ME663 - Computational Fluid Dynamics
Assignment 2

Tommaso Bocchietti

A.Y. 2023/24 - W24

Contents

List of Figures

List of Tables

Listings

1 Introduction

Compressible fluids are characterized by significantly changes in density at pressure variations. This is typically the case for gases, which can be compressed or expanded, or for liquids at high speed or high pressure.
The metrics used to describe the compressibility of a fluid are the speed of sound a and the Mach number M. The speed of sound is the speed at which waves propagate through the fluid, and it is given by the equation:

$$a = \sqrt{\frac{\gamma p}{\rho}} \tag{1}$$

Where γ is the adiabatic index, p is the pressure, and ρ is the density.
The Mach number is the ratio of the speed of an object to the speed of sound in the fluid, and it is given by the equation:

$$M = \frac{u}{a} \tag{2}$$

Where u is the velocity of the object.
In this assignment, we are going to study the compressible fluid using the Euler equations, which are a set of equations that describe the conservation of mass, momentum, and energy in a fluid. The Euler equations are given by the following system of partial differential equations:

$$\frac{\partial U}{\partial t} + \nabla \cdot F = 0 \tag{3}$$

Where U is the vector of conserved variables, and F is the vector of fluxes.
If we consider a 1D flow, the Euler equations can be written as:

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

After some calculation that involve the explicit form of $F(U)$, we can observe that Euler equations are a hyperbolic system of partial differential equations, which means that they have a wave-like solution.
The final request for this assignment is to solve numerically Equation 4 using the flux-vector splitting method, which consists of splitting the flux vector based on the eigenvalues of the Jacobian matrix of the flux vector.
In the following, we will derive the Jacobian of the flux vector (Section 3), the eigenvalues and eigenvectors of the matrix (Section 4), and the splitted flux vectors formulation (Section 5), highlighting its dependence on the eigenvalues of the Jacobian matrix.
In section 6, we will focus on how to compute the eigenvalues of the system, reporting both the Steger & Warming formulation [1] and the van Leer formulation [2].
Finally, we will implement a 1D solver for the Euler equations using the flux-vector splitting method and compare the results with the exact solution (Section 7).

2 Methods

2.1 Symbolic Analysis

Most of the request for this assignment has been solved using symbolic analysis in `Mathematica`.
For each Q#, the chunk of code used to solve that particular question is reported at the end of each section.
Here, we report the head of the `Mathematica` notebook where both U and F are initialized as symbolic variables.

```
(*Initial setup*)
U = {
    {U1},
    {U2},
    {U3}
      };

F = {
    U[[2]],
    U[[2]]^2/U[[1]] + (\[Gamma] - 1)*(U[[3]] - U[[2]]^2/(2*U[[1]])),
    U[[2]]/U[[1]] * (U[[3]] + (\[Gamma] - 1)*(U[[3]] - U[[2]]^2/(2*U[[1]])))
    };
```

Listing 1: Initialization of symbolic variables in Mathematica.

A full notebook can be found in the appendix section of this document (Section 7.2.2).

2.2 Code implementation

The code implementation has been done in `MATLAB` for the sake of simplicity and coding speed with respect to the previous assignment where the code was implemented in `C11`.
However, we would like in the future to integrate the code in the codebase of the previous assignment, where both the SCGS and SIMPLE algorithms are currently implemented.
The complete code is attached in the zip file submitted with this report.

3 (Q1) - Question #1

Derive the Jacobian matrix $A = \frac{\partial F}{\partial U}$ given:

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho E \end{bmatrix}, \quad F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (\rho E + p)u \end{bmatrix} \tag{5}$$

where ρ is the density, u is the velocity, $E = \frac{p}{(\gamma-1)\rho} + \frac{1}{2}u^2$ is the total energy, and p is the pressure.

3.1 Solution

The Jacobian matrix A is defined as:

$$A = \frac{\partial F}{\partial U} = \begin{bmatrix} \frac{\partial F_1}{\partial U_1} & \frac{\partial F_1}{\partial U_2} & \frac{\partial F_1}{\partial U_3} \\ \frac{\partial F_2}{\partial U_1} & \frac{\partial F_2}{\partial U_2} & \frac{\partial F_2}{\partial U_3} \\ \frac{\partial F_3}{\partial U_1} & \frac{\partial F_3}{\partial U_2} & \frac{\partial F_3}{\partial U_3} \end{bmatrix} \tag{6}$$

where F_i is the i-th component of the vector F and U_i is the i-th component of the vector U.
Before proceeding with the derivation of the Jacobian matrix, it's useful to express the components of the vector F in terms of the components of the vector U:

$$F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (\rho E + p)u \end{bmatrix} = \begin{bmatrix} U_2 \\ \frac{U_2^2}{U_1} + (\gamma - 1)\left(U_3 - \frac{U_2^2}{2U_1}\right) \\ \frac{U_2}{U_1}\left(U_3 + (\gamma-1)\left(U_3 - \frac{U_2^2}{2U_1}\right)\right) \end{bmatrix} \tag{7}$$

Now, we can proceed with the derivation of the Jacobian matrix A based on its definition in Equation 6:

$$A = \frac{\partial F}{\partial U} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{\gamma-3}{2}u^2 & (3-\gamma)u & \gamma - 1 \\ (\gamma-1)u^3 - \gamma u E & -\frac{3}{2}(\gamma-1)u^2 + \gamma E & \gamma u \end{bmatrix} \tag{8}$$

The code of Mathematica used to derive the Jacobian matrix A is reported below:

```
(*Q1: Jacobian Matrix A*)

A = D[F[[All, 1]], {U[[All, 1]], 1}];

U = U /. {U1 -> \[Rho], U2 -> \[Rho]*u, U3 -> \[Rho]*Subscript[e, T]};
A = A /. {U1 -> \[Rho], U2 -> \[Rho]*u, U3 -> \[Rho]*Subscript[e, T]};
```

Listing 2: Mathematica notebook for Q1.

