AN ESSENTIALLY OSCILLATION-FREE DISCONTINUOUS GALERKIN METHOD FOR HYPERBOLIC SYSTEMS

YONG LIU[†], JIANFANG LU[‡], AND CHI-WANG SHU[§]

Abstract. In this paper, we develop an essentially oscillation-free discontinuous Galerkin (OFDG) method for systems of hyperbolic conservation laws. Based on the standard discontinuous Galerkin (DG) method, the numerical damping terms are introduced so as to control the spurious oscillations, similar to the scalar case [19]. We use both the classical Runge-Kutta method and the modified exponential Runge-Kutta method in time discretization. Particularly, the latter one could avoid additional restrictions of time step size due to the numerical damping. Extensive numerical experiments are shown to demonstrate our algorithm is robust and effective.

Key words. Hyperbolic conservation laws; Compressible Euler equations; Discontinuous Galerkin method; Non-oscillatory

AMS subject classifications. 65M60

1. Introduction.

Among various numerical approaches for solving hyperbolic conservation laws, discontinuous Galerkin (DG) methods have proven to be one of the most attractive and advantageous approaches. Since complete discontinuous basis functions are used, they have some advantages different from classical finite element methods, such as the allowance of arbitrary unstructured meshes with hanging nodes, easy *h*-*p* adaptivity, and high parallel efficiency thanks to the very local data communication. The first DG method was introduced by Reed and Hill in 1973 to solve a steady linear transport problem [20]. Later, Cockburn et al. used DG methods in spatial discretization coupled with Runge-Kutta time discretization methods for solving time dependent partial differential equations, including hyperbolic conservation laws, convection diffusion equations, etc [9, 10, 8, 6, 11, 12]. In the last several decades, there are a lot of literatures for the extension of DG methods to solve other types of partial differential equations, we refer to several survey papers for more details [7, 24, 25].

It is widely known that, due to the nonlinearity of the hyperbolic equations, the solution could evolve into discontinuities in a finite time even with smooth initial and boundary conditions. This causes difficulties in numerical simulation because many numerical schemes cannot compute the shock speed correctly and even worse, the generated spurious oscillations near the discontinuity may make the scheme less robust and easily blow up for some tough problems. To control the spurious oscillations, basically there are two classes of methods to deal with it. One is to apply slope limiters after obtaining numerical solutions from DG methods, such as the *minmod* type total variation diminishing (TVD) limiter, total variation bounded (TVB) limiter

[†]LSEC, Institute of Computational Mathematics, Hua Loo-Keng Center for Mathematical Sciences, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, P.R. China. E-mail: yongliu@lsec.cc.ac.cn. Research is partially supported by the fellowship of China Postdoctoral Science Foundation No. 2020TQ0343.

[‡]South China Research Center for Applied Mathematics and Interdisciplinary Studies, South China Normal University, Canton, Guangdong 510631, China. E-mail: jflu@m.scnu.edu.cn. Research is partially supported by NSFC grant 11901213 and Guangdong Basic and Applied Basic Research Foundation 2020B1515310021.

[§]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. E-mail: chi-wang_shu@brown.edu. Research is partially supported by NSF grant DMS-2010107 and AFOSR grant FA9550-20-1-0055.

and weighted essentially non-oscillatory (WENO) limiter, etc. The limiting process can be viewed as a post-processor for the DG solution, and troubled cell indicator is often needed to identify which cell should be modified. The limiters are designed for specific needs and often involve some parameters. With suitable adjustments of these parameters, the schemes may work quite well to obtain excellent results. See e.g. [24, 34] and the references therein. Another is to add artificial diffusion terms in the weak formulations, and the artificial diffusion coefficient of the artificial diffusion should be chosen adequately, see e.g. [3, 13, 15]. More recently, in [19] we have developed an approach to control oscillations for scalar conservation laws by adding damping terms on high order coefficients of the basis functions. The formulations of added damping terms are similar in spirit to the so-called "local projection stabilization" schemes, see [1, 2]. As demonstrated in [19], the damping terms can not only control the spurious oscillations, but also preserve some basic properties of the standard DG method such as conservation, optimal a priori error estimates and superconvergence, etc.

In this paper, we extend our previous work [19] to hyperbolic systems. Our main contribution in this paper is two folds. First, we carefully construct the damping terms with the aid of the characteristic variables. In the characteristic decomposition of the Jacobian of flux functions, we present the left eigenvector matrix for compressible Euler equations as an illustration example. Then we add global damping terms for each component of the system in the weak formulation. Secondly, for some tough problems with very strong shock discontinuities, see Section 3, the damping coefficients are so large that the damping terms become quite stiff. Therefore, directly using explicit Runge-Kutta methods requires a more restricted time step size. In order to avoid this problem, we follow the idea of the modified exponential Runge-Kutta methods [16], which helps us obtain a fully discrete scheme with relatively relaxed CFL condition comparing to the case of using explicit Runge-Kutta method directly. Then we prove the strong stability properties of the modified exponential Runge-Kutta methods, under the assumption that the first order forward Euler scheme is strongly stable. We test extensive numerical examples to show that our method has good performance to treat various problems. Particularly, we compute several tough numerical examples such as the Sedov point blast problem, the high Mach number astrophysical jets problem and the shock diffraction problem, etc. The proposed algorithm works quite well without applying any limiters or artificial diffusion during the simulation.

The outline of the paper is as follows. In Section 2, we present our essentially OFDG methods for one- and two-dimensional systems of hyperbolic conservation laws, and introduce the modified exponential Runge-Kutta time discretization. We also show the numerical solutions are conservative, and prove the CFL condition would not be affected by the damping terms. In Section 3, we test various numerical examples including some very tough problems, to show that our algorithm not only controls the oscillations but also preserves high order accuracy. Finally, we give concluding remarks in Section 4.

2. Algorithm. In this section, we construct an essentially oscillation-free discontinuous Galerkin (OFDG) scheme for solving the systems of hyperbolic conservation laws. The idea of the proposed essentially OFDG scheme is to add numerical damping terms to the classical DG method.

2.1. One-dimensional system of hyperbolic conservation laws. In this subsection we consider the one-dimensional hyperbolic system as follows.

$$\begin{cases} \boldsymbol{U}_t + \boldsymbol{F}(\boldsymbol{U})_x = 0, & (x,t) \in (a,b)\times(0,T], \\ \boldsymbol{U}(x,0) = \boldsymbol{U}_0(x), & x \in (a,b) \end{cases} \tag{2.1}$$

with periodic or compactly supported boundary conditions. Here $\boldsymbol{U} = (U_1, \dots, U_m)^T$ and $\boldsymbol{F}(\boldsymbol{U}) = \big(F_1(\boldsymbol{U}), \dots, F_m(\boldsymbol{U})\big)^T$. There exists a large amount of physical models in the form of (2.1). For instance, the *p-system* includes two conservation laws that $\boldsymbol{U} = (U_1, U_2)^T$ and $\boldsymbol{F}(\boldsymbol{U}) = \big(-U_2, -p(U_1)\big)^T$, which arises from the scalar nonlinear wave equation $u_{tt} = p(u_x)_x$. For one-dimensional *compressible Euler equations*, we have $m = 3$, $\boldsymbol{U} = (\rho, \rho u, E)^T$ and $\boldsymbol{F}(\boldsymbol{U}) = \big(\rho u, \rho u^2 + p, u(E+p)\big)^T$, with $E = p/(\gamma-1) + \rho u^2/2$ for the ideal gas.

Now let us turn to the construction of the essentially OFDG scheme for (2.1). First, we take partition of the interval (a,b) into N cells, then we have

$$\begin{aligned} &a = x_{\frac12} < x_{\frac32} < \cdots < x_{N+\frac12} = b, \\ &h_j = x_{j+\frac12} - x_{j-\frac12},\ I_j = \big(x_{j-\frac12}, x_{j+\frac12}\big), \end{aligned} \tag{2.2}$$

Assume that the mesh is quasi-uniform, i.e. there exists a constant $\nu > 0$ such that

$$h \le \nu\,\rho, \quad h = \max_j h_j, \quad \rho = \min_j h_j\,. \tag{2.3}$$

Then we define the finite element space V_h^k as follows

$$V_h^k := \big\{ v \in L^2([a,b]) :\ v|_{I_j} \in P^k(I_j), \quad j = 1, \dots, N \big\}. \tag{2.4}$$

The semi-discrete essentially OFDG scheme for (2.1) is as follows: Seek $\boldsymbol{U}_h(\cdot,t) \in \big[V_h^k\big]^m$ such that for any $\boldsymbol{v}_h \in \big[V_h^k\big]^m$ we have

$$\begin{aligned} \int_{I_j} (\boldsymbol{U}_h)_t \cdot \boldsymbol{v}_h\, dx = &\int_{I_j} \boldsymbol{F}(\boldsymbol{U}_h)\cdot(\boldsymbol{v}_h)_x\, dx - \widehat{\boldsymbol{F}}_{j+\frac12}\cdot(\boldsymbol{v}_h)^-_{j+\frac12} + \widehat{\boldsymbol{F}}_{j-\frac12}\cdot(\boldsymbol{v}_h)^+_{j-\frac12} \\ &- \sum_{l=0}^{k} \frac{\sigma_j^l(\boldsymbol{U}_h)}{h_j} \int_{I_j} \big(\boldsymbol{U}_h - P_h^{l-1}\boldsymbol{U}_h\big)\cdot\boldsymbol{v}_h\, dx\,, \end{aligned} \tag{2.5}$$

where $(\boldsymbol{v}_h)^{\pm}_{j+\frac12} = \boldsymbol{v}_h\big(x^{\pm}_{j+\frac12}\big)$ and $\widehat{\boldsymbol{F}}_{j+\frac12}$ is taken to be the local Lax-Friedrichs flux [8]. Details about the numerical flux can be found in Appendix A.1. P_h^l, $l \ge 0$, is the standard L^2 projection that for any vector function $\boldsymbol{w}$, $P_h^l\boldsymbol{w} \in \big[V_h^l\big]^m$ satisfies

$$\int_{I_j} (P_h^l\boldsymbol{w} - \boldsymbol{w})\cdot\boldsymbol{v}_h\, dx = 0, \quad \forall\, \boldsymbol{v}_h \in [\mathbb{P}^l(I_j)]^m. \tag{2.6}$$

Here we define $P_h^{-1} = P_h^0$. The damping coefficients $\sigma_j^l \ge 0$ are chosen carefully such that they are small in smooth region, and becomes large near the discontinuity. In this paper, they are taken as follows

$$\sigma_j^l = \frac{2\,(2l+1)}{(2k-1)}\frac{h^l}{l!} \max_{1\le s\le m} \Big([\![\partial_x^l V_s]\!]^2_{j-\frac12} + [\![\partial_x^l V_s]\!]^2_{j+\frac12} \Big)^{\frac12}, \tag{2.7}$$

where $[\![v]\!]_{j+\frac{1}{2}} = v\big(x^+_{j+\frac{1}{2}}\big) - v\big(x^-_{j+\frac{1}{2}}\big)$ denotes the jump of v at $x = x_{j+\frac{1}{2}}$. The variables $\partial_x^l \boldsymbol{V} = \big(\partial_x^l V_1, \ldots, \partial_x^l V_m\big)^T$ are given by $\partial_x^l \boldsymbol{V} = \boldsymbol{R}^{-1}\partial_x^l \boldsymbol{U}_h$ on $x_{j+\frac{1}{2}}$, where $\boldsymbol{R}^{-1}$ is the matrix derived from the characteristic decomposition $\boldsymbol{F}'\big(\overline{(\boldsymbol{U}_h)}_{j+\frac{1}{2}}\big) = \boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^{-1}$, and $\overline{(\cdot)}_{j+\frac{1}{2}}$ stands for some average at $x_{j+\frac{1}{2}}$, such as the arithmetic mean or the Roe average (which is used in the numerical experiments later). In particular, for one-dimensional compressible Euler equations of the ideal gas, we have $\boldsymbol{R}^{-1}$ defined as

$$\boldsymbol{R}^{-1} = \big(\gamma-1\big)c^{-1}\begin{pmatrix} \frac{1}{2}uc + \frac{1}{4}(\gamma-1)u^2 & -\frac{1}{2}(\gamma-1)u - \frac{1}{2}c & \frac{1}{2}(\gamma-1) \\ c^2 - \frac{1}{2}(\gamma-1)u^2 & (\gamma-1)u & 1-\gamma \\ -\frac{1}{2}uc + \frac{1}{4}(\gamma-1)u^2 & -\frac{1}{2}(\gamma-1)u + \frac{1}{2}c & \frac{1}{2}(\gamma-1) \end{pmatrix}, \tag{2.8}$$

where u and $c = \sqrt{\gamma p/\rho}$ are computed with the quantities $\overline{(\boldsymbol{U}_h)}_{j+\frac{1}{2}}$.

Remark 2.1. *The formulation of the damping coefficients given in* (2.7) *originates from our previous work [19]. The key idea in constructing the damping coefficients is to ensure they are small in the smooth region, and become large near discontinuities. A natural and economical way is to use the jumps of the adjacent elements in the DG framework. Therefore, We construct the damping coefficients by using the jumps of the numerical solution at the element interfaces. Obviously, the formulation of damping coefficients is not unique because we can also use other indicators to detect the smoothness of the numerical solution.*

Without the extra damping terms in (2.5), the scheme is exactly the standard DG scheme [11] and it would generate some spurious oscillations when there is a discontinuity. This may lead to severe numerical difficulties such as blow-ups of numerical solution when negative pressure and/or density occurs in computing compressible Euler equations.

Proposition 2.1. *The essentially OFDG scheme* (2.5) *preserves the conservation of quantities $\boldsymbol{U}_h$ as time evolves.*

Proof. Take $\boldsymbol{v}_h = (1, 0, \ldots, 0)^T$ in (2.5), we have

$$\begin{aligned} \frac{d}{dt}\int_{I_j} (U_1)_h\, dx &= -\,(\widehat{F}_1)_{j+\frac{1}{2}} + (\widehat{F}_1)_{j-\frac{1}{2}} - \sum_{l=0}^{k} \frac{\sigma_j^l}{h_j}\int_{I_j} \big((U_1)_h - P_h^{l-1}(U_1)_h\big)\, dx \\ &= -\,(\widehat{F}_1)_{j+\frac{1}{2}} + (\widehat{F}_1)_{j-\frac{1}{2}}\,. \end{aligned}$$

Sum it over j, with the periodic or the compactly supported boundary conditions, we obtain the conservation of U_1 that

$$\frac{d}{dt}\int_a^b (U_1)_h\, dx = 0.$$

Similarly, we can obtain the conservation of other components. ∎

For the error estimates of hyperbolic systems with smooth solutions, we can follow [30] to obtain similar results under suitable assumptions.

