



U S, V., Zang, B., & New, T. H. (2021). A carbuncle cure for the Harten-Lax-van Leer contact (HLLC) scheme using a novel velocity-based sensor. *Applied Mathematics and Mechanics*, 42(9), 1259-1278. <https://doi.org/10.1007/s10483-021-2762-6>

Publisher's PDF, also known as Version of record

License (if available):  
CC BY

Link to published version (if available):  
[10.1007/s10483-021-2762-6](https://doi.org/10.1007/s10483-021-2762-6)

[Link to publication record in Explore Bristol Research](#)  
PDF-document

This is the final published version of the article (version of record). It first appeared online via Springer at <https://doi.org/10.1007/s10483-021-2762-6> .Please refer to any applicable terms of use of the publisher.

## University of Bristol - Explore Bristol Research

### General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available: <http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

## A carbuncle cure for the Harten-Lax-van Leer contact (HLLC) scheme using a novel velocity-based sensor\*

U. S. VEVEK<sup>1</sup>, B. ZANG<sup>2</sup>, T. H. NEW<sup>3,†</sup>

1. Department of Mechanical, Materials and Aerospace Engineering, University of Liverpool, Liverpool L69 3BX, United Kingdom;
  2. Department of Aerospace Engineering, University of Bristol, Bristol BS8 1TR, United Kingdom;
  3. School of Mechanical and Aerospace Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore
- (Received Mar. 20, 2021 / Revised Jun. 7, 2021)

**Abstract** A hybrid numerical flux scheme is proposed by adapting the carbuncle-free modified Harten-Lax-van Leer contact (HLLCM) scheme to smoothly revert to the Harten-Lax-van Leer contact (HLLC) scheme in regions of shear. This hybrid scheme, referred to as the HLLCT scheme, employs a novel, velocity-based shear sensor. In contrast to the non-local pressure-based shock sensors often used in carbuncle cures, the proposed shear sensor can be computed in a localized manner meaning that the HLLCT scheme can be easily introduced into existing codes without having to implement additional data structures. Through numerical experiments, it is shown that the HLLCT scheme is able to resolve shear layers accurately without succumbing to the shock instability.

**Key words** carbuncle, shock instability, Euler equation, numerical flux, Harten-Lax-van Leer contact (HLLC) scheme

**Chinese Library Classification** O241.82, O351.2

**2010 Mathematics Subject Classification** 35Q50, 74Q05, 74M20

### 1 Introduction

Simulation of high-speed, compressible flows requires proper design of inviscid numerical flux functions to capture shock waves, contact discontinuities, and shear layers in a stable and accurate manner. Grounded in the seminal work of Godunov<sup>[1]</sup>, approximate Riemann solvers represent the state-of-the-art technique in designing flux functions whereby the inviscid fluxes are obtained by assuming an approximate solution structure for the Riemann problem across

---

\* Citation: VEVEK, U. S., ZANG, B., and NEW, T. H. A carbuncle cure for the Harten-Lax-van Leer contact (HLLC) scheme using a novel velocity-based sensor. *Applied Mathematics and Mechanics (English Edition)*, **42**(9), 1259–1278 (2021) <https://doi.org/10.1007/s10483-021-2762-6>

† Corresponding author, E-mail: [dthnew@ntu.edu.sg](mailto:dthnew@ntu.edu.sg)

Project supported by the Singapore Ministry of Education AcRF Tier-2 Grant (No. MOE2014-T2-1-002) and the Graduate Research Officer Scholarship from School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore

©The Author(s) 2021

a cell interface. One of the simplest and most robust flux schemes is the Harten-Lax-van Leer (HLL) scheme<sup>[2]</sup> in which an incomplete two-wave solution structure is assumed. However, this incompleteness introduces non-physical dissipation of density and tangential momenta which precludes exact capturing of contact discontinuities and shear waves, resulting in highly diffused material interfaces and shear layers. Hence, it is unsuitable for use in many practical flow scenarios such as boundary layer flows<sup>[3]</sup>. Toro et al.<sup>[4]</sup> proposed a modification to the HLL scheme known as the HLL contact (HLLC) scheme<sup>[4]</sup>, which allowed exact capturing of the contact discontinuities and shear waves. The complete three-wave HLLC scheme has become one of the most widely used numerical flux schemes owing to its simplicity and accuracy. However, being a contact-/shear-preserving flux scheme, the HLLC scheme invariably succumbs to a multi-dimensional shock instability problem.

Quirk<sup>[5]</sup> presented the first systematic analysis for failings of approximate Riemann solvers, including the various manifestations of the shock instability problem such as carbuncle phenomena, kinked Mach stems, and odd-even decoupling. He hypothesized that strong, grid-aligned shocks provide a ‘systematic perturbation’ to the pressure field and shock-unstable flux schemes feed the pressure perturbations into the density field causing density perturbations to grow unbounded culminating in the instability. Pandolfi and D’Ambrosio<sup>[6]</sup>, through odd-even decoupling analyses, confirmed that flux schemes which account for contact surfaces explicitly are susceptible to shock instability. Xu and Li<sup>[7]</sup> mentioned that the shear-preserving ability of flux schemes provides insufficient dissipation along the shock front which leads to shock instability.

The conventional approach to overcoming the shock instability problem is to introduce additional dissipation in the vicinity of the shocks, usually by switching to a complementary shock-stable scheme. There have been several such attempts to hybridize the HLLC scheme in the recent years<sup>[8–12]</sup>. With the exception of the work of Huang et al.<sup>[11]</sup> who used the rotated Riemann solver formulation<sup>[13]</sup> with a localized velocity-based sensor, the remaining studies relied on some form of pressure-based shock sensor computed in a non-localized manner about the concerned cell interface. Among these studies, the work of Shen et al.<sup>[10]</sup> is unique and particularly illuminating. They designed a modified HLLC scheme called HLLCM (where ‘M’ refers to the modification) based on the Rankine-Hugoniot jump conditions which preserves contact waves but smears shear waves. In this respect, one may view it as a ‘two-and-a-half-wave’ Riemann solver with shear velocity dissipation across the contact wave. Using matrix stability analysis<sup>[14]</sup>, they showed that the HLLCM scheme is shock-stable. Then, they proposed a hybrid HLLC-HLLCM scheme by using a pressure-based shock sensor as mentioned before and demonstrated the effectiveness of the hybrid scheme for a variety of test cases.

Unlike the more common HLLC-HLL hybrids, the HLLC-HLLCM hybrid scheme uses a contact-preserving complementary scheme. This presents a unique opportunity to explore a radically new idea: to deactivate dissipation across shear layers rather than injecting additional dissipation along shock fronts. In other words, the HLLCM scheme is viewed as the base scheme which is applied everywhere except near shear layers where the HLLC scheme is used to resolve the shear layers accurately. The main objective of the present work is to explore the viability of this approach. In this approach, the role of the sensor now becomes one of determining regions of shear flow. To this end, a velocity-based sensor is introduced in this paper. This sensor can be computed in a localized manner as opposed to the non-local shock sensors used in most of the carbuncle cures reported in past studies<sup>[6,8–10,12,15–18]</sup>. The paper is organized as follows. The governing equations and discretization schemes are introduced in Section 2. The derivations of the HLL, HLLC, and HLLCM approximate Riemann solvers are presented in Section 3 along with suitable wave-speed estimations. The new localized, velocity-based sensor is proposed and discussed in Section 4. The results of a variety of numerical experiments are presented in Section 5. Finally, the concluding remarks are provided in Section 6.

## 2 Preliminaries

### 2.1 Governing equations

The equations governing the dynamics of inviscid gas flow are given as

$$\partial_t \mathbf{Q} + \nabla \cdot F(\mathbf{Q}) = \mathbf{0}, \quad (1)$$

where the state vector  $\mathbf{Q}$  and flux vector  $F(\mathbf{Q})$  are defined by

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho \mathbf{u} \\ \rho E \end{pmatrix}, \quad F(\mathbf{Q}) = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \mathbf{u} + p \mathbf{I} \\ \rho E \mathbf{u} + p \mathbf{u} \end{pmatrix}, \quad (2)$$

in which  $\rho$  is the gas density,  $\mathbf{u}$  is the gas velocity,  $p$  is the gas pressure, and  $E$  is the specific total energy. The total energy comprises of the internal energy  $e$  and the kinetic energy  $k$ , which can be related to the other state variables through the ideal gas equation of state as follows:

$$E = e + k = \frac{p}{(\gamma - 1)\rho} + \frac{\mathbf{u} \cdot \mathbf{u}}{2}, \quad (3)$$

where  $\gamma$  is the ratio of specific heat capacities taken to be 1.4 in the present study.

