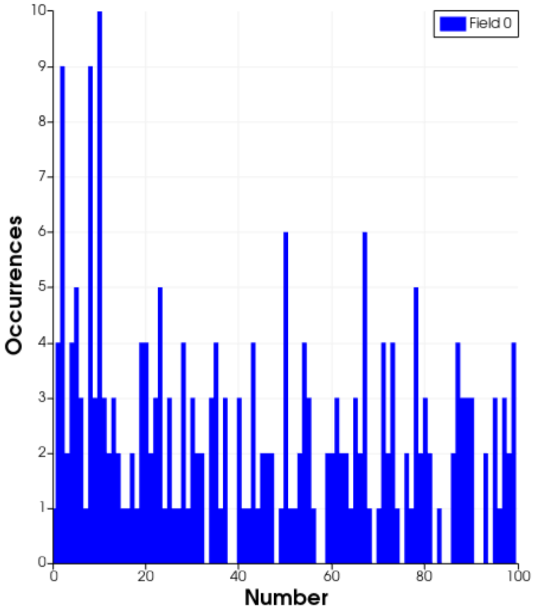
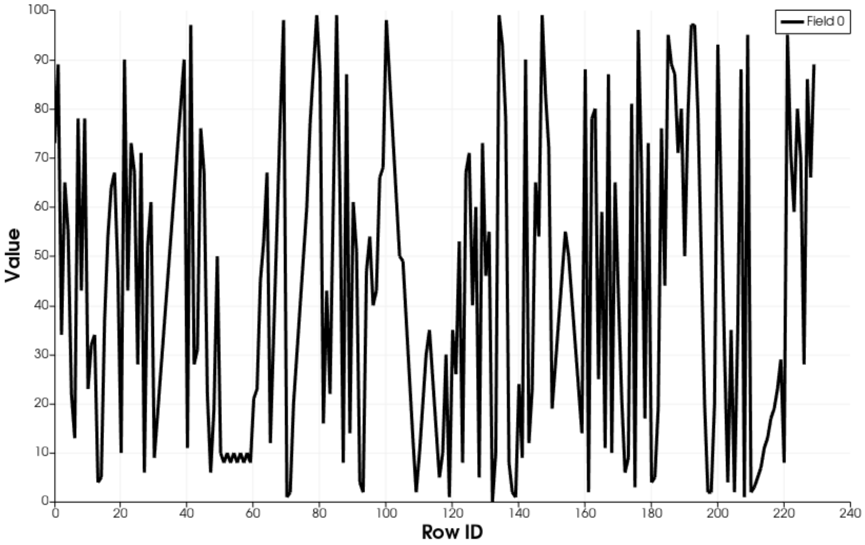
Jack Wilburn
Assignment 2

