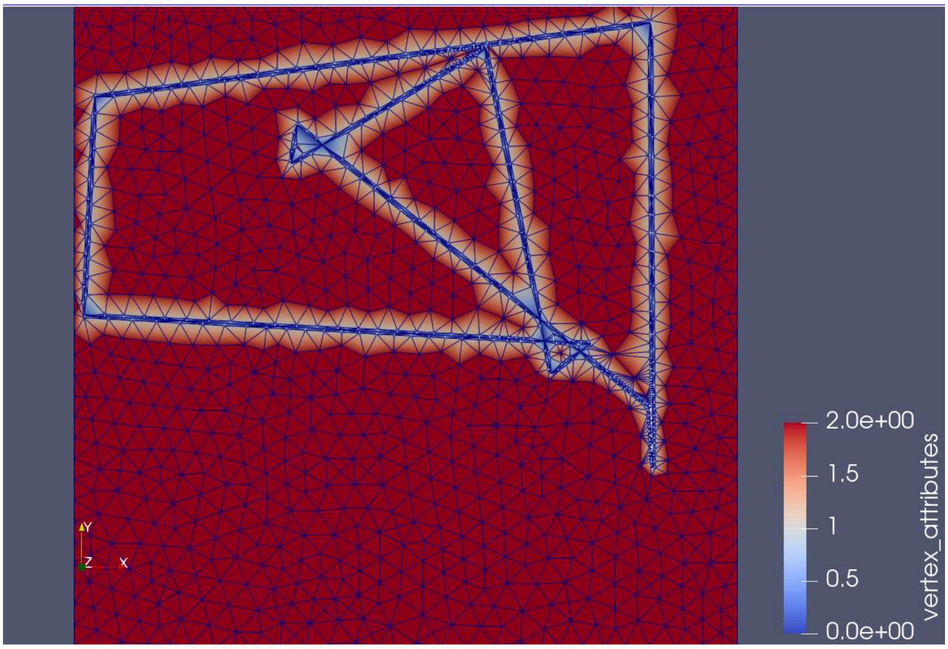
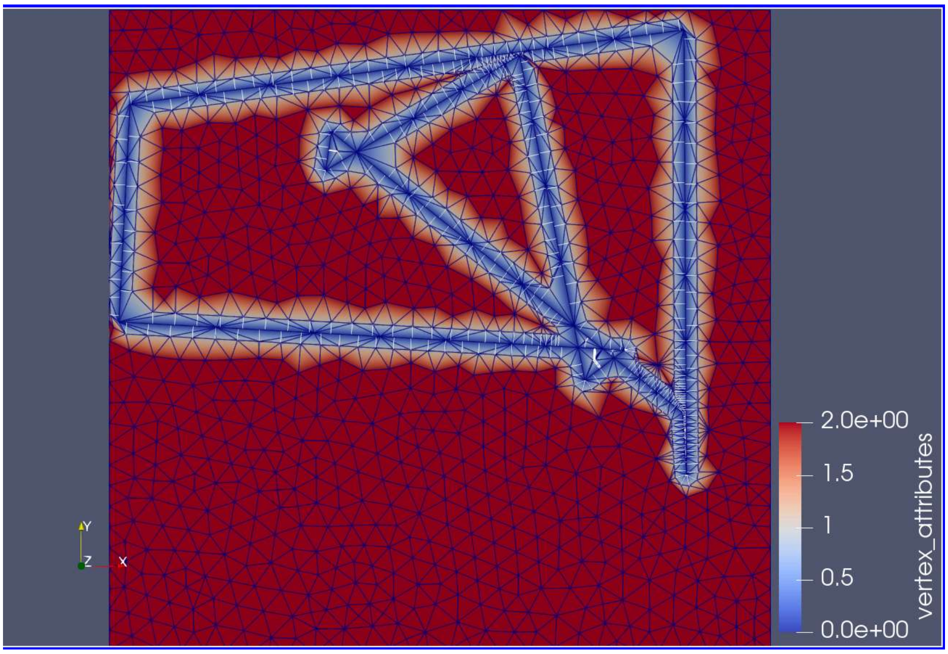
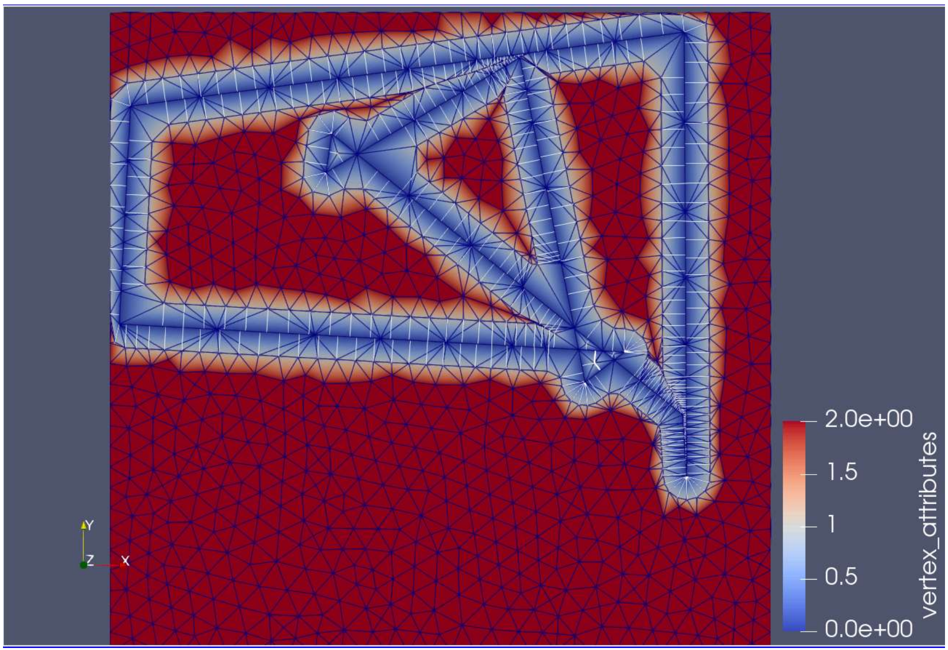
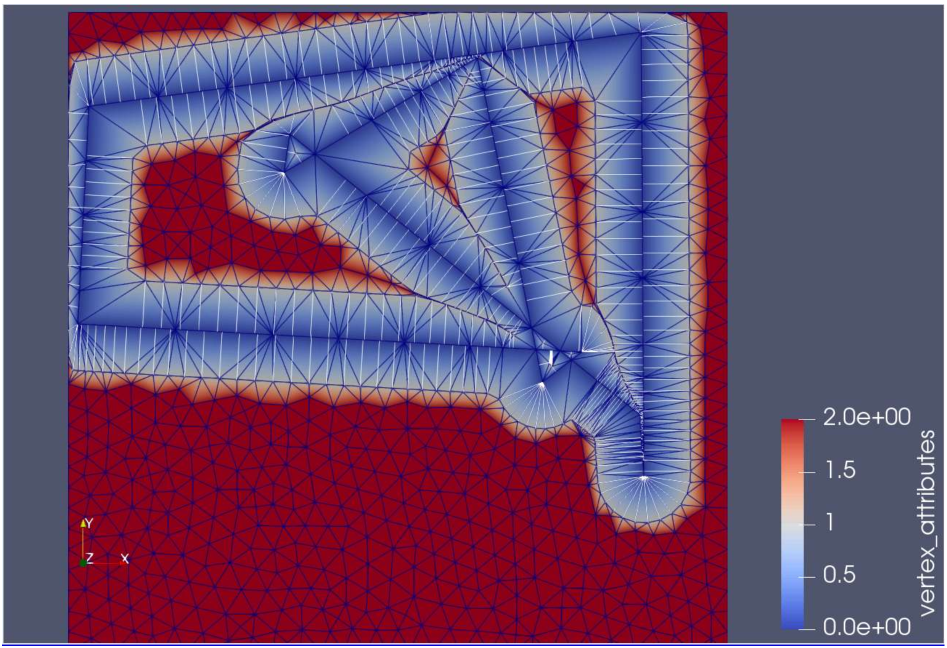
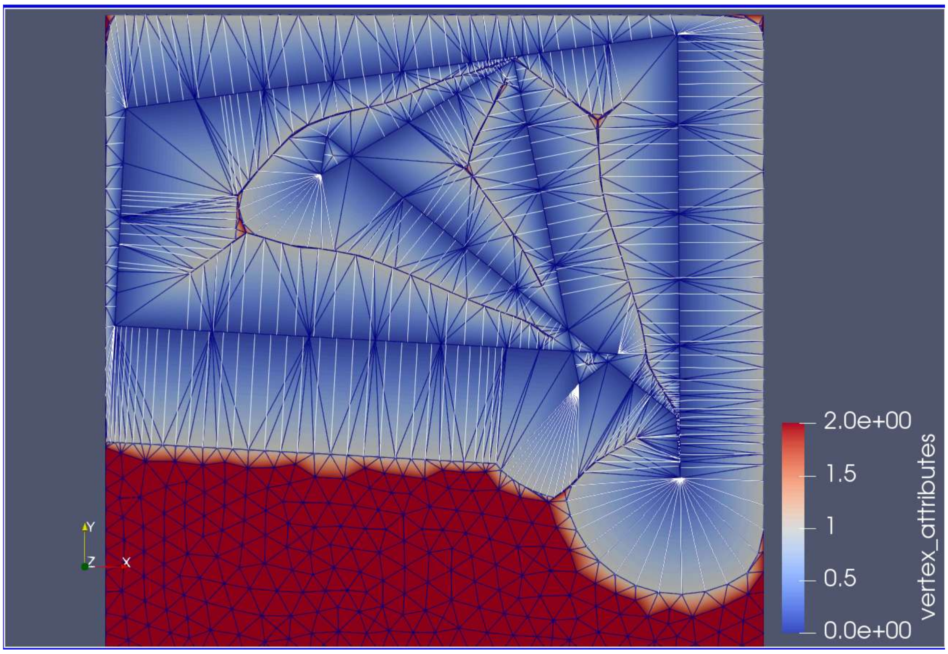
Report for the Prototype of Topological Offsets

