Accelerating shock detection using a multigrid approach

Hannah Blumhoefer* and Anupam Sharma†
Iowa State University, Ames, Iowa, USA, 50011

I. Introduction

Shock waves are mathematical singularities in a continuum description of supersonic flow that cause abrupt jumps in pressure, temperature, and density. Shock waves significantly impact flight vehicle performance, stability, and structural integrity through phenomena such as shock-boundary layer interaction and shock-induced stall/flutter. Their influence on aerodynamic forces, drag, and heat transfer underscores the need for accurate shock detection in numerical simulations and physical experiments.

In computational fluid dynamics (CFD), shocks are treated in one of two ways: a) *shock fitting*, where the computational mesh is aligned with the shock line/surface (requiring a priori knowledge of shock locations) and Rankine-Hugoniot jump conditions are applied across the shocks, or b) *shock capturing*, where the shocks are computed as a part of the simulation procedure. While shock-capturing methods do not require a priori shock information, regions, where such discontinuities exist, require special treatment to avoid producing unphysical solutions (Gibbs phenomena). In high-order ($\geq$ second) numerical schemes, shocks are treated by effectively reducing the order of accuracy of the scheme at the discontinuity, achieved by limiting the fluxes or adding numerical diffusion. High-resolution schemes for compressible flow simulations where such discontinuities are present require some means of identifying and treating non-smoothness in the solution.

Hybrid schemes built on compact finite differencing switch to shock capturing schemes, such as weighted-essentially non-oscillatory (WENO), Roe scheme, etc., in regions where discontinuities in the flow variables are identified. Examples of such schemes are the hybrid compact-WENO scheme by Pirozzoli [1], and the hybrid compact-Roe scheme explored by Visbal and Gaitonde [2]. Identification of shocks is critical for such hybrid schemes to avoid Gibbs phenomenon.

Shock detectors/sensors are numerical algorithms for identifying shock waves in fluid flows. Wu et al. [3] provides an overview of commonly used shock sensors. Traditional shock sensors, such as [4, 5], rely on first- and/or second-order derivatives of the flow variables to evaluate smoothness. Hill and Pullin [6] proposed using WENO (Weighted Essentially Non-Oscillatory) weights to assess stencil regularity; the ratio between the largest and smallest WENO weights provides a measure of smoothness. Visbal and Gaitonde [2] used a WENO-type smoothness criterion for shock detection. Such gradient/smoothness-based detectors can effectively detect discontinuities but misidentify high-wavenumber waves as

*Graduate Student, Aerospace Engineering, 537 Bissell Road, Howe Hall #1200, AIAA Student Member
†Corresponding author, sharma@iastate.edu; Professor, Aerospace Engineering, 537 Bissell Road, Howe Hall #2235, AIAA Associate Fellow.

shocks. Physics-based sensors are also widely used as shock detectors. A prominent example is the Ducros sensor [7], which uses the relative importance of dilatation to vorticity to differentiate between shock and turbulent regions. Another example is the Kanamori-Suzuki sensor [8], which uses the convergence of characteristics for shock identification. Other methods include multi-resolution wavelet sensors [9], wave-normal Mach number calculations accounting for moving shocks [10], solving shock function transport equations [11], and modal smoothness [12].

Recent developments include the use of edge-detection techniques for shock detection [13]; such techniques, for example the Canny algorithm [14], have traditionally been used for image processing. Fujimoto et al. [15] explored modifications to the Canny algorithm to detect shocks by accounting for the Rankine-Hugoniot jump conditions. Neural networks have also been explored for shock detection. Ray and Hesthaven [16] used a multilayer perceptron (an artificial neural network or ANN), trained offline using supervised learning, as a blackbox to identify "troubled" cells. Morgan et al. [17] trained and used an ANN to identify shock regions and demonstrated it for multidimensional shock problems. Liu et al. [18] proposed a convolutional neural network (CNN) utilizing a novel loss function to provide potential speedup or accuracy over conventional detectors while maintaining problem generalization. Beck et al. [19] also implemented CNNs to achieve parameter-free detection without requiring any tuning by the user.

While prior work on shock detectors has focused primarily on improving precision, there is little-to-no work on reducing the computational cost associated with shock detection. We propose a framework that uses multiple grid levels, evaluating the shock sensor progressively from coarse to fine grids, but only in regions identified as shock-positive in the previous (coarser) grid level. We refer to this as a multilevel shock-detection framework. Oliveira et al. [20] also uses a combination of fine and coarse meshes, but for the purpose of accurate shock detection, not acceleration. Their method is referred to in the literature as multilevel shock detection, but their approach and purpose differ. We demonstrate the accuracy and speedup of our multilevel framework with three commonly used shock detectors on two- and three-dimensional problems. We assert, however, that the framework is general and can be used with any mesh-based shock-detection approach.

The remainder of this paper is structured as follows. Section II describes the methodology behind the multilevel framework and the shock detection algorithms evaluated in this work. Numerical experiments evaluating the speedup and accuracy of the proposed multilevel framework are presented in section III, and the key findings are summarized in section IV.

II. Methodology

The premise of the framework is that flow discontinuities, such as shocks, can be detected using a mesh significantly coarser than required for solving the governing fluid flow equations. Since such discontinuities typically occupy a very small fraction of the computational domain, large portions of the computational domain can be rapidly determined to be free of shocks by evaluating the shock sensor over a much coarser mesh. We can save computational time without

sacrificing accuracy by progressively evaluating the shock sensor on increasingly finer meshes only in regions flagged as shock-positive on a coarser mesh. Figure 1 illustrates our proposed multilevel shock detection approach. In fig. 1, h is the cell size at the finest level, the gray lines show the computational mesh over which the governing flow equations are solved; the blue lines show the mesh used at different grid levels for shock detection; the red curve indicates a shock, and the cells flagged as shock-positive in a grid level, identified for shock detection at the next level, are highlighted in yellow.

