Homework 3

Hongsup Oh

March 2020

1 Problem1

The readfunction provided by TA is used for read dataset. And save it as raw file named Problem1 result raw. The ikt view function is used for visualization. Figure1 is result.

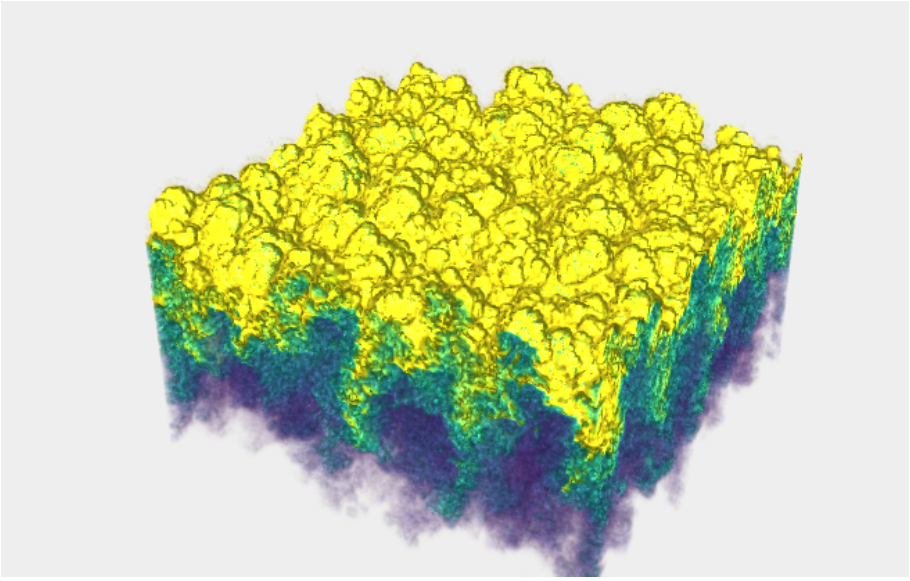


Figure 1: Result at resolution 24 and timestep 15

2 Problem2

The helpfunc is designed for interactive visualization for both resolution and timestep. For detail result, please check Problem2 of code.

3 Problem3

The PercentMixedLayerForVolume class is designed. The class including two functions such as init and PercentM. The init function defines current resolution. And the PercentM function calculate the percentage of mixed layer. If individual array's components are only 0 (zero) or 255, current array is assumed as non mixed array. Otherwise it is assumed as mixed array. For each slice of Z axis, count the number of mixed array and calculate percentage of mixed layer thickness. That is, the percentage of mixed layer thickness is the number of mixed array divided by total length of current slice. Finally average of percentage of mixed layer thickness for whole domain is calculated.