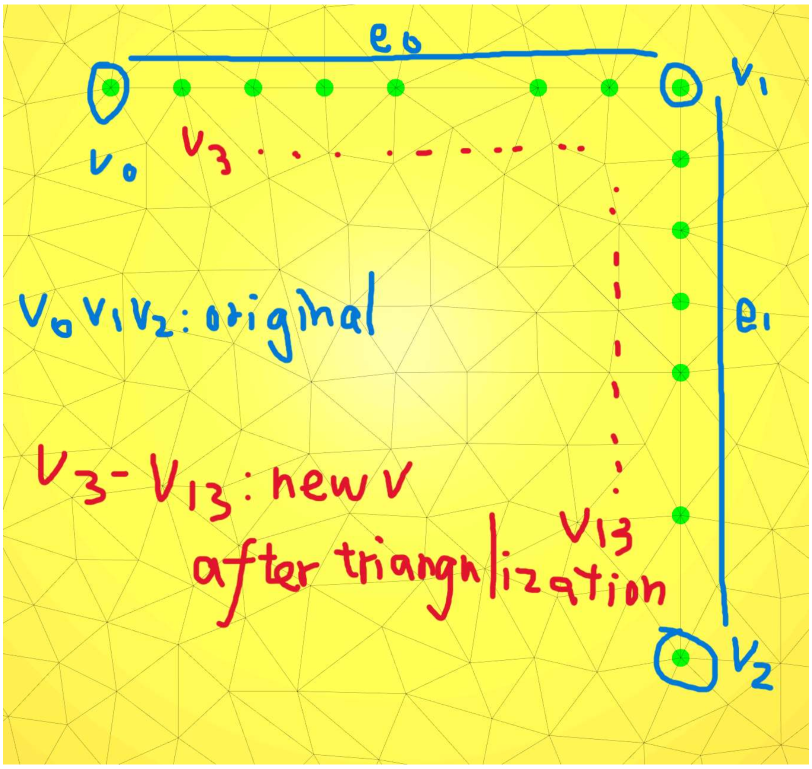
Zhouyuan Chen, 2023/08/01

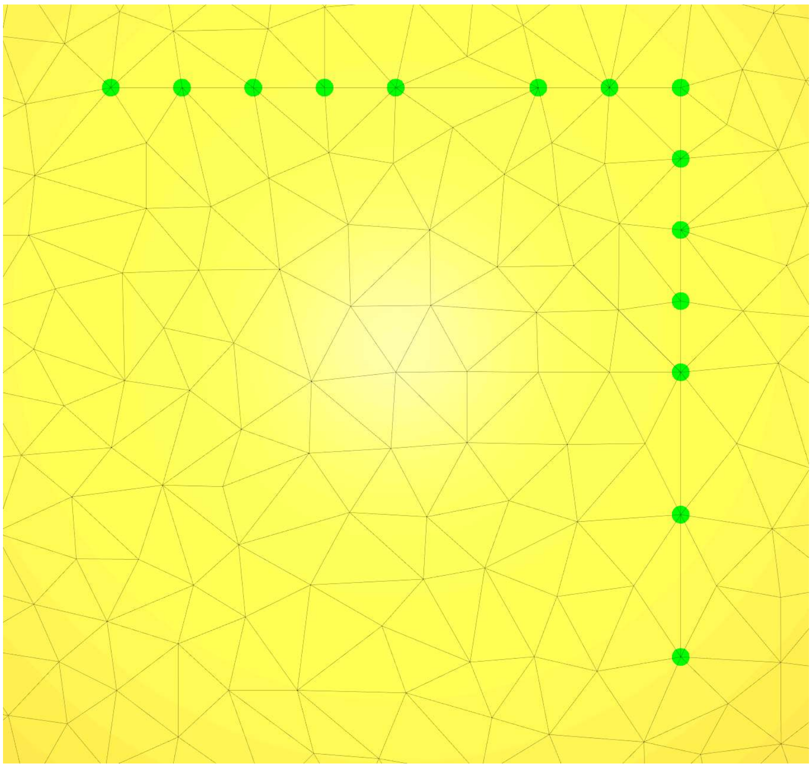
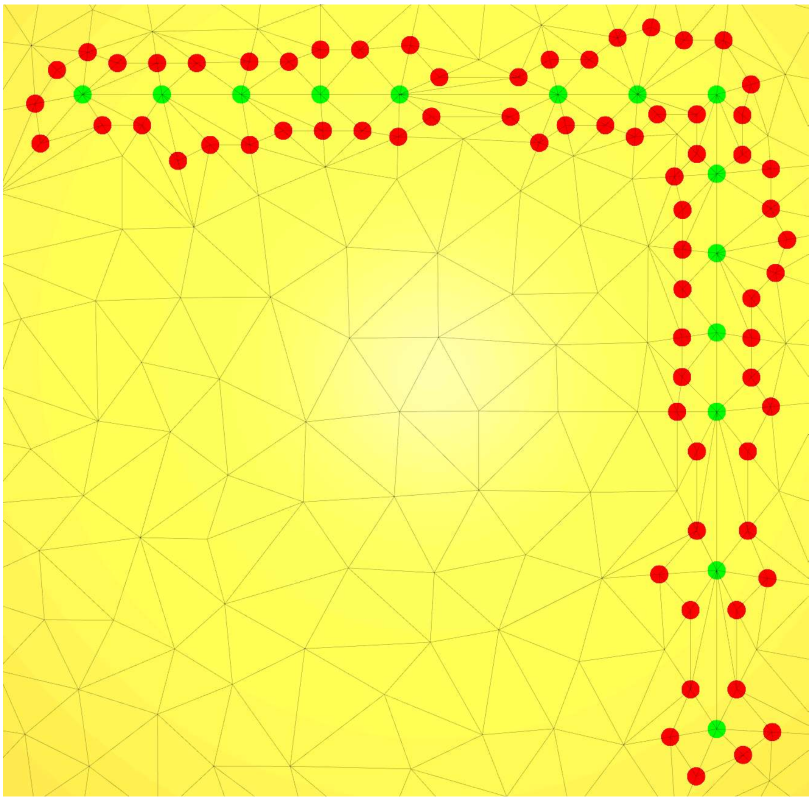
Current Result

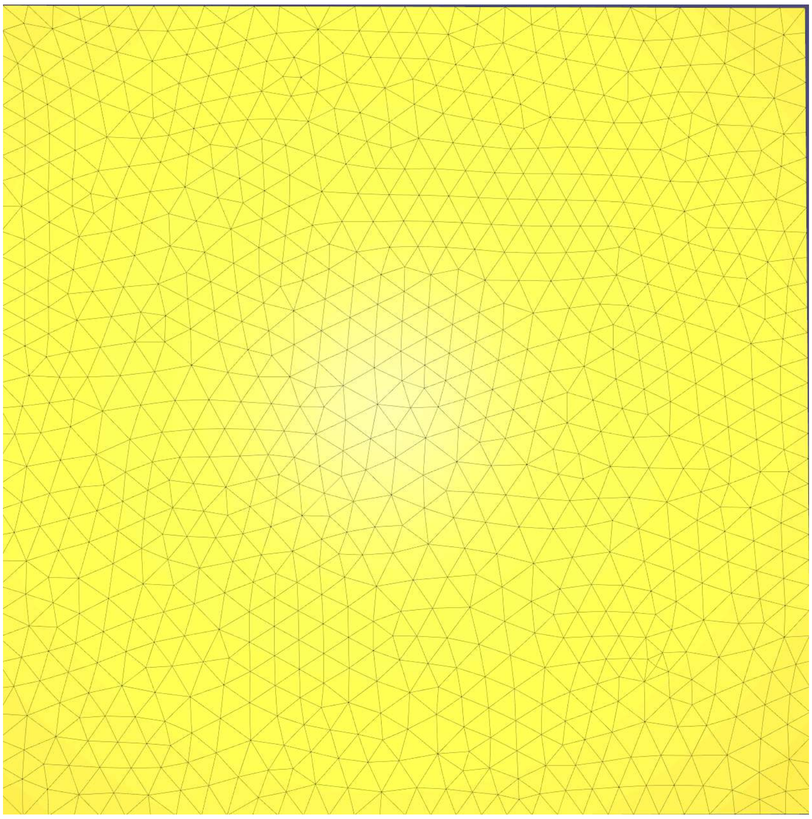
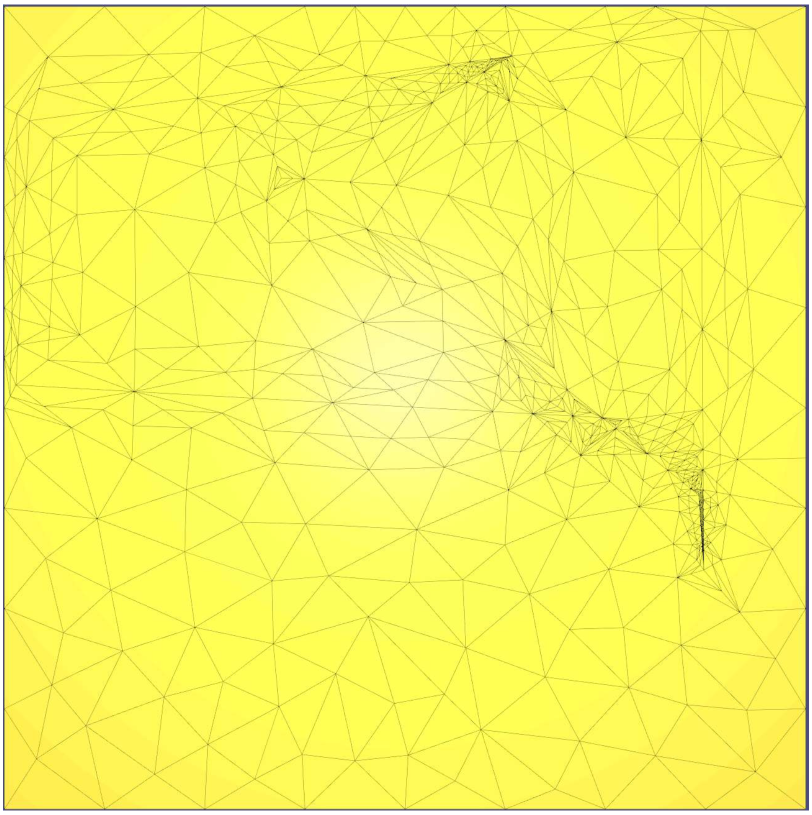
