An optimization based limiter for enforcing positivity in a semi-implicit discontinuous Galerkin scheme for compressible Navier–Stokes equations

Chen Liu[a], Gregery T. Buzzard[a], Xiangxiong Zhang[a,]

[a]*Department of Mathematics, Purdue University, 150 North University Street, West Lafayette, Indiana 47907.*

Abstract

We consider an optimization-based limiter for enforcing positivity of internal energy in a semi-implicit scheme for solving gas dynamics equations. With Strang splitting, the compressible Navier–Stokes system is split into the compressible Euler equations, which are solved by the positivity-preserving Runge–Kutta discontinuous Galerkin (DG) method, and the parabolic subproblem, which is solved by Crank–Nicolson in time with interior penalty DG method. Such a scheme is at most second order accurate in time, high order accurate in space, conservative, and preserves positivity of density. To further enforce the positivity of internal energy, we impose an optimization-based limiter for the total energy variable to post-process DG polynomial cell averages. The optimization-based limiter can be efficiently implemented by the popular first order convex optimization algorithms such as the Douglas–Rachford splitting method by using nearly optimal algorithm parameters. Numerical tests suggest that the DG method with $\mathbb{Q}^k$ basis and the optimization-based limiter is robust for demanding low-pressure problems such as high-speed flows.

Keywords: compressible Navier–Stokes, semi-implicit, discontinuous Galerkin, high order accuracy, positivity-preserving, Douglas–Rachford splitting, optimization based limiter

2000 MSC: 65M12, 65M60, 65N30, 90C25

1. Introduction

1.1. Motivation and objective

For studying viscous gas dynamics, the dimensionless compressible Navier–Stokes (NS) equations without external forces in conservative form on a bounded spatial domain $\Omega \subset \mathbb{R}^d$ over time interval $[0, T]$ are

$$\partial_t \boldsymbol{U} + \boldsymbol{\nabla} \cdot \boldsymbol{F}^{\mathrm{a}} = \boldsymbol{\nabla} \cdot \boldsymbol{F}^{\mathrm{d}}, \quad \boldsymbol{F}^{\mathrm{a}} = \begin{pmatrix} \rho \boldsymbol{u} \\ \rho \boldsymbol{u} \otimes \boldsymbol{u} + p \mathbf{I} \\ (E + p) \boldsymbol{u} \end{pmatrix} \quad \text{and} \quad \boldsymbol{F}^{\mathrm{d}} = \frac{1}{\mathrm{Re}} \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\tau} \\ \boldsymbol{u} \cdot \boldsymbol{\tau} - \boldsymbol{q} \end{pmatrix}, \tag{1}$$

where the conservative variables are density ρ, momentum $\boldsymbol{m}$, and total energy E, Re denotes the Reynolds number and $\mathbf{I} \in \mathbb{R}^{d \times d}$ denotes an identity matrix, $\boldsymbol{u} = \frac{\boldsymbol{m}}{\rho}$ is velocity and p is pressure. With the Stokes hypothesis, the shear stress tensor is given by $\boldsymbol{\tau}(\boldsymbol{u}) = 2\boldsymbol{\varepsilon}(\boldsymbol{u}) - \frac{2}{3}(\boldsymbol{\nabla} \cdot \boldsymbol{u})\mathbf{I}$, where $\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}(\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^{\mathrm{T}})$. The total energy can be expressed as $E = \rho e + \frac{\|\boldsymbol{m}\|^2}{2\rho}$, where e denotes the internal energy and $\|\cdot\|$ is the vector 2-norm. With Fourier's heat conduction law, the heat diffusion flux $\boldsymbol{q} = -\lambda \boldsymbol{\nabla} e$ with parameters $\lambda = \frac{\gamma}{\mathrm{Pr}} > 0$, where the positive constant γ is the ratio of specific heats and Pr denotes the Prandtl number. For air, we have $\gamma = 1.4$ and $\mathrm{Pr} = 0.72$. For simplicity, we only consider the ideal gas equation of state

$$p = (\gamma - 1)\rho e. \tag{2}$$

Email addresses: liu3373@purdue.edu (Chen Liu), buzzard@purdue.edu (Gregery T. Buzzard), zhan1966@purdue.edu (Xiangxiong Zhang)

The system (1) can be written as

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0 \qquad \text{in } [0,T]\times\Omega, \tag{3a}$$

$$\partial_t (\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p - \tfrac{1}{\mathrm{Re}} \nabla \cdot \boldsymbol{\tau}(\boldsymbol{u}) = \boldsymbol{0} \qquad \text{in } [0,T]\times\Omega, \tag{3b}$$

$$\partial_t E + \nabla \cdot ((E+p)\boldsymbol{u}) - \tfrac{\lambda}{\mathrm{Re}} \Delta e - \tfrac{1}{\mathrm{Re}} \nabla \cdot (\boldsymbol{\tau}(\boldsymbol{u})\boldsymbol{u}) = 0 \qquad \text{in } [0,T]\times\Omega. \tag{3c}$$

When vacuums occur, the solutions of compressible NS equations may lose continuous dependency with respect to the initial data, see [1, Theorem 2] and [2, Remark 3.3]. On the other hand, the density and internal energy of a physically meaningful solution in most applications should both be positive. For problems without any vaccum, define the set of admissible states as

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

The function $\rho e(\boldsymbol{U}) = E - \frac{\|\boldsymbol{m}\|^2}{2\rho}$ is a concave function of $\boldsymbol{U}$, which implies the set G is convex [3]. For an initial condition $\boldsymbol{U}^0 = [\rho^0, \boldsymbol{m}^0, E^0]^{\mathrm{T}} \in G$, a numerical solution preserving the positivity is preferred for the sake of not only physical meaningfulness but also numerical robustness. For the equation of state (2), negative internal energy means negative pressure, with which the linearized compressible Euler equation loses hyperbolicity and its initial value problem is ill-posed [3]. On the other hand, a conservative and positivity-preserving scheme in the sense of preserving the invariant domain G is numerically robust [4, 5, 2, 6, 7].

For solving a convection-diffusion system (3), fully explicit time stepping results in a time step constraint $\Delta t = O(\mathrm{Re}\Delta x^2)$, thus is suitable only for high Reynolds number flows in practice. In order to achieve larger time steps such as a hyperbolic CFL $\Delta t = O(\Delta x)$, a semi-implicit scheme can be used [2, 7].

The objective of this paper is to construct a high order accurate in space, conservative, and positivity-preserving scheme for solving the compressible NS equations (3). In particular, we will use the Strang splitting approach in [2, 7] with arbitrarily high order discontinuous Galerkin (DG) method for spatial discretization, which gives a scheme of at most second order accuracy in time. In general, a scheme that is high order in both time and space is preferred. On the other hand, for many fluid problems including gas dynamics problems, the solutions are often smoother with respect to the time variable, thus the spatial resolution of a numerical scheme is often more crucial for capturing fine structures in solutions than its temporal accuracy. Higher order spatial discretizations often produce better numerical solutions even if the time accuracy is only first order for various convection-diffusion problems [8, 9, 10, 7].

1.2. Existing positivity-preserving schemes for compressible NS equations

In the literature, there are many positivity-preserving schemes for compressible Euler equations, which have been well studied since 1990s. For compressible Navier–Stokes equations, most of the practical positivity-preserving schemes were developed only in the past decade.

Grapas et al. in [4] constructed a fully implicit pressure correction scheme on staggered grids, which is at most second order in space, conservative, and unconditionally positivity-preserving. Nonlinear systems must be solved for time marching. As a fully implicit scheme on a staggered grid, it seems difficult to extend it to a higher order accurate scheme.

Zhang in [5] proposed a simple nonlinear diffusion numerical flux, with which arbitrarily high order Runge–Kutta DG schemes solving (3) can be rendered positivity-preserving without losing conservation and accuracy by a simple positivity-preserving limiter in [3]. The advantages of such a fully explicit approach include easy extensions to general shear stress models and heat fluxes, and possible extensions to other types of schemes, such as high order finite volume schemes [11] and the high order finite difference WENO (weighted essentially nonoscillatory) schemes [6]. However, like many fully explicit schemes for convection-diffusion systems [12, 13, 14, 15], the time step constraint is $\Delta t = O(\mathrm{Re}\,\Delta x^2)$.

Guermond et al. in [2] introduced a semi-implicit continuous finite element scheme via Strang splitting, which preserves positivity under standard hyperbolic CFL condition $\Delta t = O(\Delta x)$. By the same operator splitting approach, in [7] we constructed a semi-implicit conservative DG scheme, with the continuous finite

element method for solving (3), and the scheme with $\mathbb{Q}^k$ ($k = 1, 2, 3$) basis can be proven positivity-preserving with $\Delta t = O(\Delta x)$.

The early pioneering work on DG methods for solving compressible NS equations was conducted by Bassi and Rebay [16, 17] as well as Baumann and Oden [18]. Advantages of DG methods include high order accuracy, flexibility in handling complex meshes and hp-adaptivity, and highly parallelizable characteristics. See [19, 20, 21] for an overview of DG methods. In this paper, we focus on constructing DG schemes within the Strang splitting approach, by which the compressible NS system (3) is splitted into a hyperbolic subproblem (H) and a parabolic subproblem (P), representing two asymptotic regimes: the vanishing viscosity limit (the compressible Euler equations) and the dominance of diffusive terms:

$$\text{(H)} \begin{cases} \partial_t \rho + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = 0, \\ \partial_t (\rho \boldsymbol{u}) + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u} + p \mathbf{I}) = \mathbf{0}, \\ \partial_t E + \boldsymbol{\nabla} \cdot ((E + p) \boldsymbol{u}) = 0, \end{cases} \qquad \text{(P)} \begin{cases} \partial_t \rho = 0, \\ \partial_t (\rho \boldsymbol{u}) - \frac{1}{\mathrm{Re}} \boldsymbol{\nabla} \cdot \boldsymbol{\tau}(\boldsymbol{u}) = \mathbf{0}, \\ \partial_t E - \frac{\lambda}{\mathrm{Re}} \Delta e - \frac{1}{\mathrm{Re}} \boldsymbol{\nabla} \cdot (\boldsymbol{\tau}(\boldsymbol{u}) \boldsymbol{u}) = 0. \end{cases} \tag{4}$$

The equation $\partial_t \rho = 0$ in the parabolic subproblem implies the variable ρ in (P) is time independent. Multiplying the second equation in (P) by $\boldsymbol{u}$ and using the identity $\boldsymbol{\nabla} \cdot (\boldsymbol{\tau}(\boldsymbol{u}) \boldsymbol{u}) = (\boldsymbol{\nabla} \cdot \boldsymbol{\tau}(\boldsymbol{u})) \cdot \boldsymbol{u} + \boldsymbol{\tau}(\boldsymbol{u}) : \boldsymbol{\nabla} \boldsymbol{u}$, we obtain the following equivalent system in non-conservative form:

$$\text{(P)} \begin{cases} \partial_t \rho = 0, & \text{(5a)} \\ \rho \partial_t \boldsymbol{u} - \frac{1}{\mathrm{Re}} \boldsymbol{\nabla} \cdot \boldsymbol{\tau}(\boldsymbol{u}) = \mathbf{0}, & \text{(5b)} \\ \rho \partial_t e - \frac{\lambda}{\mathrm{Re}} \Delta e = \frac{1}{\mathrm{Re}} \boldsymbol{\tau}(\boldsymbol{u}) : \boldsymbol{\nabla} \boldsymbol{u}. & \text{(5c)} \end{cases}$$

We use the positivity-preserving Runge–Kutta DG method [3] for subproblem (H), i.e., the Zhang–Shu method for constructing positivity-preserving schemes [22, 3, 23, 24, 25] applied to solving compressible Euler equations, which is arbitrarily high order accurate, conservative, and positivity-preserving. For the parabolic subproblem, many different types of DG methods have been developed for solving diffusion equations in literature, which include interior penalty DG [26, 27, 28, 29], local DG [30, 31], direct DG [32, 33, 34], hybridizable DG [35, 36, 37], compact DG [38, 39], and so on. In this paper, we utilize the interior penalty DG method to discretize subproblem (P). The first challenge of using DG methods for subproblem (P) is how to ensure conservation of conserved variables. In [7], we have proven that conservation can be preserved via choosing appropriate interior penalty DG forms of $\boldsymbol{\nabla} \cdot \boldsymbol{\tau}(\boldsymbol{u})$ and $\boldsymbol{\tau}(\boldsymbol{u}) : \boldsymbol{\nabla} \boldsymbol{u}$. The next major challenge is how to ensure positivity when discretizing (5c). It is very difficult to prove any positivity-preserving result for arbitrarily high order schemes solving (5c) for implicit time stepping, even if the temporal accuracy is only first order.

Consider a heat equation $\partial_t e - \Delta e = 0$ as a simplification of (5c). When using backward Euler time discretization, a systematic approach to obtaining a sufficient condition for the discrete maximum principle or positivity is to show the monotonicity of the linear system matrix. A matrix is called *monotone* if all entries of its inverse are nonnegative. The monotonicity of $\mathbb{Q}^1$ interior penalty DG on multi-dimensional structured meshes has been established in [7], also see [40, 41] for related results; and the monotonicity of continuous finite element method with $\mathbb{Q}^2$ and $\mathbb{Q}^3$ elements has been proven in [42, 43, 44]. However, for arbitrary high order schemes on unstructured meshes, the monotonicity does not hold [45]. Furthermore, for higher order implicit time marching strategy, such as the Crank–Nicolson method, the monotonicity of the linear system matrix is not enough to ensure positivity using a time step like $O(\Delta x)$. The Crank–Nicolson method with a monotone spatial discretization preserves positivity only if the time step is as small as $O(\Delta x^2)$, see [46, Appendix B] and [2, Section 5.3].

1.3. A constraint optimization approach for enforcing positivity and global conservation

To preserve positivity of internal energy, we will introduce a constraint optimization postprocessing approach. For enforcing bounds or positivity in numerical schemes solving PDEs, various optimization based approaches have been considered in the literature. We list a few such methods. Guba et al. in [47] introduced a bound-preserving limiter for spectral element method, implemented by standard quadratic

programming solvers. van der Vegt et al. in [48] considered a positivity-preserving limiter for DG scheme with implicit time integration and formulated the positivity constraints in the KKT system, implemented by an active set semismooth Newton method. Cheng and Shen in [49] introduced a Lagrange multiplier approach to preserve bounds for semilinear and quasi-linear parabolic equations, which provides a new interpretation for the cut-off method and achieves the preservation of mass by solving a nonlinear algebraic equation for the additional space independent Lagrange multiplier. Ruppenthal and Kuzmin in [50] utilized optimization-based flux correction to ensure the positivity of finite element discretization of conservation laws. The primal-dual Newton method was employed to calculate the optimal flux potentials.

Next, we describe the main idea of our approach. Let $\overline{\boldsymbol{U}_i^{\mathrm{P}}} = [\overline{\rho_i^{\mathrm{P}}}, \overline{\boldsymbol{m}_i^{\mathrm{P}}}, \overline{E_i^{\mathrm{P}}}]^{\mathrm{T}}$ be a vector denoting the cell average of the DG polynomial $\boldsymbol{U}_h^{\mathrm{P}}(\boldsymbol{x}) = [\rho_h^{\mathrm{P}}(\boldsymbol{x}), \boldsymbol{m}_h^{\mathrm{P}}(\boldsymbol{x}), E_h^{\mathrm{P}}(\boldsymbol{x})]^{\mathrm{T}}$ on the i-th cell K_i after solving subproblem (P). The density cell averages are positive, which can be ensured if using a positivity-preserving scheme for subproblem (H). The main challenge here is that in general $\overline{\boldsymbol{U}_i^{\mathrm{P}}}$ may not be in the convex invariant domain set G. We emphasize that the Zhang–Shu limiter [3] can be used only if $\overline{\boldsymbol{U}_i^{\mathrm{P}}} \in G$, which can be proven for one time step or time stage for fully explicit finite volume and DG schemes with a positivity-preserving flux [3, 5], or very special semi-implicit schemes like [7], thus these schemes can be rendered positivity-preserving by using the Zhang–Shu limiter [3] in each time step or time stage.

With a prescribed small positive number ϵ, which serves as the desired lower bound for density and internal energy, the numerical admissible state set G^ϵ is defined as follows.

$$G^\epsilon = \{\boldsymbol{U} = [\rho, \boldsymbol{m}, E]^{\mathrm{T}} : \ \rho \geq \epsilon, \ \rho e(\boldsymbol{U}) = E - \frac{\|\boldsymbol{m}\|^2}{2\rho} \geq \epsilon\}.$$

Define $\overline{E_h^{\mathrm{P}}} = [\overline{E_1^{\mathrm{P}}}, \overline{E_2^{\mathrm{P}}}, \cdots, \overline{E_N^{\mathrm{P}}}]^{\mathrm{T}}$ as the vector of all cell averages for the total energy. We propose to modify the total energy only. And we would like to modify it to another vector $\overline{E}_h = [\overline{E}_1, \overline{E}_2, \cdots, \overline{E}_N]^{\mathrm{T}}$ such that it minimizes the ℓ^2 distance to $\overline{E_h^{\mathrm{P}}}$, subject to the constraints of preserving global conservation and positivity. Specifically, given $\overline{\boldsymbol{U}_h^{\mathrm{P}}} = [\overline{\boldsymbol{U}_1^{\mathrm{P}}}, \cdots, \overline{\boldsymbol{U}_N^{\mathrm{P}}}]^{\mathrm{T}}$ with positive density $\overline{\rho_i^{\mathrm{P}}} \geq \epsilon$, find the minimizer for

$$\min_{\overline{E}_h \in \mathbb{R}^N} \left\| \overline{E}_h - \overline{E_h^{\mathrm{P}}} \right\|^2 \quad \text{subjects to} \quad \sum_{i=1}^N \overline{E}_i |K_i| = \sum_{i=1}^N \overline{E_i^{\mathrm{P}}} |K_i| \quad \text{and} \quad [\overline{\rho_i^{\mathrm{P}}}, \overline{\boldsymbol{m}_i^{\mathrm{P}}}, \overline{E}_i]^{\mathrm{T}} \in G^\epsilon, \quad \forall i, \tag{6a}$$

where $|K_i|$ is the area or volume of each cell K_i. Let $\overline{E}_h^* = [\overline{E}_1^*, \cdots, \overline{E}_N^*]^{\mathrm{T}}$ be the minimizer. Then we correct the DG polynomial cell averages for the total energy variable. Namely, let $E_i^{\mathrm{P}}(\boldsymbol{x})$ be the DG polynomial in each cell K_i, and we correct it by a constant

$$E_i(\boldsymbol{x}) = E_i^{\mathrm{P}}(\boldsymbol{x}) - \overline{E_i^{\mathrm{P}}} + \overline{E}_i^*. \tag{6b}$$

The updated or postprocessed DG polynomials $\boldsymbol{U}_h^{\mathrm{P}}(\boldsymbol{x}) = [\rho_h^{\mathrm{P}}(\boldsymbol{x}), \boldsymbol{m}_h^{\mathrm{P}}(\boldsymbol{x}), E_h(\boldsymbol{x})]^{\mathrm{T}}$ now have cell averages in the numerical admissible state set G^ϵ, and the simple Zhang–Shu positivity-preserving limiter in [3, 23] can be used to further ensure the full scheme is positivity-preserving.

Since ℓ^2 distance is minimized, the accuracy of (6a) can also be justified under suitable assumptions, which will be discussed in Section 3.2.

1.4. Efficient implementation of the constraint optimization defined postprocessing

The simple postprocessing approach (6) was considered in [51] for preserving bounds of a scalar variable in complex phase field equations. Thanks to the constraints in (6a), global conservation and positivity of the internal energy are easily achieved, and the accuracy is also easy to justify for scalar variables [51], which are the advantages of such a simple approach. On the other hand, in any optimization based approach, it is often quite straightforward to have these desired properties such as positivity, conservation, and high order

accuracy. From this perspective, the critical issue in all optimization based approaches is computational efficiency, especially for a time-dependent, demanding nonlinear system like (3).

In large-scale high-resolution fluid dynamic simulations, degree of freedoms to be processed at each time step can be quite large. Thus in general it is preferred to solve (6a) by first order optimization methods since they scale well with problem size, i.e., the complexity is $O(N)$ for each iteration, with N being the total number of cells.

In [51], it is demonstrated that the minimizer to a constraint minimization like (6a) can be efficiently computed by using the Douglas–Rachford splitting method [52] if using the nearly optimal algorithm parameters obtained from a sharp asymptotic convergence rate analysis. The Douglas–Rachford splitting method is a very popular first order splitting method, because it is equivalent to ADMM [53] and dual split Bregman method [54] with special parameters, see also [55] and references therein for the equivalence. For special convex optimization problems, it is also equivalent to PDHG [56].

There are other efficient alternative methods to solve the minimization (6a), such as the breakpoint searching algorithms [57] with an $O(N)$ computational complexity. For the ℓ^2-norm minimization (6a), the Douglas–Rachford splitting with the optimal parameters also has a provable computational complexity $O(N)$ as shown in [51], but with more flexibilities and advantages. First, the Douglas–Rachford splitting method is simple to describe and easy to implement since only three steps are needed in each iteration, which allows easy implementation, especially for efficient parallel computing. Second, it is straightforward to extend the Douglas–Rachford splitting method to other postprocessing models such as the ℓ^1-norm minimization and directly enforcing invariant domain G^ϵ, see Remark 3 and Remark 4 in Section 3.3. Though the Douglas–Rachford splitting method may no longer have a provable $O(N)$ computational complexity for ℓ^1-norm minimization, it is nontrivial or impossible to generalize other alternative methods for (6a) to ℓ^1-norm minimization. In Appendix A, we show a comparison to one simple and efficient alternative solving (6a) by the method of Lagrange multiplier, to demonstrate the practical efficiency of the Douglas–Rachford splitting for large problems.

Given the DG polynomial after solving the subproblem (P), we define the i-th cell as a bad cell if its cell average has negative internal energy, i.e., $\overline{\boldsymbol{U}_i^{\mathrm{P}}} = [\overline{\rho_i^{\mathrm{P}}}, \overline{\boldsymbol{m}_i^{\mathrm{P}}}, \overline{E_i^{\mathrm{P}}}]^{\mathrm{T}} \notin G^\epsilon$. Let r be the number of bad cells, then r/N is the bad cell ratio. It is proven in [51] that the sharp asymptotic linear convergence rate of the Douglas–Rachford splitting with the nearly optimal parameters is approximately $\frac{1-2\frac{r}{N}}{3-2\frac{r}{N}} \approx \frac{1}{3}$ when $r \ll N$. In other words, such a minimization solver is provably extremely efficient when the bad cell ratio is small, which is usually the case for a good scheme solving (3) such as Strang splitting with DG methods [7].

