PHYM004 Final Project – SPH

Jack Morton

1st April 2022

1 Introduction

1.1 Brief Outline of the Project

This project is an implementation of Smoothed Particle Hydrodynamics (SPH) in one dimension, with adaptive timestepping and per-particle smoothing lengths. The model includes artificial viscosity as well as conductivity, as described in [1]. There are Euler and second-order Runge-Kutta integration methods implemented and available for use via the command line arguments, though the Runge-Kutta method will be used throughout the rest of this report due to its superior performance. Boundary conditions are enforced with *ghost particles* that mirror the positions of the real particles across the two ends of the box. The performance of the algorithms is evaluated in **Section 2**, and results are compared with the analytical solutions of the Sod shock tube and "colliding streams" problem in **Section 3**. The effects of changing the viscosity parameters is also investigated in **Section 4**.

The adaptive timestep is calculated as the minimum of a) a CFL condition

$$\delta t_{\mathrm{CFL}} = \min_a \frac{h_a}{c_{s,a}} \tag{1}$$

where h_a is particle a's smoothing length and $c_{s,a}$ is the speed of sound at the particle's position, and b) a term involving the accelerations of the particles

$$\delta t_f = \min_a \sqrt{\frac{h_a}{|a_a|}} \tag{2}$$

where a_a is the acceleration of particle a [2].

Per-particle (variable) smoothing lengths are calculated by enforcing the following relation between density and smoothing length [1, 3]

$$h_a = \eta \frac{m_a}{\rho_a(h_a)}, \tag{3}$$

where η is a parameter usually taking a value between 1.2 and 1.5. Alongside the expression for density as a weighted sum over the neighbouring particles, we have to solve this simultaneously. For this, the Newton-Raphson root-finding method has been used.

1.2 Running and Plotting a Simulation

Take the following steps to compile, run and visualise the program:

1. Ensure that gsl is installed, providing the `gsl/gsl_vector.h` and `gsl/gsl_blas.h` headers.

2. Compile the C program by running `make` in the directory containing the Makefile
3. Run the program with `./SPH -f <output file> [...]`, appending other options (use `./SPH -?` to print the usage page)
4. Plot the results with `python3 plotting/plot.py <output file>`

Some example usages of the program are

- `./SPH -i colliding_streams -w 15 -e isothermal -N 300 -f streams.tsv` to run the colliding streams test with a box of length 15, an isothermal equation of state and 300 particles.
- `./SPH -i shock_tube -w 1 -t 0.2 -e ideal -N 360 -f shock.tsv` to run the shock tube test for 0.2 time units with a box of length unity, an ideal gas equation of state and 360 particles.

Both of these examples will run with an adaptive timestep and variable smoothing lengths, as neither were specified manually.

2 Performance of the Density Estimation

Estimating the density at a particle's position involves summing over the particle's neighbours' masses and smoothing the sum with a kernel function. **Figure 1** shows such an estimation, for the initial conditions of the Sod shock tube problem with high and low densities in the left and right sides of the box, respectively. This density estimation used a variable smoothing length. In order to determine the per-particle smoothing length, a Newton-Raphson root-finding algorithm is used to simultaneously solve for the smoothing length and density.

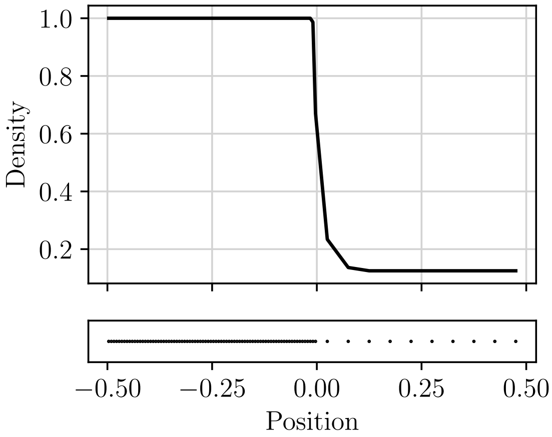

Figure 1: The density estimation (top) of the distribution of particles (bottom), using variable smoothing lengths. There are eight times more particles on the left hand side than on the right.