inflation rate	**the inflation rate is based on the average length of the scaffold**
10%	
50%	

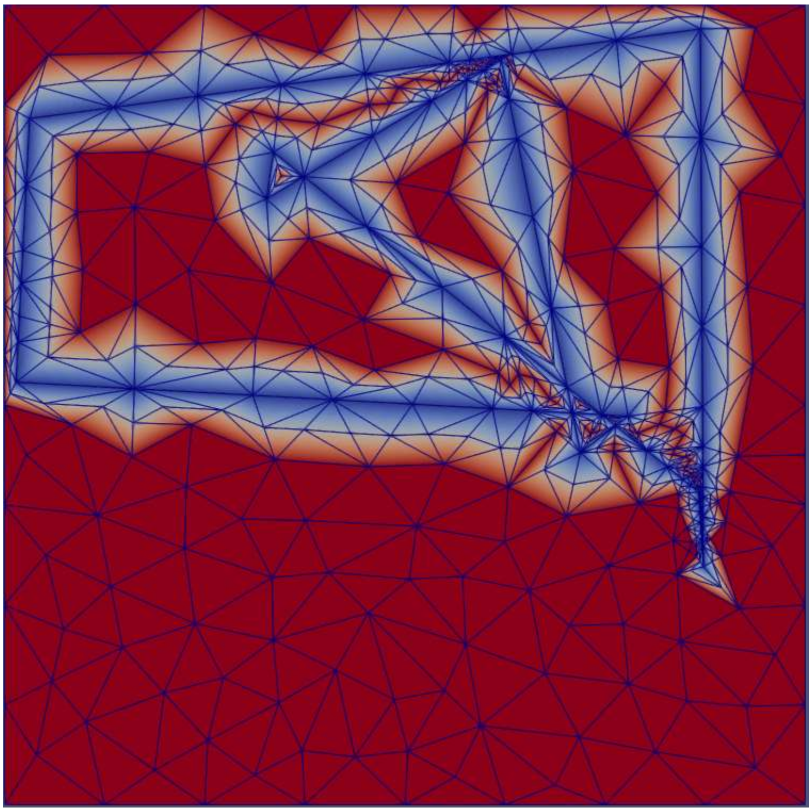
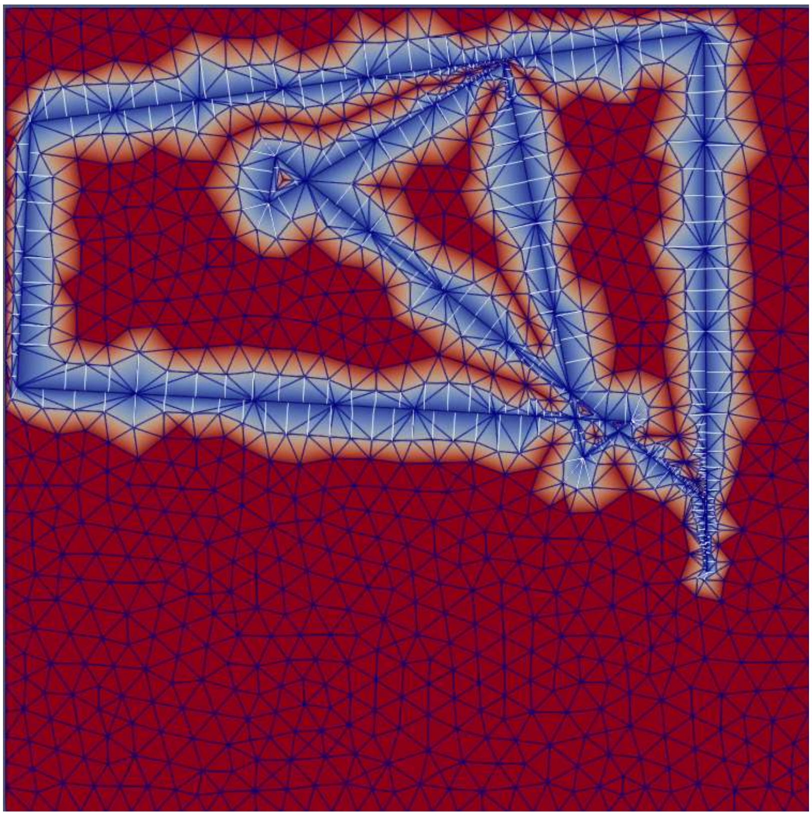
inflation rate	the inflation rate is based on the average length of the scaffold
