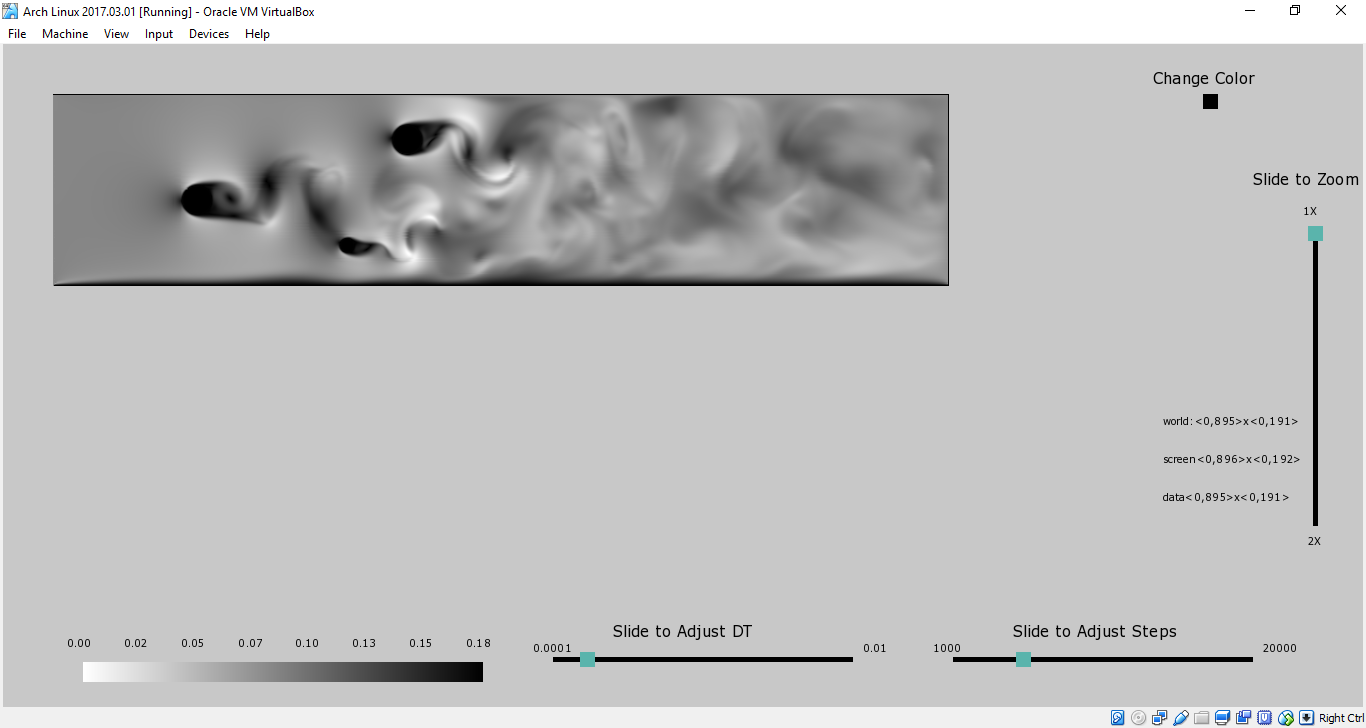
**A07P01 Report**

**To change datasets, you will have to modify line of the setup function in ofApp.cpp.**

To start, I created a class to represent the three coordinate systems. I got the vectors at each point in the world coordinate systems. Then I got the screen vectors by using interpolation and stored them in data coordinate system. Magnitude was taken from vectors stored in the data coordinates and which are actually correspond to world data, are mapped to screen coordinate system and drawn.

Apart from showing the image in grayscale, I used a simple color map with a bright blue color as the high values and a dull yellow color to represent low values. Though using this color map middle range values cannot be well identified, but it creates a good contrast to differentiate between the extreme values. I have legends to show each color corresponding to each magnitude to the vectors. I have a button to toggle between color coordinates image and grayscale image.