Figure 2 shows how the performance of this density estimation scales with the number of particles, for both fixed and variable smoothing lengths. In two or three dimensions, we would first need to find the neighbouring particles, which would increase the time complexity of the algorithm. In one dimension however, the particles never penetrate each other and so remain in the same order, allowing us to simply iterate outwards from the central particle until the kernel function falls to zero. Combined with the approach of using an approximately fixed number of neighbouring particles, this brings the time complexity of the algorithm down to the $O(N)$ seen in **Figure 2**.

Note that the case with variable smoothing lengths takes more time to run due to the root-finding routine needed to set the correct per-particle smoothing length.

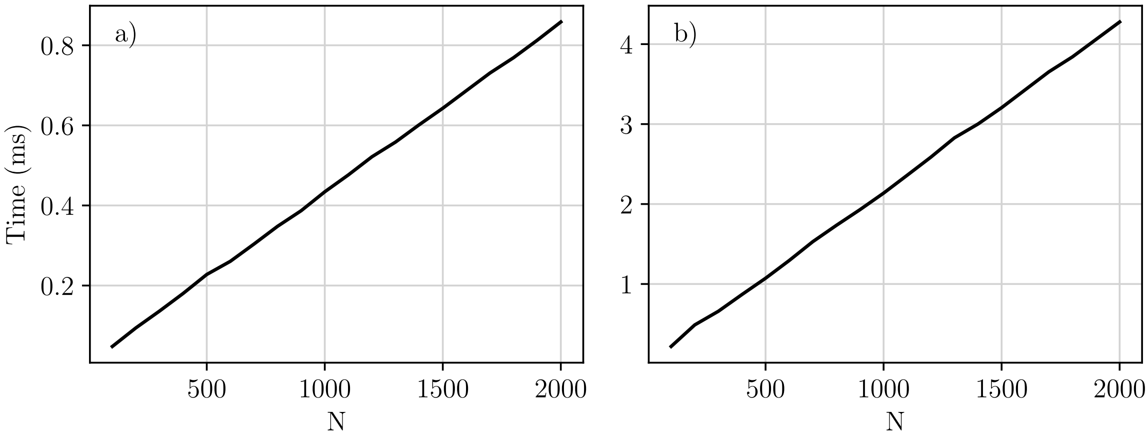

Figure 2: Time taken to estimate the density at each particle's position, as a function of the number of particles N for a) fixed smoothing lengths and b) variable smoothing lengths. The fixed smoothing length was taken to be 1.3 times the mean particle spacing.

3 Test Cases

3.1 Colliding Streams

The colliding streams problem has initial conditions of uniformly distributed particles across the box, and velocities distributed such that particles on the left and right sides of the box have velocities v_0 and $-v_0$, respectively [4]. With this, the particles collide with the opposing 'stream' when they reach the middle of the box, and a build-up of density is obtained in the centre of the box.

Figure 3 shows this problem at time $t = 1.0$ as simulated by the code, with 300 particles over a box width of 15 and an ideal equation of state with $P = (\gamma - 1)\rho u$ and $\gamma = 5/3$. The particles begin with $u = 1/(\gamma - 1)$ and $v_0 = c_s$, where $c_s = \sqrt{\gamma}$ is the adiabatic speed of sound in this case. The thick blue lines show the analytical solutions of the problem, and the code succeeds well in matching them. The system's total energy is conserved well, with no obvious deviations from its initial value as kinetic energy is converted to internal as the particles collide in the centre of the box. Total momentum is also well conserved (note the scale of the y axis). It can be seen that the particles in the higher density region are correctly given a smaller smoothing length, conserving the mass contained within the kernel function.

Figure 4 shows a comparison between simulations run with an increasing number of particles. It shows a convergence on the analytical solution with increasing N.

Figure 5 shows the same problem with an isothermal equation of state. In this case the dip in density at the centre of the box is not present, as the internal energy is constant so cannot form an overestimation at the initial collision of the streams. The total energy is not conserved for this simulation, as particles that are brought to a standstill in the central region do not transfer their energy to heat.

