

Development of a Three-Dimensional Unstructured Euler Solver for High-Speed Flows

Tudorel Petronel AFILIPOAE^{*1}, Marius STOIA-DJESKA²

*Corresponding author

^{*1}INCAS – National Institute for Aerospace Research “Elie Carafoli”
B-dul Iuliu Maniu 220, Bucharest 061126, Romania
afilipoae.tudorel@incas.ro

²“POLITEHNICA” University of Bucharest, Faculty of Aerospace Engineering
Polizu no.1-6, RO-011061, Bucharest, Romania
marius.stoia@gmail.com

DOI: 10.13111/2066-8201.2015.7.4.1

Received: 13 October 2015 / Accepted: 02 November 2015

Copyright©2015 Published by INCAS. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

The 36th “Caius Iacob” Conference on Fluid Mechanics and its Technical Applications
29 - 30 October, 2015, Bucharest, Romania, (held at INCAS, B-dul Iuliu Maniu 220, sector 6)
Section 2. Numerical Analysis

Abstract: This paper addresses the solution of the compressible Euler equations on hexahedral meshes for supersonic and hypersonic flows. Spatial discretization is accomplished by a cell-centered finite-volume formulation which employs two different upwind schemes for the computation of convective fluxes. Second-order solutions are attained through a linear state reconstruction technique that yields highly resolved flows in smooth regions while providing a sharp and clean resolution of shocks. The solution gradients required for the higher-order spatial discretization are estimated by a least-square method while Venkatakrishnan limiter is employed to preserve monotonicity and avoid oscillations in the presence of shocks. Furthermore, solutions are advanced in time by an explicit third-order Runge-Kutta scheme and convergence to steady state is accelerated using implicit residual smoothing. Flow around a circular arc in a channel and flow past a circular cylinder are studied and results are presented for various Mach numbers together with comparisons to theoretical and experimental data where possible.

Key Words: Euler equations, upwind scheme, Roe, Roe-HLL, second order scheme, linear state reconstruction, Venkatakrishnan limiter, explicit time integration.

1. INTRODUCTION

Aeronautical progress from the first half of the XXth century introduced the necessity of reliable numerical tools for investigating high Reynolds number compressible flows. For this reason, attention was directed towards the most general mathematical models that governs fluid flows such as Euler and Navier-Stokes equations. It is a well-known fact that, even until present days, theoretical solutions of these equations are still lacking in the literature except for some simple one-dimensional flows without practical interest on a larger scale.

Historically, numerical integration of Euler equations was stated for the first time in the early 1950's and it is related to the work of Courant ([16]). Throughout the years, starting

with the work of P. Lax and B. Wendroff ([14], [15]), various algorithms were developed for numerical analysis of compressible flows on both structured and unstructured grids. The necessity of using unstructured grids arises from both the flow domain geometrical complexity and the obvious motivation for obtaining more accurate solutions with the same computational effort as in structured grids.

Regarding spatial discretization, the most common methods for Euler equations are finite difference method and finite volume method. In the present work, attention is focused on a cell-centered finite volume formulation which employs Roe and Roe-HLL upwind schemes for the convective flux computation.

2. GOVERNING EQUATIONS

Formally, the time-dependent Euler equations express the conservation of mass, momentum and energy for a compressible fluid without viscosity. With the nonconductive adiabatic ideal gas assumption and in the absence of external forces, these equations are given below in integral form for a domain Ω bounded by surface $\partial\Omega$

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{W} d\Omega + \int_{\partial\Omega} \vec{\mathbf{F}}(\mathbf{W}) \cdot \vec{\mathbf{n}} dS = 0 \quad (1)$$

where

$$\mathbf{W} = [\rho \quad \rho u \quad \rho v \quad \rho w \quad \rho E]^T \quad (2)$$

and

$$\begin{aligned} \mathbf{F}_x &= [\rho u \quad \rho u^2 + p \quad \rho uv \quad \rho uw \quad \rho uH]^T \\ \mathbf{F}_y &= [\rho v \quad \rho uv \quad \rho v^2 + p \quad \rho vw \quad \rho vH]^T \\ \mathbf{F}_z &= [\rho w \quad \rho uw \quad \rho vw \quad \rho w^2 + p \quad \rho wH]^T \end{aligned} \quad (3)$$

Here, $\vec{\mathbf{n}}$ is the exterior unit normal on the boundary $\partial\Omega$. Velocity components on x, y and z are designated as u, v and w , respectively, while the term ρE represents the total energy per unit volume. With the ideal gas assumption, the static pressure and total enthalpy are expressed as

$$p = \rho(\gamma - 1) \left(E - \frac{V^2}{2} \right) \quad (4)$$

and

$$H = \frac{\gamma}{\gamma - 1} \frac{p}{\rho} + \frac{V^2}{2} \quad (5)$$

where V and γ represent the velocity vector magnitude and the ratio of specific heats, respectively.

For air, γ is prescribed as 1.4. Practically, equation (1) expresses a relationship where the rate of change of the state variables in the volume Ω is compensated by the total flux which passes across its boundary.

3. SPATIAL DISCRETIZATION

A cell-centered finite volume formulation is accomplished by dividing the domain Ω into a finite number of sub-domains Ω_i , suggestively called finite volumes, on which equation (1) is applied. Defining a vector $\bar{\mathbf{W}}_i$ of volume-averaged state variables,

$$\bar{\mathbf{W}}_i = \frac{1}{\Omega_i} \int_{\Omega_i} \mathbf{W} d\Omega \quad (6)$$

a semi-discrete form of equation (1) reads [4]:

$$\Omega_i \frac{\partial \bar{\mathbf{W}}_i}{\partial t} + \mathbf{R}_i = 0, \quad i = 1, 2, 3 \dots NVF \quad (7)$$

In equation (7) NVF is the total number of finite volumes in the computational domain and the term \mathbf{R}_i represents the summation of the numerical fluxes over the total number of interfaces which bounds the sub-domain Ω_i ,

$$\mathbf{R}_i = \sum_{j=1}^{NF} \mathbf{F}_j(\bar{\mathbf{W}}_L, \bar{\mathbf{W}}_R) S_j \quad (8)$$

where $\bar{\mathbf{W}}_L, \bar{\mathbf{W}}_R$ and S_j are the state variables to the left and right side of the interface j and the surface of the interface j , respectively.

Next, we will examine the procedure for the computation of the numerical flux \mathbf{F}_j through an interface j of the control volume Ω_i .

