Fifth-Order Finite Difference Code for Shock-Obstacle Interaction

Zahid Hussain Shah *

In this project a fifth-order finite difference code is developed in FORTRAN for simulating shock-obstacle interactions using the Euler equations using a fifth-order Weighted Essentially Non-Oscillatory (WENO) scheme for spatial discretization combined with a third-order Runge-Kutta method for time integration. Component-wise WENO reconstruction with Lax-Friedrichs flux splitting is used to handle nonlinear fluxes. The code is first validated against analytical solutions of shock-tube problems, demonstrating good agreement in capturing discontinuities and high-gradient regions. The validated code is then applied to study shock wave interactions an unconfined square obstacle and super-sonic flow over a confined high-aspect-ratio obstacle in a channel. Results show the code successfully captures key flow features including shock reflection, diffraction, formation of bow shock, and Mach stem formation. The code does not use any external libraries is written from scratch in FORTRAN.

Nomenclature

ρ	=	density
u	=	x-velocity component
v	=	y-velocity component
p	=	pressure
E	=	total energy per unit volume
e	=	specific internal energy
γ	=	specific heat ratio
c	=	speed of sound
α	=	maximum eigenvalue of Jacobian matrix
Δt	=	time step
Δx	=	grid spacing in x-direction
Δy	=	grid spacing in y-direction
CFL	=	Courant-Friedrichs-Lewy number

*Mechanical Engineering, Purdue University

I. Introduction

Shock waves in fluid are common phenomenon and are extensively studied using numerical method, since experiment are infeasible if not practically impossible [1]. However, shock waves are governed by hyperbolic system of equations pose a problem: one needs to decide between the order-of-accuracy and stability of the solution. Higher-order algorithm produces oscillations, leading to 'explosion' of the solution. Lower-order solutions like first-order upwinding are too diffusive and do not capture the important feature of the shock waves [2].

In this project, we use a high-order finite difference code in FORTRAN was developed for simulating shock-obstacle interactions using the Euler equations. The numerical method employs a fifth-order Weighted Essentially Non-Oscillatory (WENO) scheme for spatial discretization combined with a third-order Runge-Kutta method for time integration [3]. Component-wise WENO reconstruction with Lax-Friedrichs flux splitting is used to handle the nonlinear fluxes.

The code is first validated against analytical solutions of standard shock-tube problems to demonstrate its capability in capturing discontinuities and high-gradient regions. We then apply the code to study free shock wave interaction with an unconfined square obstacle and shock formation in super-sonic flow around a confined high-aspect-ratio obstacle in a channel.

II. Governing Equations

The Euler equations in conservation form in two-dimensional cartisian system for an inviscid flow are given as follows:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0 \tag{1}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(E+p) \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(E+p) \end{bmatrix} \tag{2}$$

Here ρ is the density, u, and v are the velocity components, p is the pressure, and $E = \rho e + \frac{\rho}{2}(u^2 + v^2)$ is the total energy per unit volume. We close the system by using ideal-gas equation relating specific internal energy e with thermodynamic variables as

$$e = \frac{p}{\rho(\gamma - 1)} \tag{3}$$

Here specific heat ratio $\gamma = 1.4$ assumed to be constant.

III. Numerical Method

System of equations 1 is solved using Finite Difference method using component-wise 5-th order WENO schemes. For clarity we define the scheme for one-dimesional Euler equations (extending to higher dimensions is trivial).

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \tag{4}$$

Using point values $\mathbf{U_i}$, we can get numerical solution as

$$\frac{\partial \mathbf{U_i}}{\partial t} + \frac{\mathbf{\hat{F}_{i+1/2}} - \mathbf{\hat{F}_{i-1/2}}}{\Delta x} = 0 \tag{5}$$

