A comparative study of shock capturing models for the discontinuous Galerkin method

Jian Yu[a,*], Jan S Hesthaven[b]

[a] School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, China
[b] Chair of Computational Mathematics and Simulation Schience, École Polytechnique Fédérale de Lausanne, Lausanne CH-1015, Switzerland

Abstract

Dealing with strong shocks while retaining low numerical dissipation has been one of the major challenges for high order methods like discontinuous Galerkin. In the literature, various shock capturing models have been designed based on different approaches, providing the motivation of the present work in which we compare several typical shock capturing models in terms of accuracy and robustness. The selected models consist of a derivative-based model, a highest modal decay model, an averaged modal decay model, an entropy viscosity model, and a weighted essentially non-oscillatory(WENO) limiting method. The performance for both smooth and non-smooth problems are examined with typical one- and two-dimensional cases. For smooth flows, the WENO and entropy viscosity methods can preserve the theoretical order of accuracy with the shock sensor activated, while both modal decay models rely on entirely switching off the shock sensors on sufficiently fine grids to recover the original accuracy. The derivative-based model is limited to second order accuracy in the neighborhood of smooth regions. For discontinuous problems, the results indicate that for lower orders, the WENO limiter usually generates the least dissipative results, while the viscosity models perform better for high orders. The derivative-based model is able to capture shocks and small scales with good resolution. The highest modal decay model suffers from the inaccurate estimation of the modal decay rate, which may give unusually dissipative results. The averaged modal decay model works well in terms of both accuracy and robustness, although oscillations may appear at the element level. The entropy viscosity model is observed to emphasize contact discontinuities more than the other viscosity models.

Keywords: High order methods, Discontinuous Galerkin, Shock capturing, Limiting, Artificial viscosity

1. Introduction

Computational fluid dynamics(CFD) is increasingly involved in the design process of modern aircrafts. High order methods are essential for various problems, such as turbulence flows, acoustic prediction, and long time convection of vortices from a wingtip[1], all of which pose a stringent limit on the error level of the simulation, and for which high order methods have been proved to be more efficient than their second order counterpart[2]. The last decades have seen an explosive growth in the amount of research activity in high order methods, among which discontinuous Galerkin(DG) [3] has been considered to have great potential in terms of accuracy, geometrical flexibility, h-p adaptivity, parallel efficiency, and so on.

One of the major challenges for high order methods is to handle strong compressibility(i.e. shocks) with a reasonable balance between accuracy and robustness. For methods like DG, various types of shock capturing methods have been proposed in the literature. The first approach is slope limiting, such as in [4, 5]. Despite its total variation bounded(TVB) property, the limiter shows a detrimental effect on accuracy especially for higher order accuracy, and depends strongly on emprical parameters. Improvements on these issues have been proposed[6, 7]. However, it remains difficult to retain high order accuracy with limiting. An alternative strategy is a limiter based on the weighted

*Corresponding author
Email addresses: yuj@buaa.edu.cn (Jian Yu), Jan.Hesthaven@epfl.ch (Jan S Hesthaven)

essentially non-oscillatory (WENO) reconstruction, which first identifies cells around discontinuities as troubled cells, and the WENO reconstruction is then performed. The first such ideas were proposed in [8, 9], which use only the cell averages for reconstruction and therefore require quite large stencils. In order to relieve this issue, Hermite WENO(HWENO), using both the cell average and its first derivatives was proposed in [10], and extended to two-dimensional cases in [11, 12]. However, the HWENO methods require a stencil of more than just the von Neumann neighborhood for higher order accuracy. [13] improves the HWENO method to use also the derivatives of higher than first order to reduce the stencil to be the neighborhood. However, only results for the second order case have been demonstrated. To reduce the stencil, the key is to make use of as much information as possible in each cell. Following this line, [14, 15] proposes a simple yet effective WENO limiting method which reconstructs the entire polynomial instead of just pointwise values or moments. This not only results in a stencil of the von Neumann neighborhood, but also eliminates the issue of negative weights.

Another family of methods for capturing discontinuities results from adding explicit dissipation to the original system. The key for these methods is to develop sensors which measure the smoothness of the flow field and determine the amount of required artificial dissipation. One idea is to employ derived quantities as the shock sensor, as was intiated in the context of finite difference methods in [16, 17], and later improved in [18, 19, 20]. The method tackles vortices, shocks, and contact discontinuities with artificial shear viscosity, artificial bulk viscosity, and artificial thermal conductivity, respectively. Further, [21] extends the method to unstructured grids with the spectral difference(SD) method, and [22] simplifies the method based on a scaling function of velocity dilation under the hybridizable discontinuous Galerkin(HDG) scheme. Such methods are quite robust and can handle complicated flows with great flexibility since they design specific sensors for different flow features. However, there is a general dilemma of choosing the order of the derived quantities. For higher derivatives, accuracy in smooth regions are better, while the complexity and cost for computing the derivatives are overwhelming especially for high order methods of unstructrued grids. A second idea is to explore the decay of the modal coefficients, as first proposed in [23]. The method is based on the assumption that the modal coefficients should decay similarly rapidly for smooth flows. Despite its reasonable success, the shock sensor is often too simple to make a robust method. Consequently, [24] addresses these issues, and proposes a more accurate and robust model following an alternative approach to analysing the expansion coeffcients. A third approach is the entropy-based viscosity method proposed in [25], which generates a artificial viscosity proportional to the local entropy production, and aims to be grid and approximation independent. [26] discusses implementation details in the context of DG. Furthermore, [27] extends the entropy method to viscous cases to reduce dissipation around regions where viscous effects are significant. There are also approaches which base the shock sensor on the cell residual[28] or the jump[29] along the faces.

Based on the above discussion, one can observe that the shock capturing models are based on very different approaches. It would be beneficial to compare them in the same context. Consequently, the primary motivation of this work is to consider a number of benchmarks, and compare the performance in terms of accuracy and robustness. We focus on a few techniques, including a derivative-based model, the highest modal decay model[23], the averaged modal decay model[24], the entropy viscosity model[26], and the WENO limiting method[14, 15] as a reference. The paper is organized as follows. In Section 2, the governing equations as well as the discontinuous Galerkin formulation employed in this work is described. In Section 3, the different shock capturing models are discussed. In Section 4, the numerical experiments are discussed, followed by concluding remarks in Section 5.

2. Numerical discretization

2.1. Governing equation

The conservation law is considered which takes the following form

$$\frac{\partial \boldsymbol{q}}{\partial t} + \nabla \cdot \boldsymbol{f} - \nabla \cdot \boldsymbol{g} = 0, \tag{2.1}$$

where $\boldsymbol{q}$ is the conserved variables, $\boldsymbol{f}$ is the convective flux, and $\boldsymbol{g}$ is the viscous flux. In this work, $\boldsymbol{g}$ only serves as the artificial dissipation term, and takes the Laplacian form

$$\boldsymbol{g} = \mu_{av}\boldsymbol{w}, \quad \boldsymbol{w} = \nabla \boldsymbol{q}, \tag{2.2}$$

where μ_{av} is the artificial viscosity, that needs to be computed as described in Section 3.

In this work, we will consider the one-dimensional linear transport equation, the one-dimensional Burgers' equation, and the one- and two-dimensional Euler equations, which will be introduced in the following.

For the one-dimensional linear transport equation, we have

$$f = q. \tag{2.3}$$

For the one-dimensional Burgers' equation, we have

$$f = q^2. \tag{2.4}$$

For the one-dimensional Euler system, we have

$$\boldsymbol{q} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}, \qquad \boldsymbol{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{pmatrix}, \tag{2.5}$$

where ρ is the density, u is the velocity in the x direction, E is the total energy, and p is the pressure with $p = (\gamma - 1)\left(E - \frac{1}{2}\rho u^2\right)$. $\gamma = 1.4$ is used in this work.

For the two-dimensional Euler system, we have

$$\boldsymbol{q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \qquad \boldsymbol{f} = \boldsymbol{f}^x \vec{\boldsymbol{i}} + \boldsymbol{f}^y \vec{\boldsymbol{j}}, \tag{2.6}$$

where

$$\boldsymbol{f}^x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix}, \qquad \boldsymbol{f}^y = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}, \tag{2.7}$$

and v is the velocity in the y direction. In this case, $p = (\gamma - 1)\left(E - \frac{1}{2}\rho\left(u^2 + v^2\right)\right)$.

2.2. *Formulations of the nodal discontinuous Galerkin method*

In this paper, we describe the nodal discontinuous Galerkin method[3] used in this work. For simplicity, we will only consider the scalar case here. The extension to the system case is straight-forward.

2.2.1. *The local representation in 1D*

First, each element is mapped to the standard element $I^{1\mathrm{D}} = [-1, 1]$ through the affine mapping

$$x(r) = x_{v_0} + \frac{1+r}{2}h, \quad h = x_{v_1} - x_{v_0}, \quad r \in [-1, 1], \tag{2.8}$$

where the subscripts v_0 and v_1 denote the left and right end of each element. With Eq. (2.8), the inverse map $r(x)$ follows immediately.

The nodal representation of the approximate solution in $I^{1\mathrm{D}}$ is given as

$$q_h(r,t) = \sum_{m=0}^{N_P-1} q_{p,m} l_m(r), \quad q_{p,m} \equiv q_h(r_m, t), \tag{2.9}$$

where $N_P = P + 1$, P is the highest order of the approximation polynomials, $q_{p,m}$ is the unknown value at the position $r = r_m$, and $l_m(r)$ is the Lagrange polynomial of P-th order which interpolates at the nodal points. The subscript p denotes values at nodal points. The Legendre-Gauss-Lobatto points are chosen as the nodal points.

The modal representation of the approximate solution will also be frequently needed. This can be written as

$$q_h(r,t) = \sum_{m=0}^{N_P-1} \hat{q}_m \varphi_m(r), \tag{2.10}$$

where $\varphi_m(r)$ is an orthonormal basis, and is chosen to be

$$\varphi_m(r) = \frac{Q_m^{(0,0)}(r)}{\sqrt{\chi_m}}, \qquad \chi_m = \frac{2}{2m+1}, \tag{2.11}$$

where $Q_m^{(\alpha,\beta)}(r)$ is the Jacobi polynomial of order m, and χ_m is the normalization. Note that $\alpha = 0, \beta = 0$ indicates the special case of the classical Legendre polynomial.

The nodal expression can be obtained from the modal expression through the Vandermonde matrix $\boldsymbol{V}$

$$\boldsymbol{q}_p = \boldsymbol{V}\hat{\boldsymbol{q}}, \quad V_{m,n} = \varphi_n(r_m), \quad (m,n) \in [0, N_P - 1]. \tag{2.12}$$

2.2.2. The local representation in 2D

The local representation in 2D is essentially the same as in 1D. In this work, only triangular elements (shown in Fig. 1) will be considered. The affine mapping for the triangular element is given as

$$\boldsymbol{x}(r,s) = -\frac{r+s}{2}\boldsymbol{v}_0 + \frac{r+1}{2}\boldsymbol{v}_1 + \frac{s+1}{2}\boldsymbol{v}_2, \tag{2.13}$$

where $\boldsymbol{v}_0$, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ denote the vectors of the three vertices for the standard triangular element.

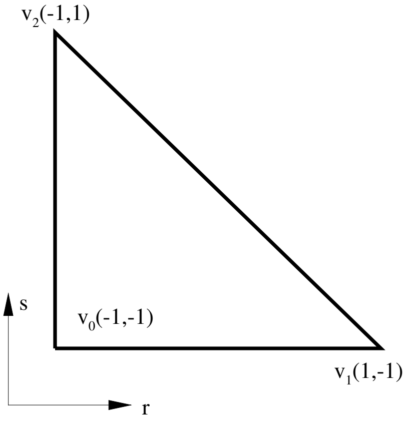


Fig. 1. Illustation of the standard triangular element

The nodal representation for 2D is given as

$$q_h(r,s,t) = \sum_{m=0}^{N_P-1} q_{p,m}(t) l_m(r,s), \quad q_{p,m}(t) \equiv q_h(r_m, s_m, t), \tag{2.14}$$

where $N_P = (P+1)(P+2)/2$. The nodal points (r_m, s_m) are chosen to be the α-optimized nodal set[3]. Note that this choice recovers the one-dimensional Legendre-Gauss-Lobatto points along the edges of the triangular element.

The modal representation for 2D is given as

$$q_h(r,s,t) = \sum_{m=0}^{N_P-1} \hat{q}_m(t) \varphi_m(r,s), \tag{2.15}$$

where the orthonormal basis function is

$$\varphi_m(r,s) = \sqrt{2} Q_i^{(0,0)}(a) Q_j^{(2i+1,0)}(b)(1-b)^i, \quad a = a(r,s), \quad b = b(r,s). \tag{2.16}$$

Here m is related to (i,j) as

$$m = j + (P+1)i - \frac{1}{2}i(i-1), \quad (i,j) \geq 0, \quad i+j \leq P. \tag{2.17}$$

Also, (a,b) denotes the collapsed coordinate system of a rectangular element transformed from the standard triangle with

$$a = 2\frac{1+r}{1-s} - 1, \qquad b = s. \tag{2.18}$$

The transformation Eq. (2.12) still holds for 2D, except that now $V_{m,n} = \varphi_n(r_m, s_m)$.

2.2.3. Discontinuous Galerkin approximation

In this section, we describe the nodal DG method for 2D. Its 1D counterpart can then be obtained in the same manner. Similar to q, we can also define the nodal approximations for $\boldsymbol{f}$, $\boldsymbol{g}$ and $\boldsymbol{w}$ as

$$\boldsymbol{f}_h(r,s,t) = \sum_{m=0}^{N_P-1} \boldsymbol{f}_{p,m} l_m(r,s), \quad \boldsymbol{f}_{p,m} \equiv \boldsymbol{f}_h(r_m, s_m, t), \tag{2.19}$$

$$\boldsymbol{g}_h(r,s,t) = \sum_{m=0}^{N_P-1} \boldsymbol{g}_{p,m} l_m(r,s), \quad \boldsymbol{g}_{p,m} \equiv \boldsymbol{g}_h(r_m, s_m, t), \tag{2.20}$$

$$\boldsymbol{w}_h(r,s,t) = \sum_{m=0}^{N_P-1} \boldsymbol{w}_{p,m} l_m(r,s), \quad \boldsymbol{w}_{p,m} \equiv \boldsymbol{w}_h(r_m, s_m, t). \tag{2.21}$$

Subsititute the above approximate representations into Eqs. (2.1) and (2.2), and recover

$$R_{h,1} = \frac{\partial q_h}{\partial t} + \nabla \cdot \boldsymbol{f}_h - \nabla \cdot \boldsymbol{g}_h, \tag{2.22}$$

$$\boldsymbol{R}_{h,2} = \boldsymbol{w}_h - \nabla q_h. \tag{2.23}$$

For the nodal DG approximation, the residuals $R_{h,1}$ and $\boldsymbol{R}_{h,2}$ are required to be orthogonal to the Lagrange basis within element K as

$$\int_K R_{h,1} l_m dK = 0, \quad \int_K \boldsymbol{R}_{h,2} l_m dK = 0, \quad m = 0, ..., N_P - 1. \tag{2.24}$$

After integration by parts, the weak form is then obtained as

$$\int_K \left(\frac{\partial q_h}{\partial t} l_m - \boldsymbol{f}_h \cdot \nabla l_m + \boldsymbol{g}_h \cdot \nabla l_m \right) dK = -\oint_{\partial K} (\boldsymbol{n} \cdot \boldsymbol{f}^* l_m - \boldsymbol{n} \cdot \boldsymbol{g}^* l_m) d\sigma, \tag{2.25}$$

$$\int_K \boldsymbol{w}_h l_m dK + \int_K q \nabla l_m dK = \oint_{\partial K} \boldsymbol{n} q^* l_m d\sigma, \tag{2.26}$$

where $\boldsymbol{f}^*$, $\boldsymbol{g}^*$, and $\boldsymbol{q}^*$ denote the numerical flux at the element interface. For $\boldsymbol{g}^*$ and q^*, the central scheme is employed, while for $\boldsymbol{f}^*$, the Lax-Friedrich scheme is used, i.e.

$$\boldsymbol{f}^* = \{\{\boldsymbol{f}_h\}\} + \frac{\lambda}{2}[\![q]\!], \tag{2.27}$$

where λ is the locally maximum wave speed.

Then, we define the mass matirx $\boldsymbol{M}$ and the stiffness matrices $\boldsymbol{S}^r$, $\boldsymbol{S}^s$ as

$$M_{m,n} = \int_K l_m l_n dK, \quad S^{\,r}_{m,n} = \int_K l_m \frac{\partial l_n}{\partial r} dK, \quad S^{\,s}_{m,n} = \int_K l_m \frac{\partial l_n}{\partial s} dK. \tag{2.28}$$

For practical computation, we have

$$\boldsymbol{M} = (\boldsymbol{V}\boldsymbol{V}^T)^{-1}, \quad \boldsymbol{S}^r = \boldsymbol{M}\boldsymbol{D}^r, \quad \boldsymbol{S}^s = \boldsymbol{M}\boldsymbol{D}^s, \tag{2.29}$$

$$\boldsymbol{D}^r = \boldsymbol{V}^r\boldsymbol{V}^{-1}, \quad \boldsymbol{D}^s = \boldsymbol{V}^s\boldsymbol{V}^{-1}, \tag{2.30}$$

$$V^r_{m,n} = \frac{\partial \varphi_n}{\partial r}|_{r_m,s_m}, \quad V^s_{m,n} = \frac{\partial \varphi_n}{\partial s}|_{r_m,s_m}. \tag{2.31}$$

The stiffness matrices in the physical coordinate system is recovered as

$$\boldsymbol{S}^x = \frac{\partial r}{\partial x}\boldsymbol{S}^r + \frac{\partial s}{\partial x}\boldsymbol{S}^s, \quad \boldsymbol{S}^y = \frac{\partial r}{\partial y}\boldsymbol{S}^r + \frac{\partial s}{\partial y}\boldsymbol{S}^s. \tag{2.32}$$

The DG approximation can then be written as

$$J\boldsymbol{M}\frac{d\boldsymbol{q}_p}{dt} - (\boldsymbol{S}^x)^T\boldsymbol{f}^x_{\boldsymbol{p}} - (\boldsymbol{S}^y)^T\boldsymbol{f}^y_{\boldsymbol{p}} + (\boldsymbol{S}^x)^T\boldsymbol{g}^x_p + (\boldsymbol{S}^y)^T\boldsymbol{g}^y_p = -\sum_e J^\sigma_e \boldsymbol{M}^\sigma_e \boldsymbol{n}_e \cdot \boldsymbol{f}^*_p + \sum_e J^\sigma_e \boldsymbol{M}^\sigma_e \boldsymbol{n}_e \cdot \boldsymbol{g}^*_p, \tag{2.33}$$

$$J\boldsymbol{M}\boldsymbol{w}_p + (\boldsymbol{S}^x)^T\boldsymbol{q}_p\vec{\boldsymbol{i}} + (\boldsymbol{S}^y)^T\boldsymbol{q}_p\vec{\boldsymbol{j}} = \sum_e J^\sigma_e \boldsymbol{M}^\sigma_e \boldsymbol{n}_e \boldsymbol{q}^*_p, \tag{2.34}$$

where J is the transformation Jacobian obtained from Eq. (2.13), and is constant for straight-sided triangular elements. J^σ_e and $\boldsymbol{M}^\sigma_e$ are the transformation Jacobian and mass matrix, respectively, along the eth edge. $\boldsymbol{M}^\sigma_e$ is a $N_p \times (P+1)$ matrix, the entry of which is given as

$$M^\sigma_{e,(m,n)} = \int_{\sigma_e} l_m l_n d\sigma. \tag{2.35}$$

Note that l_m is a Lagrange polynomial of order P, and its value is zero on points which do not reside on the e-th edge. Therefore, $\boldsymbol{M}^\sigma_e$ is not a full matrix, and only has nonzero entries in those rows, m, where $\boldsymbol{x}_m$ resides on the edge.