Consequently, the flux across each cell face j is computed using Roe's approximate Riemann solver,

$$\mathbf{F}_j = \frac{1}{2} (\mathbf{F}_{n|R} + \mathbf{F}_{n|L}) - \frac{1}{2} \bar{\mathbf{K}} (\bar{\mathbf{W}}_R - \bar{\mathbf{W}}_L) \quad (9)$$

where \mathbf{F}_n is expressed as

$$\mathbf{F}_n = \bar{\mathbf{F}} \cdot \bar{\mathbf{n}} = (\bar{\mathbf{V}} \cdot \bar{\mathbf{n}}) \left\{ \begin{array}{c} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E + p \end{array} \right\} + p \left\{ \begin{array}{c} 0 \\ n_x \\ n_y \\ n_z \\ 0 \end{array} \right\} \quad (10)$$

The matrix $\bar{\mathbf{K}}$ is computed by evaluating the Jacobian

$$\bar{\mathbf{K}} = \frac{\partial \mathbf{F}}{\partial \mathbf{W}} \quad (11)$$

with Roe-averaged quantities such as

$$\begin{aligned}
\bar{\rho} &= \sqrt{\rho_R \rho_L} \\
\bar{u}_j &= \frac{(u_j \sqrt{\rho})_R + (u_j \sqrt{\rho})_L}{\sqrt{\rho_R} + \sqrt{\rho_L}}, \quad u_{j=1,2,3} = u, v, w \\
\bar{H} &= \frac{(H \sqrt{\rho})_R + (H \sqrt{\rho})_L}{\sqrt{\rho_R} + \sqrt{\rho_L}} \\
\bar{c}^2 &= (\gamma - 1) \left[\bar{H} - \frac{1}{2} \bar{V}^2 \right]
\end{aligned} \tag{12}$$

In practice, the formulation presented in equation (9) turned out to be computationally expensive due to the explicit computation of the matrix $\bar{\mathbf{K}}$. For present computations, an equivalent formulation of the equation (9) is implemented ([2]), i.e.

$$\mathbf{F}_j = \frac{1}{2} (\mathbf{F}_{n|R} + \mathbf{F}_{n|L}) - \frac{1}{2} \sum_{k=1}^5 |\bar{\lambda}_k| \bar{\omega}_k \bar{\mathbf{r}}_k \tag{13}$$

where $\bar{\lambda}_k$, $\bar{\omega}_k$ and $\bar{\mathbf{r}}_k$ represent wave speeds, wave strengths and right eigenvectors of the Jacobian matrix, respectively. These quantities are computed with Roe-averaged values defined earlier and their expressions are given below.

$$\bar{\lambda}_{k=1,5} = [\bar{q}_n - \bar{c} \bar{q}_n \bar{q}_n + \bar{c} \bar{q}_n \bar{q}_n]^T, \quad q_n = \bar{\mathbf{V}} \cdot \bar{\mathbf{n}} \tag{14}$$

$$\bar{\omega}_{k=1,5} = \begin{Bmatrix} \frac{\delta p - \bar{\rho} \bar{c} \delta q_n}{2 \bar{c}^2} \\ \delta p - \frac{\delta p}{\bar{c}^2} \\ \frac{\delta p + \bar{\rho} \bar{c} \delta q_n}{2 \bar{c}^2} \\ \bar{\rho} \delta q_l \\ \bar{\rho} \delta q_m \end{Bmatrix} \tag{15}$$

$$\bar{\mathbf{r}}_{k=1,5} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ \bar{u} - \bar{c} n_x & \bar{u} & \bar{u} + \bar{c} n_x & l_x & m_x \\ \bar{v} - \bar{c} n_y & \bar{v} & \bar{v} + \bar{c} n_y & l_y & m_y \\ \bar{w} - \bar{c} n_z & \bar{w} & \bar{w} + \bar{c} n_z & l_z & m_z \\ \bar{H} - \bar{c} \bar{q}_n & \bar{V}^2/2 & \bar{H} + \bar{c} \bar{q}_n & \bar{q}_l & \bar{q}_m \end{bmatrix} \tag{16}$$

The vectors $\bar{\mathbf{l}}$ and $\bar{\mathbf{m}}$ are defined such as to form with the normal $\bar{\mathbf{n}}$ an orthonormal basis. In the above expressions the following notations have been used:

$$\begin{aligned}
\delta q_n &= (\bar{\mathbf{V}} \cdot \bar{\mathbf{n}})_R - (\bar{\mathbf{V}} \cdot \bar{\mathbf{n}})_L \\
\delta q_l &= (\bar{\mathbf{V}} \cdot \bar{\mathbf{l}})_R - (\bar{\mathbf{V}} \cdot \bar{\mathbf{l}})_L \\
\delta q_m &= (\bar{\mathbf{V}} \cdot \bar{\mathbf{m}})_R - (\bar{\mathbf{V}} \cdot \bar{\mathbf{m}})_L
\end{aligned} \tag{17}$$

$$\delta p = p_R - p_L, \quad \delta \rho = \rho_R - \rho_L \quad (18)$$

Roe numerical scheme presented above is often employed for the convective flux computation due to its simplicity and robustness and, more importantly, because it is strictly related to the hyperbolic nature of the Euler equations. However, this numerical scheme provides some nonphysical solutions for flows with strong normal shocks (see chapter 7). Consequently, various numerical schemes had been developed in the literature ([12], [6]) to overcome Roe scheme uncertainties. One of these numerical schemes is the Rotated Riemann solver Roe-HLL ([12]). This type of solver is based on the decomposition of the normal vector \bar{n} into two orthogonal directions. Suppose direction \bar{n}_1 has already been selected, the second direction \bar{n}_2 is simply aligned as perpendicular to \bar{n}_1 , i.e.

$$\begin{aligned} \bar{n}_1 \cdot \bar{n}_2 &= 0 \\ |\bar{n}_1| &= |\bar{n}_2| = 1 \end{aligned} \quad (19)$$

The normal is then projected onto these orthogonal directions, i.e.

$$\bar{n} = \alpha_1 \bar{n}_1 + \alpha_2 \bar{n}_2 \quad (20)$$

where $\alpha_1 \geq 0$ and $\alpha_2 \geq 0$ to preserve the same right and left states in both directions. Following this procedure, the numerical flux is decomposed as

$$F_j(\bar{n}) = \alpha_1 F_j(\bar{n}_1) + \alpha_2 F_j(\bar{n}_2) \quad (21)$$

To define the total interface flux, the numerical fluxes in the two directions \bar{n}_1 and \bar{n}_2 have to be selected. In the present code a hybrid scheme is implemented which combines the Roe flux for \bar{n}_2 and the HLL flux for \bar{n}_1 . Conclusively, the Roe-HLL flux can be expressed by

$$F_j = \frac{\lambda_R^+ F_n(W_L) - \lambda_L^- F_n(W_R)}{\lambda_R^+ - \lambda_L^-} - \frac{1}{2} \sum_{k=1}^4 |\lambda_{RHLL}^k| \bar{\omega}_{n_2}^k \bar{r}_{n_2}^k \quad (22)$$

where

$$\begin{aligned} \lambda_R^+ &= \max(0, (q_{n_1})_R + c_R, \bar{q}_{n_1} + \bar{c}) \\ \lambda_L^- &= \max(0, (q_{n_1})_L - c_L, \bar{q}_{n_1} - \bar{c}) \\ |\lambda_{RHLL}^k| &= \alpha_2 |\bar{\lambda}_{n_2}^k| - \frac{1}{\lambda_R^+ - \lambda_L^-} [\alpha_2 (\lambda_R^+ + \lambda_L^-) \bar{\lambda}_{n_2}^k + 2\alpha_1 \lambda_R^+ \lambda_L^-] \end{aligned} \quad (23)$$

