INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

DEPARTMENT OF MECHANICAL ENGINEERING

ME683 - COMPUTATIONAL GAS DYNAMICS

TERM PROJECT II

PROF. TAPAN K. MANKODI

KHADAKE VARUN DILIP

214103001

M.TECH. - A&P

Problem Statement

Introduction

Sod shock tube is a standard Riemann problem for Euler's equation which is used to validate and test the accuracy of a computational gas dynamics method. A 1 m long shock tube is separated at the middle into two sections: driver section (L) and driven section (R). The initial conditions in the two sections are as follows:

$$\begin{bmatrix}\rho_L \\ P_L \\ u_L\end{bmatrix} = \begin{bmatrix}1.0 \\ 1.0 \\ 0.0\end{bmatrix}, \begin{bmatrix}\rho_R \\ P_R \\ u_R\end{bmatrix} = \begin{bmatrix}0.125 \\ 0.1 \\ 0.0\end{bmatrix}$$

At time t = 0, the diaphragm separating the two section breaks, and a shock wave moves in the right direction in the driven section, followed by a contact discontinuity and a rarefaction wave opening up in the left direction in the driver section. The pressure and velocity are constant across the contact discontinuity, and the variables vary continuously in the rarefaction wave.

Goal

The objective of the term project is to write a program (in any language of your choice ∈ [C, C++, Fortran, Matlab, Python]) and simulate 1D shock tube problem. Details:

- Simulation Type : 1D
- Domain size : 1 m ($0 \leq x \leq 1$)
- No of cells : 400 minimum
- CFL number : 0.8 (or $\Delta t = 0.8\Delta x/\max(|\lambda|)$)
- Total Simulation time : Refer to the table for each case
- Boundary Condition : Zero gradient at the two sides. (Flux = 0)
- Flux Solver : Roe's Approximate method with Entropy Fix

Cases to solve

Table 1: Data for five Riemann problem tests

Test	ρ_L	u_L	P_L	ρ_R	u_R	P_R	Time
1	1.0	0.0	1.0	0.125	0.0	0.1	0.25
2	1.0	-2.0	0.4	1.0	2.0	0.4	0.15
3	1.0	0.0	1000.0	1.0	0.0	0.01	0.012
4	1.0	0.0	0.01	1.0	0.0	100.0	0.035
5	5.99924	19.5975	460.894	5.99242	-6.19633	46.0950	0.035

The Approximate Riemann Solver of Roe

Godunov-type schemes and all its higher-order generalisations require the solution of a Riemann problem for every cell boundary at each time-step. The exact iterative Riemann solver of Gottlieb and Groth (1978) typically needs the computation of an initial guess solution for the Newton itera-tive method, the identification of the pattern of the solution and finally some Newton iterations. It isclearly a computationally expensive procedure. Moreover, most of the structure of the resulting so- lution is lost in the Godunov's method, due to the final cell-average operation on each grid-cell. This suggests that good numerical results could be obtained by calculating an appropriate approxi-mate solution instead of the exact one.

The second Riemann solver implemented in the present code gives a direct estimation of the inter- face-fluxes following the algorithm proposed by Roe. Roe's algorithm solves exactly a linearised problem, instead of looking for an iterative solution of the exact original Riemann problem. The ap-proximate solver proposed by Roe is much less expensive in terms of computational effort than the exact one, because the exact solution of a linear Riemann problem can be more easily built. The lin-earised problem takes the form:

$$\frac{\partial U}{\partial t} + \bar{A}\frac{\partial U}{\partial x} = 0 \qquad (56)$$

where $\bar{A} = \bar{A}_{i+\frac{1}{2}}(U_i, U_{i+1})$ is a constant matrix linearising the Jacobian $A(U)$ at the interface $i + \frac{1}{2}$ .Following Roe, the matrix $\bar{A}$ has to verify these conditions:

$F_{i+1} - F_i = \bar{A}_{i+\frac{1}{2}}(U_{i+1} - U_i)$
$\bar{A}_{i+\frac{1}{2}}$ is diagonalizable and all its eigenvalues are real
$\bar{A}_{i+\frac{1}{2}}(U, U) = A(U)$

These conditions ensure that the resulting numerical scheme is conservative and consistent with the original hyperbolic problem. It is also possible to show that a discontinuity (like a shock) is seen asa solution and exactly propagated by the solver.

Roe (1981) showed that in the case of the Euler equations it is always possible to obtain the linear-ised Jacobian matrix by simply computing it in an appropriate averaged state

$$U^{Roe} = \bar{U}(U_l, U_r) \tag{57}$$

In Eq. the index l and r represent a left and a right state given by some higher-order reconstruction technique. If only a first-order accurate scheme is required, we can set $l = i$ and $r = i + 1$.

Introducing the ratio

$$D_{i+\frac{1}{2}} = \sqrt{\frac{\rho_r}{\rho_l}} \tag{58}$$

the Roe's average for (ρ, u, H) is written as:

$$\bar{\rho}_{i+\frac{1}{2}} = \sqrt{\rho_l \rho_r} = D_{i+\frac{1}{2}} \rho_r \tag{59}$$

$$\bar{u}_{i+\frac{1}{2}} = \frac{(u\sqrt{\rho})_l + (u\sqrt{\rho})_r}{\sqrt{\rho_l} + \sqrt{\rho_r}} = \frac{D_{i+\frac{1}{2}} u_r + u_l}{D_{i+\frac{1}{2}} + 1} \tag{60}$$

$$\bar{H}_{i+\frac{1}{2}} = \frac{(H\sqrt{\rho})_l + (H\sqrt{\rho})_r}{\sqrt{\rho_l} + \sqrt{\rho_r}} = \frac{D_{i+\frac{1}{2}} H_r + H_l}{D_{i+\frac{1}{2}} + 1} \tag{61}$$

and the speed of sound is given by:

$$\bar{c}^2_{i+\frac{1}{2}} = (\gamma - 1)\left(\bar{H} - \frac{\bar{u}^2}{2}\right) \tag{62}$$

Finally, it can be easily shown that the flux estimation for the exact solution of the approximate lin-earised Riemann problem is given by Eq. which has been implemented in the code:

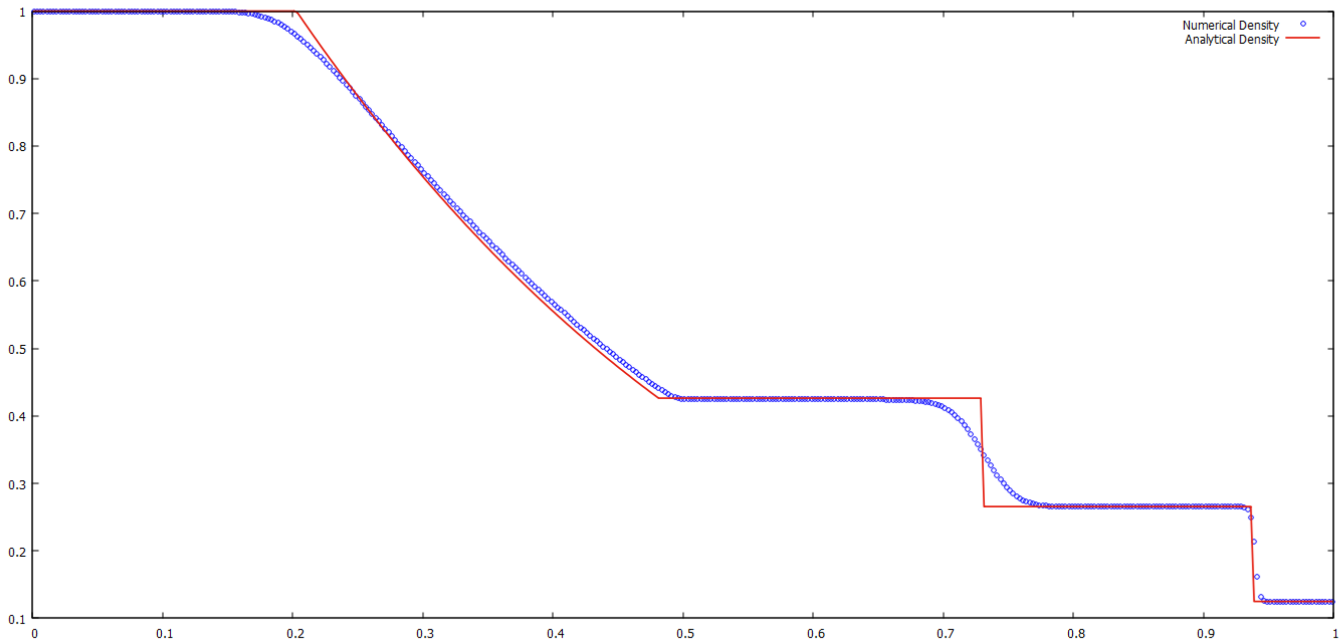
$$F_{i+\frac{1}{2}} = \frac{1}{2}[F(U_l) + F(U_r)] - \frac{1}{2}\bar{R}|\bar{\Lambda}|\bar{R}^{-1}(U_r - U_l) \tag{63}$$

where $\bar{R}^{-1}$ and $\bar{R}$ are the matrices diagonalizing the Jacobian A, $\mathbf{\Lambda}$ is the diagonalized Jacobian and $|\mathbf{\Lambda}|$ is the matrix which has on the diagonal the absolute values of the eigenvalues of A.

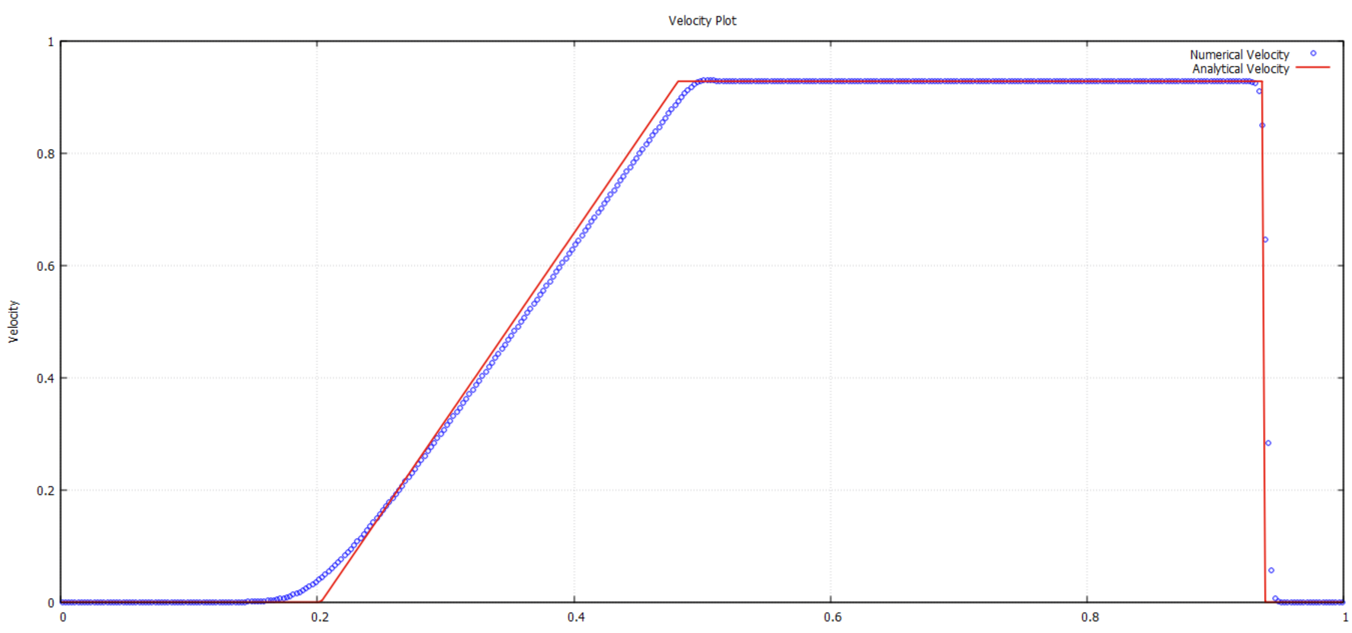
Plots for all Cases

1) Case 1:

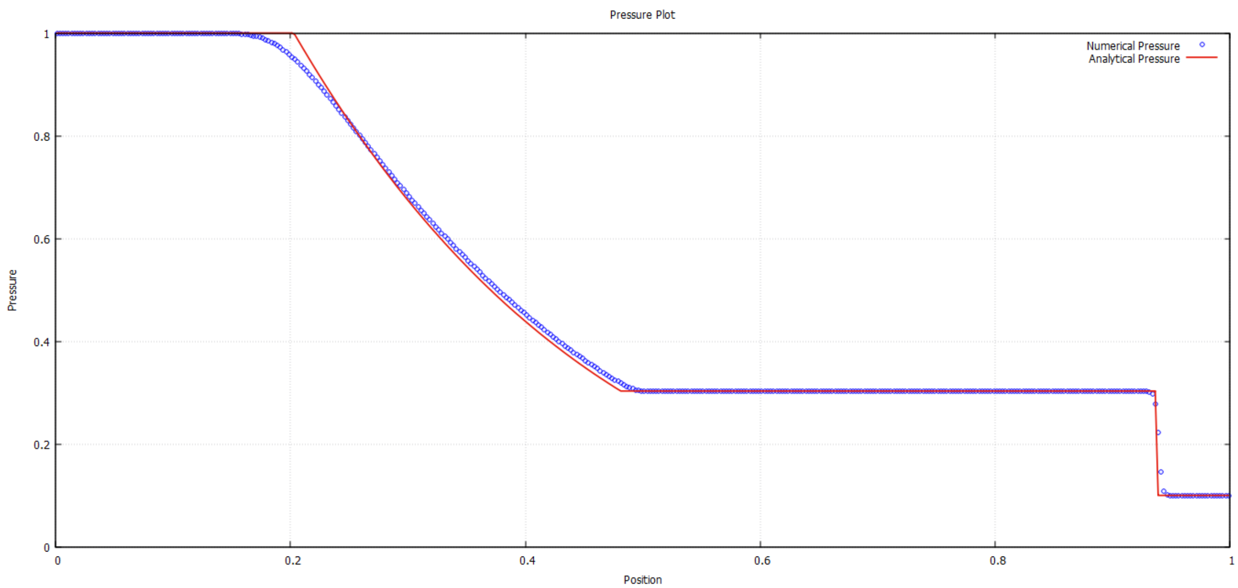
a) Density Plot



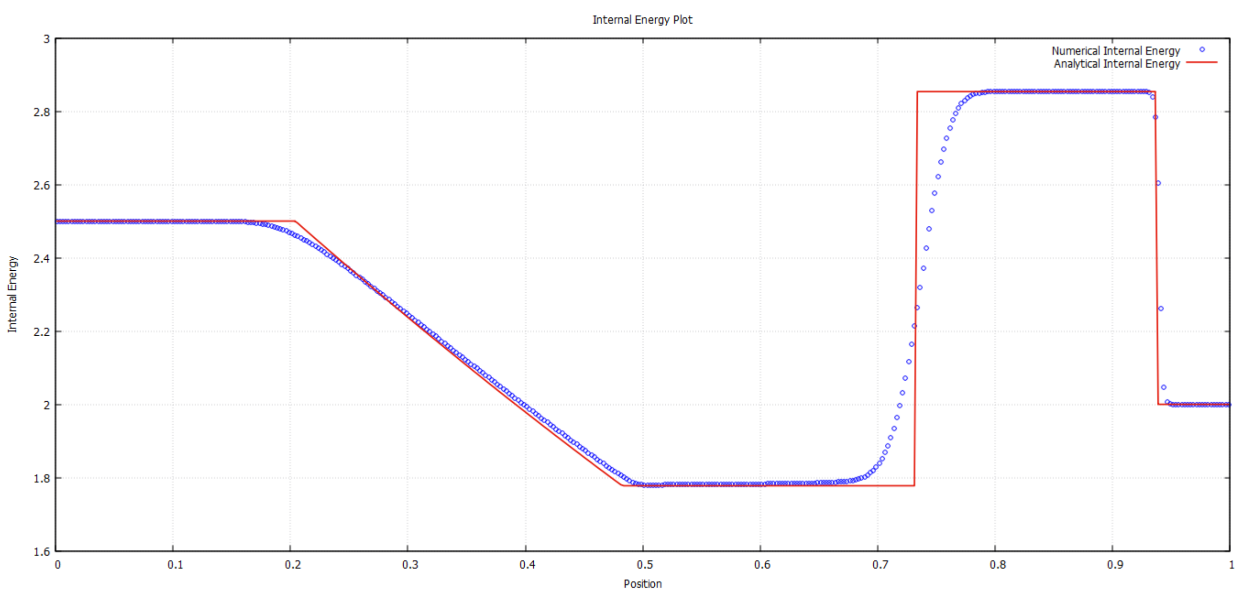
b) Velocity Plot



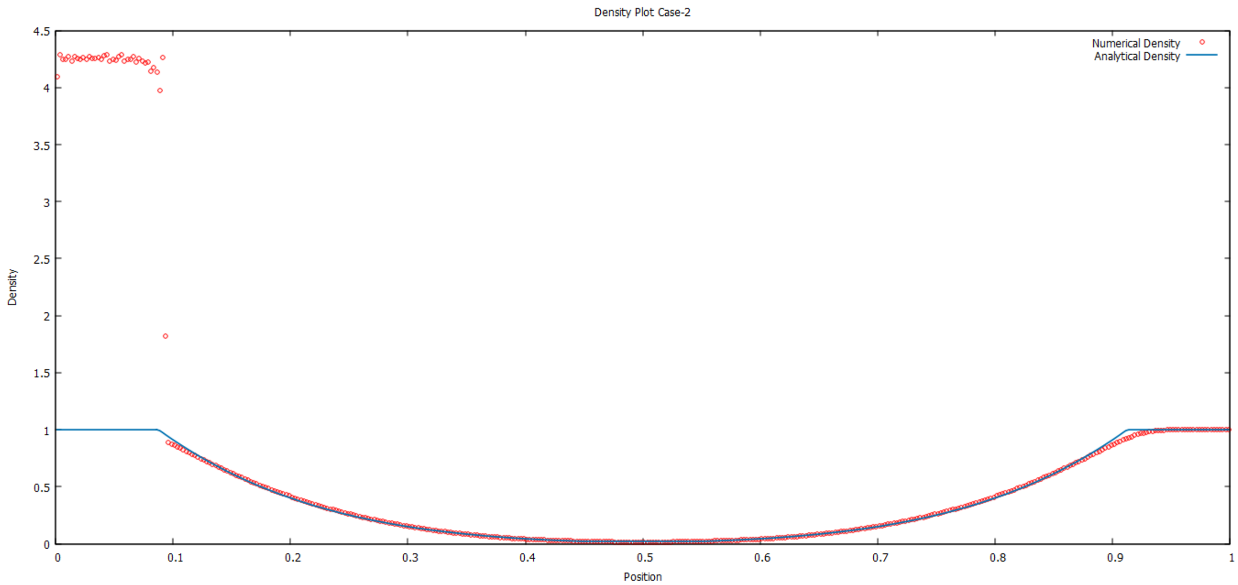
c)Pressure Plot



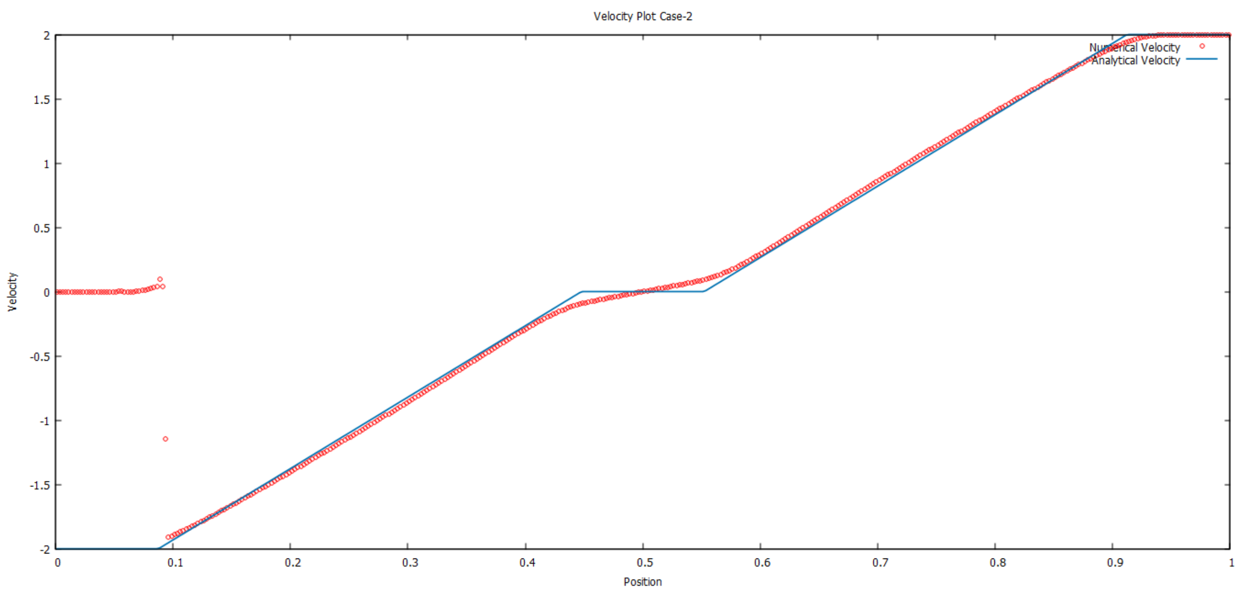
d) Internal Energy Plot



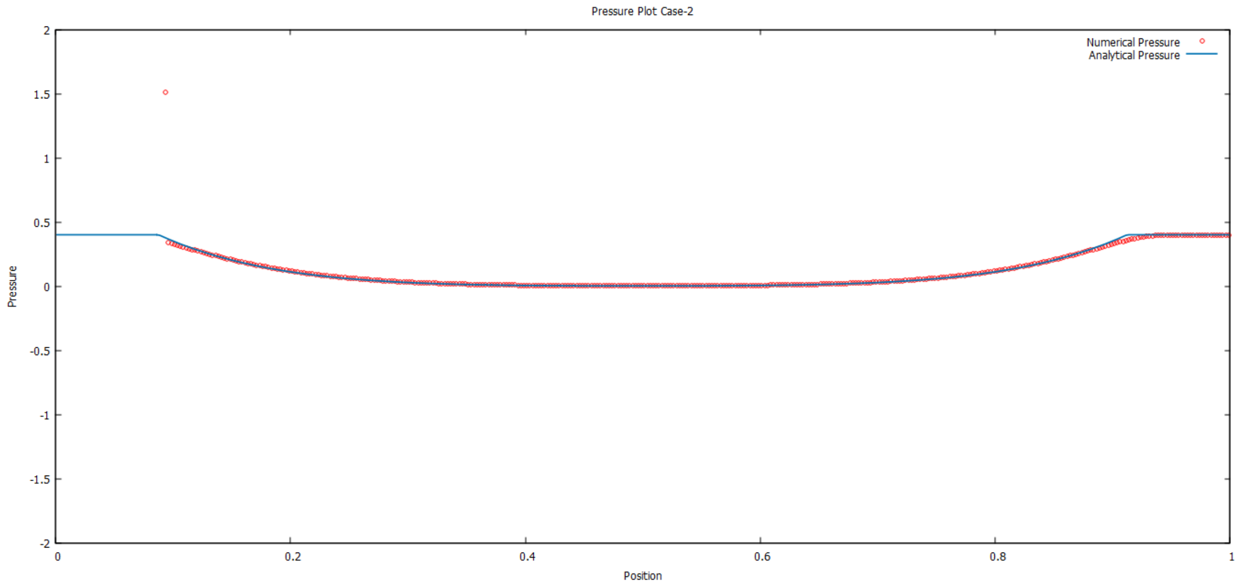
2) Case 2:

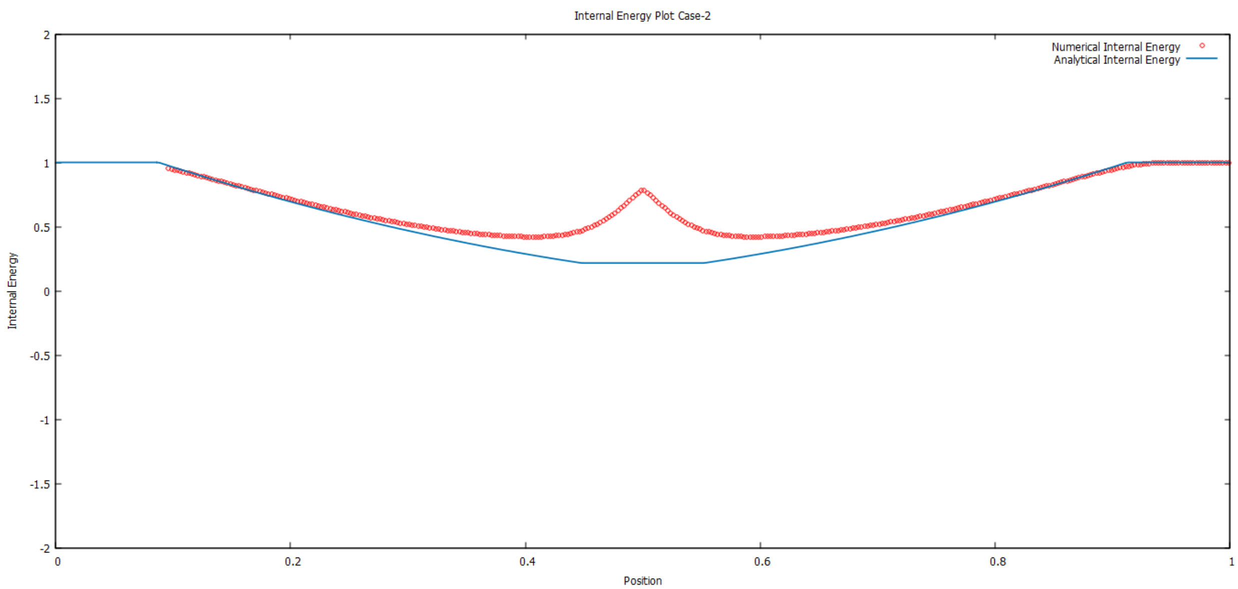


Density Plot Case-2
Numerical Density
Analytical Density
Density
Position

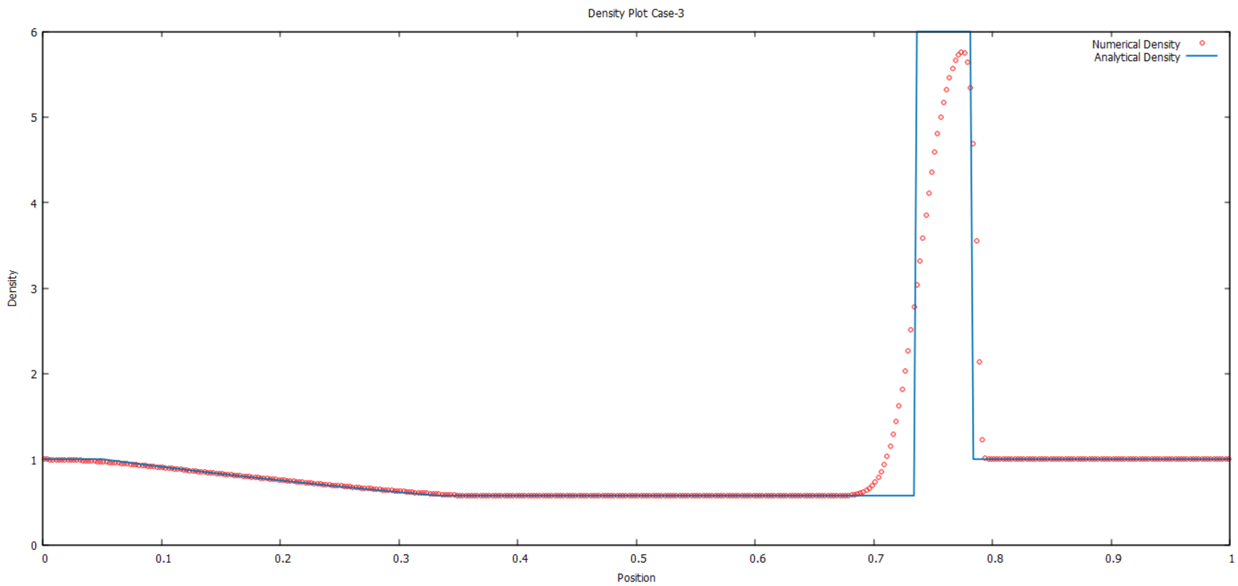


Velocity Plot Case-2
Numerical Velocity
Analytical Velocity
Velocity
Position

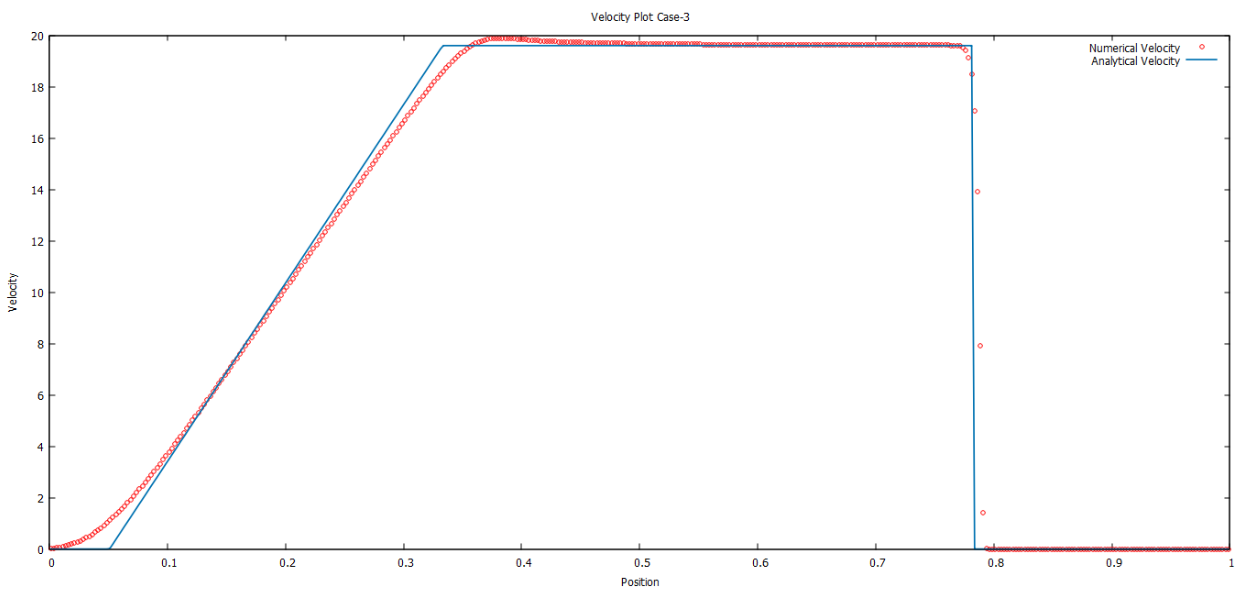

Pressure Plot Case-2
Numerical Pressure
Analytical Pressure
Pressure
Position
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
0.2
0.4
0.6
0.8
1


Internal Energy Plot Case-2
Numerical Internal Energy
Analytical Internal Energy
Internal Energy
Position
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
0.2
0.4
0.6
0.8
1

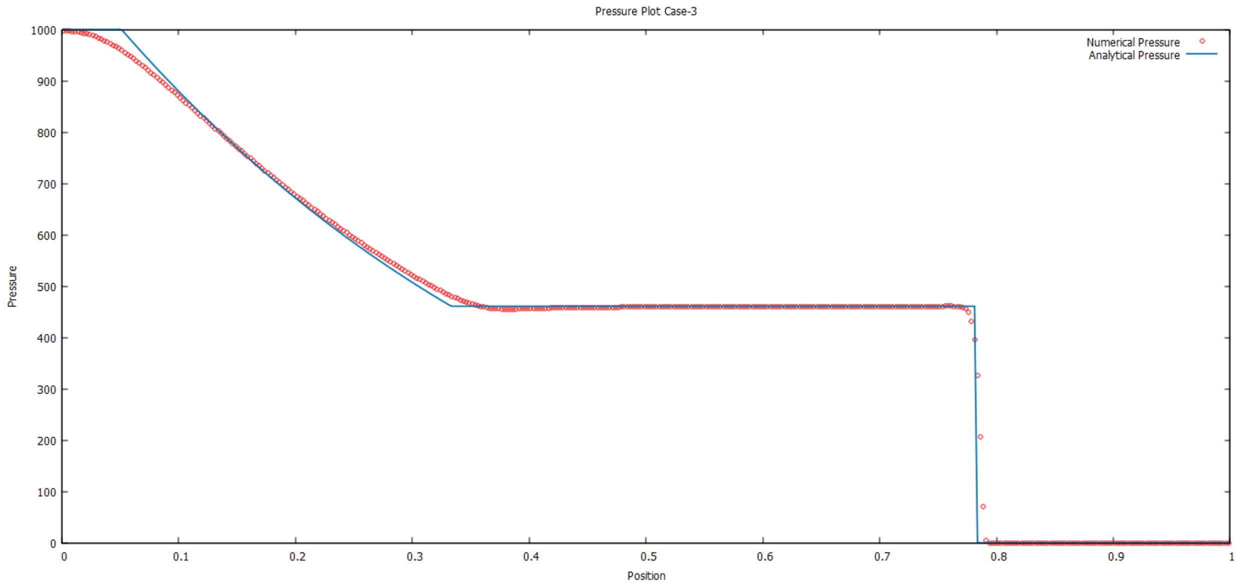
3) Case 3:

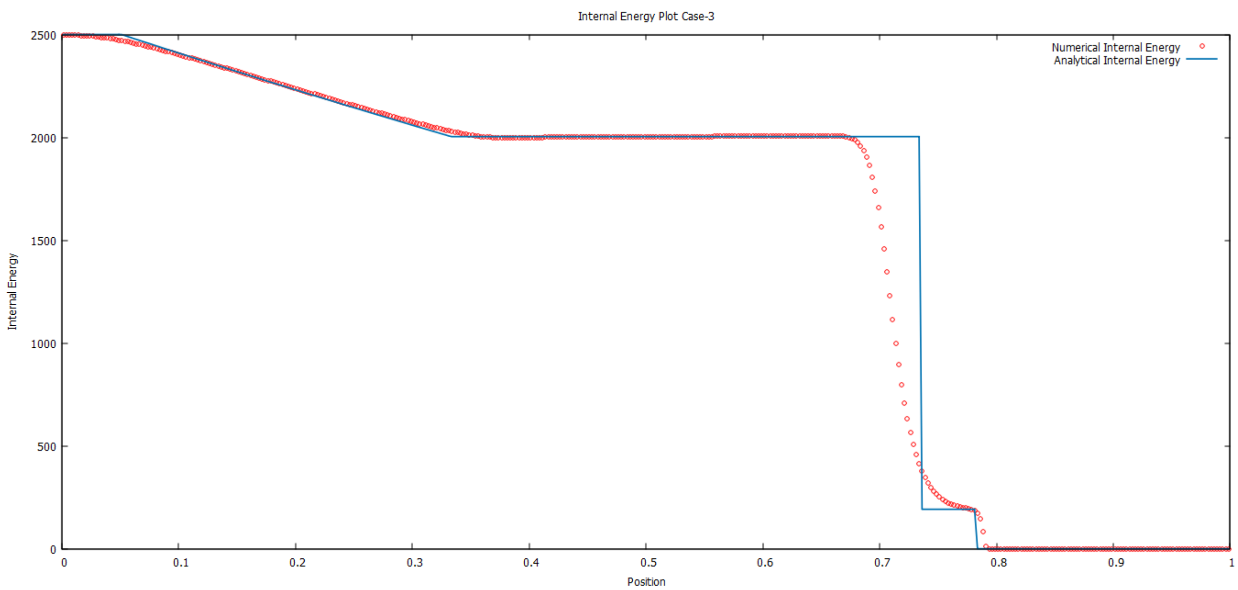


Density Plot Case-3
Numerical Density
Analytical Density
Density
Position

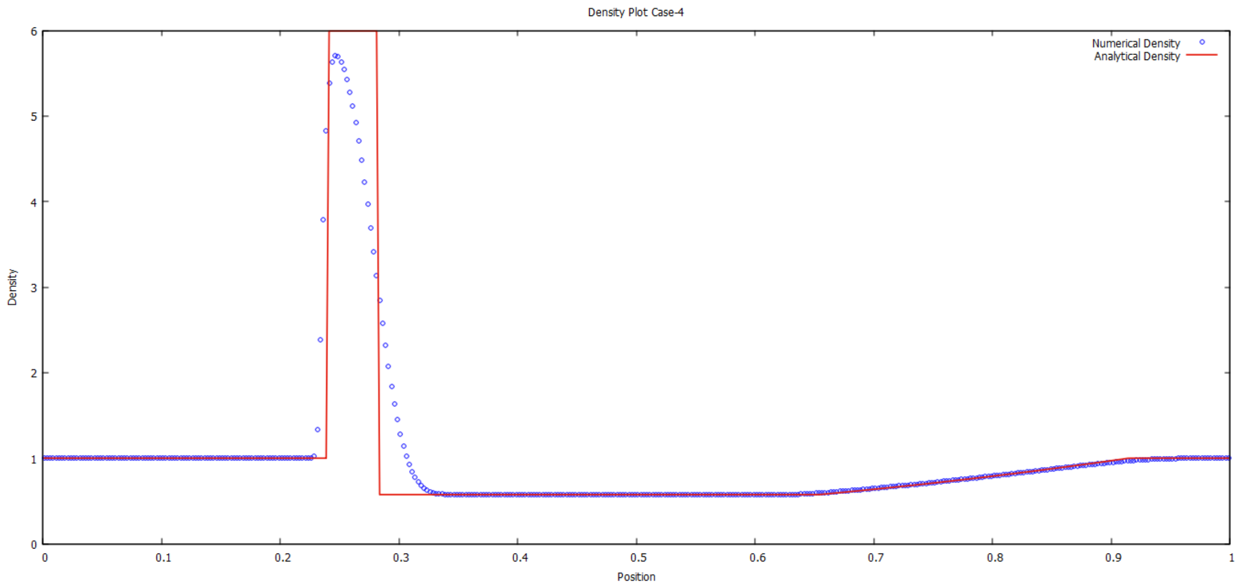


Velocity Plot Case-3
Numerical Velocity
Analytical Velocity
Velocity
Position

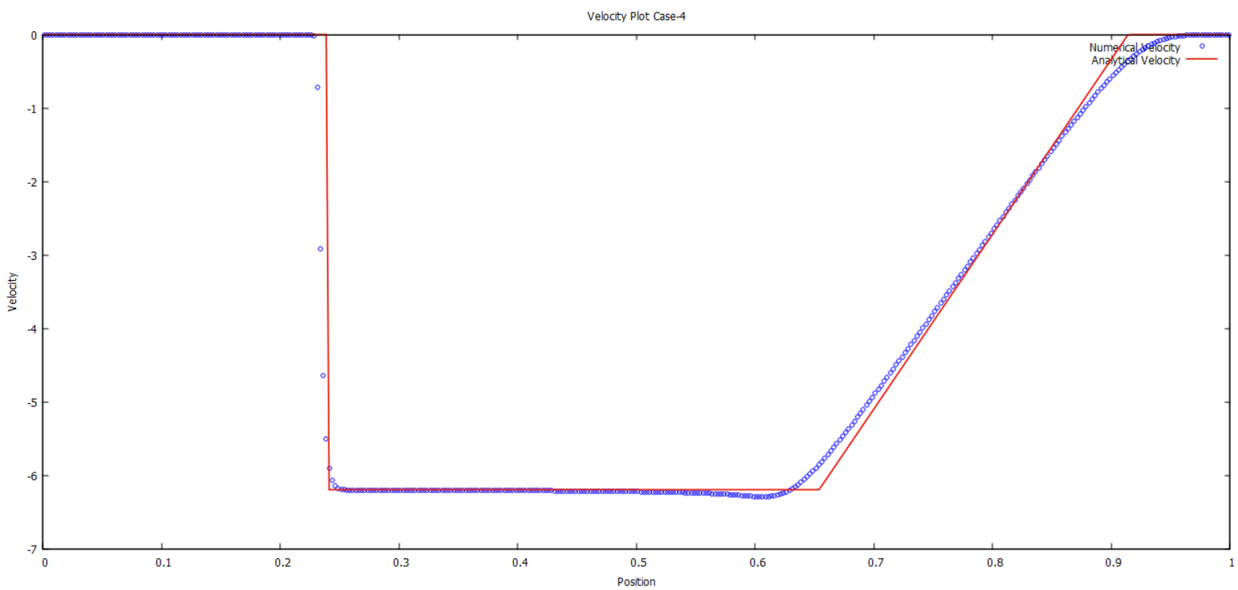

Pressure Plot Case-3
Numerical Pressure
Analytical Pressure
Pressure
Position
0
100
200
300
400
500
600
700
800
900
1000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Internal Energy Plot Case-3
Numerical Internal Energy
Analytical Internal Energy
Internal Energy
Position
0
500
1000
1500
2000
2500
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