2.3. Boundary condition for viscous terms

In this section, we describe boundary conditions when viscous terms are included. For the primitive value and its derivatives within each field, the principle is to specify explicitly one value, and all the others from the same field are extrapolated. We take the inviscid wall of the 2D Euler system as an example to explain the idea in more details, and other conditions follow in a similar manner. First, for boundary values of the primitive variables, we have

$$\left.\begin{array}{ll} \rho_{\mathrm{B}} = \rho_{\mathrm{M}}, & p_{\mathrm{B}} = p_{\mathrm{M}}, \\ (\boldsymbol{n}\cdot\boldsymbol{u})_{\mathrm{B}} = -(\boldsymbol{n}\cdot\boldsymbol{u})_{\mathrm{M}}, & (\boldsymbol{\tau}\cdot\boldsymbol{u})_{\mathrm{B}} = (\boldsymbol{\tau}\cdot\boldsymbol{u})_{\mathrm{M}}, \end{array}\right\} \tag{2.36}$$

where the subscript ‘B’ denotes the boundary value and ‘M’ the interior value along the boundary, $\boldsymbol{n}$ and $\boldsymbol{\tau}$ are the unit vectors along the normal and tangential directions of the element edge. Then, the boundary conditions for the corresponding derivatives are computed as

$$\left.\begin{array}{ll}(\boldsymbol{n}\cdot\nabla\rho)_{\mathrm{B}}=-(\boldsymbol{n}\cdot\nabla\rho)_{\mathrm{M}}, & (\boldsymbol{\tau}\cdot\nabla\rho)_{\mathrm{B}}=(\boldsymbol{\tau}\cdot\nabla\rho)_{\mathrm{M}},\\ (\boldsymbol{n}\cdot\nabla p)_{\mathrm{B}}=-(\boldsymbol{n}\cdot\nabla p)_{\mathrm{M}}, & (\boldsymbol{\tau}\cdot\nabla p)_{\mathrm{B}}=(\boldsymbol{\tau}\cdot\nabla p)_{\mathrm{M}},\\ (\boldsymbol{n}\cdot\nabla u_n)_{\mathrm{B}}=(\boldsymbol{n}\cdot\nabla u_n)_{\mathrm{M}}, & (\boldsymbol{\tau}\cdot\nabla u_n)_{\mathrm{B}}=(\boldsymbol{\tau}\cdot\nabla u_n)_{\mathrm{M}},\\ (\boldsymbol{n}\cdot\nabla u_\tau)_{\mathrm{B}}=-(\boldsymbol{n}\cdot\nabla u_\tau)_{\mathrm{M}}, & (\boldsymbol{\tau}\cdot\nabla u_\tau)_{\mathrm{B}}=(\boldsymbol{\tau}\cdot\nabla u_\tau)_{\mathrm{M}},\end{array}\right\} \tag{2.37}$$

where u_n and u_τ are equal to $\boldsymbol{n}\cdot\boldsymbol{u}$ and $\boldsymbol{\tau}\cdot\boldsymbol{u}$, respectively. As can be seen, the normal velocity u_n is specified once to be weakly zero in terms of the primitive value, and its derivatives $\boldsymbol{n}\cdot\nabla u_n$,$\boldsymbol{\tau}\cdot\nabla u_n$ are extrapolated. Similar algorithms apply to the other fields.

3. Shock capturing models

In this section, we describe the shock capturing models considered in this paper. These can be divided into two categories. One is the artificial viscosity model, which adds explicit dissipation to the original equation to stabilize discontinuities, while the other borrows the idea of WENO methods to reconstruct the solution within troubled cells.

3.1. Derivative-based model

This model is a simplified form of the artificial bulk model[20], which generally requires the computation of higher order derivatives of artificial properties and this is both expensive and inaccurate. In this work, we employ the dilation, computed as

$$\mu_\beta = c_\beta \mid \nabla\cdot\boldsymbol{u} \mid (h/P)^2, \tag{3.1}$$

instead of its higher order derivatives as the shock sensor. Here $\boldsymbol{u}$ is the scalar quantity q for scalar cases, and the velocity vector for Euler systems. c_β is an empirical parameter.

For scalar cases, $\nabla\cdot\boldsymbol{u}$ would identify locations where the scalar quantity varies abruptly, and remain small elsewhere. For the Euler systems, $\nabla\cdot\boldsymbol{u}$ is the velocity dilation which achieves large absolute values for strong compression(negative dilation) and expansion(positive dilation) regions. In the literature[22], it is typical to modify the value of $\nabla\cdot\boldsymbol{u}$ to be 0 for regions of expansion to further reduce the dissipation. We employ the original formulation. In this work, we refer to this method as the derivative-based(DB) model.

The final viscosity given by this model is determined as

$$\mu_{\mathrm{DB}} = \min(\mu_\beta, \mu_{\max}), \tag{3.2}$$

where $\mu_{\max}$ is an upper limit, given as

$$\mu_{\max} = c_{\max}(h/P)\max_{\boldsymbol{x}\in G_K} \mid \boldsymbol{f}'(q_h(\boldsymbol{x},t)) \mid, \tag{3.3}$$

where $c_{\max}$ is an empirical parameter, G_K denotes the nodal points within element K, and $\boldsymbol{f}'(q_h(\boldsymbol{x},t))$ gives the local wave speed.

3.2. Highest modal decay model

This model was first proposed by Persson and Peraire[23], and relates the strength of the discontinuity to the decay rate of the modal expansion coefficients. To explain the model, we first define the truncated solution as

$$\tilde{q}_h = \sum_{m=0}^{N_{P-1}-1} \hat{q}_m\varphi_m. \tag{3.4}$$

The shock sensor is defined to be the fraction of q_h's energy contained in the highest mode, i.e.

$$S_K = \frac{\parallel q_h - \tilde{q}_h \parallel^2_{L^2_K}}{\parallel q_h \parallel^2_{L^2_K}}, \tag{3.5}$$

where $\| \cdot \|_{L^2_K}$ denotes the standard L^2 norm on element K. For scalar cases, S_K is determined with the scalar quantity, while for Euler systems, the density is used.

Generally, a smooth function can be approximated with an expansion, in which the P-th mode scales as $1/P^2$. Then, for a smooth field, S_K should scale as $1/P^4$. Since this model relies on the decay of the highest mode, we refer to it as the highest modal decay(MDH) model. The final viscosity in [23] is given as

$$\mu_{\mathrm{MDH}} = \mu_{\max} \begin{cases} 0 & \text{if } s_K < s_0 - c_\kappa \\ \frac{1}{2}(\sin(1 + \frac{\pi(s_K - s_0)}{2c_\kappa})) & \text{if } s_0 - c_\kappa \leq s_K \leq s_0 + c_\kappa \\ 1 & \text{otherwise} \end{cases}, \tag{3.6}$$

where $s_K = \log_{10} S_K$, $s_0 = -(c_A + 4\log_{10} P)$. c_A and c_κ are empirical parameters. In this work, $\mu_{\max}$ is given by Eq. (3.3).

3.3. Averaged modal decay model

The MDH model is reasonably successful for the identification of the discontinuity and is able to give a clear criteria on when to add full viscosity(i.e. $\mu_{\max}$). However, as can be seen above, the MDH model only relies on the highest mode to determine the decay rate. As is shown in [24], frequently the modal coefficients can be rather oscillatory for both smooth and discontinuous solution, and the information contained in the highest mode may not be enough. Therefore, the MDH model may often underestimate the smoothness. To overcome this issue, more information in the modal expansion should be used to determine the decay rate more accurately. This is the motivation behind the model proposed by Klöckner et al[24]. We will first illustrate the idea of the model in 1D, and then propose a simple yet effective extenstion to multi-dimensional cases.

3.3.1. One-dimensional case

We first assume that the modal coefficients scale as

$$| \hat{q}_m | \simeq C m^{-\tau}. \tag{3.7}$$

Taking the logarithm of both sides, we obtain

$$\log | \hat{q}_m | \simeq \log(C) - \tau \log(m). \tag{3.8}$$

Similar to MDH, the working variable is chosen to be the scalar quantity for scalar cases and density for Euler systems. We compute the decay rate τ in a least-squared sense, which leads to the following problem

$$\sum_{m=1}^{N_P - 1} (\log | \hat{q}_m | - (\log(C) - \tau \log(m)))^2. \tag{3.9}$$

The decay rate τ will be recovered by prescribing the above expression to achieve a minimum value through the least-squared fitting.

As is clear, the cell average is not included in determining the decay rate(see Eq. (3.9)). Therefore, when the solution is constant with slight oscillations, the sensor senses the oscillations without realizing their amplitude, and yields a decay rate of nearly zero. This is clearly undesirable, as it would produce an overly large dissipation. To address this, a baseline modal decay is added to modify $\hat{q}_m$ as following.

First, a perfect modal decay is constructed as

$$b_m = \frac{m^{-P}}{\sqrt{\sum_{m=1}^{N_P - 1} m^{-2P}}}. \tag{3.10}$$

Then, the modal coefficients $\hat{q}_m$ are modified as

$$| \check{q}_m |^2 = | \hat{q}_m |^2 + \| q_h \|^2_{L^2_K} | b_m |^2, \quad m \in 1, ..., N_P - 1. \tag{3.11}$$

Before this model can be applied, one more issue needs to be addressed. The above method would require the modal coefficients to decay monotonely, which, however, is usually not the case. As a result, the predicted decay rate can frequently be too high, causing insufficient viscosity. Therefore, a skyline procedure is proposed to fix this issue, which modifies $\check{q}_n$ as

$$\check{q}_m = \underset{n=min(m,N_P-2),...,N_P-1}{max} \mid \check{q}_n \mid, \quad m \in \{1, 2, ..., N_P - 1\}. \tag{3.12}$$

This forces $\check{q}_m$ to be the maximum value of the original modal coefficients among all higher modes, and sets the last mode to be larger than the second-to-last mode to avoid odd-even effects of the modal coefficients.

We use $\check{q}_m$ as the final modal coefficients in Eq. (3.9), and the decay rate is close to P for situations where the solution is nearly constant. Since the model determines the smoothness indicator in an averaged sense, we call this model the averaged modal decay(MDA) model. The final viscosity is given as

$$\mu_{\text{MDA}} = \mu_{\max} \begin{cases} 1 & \text{if } \tau < 1 \\ 1 - (\tau - 1)/2 & \text{if } 1 \le \tau \le 3 \\ 0 & \text{otherwise} \end{cases}, \tag{3.13}$$

and $\mu_{\max}$ is also given by Eq. (3.3).

3.3.2. Multi-dimensional case

In this section, we propose a simple yet effective way to extend the above one-dimensional model to multi-dimensions. The idea is illustrated in the two-dimensional case and its extension to three-dimension is straightforward.

Step 1. Extract the nodal values of the specified quantity on the three edges of the triangular element.

Step 2. Along each edge, apply the one-dimensional model described in the previous section, resulting in three modal decay rates ($\tau_e, e = 0, 1, 2$) for the three edges.

Step 3. Choose the smallest decay rate τ_e among these three as the final decay rate of the element. This is then used in Eq. (3.13) to compute the piecewise-constant artificial viscosity.

Remark 1. The above extension method is found to keep the performance of the model consistent across different spacial dimensions. Also, this method preserves symmetry.

Remark 2. As noted in [24], the MDA model is specifically designed for higher accuracy orders, and will be only used in this paper with $P4$ or higher.

3.4. Entropy-based viscostiy model

This model[26] makes use of the condition of entropy around discontinuities to determine the smoothness of the solution. For conservation laws, one can define the entropy pair as $E(\boldsymbol{q}), \boldsymbol{F}(\boldsymbol{q})$, which satisfy

$$Q = \frac{\partial E}{\partial t} + \nabla \cdot \boldsymbol{F} \le 0, \tag{3.14}$$

provided q is the weak solution to the conservation law. The equality holds for smooth solution, and the residual Q decreases for discontinuities, which therefore serves as a good smoothness indicator. In the following, we will first choose the entropy pairs for the conservation laws considered in this work, and then describe how to compute the entropy-based viscosity(EV).

3.4.1. *Choosing the entropy pairs*

For the one-dimensional linear transport equation, we use

$$E = \frac{1}{2}q^2, \quad F = \frac{1}{2}q^2. \tag{3.15}$$

For the one-dimensional Burgers' equation, we have

$$E = \frac{1}{2}q^2, \quad F = \frac{2}{3}q^3. \tag{3.16}$$

For the Euler systems, we have

$$E = \frac{\rho}{\gamma - 1}\log(p/\rho^{\gamma}), \quad \boldsymbol{F} = \boldsymbol{u}E. \tag{3.17}$$

where γ is the adiabatic constant, and $\boldsymbol{u}$ is the velocity vector.

3.4.2. *Entropy-based viscosity*

The entropy-based viscosity is determined by

$$\mu_E = c_E(h/P)^2 B, \tag{3.18}$$

where c_E is an empirical parameter, and

$$B = \max\left(\max_{\boldsymbol{x}\in G_K} \mid Q(\boldsymbol{q}_h) \mid, \max_{\boldsymbol{x}\in G_{\partial K}} \mid H(\boldsymbol{q}_h) \mid\right)/T. \tag{3.19}$$

$G_{\partial K}$ denotes the nodal points along all the edges of K, and $H(\boldsymbol{q}_h)$ introduces the impact of the jump along the element edges, given as

$$H = (h/P)^{-1}\boldsymbol{n} \cdot [\![\boldsymbol{F}]\!]. \tag{3.20}$$

T acts as normalization, and is given as

$$T = \max_{\boldsymbol{x}\in\Omega}\left|E - \frac{1}{\mid\Omega\mid}\int_{\Omega} E d\Omega\right|, \tag{3.21}$$

where Ω denotes the entire computational domain. Note that for Euler systems, there is no need for normalization, and in that case $T = 1$. The final viscosity for this model is

$$\mu_{\mathrm{EV}} = \min(\mu_{\max}, \mu_E). \tag{3.22}$$

Similarly to the previous two models, $\mu_{\max}$ is given by Eq. (3.3).

3.5. *WENO limiter*

This method first relies on an indicator to identify the troubled cells where a discontinuity occurs, and employs the idea of a WENO reconstruction to capture discontinuities[14, 15]. A most distinguishable feature of this method is that instead of performing resconstruction for point values or moments individually, it reconstructs the entire polynomial of the target cell by using information from itself and its immediate neighbours. The stencil is therefore kept at a minimum regardless of the order of accuracy. In the following, we will first describe how to identify the troubled cells, and then present the WENO limiting procedure for scalar and system cases respectively.

3.5.1. Identification of troubled cells

There are a variety of methods for the identification of troubled cells in the context of DG. We have chosen the indicators according to [14, 15]. The interested readers are referred to [30] for a systematic comparison of typcial indicators.

For the one-dimensional scalar transport and Burgers' equations, we use the TVB indicator. For a j-th cell $K_j^0 = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, we define the corresponding quantities as

$$\dot{q}_j = \bar{q}_j - q^+_{j-\frac{1}{2}}, \quad \ddot{q}_j = q^-_{j+\frac{1}{2}} - \bar{q}_j, \quad \Delta_+\bar{q}_j = \bar{q}_{j+1} - \bar{q}_j, \quad \Delta_-\bar{q}_j = \bar{q}_j - \bar{q}_{j-1}, \tag{3.23}$$

where $\bar{q}_j$ denotes the cell-averaged value within K_j^0. The superscripts '−' and '+' indicate the left and right sides respectively at the element interface.

Now we are ready to introduce the TVB limiter, which is used to limit the two values $\dot{q}_j$ and $\ddot{q}_j$ as

$$\dot{q}_j^{(\mathrm{mod})} = \mathrm{TVB}(\dot{q}_j, \Delta_+\bar{q}_j, \Delta_-\bar{q}_j), \quad \ddot{q}_j^{(\mathrm{mod})} = \mathrm{TVB}(\ddot{q}_j, \Delta_+\bar{q}_j, \Delta_-\bar{q}_j), \tag{3.24}$$

where the TVB function is defined as

$$\mathrm{TVB}(a_1, ..., a_N) = \begin{cases} 0 & \text{if } |a_1| \leq c_M h^2 \\ \mathrm{minmod}(a_1, ..., a_N) & \text{otherwise} \end{cases}, \tag{3.25}$$

$$\mathrm{minmod}(a_1, ..., a_N) = \begin{cases} \epsilon \min\limits_{1 \leq m \leq N} |a_m| & \text{if } \epsilon = \mathrm{sign}(a_1) = \cdots = \mathrm{sign}(a_N) \\ 0 & \text{otherwise} \end{cases}. \tag{3.26}$$

The current cell K_j^0 is to be flagged as a troubled cell if either $\dot{q}_j$ or $\ddot{q}_j$ is modified by Eq. (3.24). In Eq. (3.25), c_M is the so-called TVB constant, and needs to be adjusted according to the specific problems.

For the Euler systems, we employ the KXRCF indicator[31]. The d-th immediate neighbour of K_j^0 is written as K_j^d. First, the boundary of the current cell K_j^0 is divided into two parts as ∂K_j^{0-} and ∂K_j^{0+}, along which the field flows into and out of K_j^0, respectively. Then, the KXRCF indicator is defined as

$$\frac{\int_{\partial K_j^{0-}} (q^0_{h,j} - q^d_{h,j}) d\sigma}{h^{\frac{P+1}{2}} \left|\partial K_j^{0-}\right| \cdot \| q^0_{h,j} \|} > c_K, \tag{3.27}$$

where c_K is an empirical parameter. Once the above inequality holds, K_j^0 is then flagged to be a troubled cell. Following [31], $\| q^0_{h,j} \|$ is chosen to be the cell-averaged value for the one-dimesional Euler system, and the maximum norm for the two-dimensional Euler system. Also, we take the density as the working variables for $q^0_{h,j}$ and $q^d_{h,j}$.

3.5.2. WENO limiting procedure for scalar cases

In this section, we illustrate the WENO limiting procedure for the two-dimensional scalar case, from which the one-dimensional case can be obtained immediately. The three neighbours of K_j^0 are denoted as K_j^1, K_j^2 and K_j^3 respectively. The limiting procedure is given in the following.

Step 1. Assume the polynomials on the reconstruction stencil to be $z_d(x, y), d = 0, 1, 2, 3$. We will first modify the polynomials as

$$\tilde{z}_d(x, y) = z_d(x, y) - \frac{1}{|K_j^0|} \int_{K_j^0} z_d(x, y) dK + \frac{1}{|K_j^0|} \int_{K_j^0} z_0(x, y) dK. \tag{3.28}$$

The above equation ensures that $\tilde{z}_d(x, y)$ has the same cell average on K_j^0 as $z_0(x, y)$. In practical computations, we first obtain the nodal values of $z_d(x, y)$ on K_j^0 through extrapolation, and then construct a polynomial with the nodal values using the nodal basis on K_j^0. Finally, we obtain $\tilde{z}_d(x, y)$ by adjusting its cell average to be equal to that of $z_0(x, y)$.

Step 2. Next we determine the linear weights for the WENO reconstruction. Since the full polynomial is used for each cell within the stencil, there is no further requirements for the linear weights c_d other than they sum to one. The choice of the linear weights is now only a matter of balance between accuracy and stability. We follow [14, 15] for the linear weights, which will be given in Section 4.

Step 3. In order to measure the smoothness of the solution within each cell of the stencil, we compute the smoothness indicator

$$\beta_d = \sum_{|l|=1}^{P} |K_j^0|^{|l|-1} \int_{K_j^0} \left(\frac{\partial^{|l|}}{\partial x^{l_1} \partial y^{l_2}} \tilde{z}_d(x,y) \right)^2 dK, d = 0, ..., 3, \tag{3.29}$$

where $l = (l_1, l_2)$.

The key issue here is to compute the derivatives of the Jacobi polynomials. For the one-dimensional case, we obtain the required high-order derivatives directly following the approach in [32]. For the two-dimensional case, we take the first-order derivative of the solution, and obtain their nodal values. Then we apply the derivative on the derivative values at nodal points repeatedly to obtain higher-order derivatives.

Step 4. With the linear weights and smoothness indicator given, we compute the nonlinear weights as

$$\omega_d = \frac{\bar{\omega}_d}{\sum_{l=0}^{3} \bar{\omega}_d}, \quad \bar{\omega}_d = \frac{c_d}{(\varepsilon + \beta_d)^2}, \tag{3.30}$$

where ε is a small number to avoid division by 0, and is fixed to be 1.0E-6 in this paper.

Step 5. Finally the reconstructed polynomial for K_j^0 is given as

$$z_j^{\text{WENO}}(x,y) = \sum_{d=0}^{3} \omega_d \tilde{z}_d(x,y) \tag{3.31}$$

It is easy to see that $z_j^{\text{WENO}}(x,y)$ has the same cell average as $z_0(x,y)$ due to Eq. (3.28).

3.5.3. WENO limiting procedure for systems

For systems, the above WENO limiting procedure needs to be applied on the characteristic variables.

