Imperial College
London

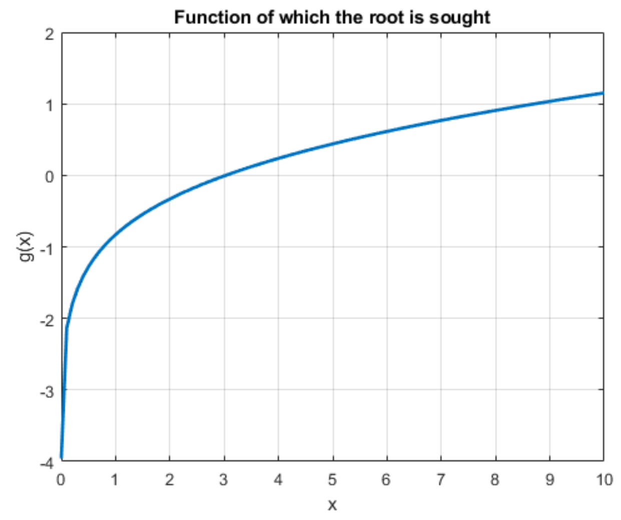
AERO96014 - Introduction to Computational Fluid Dynamics

Coursework 2

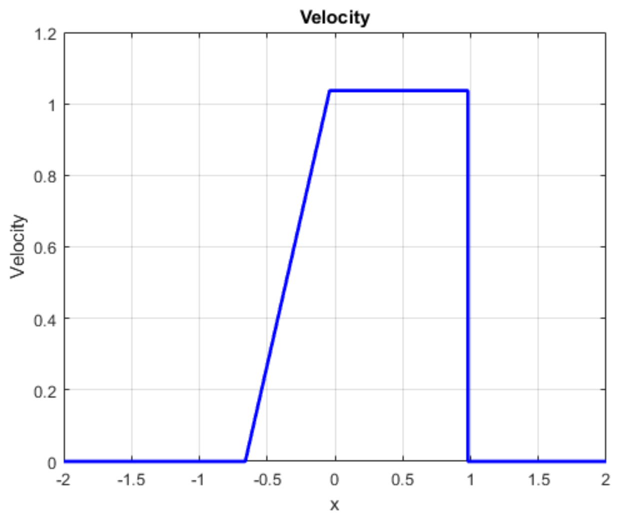
Academic Responsible:	Dr. George Papadakis
Department:	Department of Aeronautics
Course:	MEng Aeronautical Engineering
Module:	Computational Fluid Dynamics
Academic year:	2020/2021
Student:	Jo Wayne Tan
CID:	01327317
Personal tutor:	Dr. Francesco Montomoli
Date:	01/02/2021

Department of Aeronautics
South Kensington Campus
Imperial College London
London SW7 2AZ
U.K.