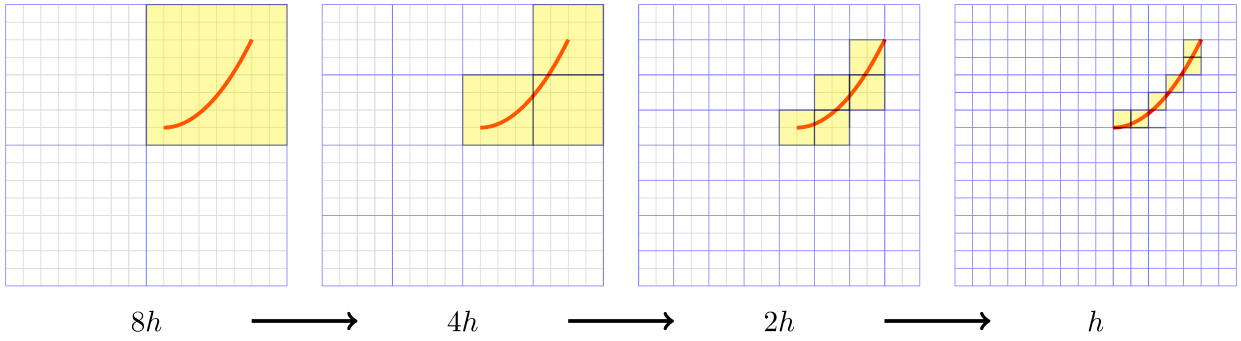


Fig. 1 Illustration of the proposed multilevel shock detection algorithm over four grid levels

A. Multilevel shock-detection approach

A hierarchical refinement process, characterized by a coarse-to-fine mesh progression, forms the foundation for developing multilevel shock detectors. The approach is similar to the second (prolongation) stage of a multigrid V-cycle (see fig. 2). Mesh refinement occurs in a sequence that adheres to grid space halving, i.e., $8h,\ 4h,\ 2h,\ h$, where h is the cell size of the (finest) mesh used for solving the governing fluid flow equations. We demonstrate the procedure with a structured single-block mesh; application to multi-block structured/Overset meshes is straightforward. The mesh and numerical solution at the n^{th} grid level are obtained by selecting every n^{th} point of the finest mesh. The restriction operator, therefore, does not require additional computation.

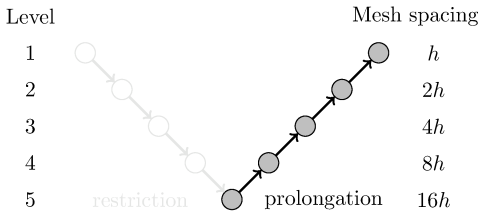


Fig. 2 Progressive shock detection over multiple grids resembles the prolongation phase of a multigrid V-cycle

The following approach is used to deal with grids where the number of mesh points that are not directly divisible into the desired number of levels of a hierarchy. Figure 3 explains how the 'leftover' points in such grids are handled. The vertical black lines in the figure show the mesh points that permit n (=3 in fig. 3) levels of hierarchy. A three-point stencil

for shock detection starting from the left (denoted by $i-1,\ i,\ i+1$ in black in the figure) is used in this region. The blue region contains the leftover points that need to separately addressed. Shock detection in this region is performed by forming a stencil starting from the right (shown in red in fig. 3). Overlap between the 'normal' and leftover regions ensures the entire domain is covered using the same stencil sizes at each level. This process is repeated for all directions containing leftover points.

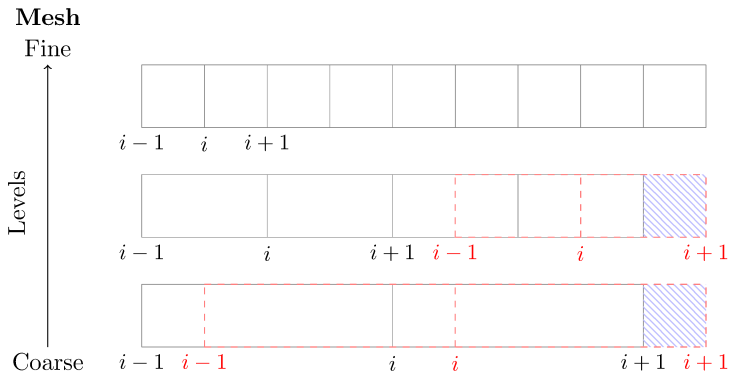


Fig. 3 Multilevel treatment when the number of mesh points along each axis does not permit exact division into the desired level of hierarchy.

Shock detection is conducted at each grid level using a predetermined, detector-dependent threshold on a sensor-relevant quantity, e.g., the magnitude of the pressure gradient. Cells/regions where the threshold is exceeded are flagged as shock-positive. Shock detection in each subsequent (refined) grid level is performed *only* in the regions flagged in the previous grid level. Successive refinement of the flagged regions ultimately results in the identification of shocks at the desired (finest) grid level (see fig. 1). Multiple grid-level progressions are considered to evaluate the computational efficiency (time savings) and accuracy of the proposed multilevel shock detection framework for four test cases, including one three-dimensional flow/shock case.

B. Shock Detectors/Sensors

Three shock sensors are selected for evaluation with the proposed multilevel shock-detection framework: Ducros, convolution-based, and WENO. The Ducros sensor [7] is widely used in CFD due to its ease of implementation and ability to differentiate between shocks and compressible turbulence [15]. The inherent efficiency of convolutional neural networks (CNNs) on graphical processing units (GPUs) has spurred their growing application in CFD for feature extraction and other critical tasks, making a convolution-based sensor an appropriate selection. The WENO detector [6] is selected because of its capability to estimate local solution regularity more precisely through its smoothness indicators, unlike traditional detectors that rely solely on low-order derivatives [21].

1. Ducros Sensor

The Ducros shock sensor enhances the Jameson sensor [5] by multiplying it with a factor (Ducros value) that distinguishes shocks from compressible turbulence through a relative comparison of dilatation and vorticity magnitude (eq. (1)). The divergence and curl operators in eq. (1) use the dimensionality of the problem, i.e, $\boldsymbol{\nabla} \cdot \boldsymbol{u}$ is evaluated as $(\partial_x u + \partial_y v)$ in 2D and $(\partial_x u + \partial_y v + \partial_z w)$ in 3D problems. The Ducros value in each cell is multiplied with the Jameson sensor evaluated along each coordinate; eq. (1) shows the expression for evaluation along the index i.

$$\Phi^{(I)}_{i,j,k} = \underbrace{\frac{(\boldsymbol{\nabla} \cdot \boldsymbol{u})^2}{(\boldsymbol{\nabla} \cdot \boldsymbol{u})^2 + |\boldsymbol{\nabla} \times \boldsymbol{u}|^2 + \epsilon_D}}_{\text{Ducros value (DV)}} \cdot \underbrace{\left| \frac{p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k}}{p_{i+1,j,k} + 2p_{i,j,k} + p_{i-1,j,k}} \right|}_{\text{Jameson sensor along } i} \tag{1}$$

where ϵ_D is a small, positive value. The cell (i, j, k) is flagged shock-positive if $\Phi^{(\mathrm{D})}_{i,j,k} = \max\left(\Phi^{(I)}_{i,j,k}, \Phi^{(J)}_{i,j,k}, \Phi^{(K)}_{i,j,k}\right)$ exceeds a predetermined problem-dependent threshold. Consideration is given to two different versions of the Ducros sensor. In the first variant, we compute the Ducros value (involving the expressions $\boldsymbol{\nabla} \cdot \boldsymbol{u}$ and $\boldsymbol{\nabla} \times \boldsymbol{u}$) as part of the shock-detection algorithm at each grid level. In contrast, in the second, we compute these only on the finest grid outside the shock-detection algorithm, i.e., the time to evaluate these quantities is not included in the computation time for shock detection. The second variant mimics the scenario where these values are precomputed elsewhere in the program (for solving the conservation laws, for example). We refer to the first variant as Ducros-I and the second as Ducros-O.