$\mathbf{\hat{F}_{i\pm1/2}}$ is the numerical flux at locations $x_{i\pm1/2}$. And $\Delta x = x_{i+1/2} - x_{i-1/2}$. Here we assume constant Δx. In order to do upwinding for linear stability, we do the flux splitting define by Lax-Friedrichs as follows

$$\mathbf{F} = \mathbf{F}^{+} + \mathbf{F}^{-} \tag{6}$$

where

$$\mathbf{F}^{\pm} = \frac{1}{2}(\mathbf{F} \pm \alpha \mathbf{U}) \tag{7}$$

Here $\alpha = max|\lambda(\mathbf{A})|$ is the maximum eigenvalue of the Jacobian matrix $\mathbf{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}}$. For Euler equations $\alpha = |u| + c$ [2], where $c = \sqrt{\gamma p/\rho}$ is the sonic of sound. Genrally one can decide the range over which α is computed. Here we take α as the maximum of $|u| + c$ over the whole computation domain (even though more diffusive than using local values). Numerical flux is then

$$\mathbf{\hat{F}_{i+1/2}} = \mathbf{\hat{F}^{+}_{i+1/2}} + \mathbf{\hat{F}^{-}_{i+1/2}} \tag{8}$$

To get the numrical fluxes we use the WENO schemes given by [3].

A. WENO Reconstruction

Let say we want to get the numerical value for first compoent of flux vector $\mathbf{F}$, and the first component be denoted by symbol f (to use same notation as [3]). Then using flux splitting we have

$$f^{\pm} = \frac{1}{2}(f \pm \alpha u) \tag{9}$$

The numerical $\hat{f}^{+}_{i+1/2}$ is obtained using the following procedure. Let (to follow same notation as [3])

$$\bar{v} \equiv f^{+} \tag{10}$$

We get for a fifth-order scheme

$$v^r_{i+1/2} = \sum_{j=0}^{2} c_{rj}\bar{v}_{i-r+j}, \quad r = 0, 1, 2 \tag{11}$$

Then we get coefficients β by

$$\beta_0 = \frac{13}{12}(\bar{v}_i - 2\bar{v}_{i+1} + \bar{v}_{i+2})^2 + \frac{1}{4}(3\bar{v}_i - 4\bar{v}_{i+1} + \bar{v}_{i+2})^2 \tag{12}$$

$$\beta_1 = \frac{13}{12}(\bar{v}_{i-1} - 2\bar{v}_i + \bar{v}_{i+1})^2 + \frac{1}{4}(\bar{v}_{i-1} - \bar{v}_{i+1})^2 \tag{13}$$

$$\beta_2 = \frac{13}{12}(\bar{v}_{i-2} - 2\bar{v}_{i-1} + \bar{v}_i)^2 + \frac{1}{4}(\bar{v}_{i-2} - 4\bar{v}_{i-1} + 3\bar{v}_i)^2 \tag{14}$$

From these β values, we get non-linear weights as

$$\alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2}, \quad r = 0, 1, 2 \tag{15}$$

where $\epsilon = 10^{-6}$ and d_r are the linear weights given by

$$d_0 = \frac{1}{10}, \quad d_1 = \frac{3}{5}, \quad d_2 = \frac{3}{10} \tag{16}$$

Finally we get the normalized non-linear weights

$$\omega_r = \frac{\alpha_r}{\sum_{s=0}^{2} \alpha_s}, \quad r = 0, 1, 2 \tag{17}$$

and then the numerical flux

$$\hat{f}^+_{i+1/2} = \sum_{r=0}^{2} \omega_r v^r_{i+1/2} \tag{18}$$

Similar procedure is used to get $\hat{f}^-_{i+1/2}$. But this time we define

$$\bar{v} = f^- \tag{19}$$

Rest of the procedure as same as above, but we need to do one additional step

$$\bar{v}_j = \bar{v}_{-j} \quad j = -2, 2 \tag{20}$$

This is just reading the vector back-to-front. Usually literature does not describe this final step. But doing so can allow us to use us same subroutine developed for $\hat{f}^+_{i+1/2}$. This way one can get $\mathbf{F}^+_{i+1/2}$ and $\mathbf{F}^-_{i+1/2}$ by following the WENO reconstruction procedure for each individual component of **F**. Numerical flux is then simply given by the equation 8.