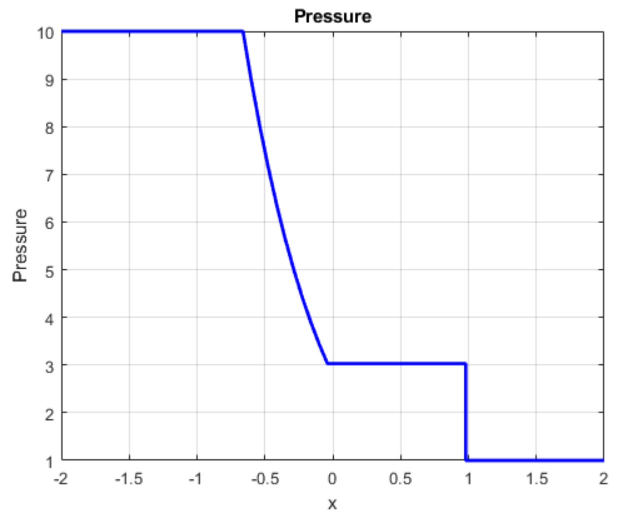
1. Analytical Solution



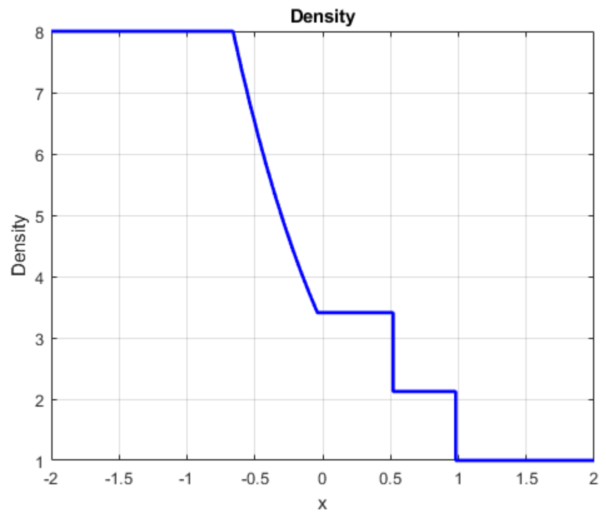
(a) *Function g(x) vs pressure ratios.*



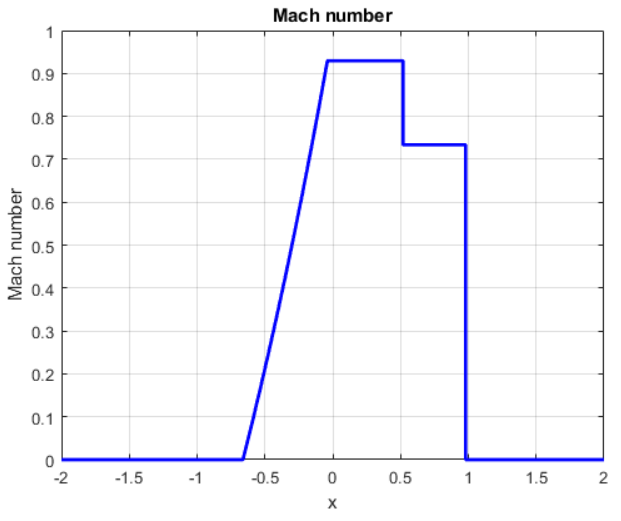
(b) *Velocity along the tube.*



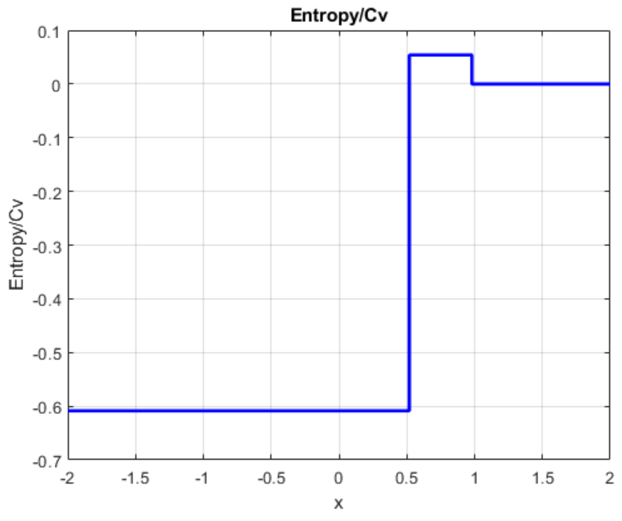
(c) *Pressure along the tube.*



(d) *Air density along the tube.*



(e) *Mach number in the domain.*



(f) *Entropy levels across the shock.*

Figure 1: Analytical solutions of the Riemann shock tube problem in the interval $-2 \leq x \leq 2$ for the time, t = 0.5, for a pressure ratio $\frac{P_2}{P_1} = 10$ and a density ratio $\frac{\rho_2}{\rho_1} = 8$.

2. First Order Upwind Schemes

Through the flux vector splitting method, the resulting 1D Euler equations become:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}^+(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{F}^-(\mathbf{U})}{\partial x} = 0 \tag{1}$$

Discretisation of the system (1) using first order upwind schemes for the spatial derivatives $\frac{\partial \mathbf{F}^+(\mathbf{U})}{\partial x}$ and $\frac{\partial \mathbf{F}^-(\mathbf{U})}{\partial x}$ depends on the physical aspects of the flow, namely the direction of the flow. This discretisation in space uses a backward difference for flux $\mathbf{F}^+$ and a forward difference for flux $\mathbf{F}^-$.

$$\frac{\partial \mathbf{F}^+(\mathbf{U})}{\partial x} \simeq \frac{\mathbf{F}_i^+(\mathbf{U}) - \mathbf{F}_{i-1}^+(\mathbf{U})}{\Delta x} \quad \text{and} \quad \frac{\partial \mathbf{F}^-(\mathbf{U})}{\partial x} \simeq \frac{\mathbf{F}_{i+1}^-(\mathbf{U}) - \mathbf{F}_i^-(\mathbf{U})}{\Delta x} \tag{2}$$

The temporal derivative $\frac{\partial \mathbf{U}}{\partial t}$ is discretised using an explicit in time scheme (for simplicity of code):

$$\frac{\partial \mathbf{U}}{\partial t} \simeq \frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t} \tag{3}$$

Substituting above finite difference approximations (2, 3) into (1) and rearranging the equation gives the discretised system in conservative form:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\left[\left(\mathbf{F}_i^+ - \mathbf{F}_{i-1}^+\right)^n + \left(\mathbf{F}_{i+1}^- - \mathbf{F}_i^-\right)^n\right] \tag{4}$$

This first order explicit upwind scheme is consistent and conservative, hence if the numerical solution converges, it converges to a weak solution according to the Lax-Wendroff Theorem [1].