2. Convolution-based Sensor

Convolutions are used in image processing for edge detection by identifying discontinuities in pixel brightness. The procedure entails convolving a filter, typically a combination of gradient and smoothing operators, with a matrix of brightness values corresponding to the image pixels. An example is the Sobel filter [22], which uses two 3×3 kernels M_X and M_Y to approximate the first derivatives along the horizontal (x) and vertical (y) directions, respectively. These kernels can be decomposed as the products of an averaging kernel and a differentiation kernel, as shown below.

$$M_X = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}; \qquad M_Y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \tag{2}$$

Shocks, associated with discontinuities in pressure, can be treated as edges appearing in a flow and thus can be detected by convolving pressure values with the Sobel kernels. At each mesh point, given by a pair of (i, j) indices, we define a 3×3 matrix P_{ij} as

$$P_{ij} = \begin{bmatrix} p_{i-1,j+1} & p_{i-1,j} & p_{i-1,j-1} \\ p_{i,j+1} & p_{i,j} & p_{i,j-1} \\ p_{i+1,j+1} & p_{i+1,j} & p_{i+1,j-1} \end{bmatrix}, \tag{3}$$

where $p_{i,j}$ is the static pressure in cell (i, j). We evaluate the convolutions $X_{ij} = M_X * P_{ij}$ and $Y_{ij} = M_Y * P_{ij}$ to estimate horizontal and vertical pressure gradients respectively; note that X_{ij} and Y_{ij} are 3×3 matrices for each (i, j) pair. The overall pressure gradient magnitude is estimated by combining the Frobenius norms of X_{ij} and Y_{ij}.

$$\Phi_{ij}^{(\mathrm{C})} = \sqrt{\|X_{ij}\|^2 + \|Y_{ij}\|^2} \tag{4}$$

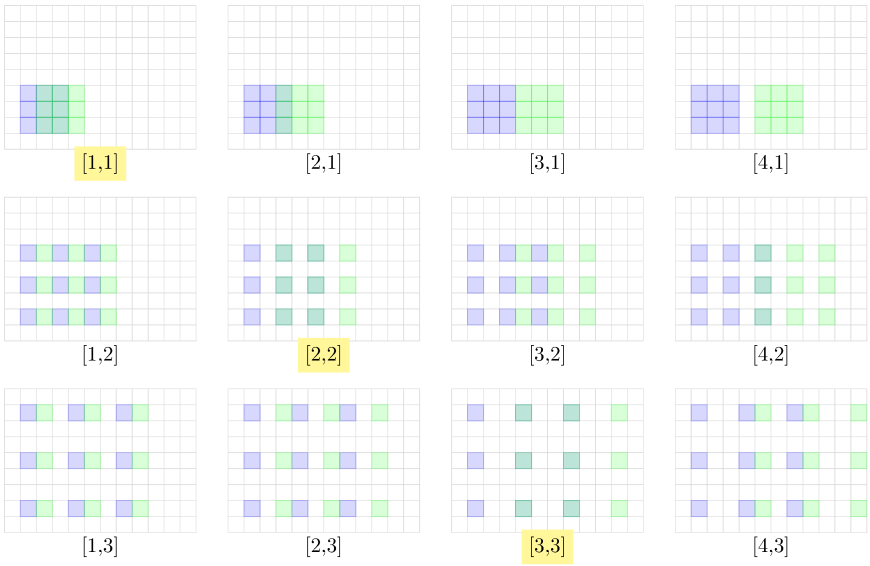
Cell (i, j) is flagged to contain a shock if $\Phi_{ij}^{(\mathrm{C})}$ exceeds a predetermined threshold, which is problem-dependent and requires tuning.

$$\mathrm{S}_{ij} = \begin{cases} 1, & \text{for } \Phi_{ij}^{(\mathrm{C})} > \text{threshold} \\ 0, & \text{otherwise} \end{cases} \qquad \forall \qquad i = 1, 2, \ldots, M \;\; j = 1, 2, \ldots, N \tag{5}$$

In convolutions, *stride* refers to the shift across cells of the filter (e.g., $M_X,\ M_Y$) when performing a convolution. For example, with a 3×3 filter, a stride of one implies a one-cell shift for successive convolutions, resulting in an overlap between convolution areas; a stride of three eliminates this overlap. *Dilation* refers to the expansion of the filter. Figure 4a illustrates stride and dilation for a 3×3 filter. The blue colored cells in fig. 4a show the filter and the green colored cells show the shifted (by stride) filter; the numbers in brackets represent [stride,dilation] values. Convolutions at different grid levels can be performed by simultaneously adjusting both the stride and dilation values, ensuring they remain equal; see subpanels labeled $[1, 1]$, $[2, 2]$, and $[3, 3]$ in fig. 4a. Figure 4b shows the stride and dilation values used for convolution-based multilevel shock detection for different grid coarseness. As an example, shock detection using four grid levels would use stride (=dilation) values of $8,\ 4,\ 2,\ 1$ on meshes with cell sizes $8h,\ 4h,\ 2h$, and h, respectively.

Extension to 3D Two distinct approaches are used to extend the convolution-based sensor to 3D. The first approach treats the entire 3D flow field as a series of 2D slices. The 2D convolution-based multilevel shock detection algorithm described above is applied to each 2D slice individually. For instance, in the z-direction, x-y slices (z = constant planes) are evaluated. This is repeated for the x and y directions (see fig. 5) for each multilevel step. For curvilinear grids where the physical coordinates (x, y, z) correspond to the computational coordinates (ξ, η, ζ), the same approach would use constant ξ, η, and ζ planes. If a shock is detected in any of the three planes passing through a grid point, the point is marked shock-positive.

The second approach uses 3D Sobel filters for shock detection. These filters are third-order tensors of dimension



(a) Stride/dilation in convolutions

# Levels	*Stride/dilation*
1	1
2	2, 1
3	4, 2, 1
4	8, 4, 2, 1
5	16, 8, 4, 2, 1

(b) Stride/dilation values for multilevel shock detection

Fig. 4 Stride and dilation in convolutions: (a) a schematic illustrating the concepts with a 3×3 filter; and (b) the progressions of stride(=dilation) values used for multilevel shock detection.

