

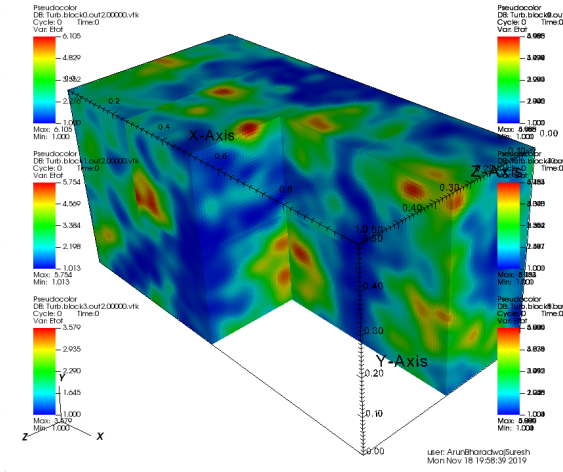
For my assignment, I chose to run the hturb example that was provided in the turb example library for the class. This example is concerned with simulating turbulence in a box. The simulation was run twice using two different Riemann solvers. The Riemann solvers used were the ones suggested in the homework assignment. For the first run, the simulation used the ROE solver, which uses the linearized version of the jacobian, which is then solved exactly at the point of discontinuity. The second run made use of the HLLC solver which makes use of Godunov-type fluxes and related integral equations to solve.

The simulation consisted of a box fragmented into 64 smaller boxes where problem was solved - the riemann solver was then used to glue together these smaller fragments to reconstruct the entire box. The athinput.turb file was edited to have  $tlim = 0.7$ . This means that the solution was solved for 8 time steps (0s to 0.7s with  $dt=0.1$ ). This was done because, each smaller fragment produced 8 files, and so for a single time step, there were  $64 \times 8 = 512$  output files, and if I wanted the simulation to last for 10 seconds there would be 51200 files. Setting aside issue about filling up space, I also noticed that the *join\_all\_vtk.sh* and *join\_vtk + .c* visualization scripts (that produces a single metadata file for VisIt) that was included with Athena did not seem to run because of a few permission issues - and so, 51200 files would be a little too much for me to parse manually. So  $tlim$  was set to a relatively low value.

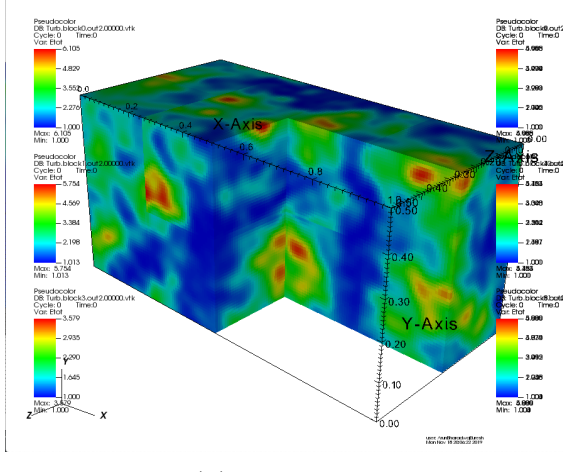
The output obtained was then transferred to my local computer where I once again tried to use the join scripts and failed because of permission issues in my computer (Because of some Xcode issues in my local computer). So I ended up manually visualizing the smaller fragments using VisIt by creating a correlation plot where I overlayed different fragments to visualize them side by side. Since it was a tedious job to merge all  $64 \times 8$  files on visnek manually ( $\times 2$  for the two solvers), I selected the first 12 fragments (which made up the bottom left  $1/5$ th of the box) for this assignment as a proof of concept. I would like to follow this up with an additional assignment submission with the full result after I figure out how to use the *c* scripts included in the visualization folder.

For the visualization, the total energy of the fluid across the box and its time evolution were plotted. This choice was made because analyzing turbulence would yield meaningful results through observing the energy profile. All that being said, presented below are the results that were observed.