Step 1. Denote the normal direction of the three edges of the triangluar element as $\boldsymbol{n}_e, e = 0, 1, 2$. Along each edge, we compute the left and right eigenmatrix $\mathcal{L}_e$ and $\mathcal{R}_e$ from the Jacobian matrix $n_e^x(\boldsymbol{f}_e^x)' + n_e^y(\boldsymbol{f}_e^y)'$.

(a) Project the polynomials z_d to characteristic fields using $z_d^c = \mathcal{L}_e z_d, d = 0, 1, 2, 3$.

(b) Perform the above steps for the scalar case on each charactieristic field and obtain the new polynomials limited by WENO given as $\tilde{z}_e^{c,\text{WENO}}$.

(c) Project the limited polynomial back by $\tilde{z}_e^{\text{WENO}} = \mathcal{R}_e \tilde{z}_e^{c,\text{WENO}}$.

Step 2. Obtain the final solution as the average of the limited polynomials $\tilde{z}_e^{\text{WENO}}$ as

$$z_j^{WENO} = \frac{\sum_e |K_j^e| \tilde{z}_e^{\text{WENO}}}{\sum_e |K_j^e|}. \tag{3.32}$$

Remark 3. In this work, the WENO limiting is only used with $P1 - P3$ for one-dimensional cases and $P1 - P2$ for two-dimensional cases. We will further discuss this with numerical experiments in Section 4.1.

3.6. Smoothing the artificial viscosity

The artificial viscosity obtained for the MDH, MDA and EV models is piecewise-constant, which may cause oscillations for high order accuracy. For the DB model, although the resulting viscosity varies with smoothness within each cell, experience indicates that further smoothing is still critical for robustness especially for cases with strong shocks. For smoothing the viscosity, there are some sophiscated examples including the PDE-based approach[33] and the filtering approach[34]. In this work, we keep things simple and perform a $C0$ smoothing method for the artificial viscosity decribed in the following.

Table 1. Basline parameters for shock capturing models

	1D transport	1D Burgers'	1D Euler	2D Euler
DB	$c_\beta = 1, c_{\max} = 0.5$	$c_\beta = 1, c_{\max} = 0.5$	$c_\beta = 1, c_{\max} = 0.5$	$c_\beta = 1, c_{\max} = 0.5$
EV	$c_E = 1, c_{\max} = 0.5$	$c_E = 1, c_{\max} = 0.5$	$c_E = 0.15, c_{\max} = 0.15$	$c_E = 0.15, c_{\max} = 0.15$
MDH	$c_A = 2.5, c_\kappa = 0.2,$ $c_{\max} = 0.5$	$c_A = 2.5, c_\kappa = 0.2,$ $c_{\max} = 0.5$	$c_A = 2.5, c_\kappa = 0.2,$ $c_{\max} = 0.25$	$c_A = 2.5, c_\kappa = 0.2,$ $c_{\max} = 0.25$
MDA	$c_{\max} = 1$	$c_{\max} = 1$	$c_{\max} = 1$	$c_{\max} = 1$
WENO	$c_M = 100, c_0 = 0.998,$ $c_1 = 0.001, c_2 = 0.001,$	$c_M = 100, c_0 = 0.998,$ $c_1 = 0.001, c_2 = 0.001,$	$c_K = 1, c_0 = 0.998,$ $c_1 = 0.001, c_2 = 0.001,$	$c_K = 1, c_0 = 0.997,$ $c_1 = 0.001, c_2 = 0.001,$ $c_3 = 0.001$

Step 1. Choose the nodal point set corresponding to $P = 2$, and conduct averaging at each node with all the cells sharing the same node.
Step 2. Construct a Lagrange polynomial of second order based on the averaged nodal values.
Step 3. With the polynomial constructed above, we compute the viscosity on the nodal points required by the discretization order.

Remark 4. This smoothing technique is applied to all the following computations except those of the DB model in 1D, for which we use the original viscosity to better demonstrate the sub-cell nature of the model.

4. Numerical experiments

In the following, we conduct a comparison of the five shock capturing models with typical benchmarks including smooth and non-smooth cases. All the results given in this section are obtained using the five-stage fourth-order low-storage explicit Runge-Kutta method[35]. For all the cases in the following, we plot point-wise results unless specified otherwise.

Before we present the detailed results, it is important to address the issue of empirical parameters. In Table 1, we list the baseline parameters for each model. These are determined from reasonable balance between accuracy and stability with minimum tuning among different cases. The baseline parameters are used throughout the paper unless specified otherwise. A quantitative comparison for these models is difficult and not the target of this work. One can typically adjust the parameters for a specific case and there is no rigorous definition of the optimal solution for a general problem. Hence, we aim to compare the behavior of different models in a qualitative fashion, less affected by the empirical parameters than one might expect.

4.1. Convergence tests with smooth problems

4.1.1. One-dimensional linear transport

The convergence test is performed with the one-dimesional linear transport equation(Eq. (2.3)). The computational domain is $[0, 2]$, and the numerical error is calculated at $t = 4.0$, with the exact solution being $q(x, t) = \sin(\pi(x - t))$. For WENO, we have set $c_M = 0.01$ for the TVB indicator to increase the number of cells touched by the limiter. Also, for WENO, it has been found that the inclusion of the downwind stencil for this case degenerates the accuracy or may render the computation unstable. Therefore, we choose the parameter as $c_0 = 0.998, c_1 = 0.002, c_2 = 0(K_j^1$ is on the left of K_j^0, and K_j^2 is on the right) for this case. The results are given in Tables 2 and 3 and compared with linear cases where no shock capturing is applied. As we can see, the DB model is restricted to second order as indicated by Eq. (3.1). The MDH model is first order due to the term $\mu_{\max}$ in Eq. (3.6) in the pre-asymptotic region and returns to the linear scheme once the resolution is fine enough or the order of accuracy is high enough to switch off the shock sensor . Both the EV and WENO methods achieves the original theoretical accuracy order despite higher numerical errors than the linear scheme. Note that the WENO limiting becomes unstable for P3 on the coarsest grid. For this case, the MDA shock sensor is not activated for all the computations.

Table 2. L^2 errors and accuracy orders of shock capturing models for the one-dimensional scalar case. $P1 - P3$.

	N	Linear		DB		EV		MDH		WENO	
		L^2 error	Order	L^2 error	Order	L^2 error	Order	L^2 error	Order	L^2 error	Order
$P1$	10	3.70E-02		5.57E-01		6.54E-01		6.83E-01		1.61E-01	
	20	6.91E-03	2.42	3.42E-01	0.70	2.87E-01	1.19	5.72E-01	0.25	5.86E-02	1.45
	40	1.53E-03	2.18	1.42E-01	1.27	3.89E-01	2.88	3.65E-01	0.65	1.44E-02	2.02
	80	3.68E-04	2.05	4.64E-02	1.62	6.46E-03	2.59	2.11E-01	0.79	2.58E-03	2.48
	160	9.12E-05	2.01	1.30E-02	1.83	9.84E-04	2.71	1.15E-01	0.87	4.90E-04	2.40
	320	2.27E-05	2.00	3.38E-03	1.95	1.33E-04	2.89	5.90E-02	0.96	1.01E-04	2.28
$P2$	10	1.71E-03		3.57E-01		1.22E-01		4.59E-01		6.40E-02	
	20	2.17E-04	2.98	1.46E-01	1.29	8.90E-03	3.78	2.17E-04	11.0	3.59E-03	4.16
	40	2.72E-05	3.00	4.70E-02	1.63	8.63E-04	3.37	2.72E-05	3.00	2.23E-04	4.01
	80	3.40E-06	3.00	1.31E-02	1.84	6.01E-05	3.84	3.40E-06	3.00	1.34E-05	4.05
	160	4.25E-07	3.00	3.39E-03	1.95	3.91E-06	3.94	4.25E-07	3.00	8.65E-07	3.96
	320	5.31E-08	3.00	8.54E-04	1.99	2.52E-07	3.95	5.31E-08	3.00	6.84E-08	3.66
$P3$	10	6.93E-05		2.21E-01		7.67E-03		6.93E-05		Unstable	
	20	4.33E-06	4.00	7.66E-02	1.53	2.78E-04	4.78	4.33E-06	4.00	3.10E-03	
	40	2.71E-07	4.00	2.26E-02	1.76	1.06E-05	4.72	2.71E-07	4.00	1.71E-04	4.18
	80	1.69E-08	4.00	5.97E-03	1.92	3.49E-07	4.92	1.69E-08	4.00	9.46E-06	4.18
	160	1.06E-09	4.00	1.51E-03	1.98	1.13E-08	4.95	1.06E-09	4.00	4.90E-07	4.27
	320	6.67E-11	3.99	3.80E-04	1.99	3.68E-10	4.94	6.67E-11	3.99	2.48E-08	4.31

Table 3. L^2 errors and accuracy orders of shock capturing models for the one-dimensional scalar case. $P4 - P5$.

	N	Linear		DB		EV		MDH		MDA	
		L^2 error	Order	L^2 error	Order	L^2 error	Order	L^2 error	Order	L^2 error	Order
$P4$	5	7.00E-05		3.55E-01		6.99E-02		7.00E-05		7.00E-05	
	10	2.18E-06	5.01	1.46E-01	1.28	5.51E-04	6.99	2.18E-06	5.01	2.18E-06	5.01
	20	6.83E-08	4.99	4.70E-02	1.64	1.01E-05	5.77	6.83E-08	4.99	6.83E-08	4.99
	40	2.16E-09	4.98	1.31E-02	1.84	1.39E-07	6.18	2.16E-09	4.98	2.16E-09	4.98
	80	6.60E-11	5.03	3.39E-03	1.95	2.40E-09	5.86	6.60E-11	5.03	6.60E-11	5.03
$P5$	5	3.75E-06		2.78E-01		1.01E-02		3.75E-06		3.75E-06	
	10	5.98E-08	5.97	1.03E-01	1.43	1.58E-05	9.33	5.98E-08	5.97	5.98E-08	5.97
	20	9.37E-10	6.00	3.16E-02	1.71	1.65E-07	6.59	9.37E-10	6.00	9.37E-10	6.00
	40	1.73E-11	5.76	8.52E-03	1.89	2.88E-11	6.89	1.73E-11	5.76	1.73E-11	5.76

Table 4. L^2 errors and accuracy orders of shock capturing models for the two-dimensional isentropic vortex. $P1 - P3$.

	N	Linear		DB		EV		MDH		WENO	
		L^2 error	Order	L^2 error	Order	L^2 error	Order	L^2 error	Order	L^2 error	Order
$P1$	20	1.59E-02		2.45E-02		1.94E-02		1.84E-02		2.17E-02	
	40	3.50E-03	2.19	1.42E-02	0.79	5.66E-03	1.77	7.45E-03	1.30	1.26E-02	0.79
	80	6.33E-04	2.47	4.17E-03	1.77	7.60E-04	2.90	6.33E-04	3.56	2.06E-03	2.60
	160	1.06E-04	2.58	6.81E-04	2.61	1.15E-04	2.72	1.06E-04	2.58	1.24E-04	4.06
	320	2.18E-05	2.28	9.13E-05	2.90	2.25E-05	2.36	2.18E-05	2.28	2.18E-05	2.51
	640	5.13E-06	2.09	1.23E-05	2.89	5.17E-06	2.12	5.13E-06	2.09	5.13E-06	2.09
$P2$	20	5.23E-03		9.10E-03		5.06E-03		2.41E-02		2.17E-02	
	40	8.64E-04	2.60	1.38E-03	2.72	8.27E-04	2.61	1.87E-02	0.37	4.74E-03	2.19
	80	1.54E-04	2.49	1.79E-04	2.95	1.52E-04	2.44	1.11E-02	0.75	1.24E-04	1.93
	160	3.46E-05	2.16	3.21E-05	2.48	3.45E-05	2.14	3.29E-03	1.75	1.52E-04	3.03
	320	8.09E-06	2.10	7.29E-06	2.14	8.08E-06	2.09	8.09E-06	8.67	7.59E-06	4.33
	640	1.72E-06	2.23	1.62E-06	2.17	1.72E-06	2.23	1.72E-06	2.23	1.57E-06	2.28
$P3$	20	2.56E-03		4.66E-03		2.86E-03		1.75E-02			
	40	3.18E-04	3.01	4.50E-04	3.37	3.26E-04	3.13	5.50E-03	1.67		
	80	1.18E-05	4.75	1.71E-05	4.71	1.22E-05	4.74	1.18E-05	8.86	Unstable	
	160	5.96E-07	4.31	7.13E-07	4.59	6.03E-07	4.34	5.96E-07	4.31		
	320	3.84E-08	3.95	4.19E-08	4.09	3.86E-08	3.97	3.84E-08	3.95		

4.1.2. Two-dimensional Euler system

In this section, we continue the convergence tests for the two-dimensional Euler system. The test case is the isentropic vortex convection problem. The computational domain is $[-10, 10] \times [-10, 10]$ with periodic boundary conditions along all boundaries. The initial vortex is located at $(x_0, y_0) = (-5, -5)$, and is convected until $t = 10$. The initial field is given as

$$\left.\begin{array}{ll} \rho = T^{\frac{1}{\gamma-1}}, & p = \rho^{\gamma}, \\ u = 1 - \frac{5}{2\pi}\exp\left(\frac{1-r^2}{2}\right)(y - y_0), & v = 1 + \frac{5}{2\pi}\exp\left(\frac{1-r^2}{2}\right)(x - x_0), \\ T = 1 - \frac{25(\gamma-1)}{8\gamma\pi^2}\exp(1 - r^2), & r = \sqrt{(x - x_0)^2 + (y - y_0)^2}. \end{array}\right\} \tag{4.1}$$

The grids are generated from uniform structured grids by dividing each quadrilateral element into two triangular elements. The number N in Tables 4 and 5 indicate the number of cells along each edge. Similar to the one-dimensional case, the MDH model is entirely switched off for sufficent grid resolution. The MDA model is triggered on the coarsest grids for this case, and recovers the accuracy of the linear scheme quickly on finer grids. For the DB model, the above initial condition indicates very weak compressbility, thus producing small values for $| \nabla \cdot \boldsymbol{u} |$. As a result, the viscosity decreases more rapidly than indicated by Eq. (3.1) to a value of around zero, adding extra reduction of the numerical error. This explains why the convergence order is higher than 2, which is the case in the one-dimensional case. The EV model is able to retain the design accuracy of the original linear scheme for all the computations. For WENO, $P1$ and $P2$ are both able to achieve the design accuracy of their linear counterparts, while $P3$ becomes unstable for all the cases. In consideration of the unstable case for WENO in the one-dimensional smooth solution, we speculate that this is caused by the extrapolation error which is generally more significant with higher order polynomials. We also point out that no results for $P3$ or higher orders are given for 2D in [14, 15]. In the following we will only conduct WENO computations with $P1 - P3$ for one-dimensional cases, and $P1 - P2$ for two-dimensional cases.

4.2. One-dimensional scalar problems with discontinuities

4.2.1. One-dimensional linear transport

The computational domain is $[0, 1]$. The initial condition is given as

Table 5. L^2 errors and accuracy orders of shock capturing models for the two-dimensional isentropic vortex. $P4 - P5$.

	N	Linear		DB		EV		MDH		MDA	
		L^2 error	Order	L^2 error	Order	L^2 error	Order	L^2 error	Order	L^2 error	Order
$P4$	10	1.27E-02		9.41E-03		7.08E-03		1.76E-02		1.21E-02	
	20	5.31E-04	4.58	1.60E-03	2.56	9.10E-04	2.96	8.36E-03	1.08	9.15E-04	3.72
	40	2.40E-05	4.47	4.70E-05	5.09	2.97E-05	4.93	2.40E-05	8.44	2.40E-05	5.25
	80	1.17E-06	4.36	2.03E-06	4.54	1.21E-06	4.62	1.17E-06	4.36	1.17E-06	4.36
	160	8.62E-08	3.76	1.17E-07	4.11	8.62E-08	3.81	8.62E-08	3.76	8.62E-08	3.76
	320	5.17E-09	4.06	6.14E-09	4.26	5.17E-09	4.06	5.17E-09	4.06	5.17E-09	4.06
$P5$	10	4.79E-03		5.68E-03		4.64E-03		1.94E-02		8.52E-03	
	20	2.83E-04	4.08	7.03E-04	3.01	3.82E-04	3.61	1.14E-02	0.76	2.83E-04	4.91
	40	8.03E-06	5.14	1.07E-05	6.04	8.43E-06	5.50	8.03E-06	10.5	8.03E-06	5.14
	80	8.70E-08	6.53	1.07E-07	6.64	8.89E-08	6.57	8.70E-08	6.53	8.70E-08	6.53
	160	7.63E-10	6.83	9.44E-10	6.83	7.70E-10	6.85	7.63E-10	6.83	7.63E-10	6.83

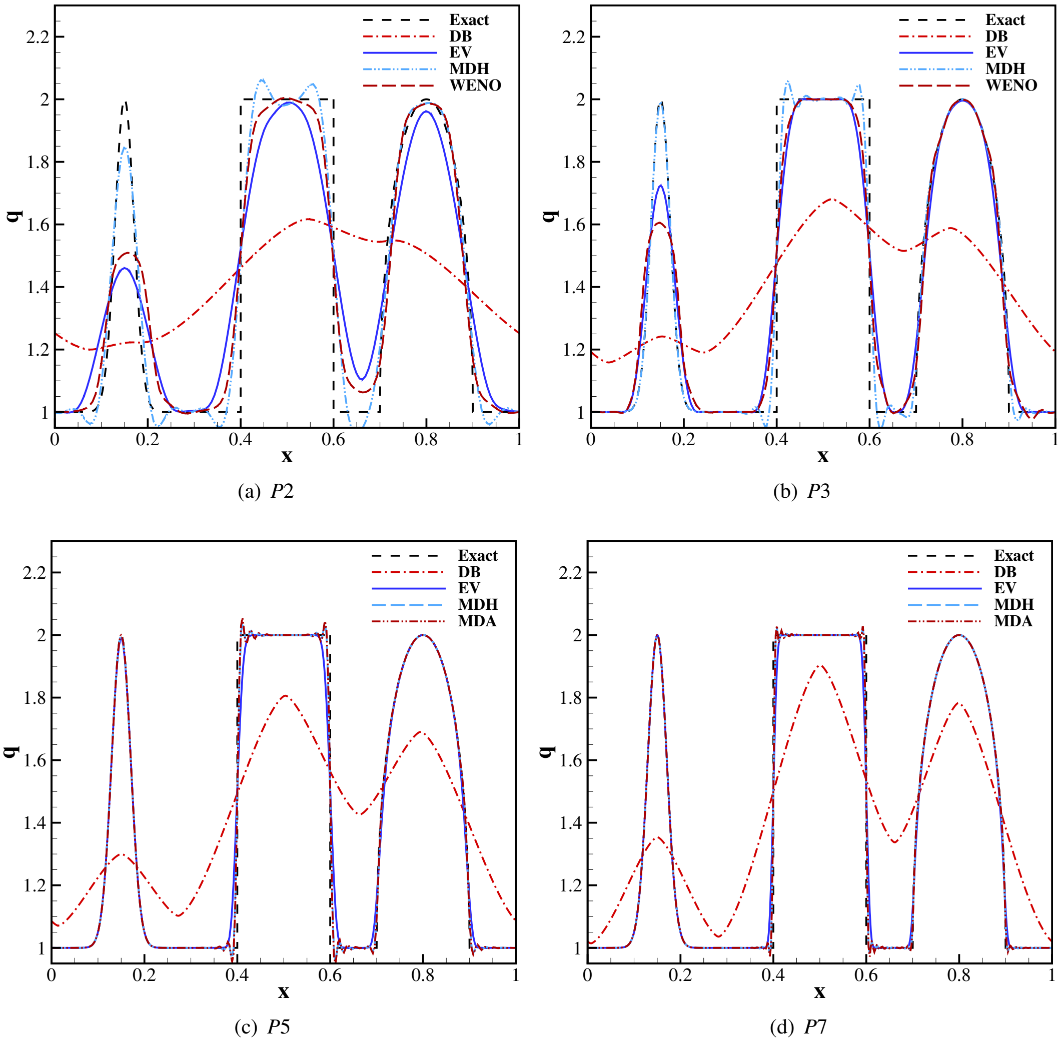
$$q(x,0) = \begin{cases} 1 + e^{-300(2x-0.3)^2} & \text{if } |2x-0.3| \leq 0.25, \\ 2 & \text{if } |2x-1.0| \leq 0.2, \\ 1 + (1 - (\frac{2x-1.6}{0.2})^2)^{\frac{1}{2}} & \text{if } |2x-1.6| \leq 0.2, \\ 1 & \text{otherwise.} \end{cases} \tag{4.2}$$

Periodic boundary conditions are applied on the boundaries. The cell number is fixed to be 80 for all the cases. The results at $t = 100$ are given in Fig. 2. The DB model is the most dissipative as indicated in the one-dimensional convergence test. The EV model is able to suppress spurious oscillations without excessive dissipation. The solution using the WENO limiting is comparable with that of the EV model for $P3$, while slightly less dissipative for $P2$. The solution by the MDH and MDA models are similar, keeping the dissipation at a very low level and showing noticable oscillations. To better understand these oscillations, we plot the cell-averaged results in Fig. 3, in which the oscillations disappear for the MDH and MDA models at higher orders. Also, as explained in [32], oscillations shown above still contain high order information and may be used to recover a high-order accurate solution. Therefore, the oscillatory solutions still make sense in the context of DG.