$3 \times 3 \times 3$, which perform differentiation along one direction and averaging along the other two. For example, the kernel in the z-direction is given by

$$M_Z(:,:,-1) = \begin{bmatrix} -1 & -2 & -1 \\ -2 & -4 & -2 \\ -1 & -2 & -1 \end{bmatrix}, \quad M_Z(:,:,0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad M_Z(:,:,1) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Convolutions $M_X * P_{ijk}$, $M_Y * P_{ijk}$, and $M_Z * P_{ijk}$ are evaluated, and their Frobenius norms combined to obtain an estimate of the overall pressure gradient, $\Phi^{(\mathrm{C})}_{ijk}$.

3. WENO Sensor

A WENO-type smoothness parameter, a combined slope and curvature variation measure, is used to detect pressure discontinuities.

$$\Phi^{(I)}_{i,j,k} = \left[a \left(p_{i+1,j,k} - p_{i-1,j,k} \right)^2 + b \left(p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k} \right)^2 \right]^n \tag{6}$$

where $a = 1/4$ and $b = 13/12$ are as defined in a fifth-order WENO scheme [23]; $n = 2$ is used to sharpen the discontinuities [2]. A shock is identified when $\Phi^{(\mathrm{W})}_{i,j,k} = \max\left(\Phi^{(I)}_{i,j,k}, \Phi^{(J)}_{i,j,k}, \Phi^{(K)}_{i,j,k}\right)$ exceeds a predetermined threshold [24].

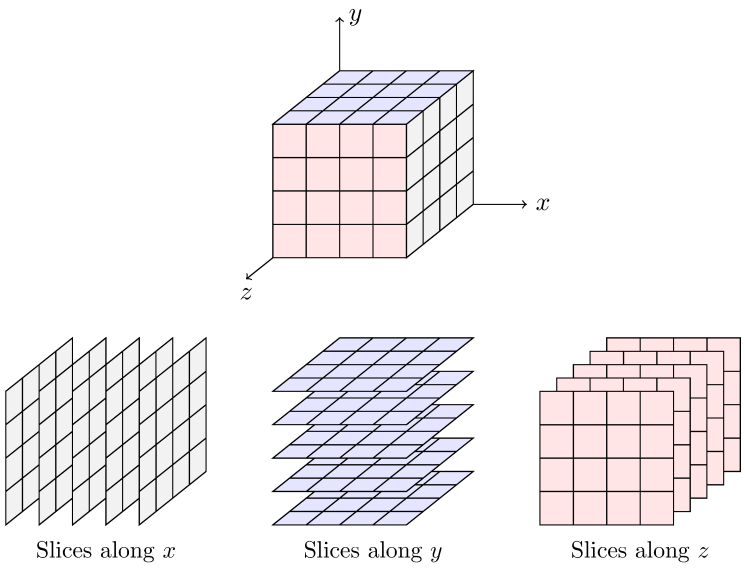


Fig. 5 2D slice decomposition for 3D shock detection

III. Numerical Experiments

The performance of the multilevel shock-detection framework is assessed using the following test cases. (1) cylindrical shock wave propagation, (2) shock-vortex interaction, (3) supersonic flow over a wall-mounted half-cylinder, and (4) spherical shock wave propagation. The first three are two-dimensional (either simulated in 2D or treated as 2D by span-averaging), whereas the fourth is three-dimensional. Numerical solutions to these problems are obtained using FDL3DI, a high-order, compressible Navier-Stokes solver. FDL3DI uses a hybrid compact-Roe scheme [2] with sixth-order accurate compact finite differencing for spatial discretization of smooth regions and an implicit, approximately-factored procedure [25] for time integration. Regions detected to have discontinuities are solved using the Roe scheme. Spatial filtering is employed to remove scales not resolved by the grid, which serves to stabilize the computation and act as a subgrid stress model. Details on the solver are available in Visbal and Gaitonde [25] and Sharma and Visbal [26].

Accuracy is assessed by comparing the shock-positive regions identified by the multilevel framework on the finest grid with a numerical Schlieren for each test case. Following Monfort et al. [27], the Schlieren value for each grid point (i, j, k) is obtained as

$$S_{ijk} = 0.8 \exp\left(-\frac{10\,|\boldsymbol{\nabla}\rho|}{\max|\boldsymbol{\nabla}\rho|}\right). \tag{7}$$

Computational time savings are reported as speedup, which is the ratio of the time required for shock detection using only the finest grid level to the time required using the multilevel framework. Speedup is reported for each detector for different numbers (up to five) of grid levels used. The computation times used to evaluate speedups are averaged over

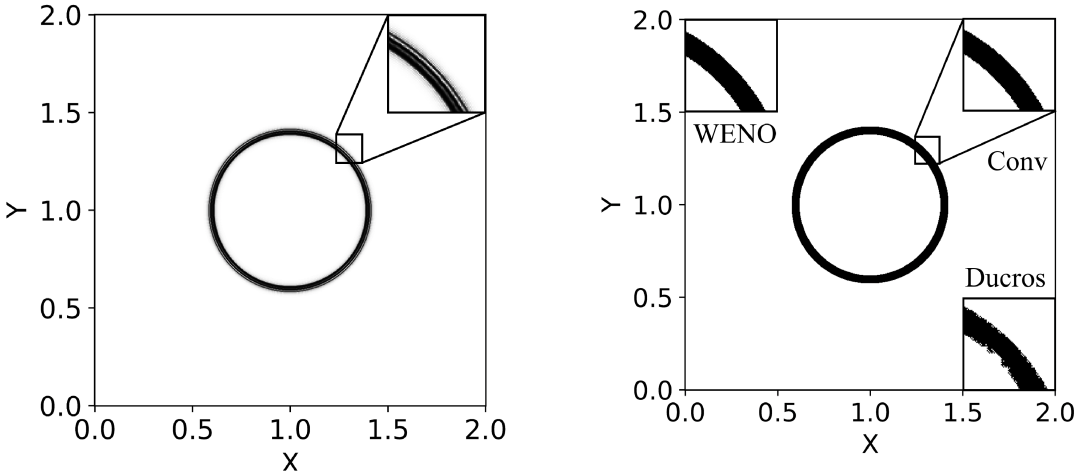
100 ensembles; the standard deviation (error) is ±0.01.