### 2.2 Spatial and temporal discretization

In finite volume methodology (FVM), the domain is discretized into non-overlapping cells, and the solution is obtained in terms of the cell averages. The cell average of a function  $\phi(\mathbf{x})$  over the  $i$ th cell  $\Omega_i$  is given by

$$\bar{\phi}_i \equiv \frac{1}{|\Omega_i|} \iiint_{\Omega_i} \phi(\mathbf{x}) dV, \quad (4)$$

where  $|\Omega_i|$  refers to the volume of the cell. Applying the averaging operation to Eq. (1) and transforming the volume integral of the divergence term into a surface integral using the Gauss theorem, as is customary in FVM, yields an ordinary differential equation (ODE) for  $\bar{\mathbf{Q}}_i$  as follows:

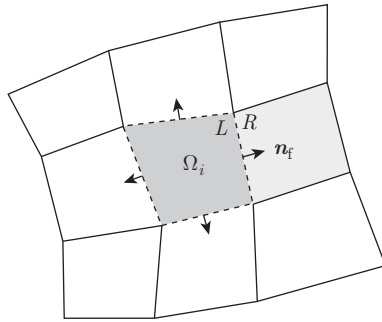
$$\frac{1}{|\Omega_i|} \iiint_{\Omega_i} (\partial_t \mathbf{Q} + \nabla \cdot F(\mathbf{Q})) dV = \mathbf{0} \Rightarrow \frac{d\bar{\mathbf{Q}}_i}{dt} = -\frac{1}{|\Omega_i|} \oint_{\partial\Omega_i} F(\mathbf{Q}) \cdot d\mathbf{S}, \quad (5)$$

where  $\partial\Omega_i$  refers to the boundary of cell  $\Omega_i$ . For a polyhedral cell, the surface integral on the right-hand side of Eq. (5) can be approximated as a summation over each of the planar faces that make up  $\partial\Omega_i$  as follows:

$$\oint_{\partial\Omega_i} F(\mathbf{Q}) \cdot d\mathbf{S} \approx \sum_{f \in \partial\Omega_i} \hat{F}(\mathbf{Q}_{f,L}, \mathbf{Q}_{f,R}, \mathbf{n}_f) S_f. \quad (6)$$

Note that the physical flux  $F$  has been replaced by a numerical flux  $\hat{F}$  in the above approximation. The numerical flux is evaluated at the face centroids using the left- and right-based interpolated state variables  $\mathbf{Q}_{f,L}$  and  $\mathbf{Q}_{f,R}$ . The left ( $L$ ) and right ( $R$ ) sides of a face are defined with respect to the face unit normal vectors  $\mathbf{n}_f$  which point out of  $\Omega_i$  as shown in Fig. 1. Lastly,  $S_f$  is the face surface area.

In the present study, we restrict to the first-order upwind scheme for spatial interpolation since higher order schemes tend to be less susceptible to shock instability problem<sup>[6]</sup>. As for the temporal discretization, the explicit third-order total-variation-diminishing (TVD) Runge-Kutta scheme<sup>[19]</sup> was used for transient problems while an implicit matrix-free lower-upper symmetric Gauss-Seidel (LU-SGS) algorithm<sup>[20]</sup> was used for steady-state problems. With the spatial and temporal schemes defined, we turn our attention to the numerical flux function itself which is the main focus of this paper.



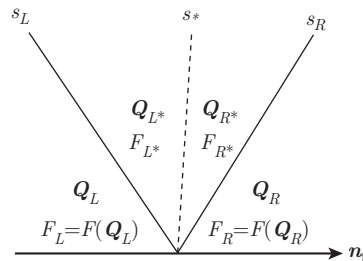
**Fig. 1** FVM applied to cell  $\Omega_i$ , where the black arrows represent the face unit normal vectors  $\mathbf{n}_f$  and the dotted black line is the cell boundary  $\partial\Omega_i$

### 3 Review of HLLC and HLLCM schemes

In the first-order upwind scheme, the variables are assumed to remain constant within each cell, i.e., no spatial variation within the cell. Hence, a discontinuity arises at each cell interface  $f$ . Godunov<sup>[1]</sup> put forth the idea of treating each discontinuity as a Riemann problem and solving this Riemann problem exactly to derive the numerical flux  $\hat{F}$ . Godunov's method is expensive for gas dynamics due to the iterative procedure involved in computing the exact solution structures which may consist of rarefaction fans, contact discontinuities, and shock waves, depending on the initial condition. Moreover, Godunov's method itself is not immune to the shock instability problem. One way to improve the efficiency is to replace the exact solution structure with an approximate one for which it is simpler to compute the intermediate states and flux. These are known as approximate Riemann solvers.

The schematic of a general 3-wave approximate Riemann solver is shown in Fig.2. The approximate solution structure consists of three waves traveling at speeds  $s_L < s_* < s_R$  in the direction normal to the face and two intermediate states  $\mathbf{Q}_{L^*}$  and  $\mathbf{Q}_{R^*}$ . For a clearer understanding of how each flux component is treated by the approximate Riemann solver, the state vector is written in the following form whereby the momentum equation is decomposed into the normal and tangential directions:

$$\mathbf{Q} = \rho(1 \ u \ v \ E)^T. \quad (7)$$



**Fig. 2** Solution structure for a general 3-wave approximate Riemann solver

From hereon, the variables  $u \equiv \mathbf{u} \cdot \mathbf{n}_f$  and  $v \equiv \mathbf{u} - u\mathbf{n}_f$  refer to the normal and tangential velocities, respectively, with respect to the face normal direction.

The approximate solution is required to obey the following Rankine-Hugoniot (RH) jump conditions:

$$F_{L^*} = F(\mathbf{Q}_L) + s_L(\mathbf{Q}_{L^*} - \mathbf{Q}_L), \quad (8a)$$

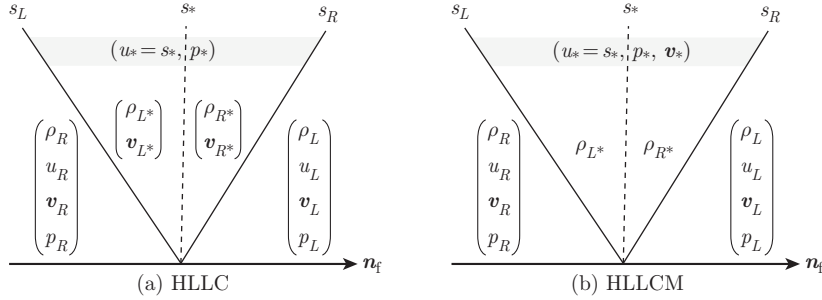
$$F_{R^*} = F(\mathbf{Q}_R) + s_R(\mathbf{Q}_{R^*} - \mathbf{Q}_R), \tag{8b}$$

$$F_{R^*} - F_{L^*} = s_*(\mathbf{Q}_{R^*} - \mathbf{Q}_{L^*}). \tag{8c}$$

Note that the intermediate fluxes  $F_{L^*}$  and  $F_{R^*}$  need not necessarily be equal to  $F(\mathbf{Q}_{L^*})$  and  $F(\mathbf{Q}_{R^*})$ , respectively. Denoting  $F_L = F(\mathbf{Q}_L)$  and  $F_R = F(\mathbf{Q}_R)$  for conciseness, the final numerical flux  $\hat{F}$  is chosen as follows:

$$\hat{F}(\mathbf{Q}_L, \mathbf{Q}_R, \mathbf{n}_f) = \begin{cases} F_L, & s_L \geq 0, \\ F_{L^*}, & s_L < 0 < s_*, \\ F_{R^*}, & s_* < 0 < s_R, \\ F_R, & s_R \leq 0. \end{cases} \tag{9}$$

The HLLC and HLLCM schemes can be derived by assuming different forms for the intermediate states  $\mathbf{Q}_{L^*}$  and  $\mathbf{Q}_{R^*}$ . These assumptions are summarized in Fig. 3.



**Fig. 3** Assumptions for the intermediate states of the HLLC and HLLCM schemes

The intermediate state for the HLLC scheme<sup>[4]</sup> is given by

$$\mathbf{Q}_{K^*}^{\text{HLLC}} = \frac{\alpha_K}{s_K - s_*} \begin{pmatrix} 1 \\ s_* \\ \mathbf{v}_K \\ E_K + (s_* - u_K) \left( s_* + \frac{p_K}{\alpha_K} \right) \end{pmatrix}, \tag{10}$$

where  $\alpha_K \equiv \rho_K(s_K - u_K)$ . Similarly, the intermediate state for the HLLCM scheme<sup>[10]</sup> is given as follows:

$$\mathbf{Q}_{K^*}^{\text{HLLCM}} = \frac{\alpha_K}{s_K - s_*} \begin{pmatrix} 1 \\ s_* \\ \frac{\alpha_R \mathbf{v}_R - \alpha_L \mathbf{v}_L}{\alpha_R - \alpha_L} \\ E_K + (s_* - u_K) \left( s_* + \frac{p_K}{\alpha_K} \right) + \frac{1}{2} \left( \frac{\alpha_R \|\mathbf{v}_R\|^2 - \alpha_L \|\mathbf{v}_L\|^2}{\alpha_R - \alpha_L} - \|\mathbf{v}_K\|^2 \right) \end{pmatrix}. \tag{11}$$

The intermediate wave-speed  $s_*$  can be obtained through the following closed form expression:

$$s_* = \frac{p_L - p_R + \alpha_R u_R - \alpha_L u_L}{\alpha_R - \alpha_L}. \tag{12}$$

Comparing Eqs. (10) and (11), it can be observed that the HLLC and HLLCM schemes behave identically when  $\mathbf{v}_L = \mathbf{v}_R$ , i.e., when the flow is locally one-dimensional. The two schemes differ in their treatment of the intermediate tangential velocity. The HLLCM introduces artificial dissipation to the intermediate tangential velocity while the HLLC scheme does not. This shear viscosity improves shock stability, but it also results in highly diffused shear layers.

### 3.1 Wave-speeds

In order to complete the definition of the flux schemes, the wave-speeds  $s_L$  and  $s_R$  are required. One possible choice proposed by Davis<sup>[21]</sup> is to compute the wave-speeds based on the minimum and maximum eigenvalues of the left and right states as follows:

$$s_L = \min(u_L - c_L, u_R - c_R), \quad s_R = \max(u_L + c_L, u_R + c_R), \quad (13)$$

where  $c = \sqrt{\gamma p / \rho}$  is the acoustic speed. Another possible choice recommended by Einfeldt<sup>[22]</sup> is to use the eigenvalues of the Roe-averaged state as follows:

$$s_L = \min(u_L - c_L, \hat{u} - \hat{c}), \quad s_R = \max(u_R + c_R, \hat{u} + \hat{c}), \quad (14)$$

where quantities with overhats ( $\hat{\cdot}$ ) refer to the respective Roe-averages. Batten et al.<sup>[23]</sup> recommend Eq. (14) for wave-speed estimation since it is less diffusive than Eq. (13) and it returns the exact shock velocity in the case of an isolated shock. Recently, Liu et al.<sup>[24]</sup> showed that the choice of wave-speed plays a significant role in the stability of the flux scheme and demonstrated that Eq. (13) provides better shock stability compared with Eq. (14). Therefore, Eq. (13) will be used in this study.

