Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 2560–2581
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Very simple, carbuncle-free, boundary-layer-resolving, rotated-hybrid Riemann solvers

Hiroaki Nishikawa [a,*], Keiichi Kitamura [a,b]

[a] W.M. Keck Foundation Laboratory for Computational Fluid Dynamics, Department of Aerospace Engineering, University of Michigan, FXB Building, 1320 Beal Avenue, Ann Arbor, MI 48109-2140, USA
[b] Fluid Dynamics Laboratory, Department of Aerospace Engineering, Nagoya University, 1 Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8603, Japan

Received 8 August 2007; received in revised form 29 October 2007; accepted 4 November 2007
Available online 17 November 2007

Abstract

In this paper, we propose new Euler flux functions for use in a finite-volume Euler/Navier–Stokes code, which are very simple, carbuncle-free, yet have an excellent boundary-layer-resolving capability, by combining two different Riemann solvers into one based on a rotated Riemann solver approach. We show that very economical Euler flux functions can be devised by combining the Roe solver (a full-wave solver) and the Rusanov/HLL solver (a fewer-wave solver), based on a rotated Riemann solver approach: a fewer-wave solver automatically applied in the direction normal to shocks to suppress carbuncles and a full-wave solver applied, again automatically, across shear layers to avoid an excessive amount of dissipation. The resulting flux functions can be implemented in a very simple and economical manner, in the form of the Roe solver with modified wave speeds, so that converting an existing Roe flux code into the new fluxes is an extremely simple task. They require only 7–14% extra CPU time and no problem-dependent tuning parameters. These new rotated fluxes are not only robust for shock-capturing, but also accurate for resolving shear layers. This is demonstrated by an extensive series of numerical experiments with standard finite-volume Euler and Navier–Stokes codes, including various shock instability problems and also an unstructured grid case.

Keywords: Shock instability; Carbuncle; Hybrid schemes; Rotated Riemann solvers; Upwind schemes

1. Introduction

In this paper, we propose new flux functions for use in a finite-volume Euler/Navier–Stokes code, which are very simple, carbuncle-free, yet have an excellent boundary-layer-resolving capability, by combining two different Riemann solvers into one based on a rotated Riemann solver approach.

[*] Corresponding author. Present address: National Institute of Aerospace, 100 Exploration Way, Hampton, VA 23666-6147, USA. Tel.: +1 757 325 6906.
E-mail address: hiro@nianet.org (H. Nishikawa).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.11.003

Despite a great deal of progress made in the past decades, current finite-volume Euler codes have a hard time suppressing unacceptable flow fields, often referred to as the carbuncle, e.g., a distorted bow shock as if a thin plate were inserted at the center perpendicular to the shock [1,2]. This happens, unfortunately, with accurate (less dissipative) flux functions such as Roe's approximate Riemann solver [3], which recognizes all wave components in the Riemann problem and introduces minimal dissipation to better resolve discontinuities [4]. Several attempts have been made to understand and cure the phenomenon [4–10], but, although they offer several useful explanations and guides to prevent it, none of them have appeared to reach yet the complete understanding and the cure of the problem [11]. There is, however, a class of flux functions that is known to be 'carbuncle-free' (if not perfectly carbuncle-free). This includes the Rusanov solver and the simplest version of the HLL Riemann solvers [12,13]. These are approximate Riemann solvers with one-wave and two-wave approximations respectively. These flux functions are very robust for inviscid calculations involving shocks, but unacceptably dissipative for shear layers, particularly for boundary layer calculations with a Navier–Stokes code.

Here, we propose a method to combine these carbuncle-free flux functions and the Roe flux function, such that shear layers are better resolved while the robustness of the dissipative fluxes is retained for shock-capturing, in a very simple but a systematic manner. To combine two flux functions, we use the rotated Riemann solver approach [14–16], which adaptively selects a direction suitable for upwinding and applies a Riemann solver along that direction, in order to capture multidimensional flow features as accurately as possible by one-dimensional physics. Originally, this approach was proposed to better resolve shocks and shear layers [14,15], but in [15] it was concluded that the gain was not very impressive in second-order accurate computations. Later, Ren [16] took this approach, not to improve accuracy, but to gain robustness for shock-capturing. He decomposed a cell-face normal into two directions: one aligned with the velocity difference vector (normal to shocks and parallel to shears) and the other orthogonal to it, and then applied the Roe solver along them. This rotated flux was shown to suppress the carbuncle by an extra dissipation introduced by the rotated flux mechanism. This, of course, does not come free; an additional cost has to be paid. He claims that the factor of CPU time is not 2 but can be made 1.5 if smartly implemented.

We follow Ren's work, but there are significant differences. One is that we apply two different Riemann solvers in the two directions. We do not rely on the additional dissipation introduced by the rotation mechanism itself, but rather apply a carbuncle-free flux function directly to the velocity difference vector direction. In the other direction, we employ the Roe solver to prevent the resulting flux from being too dissipative. This means that the combined flux becomes the carbuncle-free flux normal to shocks and the less dissipative Roe flux across shear layers. Therefore, the resulting flux has what it takes to be carbuncle-free as well as boundary-layer-resolving. Moreover, we will show that the resulting fluxes can be implemented as simple and economical as the Roe flux with an entropy fix, and they require only about 7–14% more CPU time than the Roe flux (a factor of 1.07–1.14) including the cost of computing the velocity difference vector. Another difference lies in the definition of the decomposed normal directions. It has been customary to use a cell-face normal when the velocity difference vector is too small [16]. Here, we propose to take a cell-tangent instead of normal. This does not make any differences for Ren's rotated solver (because the same flux is used in both directions), but does bring a significant impact on our rotated solvers. With this modified definition, the Roe flux will be activated instead of more dissipative solvers, for smoothly varying flows, thus leading to accuracy improvement.

The idea of hybridizing the Roe flux and the HLL flux is not new. This has already been considered by Quirk [2] to cure the carbuncle. He used a pressure gradient to detect a shock, and then switch the flux from the Roe to the HLL to avoid the carbuncle. Although very effective, the resulting algorithm does not seem to have gained popularity, apparently because of its empirical nature in the shock-sensor and also the discontinuous switching between two different fluxes. Another hybridization was proposed later by Janhunen [17] for magneto-hydrodynamics simulations, based on a positivity check on the density and the pressure. Yet, there are still other fluxes that may be considered as a combination of these two fluxes, [8,18], for example. In this paper, we construct hybrid fluxes by using the rotated Riemann solver approach, which are as simple, economical and predigested as a single Riemann solver, and do not require any problem-dependent tuning parameters.

The paper is organized as follows. In the next section, a finite-volume formulation of the Euler equations is given. In Section 3, base Riemann solvers are described, which will be combined later to construct new Riemann solvers. In Section 4, a general formulation of rotated Riemann solver is given. Then, in Section 5, we construct new flux functions by combining two different flux functions into one by the rotated Riemann solver approach, and show how they can be simplified to be comparable to the Roe solver with an entropy fix. In Section 6, the choice of the rotated direction is discussed. In Section 7, we present a series of numerical experiments in order to demonstrate the excellent performance of the new rotated-hybrid flux functions. Finally, the paper ends with concluding remarks.

2. Finite-volume discretization

Consider the Euler equations in two dimensions,

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}=\mathbf{0}, \tag{2.1}$$

$$\mathbf{U}=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{bmatrix},\quad \mathbf{F}=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uH\end{bmatrix},\quad \mathbf{G}=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vH\end{bmatrix}, \tag{2.2}$$

where ρ is the density, u and v are the velocity components in the x and y directions, respectively, and p is the static pressure. The specific total energy and enthalpy are given by

$$E=\frac{1}{\gamma-1}\frac{p}{\rho}+\frac{1}{2}(u^2+v^2), \tag{2.3}$$

$$H=\frac{\gamma}{\gamma-1}\frac{p}{\rho}+\frac{1}{2}(u^2+v^2), \tag{2.4}$$

where $\gamma=1.4$. The speed of sound, denoted by c, is given by $c=\sqrt{\gamma p/\rho}$. We consider discretizing the system with a cell-centered finite-volume algorithm over a domain dissected into a set $\{I\}$ of quadrilateral or/and triangular cells. For each cell $i\in\{I\}$, having a set of faces $\{K_i\}$, the cell-residual $\mathbf{Res}_i$ is defined as a numerical approximation of the integral over the cell of the spatial operator divided by the cell area S_i. With the midpoint rule which is sufficiently accurate for second-order accuracy, it can be written as

$$\mathbf{Res}_i=-\sum_{k\in\{K_i\}}\mathbf{\Phi}_k(\mathbf{n}_k)\Delta s_k/S_i, \tag{2.5}$$

where Δs_k is the length of cell face $k\in\{K_i\}$, $\mathbf{n}_k$ is the unit vector outward normal to that face, and $\mathbf{\Phi}_k(\mathbf{n}_k)$ is the numerical flux, which is a numerical approximation to $\mathbf{H}_{n_k}=[\mathbf{F},\mathbf{G}]\cdot\mathbf{n}_k$. The cell-averaged value, $\overline{\mathbf{U}}_i$, is then evolved by the non-zero residual in the form

$$\frac{\mathrm{d}\overline{\mathbf{U}}_i}{\mathrm{d}t}=\mathbf{Res}_i, \tag{2.6}$$

which we integrate in time by the two-stage second-order Runge–Kutta method of Shu and Osher [19],

$$\begin{aligned}
\overline{\mathbf{U}}_i^{(0)}&=\overline{\mathbf{U}}_i^{n},\\
\overline{\mathbf{U}}_i^{(1)}&=\overline{\mathbf{U}}_i^{(0)}+\Delta\mathbf{t}\mathbf{Res}_i(\overline{\mathbf{U}}_i^{(0)}),\\
\overline{\mathbf{U}}_i^{(2)}&=\frac{1}{2}\overline{\mathbf{U}}_i^{(0)}+\frac{1}{2}(\overline{\mathbf{U}}_i^{(1)}+\Delta\mathbf{t}\mathbf{Res}_i(\overline{\mathbf{U}}_i^{(1)})),\\
\overline{\mathbf{U}}_i^{n+1}&=\overline{\mathbf{U}}_i^{(2)},
\end{aligned} \tag{2.7}$$

with the time step Δt restricted by

$$\Delta t=\min_{i\in\{I\}}\left(\frac{2S_i}{\sum_{k\in\{K_i\}}|\lambda_k|_{\max}\Delta s_k}\right)\mathrm{CFL}, \tag{2.8}$$

where $|\lambda_k|_{\max}$ is the maximum characteristic wave speed normal to the cell face k.

