

# A New Algorithm for Shock Sensor Calculation at Supersonic Speeds on a 3D-Unstructured Grid

Vahid Tahmasbi<sup>1</sup>, Seyed Mohammad Hossein Karimian<sup>1</sup>

**ABSTRACT:** In this paper, 3-D supersonic flow around two types of wings is solved using a new algorithm for shock sensor calculation. A dual-time-stepping implicit method with 2nd-order accuracy is used for time integration of the equations. In each real time step, the non-linear system of equations is solved by iterating in pseudo-time, using a multi-step integration method. A cell-center finite volume scheme is applied to discretize the solution domain. Governing equations are discretized using 2nd-order central scheme of Jameson. Undesirable oscillations are prevented using artificial dissipation terms containing 2nd and 4th-order derivative terms. The second-order derivative term is proportional to shock sensor, which is a function of pressure gradient in general and is devised to capture shock waves correctly. Appropriate calculation of shock sensor is very important especially for the solution of 3-D supersonic flow on unstructured grids. In this study, a simple efficient algorithm is proposed for shock sensor calculation to stabilize solution in supersonic 3-D flows on unstructured grids. The new algorithm, implemented at an in-house code, is evaluated by comparison of its results with wind tunnel test data and upwind-type differencing scheme of Roe for a tailplane model tested at Royal Aircraft Establishment. The results show that supersonic flow with shock waves has been accurately captured.

**KEYWORDS:** Shock sensor, Supersonic speed, RAE tailplane.

## INTRODUCTION

Accurate solution of fluid flow around supersonic vehicles is one of the aerospace engineering concerns. Among the key factors for this purpose is the correct capture of high-gradient phenomena such as shock waves. In order to prevent non-physical oscillations around the shock waves, some numerical methods add artificial dissipation terms to the system of equations. The most well-known among these is the method of Jameson.

The idea of using non-linear viscosity for the numerical simulation of shock waves was introduced by von Neumann and Richtmyer (1950). Jameson *et al.* (1981) implemented a 3rd-order artificial viscosity and a shock sensor to solve Euler equations in a domain discretized by finite volume method. Jameson and Mavriplis (1986) presented an artificial viscosity that was the combination of 2nd- and 4th-order derivatives. A 2-D structured triangular mesh was used in this paper. At each face of a finite volume, 2 directions were defined. Then, shock sensor was considered proportional to the maximum value of pressure gradients in these 2 directions. This method, however, was not effective in a general 3-D unstructured grid.

Jameson and Baker (1987) developed artificial dissipation terms for 2- and 3-D unstructured grids. In this paper, shock sensor at a node was defined proportional to the pressure gradient at that node. Then, shock sensor on each face of a finite volume was considered to be the maximum one of the 2 neighbor nodes of that face. Results showed that this method was successful for the solution of 2-D flows on unstructured grids. Jameson (1995a,b) further developed previous works and analyzed the effect

1. Amirkabir University of Technology – Aerospace Engineering Department – Tehran – Iran.

Author for correspondence: Vahid Tahmasbi | Amirkabir University of Technology – Aerospace Engineering Department | 424 Hafez Ave | P.O. Box: 15875-4413 – Tehran – Iran | E-mail: vtahmasbi@gmail.com

Received: 02/11/2016 | Accepted: 06/18/2016

of artificial viscosity on accuracy and convergence of solution. Xu *et al.* (1995) essentially applied Jameson method to calculate artificial viscosity in a gas kinetic finite volume method for the solution of Euler equation on structured grids.

Most of the mentioned references and many others (Siclari 1989; Siclari and Del Guidice 1990; Jameson 1986; Volpe *et al.* 1987) in the literature are related to the calculation of artificial dissipation terms on structured grids. Flow solvers using these algorithms capture shock waves correctly in high-speed flow, eliminate oscillations around them, and converge smoothly. However, since these methods use specific directions in space to calculate shock sensor, they cannot be used on unstructured grids. It should be mentioned that artificial dissipation terms have been also introduced in some finite element methods to solve Euler equations in supersonic flows. However, these forms of artificial dissipation terms cannot be used in the context of finite volume cell-centered methods (Barter 2008; Burbeau 2010).

Other than the mentioned literature that discusses the implementation of shock sensor in artificial viscosity, there are other methods that have been introduced for shock capturing in supersonic flows. A quick review of these methods is given next.

Jiang and Shu (1996) presented a new weighted essentially non-oscillatory (WENO) scheme for Cartesian grids. The smoothness indicators of this scheme can be interpreted as a shock sensor (Visbal and Gaitonde 2005). Ducros *et al.* (1999) developed a new shock sensor, which was more accurate in distinguishing shocks from other flow features, like vortices, and more suitable for unsteady flows. Sjögreen and Yee (2004) presented a new numerical dissipation based on the artificial compression method (ACM) of Harten (1978) and a new shock sensor based on wavelet analysis and multi-resolution method of Harten (1995). Cook and Cabot (2004, 2005) presented a new artificial viscosity with spectral-like viscosity, which could damp oscillations near discontinuities without using the shock sensor. Oliveira *et al.* (2010) introduced a new 2-step shock sensor, which was able to detect shocks based on the truncation error ratio on the coarse and fine grids and the local flow gradients at left- and right-hand side finite volumes. Cook *et al.* (2013) extended the Hyperviscosity Method (Cook and Cabot 2005) to unstructured arbitrary Lagrangian-Eulerian (ALE) grids, and Shen *et al.* (2013) presented a new shock sensor, which, unlike many other shock sensors, is parameter-free and needs no user specified constants.

Against central scheme methods, upwind-type differencing schemes, such as flux-vector splitting of van Leer (1982) and the

approximate Riemann solver of Roe (1981), utilize concepts from the method of characteristics in order to determine the direction of spatial differencing to discrete hyperbolic equations. These schemes have the advantage of being naturally dissipative in contrast to central schemes in which separate dissipation terms are added to overcome oscillations arising in regions of strong gradients. Instead, central schemes are less time-consuming in comparison with upwind schemes.