4 (Q2) - Question #2

Derive the right eigenvectors r_1, r_2, and r_3 for the matrix A (Equation 8). Note that:

$$Q_A = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} \tag{9}$$

4.1 Solution

The right eigenvectors r_1, r_2, and r_3 of the matrix A are defined as the eigenvectors that satisfy the following equation:

$$AQ_A = Q_A \Lambda \tag{10}$$

where Λ is the diagonal matrix of the eigenvalues of A and Q_A is the matrix of the right eigenvectors of A.
To solve the eigenvalues and eigenvectors problem, we can use its characteristic equation:

$$\det(A - \lambda I) = 0 \tag{11}$$

where I is the identity matrix.
The characteristic equation for the matrix A is:

$$\det(A - \lambda I) = \det\left(\begin{bmatrix} -\lambda & 1 & 0 \\ \frac{\gamma-3}{2}u^2 & (3-\gamma)u - \lambda & \gamma - 1 \\ (\gamma-1)u^3 - \gamma u E & -\frac{3}{2}(\gamma-1)u^2 + \gamma E & \gamma u - \lambda \end{bmatrix}\right) = 0 \tag{12}$$

Expanding the determinant, we get:

$$-\lambda\left[((3-\gamma)u - \lambda)(\gamma u - \lambda) - \left(-\frac{3}{2}(\gamma-1)u^2\right)(\gamma - 1)\right] + \dots \tag{13}$$

$$-1\left[\left(\frac{\gamma-3}{2}u^2\right)(\gamma u - \lambda) - \left((\gamma-1)u^3 - \gamma u E\right)(\gamma - 1)\right] = 0 \tag{14}$$

Solving the characteristic equation, we find the eigenvalues of the matrix A:

$$\Lambda = diag(\begin{bmatrix} u & u+a & u-a \end{bmatrix}) \tag{15}$$

where $a = \sqrt{\gamma p/\rho}$ is the speed of sound.
The right eigenvectors r_1, r_2, and r_3 of the matrix A are:

$$r_1 = \begin{bmatrix} 1 \\ u \\ \frac{u^2}{2} \end{bmatrix}, \quad r_2 = \begin{bmatrix} 1 \\ u+a \\ H+ua \end{bmatrix}, \quad r_3 = \begin{bmatrix} 1 \\ u-a \\ H-ua \end{bmatrix} \tag{16}$$

where $H = E + p/\rho$ is the enthalpy.
If we rewrite the right eigenvectors in terms of the primitive variables contained in the vector U, and adjust the modulus of each eigenvector to match the one of the Q_A reported in [1], we get:

$$Q_A = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} = \begin{bmatrix} 1 & \frac{\rho}{2c} & -\frac{\rho}{2c} \\ u & \frac{\rho}{2c}(u+a) & -\frac{\rho}{2c}(u-a) \\ \frac{u^2}{2} & \frac{\rho}{2c}(\frac{u^2}{2} + \frac{a^2}{\gamma-1} + ua) & -\frac{\rho}{2c}(\frac{u^2}{2} + \frac{a^2}{\gamma-1} - ua) \end{bmatrix} \tag{17}$$

The code of Mathematica used to derive the right eigenvectors r_1, r_2, and r_3 is reported below:

```
(*Q2: Right eigenvectors for A*)

U = U /. {Subscript[e, T] -> a^2/(\[Gamma]*(\[Gamma] - 1)) + u^2/2 };
A = A /. {Subscript[e, T] -> a^2/(\[Gamma]*(\[Gamma] - 1)) + u^2/2 };

{\[Lambda], Q} = Eigensystem[A];

\[Lambda][[{1, 2, 3}]] = \[Lambda][[{3, 1, 2}]];
Q[[{1, 2, 3}]] = Q[[{3, 1, 2}]];

factors = {
   u*u/2,
   \[Rho]/(2*a)*(u*u/2 + a*a/(\[Gamma] - 1) + a*u),
   -\[Rho]/(2*a)*(u*u/2 + a*a/(\[Gamma] - 1) - a*u)
   };

Q = Transpose[factors Q];
```

Listing 3: Mathematica notebook for Q2.

5 (Q3) - Question #3

Derive the following flux-vector splitting:

$$F^{\pm} = \left(\frac{\rho}{2\gamma}\right) \begin{bmatrix} 2(\gamma-1)\lambda_1^{\pm} + \lambda_2^{\pm} + \lambda_3^{\pm} \\ (2-\gamma)\lambda_1^{\pm} u + \lambda_2^{\pm}(u+a) + \lambda_3^{\pm}(u-a) \\ (\gamma-1)\lambda_1^{\pm} u^2 + \frac{\lambda_2^{\pm}}{2}(u+a)^2 + \frac{\lambda_3^{\pm}}{2}(u-a)^2 + \frac{3-\gamma}{2(\gamma-1)}(\lambda_2^{\pm}+\lambda_3^{\pm})a^2 \end{bmatrix} \tag{18}$$

Which is equivalent to the following:

$$F^{\pm} = \frac{1}{\gamma} Q_A \begin{bmatrix} (\gamma-1)\rho\lambda_1^{\pm} \\ a\lambda_2^{\pm} \\ -a\lambda_3^{\pm} \end{bmatrix} \tag{19}$$

Where Q_A is the eigenvector matrix (Equations 17).