The numerical flux, $\mathbf{\Phi}_k$, is evaluated by using the left and right states across the interface. If we take simply the cell-averaged values as the left and right states, the method will be first-order accurate in space. To achieve second-order accuracy (on structured grids), we employ Van Leer's κ-schemes [20], which upgrades the cell-averaged values to a piecewise linear/quadratic polynomial over the cell, resulting the following interface values:

$$\overline{\mathbf{U}}^L_{i+1/2} = \overline{\mathbf{U}}_i + \frac{s}{4}[(1-\kappa s)\Delta_i^-\overline{\mathbf{U}}(1+\kappa s)\Delta_i^+\overline{\mathbf{U}}], \tag{2.9}$$

$$\overline{\mathbf{U}}^R_{i+1/2} = \overline{\mathbf{U}}_{i+1} - \frac{s}{4}[(1+\kappa s)\Delta_{i+1}^-\overline{\mathbf{U}} + (1-\kappa s)\Delta_{i+1}^+\overline{\mathbf{U}}], \tag{2.10}$$

where $i+1/2$ indicates the interface of the cells i and $i+1$, and

$$s = \text{ave}(\Delta_i^-\overline{\mathbf{U}}, \Delta_i^+\overline{\mathbf{U}}), \quad \Delta_i^-\overline{\mathbf{U}} = \overline{\mathbf{U}}_i - \overline{\mathbf{U}}_{i-1}, \quad \Delta_i^+\overline{\mathbf{U}} = \overline{\mathbf{U}}_{i+1} - \overline{\mathbf{U}}_i. \tag{2.11}$$

We choose $\kappa = 1/3$ to achieve a quadratic reconstruction for smooth solutions. The resulting scheme is, however, only second-order accurate (although a smaller error is expected) due to the use of the midpoint rule to evaluate the flux integral (2.5), instead of a two-point third-order Gaussian quadrature. The averaging function is chosen as the Van Albada limiter [21],

$$\text{ave}(a,b) = \frac{2(ab+\epsilon_a)}{a^2+b^2+2\epsilon_a}, \quad \epsilon_a = 10^{-6}, \tag{2.12}$$

to prevent oscillations.

3. Base Riemann solvers

A numerical flux across each interface k, $\mathbf{\Phi}_k$, may be determined by solving exactly or approximately a Riemann problem based on the one-dimensional Euler equation in the direction of the face normal $\mathbf{n}_k$,

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{H}_{n_k}}{\partial n_k} = \mathbf{0}, \tag{3.1}$$

with the initial data, $\mathbf{U}_{\mathrm{L}}$ and $\mathbf{U}_{\mathrm{R}}$ separated by the cell interface. These initial data are simply taken as the cell-averaged states in the adjacent cells (first-order accurate) or taken from linearly extrapolated states (second-order accurate). In what follows, we omit the subscript k for clarity.

One of the Riemann solvers we employ in this study is Roe's approximate Riemann solver [3]. This is based on the Roe-averaged state $\widehat{\mathbf{U}}$ that satisfies the following:

$$\Delta \mathbf{H}_n = \widehat{\mathbf{A}}_n \Delta \mathbf{U}, \tag{3.2}$$

where $\Delta \mathbf{H}_n = \mathbf{H}_n(\mathbf{U}_{\mathrm{R}}) - \mathbf{H}_n(\mathbf{U}_{\mathrm{L}})$, $\Delta \mathbf{U} = \mathbf{U}_{\mathrm{R}} - \mathbf{U}_{\mathrm{L}}$, and $\widehat{\mathbf{A}}_n = \partial \mathbf{H}_n / \partial \mathbf{U}$ evaluated by the Roe-averaged quantities,

$$\hat{\rho} = \sqrt{\rho_{\mathrm{L}}\rho_{\mathrm{R}}}, \tag{3.3}$$

$$\hat{u} = \frac{u_{\mathrm{L}}\sqrt{\rho_{\mathrm{L}}} + u_{\mathrm{R}}\sqrt{\rho_{\mathrm{R}}}}{\sqrt{\rho_{\mathrm{L}}} + \sqrt{\rho_{\mathrm{R}}}}, \tag{3.4}$$

$$\hat{v} = \frac{v_{\mathrm{L}}\sqrt{\rho_{\mathrm{L}}} + v_{\mathrm{R}}\sqrt{\rho_{\mathrm{R}}}}{\sqrt{\rho_{\mathrm{L}}} + \sqrt{\rho_{\mathrm{R}}}}, \tag{3.5}$$

$$\widehat{H} = \frac{H_{\mathrm{L}}\sqrt{\rho_{\mathrm{L}}} + H_{\mathrm{R}}\sqrt{\rho_{\mathrm{R}}}}{\sqrt{\rho_{\mathrm{L}}} + \sqrt{\rho_{\mathrm{R}}}}, \tag{3.6}$$

$$\hat{c} = \sqrt{(\gamma-1)\left(\widehat{H} - \frac{\hat{u}^2+\hat{v}^2}{2}\right)}. \tag{3.7}$$

The interface flux is then given by

$$\mathbf{\Phi}_{\mathrm{Roe}} = \frac{1}{2}[\mathbf{H}_n(\mathbf{U}_{\mathrm{L}}) + \mathbf{H}_n(\mathbf{U}_{\mathrm{R}})] - \frac{1}{2}\widehat{\mathbf{R}}_n|\widehat{\mathbf{\Lambda}}_n|\widehat{\mathbf{L}}_n\Delta\mathbf{U}, \tag{3.8}$$

where $\widehat{\mathbf{R}}_n$ is the right eigenvector matrix of $\widehat{\mathbf{A}}_n$,

$$\widehat{\mathbf{R}}_n = \begin{bmatrix} 1 & 1 & 1 & 0 \\ \hat{u} - \hat{c}n_x & \hat{u} & \hat{u} + \hat{c}n_x & -n_y \\ \hat{v} - \hat{c}n_y & \hat{v} & \hat{v} + \hat{c}n_y & n_x \\ \widehat{H} - \hat{q}_n\hat{c} & \frac{1}{2}(\hat{u}^2 + \hat{v}^2) & \widehat{H} + \hat{q}_n\hat{c} & \hat{q}_t \end{bmatrix}, \tag{3.9}$$

with $\widehat{\mathbf{L}}_n = \widehat{\mathbf{R}}_n^{-1}$, and $|\widehat{\mathbf{\Lambda}}_n|$ is a diagonal matrix defined by

$$|\widehat{\mathbf{\Lambda}}_n| = \mathrm{diag}(|\hat{\lambda}_n^1|, |\hat{\lambda}_n^2|, |\hat{\lambda}_n^3|, |\hat{\lambda}_n^4|) = \mathrm{diag}(|\hat{q}_n - \hat{c}|, |\hat{q}_n|, |\hat{q}_n + \hat{c}|, |\hat{q}_n|), \tag{3.10}$$

with $\mathbf{n} = (n_x, n_y)$, $\hat{q}_n = (\hat{u}, \hat{v}) \cdot \mathbf{n}$ and $\hat{q}_t = (\hat{u}, \hat{v}) \cdot (-n_y, n_x)$. This is often written in the following form:

$$\mathbf{\Phi}_{\mathrm{Roe}} = \frac{1}{2}[\mathbf{H}_n(\mathbf{U}_{\mathrm{L}}) + \mathbf{H}_n(\mathbf{U}_{\mathrm{R}})] - \frac{1}{2}\sum_{k=1}^{4} |\hat{\lambda}_n^k| \hat{w}_n^k \hat{\mathbf{r}}_n^k, \tag{3.11}$$

where $\hat{\mathbf{r}}_n^k$ is the k-th column of $\widehat{\mathbf{R}}_n$, and $\hat{w}_n^k = (\widehat{\mathbf{L}}_n \Delta\mathbf{U})_k$, i.e.,

$$\hat{w}_n^1 = (\Delta p - \hat{\rho}\hat{c}\Delta q_n)/(2\hat{c}^2), \tag{3.12}$$

$$\hat{w}_n^2 = \Delta\rho - \Delta p/\hat{c}^2, \tag{3.13}$$

$$\hat{w}_n^3 = (\Delta p + \hat{\rho}\hat{c}\Delta q_n)/(2\hat{c}^2), \tag{3.14}$$

$$\hat{w}_n^4 = \hat{\rho}\Delta\hat{q}_t. \tag{3.15}$$

It is well-known that this flux admits non-physical expansion shocks. This can be avoided, for example, by modifying the modulus of the eigenvalue [22], for the nonlinear fields $k = 1, 3$ as follows:

$$|\hat{\lambda}_n^k|^* = \begin{cases} |\hat{\lambda}_n^k|, & \text{if } |\hat{\lambda}_n^k| \geqslant \delta, \\ \frac{1}{2\delta}(|\hat{\lambda}_n^k|^2 + \delta^2), & \text{if } |\hat{\lambda}_n^k| < \delta, \end{cases} \tag{3.16}$$

where $\delta = 0.2$. This flux function takes into account all waves in the system, and introduces a 'right' amount of dissipation, resulting in an excellent discontinuity-capturing property including shear waves. However, it is prone to the nonlinear instability, i.e. the carbuncle.

Another Riemann solver is the HLL Riemann solver, which refers to the simpler of the two approximate Riemann solvers proposed in [13]. This is a two-wave approximation to the exact Riemann solver (the other is a three-wave approximation), and it is given by

$$\mathbf{\Phi}_{\mathrm{HLL}} = \frac{S_{\mathrm{R}}^+ \mathbf{H}_n(\mathbf{U}_{\mathrm{L}}) - S_{\mathrm{L}}^- \mathbf{H}_n(\mathbf{U}_{\mathrm{R}})}{S_{\mathrm{R}}^+ - S_{\mathrm{L}}^-} + \frac{S_{\mathrm{R}}^+ S_{\mathrm{L}}^-}{S_{\mathrm{R}}^+ - S_{\mathrm{L}}^-}\Delta\mathbf{U}, \tag{3.17}$$

where

$$S_{\mathrm{R}}^+ = \max(0, S_{\mathrm{R}}), \quad S_{\mathrm{L}}^- = \min(0, S_{\mathrm{L}}). \tag{3.18}$$

The left and right wave speeds, S_{L} and S_{R}, are set to be the minimal and maximal eigenvalues of the Jacobian matrix $\widehat{\mathbf{A}}_n$, as proposed by Einfeldt [23],

$$S_{\mathrm{L}} = \min((q_n)_{\mathrm{L}} - c_{\mathrm{L}}, \hat{q}_n - \hat{c}), \quad S_{\mathrm{R}} = \max((q_n)_{\mathrm{R}} + c_{\mathrm{R}}, \hat{q}_n + \hat{c}), \tag{3.19}$$

where with $(q_n)_{L,R} = (u, v)_{L,R} \cdot \mathbf{n}$. With this choice, the HLL flux becomes identical to the Roe flux across an isolated shock discontinuity [23]. The resulting flux is sometimes called the HLLE flux.

