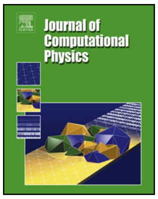
Journal of Computational Physics 396 (2019) 264–279

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



Canny-Edge-Detection/Rankine-Hugoniot-conditions unified shock sensor for inviscid and viscous flows

Takeshi R. Fujimoto*, Taro Kawasaki[1], Keiichi Kitamura

Yokohama National University, 79-5 Tokiwadai, Hodogaya-ku, Yokohama, Kanagawa 240-8501, Japan

ARTICLE INFO

Article history:
Received 8 November 2018
Received in revised form 27 June 2019
Accepted 28 June 2019
Available online 4 July 2019

Keywords:
CFD
Shock sensor
Image processing
Compressible flow
Aerodynamics
Canny-Edge-Detection

ABSTRACT

There are growing attentions to providing a simple and accurate shock detection method for computational fluid dynamics, not only for identifying the shocks in the post-processing but also for developing low-dissipation numerical schemes. However, most of conventional methods are designed for only either of them, as represented by Kanamori-Suzuki sensor [22] and Ducros sensor [1999]. The former method is quite accurate in inviscid flows but rather expensive to be incorporated into a numerical scheme, whereas the latter is very efficient and widely used while calling for an expert's care in determining actual shocks. In order to achieve both efficiency and theoretical accuracy, we developed a novel shock detection method for two-dimensional viscous/inviscid flows on Cartesian grids based on Canny-Edge-Detection [29], which is a well-known image processing method. We successfully applied the Canny-Edge-Detection to computational fluid dynamics (CFD) solutions by replacing the brightness value of digital images with pressure in CFD solutions. Then, by taking advantages of the simplicity of Canny-Edge-Detection along with Rankine-Hugoniot conditions across shocks (where 50% deviation is intentionally allowed only within the numerically captured shock), our method is designed to be both efficient and theoretically accurate, in contrast with the conventional schemes. The detection results of our method are examined in selected three test cases, in comparison with the Kanamori-Suzuki sensor and the Ducros sensor. Through the tests, we confirmed that our sensor is as accurate as Kanamori-Suzuki method, while as cheap as the Ducros sensor. In addition, the present sensor successfully detects shocks even in a viscous flow, and smoothly represents shocks oblique to the grid lines. Therefore, our sensor is eventually expected to contribute not only to the post-processing in CFD but also to developing schemes for computing high-speed flows with low cost and high accuracy.

© 2019 Elsevier Inc. All rights reserved.

1. Introduction

A shock wave is a crucial feature of compressible flows. From a physical point of view, the shock wave is regarded as a thin discontinuous layer in which Mach waves have been assembled, associated with abrupt changes in pressure, density, temperature, and other physical variables. As a result, the shock wave can be both harmful and beneficial in practical applications. For example, once the shock impinges on a body surface of aircraft/spacecraft, a boundary layer separation,

* Corresponding author.
E-mail address: fujimoto-takeshi-fc@ynu.jp (T.R. Fujimoto).

[1] Currently at University of Tokyo.

https://doi.org/10.1016/j.jcp.2019.06.071
0021-9991/© 2019 Elsevier Inc. All rights reserved.

Table 1
Shock sensors.

	Accuracy	Efficiency	Simplicity (free from threshold)	Viscous flows
Ducros et al. [6]	Fair	Very high	No	Yes
Kanamori and Suzuki [22]	Very high	Low	Yes	No
Pagendarm and Scitz [51]	Fair	Fair	No	Yes
Lovely and Haimes [52]	Fair	High	Yes	Yes
Present	High	High	Yes	Yes

huge drag, and extreme aerodynamic heating can be induced, that can be disastrous to the vehicle structure [1]. In addition, the shock also generates a loud and startling sound called 'sonic boom' from cruising aircraft [2], which has been a primary obstacle of the practical use of a supersonic transport, in spite of the technological possibility.

On the other hand, the shock has been actually applied to the medical field: Extracorporeal Shock Wave Lithotripsy (ESWL), for example, is one of the most common treatments for kidney stones in the United States of America [3], in which the kidney stone is removed by the shock waves from outside the patient's body. The shocks are also observed in detonation combustion (in which the shock travels through a duct) [4], which can be applied to a rocket engine. In astrophysics, a supernova explosion is associated with a shock wave, which is its key structure for the physical understanding of the stellar evolution [5]. For all the above cases, it is of great importance to identify the exact location of the shock wave.

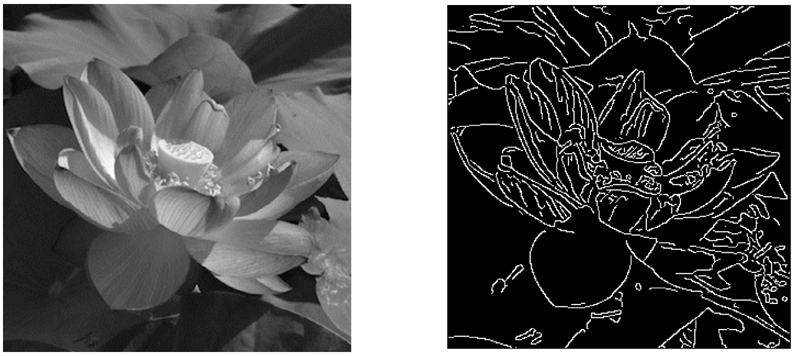
Traditionally, in post-processing of computational fluid dynamics (CFD), the shocks have been detected (by human judgment) as plotted, dense contour lines or iso-surfaces of physical values (e.g., pressure). With this manner, however, even experts can hardly identify shocks precisely because the boundaries between the shock waves and the other phenomena (i.e. endpoints of shocks) cannot be determined with confidence. In recent years, on the other hand, shock detectors (or sensors) have become powerful tools. In particular, the Ducros sensor [6] in which velocity divergence and vorticity are compared is popular nowadays because of its simplicity for implementation to distinguish the shocks form the turbulence [7]. Nevertheless, it is sensitive to velocity divergence even when vorticity is negligible [8], occasionally leading to wrong detected shocks (as will be demonstrated later). In addition, such a sensor usually calls for a user-specified threshold to determine shocks, whose values are scattered among users [9–21]. Thus, beginners and non-specialists will likely to encounter difficulties in determining the appropriate value. Kanamori and Suzuki proposed a shock detector based on the method of characteristics [22]. Although this method can yield accurate results on the rigorous grounds without tunable parameters, it needs high computational cost. Moreover, this sensor can encounter difficulties at shocks inside the cells where the numerically obtained values deviate from the theoretical value [23]. Also, the applicability of this scheme is limited to inviscid flows. There are other shock detection methods on post-processing, as summarized in a review article by Wu Ziniu [24] (and references therein), but no method seems to be free from the threshold values for accurate and efficient shock detections for viscous flows (see Table 1).

Shock sensors for CFD are not only useful for identifying the shocks in the post-process as mentioned above, but also important for developing low-dissipation numerical schemes for compressible flows where shocks are present [9–21]. In general, numerical methods with lower dissipation are preferred for simulating flows (specifically, turbulent flows) accurately. However, it is well known that computing shocks with insufficient dissipation will destabilize the solution, and it is also known that low-dissipation schemes often give rise to anomalous solutions, represented by "Carbuncle Phenomenon" [25,26]. We can handle these problems by, i) adding numerical dissipation locally around shocks[2] if their locations are clearly determined [27], or ii) increasing grid resolution near shocks, such as adaptive-mesh-refinement (AMR) [28]. For both the means, a technique for locating shocks is essential.

Now let us turn our attention to the information engineering community, where an image processing method called 'Canny-Edge-Detection' is well-known [29]. This is one of the (economical) first-order edge detection methods, in which the "edge" stands for discontinuous changes of brightness in digital images. We expect that *the Canny's method can be applied to CFD solutions easily by replacing the brightness value with pressure, since the image pixel shares the structure of Cartesian grids in CFD*. In derivation of the present shock detection method, a particular attention will be paid to the numerical errors inside the captured shockwave. The resultant method is potentially as cheap as the Ducros sensor, theoretically as accurate as the theory of characteristics (for inviscid flows), and successfully defines shocks without setting threshold values even for viscous flows (as will be demonstrated later).

The present work focuses on the Cartesian grids, that have been gaining growing attentions. In fact, there are many Cartesian grid generators and Cartesian-based CFD solvers available [30–45], presumably because i) AMR can be relatively easily implemented [4,32,42], ii) higher-order methods such as WENO [46] or Monotonicity-Preserving scheme [47] can be (almost) directly employed with slightest loss of accuracy [35,45], and iii) a very useful Immersed-Boundary-Method (IBM) [48] can be combined for flows with body-walls [38,39] (that are usually absent in astrophysical flow simulations, though), all for highly resolved flow solutions on Cartesian grids.

[2] The low-dissipation schemes typically take the form of "(numerical flux) = (central difference) − (sensor)×(dissipation term)", in which the "(sensor)" is near unity at shocks and zero otherwise [9–21].



(a) Original image (b) Edges found using Canny-Edge-Detection

Fig. 1. Canny-Edge-Detection applied to a grayscale photograph.

The objective of this work is to develop a new method of shock wave detection designed to be both efficient and theoretically accurate, in contrast with the conventional schemes by combining Canny-Edge-Detection and Rankine-Hugoniot conditions for two-dimensional viscous/inviscid flows on Cartesian grids. Our paper is organized as follows. Section 2 will briefly review the original Canny-Edge-Detection for digital images, and then combine it with the Rankine-Hugoniot conditions across the shock wave for the shock detection within compressible flow solutions on Cartesian grids. In Sec. 3, numerical examples will demonstrate accuracy, efficiency, and applicability to viscous flows of our method. Through the numerical experiments, the present method is expected to make a new option as a cheap, accurate, and easy-to-use (i.e., free from manually-operated thresholds) shock detector, and will eventually contribute to computing high-speed flows both in physics (i.e., thorough physical explanations of shockwave phenomena [49,50]) and in industries (i.e., development of quiet aircrafts [2]). Finally, Sec. 4 will summarize the current work.

