
A 2-D FINITE VOLUME NAVIER-STOKES SOLVER FOR SUPERSONIC FLOWS

Adem Gürkan TÜRK¹, Bayram ÇELİK^{1,*}

¹ Astronautical Engineering Department, Faculty of Aeronautics and Astronautics, İstanbul Technical University, İstanbul, Turkey

ABSTRACT

Two-dimensional compressible Navier-Stokes equations consist of continuity, momentum and energy equations. These coupled equations must be solved simultaneously. The values of density, two velocity components, pressure and temperature are obtained by solving the four equations and considering the assumption of calorically perfect gas. A finite volume based in house solver is developed in C++. The solver is able to solve steady or unsteady and inviscid or laminar compressible flow problems on an unstructured mesh. It uses Van Leer's Flux Vector Splitting Scheme. It has local time stepping feature for the acceleration of convergence and can run parallel on computers with shared memory architecture. The solver is designed to adapt alternative schemes for better accuracy and computing efficiency. In order to check the accuracy and code implementation, various supersonic benchmark problems are visited. The problems are steady inviscid double wedge, steady viscous flow over a cylinder, unsteady inviscid Sod's problem and unsteady inviscid forward facing step flow. The obtained results are compared with those available in literature. The comparisons show that the results are promising.

Keywords: Supersonic flows, Finite volume method, Forward step duct, Shock tube, Oblique shock

1. INTRODUCTION

For a two dimensional compressible flow of a calorically perfect gas, governing equations consists of conservations of mass, momentums in x-y directions, and energy. The equations are coupled and thereby, they must be solved simultaneously. Computational Fluid Dynamics, CFD aims to solve the equations numerically. CFD has improved along with the rapid progress in computer technology during the 1970s. In order to find the best schemes or discretization techniques in terms of accuracy and computational costs, a great number of studies were conducted, and new methodologies were developed. As a result of those efforts and progresses, new techniques were proposed such as flux vector splitting for shock capturing, artificial viscosity or thin-layer approximations for flows with high Reynolds number, and etc. CFD was not feasible for many problems in the past because of limited computer power. However, today catching the correct location of the shock, understanding of complex interactions are possible with CFD. For example, today CFD is in use for designing and manufacturing a hypersonic re-entry and air breathing spacecraft. These vehicles operate in a flow with high enthalpy where the real gas effects are non-negligible [1]. Modelling such flows realistically requires consideration of the chemical reaction in the solution.

The other major challenges in supersonic flow modelling are shock-shock and shock-boundary layer interactions that can result in high heat transfer rates between the flow and the surface. Observation of such interactions in ground facilities are expensive and limited with the operating range of the instruments used in the experiment. Underestimating consequences of such complex phenomena in design stage can change flight characteristics of a spacecraft drastically [1]. In February 1, 2003, Thermal Protection System (TPS) of USA space shuttle damaged and caused structural failure during its re-entry. Later, extended investigation revealed that the failure resulted from shock-shock interaction [2].

*Corresponding Author: celikbay@itu.edu.tr

An article by Lax is one of the most important advancements for compressible flow and wave motion [3]. The article is about weak solutions of non-linear hyperbolic equations. The method is first order accurate and allows to model structure and trajectory of a shock wave. Today, the method proposed by Lax is widely known and called as “shock capturing”. Later, Lax and Wendroff laid the foundation of second order accurate schemes with a paper that is on conservation law [4]. Almost a decade later, Mac-Cormack introduced a two step explicit method [5]. Godunov made a major contribution in the field of compressible flow by introducing a treatment for artificial viscosity [6]. The treatment is able to stabilize the oscillation that takes place in the vicinity of strong gradients. In the following years, Rosanov and then Turkel et al. introduced their third and fourth order accurate schemes, respectively [7-8]. However, the second order accurate schemes have been dominant for the last three decades. With advancement in computer technology, rapid and accurate solution needs for Navier-Stokes and Euler’s equations made high demand on powerful implicit schemes [9]. The implicit scheme of Yee [10], approximate Riemann solver of Roe [11] are among the most important studies that aim to accelerate computations. The review article by Kostoff and Cummings [9] lists and gives the details of the important advancement in the computational compressible flow.

There are several flux calculation schemes available in literature with different levels of accuracies. Liu, Osher and Chang introduced higher order WENO schemes [12]. Later, Serna and Marquina derived a fifth-order accurate Weighted Power ENO method [13]. Both schemes are using different reconstruction techniques to achieve higher order capability. Liou and Steffen proposed another type of upwind schemes, AUSM (Advection Upstream Splitting Method) [14]. This method separates convective fluxes by considering not only the eigenvalues but also pressure. AUSM has been improved and named as AUSM+ -up that is now applicable for all flow regimes [15]. *Van Leer’s Flux Vector Splitting Scheme* is first order accurate but it is very popular because of its relative simplicity. It is able to capture shock in reasonably good sharpness for transonic-supersonic flows [16]. Van Leer’s flux vector splitting scheme separates the convective fluxes into two parts as positive and negative ones. It is also differentiable at sonic point without smearing [17].

One of the key point of capturing a shock correctly is to have an adequate mesh. The mesh generation strategy in the regions of boundary layer differs from the regions near the shocks. Therefore, using a structured mesh in the computations of the flow fields increases computational costs tremendously and brings challenges. Today, there is a rich variety of solvers available in both academia and industry. Some of them are CFL3D, developed and maintained by NASA, SU2 by Stanford University, Ansys FLUENT, and OpenFOAM. All these solvers are finite volume based Navier-Stokes solvers. Finite volume method was introduced by McDonald to simulate inviscid transonic flow on turbine cascades [18]. This method solves the integral form of the Navier-Stokes equations by calculating the convective and diffusive parts of it as a summation of the fluxes at the faces of the control volumes. Even though the formulas of the mass, momentum and energy are clear, finding the values of the primitive variables or the fluxes at the faces of the control volume are not straightforward. The decision on where the variables are hold comes with some pros and cons. The two preferences for the location of holding the variables are known as *cell – centered* and *cell – vertex* schemes. For example, the fluxes at the faces can be approximated by using upwind schemes or central schemes for the convective fluxes and Galerkin Method [19] for the viscous fluxes.