3. Numerical Solution

A brief description of the Matlab code attached within the submission is presented in this section. A *for* loop is first implemented to compute the numerical solution for three different meshes with N = 100, 200 and 300 equally spaced points respectively. All variables required across the domain is re-initialised every time a new mesh is computed. A *while* loop is implemented to time march the solution until the solution time input, t=0.5, is reached. Two *for* loops within are then used to evaluate the eigenvalues ($\lambda_i^{\pm}$) and fluxes ($\mathbf{F}^{\pm}(\mathbf{U})$), and the solution vector at next time step ($\mathbf{U}_i^{n+1}$) across the interval ($-2 \leq x \leq 2$) using first order upwind scheme. As shown by the analytical solution in Figure 1(c), the flow is subsonic throughout, hence $-c < u < c$, and $u > 0$ can be assumed without loss of generality.

$$\mathbf{\Lambda} = \begin{bmatrix} u-c & & 0 \\ & u & \\ 0 & & u+c \end{bmatrix} = \mathbf{\Lambda}^+ + \mathbf{\Lambda}^-, \quad \mathbf{\Lambda}^+ = \begin{bmatrix} 0 & & 0 \\ & u & \\ 0 & & u+c \end{bmatrix}, \quad \mathbf{\Lambda}^- = \begin{bmatrix} u-c & & 0 \\ & 0 & \\ 0 & & 0 \end{bmatrix} \tag{5}$$

Once $\lambda_i^{\pm}$ are obtained, the fluxes can be found using the equation below (equation (10) from coursework handout):

$$\mathbf{F}^{\pm}(\mathbf{U}) = \frac{\rho}{2\gamma}\begin{bmatrix} \lambda_1^{\pm} + 2(\gamma-1)\lambda_2^{\pm} + \lambda_3^{\pm} \\ (u-c)\lambda_1^{\pm} + 2(\gamma-1)u\lambda_2^{\pm} + (u+c)\lambda_3^{\pm} \\ (H-uc)\lambda_1^{\pm} + (\gamma-1)u^2\lambda_2^{\pm} + (H+uc)\lambda_3^{\pm} \end{bmatrix} \tag{6}$$

Equation (5, 6) are essentially computed in the first *for* loop throughout the whole domain. Once a suitable value of Δt that guarantees stability of the numerical scheme is obtained using flow properties (explained in Question 4.), all variables needed to compute $\mathbf{U}_i^{n+1}$ using equation (4) above would have been obtained. This is the purpose of the subsequent *for* loop, it goes through all the spatial points in the interval to compute the new solution $\mathbf{U}^{n+1}$ at each point i. To implement the numerical boundary conditions to be $\mathbf{U}_0 = \mathbf{U}_1$ and $\mathbf{U}_{N+1} = \mathbf{U}_N$ at the end points $x = -2$ and $x = 2$, equation (4) is modified to be:

$$\mathbf{U}_1^{n+1} = \mathbf{U}_1^n - \frac{\Delta t}{\Delta x}\left[\left(\mathbf{F}_1^+ - \mathbf{F}_1^+\right)^n + \left(\mathbf{F}_2^- - \mathbf{F}_1^-\right)^n\right], \quad \mathbf{U}_N^{n+1} = \mathbf{U}_N^n - \frac{\Delta t}{\Delta x}\left[\left(\mathbf{F}_N^+ - \mathbf{F}_{N-1}^+\right)^n + \left(\mathbf{F}_N^- - \mathbf{F}_N^-\right)^n\right] \tag{7}$$

These values are computed at the end points hence are excluded from the *for* loop. All flow variables across the domain (ρ, u, E, p, H, c) are then updated to the next time step using the new solution vector $\mathbf{U}_i^{n+1}$, and the time, t, increases by the corresponding Δt of each specific iteration and the calculations repeated until the set time instant is reached. In summary, the scheme solves the conservative equations with a fixed mesh spacing and a variable time step calculated at each time level considering stability requirements, with a pre-defined Courant number (Question 4.).

4. Δt for Stability of Numerical Scheme

The linearized von Neumann analysis stability condition from lecture notes [1] is:

$$\sigma = \frac{\Delta t}{\Delta x}\lambda_m \leq 1, \quad \text{where} \quad \lambda_m = |u| + c \quad \text{is the max eigenvalue of } \mathbf{A}, \quad \max(\lambda_i) \tag{8}$$