2. Numerical method

In this Section, we will describe our shock wave detection method. 2.1 will describe a detailed algorithm of the original Canny-Edge-Detection. Then, 2.2-2.4 will explain the proposed method in a step-by-step manner: 2.2 will explain a step for selecting “shock candidates” by modified Canny-Edge-Detection for the shock sensor; 2.3 will describe a method for selecting shock cells from the candidates by Rankine-Hugoniot conditions; and 2.4 will summarize our proposed method.

2.1. Canny-Edge-Detection

Canny-Edge-Detector [29] is an *edge detection operator* to detect a wide range of edges in images, developed by John F. Canny in 1986. The edge detection is an image processing technique for detecting discontinuities in brightness, that is, extracting large gradient magnitudes of brightness (see Fig. 1 for example).

The process of Canny-Edge-Detection algorithm can be divided into the following four steps:

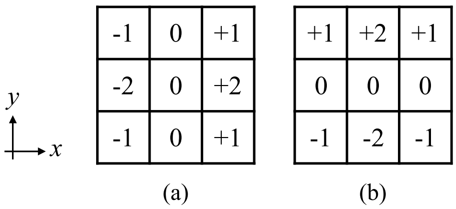
1. Smoothing images (in order to remove noises by Gaussian filter),
2. Computing gradient magnitudes and edge directions in an image,
3. Non-maximum suppression (thinning edges),
4. A final determination by a double threshold.

1) Smoothing images: The first step is filtering out any noise on the original image by using the Gaussian smoothing filter: These image noises are inevitably produced during picture taking. Since these noises can strongly affect the edge detection results, it is essential to eliminate them by smoothing the image. The equation for a two-dimensional Gaussian filter is,

$$G = \frac{1}{2\pi\sigma^2} \cdot \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{1}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.

2) Computing gradient magnitudes and edge directions in an image: The next step is to find the edge strength by computing the gradient magnitudes of brightness within the image. There are various algorithms to obtain the gradients. In this research, we use the Sobel operator [53], which is a widely used filter to compute gradients, as detailed in the following. The Sobel operator uses a pair of 3 × 3 convolution matrices/masks (Fig. 2), one for estimating the horizontal



-1	0	+1
-2	0	+2
-1	0	+1

(a)

+1	+2	+1
0	0	0
-1	-2	-1

(b)

Fig. 2. Sobel masks of 3 × 3 dimensions: (a) horizontal, (b) vertical.

gradient (Fig. 2(a)) and the other for the vertical one (Fig. 2(b)). For example, the horizontal gradient mask (Fig. 2(a)) was constructed by multiplying a horizontal averaging vector $\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}^T$ with a horizontal differential vector $\begin{bmatrix} -1 & 0 & +1 \end{bmatrix}$.

If we let a_{ij} be a brightness value on a cell (i, j) of the source image, and dx_{ij} and dy_{ij} be approximated horizontal and vertical gradients on the cell (i, j), respectively, they are computed by using the Sobel operator as follows:

$$\begin{aligned} dx_{ij} &\equiv \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \cdot \begin{bmatrix} a_{i-1j+1} & a_{ij+1} & a_{i+1j+1} \\ a_{i-1j} & a_{ij} & a_{i+1j} \\ a_{i-1j-1} & a_{ij-1} & a_{i+1j-1} \end{bmatrix}, \\ dy_{ij} &\equiv \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \cdot \begin{bmatrix} a_{i-1j+1} & a_{ij+1} & a_{i+1j+1} \\ a_{i-1j} & a_{ij} & a_{i+1j} \\ a_{i-1j-1} & a_{ij-1} & a_{i+1j-1} \end{bmatrix}, \end{aligned} \tag{2}$$

where $(\cdot)$ means inner product calculation. After calculating the approximated gradients, we can calculate a gradient magnitude d_{ij} and its direction θ_{ij} (hereinafter, this is called *gradient direction*) on the cell (i, j), by the following formula:

$$d_{ij} \equiv \sqrt{dx_{ij}^2 + dy_{ij}^2} \tag{3}$$

$$\theta_{ij} \equiv \tan^{-1}(dy_{ij}/dx_{ij}) \tag{4}$$

For instance, for

$$\begin{aligned} &\begin{bmatrix} a_{i-1j+1} & a_{ij+1} & a_{i+1j+1} \\ a_{i-1j} & a_{ij} & a_{i+1j} \\ a_{i-1j-1} & a_{ij-1} & a_{i+1j-1} \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 2 & 3 \\ 0 & -1 & 2 \end{bmatrix}, \\ dx_{ij} &\equiv \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 & 0 \\ 1 & 2 & 3 \\ 0 & -1 & 2 \end{bmatrix} = 0+0+0+2+0+2+0+0+2 = 6, \\ dy_{ij} &\equiv \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 & 0 \\ 1 & 2 & 3 \\ 0 & -1 & 2 \end{bmatrix} = 0+2+0+0+0+0+0-6-2 = -6 \end{aligned}$$

and

$$\begin{aligned} d_{ij} &\equiv \sqrt{dx_{ij}^2 + dy_{ij}^2} = 6\sqrt{2} \\ \theta_{ij} &\equiv \tan^{-1}(dy_{ij}/dx_{ij}) = \tan^{-1}(-1) = -\pi/4 \end{aligned}$$

that is, $d_{ij} = 6\sqrt{2}$ of the gradient magnitude exists with $\theta_{ij} = -\pi/4$ direction (and a potential edge, as will be explained below, lies in $+\pi/4$ and $-3\pi/4$ directions) at the cell (i, j).

3) Non-maximum suppression: Non-maximum suppression is an edge extracting technique by excluding pixels where gradient values are not maximal. Specifically, for each cell, it is checked whether or not the gradient magnitude d_{ij} of the cell (i, j) is maximal among pixels in the "gradient direction θ_{ij}" found in step 2. Then, the edge direction which is considered perpendicular to the gradient direction can be defined. We illustrated this procedure in Fig. 3. The algorithm executed for each pixel (i, j) to distinguish edge pixels from the others is:

3-1. Compare the edge magnitude value of the target pixel d_{ij} (red point in Fig. 3) with those of the pixels in the positive and negative gradient directions (blue points in Fig. 3).

3-2. If d_{ij} is the largest compared to the pixels with the same direction of the gradient θ_{ij} in the mask (we use a 3×3 mask in this paper), the target pixel is extracted as an edge "candidate." Otherwise, the pixel is excluded from the candidates.

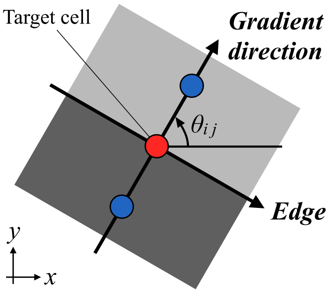


Fig. 3. Non-maximum suppression. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 2
Discretization of edge detection.

θ_{ij} range [rad]	Quantized θ_{ij} [rad]	Referenced cells	
$-\pi/8 \le \theta_{ij} < \pi/8$	0	$(i+1, j)$, $(i-1, j)$	(i)
$\pi/8 \le \theta_{ij} < 3\pi/8$	$\pi/4$	$(i+1, j+1)$, $(i-1, j-1)$	(ii)
$3\pi/8 \le \theta_{ij} < \pi/2$ or $-\pi/2 \le \theta_{ij} < -3\pi/8$	$\pi/2$	$(i, j+1)$, $(i, j-1)$	(iii)
$-3\pi/8 \le \theta_{ij} < -\pi/8$	$-\pi/4$	$(i+1, j-1)$, $(i-1, j+1)$	(iv)

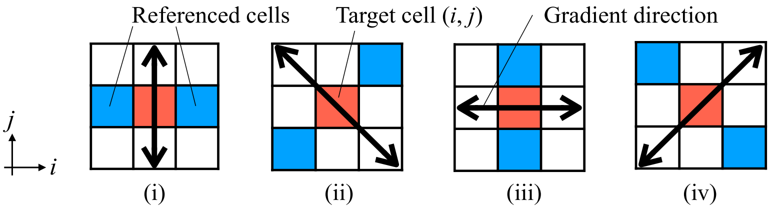


Fig. 4. Referenced cells in each case.

After conducting the above procedure, only extracted edge cells called “candidates” proceed to the final step (4. double threshold).

In the actual implementations, the algorithm discretizes the gradient directions θ_{ij} into a set of four discrete directions as shown in Table 2 and Fig. 4. In the case (i), for example, if the gradient magnitude d_{ij} of the target cell (i, j) (highlighted as red in Fig. 4) is higher than that of the referenced cells in the same direction of the gradient directions θ_{ij} (highlighted as blue in Fig. 4), the target cell is judged to be a candidate. Otherwise, the target cell is determined to be a non-edge.

4) A final determination by a double threshold: Finally, true edge cells are determined from the “candidate” pixels. While most candidates represent real edges in an image, the others were derived from merely noises and/or (continuous) color variations. In order to filter out these undesired edges, we introduce a pair of thresholds, one is higher (T_{pH}) and the other is lower (T_{pL}). Edge cells are finally determined under the following three conditions:

4-1. Determine as the edge when $d_{ij} > T_{pH}$ is true
4-2. Determine as the non-edge when $d_{ij} < T_{pL}$ is true
4-3. For the remaining cells, in which $T_{pL} \le d_{ij} \le T_{pH}$ is true, determine as the edge if more than one edge cells determined in Step 4-1 (where $d_{ij} > T_{pH}$ is true) exist in the eight surrounding cells of the current cell (i, j).

Having briefly reviewed the Canny-Edge-Detection method in the image processing, let us extend it to shock wave detection in computational fluid dynamics solutions in the subsequent subsection.