## 4 Proposed scheme

### 4.1 Motivation

In the hybrid HLLC-HLLCM scheme proposed by Shen et al.<sup>[10]</sup>, the intermediate states are defined as

$$\mathbf{Q}_{K^*}^{\text{Hybrid}} = w \mathbf{Q}_{K^*}^{\text{HLLC}} + (1 - w) \mathbf{Q}_{K^*}^{\text{HLLCM}}, \quad (15)$$

where  $w$  is a pressure-based switching function defined as follows:

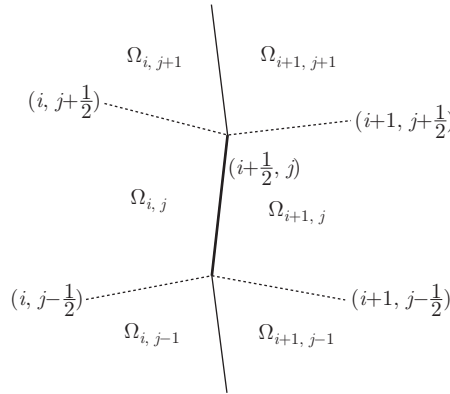
$$w = \min_{n \in NB_f} (w_f, w_n), \quad w_f = \begin{cases} 1, & \frac{\min(p_L, p_R)}{|p_R - p_L| + \varepsilon} \geq 1, \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

The sensor  $w$  must be evaluated in a non-local manner, considering not only the face  $f$  but also all its neighbors. For a structured 2D mesh, the set of neighboring faces to face  $(i+1/2, j)$  is shown in Fig. 4. Such non-local operations are unavoidable for determining regions near shocks, but they contribute to additional implementation costs. Therefore, it would be convenient to have a localized sensor instead. This provided the motivation to develop a localized ‘shear sensor’. Unlike the original hybrid HLLC-HLLCM scheme in which dissipation is introduced in regions near shocks by switching from the HLLC scheme to the HLLCM scheme, the shear sensor may be used to eliminate dissipation in regions close to shear layers by switching from the HLLCM scheme to the HLLC scheme. The construction of such a shear sensor will be presented next.

### 4.2 A localized velocity-based shear sensor

The objective is to design a sensor that detects shear layers by using state values from the immediate neighboring cells of a face. The distinguishing characteristic of a genuine shear layer is that the streamwise velocity changes more rapidly compared with the cross-stream velocity across a shear layer. With this idea in mind, a localized shear sensor  $\Theta$  could take the form

$$\Theta = \frac{|\Delta u|}{\|\Delta \mathbf{u}\|}, \quad (17)$$

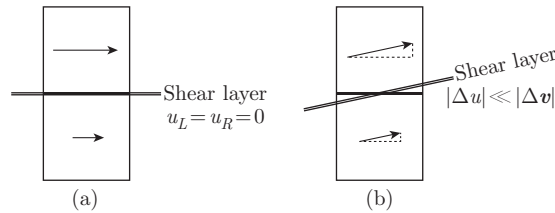


**Fig. 4** Neighboring faces (dotted lines) used in the computation of the pressure-based shock sensor at a given face (bold line)

where ‘ $\Delta$ ’ is the difference operator across the face, i.e.,  $\Delta\phi \equiv \phi_R - \phi_L$ . When a face is oriented parallel to a grid-aligned shear layer exactly as shown in Fig. 5(a),  $u_L = u_R = 0$ . Therefore, the numerator of  $\Theta$  vanishes and  $\Theta = 0$ . Even when the shear layer is not perfectly grid-aligned, as long as the face is oriented approximately parallel to a shear layer as shown in Fig. 5(b), the tangential velocity changes much faster than the normal velocity. Hence, it is expected that  $|\Delta u| \ll \|\Delta \mathbf{v}\| < \|\Delta \mathbf{u}\|$  across the face. Under these conditions, the numerator of  $\Theta$  is much smaller than its denominator and, as a result,  $\Theta$  is very small. To minimize the dissipation across genuine shear layers, the switching parameter  $w$  was taken to be

$$w = 1 - \left( \frac{\tau |\Delta u|}{\|\Delta \mathbf{u}\| + 10^{-12}} \right)^2, \tag{18}$$

where the constant  $\tau$  was set to 0.5 based on numerical experiments. A small number  $10^{-12}$  was added to the denominator to avoid division by zero.



**Fig. 5** Behavior of normal and tangential velocity differences across (a) grid-aligned and (b) non-grid-aligned shear waves

Equation (18) performed well for several benchmark problems such as hypersonic flow past a blunt body<sup>[25]</sup> and double Mach reflection<sup>[26]</sup> problems commonly used to assess the shock stability of flux schemes. However, it was still inadequate in suppressing shock instability in problems where the instability manifested as odd-even decoupling, e.g., Quirk’s test<sup>[5]</sup>. To overcome this deficiency,  $w$  was modified slightly by introducing a small reference velocity  $\varepsilon_u$  as follows:

$$w = 1 - \left( \frac{\tau |\Delta u| + \varepsilon_u}{\|\Delta \mathbf{u}\| + \varepsilon_u + 10^{-12}} \right)^2. \tag{19}$$

We require  $\varepsilon_u$  to increase faster than the velocity differences that may arise from numerical perturbations along a shock and yet remain several orders of magnitude smaller than the velocity



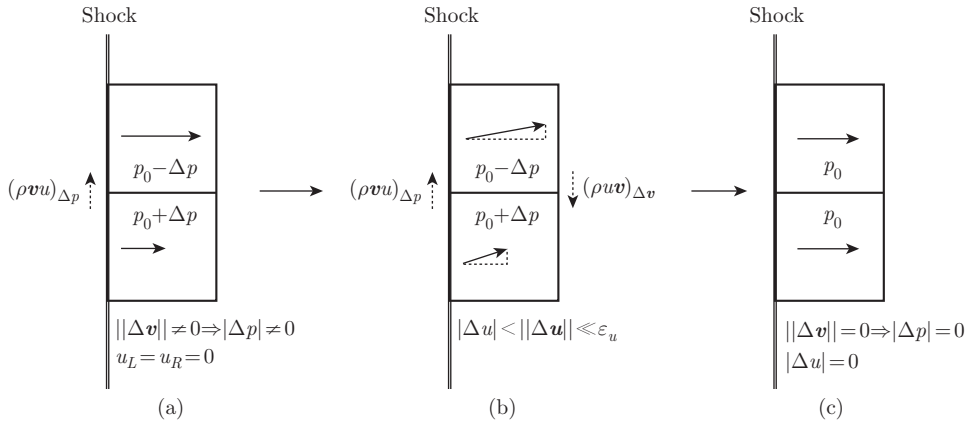
difference across a genuine shear layer. Furthermore, we also require  $\varepsilon_u$  to vanish at grid-aligned shear layers to preserve them exactly. Based on these considerations,  $\varepsilon_u$  was computed as follows:

$$\varepsilon_u = \min(10 |u_L + u_R - 2s_*|, 10^{-4}(\|u_L\| + \|u_R\|)). \quad (20)$$

The rationale behind this modification is as follows. Consider a face oriented parallel to the shock propagation direction. In the absence of any perturbations, all the properties remain the same on both sides of the face. Since  $\Delta \mathbf{v} = \mathbf{0}$ , the value of  $w$  is irrelevant. However, numerical perturbations do arise due to various reasons, and it has been found that the shock instability is linked to perturbations in the shock-normal momentum<sup>[27]</sup> which is the tangential momentum component with respect to the face. Hence, suppose that a slight perturbation is introduced only in the tangential velocity after the flow passes through the shock as shown in Fig. 6(a). Since pressure is computed based on the difference between the total and kinetic energies (see Eq. (3)), a pressure difference occurs between the two cells with a lower pressure in the cell with larger  $\|\mathbf{u}\|$  and a higher pressure in the cell with smaller  $\|\mathbf{u}\|$ . With  $u_L = u_R = 0$  and  $w = 1$  initially, the HLLC scheme is used. The HLLC scheme responds to the pressure difference (but not the tangential velocity difference which caused it) by inducing a mass flux across the face. However, the transfer of mass also results in the transfer of tangential momentum  $(\rho \mathbf{v} \mathbf{u})_{\Delta p}$  from the high-pressure cell to the low-pressure cell, thereby increasing the tangential velocity difference. As a result, the pressure perturbation continues to persist, and the process repeats.

The imbalance of pressure on the top and bottom faces of the cell changes the normal momentum in the cells, causing an increase in  $\varepsilon_u$ . As the above-described cycle continues,  $\varepsilon_u$  continues to grow. Recall that  $\varepsilon_u$  was designed to increase faster than  $\|\Delta \mathbf{u}\|$  that arises from small numerical perturbations. As such,  $|\Delta u|, \|\Delta \mathbf{u}\| \ll \varepsilon_u$ , and  $w \approx 0$ . Consequently, a tangential momentum flux  $(\rho \mathbf{u} \mathbf{v})_{\Delta u}$  is activated in the direction opposite to the pressure induced flux  $(\rho \mathbf{v} \mathbf{u})_{\Delta p}$  as shown in Fig. 6(b). This tangential momentum flux equalizes the tangential velocity difference between the cells bringing  $\Delta \mathbf{v}$  to approximately zero which, in turn, dampens the pressure perturbations and thereby ends the cycle as depicted in Fig. 6(c). In this manner, the proposed sensor alleviates the shock instability problem.