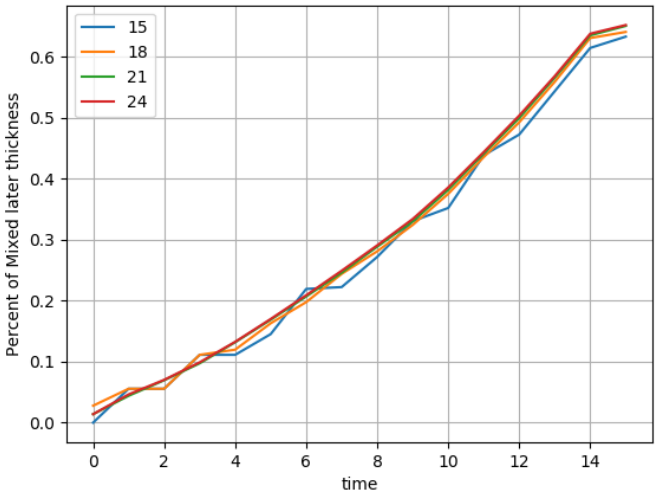


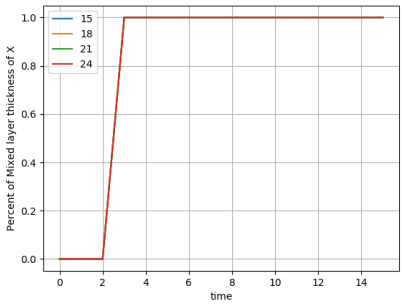
Figure 2: Percentage of mixed layer

At Figure2, the value increases from about 0 to about 0.7 over all resolutions. At low resolution such as 15 and 18, line graph is fluctuated. But the fluctuation decreases as resolution increases, and the graph becomes linear graph at resolution 21 and 24. Since I assume if a layer has any one mixed value which is between 0 to 255 then the layer is mixed layer, the percentage of the mixed layer becomes high. And whole computation time from resolution 15 to 24 and time 0 to 15 is 183234375000 sec.

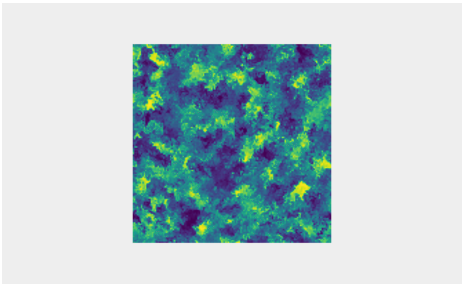
4 Problem4

The PercentMixedLayerForMiddle class is designed for measure both middle of x-orthogonal and y-orthogonal plane's percentage of the mixed layer. The method to get the percentage of mixed layer is same with previous problem.

But current section ask only using the slice located at middle of the plane. So I just calculate middle slice's percent of mixed layer thickness. The designed class has three functions such as init, PercentX, and PecentY. The init function define resolution, the PercentX function calculate the percentage of mixed layer at x-orthogonal plane and PercentY function calculate the percentage of mixed layer at y-orthogonal plane. In the problem, if the name of 3D array is Data, then x,y,z is defined by x,y,z = data.shape.



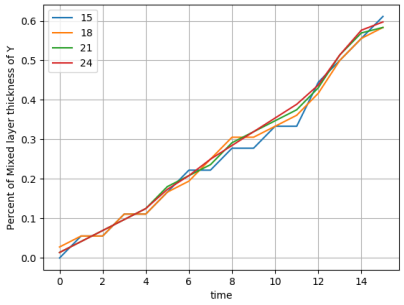
(a) X plane



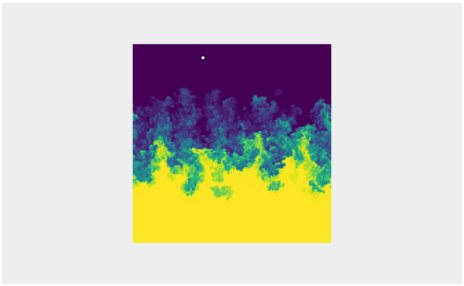
(b) X view

Figure 3: The percentage of mixed layer of X and its image

Figrure3-(a) is the percentage of mixed layer of X plane. And Figure3-(b) is corresponding plane view. In this plane, the result of the percentage of mixed layer is unreasonable. As Figure3-(b) shows, small amount of yellow phase is mixed on whole domain with blue phase. That is the reason initial timestep's results are zero and from timestep from 3 it becomes 1.



(a) Y plane



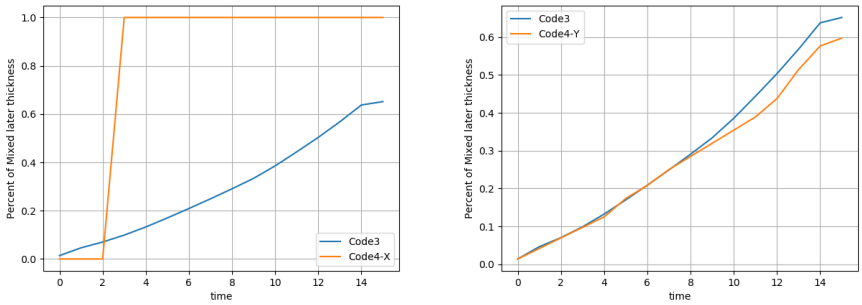
(b) Y view

Figure 4: The percentage of mixed layer of Y and its image

Figrure4-(a) is the percentage of mixed layer of Y plane. And Figure4-(b) is corresponding plane view. In this plane, the result of the percentage of mixed