The main objective of the present article is to examine a newly developed finite volume based Navier-Stokes in-house solver by dealing with the well-known benchmark problems for compressible flow. Current version of the solver accepts structured and unstructured meshes, can run on a computer with multiple cores in parallel manner, and its modular structure allows implementation of other type of schemes with minimum effort. The authors expect to contribute national literature and capabilities in the field of compressible flow.

The developed solver is tested by considering four benchmark problems available in the literature. The first one is a typical oblique shock that is generated by a wedge. Since the analytical result is available for the problem by shock-expansion theory, it is a benchmark problem to test the solver shock resolving ability of the developed solver. In order to check the accuracy of the solver especially for the discretization of the viscous terms, we consider the problem of supersonic flow over a cylinder. The comparisons of the obtained results with an experimental study [20] allows us to examine the applicability of the developed code for a real fluid (laminar) flow over object that has curved boundary. This benchmark problem, where an unstructured mesh is in use also allows us to check mesh-handling capabilities of the solver. The third benchmark problem is another well-known supersonic test problem for unsteady flow. The problem first introduced by Gary Sod in 1978, where an infinitely long 1D tube is modelled. Two regions with different densities having a discontinuity at their interface is given as initial conditions. After that, the solution is started, and created waves are observed for comparing them with the solvers of other researchers. The reason of choosing this as a test case is that analytical result is present, it is unsteady and common in the literature. Final test is forward facing step problem. In this case, generated shock is interacting with the boundaries as the time passes. This would give us the opportunity to observe the implemented boundary conditions. This test was desired to be run as a viscous flow, but that would require too much number of elements in order to catch shock-boundary layer interactions. Hence, it is decided to run as inviscid. The solution is compared with another transient inviscid solution.

Present article is designed as follows: Governing equations for compressible flows are introduced in Chapter 2, and then their temporal and spatial discretizations are explained in Chapter 3. The benchmark problems are introduced, and the obtained solutions are shown in Results and Discussions. The manuscript ends with Conclusion.

2. GOVERNING EQUATIONS

Navier-Stokes equations for a two-dimensional compressible flow in Cartesian coordinates without source, body forces, and volumetric heating terms are as follows:

$$\frac{\partial W}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = \frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y} \quad (1)$$

where

$$W = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e_t \end{bmatrix}, E = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho e_t + p)u \end{bmatrix}, F = \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ (\rho e_t + p)v \end{bmatrix}, E_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \theta_x \end{bmatrix}, \text{ and } F_v = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \theta_y \end{bmatrix}.$$

$$e_t = e + \frac{u^2 + v^2}{2} \quad (2)$$

$$\theta_x = u\tau_{xx} + v\tau_{xy} + k \frac{\partial T}{\partial x} \quad (3)$$

$$\theta_y = u\tau_{yx} + v\tau_{yy} + k \frac{\partial T}{\partial y} \quad (4)$$

For closing the system, assumption of calorically perfect gas is needed.

$$p = \rho RT \quad (5)$$

$$e = c_v T \quad (6)$$

Viscosity can be related with temperature by using Sutherland’s Law. For a Newtonian fluid, shear stresses and viscosity are related as follows:

$$\tau_{xx} = 2\mu \frac{\partial u}{\partial x} + \lambda \nabla \cdot \vec{V} \tag{7}$$

$$\tau_{xy} = \tau_{yx} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \tag{8}$$

$$\tau_{yy} = 2\mu \frac{\partial v}{\partial y} + \lambda \nabla \cdot \vec{V} \tag{9}$$

where $(\lambda = -2\mu/3)$.

3. NUMERICAL METHODS

The integral form of the Navier-Stokes equations over a volume, Ω is as follows:

$$\frac{\partial}{\partial t} \int_{\Omega} \vec{W} d\Omega + \oint_{\partial\Omega} (\vec{F}_c - \vec{F}_v) dS = 0 \tag{10}$$

The vectors appearing in the equation above can be written explicitly as

$$\vec{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e_t \end{bmatrix} \quad \vec{F}_c = \begin{bmatrix} \rho V \\ \rho u V + n_x p \\ \rho v V + n_y p \\ \rho(e_t + p/\rho)V \end{bmatrix} \quad \vec{F}_v = \begin{bmatrix} 0 \\ n_x \tau_{xx} + n_y \tau_{xy} \\ n_x \tau_{yx} + n_y \tau_{yy} \\ n_x \theta_x + n_y \theta_y \end{bmatrix} \tag{11}$$

where V denotes the velocity in the direction of \vec{n} .

$$V = [u, v]^T \cdot [n_x, n_y]^T = n_x u + n_y v \tag{11}$$

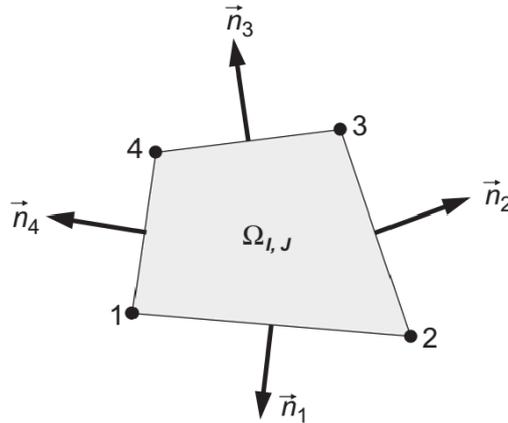


Figure 1: Schematic of control volume Ω , taken from [21]