Part 1
Q1

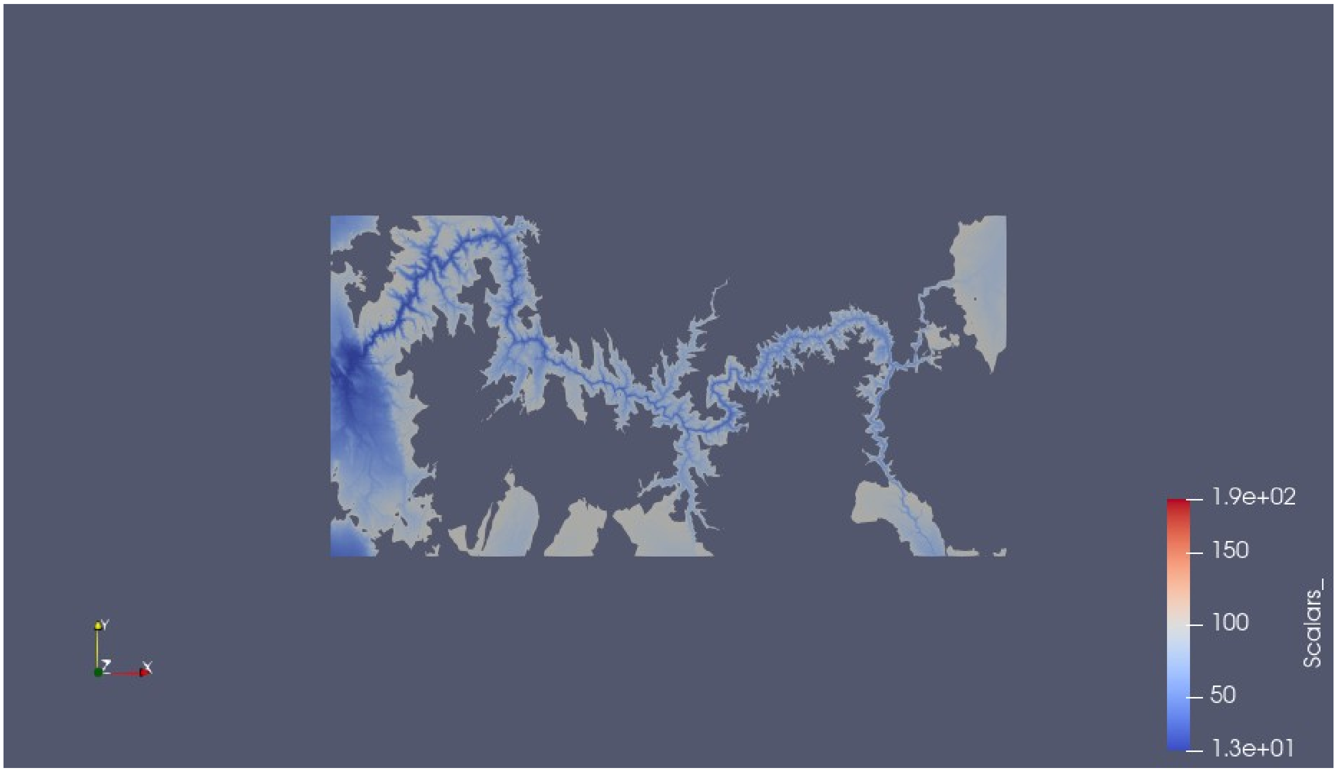


1.
    1. The number 10 occurred the most times in this dataset. It occurred 10 times.
    2. The numbers that didn't occur were 33, 38, 39, 57, 58, 69, 75, 82, 84, 85, 91, 92, and 94. So 13 numbers were unused.

2.

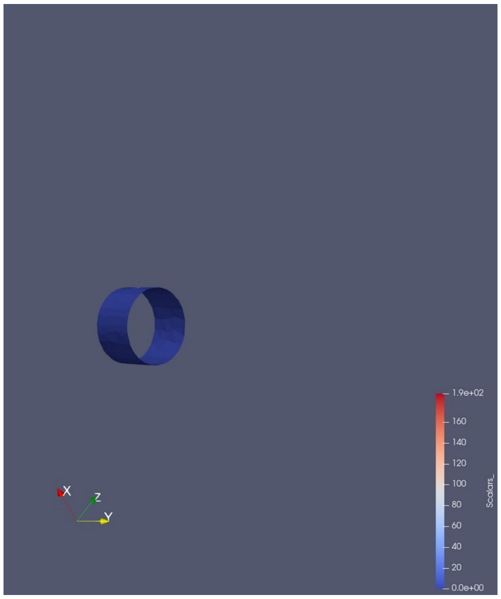


Q2

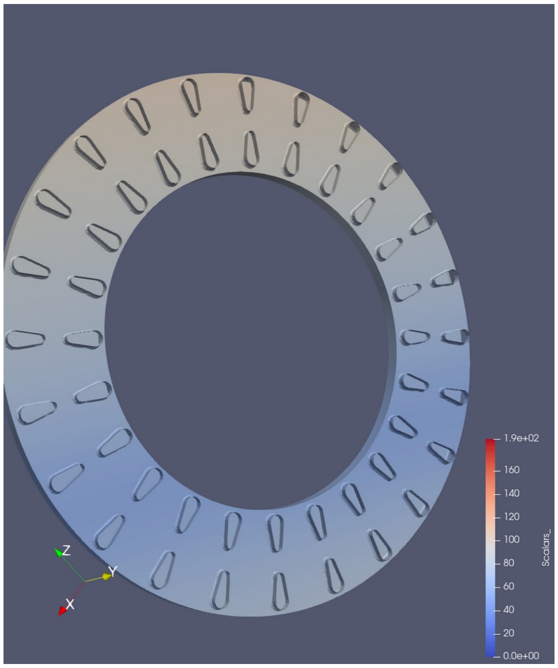


1. I used the threshold value of 98 (it was the first value over 200,000 occurrences as mentioned in the question). This gave a decent view of the river bed elevations and a little bit of the surrounding beaches. In this we miss the rim and the details of the upper canyons, but we get the full detail of the lake over to the left side of the image and the river that connects to it. It also appear to be missing large rocks/towers next to the river (they are represented as holes in the mesh.
2. There are 2915826 points in the threshold-ed image.

