

AN ENTROPY STABLE ESSENTIALLY OSCILLATION-FREE DISCONTINUOUS GALERKIN METHOD FOR HYPERBOLIC CONSERVATION LAWS ^{*}

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

Abstract. Entropy inequalities are crucial to the well-posedness of hyperbolic conservation laws, which help to select the physically meaningful one among the infinite many weak solutions. Recently, several high order discontinuous Galerkin (DG) methods satisfying entropy inequalities were proposed, see [9, 7, 10] and the references therein. However, high order numerical methods typically generate spurious oscillations in the presence of shock discontinuities. In this paper, we construct a high order entropy stable essentially oscillation-free DG (ESOFDG) method for hyperbolic conservation laws. With some suitable modification on the high order damping term introduced in [36, 35], we are able to construct an OFDG scheme with dissipative entropy. It is challenging to make the damping term compatible with the current entropy stable DG framework, that is, the damping term should be dissipative for any given entropy function without compromising high order accuracy. The key ingredient is to utilize the convexity of the entropy function and the orthogonality of the projection. Then the proposed method maintains the same properties of conservation, error estimates and entropy dissipation as the original entropy stable DG method. Extensive numerical experiments are presented to validate the theoretical findings and the capability of controlling spurious oscillations of the proposed algorithm.

Key words. Hyperbolic conservation laws; Entropy stability; Summation-by-parts; Discontinuous Galerkin method; Non-oscillatory

MSC codes. 65M12, 65M60, 76M20

1. Introduction. Hyperbolic conservation laws have been studied over the centuries in the realm of gas dynamics of continuum physics. The general form of systems of conservation laws is

$$(1.1) \quad \begin{cases} \frac{\partial \mathbf{u}}{\partial t} + \sum_{m=1}^d \frac{\partial \mathbf{f}_m(\mathbf{u})}{\partial x_m} = \mathbf{0}, & (\mathbf{x}, t) \in \mathbb{R}^d \times (0, +\infty), \\ \mathbf{u} = \mathbf{g}, & \text{on } \mathbb{R}^d \times \{t = 0\} \end{cases}$$

where $\mathbf{u} = [u_1, \dots, u_n]^T$ is a vector of functions denoting the conservative variables, $\mathbf{f}_m = [f_m^1, \dots, f_m^n]^T$ is the vector flux function, and $\mathbf{g} : \mathbb{R}^d \rightarrow \mathbb{R}^n$ is the given initial condition. It is widely known that shock waves or contact discontinuities might be developed at finite time, regardless of the smoothness of the initial or boundary conditions. Therefore it is reasonable to seek for weak solutions and interpret (1.1) in the sense of distribution.

Definition 1.1. A vector-valued function $\mathbf{u} \in L^\infty(\mathbb{R}^d \times (0, +\infty); \mathbb{R}^n)$ is called a

^{*}Submitted to the editors DATE.

Funding: Y. Liu's Research is partially supported by NSFC grant 12201621 and the Youth Innovation Promotion Association CAS. J. Lu's Research is partially supported by NSFC grant 11901213 and Science and Technology Program of Guangzhou 2023A04J1300. C.-W. Shu's Research is partially supported by NSF grant DMS-2010107 and AFOSR grant FA9550-20-1-0055.

[†]LSEC, Institute of Computational Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, P.R. China. (yongliu@lsec.cc.ac.cn).

[‡]School of Mathematics, South China University of Technology, Canton, Guangdong 510641, China. (jflu@scut.edu.cn).

[§]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. (chi-wang_shu@brown.edu)

weak solution of (1.1) if it satisfies (1.1) in the sense of distributions:

$$(1.2) \quad \int_0^{+\infty} \int_{\mathbb{R}^d} \left(\mathbf{u} \cdot \frac{\partial \varphi}{\partial t} + \sum_{m=1}^d \mathbf{f}_m(\mathbf{u}) \cdot \frac{\partial \varphi}{\partial x_m} \right) d\mathbf{x} dt + \int_{\mathbb{R}^d} \varphi \cdot \mathbf{g} d\mathbf{x} \Big|_{t=0} = 0$$

for all smooth test functions $\varphi : \mathbb{R}^d \times [0, +\infty) \rightarrow \mathbb{R}^n$ with compact support.

Unfortunately, the weak solution in Definition 1.1 turns out to be inadequate, as such weak solution would not be unique in general. In order to select the “physically relevant” solution among all weak solutions, we require the weak solution to satisfy certain entropy criterion. To this end, we first introduce the entropy pairs as follows.

Definition 1.2. A pair of functions $[U(\mathbf{u}), \mathbf{F}(\mathbf{u})]$ with $U : \mathbb{R}^n \rightarrow \mathbb{R}$, $\mathbf{F} = [F_1, \dots, F_d]^T : \mathbb{R}^n \rightarrow \mathbb{R}^d$ is called an entropy pair for (1.1) if $U(\mathbf{u})$ is convex and $\{F_m(\mathbf{u})\}_{m=1}^d$ satisfy

$$(1.3) \quad F'_m(\mathbf{u}) = U'(\mathbf{u}) \mathbf{f}'_m(\mathbf{u}), \quad m = 1, \dots, d,$$

where $U'(\mathbf{u})$ and $F'_m(\mathbf{u})$ are row vectors and $\mathbf{f}'_m(\mathbf{u})$ is the $n \times n$ Jacobian matrix.

With the setup of entropy pairs, we now specify the additional admissibility condition in order to select the physically meaningful weak solution.

Definition 1.3. A weak solution $\mathbf{u}$ of (1.1) is an entropy solution if the following inequality holds:

$$(1.4) \quad \frac{\partial U(\mathbf{u})}{\partial t} + \sum_{m=1}^d \frac{\partial F_m(\mathbf{u})}{\partial x_m} \leq 0$$

for all entropy pairs $[U(\mathbf{u}), \mathbf{F}(\mathbf{u})]$ in the weak sense, that is,

$$(1.5) \quad \int_0^{+\infty} \int_{\mathbb{R}^d} \left(U(\mathbf{u}) \frac{\partial \phi}{\partial t} + \sum_{m=1}^d F_m(\mathbf{u}) \frac{\partial \phi}{\partial x_m} \right) d\mathbf{x} dt + \int_{\mathbb{R}^d} U(\mathbf{g}) \phi(\mathbf{x}, 0) d\mathbf{x} \geq 0$$

for any $\phi \in C_c^\infty(\mathbb{R}^d \times [0, +\infty); \mathbb{R}^n)$, $\phi \geq 0$.

The existence and uniqueness of the entropy solution of (1.1) can be established for scalar conservation laws ($n = 1$), and for one-dimensional systems ($d = 1$) with small initial variation. However, the global existence and uniqueness of the entropy solutions for general hyperbolic conservation laws remain open and a good mathematical understanding of (1.1) is largely unavailable at present. For more details on the theory of hyperbolic conservation laws, we refer the readers to [23, 15] and the references cited therein.

Despite that uniqueness might not be guaranteed under the entropy conditions given in Definition 1.3, in the numerical approximation of (1.1) one would still like to seek numerical schemes which satisfy the entropy condition on the discrete level. Such a property is referred as *entropy stability*. For the first order (finite volume) method, entropy stability analysis is well-developed based on Tadmor’s entropy conservative fluxes and entropy stable fluxes [49, 50]. For the high order entropy stable finite volume methods, a notable result is the *TeCNO* scheme proposed by Fjordholm, Mishra and Tadmor [19], with the use of high order entropy conservative fluxes [37] and the sign property of the essentially non-oscillatory (ENO) reconstruction [20]. In recent years, there have been rapid developments on the entropy stable quadrature-based discontinuous Galerkin (DG) methods. In [9], Chen and Shu proposed an

entropy stable DG scheme on unstructured simplex meshes, in which they introduced special Gauss-Lobatto type quadrature rules with collocated surface quadrature points and discrete operators with the multidimensional summation-by-parts (SBP) property [27, 18]. There also exist other different types of entropy stable DG methods in the SBP framework, see e.g. [13, 14, 7, 8]. A comprehensive review of entropy stable DG methods for systems of conservation laws can be found in [10]. Entropy stable DG methods are also applied to solve other types of equations, such as convection-diffusion equations [4, 3, 22], MHD equations [2], shallow water equations [21, 40, 51], gradient flow problems [46], two-phase flow problems [41] and stochastic problems [39], etc.

For hyperbolic conservation laws, high order linear numerical schemes often generate spurious oscillations near the discontinuities (the Gibbs phenomenon), which may lead to accuracy contamination in smooth regions and less robustness of the schemes, and even possible blowups of the code. The entropy stable quadrature-based DG methods are not exempt from this deficiency. Generally, there are two kinds of treatments to overcome such a difficulty. One is to apply slope limiters on the DG solutions, such as the total variation diminishing (TVD) limiters, total variation bounded (TVB) limiters [11], and weighted ENO (WENO) limiters [42, 55], etc. The limiters work quite effectively for their simplicity, low computational cost and with little modification required on the original codes, thus they are favored by many researchers and engineers. Another treatment is to add artificial diffusion in the weak formulations, which is more convenient to perform theoretical analysis. However, it needs a subtle analysis to determine how much diffusion needs to be added, see e.g. [29]. With suitable artificial diffusion, the smoothing effect of the diffusion is obvious in suppressing the spurious oscillations.

The objective of this paper is to design a DG scheme with both entropy stability and oscillation-free properties. It seems quite challenging to obtain these two properties simultaneously. In fact, [9, Remark 4.4] states that it is hard to design entropy stable TVD/TVB limiters for hyperbolic systems. Very recently, we developed an approach to control the spurious oscillations by introducing damping in the DG formulations artificially [36, 35]. As demonstrated in [36], the proposed DG methods can not only control the spurious oscillations, but also preserve some basic properties of the standard DG methods such as conservation, optimal a priori error estimates and superconvergence, etc. In this paper, we extend the damping technique to the entropy stable quadrature-based DG methods. The newly added damping is independent of the given entropy pair and has entropy dissipation without comprising high order accuracy in the smooth region. Then entropy stability is obtained by using the convexity of the entropy functions and taking only one part of the local projections in [36]. That is, we only take the projection orthogonal to the constant states so as to achieve the entropy stability for any given entropy pair. Compared with the damping terms in [36], the one proposed here abandons the original hierarchical structure and the damping coefficients are also adjusted. By a careful theoretical analysis, the proposed algorithm still maintains the properties of the standard entropy stable quadrature-based DG methods. Finally, we take the investigations on a variety of numerical examples to show the good performance of the constructed schemes.

The rest of the paper is organized as follows. In Section 2, we briefly present some preliminary results, including continuous entropy analysis, quadrature rules on simplex elements and the corresponding SBP operators. In Section 3, we first review the matrix-vector form of the nodal DG method, and the entropy stable DG method with quadrature rules of collocated surface nodes, and then we propose the entropy stable **essentially** oscillation-free DG (ESOFDG) method. Theoretical analysis of accuracy,

conservation and entropy stability are also given. In Section 4, we present extensive numerical experiments such as accuracy tests, convex and nonconvex conservation laws, and several benchmark problems associated with compressible Euler equations. Concluding remarks are given in Section 5.

2. Preliminaries. In this section, we will firstly present the derivation of the entropy inequality for (1.1) on the PDE level. Then we introduce the quadrature rules and the summation-by-parts (SBP) operators [16, 17, 47], which mimic integration by parts at the discrete level. Next, we give a brief description of the nodal DG method. As we shall see later, the nodal DG method in the matrix-vector form is well suited in the SBP framework.

2.1. Continuous entropy analysis. As mentioned before, the functions $[U(\mathbf{u}), \mathbf{F}(\mathbf{u})]$, where $U(\mathbf{u})$ is convex and (1.3) is satisfied, is called an entropy pair. Given a strictly convex entropy function $U(\mathbf{u})$, let $\mathbf{v} = U'(\mathbf{u})^T$ be the entropy variables. Then $\mathbf{v}'(\mathbf{u}) = U''(\mathbf{u})$ is symmetric positive-definite, and the mapping $\mathbf{u} \rightarrow \mathbf{v}$ is invertible. Now let us define the potential fluxes in the following:

$$(2.1) \quad \psi_m(\mathbf{v}) = \mathbf{v}^T \mathbf{f}_m(\mathbf{u}(\mathbf{v})) - F_m(\mathbf{u}(\mathbf{v})), \quad m = 1, \dots, d.$$

One can easily verify that

$$(2.2) \quad \psi'_m(\mathbf{v}) = \mathbf{f}_m(\mathbf{u}(\mathbf{v}))^T.$$

The potential fluxes are used for the construction of the entropy conservative fluxes, see e.g. [30, 6, 13, 14]. If the solutions to conservation laws (1.1) are smooth, then they should satisfy an additional entropy conservation law as below

$$(2.3) \quad 0 = U'(\mathbf{u}) \frac{\partial \mathbf{u}}{\partial t} + \sum_{m=1}^d U'(\mathbf{u}) \mathbf{f}'_m(\mathbf{u}) \frac{\partial \mathbf{u}}{\partial x_m} = \frac{\partial U}{\partial t} + \frac{\partial F_m}{\partial x_m}.$$

When the solutions have discontinuities, it is natural to require the entropy to be dissipative. This is how the definition of entropy condition in (1.4) comes from. Integrating (1.4) in space and assuming $\mathbf{u}$ is compactly supported, we obtain the following inequality:

$$(2.4) \quad \frac{d}{dt} \int_{\mathbb{R}^d} U(\mathbf{u}) d\mathbf{x} \leq 0.$$

This means that the total amount of entropy is non-increasing with respect to time. But the existence of entropy function is not that trivial to obtain. For scalar conservation laws ($n = 1$), any convex function $U(u)$ can be taken as an entropy function, with the entropy fluxes $F_m(u) = \int U'(u) f'_m(u) du$. However, for general systems, the existence of entropy function is no longer guaranteed, and both existence and uniqueness of entropy solutions are much more challenging. Fortunately, for most systems we are interested in, such as shallow water equations, compressible Euler equations, and magnetohydrodynamic (MHD) equations, we are able to find the entropy functions with physical meaning. For more details readers can refer to [23] on the entropy analysis for systems of conservation laws.