A. Cylindrical Shock

An extension of the 1D shock tube (Sod) problem involving cylindrical wave propagation is considered. A Cartesian grid discretizes the 2D computational domain spanning two units along each coordinate. A circular diaphragm centered at (1,1) of radius $R = 0.4$ separates a high-pressure, high-density region from a low-pressure, low-density region. The initial conditions are

$$(\rho, u, v, p) = \begin{cases} (1.0,\ 0,\ 0,\ 1.0) & \forall\, r = (x^2 + y^2)^{1/2} \leq R \\ (0.125,\ 0,\ 0,\ 0.1) & \forall\, r > R \end{cases} \tag{8}$$

The diaphragm is ruptured instantaneously at $t = 0$, leading to the generation and propagation of a cylindrical/circular shock wave. Figure 6 shows the numerical Schlieren and the regions flagged as shock-positive by each detector at $t = 0.01$. Contours in the sub-panels (in the corners of the main figure) show the regions flagged as shock-positive by the three shock detectors. Multilevel detection with five levels is used for each sensor. All detectors identify the shock front. Figure 7 presents the speedups for different shock detectors. Every shock sensor benefits from the proposed multilevel framework with the highest speedup of ~7 for the WENO detector and the lowest speedup for Ducros-I.



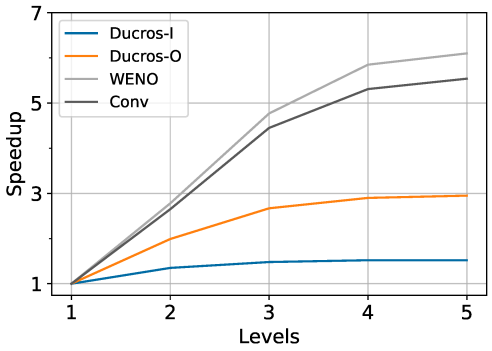
(a) Numerical Schlieren

(b) Contours in the sub-panels show the regions flagged as shock-positive

Fig. 6 Shock detection for the cylindrical shock wave problem

B. Shock-Vortex Interaction

The second test case involves the 2D shock-vortex interaction problem outlined in Visbal and Gaitonde [24]. A convecting vortex interacts with a stationary planar shock wave in a computational domain spanning $[-6, 9]$ in the x direction and $[-8, 8]$ in the y direction. The top and bottom boundaries (along y) are periodic. The inflow Mach



(a) Speedup in shock detection

Detector	Thres	Levels			
		2	3	4	5
Ducros-I	0.01	1.35	1.48	1.52	1.52
Ducros-O	0.01	1.99	2.67	2.90	2.95
WENO	3e-7	2.78	4.77	5.85	6.10
Conv	0.1	2.65	4.45	5.31	5.54

(b) Speedup factors

Fig. 7 Speedup in shock detection using the multilevel framework for the cylindrical shock problem.

number specified on the left boundary is 1.2, and the right boundary is outflow. A standing normal shock (pressure jump of 1.513) is located at $x = 2$. The vortex center is initially located at $(-1, 0)$ with the following velocity and pressure distributions.

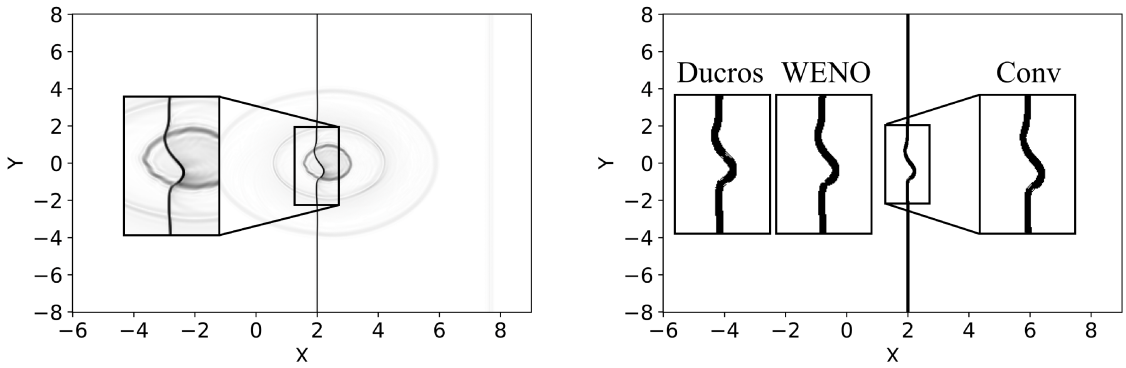
$$u = 1 - 0.4\,y\,\exp\left(-r^2/2\right), \quad v = 0.4\,(x+1)\,\exp\left(-r^2/2\right), \quad p = \frac{1}{\gamma M_\infty^2}\,\exp\left[-\frac{0.16\,\gamma\,M_\infty^2}{2}\exp\left(-r^2\right)\right]. \tag{9}$$

Figure 8 shows the numerical Schlieren images and the regions identified as shock-positive by the different sensors at $t = 3.5$ and 5.0. Shock contours are shifted along the abscissa for a compact arrangement in fig. 8. Multilevel detection with five levels is used for each sensor. Figure 9 presents the speedup with the multilevel framework for the different shock sensors. All detectors are accelerated by the multilevel framework, with WENO showing almost a nine-fold increase in computational efficiency.

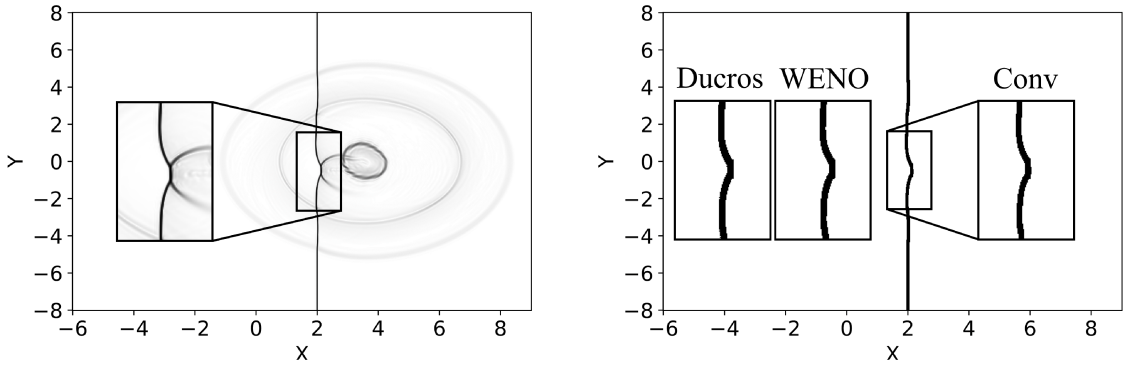
C. Wall-mounted Cylinder

Supersonic flow over a wall-mounted half cylinder is a canonical problem to study flow over optical canopies in fighter jets [28]. The half-cylinder has a unit radius, and the computational C-grid spans $[1, 6]$ in the radial direction and $[0, \pi]$ in the θ direction. The wall on which the cylinder is mounted is at $y = 0$. Freestream conditions are specified at the $r = 6$ boundary, and the cylinder and $y = 0$ plane are modeled as no-slip, adiabatic walls. The flow Reynolds and Mach numbers are 300, 000 and 2.0 respectively. The calculation is 3D with periodic boundaries along the span (in z), but the analysis here is based on span-averaged 2D results.

