

2.

Comparison between CPU and GPU based approach on Slicing:

- i. In the CPU approach, instead of triangulating each and every unit cube, the plane parallel to the XY plane has been taken as a grid structure. The grid is a collection of some squares where each point in one square is 0.5 distance apart. The scalar value associated with each point on the grid is computed by computeScalar() function, trilinear interpolation has been used here.
In the GPU approach, instead of breaking down the grid into smaller squares, the whole plane has been represented by two triangles.
- ii. a total of $256 \times 256 \times 2 \times 2 = 262144$ squares are being considered, and for each square, the program generates $262144 \times 2 = 524288$ triangles whereas in the GPU approach only 2 triangles have been used for drawing the slice.
- iii. For the CPU approach, the time spent on preprocessing is very high compared to the GPU approach.
- iv. In the CPU approach, the code needs to compute the scalar value for each vertex for all the triangles using trilinear interpolation. Hence it increases the computational complexity, whereas, in the GPU approach, this computation is avoided by doing texture mapping in the fragment shader.
- v. Both approaches almost generated similar output in terms of image quality.
- vi. Both approaches are taking 46.72 fps for rendering the output.

3.

Comparison between CPU and GPU based approaches on Isosurfaces using marching tetrahedra:

1. In CPU based approach, each small cube is considered one by one and then is divided into 6 tetrahedra. In each tetrahedron, there are 4 edges. For a given scalar value, the intersection has been computed and triangles have been generated based on that interception.

In GPU based approach, coordinates of the lowermost points have been sent as points primitives. From these points, all the computation of marching tetrahedra is being done within geometry shader then the newly generated triangles have been rendered as output.

2. In CPU based approach, the preprocessing of data takes much longer. In GPU based approach there is no preprocessing but the same computation is done within the geometry shader.
3. In CPU based approach, image quality is far better than the GPU-based approach.
4. In CPU based approach, it is giving 46.72 fps for rendering the output whereas in GPU based approach it is giving almost 4-5 fps.