices of $\gamma > 0$ at a few time steps when the limiter is invoked, then choose the best value of γ in the sense of the fastest convergence of DRS for all future time steps.

We remark that optimal parameters of λ and γ in DRS for solving (14) are given in [29], which was derived from convergence rate analysis. Unfortunately, neither the analysis nor the same parameter formulae can be extended to the invariant domain (2).

4.1. Synthetic tests in one dimension

In this part, our primary objective is to examine the optimization methods through simple tests, including linear advection governed traveling waves and manufactured perturbation of Lax shock tube data for Euler equations. We generate out-of-bound data by the baseline DG scheme without any limiters, then apply the limiters to data, without integrating optimization limiters into baseline PDE simulators, so that we only study the performance of optimizers here.

In all one-dimensional tests, we chose $\varepsilon = 10^{-13}$ for numerical admissible set G^{ε}. In addition, the convergence tolerance for both DYS and DRS methods is set to $\epsilon = 10^{-13}$.

Example 5.1 (Traveling triangle and square waves). The linear advection equation $\partial_t u + \partial_x u = 0$ enjoys the maximum principle: if initial data is in $[m, M]$, then so is the solution belongs at any later time $t > 0$. The computational domain is $\Omega = [0, 3]$. We initialize right-moving triangle and square waves as follows and evolve to time $T = 1$.

$$u^0 = \begin{cases} 4x, & \text{if } x \in (0.25, 0.5], \\ -4x + 4, & \text{if } x \in (0.5, 0.75], \\ 2, & \text{if } x \in (1.25, 1.75], \\ 1, & \text{otherwise.} \end{cases} \tag{51}$$

Thus, the exact solution $u(x, t) \in [1, 2]$ for any $x \in \Omega$ and $t \in [0, T]$. To generate out-of-bound data, we solve the equation on a uniform mesh consisting of 300 cells. We employ a fourth-order Runge-Kutta (RK) DG method without any limiter. The time step size is set to 0.001, resulting in 1000 out-of-bound data sets for postprocessing.

We apply DYS method to solve the ℓ^2 model (14) and apply direct method (`ClipAndAssuredSum`) (16) and DRS method to solve the ℓ^1 model (15), respectively. After postprocessing, all cell averages are enforced within the bounds, see Figure 1. Notice that ℓ^1 minimizers are not necessarily unique, and it has been proven that ℓ^2 minimizer is one of ℓ^1 minimizers [5]. In general, the ℓ^1 minimizers found by either `ClipAndAssuredSum` or different splitting methods may be different from one another, and also different from ℓ^2 minimizer. Nonetheless, for this particular test, up to ϵ, there is no difference in the three results in Figure 1.

In terms of performance of these optimization solvers, for the ℓ^2 model, DYS method converges within 60 iterations on all data sets at different time steps. In contrast, the ℓ^1 model requires more DRS iterations, i.e., converges within 200 iterations on all data sets at different time steps. Hereinafter, we use the *number of projection to admissible set* to denote the total count of computations of the proximal operator for indicator

function $\iota_{\Lambda_2}(\boldsymbol{x})$. The total number of projections to the admissible set $[1, 2]$ is significantly higher for the ℓ^1 model because of its inner DYS iterations, see Figure 2. We use step size $\gamma = 10^{-10}$ in DRS for the ℓ^1 model. When the iteration sequences are sufficiently close to the fixed point, both the DYS and DRS methods exhibit asymptotic linear convergence, as illustrated in Figure 2.

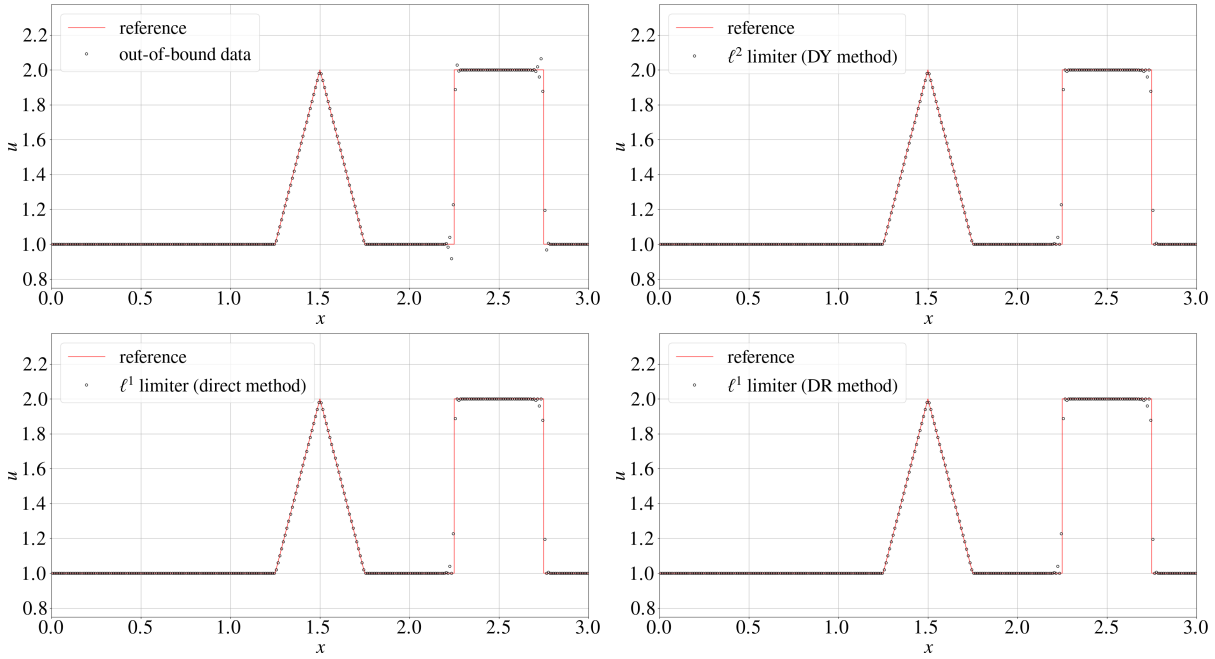

Figure 1: Top left: out-of-bound cell averages of data set 1000 (generated in time step 1000). Top right: the postprocessed result from the ℓ^2 limiter with DYS method. Bottom: the postprocessed results from the ℓ^1 limiter with direct method (`ClipAndAssuredSum`) and DRS method. There is no difference between ℓ^2 limited result and two ℓ^1 limited results up to ϵ.

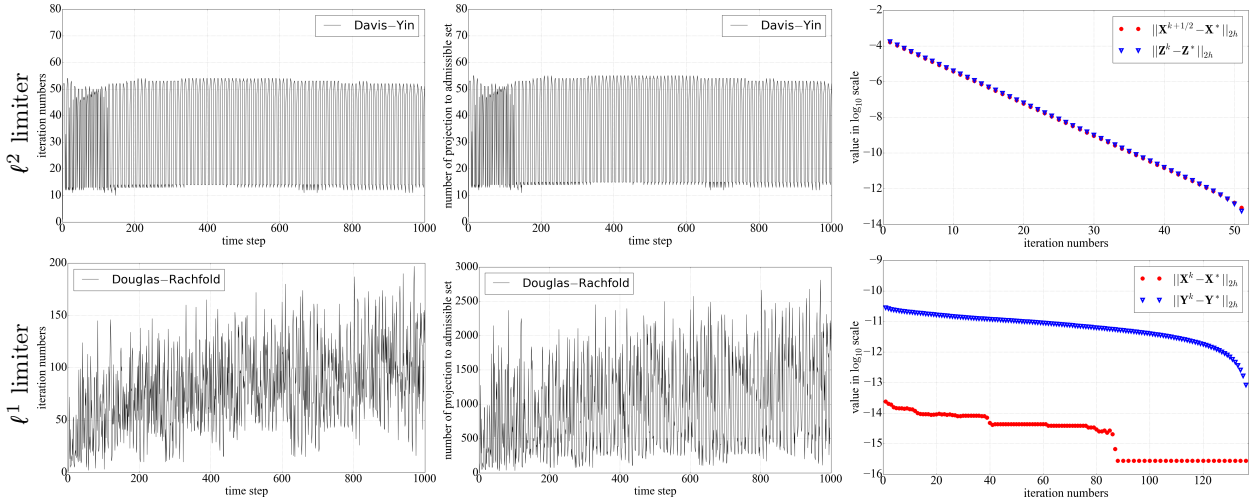

Figure 2: Top row: DYS method solving ℓ^2 model. Bottom row: DRS method nested with DYS of solving ℓ^1 model. From left to right: number of iterations for data at different time steps, number of projections, and the convergence error in processing one particular data set at time step 1000. Minimizers and fixed points, denoted by superscript star, are computed numerically by using sufficiently many iterations of splitting methods.

Example 5.2 (Perturbation of Lax shock tube data). We uniformly partition the computational domain $\Omega = [-5, 5]$ into 400 cells. To generate data violating the invariant domain, we compute cell averages of the exact solution at time $t = 1.3$ to the Lax shock tube problem for compressible Euler equations [44, Example 2] and apply perturbations near the shock location. Specifically, we perturb 5% of the cells, i.e., 20 cells in total. For the 10 cells in pre-shock region, we subtract the following random values sampled from a uniform distribution: on each cell K, we modify the cell average by

$$\overline{\boldsymbol{u}}\big|_K - \begin{bmatrix} 0.1 \max|\rho|\, \xi_\rho \\ 0.01 \max|m|\, \xi_m \\ 0.1 \max|E|\, \xi_E \end{bmatrix}, \quad \text{where} \quad \xi_\rho, \xi_m, \xi_E \overset{\text{iid}}{\sim} \text{Uniform}[1, 2]. \tag{52}$$

To ensure conservation, we add the same values back to the 10 cells in the post-shock region. In this way, we obtain one out-of-bound data set. To test our optimization solvers, we repeat and create 1000 manufactured data set.

We apply DYS method to solve the ℓ^2 model (35) and DRS method with $\gamma = 10^{-4}$ to solve the ℓ^1 model (45), respectively. After postprocessing, all cell averages are enforced within the bounds. For the ℓ^2 model, DYS method converges within 20 iterations on all data sets. Compared to the number of iterations in solving the ℓ^2 model, the ℓ^1 model takes more DRS iterations and converges within 200 iterations on all data sets. But, when compared with the total number of projections to the admissible set G^ε, the required projection numbers are significantly higher for the ℓ^1 model due to its inner DYS iterations, see Figure 3. We observe asymptotic linear convergence for both DYS and DRS methods, when iteration sequences are sufficiently close to the fixed point. See Figure 3 for an illustration.