Figure 10a shows the numerical Schlieren at one time instant. Multiple shocks that interact with each other are present in the flow. Panels b, c, and d in fig. 10 show the regions flagged as shock-positive by each detector. Multilevel detection with five levels is used for each detector. Figure 11a presents the speedup with the multilevel framework for different shock detectors. As in the previous test cases, the WENO detector shows the greatest speedup.

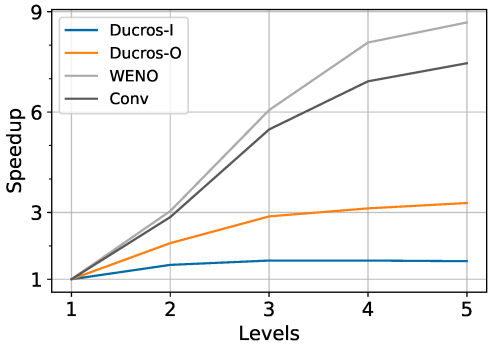


(a) Schlieren at $t = 3.5$ **(b) Shock detection at $t = 3.5$**



(c) Schlieren at $t = 5.0$ **(d) Shock detection at $t = 5.0$**

Fig. 8 Numerical Schlieren (left) and shock-positive regions identified by three sensors (right) for the shock-vortex interaction problem at $t = 3.5$ and 5.0.



(a) Speedups in shock detection

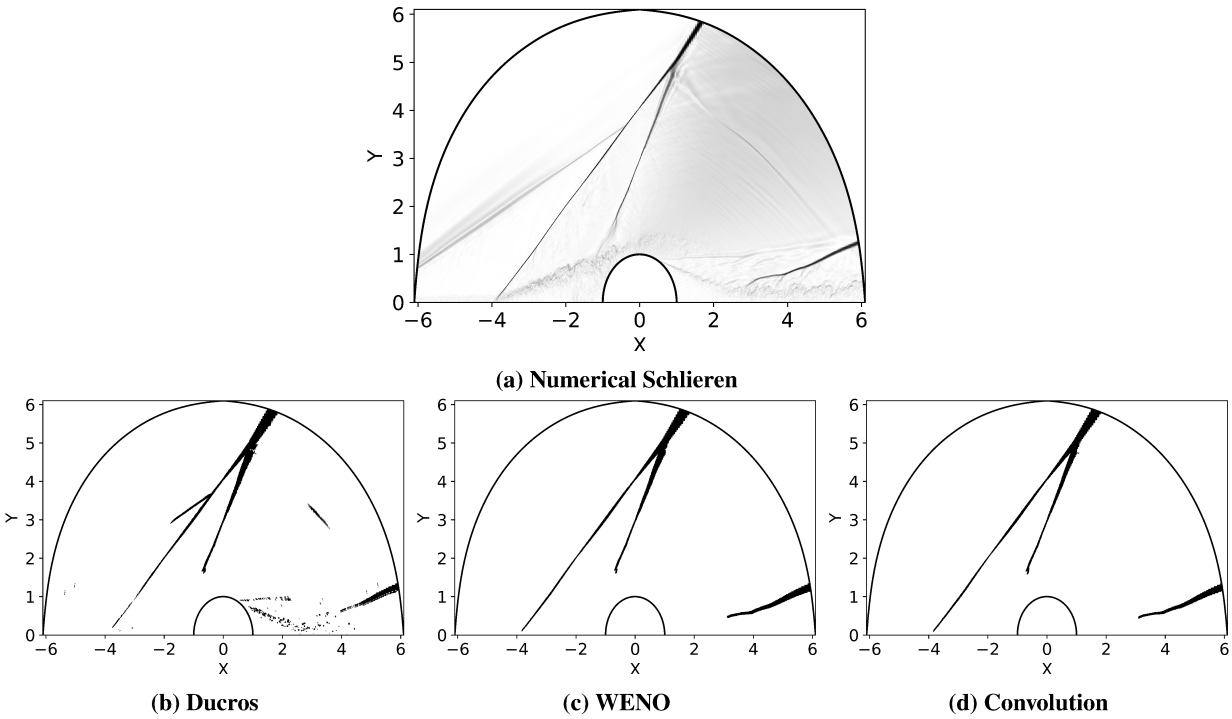
Detector	Thres	Levels			
		2	3	4	5
Ducros-I	0.005	1.43	1.56	1.56	1.54
Ducros-O	0.005	2.08	2.88	3.12	3.28
WENO	5e-5	3.04	6.06	8.08	8.68
Conv	0.7	2.86	5.48	6.92	7.46

(b) Speedup factors

Fig. 9 Speedups in shock detection using the multilevel framework for the shock-vortex interaction problem

D. Spherical Shock

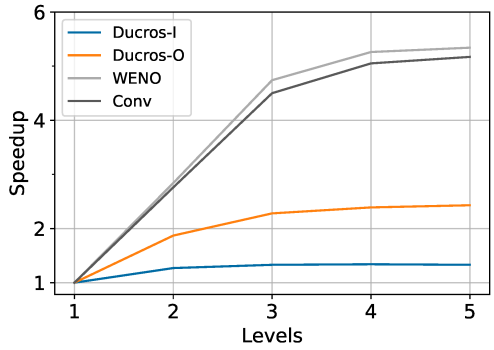
The final test case is an extension of the Sod problem to 3D. A spherical diaphragm separates a high-pressure, high-density gas from a low-pressure, low-density gas. The initial conditions are the same as in eq. (8) with the diaphragm centered at $(1, 1, 1)$. The diaphragm bursts instantaneously at $t = 0$, leading to spherically-spreading waves. Figure 12 shows the shock-positive regions for each of the detectors in comparison to the numerical Schlieren at $t = 0.01$



(a) Numerical Schlieren

(b) Ducros (c) WENO (d) Convolution

Fig. 10 Results for the wall-mounted cylinder problem. Panels b,c, & d show the shock-positive regions identified by the different sensors.



(a) Speedup

Detector	Thres	Levels			
		2	3	4	5
Ducros-I	0.003	1.27	1.33	1.34	1.33
Ducros-O	0.003	1.87	2.28	2.39	2.43
WENO	1e-5	2.84	4.74	5.26	5.34
Conv	0.4	2.76	4.50	5.05	5.17

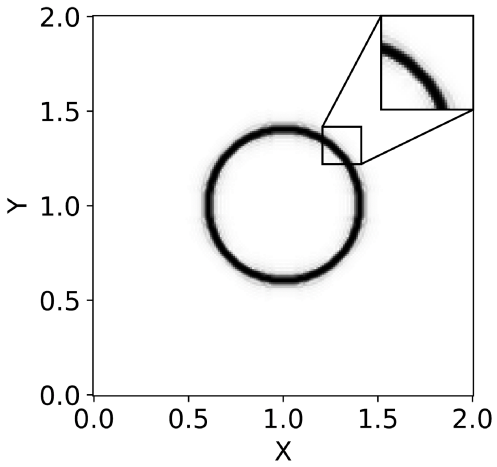
(b) Speedup factors

Fig. 11 Speedups in shock detection using the multilevel framework for supersonic flow over a wall-mounted cylinder.