B. Time Discretization using RK3

For temporal discretization we use third-order Runge-Kutta (RK3) scheme. Let

$$\frac{\partial \mathbf{U}}{\partial t} = L(\mathbf{U}) \tag{21}$$

where $L(\mathbf{U})$ is the spatial discretization operator. Then third-order Runge-Kutta [3] is given by

$$\mathbf{U}^{(1)} = \mathbf{U}^n + \Delta t L(\mathbf{U}^n) \tag{22}$$

$$\mathbf{U}^{(2)} = \frac{3}{4}\mathbf{U}^n + \frac{1}{4}\mathbf{U}^{(1)} + \frac{1}{4}\Delta t L(\mathbf{U}^{(1)}) \tag{23}$$

$$\mathbf{U}^{n+1} = \frac{1}{3}\mathbf{U}^n + \frac{2}{3}\mathbf{U}^{(2)} + \frac{2}{3}\Delta t L(\mathbf{U}^{(2)}) \tag{24}$$

Here the time-step Δt is controlled by imposing CFL condition. Where CFL is defined by

$$cfl = \frac{\Delta t}{min(\Delta x, \Delta y)}(V + a) \tag{25}$$

Here Δx and Δy are the grid spacing, while V is the velocity magnitude. Sonic speed $a = \sqrt{\gamma p/\rho}$. Time-step is decided by the lowest cfl number in the computation domain.

IV. Validation against Shock-tube Problems

Code and the numerical schemes were validated against shock-tube problems from chapter 8 [2]. Comparison made in Figures 1 and 2 for validation shows that WENO schemes used captures the high-gradients regions quite well. There are still some osciallations, which can be reduced by using characteristic-wise schemes (as apposed to component-wise). Furthermore, diffusion can be further reduced by using better approximation for parameter α used for flux-splitting in equation 7.

Details of the validations are as,

A. Validation 1

Initial conditions for Sod shock-tube problem (case 1) are as follows:

$$(\rho_L, u_L, v_L, p_L) = (1.0, 0.0, 0.0, 1000)(\rho_R, u_R, v_L, p_R) = (1.0, 0.0, 0.0, 0.01)$$

Here left side (shocked condition) is assumed to be at $x < 0.5$ initially. Figure 1 shows the numerical solution at $t = 0.012$ against exact solution (using analytical code from [2]). Total 100 points are used in $x \in [0, 1]$, while $CFL = 0.9$ is used.