3.1. Spatial and Temporal Discretization

Equation (10) can be discretised spatially over the control volume, Ω as follows.

$$\frac{\partial}{\partial t} \int_{\Omega} \vec{W} d\Omega = - \sum_{n=1}^{NF} (\vec{F}_c - \vec{F}_v)_n \Delta S_n \tag{12}$$

where NF stands for number of the faces. The term on the left hand side of the Equation (12) can be approximated by using first order finite difference formula as below:

$$\frac{\partial}{\partial t} \int_{\Omega} \vec{W} d\Omega = \Omega \frac{\partial \vec{W}}{\partial t} = \Omega \frac{\vec{W}^{n+1} - \vec{W}^n}{\Delta t} \quad (13)$$

After the spatial and temporal discretizations, Equation (10) takes the following form:

$$\vec{W}^{n+1} = \vec{W}^n - \frac{\Delta t}{\Omega} \sum_{n=1}^{NF} (\vec{F}_c - \vec{F}_v)_n \Delta S_n \quad (14)$$

Developed solver utilizes explicit type of calculations. Convective and diffusive fluxes at time n , \vec{W}^n are utilized to calculate \vec{W}^{n+1} . For steady state calculations, 3 Stage Runge -Kutta Scheme is used. Only the first and the third stages include viscous fluxes.

$$\vec{W}_i^{(1)} = \vec{W}_i^{(n)} - 0.1481 * \frac{\Delta t_i}{\Omega_i} * \sum_{n=1}^{NF} (\vec{F}_c^{(n)} - \vec{F}_v^{(n)})_n * \Delta S_n \quad (15)$$

$$\vec{W}_i^{(2)} = \vec{W}_i^{(n)} - 0.4 * \frac{\Delta t_i}{\Omega_i} * \sum_{n=1}^{NF} \vec{F}_c^{(1)} * \Delta S_n \quad (16)$$

$$\vec{W}_i^{(n+1)} = \vec{W}_i^{(n)} - 1 * \frac{\Delta t_i}{\Omega_i} * \sum_{n=1}^{NF} (\vec{F}_c^{(2)} - \vec{F}_v^{(2)})_n * \Delta S_n \quad (17)$$

$\vec{F}_n^{(1)}$ is found with the variables obtained from $\vec{W}_i^{(1)}$.

In the solution of unsteady flow problems, *Dual Time - Stepping Scheme* is used. The idea of this scheme is allowing the element to exchange all fluxes around its neighbour elements using pseudo time steps. After that, every element is integrated with the physical time step. Time integration scheme used in steady problems is used in the pseudo time stepping. There are many improvements made by other researchers in order to make this approach stable. Detailed information on the scheme can be found at [22]. For the sake of clarity and convenience, calculation procedure used in this study is presented directly as follows:

$$\vec{W}_i^{(1)} = \vec{W}_i^{(n)} - 0.1481 * \frac{\Delta t_i^*}{\Omega_i} * \left[1 + \frac{3}{2 * \Delta t} * 0.1481 * \Delta t_i^* \right]^{-1} * \sum_{n=1}^{NF} (\vec{F}_c^{(n)} - \vec{F}_v^{(n)})_n * \Delta S_n + \frac{3}{2 * \Delta t} * \Omega_i * \vec{W}_i^{(n)} - Q_i \quad (18)$$

$$\vec{W}_i^{(2)} = \vec{W}_i^{(n)} - 0.4 * \frac{\Delta t_i^*}{\Omega_i} * \left[1 + \frac{3}{2 * \Delta t} * 0.4 * \Delta t_i^* \right]^{-1} * \sum_{n=1}^{NF} (\vec{F}_c^{(1)} - \vec{F}_v^{(1)})_n * \Delta S_n + \frac{3}{2 * \Delta t} * \Omega_i * \vec{W}_i^{(n)} - Q_i \quad (19)$$

$$\vec{W}_i^{(3)} = \vec{W}_i^{(n)} - 1 * \frac{\Delta t_i^*}{\Omega_i} * \left[1 + \frac{3}{2 * \Delta t} * 1 * \Delta t_i^* \right]^{-1} * \sum_{n=1}^{NF} (\vec{F}_c^{(2)} - \vec{F}_v^{(2)})_n * \Delta S_n + \frac{3}{2 * \Delta t} * \Omega_i * \vec{W}_i^{(n)} - Q_i \quad (20)$$

$$Q_i = \frac{2}{\Delta t} * \Omega_i * \vec{W}_i^{(n)} - \frac{1}{2 * \Delta t} * \Omega_i * \vec{W}_i^{(n-1)} \quad (21)$$

After the relative error become less than $1e-5$, pseudo time stepping is skipped, and elements are integrated with physical time step.

$$\left| \frac{\bar{W}_i^{(3)} - \bar{W}_i^{(n)}}{\bar{W}_i^{(n)}} \right| < 10^{-5} \quad (22)$$

$$\bar{W}_i^{(n+1)} = \bar{W}_i^{(3)} - \frac{\Delta t}{\Omega_i} * \sum_{n=1}^{NF} \left(\vec{F}_c^{(n)} - \vec{F}_v^{(n)} \right)_n * \Delta S_n \quad (23)$$

The left and the right coming fluxes are calculated by using the primitive variables of the element on the left and right hand sides of the faces, respectively. Gradients of the velocity and temperature are needed for the calculations of the viscosity and conduction related terms appearing in the momentum and energy equations. Developed code holds the primitive variables at the center of the elements. For calculating the fluxes at the faces, Van Leer’s Flux Vector Splitting Scheme is used. Since the solver is *cell – centered*, it is convenient to find the gradients at the cell centers with Green-Gauss approach as below:

$$\nabla U \approx \frac{1}{\Omega} \int_{\partial\Omega} U \vec{n} dS \quad (24)$$

In order to find the gradients at the faces, the gradient values of the two adjacent elements are averaged. Another approach for finding the derivatives at the faces is finite difference approach. The primitive variables at the faces of two neighbour elements are known beside the vector that connects them (see Figure 2). Therefore spatial derivatives can be written as below:

$$\frac{\partial U}{\partial x} \approx \frac{U_{i+1} - U_i}{\vec{r}_{12} \cdot \hat{i}} \quad (25)$$

It is preferred to keep the values of the variables at the center of the face during the calculation of the fluxes, \vec{F}_c and \vec{F}_v . A diamond in Figure 2 represents this point.