1.5. The main result and organization of this paper

Our full scheme in this paper is a very high order accurate in space, conservative, and positivity-preserving semi-implicit DG scheme to solve the compressible NS equations (3), with a standard hyperbolic CFL $\Delta t = O(\Delta x)$. For the implicit part, the scheme is fully decoupled with two linear systems to solve sequentially for each time step. We emphasize that the spatial discretization in this paper is done by only DG methods, which is not exactly the same as the spatial scheme in [7], where the internal energy equation is discretized by continuous finite element method. The main novelties of this paper include the optimization-based postprocessing approach (6) to preserve conservation and positivity for solving the parabolic subproblem as well as a proper semi-implicit DG scheme with high order basis, which is carefully designed so that the DG scheme combined with the optimization-based positivity-preserving limiter can produce stable and solid results for challenging benchmark gas dynamics problems. The minimizer to (6a) can be efficiently computed by using the generalized Douglas–Rachford splitting method with nearly optimal parameters.

The postprocessing step (6a) only preserves the global conservation and does not preserve any local conservation property. We remark that the local conservation in the Strang splitting approach for solving (3) is already lost since the non-conservative variables are computed in (5). Nonetheless, the global conservation can be ensured [7]. Thus from this perspective, the postprocessing step (6a) is acceptable whenever the non-conservative form (5) is solved.

One can also consider a more general version of (6a) by also modifying the density and momentum variables to enforce the positivity of the internal energy $\overline{\boldsymbol{U}_i^{\mathrm{P}}} = [\overline{\rho_i^{\mathrm{P}}}, \overline{\boldsymbol{m}_i^{\mathrm{P}}}, \overline{E_i^{\mathrm{P}}}]^{\mathrm{T}} \in G^{\epsilon}$. Such a more complicated limiter is certainly more difficult to implement efficiently. On the other hand, for the Strang splitting approach in [2, 7], the momentum variable is robustly computed, which allows us to consider a simpler limiter like (6a). Most importantly, numerical tests suggest that the simple postprocessing (6) is sufficient to enforce the positivity thus the robustness for the subproblem (P) in the Strang splitting with very high order DG methods.

We emphasize that the postprocessing (6) is too simple to make a bad scheme more useful, e.g., it does not eliminate any oscillations. It is most useful for a good scheme that is stable for most testing cases yet might lose positivity thus robustness for solving challenging low pressure problems, e.g., the Strang splitting method in [2, 7]. For instance, as will be shown by numerical tests in this paper, for computing the Mach 2000 astrophysical jet problem, Strang splitting with very high order DG methods produces blow-up due to loss of positivity, but will be stable when combined with the postprocessing (6), i.e., an optimization based positivity-preserving limiter. On the other hand, there are many different kinds of DG methods for diffusion operators. With a proper choice of the interior penalty DG method, we demonstrate that global conservation can be ensured when solving the diffusion subproblem implicitly in the Strang splitting of compressible Navier-Stokes system, and only two linear systems need to be solved in the Crank–Nicolson time discretization of the diffusion subproblem. Moreover, the numerical tests suggest that such a high order DG scheme is a practical scheme producing solid results for some chanllenging benchmark problems.

The rest of this paper is organized as follows. In Section 2, we introduce the fully discrete numerical scheme. In Section 3, we discuss a high order accurate constraint optimization based postprecessing procedure, which preserves the conservation and positivity. Numerical tests are shown in Section 4. Concluding remarks are given in Section 5.

2. Numerical scheme

In this section, we describe the fully discretized numerical scheme for solving the compressible NS equations (3). Our scheme incorporates the DG spatial discretization within the Strang splitting framework.

2.1. Time discretization

Given the conserved variables $\boldsymbol{U}^n$ at time t^n ($n \geq 0$) and the step size Δt, the Strang splitting for evolving to time $t^{n+1} = t^n + \Delta t$ for the system (3) is to solve subproblems (H) and (P) separately [2, 7]. A schematic flowchart for time marching is as follows:

$$\boldsymbol{U}^n \xrightarrow[\text{step size } \frac{\Delta t}{2}]{\text{solve (H)}} \boldsymbol{U}^{\mathrm{H}} \xrightarrow[\text{step size } \Delta t]{\text{solve (P)}} \boldsymbol{U}^{\mathrm{P}} \xrightarrow[\text{step size } \frac{\Delta t}{2}]{\text{solve (H)}} \boldsymbol{U}^{n+1}. \tag{7}$$

We utilize the strong stability preserving (SSP) Runge–Kutta method to solve (H) and the θ-method with a parameter $\theta \in (0, 1]$ to solve (P). For any $n \geq 0$, the time discretization in one time step consists of the following steps.

Step 1. Given $\boldsymbol{U}^n = [\rho^n, \boldsymbol{m}^n, E^n]^{\mathrm{T}}$, we use the third order SSP Runge–Kutta method with step size $\frac{1}{2}\Delta t$ to compute $\boldsymbol{U}^{\mathrm{H}} = [\rho^{\mathrm{H}}, \boldsymbol{m}^{\mathrm{H}}, E^{\mathrm{H}}]^{\mathrm{T}}$:

$$\boldsymbol{U}^{(1)} = \boldsymbol{U}^n - \frac{\Delta t}{2}\nabla \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}^n), \tag{8a}$$

$$\boldsymbol{U}^{(2)} = \frac{3}{4}\boldsymbol{U}^n + \frac{1}{4}\Big[\boldsymbol{U}^{(1)} - \frac{\Delta t}{2}\nabla \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}^{(1)})\Big], \tag{8b}$$

$$\boldsymbol{U}^{\mathrm{H}} = \frac{1}{3}\boldsymbol{U}^n + \frac{2}{3}\Big[\boldsymbol{U}^{(2)} - \frac{\Delta t}{2}\nabla \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}^{(2)})\Big]. \tag{8c}$$

Step 2. Given $\boldsymbol{U}^{\mathrm{H}} = [\rho^{\mathrm{H}}, \boldsymbol{m}^{\mathrm{H}}, E^{\mathrm{H}}]^{\mathrm{T}}$, compute $(\boldsymbol{u}^{\mathrm{H}}, e^{\mathrm{H}})$ by solving

$$\boldsymbol{m}^{\mathrm{H}} = \rho^{\mathrm{H}}\boldsymbol{u}^{\mathrm{H}} \quad \text{and} \quad E^{\mathrm{H}} = \rho^{\mathrm{H}} e^{\mathrm{H}} + \frac{\|\boldsymbol{m}^{\mathrm{H}}\|^2}{2\rho^{\mathrm{H}}}.$$

Step 3. Given $(\boldsymbol{u}^{\mathrm{H}}, e^{\mathrm{H}})$, set $\rho^{\mathrm{P}} = \rho^{\mathrm{H}}$ due to (5a). We employ the Crank–Nicolson method to discretize (5b) and apply the θ-method, where $\theta \in (0, 1]$, to discretize (5c). For the second step in Strang splitting (7), we have

$$\begin{aligned}
&\boldsymbol{u}^* = \frac{1}{2}\boldsymbol{u}^{\mathrm{P}} + \frac{1}{2}\boldsymbol{u}^{\mathrm{H}} \quad \text{and} \quad e^* = \theta e^{\mathrm{P}} + (1-\theta)e^{\mathrm{H}}, \\
&\rho^{\mathrm{P}}\frac{\boldsymbol{u}^{\mathrm{P}} - \boldsymbol{u}^{\mathrm{H}}}{\Delta t} - \frac{1}{\mathrm{Re}}\boldsymbol{\nabla}\cdot\boldsymbol{\tau}(\boldsymbol{u}^*) = \boldsymbol{0}, \\
&\rho^{\mathrm{P}}\frac{e^{\mathrm{P}} - e^{\mathrm{H}}}{\Delta t} - \frac{\lambda}{\mathrm{Re}}\Delta e^* = \frac{1}{\mathrm{Re}}\boldsymbol{\tau}(\boldsymbol{u}^*) : \boldsymbol{\nabla}\boldsymbol{u}^*.
\end{aligned}$$

The scheme above can be implemented as first to compute $(\boldsymbol{u}^*, e^*)$ by sequentially solving two decoupled linear systems

$$\rho^{\mathrm{P}}\boldsymbol{u}^* - \frac{\Delta t}{2\mathrm{Re}}\boldsymbol{\nabla}\cdot\boldsymbol{\tau}(\boldsymbol{u}^*) = \rho^{\mathrm{H}}\boldsymbol{u}^{\mathrm{H}}, \tag{9a}$$

$$\rho^{\mathrm{P}} e^* - \frac{\theta\Delta t\,\lambda}{\mathrm{Re}}\Delta e^* = \rho^{\mathrm{H}} e^{\mathrm{H}} + \frac{\theta\Delta t}{\mathrm{Re}}\boldsymbol{\tau}(\boldsymbol{u}^*) : \boldsymbol{\nabla}\boldsymbol{u}^*, \tag{9b}$$

then set $\boldsymbol{u}^{\mathrm{P}} = 2\boldsymbol{u}^* - \boldsymbol{u}^{\mathrm{H}}$ and $e^{\mathrm{P}} = \frac{1}{\theta}e^* + (1 - \frac{1}{\theta})e^{\mathrm{H}}$.

Step 4. Given $(\rho^{\mathrm{P}}, \boldsymbol{u}^{\mathrm{P}}, e^{\mathrm{P}})$, compute $(\boldsymbol{m}^{\mathrm{P}}, E^{\mathrm{P}})$ by

$$\boldsymbol{m}^{\mathrm{P}} = \rho^{\mathrm{P}}\boldsymbol{u}^{\mathrm{P}} \quad \text{and} \quad E^{\mathrm{P}} = \rho^{\mathrm{P}} e^{\mathrm{P}} + \frac{\|\boldsymbol{m}^{\mathrm{P}}\|^2}{2\rho^{\mathrm{P}}}.$$

Step 5. Given $\boldsymbol{U}^{\mathrm{P}} = [\rho^{\mathrm{P}}, \boldsymbol{m}^{\mathrm{P}}, E^{\mathrm{P}}]^{\mathrm{T}}$, to obtain $\boldsymbol{U}^{n+1} = [\rho^{n+1}, \boldsymbol{m}^{n+1}, E^{n+1}]^{\mathrm{T}}$ in the third step in Strang splitting (7), solve (H) for another $\frac{1}{2}\Delta t$ by the third order SSP Runge–Kutta method.

We have the first order backward Euler scheme with $\theta = 1$, for which $e^{\mathrm{P}} = e^*$ and it is possible to design positivity-preserving schemes if the discrete Laplacian is monotone, e.g., $\mathbb{Q}^2$ and $\mathbb{Q}^3$ spectral element methods on uniform meshes, as shown in [7]. Unfortunately, for any $\theta < 1$, $e^{\mathrm{P}} = \frac{1}{\theta}e^* + (1 - \frac{1}{\theta})e^{\mathrm{H}}$ is not a convex combination thus it is difficult to have $e^P > 0$ even if $e^* > 0$ can be ensured by a monotone discrete Laplacian. For $\theta = \frac{1}{2}$, we have the second order Crank–Nicolson scheme. It is important to note that in each time step, only two decoupled linear systems need to be sequentially solved in (9).

2.2. Preliminary aspects of space discretization

We use the Runge–Kutta DG method to discretize subproblem (H) and the interior penalty DG method to discretize subproblem (P). For completeness, we briefly review these methods without delving into their derivation. See [3, 5, 7] for more details. For simplicity, we only consider $\mathbb{Q}^k$ polynomial basis on uniform rectangular meshes, and there is no essential difficulty to extend the main results in this paper to unstructured meshes. For example, for preserving conservation and positivity, the constraint optimization-based postprocessing approach discussed in Section 2.3 is also applicable to $\mathbb{P}^k$ polynomials on unstructured meshes.

Mesh, approximation spaces, and quadratures. Let $\mathcal{T}_h = \{K_i\}$ be a uniform partition of the computational domain Ω by square elements (cells) with the element diameter h. The unit outward normal of a cell K is denoted by $\boldsymbol{n}_K$. Let Γ_h be the set of interior faces. For each interior face $e \in \Gamma_h$ shared by cells K_{i^-} and K_{i^+}, with $i^- < i^+$, we define a unit normal vector $\boldsymbol{n}_e$ that points from K_{i^-} into K_{i^+}. For a boundary face $e = \partial K_{i^-} \cap \partial\Omega$, the normal $\boldsymbol{n}_e$ is taken to be the unit outward vector to $\partial\Omega$.

Let $\mathbb{Q}^k(K)$ be the space of polynomials of order at most k for each variable defined on a cell K. Define the following piecewise polynomial spaces:

$$M_h^k = \left\{\chi_h \in L^2(\Omega) : \ \forall K \in \mathcal{T}_h, \ \chi_h|_K \in \mathbb{Q}^k(K)\right\},$$
$$\mathbf{X}_h^k = \left\{\boldsymbol{\theta}_h \in L^2(\Omega)^d : \ \forall K \in \mathcal{T}_h, \ \boldsymbol{\theta}_h|_K \in \mathbb{Q}^k(K)^d\right\}.$$

On a reference element $\hat{K} = [-\frac{1}{2}, \frac{1}{2}]^d$, we use $(k+1)^d$ Gauss–Lobatto points to construct Lagrange interpolation polynomials $\hat{\varphi}_j$. The basis functions on each cell $K_i \in \mathcal{T}_h$ are defined by $\varphi_{ij} = \hat{\varphi}_j \circ \boldsymbol{F}_i^{-1}$, where $\boldsymbol{F}_i : \hat{K} \to K$ is an invertible mapping from the reference element to K_i. These basis are numerically orthogonal with respect to the $(k+1)^d$-point Gauss–Lobatto quadrature rule.

We summarize the quadrature rules employed in solving the hyperbolic and parabolic subproblems as well as the points to be used in the positivity-preserving limiter as follows:

1. For face and volume integrals in (H), we utilize a quadrature rule that is constructed by the tensor product of $(k+1)$-point Gauss quadrature. Denote the set of associated quadrature points here by $S_K^{\mathrm{H,int}}$ on a cell K.

2. For face and volume integrals in (P), we utilize a quadrature rule that is constructed by the tensor product of $(k+1)$-point Gauss–Lobatto quadrature. Denote the set of associated quadrature points here by S_K^{P} on a cell K.

3. The points for weak positivity of (H) are constructed by $(k+1)$-point Gauss quadrature tensor product with L-point Gauss–Lobatto quadrature in both x and y directions and we need $2L-3 \geq k$ so that the L-point Gauss–Lobatto quadrature is exact for integrating DG polynomials of degree k [5]. Denote the set of associated quadrature points here by $S_K^{\mathrm{H,aux}}$ on a cell K. Though these points form a quadrature, we do not use them for computing any integrals. Instead, they are the points to be used in the positivity-preserving limiter [3, 23, 5].

See Figure 1 for an illustration the location of these quadrature points in the $\mathbb{Q}^4$ scheme.

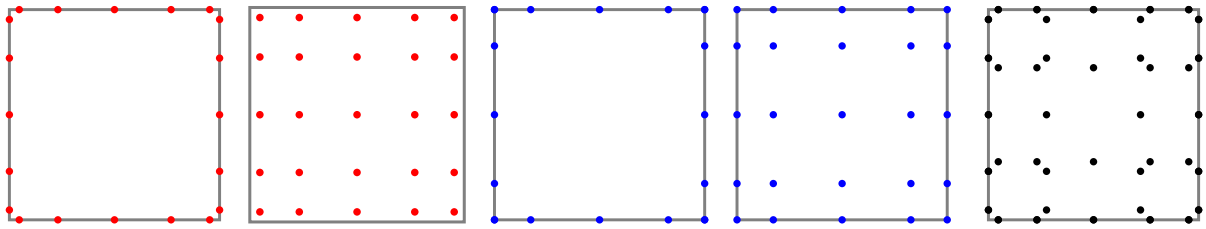

Figure 1: An illustration of the quadratures used in the $\mathbb{Q}^4$ scheme. From left to right: the quadrature points for face integrals in (H), volume integrals in (H), face integrals in (P), volume integrals in (P), and the quadrature points for weak positivity. The black points are used only in defining the positivity-preserving limiter, and they are not used in calculating any numerical integration.

Hyperbolic subproblem. One of the most popular high order accurate positivity-preserving approaches for solving compressible Euler equations $\partial_t \boldsymbol{U} + \boldsymbol{\nabla} \cdot \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}) = \boldsymbol{0}$ was introduced by Zhang and Shu in [3], also

see [5]. We utilize the same scheme to solve (H), which is defined as follows. For any piecewise polynomial test function Ψ_h, find the piecewise polynomial solution $\boldsymbol{U}_h$, such that

$$\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{U}_h, \Psi_h) = (\boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}_h), \nabla\Psi_h) - \int_{\partial K} \widehat{\boldsymbol{F}^{\mathrm{a}} \cdot \boldsymbol{n}_K}(\boldsymbol{U}_h^-, \boldsymbol{U}_h^+)\Psi_h, \tag{10}$$

where $\widehat{\boldsymbol{F}^{\mathrm{a}} \cdot \boldsymbol{n}_K}$ is any monotone flux for $\boldsymbol{F}^{\mathrm{a}}$, e.g., a Lax–Friedrichs type flux. On a face $e \subset \partial K$, the local Lax–Friedrichs flux is defined by

$$\widehat{\boldsymbol{F}^{\mathrm{a}} \cdot \boldsymbol{n}_K}(\boldsymbol{U}_h^-, \boldsymbol{U}_h^+) = \frac{\boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}_h^-) + \boldsymbol{F}^{\mathrm{a}}(\boldsymbol{U}_h^+)}{2} \cdot \boldsymbol{n}_K - \frac{\alpha_e}{2}(\boldsymbol{U}_h^+ - \boldsymbol{U}_h^-),$$

where the $\boldsymbol{U}_h^-$ (resp. $\boldsymbol{U}_h^+$) denotes the trace of $\boldsymbol{U}_h$ on the face ∂K coming from the interior (resp. exterior) of K. The factor α_e denotes the maximum wave speed with maximum taken over all $\boldsymbol{U}_h^-$ and $\boldsymbol{U}_h^+$ along the face e, namely the largest magnitude of the eigenvalues of the Jacobian matrix $\frac{\partial \boldsymbol{F}^{\mathrm{a}}}{\partial \boldsymbol{U}}$, which equals to the wave speed $|\boldsymbol{u} \cdot \boldsymbol{n}_K| + \sqrt{\gamma \frac{p}{\rho}}$ for ideal gas equation of state.

By convention, we replace $\boldsymbol{U}_h^+$ by an appropriate boundary function which realizes the boundary conditions when $\partial K \cap \partial\Omega \neq \emptyset$. For instance, if purely inflow condition $\boldsymbol{U} = \boldsymbol{U}_{\mathrm{D}}$ is imposed on ∂K, then $\boldsymbol{U}_h^+$ is replaced by $\boldsymbol{U}_{\mathrm{D}}$; if purely outflow condition is imposed on ∂K, then set $\boldsymbol{U}_h^+ = \boldsymbol{U}_h^-$; and if reflective boundary condition for fluid–solid interfaces is imposed on ∂K, then set $\boldsymbol{U}_h^+ = [\rho_h^-, \boldsymbol{m}_h^- - 2(\boldsymbol{m}_h^- \cdot \boldsymbol{n}_K)\boldsymbol{n}_K, E_h^-]^{\mathrm{T}}$.

Parabolic subproblem. We use the interior penalty DG method for discretizing (P). For convenience of introducing discrete forms in parabolic subproblem, we partition the boundary of the domain Ω into the union of two disjoint sets, namely $\partial\Omega = \partial\Omega_{\mathrm{D}} \cup \partial\Omega_{\mathrm{N}}$, where the Dirichlet boundary conditions ($\boldsymbol{u} = \boldsymbol{u}_{\mathrm{D}}$ and $e = e_{\mathrm{D}}$) are applied on $\partial\Omega_{\mathrm{D}}$ and the Neumann-type boundary conditions ($\boldsymbol{\tau}(\boldsymbol{u}) \cdot \boldsymbol{n} = \boldsymbol{0}$ and $\nabla e \cdot \boldsymbol{n} = 0$) are applied on $\partial\Omega_{\mathrm{N}}$. Here, $\boldsymbol{n}$ denotes the unit outer normal of domain Ω.

