

Computational Shock Tube Solver

University of Bath

Candidate Number: 24367

May 2, 2025



PH30110: Computational Astrophysics

Candidate Number: 24367

Department of Physics,
University of Bath, Bath
BA2 7AY, UK

Date Submitted: May 2, 2025

Contents

1	Question 1: A functional fluid solver	2
1.1	Setup A - Figure 1	2
1.2	Setup B - Figure 2	5
1.3	Alternative Methods	6
2	Question 2: Tracing the features of a shock tube	9
2.1	HLLC	9
2.2	Lax-F and Lax-W	10

Question 1: A functional fluid solver

1.1 Setup A - Figure 1

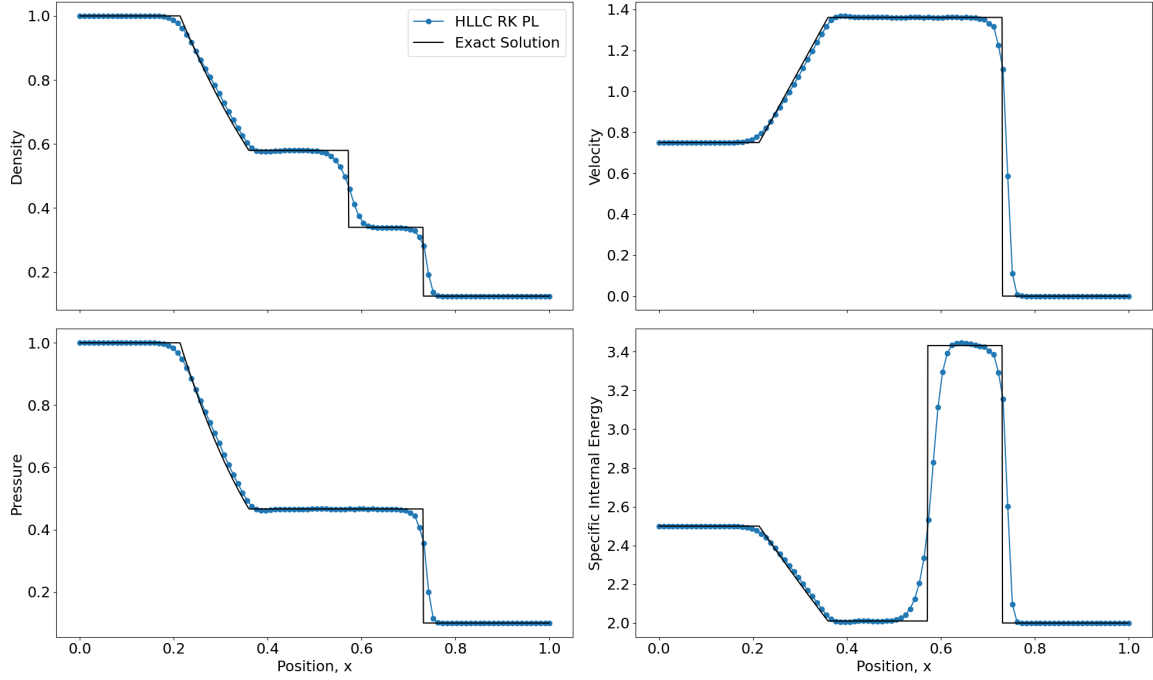


Figure 1: Comparison of the numerical solution (blue) versus the exact solution (black) shock tube result for problem A. The initial conditions are that the left state ($x < 0.3$ at $t = 0$) has $\rho = 1, p = 1$ and $v = 0.75$, whereas the right state has $\rho = 0.125, p = 0.1$ and $v = 0$. The snapshots are taken at $t = 0.2$. The close agreement between the numerical points and the exact solution curves demonstrates the accuracy of the HLLC scheme with Runge-Kutta time integration and piecewise linear (PL) reconstruction when using 100 zones.

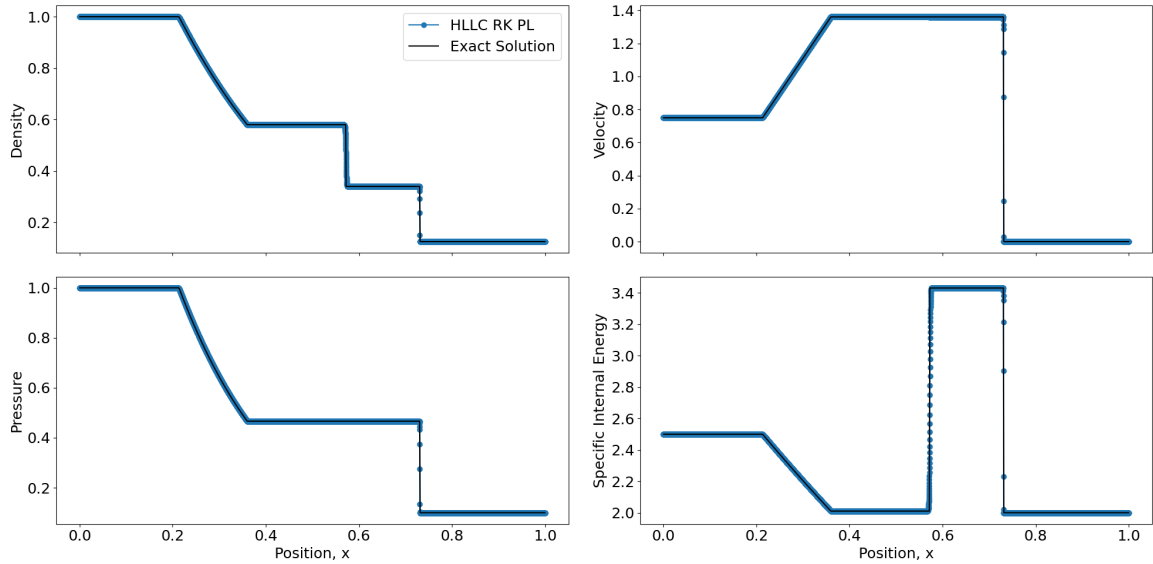


Figure 2: Comparison of the numerical solution (blue, labelled HLLC RK PL) versus the exact solution (black) shock tube result for problem A with 10,000 zones. The numerical solution (blue) more sharply resolves the discontinuities, closely matching the exact solution (black) in density, velocity, pressure, and specific internal energy. The higher resolution reduces numerical diffusion and preserves steep gradients more accurately than coarser grids.