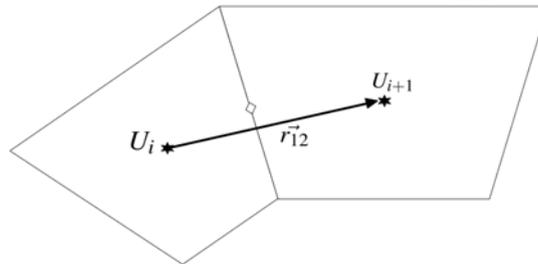


Figure 2. Schematic of two elements

3.2. Implementation of Boundary Conditions

Four type of boundary conditions, supersonic inflow, supersonic outflow, slip-wall and no-slip wall with the option of adiabatic wall or constant temperature are implemented so far.

Boundary conditions are embedded into the fluxes itself. For example, on a slip wall, \mathbf{V} must be equal to 0 because of no penetration. Then \vec{F}_c coming from the wall to the element becomes;

$$\vec{F}_c = \begin{bmatrix} 0 \\ n_x p \\ n_y p \\ 0 \end{bmatrix} \quad (26)$$

Value of \mathbf{p} can be extrapolated from the interior points. In this study, the \mathbf{p} value of the corresponding element is used directly on the $\overrightarrow{\mathbf{F}}_c$ coming from the wall. On a no-slip wall, \mathbf{V} will be equal to 0, thus $\overrightarrow{\mathbf{F}}_c$ will be same as on the case of slip wall.

If supersonic inflow will be used, all waves will be directed to inside of the domain, meaning that all primitive variables at the inflow boundary must be specified directly. In the solver, ρ , u , v , T are written in a text file, and they will be assigned automatically as inflow variables. These variables are straightly put into calculation of fluxes at the boundary face. Inflow conditions are also used as initial conditions.

$$\overrightarrow{\mathbf{F}}_{c,b} = \begin{bmatrix} \rho_i V_i \\ \rho_i u_i V_i + n_x p_i \\ \rho_i v_i V_i + n_y p_i \\ \rho_i (\mathbf{e}_{t,i} + \mathbf{p}_i / \rho_i) V_i \end{bmatrix} \quad (27)$$

If the boundary is supersonic outflow, all waves will go out and leave the domain. Thus, flux coming from the boundary to the element is calculated with all primitive variables at the the element itself if the first order extrapolation is adequate:

$$\overrightarrow{\mathbf{F}}_{c,b} = \begin{bmatrix} \rho_e V_e \\ \rho_e u_e V_e + n_x p_e \\ \rho_e v_e V_e + n_y p_e \\ \rho_e (\mathbf{e}_{t,e} + \mathbf{p}_e / \rho_e) V_e \end{bmatrix} \quad (28)$$

In wall boundary conditions, T can be calculated using extrapolation -meaning adiabatic wall-, or be given directly as a constant value. First approach corresponds to Neumann type while the latter corresponds to Dirichlet type boundary condition.

At the calculation of viscous fluxes, the only addition will be at the derivative terms. Used approach can be checked at the end of Spatial And Temporal Discretization topic. There is no neighbour element at the boundary, so only the opposite element and the value in the previous time steps at the wall are used, i.e. forward difference.

3.3. Parallelization

Current solver uses OpenMP for parallel computations. OpenMP is an API that provides developers making their computer applications run with multiple threads on computers having shared memory architecture. Furthermore, it supports multiple programming languages such as C/C++ and Fortran [23]. It uses an approach called Fork – Join Model.

Parallel solution works as follows: At the beginning of a time integration, multiple threads are created. Total element numbers are divided in smaller groups and are assigned to threads. When all the workers finish their job, the code moves forward and the workers are destroyed at the end of the time integration.

4. RESULTS AND DISCUSSION

4.1. Steady Inviscid Flow Over A Wedge

A supersonic inviscid flow over a wedge having a half angle of 15° is considered as a benchmark problem in this section. This supersonic flow will create an oblique shock over the wedge. Computational domain and the structured mesh generated for the solution are shown in Figure 3.

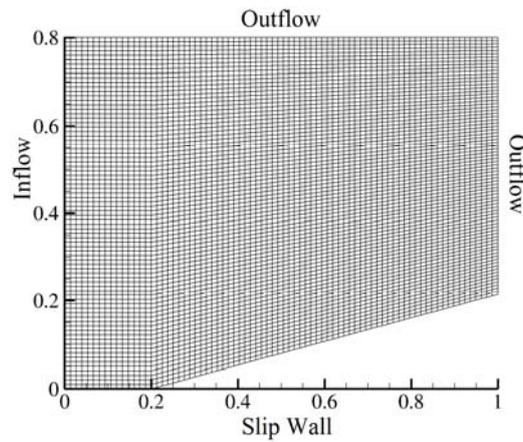


Figure 3. Computational domain, mesh and boundary conditions

Solution domain is 1 m long and 0.8 m high where the wedge starts at $x = 0.2$ m. A flow with $p = 101325$ Pa, and $T = 300$ K at $M = 2.4$ enters the domain from the left and reaches the leading edge of the wedge at $x = 0.2$ m.