In contrast to the pressure-based switch, the localized velocity-based shear sensor provides a continuous blending of the HLLC and HLLCM schemes. This blended scheme will be referred to as the HLLCT scheme whereby ‘T’ indicates the modifications pertaining to the tangential momentum.



**Fig. 6** Restoring mechanism of the proposed sensor in regions susceptible to shock instability

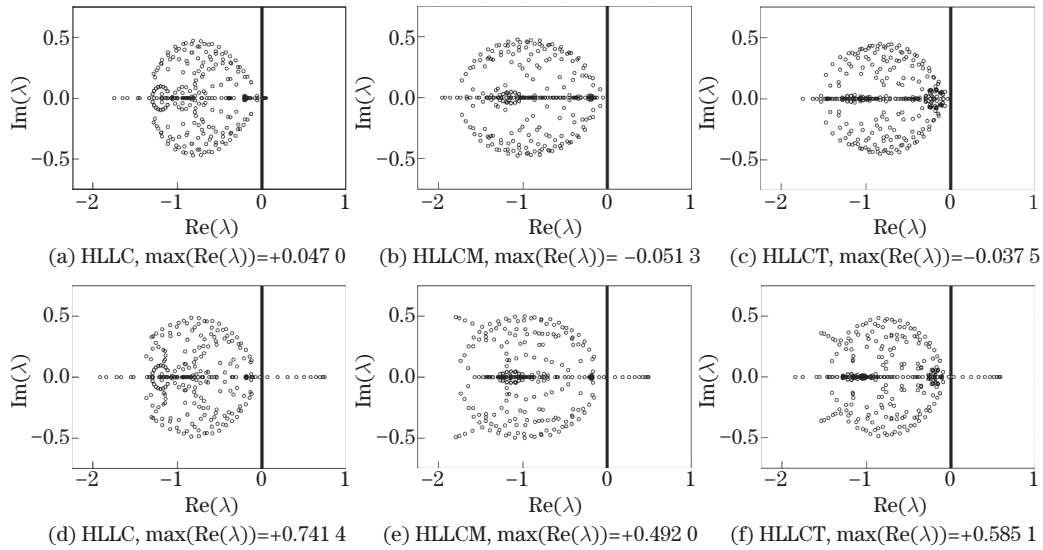
### 4.3 Matrix stability analysis

Matrix stability analysis is a linear analysis technique introduced by Dumbser et al.<sup>[14]</sup> to characterize the shock stability of a numerical flux scheme. The analyses were performed for the HLLC, HLLCM, and HLLCT schemes, using wave-speed estimates from Eqs. (13) and (14). For the wave-speed estimates from Eq. (14), the fluxes were linearized about a grid-aligned  $M = 7$  shock initialized on a dimensional  $11 \times 11$  uniform Cartesian grid as follows:

$$(\rho, u_x, u_y, p) = \begin{cases} \left(1, 1, 0, \frac{1}{\gamma M^2}\right), & i < 7, \\ \left(\frac{(\gamma + 1)M^2}{(\gamma - 1)M^2 + 2}, \frac{(\gamma - 1)M^2 + 2}{(\gamma + 1)M^2}, 0, \frac{2\gamma M^2 - (\gamma - 1)}{(\gamma + 1)\gamma M^2}\right), & i \geq 7. \end{cases} \quad (21)$$

On the other hand, as the wave-speed estimates from Eq. (13) do not admit an isolated grid-aligned shock, the fluxes were linearized about the one-dimensional numerical shock solution instead, for a truly meaningful analysis. The one-dimensional problem was initialized with an  $M = 7$  shock using Eq. (21) on a uniform 11-cell grid. Ghost cells were used to prescribe the boundary conditions. The ghost cell on the left was fixed at the supersonic state while the pressure and density were updated at the ghost cell on the right using the Dirichlet and Neumann conditions, respectively. The horizontal velocity at the right ghost cell was updated to enforce a mass flow rate of unity. The problem was computed using the HLLC scheme until the density residual dropped below  $10^{-15}$ . Then, the one-dimensional solution was mapped to the two-dimensional  $11 \times 11$  uniform Cartesian grid for the stability analysis. Since the HLLC, HLLCM, and HLLCT schemes are identical in one-dimension, the same numerical shock solution was used for all three schemes. The conserved variables  $\mathbf{Q}$  were perturbed by an amplitude of  $\pm 10^{-6}$  to approximate the flux gradients. The eigenvalues  $\lambda$  of the flux Jacobian matrices are plotted in Fig. 7. A flux scheme is deemed unstable if  $\max(\text{Re}(\lambda)) \geq 0$ , i.e., if the maximum real part of any of the eigenvalues is positive.

It can be noticed from the results that all three schemes are unstable when using the wave-speed estimates from Eq. (14) as evident from the positive values of  $\max(\text{Re}(\lambda))$ . In comparison,



**Fig. 7** Matrix stability analyses for the HLLC, HLLCM, and HLLCT schemes, using wave-speed estimates from Eq. (13) (top row) and Eq. (14) (bottom row)

using wave-speed estimates from Eq. (13) reduces  $\max(\text{Re}(\lambda))$  significantly indicating an improvement in shock stability. Yet,  $\max(\text{Re}(\lambda))$  still remains positive for the HLLC scheme. On the other hand, with  $\max(\text{Re}(\lambda)) < 0$ , both the HLLCM and HLLCT schemes are stable when using wave-speed estimates from Eq. (13).

## 5 Numerical experiments

In this section, the HLLCT scheme is compared with the HLLC scheme, the hybrid HLLC-HLLCM scheme with the pressure-based shock sensor (Eq. (27)) and, where shear layers are concerned, with the HLLCM scheme as well. The comparison is performed only for two-dimensional test cases since shock instability is multi-dimensional in nature. Moreover, the behaviors of the HLLC, HLLCM, and HLLCT schemes are identical in one-dimension.

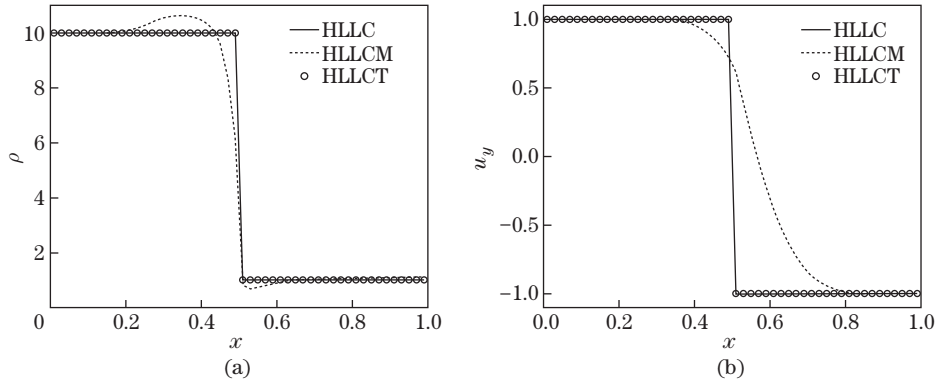
### 5.1 Shear layers

#### 5.1.1 Contact discontinuity

This problem is a test of a scheme's ability to preserve a grid-aligned contact discontinuity. The domain  $x \times y \in [0, 1] \times [0, 1]$  was discretized into a uniform  $50 \times 50$  grid. The problem was initialized as follows:

$$(p, u_x) = (1, 0), \quad (\rho, u_y) = \begin{cases} (10, 1), & x < 0.5, \\ (1, -1), & x \geq 0.5, \end{cases} \quad (22)$$

where  $u_x$  and  $u_y$  refer to the horizontal and vertical velocity components, respectively. The Neumann boundary conditions were applied on all sides. The simulations were run for 100 iterations using the implicit solver.  $\rho$  and  $u_y$  profiles along the horizontal line  $y = 0.5$  are shown in Fig. 8. It can be easily shown from Eqs. (19) and (20) that the shear sensor  $w = 1$  everywhere for this problem. Therefore, the HLLCT scheme reverts to the HLLC scheme and preserves the isolated contact discontinuity exactly. The HLLCM scheme, on the other hand, diffuses it. Due to the absence of any pressure difference, the HLLC-HLLCM scheme performed identically to the HLLC scheme as well and, hence, it is not included in Fig. 8.