5.1 Solution

We know that the flux-vector F can be expressed as:

$$F = AU = Q_A \Lambda Q_A^{-1} U \tag{20}$$

In order to proof the given flux-vector splitting, we will perform the matrix multiplication given the definitions of the eigenvector matrix Q_A (Equation 17) and the diagonal matrix of the eigenvalues Λ (Equation 15).
The flux-vector splitting can be expressed as follows:

$$F^{\pm} = \begin{bmatrix} \frac{\rho(-a+2\gamma\lambda_1^{\pm}-2\lambda_1^{\pm}+\lambda_2^{\pm}+u)}{2\gamma} \\ \frac{\rho\left(a^2+a(\lambda_2^{\pm}-2u)+u(2(\gamma-1)\lambda_1^{\pm}+\lambda_2^{\pm}+u)\right)}{2\gamma} \\ \frac{\rho\left(-2a^3+2a^2(\lambda_2^{\pm}+\gamma u)-a(\gamma-1)u(3u-2\lambda_2^{\pm})+(\gamma-1)u^2(2(\gamma-1)\lambda_1^{\pm}+\lambda_2^{\pm}+u)\right)}{4(\gamma-1)\gamma} \end{bmatrix} \tag{21}$$

From here, we still can't see the equivalence with the given expression.
We can now compute the coefficients that multiply each $\lambda_i^{\pm}$. For simplicity, we will consider the expression $\frac{F^{\pm}}{\frac{\rho}{2*\gamma}}$, obtaining the following coefficient matrix ($[\text{Known term}, \text{Coef}_{\lambda_1}, \text{Coef}_{\lambda_2}, \text{Coef}_{\lambda_3}]$)

$$\text{CoefficientList} = \begin{bmatrix} u-a & 2(\gamma-1) & 1 & 0 \\ (a-u)^2 & 2(\gamma-1)u & a+u & 0 \\ \frac{1}{2}(u-a)(\frac{2a^2}{\gamma-1}-2au+u^2) & (\gamma-1)u^2 & \frac{a^2}{\gamma-1}+au+\frac{u^2}{2} & 0 \end{bmatrix} \tag{22}$$

From now on, we can carry on the computation of the coefficients by hand, simplifying the coefficient and obtaining the desired result.

$$F^{\pm} = \frac{\rho}{2\gamma} \begin{bmatrix} u-a & + & 2(\gamma-1)\lambda_1^{\pm} & + & 1\lambda_2^{\pm} & + & 0\lambda_3^{\pm} \\ (a-u)^2 & + & 2(\gamma-1)u\lambda_1^{\pm} & + & a+u\lambda_2^{\pm} & + & 0\lambda_3^{\pm} \\ \frac{1}{2}(u-a)(\frac{2a^2}{\gamma-1}-2au+u^2) & + & (\gamma-1)u^2\lambda_1^{\pm} & + & (\frac{a^2}{\gamma-1}+au+\frac{u^2}{2})\lambda_2^{\pm} & + & 0\lambda_3^{\pm} \end{bmatrix} = \tag{23}$$

$$= \frac{\rho}{2\gamma} \begin{bmatrix} \lambda_3^{\pm} + 2(\gamma-1)\lambda_1^{\pm} + \lambda_2^{\pm} \\ \lambda_3^{\pm}(a+u) + 2(\gamma-1)u\lambda_1^{\pm} + \lambda_2^{\pm}(a+u) \\ \frac{\lambda_3^{\pm}}{2}(u^2-2au+a^2+\frac{2a^2}{\gamma-1}-a^2) + (\gamma-1)u^2\lambda_1^{\pm} + \frac{\lambda_2^{\pm}}{2}(u^2-2au+a^2+\frac{2a^2}{\gamma-1}-a^2) \end{bmatrix} = \tag{24}$$

$$= \frac{\rho}{2\gamma} \begin{bmatrix} 2(\gamma-1)\lambda_1^{\pm} + \lambda_2^{\pm} + \lambda_3^{\pm} \\ 2(\gamma-1)\lambda_1^{\pm} u + \lambda_2^{\pm}(u+a) + \lambda_3^{\pm}(u-a) \\ (\gamma-1)\lambda_1^{\pm} u^2 + \frac{\lambda_2^{\pm}}{2}(u+a)^2 + \frac{\lambda_3^{\pm}}{2}(u-a)^2 + \frac{3-\gamma}{2(\gamma-1)}(\lambda_2^{\pm}+\lambda_3^{\pm})a^2 \end{bmatrix} \tag{25}$$

The result in Equation 25 is equivalent to the given expression in Equation 18.
The code of Mathematica used to derive the coefficients list:

```
(*Q3: Flux-vector splitting*)

Qinv = Inverse[Q];
A = Q . DiagonalMatrix[\[Lambda]] . Qinv;

```

```
F = A . U /. {\[Lambda][[1]] -> \[Lambda]1, \[Lambda][[2]] -> \[Lambda]2, \[Lambda][[3]] ->
    \[Lambda]3 };

CoefficientList[F/(\[Rho]/(2*\[Gamma])), {\[Lambda]1, \[Lambda]2, \[Lambda]3}]
```

Listing 4: Mathematica notebook for Q3.