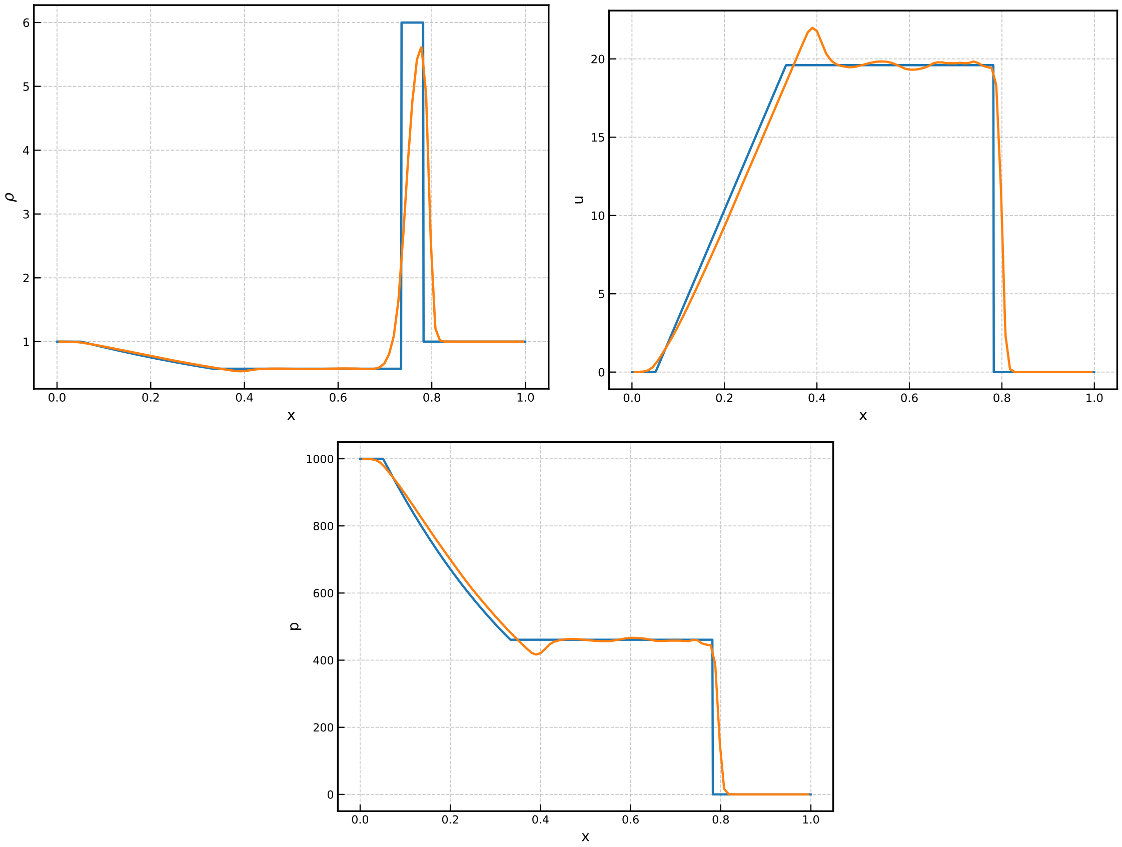

Fig. 1 Solution to the Sod shock-tube (case 1). Blue lines are the exact solution, while orange lines are the numerical solution

B. Validation 2

Initial conditions for Sod shock-tube problem (case 2) are as follows:

$$(\rho_L, u_L, v_L, p_L) = (1.0, 0.75, 0.0, 1.0)(\rho_R, u_R, v_L, p_R) = (0.125, 0.00, 0.0, 0.1)$$

Here left side (shocked condition) is assumed to be at $x < 0.3$ initially. Figure 1 shows the numerical solution at $t = 0.2$ against exact solution (using analytical code from toro). Total 100 points are used in $x \in [0, 1]$, while $CFL = 0.9$ is used.