and σ is the CFL/Courant number, in other words a CFL number ≤ 1 is required for stability. The maximum eigenvalue of Jacobian matrix $\mathbf{A}$ at current time level across the whole domain needs to be determined at each iteration. The maximum feasible time step for stability can then be determined as $\Delta t \leq \frac{\Delta x}{\max(\lambda_i)}$, and any time step below this maximum value can be chosen. Hence, this is equivalent to choosing a suitable CFL number to determine the best time step, $\frac{\sigma \Delta x}{\max(\lambda_i)} = \Delta t$. According to Steger and Warming [2], the CFL number used for a one-dimensional shock-tube flow was $\sigma \simeq 0.95$, and this was used as a starting point to determine optimal time step. It is known that the overall dispersion error increases with Courant number (as proven in Appendix), however the computational time/cost decreases with it as Δt increases. σ is therefore slowly increased by small increments from 0.95 to 1 to find the maximum allowable value in between that does not give any dispersion error, as any value above 1 breaks the stability condition and gives significant dispersion error (spurious oscillations). It was found that $\sigma > 0.98$ starts to give visible oscillations (wiggles) near discontinuities in solutions as shown in Appendices A.1 & A.2. Hence, a final CFL number of 0.98 was chosen for our numerical solution, $\sigma = 0.98 \Rightarrow \Delta t = \frac{0.98\Delta x}{\max(\lambda_i)}$ is therefore the optimal time step equation to be used in our code to evaluate Δt at each time level to ensure stability, in order to provide minimal dispersion error and maximum computational speed (elapsed time of code $\simeq$ 0.5 seconds) with minimal loss of solution characteristics and detail.

5. Results

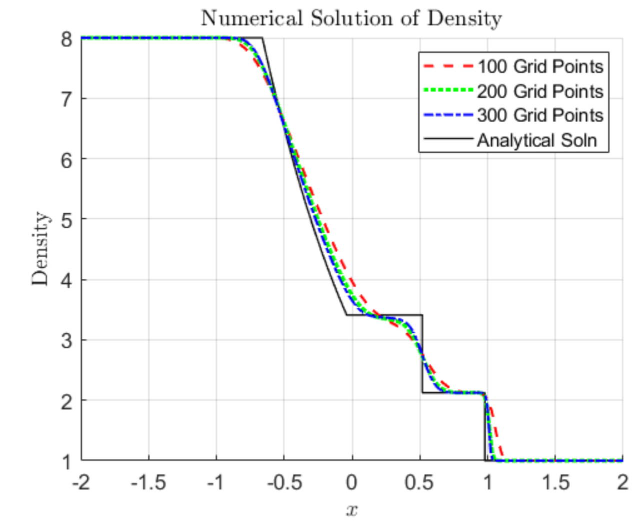
The computed results are presented in Figure 2. The solution consists of a rarefaction (or expansion) wave propagating towards the left, a shock wave and a contact discontinuity, both propagating to the right [1]. These phenomena can be shown by the two huge vertical jumps in both numerical and analytical solutions in Figure 2 (at the $x \simeq 0.5$ region and x = 1). Euler equations does not include viscous terms. Discretisation generally introduces viscosity, or more precisely second order difference terms that have viscous-like effects [1]. In this case, the numerical scheme can be thought of as discretising the following PDE according to John D. Anderson, Jr. [3]:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}(\mathbf{U})\frac{\partial \mathbf{U}}{\partial x} = \mathbf{B}(\mathbf{U})\frac{\partial^2 \mathbf{U}}{\partial x^2} + \mathbf{C}(\mathbf{U})\frac{\partial^3 \mathbf{U}}{\partial x^3} + \mathbf{O}[\ ...\ ] \tag{9}$$

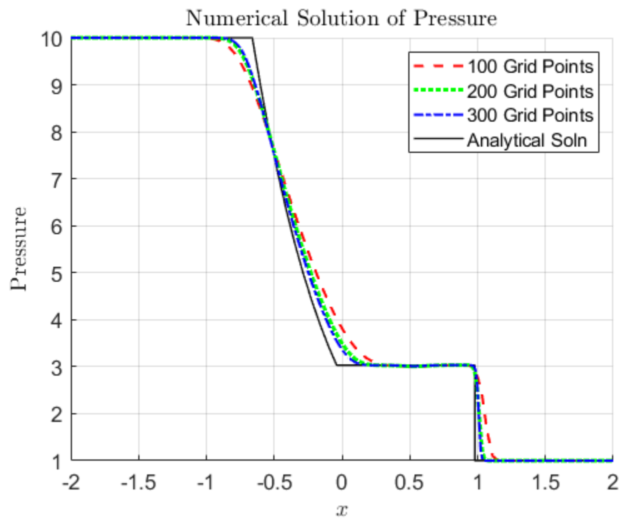
where $\mathbf{A}$ is the Jacobian of the flux vector and $\mathbf{B}$ is a coefficient that has dimensions of viscosity (acts like but has no relationship to physical viscosity) and is therefore called the (implicit) *artificial viscosity.* Hence, equation (9) is called the *modified equation* [3] (exact solution for the difference equation (4)), and the leading even order derivative term $\frac{\partial^2 \mathbf{U}}{\partial x^2}$ on the RHS of the modified equation (truncation error) appears purely for numerical reasons [1] (consequence of numerical discretisation (4)) and is the *numerical dissipation* term, much like the viscous terms in the Navier-Stokes equations. Instead of representing the dissipative aspect on the flow, this term signifies the diffusive behaviour of a numerical solution with no physical significance. It gives rise to smoothing of results near discontinuities (regions with steep gradients of ρ, u, p, for ex. a shock) as shown by the rounding of numerical solutions near sharp edges especially in Figure 2 (a, c (rounding of plateau), d & e). Numerical predictions of this first order upwind scheme at these regions of corners have relatively poor accuracy in general, hence is said to compromise the accuracy of the solution [3], although it gets better with increasing number of grid points.