Method Explanation:

The shock tube solver is designed to address one-dimensional compressible flow problems described by the Euler equations. The goal is to simulate the evolution of a gas initially separated into two distinct states by a discontinuity and then allowed to interact over time. At the core, the Euler equations model the conservation of mass, momentum and energy in a compressible fluid, and can be written in conservative form as

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0,$$

where the vector conserved variables is

$$U = \begin{pmatrix} \rho \\ m \\ E \end{pmatrix},$$

with ρ being the density, $m = \rho u$ the momentum (with u the fluid velocity) and E the total energy. The flux function $F(U)$ is given by

$$F(U) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ (E + p)u \end{pmatrix},$$

where the pressure p is related to the conserved variable via the equation of state for an ideal gas:

$$p = (\gamma - 1) \left(E - \frac{1}{2} \rho u^2 \right),$$

with $\gamma = 1.4$ for the case in problem A and problem B.

In the simulation, the computational domain is divided into discrete grid points, with initial conditions assigned according to whether the grid point lies on the left, x_{left} , or right, x_{right} , side of the discontinuity located as $x = x_{\text{disc}} = \text{Disc_loc}$. The classic shock tube cells to the left of x_{disc} are assigned one state (ρ_L, u_L, p_L) and those to the right (ρ_R, u_R, p_R) . An optional tracer variable is also introduced to help track features like contact discontinuities, used for question 2.

The spatial variation of the solution is handled using MUSCL (Monotonic Upstream-centered Scheme for Conservation Laws) reconstruction. In simple terms, while each cell initially stores only a single average value of the solution, MUSCL reconstructs a linear profile inside each cell. To do this, a slope is computed for each variable. However, to avoid introducing nonphysical oscillations near sharp discontinuities, the minmod slope limiter is applied. The minmod function is defined as

$$\text{minmod}(a, b) = \begin{cases} 0, & \text{if } ab \leq 0, \\ \text{sgn}(a) \min(|a|, |b|), & \text{otherwise,} \end{cases}$$

which means that if the differences on either side of a cell indicate a local extremum (or if they have opposite signs), the slope is set to zero. Otherwise, the smaller in magnitude difference is selected. This prevents overshoots and undershoots in the reconstructed profiles.

Using the computed slopes, left and right states at each interface (between cell i and cell $i + 1$) are approximated as follows:

$$U_i^R = U_i + \frac{1}{2} \Delta U_i, \quad U_{i+1}^L = U_{i+1} - \frac{1}{2} \Delta U_{i+1},$$

where ΔU_i is the limited slope for the cell i . This reconstruction improves the spatial accuracy of the method to second order in smooth regions.

At each interface, a Riemann problem is solved between the reconstructed left and right states. The code uses the HLLC (Harten–Lax–van Leer Contact) approximate Riemann solver, which is particularly effective in capturing the detailed wave structure of the Euler equations. In HLLC, the fastest left-going and right-going characteristic speeds, denoted S_L and S_R are first estimated. These speeds are determined from the local values of the velocity u and the sound speed c (obtained from the pressure p and using $c = \sqrt{\gamma p / \rho}$). Specifically, these speeds are given by

$$S_L = \min(u_L - c_L, u_R - c_R), \quad S_R = \max(u_L + c_L, u_R + c_R).$$

Once S_L and S_R are determined, the solver calculates an intermediate wave speed S^* corresponding to the contact discontinuity, using a formula that balances momentum and pressure differences across the interface. Based on the sign of these wave speeds, the flux at the interface is computed. For example, if $S_L \geq 0$, then the flux is entirely determined by the left state. If the speeds have differing signs, then a weighted combination incorporating the intermediate state is used. This piecewise construction of the flux ensures that the conservation laws are upheld and that shocks and contact discontinuities are treated accurately.

The time evolution of the solution is performed using a two-stage Runge–Kutta method, which is second-order accurate. The process begins by computing an intermediate solution U^* over a time step Δt based on the current flux divergence $L(U^n)$:

$$U^* = U^n + \Delta t L(U^n),$$

where $L(U^n) \approx -\frac{\partial F(U)}{\partial x}$. A second set of slopes and fluxes is then computed using this intermediate state, and the final update of the solution combines the contributions from both the initial and intermediate stages:

$$U^{n+1} = U^n + \frac{\Delta t}{2} [L(U^n) + L(U^*)].$$

This averaging step not only enhances accuracy but also helps mitigate potential instability issues. A crucial aspect of the time integration is the choice of Δt . The time step is determined using a CFL (Courant–Friedrichs–Lewy) condition:

$$\Delta t \leq \text{CFL} \frac{\Delta x}{\max_i (|u_i| + c_i)}$$

where Δx is the grid spacing, and $|u_i| + c_i$ represents the maximum signal speed at cell i . This condition ensures that the numerical scheme remains stable by preventing information from traveling more than one grid cell per time step.

From an implementation perspective, the computational domain is represented by arrays that hold the values of all the conserved variables. The code uses OpenMP to parallelize loops over grid points, which is critical for performance when dealing with a high number of grid points. In setting up the simulation, the initial conditions are distributed based on the position of the discontinuity, and special care is taken at the boundaries by copying interior values to the boundary cells. These simple boundary conditions (often called transmissive or outflow boundaries) suffice for a shock-tube setup, where the domain is typically large enough that boundary reflections do not significantly interfere with the region of interest.