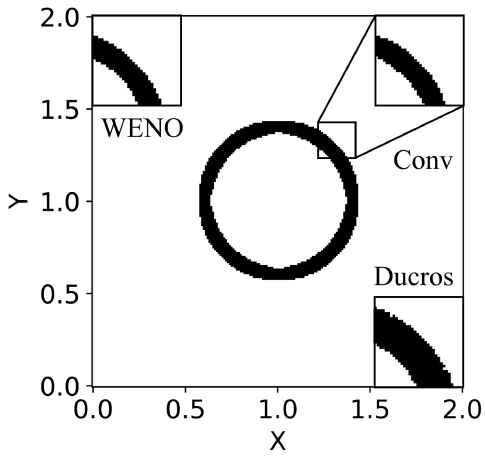
in the simulation. Figure 13a shows the speedups for the different shock detectors; the 3D Convolution detector shows the greatest speedup (~17x).

E. Accuracy

Accuracy is assessed by comparing the shock-positive cells identified by the shock sensors using the conventional, single-level approach on the highest-resolution grid with the multilevel framework results. The difference in the number of cells identified as shock-positive by the two methods is a measure of error. It is normalized by the total number of

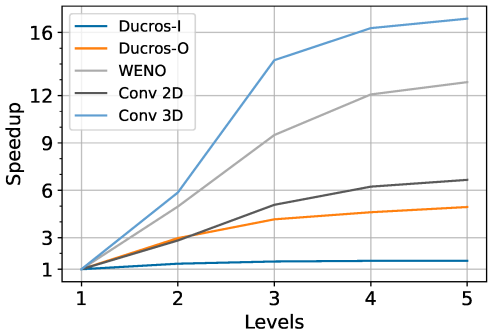


(a) Numerical Schlieren



(b) Contours in the sub-panels show the regions flagged as shock-positive

Fig. 12 Numerical Schlieren and shock-positive regions identified by three shock detectors using the multilevel framework for the spherical shock wave propagation problem.



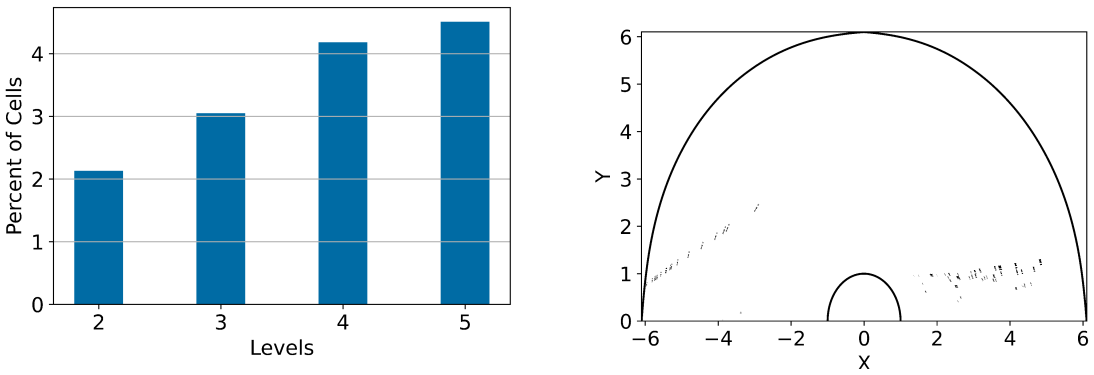
(a) Speedup

Detector	Thres	Levels			
		2	3	4	5
Ducros-I	0.03	1.45	1.54	1.55	1.58
Ducros-O	0.01	2.93	3.85	4.04	4.05
WENO	1e-4	4.74	9.08	9.97	10.00
Conv 2D	1	2.82	5.08	6.23	6.66
Conv 3D	3	5.86	14.24	16.72	16.87

(b) Speedup factors

Fig. 13 Speedups in shock detection using the multilevel framework for the spherical-shock problem

shock-positive cells identified by the single-level approach. The accuracy results for the wall-mounted cylinder with the Ducros sensor are shown in fig. 14. The percent difference in the number of cells identified as shock positive is small ($\leq$ 4.5%), with most cells located in the turbulent region. Figure 14b shows that the error is located in regions of very weak shocks (approaching Mach waves); these regions are not identified as shock-positive by the Convolution and WENO sensors (fig. 10), suggesting a larger discrepancy in identifying shock-positive regions across sensors. Hence, we argue that "error" with the multi-level framework is of little significance. The difference is negligible for the other test cases and detectors.



(a) Percent #cell difference from Level 1 **(b) Cell locations**

Fig. 14 Multilevel accuracy for the wall-mounted cylinder problem

IV. Conclusion

We present a framework to accelerate shock detection in numerical simulations (or postprocessing) by utilizing multiple grid levels, starting with a coarse mesh and adaptively refining shock regions. Three shock sensors (Ducros, convolution-based, and WENO) are evaluated within the framework. We demonstrate significant improvements in computational efficiency (speedup) without sacrificing accuracy for 2D and 3D problems. Speedups for 2D problems range between 5x to 9x for WENO and 5x to 7x for the convolution sensor. The speedup for Ducros is 2.5x to 3x when divergence and vorticity are precomputed; computing these at each level reduces the speedup to 1.5x. The multilevel approach offers greater speedups when shocks occupy smaller regions. This is evident in the 3D case, where the convolution sensor achieves the highest speedup (~17x). The proposed framework provides a simple and effective way to accelerate shock detection for various sensors.

Funding Sources

This material is based upon work supported by the US Air Force Office of Scientific Research (Award # FA9550-23-1-0016) and the National Science Foundation (Grants CBET-1935255 and 1554196). The first author acknowledges support by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate (NDSEG) Fellowship Program. We also acknowledge the computational resources provided by the Department of Defense and Iowa State University.

Acknowledgments

We acknowledge Dr. Daniel Garmann of the Air Force Research Laboratory for providing the FDL3DI simulation results for the wall-mounted cylinder problem and for overall guidance.

References

[1] Pirozzoli, S., "Conservative hybrid compact-WENO schemes for shock-turbulence interaction," *Journal of Computational Physics*, Vol. 178, No. 1, 2002, pp. 81–117.