The purpose of this paper, however, was to develop a new method for shock sensor calculation in artificial viscosity of supersonic flow solvers on unstructured grids. Therefore, other methods, including those in the References section, are not intended here.

In the present study, for discretization of the governing equations, finite volume cell-centered based method is applied. Researches carried out by Stolcis and Johnston (1990) indicated that the isotropic behavior of the shock sensor does not increase the effect of 2nd-order artificial dissipation near the discontinuities, but increases the level of the 4th-order artificial dissipation in the zero-gradient areas. This problem has been resolved by definition of non-isotropic shock sensor as a function of special form of pressure gradient on each face of finite volume (Stolcis and Johnston 1990). This shock sensor has made significant improvement in capturing shock waves. Nevertheless, in the present study, we found that the use of this type of shock sensor leads to divergence of solution in supersonic flows on 3D-unstructured grids. Therefore, a new non-isotropic shock sensor and its calculation algorithm are proposed in this paper. In fact, the non-isotropic shock sensor introduced by Stolcis and Johnston (1990) is modified by means of a simple and efficient method to be used in 3-D supersonic flow. Grid information is saved in a face-based data structure which is suitable for the calculation of shock sensor in the algorithm applied in this paper. Comparison of the results obtained in the present algorithm with the experimental data and upwind-type differencing scheme of Roe (1981) illustrates the high accuracy of this algorithm to solve 3-D supersonic flows embedding shock waves.

## EULER EQUATIONS

Euler equations are the limit of the Navier-Stokes equations for vanishing viscosity effects. These equations can be written in the Cartesian coordinate system as:

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = 0 \quad (1)$$

where:  $Q$  is the vector of conserved quantities;  $F$ ,  $G$  and  $H$  represent the convective fluxes in the following form:

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix}, F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho E + p)u \end{bmatrix}, G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (\rho E + p)v \end{bmatrix}, \quad (2)$$

$$H = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (\rho E + p)w \end{bmatrix}$$

where:  $\rho$ ,  $u$ ,  $v$ ,  $w$ ,  $p$  and  $E$  denote density, velocity components, pressure and internal energy, respectively.

Euler equations are augmented by the equation of state, which, for a perfect gas, is given by:

$$p = (\gamma - 1) \left[ \rho E - \frac{\rho(u^2 + v^2 + w^2)}{2} \right] \quad (3)$$

where:  $\gamma$  is the ratio of specific heat.

Having integrated Eq. 1 over control volume  $\Omega$  with boundary  $\partial\Omega$ , the following equation is obtained:

$$\frac{\partial}{\partial t} \int_{\Omega} Q \, dV + \oint_{\partial\Omega} (F \, dS_x + G \, dS_y + H \, dS_z) = 0 \quad (4)$$

where:  $V$  is the cell volume;  $dS_x$ ,  $dS_y$  and  $dS_z$  denote components of the outward vector normal to the surface.

The implementation of Eq. 4 to each finite volume (or cell) in the solution domain results in a set of ordinary differential equations (ODE) in the following form:

$$\frac{d}{dt} (Q_i V_i) + R_i(Q) - D_i(Q) = 0 \quad (5)$$

where:  $R_i(Q)$  and  $D_i(Q)$  denote sum of the convective fluxes and artificial viscosity terms on the surfaces of  $i^{\text{th}}$  finite volume, respectively. Note that the value of  $Q$  used for the calculation of fluxes on the finite volume surface is averaged between its values at the center of finite volume  $i$  and the neighbor finite volume of that surface.

Therefore, on the  $j^{\text{th}}$  surface of  $i^{\text{th}}$  finite volume,  $Q$  would be:

$$Q_j = \frac{Q_i + Q_k}{2} \quad (6)$$

where  $k$  indicates neighbor finite volume of the  $j^{\text{th}}$  surface of  $i^{\text{th}}$  finite volume.

## IMPLICIT TIME MARCHING SCHEME

A dual-time-stepping scheme is employed to solve the non-linear Eq. 5, which is written in the following implicit form:

$$\frac{d}{dt} (Q_i V_i) + R_i(Q^{n+1}) - D_i(Q^{n+1}) = 0 \quad (7)$$

The 2nd-order backward difference scheme is applied to discretize the transient term. This results in the following equation for each finite volume:

$$\frac{3}{2\Delta t} (Q_i^{n+1} V_i^{n+1}) - \frac{2}{\Delta t} (Q_i^n V_i^n) + \frac{1}{2\Delta t} (Q_i^{n-1} V_i^{n-1}) + R_i(Q^{n+1}) - D_i(Q^{n+1}) = 0 \quad (8)$$

where:  $i = 1, N_c$ , being  $N_c$  the total number of finite volumes in the solution domain.

