I worked with John Pharo on this worksheet. It took about three hours.

A Simple 1D SPH Code Applied to a Shock Tube Problem

I first implemented the smoothing kernel, the accelerations, and the RHS of the internal energy equation in `ws15.py`. The structure of this code is as follows.

First, some number of equally spaced particles along a 1-dimensional axis are instantiated. These are then given appropriate masses to fit an initial density profile. Other initial quantities such as internal energy, pressure, and sound speed are set, and the initial particle velocities are set to 0. Then the density is recalculated using the smoothing kernel on the particles and the initial time step is set, which depends on the sound speed.

Then the main loop begins. The following happens at each iteration. The artificial viscosity is calculated at each particle which is a dependency in calculating the accelerations of the particles. The accelerations of each particle are then calculated, as well as the velocities at half time-steps, which are then used to calculate the righthand side of the energy equation and to update the velocities at full time-steps. Based on the energy equation RHS, the internal energy of each particle is integrated via the Forward Euler method. The updated particle positions are then calculated from the half time-step velocities via the Forward Euler method. The "neighbors" of each particle, that is, the nearby particles, are determined, then used to get the density at each particle location based on the smoothing kernel. Now that the density has been calculated, related quantities like pressure, sound speed, and the new time step are all calculated. Then the loop repeats.

The figures below represent the particle positions as well as the smoothed densities at those positions for various times in the simulation. It is clear that there is some error in the simulation code because the densities of the boundary particles immediately change due to boundary effects, even though the simulation is of a medium with step-wise density distribution that extends to positive and negative infinity.

I couldn't get my Fortran compiler to work, so I'm skipping that part.

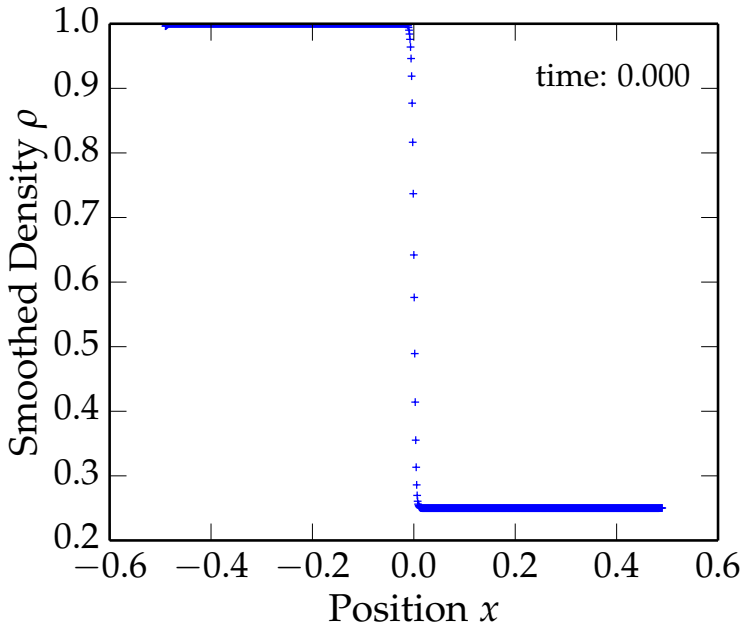

Figure 1: Initial density profile.

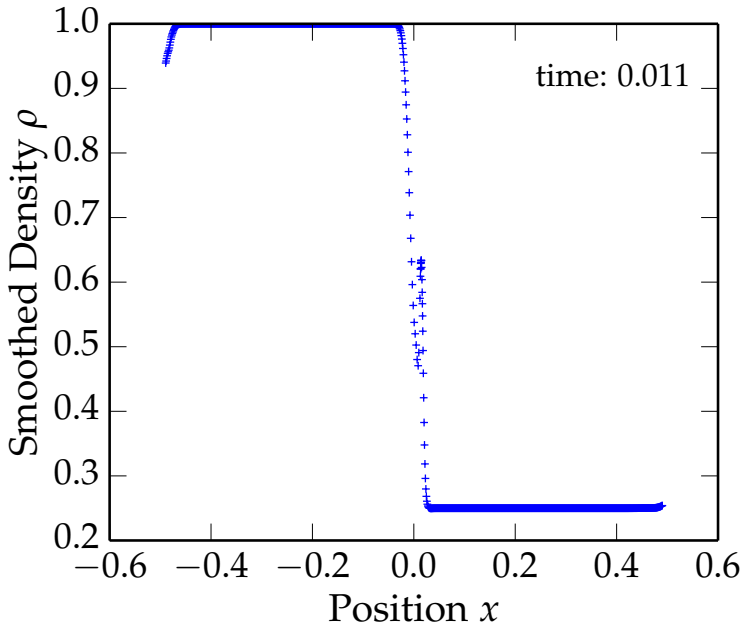

Figure 2: Density profile at the 10th simulation iteration, which corresponds to a time of 0.011.

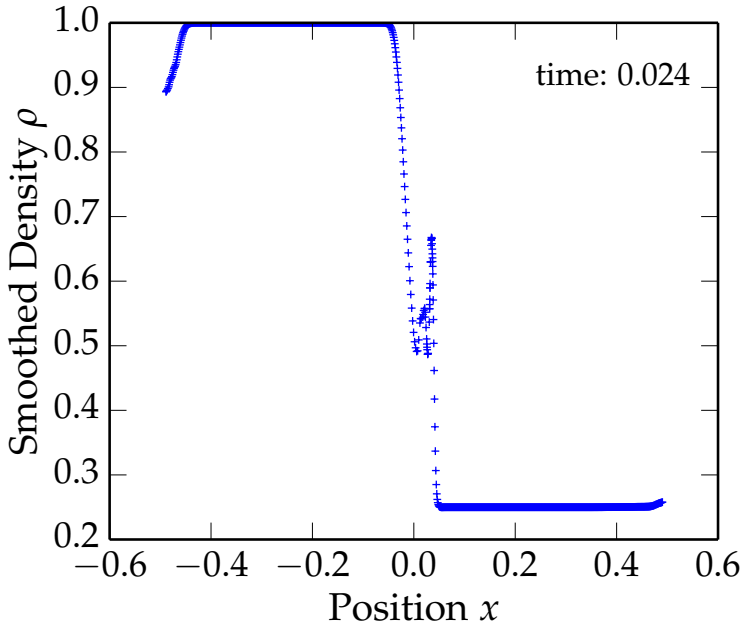

Figure 3: Density profile at the 20th simulation iteration, which corresponds to a time of 0.024.