The average and jump operators of any vector quantity $\boldsymbol{u}$ on a boundary face coincide with its trace; and on interior faces they are defined by

$$\{\!\{\boldsymbol{u}\}\!\}|_e = \frac{1}{2}\,\boldsymbol{u}|_{K_{i^-}} + \frac{1}{2}\,\boldsymbol{u}|_{K_{i^+}}\,, \quad [\![\boldsymbol{u}]\!]|_e = \boldsymbol{u}|_{K_{i^-}} - \boldsymbol{u}|_{K_{i^+}}\,, \quad e = \partial K_{i^-} \cap \partial K_{i^+}.$$

The related definitions of any scalar quantity are similar. For more details see [58]. We employ the non-symmetric interior penalty DG (NIPG) method to discretize the terms $-2\nabla \cdot \boldsymbol{\varepsilon}(\boldsymbol{u})$ and $\nabla \cdot ((\nabla \cdot \boldsymbol{u})\boldsymbol{I})$. The associated bilinear forms $a_{\boldsymbol{\varepsilon}}$ and a_λ are defined as follows:

$$\begin{aligned}
a_{\boldsymbol{\varepsilon}}(\boldsymbol{u}, \boldsymbol{\theta}) &= 2\sum_{K\in\mathcal{T}_h}\int_K \boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{\theta}) - 2\sum_{e\in\Gamma_h\cup\partial\Omega_{\mathrm{D}}}\int_e \{\!\{\boldsymbol{\varepsilon}(\boldsymbol{u})\,\boldsymbol{n}_e\}\!\} \cdot [\![\boldsymbol{\theta}]\!] \\
&\quad + 2\sum_{e\in\Gamma_h\cup\partial\Omega_{\mathrm{D}}}\int_e \{\!\{\boldsymbol{\varepsilon}(\boldsymbol{\theta})\,\boldsymbol{n}_e\}\!\} \cdot [\![\boldsymbol{u}]\!] + \frac{\sigma}{h}\sum_{e\in\Gamma_h\cup\partial\Omega_{\mathrm{D}}}\int_e [\![\boldsymbol{u}]\!] \cdot [\![\boldsymbol{\theta}]\!]\,, \\
a_\lambda(\boldsymbol{u}, \boldsymbol{\theta}) &= -\sum_{K\in\mathcal{T}_h}\int_K (\nabla \cdot \boldsymbol{u})(\nabla \cdot \boldsymbol{\theta}) + \sum_{e\in\Gamma_h\cup\partial\Omega_{\mathrm{D}}}\int_e \{\!\{\nabla \cdot \boldsymbol{u}\}\!\}\,[\![\boldsymbol{\theta} \cdot \boldsymbol{n}_e]\!] - \sum_{e\in\Gamma_h\cup\partial\Omega_{\mathrm{D}}}\int_e \{\!\{\nabla \cdot \boldsymbol{\theta}\}\!\}\,[\![\boldsymbol{u} \cdot \boldsymbol{n}_e]\!]\,.
\end{aligned}$$

And the linear form $b_{\boldsymbol{\tau}}$ associated with term $-\nabla \cdot \boldsymbol{\tau}(\boldsymbol{u})$ for the Dirichlet boundary $\partial\Omega_{\mathrm{D}}$ in (9a) is defined by

$$b_{\boldsymbol{\tau}}(\boldsymbol{\theta}) = 2\sum_{e\in\partial\Omega_{\mathrm{D}}}\int_e (\boldsymbol{\varepsilon}(\boldsymbol{\theta})\,\boldsymbol{n}) \cdot \boldsymbol{u}_{\mathrm{D}} + \frac{\sigma}{h}\sum_{e\in\partial\Omega_{\mathrm{D}}}\int_e \boldsymbol{u}_{\mathrm{D}} \cdot \boldsymbol{\theta} - \frac{2}{3}\sum_{e\in\partial\Omega_{\mathrm{D}}}\int_e \nabla \cdot \boldsymbol{\theta}\,(\boldsymbol{u}_{\mathrm{D}} \cdot \boldsymbol{n}).$$

We employ the incomplete interior penalty DG (IIPG) method to discretize the term $-\Delta e$ in (9b). The

bilinear form $a_{\mathcal{D}}$ and the linear form $b_{\mathcal{D}}$ for term $-\Delta e$ are defined as follows:

$$a_{\mathcal{D}}(e,\chi) = \sum_{K\in\mathcal{T}_h}\int_K \nabla e\cdot\nabla\chi - \sum_{e\in\Gamma_h\cup\partial\Omega_{\mathrm{D}}}\int_e \{\!\{\nabla e\cdot \boldsymbol{n}_e\}\!\}\,[\![\chi]\!] + \frac{\tilde{\sigma}}{h}\sum_{e\in\Gamma_h\cup\partial\Omega_{\mathrm{D}}}\int_e [\![e]\!]\,[\![\chi]\!],$$
$$b_{\mathcal{D}}(\chi) = \frac{\tilde{\sigma}}{h}\sum_{e\in\partial\Omega_{\mathrm{D}}}\int_e e_{\mathrm{D}}\chi.$$

For the sake of global conservation of total energy, to discrete term $\boldsymbol{\tau}(\boldsymbol{u}):\nabla\boldsymbol{u} = 2\boldsymbol{\varepsilon}(\boldsymbol{u}):\nabla\boldsymbol{u} - \frac{2}{3}((\nabla\cdot\boldsymbol{u})\mathbf{I}):\nabla\boldsymbol{u}$ in (9b), by using the tensor identity $\boldsymbol{\varepsilon}(\boldsymbol{u}):\nabla\boldsymbol{u} = \boldsymbol{\varepsilon}(\boldsymbol{u}):\boldsymbol{\varepsilon}(\boldsymbol{u})$, the DG forms $b_{\boldsymbol{\varepsilon}}$ and b_λ are designed for terms $2\boldsymbol{\varepsilon}(\boldsymbol{u}):\nabla\boldsymbol{u}$ and $-((\nabla\cdot\boldsymbol{u})\mathbf{I}):\nabla\boldsymbol{u}$, respectively.

$$b_{\boldsymbol{\varepsilon}}(\boldsymbol{u},\chi) = 2\sum_{K\in\mathcal{T}_h}\int_K \boldsymbol{\varepsilon}(\boldsymbol{u}):\boldsymbol{\varepsilon}(\boldsymbol{u})\chi + \frac{\sigma}{h}\sum_{e\in\Gamma_h}\int_e [\![\boldsymbol{u}]\!]\cdot[\![\boldsymbol{u}]\!]\,\{\!|\chi|\!\} + \frac{\sigma}{h}\sum_{e\in\partial\Omega_{\mathrm{D}}}\int_e (\boldsymbol{u}-\boldsymbol{u}_{\mathrm{D}})\cdot(\boldsymbol{u}-\boldsymbol{u}_{\mathrm{D}})\chi,$$
$$b_\lambda(\boldsymbol{u},\chi) = -\sum_{K\in\mathcal{T}_h}\int_K (\nabla\cdot\boldsymbol{u})(\nabla\cdot\boldsymbol{u})\chi.$$

The DG forms above employ penalty parameters σ and $\tilde{\sigma}$. For any $\sigma \geq 0$, the NIPG bilinear form is coercive. In particular, NIPG0 refers to the choice $\sigma = 0$, e.g., the penalty term is removed. The NIPG0 method is convergent for polynomial degrees greater than or equal to two in two dimension [58]. And more importantly, the NIPG0 method eliminates the face penalties, thereby reducing the numerical viscosity. For IIPG method, the penalty $\tilde{\sigma}$ needs to be large enough to achieve coercivity.

2.3. The simple positivity-preserving limiter

The Zhang–Shu limiter [22, 3] is a simple limiter for enforcing positivity of the approximation polynomial on a finite set S when the polynomial cell average is positive. Let $\boldsymbol{U}_K(\boldsymbol{x}) = [\rho_K, \boldsymbol{m}_K, E_K]^{\mathrm{T}}$ be the DG polynomial on cell K. A simplified version of the limiter [5] modifies the DG polynomial $\boldsymbol{U}_K(\boldsymbol{x})$ with the following steps under the assumption that $\overline{\boldsymbol{U}}_K = \frac{1}{|K|}\int_K \boldsymbol{U}_K \in G^\epsilon$.

1. First enforce positivity of density by

$$\widehat{\rho}_K = \theta_\rho(\rho_K - \overline{\rho}_K) + \overline{\rho}_K, \quad \theta_\rho = \min\left\{1, \frac{\overline{\rho}_K - \epsilon}{\overline{\rho}_K - \min\limits_{\boldsymbol{x}_q\in S_K}\rho_K(\boldsymbol{x}_q)}\right\},$$

   where $\overline{\rho}_K$ denotes the cell average of ρ_K on cell K. Notice that $\widehat{\rho}_K$ and ρ_K have the same cell average, and $\widehat{\rho}_K = \rho_K$ if $\min\limits_{\boldsymbol{x}_q\in S_K}\rho_K(\boldsymbol{x}_q) \geq \epsilon$.

2. Define $\widehat{\boldsymbol{U}}_h = [\widehat{\rho}_h, \boldsymbol{m}_h, E_h]^{\mathrm{T}}$ and enforce positivity of internal energy by

$$\widetilde{\boldsymbol{U}}_K = \theta_e(\widehat{\boldsymbol{U}}_K - \overline{\boldsymbol{U}}_K) + \overline{\boldsymbol{U}}_K, \quad \theta_e = \min\left\{1, \frac{\overline{\rho e}_K - \epsilon}{\overline{\rho e}_K - \min\limits_{\boldsymbol{x}_q\in S_K}\rho e_K(\boldsymbol{x}_q)}\right\},$$

   where $\overline{\rho e}_K = \overline{E}_K - \frac{\|\overline{\boldsymbol{m}}_K\|^2}{2\overline{\rho}_K}$ and $\rho e_K(\boldsymbol{x}_q) = E_K(\boldsymbol{x}_q) - \frac{\|\boldsymbol{m}_K(\boldsymbol{x}_q)\|^2}{2\rho_K(\boldsymbol{x}_q)}$. Notice that $\widetilde{\boldsymbol{U}}_K$ has the same cell average, the positivity is implied by the Jensen's inequality satisfied by the concave internal energy function [5].

We refer to [3, 5, 59] on the justification of its high order accuracy.

2.4. The fully discrete scheme

Let $(\cdot,\cdot)$ denote the L^2 inner product over domain Ω evaluated by Gauss quadrature in (H) and $\langle\cdot,\cdot\rangle$ denote the L^2 inner product over domain Ω evaluated by Gauss–Lobatto quadrature in (P).

Given the DG solution $\boldsymbol{U}_h^n$ at time t^n ($n \geq 0$), a schematic flowchart for evolving to time $t^{n+1} = t^n + \Delta t$ is given as:

$$\boldsymbol{U}_h^n \xrightarrow[\text{step size } \frac{\Delta t}{2}]{\text{solve (H)}} \boldsymbol{U}_h^{\mathrm{H}} \xrightarrow{L^2 \text{ proj.}} (\boldsymbol{u}_h^{\mathrm{H}}, e_h^{\mathrm{H}}) \xrightarrow[\text{step size } \Delta t]{\text{solve (P)}} (\boldsymbol{u}_h^{\mathrm{P}}, e_h^{\mathrm{P}}) \xrightarrow{L^2 \text{ proj.}} \boldsymbol{U}_h^{\mathrm{P}} \xrightarrow[\text{step size } \frac{\Delta t}{2}]{\text{solve (H)}} \boldsymbol{U}_h^{n+1},$$

where the optimization-based postprocessing will be applied to $\boldsymbol{U}_h^{\mathrm{P}}$, as will be described in Step 4 below. For any $n \geq 0$, our fully discrete scheme for solving (3) in one step consists of the following steps.

Step 1. Given $\boldsymbol{U}_h^n \in M_h^k \times \mathbf{X}_h^k \times M_h^k$, compute $\boldsymbol{U}_h^{\mathrm{H}} \in M_h^k \times \mathbf{X}_h^k \times M_h^k$ by the DG method (10) with the positivity-preserving SSP Runge–Kutta (8) [3, 5] using step size $\frac{\Delta t}{2}$. After each Runge–Kutta stage, apply the Zhang–Shu positivity-preserving limiter to ensure that all point values at $S_K^{\mathrm{H,int}}$ and $S_K^{\mathrm{H,aux}}$ have positive density and internal energy.

Step 2. Use the Zhang–Shu positivity-preserving limiter to ensure that all point values at S_K^{P} have positive density and internal energy. Given $\boldsymbol{U}_h^{\mathrm{H}} \in M_h^k \times \mathbf{X}_h^k \times M_h^k$, compute $(\boldsymbol{u}_h^{\mathrm{H}}, e_h^{\mathrm{H}}) \in \mathbf{X}_h^k \times M_h^k$ by L^2 projection

$$\langle \boldsymbol{m}_h^{\mathrm{H}}, \boldsymbol{\theta}_h \rangle = \langle \rho_h^{\mathrm{H}} \boldsymbol{u}_h^{\mathrm{H}}, \boldsymbol{\theta}_h \rangle, \quad \forall \boldsymbol{\theta}_h \in \mathbf{X}_h^k \quad \text{and} \quad \langle E_h^{\mathrm{H}}, \chi_h \rangle = \langle \rho_h^{\mathrm{H}} e_h^{\mathrm{H}}, \chi_h \rangle + \langle \frac{\boldsymbol{m}_h^{\mathrm{H}}}{2\rho_h^{\mathrm{H}}}, \boldsymbol{m}_h^{\mathrm{H}} \chi_h \rangle, \quad \forall \chi_h \in M_h^k. \tag{11}$$

Step 3. Given $(\rho_h^{\mathrm{H}}, \boldsymbol{u}_h^{\mathrm{H}}) \in M_h^k \times \mathbf{X}_h^k$, set $\rho_h^{\mathrm{P}} = \rho_h^{\mathrm{H}}$ and solve $(\boldsymbol{u}_h^*, \boldsymbol{u}_h^{\mathrm{P}}) \in \mathbf{X}_h^k \times \mathbf{X}_h^k$, such that for all $\boldsymbol{\theta}_h \in \mathbf{X}_h^k$

$$\langle \rho_h^{\mathrm{P}} \boldsymbol{u}_h^*, \boldsymbol{\theta}_h \rangle + \frac{\Delta t}{2\mathrm{Re}} a_{\boldsymbol{\varepsilon}}(\boldsymbol{u}_h^*, \boldsymbol{\theta}_h) + \frac{\Delta t}{3\mathrm{Re}} a_{\lambda}(\boldsymbol{u}_h^*, \boldsymbol{\theta}_h) = \langle \rho_h^{\mathrm{H}} \boldsymbol{u}_h^{\mathrm{H}}, \boldsymbol{\theta}_h \rangle + \frac{\Delta t}{2\mathrm{Re}} b_{\boldsymbol{\tau}}(\boldsymbol{\theta}_h), \tag{12a}$$

$$\boldsymbol{u}_h^{\mathrm{P}} = 2\boldsymbol{u}_h^* - \boldsymbol{u}_h^{\mathrm{H}}. \tag{12b}$$

Then given $(\rho_h^{\mathrm{H}}, \rho_h^{\mathrm{P}}, \boldsymbol{u}_h^*, e_h^{\mathrm{H}}) \in M_h^k \times M_h^k \times \mathbf{X}_h^k \times M_h^k$, solve for $(e_h^*, e_h^{\mathrm{P}}) \in M_h^k \times M_h^k$, such that for all $\chi_h \in M_h^k$

$$\langle \rho_h^{\mathrm{P}} e_h^*, \chi_h \rangle + \frac{\theta \Delta t \lambda}{\mathrm{Re}} a_{\mathcal{D}}(e_h^*, \chi_h) = \langle \rho_h^{\mathrm{H}} e_h^{\mathrm{H}}, \chi_h \rangle + \frac{\theta \Delta t}{\mathrm{Re}} b_{\boldsymbol{\varepsilon}}(\boldsymbol{u}_h^*, \chi_h) + \frac{2\theta \Delta t}{3\mathrm{Re}} b_{\lambda}(\boldsymbol{u}_h^*, \chi_h) + \frac{\theta \Delta t \lambda}{\mathrm{Re}} b_{\mathcal{D}}(\chi_h), \tag{12c}$$

$$e_h^{\mathrm{P}} = \frac{1}{\theta} e_h^* + (1 - \frac{1}{\theta}) e_h^{\mathrm{H}}. \tag{12d}$$

Step 4. Given $(\rho_h^{\mathrm{P}}, \boldsymbol{u}_h^{\mathrm{P}}, e_h^{\mathrm{P}}) \in M_h^k \times \mathbf{X}_h^k \times M_h^k$, compute $(\boldsymbol{m}_h^{\mathrm{P}}, E_h^{\mathrm{P}}) \in \mathbf{X}_h^k \times M_h^k$ by L^2 projection

$$\langle \boldsymbol{m}_h^{\mathrm{P}}, \boldsymbol{\theta}_h \rangle = \langle \rho_h^{\mathrm{P}} \boldsymbol{u}_h^{\mathrm{P}}, \boldsymbol{\theta}_h \rangle, \quad \forall \boldsymbol{\theta}_h \in \mathbf{X}_h^k \quad \text{and} \quad \langle E_h^{\mathrm{P}}, \chi_h \rangle = \langle \rho_h^{\mathrm{P}} e_h^{\mathrm{P}}, \chi_h \rangle + \langle \frac{\boldsymbol{m}_h^{\mathrm{P}}}{2\rho_h^{\mathrm{P}}}, \boldsymbol{m}_h^{\mathrm{P}} \chi_h \rangle, \quad \forall \chi_h \in M_h^k. \tag{13}$$

Postprocess $\boldsymbol{U}_h^{\mathrm{P}}$ by the constraint optimization-based limiting strategy, see Section 3. Then the cell averages have positive states, and we can apply the Zhang–Shu positivity-preserving limiter to ensure that all point values at $S_K^{\mathrm{H,int}}$ and $S_K^{\mathrm{H,aux}}$ have positive density and internal energy.

Step 5. Given $\boldsymbol{U}_h^{\mathrm{P}} \in M_h^k \times \mathbf{X}_h^k \times M_h^k$, compute $\boldsymbol{U}_h^{n+1} \in M_h^k \times \mathbf{X}_h^k \times M_h^k$ by the DG method (10) with the positivity-preserving SSP Runge–Kutta (8) [3, 5] using step size $\frac{\Delta t}{2}$. After each Runge–Kutta stage, apply the Zhang–Shu positivity-preserving limiter to ensure that all point values at $S_K^{\mathrm{H,int}}$ and $S_K^{\mathrm{H,aux}}$ have positive density and internal energy.

The $\boldsymbol{U}_h^0$ is obtained through the L^2 projection of the initial data $\boldsymbol{U}^0$, followed by postprocessing it with the Zhang–Shu limiter [3]. Thus, $\boldsymbol{U}_h^0$ belongs to the set of admissible states. In addition, we highlight in each time step only two decoupled linear systems (12a) and (12c) need to be solved sequentially.

Remark 1. *For $\mathbb{Q}^k$ scheme, the $\mathbb{Q}^k$ Lagrangian basis functions defined at Gauss–Lobatto points are orthogonal at the $(k+1)^d$-point Gauss–Lobatto quadrature points. Thus, in Step 2 and Step 4, no linear systems need to be solved for computing the L^2 projection.*

2.5. Global conservation of the fully discrete scheme

We first discuss the global conservation of momentum and total energy. Notice that the local conservation for mass is naturally inherited from the Runge–Kutta DG method solving compressible Euler equations. For simplicity, we only discuss conservation in the context of periodic boundary conditions. It is straightforward to extend the discussion to many other types of boundary conditions, such as the ones implemented in the numerical tests in this paper.

The following result is essentially the same as [7, Theorem 1]. However, the time discretization used in this paper is the θ-scheme for the internal energy equation, whereas the time discretization in [7, Theorem 1] is the backward Euler scheme. In addition, the spatial discretization in this paper is a DG scheme, while the spatial discretization in [7] is a combination of DG and continuous finite element method. Thus, for completeness, we include the proof of the global conservation.

Theorem 1. *Assume $\boldsymbol{U}_h^{\mathrm{P}}(\boldsymbol{x}_q)$ belongs to the set of admissible states for all $\boldsymbol{x}_q \in S_h$, then the fully discrete scheme conserves density, momentum, and total energy. We have*

$$(\rho_h^n, 1) = (\rho_h^{n+1}, 1), \quad (\boldsymbol{m}_h^n, \mathbf{1}) = (\boldsymbol{m}_h^{n+1}, \mathbf{1}), \quad (E_h^n, 1) = (E_h^{n+1}, 1).$$

Proof. Both the explicit Runge–Kutta DG method for hyperbolic subproblem (H) and the Zhang–Shu limiter conserve mass, momentum, and total energy [3, 5]. We have

$$(\rho_h^n, 1) = (\rho_h^{\mathrm{H}}, 1), \quad (\boldsymbol{m}_h^n, \mathbf{1}) = (\boldsymbol{m}_h^{\mathrm{H}}, \mathbf{1}), \quad (E_h^n, 1) = (E_h^{\mathrm{H}}, 1).$$

It is easy to verify the discrete mass conservation, since $(\rho_h^{n+1}, 1) = (\rho_h^{\mathrm{P}}, 1)$ and we set $\rho_h^{\mathrm{H}} = \rho_h^{\mathrm{P}}$ in Step 3.

For the discrete momentum conservation, we have $(\boldsymbol{m}_h^n, \mathbf{1}) = (\boldsymbol{m}_h^{\mathrm{H}}, \mathbf{1})$ and $(\boldsymbol{m}_h^{n+1}, \mathbf{1}) = (\boldsymbol{m}_h^{\mathrm{P}}, \mathbf{1})$. For $\mathbb{Q}^k$ scheme, the quadrature rules in subproblems (H) and (P) are both exact for integrating polynomials of degree k, Thus, we also have $(\boldsymbol{m}_h^{\mathrm{H}}, \mathbf{1}) = \langle \boldsymbol{m}_h^{\mathrm{H}}, \mathbf{1} \rangle$ and $(\boldsymbol{m}_h^{\mathrm{P}}, \mathbf{1}) = \langle \boldsymbol{m}_h^{\mathrm{P}}, \mathbf{1} \rangle$. Take $\boldsymbol{\theta}_h = \mathbf{1}$ in (11) and (13), we get $\langle \boldsymbol{m}_h^{\mathrm{H}}, \mathbf{1} \rangle = \langle \rho_h^{\mathrm{H}} \boldsymbol{u}_h^{\mathrm{H}}, \mathbf{1} \rangle$ and $\langle \boldsymbol{m}_h^{\mathrm{P}}, \mathbf{1} \rangle = \langle \rho_h^{\mathrm{P}} \boldsymbol{u}_h^{\mathrm{P}}, \mathbf{1} \rangle$. The above identities indicate $(\boldsymbol{m}_h^n, \mathbf{1}) = \langle \rho_h^{\mathrm{H}} \boldsymbol{u}_h^{\mathrm{H}}, \mathbf{1} \rangle$ and $(\boldsymbol{m}_h^{n+1}, \mathbf{1}) = \langle \rho_h^{\mathrm{P}} \boldsymbol{u}_h^{\mathrm{P}}, \mathbf{1} \rangle$. By selecting $\boldsymbol{\theta}_h = \mathbf{1}$ in (12a), we obtain $\langle \rho_h^{\mathrm{H}} \boldsymbol{u}_h^{\mathrm{H}}, \mathbf{1} \rangle = \langle \rho_h^{\mathrm{P}} \boldsymbol{u}_h^{\mathrm{P}}, \mathbf{1} \rangle$, namely $(\boldsymbol{m}_h^n, \mathbf{1}) = (\boldsymbol{m}_h^{n+1}, \mathbf{1})$ holds.