The above system of non-linear coupled equations is solved by iterating it in a pseudo-time, called  $\tau$ , through the following equation:

$$\frac{dQ_i}{d\tau_i} + R_i^*(Q^n) = 0 \quad (9)$$

where  $R_i^*(Q^n)$  is the unsteady residual given by:

$$R_i^*(Q^n) = \frac{3}{2\Delta t} (Q_i^{n+1} V_i^{n+1}) - \frac{2}{\Delta t} (Q_i^n V_i^n) + \frac{1}{2\Delta t} (Q_i^{n-1} V_i^{n-1}) + R_i(Q^{n+1}) - D_i(Q^{n+1}) \quad (10)$$

Having known  $Q^n$ , 4-stage Runge-Kutta scheme is used to solve Eq. 9 for  $Q_i^{n+1}$ , which is first initialized by  $Q_i^{n+1} = Q_i^n$ , and

then the following iteration procedure in pseudo-time, indexed by  $m$ , is continued until the convergence of  $Q_i^{n+1}$  is reached.

$$\begin{aligned}
 Q_i^{(0)} &= (Q_i^{n+1})^m \\
 Q_i^{(1)} &= Q_i^{(0)} - \frac{1}{4} \frac{\Delta\tau_i}{V_i} R_i^*(Q^{(0)}) \\
 Q_i^{(2)} &= Q_i^{(0)} - \frac{1}{3} \frac{\Delta\tau_i}{V_i} R_i^*(Q^{(1)}) \\
 Q_i^{(3)} &= Q_i^{(0)} - \frac{1}{2} \frac{\Delta\tau_i}{V_i} R_i^*(Q^{(2)}) \\
 Q_i^{(4)} &= Q_i^{(0)} - \frac{1}{3} \frac{\Delta\tau_i}{V_i} R_i^*(Q^{(3)}) \\
 (Q_i^{n+1})^{m+1} &= Q_i^{(4)}
 \end{aligned} \tag{11}$$

In order to minimize computational time, numerical dissipation term  $D_i$  is calculated only at the first and third stages of Eq. 11. The real-time step is not restricted in the above scheme, but the pseudo-time step is restricted by the following stability criterion at each finite volume:

$$\Delta\tau = \min \left[ \frac{CFL_i V_i}{\sum_{j=1}^{Nf_i} \lambda_j}, \frac{2\Delta t}{3} \right] \tag{12}$$

where:  $CFL_i$  is the Courant number for pseudo-time marching;  $j$  denotes the face of the  $i^{th}$  finite volume;  $Nf_i$  is the number of finite volume faces, and

$$\lambda = u \, dS_x + v \, dS_y + w \, dS_z \tag{13}$$

## BOUNDARY CONDITIONS

At the far field, non-reflecting boundary conditions were applied based on the characteristic analysis. On the solid wall boundary, the slip condition needs to be imposed. Pressure value at the solid wall is obtained by extrapolating from their values of adjacent cells.

## ARTIFICIAL DISSIPATION

In this paper, artificial dissipation term is a blend of 2nd- and 4th-order differences of flow variables  $Q$  with coefficients that depend on the local pressure gradient to prevent the

appearance of high-frequency modes corresponding to odd and even point oscillations and oscillations near the shock waves (Jameson and Mavriplis 1986). For the  $i^{th}$  finite volume, artificial dissipation term is defined as:

$$\begin{aligned}
 D_i(Q) &= \sum_{j=1}^{Nf_i} d_{ij}^{(2)} + \sum_{j=1}^{Nf_i} d_{ij}^{(4)} \\
 d_{ij}^{(2)} &= \frac{\varepsilon_{ij}^2}{2} \lambda_{ij} (Q_i - Q_k) \\
 d_{ij}^{(4)} &= \frac{\varepsilon_{ij}^4}{2} \lambda_{ij} (\nabla^2 Q_i - \nabla^2 Q_k)
 \end{aligned} \tag{14}$$

where: subscripts  $ij$  denote  $j^{th}$  surface of  $i^{th}$  finite volume;  $k$  denotes the neighbor finite volume across the  $j^{th}$  surface of  $i^{th}$  finite volume;  $\varepsilon_{ij}^{(2)}$  and  $\varepsilon_{ij}^{(4)}$  are adaptive coefficients defined as:

$$\begin{aligned}
 \varepsilon_{ij}^{(2)} &= k^{(2)} v_{ij} \\
 \varepsilon_{ij}^{(4)} &= \max \left[ 0, (k^{(4)} - \varepsilon_{ij}^{(2)}) \right]
 \end{aligned} \tag{15}$$

where:  $k^{(2)}$  and  $k^{(4)}$  are constants in the range of  $(0.5 < k^{(2)} < 1)$  and  $(1/256 < k^{(4)} < 1/32)$ , respectively;  $v_{ij}$  is the shock sensor which will be defined later.

The Laplacian operator  $\nabla^2$  and the scaling factor  $\lambda_{ij}$  are defined as:

$$\begin{aligned}
 \nabla^2 Q_i &= \sum_{j=1}^{Nf_i} (Q_k - Q_i) \\
 \lambda_{ij} &= \left| u_{ij} \, dS_x + v_{ij} \, dS_y + w_{ij} \, dS_z \right| + \\
 &\quad + c_{ij} \sqrt{(dS_x)^2 + (dS_y)^2 + (dS_z)^2}
 \end{aligned} \tag{16}$$

where:  $c_{ij}$  is the local speed of sound;  $\lambda_{ij}$  is the scaling factor calculated from the following equation:

$$\lambda_{ij} = \left( \frac{V_i}{\Delta\tau_i} + \frac{V_k}{\Delta\tau_k} \right) \tag{17}$$

The isotropic shock sensor  $v_{ij}$  is a pressure-sensitive sensor usually defined as:

$$v_{ij} = \max(v_i, v_k) \tag{18}$$

where:

$$v_i = \frac{\left| \sum_{j=1}^{Nf_i} (P_k - P_i)_j \right|}{\left| \sum_{j=1}^{Nf_i} (P_k + P_i)_j \right|} \quad (19)$$

The term  $v_k$  can be calculated in a similar manner. A non-isotropic shock sensor was defined for each face as (Stolcis and Johnston 1990):

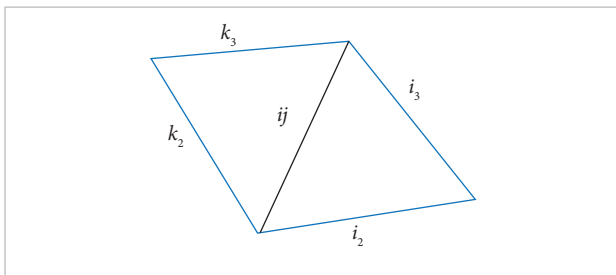
$$v_{ij} = \left| \frac{P_i - P_k}{P_i + P_k} \right| \quad (20)$$