Theorem 2.1. *For the symmetrizable system of conservation laws* (2.1)*, assume that the solution $\boldsymbol{U}$ and the flux function $\boldsymbol{F}(\boldsymbol{U})$ are sufficiently smooth with bounded derivatives. Let $\boldsymbol{U}_h$ be the numerical solution of the semi-discrete essentially OFDG*

scheme (2.5). *For the quasi-uniform meshes, if the finite element space* V_h^k *is of piecewise polynomials of degree* $k \geq 1$*, then for small enough* h *there holds the following estimate:*

$$\|\boldsymbol{U}(t) - \boldsymbol{U}_h(t)\|_{L^2([a,b])} \leq C h^{k+\frac{1}{2}}, \tag{2.9}$$

where the positive constant C *is independent of* h.

Proof. We give a detailed proof in Appendix A.2. ∎

2.2. Multidimensional system of hyperbolic conservation laws. In this subsection, we extend the one-dimensional essentially OFDG scheme (2.5) to multi-dimensional hyperbolic systems. The governing equations are given as follows

$$\begin{cases} \boldsymbol{U}_t + \sum_{i=1}^{d} \boldsymbol{F}_i(\boldsymbol{U})_{x_i} = 0, & (\boldsymbol{x}, t) \in \Omega \times (0, T], \\ \boldsymbol{U}(\boldsymbol{x}, 0) = \boldsymbol{U}_0(\boldsymbol{x}), & \boldsymbol{x} \in \Omega, \end{cases} \tag{2.10}$$

with periodic or compactly supported boundary conditions. $\boldsymbol{x} = (x_1, \ldots, x_d)^T \in \Omega$ where the domain Ω is open bounded in $\mathbb{R}^d$, and $\boldsymbol{U} = (U_1, \ldots, U_m)^T$, $\boldsymbol{F}_i(\boldsymbol{U}) = \big(F_{i,1}(\boldsymbol{U}), \ldots, F_{i,m}(\boldsymbol{U})\big)^T$.

Assume we have the partition $\mathcal{T}_h$ of Ω and $\mathcal{T}_h$ is regular. We still adopt the notations similar to Section 2.1 without causing any ambiguities.

$h = \max\limits_{K \in \mathcal{T}_h} h_K$, $h_K =$ diam K, $\rho = \min\limits_{K \in \mathcal{T}_h} \rho_K$, ρ_K is the diameter of the sphere inscribed in K.

The essentially OFDG scheme for (2.10) is presented as follows: Find $\boldsymbol{U}_h(\cdot, t) \in \big[V_h^k\big]^m$ such that

$$\begin{aligned} \int_K (\boldsymbol{U}_h)_t \cdot \boldsymbol{v}_h \, d\boldsymbol{x} = & \sum_{i=1}^{d} \left(\int_K \boldsymbol{F}_i(\boldsymbol{U}_h) \cdot (\boldsymbol{v}_h)_{x_i} \, d\boldsymbol{x} - \int_{\partial K} \widehat{\boldsymbol{F}}_i(\boldsymbol{U}_h) n_i \cdot \boldsymbol{v}_h \, dS \right) \\ & - \sum_{l=0}^{k} \frac{\sigma_K^l}{h_K} \int_K \left(\boldsymbol{U}_h - P_h^{l-1} \boldsymbol{U}_h \right) \cdot \boldsymbol{v}_h \, d\boldsymbol{x}, \quad \forall \, \boldsymbol{v}_h \in \big[V_h^k\big]^m, \end{aligned} \tag{2.11}$$

where $\widehat{\boldsymbol{F}}_i(\boldsymbol{U}_h)$ are taken as the Lax-Friedrichs flux on the element interfaces, and $\boldsymbol{n} = (n_1, \ldots, n_d)^T$ is the unit outward normal with respect to ∂K. V_h^k is the finite element space containing polynomials of degree not greater than k, i.e.

$$V_h^k := \{ v \in L^2(\Omega) : v|_K \in P^k(K), \quad \forall \, K \in \mathcal{T}_h \}. \tag{2.12}$$

P_h^l is the standard L^2 projection into V_h^l, $l \geq 0$ and we define $P_h^{-1} = P_h^0$. The damping coefficients σ_K^l are given as follows.

$$\sigma_K^l = \frac{2\,(2l+1)}{(2k-1)} \frac{h^l}{l!} \max_{1 \leq s \leq m} \sum_{|\boldsymbol{\alpha}| = l} \left(\frac{1}{N_e} \sum_{\boldsymbol{v} \in \partial K} \left([\![\partial^{\boldsymbol{\alpha}} V_s]\!]\big|_{\boldsymbol{v}} \right)^2 \right)^{\frac{1}{2}}, \tag{2.13}$$

where the vector $\boldsymbol{\alpha}$ is the multiindex of order

$$|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_d,$$

and $\partial^{\boldsymbol{\alpha}} w$ is defined as

$$\partial^{\boldsymbol{\alpha}} w = \frac{\partial^{|\boldsymbol{\alpha}|} w}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}} = \partial_{x_1}^{\alpha_1} \cdots \partial_{x_d}^{\alpha_d} w \,.$$

The variables $\partial^{\boldsymbol{\alpha}} \boldsymbol{V}\big|_{\boldsymbol{v}} = \big(\partial^{\boldsymbol{\alpha}} V_1, \ldots, \partial^{\boldsymbol{\alpha}} V_m\big)^T\big|_{\boldsymbol{v}}$ are obtained by $\partial^{\boldsymbol{\alpha}} \boldsymbol{V}\big|_{\boldsymbol{v}} = \boldsymbol{R}^{-1}\big|_{\boldsymbol{n}_e} \partial^{\boldsymbol{\alpha}} \boldsymbol{U}_h\big|_{\boldsymbol{v}}$, where $\boldsymbol{R}^{-1}\big|_{\boldsymbol{n}_e}$ is the matrix derived from the characteristic decomposition

$$\sum_{i=1}^{d} n_i \boldsymbol{F}_i'\big(\overline{(\boldsymbol{U}_h)}\big) = \boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^{-1}$$

on the element interface, $\boldsymbol{n}_e = (n_1, \ldots, n_d)$ is the unit out forward normal vector of the element interface, and $\overline{(\boldsymbol{U}_h)}$ is the arithmetic average or the Roe average (which is used in the numerical experiments later) at the element interface. N_e is the number of edges of the element K and $\boldsymbol{v} \in K$ are the vertices of K. $[\![w]\!]|_{\boldsymbol{v}}$ denotes the jump of the function w on the vertex $\boldsymbol{v}$. It should be noted that w may have several jumps on the vertex $\boldsymbol{v}$, and we only compute the jump between the element K and its adjacent neighbors. More specifically, we take the two dimensional case as an illustration example in the following.

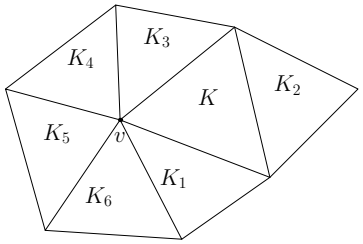


FIG. 2.1. *Illustrating graph for the jumps in the damping coefficient σ_K^l defined in* (2.13).

In Figure 2.1, we consider the jump of the function w on the vertex $\boldsymbol{v} \in K$. The triangular element K has three edges thus $N_e = 3$. The adjacent neighbors of element K are K_1, K_2, K_3, then we have

$$\Big([\![\partial^{\boldsymbol{\alpha}} w]\!]\big|_{\boldsymbol{v}}\Big)^2 = \big[(\partial^{\boldsymbol{\alpha}} w|_K - \partial^{\boldsymbol{\alpha}} w|_{K_1})^2 + (\partial^{\boldsymbol{\alpha}} w|_K - \partial^{\boldsymbol{\alpha}} w|_{K_3})^2\big]\Big|_{\boldsymbol{v}}.$$

Note that we do not take the elements K_4, K_5, K_6 into consideration, though we still have jumps between the element K and them on the vertex $\boldsymbol{v}$. Also, there are various choices of $\boldsymbol{R}$ and $\boldsymbol{R}^{-1}$ in the characteristic decomposition. Throughout this paper, for two-dimensional compressible Euler equations, $\boldsymbol{U} = (\rho, \rho u, \rho v, E)^T$, $\boldsymbol{F}_1 = \big(\rho u, \rho u^2 + p, \rho uv, u(E+p)\big)^T$, $\boldsymbol{F}_2 = \big(\rho v, \rho uv, \rho v^2 + p, v(E+p)\big)^T$, we take $\boldsymbol{R}^{-1}$ as follows.

(2.14)

$$\boldsymbol{R}^{-1} = \big(\gamma - 1\big)c^{-1} \begin{pmatrix} \frac{1}{2}(B_2 + \hat{u}c) & -\frac{1}{2}\big(B_1 u + n_1 c\big) & -\frac{1}{2}\big(B_1 v + n_2 c\big) & \frac{1}{2}B_1 \\ \hat{v}c & n_2 c & -n_1 c & 0 \\ c^2 - B_2 & B_1 u & B_1 v & -B_1 \\ \frac{1}{2}(B_2 - \hat{u}c) & -\frac{1}{2}\big(B_1 u - n_1 c\big) & -\frac{1}{2}\big(B_1 v - n_2 c\big) & \frac{1}{2}B_1 \end{pmatrix}$$

where $\hat{u}$, $\hat{v}$, c and B_1 are given as

$$\hat{u} = un_1 + vn_2, \quad \hat{v} = -un_2 + vn_1, \quad c = \sqrt{\frac{\gamma p}{\rho}}, \quad B_1 = \gamma - 1, \quad B_2 = B_1 \frac{u^2 + v^2}{2}.$$

REMARK 2.2. *The characteristic decomposition makes the algorithm complicated and also requires extra computational cost. A natural choice is to add the damping term to each component based on the current component. However, we have investigated many numerical examples and have observed that, even though this simpler approach works well for many problems, it does not work well for some of the tough problems. After testing a vast amount of problems numerically, we have decided to adopt the characteristic decomposition in the algorithm.*

PROPOSITION 2.2. *The essentially OFDG scheme* (2.11) *preserves the conservation of quantities* $\boldsymbol{U}_h$ *as time evolves.*

Proof. Take $v_h = (1, 0, \ldots, 0)^T$ in (2.11), we have

$$\begin{aligned}\frac{d}{dt}\int_K (U_1)_h \, d\boldsymbol{x} &= -\sum_{i=1}^{d} \int_{\partial K} \widehat{\boldsymbol{F}}_i(\boldsymbol{U}_h) n_i \cdot \boldsymbol{v}_h \, dS - \sum_{l=0}^{k} \frac{\sigma_K^l}{h_K} \int_K \left(\boldsymbol{U}_h - P_h^{l-1}\boldsymbol{U}_h\right) \cdot \boldsymbol{v}_h \, d\boldsymbol{x}, \\ &= -\sum_{i=1}^{d} \int_{\partial K} \widehat{F}_{i,1}(\boldsymbol{U}_h) n_i \, dS \,.\end{aligned}$$

Take summation over $K \in \mathcal{T}_h$, with the periodic or the compactly supported boundary conditions, we obtain the following conservation result for the DG scheme (2.11).

$$\frac{d}{dt} \sum_{K \in \mathcal{T}_h} \int_K (U_1)_h \, d\boldsymbol{x} = 0.$$

Similarly, we can obtain the conservation of the other components. ∎

REMARK 2.3. *In this framework, we add the damping terms to all elements including those in the smooth region. One can also use the idea of the troubled cell indicators to determine the trouble cells and add the damping terms in these cells only, then it could reduce the computational cost. This approach is feasible in implementation and it should also produce satisfactory numerical results. However, we have not pursued this approach in this paper, as such treatments would require an adjustment of some parameters (e.g. the TVB trouble cell indicators). Besides, it seems difficult to theoretically prove some properties of the DG schemes with such treatments, even for the scalar case [19].*

2.3. Time discretization. In some numerical examples, with the explicit Runge-Kutta method as the time-stepping method, the time step size of the essentially OFDG schemes (2.5) and (2.11) could be very restricted for the sake of stability. Thus, we would like to use an appropriate time discretization method which could relax this restriction. For simplicity, we only consider the scalar case, i.e. $m = 1$ in (2.1) and (2.10). We adopt the Legendre orthogonal basis functions denoted as $\phi_{K,l}, l = 0, \cdots, p$ on each element K in finite element space. Now we assume

$$\boldsymbol{U}_h(\boldsymbol{x}) = \sum_{K \in \mathcal{T}_h} \sum_{l=0}^{p} u_{K,l} \, \phi_{K,l}(\boldsymbol{x}).$$

With the aforementioned spatial discretization, we obtain the following ODE system

$$(\boldsymbol{u}_h)_t + D(\boldsymbol{u}_h) + \Sigma(\boldsymbol{u}_h)\boldsymbol{u}_h = 0, \tag{2.15}$$

where $\boldsymbol{u}_h$ denotes the coefficients of the basis functions, $D(\boldsymbol{u}_h)$ is the DG approximation of $\sum_{i=1}^{d} \boldsymbol{F}_i(\boldsymbol{U})_{x_i}$ and $\Sigma(\boldsymbol{u}_h)$ is a diagonal matrix coming from the damping terms. Since the damping term vanishes for $u_{K,0}$ from the definition, then we only need to consider the integration for the remaining coefficients. In the following, we set

$$\boldsymbol{u}_h = \big(u_{h,0}, \widehat{\boldsymbol{u}}_h\big)^T, \quad D(\boldsymbol{u}_h) = \big(D_0(\boldsymbol{u}_h), \widehat{D}(\boldsymbol{u}_h)\big)^T, \quad \Sigma(\boldsymbol{u}_h) = \begin{pmatrix} 0 & \\ & \widehat{\Sigma}(\boldsymbol{u}_h) \end{pmatrix},$$

where $\widehat{\boldsymbol{u}}_h$ denotes the coefficients of the basis functions except the first basis function, and $\widehat{D}(\boldsymbol{u}_h)$ denotes the vector of $D(\boldsymbol{u}_h)$ without $D_0(\boldsymbol{u}_h)$, and $\widehat{\Sigma}(\boldsymbol{u}_h)$ is the submatrix of $\Sigma(\boldsymbol{u}_h)$. Then the ODE system (2.15) can be rewritten as

$$\big(u_{h,0}\big)_t + D_0(\boldsymbol{u}_h) = 0, \tag{2.16a}$$

$$(\widehat{\boldsymbol{u}}_h)_t + \widehat{D}(\boldsymbol{u}_h) + \widehat{\Sigma}(\boldsymbol{u}_h)\widehat{\boldsymbol{u}}_h = 0\,. \tag{2.16b}$$