Q3



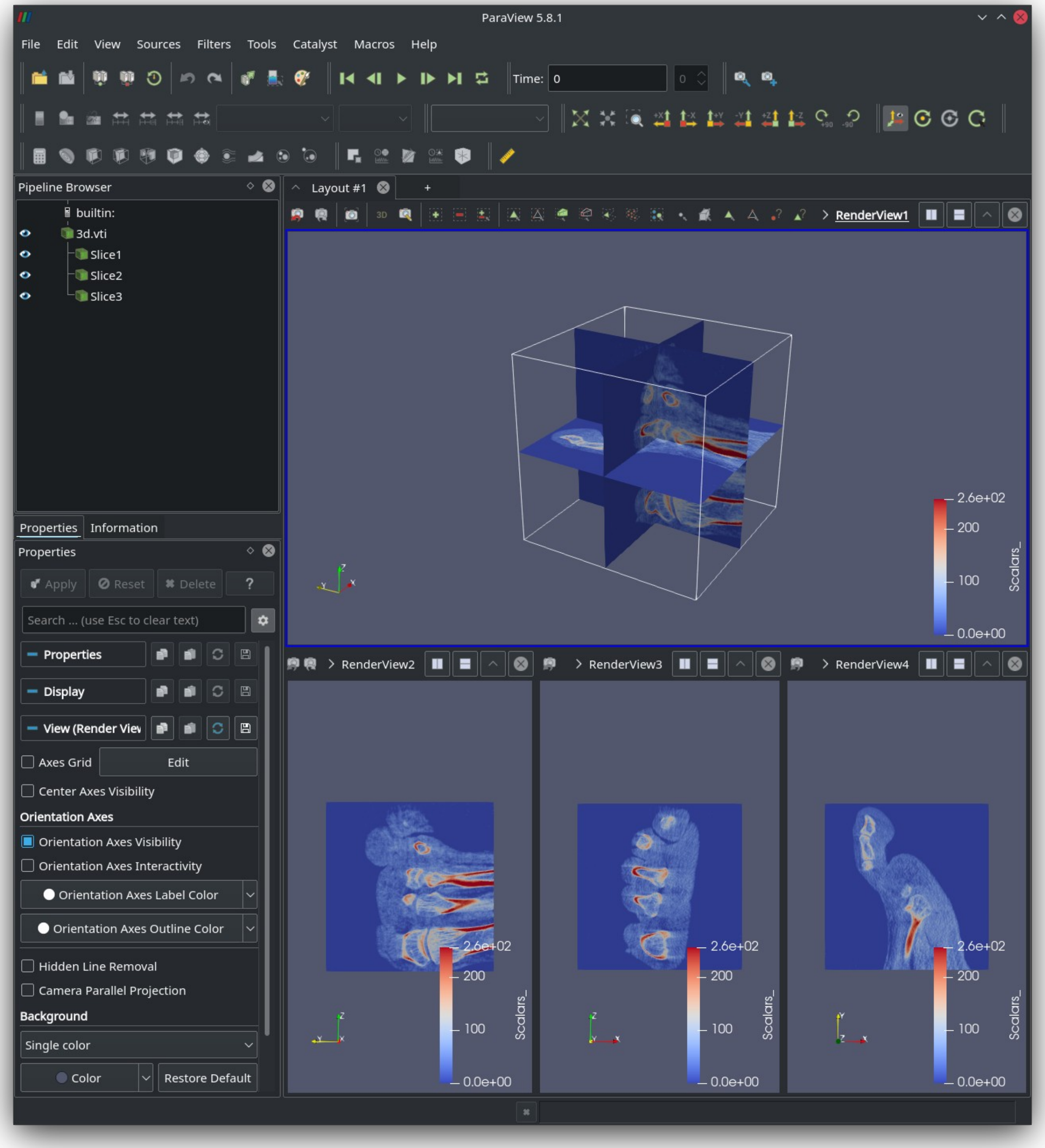
1. The best value that captured the cylinder were 0 for the minimum and 43 for the maximum.



2. There are 24 ventilation slots on this brake rotor.

Q4

1.