Flow solvers implementing this shock sensor have been successful in shock wave capturing. On unstructured grids, however, they do not perform well. To resolve this shortcoming, a new algorithm is proposed to calculate shock sensor in the present paper. This algorithm includes the following 3 steps:

- Step 1: shock sensor  $v_{ij}$  is calculated on all of the finite volume surfaces within the domain using Eq. 20. The shock sensor calculated in this step is denoted by  $N = 0$ .
- Step 2: shock sensor  $v_{ij}$  on each finite volume surface is substituted by the maximum value of  $v_{ij}$  with  $N = 0$  on this face and on the other faces of 2 finite volumes neighbor to this face. This is done for every finite volume surface within the domain. The shock sensor calculated in this step is denoted by  $N = 1$ .
- Step 3: if step 2 is repeated for  $v_{ij}$  with  $N = 1$ , then a set of new  $v_{ij}$  will be obtained, which we denote by  $N = 2$ . Shock sensors with higher  $N$  can be obtained in a similar manner.

For the sake of clarity, these 3 steps are introduced on a 2-D unstructured grid shown in Fig. 1. Assume that step 1 is done and we have  $v_{ij}$  with  $N = 0$  on all of the finite volume surfaces in the solution domain. Now  $v_{ij}$  with  $N = 1$  would be:

$$v_{ij} = \max(v_{ij}, v_{i2}, v_{i3}, v_{k2}, v_{k3}) \quad (21)$$



**Figure 1.** Edges involved in calculation of shock sensor related to edge  $ij$ .

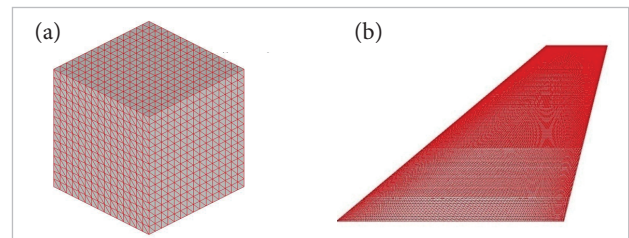
Now we have  $v_{ij}$  with  $N = 1$  on all finite volume surfaces within the domain. As mentioned before, if Eq. 21 is implemented for these new  $v_{ij}$ , one can obtain  $v_{ij}$  with  $N = 2$  on all finite volume surfaces. Note that the method offered by Stolcis and Johnston (1990) is in fact the first step of the present algorithm. In this paper, the following test cases are solved using our in-house code in which the present algorithm is implemented. In addition, the minimum value of  $N$  leading to the convergence of solution, denoted by  $N_{min}$ , is investigated in each case.

## NUMERICAL RESULTS

In this section, supersonic flow is simulated around Royal Aircraft Establishment (RAE) tailplane (Mabey *et al.* 1984) and a rectangular wing. The RAE tailplane has panel aspect ratio of 1.2, taper ratio of 0.27, leading edge sweep angle of  $50.2^\circ$  and approximately an airfoil of NACA 64A010.2. For the tailplane wing, lengths are non-dimensionalized with the chord of root. The rectangular wing has panel aspect ratio of 2 and airfoil section of NACA 64A010.2. For this configuration, lengths are non-dimensionalized with its airfoil chord. Initial conditions of the stationary fluid are  $\rho = 1 \text{ kg}\cdot\text{m}^{-3}$ ,  $P = 1 \text{ bar}$  and  $T = 300 \text{ K}$ .

A tetrahedral mesh is generated around the wings without any additional refinement near the shock. Both wings are located in the middle of the solution domain. The outer boundary of the computational domain, shown in Fig. 2a, is a cube with dimensions of  $40 \times 40 \times 40$  root chords on which regular triangular surface mesh is generated. A regular triangular mesh is generated on the wing surface as well. For the RAE tailplane, this grid is shown in Fig. 2b. Grids generated around RAE tailplane and rectangular wing have total finite volumes of 483,245 and 117,030, respectively.

In this paper, grid properties are stored using face-based data structure. In both test cases, the same values of Courant number



**Figure 2.** Surface grids on the boundaries of computational domain for RAE tailplane.

in pseudo-time ( $CFL_p$ ), Courant number in real time ( $CFL$ ), convergence criteria to stop iterations in pseudo-time ( $ER_p$ ), and convergence criteria to stop iterations in real time ( $ER$ ) are used. These values are given in Table 1.