Just like how dissipation is the result of even order derivatives of truncation error, numerical dispersion is the result of odd order derivatives ($\frac{\partial^3 \mathbf{U}}{\partial x^3}$ and so on, refer to equation (9)). The stable time step equation determined in Question 4. made sure these distortion of propagation of different phases of the wave do not exist in our solution. The implicit existence of artificial viscosity term also serves to improve the stability of the solution [3], preventing the development of these overshoots and undershoots in the form of oscillations (as shown in Appendices A.1 & A.2).

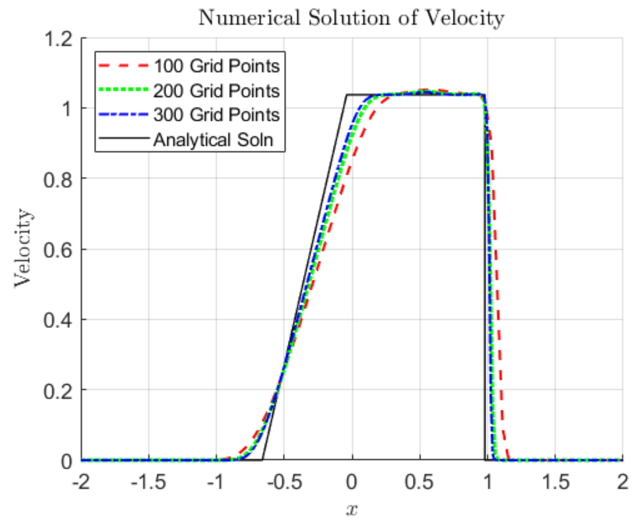
In conclusion, the Steger and Warming method successfully captured and produced the shape of the analytical solution with almost identical values in smooth/continuous regions. The solution is monotone [2] hence is non-oscillatory (do not create new extrema as shown in Figure 2) and satisfies the entropy condition when convergent [1]. The scheme is also in conservative form hence can only be first order accurate, which explains why it cannot capture discontinuities well as shown in Figure 2. This is due to the presence of diffusion errors, although this effect of smoothing by diffusion decreases with larger number of grid points, in other words more refined meshes. The numerical solution with N = 300 describes the analytical solution best in Figure 2 whilst solution with N = 100 produced the worst result due to intense dissipation errors. This is because according to $\Delta t = \frac{0.98\Delta x}{\max(\lambda_i)}$, a decrease in Δx due to a finer mesh leads to a decrease in Δt, meaning that Δt is dependent on Δx. The decrease of both Δt and Δx reduces the magnitude of the $\mathbf{B}$ coefficient of the leading even order derivative (and other even order derivatives), hence reducing the resulting dissipation error, moving the numerical solution closer to the analytical one, providing greater accuracy. This observation once again agrees with the consistency property of the solution, defined as the numerical solution should tend to the exact differential equation as Δt and Δx tend to zero. Combined with the stability condition implemented, the numerical solution guarantees convergence according to the Lax Equivalence Theorem [4]. With the correct conditions and implementation, the first order upwind Steger and Warming Flux Vector Splitting method (first order in space and time) approximates the Riemann shock tube problem very reasonably in a relatively short amount of time.