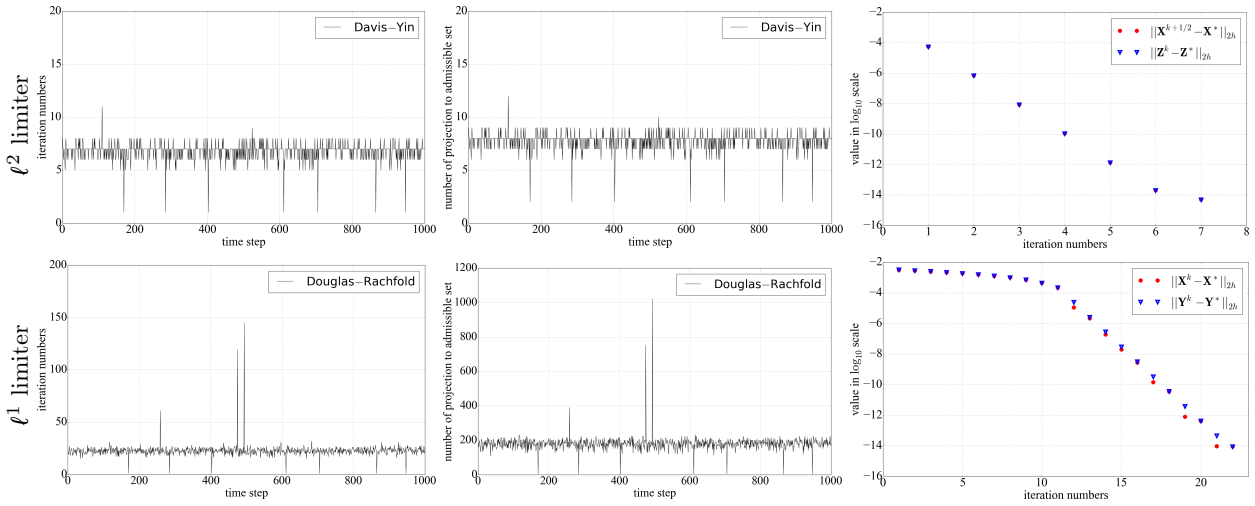

Figure 3: Top row: DYS method solving ℓ^2 model. Bottom row: DRS method nested with DYS of solving ℓ^1 model. From left to right: number of iterations for data at different time steps, total number of projections to admissible set, and the convergence error in processing the data set at time step 1000. Minimizers and fixed points, denoted by superscript star, are computed numerically by taking sufficiently many iterations.

4.2. Benchmark tests in two dimension

In this part, we apply limiters to DG schemes solving demanding test cases, including the Sedov blast wave and a high speed astrophysical jet. Our optimization-based postprocessing procedure preserves invariant domains effectively while producing high-quality solutions for challenging practical applications.

In all two-dimensional simulations, we apply the optimization-based cell average limiter after each stage of RK method, whenever there is a cell average out of the admissible set. Only when the cell averages are in the invariant domain, the Zhang-Shu limiter [46] can be further used to ensure quadrature points are also in

G. For all test cases except the astrophysical jet, we choose $\varepsilon = 10^{-13}$ in G^ε and the convergence tolerance $\epsilon = 10^{-13}$ for DYS and DRS methods. For the astrophysical jet, we use $\varepsilon = 10^{-8}$ and $\epsilon = 10^{-8}$ due to its extremely high speed. For solving the ℓ^1 limiter, DRS method with a step size $\gamma = 10^{-7}$ is used in all two dimensional examples.

Example 5.3 (Convergence study). We utilize the manufactured solution method on domain $\Omega = [0, 1]^2$ with end time $T = 0.1$ to study the spatial convergence rate of solving Euler equations. The prescribed non-polynomial solutions are:

$$\rho = \sin^{16} \pi(x + y - 2t) + 10^{-13}, \tag{53a}$$

$$\boldsymbol{u} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tag{53b}$$

$$e = \frac{10^{-13}}{(\gamma - 1)} \cdot \frac{1}{\sin^{16} \pi(x + y - 2t) + 10^{-13}}. \tag{53c}$$

The right-hand side functions and the initial and boundary conditions are defined by the manufactured solutions. The domain Ω is uniformly partitioned into square cells. For spatial discretization, we employ $\mathbb{P}^k$ ($k = 2$ and 3) DG methods, where the bases are constructed by orthonormal Legendre polynomials. The numerical integrations are evaluated by $(k + 1)^2$-point Gauss quadrature.

Let $\texttt{err}_{\Delta x}$ denote the error on a grid associated with mesh resolution Δx. The discrete L_h^2 and L_h^1 errors of density are computed by

$$L_h^2 \text{ error:} \qquad \|\rho_h^n - \rho(t^n)\|_{L_h^2}^2 = \Delta x^2 \sum_i \sum_\nu \omega_\nu \Big| \sum_j \rho_{ij}^n \varphi_{ij}(\boldsymbol{q}_\nu) - \rho(t^n, \boldsymbol{q}_\nu) \Big|^2, \tag{54a}$$

$$L_h^1 \text{ error:} \qquad \|\rho_h^n - \rho(t^n)\|_{L_h^1} = \Delta x^2 \sum_i \sum_\nu \omega_\nu \Big| \sum_j \rho_{ij}^n \varphi_{ij}(\boldsymbol{q}_\nu) - \rho(t^n, \boldsymbol{q}_\nu) \Big|, \tag{54b}$$

where φ_{ij} denotes the j-th basis on cell i, and ω_ν and $\boldsymbol{q}_\nu$ are quadrature weights and points. The errors for momentum and total energy are measured similarly. Then, the convergence rate is defined by $\ln(\texttt{err}_{\Delta x}/\texttt{err}_{\Delta x/2})/\ln(2)$.

To evaluate convergence rates, we successively refine mesh of $\Delta x = 1/25, 1/50, 1/100$ and employ a fourth-order RK method with fixed time step size $\Delta t = 5 \times 10^{-4}$. The time step size is small enough, which guarantees that the spatial error is dominated. We apply ℓ^2 limiter (35) when measuring the convergence rates with respect to L_h^2 norm; and apply ℓ^1 limiter (45) when measuring the convergence rates with respect to L_h^1 norm, see Table 1 and Table 2. The optimization-based cell average limiters are triggered and optimal convergence rates are obtained.

	$\mathbb{P}^2$ basis		$\mathbb{P}^3$ basis	
Δx	$\|\boldsymbol{U}_h^{N_T} - \boldsymbol{U}(T)\|_{L_h^2}$	rate	$\|\boldsymbol{U}_h^{N_T} - \boldsymbol{U}(T)\|_{L_h^2}$	rate
1/25	3.116×10^{-3}	—	6.514×10^{-3}	—
1/50	3.534×10^{-4}	3.141	4.846×10^{-5}	7.071
1/100	4.400×10^{-5}	3.006	1.302×10^{-6}	5.218

Table 1: The discrete L_h^2 errors and convergence rates for $\mathbb{P}^2$ and $\mathbb{P}^3$ spaces. The ℓ^2 limiter is solved by DYS method.

Example 5.4 (Sedov blast wave). This test describes a strong shock expanding from a point-source explosion in a uniform medium and involves a strong shock and low density, which makes it of great utility as a verification test for the robustness of a simulator for compressible Euler equations [30].

	$\mathbb{P}^2$ basis		$\mathbb{P}^3$ basis	
Δx	$\|\boldsymbol{U}_h^{N_T} - \boldsymbol{U}(T)\|_{L_h^1}$	rate	$\|\boldsymbol{U}_h^{N_T} - \boldsymbol{U}(T)\|_{L_h^1}$	rate
1/25	3.687×10^{-3}	—	5.040×10^{-3}	—
1/50	3.455×10^{-4}	3.416	3.710×10^{-5}	7.086
1/100	4.248×10^{-5}	3.024	1.177×10^{-6}	4.979

Table 2: The discrete L_h^1 errors and convergence rates for $\mathbb{P}^2$ and $\mathbb{P}^3$ spaces. The ℓ^1 limiter is solved by DRS method with parameter $\gamma = 10^{-3}$

.

We consider the following configuration on computational domain $\Omega = [0, 1.1]^2$ with the simulation end time $T = 1$. Uniformly partition the domain Ω into square cells with mesh resolution $\Delta x = 1.1/160$. The initials are defined as piecewise constants: the density $\rho^0 = 1$, the momentum $\boldsymbol{m}^0 = (0, 0)^{\mathrm{T}}$, and the total energy E^0 is set to 10^{-12} everywhere, except in the cell located at the lower left corner, where $0.244816/\Delta x^2$ is used. Reflective boundary conditions are imposed on the left and bottom boundaries, and outflow boundary conditions are imposed on the right and top boundaries.

Our optimization-based limiters can enforce invariant domain for SSP RKDG schemes with large time step sizes. Compared to the Zhang-Shu method for enforcing positivity [46], the advantage of integrating this optimization-based limiter with SSP RKDG methods is that the restrictive CFL from weak positivity in Zhang-Shu method [44] is no longer needed thus implementation is simplified. In this example, we employ a three-stage third-order SSP RK method in conjunction with $\mathbb{P}^2$ modal DG to solve the compressible Euler equation (28). Our time marching scheme is as follows:

$$\boldsymbol{U}^{(2)} = \boldsymbol{U}^n - \Delta t \nabla \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}^n), \tag{55a}$$

$$\boldsymbol{U}^{(3)} = \frac{3}{4}\boldsymbol{U}^n + \frac{1}{4}\Big[\boldsymbol{U}^{(2)} - \Delta t \nabla \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}^{(2)})\Big], \tag{55b}$$

$$\boldsymbol{U}^{n+1} = \frac{1}{3}\boldsymbol{U}^n + \frac{2}{3}\Big[\boldsymbol{U}^{(3)} - \Delta t \nabla \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}^{(3)})\Big]. \tag{55c}$$

We use the Lax–Friedrichs flux [46, 44]. The $\mathbb{P}^2$ basis functions are constructed using orthogonal Legendre polynomials. Numerical integrations are evaluated by the tensor product of 3-point Gauss quadrature.

When combining a DG method of k-th degree basis with a $(k+1)$-th order explicit RK scheme, the following expression provides an estimate of the effective CFL number for linear stability.

$$\mathrm{CFL} = \frac{1}{2k+1}, \quad \text{for } \mathbb{P}^k \text{ or } \mathbb{Q}^k \text{ scheme.} \tag{56}$$

Here, we use CFL = 0.2 for $\mathbb{P}^2$ scheme, which permits greater time step sizes than required for weak positivity, that is, negative cell averages are produced during the simulation. After each stage in (55), if out-of-bound cell averages appear, we apply our optimization-based postprocessing to modify the DG polynomial.

An ad-hoc strategy to improve efficiency and robustness of the optimization-based limiters is to avoid modifying regions of the computational domain which is not reached by the shock wave. Following the approach in [27, Remark 5], we define a subset of domain Ω as follows:

$$\Omega^{\lim} = \left\{ \text{cell } K_i : \text{ either } \overline{\boldsymbol{U}_h}|_{K_i} \notin G^{\varepsilon} \text{ or } \overline{E_h}|_{K_i} - \frac{1}{2}\frac{\|\overline{\boldsymbol{m}_h}|_{K_i}\|_2^2}{\overline{\rho_h}|_{K_i}} \geq 10^{-10} \right\}. \tag{57}$$

In this example, the optimization-based postprocessing is applied only to the cells in $\Omega^{\lim}$, thereby avoiding the redistribution of mass or energy into regions where the shock wave has not yet propagated to.