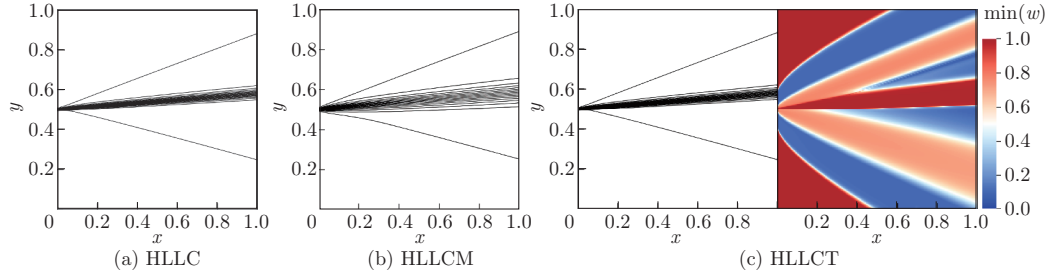
**Fig. 8** Density profile for the stationary contact discontinuity problem

#### 5.1.2 Supersonic jet

This problem tests a scheme's ability in capturing non-grid-aligned shear layers. The domain  $x \times y \in [0, 1] \times [0, 1]$  was discretized into a uniform  $200 \times 200$  grid. The problem was initialized as follows:

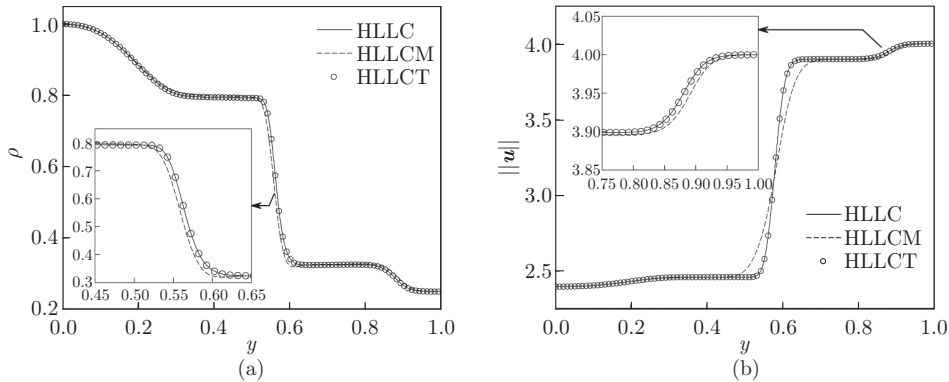
$$(\rho, u_x, u_y, p) = \begin{cases} (0.25, 4, 0, 0.25), & y > 0.5, \\ (1.0, 1.4, 0, 0.5), & y \leq 0.5. \end{cases} \quad (23)$$

The left boundary was fixed at the initial values while the Neumann boundary condition was applied to the remaining boundaries. The simulations were run for 1500 iterations which was sufficient for the density residuals to drop to  $10^{-12}$ . The final results are plotted in Fig. 9 with sixteen equispaced velocity magnitude contours from  $\|\mathbf{u}\| = 2.45$  to  $\|\mathbf{u}\| = 3.95$ . The minimum value of the shear sensor  $w$  (Eq. (19)) over the faces of each cell is plotted in Fig. 9(c) next to the velocity magnitude contours for the HLLCT scheme.



**Fig. 9** Velocity magnitude contours for the supersonic jet problem (color online)

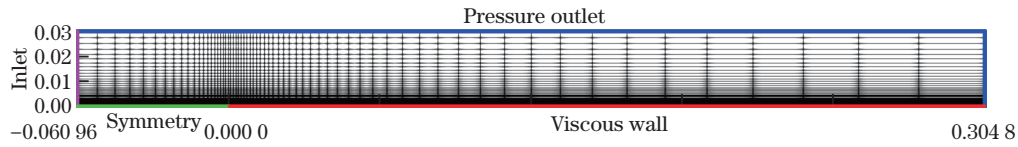
It is clear that the HLLCT scheme performs on par with the HLLC scheme in capturing the shear layer while the HLLCM scheme diffuses it. It is immediately obvious that the velocity-based shear sensor identifies the regions of shear accurately and becomes approximately equal to one.  $\rho$  and  $\|\mathbf{u}\|$  profiles sampled along the vertical line  $x = 1$  are plotted in Fig. 10 for a more quantitative comparison. This reveals that the performance of the HLLCT scheme is indistinguishable from that of the HLLC scheme. It is interesting to note that the performance of the HLLCT scheme is not affected adversely in the regions where  $w \approx 0$ . This is most likely due to the slow variations in the tangential velocities in those regions.



**Fig. 10** Density and velocity magnitude profiles for the supersonic jet problem

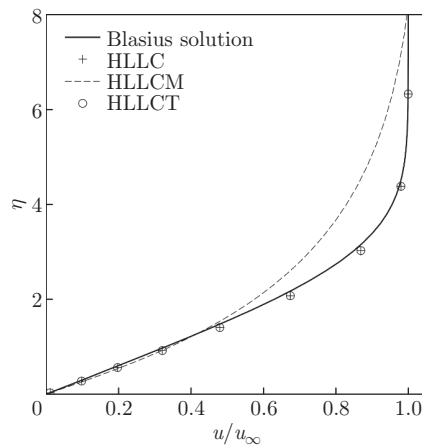
### 5.1.3 $M = 0.2$ laminar flow past flat plate

This problem is a more practical case which tests the ability of the scheme in resolving the viscous boundary layer in an  $M = 0.2$  laminar flow past a flat plate. The problem was computed on the domain  $[-0.06096, 0.3048] \times [0, 0.03]$  discretized into the non-uniform  $64 \times 64$  grid shown in Fig. 11. The grid has been refined in the vertical direction close to the wall to include approximately 30 cells within the boundary layer region. A stagnation pressure of  $p_0 = 10^5$  Pa and stagnation temperature of  $T_0 = 300$  K was specified at the inlet boundary while a static pressure of  $p = 97250$  Pa was specified at the outlet boundary. The flat plate was modeled as



**Fig. 11** Grid and boundary conditions for the  $M = 0.2$  laminar flow past the flat plate problem (color online)

an adiabatic, no-slip wall. The dynamic viscosity was taken to be  $\mu = 1.87 \times 10^{-5} \text{ N}\cdot\text{s}\cdot\text{m}^{-2}$ , and the specific heat capacity at constant pressure was taken to be  $c_p = 1004.5 \text{ J}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$ . Spatial discretization was performed using the second-order upwind scheme for the inviscid fluxes and the second-order central scheme for the viscous fluxes. The problem was computed using the HLLC, HLLCM, and HLLCT schemes until the density residuals dropped to below  $10^{-6}$ . The normalized horizontal velocities  $u/u_\infty$  at the exit plane at  $x = 0.3048 \text{ m}$  are compared with the Blasius solution in Fig. 12. It can be seen that, while the HLLCM scheme produces a diffused boundary layer, the HLLCT scheme resolves the boundary layer as accurately as the HLLC scheme which matches well with the analytical solution.



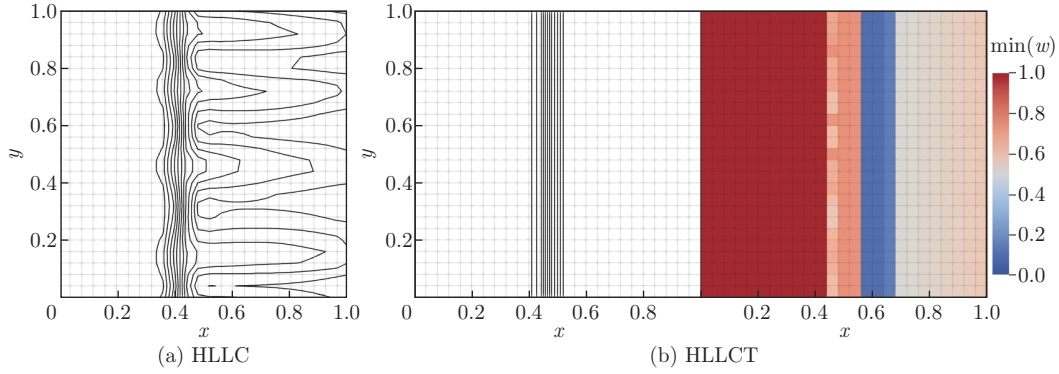
**Fig. 12** Variation of normalized horizontal velocity  $u/u_\infty$  with the Blasius parameter  $\eta = y\sqrt{xu_\infty\nu}$  at the exit plane for the laminar flow past flat plate problem. Note that every fourth point is plotted for the HLLC and HLLCT schemes for clarity

## 5.2 Shock instability problems

### 5.2.1 $M = 20$ steady shock

This problem involves a two-dimensional steady isolated standing shock. Following the work of Dumbser et al.<sup>[14]</sup>, the problem was computed on the domain  $x \times y \in [0, 1] \times [0, 1]$  discretized uniformly into a  $25 \times 25$  grid. The problem was initialized using Eq. (21) for an  $M = 20$  shock that is initially located at  $x = 0.5$ . Perturbations of the relative order of  $10^{-6}$  were introduced to the initial conditions. The left and right boundary conditions were the same as those described in Subsection 4.3. The top and bottom boundaries were set to be periodic. The problem was computed using the implicit solver for 10 000 iterations. Fourteen equispaced density contours from  $\rho = 1.2$  to  $\rho = 6.4$  are plotted in Fig. 13.

The results demonstrate that the HLLCT scheme is able to remain stable for very strong shocks. Notice that the shock position at  $x = 0.5$  is captured correctly by the HLLCT scheme. Observing the plot of  $\min(w)$  in Fig. 13(b), it can be seen that the shear dissipation is the strongest not at the shock itself but slightly downstream of it around  $x \approx 0.65$ . It is likely that



**Fig. 13** Density contours for the  $M = 20$  steady shock problem (color online)

the numerical perturbations cause the value of  $\varepsilon_u$  to become large enough about this location for the shear dissipation to be activated.