An interesting addition is the use of a tracer variable that is advected with the flow. Initially, the tracer is set to one value on one side of the discontinuity and a different value on the other side. As the simulation evolves, changes in the tracer field help identify key features such as the contact discontinuity. By analysing the locations where the tracer crosses a threshold value (for example, 0.5), the program can estimate the positions of various flow features like shocks and rarefaction waves, and optionally output these positions for further analysis. The tracer is implemented for use in question 2, further described in the appropriate section.

In summary, the shock tube solver integrates several sophisticated numerical techniques to solve the Euler equations in one dimension. The method employs MUSCL reconstruction to achieve second-order spatial accuracy while avoiding spurious oscillations via the minmod limiter. The HLLC Riemann solver is used for accurately computing fluxes at cell interfaces, ensuring that the physical discontinuities such as shock waves and contact surfaces are well-resolved. Time integration is handled through a second-order Runge–Kutta method, with the time step dynamically set by the CFL condition to maintain stability. Together, these design choices allow the code to simulate complex wave interactions in compressible flow reliably and efficiently. The blend of mathematical rigor and computational efficiency makes this solver an excellent teaching tool and a foundational piece of technology in many computational astrophysics and fluid dynamics applications.

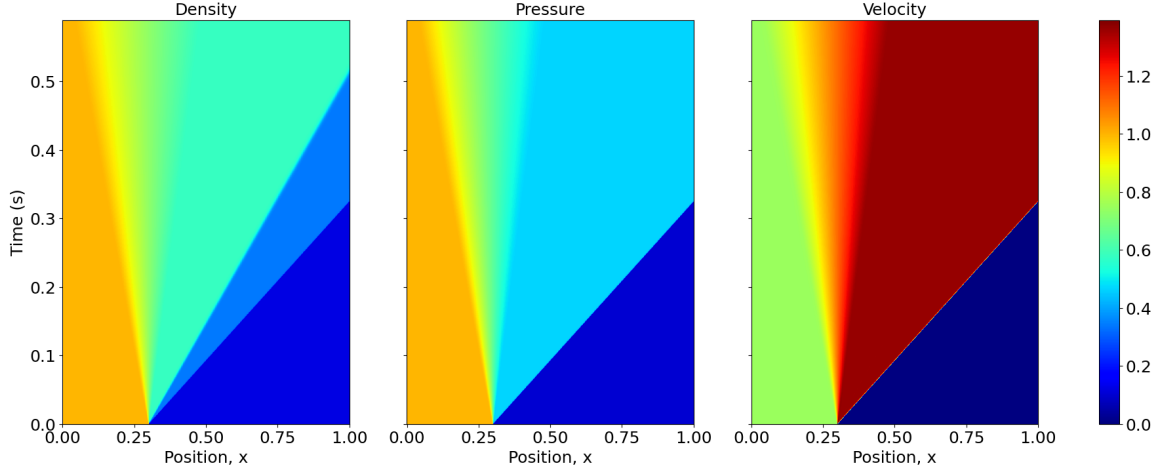


Figure 3: Spacetime evolution of the shock-tube problem from $t = 0$ to $t = 0.5$ s, shown as colour maps of (left) density, (middle) pressure, and (right) velocity. The vertical axis is time, and the horizontal axis is the spatial coordinate x . Red shading indicates higher values while blue indicates lower values, revealing the progression of shock, contact, and rarefaction features over the interval.

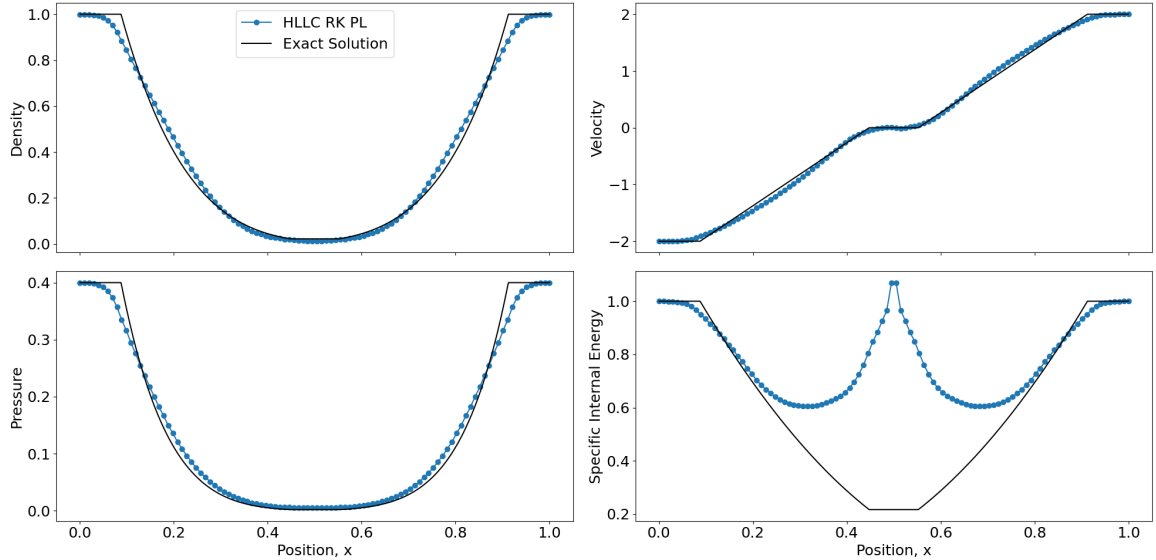


Figure 4: Numerical solution (blue) and exact solution (black) for the shock tube result for problem B. The initial conditions are that the left state ($x < 0.5$ at $t = 0$) has $\rho = 1, p = 0.4$ and $v = -2$, whereas the right state has $\rho = 1, p = 0.4$ and $v = 2$. The snapshots are taken at $t = 0.15$ with 100 zones. The good agreement between the numerical (HLLC RK PL) and exact profiles demonstrates accurate capture of the shock, contact discontinuity, and rarefaction wave.