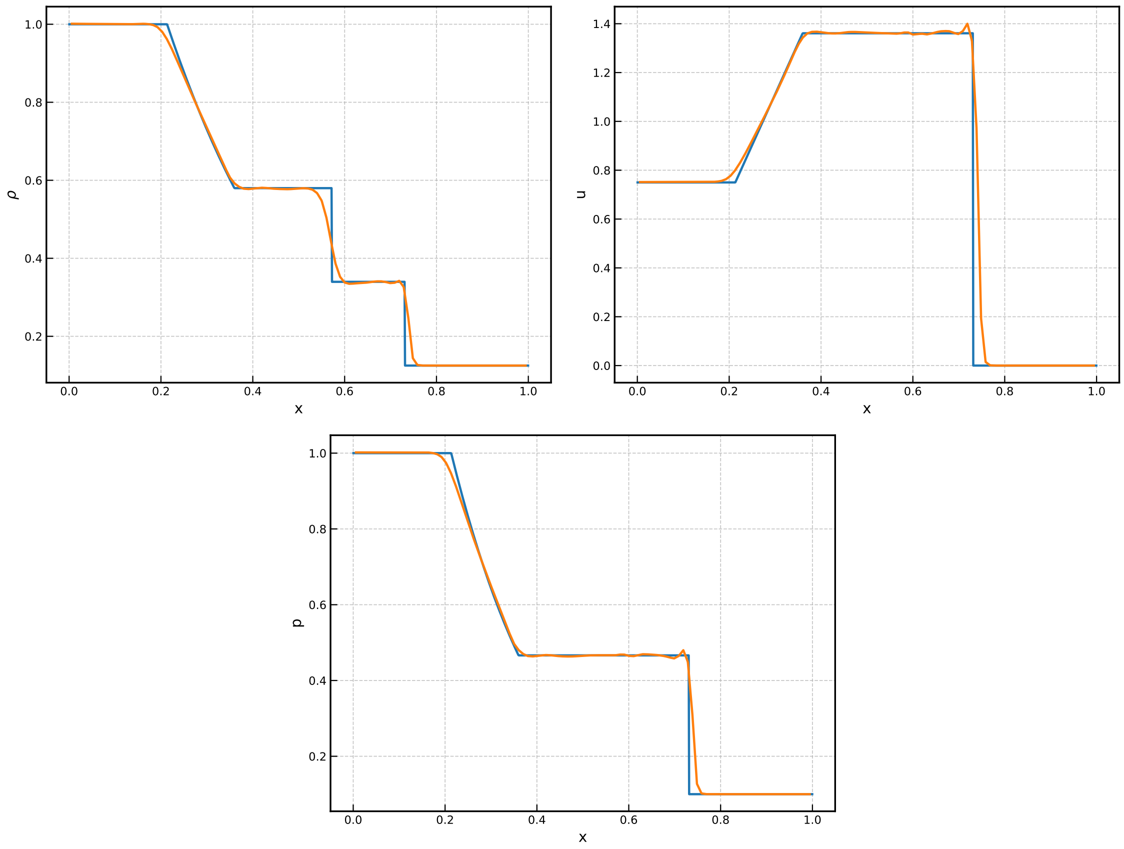

Fig. 2 Solution to the Sod shock-tube (case 2). Blue lines are the exact solution, while orange lines are the numerical solution

V. Results and Discussion

Here we show two different simulations results for shock interacting with solid obstacles. First simulations is to judge a free-shock interaction with and unconfined obstacle. Simulations should be able to capture key features like reflecting shock, diffracting shock, and Mach stem. Second simulation is for super-sonic flow over a confined obstacle. Key features for such flows are bow shock and reflection shocks.

A. Free Shock Interaction with a Square Obstacle

In this section we discuss shock-interaction with a solid square. A free shock is travelling from left-to-right. Initial conditions (same as case1 used for validation) are

$$(\rho_L, u_L, v_L, p_L) = (1.0, 0.0, 0.0, 1000)$$

$$(\rho_R, u_R, v_L, p_R) = (1.0, 0.0, 0.0, 0.01)$$

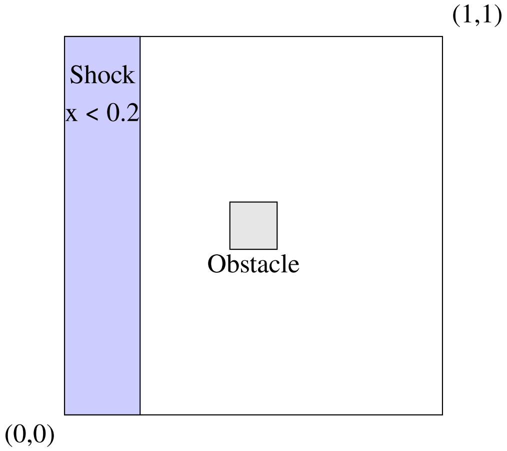

Fig. 3 Computation Domain used for free shock interacting with a square. Obstable is assumed to be at the domain center with dimesions of $\frac{1}{8} \times \frac{1}{8}$

All the the external boundaries are assumed to be Neumann boundaries with zero gradients.

$$\frac{\partial \mathbf{U}}{\partial \mathbf{n}} = 0 \quad \text{on external boundary} \tag{26}$$

One the obstacle boundary, Dirichlet no-slip boundary condition was imposed. Computational domain was descritized into 400 × 400 points and $CFL = 0.9$ was used.

Figure 4 shows contours of different variables at different moments (selected at moments between the moment before the shock arrives at the obstacle and shock interaction in the wake of the obstacle). Results show that the interaction is well captured. Along with the reflection for obstacle wall, interaction of the deflected shocks behind the obstacle can also be seen in the results.

Once the shock hits the obstacle, it and wraps around the obstacle. Resulting diffracting shocks meet at the region begind the obstacle to form a 'Mach stem' of 'Mach reflection.' On the front of the obstacle, however, a shock wave just gets reflected and moves backward with a negative velocity, as seen by the u contours.

B. Super-Sonic Flow Over an Obstacle in a Confined Channel

To judge the solution along the no-slip boundary, another simulation was performed with a high-aspect ratio obstacle in a confined channel. Super-sonic flow over a confined obstacle results in formation of bow shock and reflection shocks. This case can also be used to judge smearing of the shock wave along the no-slip boundary. This smearing is due to numerical diffusion from the algorithm.