2.2. Quadrature rules and SBP operators. In this subsection, we will briefly review the quadrature rules and SBP operators in order to rewrite the DG method under the SBP framework. First let us take the partition of the domain. Suppose

$\Omega \subset \mathbb{R}^d$ is some polygonal computational domain, and $\mathcal{T}_h = \{T_\kappa\}_{\kappa=1}^{N_h}$ is some conforming partition of Ω , and the simplex meshes $\mathcal{T}_h$ are shape regular and quasi-uniform. Assume each element T_κ is a simplex and its boundary ∂T_κ consists of $(d-1)$ -dimensional simplex faces. The set of simplex faces is denoted by

$$(2.5) \quad \Gamma_h = \{e : e = \partial T_\kappa \cap \partial T_\iota, 1 \leq \kappa, \iota \leq N_h, \kappa \neq \iota\}.$$

Let $h = \max_{T_\kappa \in \mathcal{T}_h} h_{T_\kappa}$ be the characteristic length of $\mathcal{T}_h$, where h_{T_κ} denotes the diameter of T_κ .

In order to obtain the SBP operators, the volume and surface quadrature rules should be carefully constructed. In this paper, we consider the diagonal-norm, multi-dimensional SBP operators which are defined in [27], and the SBP operators of degree k require the degree of the quadrature rules for the volume integral is at least $2k-1$ and the degree of the quadrature rules for surface integral is at least $2k$, please see [27, Theorem 3.2 and Theorem 3.4]. For multidimensional SBP operators by using a multiblock tensor-product formulation, we refer the readers to [28, 38, 48]. For each simplex T_κ , $1 \leq \kappa \leq N_h$, suppose that there is a quadrature rule of degree at least $2k-1$ on T_κ , associated with $\mathcal{N}_{Q,k}$ nodes $\{\mathbf{x}_j^\kappa\}_{j=1}^{\mathcal{N}_{Q,k}}$ and positive weights $\{\omega_j^\kappa\}_{j=1}^{\mathcal{N}_{Q,k}}$, $\mathcal{N}_{Q,k} \geq \mathcal{N}_{P,k}$ with

$$(2.6) \quad \mathcal{N}_{P,\nu} = \dim \mathcal{P}^\nu(\mathbb{R}^d) = \binom{\nu+d}{d}, \quad 0 \leq \nu \leq k.$$

For each $e \in \Gamma_h$, we also choose a (surface) quadrature rule of degree at least $2k$ on e , associated with $\mathcal{N}_{B,k}$ nodes $\{\mathbf{x}_s^e\}_{s=1}^{\mathcal{N}_{B,k}}$ and positive weights $\{\tau_s^e\}_{s=1}^{\mathcal{N}_{B,k}}$. Now we take $\{\varphi_\ell(\mathbf{x})\}_{\ell=1}^{\mathcal{N}_{P,\nu}}$ as the set of basis functions of $\mathcal{P}^\nu(\mathbb{R}^d)$, which means

$$\{\varphi_\ell(\mathbf{x})\}_{\ell=1}^{\mathcal{N}_{P,0}} \subset \{\varphi_\ell(\mathbf{x})\}_{\ell=1}^{\mathcal{N}_{P,1}} \subset \cdots \subset \{\varphi_\ell(\mathbf{x})\}_{\ell=1}^{\mathcal{N}_{P,k}}.$$

Then we define the Vandermonde matrices, whose columns are nodal values of $\{\varphi_\ell(\mathbf{x})\}_{\ell=1}^{\mathcal{N}_{P,\nu}}$:

$$(2.7) \quad V_\nu^\kappa = [\overrightarrow{\varphi_1^\kappa}, \dots, \overrightarrow{\varphi_{\mathcal{N}_{P,\nu}}^\kappa}], \quad V_\nu^e = [\overrightarrow{\varphi_1^e}, \dots, \overrightarrow{\varphi_{\mathcal{N}_{P,\nu}}^e}], \quad 0 \leq \nu \leq k,$$

where $\overrightarrow{\varphi_\ell^\kappa} = [\varphi_\ell(\mathbf{x}_1^\kappa), \varphi_\ell(\mathbf{x}_2^\kappa), \dots, \varphi_\ell(\mathbf{x}_{\mathcal{N}_{Q,k}}^\kappa)]^T$ and $\overrightarrow{\varphi_\ell^e} = [\varphi_\ell(\mathbf{x}_1^e), \varphi_\ell(\mathbf{x}_2^e), \dots, \varphi_\ell(\mathbf{x}_{\mathcal{N}_{B,k}}^e)]^T$, $1 \leq \ell \leq \mathcal{N}_{P,\nu}$. Therefore, V_ν^κ is a $\mathcal{N}_{Q,k} \times \mathcal{N}_{P,\nu}$ matrix and V_ν^e is a $\mathcal{N}_{B,k} \times \mathcal{N}_{P,\nu}$ matrix.

We also define $\mathcal{N}_{P,k} \times \mathcal{N}_{P,k}$ polynomial differential matrices $\widehat{D}_m$ such that

$$\frac{\partial \varphi_\ell}{\partial x_m}(\mathbf{x}) = \sum_{r=1}^{\mathcal{N}_{P,k}} \widehat{D}_{m,r\ell} \varphi_r(\mathbf{x}), \quad 1 \leq m \leq d,$$

where $\widehat{D}_{m,r\ell}$ is the entry of the matrix $\widehat{D}_m$ at r -th row and ℓ -th column. Then $V_k^\kappa \widehat{D}_m$ is the Vandermonde matrix of $\{\partial_{x_m} \varphi_\ell(\mathbf{x})\}_{\ell=1}^{\mathcal{N}_{P,k}}$ on T_κ . We then define the continuous and discrete inner products on T_κ and e that

$$(2.8) \quad (u, v)_{T_\kappa} = \int_{T_\kappa} uv \, d\mathbf{x}, \quad (u, v)_{T_\kappa, \omega} = \sum_{j=1}^{\mathcal{N}_{Q,k}} \omega_j^\kappa u(\mathbf{x}_j^\kappa) v(\mathbf{x}_j^\kappa) = (\overrightarrow{u^\kappa})^T M^\kappa \overrightarrow{v^\kappa},$$

$$(2.9) \quad \langle u, v \rangle_e = \int_e uv \, dS, \quad \langle u, v \rangle_{e, \tau} = \sum_{s=1}^{\mathcal{N}_{B,k}} \tau_s^e u(\mathbf{x}_s^e) v(\mathbf{x}_s^e) = (\overrightarrow{u^e})^T B^e \overrightarrow{v^e},$$

where the vector of nodal function $\vec{w}^\kappa$ and $\vec{w}^e$ are given as

$$(2.10) \quad \vec{w}^\kappa = [w(\mathbf{x}_1^\kappa), \dots, w(\mathbf{x}_{\mathcal{N}_{Q,k}}^\kappa)]^T, \quad \vec{w}^e = [w(\mathbf{x}_1^e), \dots, w(\mathbf{x}_{\mathcal{N}_{B,k}}^e)]^T.$$

and the volume mass matrix M^κ and the surfaces mass matrix B^e are diagonal matrices of quadrature weights:

$$(2.11) \quad M^\kappa = \text{diag}\{\omega_1^\kappa, \dots, \omega_{\mathcal{N}_{Q,k}}^\kappa\}, \quad B^e = \text{diag}\{\tau_1^e, \dots, \tau_{\mathcal{N}_{B,k}}^e\}.$$

According to integration by parts and algebraic accuracy of $(\cdot, \cdot)_{T_\kappa, \omega}$ and $\langle \cdot, \cdot \rangle_{e, \tau}$, we can obtain the Summation By Parts property of *modal* matrices [10] and present it in the following:

$$(2.12) \quad \widehat{M}^\kappa \widehat{D}_m + (\widehat{D}_m)^T \widehat{M}^\kappa = \sum_{e \in \partial T_\kappa} n_m^{e\kappa} \widehat{B}^e,$$

where $\widehat{M}^\kappa$ and $\widehat{B}^e$ are given as

$$(2.13) \quad \widehat{M}^\kappa = (V_k^\kappa)^T M^\kappa V_k^\kappa, \quad \widehat{B}^e = (V_k^e)^T B^e V_k^e,$$

and $\mathbf{n}^{e\kappa} = [n_1^{e\kappa}, \dots, n_d^{e\kappa}]^T$ represents the unit outward normal vector at $e \in \partial T_\kappa$. More details about (2.12) can be found in [10].

In order to obtain the *nodal* SBP property, we recall the definitions of *degree k difference matrix* D_m^κ (of the size $\mathcal{N}_{Q,k} \times \mathcal{N}_{Q,k}$) and extrapolation matrices $\{R^{e\kappa}\}_{e \in \partial T_\kappa}$ (of the size $\mathcal{N}_{B,k} \times \mathcal{N}_{Q,k}$) in [10], for which the following two conditions hold:

(i) Exactness: both D_m^κ and $R^{e\kappa}$ are exact for polynomials of degree $\leq k$, i.e.

$$(2.14) \quad D_m^\kappa V_k^\kappa = V_k^\kappa \widehat{D}_m, \quad R^{e\kappa} V_k^\kappa = V_k^e.$$

(ii) Summation-by-parts: let $S_m^\kappa = M^\kappa D_m^\kappa$ and $E^{e\kappa} = (R^{e\kappa})^T B^e R^{e\kappa}$, we have

$$(2.15) \quad \begin{aligned} S_m^\kappa + (S_m^\kappa)^T &= M^\kappa D_m^\kappa + (D_m^\kappa)^T M^\kappa \\ &= \sum_{e \in \partial T_\kappa} n_m^{e\kappa} E^{e\kappa} = \sum_{e \in \partial T_\kappa} n_m^{e\kappa} (R^{e\kappa})^T B^e R^{e\kappa}, \end{aligned}$$

$n_m^{e\kappa}$ is the m -th component of the unit outward normal $\mathbf{n}^{e\kappa}$ on $e \in \partial T_\kappa$.

To obtain the difference matrix D_m^κ , the L^2 projection matrices are also needed. With the help of the discrete inner product $(\cdot, \cdot)_{T_\kappa, \omega}$, we have the L^2 projection matrix of each order [7]:

$$(2.16) \quad P_\nu^\kappa = ((V_\nu^\kappa)^T M^\kappa V_\nu^\kappa)^{-1} (V_\nu^\kappa)^T M^\kappa, \quad 0 \leq \nu \leq k.$$

In particular, for $\nu = 0$, we have the orthogonality of the operator P_0^κ that

$$(2.17) \quad \vec{c}^T M^\kappa (\vec{u}^\kappa - V_0^\kappa P_0^\kappa \vec{u}^\kappa) = 0, \quad \forall \vec{u}^\kappa \in \mathbb{R}^{\mathcal{N}_{Q,k}},$$

where $\vec{c} = [c, c, \dots, c]^T \in \mathbb{R}^{\mathcal{N}_{Q,k}}$ is a constant vector. The existence of SBP difference matrices is ensured by the following theorem [9, 18, 27, 10]:

Theorem 2.1. Assume that we have extrapolation matrices $R^{e\kappa}$ satisfying the exactness property. Then the difference matrices, given by the formula

$$(2.18) \quad D_m^\kappa = \frac{1}{2} (M^\kappa)^{-1} \sum_{e \in \partial T_\kappa} n_m^{e\kappa} (R^{e\kappa} + V_k^e P_k^\kappa)^T B^e (R^{e\kappa} - V_k^e P_k^\kappa) + V_k^\kappa \widehat{D}_m P_k^\kappa$$

also satisfy the exactness property and the SBP property.

For the choice of the extrapolation matrices $R^{e\kappa}$, the readers can refer to [10, Remark 3.2]. Now we introduce the vector of nodal values to incorporate vector-valued functions $\mathbf{u}$:

$$(2.19) \quad \vec{\mathbf{u}}^\kappa = [\mathbf{u}(\mathbf{x}_1^\kappa), \dots, \mathbf{u}(\mathbf{x}_{\mathcal{N}_{Q,k}}^\kappa)]^T, \quad \vec{\mathbf{u}}^e = [\mathbf{u}(\mathbf{x}_1^e), \dots, \mathbf{u}(\mathbf{x}_{\mathcal{N}_{B,k}}^e)]^T,$$

as well as the Kronecker products

$$\begin{aligned} \mathbf{M}^\kappa &= M^\kappa \otimes I_n, & \mathbf{B}^e &= B^e \otimes I_n, & \mathbf{D}_m^\kappa &= D_m^\kappa \otimes I_n, & \mathbf{R}^{e\kappa} &= R^{e\kappa} \otimes I_n, \\ \widehat{\mathbf{M}}^\kappa &= \widehat{M}^\kappa \otimes I_n, & \widehat{\mathbf{D}}_m &= \widehat{D}_m \otimes I_n, & \mathbf{V}_r^\kappa &= V_r^\kappa \otimes I_n, & \mathbf{V}_r^e &= V_r^e \otimes I_n. \end{aligned}$$

The following SBP properties still hold true

$$(2.20) \quad \mathbf{S}_m^\kappa = \mathbf{M}^\kappa \mathbf{D}_m^\kappa, \quad \mathbf{E}^{e\kappa} = (\mathbf{R}^{e\kappa})^T \mathbf{B}^e \mathbf{R}^{e\kappa}, \quad \mathbf{S}_m^\kappa + (\mathbf{S}_m^\kappa)^T = \sum_{e \in \partial T_\kappa} n_m^{e\kappa} \mathbf{E}^{e\kappa}.$$

3. High order entropy stable **essentially** oscillation-free DG schemes.

In this section, we proceed to construct the entropy stable **essentially** OFDG schemes for (1.1). We first review the classic nodal DG methods and quadrature-based entropy stable DG methods with collocated surface nodes [9, 10]. The method successfully achieves entropy stability thanks to the SBP property of the corresponding discrete operators [16, 17, 47] and the *flux differencing* technique with entropy conservative fluxes [3, 9]. These two treatments are extremely important for the reason that they can recover the integration by parts and chain rule at the discrete level respectively. Now let us introduce the classic nodal DG method in the following.