6 (Q4) - Question #4

Following Q3, show that $\lambda^{\pm}$ in $F^{\pm}$ corresponding to van Leer's flux-vector splitting method are:

$$\lambda_1^{\pm} = \frac{a}{4}(M+1)^2\left[1 - \frac{(M-1)^2}{\gamma+1}\right] \tag{26}$$

$$\lambda_2^{\pm} = \frac{a}{4}(M+1)^2\left[3 - M + \frac{\gamma-1}{\gamma+1}(M-1)^2\right] \tag{27}$$

$$\lambda_3^{\pm} = \frac{a}{4}(M+1)^2\left(2\frac{M-1}{\gamma+1}\right)\left[1 + \frac{\gamma-1}{2}M\right] \tag{28}$$

6.1 Tentative solution

We have tried to prove this by taking the expected flux vector in terms of M splitting and proving the identity against Equation 25 with the given values of $\lambda_1^{\pm}$, $\lambda_2^{\pm}$, and $\lambda_3^{\pm}$ for van Leer's formulation.
However, we have not been able to prove the identity, neither by hand nor by using Mathematica.

$$F = \begin{bmatrix} \rho a M \\ \frac{\rho a^2}{\gamma}(\gamma M^2 + 1) \\ \rho a^3 M(\frac{1}{2}M^2 + \frac{1}{\gamma-1}) \end{bmatrix} \tag{29}$$

Theoretically, by imposing Equation 29 equal to Equation 25 with the given values of $\lambda_1^{\pm}$, $\lambda_2^{\pm}$, and $\lambda_3^{\pm}$ relative to van Leer's formulation, we should have been able to prove the identity.
Here follows the Mathematica code used to try to prove the identity:

```
(*Q4: van Leer's lambdas*)

\[Lambda]1 = 1/4*a*(M + 1)^2*(1 - (M - 1)^2/(\[Gamma] + 1));
\[Lambda]2 = 1/4*a*(M + 1)^2*(3 - M + (\[Gamma] - 1)/(\[Gamma] + 1)*(M - 1)^2);
\[Lambda]3 = 1/2*a*(M + 1)^2*(M - 1)/(\[Gamma] + 1)*(1 + (\[Gamma] - 1)/2*M);

FVL = {
   {\[Rho]* a* M},
   {(\[Rho]* a^2)/(\[Gamma])  (\[Gamma]* M^2 + 1)},
   {\[Rho]* a^3 *M  (1/2* M^2 + 1/(\[Gamma] - 1))}
    };

MatrixForm[F == FVL] // FullSimplify
```

Listing 5: Mathematica code used to prove the identity for van Leer's flux-vector splitting method

7 (Q5) - Question #5

Write a computer code for Test Case 1 (Handout #4, p. 352) using both Steger-Warming and van Leer flux-vector splitting methods, and compare your numerical results with the exact solutions at $t = 0.01$ for $-10 \leq x \leq 10$ in terms of density, velocity, pressure, Mach number.

7.1 Code implementation

The code for this question is implemented in `MATLAB` and can be found in the attached files to this report.
Both the Steger-Warming and van Leer flux-vector splitting methods are implemented for the 1D Euler equations. As for the order of the interpolation scheme used to compute the fluxes at the cell interfaces, the code allows for the selection of either a zero-order upwind scheme or a first-order linear upwind scheme.

7.2 Results

The results of the simulation are consistent with the exact solution provided with the assignment request.
In the following figures, we can see the results of the simulation for the Steger-Warming and van Leer flux-vector splitting methods for a mesh of 50 and 400 cells.
The computed solutions (black) have been interpolated using a zero-order upwind scheme (UDS). Exact solution are represented in red.

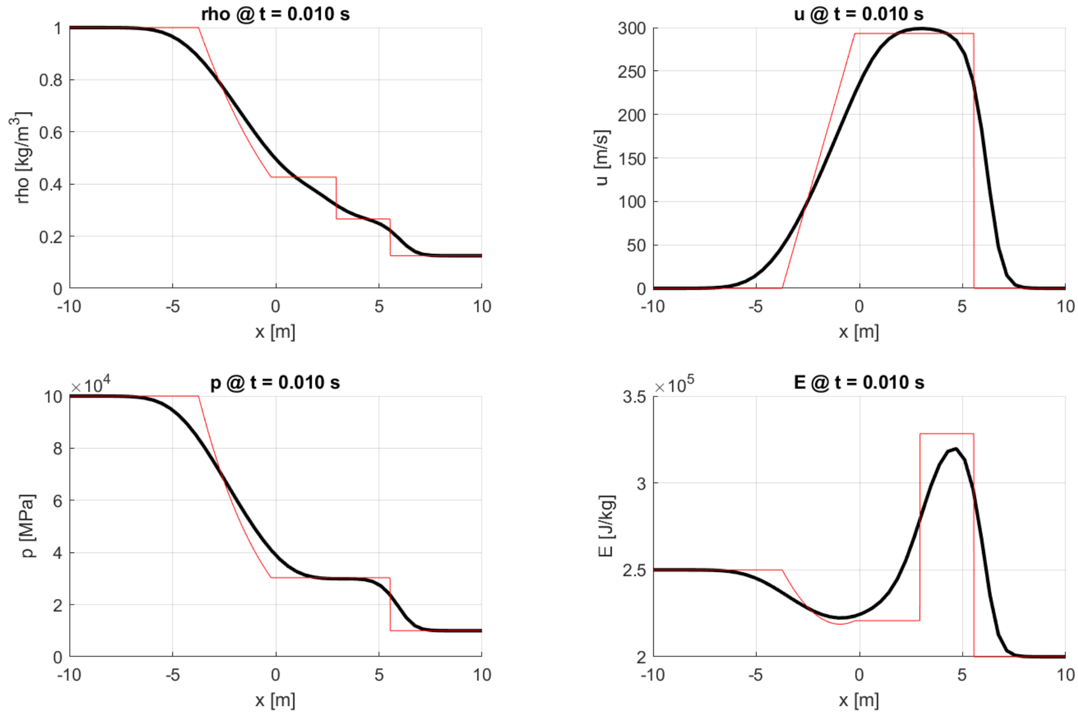

Figure 1: Steger-Warming flux-vector splitting method with 50 cells mesh.