**Table 1.** Details of solution parameters.

$CFL_p$	$CFL$	$ER_p$	$ER$
1.0	1.E+6	1.E-3	1.E-7

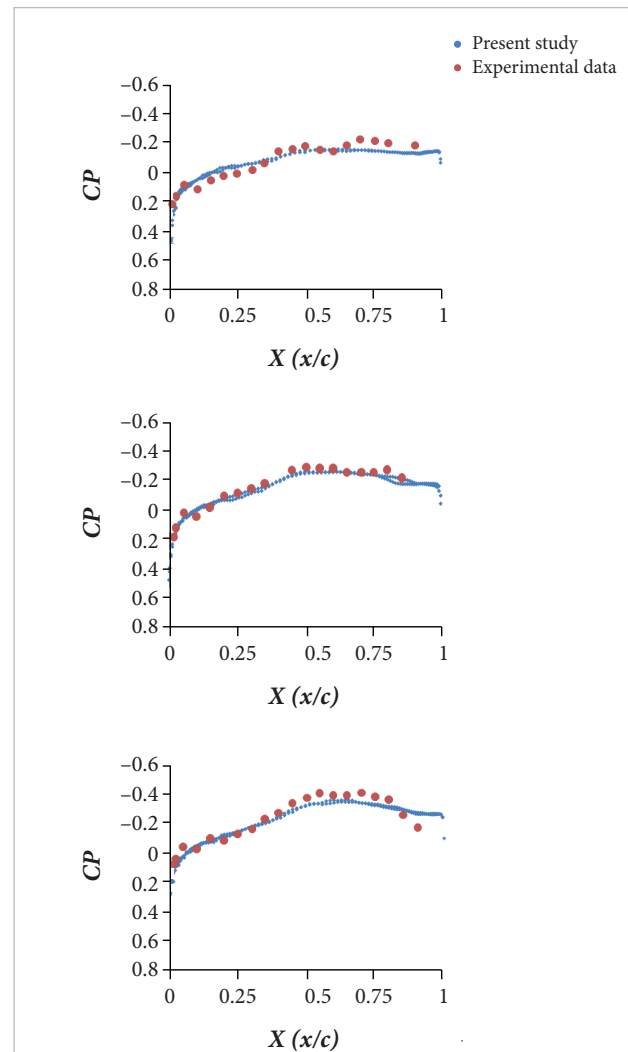
Supersonic flow around RAE tailplane is simulated under two different conditions: at free stream Mach number ( $M$ ) of 1.2 and  $0^\circ$  angle of attack (AOA) as the first case and at free stream  $M$  of 1.71 and  $0.14^\circ$  AOA as the second case to validate the present algorithm. For the first test case, after 4,045 iterations, solution error defined as  $\epsilon_r = \sum_{i=1}^{N_c} (\rho_i - \bar{\rho}_i) / \rho_i$  was reduced to less than  $ER$ , defined in Table 1 to obtain the converged solution. Note that  $N_c$  is the total number of cells and  $\bar{\rho}_i$  is the density at the previous iteration. The minimum number of pressure-gradient repetition required for the calculation of shock sensor,  $N_{min}$ , was 3 and 5 for the first and second test cases, respectively. For the first test case, pressure-coefficient distribution along the chord at semi-span sections of  $\eta$  equal to 14, 42, and 65% from the root is shown in Figs. 3. Having considered that the present solution is inviscid and viscous effects are not included, the agreement observed between the present results and the experimental data (Mabey *et al.* 1984) is very good. At position  $\eta = 65\%$  near the tip of the wing, however, some deviation from the experiment is seen near the trailing edge. This may be related to 3-D apex and tip effects that are not predicted very well by the present method at  $M = 1.2$ .

For the second test case, the solution was converged after 3,116 iterations. Similarly, pressure results of the second test case are shown in Figs. 4. Again at each semi-span section pressure-coefficient distribution along the chord is compared with results of the experiment (Mabey *et al.* 1984). The present results are in excellent agreement with experimental data. Only small differences are observed between these results in both cases, which is not surprising, since we are solving inviscid flow here. Therefore, one can conclude that the present algorithm works quite well on unstructured grids.

As seen,  $N_{min}$  increases with Mach number. To investigate this variation, the tailplane case was simulated for higher Mach number of 2.0. The solution was converged after 13,720 iterations, and the results of this case were obtained with  $N_{min} = 7$ . As seen,  $N_{min}$  has been increased again with Mach

number. Note that pressures are non-dimensionalized by free-stream pressure.

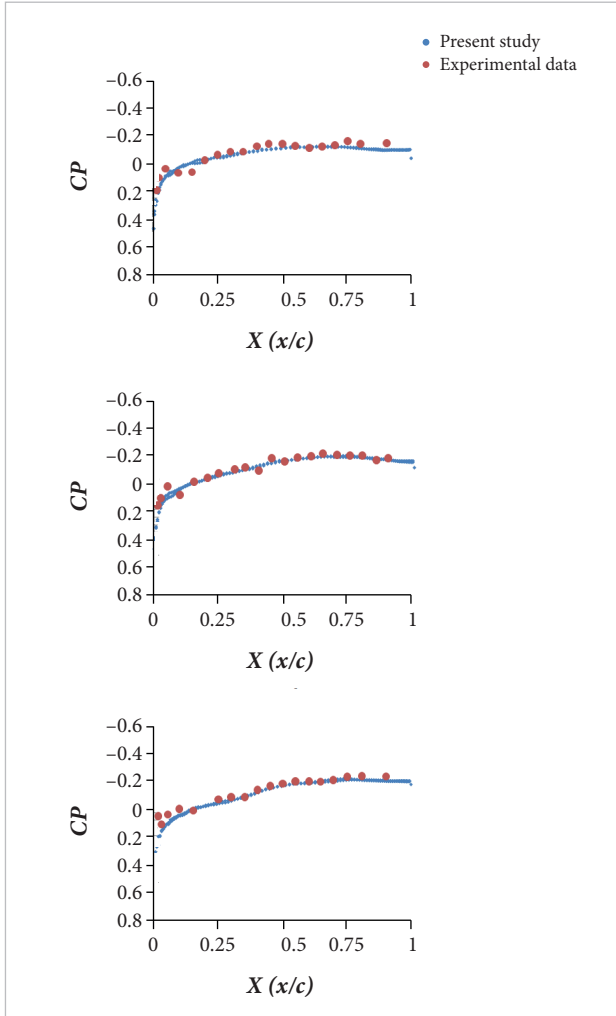
Following these results, it would be interesting to see if  $N_{min}$  increases with Mach number in other configurations, such as rectangular wing. Therefore, supersonic flow at Mach numbers of 1.2, 2.0, and 2.4 over a rectangular wing at  $0^\circ$  AOA is solved using the present algorithm. In these cases, solutions were converged after 3,875, 2,623, and 2,024 iterations, respectively. As a sample of results, the distribution of pressure coefficient along the chord in the mid-section of this wing at  $M = 2.0$ , compared with upwind scheme of Roe (1981) for identical grids, is illustrated in Fig. 5. As shown in this figure, a good agreement has been achieved between