3.1. Nodal DG schemes. In this subsection, we will briefly introduce the classic nodal DG schemes. Given polynomial degree $k \geq 0$, we define the DG finite element space:

$$(3.1) \quad \mathbf{W}_h^k := \{\mathbf{w}_h : \mathbf{w}_h|_{T_\kappa} = \mathbf{w}_h^\kappa \in [\mathcal{P}^k(T_\kappa)]^n, 1 \leq \kappa \leq N_h\},$$

and it means $\mathbf{w}_h$ is a vector of polynomials of degree k when restricted on T_κ . We seek $\mathbf{u}_h \in \mathbf{W}_h^k$ such that for each $\mathbf{w}_h \in \mathbf{W}_h^k$ and $1 \leq \kappa \leq N_h$, we have

$$(3.2) \quad \left(\frac{\partial \mathbf{u}_h^\kappa}{\partial t}, \mathbf{w}_h^\kappa \right)_{T_\kappa} - \sum_{m=1}^d \left(\mathbf{f}_m(\mathbf{u}_h^\kappa), \frac{d\mathbf{w}_h^\kappa}{dx_m} \right)_{T_\kappa} = - \sum_{e \in \partial T_\kappa} \langle \hat{\mathbf{f}}_n(\mathbf{u}_h^\kappa, \mathbf{u}_h^\kappa), \mathbf{w}_h^\kappa \rangle_e,$$

where $\hat{\mathbf{f}}_n$ is the interface numerical flux function, and $e = \partial T_\kappa \cap \partial T_{\tilde{\kappa}}$ is the interface of the element T_κ and its adjacent element $T_{\tilde{\kappa}}$. Since $\{\varphi_\ell(\mathbf{x})\}_{\ell=1}^{\mathcal{N}_{P,k}}$ are basis functions of $\mathcal{P}^k(\mathbb{R}^d)$, we can expand $\mathbf{u}_h^\kappa$ and $\mathbf{w}_h^\kappa$ under $\{\varphi_\ell(\mathbf{x})\}_{\ell=1}^{\mathcal{N}_{P,k}}$:

$$\mathbf{u}_h^\kappa(\mathbf{x}, t) = \sum_{\ell=1}^{\mathcal{N}_{P,k}} \hat{\mathbf{u}}_\ell^\kappa(t) \varphi_\ell(\mathbf{x}), \quad \mathbf{w}_h^\kappa(\mathbf{x}, t) = \sum_{\ell=1}^{\mathcal{N}_{P,k}} \hat{\mathbf{w}}_\ell^\kappa(t) \varphi_\ell(\mathbf{x}).$$

We now replace the continuous inner products by the volume quadrature rule and the surface quadrature rule and rewrite (3.2) as follows:

$$(3.3) \quad (\widehat{\mathbf{w}}^\kappa)^T \widehat{\mathbf{M}}^\kappa \frac{d\widehat{\mathbf{u}}^\kappa}{dt} - \sum_{m=1}^d (\mathbf{V}_k^\kappa \widehat{\mathbf{D}}_m \widehat{\mathbf{w}}^\kappa)^T \mathbf{M}^\kappa \widehat{\mathbf{f}}_m^\kappa = - \sum_{e \in \partial T_\kappa} (\mathbf{V}_e^e \widehat{\mathbf{w}}^\kappa)^T \mathbf{B}^e \widehat{\mathbf{f}}_n^{e\kappa,*},$$

where $\vec{\hat{\mathbf{u}}}^\kappa$, $\vec{\hat{\mathbf{w}}}^\kappa$, $\vec{\mathbf{f}}_m^\kappa$ and $\vec{\mathbf{f}}_{\mathbf{n}}^{e\kappa,*}$ are given as

$$\begin{aligned}\vec{\hat{\mathbf{u}}}^\kappa &= [\hat{\mathbf{u}}_1^\kappa, \dots, \hat{\mathbf{u}}_{\mathcal{N}_{P,k}}^\kappa]^T, \quad \vec{\hat{\mathbf{w}}}^\kappa = [\hat{\mathbf{w}}_1^\kappa, \dots, \hat{\mathbf{w}}_{\mathcal{N}_{P,k}}^\kappa]^T, \\ \vec{\mathbf{f}}_m^\kappa &= [\mathbf{f}_m(\mathbf{u}_1^\kappa), \dots, \mathbf{f}_m(\mathbf{u}_{\mathcal{N}_{Q,k}}^\kappa)]^T, \quad \vec{\mathbf{f}}_{\mathbf{n}}^{e\kappa,*} = [\hat{\mathbf{f}}_{\mathbf{n}}(\mathbf{u}_1^{e\kappa}, \mathbf{u}_1^{e\kappa}), \dots, \hat{\mathbf{f}}_{\mathbf{n}}(\mathbf{u}_{\mathcal{N}_{B,k}}^{e\kappa}, \mathbf{u}_{\mathcal{N}_{B,k}}^{e\kappa})]^T.\end{aligned}$$

Since $\vec{\hat{\mathbf{w}}}^\kappa$ can be arbitrarily chosen, we can rewrite (3.3) into the following form:

$$(3.4) \quad \frac{d\vec{\hat{\mathbf{u}}}^\kappa}{dt} - (\widehat{\mathbf{M}}^\kappa)^{-1} \sum_{m=1}^d (\mathbf{V}_k^\kappa \widehat{\mathbf{D}}_m)^T \mathbf{M}^\kappa \vec{\mathbf{f}}_m^\kappa = -(\widehat{\mathbf{M}}^\kappa)^{-1} \sum_{e \in \partial T_\kappa} (\mathbf{V}_k^e)^T \mathbf{B}^e \vec{\mathbf{f}}_{\mathbf{n}}^{e\kappa,*},$$

Multiplying (3.4) by the matrix $\mathbf{V}_k^\kappa$ from the left, with the relation $\mathbf{V}_k^\kappa (\widehat{\mathbf{M}}^\kappa)^{-1} = (\mathbf{M}^\kappa)^{-1} (\mathbf{P}_k^\kappa)^T$ we obtain the nodal formulation:

$$(3.5) \quad \frac{d\vec{\mathbf{u}}}^\kappa}{dt} - (\mathbf{M}^\kappa)^{-1} \sum_{m=1}^d (\mathbf{D}_m^\kappa)^T \mathbf{M}^\kappa \vec{\mathbf{f}}_m^\kappa = -(\mathbf{M}^\kappa)^{-1} \sum_{e \in \partial T_\kappa} (\mathbf{R}^{e\kappa})^T \mathbf{B}^e \vec{\mathbf{f}}_{\mathbf{n}}^{e\kappa,*},$$

by choosing $\vec{\mathbf{u}}^\kappa$ defined in (2.19), $\mathbf{D}_m^\kappa = \mathbf{V}_k^\kappa \widehat{\mathbf{D}}_m \mathbf{P}_k^\kappa$ and $\mathbf{R}^{e\kappa} = \mathbf{V}_k^e \mathbf{P}_k^\kappa$. According to the SBP property (2.20), we have the equivalent strong form of nodal DG formulation:

$$(3.6) \quad \frac{d\vec{\mathbf{u}}}^\kappa}{dt} + \sum_{m=1}^d \mathbf{D}_m^\kappa \vec{\mathbf{f}}_m^\kappa = (\mathbf{M}^\kappa)^{-1} \sum_{e \in \partial T_\kappa} (\mathbf{R}^{e\kappa})^T \mathbf{B}^e (\vec{\mathbf{f}}_{\mathbf{n}}^{e\kappa} - \vec{\mathbf{f}}_{\mathbf{n}}^{e\kappa,*}),$$

where $\vec{\mathbf{f}}_{\mathbf{n}}^{e\kappa}$ are the vectors of extrapolated nodal values on face $e \in \partial T_\kappa$ that

$$(3.7) \quad \vec{\mathbf{f}}_{\mathbf{n}}^{e\kappa} = \sum_{m=1}^d n_m^{e\kappa} \vec{\mathbf{f}}_m^{e\kappa}, \quad \vec{\mathbf{f}}_m^{e\kappa} = \mathbf{R}^{e\kappa} \vec{\mathbf{f}}_m^\kappa.$$

An interpretation of the relation between the modal and nodal DG formulations was given in [10, Remark 3.4].

3.2. Entropy stable DG schemes with collocated surface nodes. Now we continue to introduce the entropy stable nodal DG formulation based on (3.6). The nodal DG formulation (3.6) does not satisfy any entropy inequality because the second term on the left hand side does not satisfy the discrete chain rule. To recover the chain rule on the discrete level, we replace the difference term in (3.6) with high order difference operation of entropy conservative fluxes. This technique is termed by *flux differencing* which is crucial to the entropy balance within an element. A deeper understanding of the flux differencing technique was given in [10, Appendix A]. The modified DG formulation with collocated surface nodes [9] reads

$$(3.8) \quad \frac{d\vec{\mathbf{u}}}^\kappa}{dt} + 2 \sum_{m=1}^d \mathbf{D}_m^\kappa \circ \mathbf{F}_{m,S}(\vec{\mathbf{u}}^\kappa, \vec{\mathbf{u}}^\kappa) \vec{\mathbf{1}}^\kappa = (\mathbf{M}^\kappa)^{-1} \sum_{e \in \partial T_\kappa} (\mathbf{R}^{e\kappa})^T \mathbf{B}^e (\vec{\mathbf{f}}_{\mathbf{n}}^{e\kappa} - \vec{\mathbf{f}}_{\mathbf{n}}^{e\kappa,*})$$

where $\circ$ denotes the Hadamard (pointwise) product of vectors and matrices, and $\mathbf{F}_{m,S}(\cdot, \cdot)$ is the matrix of pairwise combinations of entropy conservative fluxes [30, 6, 13, 14]:

$$(3.9) \quad \mathbf{F}_{m,S}(\vec{\mathbf{u}}_L^\kappa, \vec{\mathbf{u}}_R^\kappa) = \begin{bmatrix} \text{diag}(\mathbf{f}_{m,S}(\mathbf{u}_{L,1}, \mathbf{u}_{R,1})) & \cdots & \text{diag}(\mathbf{f}_{m,S}(\mathbf{u}_{L,1}, \mathbf{u}_{R,\mathcal{N}_R})) \\ \vdots & \ddots & \vdots \\ \text{diag}(\mathbf{f}_{m,S}(\mathbf{u}_{L,\mathcal{N}_L}, \mathbf{u}_{R,1})) & \cdots & \text{diag}(\mathbf{f}_{m,S}(\mathbf{u}_{L,\mathcal{N}_L}, \mathbf{u}_{R,\mathcal{N}_R})) \end{bmatrix},$$

for $\vec{\mathbf{u}}_L \in \mathbb{R}^{n_{\mathcal{N}_L}}$ and $\vec{\mathbf{u}}_R \in \mathbb{R}^{n_{\mathcal{N}_R}}$. Note that this method requires the collocated surface quadrature nodes ($\{\mathbf{x}_x^e\}_{s=1}^{\mathcal{N}_{B,k}}$ is a subset of $\{\mathbf{x}_j^\kappa\}_{j=1}^{\mathcal{N}_{Q,k}}$ for $e \in \partial T_\kappa$), thus $R^{e\kappa}$ is a simple restriction onto e and diagonal. In [9], the authors proved the nodal DG scheme (3.8) is conservative, entropy stable and high order accurate, under the assumptions that $\mathbf{f}_{m,S}$ is entropy conservative, and $\hat{\mathbf{f}}_{\mathbf{n}}$ is entropy stable for any given entropy function.

3.3. Entropy stable essentially oscillation-free DG schemes. Despite the fact that the DG scheme (3.8) possesses so many good properties, it cannot eliminate numerical oscillations near the discontinuities. To control the non-physical oscillations, we adopt the recent developed damping technique in [36, 35] and construct a suitable damping term for (3.8). Note that the added damping term should not contaminate the good properties of the entropy stable DG schemes, such as conservation, entropy stability, and high order accuracy. By a careful construction, we present the newly designed nodal DG scheme as follows:

$$(3.10) \quad \begin{aligned} \frac{d\vec{\mathbf{u}}^\kappa}{dt} + 2 \sum_{m=1}^d \mathbf{D}_m^\kappa \circ \mathbf{F}_{m,S}(\vec{\mathbf{u}}^\kappa, \vec{\mathbf{u}}^\kappa) \vec{\mathbf{1}}^\kappa \\ = (\mathbf{M}^\kappa)^{-1} \sum_{e \in \partial T_\kappa} (\mathbf{R}^{e\kappa})^T \mathbf{B}^e \left(\vec{\mathbf{f}}_{\mathbf{n}}^{e\kappa} - \vec{\mathbf{f}}_{\mathbf{n}}^{e\kappa,*} \right) - \sigma^\kappa(\mathbf{u})(\vec{\mathbf{u}}^\kappa - \mathbf{V}_0^\kappa \mathbf{P}_0^\kappa \vec{\mathbf{u}}^\kappa). \end{aligned}$$

The damping coefficient $\sigma^\kappa(\mathbf{u})$ is defined as follows

$$(3.11) \quad \sigma^\kappa(\mathbf{u}) = \max_{1 \leq s \leq n} \left(\sum_{\ell=0}^1 \frac{h_{T_\kappa}^{2\ell}}{\ell+1} \sum_{|\alpha|=\ell} \frac{1}{N_e} \sum_{v \in \partial T_\kappa} \llbracket (\mathbf{L} \partial^\alpha \mathbf{u})_s \rrbracket_v^2 \right)^{\frac{1}{2}},$$

where the vector α is the multiindex of order $|\alpha| = \alpha_1 + \dots + \alpha_d$, and $\partial^\alpha \omega = \partial_{x_1}^{\alpha_1} \dots \partial_{x_d}^{\alpha_d} \omega$. The matrix $\mathbf{L}$ comes from the characteristic decomposition such that

$$\sum_{i=1}^d n_i \mathbf{f}'_i(\bar{\mathbf{u}}) = \mathbf{L}^{-1} \mathbf{A} \mathbf{L}$$

with $\mathbf{n} = [n_1, \dots, n_d]^T$ is the unit outward normal. $\bar{\mathbf{u}}$ is some average of $\mathbf{u}$ on the point $v \in \partial T_\kappa$, and one can either use the arithmetic average or the Roe average (which is used in the numerical experiments later). The characteristic decomposition is not necessarily unique and the formulation could be complicated. Fortunately, for compressible Euler systems we have a relatively simple formulation, see e.g. [35]. For illustration purpose, for two-dimensional compressible Euler systems $\mathbf{u} = [\rho, \rho u, \rho v, E]^T$, $\mathbf{f}_1 = [\rho u, \rho u^2 + p, \rho uv, u(E + p)]^T$, $\mathbf{f}_2 = [\rho v, \rho uv, \rho v^2 + p, v(E + p)]^T$, we take $\mathbf{L}$ as follows

$$(3.12) \quad \mathbf{L} = (\gamma - 1)c^{-1} \begin{pmatrix} \frac{1}{2}(B_2 + \hat{u}c) & -\frac{1}{2}(B_1 u + n_1 c) & -\frac{1}{2}(B_1 v + n_2 c) & \frac{1}{2}B_1 \\ \hat{v}c & n_2 c & -n_1 c & 0 \\ c^2 - B_2 & B_1 u & B_1 v & -B_1 \\ \frac{1}{2}(B_2 - \hat{u}c) & -\frac{1}{2}(B_1 u - n_1 c) & -\frac{1}{2}(B_1 v - n_2 c) & \frac{1}{2}B_1 \end{pmatrix}$$

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

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

For general hyperbolic systems, it might be a good way to have $\mathbf{L}$ to be orthonormal by normalizing the left characteristic matrix. Since we only have the values of the entropy variables on nodal points, we first project $\mathbf{u}$ into the finite element space to obtain the coefficients of polynomials, $\vec{\mathbf{u}}^\kappa = \mathbf{P}_k^\kappa \vec{\mathbf{u}}^\kappa$, then we can obtain the derivatives of $\mathbf{u}$. $[[\omega|_v]]$ denotes the jump of the function ω on the vertex $v \in T_\kappa$, we only consider the adjacent neighbors of element T_κ . N_e stands for the number of edges of ∂T_κ . For example, see Fig. 1, the adjacent neighbors of element K are K_1, K_2, K_3 , thus $N_e = 3$.