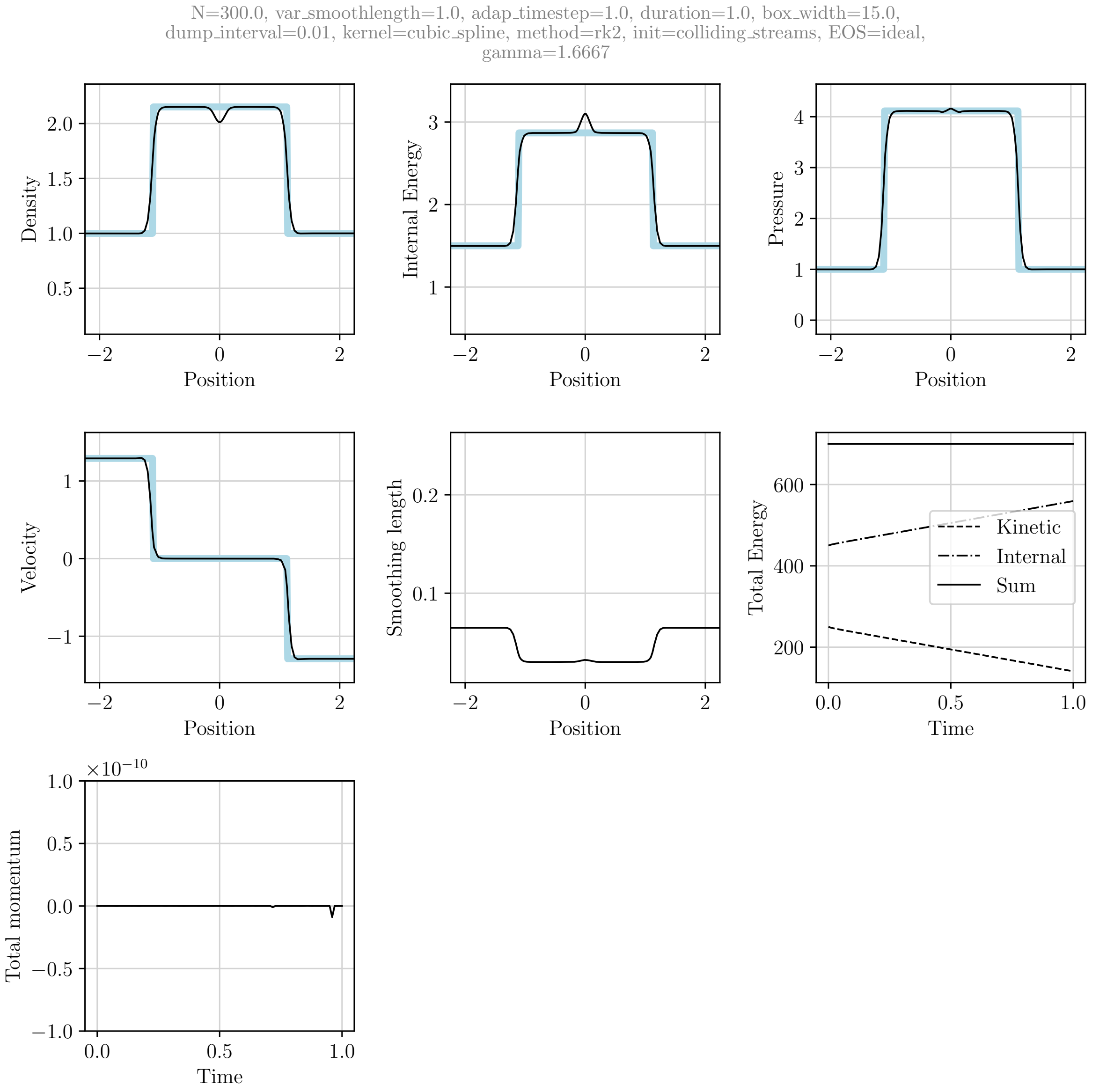

Figure 3: The colliding streams test at time $t = 1.0$ with an ideal gas equation of state and 300 particles. The analytical solution is shown with a thick blue line.

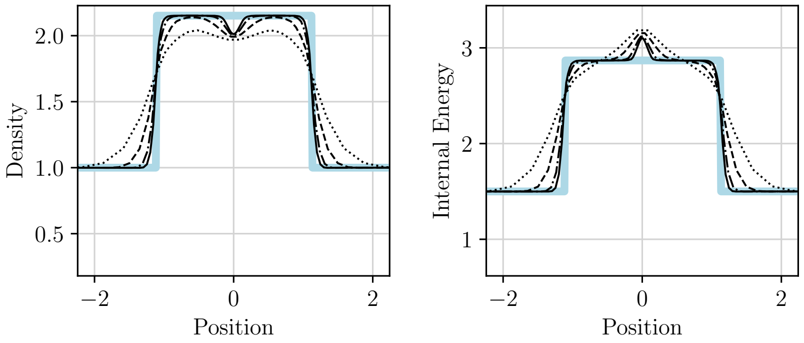

Figure 4: The colliding streams test run with 50, 100, 200 and 300 particles (the dotted, dashed, dash-dotted and solid lines, respectively). With the box width of 15, this corresponds with 3.3, 6.7, 13.3 and 20 particles per unit length. The analytical solution is plotted in a thick blue line.

