Computational Continuum Modelling
Practical 6: Exact Riemann solver for the Euler equations

Having an exact Riemann solver is useful, both for using in Godunov's method (and higher-order extensions), and also for plotting next to numerical solutions.

Although an exact solver can be written as a stand-alone code, it is recommended that you write this solver as a function within one of your Euler equation codes. It will be a function that takes in two states (either conserved states or primitive states from the initial data) as well as a final time and geometric information, and returns the exact solution for each cell in the computational domain. This way, you can use it to add data to your output file, always with the same resolution as your simulation, to compare numerical solutions to the exact one.

In doing this (rather than a stand-alone code), it is straightforward to modify this exact Riemann solver for future implementations of Godunov's method; you still take in two states to the exact solver, and return the state at $x_{i+1/2}$ (rather than the entire solution).

Some useful steps towards implementing the solver:

- Start with just solving for p^* - for the tests in Toro's book, the exact values for this variable are available. This will require defining a function for $f(p^*, \mathbf{u}_L, \mathbf{u}_R)$, and its derivative, and then implementing a root-finding algorithm.

  The derivative is with respect to p^* , and is fairly straightforward to calculate (and is also in Toro's book).

  For the root finding algorithm, the Newton-Raphson scheme from lectures is recommended.

  The accuracy to which you want to find p^* is something you need to define - two standard criteria:

  $$|p_{\text{new}} - p_{\text{old}}| < \epsilon, \quad \frac{|p_{\text{new}} - p_{\text{old}}|}{p_{\text{old}}} < epsilon$$

  with ϵ a suitable small number (between 10^{-10} and 10^{-6} is normally a good choice). These two definitions are useful - the first is obvious, but the second applies a scaling in case pressure is very small or large. Small numbers may otherwise exit the loop too soon, large numbers may have rounding precision larger than ϵ and may never converge

- Once you have p^* , compute v^* , ρ_L^* and ρ_L^* and S_L and S_R (if these last two exist). Some of these values can also be checked within the tables in Toro's book.

- At this stage, your solver is (almost) ready to be used in Godunov's method - you need to know the state at the centre of the Riemann problem. To do this, you need to know where the waves are (shock wave and head and tail of rarefactions). Things to check:

  - Is the speed of the left shock (S_L), or tail of the left rarefaction ($v_L - c_{s,L}$) positive? If yes, then all waves are to the right of the centre, and the intermediate state is u_L
  - Are all waves to the left of the centre (repeat as above, with right-hand states - note, rarefaction tail is $v_R + c_{s,R}$)
  - Is the centre in the middle of a rarefaction? Do either $v_L - c_{s,L}$ and $v^* - c_{s,L}^*$ have different signs, or $v_R + c_{s,R}$ and $v^* + c_{s,R}^*$ have different signs. If it is true for one of these, this wave covers the centre. Use equation (4.56) from Toro with $x = 0$ to compute the intermediate state.
  - Otherwise, is $v^* < 0$? If so, use $\mathbf{u}_L$, else use $\mathbf{u}_R$.

- In order to compute the profile of a Riemann problem solution at time t, you now need to discretise this solution. To compute which part of the solution a given point is in, note that waves are governed by

$$\frac{x}{t} = S$$

  where S is the speed of a given feature (shock, rarefaction head or rarefaction tail). You can work from left to right, setting values of the solution as $\mathbf{u}_L$ until the solution crosses the left-most wave (defined by S_L or $v_L - c_{s,L}$). Then move to the next appropriate state, and so on, until you cover the domain. Note that if you are in the middle of a rarefaction, you will need to compute the appropriate state from the equations given in lectures.

- Output this solution, either in its own file, or alongside your simulation results