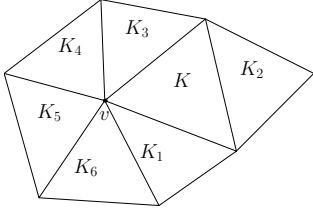


FIG. 1. Illustrating graph for the jumps in the damping coefficient σ^κ defined in (3.11).

Remark 3.1. The equation (3.11) is one of the key ingredients of the proposed method. Intuitively, It is designed as a smoothness indicator that it would be small in the smooth region and large near the discontinuous region. That means in the smooth region, the damping is small and has little effect in the DG scheme, and near the discontinuous region, the damping is large and takes effect to compress the numerical oscillations. To this end, in (3.11) we make use of the jumps of the numerical solution and its derivatives across the surface to indicate the smoothness of the numerical solution. Formally, the damping term in (3.10) makes high-order coefficients of the DG solution to be damped by a factor $\exp(-\sigma^\kappa(\mathbf{u})t)$.

Now we are ready to provide our main theorem, which states that the nodal DG method (3.10) is conservative, entropy stable, and maintains high order accuracy, under the assumptions that

- (H1) The volume quadrature rule has collocated surface quadrature nodes, and $R^{e\kappa}$ is a simple restriction onto e and $E^{e\kappa}$ is diagonal.
- (H2) The simplex meshes $\{\mathcal{T}_h\}$ are shape regular and quasi-uniform.
- (H3) All mappings and numerical fluxes (e.g. $\mathbf{v}(\mathbf{u})$, $\mathbf{f}_m(\mathbf{u})$, $\mathbf{f}_{m,S}(\mathbf{u}_L, \mathbf{u}_R)$, etc) are smooth and Lipschitz continuous.
- (H4) $\mathbf{f}_{m,S}(\mathbf{u}_L, \mathbf{u}_R)$ is entropy conservative and $\hat{\mathbf{f}}_{\mathbf{n}}$ is entropy stable with respect to the given entropy function $U(\mathbf{u})$.

Theorem 3.1. If above assumptions (H1) - (H4) hold, then the scheme (3.10) is consistent in the sense that for a smooth solution $\mathbf{u}$ of (1.1) and a smooth entropy variable $\mathbf{v}$ with respect to U , the local truncation error is of high order:

$$\begin{aligned}
 (3.13) \quad & \frac{d\mathbf{u}_j^\kappa}{dt} + 2 \sum_{m=1}^d \sum_{i=1}^{\mathcal{N}_{Q,k}} D_{m,j,i}^\kappa \mathbf{f}_{m,S}(\mathbf{u}_j^\kappa, \mathbf{u}_i^\kappa) - \sum_{e \in \partial T_\kappa} \sum_{s=1}^{\mathcal{N}_{B,k}} R_{sj}^{e\kappa} \frac{\tau_s^e}{\omega_j^\kappa} (\mathbf{f}_{\mathbf{n}}^{e\kappa} - \hat{\mathbf{f}}_{\mathbf{n}}(\mathbf{u}_s^{e\kappa}, \mathbf{u}_s^{e\tilde{\kappa}})) \\
 & + \sigma^\kappa(\mathbf{u}) \left(\mathbf{u}_j^\kappa - \sum_{i=1}^{\mathcal{N}_{Q,k}} (V_0^\kappa P_0^\kappa)_{ji} \mathbf{u}_i^\kappa \right) = \mathcal{O}(h^k), \quad j = 1, \dots, \mathcal{N}_{Q,k}.
 \end{aligned}$$

It is also conservative and entropy stable with respect to U given by

$$(3.14) \quad \frac{d}{dt} \left(\sum_{\kappa=1}^{N_h} (\vec{1}^\kappa)^T \mathbf{M}^\kappa \vec{\mathbf{u}}^\kappa \right) = 0, \quad \frac{d}{dt} \left(\sum_{\kappa=1}^{N_h} (\vec{1}^\kappa)^T M^\kappa \vec{U}^\kappa \right) \leq 0,$$

where $\vec{U}^\kappa = [U(\mathbf{u}_1^\kappa), \dots, U(\mathbf{u}_{\mathcal{N}_{Q,k}}^\kappa)]^T$.

Proof. Since $\mathbf{u}$ is a smooth function, by the definition of the damping coefficient $\sigma^\kappa(\mathbf{u}) = 0$ in (3.11), we have $\sigma^\kappa(\mathbf{u}) = 0$. From the proof in [10, Theorem 4.1], we can obtain (3.13). And due to the fact that $\sigma^\kappa(\mathbf{u})$ is a scalar and by the property of the L^2 projection matrix P_0^κ in (2.17), we have

$$(3.15) \quad \begin{aligned} & (\vec{1}^\kappa)^T \mathbf{M}^\kappa \sigma^\kappa(\mathbf{u}) (\vec{\mathbf{u}}^\kappa - \mathbf{V}_0^\kappa \mathbf{P}_0^\kappa \vec{\mathbf{u}}^\kappa) \\ &= \sigma^\kappa(\mathbf{u}) (\vec{1}^\kappa)^T \mathbf{M}^\kappa (\vec{\mathbf{u}}^\kappa - \mathbf{V}_0^\kappa \mathbf{P}_0^\kappa \vec{\mathbf{u}}^\kappa) = 0. \end{aligned}$$

Note that $\mathbf{V}_0^\kappa = \vec{1}^\kappa$, together with the conservative form of (3.8), implies the conservation of $\mathbf{u}$ in (3.10). For the entropy stability, we have

$$(3.16) \quad \frac{d}{dt} \left(\sum_{\kappa=1}^{N_h} (\vec{1}^\kappa)^T M^\kappa \vec{U}^\kappa \right) = \sum_{\kappa=1}^{N_h} (\vec{\mathbf{v}}^\kappa)^T \mathbf{M}^\kappa \frac{d}{dt} \vec{\mathbf{u}}^\kappa.$$

Since $\vec{U}^\kappa(\mathbf{V}_0^\kappa \mathbf{P}_0^\kappa \vec{\mathbf{u}}^\kappa)$ is a constant vector for each entropy variable, by (2.17) we have

$$(3.17) \quad \begin{aligned} & - (\vec{\mathbf{v}}^\kappa)^T \mathbf{M}^\kappa \sigma^\kappa(\mathbf{u}) (\vec{\mathbf{u}}^\kappa - \mathbf{V}_0^\kappa \mathbf{P}_0^\kappa \vec{\mathbf{u}}^\kappa) \\ &= -\sigma^\kappa(\mathbf{u}) \left(\vec{U}^\kappa(\vec{\mathbf{u}}^\kappa) - \vec{U}^\kappa(\mathbf{V}_0^\kappa \mathbf{P}_0^\kappa \vec{\mathbf{u}}^\kappa) \right)^T \mathbf{M}^\kappa (\vec{\mathbf{u}}^\kappa - \mathbf{V}_0^\kappa \mathbf{P}_0^\kappa \vec{\mathbf{u}}^\kappa) \quad (\text{by (3.15)}) \end{aligned}$$

By using the fact that entropy $U''(\mathbf{u}) \geq 0$, we have

$$(3.18) \quad (U'(\mathbf{u}_1) - U'(\mathbf{u}_2)) \cdot (\mathbf{u}_1 - \mathbf{u}_2) \geq 0, \quad \forall \mathbf{u}_1, \mathbf{u}_2.$$

Therefore, with the positive weights $\omega_j^\kappa \geq 0$, for $\forall 1 \leq j \leq \mathcal{N}_{Q,k}$ we have

$$(3.19) \quad \left(U'(\mathbf{u}(\mathbf{x}_j^\kappa)) - U'((\mathbf{V}_0^\kappa \mathbf{P}_0^\kappa \vec{\mathbf{u}}^\kappa)_j) \right)^T (\mathbf{u}(\mathbf{x}_j^\kappa) - (\mathbf{V}_0^\kappa \mathbf{P}_0^\kappa \vec{\mathbf{u}}^\kappa)_j) \omega_j^\kappa \geq 0.$$

Then by above inequality together with (3.17), we obtain

$$(3.20) \quad \sum_{\kappa=1}^{N_h} (\vec{\mathbf{v}}^\kappa)^T \mathbf{M}^\kappa \sigma^\kappa(\mathbf{u}) (\vec{\mathbf{u}}^\kappa - \mathbf{V}_0^\kappa \mathbf{P}_0^\kappa \vec{\mathbf{u}}^\kappa) \geq 0.$$

This indicates the newly added damping term is also entropy stable. Finally, by the entropy stability of the scheme (3.8) in [9] and (3.20), we obtain (3.14). $\square$

Remark 3.2. From the proof of Theorem 3.1, we can see that a more natural idea is to construct the damping term with the entropy variables, that is $\sigma^\kappa(\mathbf{u})(\vec{\mathbf{v}}^\kappa - \mathbf{V}_0^\kappa \mathbf{P}_0^\kappa \vec{\mathbf{v}}^\kappa)$. We can still have the properties of conservation and entropy dissipation, however, the numerical investigations indicate this choice does not control the spurious oscillations well, thus it is not adopted.

Remark 3.3. Note that the projection in the damping term in (3.10) is slightly different from [35], since here we only use the projection $\mathbf{P}_0^\kappa$ for the sake of entropy stability. This is because if $U'(\mathbf{u})$ is not a linear function with respect to $\mathbf{u}$, then it is hard to make use of the L^2 projection property to obtain the entropy stability when $U'(\mathbf{V}_r^\kappa \mathbf{P}_r^\kappa \mathbf{u}^\kappa)$, $r \geq 1$ are not the values of some polynomial at nodes.

Remark 3.4. Since the damping term in (3.10) vanishes in the evolution scheme of the cell average, the positive preserving limiter, which was developed by Zhang and Shu in [53, 54] and does not increase entropy [9, Theorem 3.7], can also be applied to ESOFDG scheme (3.10). It is worth noting that we do not use any limiters to demonstrate the robustness of the ESOFDG method in the numerical experiments.

3.4. Entropy stable essentially OFDG method on the general set of nodes. Next, we introduce three approaches to obtain the entropy stable DG methodology for arbitrary volume and surface quadrature rules. The first one is the so-called hybridized SBP operators approach proposed by Chan in [7, 8]. The key idea is to combine volume nodes and surface nodes together to obtain the hybridized SBP operators. The second approach is the global SBP operators approach, which was developed by Crean et al. in [13, 14]. The key idea is to view the nodal values on different elements as a whole, grouping them into a single global vector. Then the global SBP operators should be constructed accordingly and the entropy dissipation function with respect to U can ensure the entropy stable property. The third approach is to enforce the entropy balance directly [1], in which the method was written in the more general residual distribution framework. A simple linear correction term would be added in the original nodal DG scheme to obtain the entropy stable property. For the general cases, we present a generic form as follows:

$$(3.21) \quad \frac{d\vec{\mathbf{u}}^\kappa}{dt} = \mathbf{r}^\kappa(\vec{\mathbf{u}}^g), \quad \mathbf{u}^g = \begin{bmatrix} \vec{\mathbf{u}}^1 \\ \vdots \\ \vec{\mathbf{u}}^{N_h} \end{bmatrix}$$

with $\mathbf{u}^g$ being the global column vector of the nodal values on the whole elements. The scheme (3.21) has the following properties:

$$(3.22) \quad \sum_{\kappa=1}^{N_h} (\vec{\mathbf{1}}^\kappa)^T \mathbf{M}^\kappa \mathbf{r}^\kappa(\vec{\mathbf{u}}^g) = 0, \quad \sum_{\kappa=1}^{N_h} (\vec{\mathbf{v}}^\kappa)^T \mathbf{M}^\kappa \mathbf{r}^\kappa(\vec{\mathbf{u}}^g) \leq 0.$$

The corresponding entropy stable essentially OFDG scheme is written as

$$(3.23) \quad \frac{d\vec{\mathbf{u}}^\kappa}{dt} = \mathbf{r}^\kappa(\vec{\mathbf{u}}^g) - \sigma^\kappa(\mathbf{u})(\vec{\mathbf{u}}^\kappa - \mathbf{V}_0^\kappa \mathbf{P}_0^\kappa \vec{\mathbf{u}}^\kappa).$$

Theorem 3.2. If (H1) - (H4) hold true, the scheme (3.23) is conservative and entropy stable.