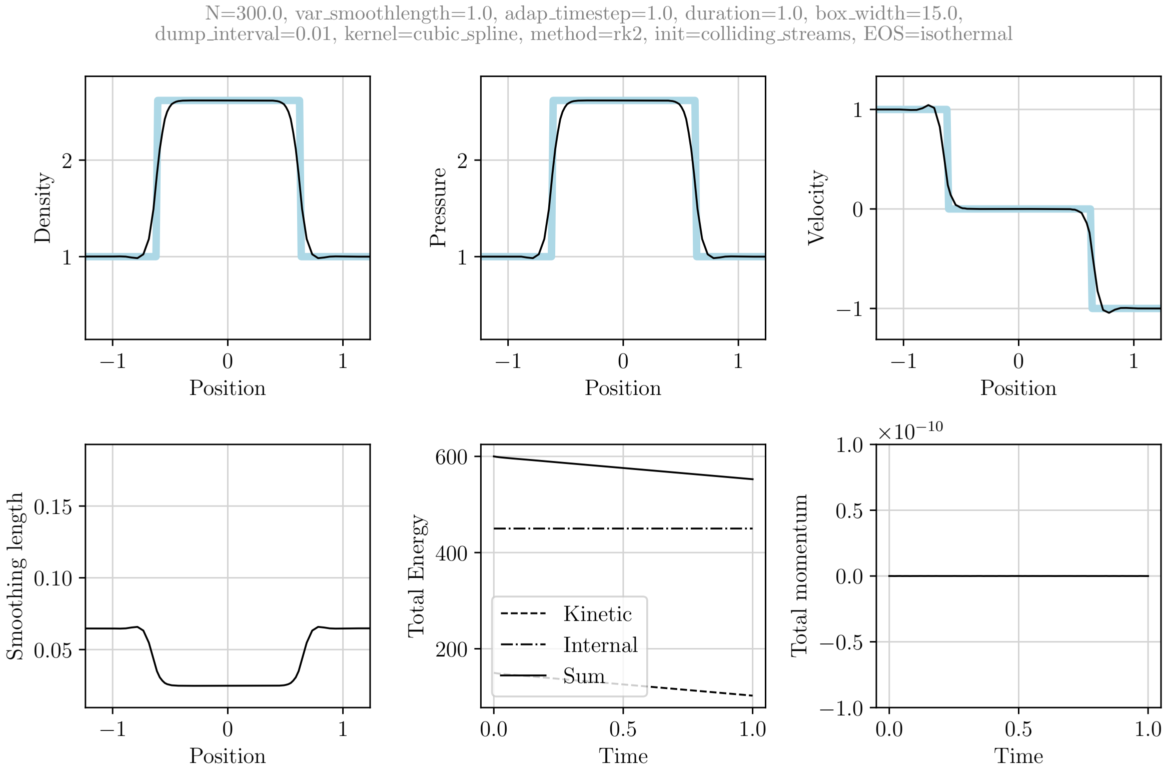

Figure 5: The colliding streams test, as in **Figure 3** but with an isothermal equation of state.

3.2 Sod Shock Tube

The Sod shock tube has initial conditions of high and low pressure on the left and right sides of the box, respectively, as in **Figure 1**. In this case, the high pressure region is at 8 times the pressure of the low pressure region. All the particles start with $v = 0$ and an ideal equation of state. **Figure 6** shows this problem as solved by the code. As with the colliding streams test, the analytical solution is matched well, and total energy is conserved.