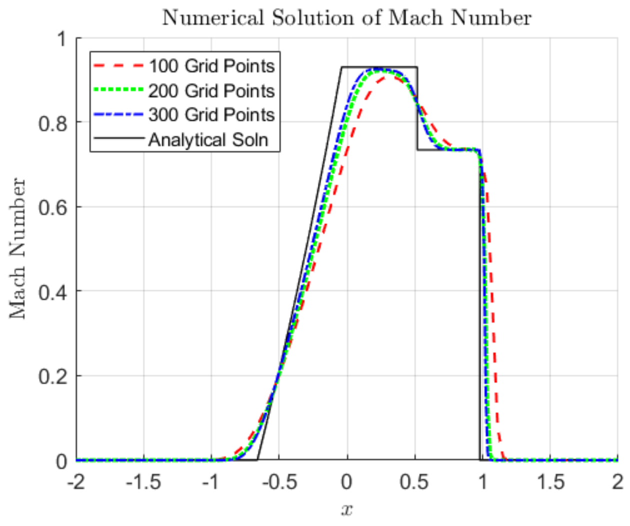
(a) *Air density along the tube.*



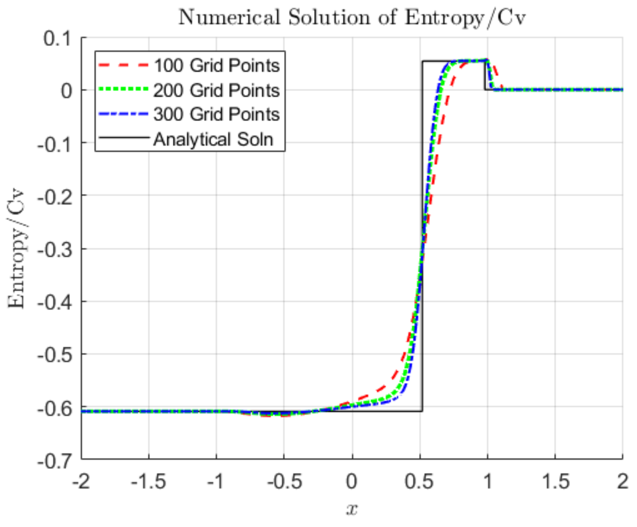
(b) *Pressure along the tube.*



(c) *Velocity along the tube.*



(d) *Mach number in the domain.*



(e) *Entropy/Cv across the shock.*

Figure 2: The numerical solution ($\sigma = 0.98$) versus the analytical solution computed at solution time t = 0.5. Numerical solution of the Riemann shock tube problem computed using three different meshes with N = 100, 200 and 300 equally spaced points in the interval $-2 \leq x \leq 2$.

References

[1] G. Papadakis. AERO96014 [part II] Lecture notes. 2020.

[2] Joseph L. Steger and R.F. Warming. Flux Vector Splitting of the Inviscid Gas Dynamics Equations with Application to Finite-Difference Methods. Computational Fluid Dynamics Branch, Ames Research Center, NASA, Moffett Field, California, 1981.

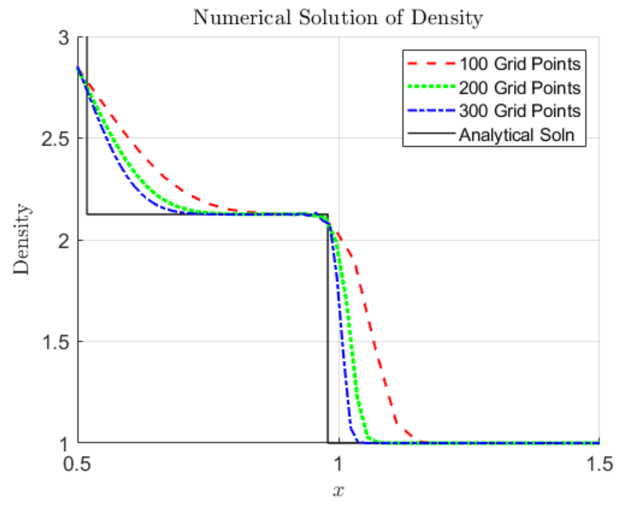
[3] Jr. John D. Anderson. Computational Fluid Dynamics The Basics with Applications. Department of Aerospace Engineering, University of Maryland, 1995.

[4] Prof. Spencer Sherwin. AERO96014 [part I] Introduction to CFD. 2020.