For the ODE system (2.16), one can apply the explicit Runge-Kutta time discretization for the time evolution directly. However, for some tough problems, the damping terms would take effect and lead to a very restricted time step size. Thus, to reduce the influence of the damping terms on the CFL condition, we follow the idea of exponential Runge-Kutta methods (see [16]) and (2.16b) is rewritten as follows.

$$\big(\widehat{\boldsymbol{u}}_h\big)_t + \widehat{D}(\boldsymbol{u}_h) + \big(\widehat{\Sigma}(\boldsymbol{u}_h) - a_0\boldsymbol{I}\big)\widehat{\boldsymbol{u}}_h + a_0\widehat{\boldsymbol{u}}_h = 0, \tag{2.17}$$

where $a_0 > 0$ is a constant and I is the identity matrix. Then we obtain the following exponential form.

$$\frac{d}{dt}\big(e^{a_0 t}\widehat{\boldsymbol{u}}_h\big) + e^{a_0 t}\big(\widehat{D}(\boldsymbol{u}_h) + (\widehat{\Sigma}(\boldsymbol{u}_h) - a_0\boldsymbol{I})\widehat{\boldsymbol{u}}_h\big) = 0, \tag{2.18}$$

and we can apply the explicit Runge-Kutta method to discretize the ODE system (2.18). For example, the classical third-order SSP RK method [26] is given as follows.

$$u_{h,0}^{(1)} = u_{h,0}^n - \tau D_0(\boldsymbol{u}_h^n), \tag{2.19a}$$

$$\widehat{\boldsymbol{u}}_h^{(1)} = e^{-a_0\tau}\big(\widehat{\boldsymbol{u}}_h^n - \tau\big(\widehat{D}(\boldsymbol{u}_h^n) + (\widehat{\Sigma}(\boldsymbol{u}_h^n) - a_0\boldsymbol{I})\widehat{\boldsymbol{u}}_h^n\big)\big), \tag{2.19b}$$

$$u_{h,0}^{(2)} = \frac{3}{4}u_{h,0}^n + \frac{1}{4}\tau\big(u_{h,0}^{(1)} - D_0(\boldsymbol{u}_h^{(1)})\big), \tag{2.19c}$$

$$\widehat{\boldsymbol{u}}_h^{(2)} = \frac{3}{4}e^{-\frac{1}{2}a_0\tau}\widehat{\boldsymbol{u}}_h^n + \frac{1}{4}e^{\frac{1}{2}a_0\tau}\Big(\widehat{\boldsymbol{u}}_h^{(1)} - \tau\big(\widehat{D}(\boldsymbol{u}_h^{(1)}) + (\widehat{\Sigma}(\boldsymbol{u}_h^{(1)}) - a_0\boldsymbol{I})\widehat{\boldsymbol{u}}_h^{(1)}\big)\Big), \tag{2.19d}$$

$$u_{h,0}^{n+1} = \frac{1}{3}u_{h,0}^n + \frac{2}{3}\tau\big(u_{h,0}^{(2)} - D_0(\boldsymbol{u}_h^{(2)})\big), \tag{2.19e}$$

$$\widehat{\boldsymbol{u}}_h^{n+1} = \frac{1}{3}e^{-a_0\tau}\widehat{\boldsymbol{u}}_h^n + \frac{2}{3}e^{-\frac{1}{2}a_0\tau}\Big(\widehat{\boldsymbol{u}}_h^{(2)} - \tau\big(\widehat{D}(\boldsymbol{u}_h^{(2)}) + (\widehat{\Sigma}(\boldsymbol{u}_h^{(2)}) - a_0\boldsymbol{I})\widehat{\boldsymbol{u}}_h^{(2)}\big)\Big). \tag{2.19f}$$

Before we proceed, we make the following assumption on (2.16):

ASSUMPTION 2.1. *Assume the standard DG scheme (without damping terms) with Euler forward time discretization satisfies the strong stability: There exists $\tau_E > 0$, such that for any $0 < \tau \le \tau_E$, then*

$$\|u_{h,0} - \tau D_0(\boldsymbol{u}_h)\| \le \|u_{h,0}\|, \quad \|\widehat{\boldsymbol{u}}_h - \tau\widehat{D}(\boldsymbol{u}_h)\| \le \|\widehat{\boldsymbol{u}}_h\|. \tag{2.20}$$

For the first unknown $u_{h,0}$, we already have the stability results by the assumption (2.20). Now we present the stability result of $\widehat{\boldsymbol{u}}_h$ in the following.

PROPOSITION 2.3. *For the third-order RK scheme* (2.19)*, if we have*

$$0 < \tau < \tau_E \text{ and } a_0 \geq \frac{1}{2}\max\left\{\|\widehat{\Sigma}(\boldsymbol{u}_h^n)\|_\infty, \|\widehat{\Sigma}(\boldsymbol{u}_h^{(1)})\|_\infty, \|\widehat{\Sigma}(\boldsymbol{u}_h^{(2)})\|_\infty\right\}, \tag{2.21}$$

then we can obtain the following stability results.

$$\|\widehat{\boldsymbol{u}}_h^{(1)}\| \leq e^{-a_0\tau}(1+a_0\tau)\|\widehat{\boldsymbol{u}}_h^n\| \leq \|\widehat{\boldsymbol{u}}_h^n\|, \tag{2.22a}$$

$$\|\widehat{\boldsymbol{u}}_h^{(2)}\| \leq e^{-\frac{1}{2}a_0\tau}\Big(1+\frac{1}{2}a_0\tau+\frac{1}{4}a_0^2\tau^2\Big)\|\widehat{\boldsymbol{u}}_h^n\| \leq \|\widehat{\boldsymbol{u}}_h^n\|, \tag{2.22b}$$

$$\|\widehat{\boldsymbol{u}}_h^{n+1}\| \leq e^{-a_0\tau}\Big(1+a_0\tau+\frac{1}{2}a_0^2\tau^2+\frac{1}{6}a_0^3\tau^3\Big)\|\widehat{\boldsymbol{u}}_h^n\| \leq \|\widehat{\boldsymbol{u}}_h^n\|. \tag{2.22c}$$

Proof. Since we have

$$a_0 \geq \frac{1}{2}\|\widehat{\Sigma}(\boldsymbol{u}_h^n)\|_\infty \Rightarrow \|\widehat{\Sigma}(\boldsymbol{u}_h^n) - a_0\boldsymbol{I}\|_\infty \leq a_0\,, \tag{2.23}$$

then from (2.19b), we obtain

$$\begin{aligned}\|\widehat{\boldsymbol{u}}_h^{(1)}\| &\leq e^{-a_0\tau}\big(\|\widehat{\boldsymbol{u}}_h^n - \tau\widehat{D}(\boldsymbol{u}_h^n)\| + \tau\|\widehat{\Sigma}(\boldsymbol{u}_h^n) - a_0\boldsymbol{I}\|_\infty\|\widehat{\boldsymbol{u}}_h^n\|\big)\\ &\leq e^{-a_0\tau}\big(1+\tau\|\widehat{\Sigma}(\boldsymbol{u}_h^n) - a_0\boldsymbol{I}\|_\infty)\|\widehat{\boldsymbol{u}}_h^n\| \leq e^{-a_0\tau}(1+a_0\tau)\|\widehat{\boldsymbol{u}}_h^n\|,\end{aligned}$$

Therefore, we obtain (2.22a). Then with (2.19d) and (2.22a), we have

$$\begin{aligned}\|\widehat{\boldsymbol{u}}_h^{(2)}\| &\leq e^{-\frac{1}{2}a_0\tau}(\frac{3}{4}+\frac{1}{4}(1+a_0\tau)^2)\|\widehat{\boldsymbol{u}}_h^n\|\\ &\leq e^{-\frac{1}{2}a_0\tau}\Big(1+\frac{1}{2}a_0\tau+\frac{1}{4}a_0^2\tau^2\Big)\|\widehat{\boldsymbol{u}}_h^n\|\,.\end{aligned}$$

Therefore, we obtain (2.22b). Similarly, by using (2.19c), (2.22a) and (2.22b), we have

$$\begin{aligned}\|\widehat{\boldsymbol{u}}_h^{n+1}\| &\leq e^{-a_0\tau}\Big(\frac{1}{3}+\frac{2}{3}(1+a_0\tau)\Big)\Big(1+\frac{1}{2}a_0\tau+\frac{1}{4}a_0^2\tau^2\Big)\Big)\|\widehat{\boldsymbol{u}}_h^n\|\\ &\leq e^{-a_0\tau}\Big(1+a_0\tau+\frac{1}{2}a_0^2\tau^2+\frac{1}{6}a_0^3\tau^3\Big)\|\widehat{\boldsymbol{u}}_h^n\|.\end{aligned}$$

□

REMARK 2.4. *As mentioned in [16], the exponential function $e^{-a_0\tau}$ in* (2.19) *decays too fast. The numerical solution would be smoothed when the damping terms take effect near the shock discontinuities. To obtain a sharper transition, we recommend to use the modified Runge-Kutta methods, i.e. replacing the exponential functions by the Taylor expansion polynomials in the time discrete scheme.*

$$\widehat{\boldsymbol{u}}_h^{(1)} = \frac{1}{s_1}\big(\widehat{\boldsymbol{u}}_h^n - \tau\big(\widehat{D}(\boldsymbol{u}_h^n) + (\widehat{\Sigma}(\boldsymbol{u}_h^n) - a_0\boldsymbol{I})\widehat{\boldsymbol{u}}_h^n\big)\big), \tag{2.24a}$$

$$\widehat{\boldsymbol{u}}_h^{(2)} = \frac{3}{4s_2}\widehat{\boldsymbol{u}}_h^n + \frac{s_1}{4s_2}\Big(\widehat{\boldsymbol{u}}_h^{(1)} - \tau\big(\widehat{D}(\boldsymbol{u}_h^{(1)}) + (\widehat{\Sigma}(\boldsymbol{u}_h^{(1)}) - a_0\boldsymbol{I})\widehat{\boldsymbol{u}}_h^{(1)}\big)\Big), \tag{2.24b}$$

$$\widehat{\boldsymbol{u}}_h^{n+1} = \frac{1}{3s_1}\widehat{\boldsymbol{u}}_h^n + \frac{2s_2}{3s_1}\Big(\widehat{\boldsymbol{u}}_h^{(2)} - \tau\big(\widehat{D}(\boldsymbol{u}_h^{(2)}) + (\widehat{\Sigma}(\boldsymbol{u}_h^{(2)}) - a_0\boldsymbol{I})\widehat{\boldsymbol{u}}_h^{(2)}\big)\Big), \tag{2.24c}$$

with $s_1 = 1+z+\frac{1}{2}z^2+\frac{1}{6}z^3$, $s_2 = 1+\frac{1}{2}z+\frac{1}{8}z^2+\frac{1}{48}z^3$, $z = a_0\tau$. More details about this modified exponential Runge-Kutta methods can be found in [16].

3. Numerical tests. In this section, we present some numerical examples to demonstrate the effectiveness and robustness of the proposed algorithm. We take the one- and two-dimensional Euler equations as illustration examples and many benchmark problems are considered. Throughout this section, we take the heat capacity ratio $\gamma = 1.4$ unless otherwise specified. Also, we take both the classic fourth order Runge-Kutta method and the modified exponential Runge-Kutta method as our time stepping method in this section, with the default being the classic fourth order Runge-Kutta method as the time stepping method unless otherwise specified. The CFL condition is different for these two kinds of RK methods for the reason that the damping term does affect the CFL condition. In 1D, the CFL condition for the classic fourth order Runge-Kutta method is

$$\tau = \frac{CFL}{\alpha/h + a_0/h}, \quad \alpha = \max_{j,s} \left|(\lambda_s)^{\pm}_{j+\frac{1}{2}}\right|,$$

where τ and h are temporal step size and spatial step size, respectively. $(\lambda_s)^{\pm}_{j+\frac{1}{2}}$, $s = 1, \ldots, m$ are the m real eigenvalues of the Jacobian $\boldsymbol{F}'(\boldsymbol{U})$ at $x^{\pm}_{j+\frac{1}{2}}$, and $a_0 = \max_j \sigma^k_j$, σ^k_j is the damping coefficient defined in (2.7). The time step would be small if a_0 is large. But for the exponential Runge-Kutta or the modified Runge-Kutta method, the CFL condition is normal that

$$\tau = \frac{CFL}{\alpha/h}.$$

The CFL condition for two-dimensional problems is similar.

3.1. One-dimensional problems. In this subsection, we present several one-dimensional numerical results. The phrase *full polynomials* in the Figures means we plot 21 points in each cell, and the phrase *cell averages* means we only plot the cell average within a cell.

Example 1. *We test a smooth problem for one-dimensional Euler equation with periodic boundary conditions. The initial conditions are*

$$\rho(x, 0) = 1 + 0.5(\sin(x))^2, \quad u(x, 0) = 1, \quad p(x, 0) = 2,$$

so that the exact solutions are given as

$$\rho(x, t) = 1 + 0.5(\sin(x - t))^2, \quad u(x, t) = 1, \quad p(x, t) = 2.$$

The computational domain is $\Omega = (0, 2\pi)$ *and the final time* $T = 1.2$.

In Table 3.1, we report the errors and convergence orders of density in L^1, L^2 and L^∞ norms in Example 1. We can see the convergence order is $k+1$ which is optimal. This indicates the damping term would not pollute the order of accuracy which has already been demonstrated in [19].

Example 2. *We consider two well-known Riemann problems for 1D Euler equations. Both of them have the following Riemann type initial conditions:*

$$\boldsymbol{U}(x, 0) = \begin{cases} \boldsymbol{U}_L, & x < 0, \\ \boldsymbol{U}_R, & x > 0. \end{cases}$$

The first test case is Sod's problem [28]. The initial conditions are

$$(\rho_L, u_L, p_L) = (1, 0, 1), \quad (\rho_R, u_R, p_R) = (0.125, 0, 0.1).$$

TABLE 3.1
Density's errors and orders in Example 1.