layer's trend is same with the result of problem3. But fluctuation of low resolution's graph larger than that of problem3. And total computation time of both X and Y plane is 35921875000 sec.

5 Problem5



(a) Z-X plane (b) Z-Y view

Figure 5: Compare the result of code3 and code4

Figure5 is comparing the result of code3 and code4 at resolution 24. At Figure5-(a), two graphs are significantly different each other. Otherwise, at Figure5-(b), two graphs are similar each other. Until timestep 8, two graphs are almost same. And computing time for code3 is 156609375000 sec, computing time for code4-X is 14703125000 sec, computing time for code4-Y is 15062500000 sec. Computation time of code4 is less than 10 percent of Code3. Thus, my conclusion is that Y-orthogonal plane is useful for measuring the percentage of mixed layer. And X-orthogonal plane is unreasonable. In addition, if approximation answer is allowed, code4-Y result is better than that of code3 because of computational time.

6 Problem6

Problem1 result raw file is used for visualization. At Paraview, data scalar type is changed to unsigned char, data extent is convert to 0-143, 0-143, 0-143 respectively x,y,z axis, data axes grid is on, and blue value increase to maximum at color map editor. And result is saved as Problem6 pvsm.

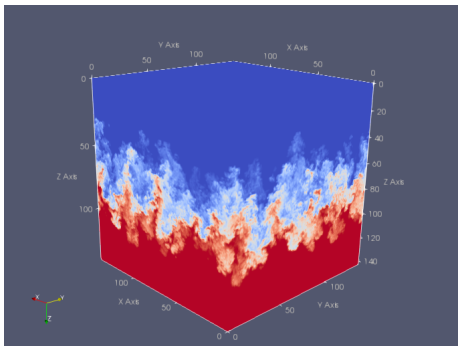


Figure 6: Paraview image of raw file

7 Problem7

a) At Paraveiw, show isosurfaces is on and add range 0 and 255. And the result is Figure7. Isosurface of 0 is invisible and only isosurface of 255 is visible. I will use marching cube theory to explain why 0 isosurface is invisible. Eight voxels compose one cube. This cube moves through whole domain. If none of the edges are cut the table then it return 0. This occurs when cube index is zero because all vertices below the isosurface. Or it also happens when all vertices are located above the isosurface. In this situation, I think it happens because of first situation (cube index is zero).

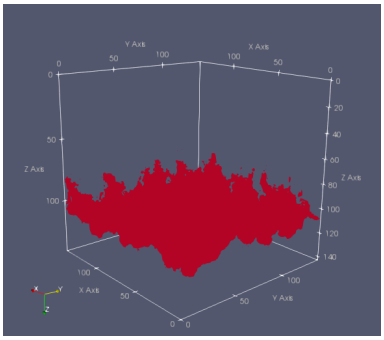


Figure 7: Isosurface of 0 and 255

b) space = [8. 21. 34. 47. 60. 73. 86. 99. 112. 125. 138. 151. 164. 177. 190. 203. 216. 229. 242. 255.] is utilized to make 20 isosurfaces which are equally spaced. Figure8 is result.