Figure 4 shows snapshot of the density at time $T = 1$. Our scheme preserves global conservation and the invariant domain. The shock location is correctly captured. The optimization cell average limiter is

triggered only at one time step during time evaluation, and the performance of the spitting methods solving the minimizations is shown in Figure 4. Besides the optimization based cell average limiter and Zhang-Shu limiter [46] for correcting quadrature point values, no additional limiters are used to remove oscillations.

Remark 3. *We can see that the ℓ^1 limiter is much more expensive to compute than the ℓ^2 limiter for 2D Sedov blast wave test. However, we will see that ℓ^1 limiter may not always be so expensive in the next example of Mach 2000 astrophysical jet.*

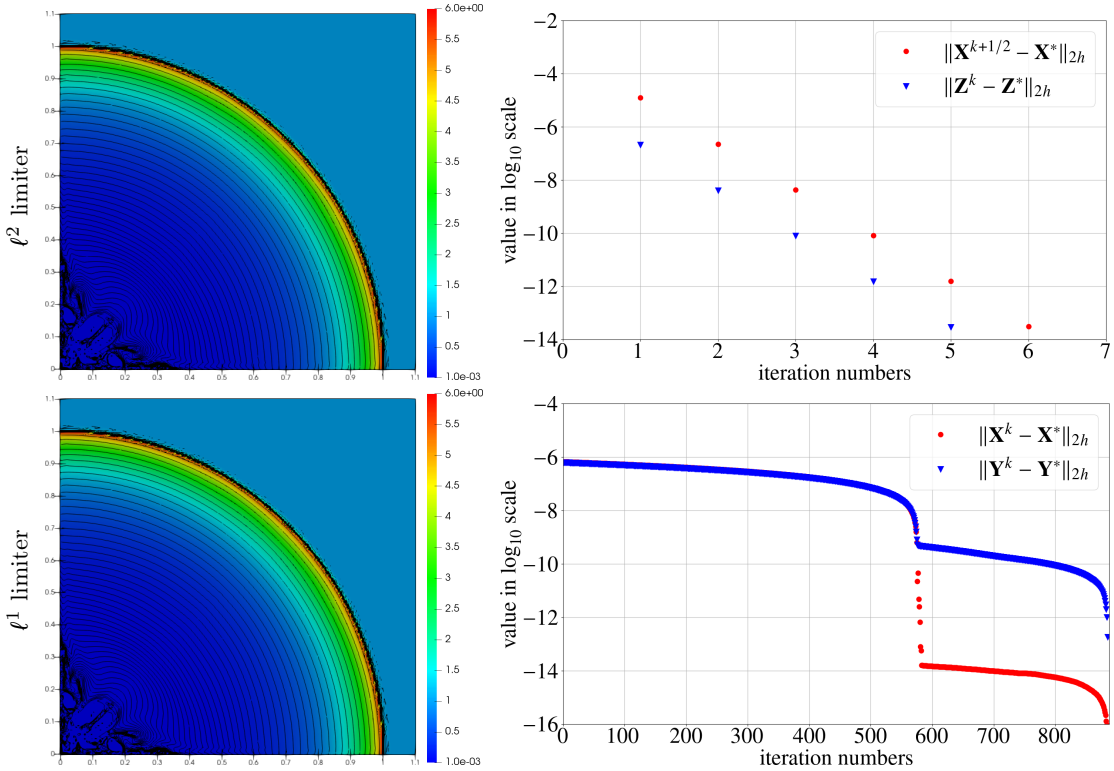

Figure 4: Sedov blast wave. Left: snapshot of density at $T = 1$. Plot of density: 50 exponentially distributed contour lines of density from 0.001 to 6. Right: number of DYS and DRS iterations in processing out-of-bound data of ℓ^2 and ℓ^1 model in the second RK stage at time step 74669, which is the only time step (during the whole time evolution) where the cell averages violate the invariant domain.

Example 5.5 (High Mach number astrophysical jet). The numerical replication of gas flows and shock wave patterns observed in Hubble Space Telescope images presents a demanding benchmark for computational methods, particularly when extreme Mach numbers are involved [14, 46]. We validate the robustness of our method by simulating Mach 2000 astrophysical jets governed by compressible Euler equations without radiative cooling.

Let us consider a high Mach number astrophysical jet configuration on the computational domain $\Omega = [0, 1] \times [-0.5, 0.5]$. We set the simulation end time $T = 0.001$. We take the ideal gas constant $\gamma = 5/3$. The initial density $\rho^0 = 0.5$, momentum $\boldsymbol{m}^0 = \mathbf{0}$, and pressure $p^0 = 0.4127$. The left edge of domain Ω is prescribed with the following inflow boundary condition:

$$[\rho, m_x, m_y, p]^{\mathrm{T}} = \begin{cases} [5,\ 4000,\ 0,\ 0.4127]^{\mathrm{T}} & \text{if } x = 0 \text{ and } |y| \le 0.05, \\ [0.5,\ 0,\ 0,\ 0.4127]^{\mathrm{T}} & \text{if } x = 0 \text{ and } |y| > 0.05, \end{cases} \tag{58}$$

while the top, right, and bottom edges of domain Ω are treated as outflow boundaries.

One advantage of our optimization-based invariant-domain-preserving approach is its applicability to numerical schemes without strong stability preserving (SSP) properties. In this example, we employ the classical four-stage fourth-order RK (RK4) method, which is not a SSP RK method, in conjunction with the $\mathbb{Q}^3$ DG spectral element method (SEM) to discretize the Euler equations (28).

$$\boldsymbol{U}^{(2)} = \boldsymbol{U}^n - \frac{\Delta t}{2}\nabla \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}^n), \tag{59a}$$

$$\boldsymbol{U}^{(3)} = \boldsymbol{U}^n - \frac{\Delta t}{2}\nabla \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}^{(2)}), \tag{59b}$$

$$\boldsymbol{U}^{(4)} = \boldsymbol{U}^n - \Delta t\nabla \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}^{(3)}), \tag{59c}$$

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n - \Delta t\Big[\frac{1}{6}\nabla \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}^n) + \frac{1}{3}\nabla \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}^{(2)}) + \frac{1}{3}\nabla \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}^{(3)}) + \frac{1}{6}\nabla \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}^{(4)})\Big], \tag{59d}$$

We uniformly partition domain Ω into square cells with mesh resolution $\Delta x = 1/640$. For spatial discretization, we employ the Lax–Friedrichs flux. The Lagrange bases of $\mathbb{Q}^3$ DG SEM are constructed via the tensor product of 4-point Gauss-Lobatto point and numerical integrations are evaluated by the tensor product of 4-point Gauss-Lobatto rule.

The fourth-order RK method is not SSP, and when combined with $\mathbb{Q}^3$ DG SEM, the resulting discretization lacks weak positivity (see [44] for its definition), i.e., negative cell averages may appear during the simulation. To enforce invariant domain without losing conservation and high order accuracy, we apply our optimization-based postprocessing to modify the DG polynomial with a CFL number $\frac{1}{7}$ as determined by (56) after each stage in (59) whenever any out-of-bound cell average is produced. The optimization-based limiters are applied to the whole computational domain for this test.

Figure 5 shows snapshots of the density field at time $T = 0.001$. Compared to the DYS method in ℓ^2 limiter, it requires more number of projections to admissible set for DRS method to solve the minimization problem in ℓ^1 limiter. When combined with Zhang-Shu limiter, both optimization-based cell average limiter (ℓ^2 and ℓ^1) robustly produce simulation results with correct shock location without other limiters for reducing oscillations. These computational results are consistent with high order DG methods stabilized by conventional non-optimization based limiters, see [31].

Remark 4. *For the astrophysical jet problem as shown in Figure 5, the computational cost of solving the ℓ^1 limiter* (45) *is still higher than the ℓ^2 limiter* (35) *whenever they are triggered. However, we can see that the cell average limiter is triggered less if using the ℓ^1 limiter, even though the ℓ^1 minimizer is not sparse as explained in previous sections. In other words, the total cost of the ℓ^1 limiter does not seem significantly higher than the ℓ^2 limiter in this test, and more importantly the ℓ^1 limiter is triggered less during the time evolution, which is more desirable.*

5. Concluding remarks

In this paper, we have developed effective splitting methods for implementing ℓ^1-norm and ℓ^2-norm optimization based limiters for enforcing the invariant domain in gas dynamics. The proposed postprocessing for high order DG schemes consists of first applying an optimization-based limiter to cell averages then by the Zhang-Shu limiter to modify quadrature point values. It is broadly applicable to DG methods with various time-marching schemes, e.g., non-SSP Runge-Kutta methods. Even though the optimization-based cell average limiters only preserve global conservation but not local conservation, the numerical tests suggest that it does not produce wrong shock locations for demanding problems such as blast waves and high speed flows. In practice, the ℓ^1-norm limiter produces almost the same result as the ℓ^2-norm limiter. In general, ℓ^2-norm limiter is cheaper to implement since it can be solved efficiently by the DYS method. On the other hand, for the astrophysical jet problem, we observe that the ℓ^1-norm limiter is triggered less during time evolution, thus the ℓ^1-norm limiter seems a better choice for this kind of problems.