☐ Axes Grid

☐ Center Axes Visibility

☑ Orientation Axes Visibility

☐ Orientation Axes Interactivity

○ Orientation Axes Label Color

○ Orientation Axes Outline Color

☐ Hidden Line Removal

☐ Camera Parallel Projection

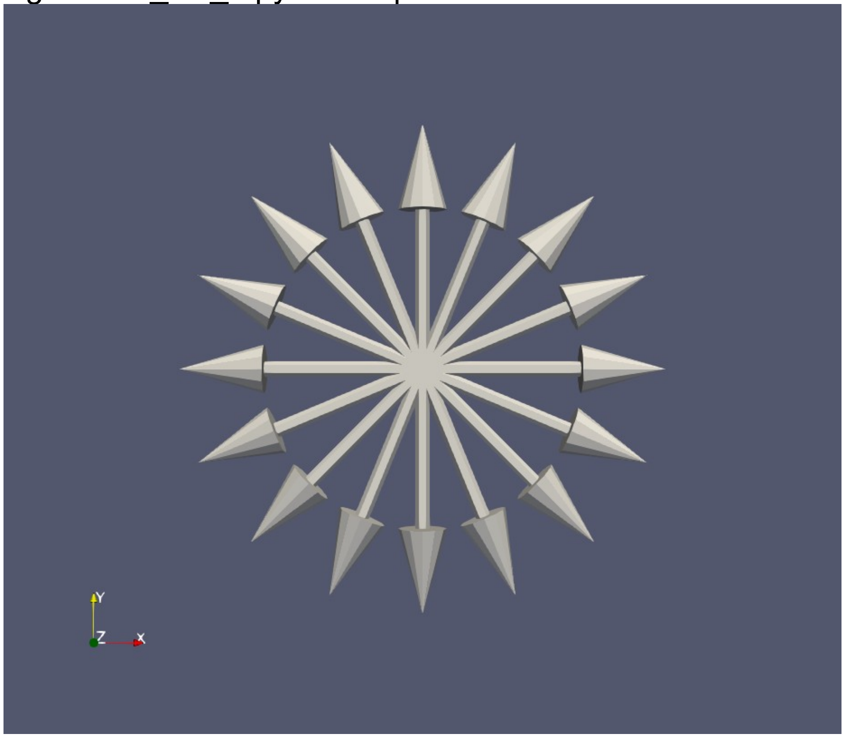
Single color

● Color

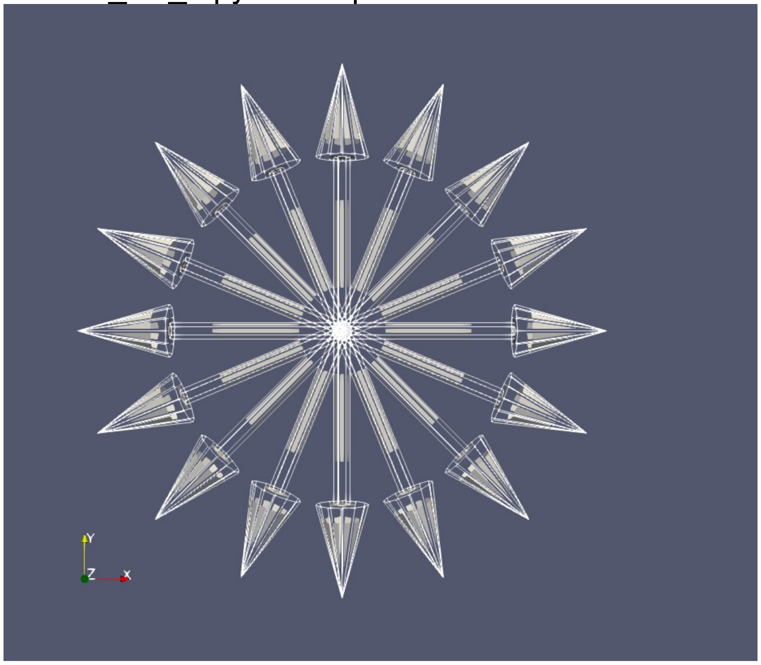
Part 2

Q1

1. See file assignment2_Q1_1.py for script