Total number of elements for three different mesh resolutions and total number of iterations for them to reach L2 norm error less than 10^{-8} are tabulated in Table 1:

Table 1. Number of elements and iteration numbers

Case	Total Cell	Number of Iterations
1	6162	497
2	25122	867
3	101442	1560

Variations of pressure and Mach values along x direction at $y = 0.25$ m are shown in Figure 4a and Figure 4b, respectively. In order to be sure that the calculated results are correct, comparing them with the oblique shock theory is necessary. From the theory:

$$M_2 = 1.79267, p_2/p_1 = 2.40481$$

The p_2/p_1 and downstream Mach number, M_2 values obtained from the developed code can be seen on Table 2 (Column 2 and 3) in addition to the percentage error relative to the theory written above (the last two columns). The error in static pressure ratio across the shock wave, p_2/p_1 and downstream Mach number M_2 for the three meshes are calculated and tabulated in Table 2. As expected, calculated values are getting closer to the theory as the total element number increases. It can be concluded that using a domain with 101442 elements is more than enough for reaching mesh independent result. Obtained Mach contours are shown in Figure 5.

Table 2. The obtained results (left) and the errors relative to the oblique shock theory (right)

Case	p_2/p_1	M_2	% p_2/p_1	% M_2
1	2.4056	1.7866	0.0362	0.3364
2	2.4052	1.7895	0.0198	0.1768
3	2.4047	1.7919	0.0015	0.0385

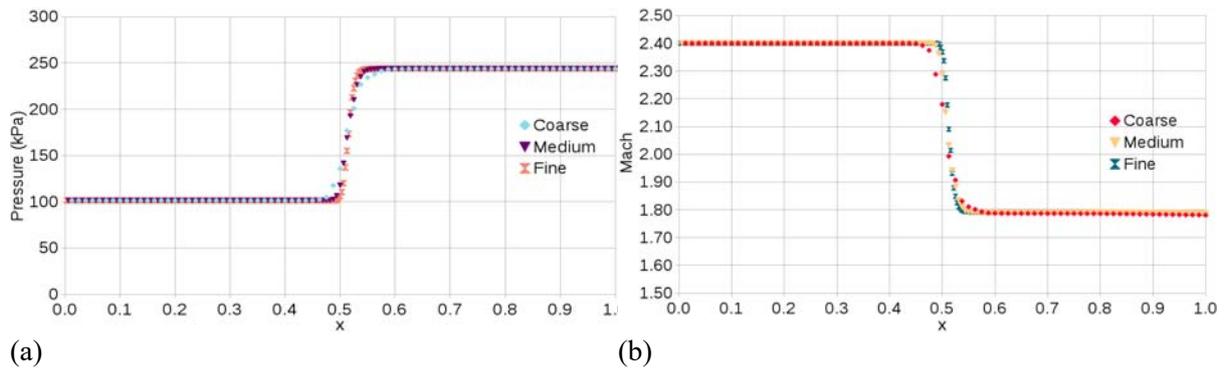


Figure 4. Comparisons of the obtained pressure and Mach variation along a line, where $y = 0.25\text{ m}$

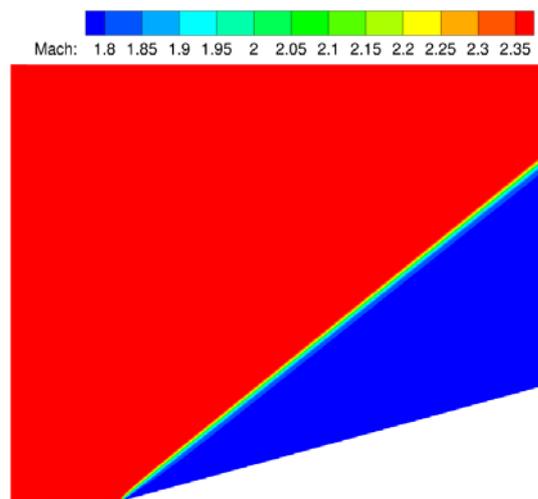


Figure 5. Obtained Mach contours

4.2. Steady Viscous Flow Around A Cylinder

A viscous flow over a circular cylinder at $M = 2.9$ is the another benchmark problem studied here. An O-type structured mesh that has a circular outer boundary is generated for the solution. The flow enters the computational domain from the left-half of the computational domain (see Figure 6). The other half of the outer boundary is set to be supersonic outflow. On the cylinder surface, no-slip and adiabatic wall boundary conditions are prescribed. Inflow conditions are also used as initial conditions, where $M = 2.90$, $p = 1139.015\text{ Pa}$, $T = 300\text{ K}$, and $Re = 7.2 \times 10^5$. Mesh sizes, relative errors in terms of pressure at the stagnation point where $\theta = 0^\circ$, and at the downstream where $\theta = 180^\circ$ can be found at Table 3. Relative errors are calculated using the experimental data presented in Reference [20].

Table 3. Number of elements used in the laminar flow around the cylinder

Case	Total Cell	Error at $\theta = 0^\circ$	Error at $\theta = 180^\circ$
1	8820	1.09449	20.19685
2	17640	0.83996	17.26973
3	35640	0.83192	16.29402

Obtained pressure coefficient distributions on the surface of the cylinder are plotted alongside the experimental data in Figure 6. Mesh having a total cell number of 8820 and the Mach contours obtained by solving the equations on the mesh computationally are shown in Figure 7a, and 7b.