To complete the rotated hybrid flux computation one needs to define the normal direction \bar{n}_1 for every interface with the assumption that \bar{n}_2 is always perpendicular to \bar{n}_1 . According to [12], the best choice is to consider the direction of the velocity vector difference taken over two adjacent cells,

$$\bar{n} = \begin{cases} \frac{\Delta \bar{q}}{\|\Delta \bar{q}\|}, & \|\Delta \bar{q}\| > \varepsilon \\ \bar{n}, & \|\Delta \bar{q}\| \leq \varepsilon \end{cases} \quad (24)$$

where $\Delta \bar{q} = [u_R - u_L \quad v_R - v_L \quad w_R - w_L]^T$ and ε is a small number. Following this approach, the direction \bar{n}_1 is always selected as normal to a shock or tangent to a shear layer if they exist.

4. SECOND ORDER SCHEME

Generally, the numerical flux computation on unstructured grids is accomplished by identifying a local one-dimensional Riemann problem at every interface of the computational domain. This implies that discontinuous states are assumed to exist on either side of a cell interface. A first order approach would require that the fluxes are computed simply by considering the volume-averaged quantities associated with the adjacent cells of the interface. However, when employed for viscous flows, first-order solutions are rather diffusive and determine excessive growth of boundary layers.

Second-order solutions are attained by redefining the right and left states of the interface. Consequently, we assume that the solution is piecewise linearly distributed over the control volume. This technique is known in the literature as piecewise linear state reconstruction. Following this approach, we can find the left and right states from

$$\begin{aligned} U_L &= \bar{U}_L + \psi_L(\bar{U}_L, \bar{U}_R, \nabla U_L)(\nabla U_L \cdot \bar{\mathbf{r}}_L) \\ U_R &= \bar{U}_R + \psi_R(\bar{U}_L, \bar{U}_R, \nabla U_R)(\nabla U_R \cdot \bar{\mathbf{r}}_R) \end{aligned} \quad (25)$$

where \bar{U}_L and \bar{U}_R are volume-averaged quantities attributed to cell centroids. The symbol ∇ denotes the gradient and the vectors $\bar{\mathbf{r}}_L$ and $\bar{\mathbf{r}}_R$ point from the cell centroid to the interface midpoint. ψ_L and ψ_R are limiter functions defined using volume-averaged quantities and solution gradients. It is imperative to mention that the variable U in equation (25) stands for any of the state variables for which the gradient needs to be computed.

To complete the states reconstruction, one needs to determine the solution gradients ∇U_L and ∇U_R . In the present work, a least-square method is implemented for gradients computation. This approach is based on the use of a first order Taylor series approximation for each direction $\bar{\mathbf{r}}_{ij}$ which points from the centroid of cell i to the centroid of cell j . The change in the solution along direction $\bar{\mathbf{r}}_{ij}$ is computed from:

$$\nabla U_i \cdot \bar{\mathbf{r}}_{ij} = U_j - U_i \quad (26)$$

When equation (26) is applied to all neighboring cells of the cell i results the following over-constrained system of linear equations ([17]):

$$\begin{bmatrix} \Delta x_{i1} & \Delta y_{i1} & \Delta z_{i1} \\ \dots & \dots & \dots \\ \Delta x_{iN} & \Delta y_{iN} & \Delta z_{iN} \end{bmatrix} \begin{Bmatrix} \partial_x U \\ \partial_y U \\ \partial_z U \end{Bmatrix} = \begin{Bmatrix} U_1 - U_i \\ \dots \\ U_N - U_i \end{Bmatrix} \quad (27)$$

where $\Delta(\bullet)_{ij} = (\bullet)_j - (\bullet)_i$ and N represents the total number of neighboring cells. Using the Gramm-Schmidt orthogonalisation from [17] ([11]), the solution of equation (27) follows immediately from

$$\nabla U_i = \sum_{j=1}^N \bar{\mathbf{w}}_{ij} (U_j - U_i) \quad (28)$$

where $\bar{\mathbf{w}}_{ij}$ are weights constructed solely from geometric information.

Second-order upwind spatial discretization requires the use of so-called limiters in order to prevent generation of oscillations and spurious solutions in regions of high gradients. The

purpose of the limiter function is to reduce the gradient used for left and right states reconstruction. In the present computations, we employ the widely used Venkatakrishnan limiter which reduces the gradient ∇U at the cell i by the factor

$$\Psi_i = \begin{cases} \frac{1}{\Delta_2} \left[\frac{(\Delta_{1,\max}^2 + \varepsilon^2)\Delta_2 + 2\Delta_2^2\Delta_{1,\max}}{\Delta_{1,\max}^2 + 2\Delta_2^2 + \Delta_{1,\max}\Delta_2 + \varepsilon^2} \right], & \Delta_2 > 0 \\ \frac{1}{\Delta_2} \left[\frac{(\Delta_{1,\min}^2 + \varepsilon^2)\Delta_2 + 2\Delta_2^2\Delta_{1,\min}}{\Delta_{1,\min}^2 + 2\Delta_2^2 + \Delta_{1,\min}\Delta_2 + \varepsilon^2} \right], & \Delta_2 < 0 \\ 1, & \Delta_2 = 0 \end{cases} \quad (29)$$

where

$$\begin{aligned} \Delta_2 &= \nabla U_i \cdot \vec{r}_L \\ U_{\min} &= \min(U_i, \min_j(U_j)) \\ U_{\max} &= \max(U_i, \max_j(U_j)) \\ \Delta_{1,\min} &= U_{\min} - U_i \\ \Delta_{1,\max} &= U_{\max} - U_i \end{aligned} \quad (30)$$

The parameter ε^2 is used to control the limitation. In practice, ε^2 is considered to be proportional with a local length scale, i.e.

$$\varepsilon^2 = (K\Delta h)^3 \quad (31)$$

where K is a unit constant and Δh is for example the cubic root of a cell volume. For detailed information about Venkatakrishnan limiter properties with respect to convergence and solution accuracy, the reader is referred to papers [17], [18] and [19].