For the discrete energy conservation, notice the basis are numerically orthogonal and similar to above, we have $(E_h^n, 1) = \langle \rho_h^{\mathrm{H}} e_h^{\mathrm{H}}, 1 \rangle + \frac{1}{2} \langle \rho_h^{\mathrm{H}} \boldsymbol{u}_h^{\mathrm{H}}, \boldsymbol{u}_h^{\mathrm{H}} \rangle$ and $(E_h^{n+1}, 1) = \langle \rho_h^{\mathrm{P}} e_h^{\mathrm{P}}, 1 \rangle + \frac{1}{2} \langle \rho_h^{\mathrm{P}} \boldsymbol{u}_h^{\mathrm{P}}, \boldsymbol{u}_h^{\mathrm{P}} \rangle$. Recall that $b_{\boldsymbol{\tau}}(\boldsymbol{\theta}) = 0$ and $b_{\mathcal{D}}(\chi) = 0$ for periodic boundary conditions, thus by (12b) and $\rho_h^{\mathrm{H}} = \rho_h^{\mathrm{P}}$, the (12a) can be written as

$$\langle \rho_h^{\mathrm{P}} \boldsymbol{u}_h^{\mathrm{P}}, \boldsymbol{\theta}_h \rangle + \frac{\Delta t}{\mathrm{Re}} a_{\boldsymbol{\varepsilon}}(\boldsymbol{u}_h^*, \boldsymbol{\theta}_h) + \frac{2\Delta t}{3\mathrm{Re}} a_{\lambda}(\boldsymbol{u}_h^*, \boldsymbol{\theta}_h) = \langle \rho_h^{\mathrm{H}} \boldsymbol{u}_h^{\mathrm{H}}, \boldsymbol{\theta}_h \rangle.$$

Plugging in $\boldsymbol{\theta}_h = (\boldsymbol{u}_h^{\mathrm{P}} + \boldsymbol{u}_h^{\mathrm{H}})/2 = \boldsymbol{u}_h^*$, we have

$$\frac{1}{2} \langle \rho_h^{\mathrm{P}} \boldsymbol{u}_h^{\mathrm{P}}, \boldsymbol{u}_h^{\mathrm{P}} \rangle + \frac{\Delta t}{\mathrm{Re}} a_{\boldsymbol{\varepsilon}}(\boldsymbol{u}_h^*, \boldsymbol{u}_h^*) + \frac{2\Delta t}{3\mathrm{Re}} a_{\lambda}(\boldsymbol{u}_h^*, \boldsymbol{u}_h^*) = \frac{1}{2} \langle \rho_h^{\mathrm{H}} \boldsymbol{u}_h^{\mathrm{H}}, \boldsymbol{u}_h^{\mathrm{H}} \rangle. \tag{14}$$

Taking $\chi_h = 1$ in (12c), we have

$$\langle \rho_h^{\mathrm{P}} e_h^*, 1 \rangle + \frac{\theta \Delta t \lambda}{\mathrm{Re}} a_{\mathcal{D}}(e_h^*, 1) = \langle \rho_h^{\mathrm{H}} e_h^{\mathrm{H}}, 1 \rangle + \frac{\theta \Delta t}{\mathrm{Re}} b_{\boldsymbol{\varepsilon}}(\boldsymbol{u}_h^*, 1) + \frac{2\theta \Delta t}{3\mathrm{Re}} b_{\lambda}(\boldsymbol{u}_h^*, 1).$$

Recall that $e^* = \theta e^{\mathrm{P}} + (1-\theta) e^{\mathrm{H}}$, we have

$$\langle \rho_h^{\mathrm{P}} e_h^{\mathrm{P}}, 1 \rangle + \frac{\Delta t \lambda}{\mathrm{Re}} a_{\mathcal{D}}(e_h^*, 1) = \langle \rho_h^{\mathrm{H}} e_h^{\mathrm{H}}, 1 \rangle + \frac{\Delta t}{\mathrm{Re}} b_{\varepsilon}(\boldsymbol{u}_h^*, 1) + \frac{2\Delta t}{3\mathrm{Re}} b_{\lambda}(\boldsymbol{u}_h^*, 1). \tag{15}$$

Adding two equations (14) and (15), with the fact that $a_{\mathcal{D}}(e_h^*, 1) = 0$ and the identities $a_{\boldsymbol{\varepsilon}}(\boldsymbol{u}_h^*, \boldsymbol{u}_h^*) = b_{\varepsilon}(\boldsymbol{u}_h^*, 1)$ and $a_{\lambda}(\boldsymbol{u}_h^*, \boldsymbol{u}_h^*) = b_{\lambda}(\boldsymbol{u}_h^*, 1)$, we obtain

$$\langle \rho_h^{\mathrm{H}} e_h^{\mathrm{H}}, 1 \rangle + \frac{1}{2} \langle \rho_h^{\mathrm{H}} \boldsymbol{u}_h^{\mathrm{H}}, \boldsymbol{u}_h^{\mathrm{H}} \rangle = \langle \rho_h^{\mathrm{P}} e_h^{\mathrm{P}}, 1 \rangle + \frac{1}{2} \langle \rho_h^{\mathrm{P}} \boldsymbol{u}_h^{\mathrm{P}}, \boldsymbol{u}_h^{\mathrm{P}} \rangle.$$

Therefore, we obtain $(E_h^n, 1) = (E_h^{n+1}, 1)$. □

3. A globally conservative and positivity-preserving postprocessing procedure

For Runge–Kutta DG method solving the hyperbolic subproblem (H), i.e., compressible Euler equations, it is well understood that the simple Zhang–Shu limiter can preserve the positivity without destroying conservation and high order accuracy [3, 5]. Let S_h be the union of sets $S_K^{\mathrm{H,int}}$ and $S_K^{\mathrm{H,aux}}$ for all $K \in \mathcal{T}_h$. By the results in [3, 5], for Step 1 and Step 5 in the fully discrete scheme in Section 2.4, we have

1. The DG polynomial $\boldsymbol{U}_h^n(\boldsymbol{x}_q) \in G$ for all $\boldsymbol{x}_q \in S_h$ gives $\boldsymbol{U}_h^{\mathrm{H}}(\boldsymbol{x}_q) \in G$ for all $\boldsymbol{x}_q \in S_h$.

2. If $\boldsymbol{U}_h^{\mathrm{P}}(\boldsymbol{x}_q) \in G$ for all $\boldsymbol{x}_q \in S_h$, then the DG polynomial $\boldsymbol{U}_h^{n+1}(\boldsymbol{x}_q) \in G$ for all $\boldsymbol{x}_q \in S_h$.

Moreover, by [7, Lemma 1], the L^2 projection step (11) in Step 2 does not affect the positivity, i.e., the positivity of e_h^{H} is ensured if conserved variables are in the invariant domain. Therefore, in order to construct a conservative and positivity-preserving scheme, we only need to enforce $\boldsymbol{U}_h^{\mathrm{P}}(\boldsymbol{x}_q) \in G^\epsilon$ for all $\boldsymbol{x}_q \in S_h$ in Step 4 without affecting the global conservation in the fully discrete scheme in Section 2.4.

When using the backward Euler time discretization (e.g., $\theta = 1$) in Step 3, positivity can be achieved if the discrete Laplacian is monotone [7]. For example, the discrete Laplacian from $\mathbb{Q}^1$ IIPG forms an M-matrix unconditionally. Moreover, the monotonicity of $\mathbb{Q}^k$ spectral element method (continuous finite element with Gauss–Lobatto quadrature) for $k = 1, 2, 3$ is proven in [42, 43, 44], see also [9, 8, 10], and such a result was used in [7] for solving (3).

To improve the time accuracy, the Crank–Nicolson scheme with $\theta = \frac{1}{2}$ can be used in Step 3. However, in this case, a monotone system matrix no longer implies the positivity of internal energy, which poses a significant challenge, though positivity might still be ensured under a small time step $\Delta t = O(\mathrm{Re}\Delta x^2)$. Instead, we consider a postprocessing procedure based on constraint optimization to ensure global conservation and positivity. The constraint optimization-based cell average limiter can be formulated as a nonsmooth convex minimization problem and efficiently solved by utilizing the generalized Douglas–Rachford splitting method [51].

3.1. A cell average postprocessing approach

By Theorem 1, the DG polynomial $\boldsymbol{U}_h^{\mathrm{P}}$ preserves the global conservation. But it may violate the positivity of internal energy. The following two-stage limiting strategy can be employed to enforce $\boldsymbol{U}_h^{\mathrm{P}}(\boldsymbol{x}_q)$ in the set of admissible states for any quadrature points $\boldsymbol{x}_q \in S_h$ without losing high order accuracy and conservation.

Step 1. Given $\boldsymbol{U}_h^{\mathrm{P}}$, if any cell average has negative internal energy, then post process all cell averages of the total energy variable without losing global conservation such that each cell average of the DG polynomial $\boldsymbol{U}_h^{\mathrm{P}}$ stays in the admissible state set G^ϵ.

Step 2. Apply the Zhang–Shu limiter to the postpocessed DG polynomial to ensure internal energy at any quadrature points in S_h is positive.

For a postprocessing procedure, minimal modifications to the original DG polynomial is often preferred. In our scheme, the density $\rho_h^{\mathrm{P}} = \rho_h^{\mathrm{H}}$ is already positive, ensured by a high order accurate positivity-preserving compressible Euler solver. Consider the scheme for solving the subproblem (P), which is fully decoupled. The momentum $\boldsymbol{m}_h^{\mathrm{P}}$ or velocity $\boldsymbol{u}_h^{\mathrm{P}}$ is stably approximated. With the given ρ_h^{P} and $\boldsymbol{u}_h^{\mathrm{P}}$, when solving (5c), which is a heat equation in the parabolic subproblem, a high order scheme may not preserve positivity in general. To this end, we consider a simple approach by only post processing the total energy variable E_h^{P} to enforce the positivity of internal energy, without losing conservation for E_h^{P}.

Let K_i $(i = 1, \cdots, N)$ be all the cells and $\overline{\boldsymbol{U}_i^{\mathrm{P}}} = [\overline{\rho_i^{\mathrm{P}}}, \overline{\boldsymbol{m}_i^{\mathrm{P}}}, \overline{E_i^{\mathrm{P}}}]^{\mathrm{T}}$ be a vector denoting the cell average of the DG polynomial $\overline{\boldsymbol{U}_h^{\mathrm{P}}}$ on the i-th cell K_i, namely $\overline{\boldsymbol{U}_i^{\mathrm{P}}} = \frac{1}{|K_i|}\int_{K_i} \boldsymbol{U}_h^{\mathrm{P}}$.

Then we apply the globally conservative postprocessing procedure (6) only to the total energy DG polynomial such that the modified DG polynomials have good cell averages, which have positive internal energy.

3.2. The accuracy of the postprocessing

It is obvious that the minimizer to (6a) preserves the global conservation of total energy and the positivity of internal energy, since these two are the constraints. Next, we discuss the accuracy of the postprocessing step (6a).

To understand how (6a) affects accuracy, consider evolving (5c) with given $\rho(\boldsymbol{x},t)=\rho_h^{\mathrm{P}}(\boldsymbol{x})$ and $\boldsymbol{u}(\boldsymbol{x},t)=\boldsymbol{u}_h^*(\boldsymbol{x})$, $\forall t$ by one time step in the Strang splitting (7), i. e., we consider the initial value problem

$$\begin{cases} \rho_h^{\mathrm{P}}\partial_t e-\frac{\lambda}{\mathrm{Re}}\Delta e=\frac{1}{\mathrm{Re}}\boldsymbol{\tau}(\boldsymbol{u}_h^*):\boldsymbol{\nabla}\boldsymbol{u}_h^*, \quad t\in(t^n,t^n+\Delta t),\\ e(\boldsymbol{x},t^n)=e_h^{\mathrm{H}}(\boldsymbol{x}). \end{cases} \tag{16}$$

Due to the inequality $\boldsymbol{\varepsilon}(\boldsymbol{u}):\boldsymbol{\varepsilon}(\boldsymbol{u})\geq\frac{1}{d}(\boldsymbol{\nabla}\cdot\boldsymbol{u})^2$, which can be easily verified by calculations (e.g., for $d=2$, $\boldsymbol{\varepsilon}(\boldsymbol{u}):\boldsymbol{\varepsilon}(\boldsymbol{u})\geq\frac{1}{2}(\boldsymbol{\nabla}\cdot\boldsymbol{u})^2\Leftrightarrow\frac{1}{4}(u_x-u_y)^2+\frac{1}{2}(u_x+u_y)^2\geq 0$), we know $\boldsymbol{\tau}(\boldsymbol{u}_h^*):\boldsymbol{\nabla}\boldsymbol{u}_h^*=2\big(\boldsymbol{\varepsilon}(\boldsymbol{u}_h^*):\boldsymbol{\varepsilon}(\boldsymbol{u}_h^*)-\frac{1}{3}(\boldsymbol{\nabla}\cdot\boldsymbol{u}_h^*)^2\big)\geq 0$. We mention that a similar property also holds for the interior penalty DG scheme at the discrete level, i. e., the right hand side of (12c) is also positive, see [7, Lemma 3]. Let e denote the exact solution to (16). Since the right-hand side of (16) is non-negative, the exact solution to (16) with an initial condition $e_h^{\mathrm{H}}>0$ is positive, thus we assume $e(\boldsymbol{x},t)\geq\epsilon_2>0$.

Noticing that ρ_h^{P} is time independent, we have $\rho_h^{\mathrm{P}}\partial_t e=\partial_t(\rho_h^{\mathrm{P}}e)$. Integrating (16) over the spatial domain Ω and using boundary condition $\boldsymbol{\nabla}e\cdot\boldsymbol{n}=0$, we get

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\int_\Omega\rho_h^{\mathrm{P}}e\,\mathrm{d}\boldsymbol{x}\right)=\frac{1}{\mathrm{Re}}\int_\Omega\boldsymbol{\tau}(\boldsymbol{u}_h^*):\boldsymbol{\nabla}\boldsymbol{u}_h^*\mathrm{d}\boldsymbol{x}.$$

Integrating the equation above over the time interval $[t^n,t^n+\Delta t]$, we have

$$\int_\Omega\rho_h^{\mathrm{P}}(\boldsymbol{x})e(\boldsymbol{x},t^n+\Delta t)\mathrm{d}\boldsymbol{x}=\int_\Omega\rho_h^{\mathrm{H}}e_h^{\mathrm{H}}\mathrm{d}\boldsymbol{x}+\frac{\Delta t}{\mathrm{Re}}\int_\Omega\boldsymbol{\tau}(\boldsymbol{u}_h^*):\boldsymbol{\nabla}\boldsymbol{u}_h^*\mathrm{d}\boldsymbol{x}. \tag{17}$$

Consider the NIPG0 method for velocity, i. e., the NIPG method with zero penalty, which is the scheme (12c) we utilized in our numerical experiments. Recall $(k+1)^d$ Gauss–Lobatto quadrature is accurate for $(2k-1)$-order polynomial. Taking $\chi_h=1$ in (12c), with (17) and the quadrature error for integrals, we have

$$\int_\Omega\rho_h^{\mathrm{P}}(\boldsymbol{x})e(\boldsymbol{x},t^n+\Delta t)\mathrm{d}\boldsymbol{x}=\langle\rho_h^{\mathrm{P}}e_h^{\mathrm{P}},1\rangle+Ch^{2k}.$$

Let $e_I(\boldsymbol{x})$ be the piecewise $\mathbb{Q}^k$ interpolation polynomial of the exact solution $e(\boldsymbol{x},t^n+\Delta t)$ at $(k+1)^d$ Gauss–Lobatto points at each cell. We have

$$\langle\rho_h^{\mathrm{P}}e_I,1\rangle=\int_\Omega\rho_h^{\mathrm{P}}(\boldsymbol{x})e(\boldsymbol{x},t^n+\Delta t)\mathrm{d}\boldsymbol{x}+Ch^{2k}=\langle\rho_h^{\mathrm{P}}e_h^{\mathrm{P}},1\rangle+Ch^{2k}.$$

Let $\tilde{e}_h(\boldsymbol{x})=e_I(\boldsymbol{x})-\frac{C}{\langle\rho_h^{\mathrm{P}},1\rangle}h^{2k}$, then $\tilde{e}_h(\boldsymbol{x})=e(\boldsymbol{x})+O(h^{k+1})$ and $\langle\rho_h^{\mathrm{P}}\tilde{e}_h,1\rangle=\langle\rho_h^{\mathrm{P}}e_h^{\mathrm{P}},1\rangle$. Define $(\boldsymbol{m}_h^{\mathrm{P}},E_h^{\mathrm{Interp}})\in\mathbf{X}_h^k\times M_h^k$ as an L^2 projection of $(\rho_h^{\mathrm{P}},\boldsymbol{u}_h^{\mathrm{P}},\tilde{e}_h)\in M_h^k\times\mathbf{X}_h^k\times M_h^k$:

$$\langle\boldsymbol{m}_h^{\mathrm{P}},\boldsymbol{\theta}_h\rangle=\langle\rho_h^{\mathrm{P}}\boldsymbol{u}_h^{\mathrm{P}},\boldsymbol{\theta}_h\rangle,\quad\forall\boldsymbol{\theta}_h\in\mathbf{X}_h^k\quad\text{and}\quad\langle E_h^{\mathrm{Interp}},\chi_h\rangle=\langle\rho_h^{\mathrm{P}}\tilde{e}_h,\chi_h\rangle+\langle\frac{\boldsymbol{m}_h^{\mathrm{P}}}{2\rho_h^{\mathrm{P}}},\boldsymbol{m}_h^{\mathrm{P}}\chi_h\rangle,\quad\forall\chi_h\in M_h^k. \tag{18}$$

Notice that $\boldsymbol{m}_h^{\mathrm{P}}$ in (18) is exactly the same as $\boldsymbol{m}_h^{\mathrm{P}}$ in (13), and only E_h^{Interp} is different.

Let $\overline{E_i^{\mathrm{Interp}}}$ be the cell average of E_h^{Interp} at the i-th cell and $\overline{E_h^{\mathrm{Interp}}}=[\overline{E_1^{\mathrm{Interp}}},\overline{E_2^{\mathrm{Interp}}},\cdots,\overline{E_N^{\mathrm{Interp}}}]^{\mathrm{T}}$. Next, we verify that $\overline{E_h^{\mathrm{Interp}}}$ satisfies both constraints in (6a), when the mesh size h is small.

- First, by taking $\chi_h = 1$ in (13) and (18), we obtain the global conservation of total energy:

$$\sum_{i=1}^{N} \overline{E_i^{\mathrm{Interp}}}|K_i| = \langle E_h^{\mathrm{Interp}}, 1\rangle = \langle E_h^{\mathrm{P}}, 1\rangle = \sum_{i=1}^{N} \overline{E_i^{\mathrm{P}}}|K_i|.$$

- Second, for small enough h such that $\frac{|C|}{\langle \rho_h^{\mathrm{P}}, 1\rangle} h^{2k} \le \frac{1}{2}\epsilon_2$, we can take $\epsilon \le \frac{1}{2}\epsilon_2 \rho_h^{\mathrm{P}}$ to have

$$\left(\epsilon_2 - \frac{|C|}{\langle \rho_h^{\mathrm{P}}, 1\rangle} h^{2k}\right) \rho_h^{\mathrm{P}} \ge \frac{1}{2}\epsilon_2 \rho_h^{\mathrm{P}} \ge \epsilon.$$

Then following the proof of Lemma 2 in [7, Section 3.2], we have

$$\overline{E_i^{\mathrm{Interp}}} - \frac{1}{2}\frac{\|\overline{\boldsymbol{m}_i^{\mathrm{P}}}\|}{\overline{\rho_i^{\mathrm{P}}}} \ge \epsilon.$$

Since $\overline{E}_h^*$ is the minimizer to (6a) and $[\overline{\rho^{\mathrm{P}}}_i, \overline{\boldsymbol{m}^{\mathrm{P}}}_i, \overline{E_i^{\mathrm{Interp}}}]^{\mathrm{T}}$ satisfies the constraints of (6a), we have

$$\left\|\overline{E}_h^* - \overline{E_h^{\mathrm{Interp}}}\right\| \le \left\|\overline{E}_h^* - \overline{E_h^{\mathrm{P}}}\right\| + \left\|\overline{E_h^{\mathrm{P}}} - \overline{E_h^{\mathrm{Interp}}}\right\| \le 2\left\|\overline{E_h^{\mathrm{P}}} - \overline{E_h^{\mathrm{Interp}}}\right\|. \tag{19}$$

To summarize the discussion for accuracy, we conclude that the accuracy of the postprocessing (6a) can be understood in the sense of (19). In other words, if considering the error approximating the exact solution of (16) in Strang splitting, then the minimizer to (6a) is not significantly worse than the DG solution E_h^{P}.

3.3. An efficient solver by Douglas–Rachford splitting with nearly optimal parameters

The key computational issue here is how to solve (6a) efficiently, and the same approach in [51] can be used. For completeness, we briefly describe the main algorithm and result in [51]. For convenience, we rewrite the minimization problem (6a) in matrix-vector form using different names for variables.

For simplicity, we only consider a uniform mesh with $|K_i| = h^d$. Extensions to non-uniform meshes are straightforward. Thus we define a matrix $\mathbf{A} = [1, 1, \cdots, 1] \in \mathbb{R}^{1\times N}$, where N is the total number of cells. A vector $\boldsymbol{w} \in \mathbb{R}^N$ is introduced to store the cell averages of DG polynomial E_h^{P}, namely the i^{th} entry of $\boldsymbol{w}$ equals $\overline{E_i^{\mathrm{P}}}$. Define the constant $b = \mathbf{A}\boldsymbol{w}$, which is the summation of all cell averages. The indicator function in constraint optimization is defined as ι_Λ for a set Λ: $\iota_\Lambda(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \in \Lambda$ and $\iota_\Lambda(\boldsymbol{x}) = +\infty$ if $\boldsymbol{x} \notin \Lambda$. Then (6a) is equivalent to the following minimization:

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} \frac{\alpha}{2}\|\boldsymbol{x} - \boldsymbol{w}\|_2^2 + \iota_{\Lambda_1}(\boldsymbol{x}) + \iota_{\Lambda_2}(\boldsymbol{x}). \tag{20}$$

where $\alpha > 0$ is a constant, and the conservation constraint and the positivity-preserving constraint give two sets

$$\Lambda_1 = \{\boldsymbol{x} : \mathbf{A}\boldsymbol{x} = b\} \quad \text{and} \quad \Lambda_2 = \{\boldsymbol{x} : x_i - \frac{\|\overline{\boldsymbol{m}}_i\|^2}{2\overline{\rho}_i} \ge \epsilon,\ \forall i = 1, \cdots, N\}.$$

Splitting algorithms naturally arise when solving minimization problem of the form $\min_{\boldsymbol{x}} f(\boldsymbol{x}) + g(\boldsymbol{x})$, where functions f and g are convex, lower semi-continuous (but not otherwise smooth), and have simple subdifferentials and resolvents. Let $F = \partial f$ and $G = \partial g$ denote the subdifferentials of f and g. Then, a sufficient and necessary condition for $\boldsymbol{x}$ being a minimizer is $\mathbf{0} \in F(\boldsymbol{x}) + G(\boldsymbol{x})$. The resolvents $\mathrm{J}_{\gamma F} = (\mathrm{I} + \gamma F)^{-1}$ and $\mathrm{J}_{\gamma G} = (\mathrm{I} + \gamma G)^{-1}$ are also called proximal operators, as $\mathrm{J}_{\gamma F}$ maps $\boldsymbol{x}$ to $\mathrm{argmin}_{\boldsymbol{z}} \gamma f(\boldsymbol{z}) + \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{x}\|_2^2$ and

$\mathrm{J}_{\gamma G}$ is defined similarly. The reflection operators are defined as $\mathrm{R}_{\gamma F} = 2\mathrm{J}_{\gamma F} - \mathrm{I}$ and $\mathrm{R}_{\gamma G} = 2\mathrm{J}_{\gamma G} - \mathrm{I}$, where I is the identity operator.