2.2. (Preliminary, straightforward) application of Canny-Edge-Detection to shock waves

Here, we propose a novel shock detection method based on the Canny edge method. Our core concept is very simple: We will apply the Canny-Edge-Detection to CFD solutions simply by *replacing the brightness value with pressure value* in the computed flow on a computational grid (composed of cells). This concept can be easily realized for two reasons: i) the image pixels share the common structure of cells on the Cartesian grids in CFD, ii) the edges in the images are similar to the shock in CFD results in a sense that each of them can be described as (a set of) places at which the gradient magnitude of brightness or pressure shows its local maximum value. The reason we use pressure rather than density, velocity or other physical values is that it is insensitive to other discontinuities such as slip line, shear layer or contact discontinuity.

The following is a procedure of our CFD version of the Canny-Edge-Detection applied to shocks.

1) Step 1: Smoothing [*NOT included*]

When we apply the Canny method to CFD, this smoothing process in the original method is intentionally omitted because smoothing can affect or even destroy the important, original, physical information; for example, it can smear out the original pressure jumps (shocks) in the numerical solution. In fact, unlike the photography, the CFD solutions are free from huge, camera-shaking noises.

1) Step 2: Computing gradient magnitudes and direction of edges

Instead of calculating edges of brightness in the digital image, we compute gradient magnitudes and the gradient directions of *"pressure"* to find edges within CFD results. If we let p_{ij} be a pressure value on a cell (i, j), and dpx_{ij} and dpy_{ij} be the horizontal and vertical approximated gradients of pressure on the cell (i, j), respectively, we can calculate dpx_{ij} and dpy_{ij} by replacing a_{ij} with p_{ij} at the right-hand side of Eq. (2) as follows:

$$\begin{aligned} dpx_{ij} &\equiv \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \cdot \begin{bmatrix} p_{i-1j+1} & p_{ij+1} & p_{i+1j+1} \\ p_{i-1j} & p_{ij} & p_{i+1j} \\ p_{i-1j-1} & p_{ij-1} & p_{i+1j-1} \end{bmatrix}, \\ dpy_{ij} &\equiv \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \cdot \begin{bmatrix} p_{i-1j+1} & p_{ij+1} & p_{i+1j+1} \\ p_{i-1j} & p_{ij} & p_{i+1j} \\ p_{i-1j-1} & p_{ij-1} & p_{i+1j-1} \end{bmatrix}, \end{aligned} \tag{5}$$

where $(\cdot)$ means the inner product calculation, again. As with the step 2 of Canny method,

$$dp_{ij} \equiv \sqrt{dpx_{ij}^2 + dpy_{ij}^2} \tag{6}$$

$$\theta p_{ij} \equiv \tan^{-1}(dpy_{ij}/dpx_{ij}) \tag{7}$$

where dp_{ij} and θp_{ij} denote the gradient magnitude and the direction of pressure on a cell (i, j), respectively.

3) Step 3: Non-maximum suppression (thinning edges)

As mentioned in 2.1, the edge pixels are extracted as the "candidates" by checking whether the gradient magnitude d_{ij} on the target pixels are maximal among those of pixels in the "gradient direction." Instead of using d_{ij} and θ_{ij} (both are concerned with brightness), we employ gradient magnitude of *pressure* dp_{ij} and its gradient direction θp_{ij} to extract cells with local maximal value as "shock candidates."

In the actual implementations, we adopt the discretized gradient directions θp_{ij} as well as the original Canny-edge-method (in Table 2, "θ_{ij}" shall be replaced with "θp_{ij}"). Hereafter, we consider using detection methods only for the candidates.

4) Step 4: The final determination by a double threshold [*NOT directly included, but to be modified*]

When applying this step to the shock detection, we encounter some difficulties:

a) The indicator (dp_{ij}) and threshold values are difficult to be evaluated based on the flow physics.
b) Threshold values (parameters) are problem dependent. In other words, each problem should have its own optimal parameters.

Regarding a), let us consider when we try to determine appropriate values of T_{pH} and T_{pL} to detect a certain shock wave. Recall that those thresholds are compared with the indicator (dp_{ij}), for evaluating whether a shock candidate is an actual shock or not. According to Eqs. (5), (6), (7), dp_{ij} is the absolute value of a gradient vector of pressure, in which pressure is differentiated along the direction of pressure θp_{ij}. However, in theory, a shock wave is a discontinuous layer, over which the derivative of pressure cannot be defined. That is why T_{pH} and T_{pL} are difficult to be evaluated in terms of flow physics.

As for b), each shock should have its own values of dp_{ij}, and we must set proper T_{pH} and T_{pL} for each problem. In other words, thresholds are problem dependent, and shocks outside the range between these (preset) values are not detected.

In the following subsection, therefore, we offer an alternative method for Step 4 to detect shocks automatically without user-defined parameters for (unknown) shocks.

2.3. Theoretical estimation based on Rankine-Hugoniot conditions (Step 4 of present method)

Instead of the final determination by a double threshold (Step 4 of Canny-Edge-Detection), we offer a new method based on the Rankine-Hugoniot conditions. In general, the following formula holds for a normal shock wave.

$$M_{2t} = \sqrt{\frac{2 + (\gamma - 1)M_1^2}{2\gamma M_1^2 - (\gamma - 1)}} \tag{8}$$

where γ is the specific heat ratio of the gas, M_1 is the Mach number upstream the shock, and M_{2t} represents the theoretical shock-downstream Mach number. The present, modified detection procedure (Step 4) is as follows:

4-1. Mach numbers M_1 and M_2 (ahead of and behind the shock, respectively) are extracted from computational results. Note that M_2, different from M_{2t}, is obtained from a CFD solution.
4-2. The theoretical post-shock Mach number (M_{2t}) is calculated by substituting M_1 into Eq. (8).
4-3. A relative error ε between M_2 and M_{2t} is calculated as,

$$\varepsilon \equiv \frac{|M_2 - M_{2t}|}{M_2} (\leq 0.5) \tag{9}$$

4-4. If ε is within 50% ($\varepsilon \leq 0.5$), the target cell is determined to be as a shock wave.

Please note that we set the maximum permissible value as 50% to accept errors due to the intermediate region of the numerical shock, based on the fact that for a Mach 10 normal shock, these errors are known to be up to 40% [23], and the fact that these errors can increase when the shock is oblique to the grid lines. It should be kept in mind that: (a) we used values on the referenced cells as shown in Table 2 and Fig. 4 for acquiring M_1 and M_2; (b) Eq. (8) holds only for the normal shock; (c) Eq. (9) can be valid only when the shock wave is stationary, that is, the shock wave does not move.

Regarding (b), when this relation is applied to general (oblique) shocks, the shock can be detected by considering its shock-normal component value. As for (c), when the shock wave is moving with supersonic speed, further care must be taken because both pre- and post-shock regions might be subsonic. Let us define a *shock-fixed* coordinate system in order to treat all shocks as locally "steady" shocks. This system can be introduced by acquiring instantaneous shock speed. In order to determine the moving velocity of the shock wave, we consider the total enthalpy h_t as in Ref. [22]. Immediate before and after the shock, total enthalpy should be conserved. In other words, by defining the velocity of the shock so that the total enthalpy is conserved, the *shock-fixed* coordinate system can be achieved as we set the shock speed to be the speed of the coordinate system. Then we estimate the velocity of coordinate systems as the velocity of the shock wave. With that velocity, the conservation of h_t between pre- and post-shock region is described as follows:

$$\begin{aligned} h_t &= \frac{\gamma}{\gamma - 1} \cdot \frac{p_1}{\rho_1} + \frac{1}{2}\left\{(u_1 - u_s)^2 - (v_1 - v_s)^2\right\} \\ &= \frac{\gamma}{\gamma - 1} \cdot \frac{p_2}{\rho_2} + \frac{1}{2}\left\{(u_2 - u_s)^2 - (v_2 - v_s)^2\right\} \end{aligned} \tag{10}$$

where u_s and v_s denote the x and y components, respectively, of the moving velocity, respectively, of the shock wave, or of the *shock-fixed* coordinate, in which subscripts 1 and 2 represent regions upstream and downstream of the shock, respectively, as shown in Fig. 5. When the angle between the velocity vector and the x-axis is designated as θp_{ij}, $u_s \tan\theta p_{ij} = v_s$ holds. Then Eq. (10) can be solved as follows:

$$\begin{cases} u_s = \dfrac{(2h_1 + u_1^2 + v_1^2) - (2h_2 + u_2^2 + v_2^2)}{2\{(u_1 - u_2) + \tan\theta p_{ij}(v_1 - v_2)\}} \\ v_s = \dfrac{(2h_1 + u_1^2 + v_1^2) - (2h_2 + u_2^2 + v_2^2)}{2\{(1/\tan\theta p_{ij})(u_1 - u_2) + (v_1 - v_2)\}} \end{cases} \tag{11}$$

where $h = p/\rho \cdot \gamma/(\gamma - 1)$. After acquiring the shock speed (u_s, v_s), we can estimate M_1 and M_2 on the *shock-fixed* coordinate as follows:

$$\begin{cases} M_1 = \sqrt{(u_1 - u_s)^2 - (v_1 - v_s)^2} \\ M_2 = \sqrt{(u_2 - u_s)^2 - (v_2 - v_s)^2} \end{cases} \tag{12}$$

Now the same method for steady shock can be used.

After the above procedures are executed for the candidate cells, there are still cells in which shocks are too "weak" to be distinguished from regular numerical jumps between cells. In this work, we exclude shocks with a small Mach ratio M_1/M_2 under 1.01 [1% change] by the following filter:

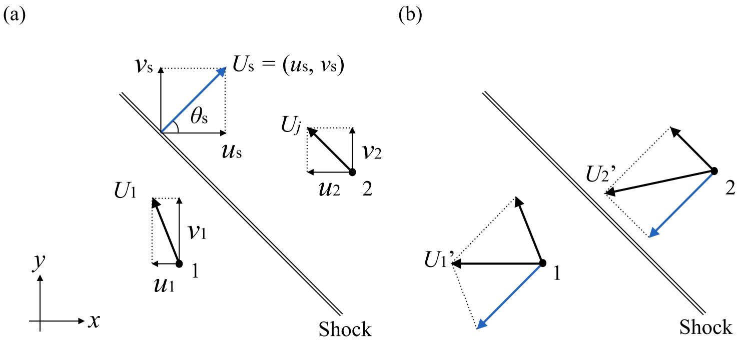


Fig. 5. Localized steady flow in a shock-fixed coordinate system: (a) original coordinate system, (b) shock-fixed coordinate system.