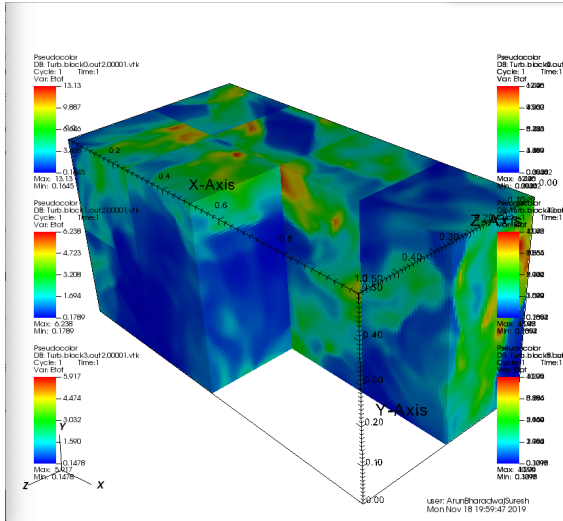
# Total Energy:



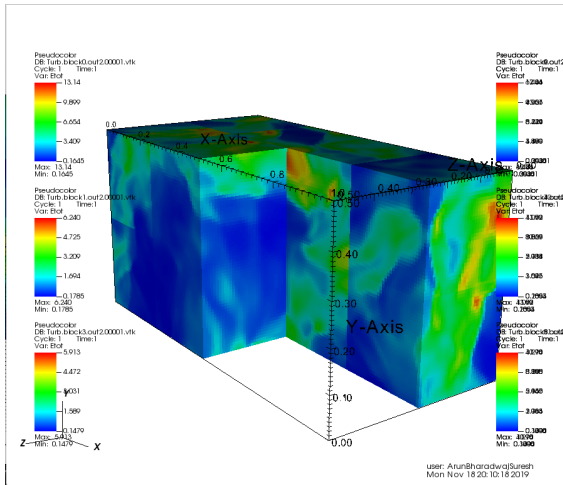
(a) Roe:  $t = 0$



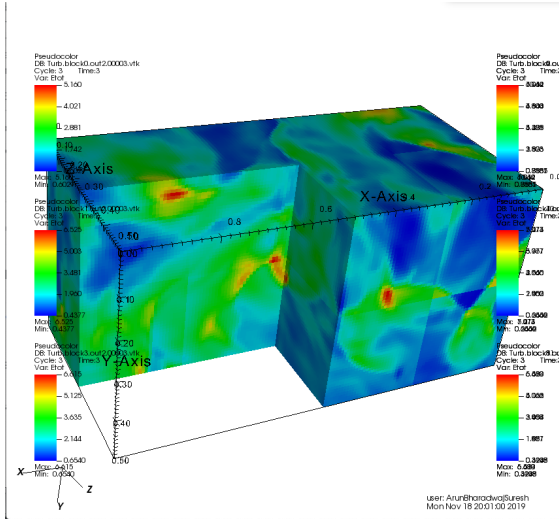
(b) Hllc:  $t = 0$



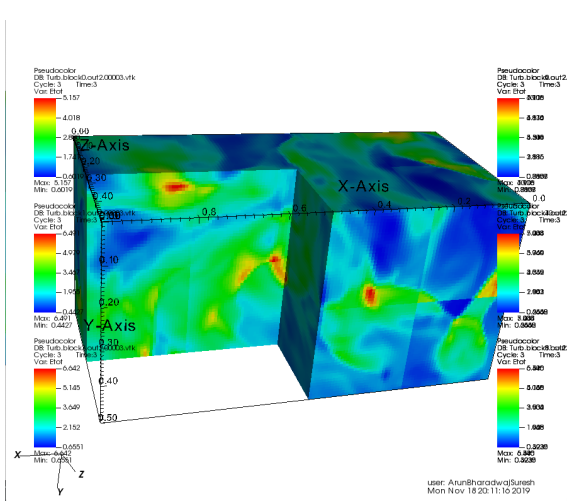
(a) Roe:  $t = 0.1$



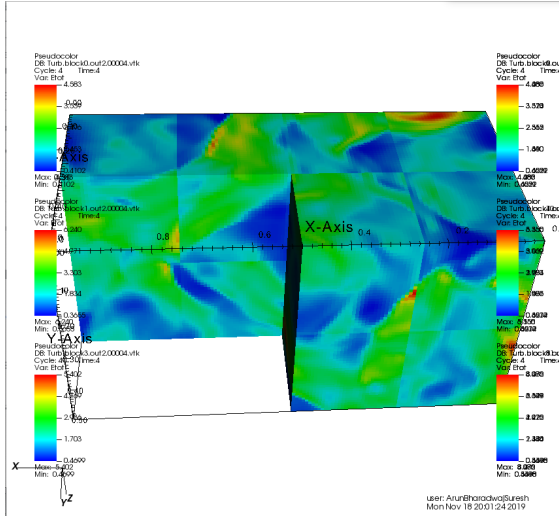
(b) Hllc:  $t = 0.1$



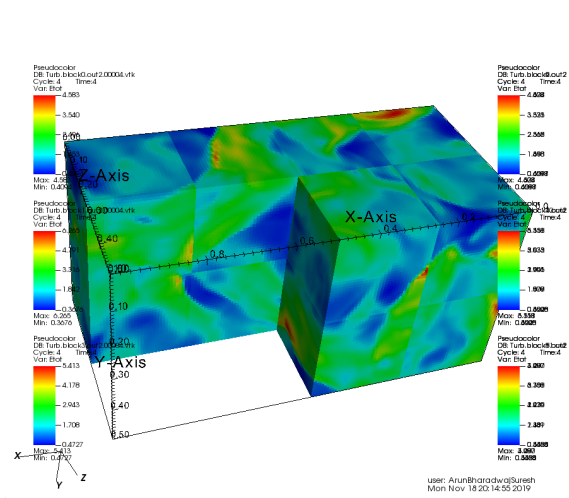
(a) Roe:  $t = 0.3$



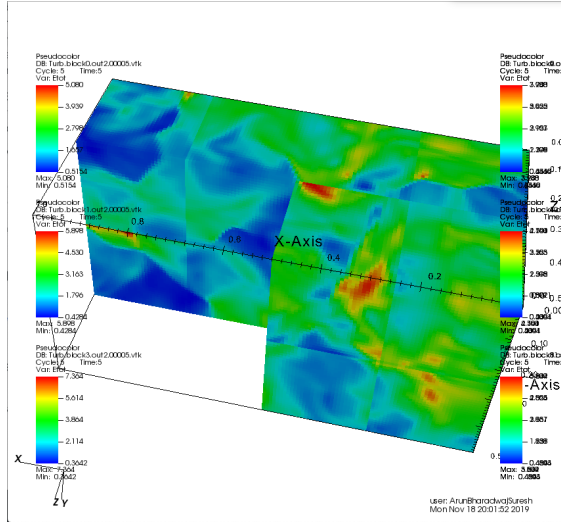
(b) Hllc:  $t = 0.3$



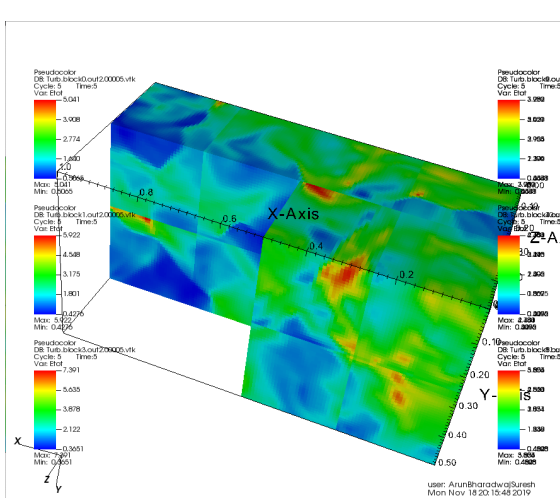
(a) Roe:  $t = 0.4$



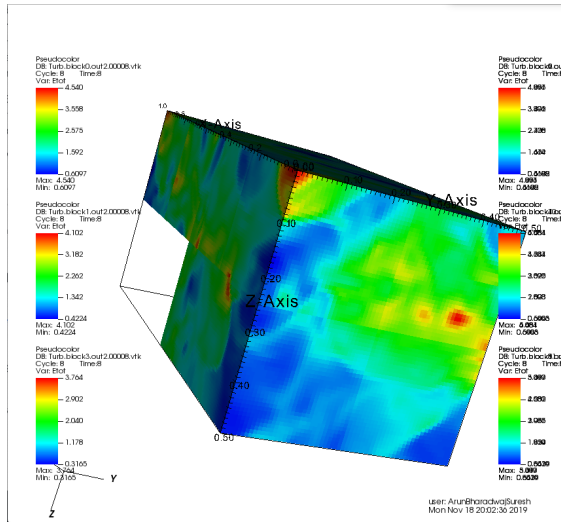
(b) Hllc:  $t = 0.4$



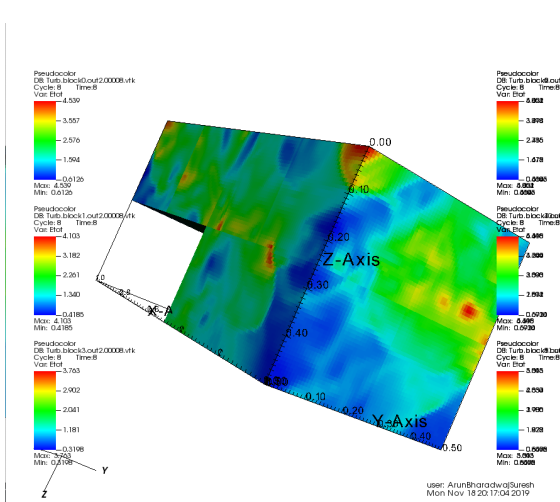
(a) Roe:  $t = 0.5$



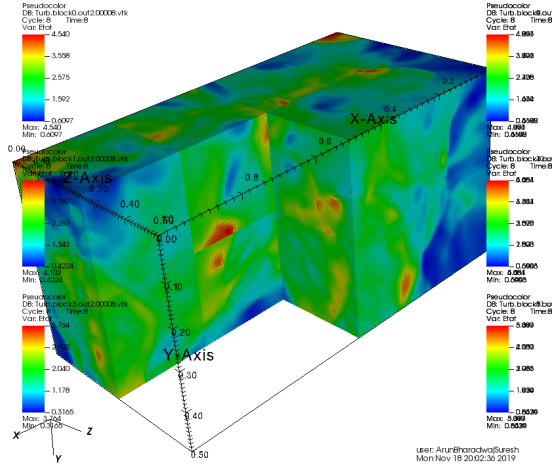