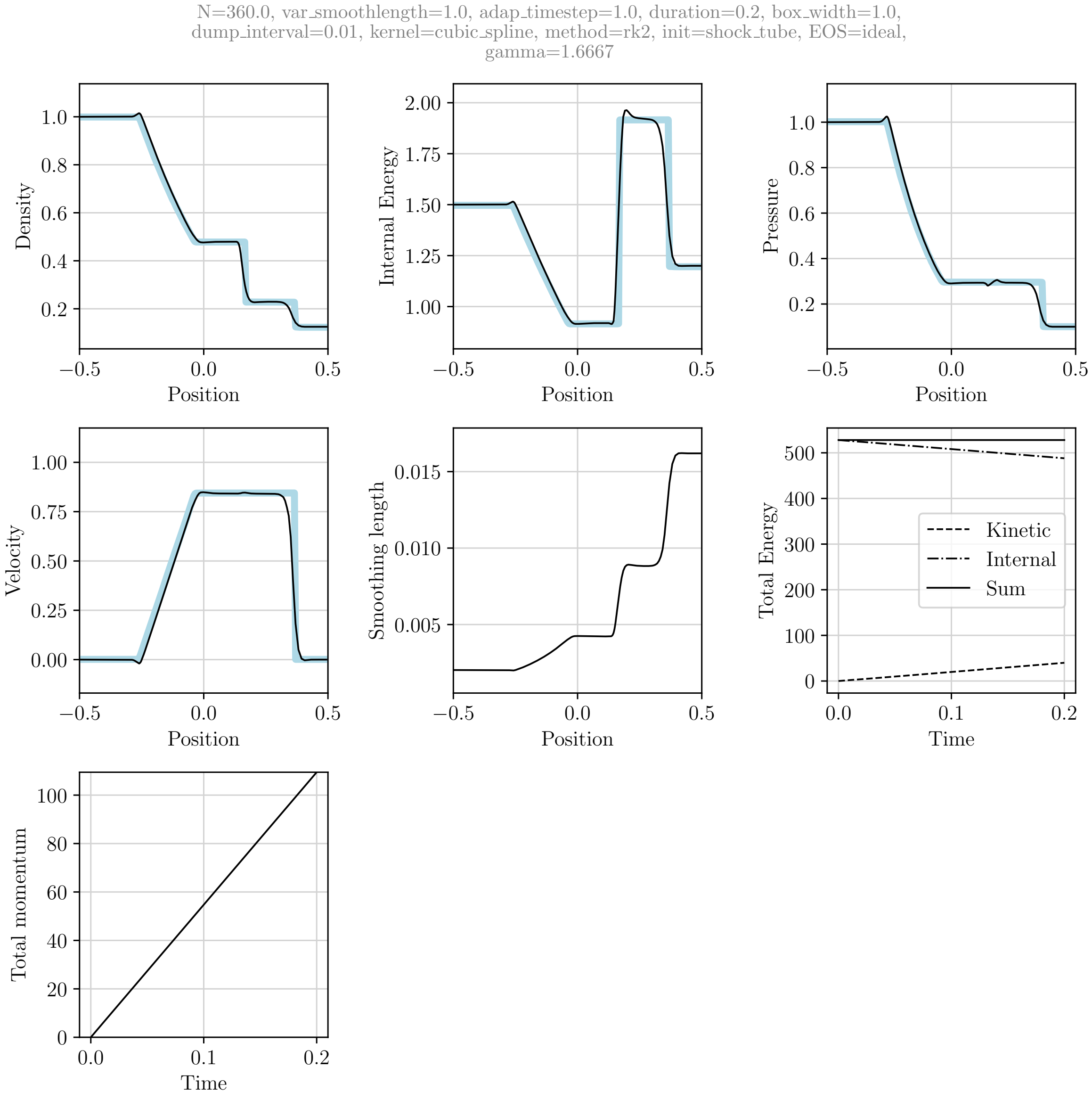

Figure 6: The Sod shock tube test at time $t = 0.2$ against the analytical solution (thick blue line).

4 Effect of the Viscosity Parameters

In the code, viscosity is modelled with two free parameters, α and β, which are coefficients for viscous terms that are linear and non-linear in velocity, respectively. Preceding this section, simulations have only been run with the typical $\alpha = 1$ and $\beta = 2$, but now the effects of these values will be investigated.

Figure 7 shows the effect of increasing α. Larger values result in a 'smoother' distribution, even though the number of particles is constant between the tests. This is likely due to the fact that the local interactions of viscosity are stronger and can act over a larger scale, thereby smoothing sharp edges of the system. The overestimate of internal energy at the centre of the box becomes more pronounced with a higher α. It should also be noted that with higher viscosity parameters the simulation can be run with slightly larger timesteps as velocities are generally lower due to the larger resistive forces.