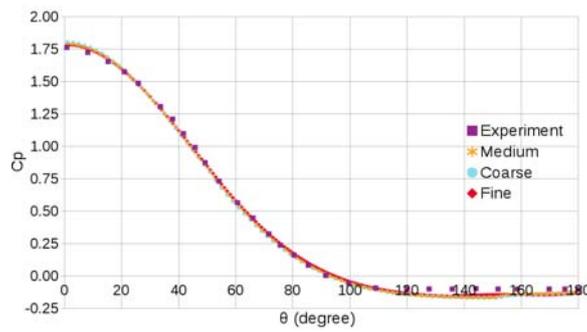


Figure 6. Pressure coefficient variation over the surface of the cylinder

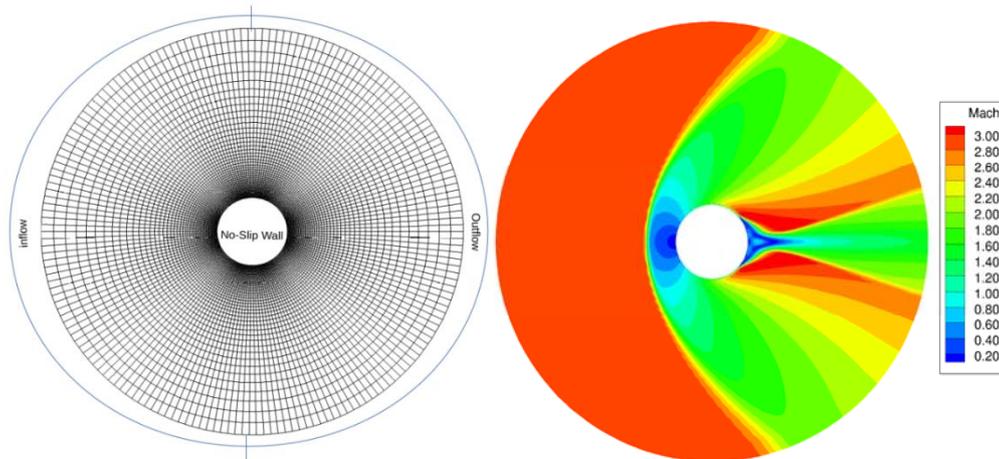


Figure 7. Mesh and Mach contours of the cylinder

4.3. The Shock Tube Problem

The third benchmark problem studied here is the shock tube problem that is also known as Sod’s problem. Shock tube is a long tube with constant cross section, where a diaphragm separates the left and right halves that are filled with two different pressures of air. The gas in the left portion is at higher pressure than the one on the right.

The computations are performed using two different meshes, course and fine meshes that have 100 and 400 cells in horizontal direction, respectively. Both meshes consist of 3 cells in vertical direction. The boundary and initial conditions used in the solution of the pressure driven inviscid flow are shown in Figure 8 and tabulated in Table 4, respectively. The length of the tube is 10 m.

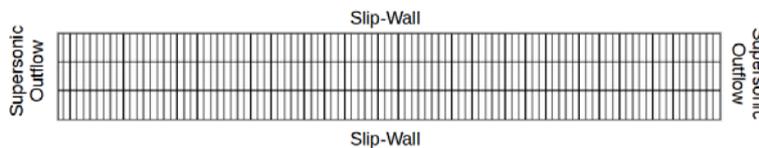


Figure 8. Domain of shock tube problem

Table 4. Initial and boundary conditions of the Riemann problem

	Left	Right
Density (kg/m^3)	1.0	0.125
Temperature (K)	348.4	278.7

The obtained density distributions along the horizontal direction at $t = 20.8 \text{ ms}$ are plotted in Figure 9 and also compared with the numerical result of [24] and the analytical solution of [25]. In addition, the density variation in time at a station, $x = 0.9 \text{ m}$ can be seen in Figure 9b.

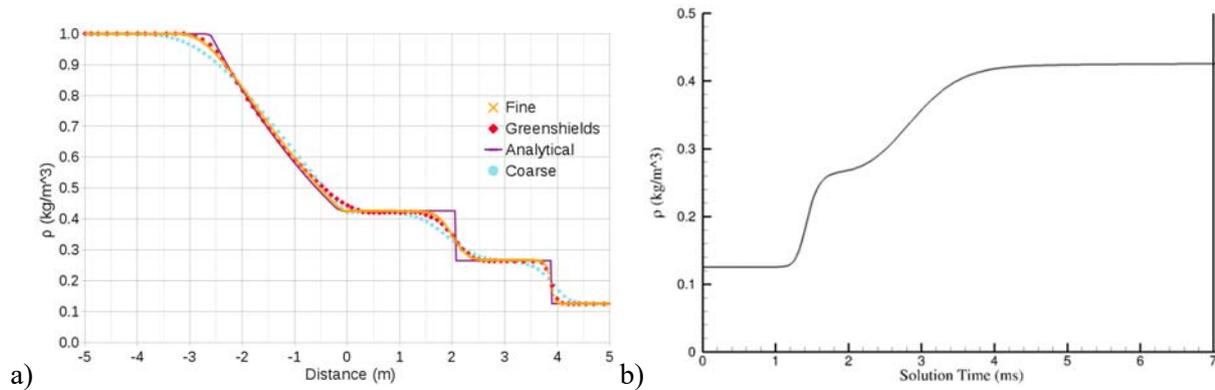


Figure 9. Density distribution along the shock tube a) and variation of density in time at $x=0.9 \text{ m}$ b)

As can be seen from the figure, as the resolution of the mesh is increased the obtained distribution matches the reference study better. On [24], second order semi-discrete, non-staggered schemes of Kurganov and Tadmor [26] with Minmod limiter is used.

4.4. Unsteady Flow on Forward Facing Step Problem

This problem was introduced by Woodward and Colella for benchmarking of Euler solvers [27]. A freestream at Mach 3 approaches a step as shown in Figure 10. A detached curve shock form in front of the step and reflects from the first top then bottom walls. Geometrical details and boundary conditions are given also in Figure 10.