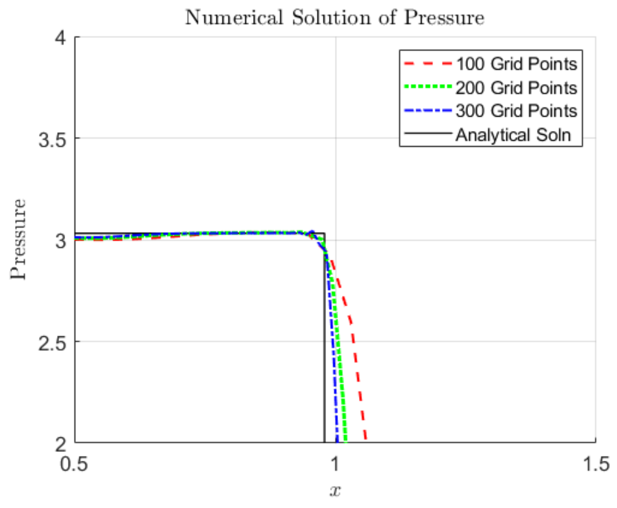
A Appendices

A.1 Numerical Solution with $\sigma = 0.99$

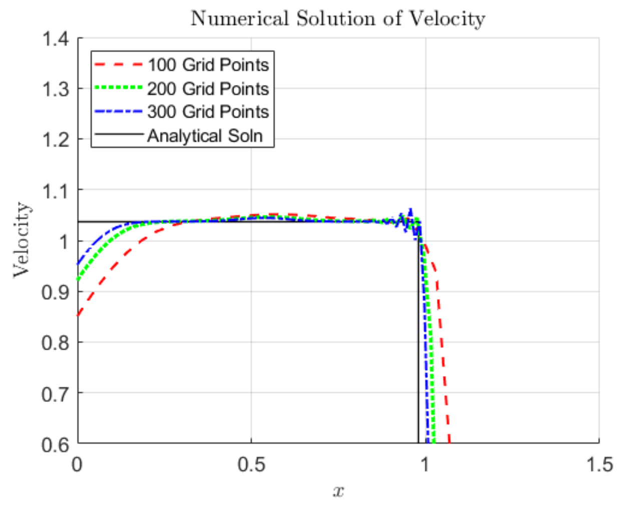
Results were plotted for $\sigma = 0.99$ to show onset of numerical oscillations in solution pass the $\sigma = 0.98$ point.