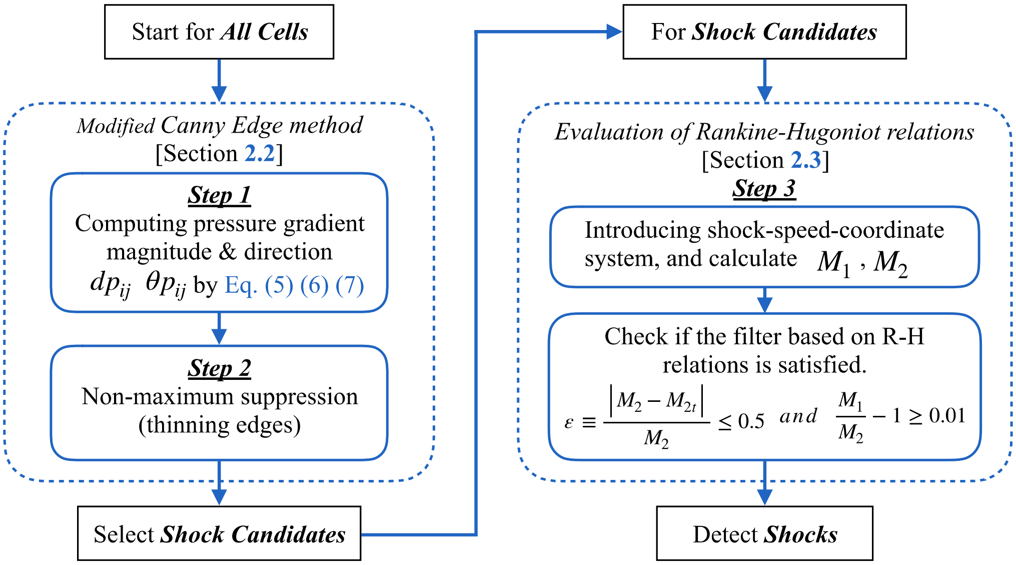

Start for All Cells
Modified Canny Edge method
[Section 2.2]
Step 1
Computing pressure gradient magnitude & direction
dp_{ij} θp_{ij} by Eq. (5) (6) (7)
Step 2
Non-maximum suppression (thinning edges)
Select Shock Candidates
For Shock Candidates
Evaluation of Rankine-Hugoniot relations
[Section 2.3]
Step 3
Introducing shock-speed-coordinate system, and calculate M_1 , M_2
Check if the filter based on R-H relations is satisfied.
$\varepsilon \equiv \frac{|M_2 - M_{2t}|}{M_2} \leq 0.5$ and $\frac{M_1}{M_2} - 1 \geq 0.01$
Detect Shocks

Fig. 6. The flowchart of this method.

$$\frac{M_1 - M_2}{M_2} = \frac{M_1}{M_2} - 1 \geq 0.01 \tag{13}$$

In summary, the whole procedure for this step is as follows:

1. Calculate the moving velocity of the shock wave, or of the shock-fixed coordinate (u_s, v_s) as Eq. (11)
2. Estimate M_1 and M_2 on the shock-fixed coordinate as Eq. (12)
3. The theoretical post-shock Mach number (M_{2t}) is calculated by substituting M_1 into Eq. (8)
4. Determine the candidates as shocks if Eqs. (9), (13) are both satisfied

2.4. Summary of this method

The following flowchart (Fig. 6) summarizes the descriptions of this section. Our shock detection method consists of three steps:

1. Compute gradient magnitudes and direction of edges by Eqs. (5), (6), (7) (Step 2 of Canny-Edge-Detection) [2.2 (2)].
2. Non-maximum suppression (Step 3 of Canny-Edge-Detection) [2.2 (3)].
3. Theoretical estimation based on Rankine-Hugoniot conditions as Eqs. (9), (13) [2.3].

In the Step 1 and Step 2, we adopted the modified Canny-Edge-Detection to select shock candidates at which a gradient magnitude of pressure shows a local maximum value. Then, in the Step 3, we adopted the Rankine-Hugoniot conditions to evaluate shock candidates based on the flow physics. In the next section, we will demonstrate the effectiveness of this method in three numerical cases in comparison with other shock detection methods.

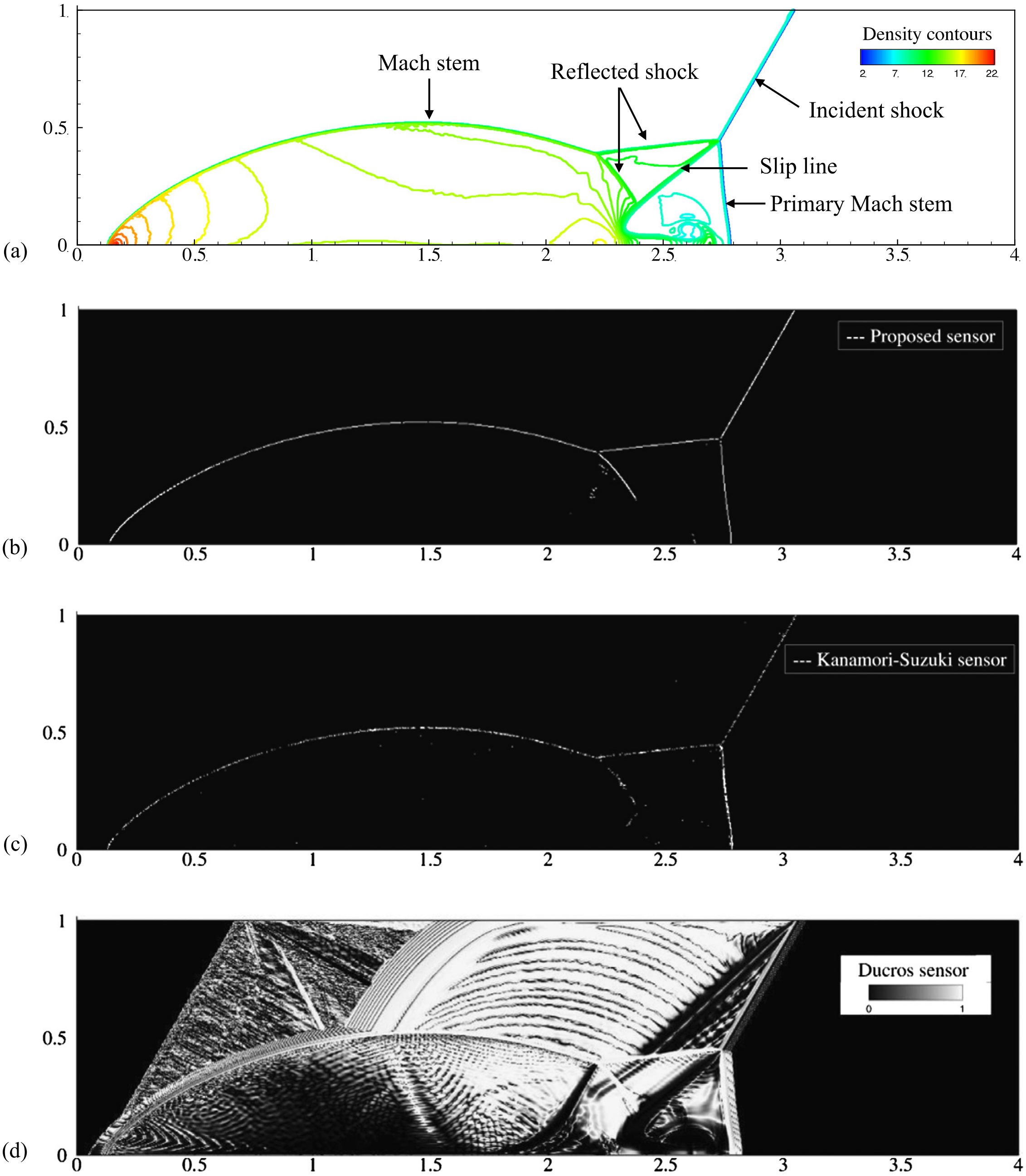


Fig. 7. Comparison of shock sensor values for Double Mach Reflection at $t = 0.2$, (a) density contours, (b) our proposed sensor, (c) Kanamori/Suzuki sensor and (d) Ducros sensor.

3. Numerical examples

In this chapter, our shock wave detection method is applied to selected numerical tests to demonstrate its efficacies, covering viscous/inviscid, supersonic/hypersonic flows: Double-Mach-Reflection (3.1); Shock/vortex-Interaction (3.2); Shock/Boundary-Layer Interaction (3.3). The above three calculations will be performed by solving the Euler or N-S equations with 2nd-order MUSCL interpolation [54] with the Van Albada's limiter [55], HR-SLAU2 numerical flux [56], and two-stage, 2nd-order Runge-Kutta time integration method [57], unless mentioned otherwise. In 3.1–3.3, the simplicity and the accuracy of our sensor will be investigated by comparisons with Kanamori-Suzuki sensor and Ducros sensor. In 3.4, we will compare responses of two different limiters (minmod [58] and Van Albada's limiter [55]) to our shock detector.

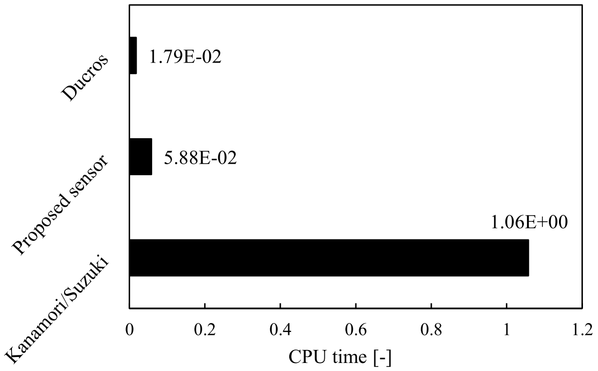


Fig. 8. Comparison of CPU times for three shock sensors.