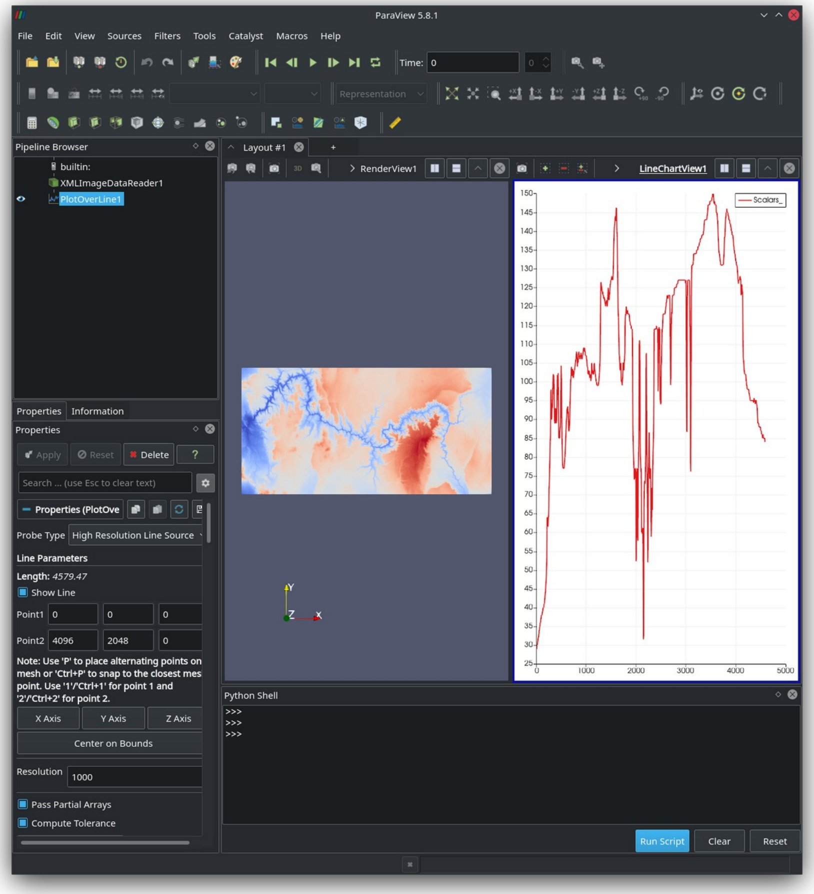


2. See file assignment2_Q1_2.py for script



Q2

1. See file assignment2_Q2_1.py for script



Probe Type High Resolution Line Source

Line Parameters

Length: *4579.47*

☑ Show Line

Point1 0 0 0

Point2 4096 2048 0

Note: Use 'P' to place alternating points on mesh or 'Ctrl+P' to snap to the closest mes point. Use '1'/'Ctrl+1' for point 1 and '2'/'Ctrl+2' for point 2.