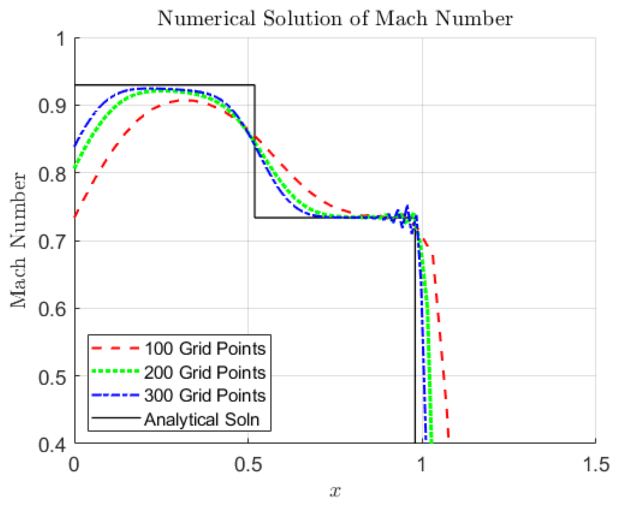
(a) *Air density along the tube.*



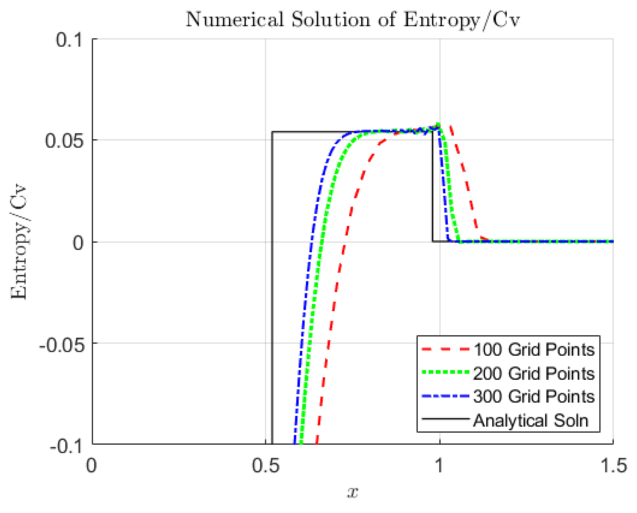
(b) *Pressure along the tube.*



(c) *Velocity along the tube.*



(d) *Mach number in the domain.*

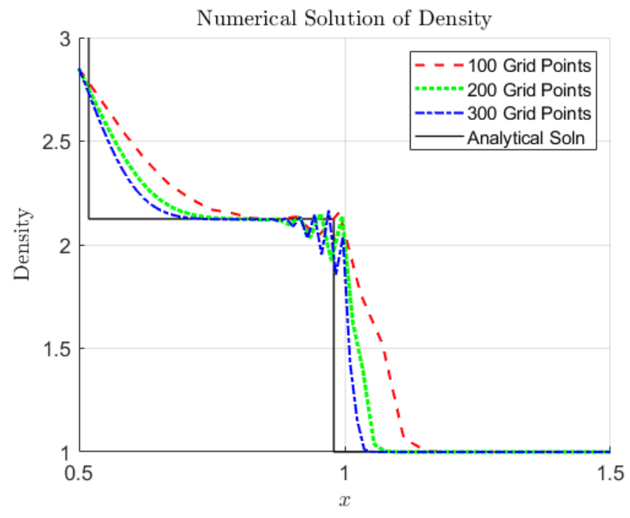


(e) *Entropy/Cv across the shock.*

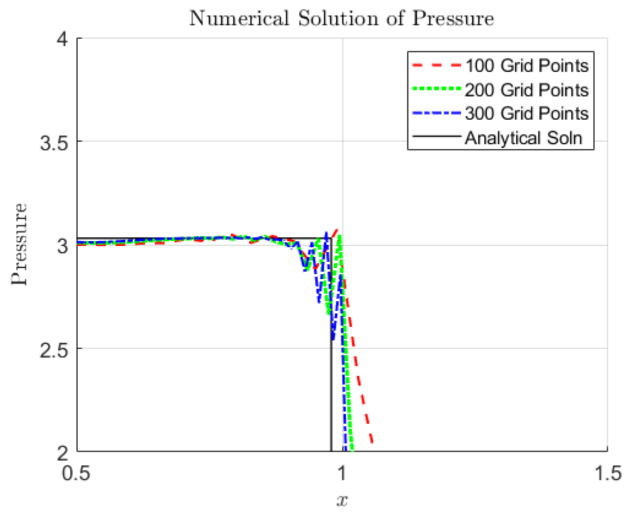
Figure 3: The numerical solution ($\sigma = 0.99$) versus the analytical solution computed at solution time t = 0.5. Numerical solution of the Riemann shock tube problem computed using three different meshes with N = 100, 200 and 300 equally spaced points in the interval $-2 \leq x \leq 2$.

A.2 Numerical Solution with $\sigma = 1.05$

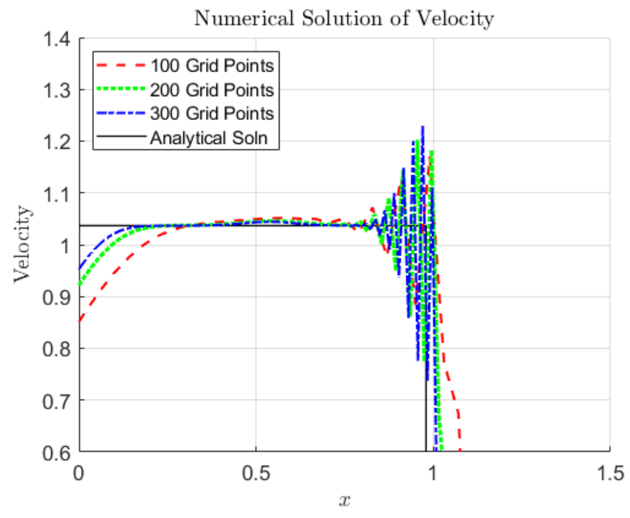
Results were plotted for $\sigma = 1.05$ to show growing numerical oscillations and onset of instability in solution as σ increases above 1.



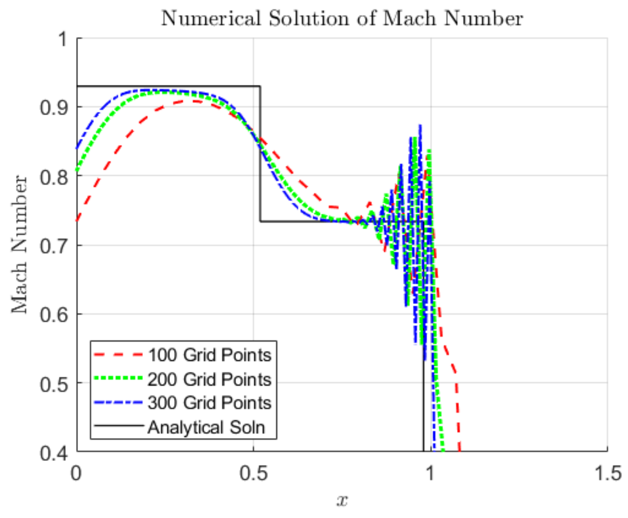
(a) *Air density along the tube.*



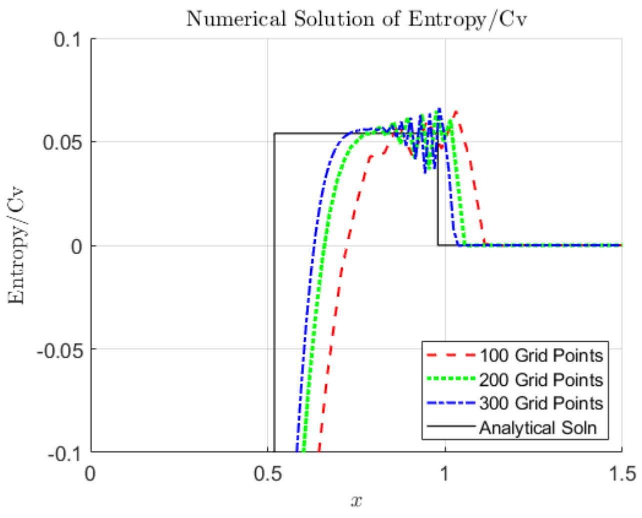
(b) *Pressure along the tube.*



(c) *Velocity along the tube.*



(d) *Mach number in the domain.*



(e) *Entropy/Cv across the shock.*

Figure 4: The numerical solution ($\sigma = 1.05$) versus the analytical solution computed at solution time t = 0.5. Numerical solution of the Riemann shock tube problem computed using three different meshes with N = 100, 200 and 300 equally spaced points in the interval $-2 \leq x \leq 2$.