Proof. The proof is similar to Theorem 3.1 and thus omitted. $\square$

4. Numerical experiments. In this section we show some numerical results to justify the good performance of the proposed algorithm. In one-dimensional problems, we use the three-point Gauss-Lobatto quadrature for $k = 1$, and four-point Gauss-Lobatto quadrature for $k = 2$ and five-point Gauss-Lobatto quadrature for $k = 3$.

We also use its tensor-product for two-dimensional problems. In the figures below, for simplicity we only plot the cell averages within each cell instead of showing the full polynomial. Note that the full polynomial solution may still contain some slight oscillations near the discontinuities, please see [35] for more details. We also use the classic fourth order Runge-Kutta method as our time-stepping method, and the time step size $\tau = O(h)$. To be more specific, we take the one-dimensional ESOFDG schemes for illustration purpose. For one-dimensional problems, we take the CFL condition as

$$\tau = \frac{CFL}{\lambda_0 + a_0} h,$$

where λ_0 is the maximum of the spectral radius of the Jacobian $\mathbf{f}'(\mathbf{u})$ over each element, and $a_0 = \max_{\kappa} \sigma^{\kappa}(\mathbf{u})$ defined in (3.11). The CFL number is a constant and one can take e.g. $CFL = c/(2k+1)$, c is a constant. Note that the standard DG methods have the empirical CFL number that $c = 1$ [12]. And in the numerical simulation, we take c less than 1 for the ESOFDG schemes.

In several numerical examples we plot the total entropy against time by the formulation

$$\text{total entropy} = \sum_{\kappa} \sum_{j=1}^{\mathcal{N}_{Q,k}} \omega_j^{\kappa} U(\mathbf{u}_h^{\kappa}(\mathbf{x}_j^{\kappa}, t_n))$$

at time level $t = t_n$. Throughout the paper, the entropy stable fluxes are chosen as the local Lax-Friedrichs (LFF) fluxes [9] that

$$h^{\text{LLF}}(a, b) = \frac{1}{2}(\mathbf{f}(a) + \mathbf{f}(b) - \alpha(b - a)),$$

where the parameter α is taken as $\alpha = \max(|f'(u_{j+\frac{1}{2}}^-)|, |f'(u_{j+\frac{1}{2}}^+)|)$ for the scalar flux function, in order to ensure the monotonicity and the entropy stability of the LLF flux on the element interfaces, we must approximate α properly. For Euler equations we use the two-rarefaction approximation [25] to estimate the bounds of wave speeds for Euler equations with $1 \leq \gamma \leq 5/3$. The explicit formulation of α can be found in [9, Theorem A.2].

Moreover, for one-dimensional compressible Euler equations with $\mathbf{u} = [\rho, \rho u, E]^T$, $\mathbf{f}(\mathbf{u}) = [\rho u, \rho u^2 + p, u(E + p)]^T$, we use the entropy function and entropy variables as

$$(4.1) \quad U = -\frac{\rho s}{\gamma - 1}, \quad \mathbf{v} = \left[\frac{\gamma - s}{\gamma - 1} - \frac{\rho u^2}{2p}, \frac{\rho u}{p}, -\frac{\rho}{p} \right]^T,$$

where $s = \log(p\rho^{-\gamma})$ is the physical specific entropy, and the entropy conservative flux was recommended by Chandrashekar in [6]:

$$(4.2) \quad \mathbf{f}_S(\vec{\mathbf{u}}_L, \vec{\mathbf{u}}_R) = \left[(\bar{\rho})^{\log} \bar{u}, (\bar{\rho})^{\log} \bar{u}^2 + \tilde{p}, \bar{u} \left(\frac{(\bar{p})^{\log}}{\gamma - 1} + \tilde{E} \right) \right]^T$$

where $\bar{z} = \frac{1}{2}(z_L + z_R)$, $(\bar{z})^{\log} = \frac{z_R - z_L}{\log z_R - \log z_L}$ and

$$\beta = \frac{\rho}{2p}, \quad \tilde{p} = \frac{\bar{\rho}}{2\bar{\beta}}, \quad (\tilde{p})^{\log} = \frac{(\bar{\rho})^{\log}}{2(\bar{\beta})^{\log}}, \quad \tilde{E} = \frac{1}{2}(\bar{\rho})^{\log}(2\bar{u}^2 - \bar{u}^2) + \tilde{p}.$$

One can also use the affordable entropy conservative flux which was suggested by Ismail and Roe [30]. Since the results of numerical examples do not have obvious

differences with these two fluxes, we only report the results when using the entropy conservative flux of Chandrashekar. For two-dimensional compressible Euler equations with $\mathbf{u} = [\rho, \rho u, \rho v, E]^T$, $\mathbf{f}_1(\mathbf{u}) = [\rho u, \rho u^2 + p, \rho uv, u(E + p)]^T$, $\mathbf{f}_2(\mathbf{u}) = [\rho v, \rho uv, \rho v^2 + p, v(E + p)]^T$, we use the entropy function and entropy variables as

$$(4.3) \quad U = -\frac{\rho s}{\gamma - 1}, \quad \mathbf{v} = \left[\frac{\gamma - s}{\gamma - 1} - \frac{\rho(u^2 + v^2)}{2p}, \frac{\rho u}{p}, \frac{\rho v}{p}, -\frac{\rho}{p} \right]^T,$$

and the entropy conservative fluxes are given as

$$(4.4) \quad \mathbf{f}_{1,S}(\vec{\mathbf{u}}_L, \vec{\mathbf{u}}_R) = \begin{bmatrix} (\bar{\rho})^{\log \bar{u}} \\ (\bar{\rho})^{\log \bar{u}^2 + \tilde{p}} \\ (\bar{\rho})^{\log \bar{u} \bar{v}} \\ \bar{u} \left(\frac{(\bar{p})^{\log}}{\gamma - 1} + \tilde{E} \right) \end{bmatrix}, \quad \mathbf{f}_{2,S}(\vec{\mathbf{u}}_L, \vec{\mathbf{u}}_R) = \begin{bmatrix} (\bar{\rho})^{\log \bar{v}} \\ (\bar{\rho})^{\log \bar{u} \bar{v}} \\ (\bar{\rho})^{\log \bar{v}^2 + \tilde{p}} \\ \bar{v} \left(\frac{(\bar{p})^{\log}}{\gamma - 1} + \tilde{E} \right) \end{bmatrix},$$

where the notations are the same as in one-dimensional case except

$$\tilde{E} = \frac{1}{2}(\bar{\rho})^{\log}(2(\bar{u}^2 + \bar{v}^2) - \overline{u^2 + v^2}) + \tilde{p}.$$

For more details we refer the readers to [6, 9, 7]. The ratio of the specific heat γ is taken to be 1.4 for air unless specified otherwise. Throughout this paper, we just use the high order numerical quadrature rule to approximate the errors in the L^2 norm instead of integrating them exactly. We put the codes for Example 2 and Example 14 on GitHub, one can download the codes at <https://github.com/YongLiu12/Entropy-stable-OFDG>

4.1. One-dimensional problems.

Example 1. We firstly consider the linear scalar conservation laws that $u_t + u_x = 0$ with periodic boundary condition. Two initial conditions are taken as follows:

- (a) The smooth case: $u_0(x) = \sin(\pi x)^2 + 1$, $x \in (-1, 1)$.
- (b) The non-smooth case:

$$u_0(x) = \begin{cases} \sin(\pi x), & -0.5 \leq x \leq 0.5, \\ 0, & \text{otherwise.} \end{cases}$$

The computational domain is $(-1, 1)$ and the final time is taken as $T = 1.2$ and $T = 5.0$ for case (a) and (b) respectively. Both of these two cases are using the square entropy function $U(u) = u^2/2$.

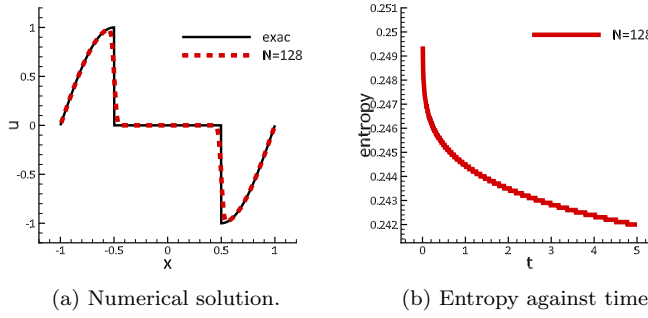
From Table 1, we observe the optimal convergence rate for the smooth solution in Example 1. In Fig. 2(a), we can clearly see the spurious oscillation is controlled well and we also plot the discrete entropy against time in Fig. 2(b). Although we have not proved the entropy stability of the fully discrete schemes with the explicit Runge-Kutta time discretization method, we are still able to observe the phenomenon of dissipative entropy.

Example 2. In this example, we consider the Burgers' equation that $u_t + \left(\frac{u^2}{2}\right)_x = 0$ with periodic boundary condition. The initial condition is $u_0(x) = 2\sin(x) + 1$, $x \in (0, 2\pi)$. Since any convex function can be chosen as the entropy function for scalar conservation laws, we adopt the entropy function $U(u) = 0.1e^u + 0.45u^2$ as an example and compute the solution at different final time $T = 0.3$ and $T = 5$. Note

TABLE 1
Errors and orders of the case (a) in Example 1 with the final time $T = 1.2$.

	$k = 1$		$k = 2$		$k = 3$	
N	$\ u - u_h\ _{L^2}$	order	$\ u - u_h\ _{L^2}$	order	$\ u - u_h\ _{L^2}$	order
16	1.819E-02	—	1.148E-03	—	2.545E-04	—
32	3.277E-03	2.473	8.899E-05	3.690	9.937E-06	4.679
64	6.291E-04	2.381	9.234E-06	3.269	3.605E-07	4.785
128	1.363E-04	2.207	1.081E-06	3.095	1.315E-08	4.777
256	3.238E-05	2.073	1.326E-07	3.027	5.482E-10	4.584
512	7.978E-06	2.021	1.649E-08	3.007	2.799E-11	4.292

FIG. 2. The numerical solution and entropy of the case (b) in Example 1 with final time $T = 5.0$, $k = 2$, $N = 128$.



that the exact solution $u(x, t)$ stays smooth when $T = 0.3$ so we can test the accuracy of the ESOFDG scheme. When $T = 5$, the exact solution develops a discontinuity and spurious oscillations may occur for classic high order DG schemes.

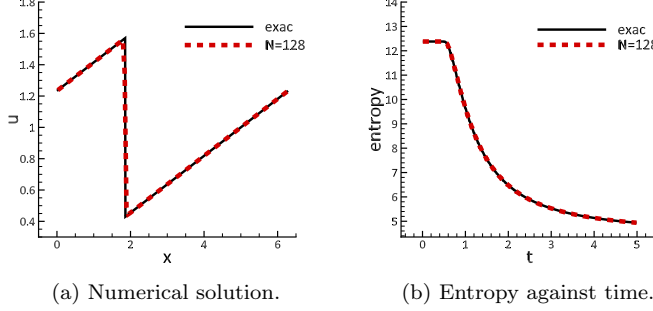
For nonlinear scalar equations, we also observe optimal convergence for $k = 1, 2, 3$ in Table 2. When $T = 5$, a shock has appeared due to the nonlinearity, and we observe the ESOFDG scheme captures the shock well without visible oscillations in Fig. 3. We also compute the evolution of the exact entropy, and we can see the numerical entropy coincides with it well in Fig. 3(b).

TABLE 2
Errors and orders in Example 2 with the final time $T = 0.3$.

	$k = 1$		$k = 2$		$k = 3$	
N	$\ u - u_h\ _{L^2}$	order	$\ u - u_h\ _{L^2}$	order	$\ u - u_h\ _{L^2}$	order
16	1.852E-02	—	3.125E-03	—	6.374E-04	—
32	4.631E-03	1.999	3.381E-04	3.028	5.770E-05	3.466
64	1.163E-03	1.993	4.326E-05	3.147	3.899E-06	3.888
128	2.901E-04	2.003	5.258E-06	3.040	2.033E-07	4.261
256	7.244E-05	2.002	6.504E-07	3.015	1.010E-08	4.331
512	1.810E-05	2.001	8.102E-08	3.005	5.488E-10	4.202

Example 3. In the following we consider the Riemann problem [32] for the one-

FIG. 3. The numerical solution and entropy in Example 2 with the final time $T = 5$, $k = 2$, $N = 128$.



dimensional nonconvex scalar hyperbolic conservation law $u_t + f(u)_x = 0$ with

$$(4.5) \quad f(u) = \frac{u(1-u)}{4}, \quad u < \frac{1}{2}; \quad f(u) = \frac{u^2}{2} - \frac{u}{2} + \frac{3}{16}, \quad u \geq \frac{1}{2}.$$

The initial condition is

$$u(x, 0) = u_l, \quad x < \frac{1}{4}; \quad u(x, 0) = u_r, \quad x \geq \frac{1}{4}.$$

The computational domain is $\Omega = (0, 1)$. We test two cases in the following:

- (i) $u_l = 0$, $u_r = 1$, and the final time is $T = 1$;
- (ii) $u_l = 1$, $u_r = 0$, and the final time is $T = 2$.

We also take three kinds of entropy functions for comparison as follows:

- [a] $U(u) = \frac{1}{2}u^2$;
- [b] $U(u) = 0.1e^u + 0.45u^2$;
- [c] $U(u) = u \arctan(20u) - \frac{1}{40} \log(1 + 400u^2)$.