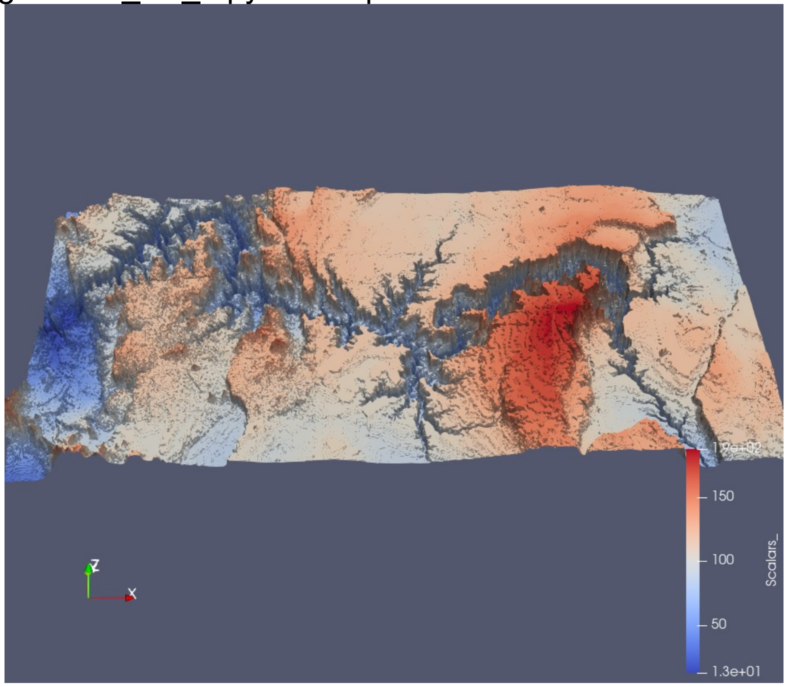
Resolution 1000

☑ Pass Partial Arrays

☑ Compute Tolerance

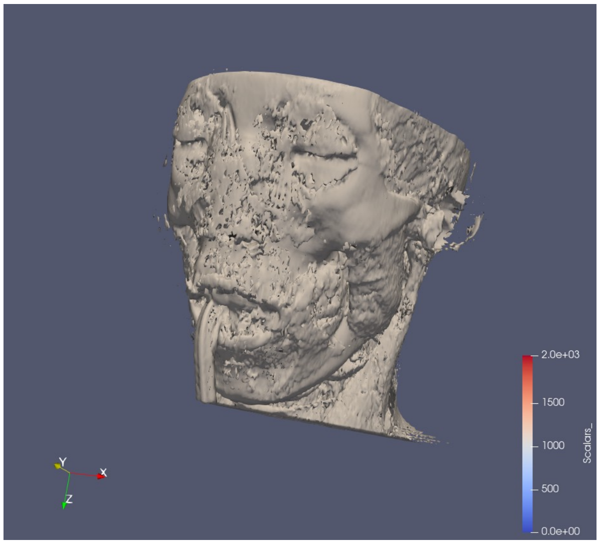
```
>>>
>>>
>>>
```

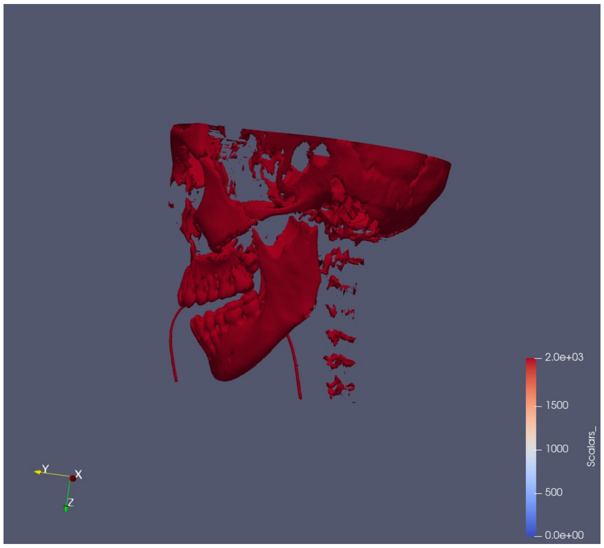
2. See file assignment2_Q2_2.py for script

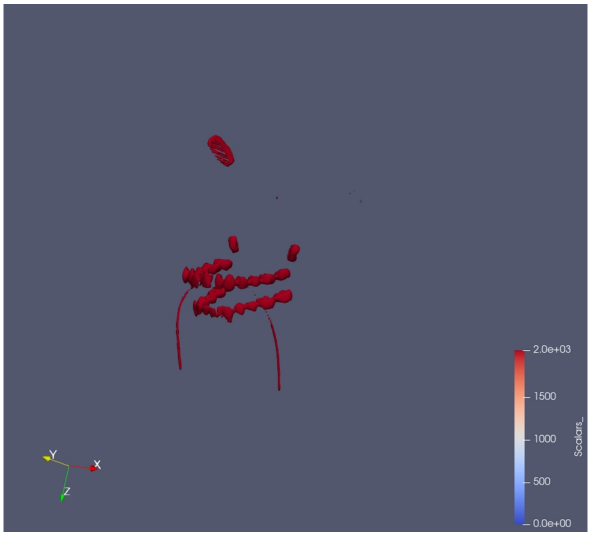


Part 3

Q1







1. The approximate value when the skin transitions to skull is 1075. The approximate value where the spine disappears is 2275. The iso-value of the for the teeth is 2900.

Q2

1. Time varying data is, simply, data that vary over time. Some examples might include weather, data data from collision simulations, personal weight over time from a smart connected scale.

   When extracting isosurfaces from time varying data we face several problems. We often run into performance issues with the search for isosurface values and the storage of the extra indices. These problems are amplified when we the data is coarse in the time dimension and we have to interpolate values between frames. Ultimately, these issues affect the smoothness of the animation transitions and the interactivity, leading to a poorer user experience.

2. The need for temporal hierarchical index tree data structures arises from the desire to minimize storage overhead of interacting with the visualization, while keeping performance high. The temporal hierarchical index tree does this by “trading the accuracy of the time-varying isosurface searches.” (p. 58)

3. The isosurface extraction algorithm first recursively traverses the temporal hierarchical index tree at a given time-step, searching for values that may lie on the isosurface. It does this until the leaf nodes are reached.

   After the candidate cells are identified, we order the cells by their minimum values and perform a lookup for each cell to see if their minimum value is greater than the iso-value. If it is, we stop the sequential search, and we’re done.

   Now repeat the above steps checking for the maximum cell value being larger than the maximum is smaller than the iso-value.

   That leaves you with the cells that are possibly on the isosurface and now you can draw the surface using their actual values at the time-step, removing the cells that are not on the surface.