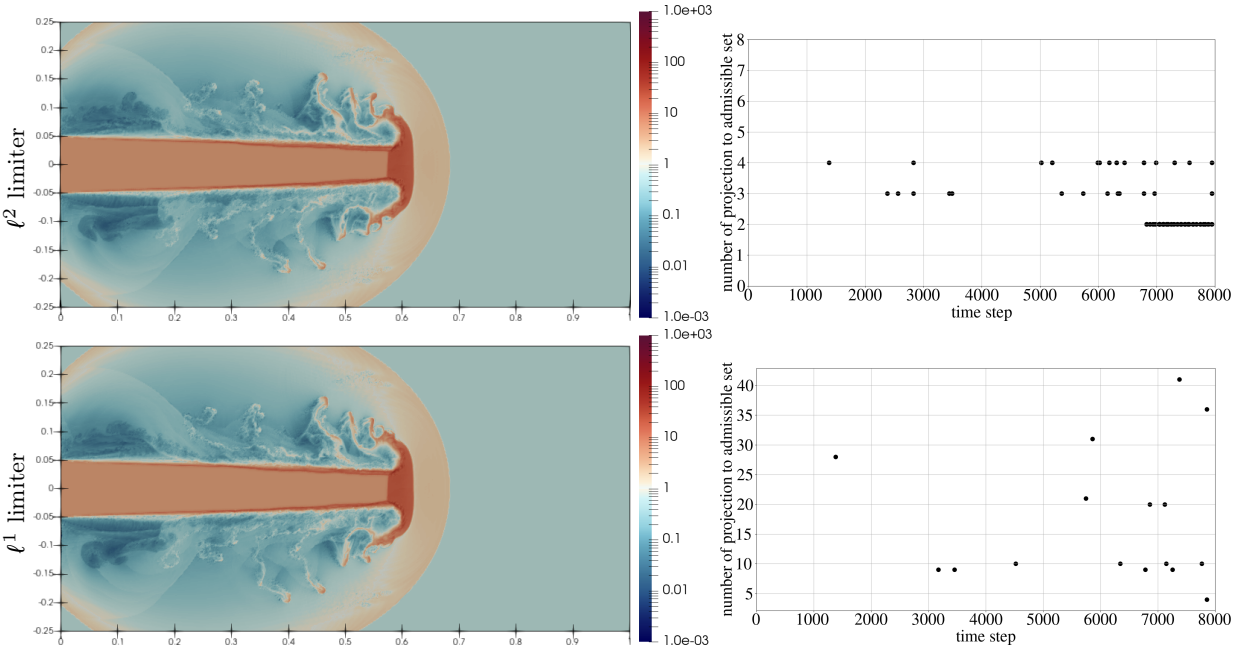

Figure 5: The Mach 2000 astrophysical jet. Top row: ℓ^2-norm limiter solved by with DYS. Bottom row: ℓ^1-norm limiter solved by DRS nested with DYS. Left: snapshot of density at $T = 0.001$ with logarithmic color scale. Right: number of projections to admissible set is number of computations of proximal operators to the indicator function $\iota_{\Lambda_2}(\mathbf{X})$ in the optimization solvers during each step of the fourth-order RK time marching, which is a fair way to compare computational cost between two optimization-based limiters.

CRediT authorship contribution statement

Chen Liu: Writing - Review & Editing, Writing – original draft, Methodology, Software, Visualization, Validation, Formal analysis, Investigation, Funding acquisition. **Dionysis Milesis:** Writing - Review & Editing, Formal analysis, Investigation. **Xiangxiong Zhang:** Writing - Review & Editing, Methodology, Formal analysis, Investigation, Conceptualization, Funding acquisition, Supervision, Project administration. **Chi-Wang Shu:** Writing - Review & Editing, Supervision, Methodology, Funding acquisition.

Data availability

Data will be made available on request.

Declaration of competing interest

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

Acknowledgements

C. Liu is supported by NSF DMS-2513106 and the Division of Research and Innovation (DRI) startup fund from University of Arkansas. C.-W. Shu is partially supported by NSF DMS-2309249. X. Zhang is partially supported by NSF DMS-2208518.

Appendix A. Computing proximal operator in ℓ^1 scalar model

Lemma 1. *For any given numbers $\gamma > 0$ and $a \in \mathbb{R}$, define $\mathrm{S}_\gamma(a) = \mathrm{sgn}(a)\max\{|a| - \gamma, 0\}$. Let $f(z) = \gamma|z| + \frac{1}{2}(z-a)^2$. We have: if $z > \mathrm{S}_\gamma(a)$, then $f(z)$ is strictly increasing; if $z < \mathrm{S}_\gamma(a)$, then $f(z)$ is strictly decreasing.*

Proof. If $z > \mathrm{S}_\gamma(a)$, we have the following scenarios:

- Case 1: $a \geq 0$ and $|a| > \gamma$.
  Then, we have $z > a - \gamma > 0 \;\Rightarrow\; f(z) = \gamma z + \frac{1}{2}(z-a)^2$, namely $f'(z) = z + \gamma - a > 0$.
- Case 2: $a \geq 0$ and $|a| \leq \gamma$.
  Then, we have $z > 0 \;\Rightarrow\; f(z) = \gamma z + \frac{1}{2}(z-a)^2$, namely $f'(z) = z + \gamma - a > 0$.
- Case 3: $a < 0$ and $|a| > \gamma$. Then, we have $z > a + \gamma$. We discuss this in two sub-cases
  - Sub-case 1: $z \geq 0 \;\Rightarrow\; f(z) = \gamma z + \frac{1}{2}(z-a)^2$, namely $f'(z) = z + \gamma - a > 0$.
  - Sub-case 2: $a + \gamma < z < 0 \Rightarrow\; f(z) = -\gamma z + \frac{1}{2}(z-a)^2$, namely $f'(z) = z - \gamma - a > 0$.
- Case 4: $a < 0$ and $|a| \leq \gamma$.
  Then, we have $z > 0 \;\Rightarrow\; f(z) = \gamma z + \frac{1}{2}(z-a)^2$, namely $f'(z) = z + \gamma - a > 0$.

Above all, when $z > \mathrm{S}_\gamma(a)$, the function $f(z)$ is increasing. Similarly, if $z < \mathrm{S}_\gamma(a)$, we have the following scenarios:

- Case 1: $a \geq 0$ and $|a| > \gamma$.
  Then, we have $z < a - \gamma$. We discuss this in two sub-cases
  - Sub-case 1: $z \leq 0 \;\Rightarrow\; f(z) = -\gamma z + \frac{1}{2}(z-a)^2$, namely $f'(z) = z - \gamma - a < 0$.
  - Sub-case 2: $0 < z < a - \gamma \Rightarrow\; f(z) = \gamma z + \frac{1}{2}(z-a)^2$, namely $f'(z) = z + \gamma - a < 0$.
- Case 2: $a \geq 0$ and $|a| \leq \gamma$.
  Then, we have $z \leq 0 \;\Rightarrow\; f(z) = -\gamma z + \frac{1}{2}(z-a)^2$, namely $f'(z) = z - \gamma - a < 0$.
- Case 3: $a < 0$ and $|a| > \gamma$.
  Then, we have $z < a + \gamma < 0 \;\Rightarrow\; f(z) = -\gamma z + \frac{1}{2}(z-a)^2$, namely $f'(z) = z - \gamma - a < 0$.
- Case 4: $a < 0$ and $|a| \leq \gamma$.
  Then, we have $z < 0 \;\Rightarrow\; f(z) = -\gamma z + \frac{1}{2}(z-a)^2$, namely $f'(z) = z - \gamma - a < 0$.

Above all, when $z < \mathrm{S}_\gamma(a)$, the function $f(z)$ is decreasing. □

Lemma 2. *For scalar conservation laws, recall the set $\Lambda_2 = \{\boldsymbol{x} \in \mathbb{R}^N : \; x_i \in [m, M], \; \forall i = 1, \cdots, N\}$ and in (23) the function $g(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{u}\|_1 + \iota_{\Lambda_2}(\boldsymbol{x})$. We have*

$$[\mathtt{prox}_g^\gamma(\boldsymbol{x})]_i = \max\{\min\{u_i + \mathrm{S}_\gamma(x_i - u_i), M\}, m\}, \tag{A.1}$$

where the subscript i denotes the i-th component in corresponding vector.

Proof. By definition

$$\begin{aligned}
\mathtt{prox}_g^\gamma(\boldsymbol{x}) &= \operatorname*{argmin}_{\boldsymbol{z}} \left\{\gamma\|\boldsymbol{z} - \boldsymbol{u}\|_1 + \gamma\iota_{\{\boldsymbol{z}:\, m \leq \boldsymbol{z} \leq M\}} + \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{x}\|_2^2\right\} \\
&= \operatorname*{argmin}_{\boldsymbol{z}} \left\{\gamma\|\boldsymbol{z} - \boldsymbol{u}\|_1 + \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{u} - (\boldsymbol{x} - \boldsymbol{u})\|_2^2 + \gamma\iota_{\{\boldsymbol{z}:\, m - \boldsymbol{u} \leq \boldsymbol{z} - \boldsymbol{u} \leq M - \boldsymbol{u}\}}\right\} \\
&= \boldsymbol{u} + \operatorname*{argmin}_{\boldsymbol{z}} \left\{\gamma\|\boldsymbol{z}\|_1 + \frac{1}{2}\|\boldsymbol{z} - (\boldsymbol{x} - \boldsymbol{u})\|_2^2 + \gamma\iota_{\{\boldsymbol{z}:\, m - \boldsymbol{u} \leq \boldsymbol{z} \leq M - \boldsymbol{u}\}}\right\} \\
&= \boldsymbol{u} + \operatorname*{argmin}_{\boldsymbol{z}} \sum_{i=1}^n \left(\gamma|z_i| + \frac{1}{2}|z_i - (x_i - u_i)|^2 + \gamma\iota_{\{z_i:\, m - u_i \leq z_i \leq M - u_i\}}\right).
\end{aligned} \tag{A.2}$$

Therefore, we have

$$[\mathtt{prox}_g^\gamma(\boldsymbol{x})]_i = u_i + \operatorname*{argmin}_{z_i} \left\{ \gamma|z_i| + \frac{1}{2}|z_i - (x_i - u_i)|^2 + \gamma \iota_{\{z_i:\, m-u_i \le z_i \le M-u_i\}} \right\}. \tag{A.3}$$

The following result is known, see [10, page 209 soft-thresholding operator].

$$\operatorname*{argmin}_{z_i} \left\{ \gamma|z_i| + \frac{1}{2}|z_i - (x_i - u_i)|^2 \right\} = \mathrm{S}_\gamma(x_i - u_i). \tag{A.4}$$

Therefore, we have

- If $\mathrm{S}_\gamma(x_i - u_i) \in [m - u_i, M - u_i]$, namely $u_i + \mathrm{S}_\gamma(x_i - u_i) \in [m, M]$, then $[\mathtt{prox}_g^\gamma(\boldsymbol{x})]_i = u_i + \mathrm{S}_\gamma(x_i - u_i)$.
- If $\mathrm{S}_\gamma(x_i - u_i) < m - u_i$, namely $u_i + \mathrm{S}_\gamma(x_i - u_i) < m$, then by Lemma 1, we know that the function $f(z) = \gamma|z| + \frac{1}{2}|z - (x_i - u_i)|^2$ is strictly increasing when $z > \mathrm{S}_\gamma(x_i - u_i)$. Notice $\mathrm{S}_\gamma(x_i - u_i) < m - u_i < M - u_i$, we have $[\mathtt{prox}_g^\gamma(\boldsymbol{x})]_i = u_i + (m - u_i) = m$.
- If $\mathrm{S}_\gamma(x_i - u_i) > M - u_i$, namely $u_i + \mathrm{S}_\gamma(x_i - u_i) > M$, then by Lemma 1, we know that the function $f(z) = \gamma|z| + \frac{1}{2}|z - (x_i - u_i)|^2$ is strictly decreasing when $z < \mathrm{S}_\gamma(x_i - u_i)$. Notice $m - u_i < M - u_i < \mathrm{S}_\gamma(x_i - u_i)$. We have $[\mathtt{prox}_g^\gamma(\boldsymbol{x})]_i = u_i + (M - u_i) = M$.