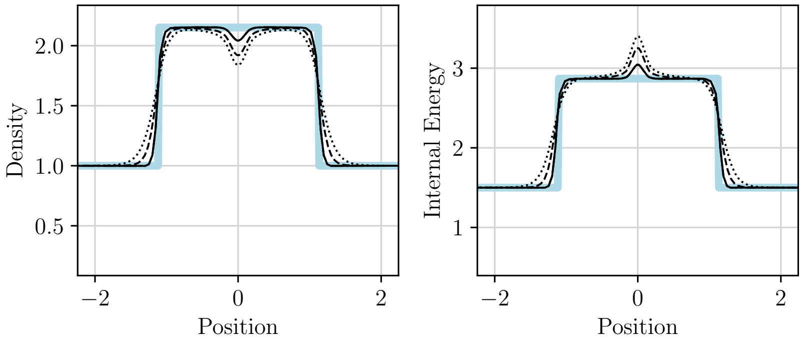

Figure 7: The ideal gas colliding streams test run with $\alpha = 1$, 3 and 5 (the solid, dashed and dotted lines, respectively) and $\beta = 0$.

Figure 8 shows the effect of increasing β. As β is the coefficient of a term non-linear in velocity, changing it has less of an effect than doing the same with α, however increasing it does seem to have a similar type of effect.

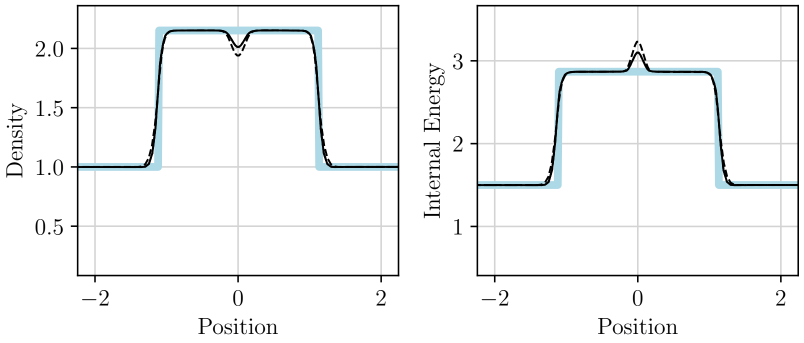

Figure 8: The ideal gas colliding streams test run with $\alpha = 1$ and $\beta = 2$ and $\beta = 10$ (the solid and dashed lines, respectively).

Decreasing α below 1 leads to small density oscillations as the streams collide, as can be seen in **Figure 9**. Similar oscillations can be seen in the shock tube problem, particularly in regions of sudden changes in internal energy.

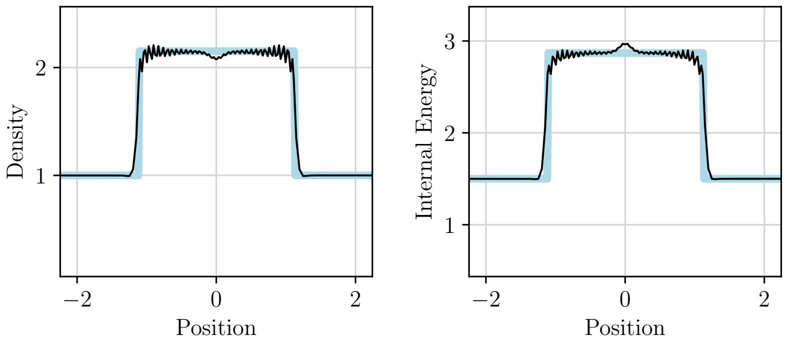

Figure 9: The ideal gas colliding streams test run with $\alpha = 0.2$ and $\beta = 2$ Small values of alpha lead to these density oscillations.

References

[1] D. J. Price, "Smoothed particle hydrodynamics and magnetohydrodynamics," *Journal of Computational Physics*, vol. 231, pp. 759–794, Feb. 2012.

[2] J. J. Monaghan, "Smoothed particle hydrodynamics.," *Annual Rev. Astron. Astrophys.*, vol. 30, pp. 543–574, Jan. 1992.

[3] S. Rosswog, "Astrophysical smooth particle hydrodynamics," *New Astronomy Reviews*, vol. 53, pp. 78–104, Apr. 2009.

[4] M. R. Bate, "The role of accretion in binary star formation."