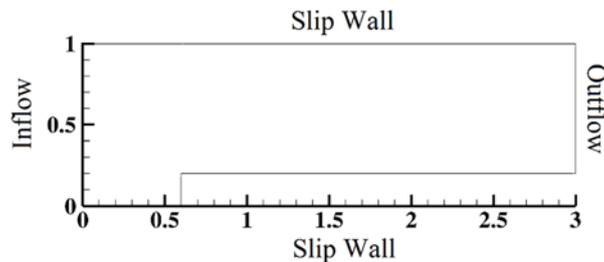


Figure 10. Geometry used in forward facing step problem

The domain is 3 m long and 1 m high. The step with the height of 0.2 m is located at $x = 0.6 \text{ m}$. A cartesian meshes with two different cell sizes of Δh are generated and used computations. Cell sizes and total element numbers are listed in Table 5. The walls of the channel are assumed to be adiabatic and freestream conditions are set also as initial conditions, where $M = 3.0$, $p = 101325 \text{ Pa}$, and $T = 300 \text{ K}$.

Table 5. Number of elements used in forward facing step problem

Case	Total Cell	Δh
1	16064	0.0125
2	64128	0.00625

The obtained results are shown and compared with the reference study of Cockburn and Shu [28] in Figure 11 at time $t = 34.56 \text{ ms}$. In the figure, there are 30 linearly distributed density contour lines in the range of 0.090338 to 6.2365.

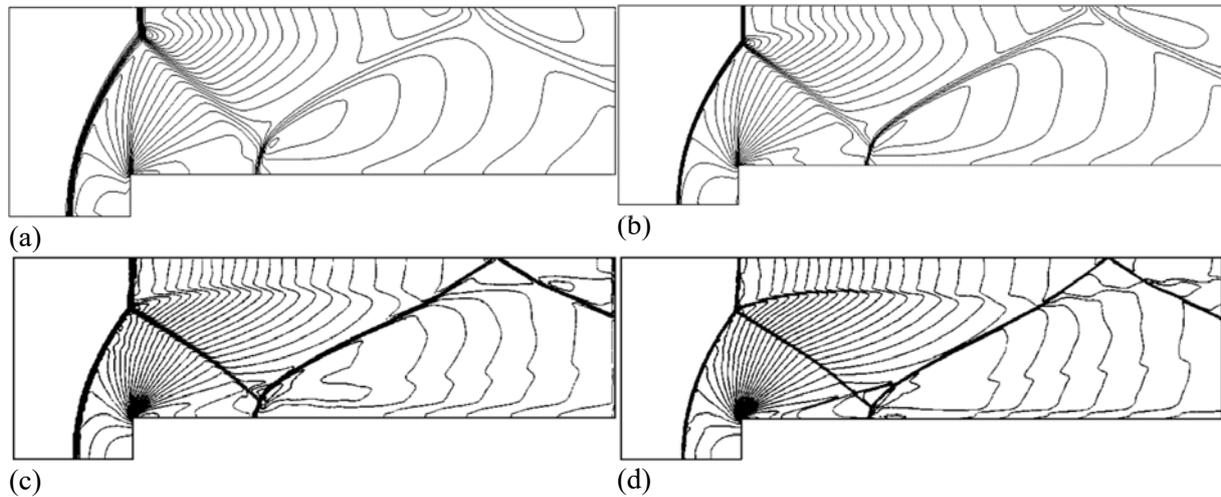


Figure 11. Comparisons of the obtained density contours (a) and (b) with Cockburn and Shu's [28] result (c), with $\Delta h = 0.0125$ and (d) $\Delta h = 0.00625$, respectively.

Cockburn and Shu uses a higher order method called Runge Kutta Discontinuous Galerkin method with local slope limiter [28]. The solver presented here is only first order in both time and space without using any limiter. Therefore, the discrepancies in the results are expected. On the other hand, qualitative comparisons of the results reveal that the developed solver is able to capture the shock with right location and strengths.

4.5. Acceleration of the Code

In order to check OpenMP implementation, the Double Wedge Problem is revisited here. The problem is solved performing both sequential and parallel runs on a laptop with 8 GB RAM and four CPU at 2.40 Ghz.

For parallel run, the computations were performed on 4 threads with a mesh of 101442 cells. The solver time is recorded for 1000 iterations. The average time per iteration is equal to 0.25680 s for a sequential run. The average time reduces to 0.06916 s for parallel run. Effect of running the code parallel in terms of relative speed is found to be 3.7128.

5. CONCLUSION

Recently developed, an unsteady, 2-D compressible Navier-Stokes solver that is based on finite volume method is introduced here. The solver reads mesh data, initial conditions, types of boundary conditions and desired numerical scheme from multiple text files. The solver utilizes upwind scheme of Van Leer's Flux Vector Splitting Scheme, and it is first order accurate in both time and space. It has local time stepping feature for fast convergence, and can run parallel on computers with shared memory architecture.

The solver is designed in such a way that other type of schemes can be integrated to the solver or the schemes in use can be modified for higher accuracy. In addition, various benchmark problems are solved to show its performance. Comparisons show that the obtained results are promising. Improving the current capabilities of the code to handle 3-D problems with higher accuracy is a must.