The generalized Douglas–Rachford splitting method for solving the minimization problem $\min_{\boldsymbol{x}} f(\boldsymbol{x}) + g(\boldsymbol{x})$ can be written as:

$$\begin{cases} \boldsymbol{y}^{k+1} = \lambda \dfrac{\mathrm{R}_{\gamma F}\mathrm{R}_{\gamma G} + \mathrm{I}}{2} \boldsymbol{y}^k + (1-\lambda)\boldsymbol{y}^k, \\ \boldsymbol{x}^{k+1} = \mathrm{J}_{\gamma G}(\boldsymbol{y}^{k+1}), \end{cases} \tag{21}$$

where $\boldsymbol{y}$ is an auxiliary variable, λ belongs to $(0, 2]$ is a parameter, and $\gamma > 0$ is step size. We get the Douglas–Rachford splitting when $\lambda = 1$ in (21). In the limiting case, $\lambda = 2$ is the Peaceman–Rachford splitting. For two convex functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$, the sequence in (21) converges for any positive step size γ and any fixed $\lambda \in (0, 2)$, see [52]. If one function is strongly convex, then $\lambda = 2$ also leads to convergence. Using the definition of reflection operators, (21) can be expressed as follows:

$$\begin{cases} \boldsymbol{y}^{k+1} = \lambda \mathrm{J}_{\gamma F}(2\boldsymbol{x}^k - \boldsymbol{y}^k) + \boldsymbol{y}^k - \lambda \boldsymbol{x}^k, \\ \boldsymbol{x}^{k+1} = \mathrm{J}_{\gamma G}(\boldsymbol{y}^{k+1}). \end{cases} \tag{22}$$

We split the objective function in (20) into

$$f(\boldsymbol{x}) = \frac{\alpha}{2}\|\boldsymbol{x} - \boldsymbol{w}\|^2 + \iota_{\Lambda_1}(\boldsymbol{x}) \quad \text{and} \quad g(\boldsymbol{x}) = \iota_{\Lambda_2}(\boldsymbol{x}).$$

Linearity implies that the set Λ_1 is convex. With ideal gas equation of state, the function ρe is concave, see [3, 5] and references therein. Thus, by Jensen's inequality, the set Λ_2 is also convex. Therefore, the function f is strongly convex and the function g is convex, given that (22) converges to the unique minimizer. After applying (22) to solve the minimization to machine precision, the positivity constraint is strictly satisfied and the conservation constraint is enforced up to round-off error. The subdifferentials and the associated resolvents are given as follows:

- The subdifferential of function f is

$$\partial f(\boldsymbol{x}) = \alpha(\boldsymbol{x} - \boldsymbol{w}) + \mathcal{R}(\mathbf{A}^{\mathrm{T}}),$$

  where $\mathcal{R}(\mathbf{A}^{\mathrm{T}})$ denotes the range of the matrix $\mathbf{A}^{\mathrm{T}}$.

- The subdifferential of function g is

$$[\partial g(\boldsymbol{x})]_i = \begin{cases} 0, & \text{if } x_i > \dfrac{\|\overline{\boldsymbol{m}}_i\|^2}{2\overline{\rho}_i} + \epsilon, \\ [-\infty, 0], & \text{if } x_i = \dfrac{\|\overline{\boldsymbol{m}}_i\|^2}{2\overline{\rho}_i} + \epsilon. \end{cases}$$

- For the function $f(\boldsymbol{x}) = \frac{\alpha}{2}\|\boldsymbol{x} - \boldsymbol{w}\|_2^2 + \iota_{\Lambda_1}(\boldsymbol{x})$, the associated resolvent is

$$\mathrm{J}_{\gamma F}(\boldsymbol{x}) = \frac{1}{\gamma\alpha + 1}\left(\mathbf{A}^{+}(b - \mathbf{A}\boldsymbol{x}) + \boldsymbol{x}\right) + \frac{\gamma\alpha}{\gamma\alpha + 1}\boldsymbol{w}, \tag{23}$$

  where $\mathbf{A}^{+} = \mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{A}^{\mathrm{T}})^{-1}$ denotes the pseudo inverse of the matrix $\mathbf{A}$.

- For the function $g(\boldsymbol{x}) = \iota_{\Lambda_2}(\boldsymbol{x})$, the associated resolvent is $\mathrm{J}_{\gamma G}(\boldsymbol{x}) = \mathrm{S}(\boldsymbol{x})$, where S is a cut-off operator defined by

$$[\mathrm{S}(\boldsymbol{x})]_i = \max\left(x_i, \frac{\|\overline{\boldsymbol{m}}_i\|^2}{2\overline{\rho}_i} + \epsilon\right), \quad \forall i = 1, \cdots, N. \tag{24}$$

Define parameter $c = \frac{1}{\gamma\alpha+1}$, which gives $\frac{\gamma\alpha}{\gamma\alpha+1} = 1-c$. Using the expressions of resolvents in (23) and (24), we obtain the generalized Douglas–Rachford splitting method for solving the minimization problem (20) in matrix-vector form:

$$\begin{cases} \boldsymbol{z}^k = 2\boldsymbol{x}^k - \boldsymbol{y}^k, \\ \boldsymbol{y}^{k+1} = \lambda c\big(\mathbf{A}^+(b - \mathbf{A}\boldsymbol{z}^k) + \boldsymbol{z}^k\big) + \lambda(1-c)\boldsymbol{w} + \boldsymbol{y}^k - \lambda\boldsymbol{x}^k, \\ \boldsymbol{x}^{k+1} = \mathrm{S}(\boldsymbol{y}^{k+1}). \end{cases} \tag{25}$$

As a brief summary, after obtaining the DG polynomial E_h^{P}, compute cell averages to generate vector $\boldsymbol{w}$, where the i^{th} entry of $\boldsymbol{w}$ equals $\overline{E_h^{\mathrm{P}}}|_{K_i}$, then our cell average limiter can be implemented as follows.

Algorithm DR. To start the generalized Douglas–Rachford iteration, set $\boldsymbol{y}^0 = \boldsymbol{w}$, $\boldsymbol{x}^0 = \mathrm{S}(\boldsymbol{w})$, and $k = 0$. Compute parameters c and λ by using formula in Remark 2, and select a small ϵ for numerical tolerance of the conservation error.

Step 1. Compute intermediate variable $\boldsymbol{z}^k = 2\boldsymbol{x}^k - \boldsymbol{y}^k$.

Step 2. Compute auxiliary variable $\boldsymbol{y}^{k+1} = \lambda c\big(\mathbf{A}^+(b - \mathbf{A}\boldsymbol{z}^k) + \boldsymbol{z}^k\big) + \lambda(1-c)\boldsymbol{w} + \boldsymbol{y}^k - \lambda\boldsymbol{x}^k$.

Step 3. Compute $\boldsymbol{x}^{k+1} = \mathrm{S}(\boldsymbol{y}^{k+1})$.

Step 4. It is convenient to employ the norm $\|\cdot\|_h = h^{d/2}\|\cdot\|$ to measure the conservation error, which is an approximation to the L^2-norm. If stopping criterion $\|\boldsymbol{y}^{k+1} - \boldsymbol{y}^k\|_h < \epsilon$ is satisfied, then terminate and output $\boldsymbol{x}^* = \boldsymbol{x}^{k+1}$, otherwise set $k \leftarrow k+1$ and go to Step 1.

In the algorithm above, $2\boldsymbol{x}^k$ can be regarded as $\boldsymbol{x}^k + \boldsymbol{x}^k$; the $\lambda(1-c)\boldsymbol{w}$ remains unchanged during iteration; and each entry of $\mathbf{A}^+(b - \mathbf{A}\boldsymbol{z}^k) + \boldsymbol{z}^k$ can be computed by $z_i^k + \frac{1}{N}(b - \sum_i z_i^k)$, thus if only counting number of computing multiplications and taking maximum, the computational complexity of each iteration is $3N+1$.

Remark 2. *The analysis in [51] proves the asymptotic linear convergence and suggests a simple choice of nearly optimal parameters c and λ in* (25)*. Let $\hat{r}$ be the number of bad cells defined by $\overline{\boldsymbol{U}_i^{\mathrm{P}}} \notin G^\epsilon$ and let $\hat{\theta} = \cos^{-1}\sqrt{\frac{\hat{r}}{N}}$, then we have:*

$$\begin{cases} c = \frac{1}{2},\ \lambda = \frac{4}{2-\cos(2\hat{\theta})}, & \text{if } \hat{\theta} \in (\frac{3}{8}\pi, \frac{1}{2}\pi], \\ c = \frac{1}{(\cos\hat{\theta}+\sin\hat{\theta})^2},\ \lambda = \frac{2}{1+\frac{1}{1+\cot\hat{\theta}} - \frac{1}{(\cos\hat{\theta}+\sin\hat{\theta})^2}}, & \text{if } \hat{\theta} \in (\frac{1}{4}\pi, \frac{3}{8}\pi], \\ c = \frac{1}{(\cos\hat{\theta}+\sin\hat{\theta})^2},\ \lambda = 2, & \text{if } \hat{\theta} \in (0, \frac{1}{4}\pi]. \end{cases} \tag{26}$$

Remark 3. *By splitting the Navier–Stokes system into the Euler system and a parabolic system, to preserve positivity, we may postprocess only a scalar variable, i.e, the total energy, which is the main advantage of the splitting approach. It is also possible to postprocess the DG solutions to the convection-diffusion Navier-Stokes system, by a similar Douglas–Rachford cell average limiter to preserve the invariant domain G^ϵ, for which the operator $\mathrm{J}_{\gamma G}$ in* (21) *becomes the projection to admissible set G^ϵ.*

Remark 4. *Compared to other alternative methods for solving* (20) *such as breakpoint searching algorithms [57] and the method of Lagrangian multipliers in the Appendix, the Douglas–Rachford algorithm* (21) *is more flexible for other minimization models such as replacing the ℓ^2-norm in* (20) *by the ℓ^1-norm, for which the Shrinkage operator would appear in $\mathrm{J}_{\gamma F}$.*

3.4. Implementation

We provide details on implementing our scheme. The time-stepping strategy employed to solve subproblem (H) is identical to the one described in Section 3.2 of [60]. For the sake of completeness, we include a list of the steps below.

Algorithm H. At time t^n, select a trial hyperbolic step size Δt^{H}. The parameter ϵ is a prescribed small positive number for numerical admissible state set G^ϵ. The input DG polynomial $\boldsymbol{U}_h^n$ satisfies $\boldsymbol{U}_h^n(\boldsymbol{x}_q) \in G^\epsilon$, for all $\boldsymbol{x}_q \in S_h$.

Step H1. Given DG polynomial $\boldsymbol{U}_h^n$, compute the first stage to obtain $\boldsymbol{U}_h^{(1)}$.

- If the cell averages $\overline{\boldsymbol{U}}_K^{(1)} \in G^\epsilon$, for all $K \in \mathcal{T}_h$, then apply Zhang–Shu limiter described in Section 2.3 to obtain $\widetilde{\boldsymbol{U}}_h^{(1)}$ and go to Step H2.
- Otherwise, recompute the first stage with halved step size $\Delta t^{\mathrm{H}} \leftarrow \frac{1}{2}\Delta t^{\mathrm{H}}$. Notice, when Δt^{H} satisfies the positivity-preserving hyperbolic CFL proven in [3] (see also [5]), the $\overline{\boldsymbol{U}}_K^{(1)} \in G^\epsilon$ is guaranteed.

Step H2. Given DG polynomial $\widetilde{\boldsymbol{U}}_h^{(1)}$, compute the second stage to obtain $\boldsymbol{U}_h^{(2)}$.

- If the cell averages $\overline{\boldsymbol{U}}_K^{(2)} \in G^\epsilon$, for all $K \in \mathcal{T}_h$, then apply Zhang–Shu limiter to obtain $\widetilde{\boldsymbol{U}}_h^{(2)}$ and go to Step H3.
- Otherwise, return to Step H1 and restart the computation with halved step size $\Delta t^{\mathrm{H}} \leftarrow \frac{1}{2}\Delta t^{\mathrm{H}}$. Notice that the results proven in [3] ensure that there is not an infinite restarting loop, see [5].

Step H3. Given DG polynomial $\widetilde{\boldsymbol{U}}_h^{(2)}$, compute the third stage to obtain $\boldsymbol{U}_h^{(3)}$.

- If the cell averages $\overline{\boldsymbol{U}}_K^{(3)} \in G^\epsilon$, for all $K \in \mathcal{T}_h$, then apply Zhang–Shu limiter to obtain $\boldsymbol{U}_h^{\mathrm{H}}$. We finish the current SSP Runge–Kutta.
- Otherwise, return to Step H1 and restart the computation with halved step size $\Delta t^{\mathrm{H}} \leftarrow \frac{1}{2}\Delta t^{\mathrm{H}}$. Notice that the results proven in [3] ensure that there is not an infinite restarting loop, see [5].

The time-stepping strategy for solving the compressible NS equations is as follows. The initial condition $\boldsymbol{U}_h^0$ is constructed by L^2 projection of $\boldsymbol{U}^0$ with Zhang–Shu limiter on S_h, e.g., we have $\boldsymbol{U}_h^0(\boldsymbol{x}_q) \in G^\epsilon$, for all $\boldsymbol{x}_q \in S_h$.

Algorithm CNS. At time t^n, select a desired time step size Δt. The parameter ϵ is a prescribed small positive number for numerical admissible state set G^ϵ. The input DG polynomial $\boldsymbol{U}_h^n$ satisfies $\boldsymbol{U}_h^n(\boldsymbol{x}_q) \in G^\epsilon$, for all $\boldsymbol{x}_q \in S_h$.

Step CNS1. Given DG polynomial $\boldsymbol{U}_h^n$, solve subproblem (H) form time t^n to $t^n + \frac{\Delta t}{2}$.

- Set $m = 0$. Let $t^{n,0} = t^n$ and $\boldsymbol{U}_h^{n,0} = \boldsymbol{U}_h^n$.
- Given $\boldsymbol{U}_h^{n,m}$ at time $t^{n,m}$, solve (H) to compute $\boldsymbol{U}_h^{n,m+1}$ by the Algorithm H. Let $t^{n,m+1} = t^{n,m} + \Delta t^{\mathrm{H}}$. If $t^{n,m+1} = t^n + \frac{\Delta t}{2}$, then apply Zhang–Shu limiter for $\boldsymbol{U}_h^{n,m+1}$ on all Gauss–Lobatto points in S_K^{P}, for all $K \in \mathcal{T}_h$, we obtain $\boldsymbol{U}_h^{\mathrm{H}}$. Go to Step CNS2. Otherwise, set $m \leftarrow m + 1$ and repeat solving (H) by Algorithm H until reaching $t^n + \frac{\Delta t}{2}$. Let L be the smallest integer satisfying $2L - 3 \geq k$ for $\mathbb{Q}^k$ basis, when using $\mathbb{Q}^k$ DG method to compute $\boldsymbol{U}_h^{n,m+1}$, we can take

$$\Delta t^{\mathrm{H}} = \min\left\{ a \frac{1}{\max_e \alpha_e} \frac{1}{L(L-1)} \Delta x,\ t^n + \frac{\Delta t}{2} - t^{n,m} \right\}$$

  as a trial hyperbolic step size to start Algorithm H. We refer to [5] for choosing the value of parameter a on above.

Step CNS2. Given DG polynomial $\boldsymbol{U}_h^{\mathrm{H}}$, take L^2 projection to compute $(\boldsymbol{u}_h^{\mathrm{H}}, e_h^{\mathrm{H}})$.

Step CNS3. Given DG polynomials $(\rho_h^{\mathrm{H}}, \boldsymbol{u}_h^{\mathrm{H}}, e_h^{\mathrm{H}})$, solve subproblem (P) form time t^n to $t^n + \Delta t$.

Step CNS4. Given DG polynomials $(\rho_h^{\mathrm{P}}, \boldsymbol{u}_h^{\mathrm{P}}, e_h^{\mathrm{P}})$, take L^2 projection to compute $\boldsymbol{U}_h^{\mathrm{P}}$.

- Notice that the postprocessing (6) can be applied to either the whole computational domain or a large enough local region containing negative cells. When possible, first define a local region of trouble cells defined by $\overline{\boldsymbol{U}_i^{\mathrm{P}}} \notin G^\epsilon$. Let $T \subseteq \{1, 2, \cdots, N\}$ be the indices of the local region containing all cells with negative averages $\overline{\boldsymbol{U}_i^{\mathrm{P}}} \notin G^\epsilon$, and let $|T|$ be the number of cells in the local region marked by indices in the set T. Then the postprocessing on the local region is given by

$$\min_{\overline{E}_i} \sum_{i\in T} \left|\overline{E}_i - \overline{E_i^{\mathrm{P}}}\right|^2 \text{ subjects to } \sum_{i\in T} \overline{E}_i|K_i| = \sum_{i\in T} \overline{E_i^{\mathrm{P}}}|K_i| \quad \text{and} \quad [\overline{\rho_i^{\mathrm{P}}}, \overline{\boldsymbol{m}_i^{\mathrm{P}}}, \overline{E}_i]^{\mathrm{T}} \in G^\epsilon, \ \forall i \in T. \tag{27a}$$

  Let $\overline{E}_h^* = [\overline{E}_1^*, \cdots, \overline{E}_N^*]^{\mathrm{T}}$ be the minimizer. Then we correct the DG polynomial cell averages for the total energy variable by a constant

$$E_i(\boldsymbol{x}) = E_i^{\mathrm{P}}(\boldsymbol{x}) - \overline{E_i^{\mathrm{P}}} + \overline{E}_i^*, \quad \forall i \in T. \tag{27b}$$

  Notice that T cannot contain only the negative cells, which will cause the feasible set in (27a) to be empty, i.e., it is impossible to modify only negative cells to achieve positivity, without affecting conservation. If it is difficult to define such a set T, we can simply take $T = \{1, 2, \cdots, N\}$, i.e., the whole computational domain. For certain problems, it is straightforward to define a proper T, see the remark below.
- Solve (27a) for the region defined by indices in T by the Douglas–Rachford splitting algorithm (25) with nearly optimal parameters (26) using $\hat{\theta} = \cos^{-1}\sqrt{\frac{\hat{r}}{|T|}}$. Then update or postprocess the cell averages of the DG polynomial $\boldsymbol{U}_h^{\mathrm{P}}$ by (27b).
- With positive cell averages $\overline{\boldsymbol{U}_i^{\mathrm{P}}} \in G^\epsilon$ ensured by the postprocessing step (6), we can apply the Zhang–Shu limiter to $\boldsymbol{U}_h^{\mathrm{P}}$ to ensure positivity on all points in S_h.

Step CNS5. Given DG polynomial $\boldsymbol{U}_h^{\mathrm{P}}$, use adaptive time-stepping strategy to solve subproblem (H) form time $t^n + \frac{\Delta t}{2}$ to $t^n + \Delta t$.

Remark 5. *For the sake of robustness and efficiency, whenever possible, one should apply the postprocessing* (27) *to a subset of cells (i.e., T is a strict subset of $\{1, 2, \cdots, N\}$) containing all trouble cells and also some good cells, rather than the whole computational domain (i.e., $T = \{1, 2, \cdots, N\}$). For example, in the 2D Sedov blast wave test in Section 4.5, the initial total energy is 10^{-12} everywhere except in the cell at the lower left corner, and we can define T as*

$$T = \left\{ i : \text{either} \quad \overline{\boldsymbol{U}_i^{\mathrm{P}}} \notin G^\epsilon \quad \text{or} \quad \overline{E_i^{\mathrm{P}}} - \frac{1}{2}\|\overline{\boldsymbol{m}_i^{\mathrm{P}}}\|/\overline{\rho_i^{\mathrm{P}}} \geq 10^{-10} \right\}. \tag{28}$$

By such a definition of T for each time step, the gray region in the Figure 2 will not be modified by the postprocessing. Note, the number of cells contained in T may various at each time step.

4. Numerical experiments

In this section, we validate our full numerical scheme through representative two-dimensional benchmark tests, including the Lax shock tube, double rarefraction, Sedov blast wave, shock diffraction, shock reflection-diffraction, and high Mach number astrophysical jet problems.

For penalty parameters in interior penalty DG method for solving (P), in the $\mathbb{Q}^1$ scheme, we set $\sigma = 2$ on Γ_h, $\sigma = 4$ on $\partial\Omega$, and $\tilde{\sigma} = 2$; in the $\mathbb{Q}^k$ $(k \geq 2)$ schemes, we set $\sigma = 0$ on all faces, namely using NIPG0 method for the velocity, and $\tilde{\sigma} = 2^k$ for the internal energy. We take $\epsilon = 10^{-13}$ as the lower bound for the numerical admissible state set in all tests except the astrophysical jet simulations, where $\epsilon = 10^{-8}$ is used.

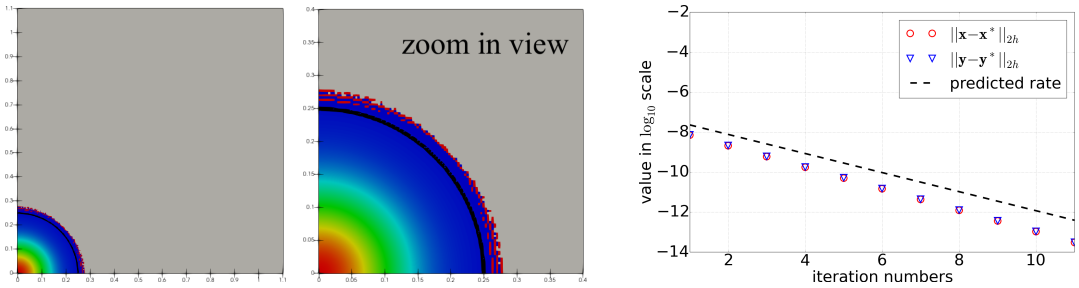

Figure 2: DG with $\mathbb{Q}^2$ basis for 2D Sedov blast wave test. The middle figure is the zoom view of the left figure: the shock is marked black; the negative cells are highlighted by the red marks; by the definition (28), T does not include cells in the gray region in which the exact solution is supposed to be a constant. Right: the actual convergence rate of the Douglas–Rachford splitting algorithm (25) with nearly optimal parameters (26) for solving (27a) for the 2D Sedov problem (at one particular time step for the left figure) matches well the predicated rate from analysis (asymptotic linear convergence from analysis using the estimated principle angle $\hat{\theta} = \cos^{-1}\sqrt{\frac{\hat{r}}{|T|}}$), see [51] for more details on such a provable convergence rate.