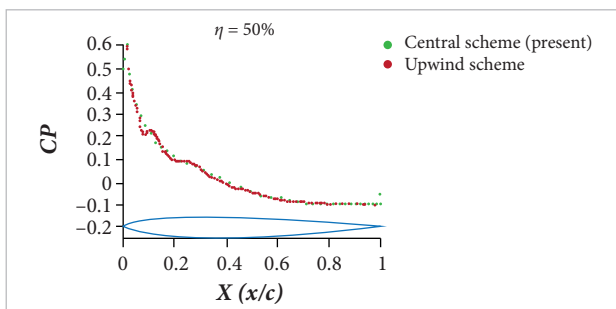
**Figure 3.** RAE tailplane at  $M = 1.2$  and  $AOA = 0^\circ$ . Comparison of pressure coefficient with experimental data (a)  $\eta = 14\%$ , (b)  $\eta = 42\%$ , (c)  $\eta = 65\%$ .



the upwind scheme of Roe (1981) and the present scheme.  $N_{min}$ , which led to the convergence of solution for Mach numbers of 1.2, 2.0, and 2.4, is given in Table 2. One more time, its



**Figure 4.** RAE tailplane at  $M = 1.71$  and  $AOA = 0.14^\circ$ . Comparison of pressure coefficient with experimental data (a)  $\eta = 14\%$ , (b)  $\eta = 42\%$ , (c)  $\eta = 65\%$ .



**Figure 5.** Pressure coefficient along the chord in the middle of rectangular wing at  $M = 2.0$  and  $AOA = 0^\circ$ .

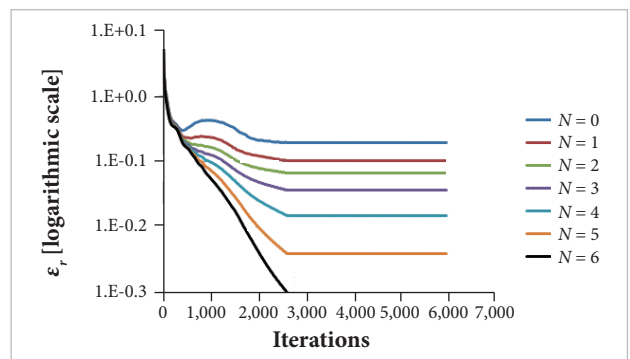
values have increased with Mach number. The increase in  $N_{min}$  should be due to the addition of non-linearities and flow gradients with Mach number in flow regions, such as the shock waves.

**Table 2.** Variation of  $N_{min}$  with Mach for the rectangular wing.

Mach Number	1.2	2	2.4
$N_{min}$	0	6	10

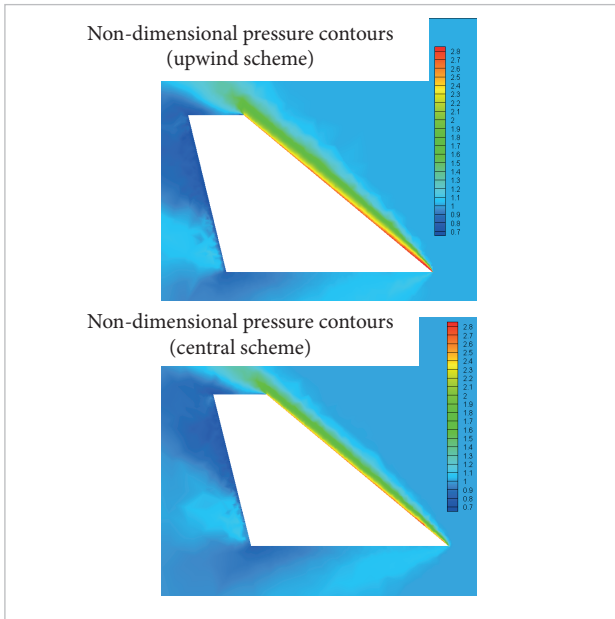
The effect of  $N$  on the solution convergence is investigated as well. For this purpose, convergence rate of solution for rectangular wing at  $M = 2.0$  and  $AOA = 0^\circ$  is plotted versus the iteration in one real-time step (Fig. 6). Because of the implicit treatment, the solution is obtained in one large real-time step, and the error is plotted versus pseudo-time iteration in this figure. According to Table 1, the minimum error to stop iterations in pseudo-time is  $10^{-3}$ . As seen in Fig. 6, the solution convergence is achieved for  $N$  not less than 6.0. It is quite interesting that, regardless of the  $N$  value, the minimum error is achieved after about 2,600 pseudo-iterations. One should know that the cost of computation in each iteration increases with  $N$ . In addition, the results show that the error reduction becomes more monotonic at higher  $N$  values.

Obviously, a 3-D oblique shock wave will be formed in the front of RAE tailplane in  $M = 2.0$  and  $AOA = 0^\circ$ . It is not an easy task to depict this shock wave, since it is both oblique and 3-D. The strongest segment of this shock wave can be seen in a plane parallel to the free stream flow. As shown in Figs. 7 and 8, non-dimensional pressure and Mach contours on this plane obtained from the present method are compared with the results of upwind scheme of Roe (1981). Note that both results are obtained on the same grid. Contours of both methods confirm formation of oblique shock wave in the