3.1. Double Mach reflection (inviscid and hypersonic)

A planar shock wave in an inviscid fluid at Mach 10 reflected by a 30-degree ramp is computed here. The computational domain is $[0,4] \times [0,1]$ divided by 1200×300 Cartesian cells ($\Delta x = \Delta y = 1/300$). The initial condition of the problem is:

$$(\rho, u, v, p) = \begin{cases} (8, 8.25\cos(\frac{\pi}{6}), -8.25\sin(\frac{\pi}{6}), 116.5) & \text{for } x \le \frac{1}{6} + \frac{y}{\sqrt{3}}, \\ (1.4, 0, 0, 1) & \text{for } x > \frac{1}{6} + \frac{y}{\sqrt{3}}. \end{cases} \tag{14}$$

The lower boundary is treated as a slip wall, and the left and right boundary values are fixed to the left and right states of the shock, respectively. The condition on the upper boundary is changed every moment according to the shock propagation. The time step is prescribed at $\Delta \mathrm{t} = 2.\mathrm{e}-5$ (CFL ≈ 0.08), and the computation is conducted to 10000 steps ($t = 0.2$). Fig. 7(a) shows density contours of the solution at $t = 0.2$.

Fig. 7(b), (c), (d) show the shock detection results by, (b) our proposed sensor, (c) Kanamori-Suzuki sensor and (d) Ducros sensor, respectively. In each example, detected cells are plotted as white. As for (d), the Ducros sensor values are displayed from 0 to 1 (the user must specify the threshold value, such as 0.65, beyond which the cell is regarded to contain the shock), with a grayscale (where the white color represents a high value).

A qualitative comparison of the density contours and the result of our proposed sensor suggests that our sensor detects the shock structure quite well. It is also noteworthy that slip lines are successfully excluded, where density gradient magnitude (rather than pressure gradient magnitude) is large. Although a few false detected cells are observed in the neighborhood of the reflected shock, these error cells account for only 0.7% of the shock candidate cells and are scattered (not forming a shock “wave”). Thus, they can easily be distinguished from the actual shock waves.

Also, the position of the detected shock wave by our sensor almost agrees with the result by the Kanamori-Suzuki sensor. Thus, in this case, it is considered that our sensor realizes almost as the same high accuracy as Kanamori-Suzuki sensor. Furthermore, our detected points are continuously connected along the shock waves, while in the result of Kanamori/Suzuki detected cells are discontinuous. This difference arises from the errors by the intermediate region of the numerical shock. Our method allows errors up to 50% from the intermediate region of the numerical shock in Eq. (9), whereas the Kanamori method does not.

We see that Ducros sensor is active, not only around shock waves, but in many other places where vorticity is negligible. This is one of typical examples that the Ducros sensor is sensitive to the velocity divergence when the vorticity is negligible, leading to erroneous detection (see Appendix for the formulation of Ducros sensor, Eq. (A.1)).

Next, the CPU time was measured (and averaged for 1,000 runs) for each case and compared in order to evaluate the calculation cost of our sensor. The results in Fig. 8 show that our method is approximately 18 times faster than Kanamori-Suzuki sensor, and only three times slower than the Ducros sensor, i.e., its cost is within the same order of the Ducros sensor's. Therefore, our sensor has been demonstrated to be as accurate as Kanamori-Suzuki sensor, and nearly as efficient as Ducros sensor.

3.2. Shock/vortex interaction (inviscid and supersonic)

Jiang–Shu's (inviscid and unsteady) shock/vortex interaction [46] is computed here. The computational domain is $[0,2] \times [0,1]$ divided by 800×400 Cartesian cells. The inflow condition is imposed at the left boundary, the outflow condition at the right boundary (with pressure fixed), and the slip condition is applied to the top and bottom boundaries. The initial conditions are as follows. A Mach 1.1 shockwave is given at $x = 0.5$. Its left side is:

$$(\rho, u, v, p)_L = (1, \sqrt{\gamma}, 0, 1) \quad \text{for } x \le 0.5 \tag{15}$$

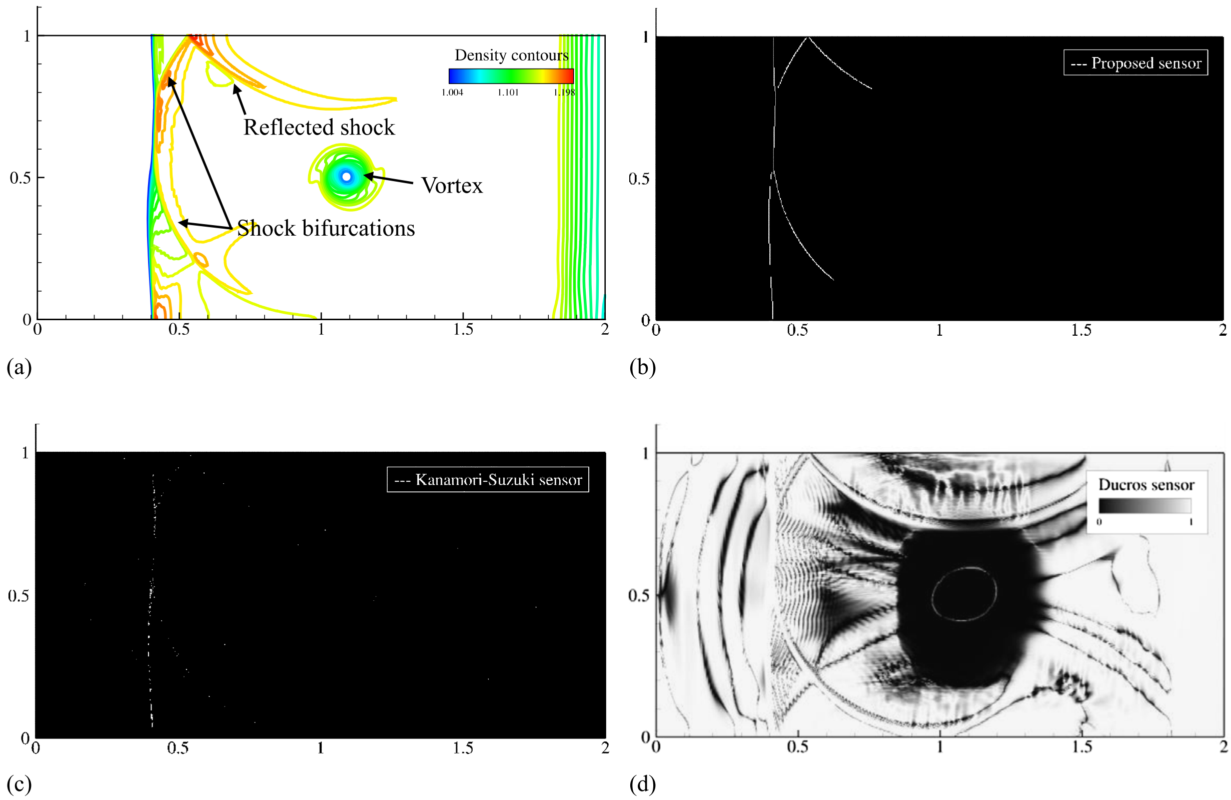


Fig. 9. Comparison shock sensor value fields for Shock/vortex Interaction at $t = 0.8$: (a) density contours, (b) our proposed sensor, (c) Kanamori-Suzuki sensor and (d) Ducros sensor.

The right-side condition is then provided to satisfy the Rankine–Hugoniot conditions. Furthermore, the following vortex centered at $(x_c, y_c) = (0.25, 0.5)$ is superimposed as perturbations to the initial flow.

$$\left.\begin{aligned}
\Delta u &= \varepsilon\tau\exp\left(\alpha\left(1-\tau^2\right)\right)\cdot\frac{\Delta y}{r}\\
\Delta v &= -\varepsilon\tau\exp\left(\alpha\left(1-\tau^2\right)\right)\cdot\frac{\Delta x}{r}\\
\Delta T &= -\frac{(\gamma-1)\varepsilon^2\exp(2\alpha(1-\tau^2))}{4\alpha\gamma}\\
\Delta p &= (\Delta T+1)^{\frac{\gamma}{\gamma-1}}-1\\
\Delta\rho &= (\Delta p+1)^{\frac{1}{\gamma}}-1
\end{aligned}\right\} \tag{16a}$$

where

$$\tau = r/r_c, \quad r=\sqrt{(\Delta x)^2+(\Delta y)^2}, \quad \Delta x = x - x_c, \quad \Delta y = y - y_c, \quad \varepsilon = 0.3, \quad r_c = 0.05, \quad \alpha = 0.204. \tag{16b}$$

The time step is prescribed at $\Delta t = 5.e-4$ (CFL $\approx$ 0.54), and the computation is conducted to 1600 steps ($t = 0.8$). Fig. 9(a) shows density contours at $t = 0.8$, in which calculation was performed by solving the Euler equations with the same methods as in the previous subsection. There exists *a bifurcation shock* arising from interactions between initial shock, vortex, and *reflected unsteady shock* from the upper wall. Fig. 9(b), (c), (d) show the shock detection results by, (b) our proposed sensor, (c) Kanamori-Suzuki sensor and (d) Ducros sensor, respectively.