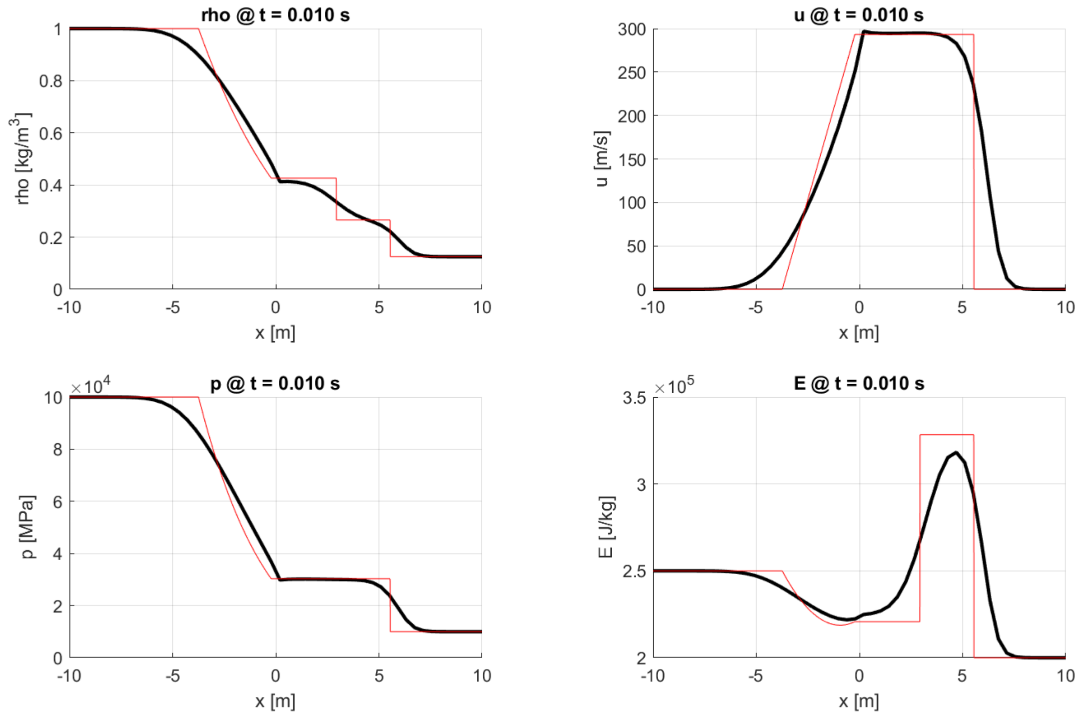

Figure 2: Van Leer flux-vector splitting method with 50 cells mesh.

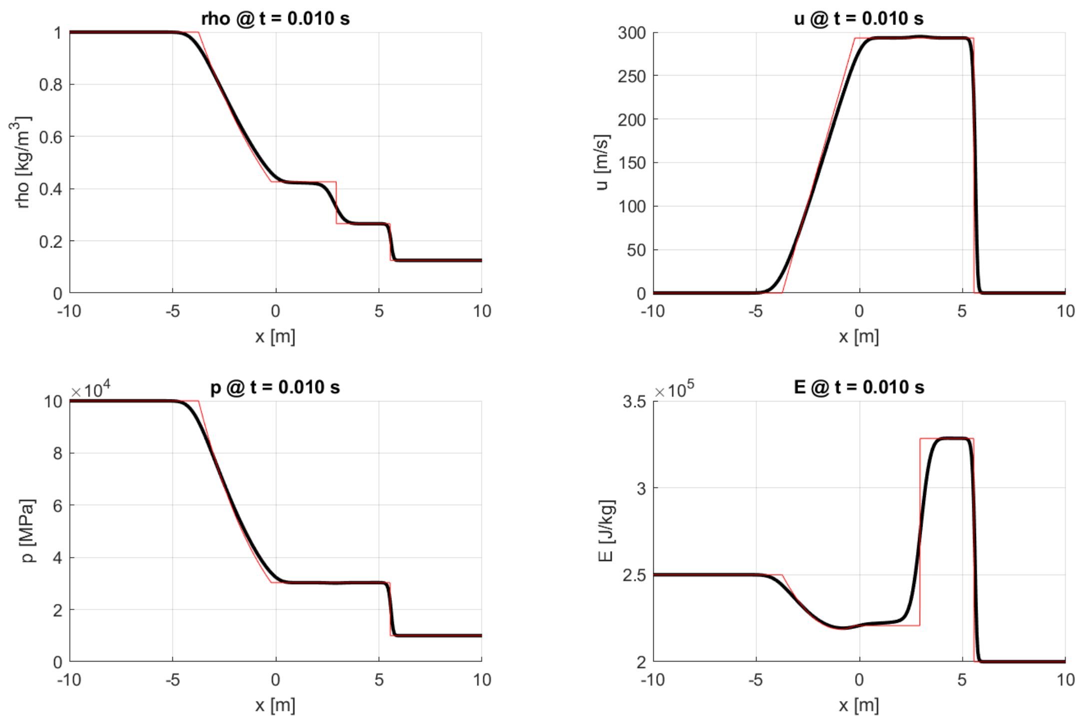

Figure 3: Steger-Warming flux-vector splitting method with 400 cells mesh.