The ideal gas constant is $\gamma = 1.4$ and the Prandtl number is $\mathrm{Pr} = 0.72$. **The Reynolds number for all tests is** $\mathrm{Re} = 1000$ **unless otherwise specified**.

In all physical simulations, we use $\theta = \frac{1}{2}$ in (9), namely utilizing the second order Crank–Nicolson method to solve (P). The postprocessing step for total energy variable after solving (P) is only triggered in the accuracy test in Section 4.2, the Sedov blast wave test, and astrophysical jets test.

4.1. Accuracy tests

We verify the order of accuracy of our numerical scheme by utilizing the method of manufactured smooth solutions. Let the computational domain $\Omega = [0,1]^2$ and select the end time $T = 0.1024$. The prescribed non-polynomial solutions are as follows:

$$\begin{aligned}
\rho &= \exp(-t)\sin 2\pi(x+y) + 2,\\
\boldsymbol{u} &= \begin{bmatrix} \exp(-t)\cos(2\pi x)\sin(2\pi y) + 2 \\ \exp(-t)\sin(2\pi x)\cos(2\pi y) + 2 \end{bmatrix},\\
e &= \frac{1}{2}\exp(-t)\cos(2\pi(x+y)) + 1.
\end{aligned}$$

Taking Reynolds number $\mathrm{Re} = 1$ and parameter $\lambda = 1$ in (3), the boundary conditions and the right-hand side of the compressible NS equations are computed by above manufactured solutions. Define the discrete L_h^2 error of density by

$$\|\rho_h^n - \rho(t^n)\|_{L_h^2}^2 = \Delta x^2 \sum_{i=1}^{N}\sum_{\nu=1}^{N_{\mathrm{q}}^{\mathrm{H,vol}}} \omega_\nu \Big| \sum_{j=1}^{N_{\mathrm{loc}}} \rho_{ij}^n \hat{\varphi}_j(\hat{\boldsymbol{q}}_\nu) - \rho(t^n) \circ \boldsymbol{F}_i(\hat{\boldsymbol{q}}_\nu) \Big|^2,$$

where ω_ν and $\hat{\boldsymbol{q}}_\nu$ are the Gauss quadrature weights and points used in evaluating volume integrals in (H). The discrete L_h^2 errors for momentum and total energy are measured similarly. In addition, the discrete L_h^2 for $\boldsymbol{U}_h^n$ is defined by

$$\|\boldsymbol{U}_h^n - \boldsymbol{U}(t^n)\|_{L_h^2}^2 = \|\rho_h^n - \rho(t^n)\|_{L_h^2}^2 + \|\boldsymbol{m}_h^n - \boldsymbol{m}(t^n)\|_{L_h^2}^2 + \|E_h^n - E(t^n)\|_{L_h^2}^2.$$

If $\mathtt{err}_{\Delta x}$ denotes the error on a mesh with resolution Δx, then the rate is given by $\ln(\mathtt{err}_{\Delta x}/\mathtt{err}_{\Delta x/2})/\ln 2$.

For temporal convergence rate tests, we use $\mathbb{Q}^3$ scheme and fix the mesh resolution $\Delta x = 1/64$ small enough such that the time error dominates. We choose NIPG method with $\sigma = 0$ to solve the second

equation in subproblem (P) and choose IIPG method with $\tilde{\sigma} = 8$ to solve the third equation in subproblem (P). We observe the optimal temporal convergence rates, see Table 1.

For spatial convergence rate tests, we use $\theta = \frac{1}{2}$ and fix time step size $\Delta t = 3.125 \times 10^{-6}$ small enough such that the spatial error dominates and the hyperbolic CFL is satisfied. We choose NIPG method with $\sigma = 2$ on Γ_h and $\sigma = 4$ on $\partial\Omega$ for $\mathbb{Q}^1$ scheme; and $\sigma = 0$ for $\mathbb{Q}^k$ ($k \geq 2$) scheme to solve the second equation in subproblem (P). We choose IIPG method with $\tilde{\sigma} = 2^k$ to solve the third equation in subproblem (P). For $\mathbb{Q}^1$, $\mathbb{Q}^3$, and $\mathbb{Q}^5$ schemes, we obtain the optimal spatial convergence rates, see Table 2. For $\mathbb{Q}^2$ and $\mathbb{Q}^4$ schemes, the convergence is suboptimal, which is as expected, since the NIPG and IIPG methods are suboptimal for even order spaces.

θ	Δt	$\|\boldsymbol{U}_h^{N_T} - \boldsymbol{U}(T)\|_{L_h^2}$	Δt	$\|\boldsymbol{U}_h^{N_T} - \boldsymbol{U}(T)\|_{L_h^2}$	rate	Δt	$\|\boldsymbol{U}_h^{N_T} - \boldsymbol{U}(T)\|_{L_h^2}$	rate
1	$4 \cdot 10^{-4}$	$1.599 \cdot 10^{-2}$	$2 \cdot 10^{-4}$	$7.988 \cdot 10^{-3}$	1.001	$1 \cdot 10^{-4}$	$3.997 \cdot 10^{-3}$	0.999
$\frac{1}{2}$	$4 \cdot 10^{-4}$	$1.393 \cdot 10^{-3}$	$2 \cdot 10^{-4}$	$3.601 \cdot 10^{-4}$	1.952	$1 \cdot 10^{-4}$	$9.140 \cdot 10^{-5}$	1.978

Table 1: Test of accuracy. The temporal error and convergence rates. $\theta = 1$ backward Euler scheme for internal energy in subproblem (P). $\theta = \frac{1}{2}$ Crank–Nicolson scheme for internal energy in subproblem (P).

k	Δx	$\|\boldsymbol{U}_h^{N_T} - \boldsymbol{U}(T)\|_{L_h^2}$	Δx	$\|\boldsymbol{U}_h^{N_T} - \boldsymbol{U}(T)\|_{L_h^2}$	rate	Δx	$\|\boldsymbol{U}_h^{N_T} - \boldsymbol{U}(T)\|_{L_h^2}$	rate
1	$1/2^4$	$1.209 \cdot 10^{-1}$	$1/2^5$	$3.071 \cdot 10^{-2}$	1.977	$1/2^6$	$7.728 \cdot 10^{-3}$	1.991
2	$1/2^4$	$5.116 \cdot 10^{-2}$	$1/2^5$	$1.413 \cdot 10^{-2}$	1.856	$1/2^6$	$3.718 \cdot 10^{-3}$	1.926
3	$1/2^3$	$4.945 \cdot 10^{-3}$	$1/2^4$	$2.974 \cdot 10^{-4}$	4.056	$1/2^5$	$1.813 \cdot 10^{-5}$	4.036
4	$1/2^3$	$3.221 \cdot 10^{-4}$	$1/2^4$	$1.677 \cdot 10^{-5}$	4.264	$1/2^5$	$1.012 \cdot 10^{-6}$	4.051
5	$1/2^2$	$7.374 \cdot 10^{-4}$	$1/2^3$	$1.387 \cdot 10^{-5}$	5.733	$1/2^4$	$2.087 \cdot 10^{-7}$	6.054

Table 2: Test of accuracy. The spatial error and convergence rates. From top to bottom: the $\mathbb{Q}^1, \mathbb{Q}^2, \cdots, \mathbb{Q}^5$ schemes using a very small time step for a smooth solution.

4.2. Convergence study for testing of preserving positivity

In this part, we verify our numerical algorithm preserves positivity. Let the computational domain $\Omega = [0, 1]^2$ and the end time $T = 0.1024$. The prescribed manufactured solutions are as follows:

$$\rho = 1, \quad \boldsymbol{u} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad e = \frac{1}{\gamma - 1}(\sin^8(2\pi(x + y)) + 10^{-12}).$$

Taking Reynolds number Re = 1 and Prandtl number Pr = 1.4, namely with $\gamma = 1.4$ we have $\lambda = 1$. The boundary conditions and the system right-hand side are defined by the prescribed solutions. We utilize the same L_h^2 norm to measure error.

We use the second order Crank–Nicolson time discretization for internal energy in parabolic sub-problem. Fix the time step size $\Delta t = 3.125 \times 10^{-6}$ small enough such that the spatial error dominates. We choose NIPG method with $\sigma = 2$ on Γ_h and $\sigma = 4$ on $\partial\Omega$ for $\mathbb{Q}^1$ scheme; and $\sigma = 0$ for $\mathbb{Q}^k$ ($k \geq 2$) scheme to solve the second equation in subproblem (P). We choose IIPG method with $\tilde{\sigma} = 2^k$ to solve the third equation in subproblem (P). We obtain the expected convergence rates, see Table 3.

4.3. Lax shock tube problem

We choose the computational domain $\Omega = [-5, 5] \times [0, 2]$ and set the simulation end time $T = 1.3$. We uniformly partition domain Ω by square cells with mesh resolution $\Delta x = 1/100$. The initial conditions for

k	Δx	$\|\boldsymbol{U}_h^{N_T}-\boldsymbol{U}(T)\|_{L_h^2}$	Δx	$\|\boldsymbol{U}_h^{N_T}-\boldsymbol{U}(T)\|_{L_h^2}$	rate	Δx	$\|\boldsymbol{U}_h^{N_T}-\boldsymbol{U}(T)\|_{L_h^2}$	rate	Postprocessing
1	$1/2^5$	$2.858 \cdot 10^{-2}$	$1/2^6$	$6.804 \cdot 10^{-3}$	2.071	$1/2^7$	$1.692 \cdot 10^{-3}$	2.008	Yes
2	$1/2^5$	$6.301 \cdot 10^{-3}$	$1/2^6$	$1.518 \cdot 10^{-3}$	2.054	$1/2^7$	$3.749 \cdot 10^{-4}$	2.018	Yes
3	$1/2^4$	$2.018 \cdot 10^{-2}$	$1/2^5$	$2.063 \cdot 10^{-4}$	6.612	$1/2^6$	$9.680 \cdot 10^{-6}$	4.414	No
4	$1/2^4$	$2.320 \cdot 10^{-4}$	$1/2^5$	$1.121 \cdot 10^{-5}$	4.372	$1/2^6$	$6.245 \cdot 10^{-7}$	4.166	Yes
5	$1/2^3$	$4.614 \cdot 10^{-2}$	$1/2^4$	$5.697 \cdot 10^{-4}$	6.340	$1/2^5$	$7.187 \cdot 10^{-7}$	9.631	No

Table 3: Test of accuracy. The spatial error and convergence rates. From top to bottom: the $\mathbb{Q}^1, \mathbb{Q}^2, \cdots, \mathbb{Q}^5$ schemes using a very small time step for a smooth solution. In last column, "Yes" indicates the postprocesing (6) is triggered, otherwise "No".

density ρ^0, velocity $\boldsymbol{u}^0 = [u_x^0, u_y^0]^{\mathrm{T}}$, and pressure p^0 are prescribed as follows:

$$[\rho^0, u_x^0, u_y^0, p^0]^{\mathrm{T}} = \begin{cases} [0.445,\ 0.698,\ 0,\ 3.528]^{\mathrm{T}} & \text{if } \ x \in [-5, 0), \\ [0.5,\ 0,\ 0,\ 0.571]^{\mathrm{T}} & \text{if } \ x \in [0, 5]. \end{cases}$$

The top and bottom boundaries are set to be reflective when solving subproblem (H) and to be Neumann-type when solving subproblem (P). Dirichlet boundary conditions are applied to the left and right boundaries for both subproblems (H) and (P), with values equal to the initials before the wave reaches the boundary. The Figure 3 shows snapshots of the density field at the simulation final time $T = 1.3$ in mountain view.

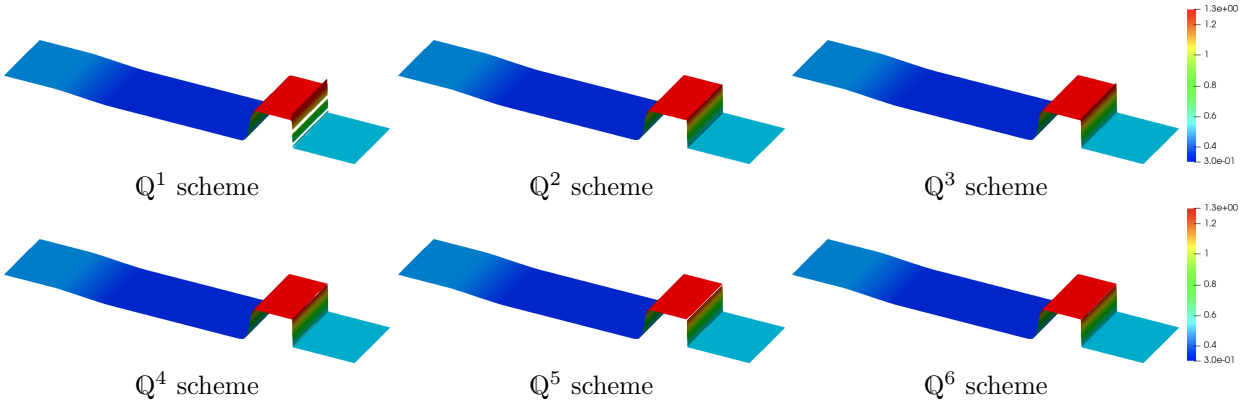

Figure 3: Lax shock tube. The density field snapshots at time $T = 1.3$ are displayed in the mountain view.

4.4. Double rarefaction

We choose the computational domain $\Omega = [-1, 1] \times [0, 1]$ and set the simulation end time $T = 0.6$. We uniformly partition domain Ω by square cells with mesh resolution $\Delta x = 1/640$ for $\mathbb{Q}^1$ and $\mathbb{Q}^2$ schemes, $\Delta x = 1/480$ for $\mathbb{Q}^3$ and $\mathbb{Q}^4$ schemes, and $\Delta x = 1/400$ for $\mathbb{Q}^5$ and $\mathbb{Q}^6$ schemes. The initial conditions for density ρ^0, velocity $\boldsymbol{u}^0 = [u_x^0, u_y^0]^{\mathrm{T}}$, and pressure p^0 are prescribed as follows:

$$[\rho^0, u_x^0, u_y^0, p^0]^{\mathrm{T}} = \begin{cases} [7,\ -1,\ 0,\ 0.2]^{\mathrm{T}} & \text{if } \ x \in [-1, 0), \\ [7,\ 1,\ 0,\ 0.2]^{\mathrm{T}} & \text{if } \ x \in [0, 1]. \end{cases}$$

When solving subproblem (H), reflective boundary conditions are set for the top and bottom boundaries, while outflow conditions are set for the left and right boundaries. When solving subproblem (P), Neumann-type boundary conditions are applied to all boundaries. The Figure 4 shows snapshots of density field at the simulation final time $T = 0.6$ in mountain view.

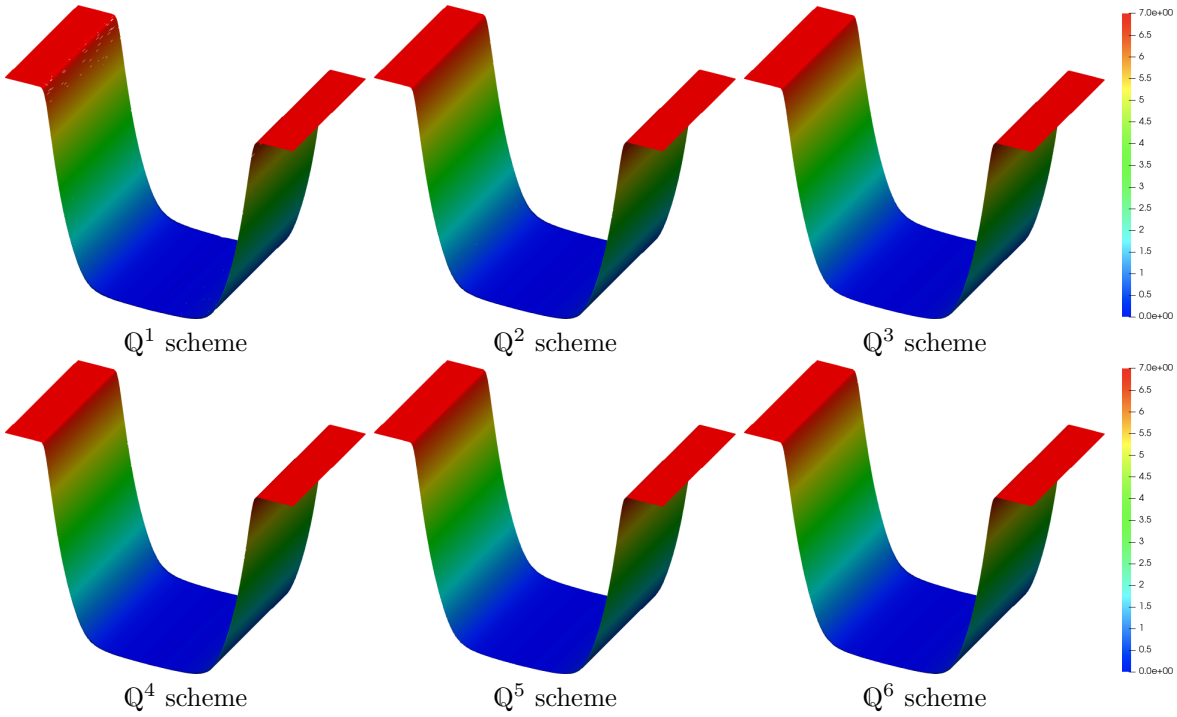

Figure 4: Double rarefaction. The density field snapshots at time $T = 0.6$ are displayed in the mountain view.

4.5. Sedov blast wave

The Sedov blast wave test is a standard benchmark in hyperbolic conservation law. It involves a blast wave generated by a strong explosion, which involves low density, low pressure, and a strong shock. This test holds great value in validating a positivity-preserving scheme.

Let the computational domain $\Omega = [0, 1.1]^2$ and the simulation end time $T = 1$. We uniformly partition domain Ω by square cells with mesh resolution $\Delta x = 1.1/320$. The initials are prescribed as piecewise constants: density $\rho^0 = 1$ and velocity $\boldsymbol{u}^0 = \mathbf{0}$, for all points in Ω; the total energy E^0 equals to 10^{-12} everywhere except the cell at the lower left corner, where $0.244816/\Delta x^2$ is used. When solving subproblem (H), reflective boundary conditions are set for the left and bottom boundaries, while outflow conditions are set for the top and right boundaries. When solving subproblem (P), Neumann-type boundary conditions are applied to all boundaries.

The Figure 5 shows snapshots of density field at the simulation final time $T = 1$. The postprocessing (27) with (28) is used and necessary in all these tests. See Figure 6. Our numerical algorithm preserves conservation and the shock location is correct.

4.6. Shock diffraction

In this test, we consider a right-moving high-speed shock, which is perpendicular to solid surface at initial and moves towards undisturbed air ahead. As the shock crosses the right corner, a region of low density and low pressure emerges, making this a challenging benchmark for conservation law.

Let the computational domain Ω be the union of $[0, 1]\times[6, 11]$ and $[1, 13]\times[0, 11]$. We set the simulation end time $T = 2.3$. The initial condition is a pure right-moving shock of Mach number 5.09, initially located at $\{x = 0.5, 6 \le y \le 12\}$, moving into undisturbed air ahead of the shock with a density of 1.4 and a pressure of 1. When solving subproblem (H), the left boundary is inflow, while the right and bottom boundaries are outflow. The fluid–solid boundaries $\{y = 6, 0 \le x \le 1\}$ and $\{x = 1, 0 \le y \le 6\}$ are reflective. In addition, the flow values on the top boundary are set to accurately depict the motion of the Mach 5.09

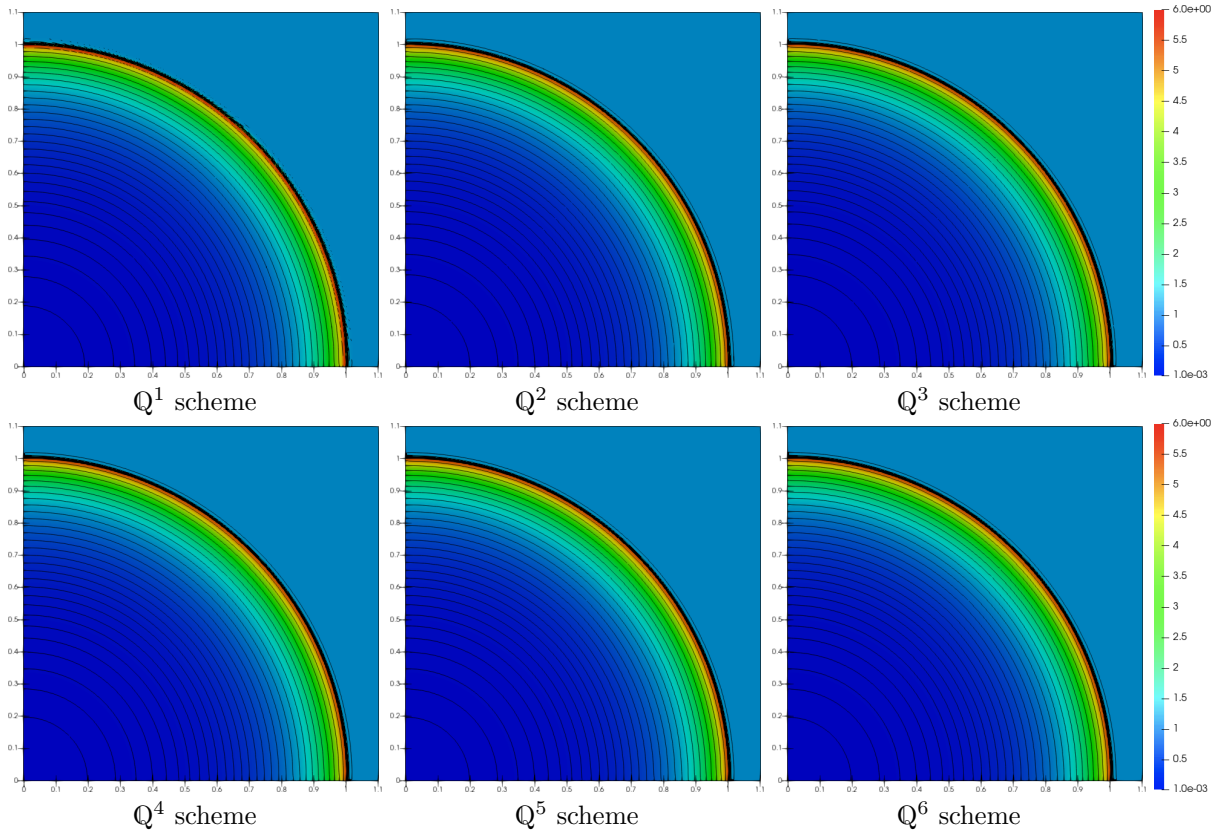

Figure 5: Sedov blast wave. The snapshots of density profile are taken at $T = 1$. Plot of density: 50 exponentially distributed contour lines of density from 0.001 to 6.