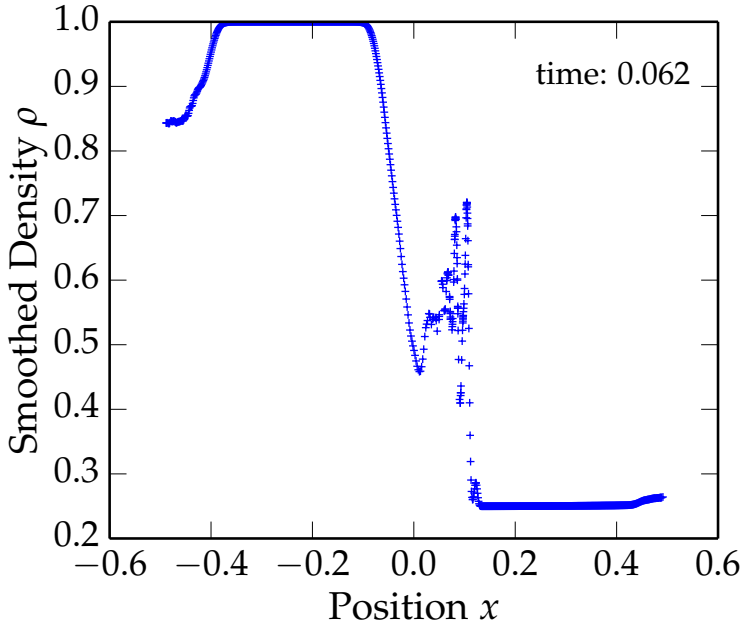

Figure 4: Density profile at the 50th simulation iteration, which corresponds to a time of 0.062.

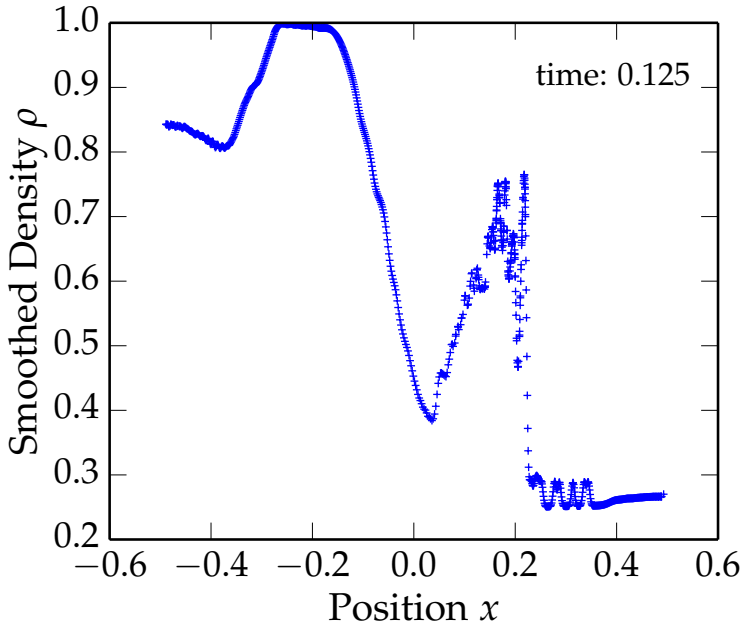

Figure 5: Density profile at the 100th simulation iteration, which corresponds to a time of 0.062.

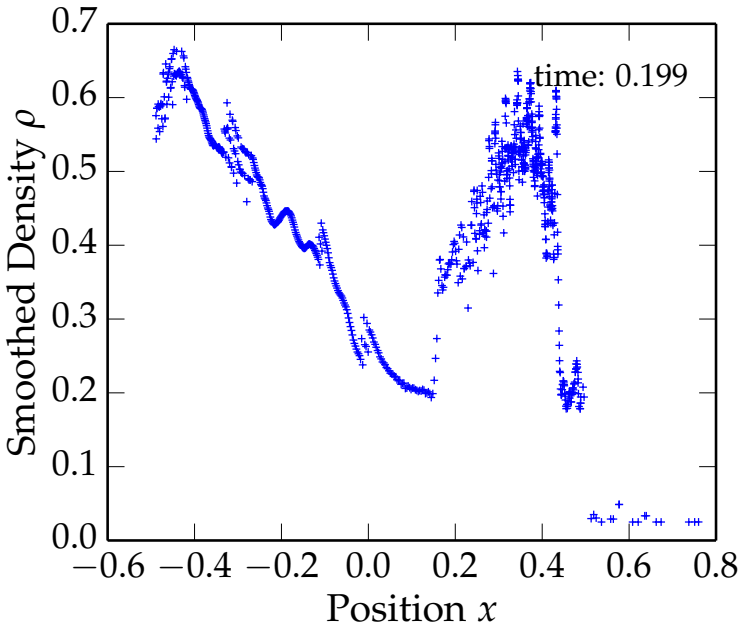

Figure 6: Density profile at the 175th simulation iteration, which corresponds to a time of 0.199.