Above all, we conclude the proof. □

Appendix B. Projection to numerical admissible set in one dimension

Appendix B.1. Derivation from KKT conditions

Let $[u, v, w]^{\mathrm{T}}$ denote an out-of-bound cell average produced by a simulation in one-dimensional domain. For a small prescribed $\varepsilon > 0$, the set G^ε in (2) becomes

$$G^\varepsilon = \left\{ \boldsymbol{U} = [\rho, m, E]^{\mathrm{T}} :\ \rho \ge \varepsilon,\ \ \rho e(\boldsymbol{U}) = E - \frac{m^2}{2\rho} \ge \varepsilon \right\}. \tag{B.1}$$

Since the set G^ε is closed and convex, the projection is unique. To find the projection of $[u, v, w]^{\mathrm{T}}$ on set G^ε, we need to solve the following minimization problem: find $[\rho, m, E]^{\mathrm{T}}$ that

$$\begin{aligned} &\min_{\rho,m,E} \ \ \frac{1}{2}\left(|\rho - u|^2 + |m - v|^2 + |E - w|^2\right) \\ &\text{subject to:} \ \ \varepsilon - \rho \le 0 \quad \text{and} \quad \varepsilon - E + \frac{m^2}{2\rho} \le 0. \end{aligned} \tag{B.2}$$

The exact solution can be derived from the Lagrangian

$$L(\rho, m, E, \lambda, \mu) = \frac{1}{2}\left(|\rho - u|^2 + |m - v|^2 + |E - w|^2\right) + \lambda(\varepsilon - \rho) + \mu(\varepsilon - E + \frac{m^2}{2\rho}),$$

and its KKT conditions:

- Stationarity condition

$$\rho - u - \lambda - \mu\frac{m^2}{2\rho^2} = 0, \tag{B.3a}$$

$$m - v + \mu\frac{m}{\rho} = 0, \tag{B.3b}$$

$$E - w - \mu = 0. \tag{B.3c}$$

- Complementary slackness

$$\lambda(\varepsilon - \rho) = 0, \tag{B.3d}$$

$$\mu\Big(\varepsilon - E + \frac{m^2}{2\rho}\Big) = 0. \tag{B.3e}$$

- Primal feasibility

$$\varepsilon - \rho \le 0, \tag{B.3f}$$

$$\varepsilon - E + \frac{m^2}{2\rho} \le 0. \tag{B.3g}$$

- Dual feasibility

$$\lambda \ge 0, \tag{B.3h}$$

$$\mu \ge 0. \tag{B.3i}$$

From the dual feasibility, depending on whether λ and μ are zero or positive, there are four cases:

Case 1 ($\mu = 0$ and $\lambda > 0$). When μ is zero, the (B.3a)-(B.3c) gives $\rho = u + \lambda$, $m = v$, and $E = w$. When $\lambda > 0$, (B.3d) implies $\rho = \varepsilon$. From (B.3a), we get $\lambda = \varepsilon - u$. Thus, if $[\varepsilon, v, w]^{\mathrm{T}}$ is a solution of (B.3), the constraints $\varepsilon > u$ and (B.3g) need to be satisfied.

Case 2 ($\mu = 0$ and $\lambda = 0$). When both μ and λ are zero, the (B.3a)-(B.3c) gives: $\rho = u$, $m = v$, and $E = w$. Thus, if $[u, v, w]^{\mathrm{T}}$ is a solution of (B.3), the given point should belong to the set G^{ε}, i.e., the projection point is itself.

Case 3 ($\mu > 0$ and $\lambda > 0$). When both μ and λ are positive, the (B.3d) gives $\rho = \varepsilon$ and (B.3e) becomes

$$E = \frac{1}{2\varepsilon}m^2 + \varepsilon. \tag{B.4}$$

If $v = 0$, by (B.3b), we have $m = 0$, which further implies $E = \varepsilon$. From (B.3c), we get $\mu = \varepsilon - w$ and from (B.3a), we get $\lambda = \varepsilon - u$. Thus, if $[\varepsilon, 0, \varepsilon]^{\mathrm{T}}$ is a solution of (B.3), the constraints $\varepsilon > u$ and $\varepsilon > w$ need to be satisfied.

If $v \neq 0$, then $m \neq 0$. Otherwise, (B.3b) gives $v = 0$, which leads to a contradiction. Substituting $\rho = \varepsilon$ into (B.3b) to solve μ, and further substituting the result into (B.3a), we have

$$\mu = \varepsilon\Big(\frac{v}{m} - 1\Big) \quad \text{and} \quad \lambda = \varepsilon - u - \frac{1}{2\varepsilon}m(v - m). \tag{B.5}$$

Combine (B.3c) with (B.4) and (B.5), we obtain

$$\frac{1}{2\varepsilon}m^2 + \varepsilon = E = \mu + w = \varepsilon\Big(\frac{v}{m} - 1\Big) + w. \tag{B.6}$$

Multiplying $2\varepsilon m$ on both sides above, we get a cubic equation with respect to m as follows.

$$m^3 + (4\varepsilon^2 - 2\varepsilon w)m - 2\varepsilon^2 v = 0. \tag{B.7}$$

After getting all the real roots of using the method in Appendix D, we compute E from (B.4), and μ, λ from (B.5). Thus, if the result here is a solution of (B.3), the constraints $\mu > 0$ and $\lambda > 0$ for (B.5) need to be satisfied.

Case 4 ($\mu > 0$ and $\lambda = 0$). When μ is positive and λ is zero, the KKT condition (B.3) simplify to $\rho \geq \varepsilon$ and

$$\rho - u - \mu \frac{m^2}{2\rho^2} = 0, \tag{B.8a}$$

$$m - v + \mu \frac{m}{\rho} = 0, \tag{B.8b}$$

$$E - w - \mu = 0, \tag{B.8c}$$

$$\varepsilon - E + \frac{m^2}{2\rho} = 0. \tag{B.8d}$$

If $v = 0$, by (B.8b), we have $m = 0$. By (B.8a) and (B.8d), we get $\rho = u$ and $E = \varepsilon$. From (B.8c), we get $\mu = \varepsilon - w$. Thus, if $[u, 0, \varepsilon]^{\mathrm{T}}$ is a solution of (B.3), the constraints $\varepsilon > w$ and $\varepsilon \leq u$ need to be satisfied.

If $v \neq 0$, then $m \neq 0$. Otherwise, (B.8b) gives $v = 0$, which leads to a contradiction. Substituting into (B.8b) and (B.8a), we have

$$\rho - u - \frac{m}{2\rho}\Big(\mu \frac{m}{\rho}\Big) = 0 \quad \Rightarrow \quad \frac{m}{\rho}(v - m) = 2(\rho - u), \tag{B.9}$$

which is a quadratic equation with respect to m, namely $m^2 - vm + 2\rho^2 - 2u\rho = 0$. Thus, let $\Delta = -8\rho^2 + 8u\rho + v^2$ denote the discriminant. We have

$$m = \frac{v}{2} - \frac{1}{2}\sqrt{\Delta} \quad \text{or} \quad m = \frac{v}{2} + \frac{1}{2}\sqrt{\Delta}. \tag{B.10}$$

From (B.8c) and (B.8d), we get

$$E = \varepsilon + \frac{m^2}{2\rho}, \tag{B.11}$$

$$\mu = \varepsilon + \frac{m^2}{2\rho} - w. \tag{B.12}$$

Therefore, once we compute ρ, the (B.8) is solved. Substitute (B.12) into (B.8b). After multiplying $2\rho^2$, we end up with a quadratic equation with respect to ρ, as follows. Notice that $v \neq 0$ implies $m \neq v$.

$$2(m - v)\rho^2 + 2(\varepsilon - w)m\rho + m^3 = 0. \tag{B.13}$$

To solve ρ, let us first substitute $m = \frac{v}{2} - \frac{1}{2}\sqrt{\Delta}$ into above. The other root in (B.10) can be addressed in the same manner. We get

$$\sqrt{\Delta}(3v^2 + \Delta + 8(\varepsilon - w)\rho + 8\rho^2) = v^3 + 3v\Delta + 8(\varepsilon - w)v\rho - 8v\rho^2. \tag{B.14}$$

Recall the discriminant $\Delta = -8\rho^2 + 8u\rho + v^2$. Square both sides above and after some simplification, we obtain

$$4\big(2v^2 + (\varepsilon + u - w)^2\big)\rho^2 - 4u\big(2v^2 + (\varepsilon + u - w)^2\big)\rho + 2uv^2(w - \varepsilon) - v^4 = 0. \tag{B.15}$$

The two roots of above quadratic equation are:

$$\rho = \frac{u}{2} + \frac{1}{2}\frac{\sqrt{u^2(2v^2 + (\varepsilon + u - w)^2) - 2uv^2(w - \varepsilon) + v^4}}{\sqrt{2v^2 + (\varepsilon + u - w)^2}}, \tag{B.16a}$$

$$\text{or} \quad \rho = \frac{u}{2} - \frac{1}{2}\frac{\sqrt{u^2(2v^2 + (\varepsilon + u - w)^2) - 2uv^2(w - \varepsilon) + v^4}}{\sqrt{2v^2 + (\varepsilon + u - w)^2}}. \tag{B.16b}$$

For the second root in (B.10), substitute $m = \frac{v}{2} + \frac{1}{2}\sqrt{\Delta}$ into (B.13). After simplification, we end up with exactly the same equation as (B.15), namely the expressions in (B.16) are still the roots. Thus, if the result here is a solution of (B.3), the constraints $\mu > 0$ for (B.12) and $\rho \geq \varepsilon$ for (B.16) need to be satisfied.

Appendix B.2. Algorithm of projection onto the invariant domain set

Given $[x, y, z]^{\mathrm{T}} \notin G^{\varepsilon}$, find the projection of $[x, y, z]^{\mathrm{T}}$ on set G^{ε}.

Step 1. Compute candidate points from KKT conditions.