Nonconvex hyperbolic conservation laws are very challenging in computation, because if their numerical schemes are not carefully constructed, they may fail to converge to the unique entropy solution or may be too slow to converge that would require impractically fine meshes [32]. These three kinds of entropy functions correspond to different entropy stability. The first entropy is the classic square entropy, thus the entropy stability is equivalent to the L^2 -norm stability which is obtained in the classic DG framework. The second entropy is artificially constructed which is the same as in Example 2. The third one can be viewed as a smooth approximation of the Kruzhkov's entropy $U(u) = |u|$ [33]. In Fig. 4, we observe the ESOFDG scheme works well for all three kinds of entropy functions. The discrete entropy against time is also plotted for different initial conditions. In Fig. 4(d), the entropy increases for the reason that the boundary terms do not vanish and we can only obtain entropy stability instead of entropy dissipation that

$$\frac{d}{dt} \left(\sum_{\kappa=1}^{N_h} (\vec{1}^{\kappa})^T M^{\kappa} \vec{U}^{\kappa} \right) \leq C$$

where $C > 0$ is some positive constant. For instance, with entropy [a] we have the entropy flux

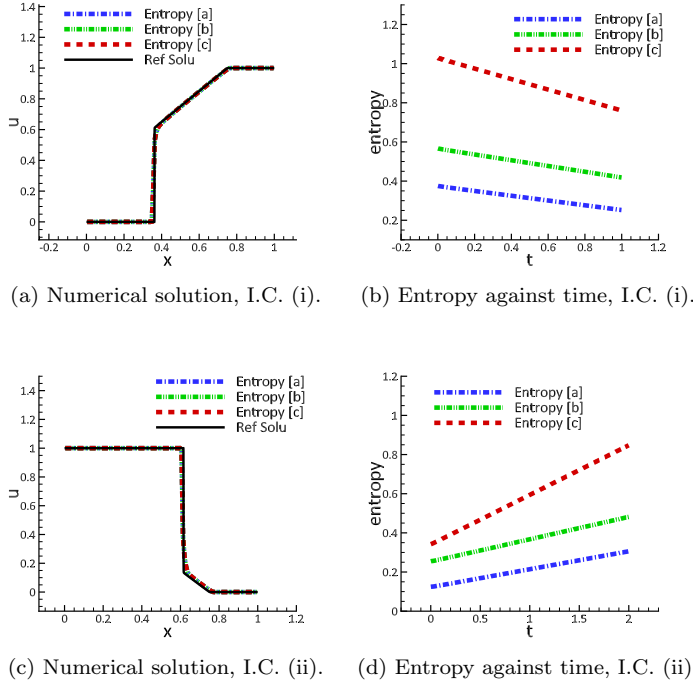
$$F(u) = \int U'(u) f'(u) du = \begin{cases} -\frac{u^3}{6} + \frac{u^2}{8}, & u < \frac{1}{2}, \\ \frac{u^3}{3} - \frac{u^2}{4} + \frac{1}{32}, & u \geq \frac{1}{2}. \end{cases}$$

Then for the I.C. (ii) we have

$$\frac{d}{dt} \left(\sum_{\kappa=1}^{N_h} (\vec{1}^{\kappa})^T M^{\kappa} \vec{U}^{\kappa} \right) \leq -F(u_r) + F(u_l) = \frac{11}{96} \approx 0.11.$$

From Fig. 4(d), the tangent of the evolution curve of entropy [a] is about 0.09 which is less than 0.11.

FIG. 4. The numerical solution and entropy in Example 3 with I.C. (i) and (ii), $k = 2, N = 128$.



Example 4. Consider a Riemann problem for one-dimensional Buckley-Leverett equation with the flux function defined as

$$(4.6) \quad f(u) = \frac{4u^2}{4u^2 + (1-u)^2},$$

and the initial condition is given as

$$(4.7) \quad u(x, 0) = u_l, \quad x \leq 0; \quad u(x, 0) = u_r, \quad x > 0.$$

We consider two kinds of initial conditions as follows:

- (i) $u_l = 2, \quad u_r = -2;$
- (ii) $u_l = -3, \quad u_r = 3.$

We take the computational domain $\Omega = (-4, 4)$, and the final time $T = 10$. We take three kinds of entropy functions as follows:

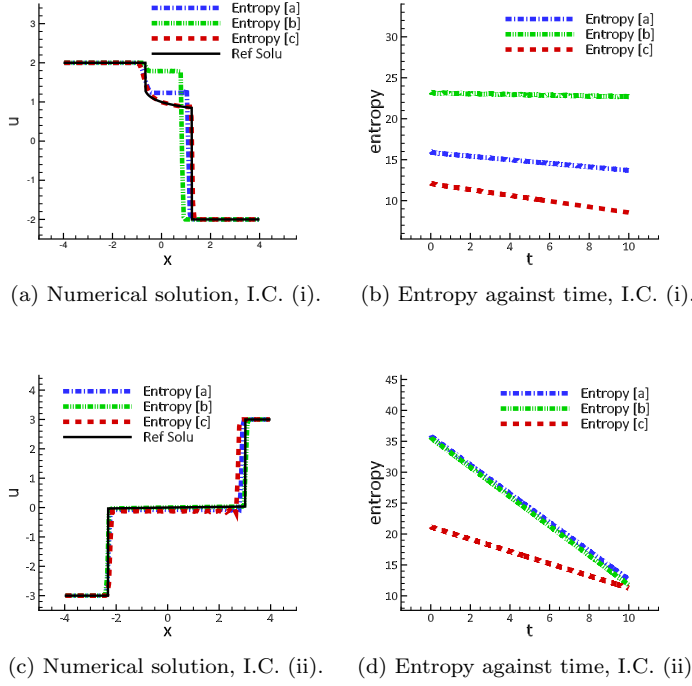
[a] $U(u) = \frac{1}{2}u^2;$

[b] $U(u) = u \arctan(20u) - \frac{1}{40} \log(1 + 400u^2);$

[c] $U(u) = (u - 1) \arctan(u - 1) - \frac{1}{2} \log(u^2 - 2u + 2).$

The first entropy function is the standard square entropy, and the second one is a mollified version of the Kruzhkov's entropy $U(u) = |u|$ [33] and the third one is a mollified version of the translated Kruzhkov's entropy $U(u) = |u - 1|$. In Fig. 5(a) and (b), we observe that numerical solutions of entropies [a] and [b] do not agree with the reference solution. But the numerical solution of entropy [c] gives satisfactory results thanks to the carefully chosen entropy function. As mentioned before, the entropy [c] is an approximation to the entropy function $U(u) = |u - 1|$, which emphasizes the state near $u = 1$. For the initial condition (ii), the numerical solution obtained by using entropy [b] is better than the other two, see Fig. 5(c) and (d). Similar observations are also mentioned in [9]. This implies the entropy pair should be carefully chosen for the nonconvex hyperbolic conservation laws, otherwise the numerical scheme might generate the entropy violating solutions.

FIG. 5. The numerical solution and entropy in Example 4 with I.C. (i) and (ii), $k = 2, N = 128$.



Example 5. Now let us consider two well-known Riemann problems for one-dimensional

Euler equations. Both of them have the following Riemann type initial conditions:

$$\mathbf{U}(x, 0) = \mathbf{U}_L, \quad x < 0; \quad \mathbf{U}(x, 0) = \mathbf{U}_R, \quad x > 0.$$

The first test case is Sod's problem [45]. The initial conditions are

$$[\rho_L, u_L, p_L]^T = [1, 0, 1]^T, \quad [\rho_R, u_R, p_R]^T = [0.125, 0, 0.1]^T.$$

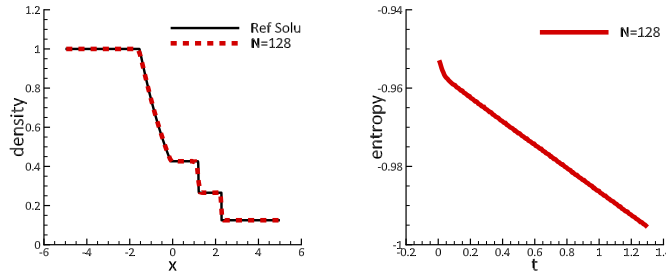
The second one is Lax's problem [34]. The initial conditions are

$$[\rho_L, u_L, p_L]^T = [0.445, 0.698, 3.528]^T, \quad [\rho_R, u_R, p_R]^T = [0.5, 0, 0.571]^T.$$

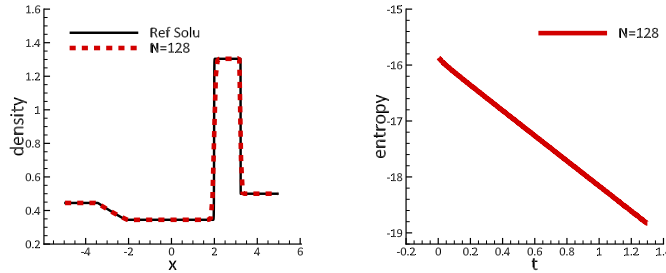
The computational domain $\Omega = (-5, 5)$ and the terminal time $T = 1.3$.

In Example 5, we consider the shock tube problems with one-dimensional compressible Euler systems. The density of the numerical solutions are shown in Fig. 6. The numerical solutions of both Sod's problem and Lax's problem have good performances without obvious oscillations.

FIG. 6. The density profile and entropy for Sod's problem and Lax's problem in Example 5 with the final time $T = 1.3$, $k = 2$, $N = 128$.



(a) Density profile, Sod's problem. (b) Entropy vs. time, Sod's problem.



(c) Density profile, Lax's problem. (d) Entropy vs. time, Lax's problem.

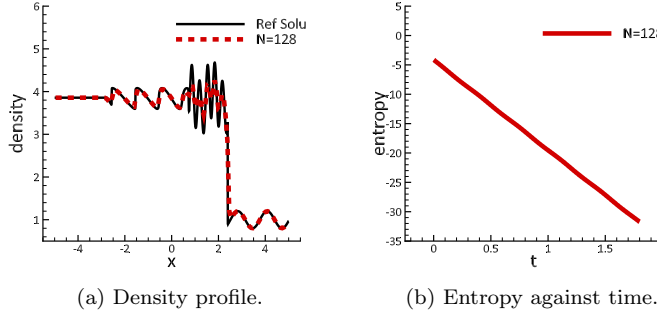
Example 6. Now we consider the Shu-Osher's problem [44]. This example describes the interaction between a right-moving Mach = 3 shock and sine waves in density. It is often used to test high order numerical schemes because both shocks and complicated smooth flow structures co-exist. The initial conditions 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 $T = 1.8$.

The plots of density with 128 cells are displayed in Fig. 7(a) and the discrete entropy is also plotted in Fig. 7(b). We again observe the entropy decreases as time evolves. The ESOFDG scheme has comparable performance to the one in [44] without using any limiters.

FIG. 7. The density profile and entropy for Shu-Osher's problem in Example 6 with the final time $T = 1.8$, $k = 2$, $N = 128$.



Example 7. We consider here the interaction of two blast waves [52]. This problem involves multiple reflections of shocks and rarefaction waves off the walls. The 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 $T = 0.038$.

This example easily generates negative density and negative pressure numerically if no oscillation control procedure is used. The classic DG scheme tends to blow up due to the occurrence of the negative density and negative pressure. The ESOFDG scheme, on the other hand, could proceed without using any limiters. Fig. 8 shows the profile of density at $T = 0.038$ with 800 cells. All shocks and structures are resolved correctly without obvious oscillations.

Example 8. We consider the one-dimensional Sedov point blast problem [54] which models the expanding wave by an intense explosion in the perfect gas. The authors successfully computed this problem by using both the positivity preserving limiter and TVB limiter in [54]. The initial conditions are $\rho(x, 0) = 1$, $u(x, 0) = 0$ and $E(x, 0) = 10^{-12}$ everywhere except $E(x, 0) = E_0/h_0$, $E_0 = 3,200,000$ in the center cell, h_0 is the length of the center cell. The computational domain is $\Omega = (-2, 2)$ and the final time is $T = 10^{-3}$. The formula of the exact solution can be found in [43].

In Fig. 9, we show the profile of density with 128 cells at time $T = 10^{-3}$ for the one-dimensional Sedov point blast problem in Example 8. The ESOFDG scheme gives satisfactory numerical results and we again observe that the discrete entropy decreases with time during the simulation, which indicates the fully discrete scheme is also entropy stable.

FIG. 8. The density profile and entropy for two blast waves problem in Example 7 with the final time $T = 0.038$, $k = 2$, $N = 800$.

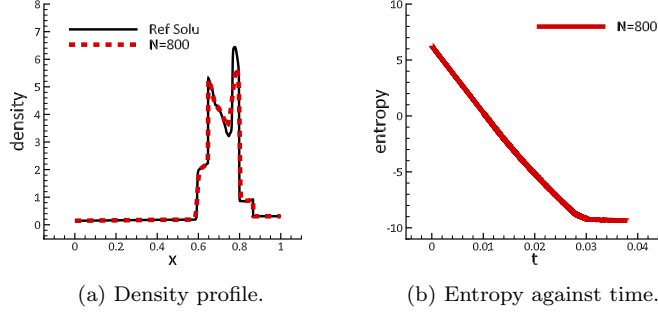
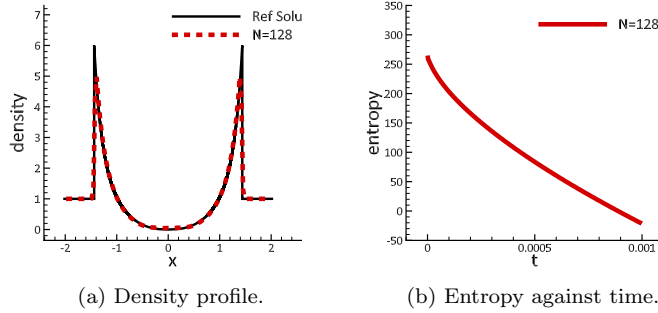


FIG. 9. The density profile and entropy for Sedov point blast problem in Example 8 with the final time $T = 10^{-3}$, $k = 2$, $N = 128$.



4.2. Two-dimensional problems.

Example 9. Consider the two-dimensional linear scalar conservation law

$$u_t + u_x + u_y = 0, \quad (x, y) \in \Omega$$

with periodic boundary condition. We consider two initial conditions in the following.

- (a) The smooth case: $u_0(x, y) = \sin(2x) \cos(2y) + 0.5$, $\Omega = (0, \pi) \times (0, \pi)$. The final time $T = 1.2$.
- (b) The non-smooth case:

$$u_0(x, y) = \begin{cases} 1, & r \leq \frac{1}{8}(3 + 3^{\sin 5\theta}), \\ 0, & \text{elsewhere,} \end{cases}$$

where (r, θ) are the polar coordinates. The computational domain $\Omega = (-1, 1) \times (-1, 1)$. The final time is $T = 1.8$.