	N	L^1 error	order	L^2 error	order	L^∞ error	order
P^1	16	1.156E-02	–	1.399E-02	–	2.469E-02	–
	32	1.902E-03	2.604	2.440E-03	2.519	5.433E-03	2.184
	64	3.414E-04	2.478	4.223E-04	2.531	9.492E-04	2.517
	128	6.488E-05	2.396	8.175E-05	2.369	2.188E-04	2.117
	256	1.430E-05	2.181	1.835E-05	2.156	6.206E-05	1.818
	512	3.368E-06	2.086	4.436E-06	2.048	1.644E-05	1.916
P^2	16	1.057E-03	–	1.227E-03	–	2.787E-03	–
	32	1.126E-04	3.230	1.450E-04	3.081	4.664E-04	2.579
	64	1.148E-05	3.294	1.614E-05	3.168	6.112E-05	2.932
	128	1.304E-06	3.138	1.888E-06	3.095	7.599E-06	3.008
	256	1.571E-07	3.054	2.295E-07	3.040	9.394E-07	3.016
	512	1.938E-08	3.019	2.835E-08	3.017	1.165E-07	3.011
P^3	16	4.096E-05	–	5.290E-05	–	1.732E-04	–
	32	1.636E-06	4.646	2.135E-06	4.631	7.824E-06	4.469
	64	7.628E-08	4.423	9.976E-08	4.419	3.998E-07	4.291
	128	4.329E-09	4.139	5.413E-09	4.204	2.168E-08	4.205
	256	2.607E-10	4.053	3.226E-10	4.069	1.276E-09	4.086
	512	1.611E-11	4.016	1.989E-11	4.019	7.759E-11	4.040

The second one is Lax's problem [18]. The initial conditions are

$$(\rho_L, u_L, p_L) = (0.445, 0.698, 3.528), \quad (\rho_R, u_R, p_R) = (0.5, 0, 0.571).$$

For the two problems, we take the same computational domain $\Omega = (-5, 5)$ *and the final time is* $T = 1.3$.

In Figures 3.2 and 3.3, we plot the density profiles for the shock tube problems in Example 2. In both figures, we can see some small oscillations near the shock for full polynomials, while for cell averages, the spurious oscillations are effectively reduced. This is similar to the effect of TVB limiters in obtaining the total variation bounded in the means (TVBM) DG schemes in [10].

EXAMPLE 3. *Consider the Shu-Osher's problem [27]. This is a shock entropy wave interaction problem describing the interaction between a right-moving Mach = 3 shock and sine waves in density. This problem is very suitable for high order numerical schemes because both shocks and complicated smooth flow feature co-exist. The initial conditions are given as*

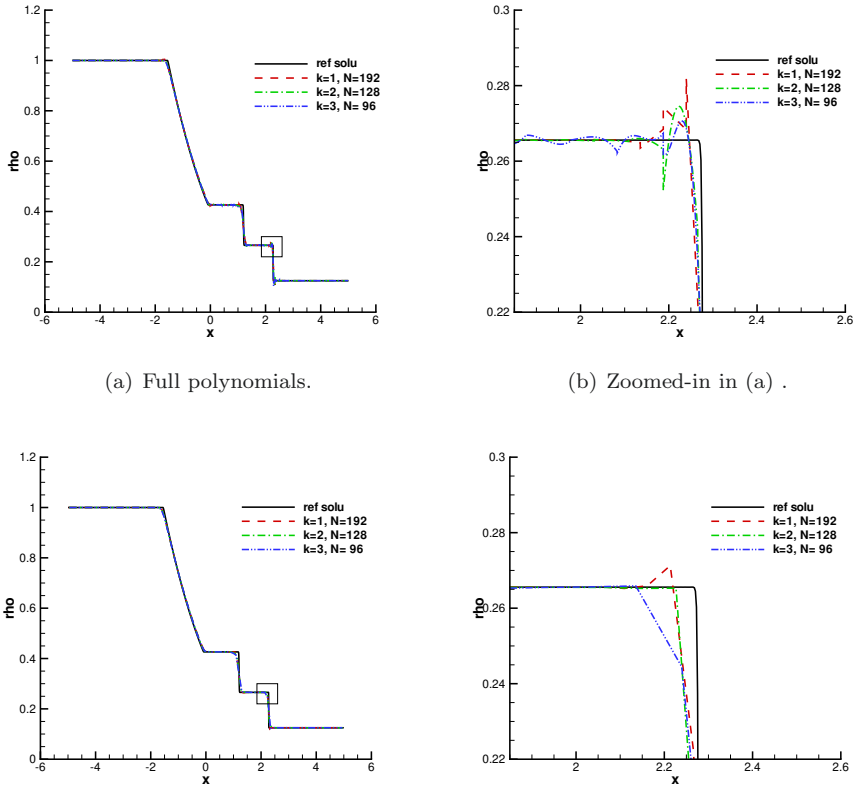
$$\begin{aligned} &\rho(x,0) = 3.857143, \quad u(x,0) = 2.629369, \quad p(x,0) = 10.33333, && x < -4, \\ &\rho(x,0) = 1 + 0.2\sin(5x), \quad u(x,0) = 0, \quad p(x,0) = 1, && x > -4. \end{aligned}$$

The computational domain is $\Omega = (-5, 5)$ *and the final time is* $T = 1.8$.

In Figure 3.4, we show the density profiles when five waves have passed through the shock in Example 3. The reference solution is computed by the fifth order finite difference WENO method with $N = 4096$. With fixed degrees of freedom, the numerical solutions of $k = 2, 3$ behaves better than $k = 1$, indicating that the higher order numerical schemes can resolve the waves better.

EXAMPLE 4. *We consider here the interaction of two blast waves [29]. This problem involves multiple reflections of shocks and rarefactions off the walls. The*

FIG. 3.2. *Density profile, Sod's problem in Example 2, $T = 1.3$.*



(a) Full polynomials.

(b) Zoomed-in in (a) .

(c) Cell averages.

(d) Zoomed-in in (c).

initial conditions are given as

$$\rho(x,0) = 1, \quad u(x,0) = 1, \quad p(x,0) = \begin{cases} 10^3, & 0 < x < 0.1, \\ 10^{-2}, & 0.1 < x < 0.9, \\ 10^2, & 0.9 < x < 1. \end{cases}$$

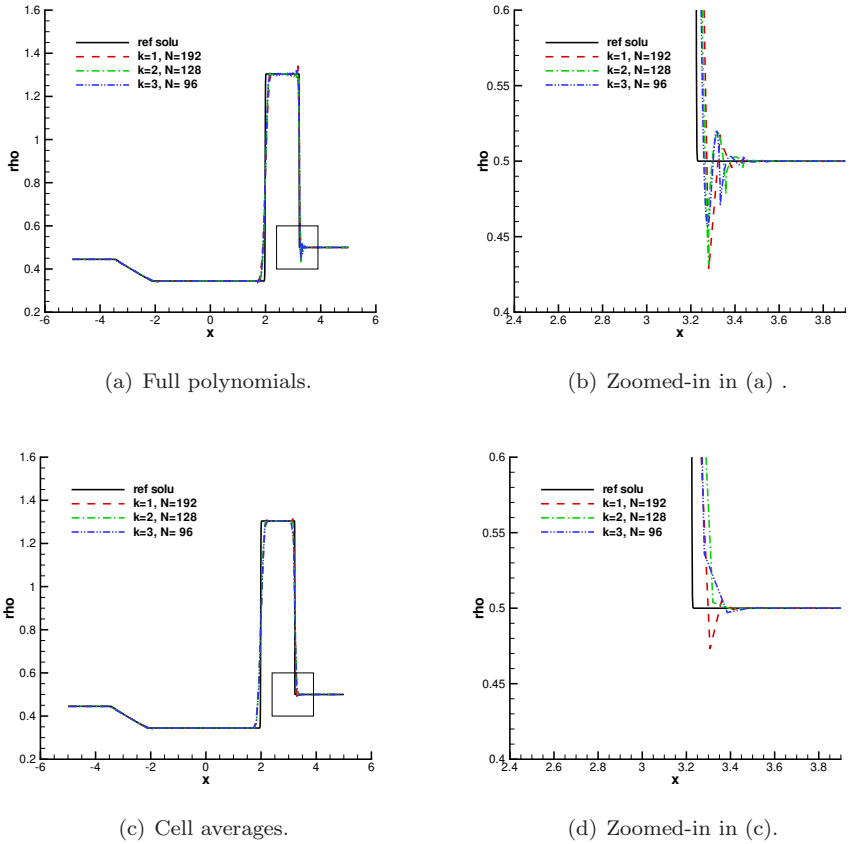
The computational domain is $\Omega = (0,1)$ and the reflective boundary conditions are imposed on both left and right boundaries. The final time is $T = 0.038$.

In Figure 3.5, we show the density profiles of the interaction of two blast waves at $T = 0.038$ in Example 4. The reference solution is computed by the fifth order finite difference WENO method with $N = 4096$. We can see no spurious oscillations near the discontinuities. In Figure 3.6, we also make a comparison between the damping coefficient (2.7) and the following one.

$$\tilde{\sigma}_j^l = \frac{2\,(2l+1)}{(2k-1)} \frac{h^l}{l!} \max_{1 \le s \le m} \left([\![\partial_x^l (U_s)_h]\!]^2_{j-\frac{1}{2}} + [\![\partial_x^l (U_s)_h]\!]^2_{j+\frac{1}{2}} \right)^{\frac{1}{2}}. \tag{3.1}$$

The *Damping* 1 stands for $\tilde{\sigma}_j^l$ in (3.1) and *Damping* 2 stands for σ_j^l in (2.7). The difference between these two damping coefficients is that σ_j^l depends on a characteristic

FIG. 3.3. *Density profile, Lax's problem in Example 2, $T = 1.3$.*



(a) Full polynomials.

(b) Zoomed-in in (a) .

(c) Cell averages.

(d) Zoomed-in in (c).

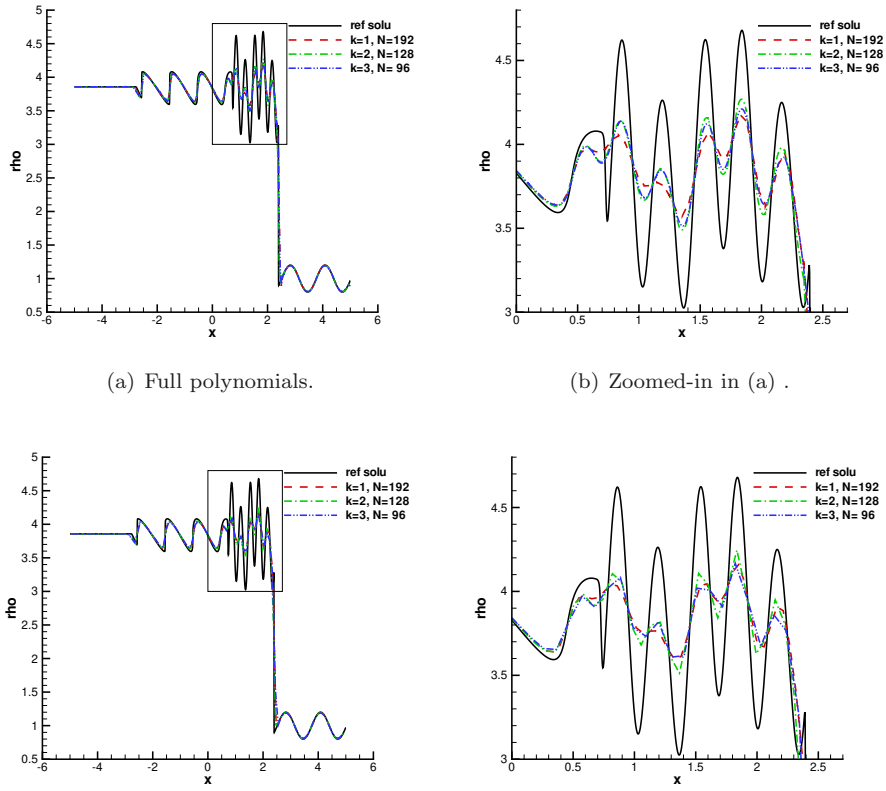
decomposition, while $\tilde{\sigma}_j^l$ depends on the physical variables $\boldsymbol{U}_h$ directly. From Figure 3.6, we can see $\tilde{\sigma}_j^l$ introduces too much damping comparing to σ_j^l in (2.7).

EXAMPLE 5. *We consider the one-dimensional Sedov point blast problem [32] which models the expanding wave by an intense explosion in the perfect gas. This problem is a typical low density problem involving shocks, and in [32] the authors successfully computed this problem by using both the positivity preserving limiter and TVB limiter. The formula of the exact solution can be found in [21]. Initially, we take the density $\rho(x,0) = 1$, the velocity $u(x,0) = 0$ and total energy is $E(x,0) = 10^{-12}$ everywhere except in the center cell that $E(x,0) = E_0/h_0$, $E_0 = 3,200,000$ and h_0 is the length of the center cell. The computational domain is $\Omega = (-2,2)$ and the final time is $T = 10^{-3}$.*

In Figure 3.7, we show the profile of density for the 1D Sedov point blast problem in Example 5 with fixed degrees of freedom. Comparing to the exact solution, the proposed algorithm gives satisfactory numerical results. This indicates our method is robust and can reduce the spurious oscillations effectively.

3.2. Two-dimensional problems. In this subsection, we show the numerical results of some benchmark problems in two dimensions, including the steady state

FIG. 3.4. *Density profile, Shu-Osher's problem in Example 3, $T = 1.8$.*



(a) Full polynomials.

(b) Zoomed-in in (a) .

(c) Cell averages.

(d) Zoomed-in in (c).

problems. Steady state problems are very important both in gas dynamics and in other fields of applications. For some numerical schemes like ENO or TVD schemes, the residual does not settle down to machine zero during the time evolution. We will test some steady state problems to validate the effectiveness and good performance of our method. To this end, we define the *average residual* [35] as

$$Res_A = \sum_{i=1}^{N} \frac{|R1_i| + |R2_i| + |R3_i| + |R4_i|}{4 \times N_x \times N_y}, \tag{3.2}$$

where $R1_i, R2_i, R3_i, R4_i$ are defined as

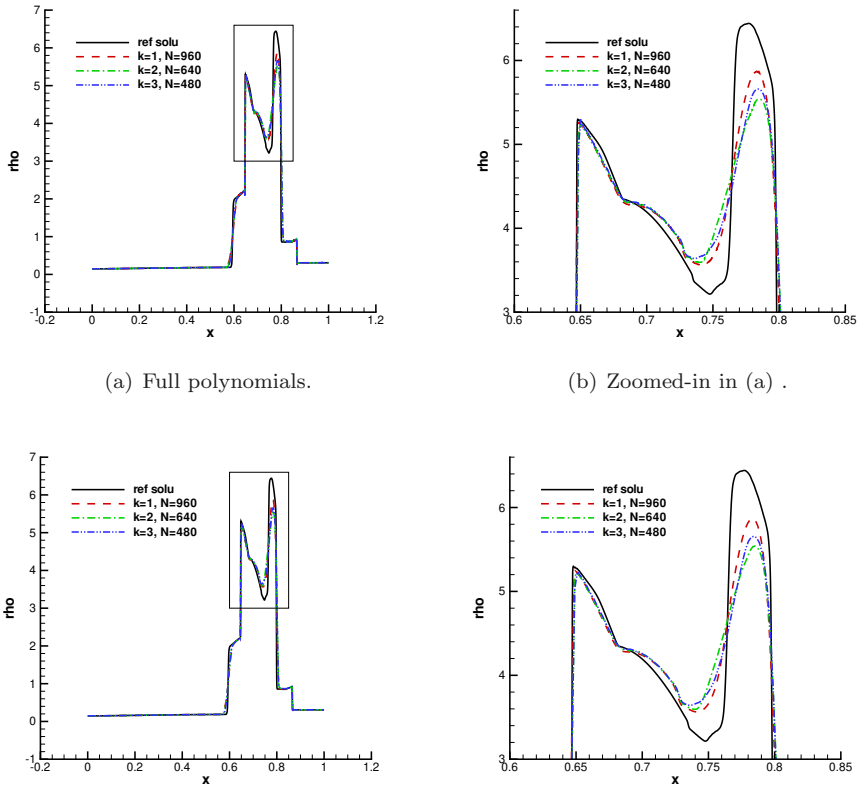
$$R1_i = \frac{\rho_i^{n+1} - \rho_i^n}{\tau}, R2_i = \frac{(\rho u)_i^{n+1} - (\rho u)_i^n}{\tau}, R3_i = \frac{(\rho v)_i^{n+1} - (\rho v)_i^n}{\tau}, R4_i = \frac{E_i^{n+1} - E_i^n}{\tau}.$$

N is the total number of partitioned cells. All figures plotted in this subsection are the cell averages.