Qualitatively, compared with Fig. 9(a), all the shocks look successfully detected in our method. Also, false detection was not seen at all in this case. When comparing Kanamori-Suzuki sensor with our sensor, the position of the shock detected by Kanamori-Suzuki sensor coincides well with our detection. However, in the Kanamori-Suzuki sensor, it is found that i) the detection spots are scattered greatly, and that ii) almost no reflected shock waves are captured. This is because the reflected shocks are oblique to the Cartesian grid lines and the errors on the numerical shocks are especially so large that Kanamori/Suzuki sensor cannot capture these shocks in this case. On the contrary, since our method allows 50% errors from the theoretical value inside the shock, we succeeded in detecting all the shock waves smoothly including reflected shocks

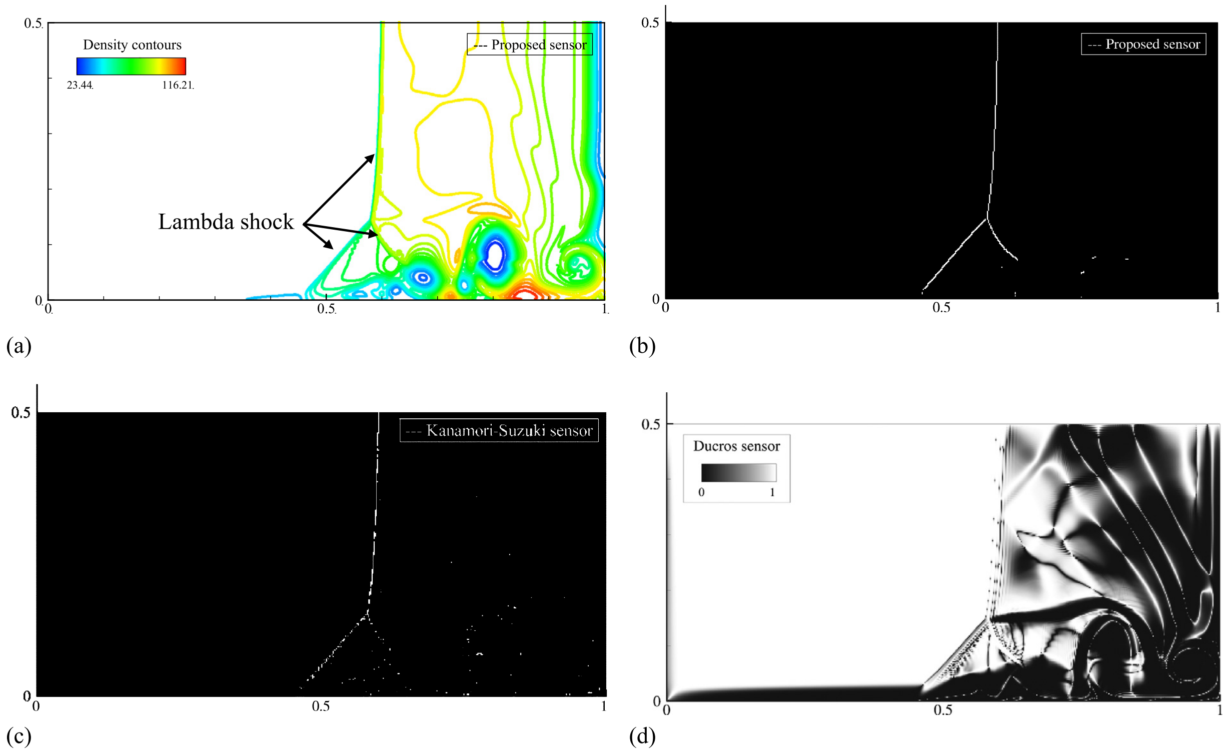


Fig. 10. Comparison of shock sensor value fields for Shock/Boundary-Layer Interaction at $t = 1$: (a) density contours, (b) our proposed sensor, (c) Kanamori/Suzuki sensor and (d) Ducros sensor.

(no false detection was found in this case). From these results, it can be said that our method is suitable in the case where oblique shocks with larger numerical errors present.

Looking at the results of the Ducros sensor, we can see that the sensor is (unexpectedly) activated in most areas where vorticity is low (see Appendix for its formulation, Eq. (A.1)), and any amount of disturbance arises due to the shocks, except for a region near the vortex. Therefore, it is confirmed that the Ducros sensor is too sensitive to detect shocks. On the contrary, our sensor does not suffer from such a difficulty.

3.3. Shock/boundary-layer interaction (viscous and supersonic)

Then we will deal with a 2D shock/boundary-layer interaction problem proposed by Daru et al. [59]. The computational space is $[0, 1] \times [0, 0.5]$ divided by 500×250 Cartesian cells ($\Delta x = \Delta y = 1/500$). The symmetry condition is applied to the upper boundary, while the adiabatic wall condition is imposed on the other boundaries. The initial conditions are:

$$(\rho, u, v, p) = \begin{cases} (120, 0, 0, 120/\gamma) & \text{for } x \leq -0.5, \\ (1.2, 0, 0, 1.2/\gamma) & \text{for } x > -0.5. \end{cases} \tag{17}$$

After the diaphragm rupture at $t = 0$, a shockwave traveling to the right reflects at the right boundary ($t = 0.2 - 0.3$) and comes back to left while interacting with a boundary-layer developed over the lower wall boundary, creating a bifurcation shock (or *lambda-shock*) followed by vortices. The AUSM+-up [60] is used for the inviscid term and the 2nd-order central difference for the viscous term. The time step is prescribed at $\Delta t = 1.25$e-6 (CFL ≈ 0.52), and the computation is conducted to 800,000 steps ($t = 1$). The Reynolds number is set to 200 as in [59,61]. Fig. 10(a) shows density contours of the result at $t = 1$. Fig. 10(b), (c), (d) show the detection results by, (b) our proposed sensor, (c) Kanamori-Suzuki sensor and (d) Ducros sensor, respectively.

Compared with the density distribution, our method accurately captures the position of the shock wave near the boundary layer. Although misdetections are found at several locations around the vortex in the boundary layer, these error cells are, again, easily be distinguished from actual shock waves.

When comparing Kanamori-Suzuki sensor with our sensor, the lower right part of the lambda shock was not detected by Kanamori-Suzuki sensor as opposed to the present sensor. This is understandable remembering the fact that Kanamori-Suzuki was based on the theory of characteristics in inviscid flows. The Ducros sensor (Fig. 10(d)), again, yields many potentially shocked (white) cells. Considering these, our sensor nearly successfully detected the shocks even in this viscous flow example.

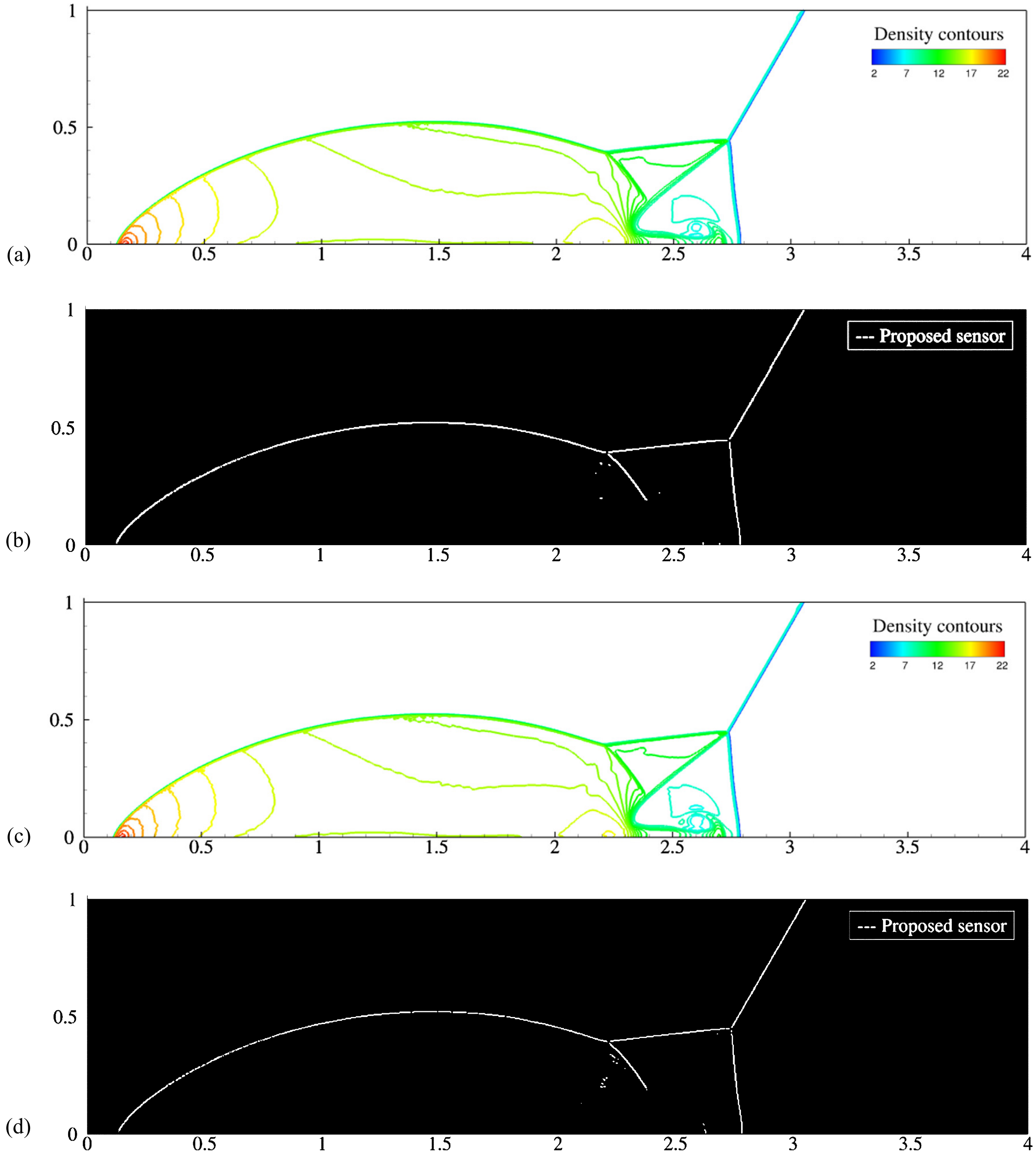


Fig. 11. Density contours and detection results of our sensor at $t = 0.2$, (a) density contours (minmod limiter), (b) our detection result (minmod limiter), (c) density contours (Van Albada's limiter), (d) our detection result (Van Albada's limiter).