### 5.2.2 Quirk's test: odd-even decoupling problem

This classic test proposed by Quirk<sup>[5]</sup> involves an  $M = 6$  shock propagating along a perturbed grid. The domain  $x \times y \in [0, 800] \times [-20, 20]$  was discretized into a uniform  $800 \times 40$  mesh. The horizontal line  $y = 0$  was then perturbed in a sawtooth fashion as follows:

$$y_i = \begin{cases} +10^{-6}, & i \text{ is odd,} \\ -10^{-6}, & i \text{ is even.} \end{cases} \quad (24)$$

The problem was initialized with the shock positioned at  $x = 5$  as follows:

$$(\rho, u_x, u_y, p) = \begin{cases} (5.268\ 29, 5.751\ 74, 0, 41.833\ 33), & x < 5, \\ (1, 0, 0, 1), & x \geq 5. \end{cases} \quad (25)$$

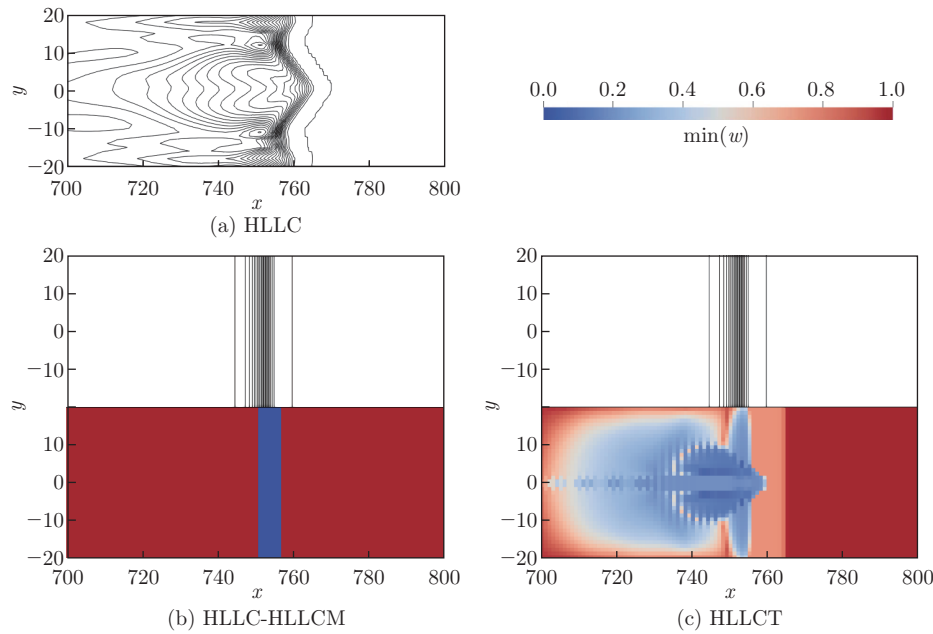
The transient simulation was performed until  $t = 105$  s at which point the shock front would have propagated to  $x = 750$ . The final results are shown in Fig. 14 using twenty-five equispaced density contours from  $\rho = 1$  to  $\rho = 5.8$ . The pressure-based shock sensor (Eq. (16)) and velocity-based shear sensor are plotted below the density contours of the HLLC-HLLCM and HLLCT schemes, respectively.

While the HLLC scheme breaks down the shock front and produces a prominent carbuncle, both the HLLC-HLLCM and HLLCT schemes keep it intact. It can be observed from Fig. 14(c) that shear dissipation is the most active around  $y = 0$  where the grid is perturbed. It is interesting to note that the region of increased dissipation around  $y = 0$  resembles the shock instability pattern observed for the HLLC scheme in Fig. 14(a).

The computation was repeated for an  $M = 20$  shock to demonstrate that the choice of numerical constants in Eq. (20) applies for a wide range of shock strengths. The computation was performed on the extended domain  $x \times y \in [0, 1\ 500] \times [-20, 20]$  which was discretized into a uniform  $1\ 500 \times 40$  mesh. The horizontal line  $y = 0$  was perturbed as before. The problem was initialized with the shock positioned at  $x = 5$  as follows:

$$(\rho, u_x, u_y, p) = \begin{cases} (5.925\ 93, 19.671, 0, 466.5), & x < 5, \\ (1, 0, 0, 1), & x \geq 5. \end{cases} \quad (26)$$

The problem was computed until  $t = 61$  s at which point the shock would have propagated to approximately  $x = 1\ 448.5$ . The final results are shown in Fig. 15 using twenty-six equispaced



**Fig. 14** Density contours for Quirk's test for  $M = 6$  (color online)

density contours from  $\rho = 1$  to  $\rho = 6$ . Once again, the HLLC scheme breaks down the shock front completely. Interestingly, the HLLC-HLLCM hybrid scheme also becomes susceptible to the shock instability due to insufficient dissipation behind the shock. It can be seen from Fig. 15(b) that the pressure-based sensor does not encompass the entire shock. On the other hand, The HLLCT scheme remains stable and captures the shock front accurately. This test demonstrates that the choice of the numerical constants used for  $\varepsilon_u$  in Eq. (20) is applicable for a wide range of shock strengths.

### 5.2.3 Hypersonic flow past blunt body

This problem involves an  $M_\infty = 20$  flow past a cylinder. The gridpoints  $(x_{i,j}, y_{i,j})$  were obtained from the expressions

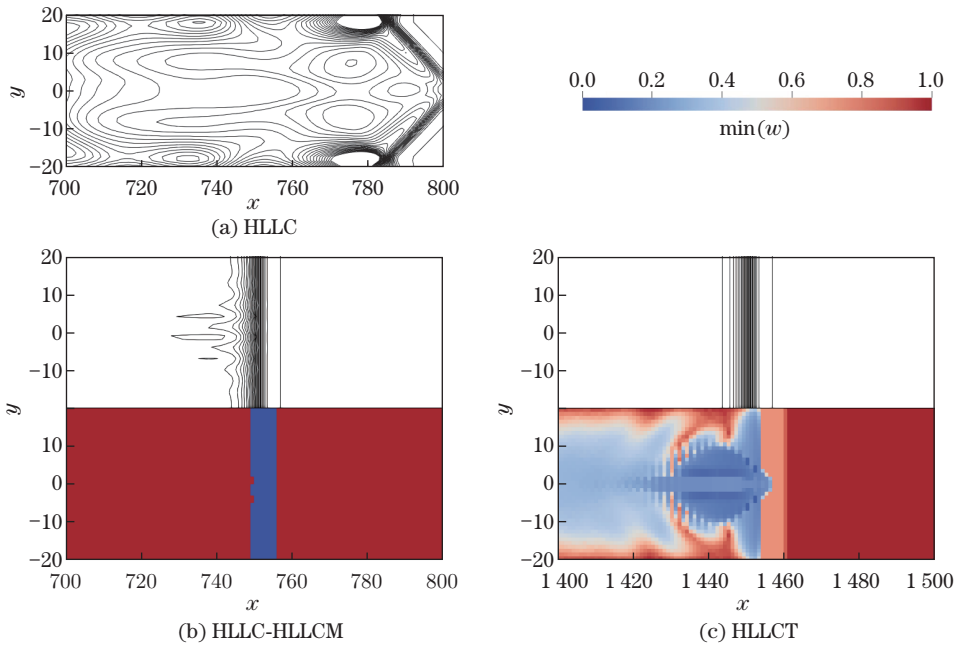
$$x_{i,j} = \frac{2x_{ci} - \sqrt{4x_{ci}^2 - 4(1 + \tan^2 \theta_j)(c_i^2 - r_i^2)}}{2(1 + \tan^2 \theta_j)}, \quad y_{i,j} = -x_{i,j} \tan \theta_j, \quad (27)$$

where  $\theta_j = \frac{11\pi}{15}(j/J - 0.5)$ ,  $x_{ci} = 1.8(I - i)/I$ , and  $r_i = 1 + 2.4(I - i)/I$ . The indices  $i$  and  $j$  run from 0 to  $I$  and 0 to  $J$ , respectively. Hence, the grid consists of  $I \times J$  cells in total. A  $10 \times 200$  grid is shown in Fig. 16(a). The problem was computed on a  $20 \times 400$  grid to trigger a prominent carbuncle. The problem was initialized as follows:

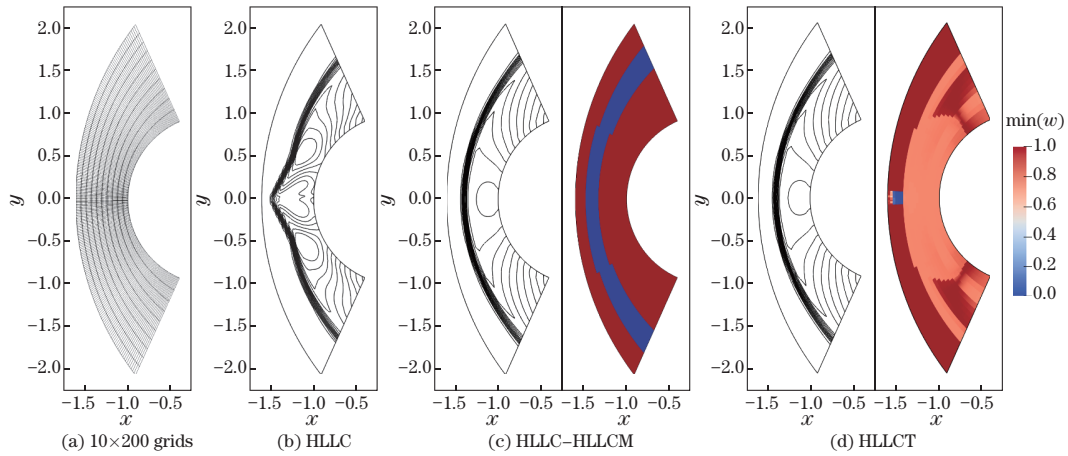
$$(\rho, u_x, u_y, p) = (1.4, 20, 0, 1). \quad (28)$$

The initial conditions were applied at the inlet. The reflective boundary condition was prescribed along the cylinder, and the Neumann boundary conditions were applied at the top and bottom boundaries. The simulations were run until the density residuals dropped down to  $10^{-11}$ . The final results are shown in Fig. 16 using fifteen equispaced density contours from  $\rho = 1.4$  to  $\rho = 8.4$ . The pressure-based shock sensor and velocity-based shear sensor are plotted next to the density contours of the HLLC-HLLCM and HLLCT schemes, respectively.