EXAMPLE 6. *We test a smooth problem for two-dimensional Euler equations with periodic boundary conditions. The initial conditions are given as*

$$\rho(x, y, t) = 1 + 0.2 \sin(\pi(x + y)), \quad u = 0.7, \quad v = 0.3, \quad p = 1,$$

FIG. 3.5. *Density profile, The two blast waves problem in Example 4, $T = 0.038$.*



(a) Full polynomials.

(b) Zoomed-in in (a) .

(c) Cell averages.

(d) Zoomed-in in (c).

so that the exact solutions are

$$\rho(x, y, t) = 1 + 0.2\sin(\pi(x + y - t)), \quad u = 0.7, \quad v = 0.3, \quad p = 1.$$

The computational domain is $(0, 2) \times (0, 2)$ *and the final time is* $T = 2$.

In Table 3.2, we report the errors and convergence orders of density in L^1, L^2 and L^∞ norms in Example 6. We can see the convergence order is slightly larger than $k+1$ especially in L^2 norm on the coarse meshes and this phenomenon has been explained in [19].

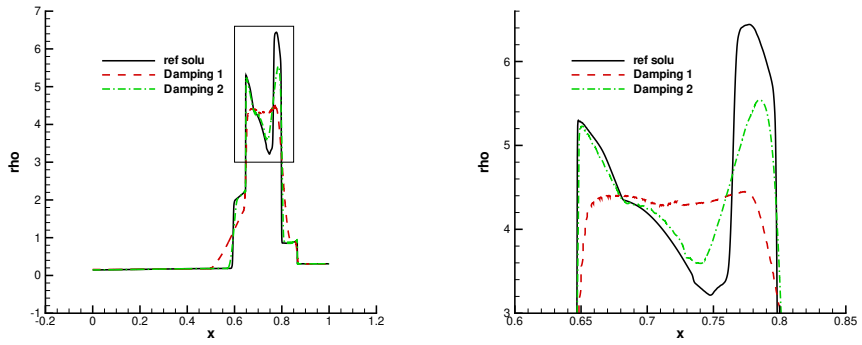
EXAMPLE 7. *We test a steady state smooth problem for two-dimensional Euler equations. The initial conditions are given as*

$$\rho(x, y, 0) = 1 + 0.2\sin(x - y), \quad u(x, y, 0) = 1, \quad v(x, y, 0) = 1, \quad p(x, y, 0) = 1.$$

The boundary conditions are prescribed by the initial conditions so that the exact solutions stay the same as initial conditions as time evolves. The computational domain is $(0, 2) \times (0, 2)$ *and we take the final time* $T = 10$.

In Table 3.3, we report the errors and convergence orders of density in L^1, L^2 and L^∞ norms in Example 7 after a long time simulation. We can see clearly the $(k+1)$-th

FIG. 3.6. *Density profile, Two blast waves problem in Example 4, $k = 2$, $N = 640$. Damping 1 stands for $\tilde{\sigma}_j^l$ in (3.1), and Damping 2 stands for σ_j^l in (2.7).*



(a) Full polynomials. (b) Zoomed-in in (a) .

TABLE 3.2
Density's errors and orders in Example 6, $T = 2$.

	$N_x \times N_y$	L^1 error	order	L^2 error	order	L^∞ error	order
P^1	16×16	1.134E-02	–	1.306E-02	–	2.297E-02	–
	32×32	1.456E-03	2.962	1.896E-03	2.785	4.171E-03	2.461
	64×64	2.201E-04	2.726	2.816E-04	2.751	6.527E-04	2.676
	128×128	3.686E-05	2.578	4.671E-05	2.592	1.099E-04	2.570
	256×256	7.375E-06	2.321	9.451E-06	2.305	3.343E-05	1.718
P^2	16×16	5.494E-04	–	6.267E-04	–	1.142E-03	–
	32×32	4.458E-05	3.623	5.307E-05	3.562	1.477E-04	2.951
	64×64	3.948E-06	3.497	4.988E-06	3.411	1.949E-05	2.922
	128×128	3.891E-07	3.343	5.267E-07	3.244	2.426E-06	3.006
	256×256	4.349E-08	3.162	6.150E-08	3.098	3.004E-07	3.013
P^3	16×16	2.063E-05	–	2.545E-05	–	7.129E-05	–
	32×32	7.493E-07	4.783	9.249E-07	4.559	3.024E-06	4.559
	64×64	3.331E-08	4.492	4.102E-08	4.328	1.506E-07	4.328
	128×128	1.766E-09	4.237	2.198E-09	4.173	8.345E-09	4.173
	256×256	1.041E-10	4.085	1.307E-10	4.071	4.896E-10	4.091

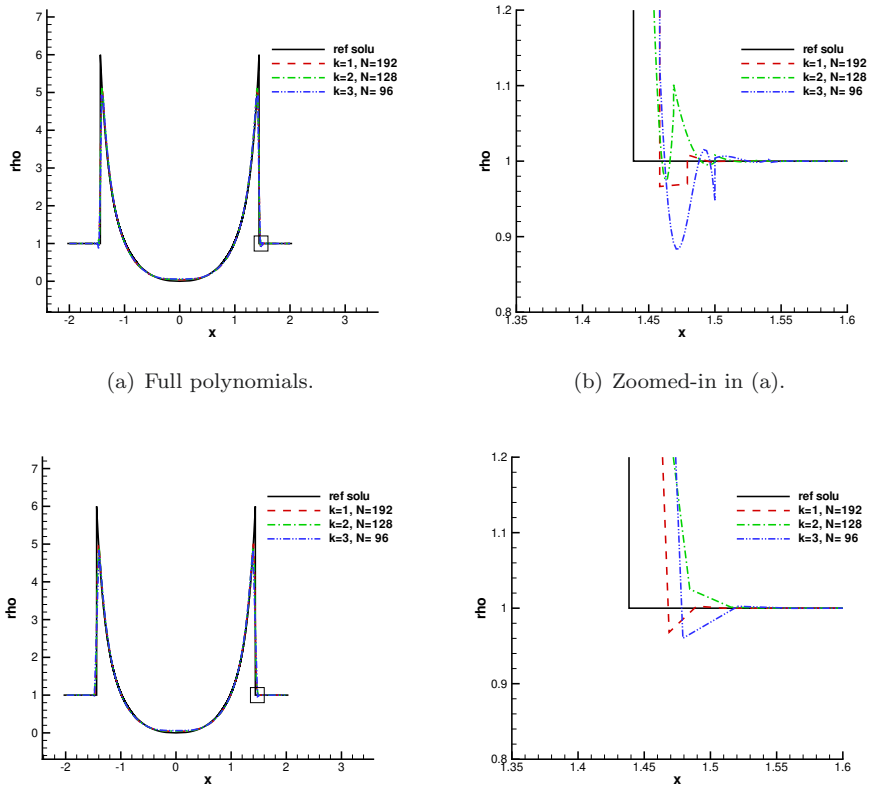
order of convergence for the steady state solution. In Figure 3.8, we show the average residual against time for $k = 1, 2, 3$. The results also verify the good performance of the proposed algorithm.

EXAMPLE 8. *Consider shock reflection problem in [35]. The initial conditions are given as*

$$\rho(x, y, 0) = 1, \quad u(x, y, 0) = 2.9, \quad v(x, y, 0) = 0, \quad p(x, y, 0) = 1/1.4.$$

The computational domain is $(0, 4) \times (0, 1)$. We have inflow boundary conditions on

FIG. 3.7. *Density profile, Sedov point blast problem in Example 5, $k = 2$, $N = 128$.*



(a) Full polynomials.

(b) Zoomed-in in (a).

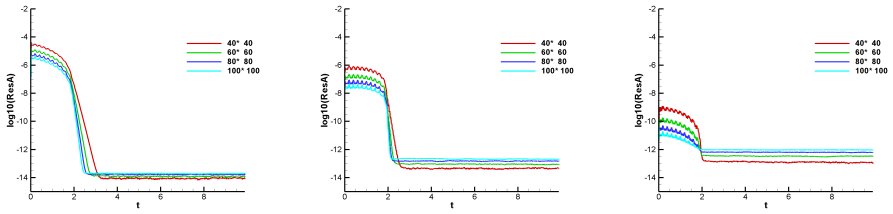
(c) Cell averages.

(d) Zoomed-in in (c).

TABLE 3.3
Density's errors and orders in Example 7.

	$N_x \times N_y$	L^1 error	order	L^2 error	order	L^∞ error	order
P^1	40×40	2.046E-05	–	3.369E-05	–	2.363E-04	–
	60×60	9.043E-06	2.014	1.502E-05	1.992	1.058E-04	1.982
	80×80	5.071E-06	2.011	8.466E-06	1.994	5.971E-05	1.988
	100×100	3.240E-06	2.008	5.425E-06	1.995	3.830E-05	1.990
P^2	40×40	4.552E-07	–	5.755E-07	–	3.030E-06	–
	60×60	1.330E-07	3.034	1.685E-07	3.030	9.011E-07	2.990
	80×80	5.572E-08	3.025	7.062E-08	3.022	3.808E-07	2.994
	100×100	2.841E-08	3.019	3.602E-08	3.017	1.953E-07	2.992
P^3	40×40	1.255E-09	–	1.921E-09	–	1.634E-08	–
	60×60	2.469E-10	4.010	3.773E-10	4.014	3.222E-09	4.005
	80×80	7.799E-11	4.005	1.191E-10	4.010	1.018E-09	4.004
	100×100	3.192E-11	4.003	4.869E-11	4.007	4.167E-10	4.003

FIG. 3.8. *The average residual against time in Example 7.*



(a) P^1-element. (b) P^2-element. (c) P^3-element.

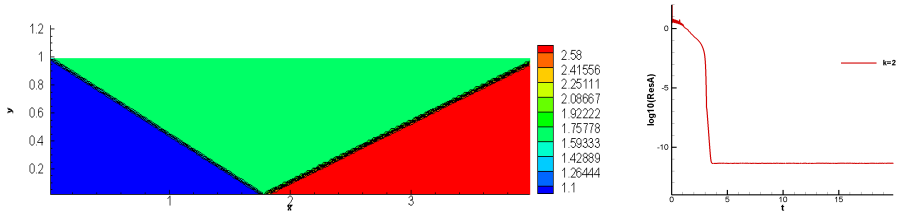
both left and upper boundaries:

$$(\rho, u, v, p) = \begin{cases} (1, 2.9, 0, 1/1.4), & \text{at the left boundary } x = 0, \\ (1.69997, 2.61934, -0.50632, 1.52819), & \text{at the upper boundary } y = 1. \end{cases}$$

We have outflow boundary conditions at the right boundary $x = 4$ and reflection boundary conditions at the bottom boundary $y = 0$. The final time is $T = 20$.

In Figure 3.9, we plot the density contour and the average residual against time for shock reflection problem in Example 8 with $k = 2$, $N_x \times N_y = 200 \times 50$. The average residual starts to remain unchanged at about $t = 3.5$.

FIG. 3.9. *The shock reflection problem in Example 8, $k = 2$, $N_x \times N_y = 200 \times 50$. Left figure: Density contour. Right figure: The average residual against time.*



(a) Density contour. (b) Average residual.

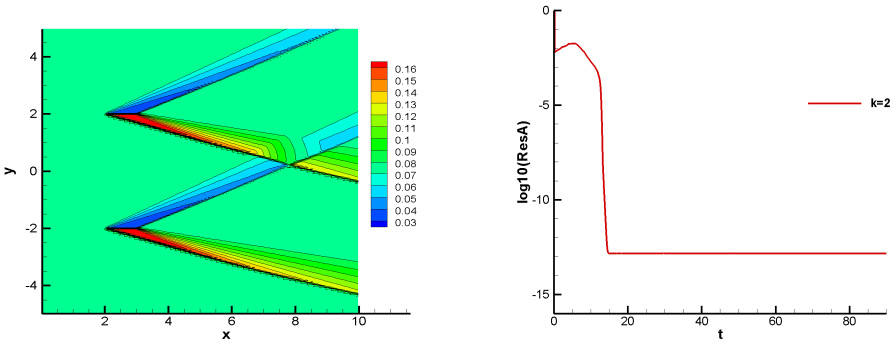
EXAMPLE 9. *Consider a supersonic flow past two plates with an attack angle of 15° [35]. The computational domain is $(0, 10) \times (-5, 5)$ and two plates are set at $x \in (2, 3)$, $y = -2$ and $x \in (2, 3)$, $y = 2$. The initial conditions are given as*

$$\rho(x, y, 0) = 1, \quad u(x, y, 0) = \cos\left(\frac{\pi}{12}\right), \quad v(x, y, 0) = \sin\left(\frac{\pi}{12}\right), \quad p = \frac{1}{\gamma M_\infty^2},$$

where $M_\infty = 3$ is the Mach number of free stream. We have inflow boundary conditions at both left and bottom boundaries and outflow boundary conditions at both right and upper boundaries, and no-penetration boundary conditions are imposed on two plates. We take the final time $T = 100$.

In Figure 3.10, we plot the pressure contour and average residual against time of the steady state solutions in Example 9, $k = 2$, $N_x \times N_y = 200 \times 200$. We can see

FIG. 3.10. *The pressure contour and average residual against time of a supersonic flow past two plates with an attack angle* 15° *in Example 9,* $k = 2$, $N_x \times N_y = 200 \times 200$.