The HLL flux can be written also as

$$\mathbf{\Phi}_{\mathrm{HLL}} = \frac{1}{2}[\mathbf{H}_n(\mathbf{U}_{\mathrm{L}}) + \mathbf{H}_n(\mathbf{U}_{\mathrm{R}})] + \frac{S_{\mathrm{R}}^+ S_{\mathrm{L}}^-}{S_{\mathrm{R}}^+ - S_{\mathrm{L}}^-}\Delta\mathbf{U} - \frac{1}{2}\frac{S_{\mathrm{R}}^+ + S_{\mathrm{L}}^-}{S_{\mathrm{R}}^+ - S_{\mathrm{L}}^-}\Delta\mathbf{H}_n, \tag{3.20}$$

where

$$\Delta\mathbf{H}_n = \mathbf{H}_n(\mathbf{U}_{\mathrm{R}}) - \mathbf{H}_n(\mathbf{U}_{\mathrm{L}}). \tag{3.21}$$

This flux difference can be decomposed by the Roe linearization, so that the dissipation terms merge into a single form similar to that of the Roe flux as will be shown later. Then, it can be shown that the choice of the wave speeds (3.19) results in a stable scheme, ensuring the positivity of the dissipation matrix (see [23]). This flux function is known to be carbuncle-free [4], but more dissipative than the Roe flux, particularly for shear layers.

We consider also the Rusanov solver [12]. This is a one-wave approximate Riemann solver, which can be derived from either of the two solvers described above, by reducing the number of waves. To derive from the Roe flux, simply replace the eigenvalue matrix $|\widehat{\mathbf{\Lambda}}_n|$ by a single scalar $\widehat{S}_n$ defined as the spectral radius of $\widehat{\mathbf{A}}_n$,

$$\widehat{S}_n = \rho(\widehat{\mathbf{A}}_n) = \max(|\hat{u}_n + \hat{c}|, |\hat{u}_n - \hat{c}|) = |\hat{u}_n| + \hat{c}. \tag{3.22}$$

That is to say, we set

$$|\widehat{\mathbf{\Lambda}}_n| = \widehat{S}_n \mathbf{I}, \tag{3.23}$$

where $\mathbf{I}$ is the identity matrix, and we obtain, from (3.8),

$$\mathbf{\Phi}_{\text{Rusanov}} = \frac{1}{2}[\mathbf{H}_n(\mathbf{U}_{\text{L}}) + \mathbf{H}_n(\mathbf{U}_{\text{R}})] - \frac{1}{2}\widehat{\mathbf{R}}_n(\widehat{S}_n\mathbf{I})\widehat{\mathbf{L}}_n\Delta\mathbf{U} \tag{3.24}$$

$$= \frac{1}{2}[\mathbf{H}_n(\mathbf{U}_{\text{L}}) + \mathbf{H}_n(\mathbf{U}_{\text{R}})] - \frac{\widehat{S}_n}{2}\Delta\mathbf{U}. \tag{3.25}$$

This is the Rusanov flux (also sometimes called the Lax–Friedrichs flux or the local Lax–Friedrichs flux, depending on the definition of $\widehat{S}_n$). The resulting scheme is stable in the sense that the coefficient in the dissipation term, i.e., $\widehat{S}_n$ is positive. Note that this is even more dissipative than the HLL and Roe fluxes (although equivalent to the Roe and HLL fluxes in the case of an isolated shock associated with the speed $|\hat{\lambda}_n^3|$), but this is also known as robust as the HLL flux, i.e., 'carbuncle-free'.

4. General form of rotated Riemann solvers

In the standard finite-volume approach, no matter how we solve the Riemann problem, the direction for the one-dimensional problem is completely fixed by the mesh geometry. Obviously, this grid-aligned normal may not always be appropriate for upwinding. Then, methods of rotated Riemann solvers have been proposed in order to choose freely the direction of the one-dimensional Riemann problem for an improved accuracy (see [15] and references therein). Although these methods were originally introduced to improve accuracy for shock-capturing, our purpose is mainly to gain robustness rather than better accuracy, in the same spirit as in [16]. Better accuracy can also be obtained in our approach, by a smart choice of the direction (see Section 6). In the following, we describe a prototype of rotated Riemann solvers, which we will see later can be dramatically simplified by combining two different fluxes.

Rotated Riemann solvers are based on the decomposition of the normal vector $\mathbf{n}$ into two orthogonal directions. Suppose we have selected $\mathbf{n}_1$ (the choice will be discussed in Section 6). Then, we align the other direction $\mathbf{n}_2$ as perpendicular to $\mathbf{n}_1$, i.e.,

$$\mathbf{n}_1 \cdot \mathbf{n}_2 = 0, \tag{4.1}$$

where we assume also that $|\mathbf{n}_1| = |\mathbf{n}_2| = 1$. The cell-face normal $\mathbf{n}$ is now projected onto these orthogonal directions,

$$\mathbf{n} = \alpha_1\mathbf{n}_1 + \alpha_2\mathbf{n}_2, \tag{4.2}$$

where $\alpha_1 = \mathbf{n}\cdot\mathbf{n}_1$, $\alpha_2 = \mathbf{n}\cdot\mathbf{n}_2$. In doing this, we choose the vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ such that $\alpha_1 \geqslant 0$ and $\alpha_2 \geqslant 0$, so that we keep the same left and right states in both directions (see Fig. 1). The interface flux is then decomposed as

$$\mathbf{\Phi} = \mathbf{\Phi}(\mathbf{n}) = \alpha_1\mathbf{\Phi}(\mathbf{n}_1) + \alpha_2\mathbf{\Phi}(\mathbf{n}_2). \tag{4.3}$$

To define the interface flux, we need to choose numerical fluxes in two directions, $\mathbf{\Phi}(\mathbf{n}_1)$ and $\mathbf{\Phi}(\mathbf{n}_2)$. These can be determined by solving two Riemann problems: one in the direction of $\mathbf{n}_1$ and the other in the

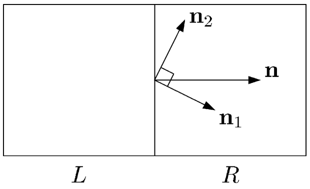

Fig. 1. Decomposition of the face normal.

direction of $\mathbf{n}_2$. For example, if we use the Roe solver for both directions, we obtain the rotated Riemann solver of Ren [16]

$$\mathbf{\Phi}_{\text{Ren}} = \frac{1}{2}[\mathbf{H}_n(\mathbf{U}_L) + \mathbf{H}_n(\mathbf{U}_R)] - \frac{1}{2}\sum_{m=1,2} \alpha_m \widehat{\mathbf{R}}_{n_m} |\widehat{\mathbf{\Lambda}}_{n_m}|^* \widehat{\mathbf{L}}_{n_m} \Delta\mathbf{U}, \tag{4.4}$$

where $|\widehat{\mathbf{\Lambda}}_{n_m}|^*$ is the diagonal matrix of the modified eigenvalues (3.16). Observe that compared with the original formula only the dissipation term is different. Its property depends on the choice of the direction $\mathbf{n}_1$. For example, Ren [16] showed by taking $\mathbf{n}_1$ as a velocity difference vector that this flux function introduces more dissipation in the linear field and prevents carbuncles effectively. Obviously this is more expensive than the original one, but the cost can be minimized, for example by sharing the averaged state and the eigenvector of the entropy wave in the two directions. Ren [16] claims that the cost can be made 1.5 times the original Roe flux. In this paper, we construct rotated flux functions based on the same rotated Riemann solver approach whose cost is nearly the same as the Roe flux.

5. Rotated-hybrid Riemann solvers

We now construct new flux functions by combining the Roe flux and the fewer-wave solvers, by applying them in the two normal directions in the framework of the rotated Riemann solvers.

Firstly, we consider employing the Roe flux function (3.8) in $\mathbf{n}_2$ and the Rusanov flux (3.24) in $\mathbf{n}_1$. This gives

$$\mathbf{\Phi}_{\text{RR}} = \alpha_1 \mathbf{\Phi}_{\text{Rusanov}}(\mathbf{n}_1) + \alpha_2 \mathbf{\Phi}_{\text{Roe}}(\mathbf{n}_2) \tag{5.1}$$

$$= \frac{1}{2}[\mathbf{H}_n(\mathbf{U}_L) + \mathbf{H}_n(\mathbf{U}_R)] - \frac{\alpha_2}{2} \widehat{\mathbf{R}}_{n_2} |\widehat{\mathbf{\Lambda}}_{n_2}|^* \widehat{\mathbf{L}}_{n_2} \Delta\mathbf{U} - \frac{\alpha_1}{2} \widehat{\mathbf{R}}_{n_1} (\widehat{S}_{n_1} \mathbf{I}) \widehat{\mathbf{L}}_{n_1} \Delta\mathbf{U}. \tag{5.2}$$

Since $\widehat{\mathbf{R}}_{n_1} \mathbf{I} \widehat{\mathbf{L}}_{n_1} = \mathbf{I} = \widehat{\mathbf{R}}_{n_2} \mathbf{I} \widehat{\mathbf{L}}_{n_2}$, this can be combined into the following:

$$\mathbf{\Phi}_{\text{RR}} = \frac{1}{2}[\mathbf{H}_n(\mathbf{U}_L) + \mathbf{H}_n(\mathbf{U}_R)] - \frac{1}{2} \widehat{\mathbf{R}}_{n_2} (\alpha_2 |\widehat{\mathbf{\Lambda}}_{n_2}|^* + \alpha_1 \widehat{S}_{n_1} \mathbf{I}) \widehat{\mathbf{L}}_{n_2} \Delta\mathbf{U}, \tag{5.3}$$

which we write as

$$\mathbf{\Phi}_{\text{RR}} = \frac{1}{2}[\mathbf{H}_n(\mathbf{U}_L) + \mathbf{H}_n(\mathbf{U}_R)] - \frac{1}{2}\sum_{k=1}^{4} |\hat{s}_{\text{RR}}^k| \hat{w}_{n_2}^k \hat{\mathbf{r}}_{n_2}^k, \tag{5.4}$$

where

$$|\hat{s}_{\text{RR}}^k| = \alpha_2 |\hat{\lambda}_{n_2}^k|^* + \alpha_1 \widehat{S}_{n_1}. \tag{5.5}$$

Since α_1, α_2 and $\widehat{S}_{n_1}$ are all non-negative, this scheme is stable. Note that this flux is in the form of the Roe flux with *modified* wave speeds (compare this with (3.11)). In particular, the eigenvectors and the wave strengths are needed to be computed for $\mathbf{n}_2$ only. The vector $\mathbf{n}_1$ is needed only to compute the wave speed $\widehat{S}_{n_1}$. Hence, the cost is comparable to the Roe flux with an entropy fix incorporated in the wave speeds. We will refer to this flux as the Rotated-Roe–Rusanov (Rotated-RR) flux.