- The point $[\rho, m, E]^{\mathrm{T}} = [\varepsilon, y, z]^{\mathrm{T}}$ is a candidate, if $x < \varepsilon$ and $z - \frac{y^2}{2\varepsilon} \geq \varepsilon$.
- The point $[\rho, m, E]^{\mathrm{T}} = [\varepsilon, 0, \varepsilon]^{\mathrm{T}}$ is a candidate, if $x < \varepsilon$, $y = 0$, and $z < \varepsilon$.
- If $y \neq 0$, solve cubic equation $m^3 + (4\varepsilon^2 - 2\varepsilon z)m - 2\varepsilon^2 y = 0$ to obtain all real roots. Examine each real root individually. Let m_r denote a real root and let $E_r = \frac{m_r^2}{2\varepsilon} + \varepsilon$. The point $[\rho, m, E]^{\mathrm{T}} = [\varepsilon, m_r, E_r]^{\mathrm{T}}$ is a candidate, if $\frac{y}{m_r} > 1$ and $2\varepsilon x + m_r(y - m_r) < 2\varepsilon^2$.
- The point $[\rho, m, E]^{\mathrm{T}} = [x, 0, \varepsilon]^{\mathrm{T}}$ is a candidate, if $x \geq \varepsilon$, $y = 0$, and $z < \varepsilon$.
- Compute ρ_1 and ρ_2 using the following formulas only if they are real.

$$\rho_1 = \frac{x}{2} + \frac{1}{2}\frac{\sqrt{x^2(2y^2 + (\varepsilon + x - z)^2) - 2xy^2(z - \varepsilon) + y^4}}{\sqrt{2y^2 + (\varepsilon + x - z)^2}},$$
$$\rho_2 = \frac{x}{2} - \frac{1}{2}\frac{\sqrt{x^2(2y^2 + (\varepsilon + x - z)^2) - 2xy^2(z - \varepsilon) + y^4}}{\sqrt{2y^2 + (\varepsilon + x - z)^2}}.$$

For real values of ρ_1 and ρ_2, compute m_1 and m_2 using the following formulas only if they are real.

$$m_1(\rho) = \frac{1}{2}y - \frac{1}{2}\sqrt{-8\rho^2 + 8x\rho + y^2} \quad \text{and} \quad m_2(\rho) = \frac{1}{2}y + \frac{1}{2}\sqrt{-8\rho^2 + 8x\rho + y^2}.$$

Then, for real point:

- The point $[\rho, m, E]^{\mathrm{T}} = [\rho_1, m_1(\rho_1), \varepsilon + \frac{m_1(\rho_1)^2}{2\rho_1}]^{\mathrm{T}}$ is a candidate, if $\rho_1 \geq \varepsilon$ and $\varepsilon + \frac{m_1(\rho_1)^2}{2\rho_1} > z$.
- The point $[\rho, m, E]^{\mathrm{T}} = [\rho_2, m_1(\rho_2), \varepsilon + \frac{m_1(\rho_2)^2}{2\rho_2}]^{\mathrm{T}}$ is a candidate, if $\rho_2 \geq \varepsilon$ and $\varepsilon + \frac{m_1(\rho_2)^2}{2\rho_2} > z$.
- The point $[\rho, m, E]^{\mathrm{T}} = [\rho_1, m_2(\rho_1), \varepsilon + \frac{m_2(\rho_1)^2}{2\rho_1}]^{\mathrm{T}}$ is a candidate, if $\rho_1 \geq \varepsilon$ and $\varepsilon + \frac{m_2(\rho_1)^2}{2\rho_1} > z$.
- The point $[\rho, m, E]^{\mathrm{T}} = [\rho_2, m_2(\rho_2), \varepsilon + \frac{m_2(\rho_2)^2}{2\rho_2}]^{\mathrm{T}}$ is a candidate, if $\rho_2 \geq \varepsilon$ and $\varepsilon + \frac{m_2(\rho_2)^2}{2\rho_2} > z$.

Step 2. Substitute all candidate points into the objective function

$$|\rho - x|^2 + |m - y|^2 + |E - z|^2.$$

Pick the point $[\rho, m, E]^{\mathrm{T}}$ from the candidate points such that minimizes the value of the objective function, which gives the projection point.

Appendix C. Projection to numerical admissible set in two dimensions

Appendix C.1. Derivation from KKT conditions

Let $[u, v_1, v_2, w]^{\mathrm{T}}$ denote an out-of-bound cell average produced by a simulation in two-dimensional domain. For a small prescribed $\varepsilon > 0$, the numerical admissible set G^{ε} in (2) becomes

$$G^{\varepsilon} = \left\{ \boldsymbol{U} = [\rho, m_1, m_2, E]^{\mathrm{T}} : \ \rho \geq \varepsilon, \ \ \rho e(\boldsymbol{U}) = E - \frac{m_1^2 + m_2^2}{2\rho} \geq \varepsilon \right\}. \tag{C.1}$$

The projection point on the closed convex set G^ε is unique. To find the projection of $[u, v_1, v_2, w]^{\mathrm{T}}$, we need to solve the following minimization problem: find $[\rho, m_1, m_2, E]^{\mathrm{T}}$ that

$$\begin{aligned} \min_{\rho, m_1, m_2, E} \quad & \frac{1}{2}\left(|\rho - u|^2 + |m_1 - v_1|^2 + |m_2 - v_2|^2 + |E - w|^2\right) \\ \text{subject to:} \quad & \varepsilon - \rho \le 0 \quad \text{and} \quad \varepsilon - E + \frac{m_1^2 + m_2^2}{2\rho} \le 0. \end{aligned} \tag{C.2}$$

We derive an exact solution using the following KKT conditions:

- Stationarity condition

$$\rho - u - \lambda - \mu \frac{m_1^2 + m_2^2}{2\rho^2} = 0, \tag{C.3a}$$
$$m_1 - v_1 + \mu \frac{m_1}{\rho} = 0, \tag{C.3b}$$
$$m_2 - v_2 + \mu \frac{m_2}{\rho} = 0, \tag{C.3c}$$
$$E - w - \mu = 0. \tag{C.3d}$$

- Complementary slackness

$$\lambda(\varepsilon - \rho) = 0, \tag{C.3e}$$
$$\mu\left(\varepsilon - E + \frac{m_1^2 + m_2^2}{2\rho}\right) = 0. \tag{C.3f}$$

- Primal feasibility

$$\varepsilon - \rho \le 0, \tag{C.3g}$$
$$\varepsilon - E + \frac{m_1^2 + m_2^2}{2\rho} \le 0. \tag{C.3h}$$

- Dual feasibility

$$\lambda \ge 0, \tag{C.3i}$$
$$\mu \ge 0. \tag{C.3j}$$

From the dual feasibility, depending on λ and μ are zero or positive, we have four different scenarios. Let us discuss case by case.

Case 1 ($\mu = 0$ and $\lambda > 0$). When μ is zero, the (C.3a)-(C.3d) gives $\rho = u + \lambda$, $m_1 = v_1$, $m_2 = v_2$, and $E = w$. When $\lambda > 0$, the (C.3e) implies $\rho = \varepsilon$. From (C.3a), we get $\lambda = \varepsilon - u$. Thus, if $[\varepsilon, v_1, v_2, w]^{\mathrm{T}}$ is a solution of (C.3), the constraints $\varepsilon > u$ and (C.3h) need to be satisfied.

Case 2 ($\mu = 0$ and $\lambda = 0$). When both μ and λ are zero, the (C.3a)-(C.3d) gives $\rho = u$, $m_1 = v_1$, $m_2 = v_2$, and $E = w$. Thus, if $[u, v_1, v_2, w]^{\mathrm{T}}$ is a solution of (C.3), the given point already belongs to the set G^ε, i.e., the projection point is itself.

Case 3 ($\mu > 0$ and $\lambda > 0$). When both μ and λ are positive, the (C.3e) gives $\rho = \varepsilon$ and (C.3f) becomes

$$E = \frac{1}{2\varepsilon}(m_1^2 + m_2^2) + \varepsilon. \tag{C.4}$$

If $v_1 = 0$ and $v_2 = 0$, by (C.3b) and (C.3c), we have $m_1 = 0$ and $m_2 = 0$, which further implies $E = \varepsilon$. From (C.3d), we get $\mu = \varepsilon - w$ and from (C.3a), we get $\lambda = \varepsilon - u$. Thus, if $[\varepsilon, 0, 0, \varepsilon]^{\mathrm{T}}$ is a solution of (C.3), the constraints $\varepsilon > u$ and $\varepsilon > w$ need to be satisfied.

If $v_1 = 0$ and $v_2 \neq 0$, then (C.3b) gives $m_1 = 0$ and (C.3c) gives $m_2 \neq 0$. Therefore, (C.3a)-(C.3d) become

$$\rho - u - \lambda - \mu\frac{m_2^2}{2\rho^2} = 0, \tag{C.5a}$$

$$m_2 - v_2 + \mu\frac{m_2}{\rho} = 0, \tag{C.5b}$$

$$E - w - \mu = 0. \tag{C.5c}$$

Furthermore, the KKT condition (C.3) degenerates to the associated one-dimensional case, which can be solved by using the method in Section Appendix B Case 3.

If $v_1 \neq 0$, multiply (C.3b) by v_2 and multiply (C.3c) by v_1, respectively. We have

$$\left.\begin{aligned}\Big(1 + \frac{\mu}{\rho}\Big)m_1 v_2 = v_1 v_2\\ \Big(1 + \frac{\mu}{\rho}\Big)m_2 v_1 = v_1 v_2\end{aligned}\right\} \quad \Rightarrow \quad m_1 v_2 = m_2 v_1 \quad \Rightarrow \quad m_2 = \frac{v_2}{v_1}m_1. \tag{C.6}$$

When $v_1 \neq 0$, it implies $m_1 \neq 0$. Otherwise, (C.3b) gives $v_1 = 0$, which leads to a contradiction. Let $a = 1 + v_2^2/v_1^2$. Substituting $\rho = \varepsilon$ into (C.3b) to solve μ, and further substituting the result into (C.3a), we have

$$\mu = \varepsilon\Big(\frac{v_1}{m_1} - 1\Big) \quad \text{and} \quad \lambda = \varepsilon - u - \frac{a}{2\varepsilon}m_1(v_1 - m_1). \tag{C.7}$$

By substituting (C.6) into (C.4) and combining the result with (C.3d) and (C.7), we obtain

$$\frac{a}{2\varepsilon}m_1^2 + \varepsilon = E = \mu + w = \varepsilon\Big(\frac{v_1}{m_1} - 1\Big) + w. \tag{C.8}$$

Multiplying $2\varepsilon m_1$ on both sides above, we get a cubic equation with respect to m_1 as follows.

$$a m_1^3 + (4\varepsilon^2 - 2\varepsilon w)m_1 - 2\varepsilon^2 v_1 = 0.$$

After getting all the real roots of using the method in Appendix D, we compute m_2 form (C.6), E from (C.8), and μ, λ from (C.7). Thus, if the result here is a solution of (C.3), the constraints $\mu > 0$ and $\lambda > 0$ for (C.7) need to be satisfied.

Case 4 ($\mu > 0$ and $\lambda = 0$). When μ is positive and λ is zero, the KKT condition (C.3) simplify to $\rho \geq \varepsilon$ and

$$\rho - u - \mu\frac{m_1^2 + m_2^2}{2\rho^2} = 0, \tag{C.9a}$$

$$m_1 - v_1 + \mu\frac{m_1}{\rho} = 0, \tag{C.9b}$$

$$m_2 - v_2 + \mu\frac{m_2}{\rho} = 0, \tag{C.9c}$$

$$E - w - \mu = 0, \tag{C.9d}$$

$$\varepsilon - E + \frac{m_1^2 + m_2^2}{2\rho} = 0. \tag{C.9e}$$

If $v_1 = 0$ and $v_2 = 0$, we have $m_1 = 0$ and $m_2 = 0$. By (C.9a) and (C.9e), we get $\rho = u$ and $E = \varepsilon$. From (C.9d), we get $\mu = \varepsilon - w$. Thus, if $[u, 0, 0, \varepsilon]^{\mathrm{T}}$ is a solution of (C.3), the constraints $\varepsilon > w$ and $\varepsilon \le u$ need to be satisfied.