Furthermore, we plot the temporal history of the artificial viscosity and the troubled-cell indicator for $t = [0, 0.04]$ in Fig. 4, where the peak values of artificial vsicosity at $t = 0$ and $t = 0.04$ are also given. To better visualize the results, we display the logarithm of the viscosity and plot the results with a relative frame which travels at the same speed as the waves. As can be seen, all the models capture the discontinuities at the right places. At $t = 0$, the initial discontinuity pushes the viscosity of all the models to the upper limit, i.e. $\mu_{\max}$, thus giving similar viscosity values. Large difference among them emerges after a very small time at $t = 0.04$. For DB, the viscosity decreases very slowly relative to the EV model. Also, the viscosity of DB is applied in a region much larger than the others. For the MDH and MDA models, they both reduce to zero viscosity very quickly. However, one can still observe that the MDA model generates a smoother viscosity than the MDH model in a temporal sense probably due to the averaging feature of MDA. The WENO limiting tracks the discontinuities similarly to the DB model, but is able to maintain low dissipation thanks to its high order reconstruction property.

4.2.2. One-dimensional Burgers' equation

The computational domain is $[0, 1]$, and the intial condition is $q(x, 0) = \sin(2\pi x)$. For this nonlinear case, a discontinuity develops from the smooth initial condition. The computation is terminated at $t = 0.3$, and the cell number is fixed to be 80 for all results. In Fig. 5, we plot the results for $P3$ and $P7$ since the results with other orders are very similar. It can be seen that all the shock capturing models are able to suppress the oscillations well. Also, we draw the temporal history of the artificial viscosity and the troubled-cell indicator for $t = [0, 0.3]$ in Fig. 6. As the discontinuity gradually develops, the shock sensors are activated after a certain point when the solution is steep enough. For this nonlinear case, the action region of artificial viscosity of DB is much smaller than those of the other viscosity models, due to its sub-cell nature. Also, the viscosity for MDH is rather oscillatory, which is attributed to



(a) *P*2

(b) *P*3

(c) *P*5

(d) *P*7

Fig. 2. Solution for the one-dimensional transport equation with 80 elements at $t = 100$.

its dependence on the decay rate of just the highest mode as analyzed in Section 3.3. A similar phenomenon is also observed for MDH in [32]. On the contrary, MDA is able to achieve a smooth viscosity similar to that of EV. For the WENO limiting, the cells with local extremum around the discontinuity is flagged as troubled cells, which is an attribution to the TVB function. Note that for a better illustration of the indicator behavior, we have set $c_M = 30$ just for Fig. 6e.

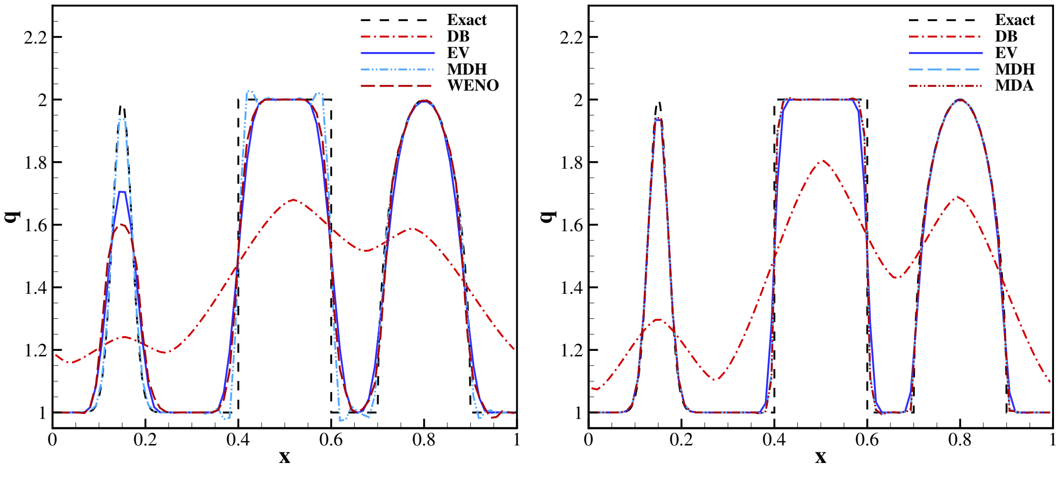
4.3. *Shu-Osher problem*

This problem is a one-dimesional model of shock/turbulence interaction, which contains complex yet typical features of compressible turbulence including shocklets and fluctuations. Therefore, a shock-capturing model is required to suppress spurious oscillations without causing too much dissipation for fluctuations. The computational domain is $[-5, 5]$, and the initial condition is given as

$$(\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.333333) & x < -4, \\ (1.0 + 0.2\sin(5x), 0, 1) & x > -4. \end{cases} \tag{4.3}$$

The computation is terminated at $t = 1.8$. The results on fine grids(the number of elements is fixed to be 200 for $P2, P3$, and 150 for $P5, P7$) are given first in Figs. 7 to 10. The results improve and agree with the reference solution as P increases. For this case, WENO displays more oscillations at shocklets than viscosity models with lower orders, while for higher orders, DB obtains smoother results around shocks than the other viscosity models. In Fig. 11, we further plot the temporal history of the artificial viscosity and the troubled-cell indicator, which provides better insights into the characteristics of the different models. Note that the logarithm of the viscosity is plotted. All the shock sensors capture the major structures of this complex problem. The DB model generates more localized viscosity than the other viscosity models around strong shocks, while maintaining larger viscosity around fluctuations. On the contrary, the EV model spreads the viscosity around shocks more widely while achieving low viscosity for fluctuations. The MDH and MDA models behave similarly and activate viscosity in very small regions, with a smoother variation in the temporal direction for MDA. For the WENO limiting, the troubled-cells are mainly distributed around the Mach 3 shock, which is likely the cause of the oscillations around shocklets.

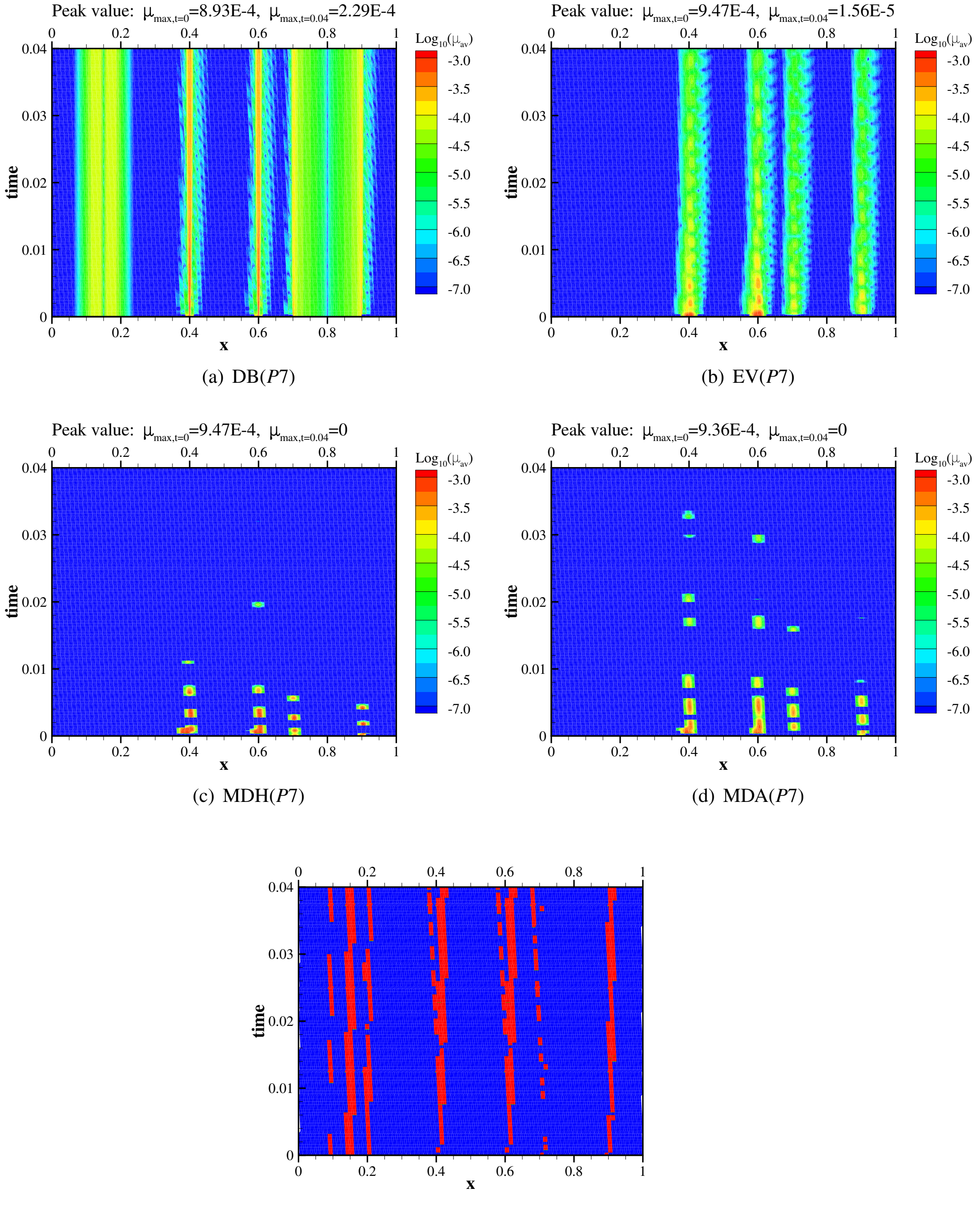
In order to further analyze the models, we repeat the computations on coarser grids based on the same number of degree of freedoms(DOFs) (equal to 480) for all the results(Fig. 12). It can be seen that with the same DOFs, the WENO limiting performs significantly worse for higher order accuracy, while all the viscosity models improve to a



(a) $P3$

(b) $P5$

Fig. 3. Cell-averaged solution for the one-dimensional transport equation with 80 elements at $t = 100$.



(a) DB(*P*7)

(b) EV(*P*7)

(c) MDH(*P*7)

(d) MDA(*P*7)

(e) WENO(*P*3)

Fig. 4. Temporal history of artificial viscosity and troubled-cell indicator for the one-dimensional transport equation with 80 elements for $t \in [0, 0.04]$.

certain extent. This is attributed to the large extrapolation error of the WENO recontructrion method on coarser grids. If we examine the results more carefully, we find that the DB model improves consistently from $P2$ to $P7$, while the improvement for the other three viscosity models stops at $P5$, and deteriorates a bit with $P7$. The source causing this phenomenon comes from the smoothness of the artificial viscosity. Recall that DB has a sub-cell nature, while the EV, MDH and MDA models generate piece-wise constant viscosity first, and then rely on a quadratic smoothing technique to obtain a distribution of $C0$ continuity. Therefore, for higher orders(i.e. coarser grids), the DB model adapts to complex flows better in a sub-cell level than its piece-wise constant counterparts, as indicated by Fig. 12.

4.4. Two-dimesnional Riemann problems

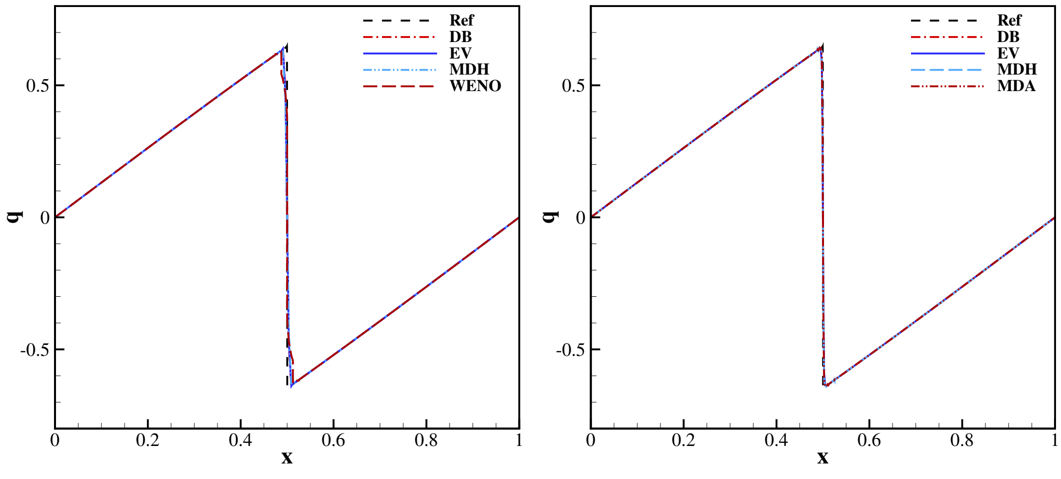
In the following, we will consider two-dimensional problems with shocks. First we consider the two-dimensional Riemann problems[36], including Case 4 and Case 12. For both cases, the computational domain is $[0, 1] \times [0, 1]$. The grid is obtained by dividing each quadrilateral element of a structured grid into two triangles. The number of elements along each edge is fixed to 160.

4.4.1. Case 4

For this case, the initial condition is given as

$$(\rho, u, v, p) = \begin{cases} (1.1, 0, 0, 1.1) & 0.5 < x < 1, 0.5 < y < 1, \\ (0.5065, 0.8939, 0.0, 0.35) & 0 < x < 0.5, 0.5 < y < 1, \\ (1.1, 0.8939, 0.8939, 1.1) & 0 < x < 0.5, 0 < y < 0.5, \\ (0.5065, 0, 0.8939, 0.35) & 0.5 < x < 1, 0 < y < 0.5. \end{cases} \tag{4.4}$$

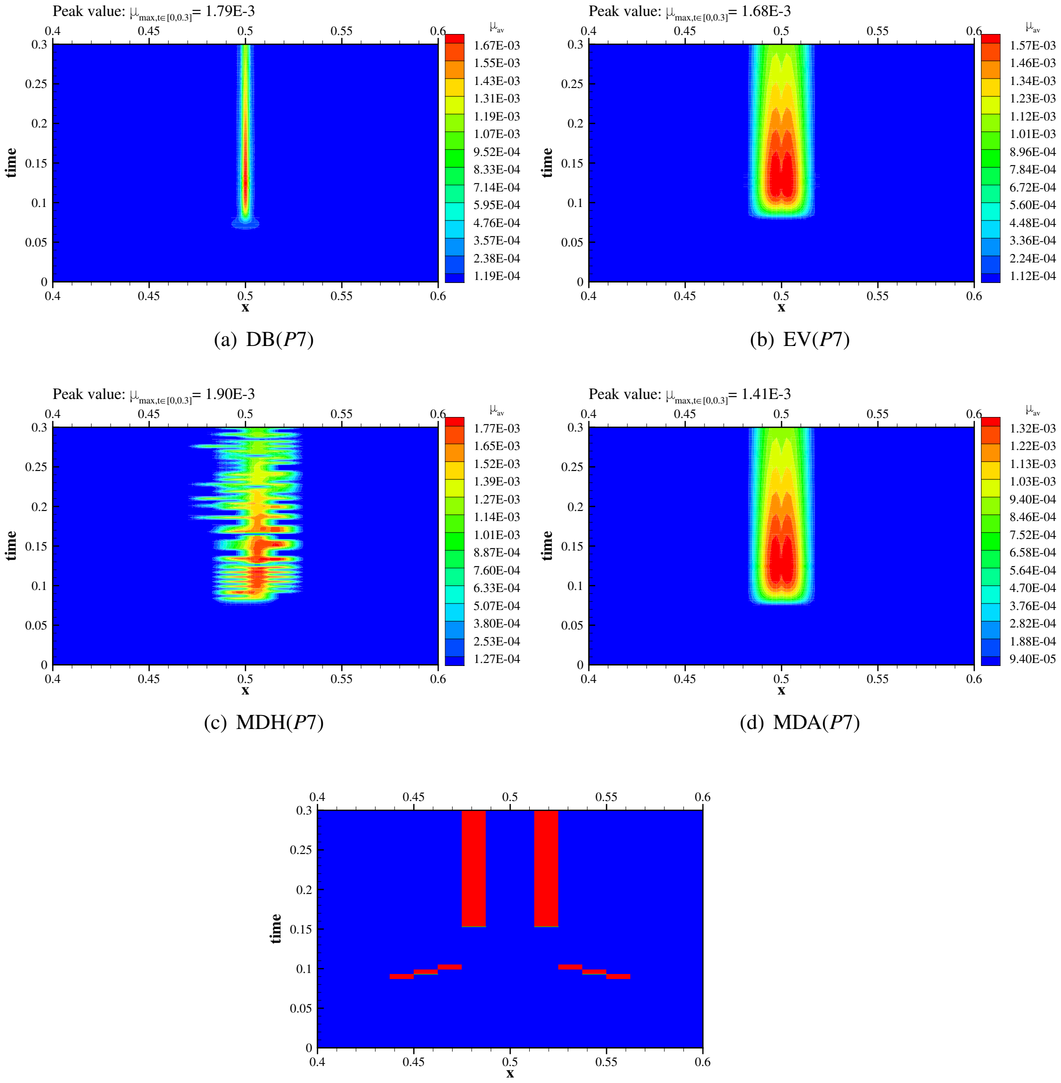
The density results at $t = 0.25$ are shown in Figs. 13 to 15. The DB model is shown to be more dissipative than EV and MDA, which are slightly oscillatory. The WENO limiting is more oscillatory than the viscosity models under the same condition. The MDH model displays an unusual phenomenon for $P2$, which is much more dissipative than for $P1$ on the same grid. After examing the artificial viscosity distribution (not shown), we find that although the peak value of the viscosity for $P2$ is lower than for $P1$, the high viscosity region for $P2$ is large causing excessive dissipation. This can be attributed to the oscillatory behavior of the viscosity as mentioned above. For $P4$, the MDH model returns to a normal level of dissipation. For a better illustration of the oscillations, we plot the cell-averaged solution in Fig. 16, in which the results are smoother.



(a) $P3$

(b) $P7$

Fig. 5. Solution for the one-dimensional Burgers' equation with 80 elements at $t = 0.3$.