5. TIME INTEGRATION

The system of first-order ordinary differential equations expressed in (7) is integrated in time using an explicit three-stage Runge-Kutta method developed by Shu and Osher ([8]):

$$\begin{aligned} \bar{W}_i^{(1)} &= \bar{W}_i^n - \frac{\Delta t}{\Omega_i} \mathbf{R}_i(\bar{W}_i^n) \\ \bar{W}_i^{(2)} &= \frac{3}{4} \bar{W}_i^n + \frac{1}{4} \bar{W}_i^{(1)} - \frac{1}{4} \frac{\Delta t}{\Omega_i} \mathbf{R}_i(\bar{W}_i^{(1)}) \\ \bar{W}_i^{n+1} &= \frac{1}{3} \bar{W}_i^n + \frac{2}{3} \bar{W}_i^{(2)} - \frac{2}{3} \frac{\Delta t}{\Omega_i} \mathbf{R}_i(\bar{W}_i^{(2)}) \end{aligned} \quad (32)$$

The time integration is carried out using a global time step chosen to be the minimum of the local time steps computed for each control volume i . To preserve numerical stability of the Runge-Kutta scheme, the time step for a cell i in the flow domain is computed from ([17]):

$$\Delta t_i = CFL \frac{\Omega_i}{\Lambda_c^x + \Lambda_c^y + \Lambda_c^z} \quad (33)$$

with

$$\begin{aligned}
\Lambda_c^x &= (|u| + c) \Delta S_i^x \\
\Lambda_c^y &= (|v| + c) \Delta S_i^y \\
\Lambda_c^z &= (|w| + c) \Delta S_i^z
\end{aligned} \tag{34}$$

where $CFL \leq 1$ and ΔS_i^x , ΔS_i^y and ΔS_i^z represent projections of the control volume i on yz , xz and xy plane respectively. Finally, the global time step follows immediately from:

$$\Delta t = \min_i (\Delta t_i) \tag{35}$$

The iterative process reaches convergence when the scaled density residual defined as

$$\varepsilon_\rho = \frac{\sqrt{NVF \sum_{i=1}^{NVF} (\rho_i^{n+1} - \rho_i^n)^2}}{\sum_{i=1}^{NVF} \rho_i^{n+1}} \tag{36}$$

is reduced by at least four orders in magnitude.

Usually, an explicit time-stepping scheme employs a relatively small time-step, due to stability restrictions, as it must be adapted to the smallest cell in the computational grid. Larger time step can be obtained by averaging the residuals \mathbf{R}_i with the ones corresponding to neighboring cells. This process increases the stencil of the numerical scheme and is denoted in literature as implicit residual smoothing ([5], [7]). Precisely, the residuals are filtered through a smoothing operator, i.e.

$$\bar{\mathbf{R}}_i = \mathbf{R}_i + \varepsilon \nabla^2 \bar{\mathbf{R}}_i \tag{37}$$

where

$$\nabla^2 \bar{\mathbf{R}}_i = \sum_j (\bar{\mathbf{R}}_j - \bar{\mathbf{R}}_i) \tag{38}$$

and ε is a smoothing parameter prescribed as 0.5. The summation in equation (38) uses the residuals from the neighboring cells that share an interface with cell i . For the resulting set of equations two Jacobi iterations are adequate to provide a reasonable approximation of $\bar{\mathbf{R}}_i$ at all cell centers. The present computations performed residual smoothing during each Runge-Kutta interior cycle and resulted in a CFL number of 1.8.

6. BOUNDARY CONDITIONS

Flow boundary conditions are implemented using ghost cells which are defined as images of adjacent interior cells across the exterior boundary. At wall boundaries, the flow tangency condition is imposed by creating an image for the interior velocity vector within corresponding ghost cell. Precisely, velocity components in the ghost cell are computed by subtracting twice the normal velocity from the adjacent interior cell velocity vector ([20], [21]):

$$\begin{aligned}
u_{ghost} &= u_{interior} - 2Un_x \\
v_{ghost} &= v_{interior} - 2Un_y \\
w_{ghost} &= w_{interior} - 2Un_z
\end{aligned} \tag{39}$$

where

$$U = \vec{V}_{interior} \cdot \vec{n} \quad (40)$$

Zero mass and energy flux through the wall surface is enforced by setting pressure and density in the ghost cells equal to corresponding interior values. This approach allows only a flux for the pressure terms of the momentum equations through a solid boundary.

Supersonic inlet boundary conditions are imposed simply by computing the state variables in the ghost cells using free-stream variables while at the supersonic outlet the state variables are extrapolated from the interior. Symmetry boundary conditions are also required due to the fact that two-dimensional flows are studied using three-dimensional grids. The procedure for implementing this type of conditions is similar to that employed for the solid boundary conditions.

7. RESULTS

A range of numerical studies were carried out in this section to show the code accuracy and capability of providing non-oscillating solutions for shock-wave dominated flows. Supersonic solutions for flow around a circular arc in a channel and flow past a circular cylinder were assessed for various Mach numbers. For the first case, a grid sensitivity analysis was performed by computing supersonic solutions at the same conditions on four types of meshes. In the second case, comparisons to theoretical data were made with respect to normal shock resolution, stagnation pressure and shock-wave standoff distance.

Flow around circular arc in a channel

The computational domain employed for this flow consists of a rectangular channel and a circular arc situated on its lower side (Fig. 1). Supersonic solutions were computed at $M_\infty = 2.0$ on four different computational grids to make an assessment of the grid sensitivity. Mesh size specifications are presented in Table 1 and Fig. 2.

Table 1 Mesh size specifications

	Mesh 1	Mesh 2	Mesh 3	Mesh 4
Total Cells	6000	24000	54000	96000
Total Nodes	8484	32964	73444	129924

The grid sensitivity analysis was carried out using Roe scheme with a first-order spatial discretization. The computations were performed using the three-stage Runge-Kutta scheme with global time-stepping, implicit residual smoothing and a *CFL* number of 1.8. Initial solution was set from free-stream variables and the iterative time integration was run until the normalized density residual dropped by at least four orders in magnitude. The Roe scheme showed excellent convergence properties given the fact that the normalized density residual decreased by approximately seven orders in magnitude the moment when calculations stopped (Fig. 3). Pressure contours and lower-wall pressure distributions are plotted in Fig. 5 and Fig. 6 respectively. It can be easily noticed the effect of mesh size over the shock resolution. Here, better shock resolutions are provided by finer meshes. The same effect can be observed in the wall pressure distributions where the best approximation is encountered for the finest mesh (Mesh 4). However, the solutions are strongly limited by the first-order spatial discretization since the two pressure peaks should have the same magnitude due to the fact that they arise from the same shock wave reflected by the upper-wall of the channel. For this reason, second-order solutions were computed on Mesh 4 using

both Roe and Roe-HLL schemes. Pressure contours and wall pressure distributions are presented in Fig. 7 and Fig. 6 respectively in comparison with first-order solutions.