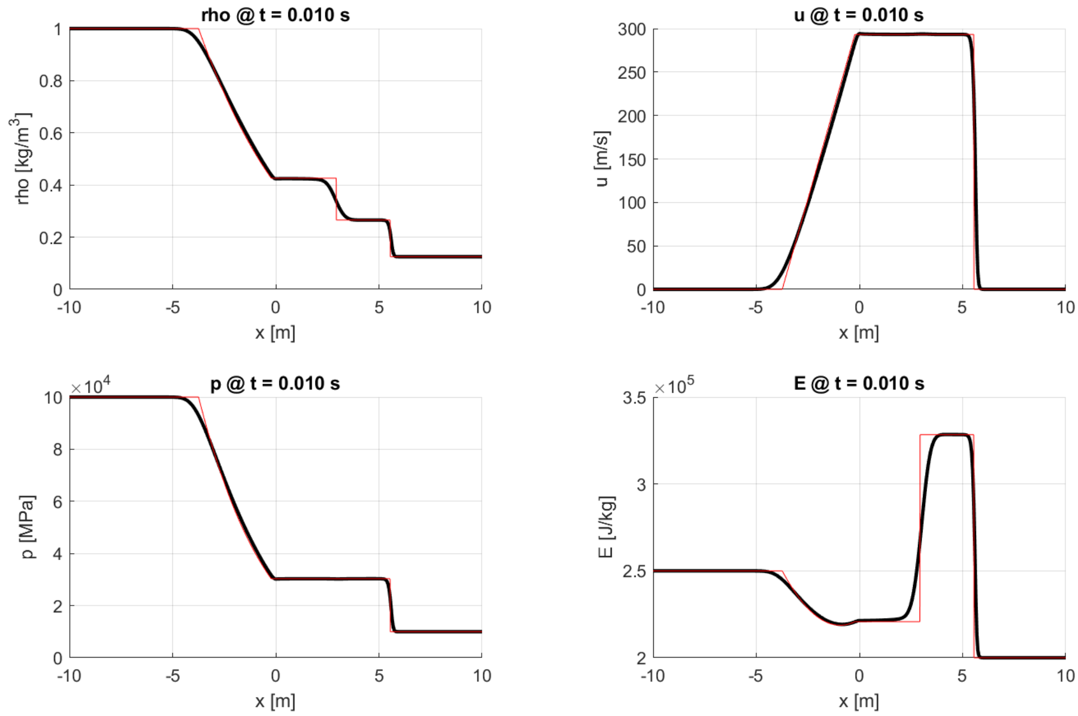

Figure 4: Van Leer flux-vector splitting method with 400 cells mesh.

7.2.1 Steger-Warming vs. Van Leer splitting methods

As we can see observing Figures 1 and 2, the Steger-Warming method tends to be more diffusive than the van Leer method, which is more accurate in capturing the shock wave. In particular, by observing the step ahead of the shock wave (i.e., the region behind the contact point and the expansion wave), we can see that the Steger-Warming method tends to over-diffuse the solution and doesn't capture the shock wave, while the van Leer method is more accurate in capturing the discontinuity between the two regions.

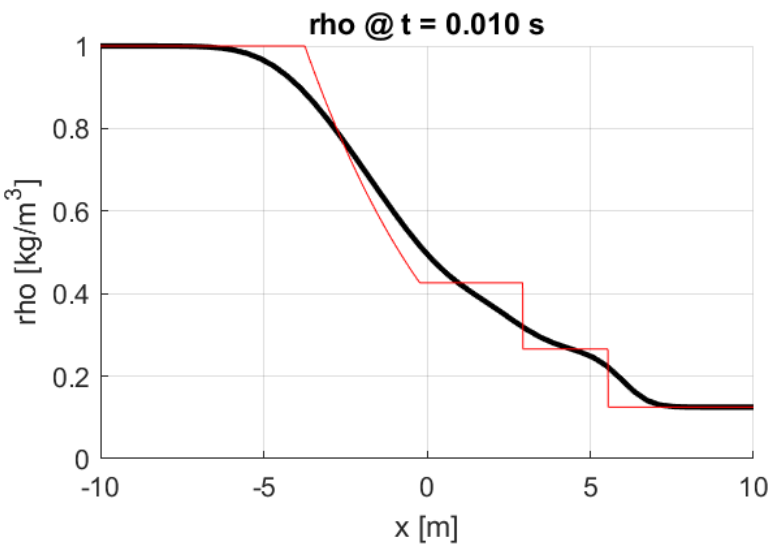

Figure 5: Density computed with Steger-Warming flux-vector splitting method (50 cells mesh).