(a) DB(*P*7)

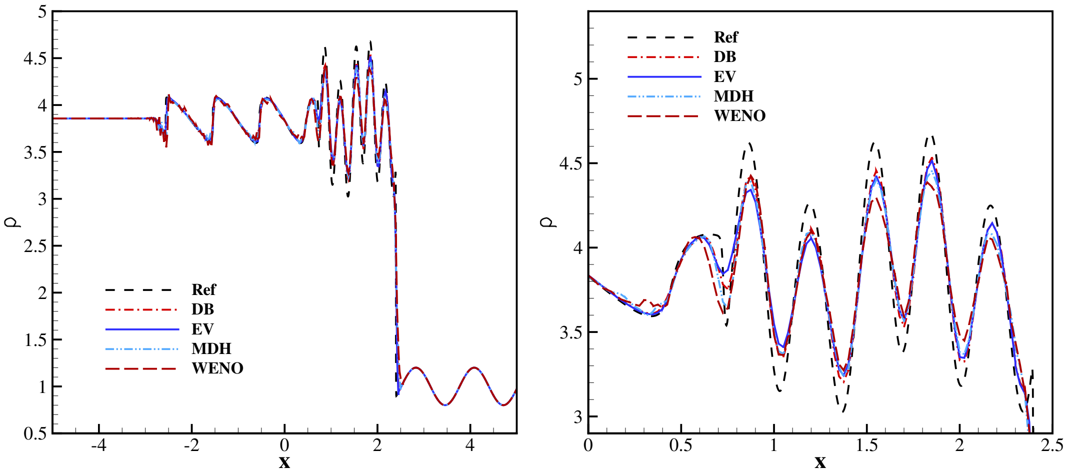
(b) EV(*P*7)

(c) MDH(*P*7)

(d) MDA(*P*7)

(e) WENO(*P*3)

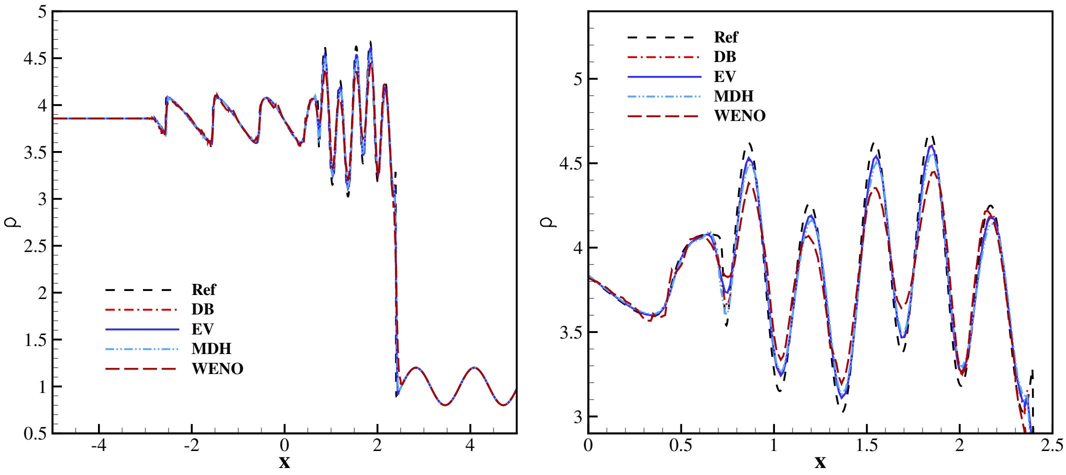
Fig. 6. Temporal history of artificial viscosity and troubled-cell indicator for the one-dimensional tranport equation with 80 elements for $t \in [0, 0.3]$.



(a) Global profiles

(b) Close up of the solution around fluctuations

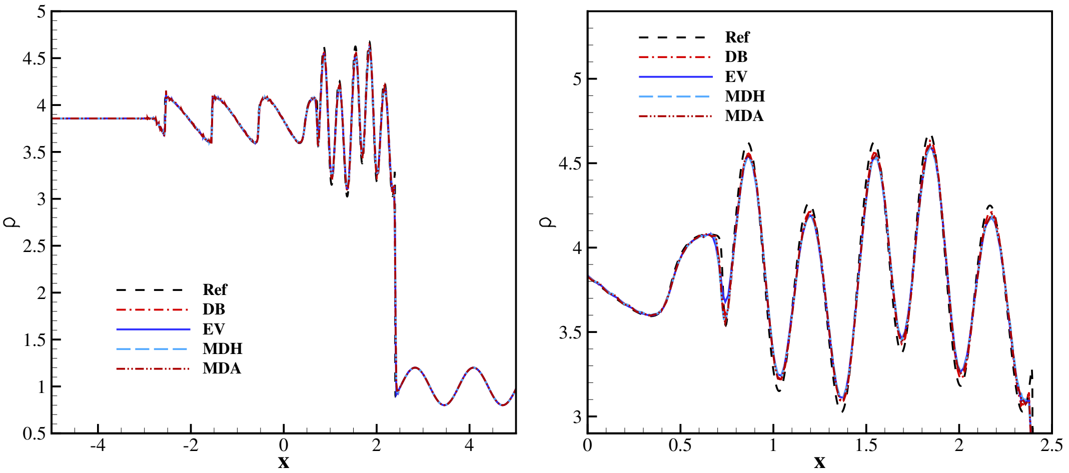
Fig. 7. Density solution for the one-dimensional Shu-Osher problem with 200 $P2$ elements at $t = 1.8$.



(a) Global profiles

(b) Close up of the solution around fluctuations

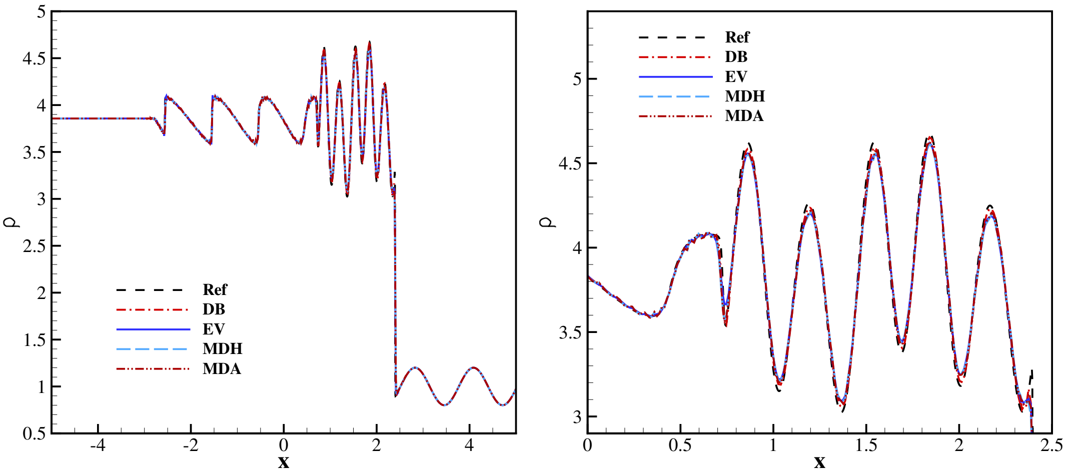
Fig. 8. Density solution for the one-dimensional Shu-Osher problem with 200 $P3$ elements at $t = 1.8$.



(a) Global profiles

(b) Close up of the solution around fluctuations

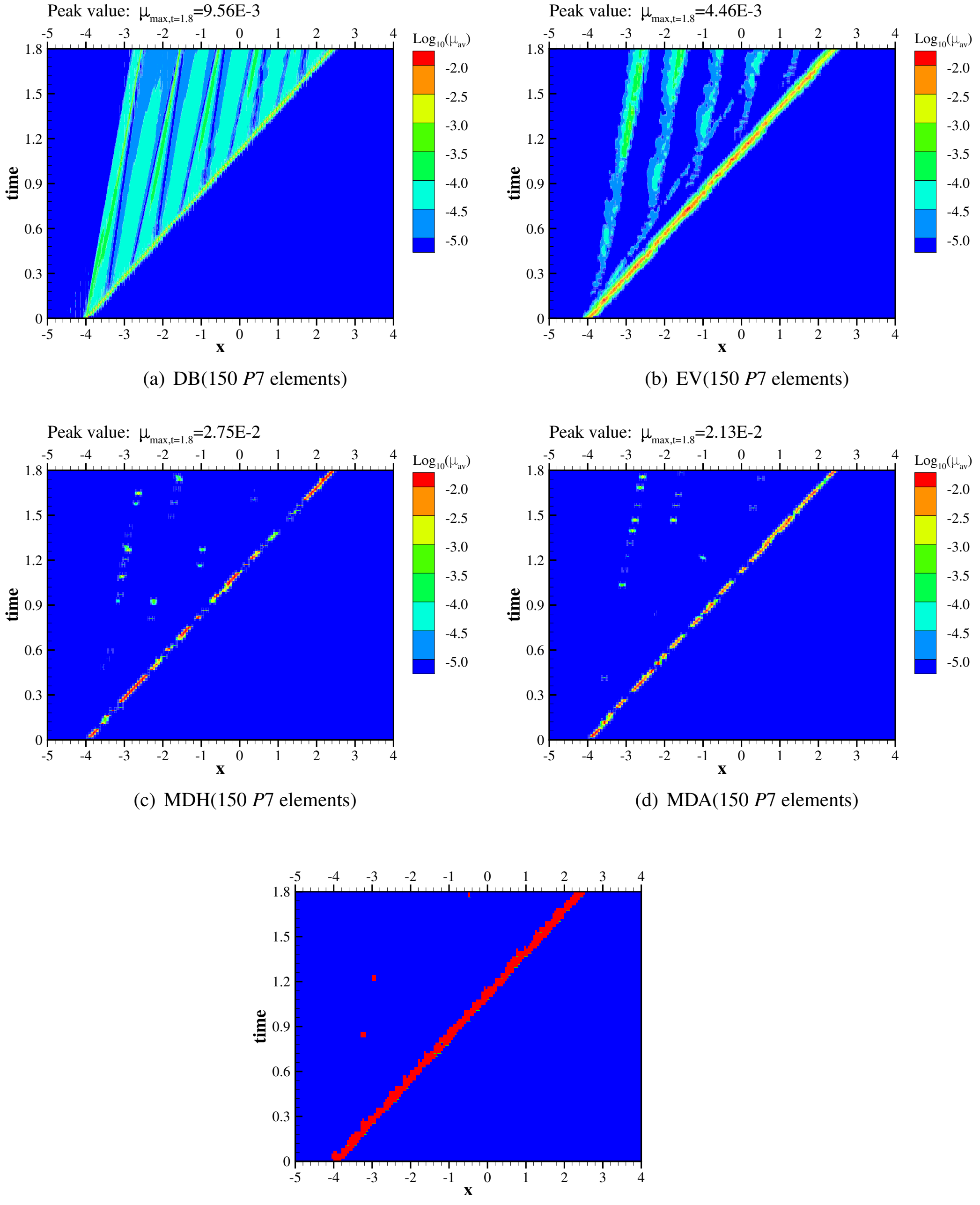
Fig. 9. Density solution for the one-dimensional Shu-Osher problem with 150 $P5$ elements at $t = 1.8$.



(a) Global profiles

(b) Close up of the solution around fluctuations

Fig. 10. Density solution for the one-dimensional Shu-Osher problem with 150 $P7$ elements at $t = 1.8$.



(a) DB(150 *P*7 elements)

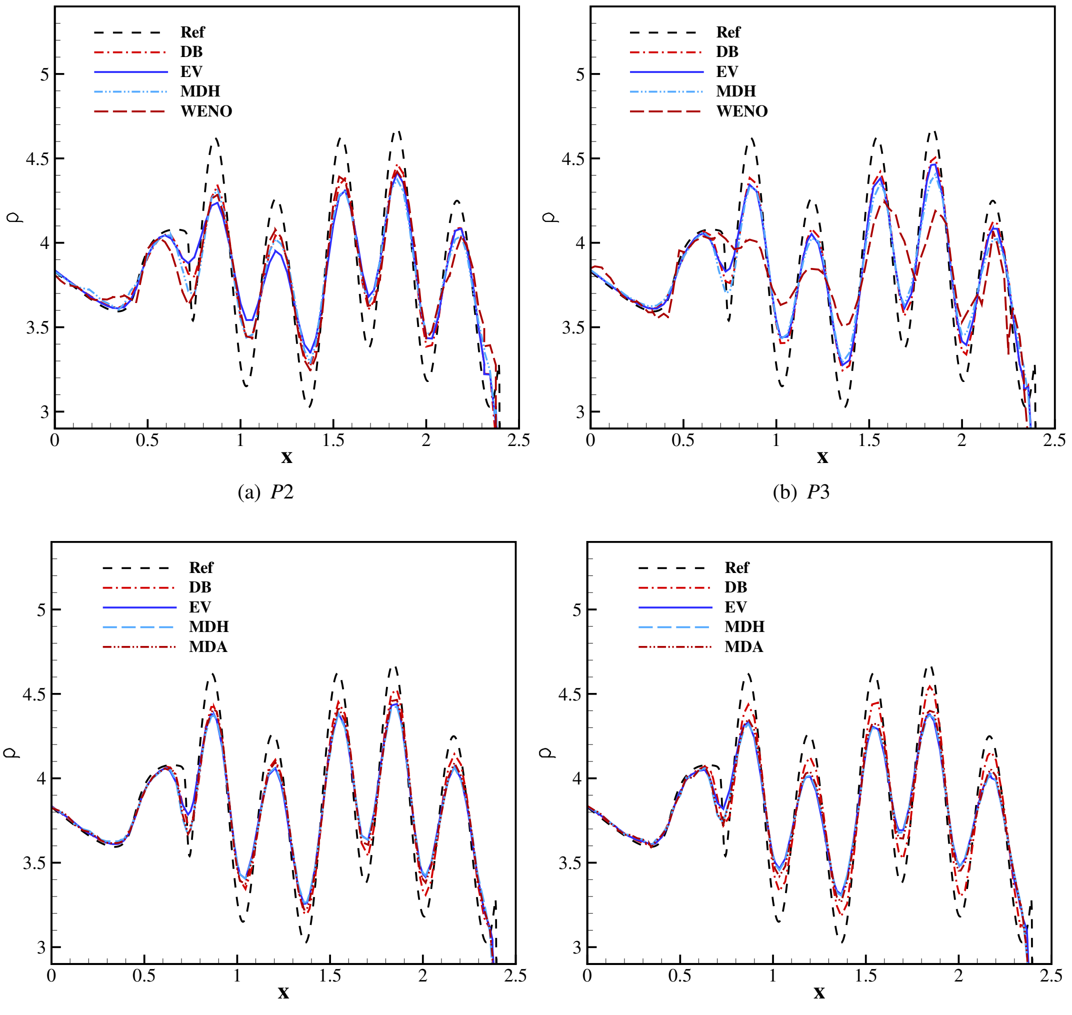
(b) EV(150 *P*7 elements)

(c) MDH(150 *P*7 elements)

(d) MDA(150 *P*7 elements)

(e) WENO(200 *P*3 elements)

Fig. 11. Temporal history of artificial viscosity and troubled-cell indicator for the one-dimensional Shu-Osher problem for $t \in [0, 1.8]$.



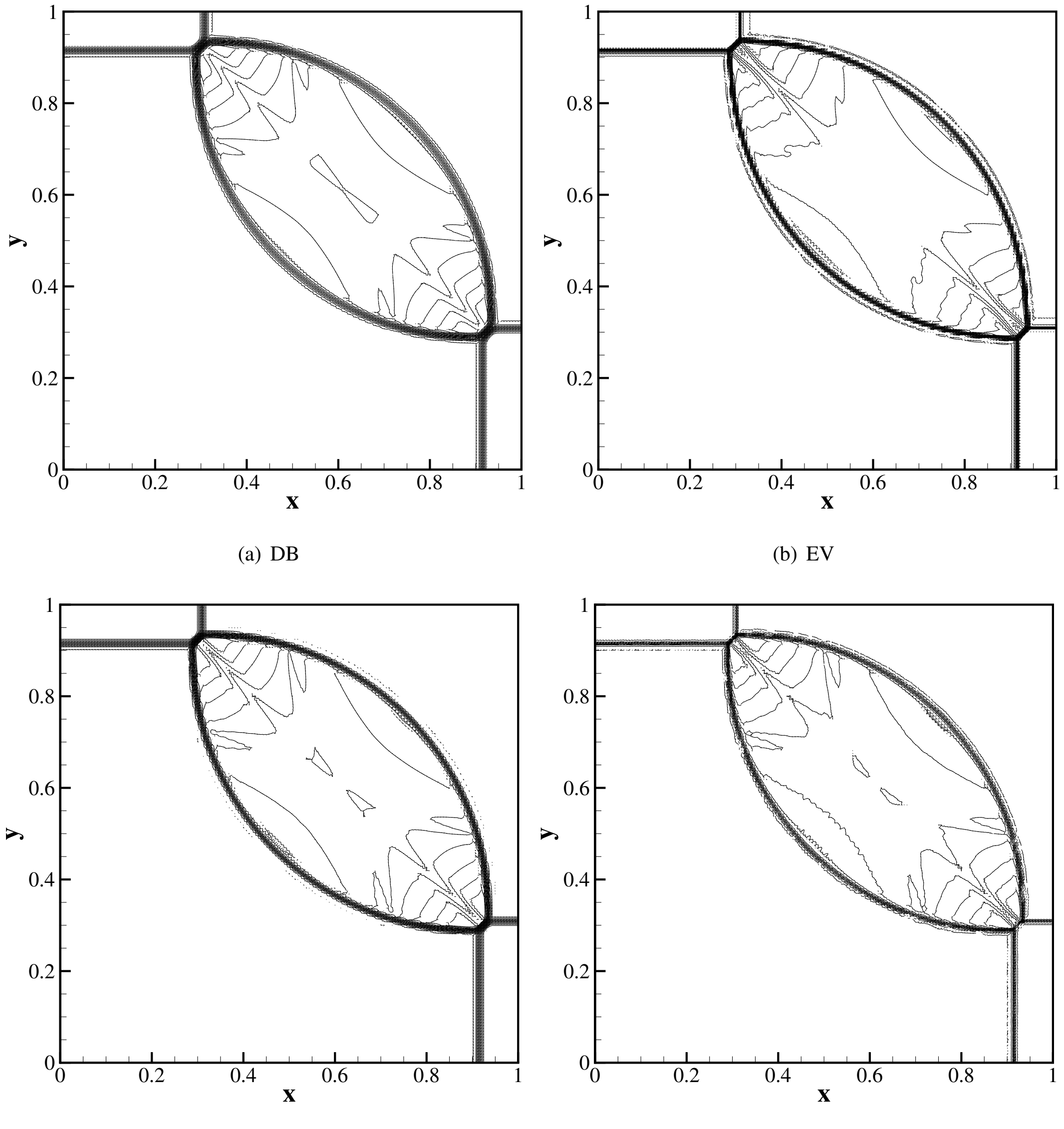
(a) *P*2

(b) *P*3

(c) *P*5

(d) *P*7

Fig. 12. Density solution for the one-dimensional Shu-Osher problem with DOFs equal to 480 at $t = 1.8$.



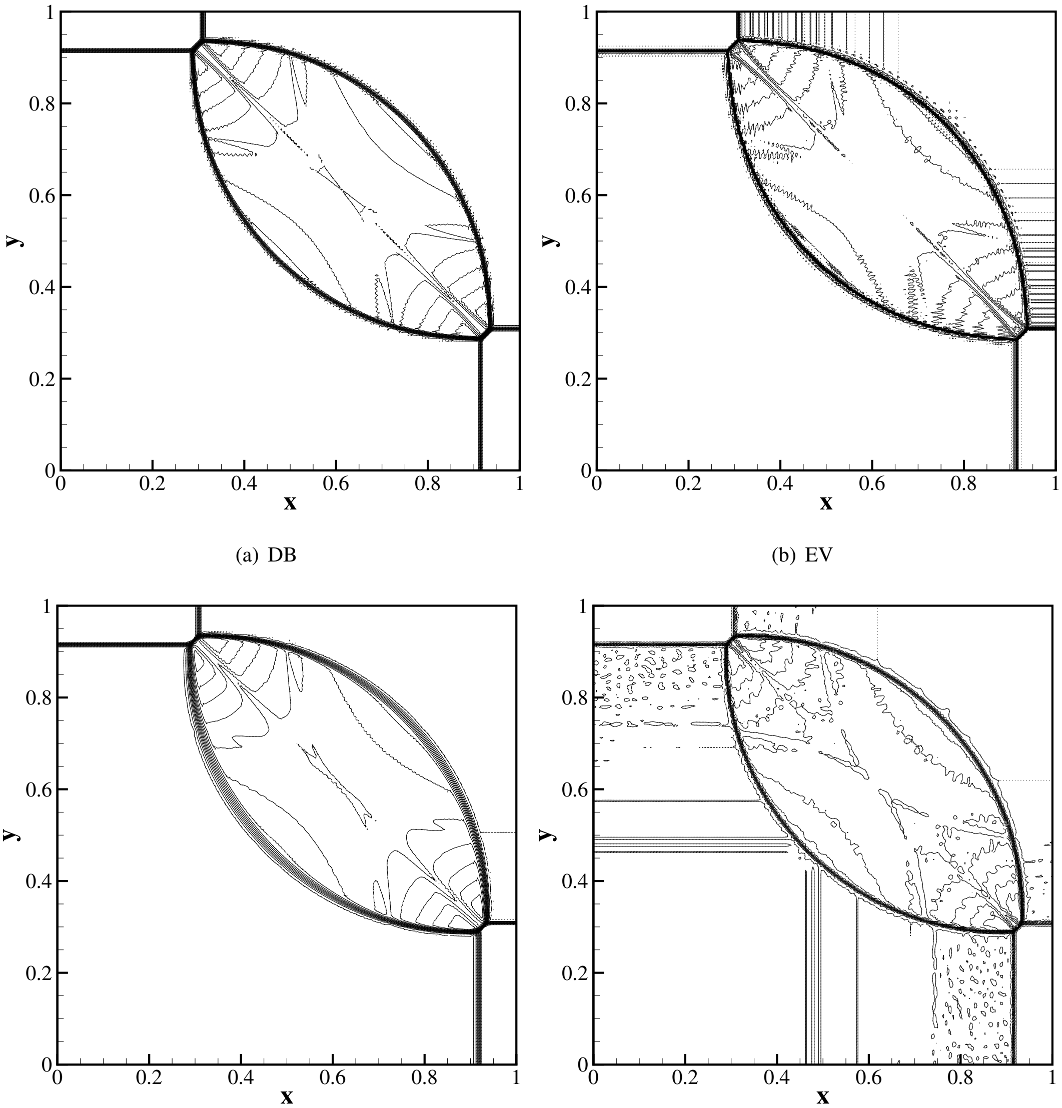
(a) DB

(b) EV

(c) MDH

(d) WENO

Fig. 13. Density solution for the two-dimensional Riemann problem (Case 4) with 160 × 160 × 2 $P1$ elements at $t = 0.25$. Thirty equally spaced contours from 0.255 to 1.9.