1.2 Setup B - Figure 2

To further test the solver discussed in section 1.1, Figure 4 illustrates a comparison where the numerical solution (displayed in blue) does not capture the exact solution (shown in black) as well as it does in the previous problem. In this case, while the HLLC RK PL method still reproduces the overall structure of the flow, it exhibits noticeable discrepancies near the discontinuities. The shock front, for example, is slightly under-resolved, and subtle oscillations may appear in the contact region. These deviations highlight the limitations of the numerical method in exactly replicating the idealised sharp transitions found in the exact solution, emphasising the trade-off between numerical stability and precision in complex flow regimes. This is still present, even when the number of zones is increased to 10,000, displayed in Figure 5 where a sharp peak in the Specific Internal Energy plot is still present.

Figure 4 thus serves as a reminder that even though modern techniques like the HLLC method

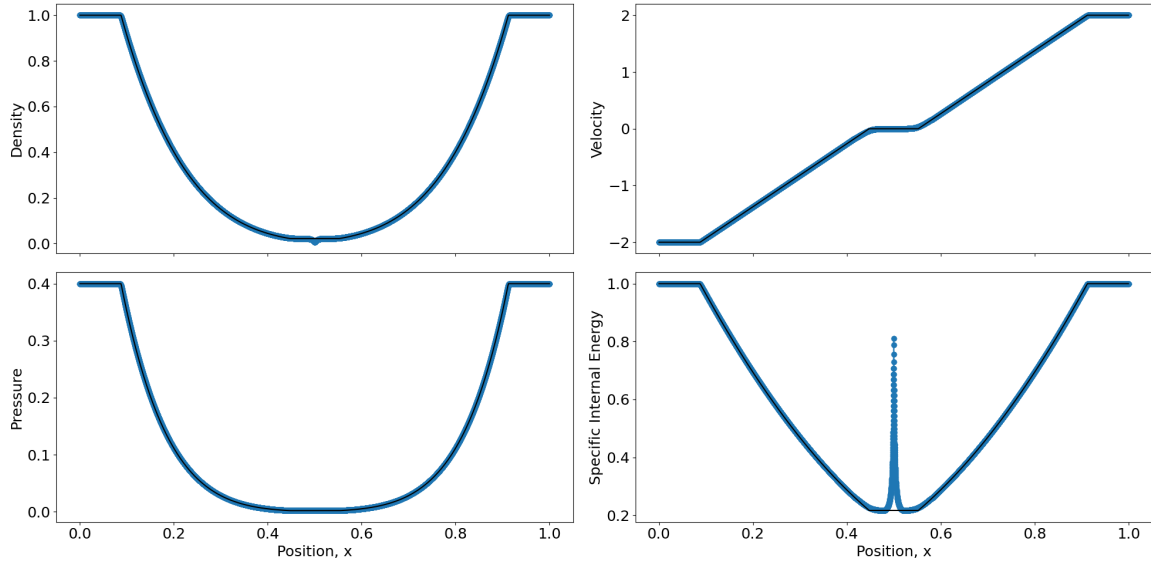


Figure 5: Numerical solution (blue) and exact solution (black) for the shock tube result for problem B. The initial conditions are that the left state ($x < 0.5$ at $t = 0$) has $\rho = 1, p = 0.4$ and $v = -2$, whereas the right state has $\rho = 1, p = 0.4$ and $v = 2$. The snapshots are taken at $t = 0.15$ with 10,000 zones. The good agreement between the numerical (HLLC RK PL) and exact profiles demonstrates accurate capture of the shock, contact discontinuity, and rarefaction wave.

combined with Runge–Kutta time integration and piecewise linear reconstruction offer significant improvements over classical schemes such as Lax–Friedrichs or Lax–Wendroff, capturing the exact solution remains a challenging endeavour when dealing with steep gradients and rapid wave interactions.

1.3 Alternative Methods

The program design gives the user the option to toggle between different solvers, other than the most advanced one already discussed. To mention two alternative methods, the Lax–Friedrichs and Lax–Wendroff methods both represent classic finite difference schemes developed to solve hyperbolic partial differential equations. They have long been used as introductory methods in computational fluid dynamics but come with notable trade-offs that limit their performance when compared to more modern approaches like the HLLC scheme combined with Runge–Kutta time integration and piecewise linear (PL) reconstruction.

The Lax–Friedrichs method is a first-order accurate scheme well known for its simplicity and robustness. Its primary drawback is its inherent diffusivity: it tends to smear out steep gradients and discontinuities such as shocks or contact surfaces. While this smoothing behaviour can contribute to numerical stability, it often results in a loss of key physical features. In practical terms, the excessive numerical diffusion can lead to solutions that are overly smooth, obscuring the detailed structure of shock waves—a critical aspect for accurate modeling of compressible flows. This is shown in Figure 6 when compared to the HLLC RK PL method outlined above. Figure 8 demonstrates this solver working for the setup in problem B. As shown, it quite clearly does not numerically solve as well as the more complex, HLLC RK PL method.

In contrast, the Lax–Wendroff method is designed to be second-order accurate in both space and time. By incorporating additional terms in its Taylor series expansion, it achieves a higher level of accuracy in regions where the solution is smooth. However, this scheme is not without its challenges. The Lax–Wendroff method is prone to oscillations near discontinuities—a phenomenon often referred to as numerical dispersion or the Gibbs phenomenon. These oscillations can manifest as spurious wiggles around shock fronts and contact discontinuities, making it less reliable when capturing sharp transitions in the flow. A comparison of the Lax–Friedrichs and Lax–Wendroff methods are shown in Figure 7. Additionally, Figure 9 demonstrates that the Lax–Wendroff is incapable of solving the complex setup for problem B, further highlighting the need for more complex and accurate solvers.