If $v_1 = 0$ and $v_2 \neq 0$, then (C.9b) gives $m_1 = 0$ and (C.9c) gives $m_2 \neq 0$. Thus, (C.9a)-(C.9e) become

$$\rho - u - \mu\frac{m_2^2}{2\rho^2} = 0, \tag{C.10a}$$

$$m_2 - v_2 + \mu\frac{m_2}{\rho} = 0, \tag{C.10b}$$

$$E - w - \mu = 0, \tag{C.10c}$$

$$\varepsilon - E + \frac{m_2^2}{2\rho} = 0, \tag{C.10d}$$

which can be solved by using the method in Section Appendix B Case 4.

If $v_1 \neq 0$, recall $a = 1 + v_2^2/v_1^2$ and $m_2 v_1 = m_1 v_2$, by eliminating m_2 in (C.9a) and substituting (C.9b) into (C.9a), we have

$$\rho - u - a\frac{m_1}{2\rho}\Big(\mu\frac{m_1}{\rho}\Big) = 0 \quad \Rightarrow \quad a\frac{m_1}{\rho}(v_1 - m_1) = 2(\rho - u),$$

which is a quadratic equation with respect to m_1, namely $am_1^2 - av_1m_1 + 2\rho^2 - 2u\rho = 0$. Thus, let $\Delta = -8a\rho^2 + 8au\rho + a^2v_1^2$ denote the discriminant. We have

$$m_1 = \frac{v_1}{2} - \frac{1}{2a}\sqrt{\Delta} \quad \text{or} \quad m_1 = \frac{v_1}{2} + \frac{1}{2a}\sqrt{\Delta}. \tag{C.11}$$

Similarly, using $m_2 = m_1 v_2/v_1$ to eliminate m_2 in (C.9e) and adding (C.10c) and (C.9e) together, we have

$$E = \varepsilon + a\frac{m_1^2}{2\rho}, \tag{C.12}$$

$$\mu = \varepsilon + a\frac{m_1^2}{2\rho} - w. \tag{C.13}$$

Therefore, once we compute ρ, the (C.9) is solved. Substitute (C.13) into (C.9b). After multiplying $2\rho^2$, we end up with a quadratic equation with respect to ρ, as follows. Notice that $v_1 \neq 0$ implies $m_1 \neq v_1$.

$$2(m_1 - v_1)\rho^2 + 2(\varepsilon - w)m_1\rho + am_1^3 = 0. \tag{C.14}$$

To solve ρ, let us first substitute $m_1 = \frac{v_1}{2} - \frac{1}{2a}\sqrt{\Delta}$ into above. The other root in (C.11) can be addressed in the same manner. We get

$$\sqrt{\Delta}\Big(3v_1^2 + \frac{\Delta}{a^2} + 8(\varepsilon - w)\frac{\rho}{a} + \frac{8}{a}\rho^2\Big) = av_1^3 + \frac{3}{a}v_1\Delta + 8(\varepsilon - w)v_1\rho - 8v_1\rho^2. \tag{C.15}$$

Recall the discriminant $\Delta = -8a\rho^2 + 8au\rho + a^2v_1^2$. Square both sides above and after some simplification, we obtain

$$4\Big(2v_1^2 + \frac{1}{a}(\varepsilon + u - w)^2\Big)\rho^2 - 4u\Big(2v_1^2 + \frac{1}{a}(\varepsilon + u - w)^2\Big)\rho + 2uv_1^2(w - \varepsilon) - av_1^4 = 0. \tag{C.16}$$

The two roots of above quadratic equation are:

$$\rho = \frac{u}{2} + \frac{1}{2}\frac{\sqrt{u^2(2v_1^2 + \frac{1}{a}(\varepsilon + u - w)^2) - 2uv_1^2(w - \varepsilon) + av_1^4}}{\sqrt{2v_1^2 + \frac{1}{a}(\varepsilon + u - w)^2}}, \tag{C.17a}$$

$$\text{or} \quad \rho = \frac{u}{2} - \frac{1}{2}\frac{\sqrt{u^2(2v_1^2 + \frac{1}{a}(\varepsilon + u - w)^2) - 2uv_1^2(w - \varepsilon) + av_1^4}}{\sqrt{2v_1^2 + \frac{1}{a}(\varepsilon + u - w)^2}}. \tag{C.17b}$$

For the second root in (C.11), substitute $m_1 = \frac{v_1}{2} + \frac{1}{2a}\sqrt{\Delta}$ into (C.14). After simplification, we end up with exactly the same equation as (C.16), namely the expressions in (C.17) are still the roots. Thus, if the result here is a solution of (C.3), the constraints $\mu > 0$ for (C.13) and $\rho \geq \varepsilon$ for (C.17) need to be satisfied.

Finally, among all the points listed above, we pick the point $[\rho, m_1, m_2, E]^{\mathrm{T}}$ which gives the smallest value of the objective function, and it is the projection point.

Appendix C.2. Algorithm of projection onto the invariant domain set

Given $[x, y_1, y_2, z]^{\mathrm{T}} \notin G^{\varepsilon}$, find the projection $[\rho, m_1, m_2, E]^{\mathrm{T}}$ on set G^{ε}.

Step 1. Compute candidate points from KKT conditions.

- The point $[\rho, m_1, m_2, E]^{\mathrm{T}} = [\varepsilon, y_1, y_2, z]^{\mathrm{T}}$ is a candidate, if $x < \varepsilon$ and $z - \frac{1}{2\varepsilon}(y_1^2 + y_2^2) \geq \varepsilon$.
- The point $[\rho, m_1, m_2, E]^{\mathrm{T}} = [\varepsilon, 0, 0, \varepsilon]^{\mathrm{T}}$ is a candidate, if $x < \varepsilon$, $y_1 = 0$, $y_2 = 0$, and $z < \varepsilon$.
- If $y_1, y_2 \neq 0$, solve the cubic equation $am_1^3 + (4\varepsilon^2 - 2\varepsilon z)m_1 - 2\varepsilon^2 y_1 = 0$ with $a = 1 + y_2^2/y_1^2$ to obtain all real roots. Examine each real root individually. Let m_{1r} denote a real root. Then the point

$$[\rho, m_1, m_2, E]^{\mathrm{T}} = [\varepsilon, m_{1r}, m_{2r}, E_r]^{\mathrm{T}}, \quad \text{where} \quad m_{2r} = \frac{y_2}{y_1} m_{1r} \quad \text{and} \quad E_r = \frac{a}{2\epsilon} m_{1r}^2 + \varepsilon,$$

  is a candidate, if $\frac{y_1}{m_{1r}} > 1$ and $2\varepsilon x + a m_{1r}(y_1 - m_{1r}) < 2\varepsilon^2$.
- The point $[\rho, m_1, m_2, E]^{\mathrm{T}} = [x, 0, 0, \varepsilon]^{\mathrm{T}}$ is a candidate, if $x \geq \varepsilon$, $y_1 = 0$, $y_2 = 0$, and $z < \varepsilon$.
- Compute ρ_1 and ρ_2 using the following formulas only if they are real.

$$\rho_1 = \frac{x}{2} + \frac{1}{2}\frac{\sqrt{x^2(2y_1^2 + \frac{1}{a}(\varepsilon + x - z)^2) - 2xy_1^2(z - \varepsilon) + ay_1^4}}{\sqrt{2y_1^2 + \frac{1}{a}(\varepsilon + x - z)^2}},$$

$$\rho_2 = \frac{x}{2} - \frac{1}{2}\frac{\sqrt{x^2(2y_1^2 + \frac{1}{a}(\varepsilon + x - z)^2) - 2xy_1^2(z - \varepsilon) + ay_1^4}}{\sqrt{2y_1^2 + \frac{1}{a}(\varepsilon + x - z)^2}}.$$

  For real values of ρ_1 and ρ_2, compute $m_{1\alpha}$ and $m_{1\beta}$ using the following formulas only if they are real.

$$m_{1\alpha}(\rho) = \frac{1}{2}y_1 - \frac{1}{2a}\sqrt{-8a\rho^2 + 8ax\rho + a^2y_1^2} \quad \text{and} \quad m_{1\beta}(\rho) = \frac{1}{2}y_1 + \frac{1}{2a}\sqrt{-8a\rho^2 + 8ax\rho + a^2y_1^2}.$$

  Then, for real points:

  - The point $[\rho, m_1, m_2, E]^{\mathrm{T}} = [\rho_1, m_{1\alpha}(\rho_1), \frac{y_2}{y_1}m_{1\alpha}(\rho_1), \varepsilon + \frac{m_{1\alpha}(\rho_1)^2}{2\rho_1}]^{\mathrm{T}}$ is a candidate, if $\rho_1 \geq \varepsilon$ and $\varepsilon + \frac{m_{1\alpha}(\rho_1)^2}{2\rho_1} > z$.
  - The point $[\rho, m_1, m_2, E]^{\mathrm{T}} = [\rho_2, m_{1\alpha}(\rho_2), \frac{y_2}{y_1}m_{1\alpha}(\rho_2), \varepsilon + \frac{m_{1\alpha}(\rho_2)^2}{2\rho_2}]^{\mathrm{T}}$ is a candidate, if $\rho_2 \geq \varepsilon$ and $\varepsilon + \frac{m_{1\alpha}(\rho_2)^2}{2\rho_2} > z$.
  - The point $[\rho, m_1, m_2, E]^{\mathrm{T}} = [\rho_1, m_{1\beta}(\rho_1), \frac{y_2}{y_1}m_{1\beta}(\rho_1), \varepsilon + \frac{m_{1\beta}(\rho_1)^2}{2\rho_1}]^{\mathrm{T}}$ is a candidate, if $\rho_1 \geq \varepsilon$ and $\varepsilon + \frac{m_{1\beta}(\rho_1)^2}{2\rho_1} > z$.
  - The point $[\rho, m_1, m_2, E]^{\mathrm{T}} = [\rho_2, m_{1\beta}(\rho_2), \frac{y_2}{y_1}m_{1\beta}(\rho_2), \varepsilon + \frac{m_{1\beta}(\rho_2)^2}{2\rho_2}]^{\mathrm{T}}$ is a candidate, if $\rho_2 \geq \varepsilon$ and $\varepsilon + \frac{m_{1\beta}(\rho_2)^2}{2\rho_2} > z$.

Step 2. Substitute all candidate points into the objective function

$$|\rho - x|^2 + |m_1 - y_1|^2 + |m_2 - y_2|^2 + |E - z|^2.$$

Pick the point $[\rho, m_1, m_2, E]^{\mathrm{T}}$ from the candidate points such that minimizes the value of the objective function, which gives the projection point.

Appendix D. Real roots of depressed cubic equation

We list the following formulae to compute all the real roots of a depressed cubic equation in the form of $x^3 + px + q = 0$, where $p, q \in \mathbb{R}$. For more details, see [13] and [47, Appendix A].