Here, very sharp and clean resolution of the shock waves can be noted for both numerical schemes. Furthermore, the wall pressure distributions show excellent agreement between Roe and Roe-HLL solutions for both first and second-order spatial discretizations.

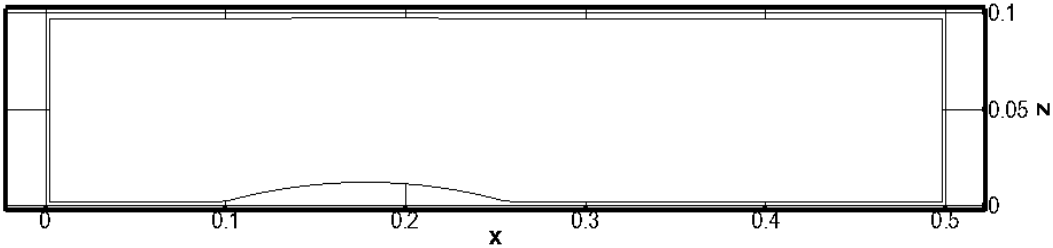


Fig. 1 Computational domain for flow around circular arc

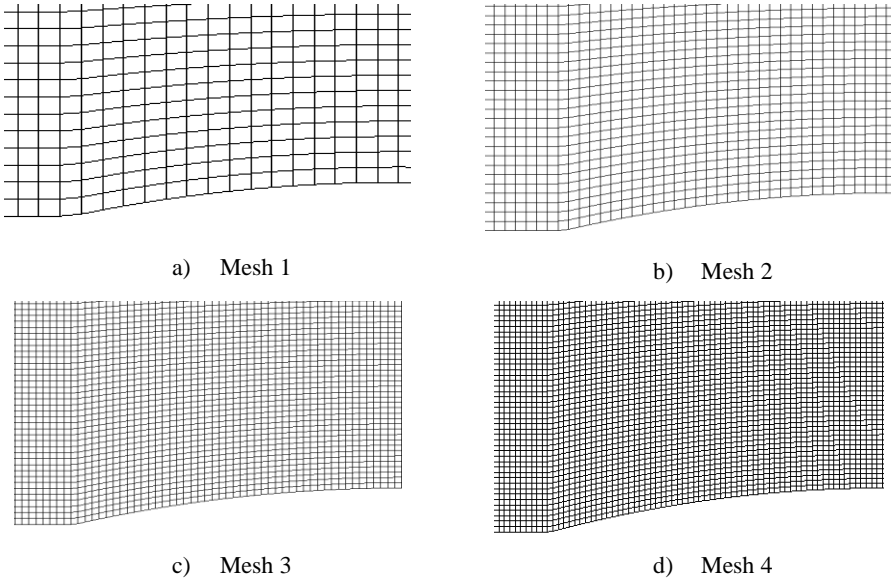


Fig. 2 Computational grids employed for grid sensitivity analysis

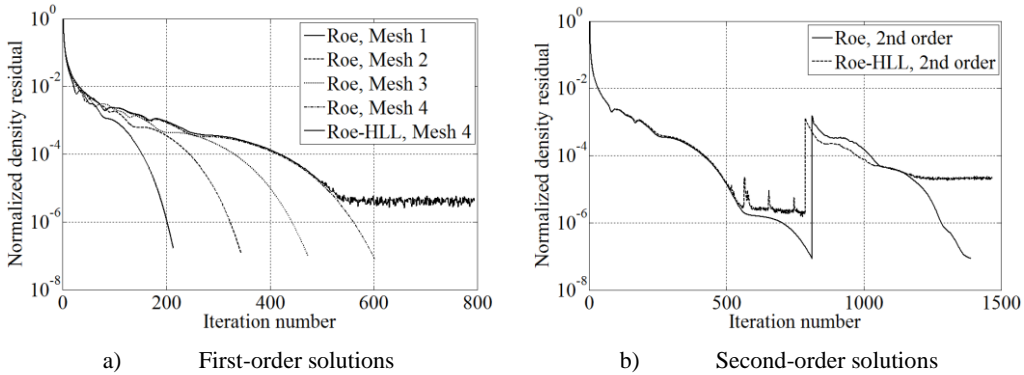


Fig. 3 Convergence histories for flow around circular arc

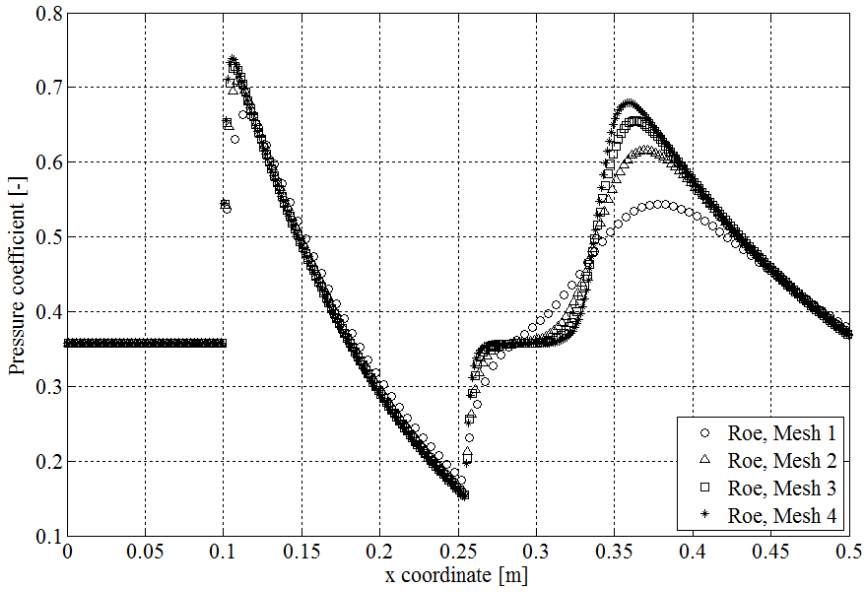
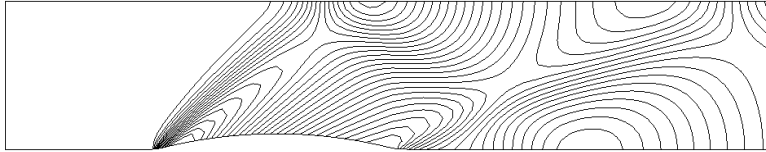
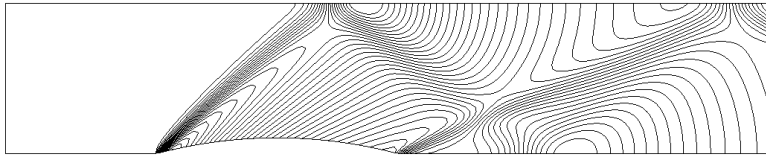