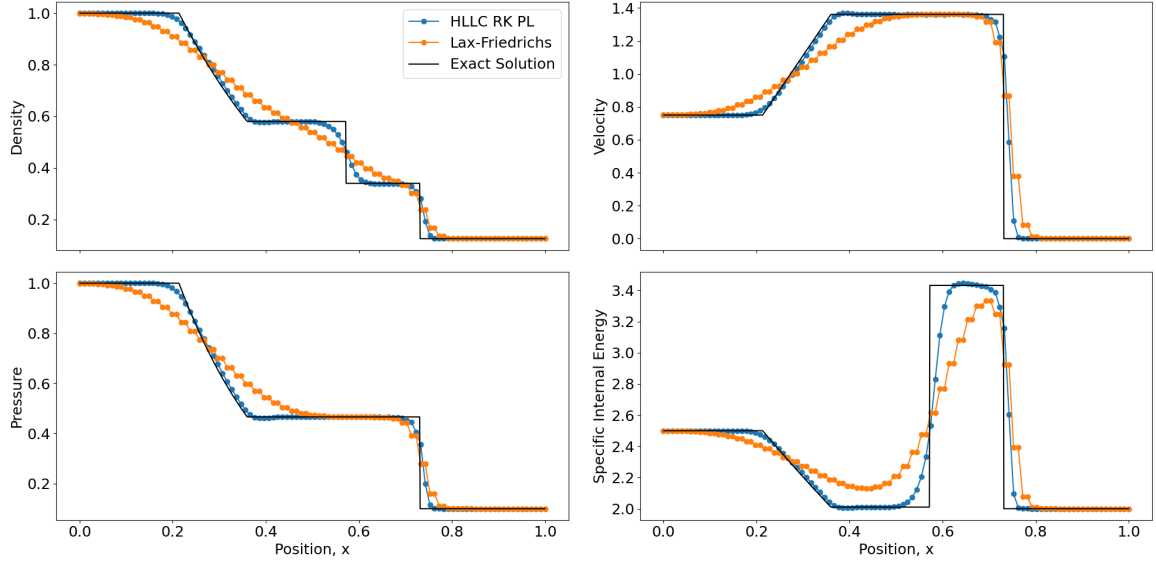


Figure 6: Comparison of the HLLC scheme with Runge–Kutta time integration and piecewise linear reconstruction (blue) against the Lax–Friedrichs scheme (orange) for the shock tube in problem A, at $t = 0.2$ s using 100 zones. The HLLC method preserves sharper discontinuities, while the Lax–Friedrichs scheme is more diffusive near shocks.

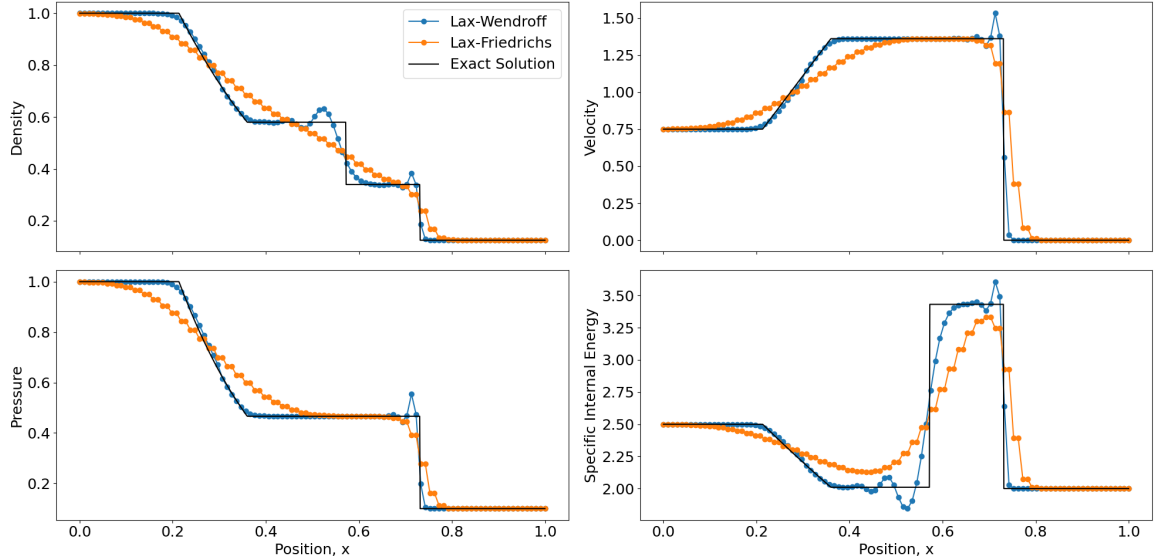


Figure 7: Comparison of numerical solutions from the Lax–Wendroff scheme (blue) and Lax–Friedrichs scheme (orange) with the exact solution (black) for the shock tube in problem A, at $t = 0.2$ s using 100 zones. While both schemes reproduce the main features of the exact solution, the Lax–Friedrichs method is more diffusive near discontinuities, whereas the Lax–Wendroff scheme captures steeper gradients, albeit with slight oscillations around shocks.

When compared to these classical approaches, the HLLC (Harten–Lax–van Leer Contact) method combined with a Runge–Kutta time integrator and piecewise linear reconstruction (often implemented via MUSCL techniques with a minmod limiter) stands out for several reasons. The HLLC scheme is tailored to accurately resolve the full wave structure of the Euler equations by distinguishing between left-, right-, and contact waves. This capability, along with the second-order accuracy in both space and time provided by the Runge–Kutta integration and PL reconstruction, allows for a much sharper and more faithful representation of discontinuities like shocks and contact surfaces. The careful balance between accuracy and monotonicity (enforced by the slope limiter) helps prevent the unphysical oscillations that the Lax–Wendroff method tends to exhibit while avoiding the excessive smoothing characteristic of the Lax–Friedrichs method.