Secondly, we consider employing the HLL flux instead of the Rusanov. This will improve the shock-capturing capability in the direction of $\mathbf{n}_1$ as the HLL solver can capture isolated shock discontinuities. Applying the Roe flux function (3.8) in $\mathbf{n}_2$ and the HLL flux (3.20) in $\mathbf{n}_1$, we get

$$\mathbf{\Phi}_{\text{RHLL}} = \alpha_1 \mathbf{\Phi}_{\text{HLL}}(\mathbf{n}_1) + \alpha_2 \mathbf{\Phi}_{\text{Roe}}(\mathbf{n}_2) \tag{5.6}$$

$$= \frac{1}{2}[\mathbf{H}_n(\mathbf{U}_{\text{L}}) + \mathbf{H}_n(\mathbf{U}_{\text{R}})] - \frac{\alpha_2}{2}\widehat{\mathbf{R}}_{n_2}|\widehat{\mathbf{\Lambda}}_{n_2}|^* \widehat{\mathbf{L}}_{n_2}\Delta\mathbf{U} + \alpha_1 \frac{S_{\text{R}}^+ S_{\text{L}}^-}{S_{\text{R}}^+ - S_{\text{L}}^-}\Delta\mathbf{U} - \frac{\alpha_1}{2}\frac{S_{\text{R}}^+ + S_{\text{L}}^-}{S_{\text{R}}^+ - S_{\text{L}}^-}\Delta\mathbf{H}_{n_1}, \tag{5.7}$$

where S_{R}^+ and S_{L}^- are to be computed with $\mathbf{n}_1$. This is again stable since each dissipation matrix has positive eigenvalues, as described in Section 3. To further simplify this, we use the identity,

$$\Delta\mathbf{H}_n = \alpha_1 \Delta\mathbf{H}_{n_1} + \alpha_2 \Delta\mathbf{H}_{n_2}, \tag{5.8}$$

which is rewritten (by the Roe linearization [3], i.e., $\Delta\mathbf{H}_{n_2} = \widehat{\mathbf{A}}_{n_2}\Delta\mathbf{U} = \widehat{\mathbf{R}}_{n_2}\widehat{\mathbf{\Lambda}}_{n_2}\widehat{\mathbf{L}}_{n_2}\Delta\mathbf{U}$) as

$$\alpha_1 \Delta\mathbf{H}_{n_1} = \Delta\mathbf{H}_n - \alpha_2 \widehat{\mathbf{R}}_{n_2}\widehat{\mathbf{\Lambda}}_{n_2}\widehat{\mathbf{L}}_{n_2}\Delta\mathbf{U}, \tag{5.9}$$

to replace the last term in (5.7), resulting

$$\mathbf{\Phi}_{\text{RHLL}} = \frac{S_{\text{R}}^+ \mathbf{H}_n(\mathbf{U}_{\text{L}}) - S_{\text{L}}^- \mathbf{H}_n(\mathbf{U}_{\text{R}})}{S_{\text{R}}^+ - S_{\text{L}}^-} - \frac{1}{2}\widehat{\mathbf{R}}_{n_2}\left[\alpha_2|\widehat{\mathbf{\Lambda}}_{n_2}|^* - \alpha_2 \frac{S_{\text{R}}^+ + S_{\text{L}}^-}{S_{\text{R}}^+ - S_{\text{L}}^-}\widehat{\mathbf{\Lambda}}_{n_2} - 2\alpha_1 \frac{S_{\text{R}}^+ S_{\text{L}}^-}{S_{\text{R}}^+ - S_{\text{L}}^-}\right]\widehat{\mathbf{L}}_{n_2}\Delta\mathbf{U}. \tag{5.10}$$

We thus arrive at the following very simple formula:

$$\mathbf{\Phi}_{\text{RHLL}} = \frac{S_{\text{R}}^+ \mathbf{H}_n(\mathbf{U}_{\text{L}}) - S_{\text{L}}^- \mathbf{H}_n(\mathbf{U}_{\text{R}})}{S_{\text{R}}^+ - S_{\text{L}}^-} - \frac{1}{2}\sum_{k=1}^{4}|\hat{s}_{\text{RHLL}}^k|\hat{w}_{n_2}^k \hat{\mathbf{r}}_{n_2}^k, \tag{5.11}$$

where

$$|\hat{s}_{\text{RHLL}}^k| = \alpha_2|\hat{\lambda}_{n_2}^k|^* - \frac{1}{S_{\text{R}}^+ - S_{\text{L}}^-}[\alpha_2(S_{\text{R}}^+ + S_{\text{L}}^-)\hat{\lambda}_{n_2}^k + 2\alpha_1 S_{\text{R}}^+ S_{\text{L}}^-]. \tag{5.12}$$

Comparing this with the HLL flux in (3.17), we observe that this is as if a matrix dissipation has been introduced into the HLL flux. This implies a better resolution than the HLL flux itself can provide. We find it interesting that a flux proposed in [8], called RoeM2, has a very similar structure, i.e., the HLL with a detailed dissipation term, although the mechanism of the dissipation seems very different. Alternatively, this rotated flux may be viewed as a Roe-type flux with *modified* wave speeds and the arithmetic average of the physical flux replaced by the wave-speed average. Again, the cost is expected to be comparable to the Roe flux with modified wave speeds. We will refer to this flux as the Rotated-Roe-HLL (Rotated-RHLL) flux.

We point out that it is extremely simple to convert an existing Roe flux code into these rotated-hybrid fluxes. After computing as usual the left and right physical fluxes, $\mathbf{H}_n(\mathbf{U}_{\text{L}})$ and $\mathbf{H}_n(\mathbf{U}_{\text{R}})$, we simply redefine the interface normal as $\mathbf{n}_2$, and compute the dissipation term exactly as in the Roe flux with *modified* wave speeds (5.5) or (5.12).

6. Choice of $\mathbf{n}_1$

To completely define our rotated-hybrid fluxes, the normal direction $\mathbf{n}_1$ must be determined at every interface. We assume always that $\mathbf{n}_2$ is taken as an orthogonal direction with $\alpha_2 \geqslant 0$. In this work, as in the previous works [15,16], we consider aligning $\mathbf{n}_1$ with the velocity difference vector taken over two adjacent cells,

$$\mathbf{n}_1 = \begin{cases} \dfrac{\Delta\vec{q}}{\|\Delta\vec{q}\|}, & \text{if } \|\Delta\vec{q}\| > \epsilon, \\ \mathbf{n}, & \text{otherwise,} \end{cases} \tag{6.1}$$

where $\Delta\vec{q} = (\Delta u, \Delta v) = (u_{\text{R}} - u_{\text{L}}, v_{\text{R}} - v_{\text{L}})$, $\|\Delta\vec{q}\| = \sqrt{(\Delta u)^2 + (\Delta v)^2}$ and ϵ is a small number. This choice is known to select the direction normal to a shock or parallel to a shear if they exist [15].

In the case of a shock, this means that our rotated Riemann solvers become the Rusanov/HLL Riemann solver in the direction normal to the shock, and the Roe flux in the parallel direction (see Fig. 2). It may be argued that the nonlinear instability is caused by a transverse perturbation and therefore a dissipative flux should be applied parallel to the shock [6]. In fact, the rotated fluxes designed this way do it automatically: for a cell-face perpendicular to a shock, any significant transverse disturbance will orient $\mathbf{n}_1$ toward the face normal (creating non-zero α_1), and thus the Rusanov/HLL solver is applied parallel to the shock by a factor of

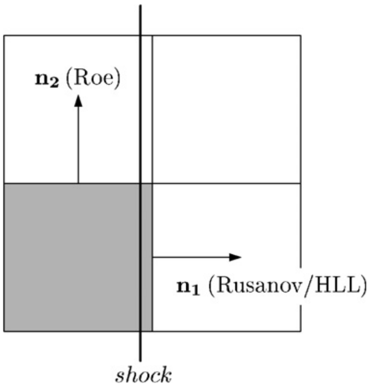

Fig. 2. Selected normals for a grid-aligned shock wave, by (6.2).

α_1. Note that our rotated Riemann solvers are always more dissipative than the Roe flux whenever there exists a significant amount of velocity difference in the direction normal to the cell-face (so that $\alpha_1 \neq 0$) because they introduce the Rusanov/HLL solver in that direction. On the other hand, it may also be argued that the transverse perturbation is caused by a perturbation normal to the shock, implying that a dissipative flux normal to the shock will prevent the nonlinear instability in its early stage, as analyzed in [24]. In this viewpoint, our rotated fluxes will eliminate the nonlinear instability by applying the Rusanov/HLL flux directly normal to the shock. Either way, numerical experiments indicate that our rotated Riemann solvers completely eliminate the shock instability for a wide range of problems.

On the other hand, in the case of a shear layer, the rotated fluxes are in general more dissipative than the Roe flux. However, if the cell interface is aligned with the shear layer, as is often the case with boundary layer calculations, we will have $\alpha_1 = 0$. This turns off the dissipative fewer-wave solver, and we end up with the standard grid-aligned Roe flux across the shear layer (see Fig. 3). This will have a great impact on the resolution of the boundary layer because it is the cross diffusion, rather than the streamwise diffusion, that contaminates the solution inside the boundary layer. For the face oriented normal to the shear layer, on the other hand, the rotated flux reduces to a grid-aligned dissipative fewer-wave solver (see Fig. 3). But in this case, the solution variation is expected to be small compared with the cross-variation, and therefore its effect on the resolution would be very small. Our numerical experiments confirmed this assertion. It is still wished, nevertheless, that the rotated flux fully becomes the Roe flux also in the streamwise direction. This may be realized for our rotated Riemann solvers, by the following modified definition of $\mathbf{n}_1$:

$$\mathbf{n}_1 = \begin{cases} \dfrac{\Delta\vec{q}}{\|\Delta\vec{q}\|}, & \text{if } \|\Delta\vec{q}\| > \epsilon, \\ \mathbf{n}_\perp, & \text{otherwise,} \end{cases} \tag{6.2}$$

where $\mathbf{n}_\perp$ denotes the tangential direction to the geometric face normal. This differs from the previous one in the alternative direction for a small velocity difference, i.e., the cell-face tangent instead of the normal. It does

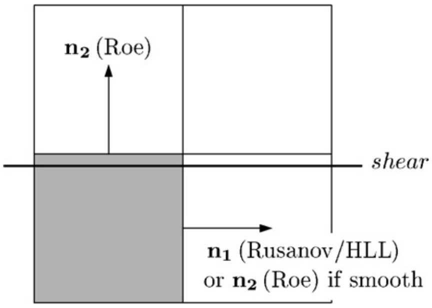
$\mathbf{n_2}$ (Roe)
shear
$\mathbf{n_1}$ (Rusanov/HLL)
or $\mathbf{n_2}$ (Roe) if smooth

Fig. 3. Selected normals for a grid-aligned shear wave.

not cause any change to the Ren's rotated Riemann solver because his solver uses the Roe flux in both directions, but it does bring a significant change into our rotated Riemann solvers. Specifically, this effectively turns off the dissipative solver (since $\alpha_1 = \mathbf{n} \cdot \mathbf{n}_1 = \mathbf{n} \cdot \mathbf{n}_\perp = 0$) and recovers the grid-aligned Roe flux for a smoothly varying streamwise velocity field (Fig. 3). Note that for smooth flows we have

$$\|\Delta\vec{q}\| = \mathrm{O}(h), \tag{6.3}$$

where h is a mesh size. This means that the more we refine the mesh the more the Roe flux will be used, except at discontinuities for which the velocity difference remains finite as $h \to 0$. This is even more so for second-order schemes, in which we use extrapolated values (2.9) and (2.10) to compute the velocity difference vector. In this case, it is easy to show that

$$\|\Delta\vec{q}\| = \mathrm{O}(h^3) \tag{6.4}$$

for smooth flows. Therefore, it is much more likely in the second-order schemes than in the first-order schemes that the Roe flux will be used in most places for sufficiently resolved flows and the dissipative flux will be introduced only at discontinuities. In fact, numerical experiments show that the condition $\|\Delta\vec{q}\| > \epsilon$ is satisfied only near discontinuities or strongly varying flow regions, and this is not sensitive to the value of ϵ unless it is exceptionally large. This is consistent with Ren's results [16]. In this work, we set $\epsilon = 10^{-12}U_{\mathrm{ref}}$ (simply a very small number), where U_{ref} is a reference velocity magnitude, using the modified definition of $\mathbf{n}_1$ (6.2) for all problems, and demonstrate that our rotated fluxes are fully capable without problem-dependent tuning of ϵ. It should be noted here that once such an extremely small value is chosen for ϵ, the two choices, (6.1) and (6.2), would not make a big difference in the results because when $\|\Delta\vec{q}\| < \epsilon$ is true, the left and right states will be almost identical, implying that the choice of flux functions is insignificant.