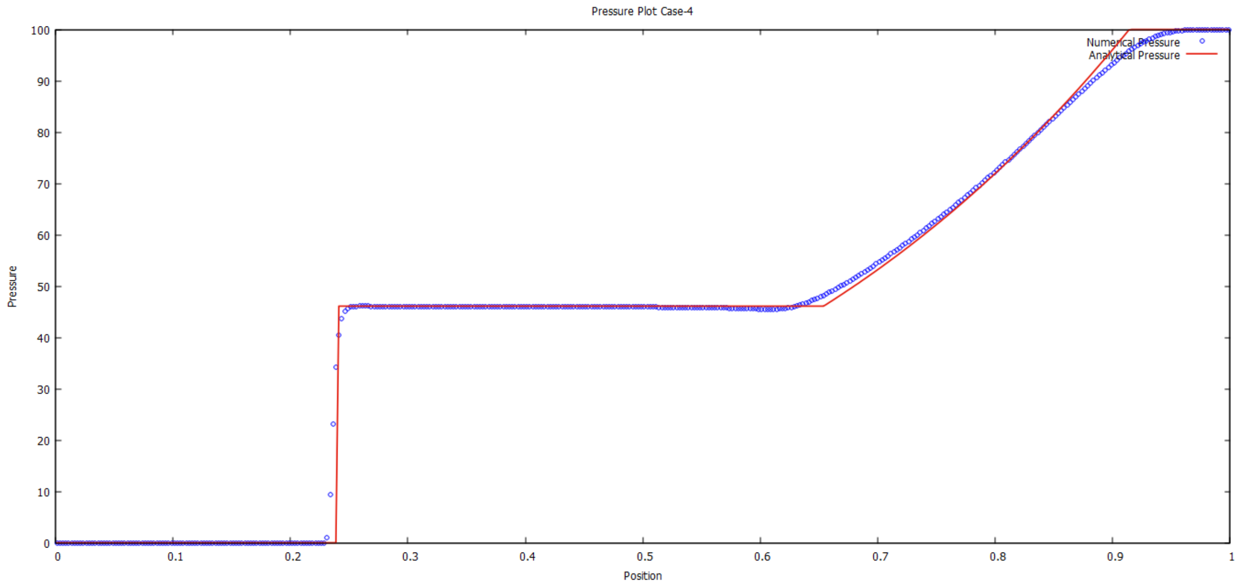
4) Case 4:

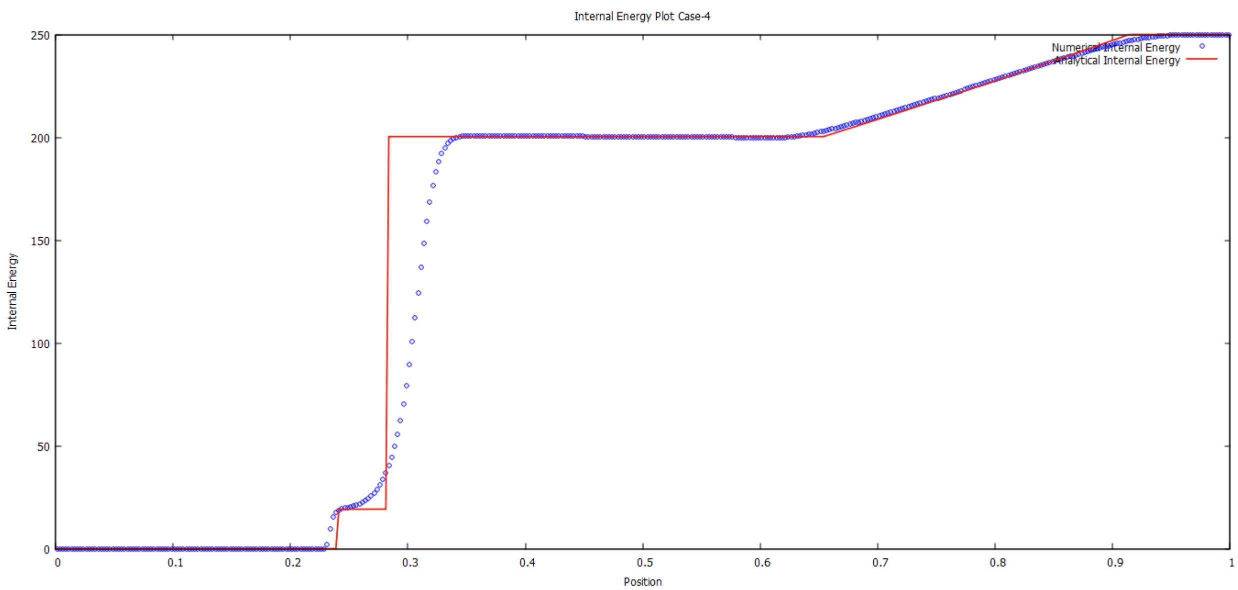


Density Plot Case-4
Numerical Density
Analytical Density
Density
Position

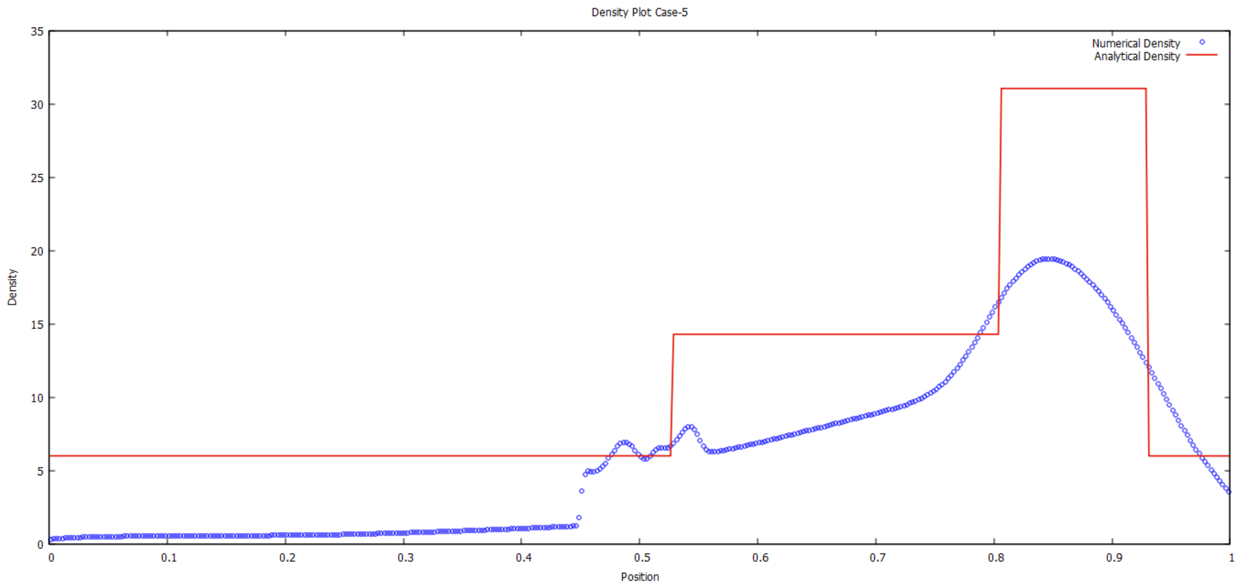


Velocity Plot Case-4
Numerical Velocity
Analytical Velocity
Velocity
Position

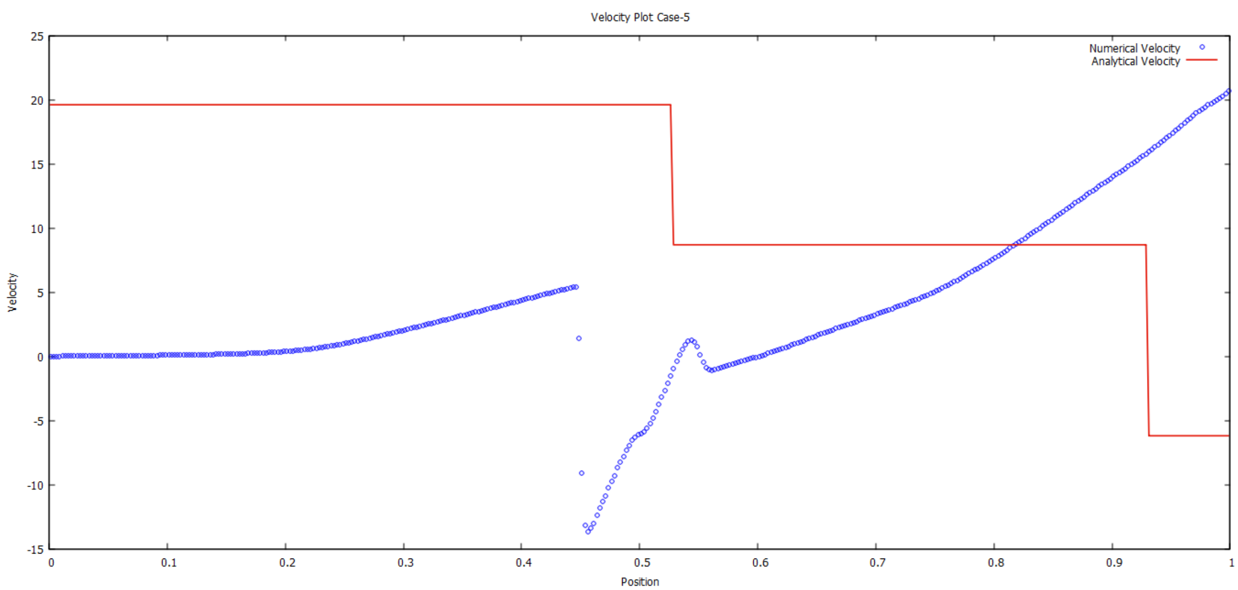

Pressure Plot Case-4
Numerical Pressure
Analytical Pressure
Pressure
Position
0
10
20
30
40
50
60
70
80
90
100
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Internal Energy Plot Case-4
Numerical Internal Energy
Analytical Internal Energy
Internal Energy
Position
0
50
100
150
200
250
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