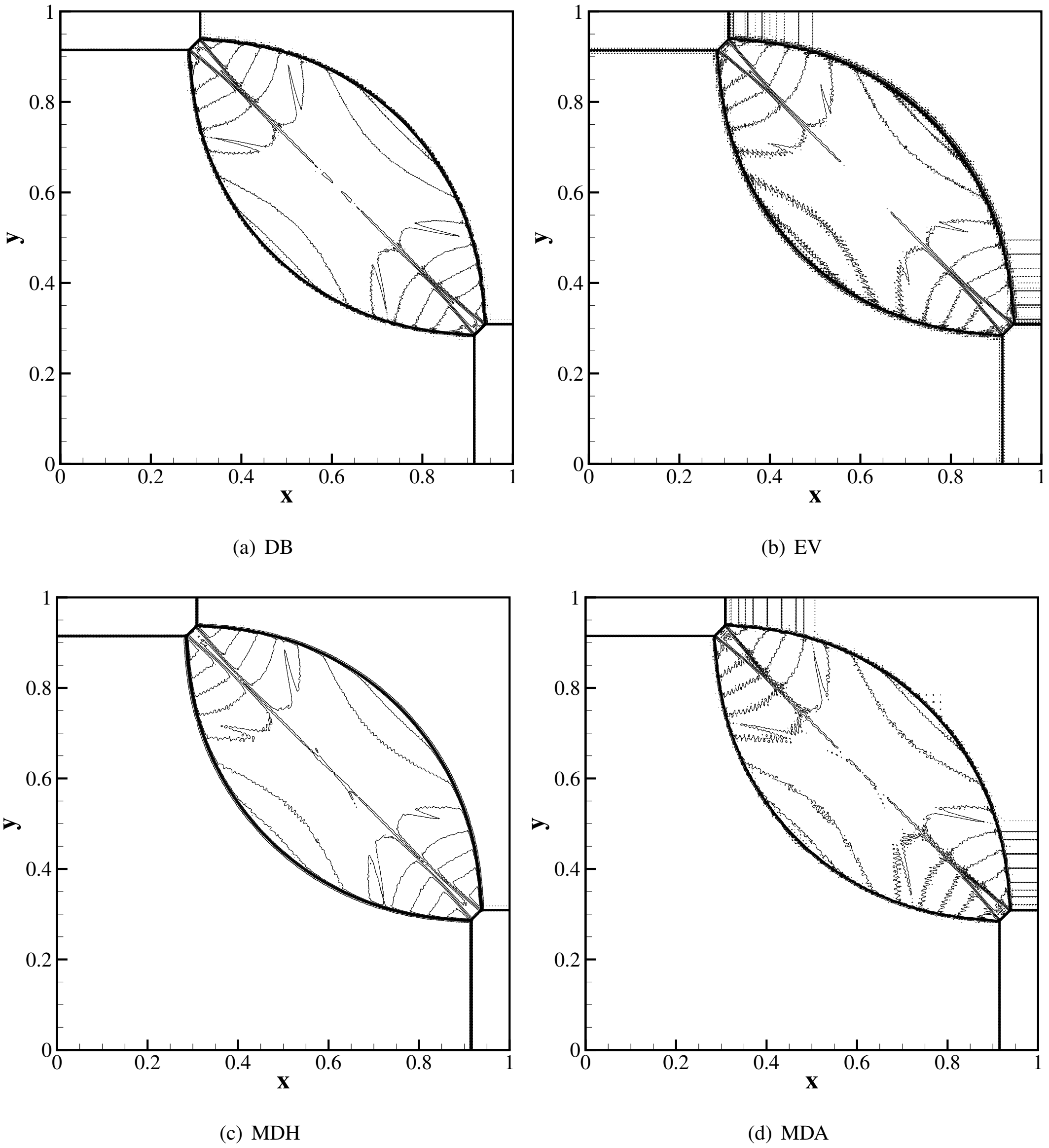
(a) DB

(b) EV

(c) MDH

(d) WENO

Fig. 14. Density solution for the two-dimensional Riemann problem (Case 4) with 160 × 160 × 2 $P2$ elements at $t = 0.25$. Thirty equally spaced contours from 0.255 to 1.9.



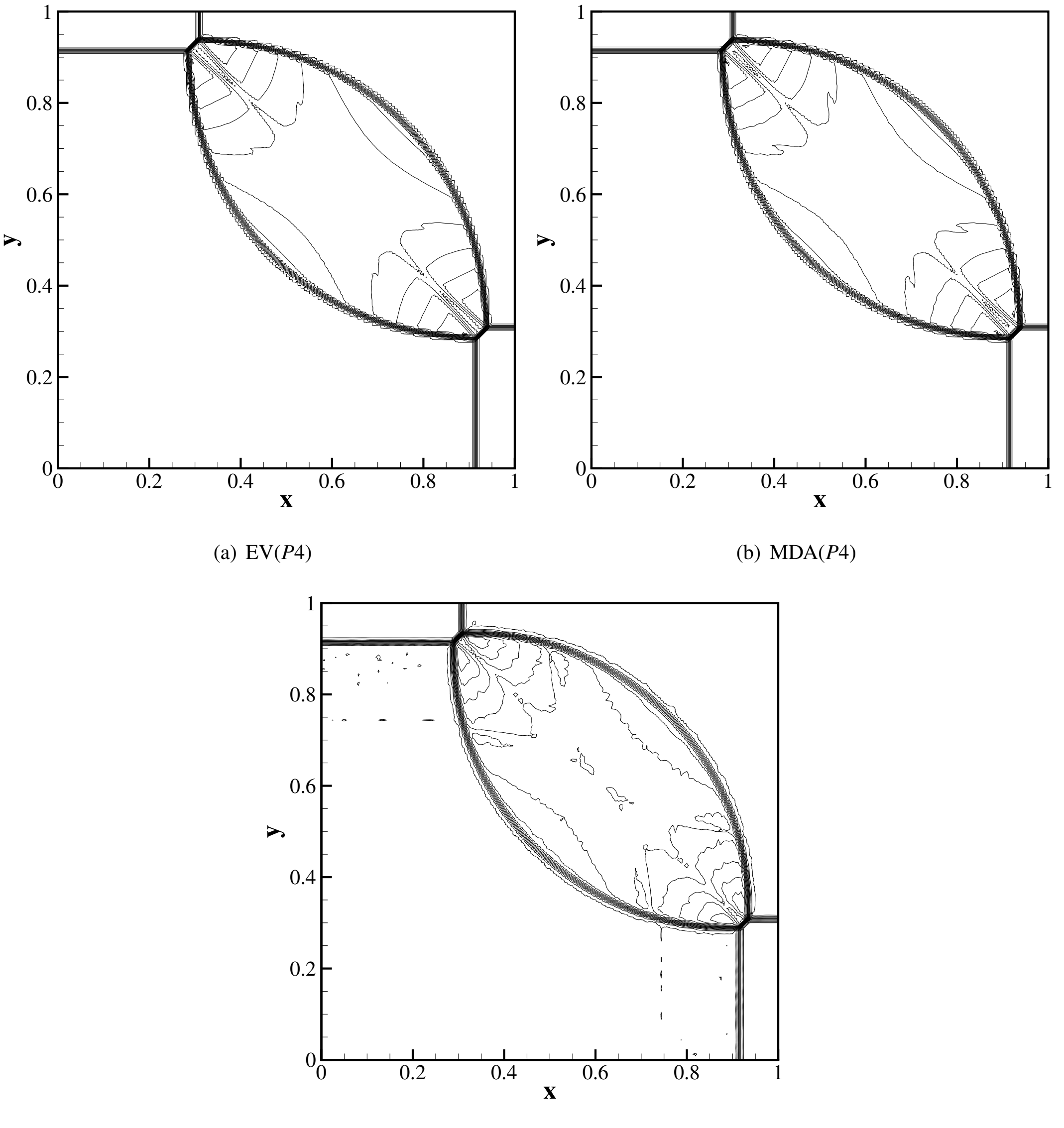
(a) DB

(b) EV

(c) MDH

(d) MDA

Fig. 15. Density solution for the two-dimensional Riemann problem (Case 4) with 160 × 160 × 2 $P4$ elements at $t = 0.25$. Thirty equally spaced contours from 0.255 to 1.9.



(a) EV($P4$)

(b) MDA($P4$)

(c) WENO($P2$)

Fig. 16. Cell-averaged density solution for the two-dimensional Riemann problem (Case 4) with 160 × 160 × 2 elements at t = 0.25. Thirty equally spaced contours from 0.255 to 1.9.

4.4.2. Case 12

For this case, the initial contintion is given as

$$(\rho, u, v, p) = \begin{cases} (0.5313, 0, 0, 0.4) & 0.5 < x < 1, 0.5 < y < 1, \\ (1.0, 0.7276, 0.0, 1.0) & 0 < x < 0.5, 0.5 < y < 1, \\ (0.8, 0, 0, 1) & 0 < x < 0.5, 0 < y < 0.5, \\ (1.0, 0, 0.7276, 1.0) & 0.5 < x < 1, 0 < y < 0.5. \end{cases} \tag{4.5}$$

The density results at $t = 0.25$ are shown in Figs. 17 to 19. The results of the comparison among different models are generally similar to Case 4. The DB model is more dissipative than EV and MDA, and the MDH model shows the same unusual behavior in terms of consistency. The WENO limiting generates the most oscillatory results under the same condition among all the models.

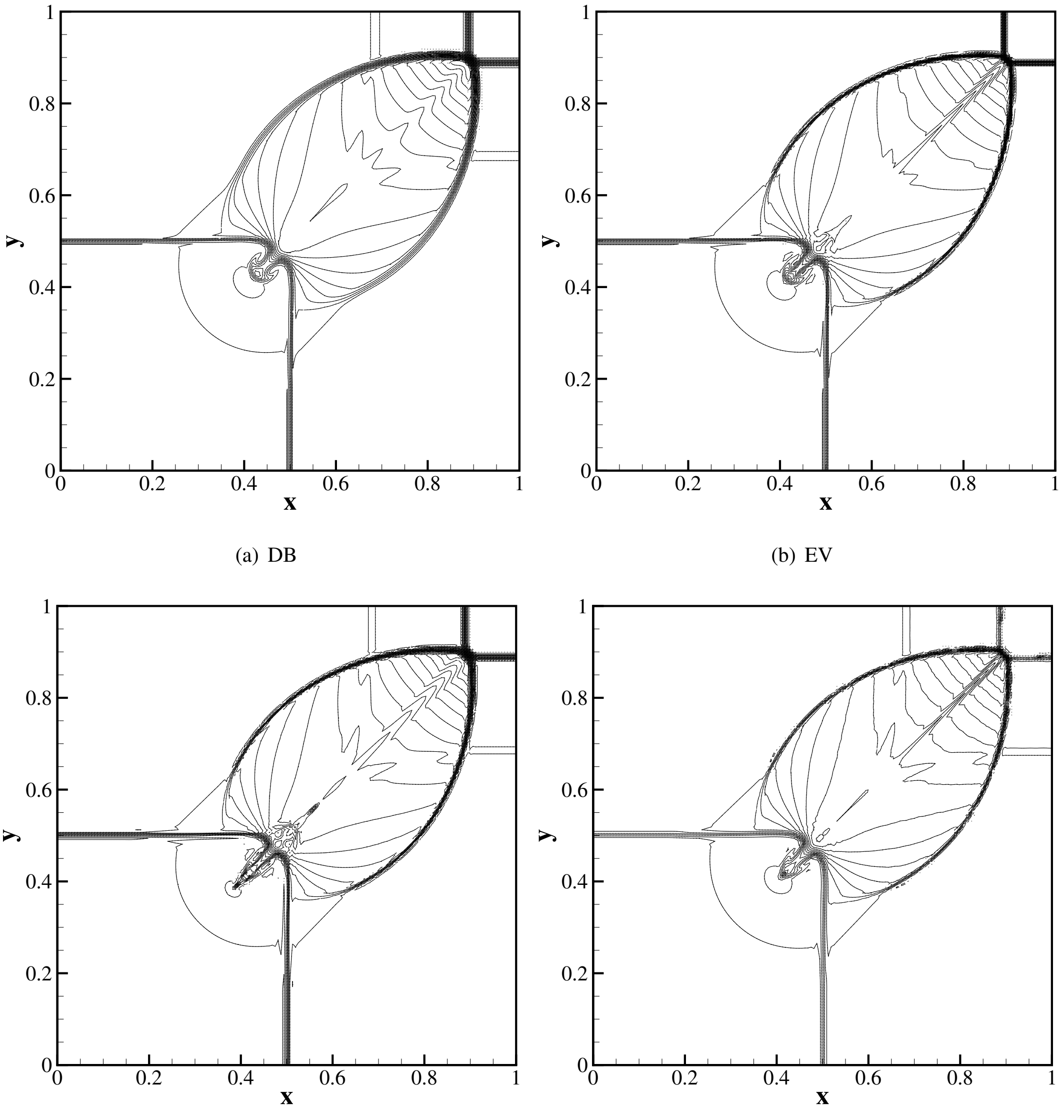
4.5. Double Mach problem

This case consists of a Mach 10 shock moving along and interacting with a solid wall. The computational domain is $[0, 4] \times [0, 1]$, in which the left boundary is set to be the post condition of the shock, and the right boundary is extrapolation. The initial shock makes a 60 angle with the x-direction and intersects with the x-axis at $x = 1/6$. The upper boundary is set to describe the motion of the Mach 10 shock. At $x \in [1/6, 4]$ of the bottom boundary is an inviscid wall, and the rest is set as the post shock condition. The shock moves rightwards and develops complex structures due to the interaction with the wall. The grid is obtained through dividing each quadrilateral element of a uniform structured grid into two triangles. We present the results for the density on a grid of $816 \times 204 \times 2$ in Figs. 20 to 22. The WENO limiting model is less dissipative and more oscillatory than the viscosity models under the same conditions. The DB model is able to achieve a reasonable balance between accuracy and stability. For the MDH model, the situation is similar to the two-dimensional Riemann problems, i.e. excessive dissipation is observed for $P2$ and comparable solution to DB can be obtained for $P1$ and $P4$. The MDA model is the least dissipative among all the viscosity models. The situation is a bit more complicated for EV, which shows excessive dissipation for contact discontinuities despite slightly sharper shock resolution.

To further compare the characteristics of different models, we plot the logarithm of the viscosity distribution and the troubled-cell indicator at the final time for typical cases in Fig. 23. All methods are able to capture the complex flow structures. The EV model generates more viscosity in terms of both value and action region for contact lines, causing less roll-up compared with the other models. The DB model generates the least dissiapation for contact discontinuities due to its dependency on dilation. The MDH and MDA models are similar, with much narrower action region. Similarly to EV, the WENO limiting flags the entire roll-up region as troubled cells, but is able to maintain low dissipation thanks to its high order reconstruction. Also, we give the cell-averaged solution for typical cases in Fig. 24. As can be seen, the cell-averaged solution is smoother, indicating the oscillations shown in the previous figures reside at the element level.

4.6. Forward step problem

This models a supersonic flow of Mach 3 over a step in a wind tunnel. A sample grid employed for this case is given in Fig. 25. The grid is fully unstructured with a uniform distribution of points along the edges. The freestream enters the domain from the left and the right boundary is set to be extrapolation. No special treatment for either the numerical methods or the grid is applied around the step corner. Note that for this case, we adjust the parameters to $c_E = 0.25, c_{\max} = 0.25$ to obtain a stable result for the EV model. We plot the density results for different models in Figs. 26 to 28 at $t = 3$. As can be seen, all models generate noticable entropy layers along the bottom wall due to the singularity around the corner. The DG method in this work reduces to the Legendre-Gauss-Lobatto points along the element edges, in which case the boundary condtion would be $\boldsymbol{u} = 0$ for the corner point as mentioned in [25], therefore generating strong entropy layers. For further validation of our results, we invoke a comparision with [27], in which Lobatto points are used for flux evaluation and the results therein for the forward step case are very similar to ours. On the other hand, using a method with Gauss quadrature points containing no end point for flux on the element



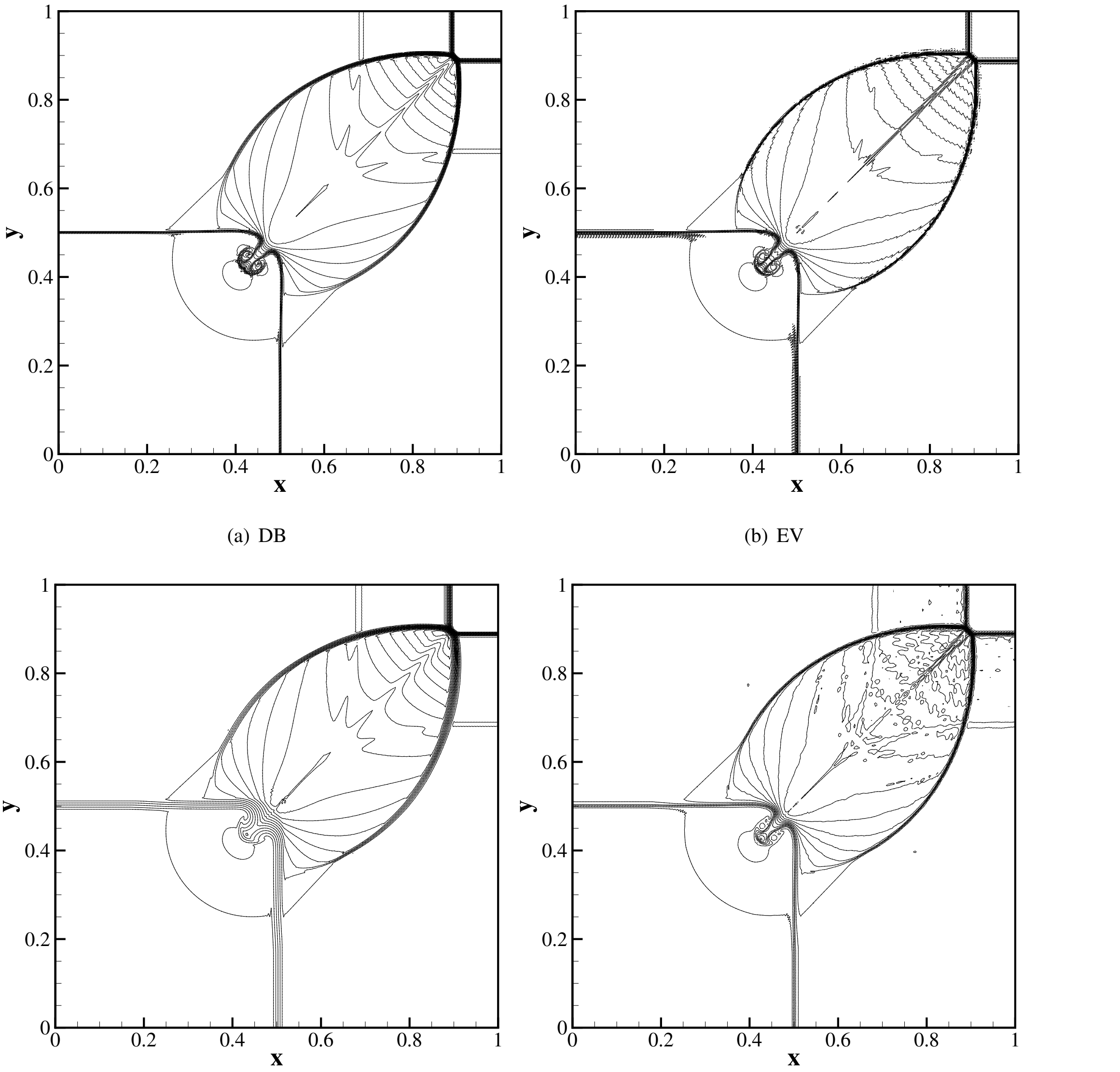
(a) DB

(b) EV

(c) MDH

(d) WENO

Fig. 17. Density solution for the two-dimensional Riemann problem (Case 12) with 160 × 160 × 2 $P1$ elements at $t = 0.25$. Thirty equally spaced contours from 0.515 to 1.665.