This choice of normals, as being dependent of the solution, introduces nonlinearity in the algorithm, and therefore is likely to cause convergence difficulties for steady calculations. To alleviate this, we re-calculate the normals at every time step only until the residual is reduced by an order of magnitude from the start-up, and freeze them thereafter. But we may re-calculate them a few times more: once when the residual shows reduction of two orders of magnitude and also once when it shows three orders of magnitude reduction. This will better adapt the normals to a converged flow solution. A similar strategy was taken in [15] where the normals were re-calculated at every 200 iterations.

It should be kept in mind that our rotated-hybrid fluxes do not simply switch between the Roe flux and the Rusanov/HLL flux. With the definition (6.2), the rotated-hybrid fluxes can be expressed as follows:

$$\mathbf{\Phi}(\mathbf{n}) = \begin{cases} \alpha_1 \mathbf{\Phi}_{\mathrm{Rusanov/HLL}}(\mathbf{n}_1) + \alpha_2 \mathbf{\Phi}_{\mathrm{Roe}}(\mathbf{n}_2), & \text{if } \|\Delta\vec{q}\| > \epsilon, \\ \mathbf{\Phi}_{\mathrm{Roe}}(\mathbf{n}), & \text{otherwise.} \end{cases} \tag{6.5}$$

In this form, it is clear that even if the rotated-solver is activated (when $\|\Delta\vec{q}\| > \epsilon$), it introduces the Rusanov/HLL flux only if $\alpha_1 \neq 0$ and only by a factor of α_1 ($\leqslant 1$). The flux becomes fully identical to the Rusanov/HLL flux only if $\alpha_1 = 1$, i.e., the velocity difference vector is perfectly aligned with the cell-face normal (e.g., a grid-aligned normal shock). This indicates that the rotated-hybrid fluxes are generally much less dissipative than the Rusanov/HLL fluxes themselves.

A final remark is on the extension of the rotated Riemann solvers to three dimensions. This can be done in a very straightforward manner as pointed out by Ren [16]. It only amounts to defining $\mathbf{n}_2$ as perpendicular to the principal direction $\mathbf{n}_1$ chosen exactly as above. For example, the following formula will do:

$$\mathbf{n}_2 = \frac{(\mathbf{n}_1 \times \mathbf{n}) \times \mathbf{n}_1}{\|(\mathbf{n}_1 \times \mathbf{n}) \times \mathbf{n}_1\|}. \tag{6.6}$$

This is applicable to any types of cells, e.g., hexahedra or tetrahedra. Incidentally, our rotated fluxes offer nothing new for one-dimensional grids. They are inherently multidimensional.

7. Results

We made an extensive series of numerical experiments using the rotated-hybrid Riemann solvers. In the first subsection, we present results for six shock instability problems, with both first-order and second-order schemes, including also an unstructured grid case. A focus is on shock instabilities here. In the second subsection, we present results for three problems which involve shear layers. Obviously, a focus here is on accuracy for resolving shear layers.

All computations were performed with double-precision. The rotated-hybrid fluxes, the Rotated-RR and the Rotated-RHLL, were implemented in the forms (5.4) and (5.11) respectively. We employed the modified definition of $\mathbf{n}_1$ (6.2) with $\epsilon = 10^{-12}U_{\mathrm{ref}}$ to determine the normals for *all* test problems. For steady calculations, we freeze the normals after the density residual experiences an order of magnitude reduction, and re-compute once each when the residual goes down two and three orders of magnitude. For second-order schemes, we used extrapolated values (2.9) and (2.10) to compute the velocity difference vector in (6.2).

7.1. Shock instability problems

7.1.1. Quirk's odd–even grid perturbation problem

This is a well-known test case first studied by Quirk [2], and later by several others [6,16,25]. A single shock travels from the left to the right at Mach 6, in a domain covered by a 20 × 800 structured grid with unit spacing, involving the following grid perturbations:

$$y(i,j\mathrm{mid}) = \begin{cases} y_{\mathrm{mid}} + 0.001h & \text{for } i \text{ even}, \\ y_{\mathrm{mid}} - 0.001h & \text{for } i \text{ odd}, \end{cases} \tag{7.1}$$

where $y(i,j\mathrm{mid})$ is the y coordinate of a vertex $(i,10)$, y_{mid} is the y coordinate of the halfway line, and h is the cell size. This simple flow with such extremely small perturbations is known to cause many upwind flux functions to fail catastrophically (see [6]). Here, we compare our rotated-hybrid fluxes with the Roe flux in the first-order scheme. Computations were done for 3000 time steps with $\mathrm{CFL} = 0.4$. The shock was set initially at $i = 25$. The normals were computed at every time step and never frozen. As shown in Fig. 4, the shock was destroyed by the Roe flux, and a typical carbuncle was developed. On the other hand, the Rotated-RHLL flux did not suffer and kept the shock all the way through. The Rotated-RR flux was also found to be free of instability. The result of the Rotated-RR is almost perfectly identical to that by the Rotated-RHLL, and therefore not shown.

7.1.2. Steady normal shock instability problem

This is a steady shock test case taken from [11]. A stationary normal shock sits on a 50 × 20 two-dimensional grid with uniform spacings. Unless the shock is located at a mesh interface, there must be at least one intermediate state. To investigate the effect of the shock position, an initial solution is set up with a varied intermediate state

$$\rho_M = \delta_I \rho_{\mathrm{L}} + (1 - \delta_I)\rho_{\mathrm{R}}, \tag{7.2}$$

where ρ_M, ρ_{L} and ρ_{R} are the density in the intermediate, left and right states respectively, and $\delta_I = 0.0, 0.1, 0.2, \ldots, 0.9$ is a shock position parameter, e.g., $\delta_I = 0.0$ corresponds to a shock exactly on a mesh interface. It was reported in [11] that all the fluxes they tested (the Roe flux, the HLL flux, AUSM$^+$ [26], AUSM$^+$-up [27], AUSMPW+ [28], RoeM2 [8], and an entropy-consistent flux [10,29]) were not able to keep

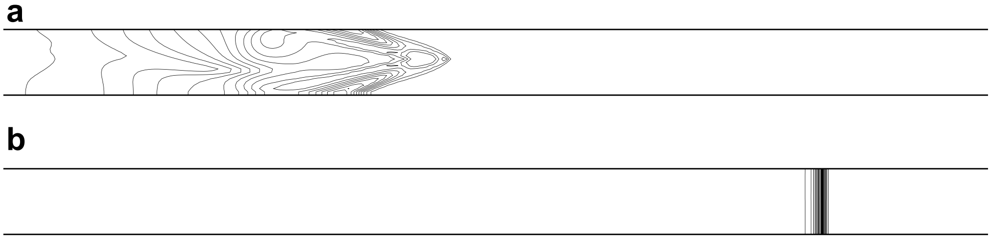

Fig. 4. Odd–even instability: density contours at 2000 time steps. (a) Roe with E-fix and (b) Rotated-RHLL.

the initial shock for all shock locations, creating a carbuncle or significantly non-uniform solutions in y-direction.

We specify a uniform state with Mach 6 on the left. Then, all other quantities on the left and right states are figured automatically as described in [9]. The density in the intermediate cell is given by (7.2), and then all other intermediate quantities are calculated by the Hugoniot curve connecting the left and right states (see [5]). The top and bottom boundaries are taken as periodic. At the outflow boundary, the mass flux $\rho u = 1.0$ is specified in the ghost cells for the purpose of preserving the initial shock location, and other quantities are simply copied from the adjacent interior cells. Here again, we compare our rotated-hybrid fluxes with the Roe flux in the first-order scheme. All calculations were stopped at 40,000 time steps with $\mathrm{CFL} = 0.5$. The normals were computed at every time step and never frozen. The results for $\delta_I = 0.0$ and 0.4 are shown in Fig. 5. The Roe flux failed to preserve the initial shock for all values of δ_I, resulting a carbuncle. For other shock locations, the solution developed the same kind of carbuncle and it was quickly washed away out of the domain through the downstream boundary. For this reason, we do not show other results. On the other hand, both the Rotated-RR and Rotated-RHLL fluxes reduced the residuals by 6–7 orders of magnitude, and successfully preserved the initial shock for *all shock locations*. Results are shown in Fig. 5 for the Rotated-RHLL flux with $\delta_I = 0.0$ and 0.4 only. Other results, for other locations as well as by the Rotated-RR, are similar to these results and therefore omitted.

7.1.3. $M_\infty = 8$ *flow over a cylinder*

We consider $M_\infty = 8$ flow over a cylinder. This is yet another test case for which the Roe flux suffers from the carbuncle. The grid is 160 × 80 structured grid, as shown in Fig. 6 (every other points are displayed for better visibility). The first-order scheme was used with the rotated-hybrid fluxes and the Roe flux. As expected, the Roe flux created a distorted solution as shown in Fig. 7(a). On the other hand, our rotated-hybrid fluxes produced perfectly acceptable solutions shown in Fig. 7(b) and (c). We remark that the solutions are fully converged for all cases ($\mathrm{CFL} = 0.5$): the density residuals dropped to 10^{-16} (see Fig. 8). For the rotated-hybrid fluxes, freezing the normal vectors after an order of magnitude reduction of the residuals was essential for a full convergence. If we continued to re-compute the normals at every time step, the residuals would stagnate after a few orders of magnitude reduction (although the solutions are very similar to fully converged ones).