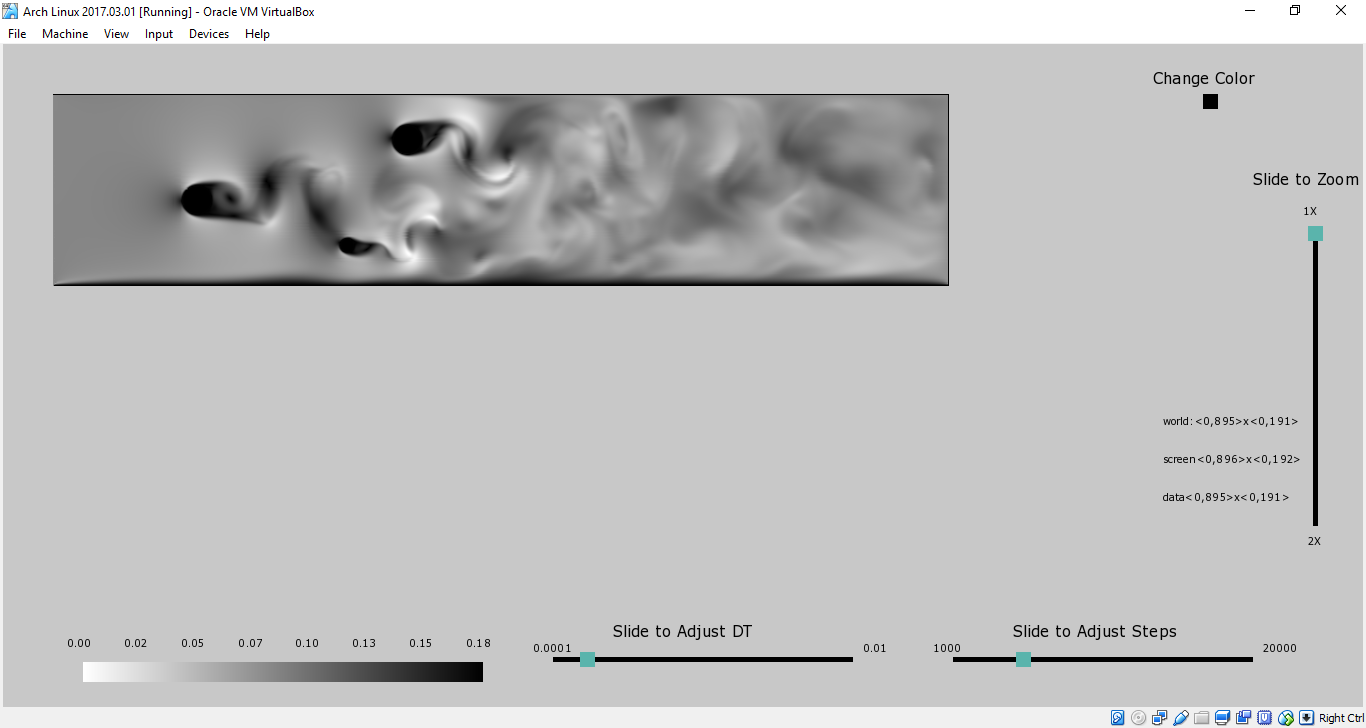
I show the bounds of the three coordinate systems to give the user better context of the image in (x,y) format. I have also enabled zoom and pan functionality. To zoom, the user should use the slider which goes from 1X which the original size of the image to 2X which is twice the size. The same zooming slider is used to change both length and width. Both dimensions increase by the fraction chosen on the slider. Having two different sliders for length and width would just make the tool more crowded.