The HLLC scheme produces a very prominent carbuncle, but the HLLCT scheme is carbuncle-free similar to the HLLC-HLLCM scheme. It is interesting to note that  $\min(w)$  remains at



**Fig. 15** Density contours for Quirk's test for  $M = 20$  (color online)



**Fig. 16** Density contours for the  $M_\infty = 20$  flow past a cylinder problem (color online)

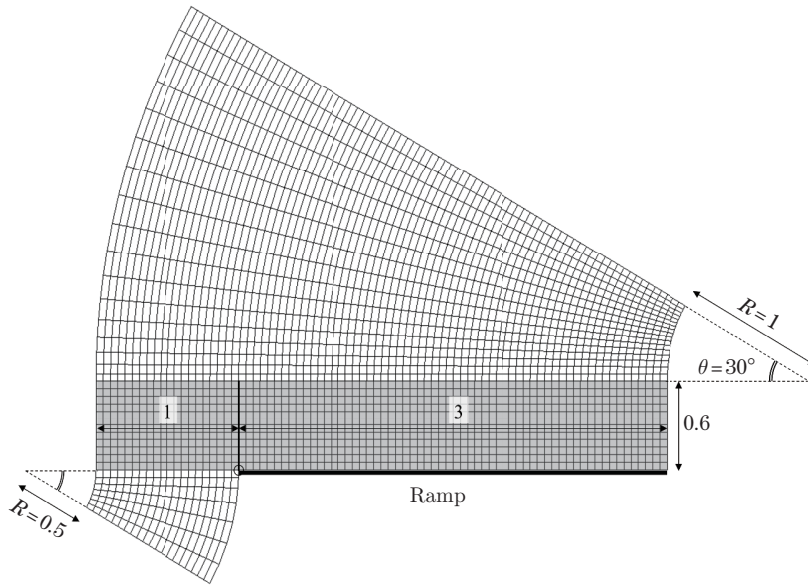
about 0.75 near the shock for the HLLCT scheme and yet, this is sufficient to suppress the shock instability in this problem.

#### 5.2.4 Double Mach reflection problem

The double Mach reflection (DMR) problem proposed by Woodward and Colella<sup>[26]</sup> involves an  $M = 10$  shock impinging on an inclined ramp to form a system of complex shock reflections. The DMR problem was solved on a modified computational domain proposed by Vevek et al.<sup>[28]</sup> to prevent the formation of numerical artefacts. A coarse mesh of the modified domain is shown in Fig. 17.

The computational domain consists of a rectangular region  $x \times y \in [-1, 3] \times [0, 0.6]$  (shaded





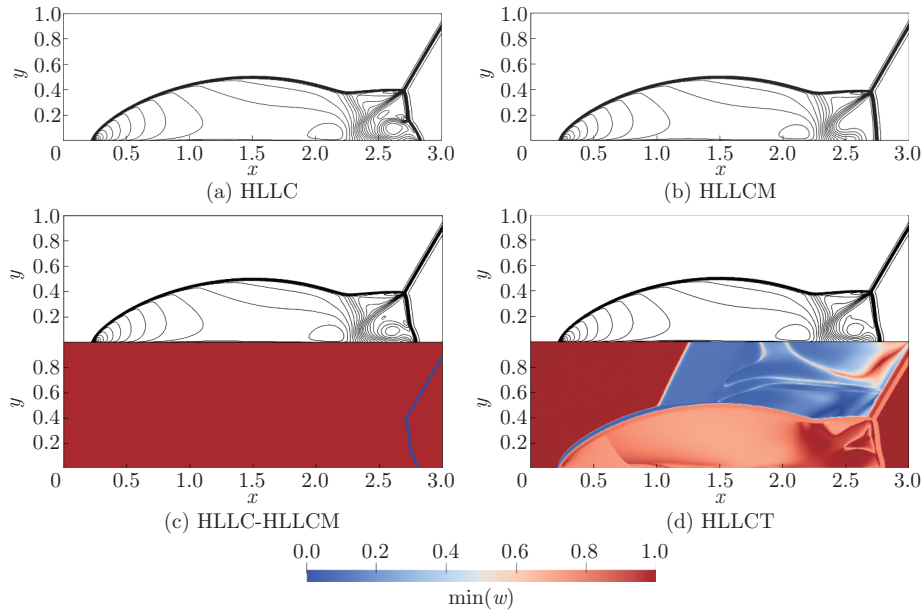
**Fig. 17** Modified computational domain for simulating the DMR problem

region in Fig. 17) that is discretized using uniform Cartesian mesh. Prominent flow features of the DMR problem are confined to this region. The top side and portion of the bottom side of the rectangular region are rotated by  $30^\circ$  to form the extensions above and below, respectively. The ramp surface extends from the origin (small circle in Fig. 17) to the right along the bottom side of the rectangular region. The advantage of this setup is that the oblique shock oriented at  $30^\circ$  to the horizontal ramp lies normal to the boundaries created by the extensions which allows one to use simple Neumann boundary conditions to track the moving shock exactly. The reflective boundary conditions are applied along the ramp. The exact pre- and post-shock conditions

$$\begin{cases} (\rho, u_x, u_y, p)_{\text{pre}} = (1.4, 0, 0, 1), \\ (\rho, u_x, u_y, p)_{\text{post}} = (8, -8.25 \cos 30^\circ, 8.25 \sin 30^\circ, 116.5), \end{cases} \quad (29)$$

were prescribed at the right and left boundaries, respectively. The shock was initially positioned at  $x = -0.9$  and allowed to propagate for  $t = 0.08$  s at which time it would approximately impinge on the ramp. Next, the region to the left shock is re-initialized with the exact post-shock conditions given above to eliminate the numerical noise generated as the shock is smeared on the grid. Then, the simulation is continued for another 0.19 s until  $t = 0.27$  s. The final results are plotted in Fig. 18 using forty-three equispaced density contours from  $\rho = 1.4$  to  $\rho = 22.4$ . The pressure-based shock sensor and velocity-based shear sensor are plotted below the density contours of the HLLC-HLLCM and HLLCT schemes, respectively.

The HLLC scheme produces a kinked Mach stem near  $x = 2.5$ . While the HLLC-HLLCM scheme reduces the extent of the kink, it does not eliminate it completely. On the other hand, the HLLCT scheme removes the kink entirely. Secondly, it is clear from the plot of the shear sensor in Fig. 18(c) that the value of  $w$  is approximately one near the slip lines. Consequently, the HLLCT scheme behaves like the HLLC scheme and captures the slip lines with little dissipation. Comparing Figs. 18(b) and 18(d), it can be seen that the HLLCM scheme has destroyed the roll-up of the primary slip line near the bottom wall while the HLLCT scheme has improved the resolution of the slip line significantly.



**Fig. 18** Density contours for the DMR problem (color online)

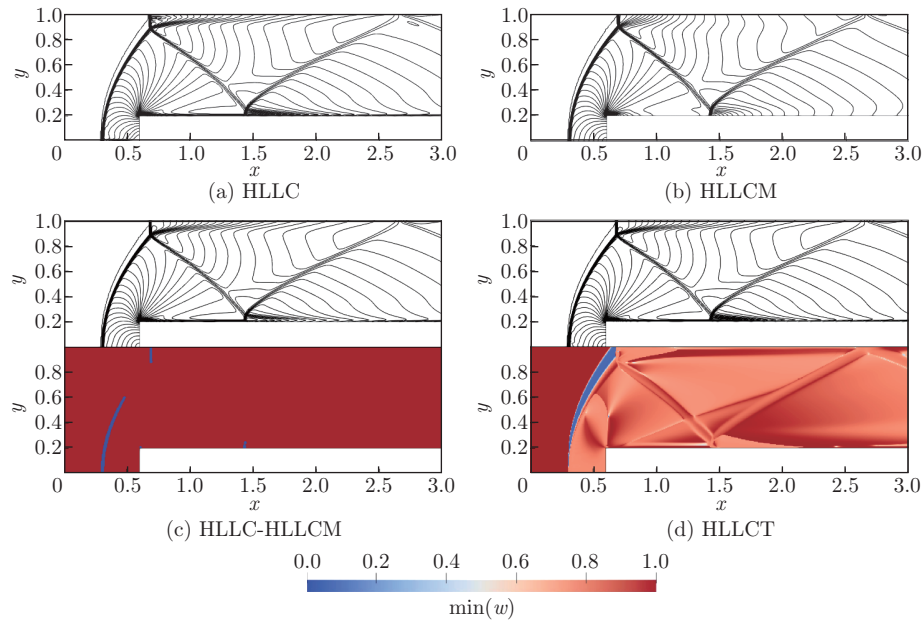
### 5.2.5 Forward-facing step problem

This problem involves an  $M_\infty = 3$  flow impinging on a forward-facing step<sup>[29]</sup>. The problem was solved on the domain  $[x, y] \in [0, 3] \times [0, 1]$  discretized uniformly using cells of width  $\Delta x = \Delta y = 1/160$ . The 0.2-unit high step extends to the right from  $x = 0.6$ . The problem was initialized as follows:

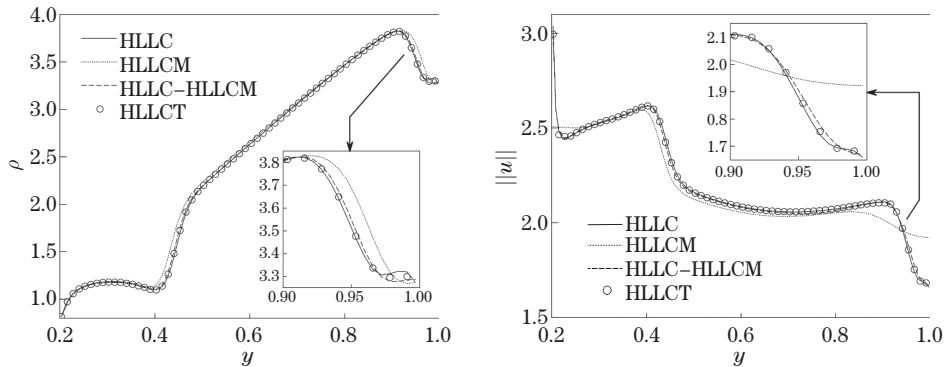
$$(\rho, u_x, u_y, p) = (1.4, 3, 0, 1). \quad (30)$$

The left boundary was set to supersonic inflow using the above initial conditions while the Neumann boundary condition was applied at the right boundary. All other boundaries were set to be reflective. The special treatment suggested by Woodward and Colella<sup>[26]</sup> was applied at the step corner to minimize the effects of the unphysical ‘boundary layer’. The problem was computed till  $t = 3$  s, and the final results are plotted in Fig. 19, using the forty-one velocity magnitude contours from  $\|\mathbf{u}\| = 0$  to  $\|\mathbf{u}\| = 4$  to show the slip line emanating from the triple point clearly. The pressure-based shock sensor and velocity-based shear sensor are plotted below the velocity magnitude contours of the HLLC-HLLCM and HLLCT schemes, respectively.