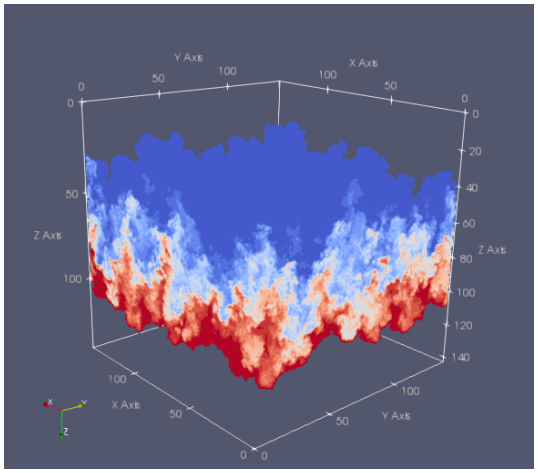


Figure 8: 20 equally spaced Isosurfaces

8 Problem8

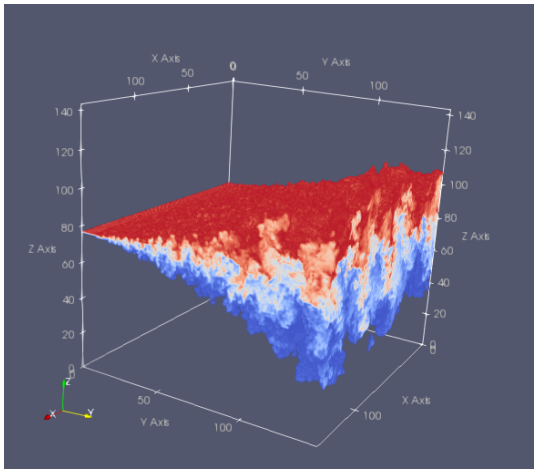


Figure 9

The FindMixedLayer is defined to make the result. The class has three functions such as init, combineY, and combineX. The init funcitone define current resolution. The CombineY function combines layer through y-axis. And the

CombineX function combines layer through x-axis. Two combines functions' results are same. I just want to check the result of two direction's combine. My answer follow CombineY function. New empty 3D layer named total is defined and divide the layer as 16 pieces. And each time's piece is input at each piece from 0 to 15 timestep. Figure9 is result at resolution 24. It is possible to check mixed layer thickness is linearly increase and it will be useful for checking diffusion rate.

9 Problem9

a) I read Accelerated Isosurface Extraction Approaches. In this section, main goal is how to extract isosurface with high speed and low memory. Marching cube method is famous. But it's computation time is high for large dataset, so it is not optimal method for time complexity. Isosurface extraction is categorized by three part, namely geometric, value-base,and image-base. And there are two acceleration method. First method is Near-Optimal Isosurface Extraction (NOISE) method. It uses span space as the underlying domain. In addition, kd-trees is employed as a means for simultaneously ordering the cells according to their maximum and minimum values. It is my first experience seeing the tree structure is really used for real situation for fast process, so it is helpful for me to improve my knowledge. Second method is View-Dependant Isosurface Extraction. It tries to extract specific Isosurface's visible part. That is, if the Isosurface is not visible, the algorithm not extract that part. Thus both methods are helpful to reduce the computational time of Isosurface extraction.

b) First, I learn Isosurface can be both structured and unstructured grid. Since I am a mechanical engineer. I just learn structured grid at Finite Element Method class. In addition, I learn if the size of dataset is huge, then not only the size of dataset but also the size of Isosurface can cause trouble such as time-consuming.

c) Near-Optimal Isosurface Extraction is one advanced method. It use span space decomposition and employ kd-tree. Using the kd-tree containing pointers to the data cells as an index to the dataset, the algorithm can rapidly answer isosurface queries. And the number of cells intersecting the isosurface can be found by incrementing a counter rather than constructing polygons from a node. That is, extraction time is improved.

d) In the part of kd-tree, it is not easy to find median of data in optimal time. And it will require total four list of pointers so it can cause memory problem. This paper keep trying to explain how to reduce the memory consumption. For example, pointerless Kd-Tree reduce the memory requirements to two real number.