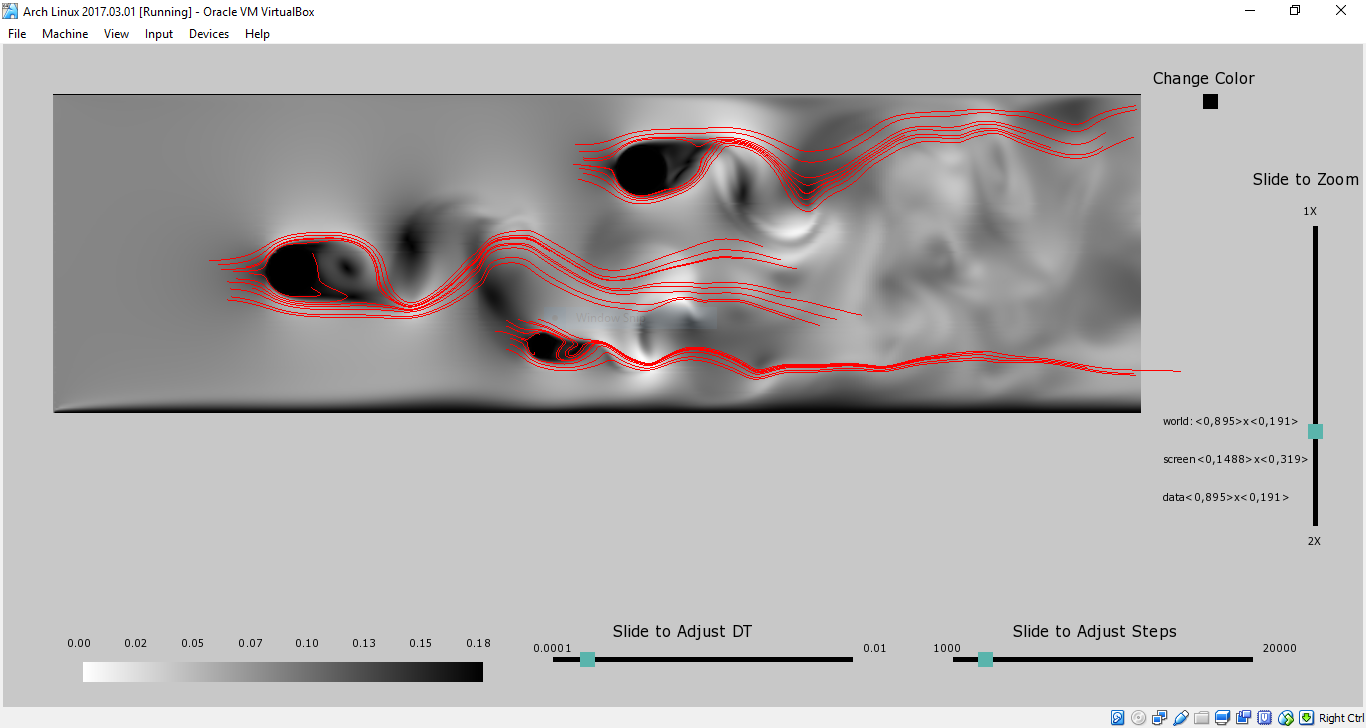
For panning within the image, the user can use the left arrow key to move left, the right arrow key to move right, the up arrow key to move upwards and down arrow key to move down. This was inspired from how we usually scroll in windows laptops. This seemed as the most convenient way for the user to pan across the entire image.

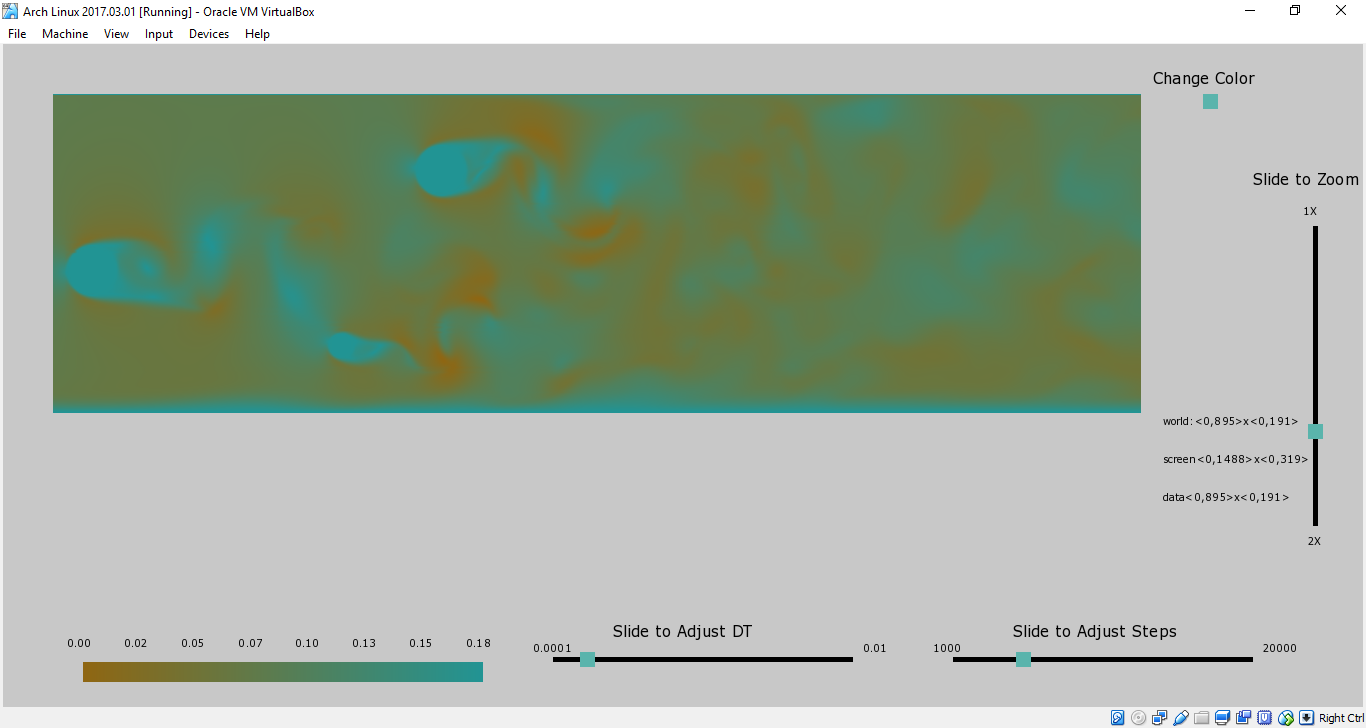
The streamlines where constructed in a bright red color so that they are clearly visible over both grayscaled image and the color coordinates image. I used RK-4 integration to load the streamlines. The value for time differential and number of steps was left for the user to adjust using two sliders for each at the both. While, zooming and panning I am storing the streamlines in world coordinates which was suggested on piazza, I only remap them to screen coordinates while zooming and panning. **To remove the streamlines form the image and start over, press ‘R’ key.**