[2] Visbal, M., and Gaitonde, D., "Shock capturing using compact-differencing-based methods," *43rd AIAA aerospace sciences meeting and exhibit*, 2005, p. 1265.

[3] Wu, Z., Xu, Y., Wang, W., and Hu, R., "Review of Shock Wave Detection Method in CFD post-processing," *Chinese Journal of Aeronautics*, Vol. 26, No. 3, 2020, pp. 501–513. https://doi.org/10.1016/j.cja.2013.05.001.

[4] Harten, A., "The artificial compression method for computation of shocks and contact discontinuities. III. Self-adjusting hybrid schemes," *Mathematics of Computation*, Vol. 32, No. 142, 1978, pp. 363–389.

[5] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods using Runge Kutta Time Stepping Schemes," *14th Fluid and Plasma Dynamics Conference*, Palo Alto, CA, USA, 1981.

[6] Hill, D., and Pullin, D., "Hybrid Tuned Center Difference WENO method for Large Eddy Simulations in the Presence of Strong Shocks," *Journal of Computational Physics*, Vol. 194, No. 2, 2004, pp. 435–450. https://doi.org/10.1016/j.jcp.2003.07.032.

[7] Ducros, F., Ferrand, V., Nicoud, F., Weber, C., Darracq, D., Gacherieu, C., and Poinsot, T., "Large-eddy simulation of the shock/turbulence interaction," *Journal of Computational Physics*, Vol. 152, No. 2, 1999, pp. 517–549. https://doi.org/10.1006/jcph.1999.6238.

[8] Kanamori, M., and Suzuki, K., "Shock wave detection in two-dimensional flow based on the theory of characteristics from CFD Data," *Journal of Computational Physics*, Vol. 230, 2011, pp. 3085–3092. https://doi.org/10.1016/j.jcp.2011.01.007.

[9] Sjögreen, B., and Yee, H. C., "Multiresolution wavelet based adaptive numerical dissipation control for high order methods," *Journal of Scientific Computing*, Vol. 20, 2004, pp. 211–255.

[10] Lovely, D., and Haimes, R., "Shock Detection from Computational Fluid Dynamics Results," *14th Computational Fluid Dynamics Conference*, 1999. https://doi.org/10.2514/6.1999-3285.

[11] Rathi, H., and Sinha, K., "Numerical Detection of Shock Location and Shock Strength in Unsteady Flow Computations," *AIAA AVIATION 2021 FORUM*, 2021. https://doi.org/10.2514/6.2021-2867.

[12] Persson, P.-O., and Peraire, J., "Sub-Cell Shock Capturing for Discontinous Galerkin Methods," *44th AIAA Aerospace Sciences Meeting and Exhibit*, 2006. https://doi.org/10.2514/6.2006-112.

[13] Cui, S., Wang, Y., Qian, X., and Deng, Z., "Image Processing Techniques in Shockwave Detection and Modeling," *Journal of Signal and Information Processing*, Vol. 4, No. 3, 2013, pp. 109–113. https://doi.org/10.4236/jsip.2013.43b019.

[14] Canny, J., "A computational approach to edge detection," *Readings in Computer Vision*, 1987, pp. 184–203. https://doi.org/10.1016/b978-0-08-051581-6.50024-6.

[15] Fujimoto, T., Kawasaki, T., and Kitamura, K., "Canny Edge Detection/Rankine Hugoniot conditions unified shock sensor for inviscid and viscous flows," *Journal of Computational Physics*, Vol. 396, 2019, pp. 264–279. https://doi.org/10.1016/j.jcp.2019.06.071.

[16] Ray, D., and Hesthaven, J., "An artificial neural network as a troubled-cell indicator," *Journal of Computational Physics*, Vol. 367, 2018, pp. 166–192. https://doi.org/10.1016/j.jcp.2018.04.029.

[17] Morgan, N., Tokareva, S., Liu, X., and Morgan, A., "A machine learning approach for detecting shocks with high-order hydrodynamic methods," *AIAA Scitech 2020 Forum*, 2020. https://doi.org/10.2514/6.2020-2024.

[18] Liu, Y., Lu, Y., Wang, Y., Sun, D., Deng, L., Wang, F., and Lei, Y., "A CNN-based shock detection method in flow visualization," *Computers and Fluids*, Vol. 184, 2019, pp. 1–9. https://doi.org/10.1016/j.compfluid.2019.03.022.

[19] Beck, A., Zeifang, J., Schwarz, A., and Flad, D., "A neural network based shock detection and localization approach for discontinuous Galerkin methods," *Journal of Computational Physics*, Vol. 423, 2020. https://doi.org/10.1016/j.jcp.2020.109824.

[20] Oliveira, M., Lu, P., Liu, X., and Liu, C., "A new shock/discontinuity detector," *International Journal of Computer Mathematics*, Vol. 87, No. 13, 2010, pp. 3063–3078.

[21] Zhao, G.-Y., Sun, M.-B., and Pirozzoli, S., "On shock sensors for Hybrid Compact Weno Schemes," *Computers & Fluids*, Vol. 199, 2020. https://doi.org/10.1016/j.compfluid.2020.104439.

[22] Sobel, I., "Camera models and machine perception," Ph.D. thesis, Stanford, 1970.

[23] Jiang, G.-S., and Shu, C.-W., "Efficient implementation of weighted ENO schemes," *Journal of computational physics*, Vol. 126, No. 1, 1996, pp. 202–228.

[24] Visbal, M., and Gaitonde, D., "Shock Capturing Using Compact Finite Differencing Based Methods," *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005. https://doi.org/10.2514/6.2005-1265.

[25] Visbal, M. R., and Gaitonde, D. V., "On the Use of Higher-Order Finite-Difference Schemes on Curvilinear and Deforming Meshes," *Journal of Computational Physics*, Vol. 181, No. 1, 2002, pp. 155–185.

[26] Sharma, A., and Visbal, M., "Numerical investigation of the effect of airfoil thickness on onset of dynamic stall," *Journal of Fluid Mechanics*, Vol. 870, 2019, p. 870–900. https://doi.org/10.1017/jfm.2019.235.

[27] Monfort, M., Luciani, T., Komperda, J., Ziebart, B., Mashayek, F., and Marai, G., "A Deep Learning Approach to Identifying Shock Locations in Turbulent Combustion Tensor Fields," *Mathematics and Visualization*, 2017, pp. 375–392. https://doi.org/10.1007/978-3-319-61358-1_16.

[28] Morgan, P., Sherer, S., and Visbal, M., "Numerical investigation of supersonic flow over a wall-mounted cylinder," *54th AIAA Aerospace Sciences Meeting*, 2016. https://doi.org/10.2514/6.2016-0046.