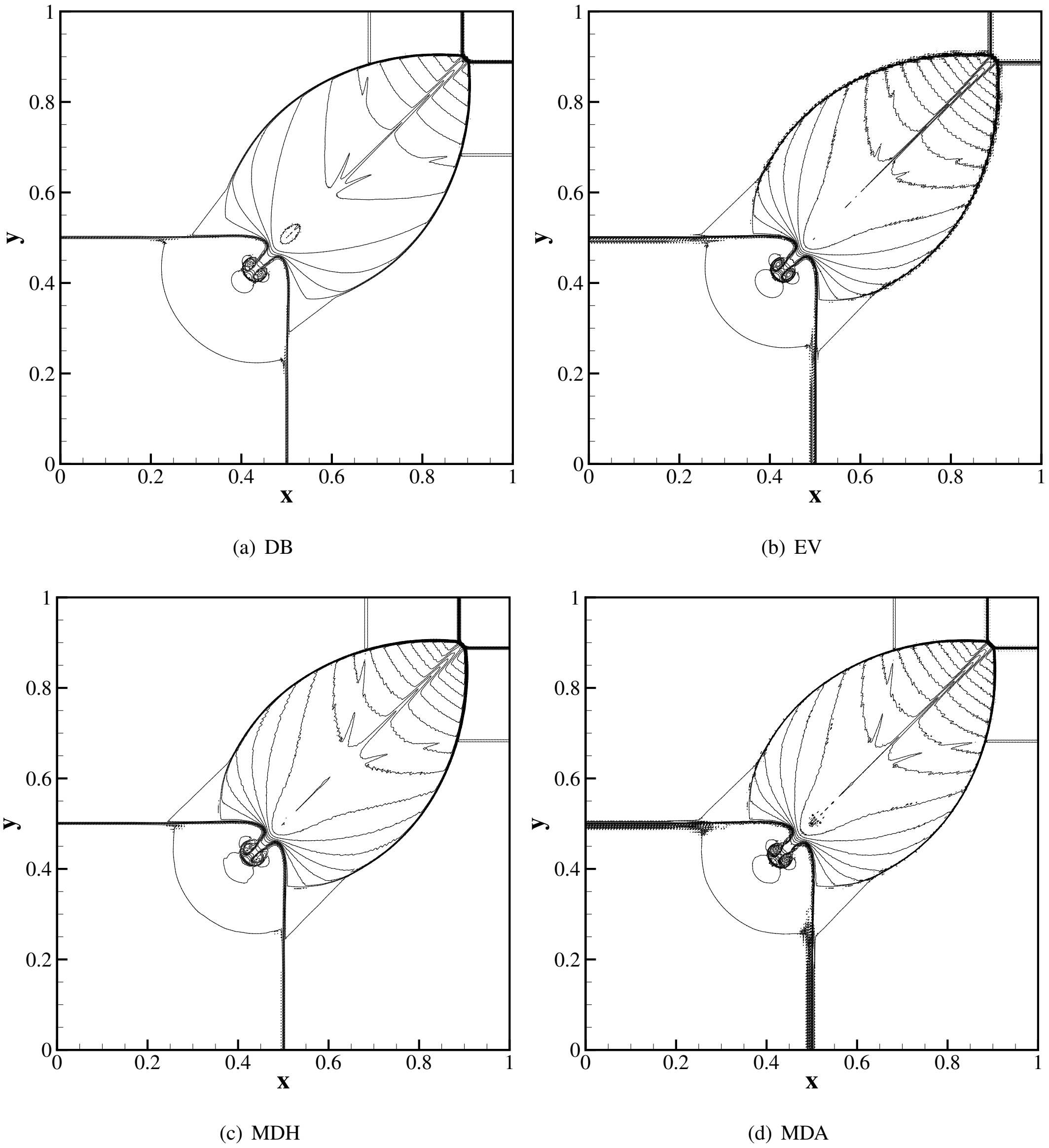
(a) DB

(b) EV

(c) MDH

(d) WENO

Fig. 18. Density solution for the two-dimensional Riemann problem (Case 12) with 160 × 160 × 2 $P2$ elements at $t = 0.25$. Thirty equally spaced contours from 0.515 to 1.665.



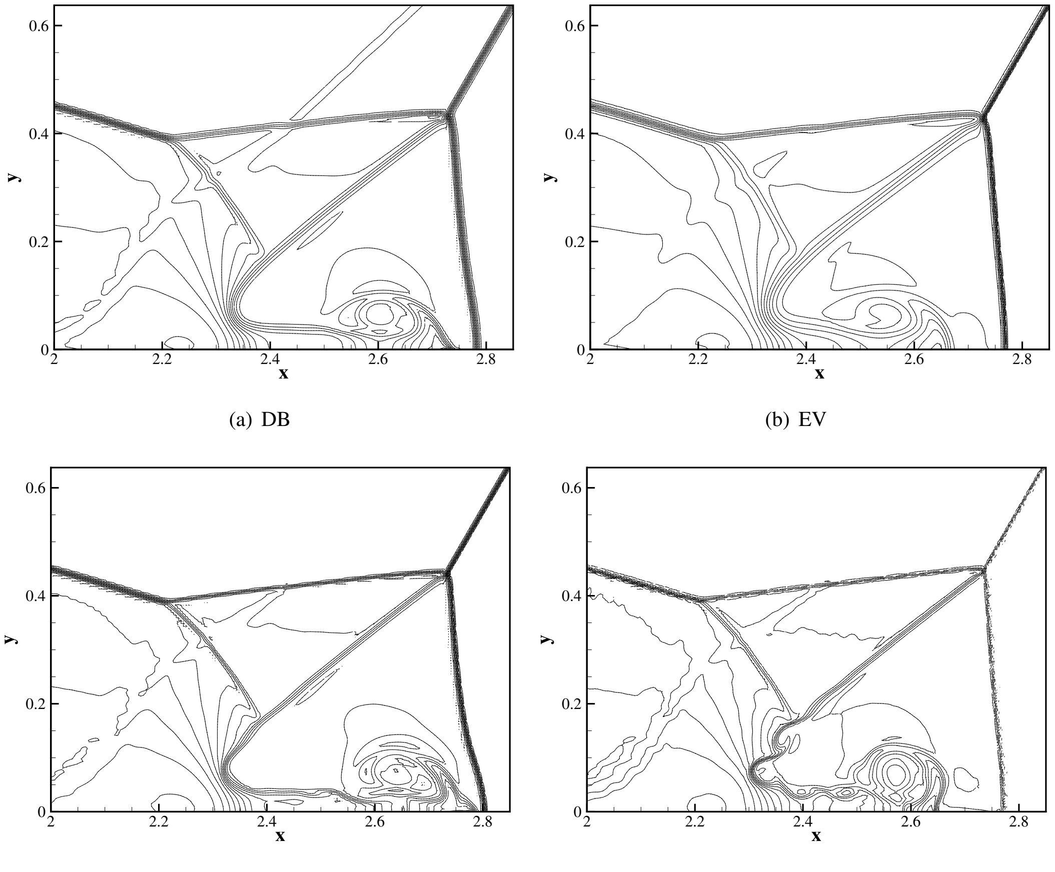
(a) DB

(b) EV

(c) MDH

(d) MDA

Fig. 19. Density solution for the two-dimensional Riemann problem (Case 12) with 160 × 160 × 2 $P4$ elements at t = 0.25. Thirty equally spaced contours from 0.515 to 1.665.



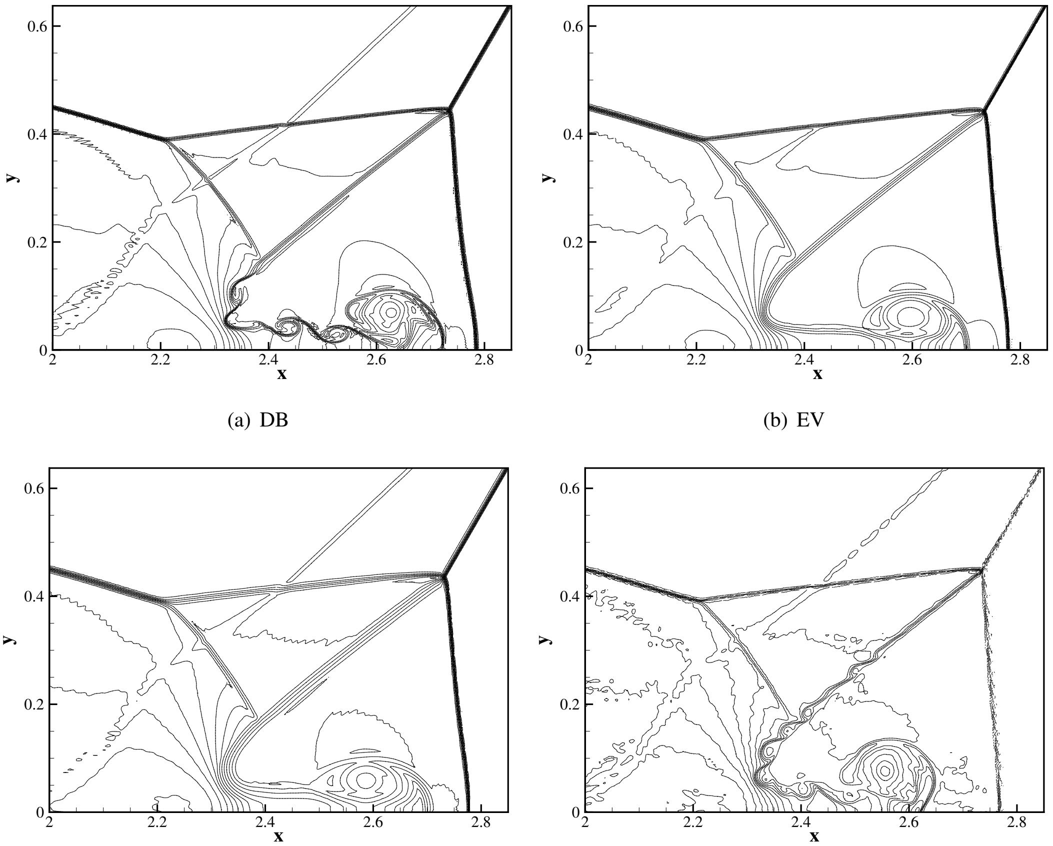
(a) DB

(b) EV

(c) MDH

(d) WENO

Fig. 20. Density solution for the Double Mach problem with 816 × 204 × 2 $P1$ elements at $t = 0.2$. Thirty equally spaced contours from 1.85 to 22.69.



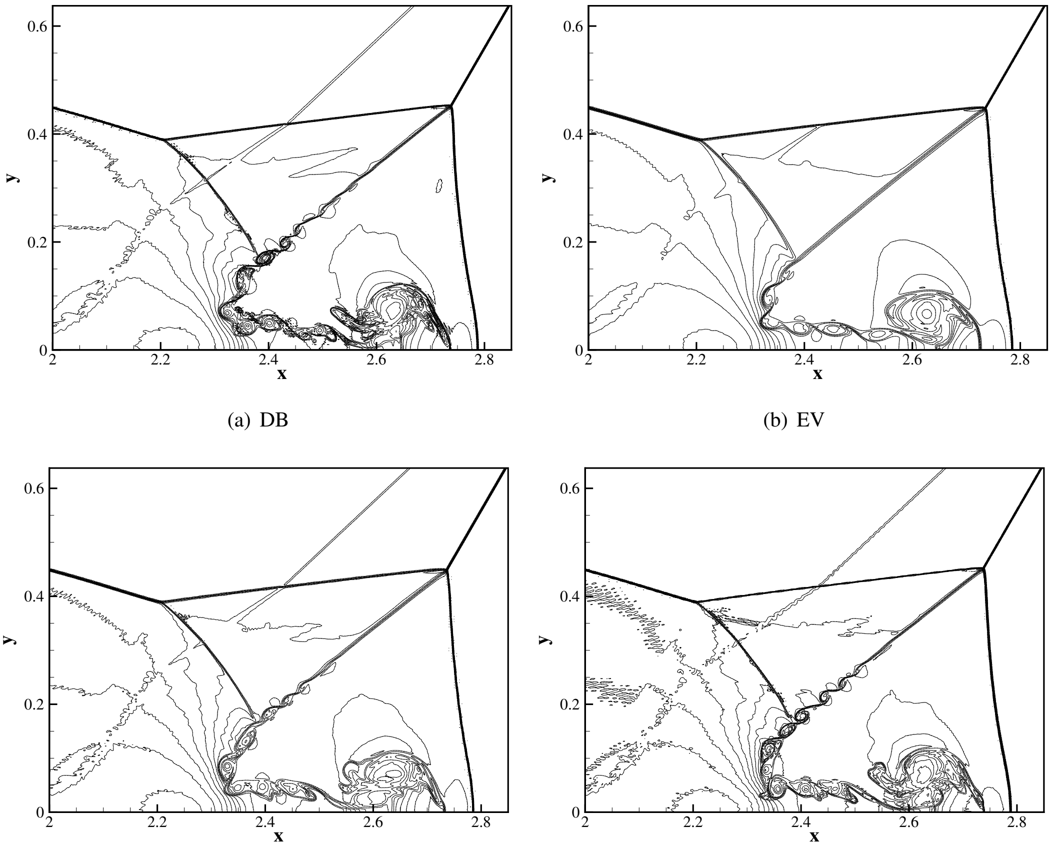
(a) DB

(b) EV

(c) MDH

(d) WENO

Fig. 21. Density solution for the Double Mach problem with $816 \times 204 \times 2$ $P2$ elements at $t = 0.2$. Thirty equally spaced contours from 1.85 to 22.69.



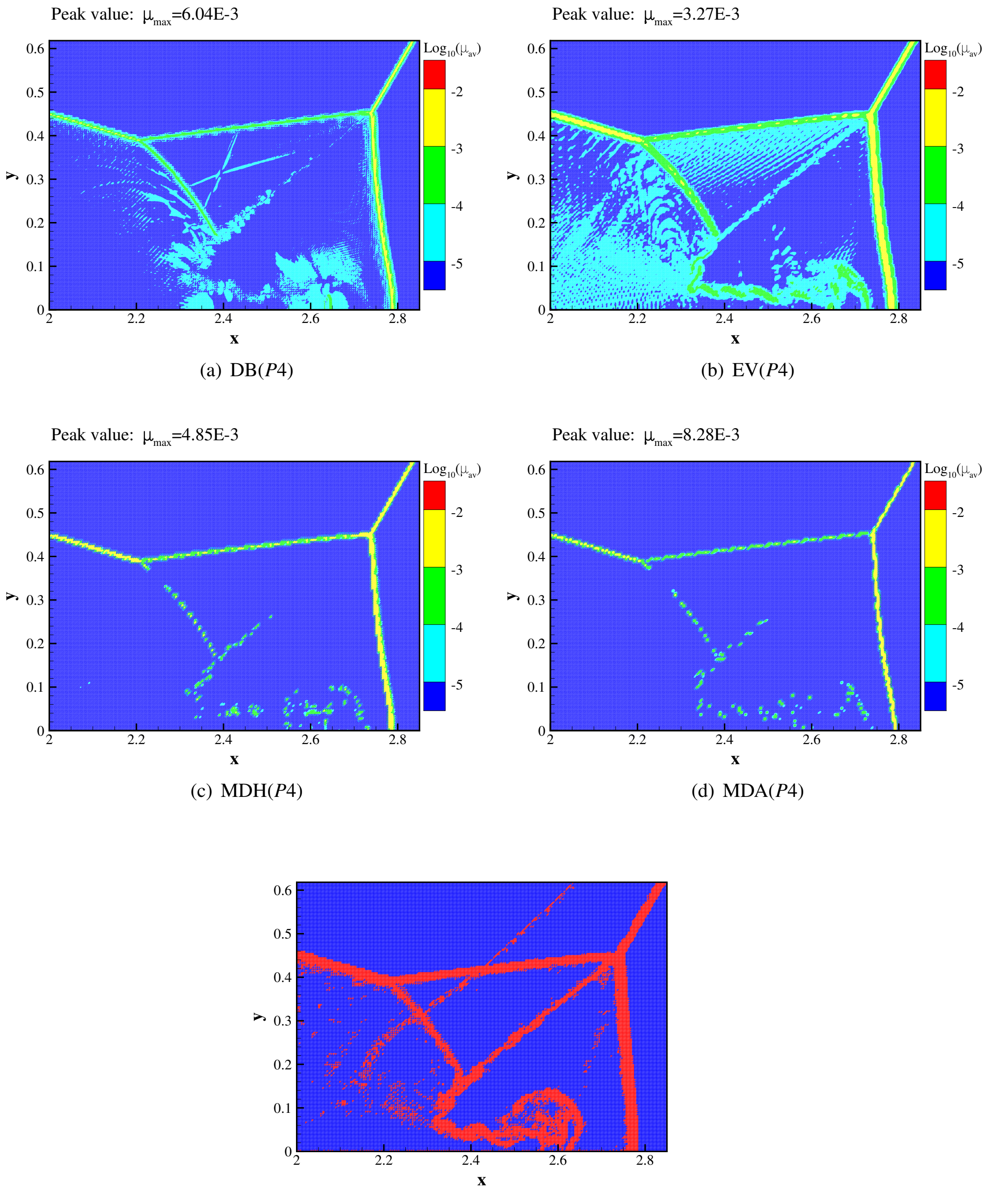
(a) DB

(b) EV

(c) MDH

(d) MDA

Fig. 22. Density solution for the Double Mach problem with $816 \times 204 \times 2$ $P4$ elements at $t = 0.2$. Thirty equally spaced contours from 1.85 to 22.69.



(a) DB(*P*4)

(b) EV(*P*4)

(c) MDH(*P*4)

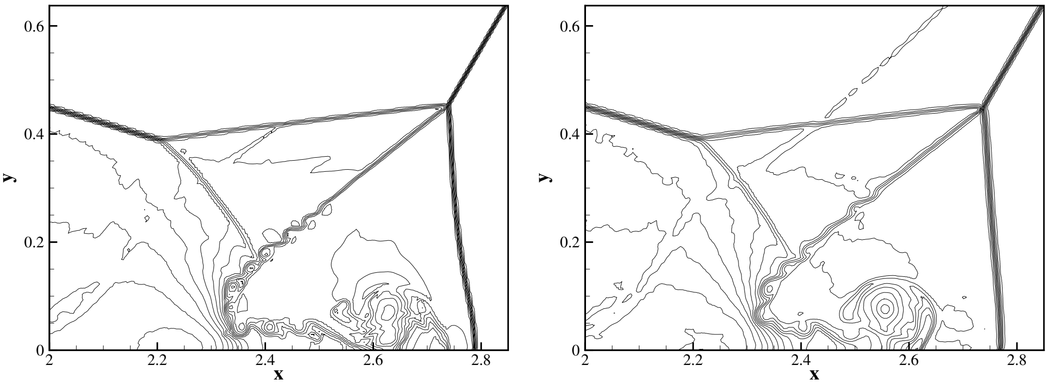
(d) MDA(*P*4)

(e) WENO(*P*2)

Fig. 23. Artificial viscosity and troubled-cell indicator for the Double Mach problem with 816 × 204 × 2 elements at $t = 0.2$.

edge may relieve this situation. However, we stress that in our case the results still suffice for a valid comparison of different models.

For the comparison, we observe that WENO gives the most oscillatory results for this case under the same grid and order. For lower orders ($P1-P2$), DB is more dissipative than EV in the development of the contact discontinuity, while the contrary is true for higher orders($P4$). The MDH model is again observed to show unusually large dissipation for $P2$, and otherwise gives similar results to the other viscosity models. The MDA model is still the least dissipative among all the viscosity models under the same condition. We also plot the logarithm of the viscosity and the troubled-cell indicator in Fig. 29. All models resolve the complex discontinuity structures well. Due to its dependency on velocity dilation, DB generates noticable viscosity around expansion waves. EV generates more significant viscosity around the contact discontinuities than the other viscosity models. The action regions of the viscosity for MDH and MDA are localized compared to the others. The distribution of troubled-cells in the WENO limiting agrees well with those of the artificial viscosity models.