We use the entropy function $U(u) = 0.1e^u + 0.45u^2$ for both cases.

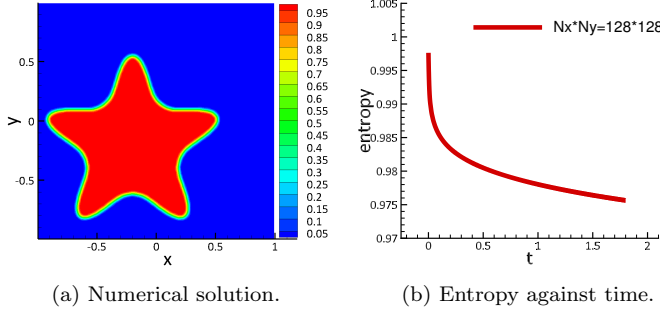
In Table 3, we report the errors and convergence orders of the numerical solution in L^2 norm for $k = 1, 2, 3$ in Example 9. We observe the $(k+1)$ th order of convergence

for all k , which is better than the prediction of truncation error analysis. For non-smooth solutions, we observe the ESOFDG scheme sharply captures the interfaces of the shock in Fig. 10.

TABLE 3
Errors and orders of the case (a) in Example 9 with the final time $T = 1.2$.

	$k = 1$		$k = 2$		$k = 3$	
N	$\ u - u_h\ _{L^2}$	order	$\ u - u_h\ _{L^2}$	order	$\ u - u_h\ _{L^2}$	order
16×16	1.903E-02	—	1.177E-03	—	8.232E-05	—
32×32	4.065E-03	2.227	9.589E-05	3.618	3.106E-06	4.728
64×64	9.671E-04	2.072	9.579E-06	3.323	1.402E-07	4.469
128×128	2.386E-04	2.019	1.108E-06	3.112	7.651E-09	4.196
256×256	5.946E-05	2.005	1.355E-07	3.031	4.589E-10	4.059
512×512	1.485E-05	2.001	1.685E-08	3.008	2.841E-11	4.014

FIG. 10. The numerical solution and entropy of the case (b) in Example 9 with final time $T = 1.8$, $k = 2$, $N_x \times N_y = 128 \times 128$.



Example 10. Consider the following two-dimensional Burgers' equation

$$u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0, \quad (x, y) \in \Omega.$$

with the two cases in the following:

- (a) The initial condition is $u_0(x, y) = \sin(\pi(x + y))$ and periodic boundary condition. The computational domain is $\Omega = (0, 2) \times (0, 2)$, and we take the final time that $T = 0.2$.
- (b) *Riemann problem* [24]. The initial condition is

$$u_0(x, y) = \begin{cases} -\frac{1}{5}, & \text{if } x < \frac{1}{2}, y > \frac{1}{2}, \\ -1, & \text{if } x \geq \frac{1}{2}, y > \frac{1}{2}, \\ \frac{1}{2}, & \text{if } x < \frac{1}{2}, y \leq \frac{1}{2}, \\ \frac{4}{5}, & \text{if } x \geq \frac{1}{2}, y \leq \frac{1}{2}. \end{cases}$$

The exact solution for $t > 0$ can be found in [24]. The computational domain is $\Omega = (0, 1) \times (0, 1)$ and the final time is $T = 0.5$.

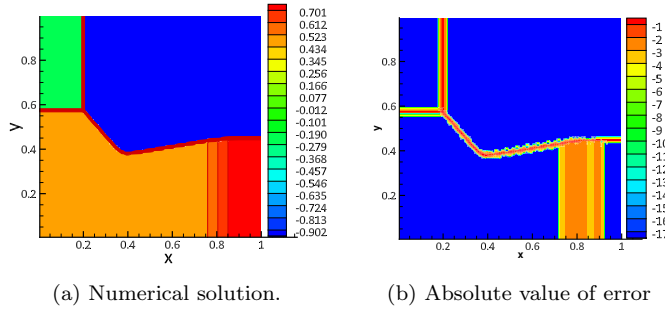
We take $U(u) = \cosh(u)$ as the entropy function for both cases.

For the case (a) in Example 10, the errors and orders are shown in Table 4 and the convergence rates are optimal at least for $k = 1, 2$. The numerical solution of case (b) in Example 10 is displayed in the left panel of Fig. 11, and the absolute value error is plotted in the right panel where it uses the base 10 logarithmic scale. We can see the numerical error is small away from the shock waves.

TABLE 4
Errors and orders of the case (a) in Example 10 with the final time $T = 0.2$.

N	$k = 1$		$k = 2$		$k = 3$	
	$\ u - u_h\ _{L^2}$	order	$\ u - u_h\ _{L^2}$	order	$\ u - u_h\ _{L^2}$	order
16×16	1.462E-02	—	2.319E-03	—	4.225E-04	—
32×32	3.723E-03	1.973	3.161E-04	2.875	3.199E-05	3.723
64×64	9.336E-04	1.996	4.090E-05	2.950	2.568E-06	3.639
128×128	2.341E-04	1.996	5.305E-06	2.947	2.347E-07	3.452
256×256	5.900E-05	1.988	7.063E-07	2.909	2.229E-08	3.397
512×512	1.500E-05	1.975	9.575E-08	2.883	2.092E-09	3.413

FIG. 11. The numerical solution and entropy of the case (b) in Example 10 with final time $T = 0.5$, $k = 2$, $N_x \times N_y = 128 \times 128$. Error is shown in the base 10 logarithmic scale.



Example 11. We now test the Riemann problem proposed in [32] which is very challenging to many high-order numerical schemes for the reason that the solution has a two-dimensional composite wave structure. We have the flux functions as

$$(4.8) \quad \mathbf{f}(u) = [\sin u, \cos u]^T,$$

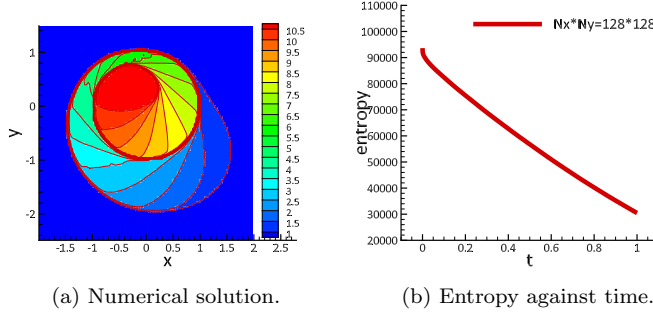
and the initial condition is given as

$$(4.9) \quad u(x, y, 0) = \begin{cases} \frac{7\pi}{2}, & x^2 + y^2 < \frac{1}{2}, \\ \frac{\pi}{4}, & \text{otherwise.} \end{cases}$$

The computational domain is $\Omega = (-2, 2) \times (-2.5, 1.5)$, and the final time is $T = 1$. The entropy function is $U(u) = \cosh(u)$.

The numerical result of Example 11 is shown in Fig. 12. With this suitable entropy function, the ESOFDG scheme provides satisfactory results. We again observe that the discrete entropy monotonically decays which indicates the fully discrete scheme is entropy stable.

FIG. 12. The numerical solution and entropy in Example 11 with final time $T = 1$, $k = 2$, $N_x \times N_y = 128 \times 128$.



Example 12. In this example, we consider the shock vortex interactions in two dimensions [5, 31]. A stationary Mach 1.1 shock is positioned at $x = 0.5$, perpendicular to the x -axis. Its left state is $[\rho, u, v, p]^T = [1, 1.1\sqrt{\gamma}, 0, 1]^T$. 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:

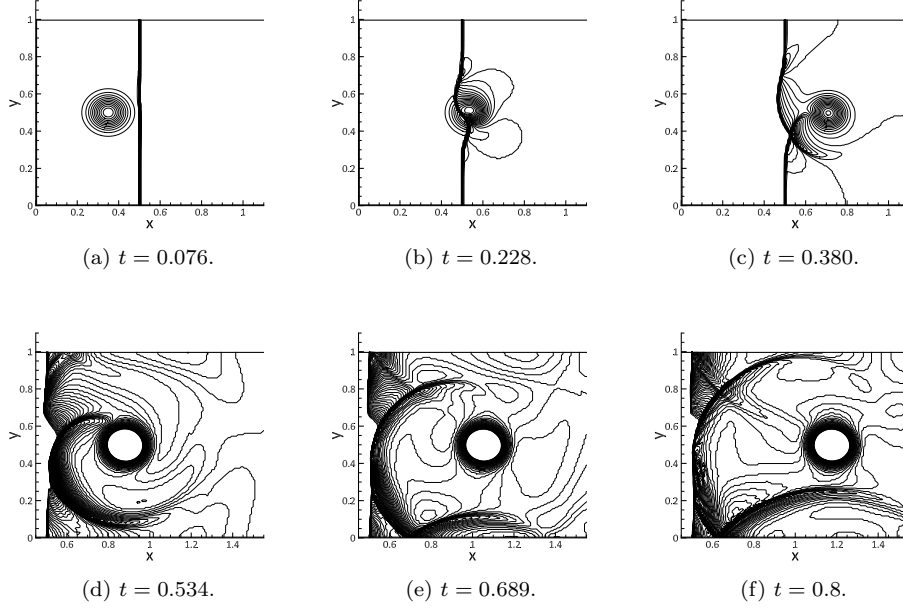
$$(4.10) \quad [\delta u, \delta v]^T = \frac{\varepsilon}{r_c} e^{\alpha(1-\tau^2)} [-\bar{y}, \bar{x}]^T, \quad \delta T = -\frac{(\gamma-1)\varepsilon^2}{4\alpha\gamma} e^{2\alpha(1-\tau^2)}, \quad \delta S = 0,$$

where $[\bar{x}, \bar{y}]^T = [x - x_c, y - y_c]^T$, $r = (\bar{x}^2 + \bar{y}^2)^{1/2}$ and $\tau = r/r_c$. The parameters are taken in the same way as in [31] that $[x_c, y_c]^T = [0.25, 0.5]^T$, $\varepsilon = 0.3$, $r_c = 0.05$ and $\alpha = 0.204$. The computational domain is taken as $(0, 2) \times (0, 1)$ and 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 Fig. 13, we plot the vortex interacting with the stationary shock wave at different time. Since we change the direction of the perturbation of the velocity compared with that in [31], we can see the solution at $t = 0.8$ in which one branch of the shock bifurcations has reached the bottom boundary and has been reflected. The ESOFDG scheme captures the reflection well and the results are comparable to those in [31].

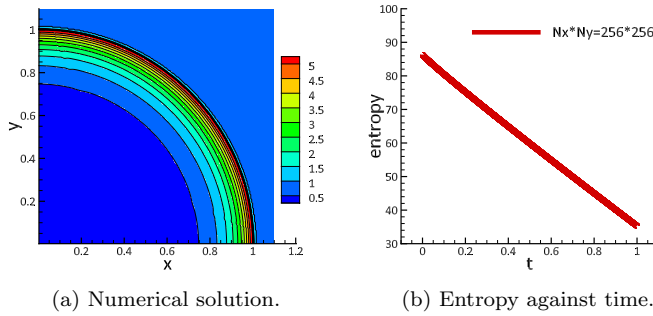
Example 13. Now let us consider the two-dimensional Sedov point blast problem [54]. The initial states are $\rho(x, y, 0) = 1$, $u(x, y, 0) = v(x, y, 0) = 0$ and $E(x, y, 0) = 10^{-12}$ everywhere except $E(x, 0) = E_0/S_0$ in the lower left corner cell, where $E_0 = 0.244816$ and S_0 is the area of the lower left corner cell. The computational domain is $\Omega = (0, 1.1) \times (0, 1.1)$ and the final time is $T = 1$. The numerical boundary treatment is to extend the DG solutions of ρ, v, E as even functions and u as an odd function with respect to the left boundary, and extend the DG solutions of ρ, u, E as even functions and v as an odd function with respect to the bottom boundary. The formula of the exact solution can also be found in [43].

FIG. 13. Pressure contours of the two-dimensional shock vortex interaction problem in Example 12, $k = 2$, $N_x \times N_y = 256 \times 128$. 30 contours: (a) $t = 0.078$; (b) $t = 0.231$; (c) $t = 0.386$. 90 contours from 1.19 to 1.37: (d) $t = 0.544$; (e) $t = 0.701$; (f) $t = 0.8$.



In Fig. 14, we show the density profile of the two-dimensional Sedov point blast problem in Example 13. Same as in the one-dimensional problem, a typical low density would appear along with shock discontinuities. The ESOFDG scheme works well again without using any limiters.

FIG. 14. Density contour of the two-dimensional Sedov point blast problem and entropy against time in Example 13 with final time $T = 1$, $k = 2$, $N_x \times N_y = 256 \times 256$.



Example 14. Now let us consider the double Mach reflection problem [52]. Initially, a Mach 10 shock attacks the horizontal wall with a 60° angle. The reflecting wall lies at the bottom of domain starting from $x = 1/6$. The initial conditions are

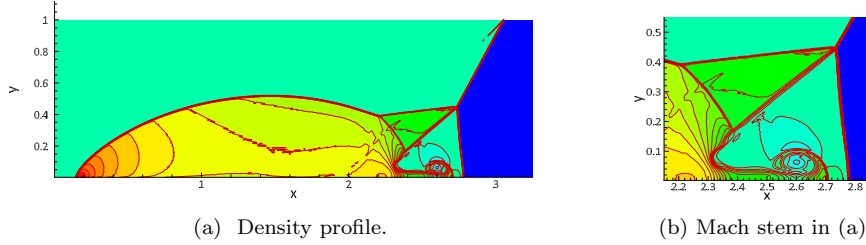
given as follows.

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

The computational domain is $(0, 4) \times (0, 1)$ and the final time is taken to be 0.2. 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.

In Fig. 15, we plot the density contours of the double Mach reflection problem in Example 14 with $k = 2$ on a grid with $h_x = h_y = 1/256$. We can see the flow structure are resolved very clearly, and no instability occurs for our proposed algorithm.