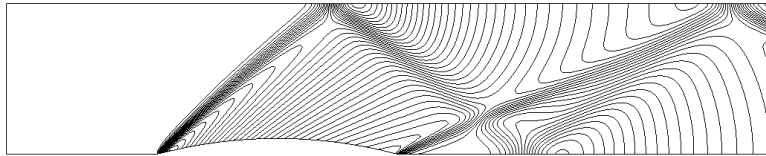
Fig. 4 First-order wall pressure distributions for Roe scheme



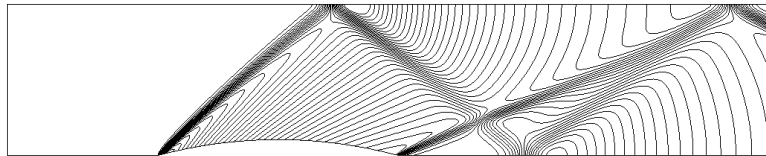
a) Mesh 1



b) Mesh 2



c) Mesh 3



d) Mesh 4

Fig. 5 First-order pressure contours for Roe scheme

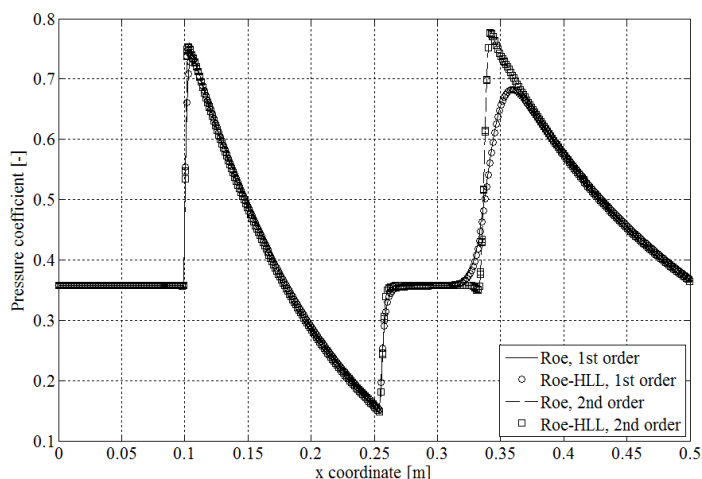
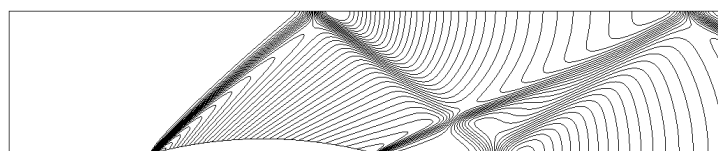
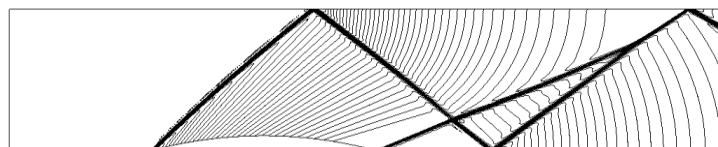


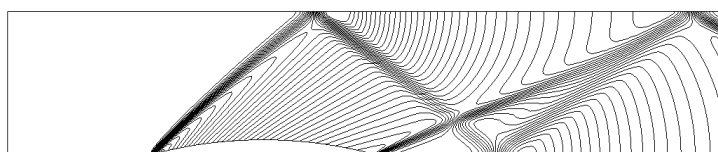
Fig. 6 Wall pressure distributions comparison between first and second-order solutions for Mesh 4



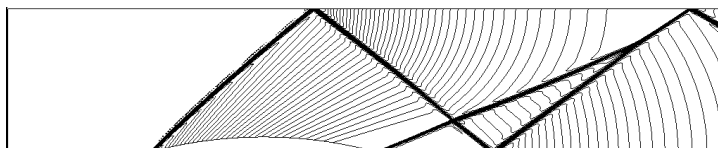
a) Roe, first-order



b) Roe, second-order



c) Roe-HLL, first-order



d) Roe-HLL, second-order

Fig. 7 Pressure contours comparison between first and second-order solutions for Mesh 4

Flow past a circular cylinder

Several computations were made for the flow past a circular cylinder at four different Mach numbers ($M_\infty = 2.5, 3.0, 5.0, 7.0$) to demonstrate the code robustness in obtaining accurate physical solutions for flows with strong normal shocks. The computational grid consists of 45000 cells and 61004 nodes and is shown in Fig. 9. First and second-order solutions were

computed using only the Roe-HLL scheme due to the fact that the Roe scheme provided solutions with nonphysical meaning (see Fig. 10). These nonphysical solutions were obtained by various researchers and are denoted in literature as “carbuncles” ([12], [13]). The main reason for this numerical instability is that the Roe scheme doesn’t provide sufficient numerical dissipation to deal with high solution gradients in strong normal shocks. The solutions were started from free-stream initial conditions and run until the normalized density residual dropped by five orders in magnitude at a point when convergence stagnated (Fig. 8). A second-order solution was computed only for $M_\infty = 2.5$ since pressure contours and normal shocks resolutions showed no noticeable differences between first and second-order solutions due to the relatively dense computational grid (Fig. 11, Fig. 13). Comparisons with theoretical stagnation pressures ([9]) and Hugoniot-Rankine relations for pressure before and after the normal shocks ([1], [3]) show excellent agreement between theory and presented method (see Fig. 14). Furthermore, Fig. 12 shows a good agreement between shock wave standoff distances predicted by present computations and experimental data from various researchers ([10]).