(a) MDA($P4$)

(b) WENO($P2$)

Fig. 24. Cell-averaged density solution for the Double Mach problem with 816 × 204 × 2 elements at $t = 0.2$. Thirty equally spaced contours from 1.85 to 22.69.

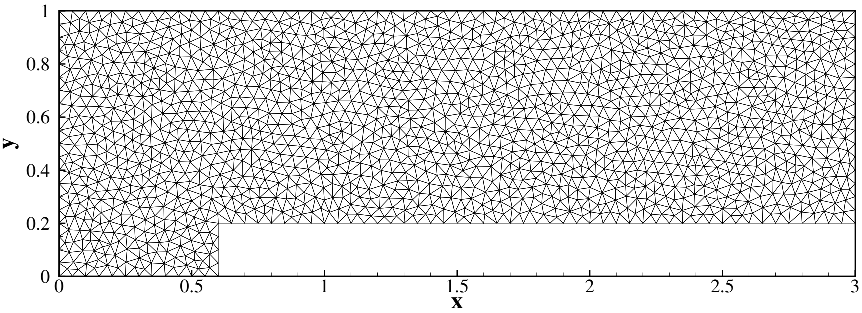
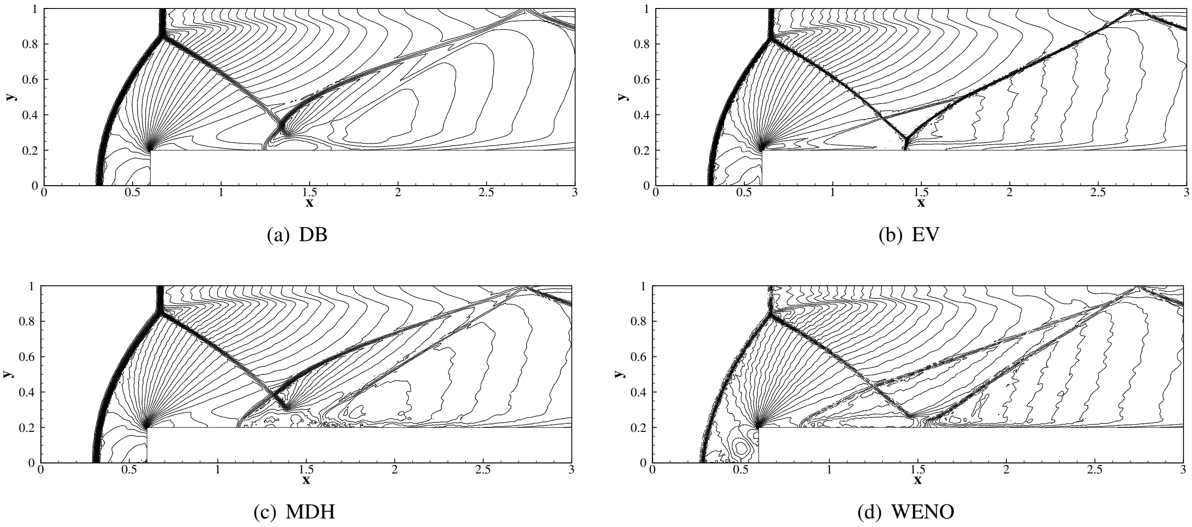


Fig. 25. A sample grid for the forward step problem($h = 1/20$).



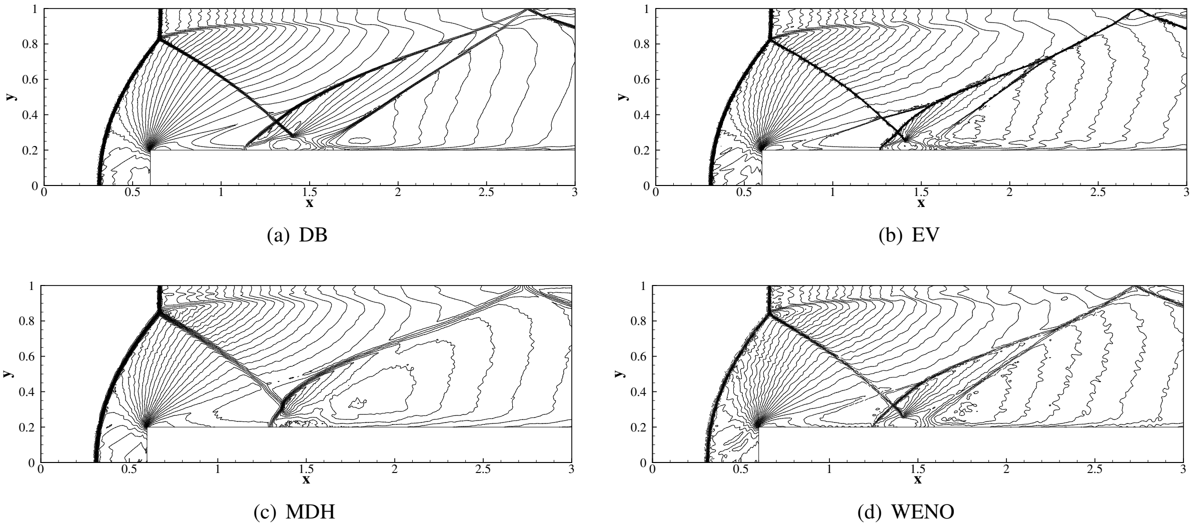
(a) DB

(b) EV

(c) MDH

(d) WENO

Fig. 26. Density solution for the forward step problem with 50486 $P1$ elements ($h = 1/80$) at $t = 3$. Thirty equally spaced contours from 0.42 to 6.466.



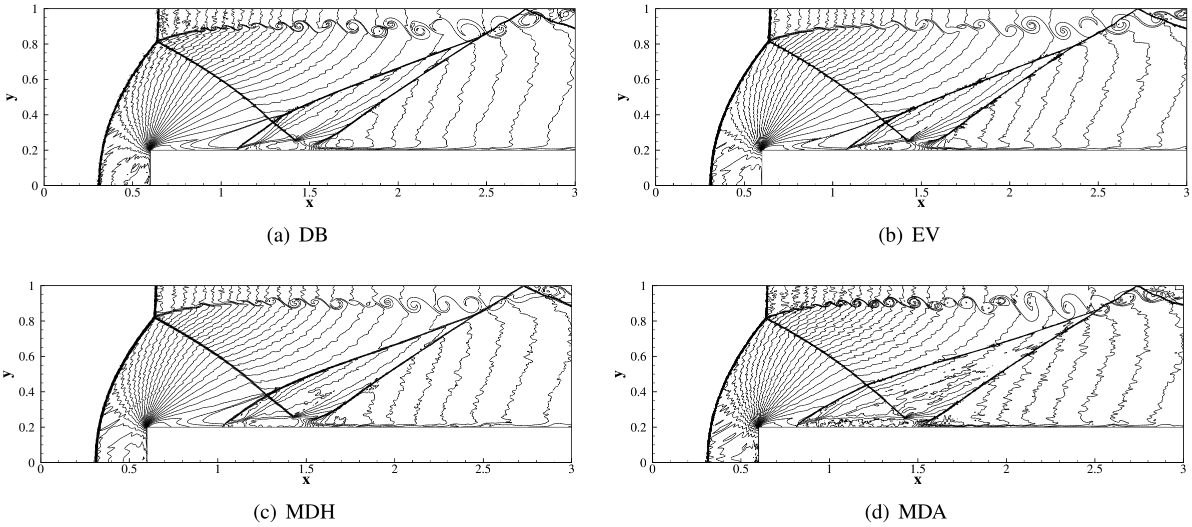
(a) DB

(b) EV

(c) MDH

(d) WENO

Fig. 27. Density solution for the forward step problem with 50486 $P2$ elements ($h = 1/80$) at $t = 3$. Thirty equally spaced contours from 0.42 to 6.466.



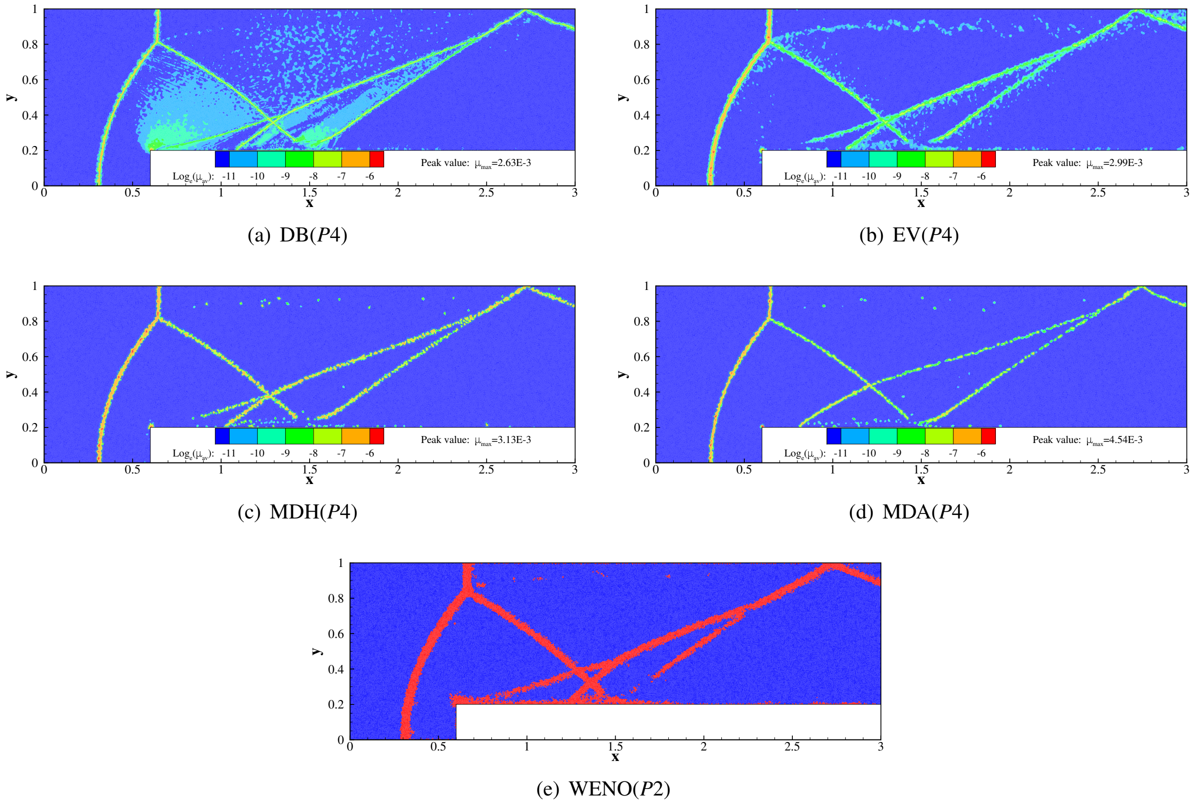
(a) DB

(b) EV

(c) MDH

(d) MDA

Fig. 28. Density solution for the forward step problem with 50486 $P4$ elements ($h = 1/80$) at $t = 3$. Thirty equally spaced contours from 0.42 to 6.466.



(a) DB($P4$)

(b) EV($P4$)

(c) MDH($P4$)

(d) MDA($P4$)

(e) WENO($P2$)

Fig. 29. Artificial viscosity and troubled-cell indicator for the forward step problem with 50486 elements ($h = 1/80$) at $t = 3$.

5. Conclusion

In this work, we conduct a comparative study of various shock capturing models for DG in terms of accuracy and robustness. Five typical models are selected to gain insights into their performance, including the DB model, the EV model, the MDH model, the MDA model and the WENO limiting method. A wide range of benchmark cases are employed for comparison comprising both smooth and unsmooth problems. One of the major challenges for the shock capturing models is their relatively strong dependency on empirical parameters, which one generally needs to tune for optimal results. In this work, we seek baseline parameters for each model and reduce the adjustment of these parameters to a minimum level among different cases. The behaviors of different models are compared in a qualitative fashion, weakly impacted by different choices of the empirical parameters.

For the smooth cases, convergence tests are performed to evaluate the accuracy of the shock capturing models for smooth flows. The DB model is limited to second order accuracy theoretically and may recover some accuracy for flow regions of weak compressibility. The WENO method is able to preserve the theoretical order of accuracy regardless of its action region, while the MDH models relies on entirely switching off the shock sensor on sufficiently refined grids to recover the original accuracy. The MDA model is similar to MDH, execpt that MDA switches off the viscosity more rapidily, thus causing less dissipation. The EV model is able to maintain the original order of accuracy even with the viscosity activated.

For non-smooth flows, all the models considered in this work are able to capture discontinuous structures well despite their noticable difference. The DB model can capture shocks and complex small scales with good resolution and has an sub-cell property, which makes it attractive for higher orders on coarse grids. For 2D however, smoothing is required by DB for cases of strong shocks, consequently affecting its advantage at subcell resolution. The EV model dissipates contact discontinuites more than the other viscosity models, thus frequentlly showing more dissipative results for complex flow structures. The MDA model shows very restricted action region of artificial viscosity, generally producing the least dissipative results. The MDH model is similar to MDA, except MDH's oscillatory feature in artificial viscosity, i.e. MDH may exhibit inconsistent behavior with the same set of parameters for P increasing and it has a chance of yielding unusually diffusive results for certain orders. All the viscosity models are observed to work better at higher orders, while the WENO limiting is able to achieve less dissipative results for lower orders than the viscosity models. However, it is generally difficult to apply the present WENO method to higher orders, probably due to the larger extrapolation error of the polynomials of higher orders. In terms of parameter tuning, DB, MDA and WENO are more robust than the others since they are able to generate reasonable results for a wide range of cases with the same set of parameters. On the contrary, MDH and EV require more attention on the adjustment of the parameters for different cases.

Acknowledgments

The authors thank Prof. Tim Warburton and Dr. David Medina for helpful discussions on the DG code. Yu would like to acknowledge support by the National Natural Science Foundation of China (Grant No. 11402016) and by the China Scholarship Council (Grant No. 201606025011). All the simulations are conducted on the high performance computing clusters of SCITAS at EPFL.

References

[1] N. Kroll, ADIGMA: A European project on the development of adaptive higher order variational methods for aerospace applications, in: 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition, 2009, p. 176.

[2] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, et al., High-order CFD methods: current status and perspective, International Journal for Numerical Methods in Fluids 72 (8) (2013) 811–845.

[3] J. S. Hesthaven, T. Warburton, Nodal discontinuous Galerkin methods: algorithms, analysis, and applications, Springer Science & Business Media, 2007.

[4] B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems, Journal of Computational Physics 84 (1) (1989) 90–113.

[5] B. Cockburn, C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems, Journal of Computational Physics 141 (2) (1998) 199–224.

[6] A. Burbeau, P. Sagaut, C.-H. Bruneau, A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods, Journal of Computational Physics 169 (1) (2001) 111–150.

[7] L. Krivodonova, Limiters for high-order discontinuous Galerkin methods, Journal of Computational Physics 226 (1) (2007) 879–896.

[8] J. Qiu, C.-W. Shu, Runge-Kutta discontinuous Galerkin method using WENO limiters, SIAM Journal on Scientific Computing 26 (3) (2005) 907–929.

[9] J. Zhu, J. Qiu, C.-W. Shu, M. Dumbser, Runge-Kutta discontinuous Galerkin method using WENO limiters II: Unstructured meshes, Journal of Computational Physics 227 (9) (2008) 4330–4353.

[10] J. Qiu, C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: One-dimensional case, Journal of Computational Physics 193 (1) (2004) 115–135.

[11] J. Qiu, C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: Two dimensional case, Computers & Fluids 34 (6) (2005) 642–663.

[12] J. Zhu, J. Qiu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method, III: Unstructured meshes, Journal of Scientific Computing 39 (2) (2009) 293–321.

[13] H. Luo, J. D. Baum, R. Löhner, A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids, Journal of Computational Physics 225 (1) (2007) 686–713.

[14] X. Zhong, C.-W. Shu, A simple weighted essentially nonoscillatory limiter for Runge-Kutta discontinuous Galerkin methods, Journal of Computational Physics 232 (1) (2013) 397–415.

[15] J. Zhu, X. Zhong, C.-W. Shu, J. Qiu, Runge-Kutta discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes, Journal of Computational Physics 248 (2013) 200–220.

[16] A. W. Cook, W. H. Cabot, A high-wavenumber viscosity for high-resolution numerical methods, Journal of Computational Physics 195 (2) (2004) 594–601.

[17] A. W. Cook, W. H. Cabot, Hyperviscosity for shock-turbulence interactions, Journal of Computational Physics 203 (2) (2005) 379–385.

[18] B. Fiorina, S. K. Lele, An artificial nonlinear diffusivity method for supersonic reacting flows with shocks, Journal of Computational Physics 222 (1) (2007) 246–264.

[19] S. Kawai, S. K. Lele, Localized artificial diffusivity scheme for discontinuity capturing on curvilinear meshes, Journal of Computational Physics 227 (22) (2008) 9498–9526.

[20] A. Mani, J. Larsson, P. Moin, Suitability of artificial bulk viscosity for large-eddy simulation of turbulent flows with shocks, Journal of Computational Physics 228 (19) (2009) 7368–7374.

[21] S. Premasuthan, C. Liang, A. Jameson, Computation of flows with shocks using the spectral difference method with artificial viscosity, I: Basic formulation and application, Computers & Fluids 98 (2014) 111–121.

[22] D. Moro, N. C. Nguyen, J. Peraire, Dilation-based shock capturing for high-order methods, International Journal for Numerical Methods in Fluids 82 (7) (2016) 398–416.

[23] P.-O. Persson, J. Peraire, Sub-cell shock capturing for discontinuous Galerkin methods, in: 44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA paper 2006-112, 2006.

[24] A. Klöckner, T. Warburton, J. S. Hesthaven, Viscous shock capturing in a time-explicit discontinuous Galerkin method, Mathematical Modelling of Natural Phenomena 6 (3) (2011) 57–83.

[25] J.-L. Guermond, R. Pasquetti, B. Popov, Entropy viscosity method for nonlinear conservation laws, Journal of Computational Physics 230 (11) (2011) 4248–4267.

[26] V. Zingan, J.-L. Guermond, J. Morel, B. Popov, Implementation of the entropy viscosity method with the discontinuous Galerkin method, Computer Methods in Applied Mechanics and Engineering 253 (2013) 479–490.

[27] A. Chaudhuri, G. B. Jacobs, W.-S. Don, H. Abbassi, F. Mashayek, Explicit discontinuous spectral element method with entropy generation based artificial viscosity for shocked viscous flows, Journal of Computational Physics 332 (2017) 99–117.

[28] R. Hartmann, Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier-Stokes equations, International Journal for Numerical Methods in Fluids 51 (9-10) (2006) 1131–1156.

[29] F. Bassi, A. Crivellini, A. Ghidoni, S. Rebay, High-order discontinuous Galerkin discretization of transonic turbulent flows, in: 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition, AIAA paper 2009-180, 2009.

[30] J. Qiu, C.-W. Shu, A comparison of troubled-cell indicators for Runge-Kutta discontinuous Galerkin methods using weighted essentially nonoscillatory limiters, SIAM Journal on Scientific Computing 27 (3) (2005) 995–1013.

[31] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, J. E. Flaherty, Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws, Applied Numerical Mathematics 48 (3-4) (2004) 323–338.

[32] J. S. Hesthaven, Numerical methods for conservation laws: From analysis to algorithms, SIAM Publishing, 2017.

[33] G. E. Barter, D. L. Darmofal, Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation, Journal of Computational Physics 229 (5) (2010) 1810–1827.

[34] H. Abbassi, F. Mashayek, G. B. Jacobs, Shock capturing with entropy-based artificial viscosity for staggered grid discontinuous spectral element method, Computers & Fluids 98 (2014) 152–163.

[35] M. H. Carpenter, C. A. Kennedy, Fourth-order 2N-storage Runge-Kutta schemes, NASA Report TM 109112, NASA Langley Research Center (1994).

[36] A. Kurganov, E. Tadmor, Solution of two-dimensional riemann problems for gas dynamics without riemann problem solvers, Numerical Methods for Partial Differential Equations 18 (5) (2002) 584–608.