Introduction to Computational Multiphysics
Practical 2: Level set methods.

The aim of this practical is to introduce a level set to your one-dimensional CFD code(s) in preparation for implementing a sharp interface method. This will require introducing a new vector quantity to store the level set function throughout your computational domain. It is recommended that you keep the level set function separately from your vector of conserved variables since it used a different numerical method to evolve it. The level set function will obey the evolution equation

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0.$$

where $\mathbf{v}$ is the velocity of the fluid. The numerical evolution of the level set function can be treated separately from the CFD evolution, but will happen with the same time step. This time step is still calculated using the CFL condition for the fluid equations - the time step from the level set equation is limited by v, not $|v| + c_s$, so never contributes to the overall stable time step. Transmissive boundaries are appropriate for the level set function.

In addition to the level set function, this practical also requires you to introduce a second set of conserved variables into your code. Again this will be required for a sharp interface method. Here it is recommended that you store your two variables separately, e.g. you have a vector of array objects $\mathbf{u}_1$ and a second one $\mathbf{u}_2$. At the moment these two objects will be updated using exactly the same flux routines, but they may contain different initial data. It is likely to be best (but not essential) when outputting to place $\mathbf{u}_1$, $\mathbf{u}_2$ and ϕ in a single file.

Exercises:

1. Add a level set function to your 1D Euler equation code. The first-order upwind difference scheme should be implemented to evolve the level set function. The velocity field for this update should come from your Euler equation solution.

   To test the level set function, repeat Toro's five tests. The initial data for the level set function should be that $\phi = 0$ at $x = 0.5$ (the initial discontinuity in the equations). It does not matter if $\phi > 0$ corresponds to $x < 0.5$ or $x > 0.5$.

   For these tests, you expect the level set function to follow the contact discontinuity (i.e. the $\phi = 0$ point should be located somewhere within this region). To show this, you only need to plot the level set function itself and the density. Ideally these should be on the same plot. Some tests may perform better than others (test 2 might not work as expected).

2. Add a second vector of conserved variables to your code. Every function call within your time update loop (and initial data) will need to be called twice, once for $\mathbf{u}_1$ and once for $\mathbf{u}_2$.

   Initially give $\mathbf{u}_1$ and $\mathbf{u}_2$ the same initial data, and verify (for yourself) that they give the same result. Then give $\mathbf{u}_1$ and $\mathbf{u}_2$ different initial data (taken from Toro's tests) and verify that each one now gives the correct individual result. You do not need to provide any plots from this part of the exercise. Since this exercise is simply to test the implementation of the second set of variables, don't worry about what velocity the level set function has.

3. We shall now use the two velocity fields from the last task to alter how the level set function evolves. For $\phi < 0$ the velocity field should come from $\mathbf{u}_1$ and for $\phi > 0$ it should come from $\mathbf{u}_2$.

   Initially, verify for yourself that if the initial data for $\mathbf{u}_1$ and $\mathbf{u}_2$ is equal, nothing changes for the level set position from the first task. Once this works, some examples to try are:

   - Initial data for $\mathbf{u}_1$ comes from test 1, and for $\mathbf{u}_2$ from test 3.
   - Initial data for $\mathbf{u}_1$ comes from test 1, and for $\mathbf{u}_2$, also from test 1, but with the initial data flipped (so that e.g. $(1, 0, 1)$ is the state for $x > 0.5$)
   - Initial data for $\mathbf{u}_1$ is from test 2 and any other initial data is used for $\mathbf{u}_2$
   - Initial data for $\mathbf{u}_2$ is from test 2, and for $\mathbf{u}_1$ use the initial data from the previous example, but flipped. Physically you would expect this to give a symmetric location of $\phi = 0$ to the previous example, but given the behaviour of the second-order methods we have worked with, it might not.

   For these tests, it is less obvious what should be plotted. Density plots for both $\mathbf{u}_1$ and $\mathbf{u}_2$ should be shown with the level set function. It might also be interesting to see a velocity plot where for $\phi < 0$ the velocity of $\mathbf{u}_1$ is plotted, and for $\phi > 0$, the velocity of $\mathbf{u}_2$ is plotted. This can be done in e.g. gnuplot and does not have to be an additional output quantity.