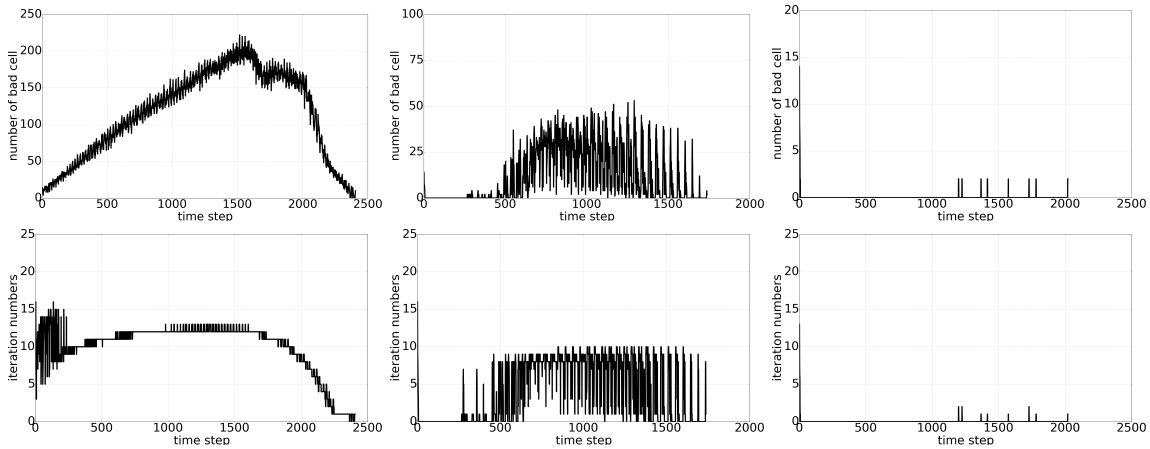

Figure 6: From left to right $\mathbb{Q}^2$, $\mathbb{Q}^4$, $\mathbb{Q}^6$ DG schemes. Top: the number of bad cells after solving (P) at each time step (the DG polynomial cell averages are not in the admissible set). Bottom: the number of Douglas–Rachford iterations need to reach round-off convergence for solving (27a) with (28).

shock. When solving subproblem (P), Neumann-type boundary conditions are applied to the fluid–solid surfaces, while Dirichlet boundary conditions are applied to the remaining boundaries. The Dirichlet data

on the left and top boundaries are determined by the inflow data and the exact motion of the Mach 5.09 shock. Additionally, the Dirichlet data on the right and bottom boundaries remain unchanged from their initial values before the shock wave reaches the boundary.

The Figure 7 displays snapshots of density field at the simulation final time $T = 2.3$. The results are comparable to those in [5].

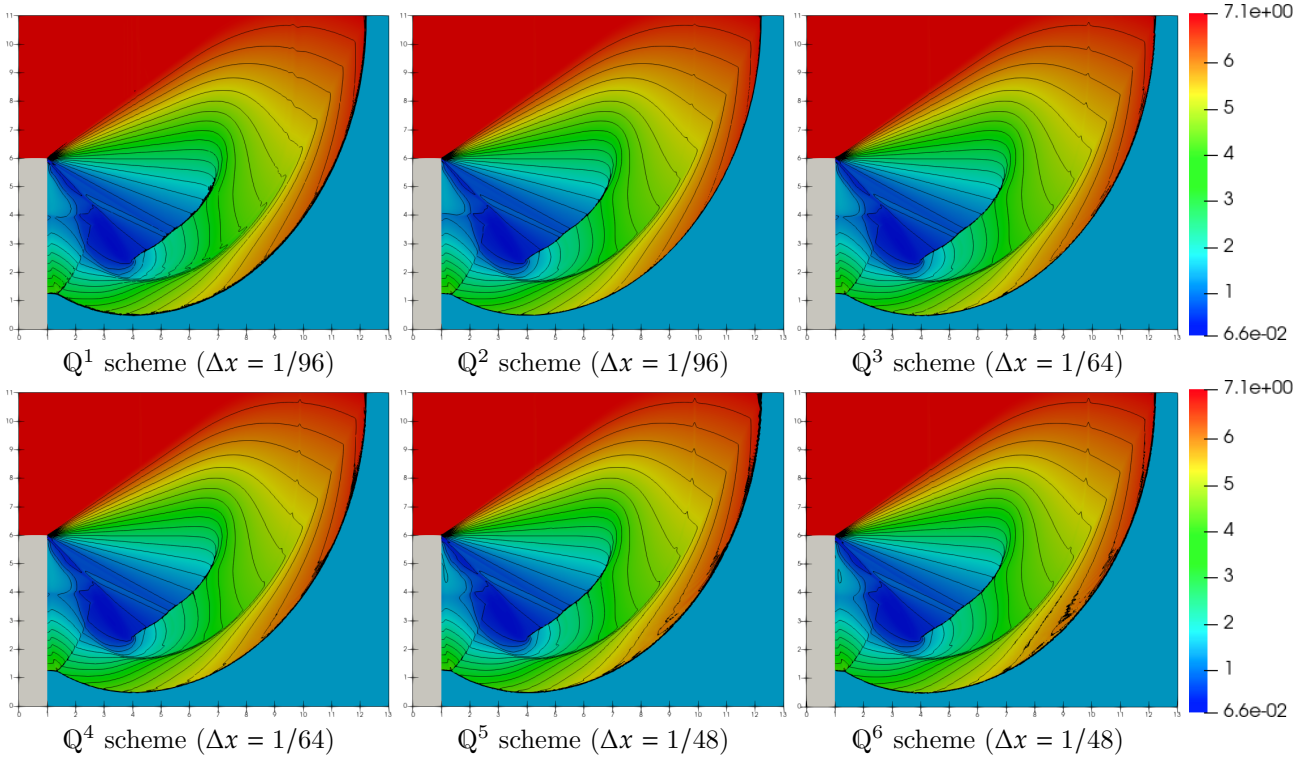

Figure 7: Shock diffraction. The snapshots of density profile are taken at $T = 2.3$. The gray colored region denotes solid. Plot of density: 20 equally spaced contour lines from 0.066227 to 7.0668.

4.7. Mach 10 *shock reflection and diffraction*

The high-speed shock reflection and diffraction test is a widely used benchmark [6]. We consider a Mach 10 shock that moves to the right with a sixty-degree incident angle to the solid surface. As the shock across the sharp corner, areas of low density and low pressure appear. In the region of shock reflection, vortices are formed due to Kelvin–Helmholtz instabilities.

Let the computational domain Ω be the union of $[0,4]\times[0,1]$ and $[1,4]\times[-1,0]$. We set the simulation end time $T = 0.2$. The initial condition is a right-moving shock of Mach number 10 positioned at $(\frac{1}{6},0)$ with a sixty-degree angle to the x-axis. The shock is moving into undisturbed air ahead of it, which has a density of 1.4 and a pressure of 1. In the post-shock region, the density is 8, the velocity is $[4.125\sqrt{3},-4.125]^{\mathrm{T}}$, and the pressure is 116.5.

When solving subproblem (H), the left boundary is inflow, while the right and bottom boundaries are outflow. Part of the fluid–solid boundaries $\{y = 0, \frac{1}{6} \le x \le 1\}$ and $\{x = 1, -1 \le y \le 0\}$ are reflective, and the post-shock condition is imposed at $\{y = 0, 0 \le x \le \frac{1}{6}\}$. On the boundary with post-shock condition, the density, velocity, and pressure are fixed in time with the initial values to make the reflected shock stick to the solid wall. In addition, the flow values on the top boundary are set to accurately depict the motion of the Mach 10 shock. When solving subproblem (P), Neumann-type boundary conditions are applied to part of

the fluid–solid surfaces associated with the reflective boundary in subproblem (H), while Dirichlet boundary conditions are applied to the remaining boundaries. The Dirichlet data on the left and top boundaries are determined by the inflow data and the exact motion of the Mach 10 shock. Additionally, the Dirichlet data on the right and bottom boundaries remain unchanged from their initial values before the shock wave reaches the boundary.

From Figure 8, we see our scheme produces satisfactory non-oscillatory solutions with correct shock location and well-captured rollups. These test results are consistent with the observations for fully explicit high order accurate schemes in [5].

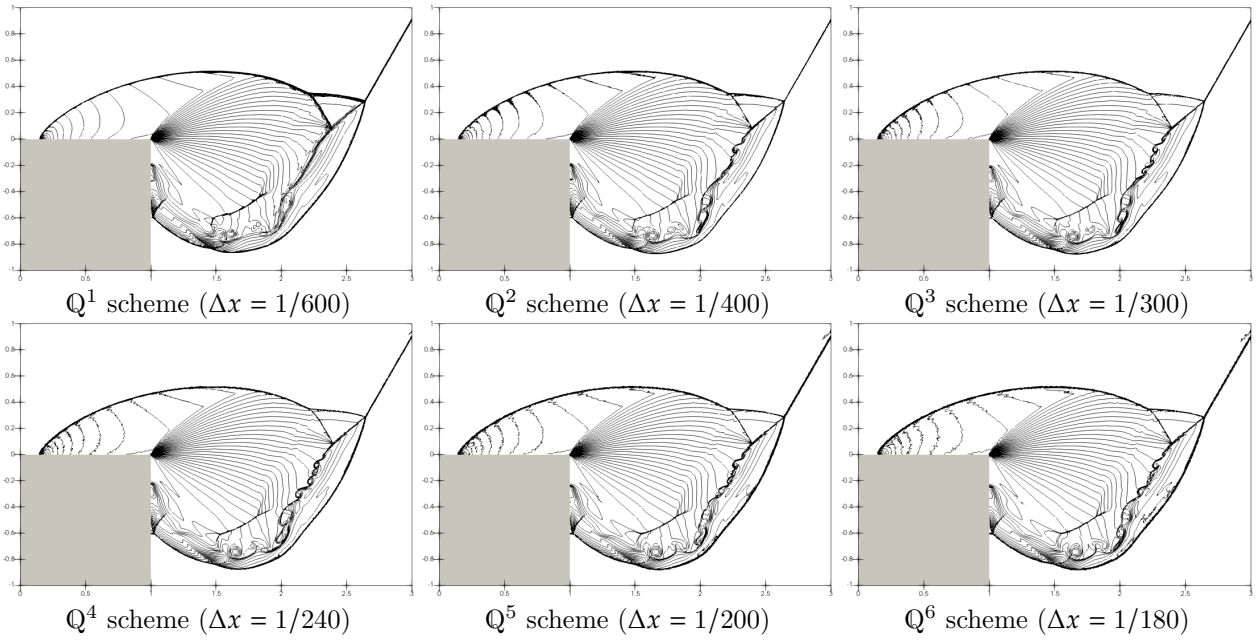

Figure 8: Mach 10 shock reflection and diffraction. The snapshots of density profile are taken at $T = 0.2$. The gray colored region denotes solid. Plot of density: 50 equally space contour lines from 0 to 25. Only contour lines are plotted. We can observe that the scheme with higher order spatial accuracy indeed induces less artificial viscosity, despite that the temporal accuracy is at most second order.

4.8. High Mach number astrophysical jet

To replicate the gas flows and shock wave patterns observed in the Hubble Space Telescope images, one can utilize theoretical models within a gas dynamics simulator, see [61, 62, 63]. We consider the Mach 2000 astrophysical jets without radiative cooling to demonstrate the robustness of our scheme.

Let the computational domain $\Omega = [0, 1] \times [-0.5, 0.5]$. We set the simulation end time $T = 0.001$. In this example, we use the ideal gas constant $\gamma = 5/3$. The initial density $\rho^0 = 0.5$, velocity $\boldsymbol{u}^0 = \mathbf{0}$, and pressure $p^0 = 10^{-6}$. When solving subproblem (H), the following inflow boundary conditions are set for the left boundary

$$[\rho, u_x, u_y, p]^{\mathrm{T}} = \begin{cases} [5,\ 800,\ 0,\ 0.4127]^{\mathrm{T}} & \text{if } x = 0 \text{ and } |y| \le 0.05, \\ [0.5,\ 0,\ 0,\ 10^{-6}]^{\mathrm{T}} & \text{if } x = 0 \text{ and } |y| > 0.05, \end{cases}$$

while the outflow boundary conditions are set for the top, right, and bottom boundaries. When solving subproblem (P), Dirichlet boundary condition is applied to the left boundary, while Neumann-type boundary conditions are applied to the remaining boundaries. The Dirichlet data on the left boundary are determined by the inflow data of the Mach 2000 astrophysical jet.

We take $\epsilon = 10^{-8}$ in defining G^ϵ and the Zhang–Shu limiter in Section 2.3. The postprocessing of DG cell averages is necessary in these simulations. For the sake of robustness and efficiency in the postprocessing step, we define the local region T as the set of indices

$$T = \left\{ i : \text{either} \quad \overline{\boldsymbol{U}_i^{\mathrm{P}}} \notin G^\epsilon \quad \text{or} \quad \overline{E_i^{\mathrm{P}}} - \frac{1}{2}\|\overline{\boldsymbol{m}_i^{\mathrm{P}}}\|/\overline{\rho_i^{\mathrm{P}}} \geq 2 * 10^{-6} \right\}. \tag{29}$$

The Figure 9 shows snapshots of density field at the simulation final time $T = 0.001$. See the performance of Douglas–Rachford splitting for solving (27a) in Figure 10.

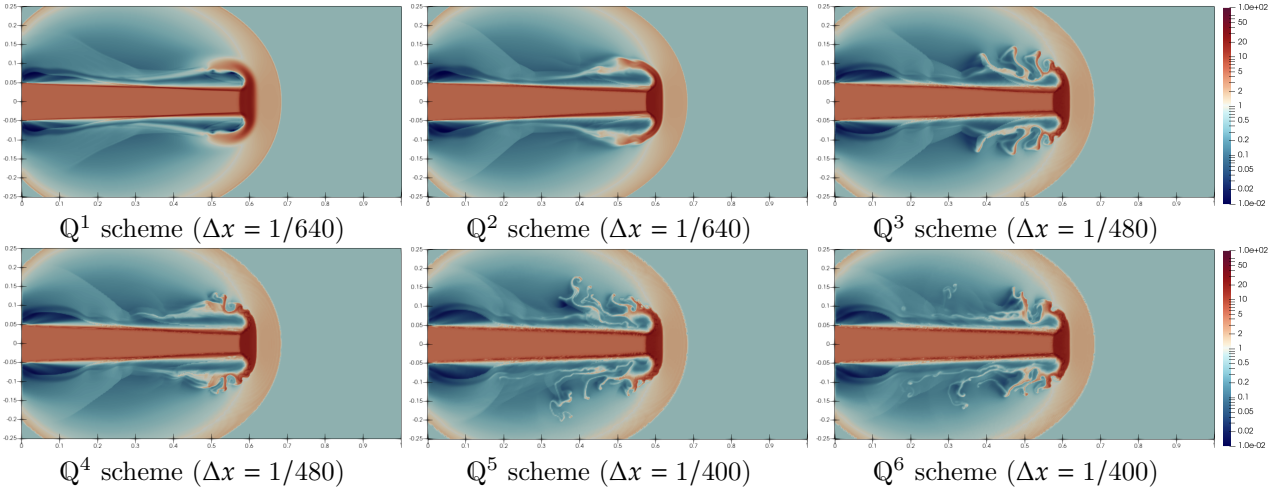

Figure 9: Astrophysical jets. The snapshots of the density filed at $T = 0.001$. Scales are logarithmic. We can observe that the scheme with higher order spatial accuracy indeed induces less artificial viscosity, despite that the temporal accuracy is at most second order.

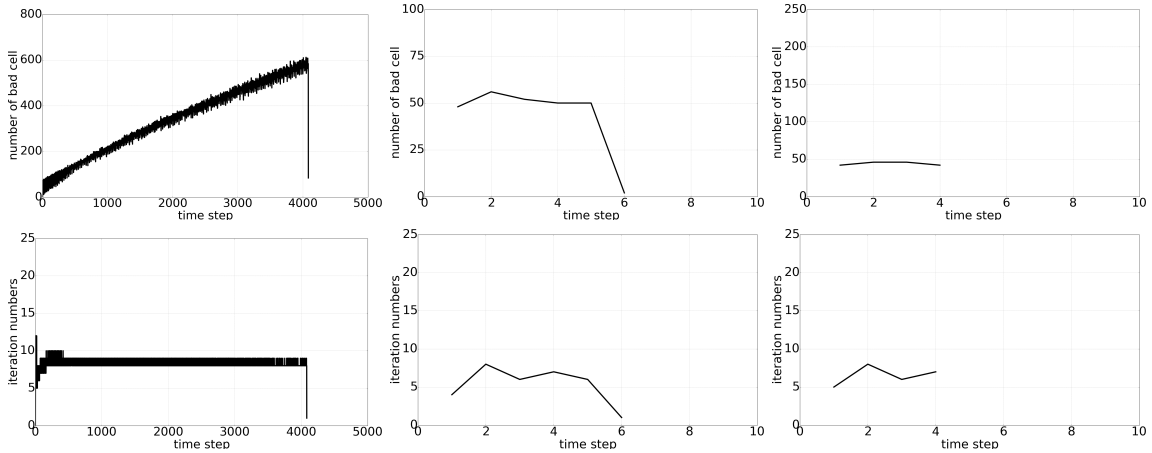

Figure 10: From left to right $\mathbb{Q}^2$, $\mathbb{Q}^4$, $\mathbb{Q}^6$ DG schemes. Top: the number of bad cells after solving (P) at each time step (the DG polynomial cell averages are not in the admissible set). Bottom: the number of Douglas–Rachford iterations need to reach round-off convergence for solving (27a) with (29).

5. Concluding remarks

In this paper, we have constructed a semi-implicit DG scheme that is high order accurate in space, conservative, and positivity-preserving for solving the compressible NS equations. The time step constraint follows the standard hyperbolic CFL condition $\Delta t = \mathcal{O}(\Delta x)$. Our scheme is fully decoupled, requiring only the sequential solving of two linear systems at each time step to achieve second order accuracy in time. Conservation and positivity are ensured through a postprocessing of the cell averages of total energy variable. A high order accurate cell average limiter can be formulated as a constraint minimization, which can be efficiently computed by using the generalized Douglas–Rachford splitting method with nearly optimal parameters. Numerical tests suggest that such a simple and efficient postprocessing of the total energy variable indeed renders the semi-implicit high order DG method with Strang splitting much more robust. Ongoing and future work consists of extensions from the ℓ^2-norm minimization postprocessing to the ℓ^1-norm minimization, and also generalizations to directly enforcing the convex invariant domain.

Acknowledgments

Research is supported by NSF DMS-2208515.

Appendix A. The method of Lagrange multiplier

Given a matrix $\mathbf{A} = [1, 1, \cdots, 1] \in \mathbb{R}^{1\times N}$ and a vector $\boldsymbol{w} \in \mathbb{R}^N$. Define a constant $b = \mathbf{A}\boldsymbol{w}$ and assume $\sum_{i=1}^N w_i > 0$. Let us consider the following constrained minimization problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{w}\|_2^2 \ \text{ subject to } \ \mathbf{A}\boldsymbol{x} = b \ \text{ and } \ x_i \geq 0 \ \text{ for all } \ i \in \{1, \cdots, N\}. \tag{A.1}$$

Consider the Lagrangian function with multipliers λ_i and γ

$$L = \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{w}\|_2^2 + \gamma\Big(\sum_{i=1}^N x_i - b\Big) + \sum_{i=1}^N(-\lambda_i x_i),$$

and its Karush–Kuhn–Tucker (KKT) conditions, which are given as

$$\frac{\partial L}{\partial x_i} = x_i - w_i + \gamma - \lambda_i = 0, \tag{A.2a}$$

$$-\lambda_i x_i = 0, \tag{A.2b}$$

$$\lambda_i \geq 0, \tag{A.2c}$$

$$-x_i \leq 0, \tag{A.2d}$$

$$\sum_{i=1}^N x_i = b. \tag{A.2e}$$

For the constrained minimization problem (A.1), the KKT condition (A.2) is both sufficient and necessary. In the rest of this part, let us assume there exists at least one entry in $\boldsymbol{w}$ that is strictly less than 0. Otherwise, the minimizer of the constraint optimization problem (A.1) is $\boldsymbol{w}$, which is trivial.

Lemma 1. *If there exists an entry in $\boldsymbol{w}$ less than 0, then $\gamma \neq 0$.*

Proof. Assume $\gamma = 0$. Then (A.2a) becomes $x_i - w_i - \lambda_i = 0$, namely we have $\lambda_i = x_i - w_i$. Summing over i from 1 to N, we get

$$\sum_{i=1}^N \lambda_i = \sum_{i=1}^N x_i - \sum_{i=1}^N w_i = b - b = 0.$$

Notice (A.2c) gives $\lambda_i \geq 0$, we have $\lambda_i = 0$ for all i. Thus $x_i = w_i$ for all i, which contradicts the existence of a negative entry in $\boldsymbol{w}$. □

Let $B = \{j : x_j = 0\}$ denote the set of all indexes, as represented in the minimizer $\boldsymbol{x}$ of (A.1), touching the boundary of the feasible region. Let $\#B$ be the number of elements in set B. The next lemma shows that if an entry of the vector $\boldsymbol{w}$ is less than 0, then the minimizer plugs that entry back to the boundary of the feasible region.