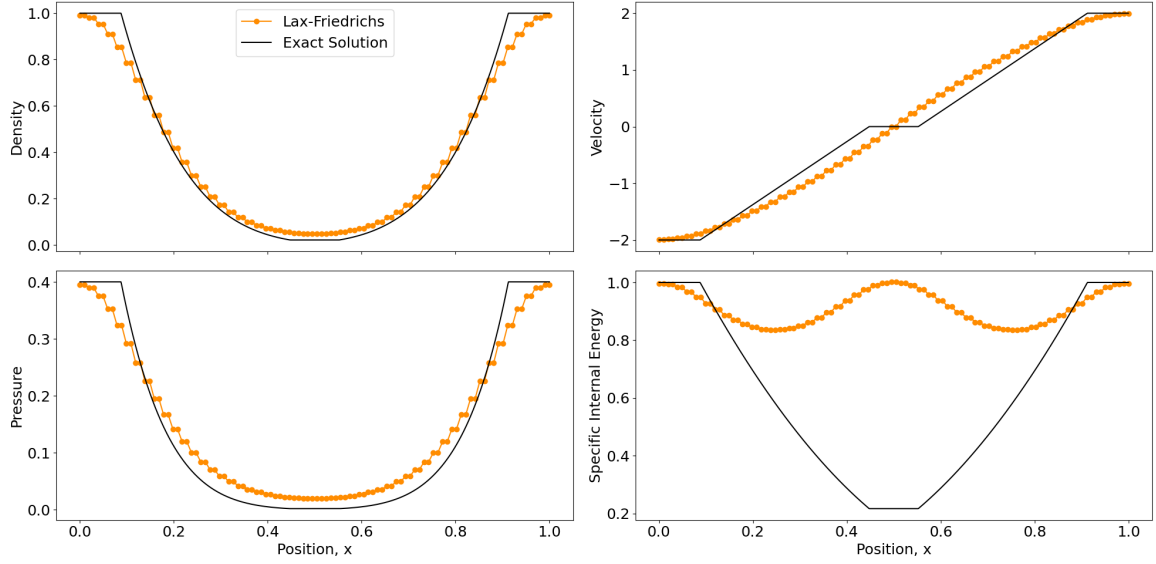


Figure 8: Numerical solution (orange) and exact solution (black) for the shock tube result for problem B, using the Lax-Friedrichs method. The initial conditions are that the left state ($x < 0.5$ at $t = 0$) has $\rho = 1, p = 0.4$ and $v = -2$, whereas the right state has $\rho = 1, p = 0.4$ and $v = 2$. The snapshots are taken at $t = 0.15$ with 100 zones.

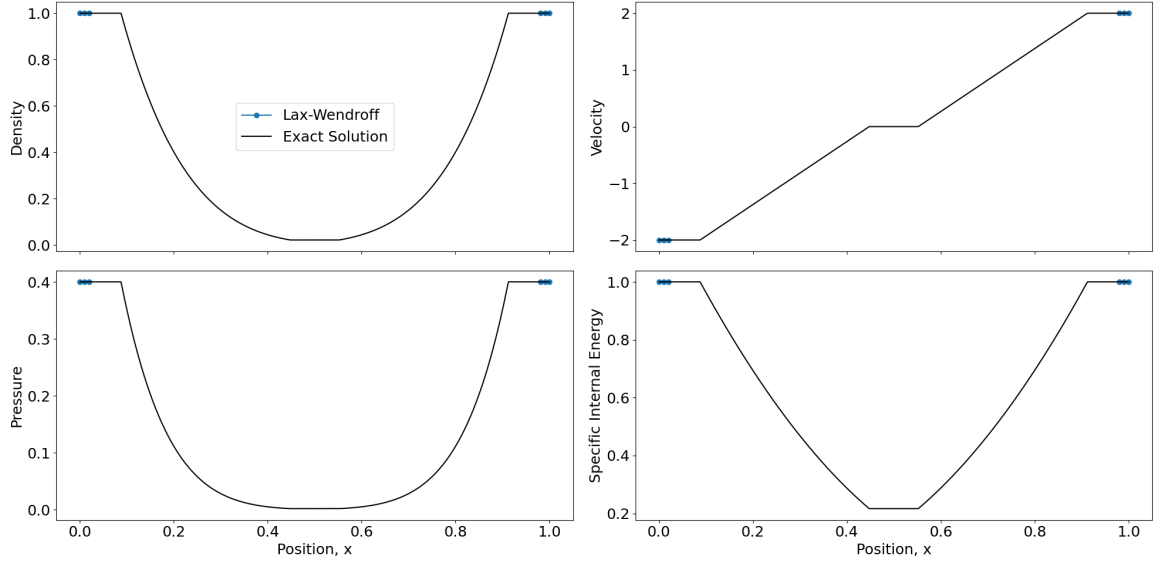


Figure 9: Numerical solution (orange) and exact solution (black) for the shock tube result for problem B, using the Lax-Wendroff method. The initial conditions are that the left state ($x < 0.5$ at $t = 0$) has $\rho = 1, p = 0.4$ and $v = -2$, whereas the right state has $\rho = 1, p = 0.4$ and $v = 2$. The snapshots are taken at $t = 0.15$ with 100 zones. The plot shows that the Lax-Wendroff is not capable of solving the more complex setup in problem B.

The basic solvers were first implemented to offer a foundation, before later developing into the more complex HLLC solver. While Lax-Friedrichs and Lax-Wendroff methods have historically been valuable and instructive, their inherent diffusion and oscillatory tendencies respectively, make them less effective for problems requiring precise shock capturing. Modern solvers like the HLLC RK PL combine the best of both worlds, delivering high accuracy with robust shock resolution and minimal spurious oscillations.