1. If $p = q = 0$, then the equation has a triple real root.

$$x_1 = x_2 = x_3 = 0. \tag{D.1}$$

2. If $4p^3 + 27q^2 > 0$, then the equation only has one real root.

$$x_1 = -\frac{1}{3}(\sqrt[3]{Y_1} + \sqrt[3]{Y_2}). \tag{D.2}$$

   Here $Y_1 = \frac{3}{2}(9q + \sqrt{12p^3 + 81q^2})$ and $Y_2 = \frac{3}{2}(9q - \sqrt{12p^3 + 81q^2})$.

3. If $4p^3 + 27q^2 = 0$ (but $p \neq 0$ or $q \neq 0$), then the equation has three real roots, one of which is a double root.

$$x_1 = \frac{3q}{p}, \tag{D.3a}$$

$$x_2 = x_3 = -\frac{3q}{2p}. \tag{D.3b}$$

4. If $4p^3 + 27q^2 < 0$ (this implies $p < 0$), then the equation has three unequal real roots.

$$x_1 = -\frac{2\sqrt{-3p}}{3}\cos\frac{\theta}{3}, \tag{D.4a}$$

$$x_2 = \frac{\sqrt{-3p}}{3}(\cos\frac{\theta}{3} + \sqrt{3}\sin\frac{\theta}{3}), \tag{D.4b}$$

$$x_3 = \frac{\sqrt{-3p}}{3}(\cos\frac{\theta}{3} - \sqrt{3}\sin\frac{\theta}{3}). \tag{D.4c}$$

   Here, the $\theta = \arccos\big(-\frac{3\sqrt{3}q}{2p\sqrt{-p}}\big)$.

The advantage of using the formulae above is that it allows us to only use real number operations to obtain real roots. Eliminating the need for complex number operations simplifies implementation.

References

[1] Anshika, A., Li, J., Ghosh, D., Zhang, X., 2024. A three-operator splitting scheme derived from three-block ADMM. arXiv preprint arXiv:2411.00166 .

[2] Bochev, P., Ridzal, D., D'Elia, M., Perego, M., Peterson, K., 2020. Optimization-based, property-preserving finite element methods for scalar advection equations and their connection to algebraic flux correction. Computer Methods in Applied Mechanics and Engineering 367, 112982.

[3] Bochev, P., Ridzal, D., Scovazzi, G., Shashkov, M., 2012. Constrained-optimization based data transfer: A new perspective on flux correction, in: Flux-Corrected Transport: Principles, Algorithms, and Applications. Springer, pp. 345–398.

[4] Bochev, P., Ridzal, D., Shashkov, M., 2013. Fast optimization-based conservative remap of scalar fields through aggregate mass transfer. Journal of Computational Physics 246, 37–57.
[5] Bradley, A.M., Bosler, P.A., Guba, O., Taylor, M.A., Barnett, G.A., 2019. Communication-efficient property preservation in tracer transport. SIAM Journal on Scientific Computing 41, C161–C193.
[6] Briceno-Arias, L.M., 2015. Forward-Douglas–Rachford splitting and forward-partial inverse method for solving monotone inclusions. Optimization 64, 1239–1261.
[7] Chen, Y., Xiu, D., Zhang, X., 2025. On enforcing nonnegativity in polynomial approximations in high dimensions. SIAM Journal on Scientific Computing 47, A866–A888.
[8] Dai, Y.H., Fletcher, R., 2006. New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds. Mathematical Programming 106, 403–421.
[9] Davis, D., Yin, W., 2017. A three-operator splitting scheme and its optimization applications. Set-valued and variational analysis 25, 829–858.
[10] Demanet, L., Zhang, X., 2016. Eventual linear convergence of the Douglas–Rachford iteration for basis pursuit. Mathematics of Computation 85, 209–238.
[11] Douglas, J., Rachford, H.H., 1956. On the numerical solution of heat conduction problems in two and three space variables. Transactions of the American mathematical Society 82, 421–439.
[12] Einfeldt, B., Munz, C.D., Roe, P.L., Sjögreen, B., 1991. On Godunov-type methods near low densities. Journal of computational physics 92, 273–295.
[13] Fan, S., 1989. A new extracting formula and a new distinguishing means on the one variable cubic equation. Natural science journal of Hainan teachers college 2, 91.
[14] Gardner, C.L., Dwyer, S.J., 2009. Numerical simulation of the XZ Tauri supersonic astrophysical jet. Acta Mathematica Scientia 29, 1677–1683.
[15] Gil Torres, E., Jacobs, M., Zhang, X., 2025. Asymptotic linear convergence of ADMM for isotropic TV norm compressed sensing. arXiv preprint arXiv:2505.01240 .
[16] Grapsas, D., Herbin, R., Kheriji, W., Latché, J.C., 2016. An unconditionally stable staggered pressure correction scheme for the compressible Navier–Stokes equations. The SMAI journal of computational mathematics 2, 51–97.
[17] Guba, O., Taylor, M., St-Cyr, A., 2014. Optimization-based limiters for the spectral element method. Journal of Computational Physics 267, 176–195.
[18] Guermond, J.L., Maier, M., Popov, B., Tomas, I., 2021. Second-order invariant domain preserving approximation of the compressible Navier–Stokes equations. Computer Methods in Applied Mechanics and Engineering 375, 113608.
[19] Guermond, J.L., Nazarov, M., Popov, B., Tomas, I., 2018. Second-order invariant domain preserving approximation of the Euler equations using convex limiting. SIAM Journal on Scientific Computing 40, A3211–A3239.
[20] Guermond, J.L., Popov, B., 2017. Invariant domains and second-order continuous finite element approximation for scalar conservation equations. SIAM Journal on Numerical Analysis 55, 3120–3146.
[21] Ha, Y., Gardner, C.L., Gelb, A., Shu, C.W., 2005. Numerical simulation of high mach number astrophysical jets with radiative cooling. Journal of Scientific Computing 24, 29–44.
[22] Hajduk, H., 2021. Monolithic convex limiting in discontinuous Galerkin discretizations of hyperbolic conservation laws. Computers & Mathematics with Applications 87, 120–138.
[23] Kuzmin, D., 2020. Monolithic convex limiting for continuous finite element discretizations of hyperbolic conservation laws. Comput. Methods Appl. Mech. Engrg. 361, 112804.
[24] Kuzmin, D., Hajduk, H., 2024. Property-preserving numerical schemes for conservation laws. World Scientific.
[25] Lin, Y., Chan, J., 2024. High order entropy stable discontinuous Galerkin spectral element methods through subcell limiting. Journal of Computational Physics 498, 112677.
[26] Lions, P.L., Mercier, B., 1979. Splitting algorithms for the sum of two nonlinear operators. SIAM Journal on Numerical Analysis 16, 964–979.
[27] Liu, C., Buzzard, G.T., Zhang, X., 2024a. An optimization based limiter for enforcing positivity in a semi-implicit discontinuous Galerkin scheme for compressible Navier–Stokes equations. Journal of Computational Physics 519, 113440.
[28] Liu, C., Hu, J., Taitano, W.T., Zhang, X., 2025a. An optimization-based positivity-preserving limiter in semi-implicit discontinuous Galerkin schemes solving Fokker–Planck equations. Computers and Mathematics with Applications 192, 54–71.
[29] Liu, C., Riviere, B., Shen, J., Zhang, X., 2024b. A simple and efficient convex optimization based bound-preserving high order accurate limiter for Cahn–Hilliard–Navier–Stokes system. SIAM Journal on Scientific Computing 46, A1923–A1948.
[30] Liu, C., Sun, Z., Zhang, X., 2025b. A bound-preserving Runge–Kutta discontinuous Galerkin method with compact stencils for hyperbolic conservation laws. Journal of Computational Physics , 114071.
[31] Liu, C., Zhang, X., 2023. A positivity-preserving implicit-explicit scheme with high order polynomial basis for compressible Navier–Stokes equations. Journal of Computational Physics 493, 112496.
[32] Pazner, W., 2021. Sparse invariant domain preserving discontinuous Galerkin methods with subcell convex limiting. Computer Methods in Applied Mechanics and Engineering 382, 113876.
[33] Peaceman, D.W., Rachford, Jr, H.H., 1955. The numerical solution of parabolic and elliptic differential equations. Journal of the Society for industrial and Applied Mathematics 3, 28–41.
[34] Peterson, K., Bochev, P., Ridzal, D., 2024. Optimization-based, property-preserving algorithm for passive tracer transport. Computers & Mathematics with Applications 159, 267–286.
[35] Raguet, H., 2019. A note on the forward-Douglas–Rachford splitting for monotone inclusion and convex optimization. Optimization Letters 13, 717–740.
[36] Raguet, H., Fadili, J., Peyré, G., 2013. A generalized forward-backward splitting. SIAM Journal on Imaging Sciences 6,

1199–1226.
[37] Rueda-Ramírez, A.M., Pazner, W., Gassner, G.J., 2022. Subcell limiting strategies for discontinuous Galerkin spectral element methods. Computers & Fluids 247, 105627.
[38] Ruppenthal, F., Kuzmin, D., 2023. Optimal control using flux potentials: A way to construct bound-preserving finite element schemes for conservation laws. Journal of Computational and Applied Mathematics 434, 115351.
[39] van der Vegt, J.J., Xia, Y., Xu, Y., 2019. Positivity preserving limiters for time-implicit higher order accurate discontinuous Galerkin discretizations. SIAM journal on scientific computing 41, A2037–A2063.
[40] Wang, C., Zhang, X., Shu, C.W., Ning, J., 2012. Robust high order discontinuous Galerkin schemes for two-dimensional gaseous detonations. Journal of Computational Physics 231, 653–665.
[41] Xu, Z., 2014. Parametrized maximum principle preserving flux limiters for high order schemes solving hyperbolic conservation laws: one-dimensional scalar problem. Math. Comp. 83, 2213–2238.
[42] Yee, B.C., Olivier, S.S., Haut, T.S., Holec, M., Tomov, V.Z., Maginot, P.G., 2020. A quadratic programming flux correction method for high-order DG discretizations of SN transport. Journal of Computational Physics 419, 109696.
[43] Zalesak, S.T., 1979. Fully multidimensional flux-corrected transport algorithms for fluids. Journal of computational physics 31, 335–362.
[44] Zhang, X., 2017. On positivity-preserving high order discontinuous Galerkin schemes for compressible Navier–Stokes equations. Journal of Computational Physics 328, 301–343.
[45] Zhang, X., Shu, C.W., 2010a. On maximum-principle-satisfying high order schemes for scalar conservation laws. Journal of Computational Physics 229, 3091–3120.
[46] Zhang, X., Shu, C.W., 2010b. On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. Journal of Computational Physics 229, 8918–8934.
[47] Zhao, Z., Zhu, J., Chen, Y., Qiu, J., 2019. A new hybrid WENO scheme for hyperbolic conservation laws. Computers & Fluids 179, 422–436.