(b) Hllc:  $t = 0.5$



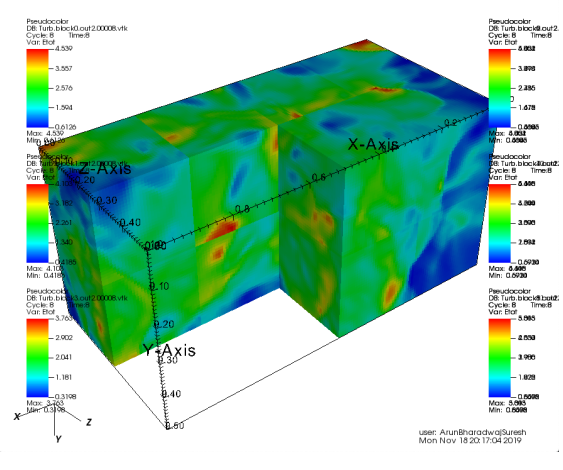
(a) Roe:  $t = 0.7$  view: back



(b) Hllc:  $t = 0.7$  view: back



(a) Roe:  $t = 0.7$  view: front



(b) Hllc:  $t = 0.7$  view: front

It is apparent that both Roe and HLLC solvers very produce similar results. Upon very close inspection, we notice that the HLLC does a slightly better job at gluing the fragments than Roe. This can be seen a bit easily in the front view of the box, on the rear end and the lower side of the L cut, when  $t = 0.7$  - where the lines that join the boxes are smoothed out much better than the Roe solver. But other than very specific details like that - both outputs seem to coincide very well. The only realistic way I was able to say that these results were from two different runs by eyeballing - were because of the different values at the color index during the different time frames.

This close similarity between the two solvers, may also be observed because the visualization did not make use of the entire box, but instead just a small part of it. Perhaps in areas with much more serious turbulence, the differences could be much more significant. I hope to answer this question in a follow up assignment submission where I analyze the entire box (after I figure out how to run the C script).