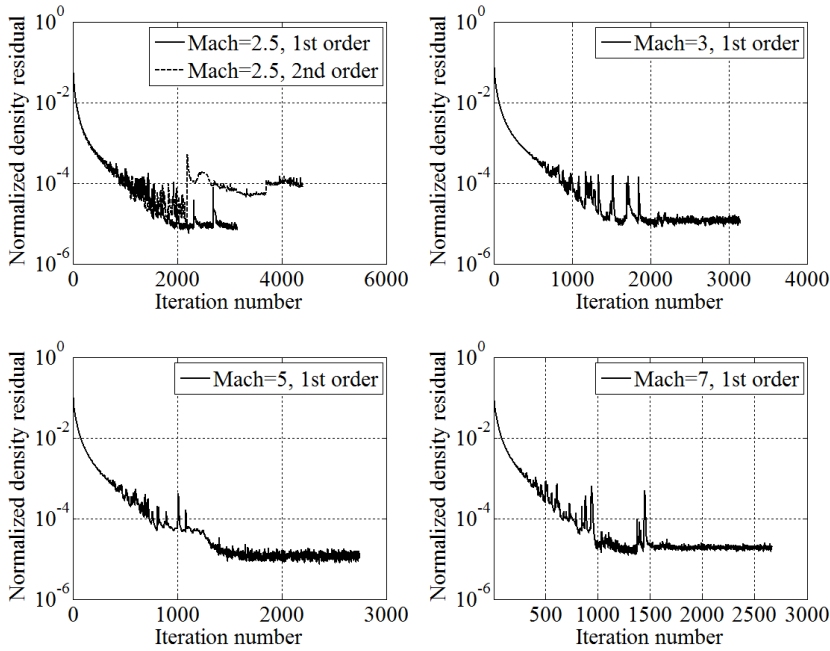


Fig. 8 Convergence histories for flow past circular cylinder

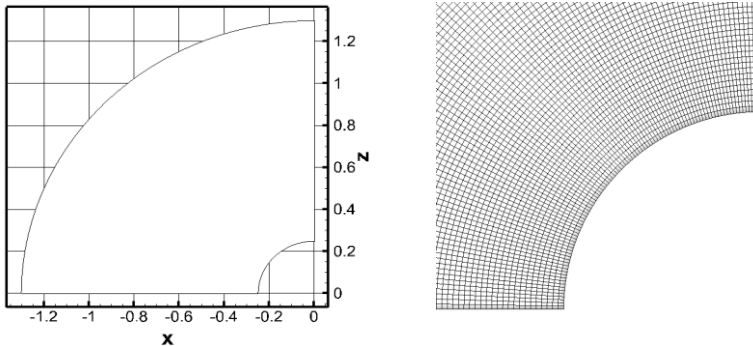


Fig. 9 Computational domain for flow past circular cylinder

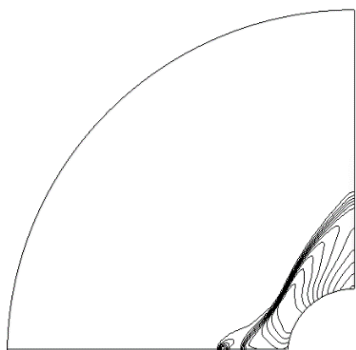


Fig. 10 Numerical instability for Roe scheme at $M_\infty = 5.0$

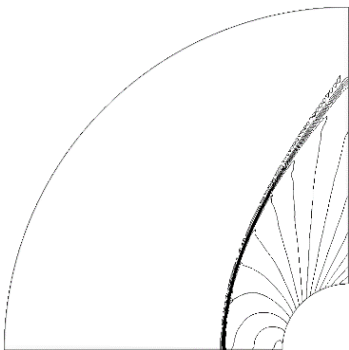


Fig. 11 Second-order pressure contours for Roe-HLL scheme at $M_\infty = 2.5$

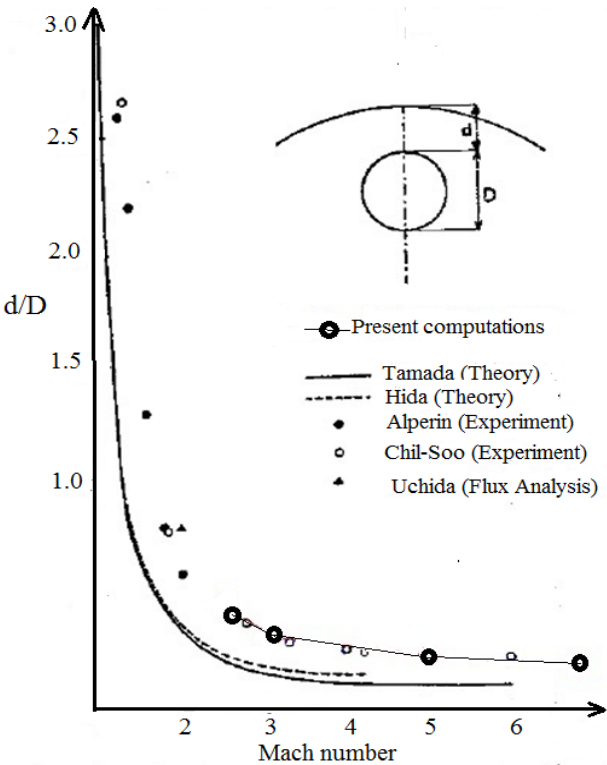
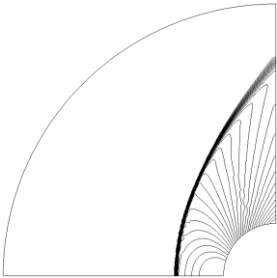
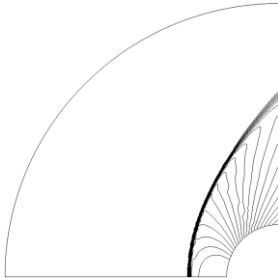