Lemma 2. *Assume there exists at least one entry in vector $\boldsymbol{w}$ that is strictly less than 0. Then for any index i so that $w_i \le 0$, we have $i \in B$ and hence $x_i = 0$.*

Proof. From (A.2b), we only need to show $\lambda_i > 0$. By (A.2a), we have $x_i - w_i + \gamma = \lambda_i$. Summing over i from 1 to N, we get

$$\underbrace{\sum_{i=1}^{N} x_i - \sum_{i=1}^{N} w_i}_{=\, b-b \,=\, 0} + N\gamma = \sum_{i=1}^{N} \lambda_i \quad \Rightarrow \quad \gamma = \frac{1}{N}\sum_{i=1}^{N} \lambda_i.$$

Thus, by (A.2c), we know $\gamma \ge 0$. Furthermore, by Lemma 1, we get $\gamma > 0$. Notice (A.2d) gives $x_i \ge 0$. Therefore, under the condition $w_i \le 0$, the (A.2a) implies $\lambda_i = x_i - w_i + \gamma > 0$. □

Lemma 3. *The solution of the constrained minimization problem* (A.1) *satisfies:*

- *If $x_i = 0$, then we have*

$$\lambda_i - \frac{1}{N}\sum_{j\in B} \lambda_j = -w_i, \quad \forall i \in B. \tag{A.3}$$

- *If $x_i > 0$, then we have*

$$x_i = w_i + \frac{1}{N - \#B}\sum_{j\in B} w_j. \tag{A.4}$$

Proof. From (A.2a), we have $\gamma = \lambda_i + w_i - x_i$. Summing over i from 1 to N, we get

$$N\gamma = \sum_{i=1}^{N} \lambda_i + \underbrace{\sum_{i=1}^{N} w_i - \sum_{i=1}^{N} x_i}_{=\, b-b \,=\, 0} = \sum_{i\in B} \lambda_i + \sum_{i\notin B} \lambda_i.$$

Recall that the set $B = \{j : x_j = 0\}$. By (A.2d), $i \notin B$ gives $x_i > 0$. By (A.2b), we have $\lambda_i = 0$ for all $i \notin B$. Thus, we get

$$\gamma = \frac{1}{N}\sum_{i\in B} \lambda_i. \tag{A.5}$$

If $x_i = 0$, then $i \in B$ and (A.2a) becomes $\lambda_i - \gamma = -w_i$, so replacing γ with (A.5), we obtain (A.3). Summing over $i \in B$ of (A.3), we have

$$\sum_{i\in B} \lambda_i - \frac{\#B}{N}\sum_{j\in B} \lambda_j = -\sum_{i\in B} w_i \quad \Rightarrow \quad \sum_{j\in B} \lambda_j = -\frac{N}{N - \#B}\sum_{j\in B} w_j.$$

If $x_i > 0$, then by (A.2b) we have $\lambda_i = 0$. Again, (A.2a) and (A.5) gives

$$x_i = w_i - \gamma = w_i - \frac{1}{N}\sum_{j\in B} \lambda_j = w_i + \frac{1}{N - \#B}\sum_{j\in B} w_j.$$

Therefore, we conclude the proof. □

Lemma 4. *If $w_{i_1} \geq w_{i_2} > 0$, then $x_{i_1} = 0$ implies $x_{i_2} = 0$, namely $i_1 \in B$ implies $i_2 \in B$.*

Proof. Let us first deal with the case $w_{i_1} > w_{i_2}$. If the vector $\boldsymbol{x}$ is a solution of the minimization problem (A.1) with $x_{i_1} = 0$ and $x_{i_2} > 0$, then we will construct a solution vector $\tilde{\boldsymbol{x}}$ such that $\tilde{x}_i = x_i$ for all $i \notin \{i_1, i_2\}$ and $\tilde{x}_{i_1} = x_{i_2}$ and $\tilde{x}_{i_2} = 0$.

- *Check constraint*: since $\tilde{x}_i = x_i$ for all $i \notin \{i_1, i_2\}$, we only need to check $\tilde{x}_{i_1} + \tilde{x}_{i_2} = x_{i_1} + x_{i_2}$. This holds since $\tilde{x}_{i_1} + \tilde{x}_{i_2} = x_{i_2} + 0$ and $x_{i_1} + x_{i_2} = 0 + x_{i_2}$.
- *Compare* 2*-norm*: we have $(w_{i_1} - x_{i_1})^2 + (w_{i_2} - x_{i_2})^2 > (w_{i_1} - \tilde{x}_{i_1})^2 + (w_{i_2} - \tilde{x}_{i_2})^2$, which can be easily verified as follows

$$\begin{aligned} & (w_{i_1} - x_{i_1})^2 + (w_{i_2} - x_{i_2})^2 > (w_{i_1} - \tilde{x}_{i_1})^2 + (w_{i_2} - \tilde{x}_{i_2})^2 \\ \Leftrightarrow \quad & w_{i_1}^2 + (w_{i_2} - x_{i_2})^2 > (w_{i_1} - x_{i_2})^2 + w_{i_2}^2 \\ \Leftrightarrow \quad & w_{i_1}^2 + w_{i_2}^2 - 2w_{i_2}x_{i_2} + x_{i_2}^2 > w_{i_1}^2 - 2w_{i_1}x_{i_2} + x_{i_2}^2 + w_{i_2}^2 \\ \Leftrightarrow \quad & (w_{i_1} - w_{i_2})x_{i_2} > 0, \end{aligned}$$

  which holds when $w_{i_1} > w_{i_2}$ and $x_{i_2} > 0$.

Hence we have constructed a vector $\tilde{\boldsymbol{x}}$ that satisfies the constraint but has smaller objective value, which contradicts that $\boldsymbol{x}$ is the unique minimizer of (A.1).

In case of $w_{i_1} = w_{i_2}$, we use contradiction argument to show the vector $\boldsymbol{x}$ with $x_{i_1} = 0$ and $x_{i_2} > 0$ is not a solution of the minimization problem (A.1). We construct a vector $\tilde{\boldsymbol{x}}$ such that $\tilde{x}_i = x_i$ for all $i \notin \{i_1, i_2\}$ and $\tilde{x}_{i_1} = \frac{1}{2}x_{i_2}$ and $\tilde{x}_{i_2} = \frac{1}{2}x_{i_2}$.

- *Check constraint*: since $\tilde{x}_i = x_i$ for all $i \notin \{i_1, i_2\}$, we only need to check $\tilde{x}_{i_1} + \tilde{x}_{i_2} = x_{i_1} + x_{i_2}$. This holds since $\tilde{x}_{i_1} + \tilde{x}_{i_2} = x_{i_2}$ and $x_{i_1} + x_{i_2} = x_{i_2}$.
- *Compare* 2*-norm*: we have $(w_{i_1} - x_{i_1})^2 + (w_{i_2} - x_{i_2})^2 > (w_{i_1} - \tilde{x}_{i_1})^2 + (w_{i_2} - \tilde{x}_{i_2})^2$, which can be easily verified as follows

$$\begin{aligned} & (w_{i_1} - x_{i_1})^2 + (w_{i_2} - x_{i_2})^2 > (w_{i_1} - \tilde{x}_{i_1})^2 + (w_{i_2} - \tilde{x}_{i_2})^2 \\ \Leftrightarrow \quad & w_{i_1}^2 + (w_{i_2} - x_{i_2})^2 > (w_{i_1} - \frac{1}{2}x_{i_2})^2 + (w_{i_2} - \frac{1}{2}x_{i_2})^2 \\ \Leftrightarrow \quad & w_{i_1}^2 - (w_{i_1} - \frac{1}{2}x_{i_2})^2 > (w_{i_2} - \frac{1}{2}x_{i_2})^2 - (w_{i_2} - x_{i_2})^2 \\ \Leftrightarrow \quad & x_{i_2}(2w_{i_1} - \frac{1}{2}x_{i_2}) > x_{i_2}(2w_{i_2} - \frac{3}{2}x_{i_2}) \\ \Leftrightarrow \quad & w_{i_1} - w_{i_2} > -\frac{1}{2}x_{i_2} \end{aligned}$$

  This hold when $w_{i_1} = w_{i_2}$ and $x_{i_2} > 0$.

The proof is now concluded. □

The Lemma 2 indicates the following: if the i-th entry of the vector $\boldsymbol{w}$ is non-positive, $w_i \leq 0$, then we need to set $x_i = 0$. Lemma 3 gives the structure of the exact solution to the minimization problem (A.1). Lemma 4 helps us to construct the following algorithm to find the set B and obtain the solution to (A.1).

- Step 1. If $w_i \leq 0$, then set $x_i = 0$ and push i in set B.
- Step 2. **Sort all entries** $w_i > 0$ in $\boldsymbol{w}$ in ascending order.
- Step 3. Compute the "total out-of-bound mass" of set B by the following formula:

$$-\sum_{j \in B} w_j.$$

- Step 4. Check whether the smallest w_{i_s}, where $i_s \notin B$, satisfies

$$w_{i_s} + \frac{1}{N - \#B} \sum_{j \in B} w_j > 0. \tag{A.6}$$

If (A.6) holds, then uniformly allocate the "total out-of-bound mass" of set B to all other "good entries" by formula

$$x_i = w_i + \frac{1}{N - \#B} \sum_{j \in B} w_j \quad \text{for all } i \notin B.$$

Otherwise, push i_s into set B and go to Step 3. Note: if there are multiple entries with the same smallest value, then push all of them into set B.

The complexity of sorting algorithm `std::sort()` in C++ is $O(N \log(N))$ on the best and average case scenarios. Additionally, sophisticated coding skills are required for implementing sorting algorithms on a distributed memory system. In comparison, the complexity of the DR algorithm is $O(N)$ and DR is very amenable to parallelization.

Comparison of optimization algorithms. We create synthetic data to let $\boldsymbol{w}$ in (A.1) be defined as point values of the following function on a uniform grid of size 1000^2 on the domain $[0,1]^2$:

$$f(x,y) = \begin{cases} -0.25, & -\frac{\delta}{4} + 0.25 \le x \le \frac{\delta}{4} + 0.25 \\ -0.25, & -\frac{\delta}{4} + 0.75 \le x \le \frac{\delta}{4} + 0.75 \\ \cos^8(2\pi x) + 10^{-13}, & \text{otherwise} \end{cases},$$

where $\delta > 0$ is a parameter. A different value of δ gives a different ratio of negative point values, and we consider values of δ such that the ratio of negative point values is $1\%, 2\%, 5\%, 10\%$ and 20%.

We then solve (A.1) with $b = \mathbf{A}\boldsymbol{w}$ by both the method of Lagrange multiplier and the Douglas–Rachford method. For each optimization method, we solve (A.1) to machine precision 100 times and compare the average CPU time for solving it once on a single Intel Xeon CPU E5-2660 v3 2.60GHz. The Table A.4 shows the computational time of finding the minimizer up to machine precision.

The time cost of the DR algorithm increases as the ratio of negative points increases, which is however still faster than the Lagrange multiplier approach for large data set, due to the $O(N \log(N))$ sorting operation. Notice that as the number of negative points increases, the data set requiring sorting becomes smaller, resulting in a decrease in the time cost of the Lagrange multiplier method. However, in large-scale simulations with a good base scheme such as a proper DG scheme in this paper, the percentage of negative points is typically small. Such a comparison suggests that the DR algorithm is a preferable option from the efficiency perspective.

bad cells %	1%	2%	5%	10%	20%
LM	1.426 s	1.500 s	1.509 s	1.418 s	1.130 s
DR	0.378 s	0.467 s	0.565 s	0.656 s	0.846 s

Table A.4: The CPU time for applying the method of Lagrange multiplier and the DR algorithm to solve the minimization (A.1) for a problem of size 10^6 for problems with different ratios of negative points (bad cells). The time unit is second. The "LM" refers to the method of Lagrange multiplier and the "DR" refers to the Douglas–Rachford splitting algorithm.

References

[1] D. Hoff, D. Serre, The failure of continuous dependence on initial data for the Navier–Stokes equations of compressible flow, SIAM Journal on Applied Mathematics 51 (4) (1991) 887–898.

[2] J.-L. Guermond, M. Maier, B. Popov, I. Tomas, Second-order invariant domain preserving approximation of the compressible Navier–Stokes equations, Computer Methods in Applied Mechanics and Engineering 375 (2021) 113608.
[3] X. Zhang, C.-W. Shu, On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes, Journal of Computational Physics 229 (23) (2010) 8918–8934.
[4] D. Grapsas, R. Herbin, W. Kheriji, J.-C. Latché, An unconditionally stable staggered pressure correction scheme for the compressible Navier–Stokes equations, The SMAI journal of computational mathematics 2 (2016) 51–97.
[5] X. Zhang, On positivity-preserving high order discontinuous Galerkin schemes for compressible Navier–Stokes equations, Journal of Computational Physics 328 (2017) 301–343.
[6] C. Fan, X. Zhang, J. Qiu, Positivity-preserving high order finite difference WENO schemes for compressible Navier–Stokes equations, Journal of Computational Physics 467 (2022) 111446.
[7] C. Liu, X. Zhang, A positivity-preserving implicit-explicit scheme with high order polynomial basis for compressible Navier–Stokes equations, Journal of Computational Physics 493 (2023) 112496.
[8] J. Shen, X. Zhang, Discrete maximum principle of a high order finite difference scheme for a generalized Allen–Cahn equation, Communications in Mathematical Sciences 20 (5) (2022) 1409–1436.
[9] J. Hu, X. Zhang, Positivity-preserving and energy-dissipative finite difference schemes for the Fokker–Planck and Keller–Segel equations, IMA Journal of Numerical Analysis 43 (3) (2023) 1450–1484.
[10] C. Liu, Y. Gao, X. Zhang, Structure preserving schemes for Fokker–Planck equations of irreversible processes, Journal of Scientific Computing 98 (1) (2024) 4.
[11] C. Fan, X. Zhang, J. Qiu, Positivity-preserving high order finite volume hybrid hermite WENO schemes for compressible Navier–Stokes equations, Journal of Computational Physics 445 (2021) 110596.
[12] X. Zhang, Y. Liu, C.-W. Shu, Maximum-principle-satisfying high order finite volume weighted essentially nonoscillatory schemes for convection-diffusion equations, SIAM Journal on Scientific Computing 34 (2) (2012) A627–A658.
[13] Z. Chen, H. Huang, J. Yan, Third order maximum-principle-satisfying direct discontinuous Galerkin methods for time dependent convection diffusion equations on unstructured triangular meshes, Journal of Computational Physics 308 (2016) 198–217.
[14] S. Srinivasan, J. Poggie, X. Zhang, A positivity-preserving high order discontinuous Galerkin scheme for convection–diffusion equations, Journal of Computational Physics 366 (2018) 120–143.
[15] Z. Sun, J. A. Carrillo, C.-W. Shu, A discontinuous Galerkin method for nonlinear parabolic equations and gradient flow problems with interaction potentials, Journal of Computational Physics 352 (2018) 76–104.
[16] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, Journal of computational physics 131 (2) (1997) 267–279.
[17] F. Bassi, S. Rebay, Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier–Stokes equations, International journal for numerical methods in fluids 40 (1-2) (2002) 197–207.
[18] C. E. Baumann, J. T. Oden, A discontinuous hp finite element method for the Euler and Navier–Stokes equations, International Journal for Numerical Methods in Fluids 31 (1) (1999) 79–95.
[19] B. Cockburn, G. E. Karniadakis, C.-W. Shu, Discontinuous Galerkin methods: theory, computation and applications, Vol. 11, Springer Science & Business Media, 2012.
[20] C.-W. Shu, Discontinuous Galerkin method for time-dependent problems: survey and recent developments, Recent Developments in Discontinuous Galerkin Finite Element Methods for Partial Differential Equations: 2012 John H Barrett Memorial Lectures (2014) 25–62.
[21] D. N. Arnold, F. Brezzi, B. Cockburn, L. D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM journal on numerical analysis 39 (5) (2002) 1749–1779.
[22] X. Zhang, C.-W. Shu, On maximum-principle-satisfying high order schemes for scalar conservation laws, Journal of Computational Physics 229 (9) (2010) 3091–3120.
[23] X. Zhang, C.-W. Shu, Positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations with source terms, Journal of Computational Physics 230 (4) (2011) 1238–1248.
[24] X. Zhang, Y. Xia, C.-W. Shu, Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes, Journal of Scientific Computing 50 (1) (2012) 29–62.
[25] X. Zhang, C.-W. Shu, A minimum entropy principle of high order schemes for gas dynamics equations, Numerische Mathematik 121 (3) (2012) 545–563.
[26] V. Girault, B. Riviere, M. Wheeler, A discontinuous Galerkin method with nonoverlapping domain decomposition for the Stokes and Navier–Stokes problems, Mathematics of computation 74 (249) (2005) 53–84.
[27] C. Liu, F. Frank, F. O. Alpak, B. Riviere, An interior penalty discontinuous Galerkin approach for 3D incompressible Navier–Stokes equation for permeability estimation of porous media, Journal of Computational Physics 396 (2019) 669–686.
[28] R. Masri, C. Liu, B. Riviere, A discontinuous Galerkin pressure correction scheme for the incompressible Navier–Stokes equations: Stability and convergence, Mathematics of Computation 91 (336) (2022) 1625–1654.
[29] R. Masri, C. Liu, B. Riviere, Improved a priori error estimates for a discontinuous Galerkin pressure correction scheme for the Navier–Stokes equations, Numerical Methods for Partial Differential Equations (2023).
[30] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, SIAM journal on numerical analysis 35 (6) (1998) 2440–2463.
[31] P. Castillo, B. Cockburn, I. Perugia, D. Schötzau, An a priori error analysis of the local discontinuous Galerkin method for elliptic problems, SIAM Journal on Numerical Analysis 38 (5) (2000) 1676–1706.
[32] H. Liu, J. Yan, The direct discontinuous Galerkin (DDG) method for diffusion with interface corrections, Communications in Computational Physics 8 (3) (2010) 541.

[33] M. Zhang, J. Yan, Fourier type error analysis of the direct discontinuous Galerkin method and its variations for diffusion equations, Journal of Scientific Computing 52 (3) (2012) 638–655.
[34] H. Liu, Optimal error estimates of the direct discontinuous Galerkin method for convection-diffusion equations, Mathematics of computation 84 (295) (2015) 2263–2295.
[35] B. Cockburn, B. Dong, J. Guzman, M. Restelli, R. Sacco, A hybridizable discontinuous Galerkin method for steady-state convection-diffusion-reaction problems, SIAM Journal on Scientific Computing 31 (5) (2009) 3827–3846.
[36] J. Peraire, N. Nguyen, B. Cockburn, A hybridizable discontinuous Galerkin method for the compressible Euler and Navier–Stokes equations, in: 48th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, 2010, p. 363.
[37] N. C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations, Journal of Computational Physics 230 (4) (2011) 1147–1170.
[38] J. Peraire, P.-O. Persson, The compact discontinuous Galerkin (CDG) method for elliptic problems, SIAM Journal on Scientific Computing 30 (4) (2008) 1806–1824.
[39] A. Uranga, P.-O. Persson, M. Drela, J. Peraire, Implicit large eddy simulation of transitional flows over airfoils and wings, in: 19th AIAA Computational Fluid Dynamics, American Institute of Aeronautics and Astronautics, Inc., 2009, p. 4131.
[40] T. L. Horváth, M. E. Mincsovics, Discrete maximum principle for interior penalty discontinuous Galerkin methods, Central European Journal of Mathematics 11 (4) (2013) 664–679.
[41] H. Li, X. Zhang, A monotone Q^1 finite element method for anisotropic elliptic equations, arXiv preprint arXiv:2310.16274 (2023).
[42] H. Li, X. Zhang, On the monotonicity and discrete maximum principle of the finite difference implementation of C^0-Q^2 finite element method, Numerische Mathematik 145 (2) (2020) 437–472.
[43] L. J. Cross, X. Zhang, On the monotonicity of Q^2 spectral element method for Laplacian on quasi-uniform rectangular meshes, Communications in Computational Physics 35 (1) (2024) 160–180.
[44] L. J. Cross, X. Zhang, On the monotonicity of Q^3 spectral element method for Laplacian, arXiv preprint arXiv:2010.07282 (2023).
[45] W. Höhn, H. D. Mittelmann, Some remarks on the discrete maximum-principle for finite elements of higher order, Computing 27 (2) (1981) 145–154.
[46] H. Li, X. Zhang, A high order accurate bound-preserving compact finite difference scheme for two-dimensional incompressible flow, Communications on Applied Mathematics and Computation 6 (1) (2024) 113–141.
[47] O. Guba, M. Taylor, A. St-Cyr, Optimization-based limiters for the spectral element method, Journal of Computational Physics 267 (2014) 176–195.
[48] J. J. van der Vegt, Y. Xia, Y. Xu, Positivity preserving limiters for time-implicit higher order accurate discontinuous Galerkin discretizations, SIAM journal on scientific computing 41 (3) (2019) A2037–A2063.
[49] Q. Cheng, J. Shen, A new lagrange multiplier approach for constructing structure preserving schemes, II. Bound preserving, SIAM Journal on Numerical Analysis 60 (3) (2022) 970–998.
[50] F. Ruppenthal, D. Kuzmin, Optimal control using flux potentials: A way to construct bound-preserving finite element schemes for conservation laws, Journal of Computational and Applied Mathematics 434 (2023) 115351.
[51] C. Liu, B. Riviere, J. Shen, X. Zhang, 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 (3) (2024) A1923–A1948.
[52] P.-L. Lions, B. Mercier, Splitting algorithms for the sum of two nonlinear operators, SIAM Journal on Numerical Analysis 16 (6) (1979) 964–979.
[53] M. Fortin, R. Glowinski, Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems, Elsevier, 2000.
[54] T. Goldstein, S. Osher, The split Bregman method for L1-regularized problems, SIAM journal on imaging sciences 2 (2) (2009) 323–343.
[55] L. Demanet, X. Zhang, Eventual linear convergence of the Douglas–Rachford iteration for basis pursuit, Mathematics of Computation 85 (297) (2016) 209–238.
[56] A. Chambolle, T. Pock, An introduction to continuous optimization for imaging, Acta Numerica 25 (2016) 161–319.
[57] K. C. Kiwiel, Breakpoint searching algorithms for the continuous quadratic knapsack problem, Mathematical Programming 112 (2008) 473–491.
[58] B. Riviere, Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation, Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics, 2008.
[59] Z. Xu, X. Zhang, Bound-preserving high-order schemes, in: Handbook of numerical analysis, Vol. 18, Elsevier, 2017, pp. 81–102.
[60] C. Wang, X. Zhang, C.-W. Shu, J. Ning, Robust high order discontinuous Galerkin schemes for two-dimensional gaseous detonations, Journal of Computational Physics 231 (2) (2012) 653–665.
[61] C. L. Gardner, S. J. Dwyer, Numerical simulation of the XZ Tauri supersonic astrophysical jet, Acta Mathematica Scientia 29 (6) (2009) 1677–1683.
[62] Y. Ha, C. L. Gardner, A. Gelb, C.-W. Shu, Numerical simulation of high Mach number astrophysical jets with radiative cooling, Journal of Scientific Computing 24 (2005) 29–44.
[63] W. Tong, R. Yan, G. Chen, On a class of robust bound-preserving MUSCL-Hancock schemes, Journal of Computational Physics 474 (2023) 111805.