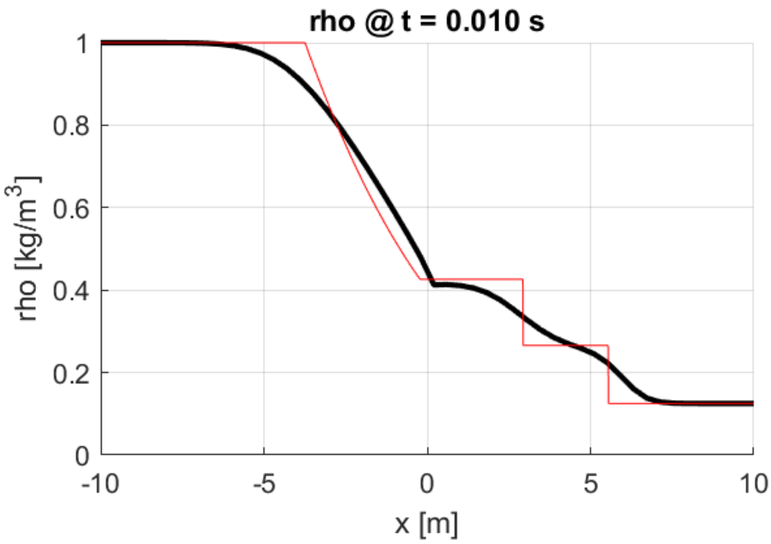

Figure 6: Density computed with van Leer flux-vector splitting method (50 cells mesh).

7.2.2 Order of the interpolation scheme (UDS vs. LUDS)

The code allows for the selection of either a zero-order upwind scheme (UDS) or a first-order linear upwind scheme (LUDS) to compute the fluxes at the cell interfaces.
However, as also explained in literature, the first-order linear upwind scheme tends to be unstable, generating strong oscillations at the discontinuity regions.
In the following, we can see the results of the simulation for the Steger-Warming method and the van Leer method with a limited first-order linear upwind scheme (LUDS) for a 50 cells mesh. In particular, the adopted interpolation scheme is the following:

$$U_e = U_i + \frac{1}{4}\left(U_i - U_{i-1}\right) \tag{30}$$

Notice the weight of the interpolation scheme is set to $\frac{1}{4}$ to limit the oscillations generated by the complete first-order linear upwind scheme.

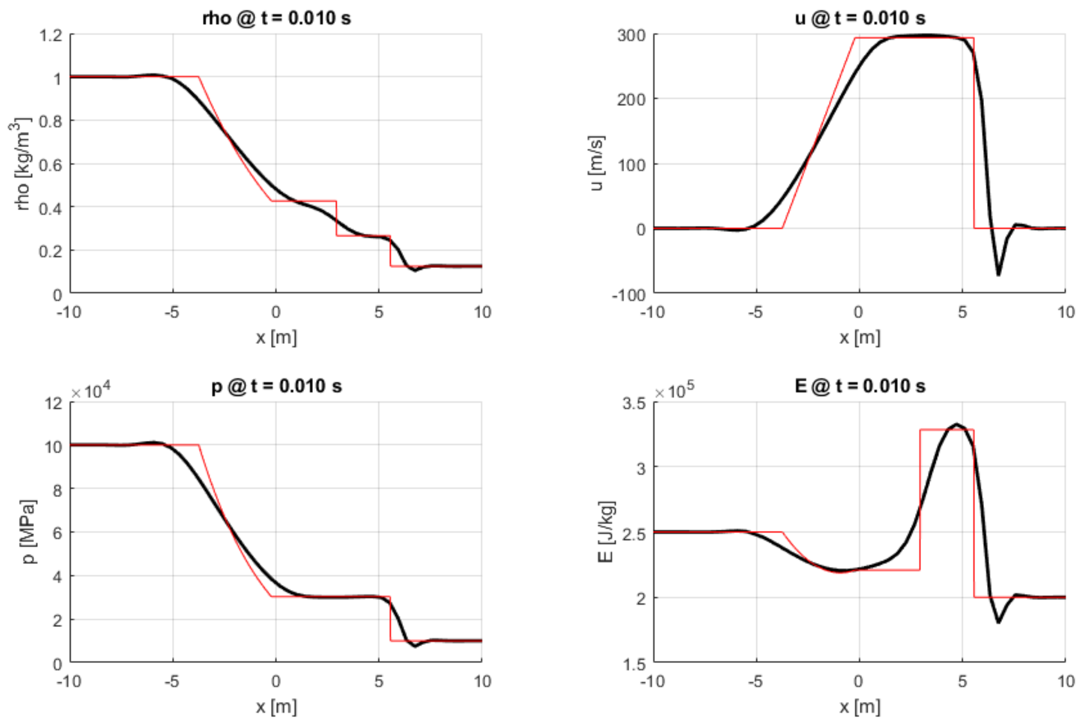

Figure 7: Steger-Warming flux-vector splitting method with 50 cells mesh and limited LUDS interpolation scheme.