Question 2: Tracing the features of a shock tube

2.1 HLLC

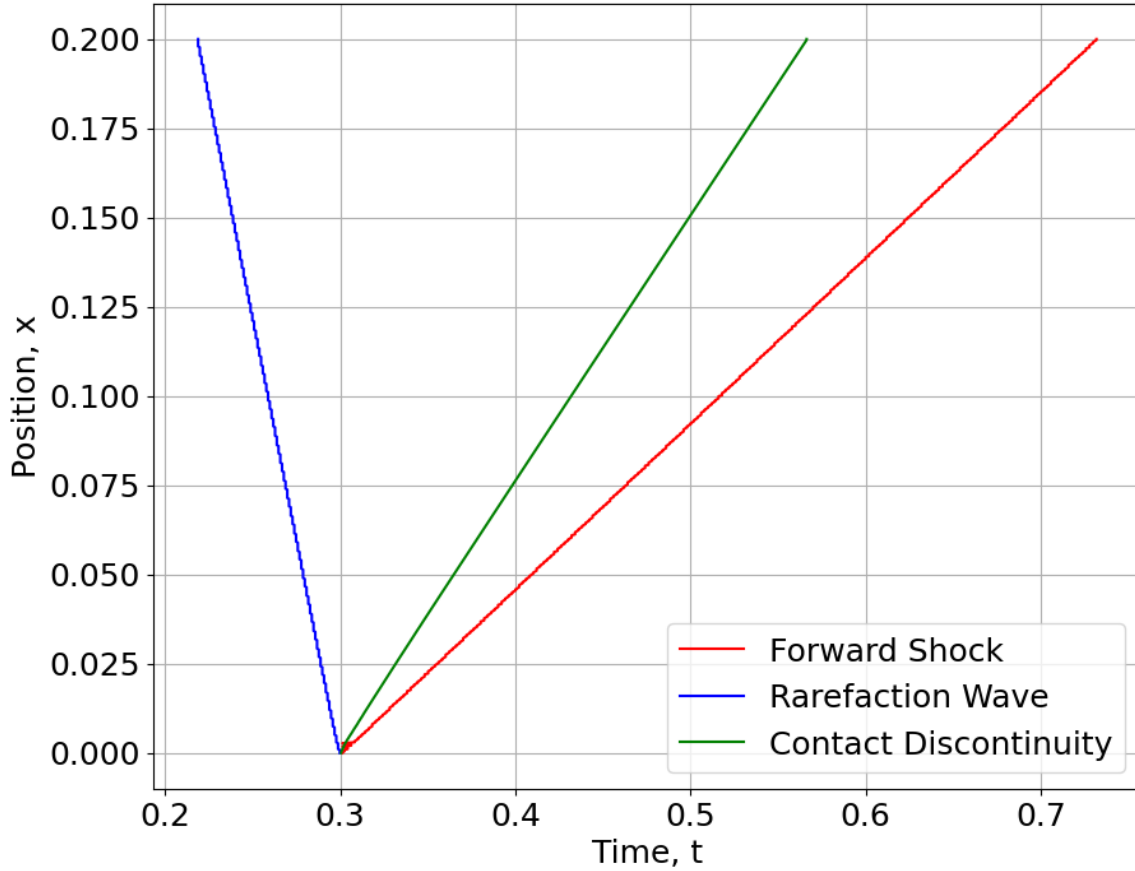


Figure 10: Time evolution of the main wave features in a one-dimensional shock-tube problem, showing the forward shock (red), rarefaction wave (blue), and contact discontinuity (green) as they propagate through the tube using 1000 zones and the HLLC solver. The horizontal axis is time, and the vertical axis is the position of each feature. This tracking provides a clear illustration of how shocks, contacts, and rarefactions move at different characteristic speeds within the flow.

For Question 2, the shock tube program in Question 1, which advances the one-dimensional Euler equations using MUSCL reconstruction and the HLLC approximate Riemann solver, was extended to track the evolving wave features within the flow. In the underlying method, second-order spatial accuracy is achieved by reconstructing a piecewise linear profile in each cell, with a minmod limiter ensuring that no spurious oscillations arise near discontinuities. The HLLC solver then accurately computes the interface fluxes by considering the fastest left- and right-going waves, along with an intermediate contact wave. This flux information is integrated in time via a two-stage Runge–Kutta method (Heun’s method), yielding second-order temporal accuracy as well.

The approach for Question 2 was the addition of an advected tracer variable. Initially set to different constant values on either side of the discontinuity (for example, 1.0 on the left and 0.0 on the right), the tracer serves as a marker for the contact discontinuity. As the simulation evolves, the point where the tracer value transitions through approximately 0.5 is used to precisely identify the location of the contact surface. In parallel, the code identifies the forward shock position by locating the largest jump in pressure between adjacent cells, and it recognises the rarefaction region by detecting where the density drops below a threshold relative to its initial value.

At each time step, the positions of these key wave features, the forward shock, the contact discontinuity, and the rarefaction wave are recorded and were later visualised in Figure 10. This dynamic visualisation provides a clear illustration of how each wave propagates: the forward shock advances rapidly through the shock tube, the contact discontinuity moves at an intermediate speed,

and the rarefaction wave expands more gradually, in the opposite direction (inwards in the case of the star). The ability to track these features concurrently not only reinforces the physical fidelity of the simulation but also offers valuable insight into the characteristic speeds and interactions present for compressible flows.

In the broader context, these wave features directly parallel the mechanisms involved in supernova shock breakout, where a forward shock emerges from the star and interacts with its surroundings, while a rarefaction wave or reverse shock propagates back inward. The tracer-based contact discontinuity highlights the boundary between different material layers, underscoring the importance of accurately tracking fluid interfaces as they move. By plotting each wave feature over time, as illustrated in Figure 10, insights into the characteristic speeds of shocks, contacts, and rarefactions can be seen which ultimately govern the observable emission and dynamical evolution in explosive events. This systematic approach, combining a robust numerical scheme with effective wave-feature tracking, provides a valuable framework for interpreting the early-time signals that might be detected by modern astrophysical observatories.