(a) Pressure contour. (b) Average residual.

the residual remains unchanged at around $t = 15$. This again indicates our method works well in getting the residuals down to machine zeros.

EXAMPLE 10. *In this example, we consider the vortex evolution problems in 2D. An isentropic vortex perturbation centered at* (x_c, y_c) *is added to the velocity* (u, v)*, temperature* $(T = p/\rho)$ *and entropy* $(S = ln(p/\rho^\gamma))$ *of the flow, given in the following:*

$$(\delta u, \delta v) = \frac{\varepsilon}{r_c} e^{\alpha(1-\tau^2)}(\bar{y}, -\bar{x}),\ \delta T = -\frac{(\gamma-1)\varepsilon^2}{4\alpha\gamma} e^{2\alpha(1-\tau^2)},\ \delta S = 0, \tag{3.3}$$

where $(\bar{x}, \bar{y}) = (x - x_c, y - y_c)$, $r = (\bar{x}^2 + \bar{y}^2)^{1/2}$ *and* $\tau = r/r_c$. *Here* ε *is the strength of the vortex,* α *controls the decay rate of the vortex and* r_c *is the critical radius that vortex has the maximum strength. We test following two cases:*

(a) Consider the idealized problem similar to [23]: The mean flow is $(\rho, u, v, p) = (1, 1, 1, 1)$ *(diagonal flow). The parameters are taken as* $\varepsilon = 5/2\pi$, $r_c = 1$, $\alpha = 1/2$, $(x_c, y_c) = (4.5, 4.5)$. *The exact solution is the passive convection of the vortex along with the mean velocity. The computational domain is* $[0, 10] \times [0, 10]$ *with the exact boundary conditions. The final time is* $T = 1$.

(b) Consider the shock vortex interactions in 2D [4, 17]. The computational domain is taken as $[0, 2] \times [0, 1]$. *A stationary Mach 1.1 shock is positioned at* $x = 0.5$ *and perpendicular to the x-axis. Its left state is* $(\rho, u, v, p) = (1, 1.1\sqrt{\gamma}, 0, 1)$. *We take the same values of these parameters as in [17] that* $(x_c, y_c) = (0.25, 0.5)$, $\varepsilon = 0.3$, $r_c = 0.05$ *and* $\alpha = 0.204$. *The final time is* $T = 0.8$. *The left and right boundary conditions are inflow and outflow respectively, and reflecting boundary conditions are imposed on the upper and lower boundaries.*

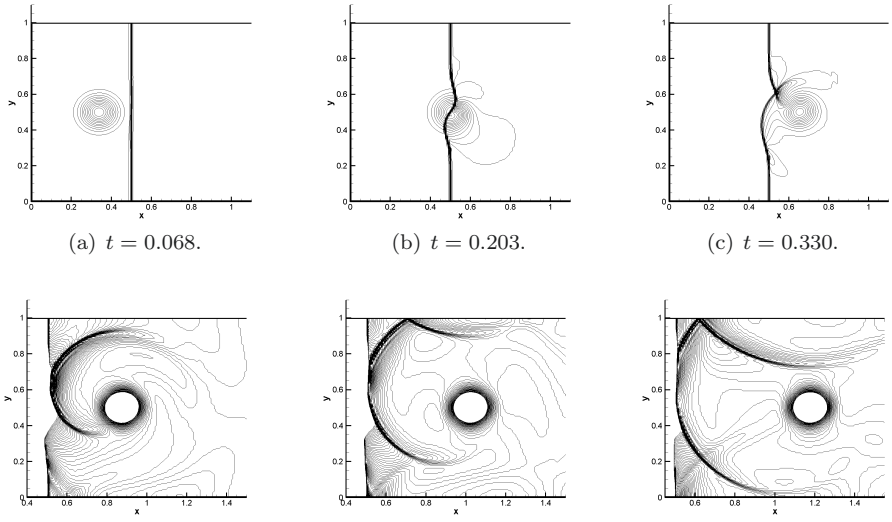
In Table 3.4, we show the error and orders of density for the case (a) in Example 10. The convergence order is slightly less than 3 for $k = 2$, while it is slightly larger than $k+1$ for $k = 1, 3$ at least in L^1 and L^2 norms. In Figure 3.11, we plot the vortex interacted with the stationary shock wave at different time. We can see one branch of the shock bifurcation has been reflected by the upper boundary in the figures $t = 0.662$ and $t = 0.8$. The results are comparable to those in [17].

EXAMPLE 11. *Now let us consider the two-dimensional Sedov point blast problem [32]. As mentioned in Example 5, the formula of exact solution can be found in*

TABLE 3.4
Density's errors and orders of vortex evolution problem in Example 10, case (a), $T = 1$.

	$N_x \times N_y$	L^1 error	order	L^2 error	order	L^∞ error	order
P^1	16×16	5.965E-03	–	1.228E-02	–	7.847E-02	–
	32×32	1.185E-03	2.332	2.414E-03	2.347	1.809E-02	2.117
	64×64	2.032E-04	2.544	4.202E-04	2.522	4.017E-03	2.171
	128×128	3.984E-05	2.351	8.375E-05	2.327	1.098E-03	1.871
	256×256	8.946E-06	2.155	1.929E-05	2.118	2.858E-04	1.942
P^2	16×16	8.704E-04	–	1.619E-03	–	1.830E-02	–
	32×32	9.816E-05	3.148	1.769E-04	3.194	2.397E-03	2.933
	64×64	1.294E-05	2.924	2.351E-05	2.911	3.504E-04	2.774
	128×128	1.888E-06	2.776	3.651E-06	2.687	5.075E-05	2.787
	256×256	2.762E-07	2.773	5.610E-07	2.702	7.650E-06	2.730
P^3	16×16	2.244E-04	–	4.513E-04	–	3.881E-03	–
	32×32	9.874E-06	4.506	2.055E-05	4.457	2.486E-04	3.964
	64×64	4.199E-07	4.555	9.251E-07	4.473	1.522E-05	4.030
	128×128	1.930E-08	4.443	4.382E-08	4.400	8.370E-07	4.185
	256×256	9.510E-10	4.343	2.160E-09	4.342	4.589E-08	4.189

FIG. 3.11. *The pressure contours of the 2D shock vortex interaction problem in Example 10, case (b), $k = 2$, $N_x \times N_y = 256 \times 128$. 30 contours: (a) $t = 0.068$; (b) $t = 0.203$; (c) $t = 0.330$. 90 contours from 1.19 to 1.37: (d) $t = 0.529$; (e) $t = 0.662$; (f) $t = 0.8$.*



(a) $t = 0.068$. (b) $t = 0.203$. (c) $t = 0.330$.

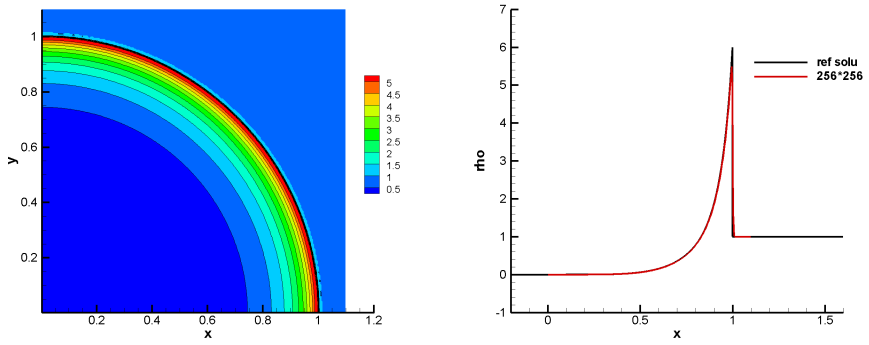
(d) $t = 0.529$. (e) $t = 0.662$. (f) $t = 0.8$.

[21]. The computational domain is $\Omega = (0, 1.1) \times (0, 1.1)$. We take the initial density $\rho(x, y, 0) = 1$, the velocity $u(x, y, 0) = v(x, y, 0) = 0$ and total energy is $E(x, y, 0) = 10^{-12}$ everywhere except in the lower left corner cell that $E(x, 0) = E_0/S_0$ where $E_0 = 0.244816$ and S_0 is the area of the lower left corner cell. The numerical boundary treatment is to extend the DG solutions of ρ, v, E as even functions and u as odd

function with respect to the left boundary, and extend the DG solutions of ρ, u, E as even functions and v as odd function with respect to the bottom boundary. The final time is $T = 1$.

In Figure 3.12, we plot the density profiles of the 2D Sedov point blast problem in Example 11. We take a slice of the numerical solution at $y = 0$ and we can see no spurious oscillations occur near the discontinuity.

FIG. 3.12. *The density profiles of the 2D Sedov point blast problem in Example 11, $k = 2$, $N_x \times N_y = 256 \times 256$.*



(a) Density contour. (b) A slice at $y = 0$ in (a).

EXAMPLE 12. *Consider the double Mach reflection problem [29]. The computational domain is $(0, 4) \times (0, 1)$. The problem describes a Mach 10 shock initially makes a 60° angle with the horizontal wall. The reflecting wall lies at the bottom of the computational domain starting from $x = 1/6$. The initial conditions are given as follows.*

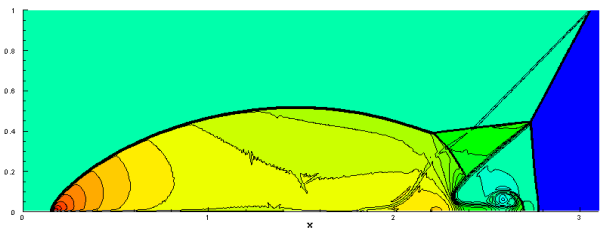
$$(\rho, u, v, p) = \begin{cases} \left(8, 8.25\cos\left(\frac{\pi}{6}\right), -8.25\sin\left(\frac{\pi}{6}\right), 116.5\right), & x < \frac{1}{6} + \frac{y}{\sqrt{3}}, \\ (1.4, 0, 0, 1), & x > \frac{1}{6} + \frac{y}{\sqrt{3}}. \end{cases}$$

We have inflow boundary conditions for the left boundary and outflow boundary condition at the right boundary. For the bottom boundary, the exact post-shock condition is imposed for the part from $x = 0$ to $x = 1/6$ and a reflective boundary condition is used for the rest. For the upper boundary, the post-shock condition is imposed for the part from $x = 0$ to $x = 1/6 + (1 + 20t)/\sqrt{3}$ and the pre-shock condition is used for the rest. The final time is taken to be 0.2.

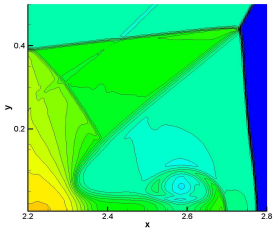
In Figure 3.13, we show the density contours of double Mach reflection problem in Example 12 with $k = 2$ on two kinds of grids. We use the exponential third order Runge-Kutta (2.19) method as the time stepping method. We can see the flow structures are better resolved with the refined mesh and the Kelvin-Helmholtz instability can be seen in the Mach stem.

EXAMPLE 13. *Consider the high Mach number astrophysical jets problem [14, 32]. Since the Mach number of the jet is extremely high, negative pressure and density could easily appear during numerical computation, leading to the crash of the program. To overcome this difficulty, a positivity preserving limiter was developed in [31, 32],*

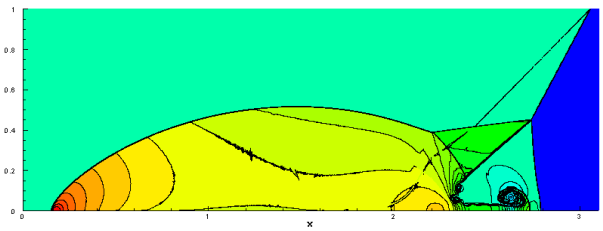
FIG. 3.13. *Density contours of double Mach reflection at $t = 0.2$ in Example 12, 30 contour lines from 1.731 to 20.92, $k = 2$.*



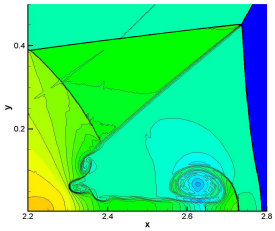
(a) $h_x = h_y = 1/240$.



(b) Mach stem in (a).



(c) $h_x = h_y = 1/960$.



(d) Mach stem in (c).

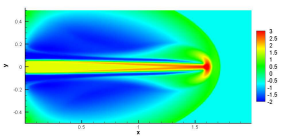
which successfully preserved the positivity of the relevant physical quantities. Now we compute the high Mach number astrophysical jets without the positivity preserving limiter. We consider two cases: Mach = 80 and Mach = 2000 in the following. Note that the heat capacity ratio $\gamma = 5/3$ in this example.

For the Mach 80 problem, the computational domain is $(0, 2)\times(-0.5, 0.5)$ with the ambient gas $(\rho, u, v, p) = (0.5, 0, 0, 0.4127)$ initially. The jet locates at $y \in (-0.05, 0.05)$, $x = 0$, and the physical values are $(\rho, u, v, p) = (5, 30, 0, 0.4127)$. The boundary conditions of the rest boundaries are outflow. The terminal time is 0.07.

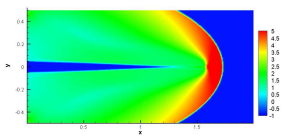
For the Mach 2000 problem, the computational domain is $(0, 1) \times (-0.25, 0.25)$, which is full of the ambient gas with $(\rho, u, v, p) = (0.5, 0, 0, 0.4127)$ initially. The jet locates on $y \in (-0.05, 0.05)$, $x = 0$. The physical values of jet are $(\rho, u, v, p) = (5, 800, 0, 0.4127)$. The boundary conditions of the rest boundaries are outflow. The terminal time is 0.001.

In Figures 3.14 and 3.15, we show the numerical results of Mach 80 and Mach 2000 astrophysical jets in Example 13, $k = 2$, $N_x \times N_y = 320 \times 160$. We obtain satisfactory results comparing to the results in [32] without any occurrence of instability.

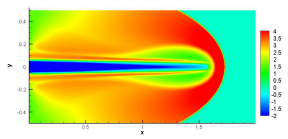
FIG. 3.14. *High Mach astrophysical jets in Example 13, Mach = 80, $k = 2$, $N_x \times N_y = 320 \times 160$. Scales are logarithmic.*



(a) Density contour.

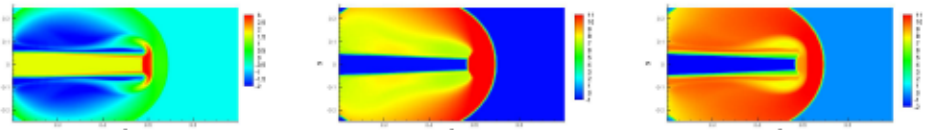


(b) Pressure contour.