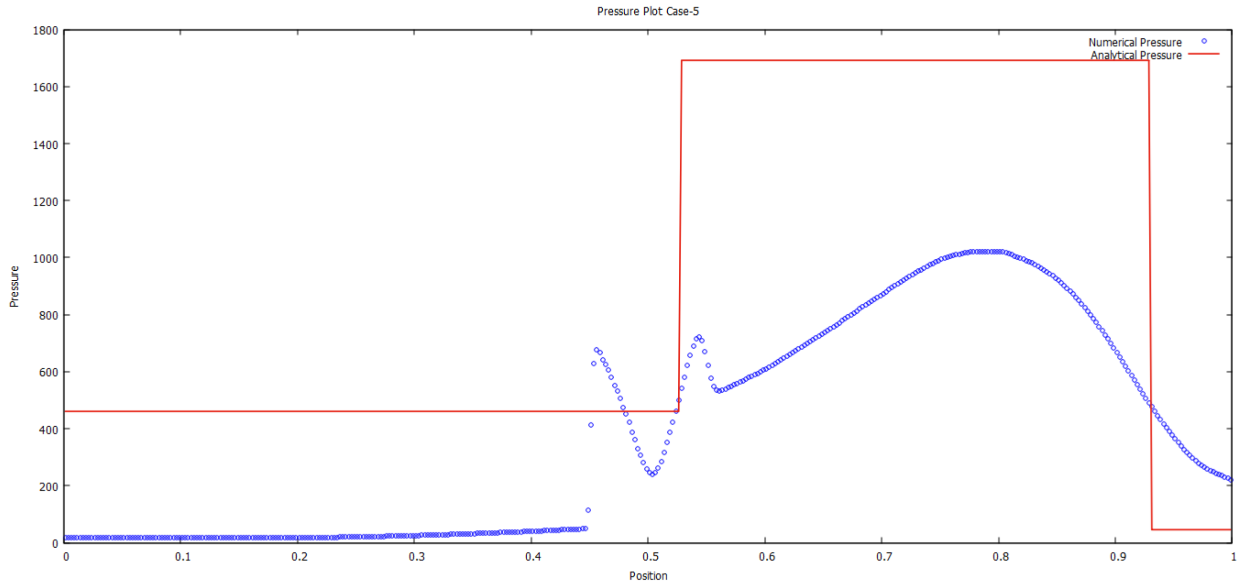
5) Case 5:

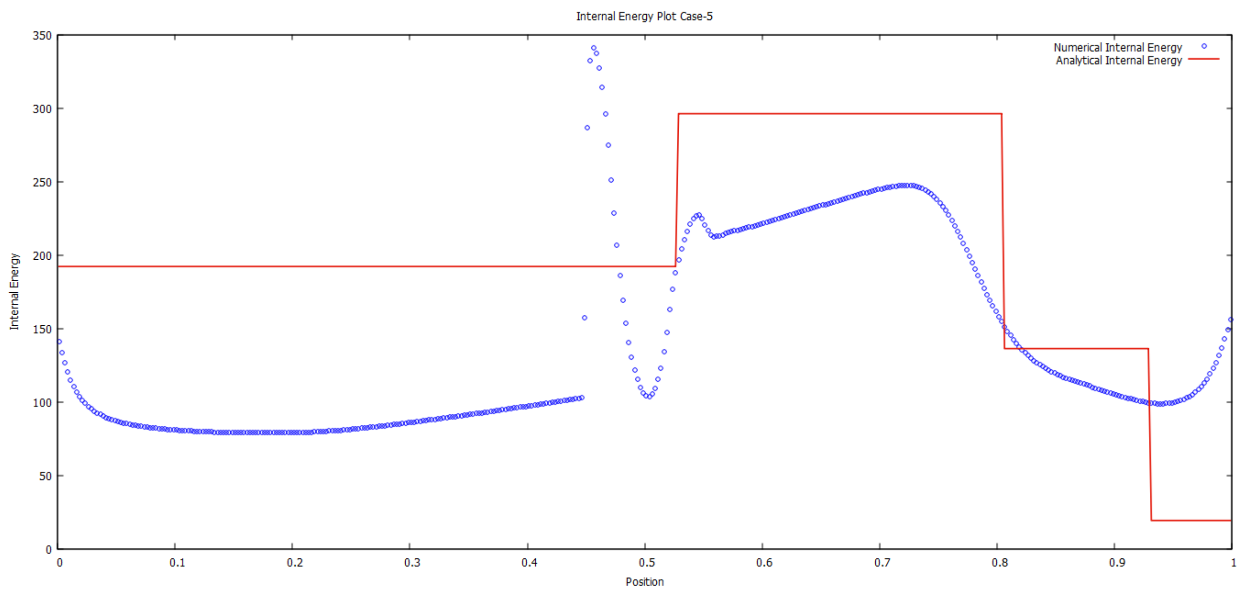


Density Plot Case-5
Numerical Density
Analytical Density
Density
Position
0
5
10
15
20
25
30
35
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



Velocity Plot Case-5
Numerical Velocity
Analytical Velocity
Velocity
Position
-15
-10
-5
0
5
10
15
20
25
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1





Conclusion:

Roe's algorithm for approximate solution of Euler's equation runs well for Cases 1,3,4 but the algorithm does not give desired results for case 2 and 5.