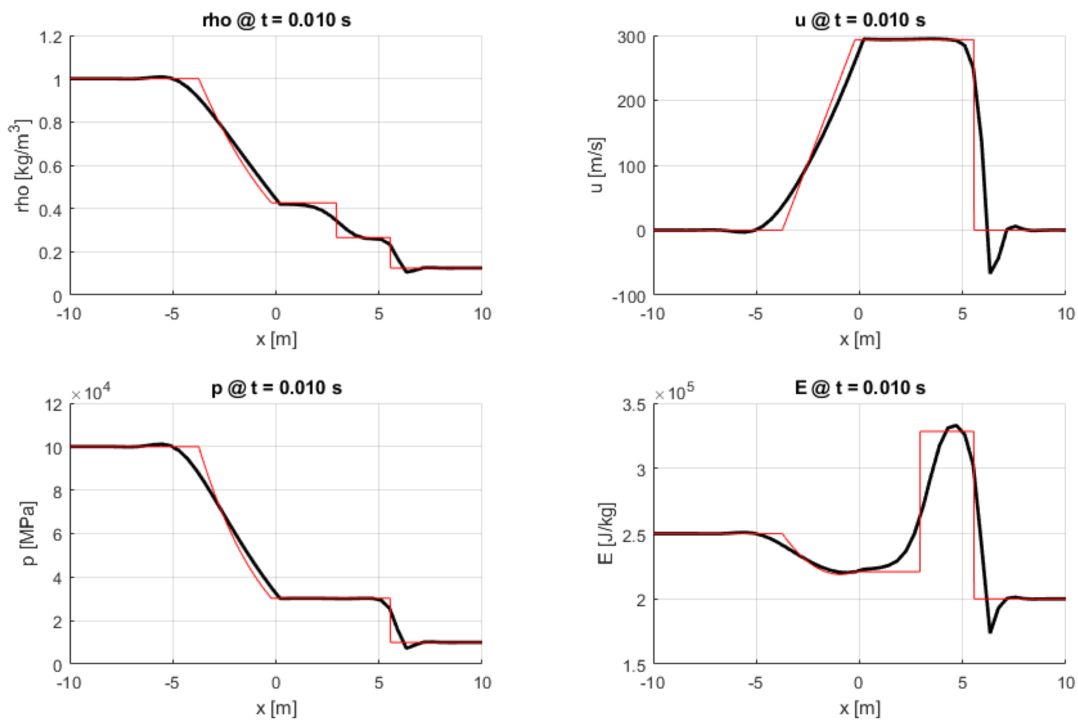

Figure 8: Van Leer flux-vector splitting method with 50 cells mesh and limited LUDS interpolation scheme.

In Figures 7 and 8, the oscillations generated by the first-order linear upwind scheme are visible, especially in the plot of the velocity and total energy.

References

[1] Joseph L Steger and R.F Warming. Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods. *Journal of Computational Physics*, 40(2):263–293, 1981.

[2] Bram van Leer. Flux-vector splitting for the euler equation. volume 170, 01 1982.

A Mathematica code

Here follows the `Mathematica` notebook used for symbolic analysis of the discretized schemes.

```
(*Tommaso Bocchietti*)
(*ME663-Computational Fluid Dynamics*)
(*Compressible flow matrices calculations*)
Clear["Global`*"]


(*Initial setup*)
U = {
{U1},
{U2},
{U3}
 };

F = {
U[[2]],
U[[2]]^2/U[[1]] +(\[Gamma]-1)*(U[[3]]-U[[2]]^2/(2*U[[1]])),
U[[2]]/U[[1]] * (U[[3]]+(\[Gamma]-1)*(U[[3]]-U[[2]]^2/(2*U[[1]])))
};


(*Q1: Jacobian Matrix A*)
A = D[F[[All,1]], {U[[All,1]], 1}];
U = U/.{U1->\[Rho],U2->\[Rho]*u,U3->\[Rho]*Subscript[e, T]};
A = A/.{U1->\[Rho],U2->\[Rho]*u,U3->\[Rho]*Subscript[e, T]};

MatrixForm[A] //FullSimplify
(*Q2: Right eigenvectors for A*)
U = U/.{Subscript[e, T]->a^2/(\[Gamma]*(\[Gamma]-1))+u^2/2 };
A=A/.{Subscript[e, T]->a^2/(\[Gamma]*(\[Gamma]-1))+u^2/2 };

{\[Lambda], Q}=Eigensystem[A];

\[Lambda][[{1,2,3}]]=\[Lambda][[{3,1,2}]];
Q[[{1,2,3}]]=Q[[{3,1,2}]];

factors = {
u*u/2,
\[Rho]/(2*a)*(u*u/2+a*a/(\[Gamma]-1)+a*u),
-\[Rho]/(2*a)*(u*u/2+a*a/(\[Gamma]-1)-a*u)
};

Q=Transpose[factors Q];

MatrixForm[\[Lambda]]//FullSimplify
MatrixForm[Q]//FullSimplify


(*Q3: Flux-vector splitting*)
Qinv = Inverse[Q];
```

```
A = Q.DiagonalMatrix[\[Lambda]].Qinv;
F = A.U/. {\[Lambda][[1]]->\[Lambda]1, \[Lambda][[2]]->\[Lambda]2, \[Lambda][[3]]->\[Lambda]3 };
MatrixForm[F]//FullSimplify
MatrixForm[CoefficientList[F/(\[Rho]/(2*\[Gamma])), {\[Lambda]1, \[Lambda]2, \[Lambda]3}]]//FullSimplify


(*Q4: van Leer's lambdas*)
\[Lambda]1 = 1/4*a*(M+1)^2*(1-(M-1)^2/(\[Gamma]+1));
\[Lambda]2 = 1/4*a*(M+1)^2*(3-M+(\[Gamma]-1)/(\[Gamma]+1)*(M-1)^2);
\[Lambda]3 = 1/2*a*(M+1)^2*(M-1)/(\[Gamma]+1)*(1+(\[Gamma]-1)/2*M);

FVL = {
{\[Rho]* a* M},
{(\[Rho]* a^2)/(\[Gamma]) (\[Gamma]* M^2+1)},
{\[Rho]* a^3 *M (1/2* M^2+ 1/(\[Gamma]-1))}
 };
MatrixForm[F==FVL] //FullSimplify
```
Listing 6: Mathematica notebook used for symbolic analysis.