Fig. 12 Comparison between computed and experimental shock standoff distances



a) $M_\infty = 2.5$



b) $M_\infty = 3.0$

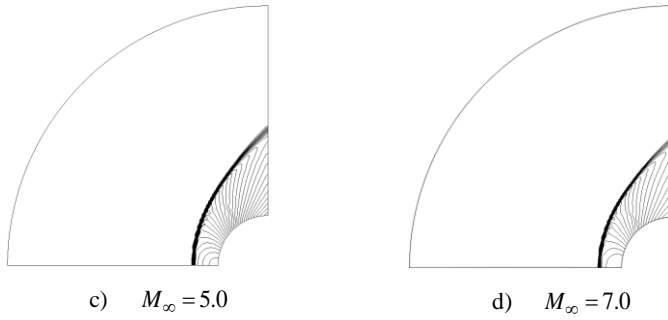


Fig. 13 First-order pressure contours for Roe-HLL scheme

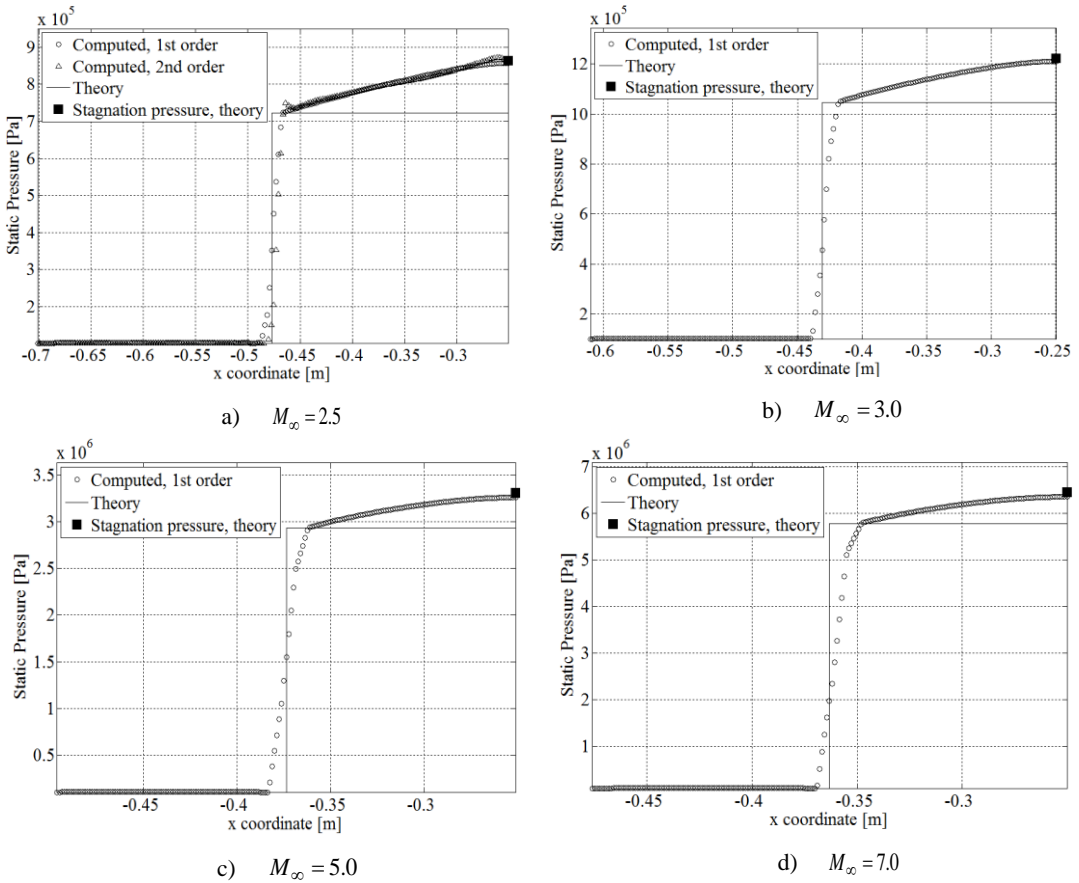


Fig. 14 Comparison between computed and theoretical normal shock resolution

8. CONCLUDING REMARKS

This paper assessed the development of a three-dimensional Euler solver for unstructured hexahedral meshes. Two upwind schemes for the computation of convective fluxes have been presented in the context of a cell-centered finite volume spatial discretization. The algorithm employs an explicit temporal discretization with global time-stepping and implicit residual smoothing. Higher-order solutions are accomplished by a linear state reconstruction technique that yields highly resolved solutions in smooth regions while providing a clean

resolution of shocks. Solution gradients required for the higher-order spatial discretization are computed using a least-square method and Venkatakrishnan limiter is employed to ensure the solution monotonicity and to avoid oscillations across shocks.

Results have been presented for two configurations at various supersonic and hypersonic speeds to show the accuracy and the robustness of the flow solver in resolving shock-wave dominated flows. A grid sensitivity analysis has been performed that showed a relatively high sensitivity of the solution to mesh size for a first-order spatial discretization. Solutions improved considerably when a higher-order discretization had been employed yielding a very high resolution of shock waves. Comparisons to theoretical and experimental data with respect to stagnation pressure, normal shock resolution and shock wave standoff distance showed excellent agreement with the presented method.

REFERENCES

- [1] E. Carafoli and V. N. Constantinescu, *Dinamica fluidelor compresibile*, Ed. Academiei, București, 1984.
- [2] S. Dănilă and C. Berbente, *Metode numerice în dinamica fluidelor*, Ed. Academiei Române, București, 2003.
- [3] V. N. Constantinescu and S. Găletușe, *Mecanica fluidelor și elemente de aerodinamică*, Ed. Didactică și Pedagogică, București, 1983.
- [4] J.B. Vos, P. Leyland, V. Van Kemenade, C. Gacherieu, N. Duquesne, P. Lodstedt, C. Weber, A. Ytterstrom and C. Saint Requier, *NSMB Handbook 4.5*, IMHEF-DGM-EPFL, Department of Aeronautics KTH, 1999.
- [5] N. T. Frink, Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Meshes, *AIAA Journal*, Vol. **30**, 1992.
- [6] R. Radespiel and N. Kroll, Accurate flux vector splitting for shocks and shear layers, *Journal of Computational Physics* **121**, 66-78, 2007.
- [7] N. T. Frink, P. Parikh and S. Pirzadeh, A Fast Upwind Solver for the Euler Equations on Three-Dimensional Unstructured Meshes, *AIAA 91-0102*, Jan. 1991.
- [8] Sigal Gottlieb, On High Order Strong Stability Preserving Runge-Kutta and Multi Step Time Discretizations, *Journal of scientific computing*, vol. **25**, 2005.
- [9] K. Masatsuka, *I do like CFD –Governing equations and exact solutions*, vol. 1, 2009.
- [10] K. Chul-Soo, Experimental Studies of Supersonic Flow past a Circular Cylinder, *Journal of the Physical Society of Japan*, **11**: 439–+, April 1956.
- [11] I. N. Bronshtein and K. A. Semendyayev, *Handbook of Mathematics*, K. A. Hirsch, Van Nostrand Reinhold Company, 1985.
- [12] Hiroaki Nishikawa and Keiichi Kitamura, Very simple, carbuncle-free, boundary-layer-resolving, rotated-hybrid Riemann solvers, *Journal of Computational Physics*, 2007.
- [13] M. Pandolfi and D. D'Ambrosio, Numerical instabilities in upwind methods: Analysis and cures for the 'carbuncle' phenomenon, *Journal of Computational Physics* **166**, 271–301, 2001.
- [14] P. Lax and B. Wendroff, Difference schemes with high order accuracy, *Comm. Pure and Applied Mathematics*, 1960.
- [15] P. Lax and B. Wendroff, Systems of conservation laws, *Comm. Pure and Applied Mathematics*, 1960.
- [16] I. E. R. Courant and M. Rees, On the solution of nonlinear hyperbolic differential equations by finite differences, *Comm. Pure and Applied Mathematics*, 1952.
- [17] J. Blazek, *Computational Fluid Dynamics: Principles and Applications*, Kidlington, Oxford OX5 1GB, 2001.
- [18] M. Aftosmis, D. Gaitonde and T. Sean Tavares, On the Accuracy, Stability and Monotonicity of Various Reconstruction Algorithms for Unstructured Meshes, *AIAA Paper 94-0415*, 1994.
- [19] V. Venkatakrishnan, Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters, *Journal of Computational Physics* **118**, 120-130, 1995.
- [20] N. T. Frink, P. Parikh and S. Pirzadeh, A fast Upwind Solver for The Euler Equations on Three-Dimensional Unstructured Meshes, *AIAA Paper 91-0102*, 1991.
- [21] N. T. Frink, Recent Progress Towards Three-Dimensional Unstructured Navier-Stokes Flow Solver, *AIAA Paper 94-0061*, 1994.