NOMENCLATURE

<u>Symbols</u>	<u>Description</u>	<u>Symbols</u>	<u>Description</u>
c	specific heat	ΔS	length of a face in a control volume
CFL	Courant–Friedrichs–Lewy condition	Δt	time step
e	Energy	θ	sum of the conduction and work done by viscous stresses
\vec{F}	flux vector	μ	dynamic viscosity
$\hat{i}, \hat{j}, \hat{k}$	unit vectors in Cartesian coordinates	ρ	Density
k	thermal conductivity	τ	shear stress
M	Mach number	Ω	area of a control volume
\vec{n}	unit normal vector	<u>Superscripts</u>	
p	Pressure	n	time level
Pr	Prandtl number	<u>Subscripts</u>	
R	specific gas constant	ab	From a to b
\vec{r}	a direction vector	c	Convective
Re	Reynolds number	n	n^{th} face in a control volume
T	Temperature	t	Total
u, v	velocity components in x and y direction	v	Viscous
U	an arbitrary scalar quantity	x	x direction in Cartesian coordinates
V	contravariant velocity	y	y direction in Cartesian coordinates
\vec{W}	conservative variables	b	Boundary
γ	specific heat ratio	e	Element
E, F	Fluxes in Cartesian coordinates	i	element id
ΔS	length of a face in a control volume		
Δt	time step		

REFERENCES

- [1] Nompelis I, Candler G, Holden M. Computational investigation of hypersonic viscous/inviscid interactions in high enthalpy flows. In: 36th AIAA Thermophysics Conference; 2003. pp. 3642.
- [2] Hu ZM, Myong RS, Wang C, Cho TH, Jiang ZL. Numerical study of the oscillations induced by shock/shock interaction in hypersonic double-wedge flows. Shock Waves 2008; 18(1): 41–51. doi: 10.1007/s00193-008-0138-x.
- [3] Lax PD. Weak solutions of non-linear hyperbolic equations and their numerical approximations. Communications on Pure and Applied Mathematics, 7 (1954), pp. 159-193.
- [4] Lax P, Wendroff B. Systems of conservation laws. Communications on Pure and Applied Mathematics, 13 (1960), pp. 217-237.
- [5] MacCormack RW. The effect of viscosity in hypervelocity impact cratering. AIAA paper 69-0354, April–May 1969.
- [6] Godunov SK. Finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. Matematicheskii Sbornik, 47 (1959), pp. 271-306.

- [7] Rusanov VV. On difference schemes of third order accuracy for nonlinear hyperbolic systems. *Journal of Computational Physics*, 5 (3) (1970), pp. 507-516.
- [8] Turkel E, Abarbanel S, Gottlieb D. Multidimensional difference schemes with fourth-order accuracy. *Journal of Computational Physics*, 21 (1) (1976), pp. 85-113.
- [9] Kostoff RN, Cummings RM. Highly cited literature of high-speed compressible flow research. *Aerospace Science and Technology*, 26(1), 2013, pp. 216-234.
- [10] Yee HC. On symmetric and upwind TVD schemes, NASA TM 88325 1986.
- [11] Roe PL. Characteristics-based schemes for the Euler equations *Annual Review of Fluid Mechanics*, 18 (1986), pp. 337-365.
- [12] Liu XD, Osher S, Chan T. Weighted essentially non-oscillatory schemes. *J Comp Phys* 1994; 115(1): 200-212.
- [13] Serna S, Marquina A. Power ENO methods: a fifth-order accurate weighted power ENO method. *J Comp Phys* 2004; 194(2): 632-658.
- [14] Liou MS, Steffen CJ. A new flux splitting scheme. *J Comp Phys* 1993; 107: 23-39.
- [15] Liou MS. A sequel to AUSM, Part II: AUSM+_{up} for all speeds. *J Comp Phys* 2006; 214(1): 137-170.
- [16] Yee HC, Klopfer GH, Montagne JL. High-resolution shock-capturing schemes for inviscid and viscous hypersonic flows. *J Comp Phys* 1990; 88(1): 31-61.
- [17] Hoffman K, Chiang ST. *Computational Fluid Dynamics. Volume II*, 4th ed. Kansas, USA. 2000. pp 178.
- [18] McDonald PW. The computation of transonic flow through two-dimensional gas turbine cascades. In: *ASME International Gas Turbine Conference and Products Show*; 1971; Texas, USA. doi: 10.1115/71-GT-89.
- [19] Barth T. Numerical aspects of computing high Reynolds number flows on unstructured meshes. In: *29th Aerospace Sciences Meeting*, 1991. pp. 721.
- [20] Gowen FE, Perkins EW. Drag of circular cylinders for a wide range of Reynolds numbers and Mach numbers. *National Advisory Committee for Aeronautics* 1953.
- [21] Blazek J. *Computational Fluid Dynamics: Principles and Applications*. 3rd ed. Oxford: Elsevier Ltd., 2009. pp. 75.
- [22] Jameson A. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. In: *10th Computational Fluid Dynamics Conference*; 1991. p. 1596.
- [23] Barney B. n.d. Introduction to Parallel Computing (EC3500). Retrieved March 1, 2017, from <https://computing.llnl.gov/tutorials/openMP/>.

- [24] Greenshields CJ, Weller HG, Gasparini L, Reese JM. Implementation of semi-discrete, non-staggered central schemes in a colocated, polyhedral, finite volume framework, for high-speed, viscous flows. *International Journal for Numerical Methods in Fluids* 2009; 601–629. doi: 10.1002/flid.2069..
- [25] Anderson JD. *Modern Compressible Flow*. 3rd ed. New York: McGraw-Hill, 2003. pp. 202.
- [26] Kurganov A, Tadmor E. New High-Resolution Central Schemes for Nonlinear Conservation Laws and Convection–Diffusion Equations. *J Comp Phys* 2000; 160(1): 241–282.
- [27] Woodward P, Colella P. The numerical simulation of two-dimensional fluid flow with strong shocks. *J Comput Phys* 1984; 54(1): 115–173. doi:10.1016/0021-9991(84)90142-6.
- [28] Cockburn B, Shu CW. The Runge–Kutta Discontinuous Galerkin Method for Conservation Laws V. *J Comput Phys* 1998; 141(2): 199–224. doi:10.1006/jcph.1998.5892.