The HLLC scheme results in instabilities downstream of the normal shock which are eliminated by the other three schemes. While the HLLCM scheme diffuses the slip lines issuing from the triple point almost completely, both the HLLC-HLLCM and HLLCT schemes are able to capture them well. Observing the plot of  $\min(w)$  in Fig. 19(d), it is evident that the shear sensor identifies the slip lines accurately and attains a value close to unity, thus reducing the dissipation across the slip lines. For a more quantitative comparison,  $\rho$  and  $\|\mathbf{u}\|$  profiles sampled along the vertical line  $x = 1.25$  are plotted in Fig. 20. The results near the slip lines are plotted in the insets. Interestingly, the HLLCT scheme performs slightly better than the HLLC-HLLCM scheme as benchmarked against the HLLC scheme. This is mostly likely because, unlike the pressure-based shock sensor, the velocity-based shear sensor does not vanish completely around the shock. Therefore, the HLLCT scheme introduces lesser dissipation than the HLLC-HLLCM scheme. Nevertheless, this is sufficient in eliminating the shock instability.



**Fig. 19** Velocity magnitude contours for the forward-facing step problem (color online)



**Fig. 20** Density and velocity magnitude profiles for the forward-facing step problem

### 5.3 Computational efficiency

It has been demonstrated thus far that the HLLCT scheme is able to resolve shear layers well without succumbing to the shock instability problem. However, it is also important to ensure that the scheme is computationally efficient. In Table 1, a comparison of the computation time of the different schemes is provided for the two largest problems considered in this study, namely, the DMR and forward-facing step problems. The computation time has been normalized with respect to that of the HLLC scheme. It is clear that the HLLCT scheme remains competitive with the original HLLC scheme. Though the HLLC-HLLCM scheme takes the longest time amongst the schemes, it is not far behind the others. This shows that, if implemented properly, shock sensor computation is not necessarily a demanding operation. Its main drawback lies in introducing it within existing codes which have been designed to compute fluxes in a localized manner. In this regard, it is straightforward to incorporate the HLLCT scheme in such codes.

**Table 1** Comparison of normalized computation time

Problem	HLLC	HLLCM	HLLC-HLLCM	HLLCT
DMR	1.000	0.964	1.011	1.006
Forward-facing step	1.000	0.949	1.016	0.984

## 6 Conclusions

In conventional carbuncle cures, the HLLC scheme is coupled with a more diffusive, shock-stable scheme, and the complementary scheme is applied in the vicinity of shocks. This requires a suitable shock sensor which needs to be computed in a non-local manner, considering all the neighboring faces to assure the elimination of the shock instability. The contact-preserving shock-stable HLLCM scheme proposed by Shen et al.<sup>[10]</sup> presents a unique opportunity to explore the viability of a different approach to curing the carbuncle problem. Since the HLLCM scheme introduces spurious dissipation only across shear layers, it was hypothesized that it might be sufficient to restrict the action of the HLLC scheme to regions of shear and use the HLLCM scheme everywhere else. To evaluate this hypothesis, a hybrid HLLC-HLLCM scheme called the HLLCT scheme was designed. The crucial advantage of the HLLCT scheme comes from a velocity-based shear sensor which can be computed in a localized manner based on the left and right states only. Numerical experiments on grid-aligned and non-grid-aligned shear layers show that the HLLCT scheme is able to capture shear layers with as little dissipation as the HLLC scheme itself, attesting to the effectiveness of the shear sensor. Interestingly, the use of the non-shear-preserving HLLCM scheme in regions away from shear layers does not seem to have an adverse effect on the results. More importantly, the HLLCT scheme is also found to be free from any shock instabilities for a number of classic test cases involving strong grid-aligned shocks for which the HLLC scheme fails.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- [1] GODUNOV, S. K. A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics. *Sbornik: Mathematics*, **47**, 357–393 (1959)
- [2] HARTEN, A., LAX, P., and LEER, B. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, **25**, 35–61 (1983)
- [3] LIU, M. S. Mass flux schemes and connection to shock instability. *Journal of Computational Physics*, **160**, 623–648 (2000)
- [4] TORO, E. F., SPRUCE, M., and SPEARES, W. Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, **4**, 25–34 (1994)
- [5] QUIRK, J. J. A contribution to the great Riemann solver debate. *Upwind and High-Resolution Schemes*, Springer, Berlin, 550–569 (1997)
- [6] PANDOLFI, M. and D’AMBROSIO, D. Numerical instabilities in upwind methods: analysis and cures for the “carbuncle” phenomenon. *Journal of Computational Physics*, **166**, 271–301 (2001)
- [7] XU, K. and LI, Z. Dissipative mechanism in Godunov-type schemes. *International Journal for Numerical Methods in Fluids*, **37**, 1–22 (2001)
- [8] SIMON, S. and MANDAL, J. C. A cure for numerical shock instability in HLLC Riemann solver using antidiffusion control. *Computers & Fluids*, **174**, 144–166 (2018)

- 
- [9] SIMON, S. and MANDAL, J. C. A simple cure for numerical shock instability in HLLC Riemann solver. *Journal of Computational Physics*, **378**, 477–496 (2019)
- [10] SHEN, Z., YAN, W., and YUAN, G. A robust HLLC-type Riemann solver for strong shock. *Journal of Computational Physics*, **309**, 185–206 (2016)
- [11] HUANG, K., WU, H., YU, H., and YAN, D. Cures for numerical shock instability in HLLC solver. *International Journal for Numerical Methods in Fluids*, **65**, 1026–1038 (2011)
- [12] KIM, S. D., LEE, B. J., LEE, H. J., and JEUNG, I. S. Robust HLLC Riemann solver with weighted average flux scheme for strong shock. *Journal of Computational Physics*, **228**, 7634–7642 (2009)
- [13] REN, Y. X. A robust shock-capturing scheme based on rotated Riemann solvers. *Computers & Fluids*, **32**, 1379–1403 (2003)
- [14] DUMBSER, M., MOSCHETTA, J. M., and GRESSIER, J. A matrix stability analysis of the carbuncle phenomenon. *Journal of Computational Physics*, **197**, 647–670 (2004)
- [15] CHEN, S. S., YAN, C., LIN, B. X., and LI, Y. S. A new robust carbuncle-free Roe scheme for strong shock. *Journal of Scientific Computing*, **77**, 1250–1277 (2018)
- [16] KIM, S. S., KIM, C., RHO, O. H., and HONG, K. S. Cures for the shock instability: development of a shock-stable Roe scheme. *Journal of Computational Physics*, **185**, 342–374 (2003)
- [17] CHEN, S. S., LIN, B. X., LI, Y. S., and YAN, C. HLLC+: low-Mach shock-stable HLLC-type Riemann solver for all-speed flows. *SIAM Journal on Scientific Computing*, **42**, B921–B950 (2020)
- [18] SIMON, S. and MANDAL, J. C. Strategies to cure numerical shock instability in the HLLEM Riemann solver. *International Journal for Numerical Methods in Fluids*, **89**, 533–569 (2019)
- [19] GOTTLIEB, S. and SHU, C. W. Total variation diminishing Runge-Kutta schemes. *Mathematics of Computation*, **67**, 73–85 (1998)
- [20] LUO, H., BAUM, J. D., and LÖHNER, R. An accurate, fast, matrix-free implicit method for computing unsteady flows on unstructured grids. *Computers & Fluids*, **30**, 137–159 (2001)
- [21] DAVIS, S. Simplified second-order Godunov-type methods. *SIAM Journal on Scientific and Statistical Computing*, **9**, 445–473 (1988)
- [22] EINFELDT, B. On Godunov-type methods for gas dynamics. *SIAM Journal on Numerical Analysis*, **25**(2), 294–318 (1988)
- [23] BATTEN, P., CLARKE, N., LAMBERT, C., and CAUSON, D. On the choice of wavespeeds for the HLLC Riemann solver. *SIAM Journal on Scientific Computing*, **18**, 1553–1570 (1997)
- [24] LIU, L., LI, X., and SHEN, Z. Overcoming shock instability of the HLLE-type Riemann solvers. *Journal of Computational Physics*, **418**, 109628 (2020)
- [25] PEERY, K. M. and IMLAY, S. T. Blunt-body flow simulations. *24th Joint Propulsion Conference*, American Institute of Aeronautics and Astronautics, Massachusetts (1988)
- [26] WOODWARD, P. and COLELLA, P. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, **54**, 115–173 (1984)
- [27] SHEN, Z., YAN, W., and YUAN, G. A stability analysis of hybrid schemes to cure shock instability. *Communications in Computational Physics*, **15**, 1320–1342 (2015)
- [28] VEVEK, U. S., ZANG, B., and NEW, T. H. On alternative setups of the double Mach reflection problem. *Journal of Scientific Computing*, **78**, 1291–1303 (2019)
- [29] EMERY, A. F. An evaluation of several differencing methods for inviscid fluid flow problems. *Journal of Computational Physics*, **2**, 306–331 (1968)