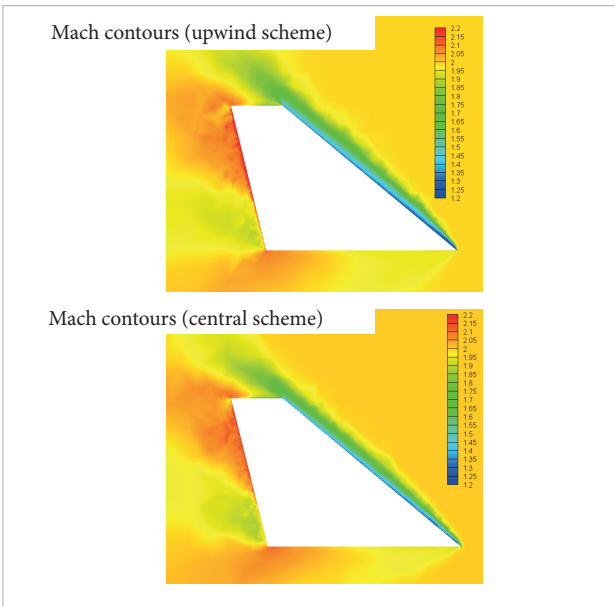


**Figure 6.** Effect of  $N$  on the rate of convergence with iteration at the first real-time step for rectangular wing at  $M = 2.0$  and  $AOA = 0^\circ$ .

leading edge region. Since flow behind the oblique shock slips in the spanwise direction along the leading edge, pressure and Mach do not reach their values at stagnation point. The results presented in these figures show that the oblique shock wave is captured thinner and in a location closer to the leading edge in the present method. This observation is quantitatively

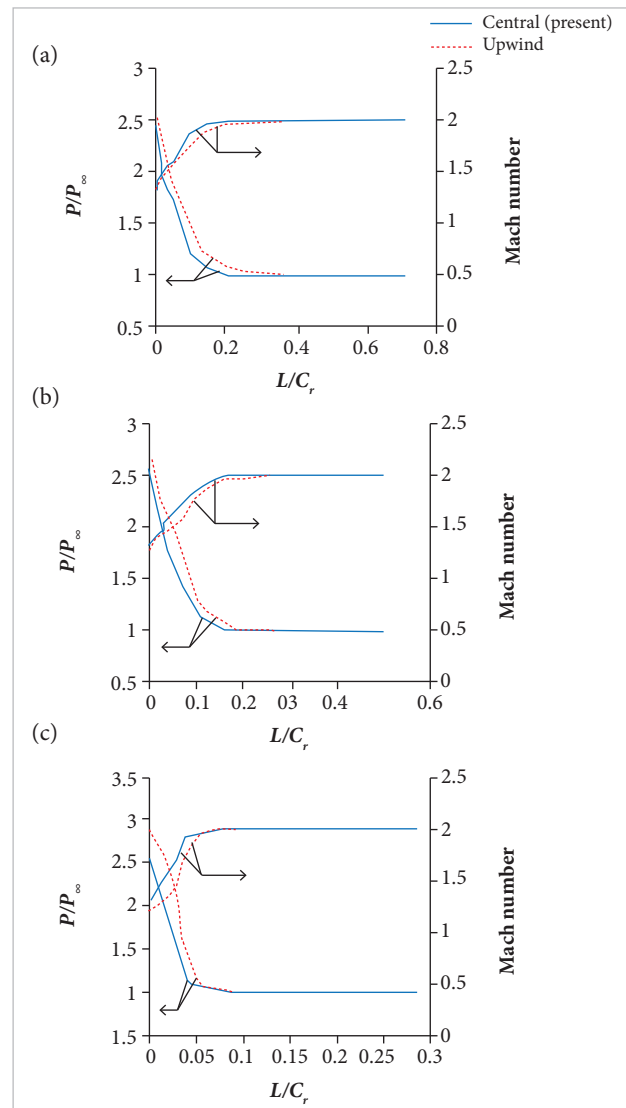


**Figure 7.** Non-dimensional pressure contours on a plane parallel to the free stream flow for the RAE tailplane at  $M = 2.0$  and  $AOA = 0^\circ$ .



**Figure 8.** Mach contours on a plane parallel to the free stream flow for the RAE tailplane at  $M = 2.0$  and  $AOA = 0^\circ$ .

demonstrated in Figs. 9, in which non-dimensional pressure and Mach number variations are plotted along a line parallel to the free stream from the leading edge ( $L/C_r = 0$ ) towards the upstream at different semi span positions.  $L$  is the distance from leading towards the upstream and  $C_r$  is the wing chord at its root. As seen in these figures, the location where gradients of pressure and Mach start is closer to the leading edge in the present method. The difference between the results of the present method and those of the Roe (1981) method is due to the order of methods in space. The present method is 2nd-order in space, but Roe (1981) method is 1st-order in space.



**Figure 9.** RAE tailplane at  $M = 2.0$  and  $AOA = 0^\circ$ . Non-dimensional pressure and Mach number variations along a line parallel to the free stream from the leading edge ( $L/C_r = 0$ ) towards the upstream. (a)  $\eta = 90\%$ , (b)  $\eta = 50\%$ , (c)  $\eta = 10\%$ .



## CONCLUSION

In this paper, 3-D supersonic flow around 2 types of wings is solved on unstructured grids using a new algorithm for shock sensor calculation. This shock sensor, which is based on repetitious calculation of pressure-gradient, is very effective in capturing shock waves without oscillations, leading to a stable and converged solution. The present algorithm is a modified version of the method introduced earlier in the literature. Face-based data structure is used here, since it is compatible with the data required for the shock sensor calculation. To validate the present algorithm, its results are compared with the experimental data and the results of Roe (1981) upwind

scheme. A comparison shows that the present algorithm predicts shock waves and the supersonic flow fields accurately. Analysis of the results indicates that the minimum number of pressure-gradient repetition required for the calculation of shock sensor in the present algorithm increases with Mach number at least for the 2 types of wings tested here.

## AUTHOR'S CONTRIBUTION

Tahmasbi V performed the simulation and Karimian SMH wrote the main paper. Both authors discussed the results and implications and commented on the manuscript at all stages.