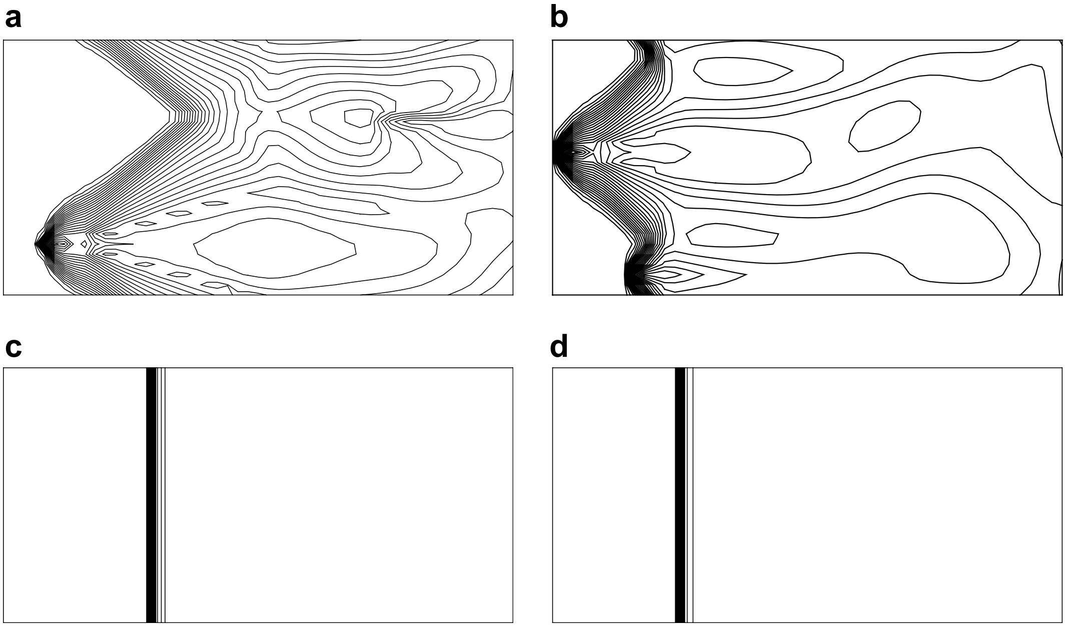

Fig. 5. Steady normal shock instability problem: Mach contours. (a) Roe with E-fix: $\delta_I = 0.0$, (b) Roe with E-fix: $\delta_I = 0.4$, (c) Rotated-RHLL: $\delta_I = 0.0$ and (d) Rotated-RHLL: $\delta_I = 0.4$.

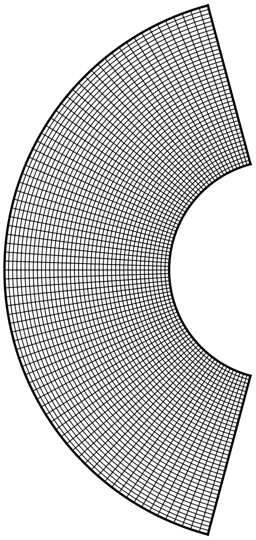

Fig. 6. Computational grid.

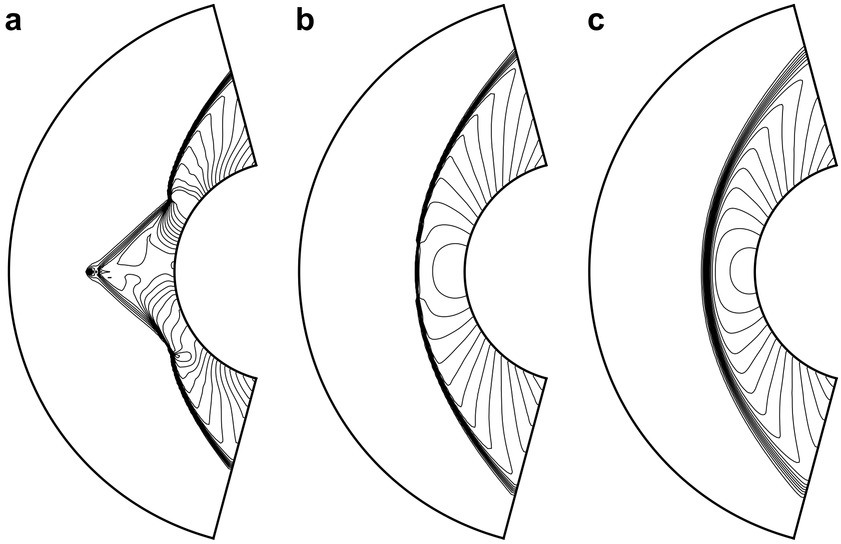

Fig. 7. $M_\infty = 8$ flow over a cylinder: pressure contours. (a) Roe with E-fix, (b) Rotated-RHLL and (c) Rotated-RR.

7.1.4. Unstructured grid case: $M_\infty = 20$ hypersonic flow over a cylinder

Here, we apply our rotated-hybrid fluxes to a more severe case: $M_\infty = 20$ hypersonic flow over a cylinder on an unstructured triangular grid. We used a first-order cell-centered finite-volume Euler code with the Roe, the Rotated-RR and the Rotated-RHLL implemented to compute the interface flux across a side of a triangular element. The grid is a Delaunay triangulation with 11,280 cells and 5881 nodes (160 nodes on the cylinder), and completely irregular (see Fig. 9: a blow-up of the complete grid). We found that the Roe flux created a carbuncle as shown in Fig. 10. On the other hand, the Rotated-RHLL flux did not allow the carbuncle to appear and produced a correct solution (see Fig. 11). We also found, to be somewhat unexpected, that the Rotated-RR flux created, although weaker, a carbuncle (see Fig. 12). This indicates that although generally the carbuncle is suppressed by adding an extra dissipation, adding it too much also provokes the carbuncle. This agrees with results reported in [11]. We remark also that all computations are fully converged (of the

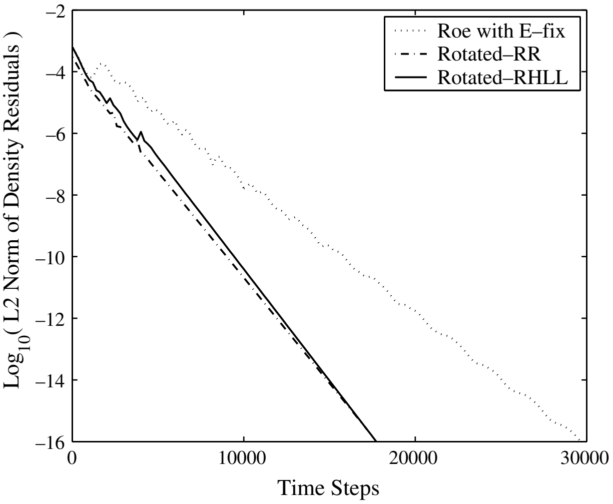

Fig. 8. Convergence histories.

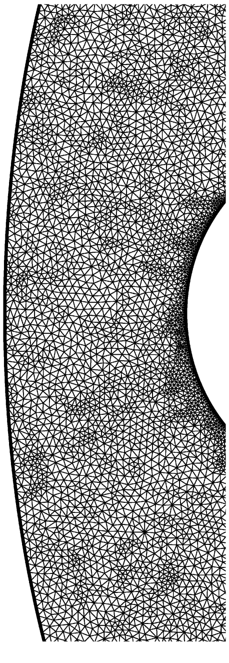

Fig. 9. Unstructured grid.

order of 10^{-16} after 50,000 to 60,000 time steps), except for the Roe flux where the residuals stagnated after two orders of magnitude reduction.

7.1.5. Shock diffraction

A shock diffraction problem is another test case for which many Godunov-type fluxes are known to fail [18]. A Mach 5.09 right-running normal shock diffracts around a 90° corner. We use a square domain, and locate the corner at the midpoint of the left boundary: the lower half is treated as a wall; the top half is taken as an inflow. The top boundary is taken as a wall also, and the right and bottom boundaries are taken as outflow. The grid is 400×400 structured. All computations were performed by second-order schemes with $\mathrm{CFL} = 0.95$ up to $t = 0.18$. Results are shown in Fig. 13. It is evident that the Roe flux suffers from a shock instability. But the rotated-hybrid fluxes do not suffer at all. It is also observed, in this example, that the

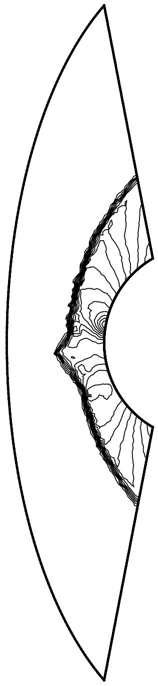

Fig. 10. Roe with E-fix.

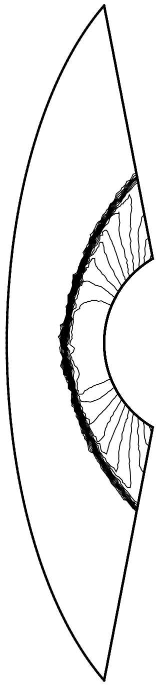

Fig. 11. Rotated-RHLL.

Rotated-RR flux results in a somewhat more dissipative solution than that of the Rotated-RHLL flux. It is notably dissipative near the left wall where a shear layer is developing. But at the same time, we observe that the shocks are captured by Rotated-RR flux as crisply as those by the HLL flux for second-order schemes

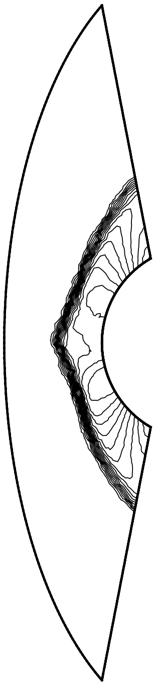

Fig. 12. Rotated-RR.

(compare these with first-order results 8(b) and (c)). Shown in Fig. 13(d) is a vector plot of $\alpha_1 \mathbf{n}_1$ (on a finer grid with the face midpoints included as nodes). This is for the Rotated-RHLL flux but essentially the same for the Rotated-RR flux. It may be hard to see the directions of plotted vectors, but they are not very important (they are actually well aligned with shock normals; this has already been shown in [16]). Rather, the purpose of this plot here is to show that the Rusanov/HLL flux has been introduced only in limited regions, i.e., discontinuities and strongly varying flow fields (observe that $\alpha_1 \approx 0$ in a large part of the domain). This confirms that the rotated-hybrid fluxes reduce to the Roe flux for smoothly varying flow fields. It also suggests that the quantity $\alpha_1 \mathbf{n}_1$ could serve as a feature detector for adaptive mesh refinement. Note in particular that it carries information about feature directions, e.g., shock normals. This is a valuable piece of information for anisotropic adaptation.