Overall, the approach used in Question 2 leverages numerical techniques such as MUSCL reconstruction, HLLC flux evaluation, and second-order Runge–Kutta time integration, while the addition of a tracer field allows for detailed tracking of multiple wave phenomena. This combination of methods underscores both the mathematical rigour and computational efficiency required to accurately capture the complex wave dynamics observed in astrophysical and fluid dynamic applications.

2.2 Lax-F and Lax-W

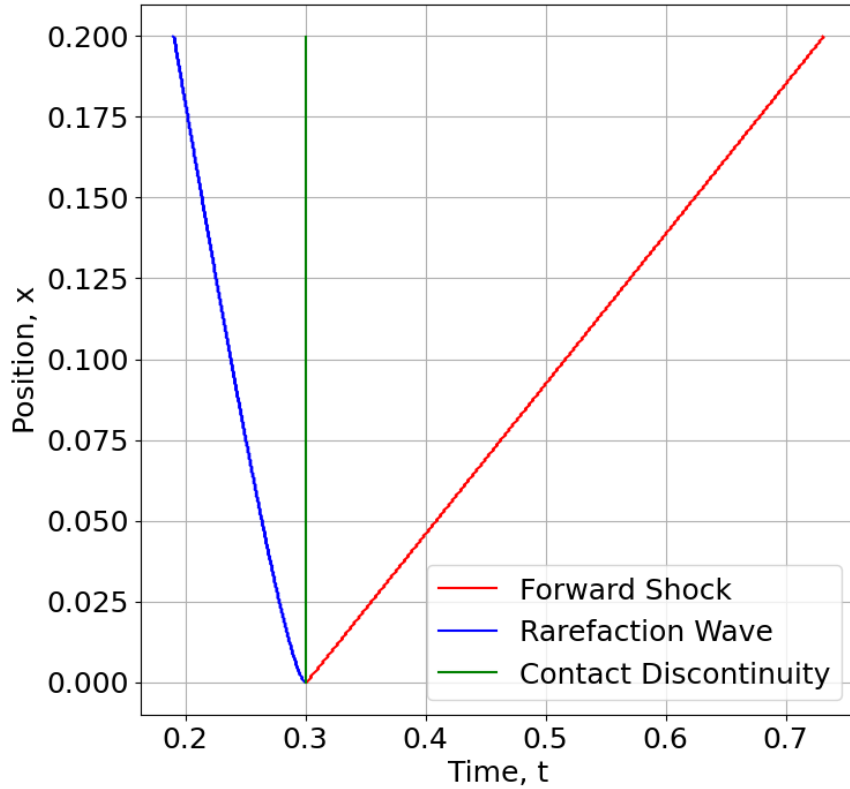


Figure 11: Time evolution of the main wave features in a one-dimensional shock-tube problem, showing the forward shock (red), rarefaction wave (blue), and contact discontinuity (green) as they propagate through the tube using 1000 zones using the Lax-Friedrichs solver. The horizontal axis is time, and the vertical axis is the position of each feature. This tracking provides a clear illustration of how shocks, contacts, and rarefactions move at different characteristic speeds within the flow. The Lax-Friedrichs solver is unable to solve the contact discontinuity.

End of Report

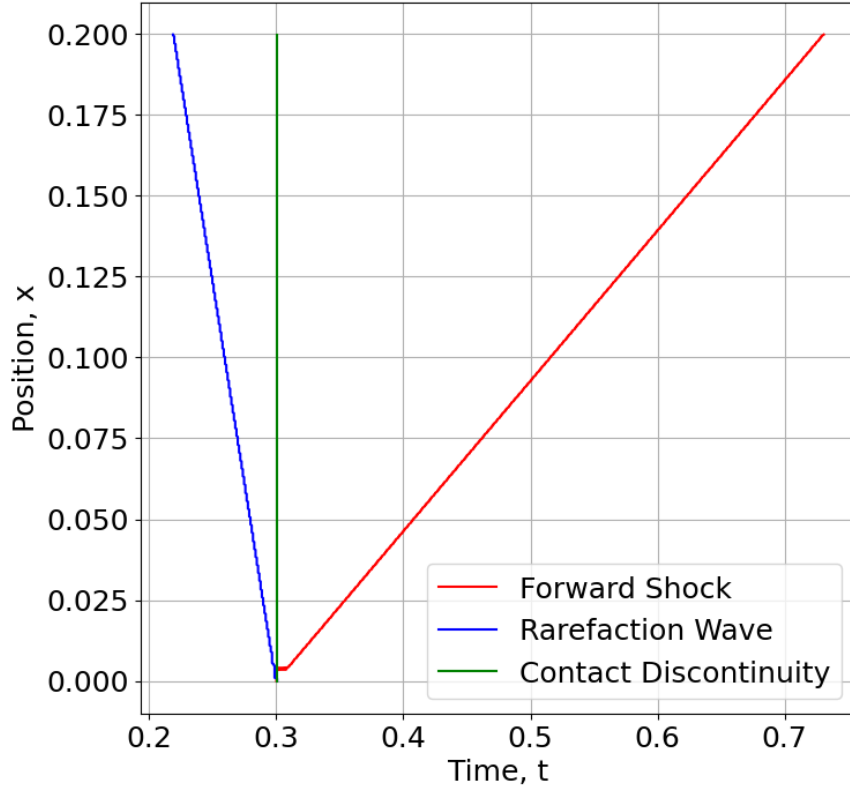


Figure 12: Time evolution of the main wave features in a one-dimensional shock-tube problem, showing the forward shock (red), rarefaction wave (blue), and contact discontinuity (green) as they propagate through the tube using 1000 zones using the Lax-Wendroff solver. The horizontal axis is time, and the vertical axis is the position of each feature. This tracking provides a clear illustration of how shocks, contacts, and rarefactions move at different characteristic speeds within the flow. Similar to the Lax-Friedrichs solver, the Lax-Wendroff also cannot resolve the contact discontinuity.