3.4. Effects of slope limiter

Finally, we will discuss influences of the slope limiter. The same example (Double-Mach-Reflection) as in 3.1 is calculated by changing the limiters from minmod to Van Albada's, which is also widely-used. Fig. 11 shows density contours and detection results of our sensor at $t = 0.2$. We see that the shock geometry detected by our sensor is almost identical between both cases, except for a very small number of error cells. However, since these errors are less than 0.01% of all the cells, their influence can be almost ignored (the ratio of error cells to all the cells is 0.004% for minmod, and 0.01% for Van Albada).

Then, we will investigate the distribution of the relative error ε defined in Eq. (9) in the detected shocked cells. Fig. 12 shows a histogram of the relative error ε of the detected cells (i.e., cells with $\varepsilon \leq 0.5$). The horizontal axis is the dimensionless number of cells (= the number of cells of interest per total cell numbers). We can see that the distributions are similar

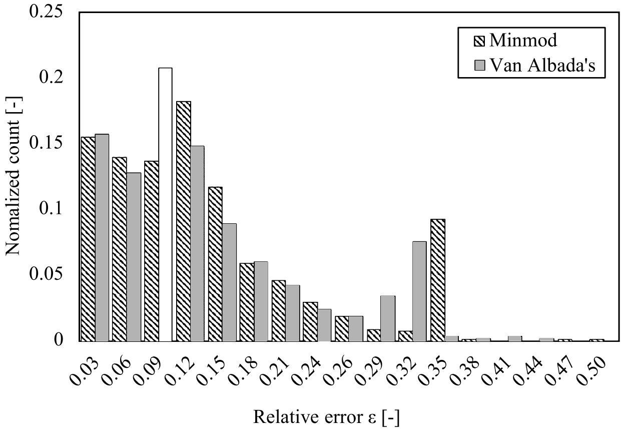


Fig. 12. Comparison of shocked cells having errors between limiters.

in that both results having a large peak around $\varepsilon = 0.1$ and small peaks near $\varepsilon = 0.35$. The cells distributed in the small peak near $\varepsilon = 0.35$ in Fig. 12 correspond to the primary Mach stem near $x = 2.7$ in Fig. 7(a), and the remaining shock waves are distributed in large peaks near $\varepsilon = 0.1$ in Fig. 12. It is noteworthy that there are only few cells with ε greater than 0.38 for both cases. This result shows that, for either of the limiters, it will not significantly affect the detection result if we change the threshold of ε in Eq. (9) between 0.4 and 0.5. From these results, our sensor has been demonstrated consistent between the two limiters and also within the range of $0.4 < \varepsilon < 0.5$.

4. Conclusions

In this study, we developed a new method of shock wave detection by combining Canny-Edge-Detection and Rankine-Hugoniot conditions for two-dimensional viscous/inviscid flows on Cartesian grids. Our method is designed to achieve better efficiency and theoretical accuracy, compared with conventional schemes. A particular attention is paid to cells inside the numerically captured shock, by allowing 50% deviations from Hugoniot curve.

We examined the present shock detector in three test cases: Double-Mach-Reflection, Shock/Vortex-Interaction, and Shock/Boundary-Layer-Interaction. In these tests, we compared the result of our method with two representative shock sensors, i.e., Kanamori-Suzuki sensor and Ducros sensor. The former is based on the theory of characteristics and quite accurate in inviscid flows, whereas the latter is very efficient and widely used. In the numerical examples, we confirmed that our method is as accurate as Kanamori-Suzuki method, while as cheap as the Ducros sensor. Moreover, our sensor successfully detected shocks even in a viscous flow, and smoothly represented shocks oblique to the grid lines. In addition, our sensor is free from threshold value settings within the recommended range, and insensitive to the selected slope limiters between minmod and Van Albada.

Acknowledgements

We would like to express our sincere gratitude to Dr. Masashi Kanamori at Japan Aerospace Exploration Agency (JAXA), for giving us valuable advice at the beginning of this work. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Appendix. Ducros sensor

The Ducros sensor ϕ is written as:

$$\phi = \frac{(\nabla \cdot \boldsymbol{u})^2}{(\nabla \cdot \boldsymbol{u})^2 + (|\nabla \times \boldsymbol{u}|)^2 + \delta} \tag{A.1}$$

where δ is a small positive number (1×10^{-30}) to prevent the division by zero where both $\nabla \cdot \boldsymbol{u}$ and $|\nabla \times \boldsymbol{u}|$ are almost zero. This function varies from 0 for regions where the vorticity $|\nabla \times \boldsymbol{u}|$ is sufficiently large compared to velocity divergence (e.g., vortices, boundary-layers), to 1 where velocity divergence is dominant (e.g., shocks). Though this sensor is widely used because of its simplicity, it is sensitive to the velocity divergence $\nabla \cdot \boldsymbol{u}$ even when the vorticity $|\nabla \times \boldsymbol{u}|$ is negligibly small (e.g., smooth flows), sometimes leading to erroneous detections. Therefore, it is of great importance to set an appropriate threshold ϕ, at what value the detected phenomenon is regarded as a shock. This threshold is, however, at the hands of users.

References

[1] T. Horvath, Experimental aerothermodynamics in support of the Columbia accident investigation, in: 42nd AIAA Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings, Reno, Nevada, 2004.

[2] M. Kanamori, T. Takahashi, Y. Makino, Effect of low-boom waveform on focus boom using lossy nonlinear Tricomi analysis, AIAA J. 55 (6) (2017) 2029–2042, https://doi.org/10.2514/1.J055226.

[3] C. Chaussy, E. Schmiedt, D. Jocham, W. Brendel, B. Forssmann, V. Walther, First clinical experience with extracorporeally induced destruction of kidney stones by shock waves, J. Urol. 127 (3) (1982 Mar.) 417–420, https://doi.org/10.1016/S0022-5347(17)53841-0, PubMed.

[4] X. Cai, J. Liang, R. Deiterding, Y. Che, Z. Lin, Adaptive mesh refinement based simulations of three-dimensional detonation combustion in supersonic combustible mixtures with a detailed reaction model, Int. J. Hydrog. Energy 41 (2016) 3222–3239, https://doi.org/10.1016/j.ijhydene.2015.11.093.

[5] N. Ohnishi, K. Kotake, S. Yamada, Numerical analysis on standing accretion shock instability with neutrino heating in the supernova cores, Astrophys. J. 667 (1) (2007) 375–381, https://doi.org/10.1086/500554.

[6] F. Ducros, V. Ferrand, F. Nicoud, C. Weber, D. Darracq, C. Gacherieu, T. Poinsot, Large-eddy simulation of the shock/turbulence interaction, J. Comput. Phys. 152 (1999) 517–549, https://doi.org/10.1006/jcph.1999.6238.

[7] E. Garnier, P. Sagaut, M. Deville, A class of explicit ENO filters with application to unsteady flows, J. Comput. Phys. 170 (2001) 184–204, https://doi.org/10.1006/jcph.2001.6732.

[8] S. Pirozolli, Numerical methods for high-speed flows, Annu. Rev. Fluid Mech. 43 (2011) 163–194, https://doi.org/10.1146/annurev-fluid-122109-160718.

[9] C.M. Winkler, A.J. Dorgany, M. Mani, A Reduced Dissipation Approach for Unsteady Flows on Unstructured Grids, AIAA 2012-0570, https://doi.org/10.2514/6.2012-570, 2012.

[10] S.M. Murman, N.K. Burgess, L.T. Diosady, A. Garai, A DGSEM shock-capturing scheme for scale-resolving simulations, in: 23rd AIAA Computational Fluid Dynamics Conference, AIAA AVIATION Forum, AIAA 2017-4106, https://doi.org/10.2514/6.2017-4106.

[11] K. Mohamed, S. Nadarajah, M. Paraschivoiu, Eddy-preserving limiter for unsteady subsonic flows, AIAA J. 50 (2) (2012) 429–446, https://doi.org/10.2514/1.J051200.

[12] C. Bogey, N. Cacqueray, C. Bailly, A shock-capturing methodology based on adaptative spatial filtering for high-order non-linear computations, J. Comput. Phys. 228 (2009) 1447–1465, https://doi.org/10.1016/j.jcp.2008.10.042.

[13] J.M. Brock, P.K. Subbareddy, G.V. Candler, Detached-eddy simulations of hypersonic capsule wake flow, AIAA J. 53 (1) (2015) 70–80, https://doi.org/10.2514/1.J052771.

[14] G. Salazar, J.R. Edwards, Mach 6 wake flow simulations using a large-eddy simulation/Reynolds-averaged Navier–Stokes model, J. Spacecr. Rockets 51 (4) (2014) 1329–1348, https://doi.org/10.2514/1.A32729.

[15] N. Tajallipour, B.B. Owlam, M. Paraschivoiu, Self-adaptive upwinding for large eddy simulation of turbulent flows on unstructured elements, J. Aircr. 46 (3) (2009) 915–926, https://doi.org/10.2514/1.38945.

[16] S. Camarri, M.V. Salvetti, B. Koobus, A. Dervieux, A low-diffusion MUSCL scheme for LES on unstructured grids, Comput. Fluids 33 (2004) 1101–1129, https://doi.org/10.1016/j.compfluid.2003.10.002.

[17] M. Caraeni, A. West, D. Caraeni, Turbulent jet noise simulation and propagation using a 3rd order MUSCL/CD scheme on unstructured grid and Ffowcs-Williams Hawkings, in: Int. Conf. on Jets, Wakes and Separated Flows, June 16–18, KTH Mechanics, Stockholm, Sweden, 2015.

[18] M.L. Shur, P.R. Spalart, M.K. Strelets, A.K. Travin, Towards the prediction of noise from jet engines, Int. J. Heat Fluid Flow 24 (2003) 551–561, https://doi.org/10.1016/S0142-727X(03)00049-3.

[19] H. Xia, P.G. Tucker, S. Eastwood, Large-eddy simulations of chevron jet flows with noise predictions, Int. J. Heat Fluid Flow 30 (2009) 1067–1079, https://doi.org/10.1016/j.ijheatfluidflow.2009.05.002.