7.1.6. Double-Mach reflection

It is well known that the Roe flux produces a spurious triple point, also known as the kinked Mach stem, in the double-Mach reflection problem. This test case was first studied by Woodward and Colella [30], and later by many others. We use the setup of Quirk [2]: Mach 5.5 right-running normal shock wave hits a 30° ramp. The grid is 400 × 200 structured grid. Setting the shock at the left boundary at $t = 0$, we computed a solution at $t = 4$ by second-order schemes. All computations were performed with CFL $= 0.7$. As shown in Fig. 14, the Roe flux produced the kinked Mach stem, but both the Rotated-RHLL and Rotated-RR fluxes did not suffer such a failure. Also observed is that the Rotated-RR flux gives a bit more dissipative results (near the wall), as in the previous test case. Shown in Fig. 14(d) is a vector plot of $\alpha_1 \mathbf{n}_1$. Exactly as in the previous test case, we observe that the Rusanov/HLL flux has been introduced only near discontinuities and regions of strong variation. Again, we see that the quantity $\alpha_1 \mathbf{n}_1$ captured the flow feature very well, and can be very useful for grid adaptation.

For this test case, we also made a cost comparison. We calculated an average CPU time per time step for the three different fluxes. This includes computations of the normals also. The result is that the Rotated-RR flux and the Rotated-RHLL flux cost respectively 1.07 and 1.14 times more CPU time than the Roe flux. We remark that the Rotated-RHLL is more expensive simply by the cost of max/min functions used to compute S_L^- and S_R^+.

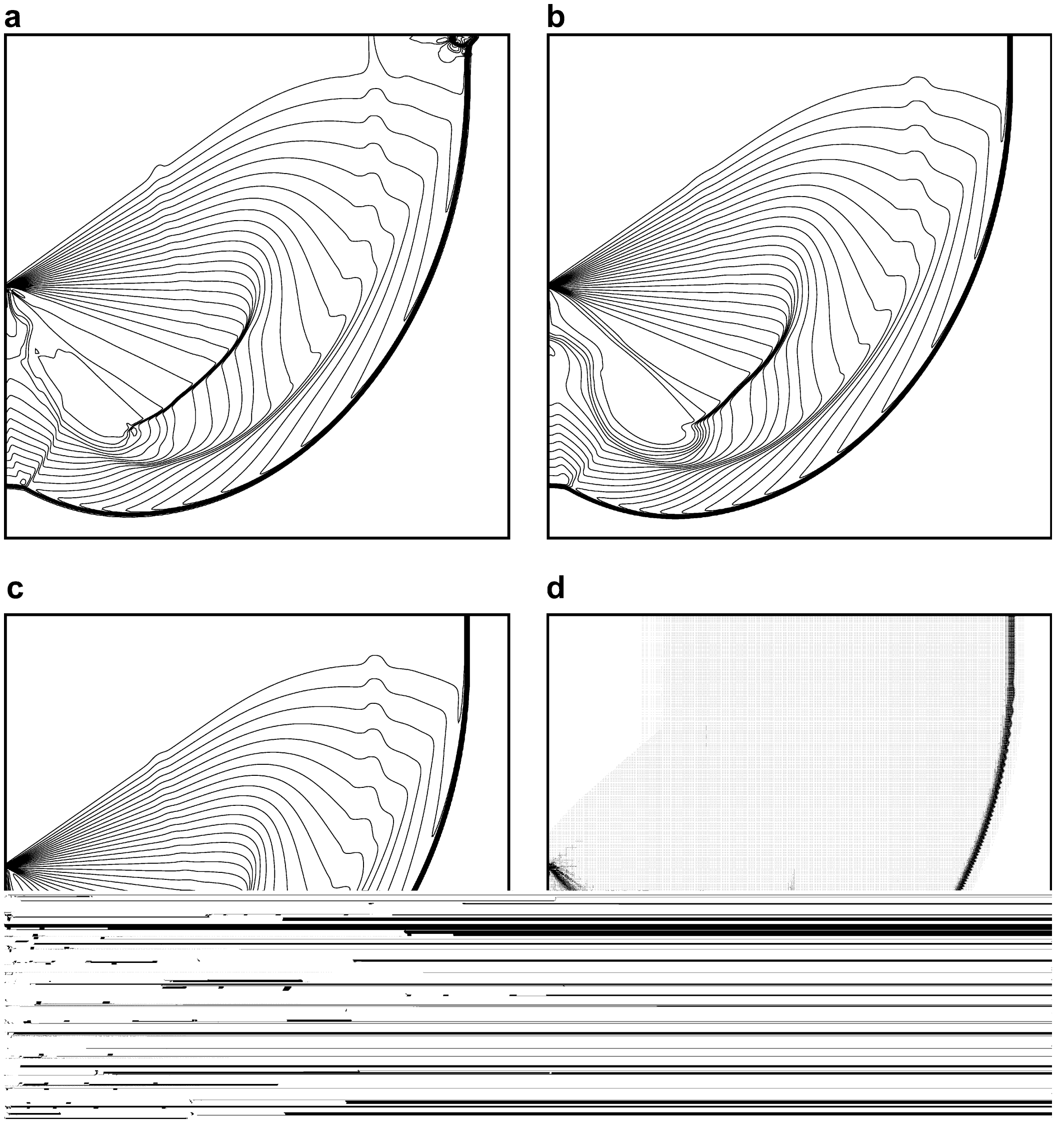

Fig. 13. Shock diffraction: density contours. (a) Roe with E-fix, (b) Rotated-RR, (c) Rotated-RHLL and (d) Vector Plot of $\alpha_1\mathbf{n}_1$ (Rotated-RHLL).

7.2. Shear layers

7.2.1. Contact discontinuity

In order to demonstrate the shear-resolving capability of the rotated-hybrid fluxes, we conducted a simple test of a grid-aligned contact discontinuity. In a square domain of 50×50 uniform grid $(0 \leqslant x \leqslant 0.05,\ 0 \leqslant y \leqslant 0.05)$, we specify $\rho = 10$ on the left half and $\rho = 1$ on the right half with other quantities uniform throughout the domain. We computed solutions by first-order schemes for 100 times steps (CFL $= 0.65$), with the Roe, HLL and Rotated-RHLL fluxes. Solutions are fully converged (density residuals down to 10^{-16}) with the Roe and Rotated-RHLL fluxes, but not with the HLL. The HLL continued to diffuse the solution after 100 time steps, and eventually smeared it out completely. A section plot of density is shown in Fig. 15. As expected, the HLL flux diffuses the contact while the Roe flux preserves it. The Rotated-RHLL flux, as designed, reduced to the Roe flux for faces aligned with the contact wave, and produced a perfectly comparable solution to that of the Roe flux. Also, we note that also for faces perpendicular to the contact

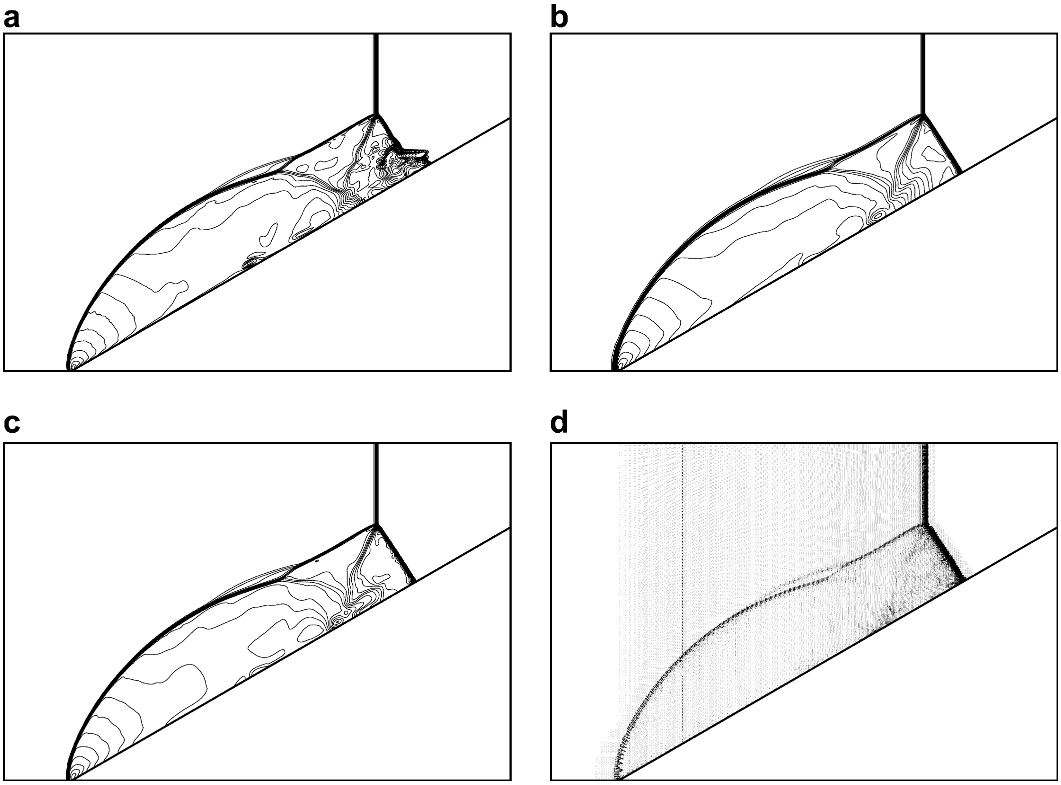

Fig. 14. Double Mach reflection: density contours. (a) Roe with E-fix, (b) Rotated-RR, (c) Rotated-RHLL and (d) vector plot of $\alpha_1 \mathbf{n}_1$ (Rotated-RHLL).

wave, the Rotated-RHLL flux becomes the Roe flux because $\|\Delta\vec{q}\| < \epsilon$. The same results were obtained by the Rusanov and Rotated-RR fluxes, and therefore not shown.

7.2.2. Supersonic jet

To further demonstrate the shear-resolving ability, we compared the Rotated-RHLL flux with the Roe and HLL fluxes for a non-aligned shear wave. This is a one-dimensional Riemann problem in a two-dimensional domain with time-like x-axis. On the left boundary of a square domain, we give, on the top half,

$$\begin{bmatrix} \rho \\ u \\ v \\ p \end{bmatrix}_{\text{Top}} = \begin{bmatrix} 0.25 \\ 4.0 \\ 0 \\ 0.25 \end{bmatrix}, \tag{7.3}$$

and on the bottom half,

$$\begin{bmatrix} \rho \\ u \\ v \\ p \end{bmatrix}_{\text{Bottom}} = \begin{bmatrix} 1 \\ 2.4 \\ 0 \\ 0.5 \end{bmatrix}. \tag{7.4}$$

The top, the bottom, and the right boundaries were treated as free-boundary. Computations were made by first-order schemes on a 200 × 200 grid for 2000 time steps (CFL = 0.5). The Roe flux fully converged (the density residual down to 10^{-16}), and the HLL and Rotated-RHLL fluxes also converged at a level of 10^{-12} (7–8 orders of magnitude reduction). Density contours are shown in Fig. 16. Clearly, the Rotated-RHLL flux

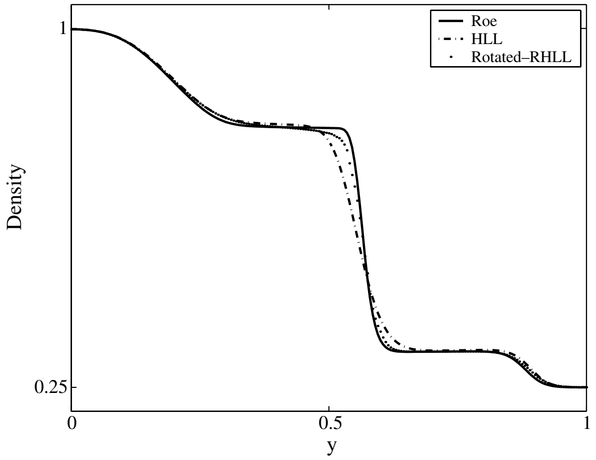

Fig. 17. Supersonic jet: section plot of density at $x = 1$.