FIG. 15. Density contours of double Mach reflection at $t = 0.2$ in Example 14, 30 contour lines from 1.731 to 20.92, $k = 2$, $h_x = h_y = 1/256$.



Example 15. In the last example, we test the high Mach number astrophysical jets problem [26, 54]. The code could easily blow up since the negative pressure and density could easily appear during numerical computation. Conventionally, a positivity preserving limiter was developed to preserve the positivity of the relevant physical quantities in [53, 54]. Now we compute the high Mach number astrophysical jets without using any positivity preserving limiter. We consider two cases: $Mach = 80$ and $Mach = 2000$ in the following. Note that the heat capacity ratio $\gamma = 5/3$.

- (a) For the Mach 80 problem, the jet initially locates at $y \in (-0.05, 0.05)$, $x = 0$, with the physical values $[\rho, u, v, p]^T = [5, 30, 0, 0.4127]^T$ and the ambient gas is $[\rho, u, v, p]^T = [0.5, 0, 0, 0.4127]^T$. The computational domain is $(0, 2) \times (-0.5, 0.5)$ and the terminal time is 0.07. The boundary conditions of the rest boundaries are outflow.
- (b) For the Mach 2000 problem, the jet initially locates at $y \in (-0.05, 0.05)$, $x = 0$. The physical values of the jet are $[\rho, u, v, p]^T = [5, 800, 0, 0.4127]^T$ and the ambient gas is $[\rho, u, v, p]^T = [0.5, 0, 0, 0.4127]^T$. The computational domain is $(0, 1) \times (-0.25, 0.25)$ and the terminal time is 0.001. The boundary conditions of the rest boundaries are outflow.

In Fig. 16 and 17, we show the density, pressure and temperature contours of Mach 80 and Mach 2000 astrophysical jets in Example 15, $k = 2$, $N_x \times N_y = 320 \times 160$.

The ESOFDG scheme successfully obtain satisfactory results compared to the results in [54] without any occurrence of instability.

FIG. 16. *High Mach astrophysical jets in Example 15, Mach = 80, $k = 2$, $N_x \times N_y = 320 \times 160$. Scales are logarithmic.*

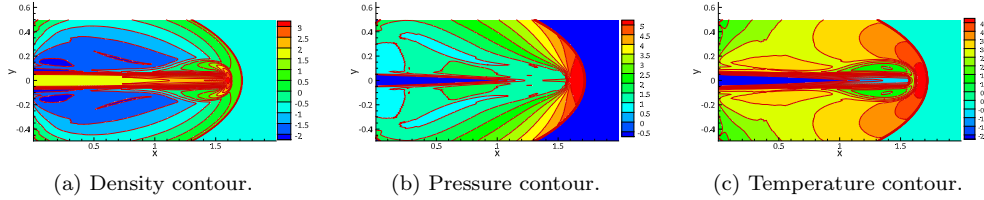
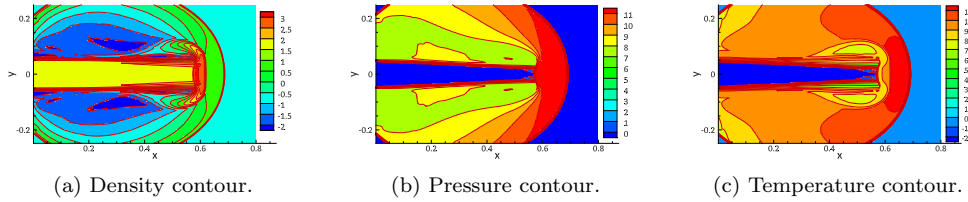


FIG. 17. *High Mach astrophysical jets in Example 15, Mach = 2000, $k = 2$, $N_x \times N_y = 320 \times 160$. Scales are logarithmic.*



5. Concluding remarks. In this paper, we propose an entropy stable **essen-**
tially oscillation-free discontinuous Galerkin method for hyperbolic conservation laws. The entropy stable DG method [9, 10] has attracted much attention from the date of its birth. Several key ingredients like summation-by-parts operators, flux differencing technique and entropy conservative fluxes and entropy stable fluxes are incorporated in the nodal DG formulation. This also brings challenges if one wants to apply the damping technique [36, 35] to the current entropy stable DG framework. Thanks to the convexity of the entropy functions and the orthogonality of the projection, we are able to construct a damping term similar to the original ones. This indicates the constructed scheme would preserve several properties such as conservation, entropy stability and high order accuracy of the entropy stable DG method, in the meantime it can also suppress the spurious oscillations as demonstrated in the numerical tests. We are aware that currently there is no theoretical analysis of the oscillation control mechanism with the damping term, and this is a possible investigation direction of our future study.

REFERENCES

- [1] R. Abgrall. A general framework to construct schemes satisfying additional conservation relations. Application to entropy conservative and entropy dissipative schemes. *J. Comput. Phys.*, 372 (2018), 640 – 666.
- [2] M.W. Böhm, A.R. Winters, G.J. Gassner, D. Derigs, F.J. Hindenlang and J. Saur. An entropy stable nodal discontinuous Galerkin method for the resistive MHD equations. Part I: Theory and numerical verification. *J. Comput. Phys.*, 422 (2020), 108076.

- [3] M.H. Carpenter, T.C. Fisher, E.J. Nielsen and S.H. Frankel. Entropy stable spectral collocation schemes for the Navier-Stokes equations: discontinuous interfaces. *SIAM J. Sci. Comput.*, 36 (2014), B835 – B867.
- [4] M.H. Carpenter, T.C. Fisher, E.J. Nielsen, M. Parsani, M. Svärd and N. Yamaleev. Entropy stable summation-by-parts formulations for compressible computational fluid dynamics. *Handb. Numer. Anal.*, 17 (2016), 495 – 524.
- [5] J. Casper. Finite-volume implementation of high-order essentially nonoscillatory schemes in two dimensions. *AIAA Journal*, 30 (1992), 2829 – 2835.
- [6] P. Chandrashekar. Kinetic energy preserving and entropy stable finite volume schemes for compressible Euler and Navier-Stokes equations. *Commun. Comput. Phys.*, 14 (2013), 1252 – 1286.
- [7] J. Chan. On discretely entropy conservative and entropy stable discontinuous Galerkin methods. *J. Comput. Phys.*, 362 (2018), 346 – 374.
- [8] J. Chan, D.C. Del Rey Fernández and M.H. Carpenter. Efficient entropy stable Gauss collocation methods. *SIAM J. Sci. Comput.*, 41 (2019), A2938 – A2966.
- [9] 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.
- [10] T. Chen and C.-W. Shu. Review of entropy stable discontinuous Galerkin methods for systems of conservation laws on unstructured simplex meshes. *CSIAM Trans. Appl. Math.*, 1 (2020), 1 – 52.
- [11] 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.
- [12] B. Cockburn and C.-W. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.*, 16 (2001), 173 – 261.
- [13] J. Crean, J.E. Hicken, D.C. Del Rey Fernández, D.W. Zingg and M.H. Carpenter. High order, entropy-stable discretizations of the Euler equations for complex geometries. In 23rd AIAA Computational Fluid Dynamics Conference. American Institute of Aeronautics and Astronautics, 2017.
- [14] J. Crean, J.E. Hicken, D.C. Del Rey Fernández, D.W. Zingg and M.H. Carpenter. Entropy stable summation-by-parts discretization of the Euler equations on general curved elements. *J. Comput. Phys.*, 356 (2018), 410 – 438.
- [15] C.M. Dafermos. *Hyperbolic Conservation Laws in Continuum Physics*. Vol. 325, Springer-Verlag, Berlin, 2010.
- [16] D.C. Del Rey Fernández, P.D. Boom and D.W. Zingg. A generalized framework for nodal first derivative summation-by-parts operators. *J. Comput. Phys.*, 266 (2014), 214 – 239.
- [17] D.C. Del Rey Fernández, J.E. Hicken and D.W. Zingg. Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. *Comput. & Fluids*, 95 (2014), 171 – 196.
- [18] D.C. Del Rey Fernández, J.E. Hicken and D.W. Zingg. Simultaneous approximation terms for multi-dimensional summation-by-parts operators. *J. Sci. Comput.*, 75 (2018), 83 – 110.
- [19] U.S. Fjordholm, S. Mishra and E. Tadmor. Arbitrarily high-order accurate entropy stable essentially nonoscillatory schemes for systems of conservation laws. *SIAM J. Numer. Anal.*, 50 (2012), 544 – 573.
- [20] U.S. Fjordholm, S. Mishra and E. Tadmor. ENO reconstruction and ENO interpolation are stable. *Found. Comput. Math.*, 13 (2013), 139 – 159.
- [21] G.J. Gassner, A.R. Winters and D.A. Kopriva. A well balanced and entropy conservative discontinuous Galerkin spectral element method for the shallow water equations. *Appl. Math. Comput.*, 272 (2016), 291 – 308.
- [22] G.J. Gassner, A.R. Winters, F.J. Hindenlang and D.A. Kopriva. The BR1 scheme is stable for the compressible Navier-Stokes equations. *J. Sci. Comput.*, 77 (2018), 154 – 200.
- [23] E. Godlewski and P.-A. Raviart. *Hyperbolic systems of conservation laws*. Ellipses, 1991.
- [24] J.-L. Guermond, R. Pasquetti and B. Popov. Entropy viscosity method for nonlinear conservation laws. *J. Comput. Phys.*, 230 (2011), 4248 – 4267.
- [25] J.-L. Guermond, B. Popov, Fast estimation from above of the maximum wave speed in the Riemann problem for the Euler equations, *J. Comput. Phys.*, 321 (2016), 908–926.
- [26] 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.
- [27] J.E. Hicken, D.C. Del Rey Fernández and D.W. Zingg. Multidimensional summation-by-parts operators: general theory and application to simplex elements. *SIAM J. Sci. Comput.*, 38 (2016), A1935 – A1958.

- [28] J.E. Hicken and D.W. Zingg. A parallel Newton-Krylov solver for the Euler equations discretized using simultaneous approximation terms. *AIAA J.*, 46 (2008), 2773 – 2786
- [29] A. Hildebrand and S. Mishra. Entropy stable shock capturing space time discontinuous Galerkin schemes for systems of conservation laws. *Numer. Math.*, 126 (2014), 103 – 151.
- [30] F. Ismail and P.L. Roe. Affordable, entropy-consistent Euler flux functions II: Entropy production at shocks. *J. Comput. Phys.*, 228 (2009), 5410 – 5436.
- [31] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126 (1996), 202 – 228.
- [32] A. Kurganov, G. Petrova and B. Popov. Adaptive semi-discrete central-upwind schemes for nonconvex hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 29 (2007), 2381 – 2401.
- [33] S.N. Kruzhkov. First order quasilinear equations in several independent variables. *Mat. Sb.*, 123 (1970), 228 – 255.
- [34] P.D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Comm. Pure Appl. Math.*, 7 (1954), 159 – 193.
- [35] Y. Liu, J. Lu and C.-W. Shu. An essentially oscillation-free discontinuous Galerkin method for hyperbolic systems. *SIAM J. Sci. Comput.*, 44 (2022), A230 – A259.
- [36] 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.
- [37] P.G. Lefloch, J.-M. Mercier and C. Rohde. Fully discrete, entropy conservative schemes of arbitrary order. *SIAM J. Numer. Anal.*, 40 (2002), 1968 – 1992.
- [38] J. Nordström, J. Gong, E. van der Weide and M. Svård. A stable and conservative high order multi-block method for the compressible Navier-Stokes equations. *J. Comput. Phys.*, 228 (2009), 9020 – 9035.
- [39] P. Öffner, J. Glaubitz and H. Ranocha. Stability of correction procedure via reconstruction with summation-by-parts operators for Burgers equation using a polynomial chaos approach. *ESAIM: M2AN*, 52 (2018), 2215 – 2245.
- [40] H. Ranocha. Shallow water equations: split-form, entropy stable, well-balanced, and positive preserving numerical methods. *GEM-Int. J. Geomath*, 8 (2017), 85 – 133.
- [41] F. Renac. Entropy stable DGSEM for nonlinear hyperbolic systems in nonconservative form with application to two-phase flows. *J. Comput. Phys.*, 382 (2019), 1 – 26.
- [42] J.-X. Qiu, and C.-W. Shu. Runge-Kutta discontinuous Galerkin method using WENO limiters. *SIAM J. Sci. Comput.*, 26 (2005), 907 – 929.
- [43] L.I. Sedov. Similarity and Dimensional Methods in Mechanics. Academic Press, New York, 1959.
- [44] C.-W. Shu and S. Osher. Efficient implementation of essentially nonoscillatory shock-capturing schemes. II. *J. Comput. Phys.*, 83 (1989), 32 – 78.
- [45] G.A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.*, 27 (1978), 1 – 31.
- [46] Z. Sun, J.A. Carillo and C.-W. Shu. An entropy stable high-order discontinuous Galerkin method for cross-diffusion gradient flow systems. *Kinet. Relat. Mod.*, 12 (2019), 885 – 908.
- [47] M. Svård and J. Nordström. Review of summation-by-parts schemes for initial-boundary-value problems. *J. Sci. Comput.*, 58 (2014), 61 – 89.
- [48] M. Svård, K. Mattsson and J. Nordström. Steady-state computations using summation-by-parts operators. *J. Sci. Comput.*, 24 (2005), 79 – 95.
- [49] E. Tadmor. The numerical viscosity of entropy stable schemes for systems of conservation laws. I. *Math. Comp.*, 49 (1987), 91 – 103.
- [50] E. Tadmor. Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems. *Acta Numer.*, 12 (2003), 451 – 512.
- [51] N. Wintermeyer, A.R. Winters, G.J. Gassner and D.A. Kopriva. An entropy stable nodal discontinuous Galerkin method for the two dimensional shallow water equations on unstructured curvilinear meshes with discontinuous bathymetry. *J. Comput. Phys.*, 340 (2017), 200 – 242.
- [52] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, 54 (1984), 115 – 173.
- [53] X. Zhang and C.-W. Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *J. Comput. Phys.*, 229 (2010), 3091 – 3120.
- [54] 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.
- [55] 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.