Figure 5 shows the computation domain used for this case. Initially all the domain (excluding the obstacle) is given the values

$$(\rho, u, v, p) = (1.89, 948.68, 0.0, 100000) \tag{27}$$

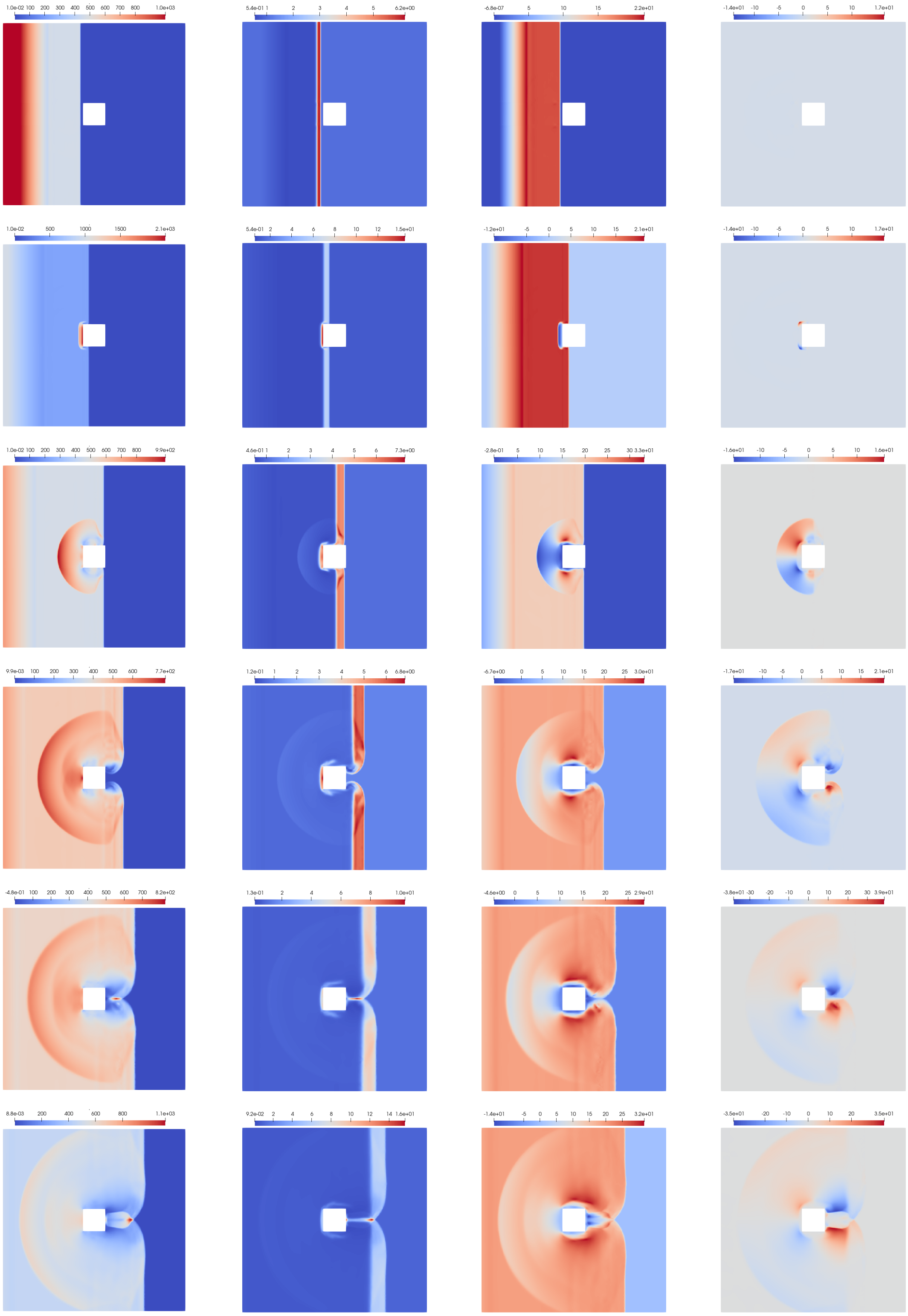

Fig. 4 From left to right: Pressure (p), density (ρ), x-velocity (u), and y-velocity (v) contours showing shock interaction with square obstacle at different times

All the units are S.I units. $u = 948.68 m/s$ responds to Mach number of 3. Left boundary is a uniform inlet velocity of $u = 948.68$, $v = 0$ and the right boudary condition is zero-gradient, while all the other boundaries are no-slip. Domain is descritized with 800×200 points in x and y directions, respectively. $CFL = 0.5$ was used for time-step calculuations.