100%	
200%	

inflation rate	the inflation rate is based on the average length of the scaffold
500%	

previous skeleton description	
original_describe	

previous skeleton description	
triangulization	
inflation	

previous skeleton description	
remeshing_general	
remeshing_input	

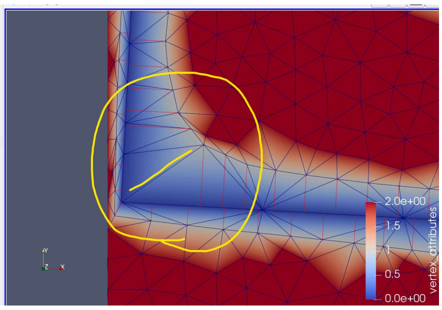
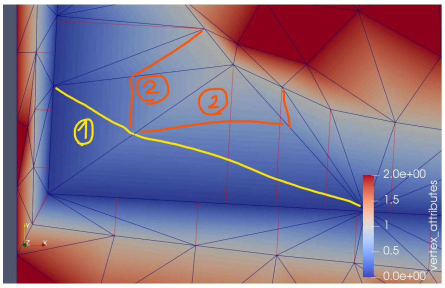
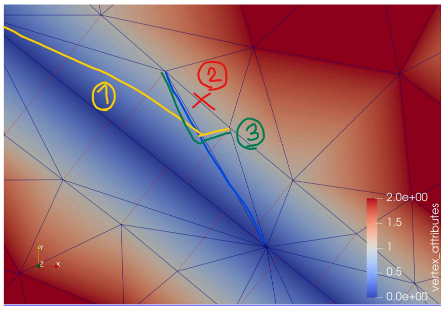
<table>
<tr><th>previous skeleton description</th><th></th></tr>
