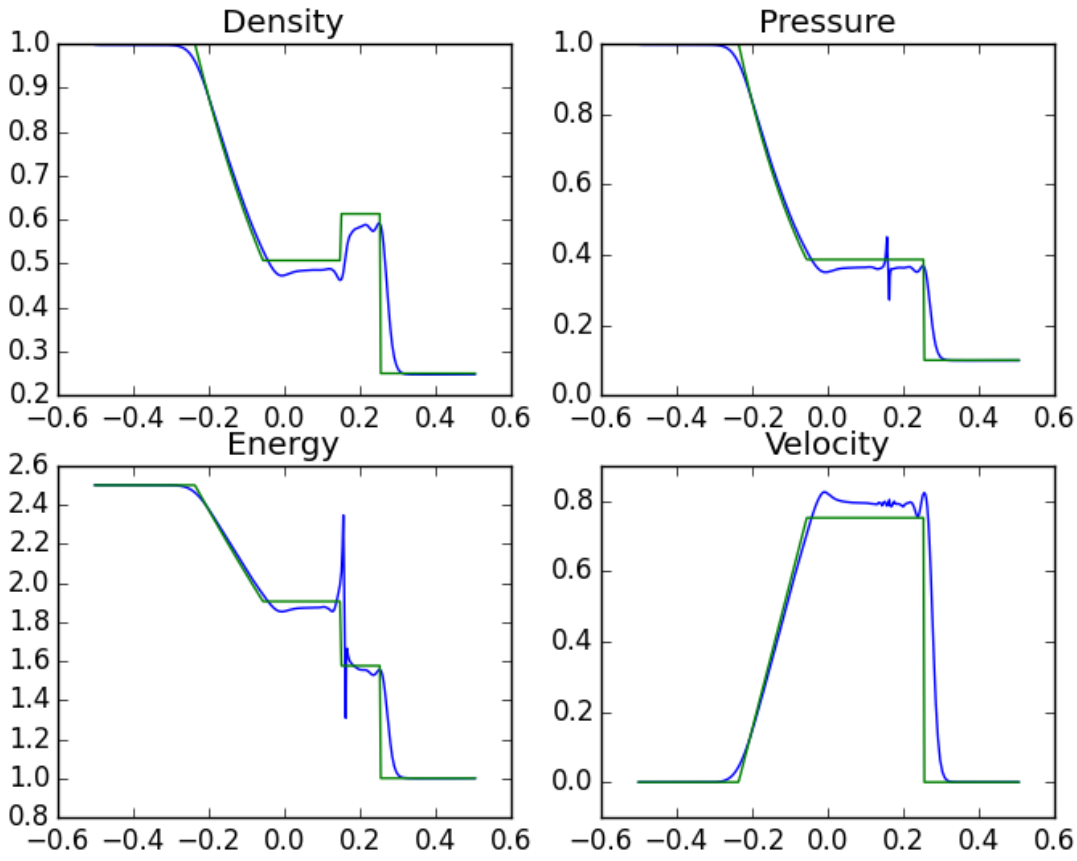
Sod's shock tube using SPH (with binning)

Parameters:

- Number of points in the physical domain = varying
- Physical Domain = [-0.5, 0.5]
- Extension using ghost points = upto 0.6 on each side
- time step = 1e-4
- Final time = 0.001 seconds
- h_factor = 2

Results:

- h_factor = 2; time step = 1e-4 sec (result with binning)



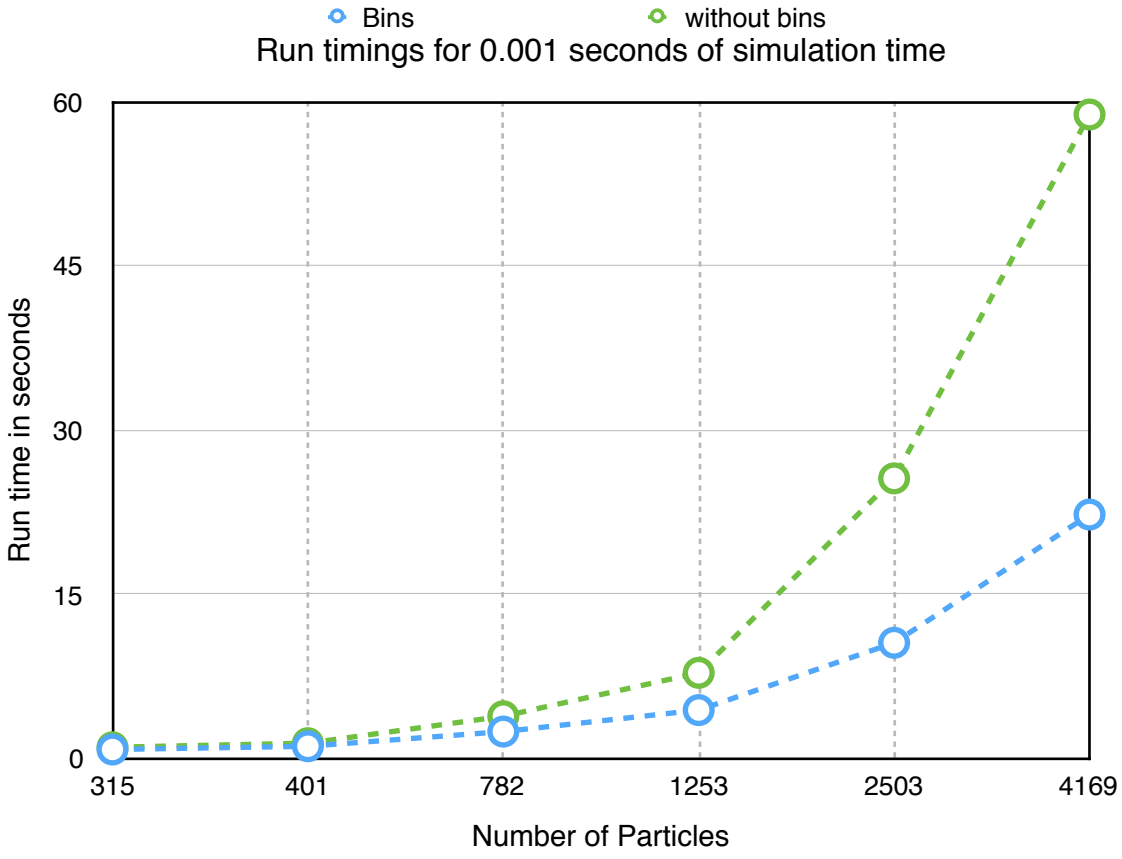
Timing Performance:

I ran the code for 0.001 seconds with a time step of 1e-4 to see the time each version (with bins (NNPS) & without bins) takes:

Here is the performance improvement table & plot:

Performance

Particles	runtime with bins	runtime without bins	Speed up
315	0.864	1.07	1.238
401	1.16	1.44	1.241
782	2.5	3.9	1.560
1253	4.46	7.86	1.762
2503	10.6	25.6	2.415
4169	22.3	58.8	2.637



We can clearly see a performance improvement in the run time when we use binning (NNPS). The improvement margin is smaller for lesser particles because I have already used numpy operations (which are implemented in C) in both codes.

Testing of whether binning is working or not:

I wrote a test function to check whether my implementation of binning is correct or not. For this I generated 30 uniform random numbers in [0,1) and binned them with a bin width of 0.2. I have plotted the results below with different colours for different bins. I have also found the nearest neighbours for particle numbered 10. (different y locations are used just for segregation purpose and better visualisation)