[20] K. Kitamura, A. Hashimoto, Reduced dissipation AUSM-family fluxes: HR-SLAU2 and HR-AUSM$^+$-up for high resolution unsteady flow simulations, Comput. Fluids 126 (2016) 41–57, https://doi.org/10.1016/j.compfluid.2015.11.014.

[21] K. Mohamed, S. Nadarajah, M. Paraschivoiu, Detached-eddy simulation of a wing tip vortex at dynamic stall conditions, J. Aircr. 46 (4) (2009) 1302–1313, https://doi.org/10.2514/1.40685.

[22] M. Kanamori, K. Suzuki, Shock wave detection in two-dimensional flow based on the theory of characteristics from CFD data, J. Comput. Phys. (2011) 3085–3092, https://doi.org/10.1016/j.jcp.2011.01.007.

[23] T.J. Barth, Some Notes on Shock-Resolving Flux Functions, Part 1: Stationary Characteristics, Ames Research Center, 1989, NASA TM 101087.

[24] Z. Wu, Y. Xu, W. Wang, R. Hu, Review of shock wave detection method in CFD post-processing, Chin. J. Aeronaut. 26 (3) (2013) 501–513, https://doi.org/10.1016/j.cja.2013.05.001.

[25] K.M. Perry, S.T. Imlay, Blunt body flow simulations, in: 24th AIAA Joint Propulsion Conference, Boston, MA, 1988, AIAA Paper No. 88-2904.

[26] K. Kitamura, P.L. Roe, F. Ismail, Evaluation of Euler fluxes for hypersonic flow computations, AIAA J. 47 (1) (2009), https://doi.org/10.2514/1.33735.

[27] K. Kitamura, E. Shima, Towards shock-stable and accurate hypersonic heating computations: a new pressure flux for AUSM-family schemes, J. Comput. Phys. 245 (2013) 62–83, https://doi.org/10.1016/j.jcp.2013.02.046.

[28] C. Tenaud, O. Roussel, L. Bentaleb, Unsteady compressible flow computations using an adaptive multiresolution technique coupled with a high-order one-step shock-capturing scheme, Comput. Fluids 120 (2015) 111–125, https://doi.org/10.1016/j.compfluid.2015.07.025.

[29] J.F. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 (6) (1986) 679–698, https://doi.org/10.1109/TPAMI.1986.4767851.

[30] M.J. Aftosmis, M.J. Berger, J.J. Alonso, Applications of a Cartesian Mesh Boundary-Layer Approach for Complex Configurations, AIAA Paper 2006-0652, 2006, https://doi.org/10.2514/6.2006-652.

[31] Y. Kidron, Y. Mor-Yossef, Y. Levy, Robust cartesian grid flow solver for high-Reynolds-number turbulent flow simulations, AIAA J. 48 (2010) 1130–1140, https://doi.org/10.2514/1.45817.

[32] Z.J. Wang, A quadtree-based adaptive cartesian/quad grid flow solver for Navier-Stokes equations, Comput. Fluids 27 (1998) 529–549, https://doi.org/10.1016/S0045-7930(97)00070-4.

[33] Z.J. Wang, R.F. Chen, Anisotropic solution-adaptive viscous Cartesian grid method for turbulent flow simulation, AIAA J. 40 (2002) 1969–1978, https://doi.org/10.2514/2.1558.

[34] T. Kamatsuchi, Turbulent Flow Simulation around Complex Geometries with Cartesian Grid Method, AIAA Paper 2007-1459, 2007, https://doi.org/10.2514/6.2007-1459.

[35] A. Ochi, M.K. Ibrahim, Y. Nakamura, Computational fluid dynamics validation study of wake-capturing capability for a flat-plate wake, J. Aircr. 47 (2010) 441–449, https://doi.org/10.2514/1.33502.

[36] A. Hashimoto, K. Murakami, T. Aoyama, K. Ishiko, M. Hishida, M. Sakashita, P. Lahur, Toward the fastest unstructured CFD code 'FaSTAR', AIAA-2012-1075, https://doi.org/10.2514/6.2012-1075, 2012.

[37] K. Kitamura, K. Fujimoto, E. Shima, K. Kuzuu, Z.J. Wang, Validation of an arbitrary unstructured CFD code for aerodynamic analyses, Trans. Jpn. Soc. Aeronaut. Space Sci. 53 (182) (2011) 311–319, https://doi.org/10.2322/tjsass.53.311.

[38] Y. Tamaki, M. Harada, T. Imamura, Near-wall modification of Spalart–Allmaras turbulence model for immersed boundary method, AIAA J. 55 (9) (2017) 3027–3039, https://doi.org/10.2514/1.J055824.

[39] Y. Tamaki, T. Imamura, Turbulent flow simulations of the common research model using immersed boundary method, AIAA J. 56 (6) (2018) 2271–2282, https://doi.org/10.2514/1.J056654.

[40] D.S. Balsara, Total variation diminishing scheme for adiabatic and isothermal magnetohydrodynamics, Astrophys. J. Suppl. Ser. 116 (1998) 119–131, https://doi.org/10.1086/313093.

[41] J.M. Stone, T.A. Gardiner, P. Teuben, J.F. Hawley, J.B. Simon, Athena: a new code for astrophysical MHD, Astrophys. J. Suppl. Ser. 178 (1) (2008) 137–177, https://doi.org/10.1086/588755.

[42] Y. Matsumoto, Y. Asahina, Y. Kudoh, T. Kawashima, J. Matsumoto, H.R. Takahashi, T. Minoshima, S. Zenitani, T. Miyoshi, R. Matsumoto, Magnetohydrodynamic simulation code CANS+: assessments and applications, e-prints, arXiv:1611.01775, 2016.

[43] P. Tzeferacos, M. Fatenejad, N. Flocke, C. Graziani, G. Gregori, D.Q. Lamb, D. Lee, J. Meinecke, A. Scopatz, K. Weide, FLASH MHD simulations of experiments that study shock-generated magnetic fields, High Energy Density Phys. 17 (A) (2015) 24–31, https://doi.org/10.1016/j.hedp.2014.11.003.

[44] A. Mignone, G. Bodo, S. Massaglia, T. Matsakos, O. Tesileanu, C. Zanni, A. Ferrari, PLUTO: a numerical code for computational astrophysics, arXiv: astro-ph/0701854, 2007. (Accessed 19 October 2017).

[45] M. Sugiura, Y. Tanabe, H. Sugawara, N. Matayoshi, H. Ishii, Numerical simulations and measurements of the helicopter wake in ground effect, J. Aircr. 54 (1) (2017) 209–219, https://doi.org/10.2514/1.C033665.

[46] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228, https://doi.org/10.1006/jcph.1996.0130.

[47] A. Suresh, H.T. Huynh, Accurate monotonicity-preserving schemes with Runge–Kutta time stepping, J. Comput. Phys. 136 (1) (1997) 83–99, https://doi.org/10.1006/jcph.1997.5745.

[48] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. (ISSN 0021-9991) 10 (2) (1972) 252–271, https://doi.org/10.1016/0021-9991(72)90065-4.

[49] F. Grossi, M. Braza, Y. Hoarau, Prediction of transonic buffet by delayed detached-eddy simulation, AIAA J. 52 (10) (2014) 2300–2312, https://doi.org/10.2514/1.J052873.

[50] X. Huang, G. Chandola, D. Estruch-Samper, Unsteady separation shock dynamics in a Mach 4 shock-wave/turbulent-boundary-layer interaction, in: 31st International Symposium on Shock Waves, Nagoya, Japan, vol. SBM000234, 2017.

[51] H.G. Pagendarm, B. Seitz, An algorithm for detection and visualization of discontinuities in scientific data fields applied to flow data with shock waves, in: P. Palamidese (Ed.), Scientific Visualization: Advanced Software Techniques, Prentice Hall India, Delhi, 1993.

[52] L. David, H. Robert, Shock detection from computational fluid dynamics results, in: AIAA 14th Computational Fluid Dynamics Conference, Fluid Dynamics and Co-Located Conferences.

[53] I. Sobel, An isotropic 3×3 gradient operator, in: H. Freeman (Ed.), Machine Vision for Three-Dimensional Scenes, Academic Press, New York, NY, USA, 1990, pp. 376–379.

[54] B. Van Leer, Toward the ultimate conservative difference scheme. 4 a new approach to numerical convection, J. Comput. Phys. 23 (1977) 276–299, https://doi.org/10.1016/0021-9991(77)90095-X.

[55] G.D. Van Albada, B. Van Leer, W.W. Roberts, A comparative study of computational methods in cosmic gas dynamics, Astron. Astrophys. 108 (1982) 76–84.

[56] K. Kitamura, A. Hashimoto, Reduced dissipation AUSM-family fluxes: HR-SLAU2 and HR-AUSM^{+}-up for high resolution unsteady flow simulations, Comput. Fluids 126 (2016) 41–57, https://doi.org/10.1016/j.compfluid.2015.11.014.

[57] C. Hirsch, Numerical Computation of Internal and External Flows: the Fundamentals of Computational Fluid Dynamics, John Wiley & Sons Inc. Print on, New York, 2007.

[58] P.L. Roe, Characteristic-based schemes for the Euler equations, Annu. Rev. Fluid Mech. 18 (1986) 337–365, https://doi.org/10.1146/annurev.fl.18.010186.002005.

[59] V. Daru, C. Tenaud, Evaluation of TVD high-resolution schemes for unsteady viscous shocked flows, Comput. Fluids 30 (2001) 89–113, https://doi.org/10.1016/S0045-7930(00)00006-2.

[60] M.-S. Liou, A sequel to AUSM, Part II: AUSM$+$-up for all speeds, J. Comput. Phys. 214 (2006) 137–170, https://doi.org/10.1016/j.jcp.2005.09.020.

[61] J.S. Park, S.H. Yoon, C. Kim, Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids, J. Comput. Phys. 229 (2010) 788–812, https://doi.org/10.1016/j.jcp.2009.10.011.