## REFERENCES

- Barter GE (2008) Shock capturing with PDE-based artificial viscosity for an adaptive, higher-order discontinuous Galerkin finite element method (PhD thesis). Cambridge: Massachusetts Institute of Technology.
- Burbeau A (2010) A node-centered artificial viscosity method for two-dimensional Lagrangian hydrodynamics calculations on a staggered grid. *Comm Computat Phys* 8(4):877-900. doi: 10.4208/cicp.030709.161209a
- Cook AW, Cabot WH (2004) A high-wavenumber viscosity for high-resolution numerical methods. *J Comput Phys* 195(2):594-601. doi: 10.1016/j.jcp.2003.10.012
- Cook AW, Cabot WH (2005) Hyperviscosity for shock-turbulence interactions. *J Comput Phys* 203(2):379-385. doi: 10.1016/j.jcp.2004.09.011
- Cook AW, Ulitsky M, Miller D (2013) Hyperviscosity for unstructured ALE meshes. *Int J Comput Fluid Dynam* 27(1):32-50. doi: 10.1080/10618562.2012.756477
- Ducros F, Ferrand V, Nicoud F, Weber C, Darracq D, Gacherieu C, Poinot T (1999) Large-eddy simulation of the shock/turbulence interaction. *J Comput Phys* 152(2):517-549. doi: 10.1006/jcph.1999.6238
- Harten A (1978) The artificial compression method for computation of shocks and contact discontinuities. III. Self-adjusting hybrid schemes. *Math Comput* 32(142):363-389. doi: 10.2307/2006149
- Harten A (1995) Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Comm Pure Appl Math* 48(12):1305-1342. doi: 10.1002/cpa.3160481201
- Jameson A (1986) A vertex based multigrid algorithm for three-dimensional compressible flow calculations. Proceedings of the ASME Symposium on Numerical Methods for Compressible Flow; Anaheim, USA.
- Jameson A (1995a) Analysis and design of numerical schemes for gas dynamics. 1: artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *Int J Comput Fluid Dynam* 4(3-4):171-218. doi:10.1080/10618569508904524
- Jameson A (1995b) Analysis and design of numerical schemes for gas dynamics. 2: artificial diffusion and discrete shock structure. *Int J Comput Fluid Dynam* 5(1-2):1-38. doi: 10.1080/10618569508940734
- Jameson A, Baker T (1987) Improvements to the aircraft Euler method. Proceedings of the 25th AIAA Aerospace Sciences Meeting; Reno, USA.
- Jameson A, Mavriplis D (1986) Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh. *AIAA J* 24(4):611-618. doi: 10.2514/3.9315
- Jameson A, Schmidt W, Turkel E (1981) Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. Proceedings of the 14th Fluid and Plasma Dynamics Conference; Palo Alto, USA.
- Jiang G, Shu C (1996) Efficient implementation of weighted ENO schemes. *J Comput Phys* 1(126):202-228. doi:10.1006/jcph.1996.0130
- Mabey DG, Welsh BL, Cripps BE (1984) Measurements of steady and oscillatory pressures on a low aspect ratio model at subsonic and supersonic speeds. Bedford: British Royal Aerospace Establishment. Report No.: TR-84095.
- Oliveira M, Lu P, Liu X, Liu C (2010) A new shock/discontinuity detector. *Int J Comput Math* 87(13):3063-3078. doi: 10.1080/00207160902919284
- Roe PL (1981) Approximate Riemann solvers, parameter vectors, and difference schemes. *J Comput Phys* 43(2):357-372. doi: 10.1016/0021-9991(81)90128-5
- Shen Y, Cui K, Yang G, Zha G (2013) Parameter-free shock detector and high order hybrid algorithm for shock/complex flowfield interaction. Proceedings of the 51st AIAA Aerospace Sciences Meeting; Texas, USA.
- Siclari MJ (1989) Three-dimensional hybrid finite volume solutions to the Euler equations for supersonic-hypersonic aircraft. Proceedings of the 25th AIAA Aerospace Sciences Meeting; Reno, USA.

- Siclari MJ, Del Guidice P (1990) Hybrid finite volume approach to Euler solutions for supersonic flows. *AIAA J* 28(1):66-74. doi: 10.2514/3.10354
- Sjögreen B, Yee HC (2004) Multiresolution wavelet based adaptive numerical dissipation control for high order methods. *J Sci Comput* 20(2):211-255. doi:10.1023/B:JOMR.0000008721.30071.e4
- Stolcis L, Johnston L (1990) Solution of the Euler equations on unstructured grids for two dimensional compressible flows. *Aeronaut J* 94(930):181-195.
- Van Leer B (1982) Flux vector splitting for the Euler Equations. In: Krause E, editor. *Eighth International Conference on Numerical Methods in Fluid Dynamics: proceedings of the Conference*, Rheinisch-Westfälische Technische Hochschule, Aachen, Germany, June 28-July 2, 1982. Berlin, New York: Springer-Verlag. p. 507-512. (Lecture Notes in Physics, 170).
- Visbal MR, Gaitonde DV (2005) Shock capturing using compact-differencing-based methods. *Proceedings of the 43rd AIAA Aerospace Sciences Meeting*; Reno, USA.
- Volpe G, Siclari MJ, Jameson A (1987) A new multigrid Euler method for fighter-type configurations. *Proceedings of the 8th AIAA Computational Fluid Dynamics Conference*; Honolulu, USA.
- Von Neumann J, Richtmyer RD (1950) A method for the numerical calculation of hydrodynamic shocks. *J Appl Phys* 21(3):232-237. doi: 10.1063/1.1699639
- Xu K, Martinelli L, Jameson A (1995) Gas-kinetic finite volume methods, flux-vector splitting and artificial diffusion. *J Comput Phys* 120(1):48-65. doi: 10.1006/jcph.1995.1148