(c) Temperature contour.

FIG. 3.15. *High Mach astrophysical jets in Example 13, Mach = 2000, $k = 2$, $N_x \times N_y = 320 \times 160$. Scales are logarithmic.*



(a) Density contour. (b) Pressure contour. (c) Temperature contour.

EXAMPLE 14. *This is another popular test problem for two-dimensional Euler equations. In [11, 32], the authors studied a Mach 5.09 shock diffracting at a 90° degree. Here we study a Mach 10 shock diffracting at a 120° degree [33, 5]. The computational domain and the triangular mesh with $h = 1/4$ are demonstrated in Figure 3.16. The initial condition and boundary conditions are the same as in [33, 5]. This example is very difficult to simulate for the reason that the negative density and pressure appear easily due to the numerical oscillations.*

In the shock diffraction problem in Example 14, we use the third order modified exponential Runge-Kutta method as the time discretization method. In Figure 3.16, we give the unstructured mesh with $h = 1/4$ in Example 14 for illustration purpose. In Figure 3.17, we show the density and pressure contours of shock diffraction problem at $t = 0.9$ with $k = 2$, $h = 1/20, 1/40$. Our results are comparable to those in [33, 5], which implies the proposed method has good performance without using any limiters.

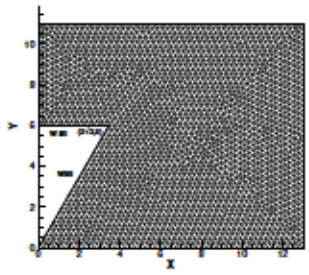
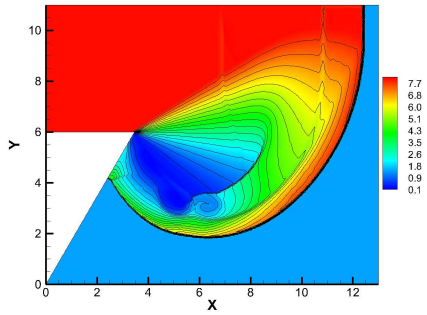


FIG. 3.16. *Shock diffraction in Example 14, illustration of the computational domain and the unstructured mesh with $h = 1/4$.*

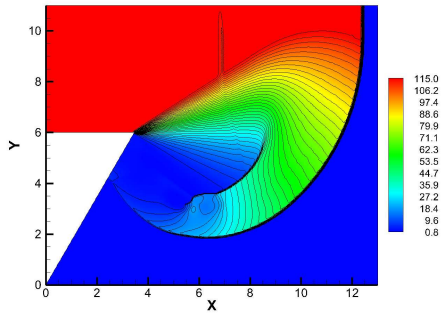
EXAMPLE 15. *Our final example is an inviscid, compressible Mach 3 flow moving towards a circular cylinder, which locates at the origin with radius 1 on the x-y plane, from the left. Our computational domain is $\Omega = [-3, 3] \times [-6, 6]$. At the left boundary $x = -3$, the inflow boundary condition is used and outflow boundary condition is imposed at the boundaries $x = 3$, $y = \pm 6$. We used the solid wall boundary condition on the surface of the cylinder, $(u, v) \cdot \boldsymbol{n} = 0$. The terminal time $T = 40$ makes the numerical solution to reach the steady state in the subregion $(-3, 0) \times (-6, 6)$.*

In Example 15, we also use the third order modified exponential Runge-Kutta method as the time discretization method instead of the explicit Runge-Kutta method. In Figure 3.18, we present the sketched triangular mesh of the computational domain with $h = 1/4$ in Example 15. In Figure 3.19, we show the density and pressure contours of steady state solutions for the problem of *flow past a cylinder*. We can observe the bow shock is well-captured by our method with the meshes $h = 1/20$ and $h = 1/40$.

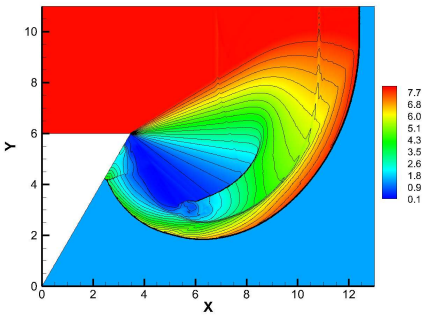
FIG. 3.17. *Shock diffraction in Example 14, Mach = 10, $k = 2$.*



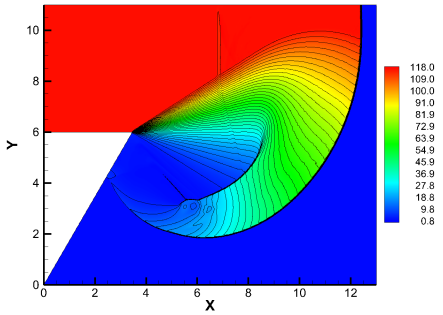
(a) Density contour: 20 equally spaced contour flood and lines from $\rho = 0.07$ to 8.1, $h = 1/20$.



(b) Pressure contour: 40 equally spaced contour flood and lines from $p = 0.8$ to 115, $h = 1/20$.



(c) Density contour: 20 equally spaced contour flood and lines from $\rho = 0.07$ to 8.1, $h = 1/40$.



(d) Pressure contour: 40 equally spaced contour flood and lines from $p = 0.8$ to 118, $h = 1/40$.

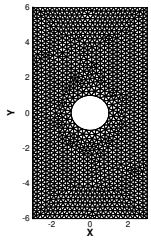
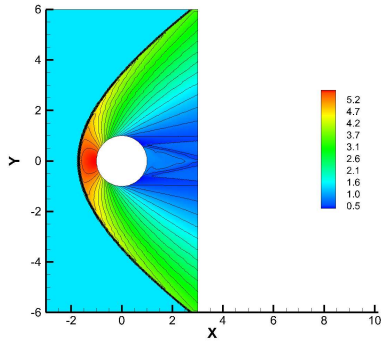


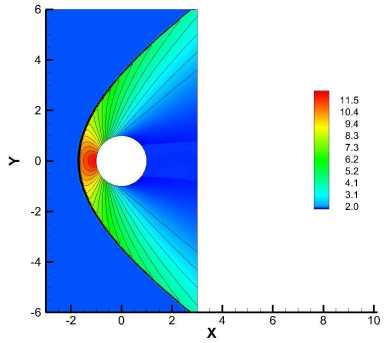
FIG. 3.18. *Flow past a cylinder in Example 15, illustration of the computational domain and the unstructured mesh with $h = 1/4$.*

4. Concluding remarks. In this paper, we extend our previous work, the oscillation-free discontinuous Galerkin (OFDG) method to solve hyperbolic systems. To construct the damping terms in the numerical scheme, we need to perform characteristic decomposition of the Jacobian of flux functions and the left eigenvector matrix is being carefully chosen. Coupled with the exponential Runge-Kutta method,

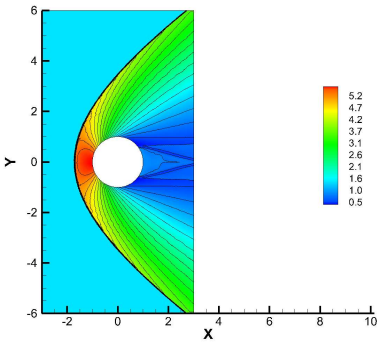
FIG. 3.19. *Flow past a cylinder in Example 15, Mach = 3, $k = 2$.*



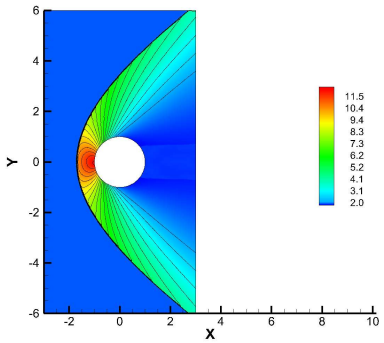
(a) Density contour: 20 equally spaced contour flood and lines from $\rho = 0.5$ to 5.5, $h = 1/20$.



(b) Pressure contour: 20 equally spaced contour flood and lines from $p = 2$ to 12, $h = 1/20$.



(c) Density contour: 20 equally spaced contour flood and lines from $\rho = 0.5$ to 5.5, $h = 1/40$.



(d) Pressure contour: 20 equally spaced contour flood and lines from $p = 2$ to 12, $h = 1/40$.

we show the CFL condition of the fully discrete scheme would not be polluted by the damping terms. A variety of numerical experiments demonstrate the new DG method can control spurious oscillations automatically without any additional post processes, even for some tough problems with very low density and pressure. Though the OFDG method does not guarantee the preservation of some physical structures like positivity preserving, one can easily apply the positive preserving limiter in [32] to obtain guaranteed positivity based on this framework. Currently, there is no theoretical result on the entropy stability of the OFDG method for hyperbolic systems, thus to obtain an entropy stable OFDG scheme will be one of our future works.

Appendix A. Some implementation details and technical proofs. In this appendix, we provide some implementation details and technical proofs of error estimates of the numerical schemes .

A.1. Implementation details: Quadrature rules and numerical flux. The DG formulation often involves some computation of volume integrals. The integrals are often complex thus the exact integration is impossible. The common

approach is to use the numerical quadrature with sufficient accuracy. Throughout this paper, we use the five point Gauss-Lobatto quadrature in 1D problems:

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{i=1}^{5} w_i f(x_i), \tag{A.1}$$

where $x_1 = -1, x_2 = -\sqrt{3/7}, x_3 = 0, x_4 = \sqrt{3/7}, x_5 = 1$, $w_1 = 1/10, w_2 = 49/90, w_3 = 32/45, w_4 = 49/90, w_5 = 1/10$. We use the tensor product of 1D Gauss-Lobatto quadrature on the 2D Cartesian grid. For the triangular mesh, we use the quadrature rule in Table A.5 on reference triangle element $K = \{(\xi, \eta) : 0 \le \xi,\, \eta,\, \xi + \eta \le 1\}$.

TABLE A.5
Six points quadrature rule with order 4.

ξ	η	w
0.44594849091597	0.44594849091597	0.22338158967801
0.44594849091597	0.10810301816807	0.22338158967801
0.10810301816807	0.44594849091597	0.22338158967801
0.09157621350977	0.09157621350977	0.10995174365532
0.09157621350977	0.81684757298046	0.10995174365532
0.81684757298046	0.09157621350977	0.10995174365532

$$\int_K f(\xi, \eta)\,d\xi\,d\eta \approx \sum_{i=1}^{6} \frac{1}{2} w_i f(\xi_i, \eta_i) \tag{A.2}$$

The numerical flux we use in this paper is the local Lax-Friedrichs flux. For illustration purpose, we only present the 1D numerical flux here. The local Lax-Friedrichs flux is defined as follows:

$$\hat{\boldsymbol{F}}_{j+\frac{1}{2}} = \frac{1}{2}\Big(\boldsymbol{F}\big((\boldsymbol{U}_h)^-_{j+\frac{1}{2}}\big) + \boldsymbol{F}\big((\boldsymbol{U}_h)^+_{j+\frac{1}{2}}\big) - \alpha_{j+\frac{1}{2}}\big((\boldsymbol{U}_h)^+_{j+\frac{1}{2}} - (\boldsymbol{U}_h)^-_{j+\frac{1}{2}}\big)\Big), \tag{A.3}$$

with $\alpha_{j+\frac{1}{2}} = \max_{1\le s\le m}\{\big|(\lambda_s)^-_{j+\frac{1}{2}}\big|, \big|(\lambda_s)^+_{j+\frac{1}{2}}\big|\}$, where $(\lambda_s)^{\pm}_{j+\frac{1}{2}}$, $s = 1, \ldots, m$ are the m real eigenvalues of the Jacobian $\boldsymbol{F}'(\boldsymbol{U})$ at $x^{\pm}_{j+\frac{1}{2}}$.

A.2. Error estimates for one-dimensional symmetrizable systems. We assume that the first-order conservation laws (2.1) is symmetrizable system, i.e. there exists a mapping $\boldsymbol{U}(\boldsymbol{V})$: $\mathbb{R}^m \to \mathbb{R}^m$ such that (2.1) is transformed into the following form:

$$\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{V}}\boldsymbol{V}_t + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{V}}\boldsymbol{V}_x = 0, \tag{A.4}$$

the matrix $\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{V}}$ is symmetric positive definite (SPD) and the matrix $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{V}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}}\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{V}}$ is also symmetric. We further assume that each component of $\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{V}}$ is Lipschitz continuous with respect to the variable $\boldsymbol{V}$. Following the standard analysis in finite element error estimates, we define $\boldsymbol{\xi} = \Pi_h \boldsymbol{U} - \boldsymbol{U}_h$ and $\boldsymbol{\eta} = \Pi_h \boldsymbol{U} - \boldsymbol{U}$, where Π_h is the local L^2-projection. The error is decomposed by $\boldsymbol{e} = \boldsymbol{U} - \boldsymbol{U}_h = \boldsymbol{\xi} - \boldsymbol{\eta}$. From (2.5) and

consistent property of scheme, we obtain error equations as follows.