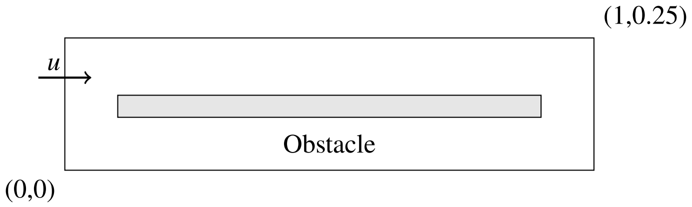

Fig. 5 Computation Domain used for shock interacting with an obstacle in a confined channel. Obstacle is placed at the center with dimensions of $0.8 \times \frac{1}{24}$

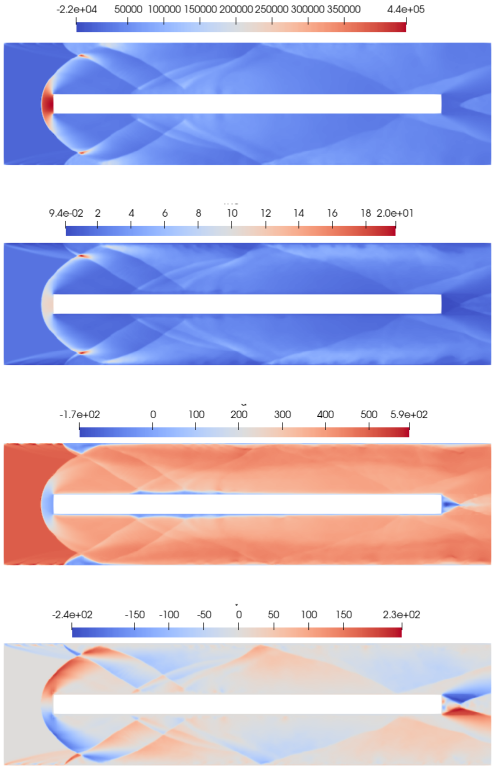

Fig. 6 p, rho, u, v **(from top to bottom) contours at different time instances showing shock-wave interaction with the obstacle in a channel. Results are shown for** $t = 3.8e^{-3} s$

Figure 6 shows that a bow shock is formed in front of the obstacle, while several shocks are formed between the obstacle and the side wall due to the shock reflection from solid (no-slip) boundaries.

Figure 6 shows that the algorithm handles the no-slip boundary conditions well. Density contours, however, show some smearing (over-diffusion) close to the upper and lower walls. This could be due to the fact that we are choosing α as maximum over whole computation domain. Since velocity (as $\alpha = |u| + c$) close to the walls is much lower than the maximum velocity in the computation domain, the regions close to the wall get relatively more diffusion due to flux-splitting. One can use local α based to redeme the problem but then it becomes a bit more tricky to code the boundary conditions (I tried local α but solution close to the boundary was not physical. Could have been a bug too).

VI. Conclusions

A fifth-order WENO finite difference code was successfully developed and validated for simulating shock-obstacle interactions using the Euler equations. The validation against shock-tube problems show a good agreement with analytical solutions, accurately capturing discontinuities and high-gradient regions. Furthermore, shock-obstacle interaction simulations, the code effectively resolved key flow features like shock reflection, diffraction, Mach stem formation, and bow shock appearance.

Overall, the algorithm shows a promising capability for simulation compressible, inviscid flows. Certain improvements can still be made, including implementation of local α for flux-splitting and implementation of characteristic-wise decomposition (although it can add significant computation cost).

References

[1] Song, H., and Wu, B., "Physics-based modeling and micro-burr removal mechanism analysis for laser-induced plasma deburring," *Journal of Manufacturing Processes*, Vol. 75, 2022, pp. 1217–1229.

[2] Toro, E. F., *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*, Springer Science & Business Media, 2013.

[3] Cockburn, B., Shu, C.-W., Johnson, C., Tadmor, E., and Shu, C.-W., *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, Springer, 1998.