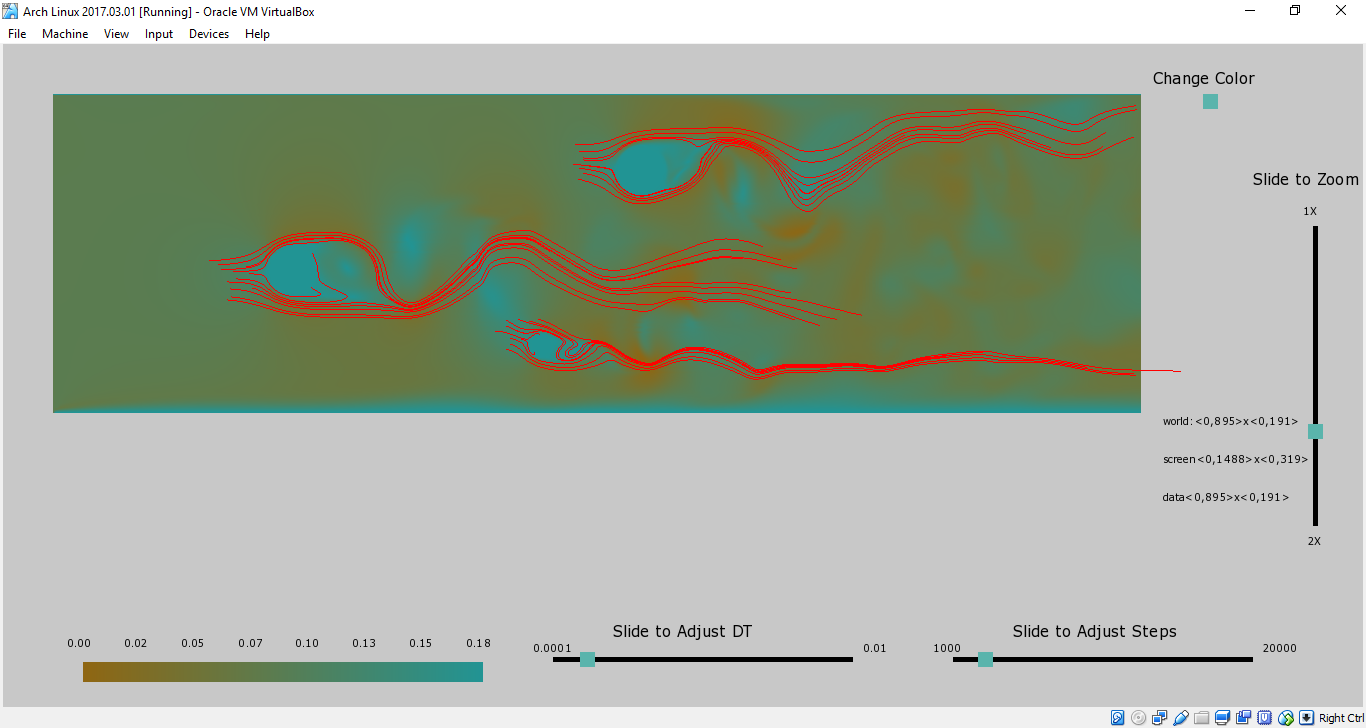
I decided to inspect cosine.vti and 3cylflow.vti.

**Cosine.vti –**

I see two artefacts which look like fingerprints. The flow looks circular concentrated towards the centre or all the lines flowing form centre. I also see other asymptotic lines in addition to above. I have included both screenshots below, with grayscale image and color coordinated.







**3cylflow.vti-**

Here, I see three objects in motion, towards left direction. I determined the direction of motion form the flow of the streamlines. Streamlines look like the wind or fluid motion around a moving object. I observed other waves farther from the three objects. I have included screenshots for both grayscale image and color coordinated image.