(A.5)
$$\int_{I_j} \boldsymbol{\xi}_t \cdot \boldsymbol{v}_h \, dx + \sum_{l=0}^{k} \frac{\sigma_j^l(\boldsymbol{U}_h)}{h_j} \int_{I_j} \big(\boldsymbol{\xi} - P_h^{l-1}\boldsymbol{\xi}\big) \cdot \boldsymbol{v}_h \, dx$$
$$= \int_{I_j} \boldsymbol{\eta}_t \cdot \boldsymbol{v}_h \, dx + \mathcal{K}_j(\boldsymbol{U}, \boldsymbol{U}_h; \boldsymbol{v}_h) + \sum_{l=0}^{k} \frac{\sigma_j^l(\boldsymbol{U}_h)}{h_j} \int_{I_j} \big(\Pi_h \boldsymbol{U} - P_h^{l-1}\Pi_h \boldsymbol{U}\big) \cdot \boldsymbol{v}_h \, dx$$

for any $\boldsymbol{v}_h \in [V_h^k]^m$ and $1 \le j \le N$, and

(A.6)
$$\mathcal{K}_j(\boldsymbol{U}, \boldsymbol{U}_h; \boldsymbol{v}_h) = \mathcal{H}_j(\boldsymbol{U}, \boldsymbol{v}_h) - \mathcal{H}_j(\boldsymbol{U}_h, \boldsymbol{v}_h),$$
$$\mathcal{H}_j(\boldsymbol{U}_h, \boldsymbol{v}_h) = \int_{I_j} \boldsymbol{F}(\boldsymbol{U}_h) \cdot (\boldsymbol{v}_h)_x \, dx - \widehat{\boldsymbol{F}}_{j+\frac{1}{2}} \cdot (\boldsymbol{v}_h)^-_{j+\frac{1}{2}} + \widehat{\boldsymbol{F}}_{j-\frac{1}{2}} \cdot (\boldsymbol{v}_h)^+_{j-\frac{1}{2}}.$$

We take the test function $\boldsymbol{v}_h = \frac{\partial \boldsymbol{V}}{\partial \boldsymbol{U}}(\boldsymbol{U}_c)\boldsymbol{\xi}$ in (A.5), where $\boldsymbol{U}_c$ is the piecewise constant vector-valued function that is equal to the vector $\boldsymbol{U}(x_j, t)$ in each element I_j. We denote the weighted norm

(A.7)
$$\|\boldsymbol{p}\|_V = \Big\| \Big(\frac{\partial \boldsymbol{V}}{\partial \boldsymbol{U}}\Big)^{\frac{1}{2}} (\boldsymbol{U}_c)\boldsymbol{p} \Big\|,$$

for any vector-valued function $\boldsymbol{p}$. By the L^2-projection, the first term of right hand side of (A.5) vanishes. Thus summing over j, we obtain

(A.8)
$$\int_a^b \boldsymbol{\xi}^T \frac{\partial \boldsymbol{V}}{\partial \boldsymbol{U}}(\boldsymbol{U}_c)\boldsymbol{\xi}_t \, dx + \sum_{j=1}^{N}\sum_{l=0}^{k} \frac{\sigma_j^l(\boldsymbol{U}_h)}{h_j} \int_{I_j} \big(\boldsymbol{\xi} - P_h^{l-1}\boldsymbol{\xi}\big)^T \frac{\partial \boldsymbol{V}}{\partial \boldsymbol{U}}(\boldsymbol{U}_c)\big(\boldsymbol{\xi} - P_h^{l-1}\boldsymbol{\xi}\big) \, dx$$
$$= \sum_{j=1}^{N} \mathcal{K}_j\Big(\boldsymbol{U}, \boldsymbol{U}_h; \frac{\partial \boldsymbol{V}}{\partial \boldsymbol{U}}(\boldsymbol{U}_c)\boldsymbol{\xi}\Big) + \sum_{j=1}^{N}\sum_{l=0}^{k} \frac{\sigma_j^l(\boldsymbol{U}_h)}{h_j} \int_{I_j} \boldsymbol{\xi}^T \frac{\partial \boldsymbol{V}}{\partial \boldsymbol{U}}(\boldsymbol{U}_c)\big(\Pi_h \boldsymbol{U} - P_h^{l-1}\Pi_h \boldsymbol{U}\big) \, dx$$
$$= A_1 + A_2 \,.$$

By lemma 4.2 in [30], we have the estimate of A_1:

(A.9)
$$A_1 \le (C + C_\star \|\boldsymbol{e}\|_\infty^2) h^{2k+1} + (C + C_\star h^{-1}\|\boldsymbol{e}\|_\infty^2)\|\boldsymbol{\xi}\|_V^2.$$

Next, we proceed to estimate A_2. Since we have

$$\begin{aligned}
&[\![(\mathbf{R}^{-1}\partial_x^l(\boldsymbol{U}_h - \boldsymbol{U}))_s]\!]^2_{j-\frac{1}{2}} + [\![(\mathbf{R}^{-1}\partial_x^l(\boldsymbol{U}_h - \boldsymbol{U}))_s]\!]^2_{j+\frac{1}{2}} \\
&\quad \lesssim [\![(\partial_x^l \boldsymbol{\xi})_s]\!]^2_{j-\frac{1}{2}} + [\![(\partial_x^l \boldsymbol{\xi})_s]\!]^2_{j+\frac{1}{2}} + [\![(\partial_x^l \boldsymbol{\eta})_s]\!]^2_{j-\frac{1}{2}} + [\![(\partial_x^l \boldsymbol{\eta})_s]\!]^2_{j+\frac{1}{2}} \\
&\quad \lesssim h^{-2l-1}\|\xi_s\|^2_{L^2(I_{j-1}\cup I_j \cup I_{j+1})} + h^{2k-2l+1}\|U_s\|^2_{H^{k+1}(I_{j-1}\cup I_j \cup I_{j+1})} \,.
\end{aligned}$$

Then we have the estimation of damping coefficients in the following.

(A.10)
$$|\sigma_j^l| \lesssim h^{-\frac{1}{2}}\|\boldsymbol{\xi}\|_{L^2(I_{j-1}\cup I_j \cup I_{j+1})} + h^{k+\frac{1}{2}}\|\mathbf{U}\|_{H^{k+1}(I_{j-1}\cup I_j \cup I_{j+1})}.$$

We also have

$$\begin{aligned}
&\|\Pi_h \boldsymbol{U} - P_h^{l-1}\Pi_h \boldsymbol{U}\|_{L^2(I_j)} \\
&\quad \le \|\Pi_h \boldsymbol{U} - \boldsymbol{U}\|_{L^2(I_j)} + \|\boldsymbol{U} - P_h^{l-1}\boldsymbol{U}\|_{L^2(I_j)} + \|P_h^{l-1}\boldsymbol{U} - P_h^{l-1}\Pi_h \boldsymbol{U}\|_{L^2(I_j)} \\
&\quad \le Ch^{k+1}\|\boldsymbol{U}\|_{H^{k+1}(I_j)} + Ch^{\min(1,l)+\frac{1}{2}}\|\boldsymbol{U}\|_{W^{l,\infty}(I_j)} + Ch^{k+1}\|\boldsymbol{U}\|_{H^{k+1}(I_j)} \\
&\quad \le C(h^{k+1} + h^{\min(1,l)+\frac{1}{2}})\|\boldsymbol{U}\|_{H^{k+1}(\Omega)}.
\end{aligned}$$

Here we used the Sobolev inequality

$$\|\boldsymbol{U}\|_{W^{l,\infty}(\Omega)} \leq C\|\boldsymbol{U}\|_{H^{k+1}(\Omega)}, \quad \forall\, 0 \leq l \leq k. \tag{A.11}$$

Therefore, combine (A.10) and (A.11) and apply Cauchy-Schwarz inequality, we have

$$A_2 \leq C(\|\boldsymbol{\xi}\|\|\boldsymbol{\xi}\|_V + h^{k+1}\|\boldsymbol{\xi}\|_V)\|\boldsymbol{U}\|_{H^{k+1}(\Omega)}. \tag{A.12}$$

Plugging (A.9) and (A.12) into (A.8) we have

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}\|\boldsymbol{\xi}\|_V^2 \leq & \frac{1}{2}\int_a^b \boldsymbol{\xi}^T\Big(\frac{\partial \boldsymbol{V}}{\partial \boldsymbol{U}}\Big)_t(\boldsymbol{U}_c)\boldsymbol{\xi}\,dx + (C + C_\star h^{-1}\|\boldsymbol{e}\|_\infty^2)\|\boldsymbol{\xi}\|_V^2 \\ & + (C + C_\star\|\boldsymbol{e}\|_\infty^2)h^{2k+1} + C(\|\boldsymbol{\xi}\|\|\boldsymbol{\xi}\|_V + h^{k+1}\|\boldsymbol{\xi}\|_V)\|\boldsymbol{U}\|_{H^{k+1}(\Omega)}.\end{aligned} \tag{A.13}$$

Under the a priori assumption of error, for h small enough there holds

$$\|\boldsymbol{U} - \boldsymbol{U}_h\|_\infty \leq Ch^{\frac{1}{2}}. \tag{A.14}$$

This a priori assumption, which can be justified (see [19]), is unnecessary for linear flux function, i.e $\boldsymbol{F}(\boldsymbol{U}) = \mathbb{C}\boldsymbol{U}$, where $\mathbb{C}$ is a constant matrix. And in the symmetrization theory [22], the $\|\cdot\|_V$ norm is equivalent to the usual L^2-norm $\|\cdot\|$. Therefore, by (A.13) we obtain

$$\frac{d}{dt}\|\boldsymbol{\xi}\|_V^2 \leq C(h^{2k+1} + \|\boldsymbol{\xi}\|_V^2). \tag{A.15}$$

Finally, the Gronwall's inequality, triangle inequality and approximation of projection imply that

$$\|\boldsymbol{U} - \boldsymbol{U}_h\| \leq Ch^{k+\frac{1}{2}}. \tag{A.16}$$

Remark A.1. *Since we use the local Lax-Friedrichs flux, we only obtain the sub-optimal error estimates here because of A_1, not A_2. For certain special numerical flux functions, we can also upgrade the error estimates to be optimal, i.e., $\mathcal{O}(h^{k+1})$. Thus, the added damping term does not reduce any accuracy of the original scheme when the exact solution is sufficiently smooth.*

REFERENCES

[1] R. Becker and M. Braack, *A two-level stabilization scheme for the Navier-Stokes equations*, Numerical Mathematics and Advanced Applications, 123 – 130, Springer, Berlin, 2004.

[2] M. Braack and E. Burman, *Local projection stabilization for the Oseen problem and its interpretation as a variational multiscale method*, SIAM J. Numer. Anal. 43 (2006), 2544 – 2566.

[3] F. Bassi and S. Rebay, *Accurate 2D Euler computations by means of a high order discontinuous finite element method*, 14th ICNMFD, Lect. Notes in Phys. 453 (1995), 234 – 240.

[4] J. Casper, *Finite-volume implementation of high-order essentially nonoscillatory schemes in two dimensions*, AIAA Journal 30 (1992), 2829 – 2835.

[5] T. Chen and C.-W. Shu, *Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation laws*, J. Comput. Phys. 345 (2017), 427–461.

[6] B. Cockburn, S. Hou and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case*, Math. Comp. 54 (1990), 545 – 581.

[7] B. Cockburn, G.E. Karniadakis and C.-W. Shu, *The development of discontinuous Galerkin methods. Discontinuous Galerkin methods*, 3 – 50, Lect. Notes Comput. Sci. Eng. 11, Springer, Berlin, 2000.

[8] B. Cockburn, S.-Y. Lin and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems*, J. Comput. Phys. 84 (1989), 90 – 113.

[9] B. Cockburn and C.-W. Shu, *The Runge-Kutta local projection P^1-discontinuous-Galerkin finite element method for scalar conservation laws*, ESAIM Math. Model. Numer. Anal. 25 (1991), 337 – 361.

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

[11] B. Cockburn and C.-W. Shu, *The Runge-Kutta discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems*, J. Comput. Phys. 141 (1998), 199 – 224.

[12] B. Cockburn and C.-W. Shu, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal. 35 (1998), 2440 – 2463.

[13] R. Hartmann, *Adaptive discontinuous Galerkin methods with shock capturing for the compressible Navier Stokes equations*, Internat. J. Numer. Methods Fluids 51 (2006), 1131 – 1156.

[14] Y. Ha, C. Gardner, A. Gelb and C.-W. Shu, *Numerical simulation of high Mach number astrophysical jets with radiative cooling*, J. Sci. Comput. 24 (2005), 597 – 612.

[15] A. Hiltebrand and S. Mishra, *Entropy stable shock capturing space time discontinuous Galerkin schemes for systems of conservation laws*, Numer. Math. 126 (2014), 103 – 151.

[16] J. Huang and C.-W. Shu, *Bound-preserving modified exponential Runge-Kutta discontinuous Galerkin methods for scalar hyperbolic equations with stiff source terms*, J. Comput. Phys. 361 (2018), 111–135.

[17] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys. 126 (1996), 202 – 228.

[18] P.D. Lax, *Weak solutions of nonlinear hyperbolic equations and their numerical computation*, Comm. Pure Appl. Math. 7 (1954), 159 – 193.

[19] J. Lu, Y. Liu and C.-W. Shu, *An oscillation-free discontinuous Galerkin method for scalar hyperbolic conservation laws*, SIAM J. Numer. Anal. 59 (2021), 1299–1324.

[20] W.H. Reed and T.R. Hill, *Triangular mesh methods for the neutron transport equation*, Los Alamos Scientific Laboratory report LA-UR-73-479, NM, 1973.

[21] L.I. Sedov, *Similarity and Dimensional Methods in Mechanics*, Academic Press, New York, 1959.

[22] C. W. Schulz-Rinne, *Classification of the Riemann problem for two-dimensional gas dynamics*, SIAM J. Math. Anal. 24 (1993), 76 – 88.

[23] C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, Advanced numerical approximation of nonlinear hyperbolic equations (Cetraro, 1997), 325 – 432, Lecture Notes in Math., 1697, Fond. CIME/CIME Found. Subser., Springer, Berlin, 1998.

[24] C.-W. Shu, *Discontinuous Galerkin methods: general approach and stability*, Numerical solutions of partial differential equations, 149 – 201, Adv. Courses Math. CRM Barcelona, Birkhäuser, Basel, 2009.

[25] C.-W. Shu, *Discontinuous Galerkin methods for time-dependent convection dominated problems: basics, recent developments and comparison with other methods. Building bridges: connections and challenges in modern approaches to numerical partial differential equations*, 369 – 397, Lect. Notes Comput. Sci. Eng. 114, Springer, 2016.

[26] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys. 77 (1988), 439 – 471.

[27] C.-W. Shu and S. Osher, *Efficient implementation of essentially nonoscillatory shock-capturing schemes. II*, J. Comput. Phys. 83 (1989), 32 – 78.

[28] G.A. Sod, *A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws*, J. Comput. Phys. 27 (1978), 1 – 31.

[29] P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, J. Comput. Phys. 54 (1984), 115 – 173.

[30] Q. Zhang and C.-W. Shu, *Error estimates to smooth solutions of Runge-Kutta discontinuous Galerkin method for symmetrizable systems of conservation laws*, SIAM J. Numer. Anal. , 44 (2006), 1703–1720.

[31] X. Zhang and C.-W. Shu, *On maximum-principle-satisfying high order schemes for scalar conservation laws*, J. Comput. Phys. 229 (2010), 3091 – 3120.

[32] X. Zhang and C.-W. Shu, *On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes*, J. Comput. Phys. 229 (2010),

8918 – 8934.
[33] X. Zhang, Y. Xia and C.-W. Shu, *Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes*, J. Sci. Comput. 50 (2012), 29 – 62.
[34] X. Zhong and C.-W. Shu, *A simple weighted essentially nonoscillatory limiter for Runge-Kutta discontinuous Galerkin methods*, J. Comput. Phys. 232 (2013), 397 – 415.
[35] J. Zhu and C.-W. Shu, *Numerical study on the convergence to steady state solutions of a new class of high order WENO schemes*, J. Comput. Phys. 349 (2017), 80 – 96.