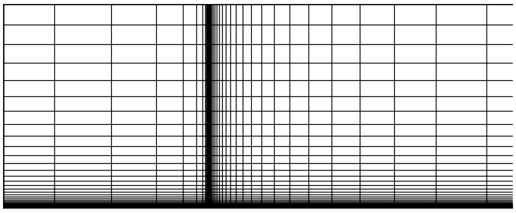

Fig. 18. Computational grid for flat-plate boundary calculations.

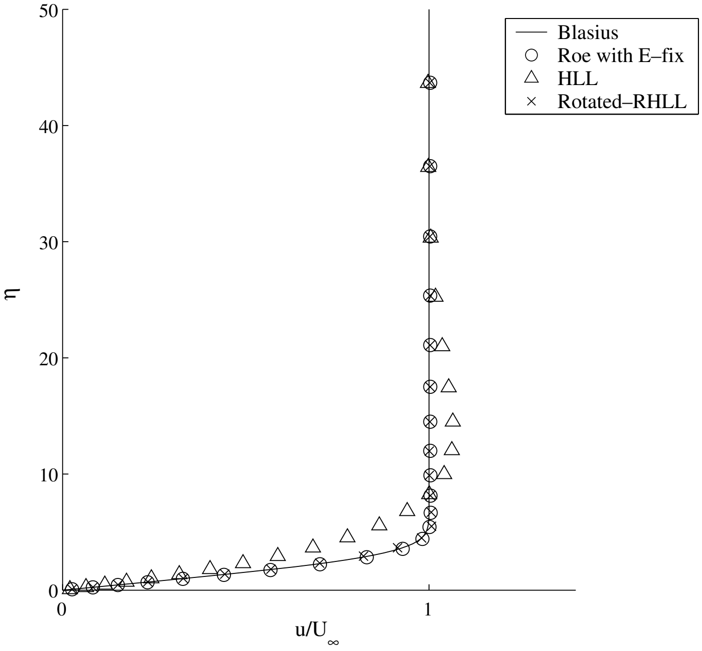

Fig. 19. Velocity profile for the flat-plate boundary layer problem: $\eta = \frac{y\sqrt{Re_x}}{x}$, $Re_x = \frac{\rho U_\infty x}{\mu}$ where ρ and μ are the local density and viscosity coefficient, and U_∞ is the free-stream velocity magnitude.

the Rotated-RHLL flux gives an almost identical result with that of the Roe flux whereas the HLL flux gives very dissipative and inaccurate solution. Very similar results were obtained by the Rusanov and Rotated-RR fluxes, and therefore not shown. We also point out that the solution by the Rotated-RHLL flux at 50,000 time steps is in fact nearly converged, being almost indistinguishable from the one at 250,000 time steps in Fig. 19.

8. Concluding remarks

We have developed very simple rotated-hybrid Riemann solvers, by combining the Roe solver and its simplified versions, i.e., the Rusanov and HLL solvers. The resulting flux functions are not only robust for nonlinear shock instability, but also accurate for resolving shear layers. Furthermore, they are particularly simple and economical: the same form as the Roe flux with modified wave speeds, so that implementing them into an existing code (with a Roe-type flux already built in) is an extremely simple task. A cost comparison indicates that the new flux functions require only 7–14% more CPU time than the Roe flux. They are also directly applicable to both structured and unstructured grids. Although the Rotated-RR flux was found to create a carbuncle on an unstructured grid, the Rotated-RHLL flux did not suffer at all and produced perfectly acceptable solutions for all cases. Moreover, it was shown also that the vector quantity $\alpha_1\mathbf{n}_1$ was non-zero only at discontinuities and strongly varying flow fields, and therefore it could serve as a directional feature detector for anisotropic mesh adaptation. Efficiency, robustness and accuracy demonstrated by an extensive series of numerical experiments indicate that these simple rotated-hybrid Riemann solvers offer a quick and effective cure to those finite-volume Euler/Navier–Stokes codes that suffer from the carbuncle phenomenon.

Finally, we remark that although this paper focused on particular combinations of flux functions (the Roe, and the HLL or the Rusanov), other combinations are also possible. For example, we may employ the three-wave HLL approximate Riemann solver [13] or the HLLC solver [31] instead of the Roe solver, and we still obtain very simple flux functions. Yet, if we are willing to pay the cost of computing two flux functions per interface, we can combine any two flux functions in the form of (6.5) to devise a new rotated-hybrid flux that might work even better. Exploring such other combinations is left as a future work.

Acknowledgements

The second author thanks Professor Nakamura and his colleagues at Nagoya University for their support for computational resources. He is grateful to the Student Exchange program between the University of Michigan and Nagoya University. The authors are grateful to Dr. Farzad Ismail at the University of Iowa, Mr. Yoshifumi Suzuki at the University of Michigan, and Dr. Hiroshi Koyama at the University of Maryland for their valuable comments, and also to Dr. Jeffery A. White at NASA Langley Research Center for his comments particularly on the normalization of $\mathbf{n}_2$ in three-dimensions. Finally, the authors are grateful to the reviewers for their constructive comments.

References

[1] K.M. Peery, S.T. Imlay, Blunt-body flow simulations, AIAA Paper 88-2904, 1988.
[2] J.J. Quirk, A contribution to the great Riemann solver debate, International Journal for Numerical Methods in Fluids 18 (1994) 555–574.
[3] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, Journal of Computational Physics 43 (1981) 357–372.
[4] M. Pandolfi, D. D'Ambrosio, Numerical instabilities in upwind methods: analysis and cures for the "carbuncle" phenomenon, Journal of Computational Physics 166 (2000) 271–301.
[5] Y. Chauvat, J.-M. Moschetta, J. Gressier, Shock wave numerical structure and the carbuncle phenomenon, International Journal for Numerical Methods in Fluids 47 (2005) 903–909.
[6] M.S. Liou, Mass flux schemes and connection to shock instability, Journal of Computational Physics 160 (2000) 623–648.
[7] K. Xu, Z. Li, Dissipative mechanism in Godunov-type schemes, International Journal for Numerical Methods in Fluids 37 (2001) 1–22.
[8] S.-S. Kim, C. Kim, O.-H. Rho, S.K. Hong, Cures for the shock instability: development of a shock-stable Roe scheme, Journal of Computational Physics 185 (2003) 342–374.

[9] M. Dumbser, J.-M. Morschetta, J. Gressier, A matrix stability analysis of the carbuncle phenomenon, Journal of Computational Physics 197 (2004) 647–670.

[10] F. Ismail, Toward a Reliable Prediction of Shocks in Hypersonic Flows: Resolving Carbuncles with Entropy and Vorticity Control, Ph.D. Thesis, University of Michigan, Ann Arbor, MI, 2006.

[11] K. Kitamura, P.L. Roe, F. Ismail, An evaluation of Euler fluxes for hypersonic flow computations, AIAA Paper 2007-4465, 2007.

[12] V.V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, Journal of Computational and Mathematical Physics USSR 1 (1961) 267–279.

[13] A. Harten, P.D. Lax, B. van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Reviews 25 (1) (1983) 35–61.

[14] S.F. Davis, A rotationally biased upwind difference scheme for the Euler equations, Journal of Computational Physics 56 (1984) 65–92.

[15] D.W. Levy, K.G. Powell, B. van Leer, Use of a rotated Riemann solver for the two-dimensional Euler equations, Journal of Computational Physics 106 (1993) 201–214.

[16] Y.-X. Ren, A robust shock-capturing scheme based on rotated Riemann solvers, Computers and Fluids 32 (2003) 1379–1403.

[17] P. Janhunen, A positive conservative method for magnetohydrodynamics based on HLL and Roe methods, Journal of Computational Physics 166 (2000) 649–661.

[18] B. Einfeldt, C.D. Munz, P.L. Roe, B. Sjögreen, On Godunov-type methods near low densities, Journal of Computational Physics 92 (1991) 273–295.

[19] C.-W. Shu, S.J. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of Computational Physics 77 (1988) 439–471.

[20] B. van Leer, Upwind-difference methods for aerodynamic problems governed by the Euler equations, in: Fluid Mechanics, Lectures in Applied Mathematics, vol. 22, 1985, pp. 327–335.

[21] G.D. van Albada, B. van Leer, J.W.W. Roberts, A comparative study of computational methods in cosmic gas dynamics, Astronomy and Astrophysics 108 (1) (1982) 76–84.

[22] A. Harten, High resolution schemes for hyperbolic conservation laws, Journal of Computational Physics 49 (1983) 357–393.

[23] B. Einfeldt, On Godunov-type methods for gas dynamics, SIAM Journal on Numerical Analysis 25 (2) (1988) 294–318.

[24] P. Roe, H. Nishikawa, F. Ismail, L. Scalabrin, On carbuncles and other excrescences, in: 17th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2005-4872, Toronto, 2005.

[25] R. Sanders, E. Morano, M.-C. Druguet, Multidimensional dissipation for upwind schemes: stability and applications to gas dynamics, Journal of Computational Physics 145 (1998) 511–537.

[26] M.S. Liou, A sequel to AUSM: AUSM$^+$, Journal of Computational Physics 129 (1996) 364–382.

[27] M.S. Liou, A sequel to AUSM, Part II: AUSM$^+$-up for all speeds, Journal of Computational Physics 214 (2006) 137–170.

[28] K.H. Kim, C. Kim, O.-H. Rho, Methods for the accurate computations of hypersonic flows: AUSMPW+ scheme, Journal of Computational Physics 174 (2001) 38–80.

[29] F. Ismail, P.L. Roe, H. Nishikawa, A proposed cure to the carbuncle phenomenon, in: Fourth International Conference on Computational Fluid Dynamics, Gent, Belgium, 2006.

[30] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, Journal of Computational Physics 54 (1984) 115–173.

[31] E.F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the HLL-Riemann solver, Shock Waves 4 (1994) 25–34.