<tr><td>before remeshing</td><td></td></tr>
<tr><td>after remeshing optimization</td><td></td></tr>
<tr><td></td><td>NOTE: the pictures above are displayed by the libigl, you can have a closer look by using Paraview, checking the .vtk files under D:/(this is the default direction) after pressing bottom 1. Besides, please don't export the file while opening it.</td></tr>
</table>

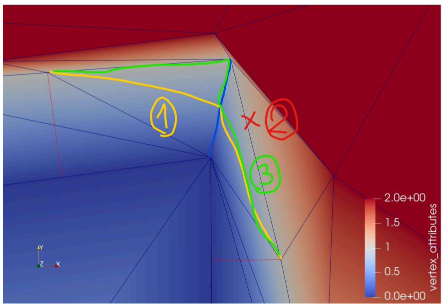
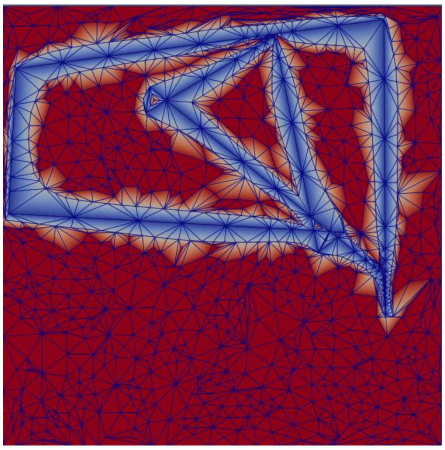
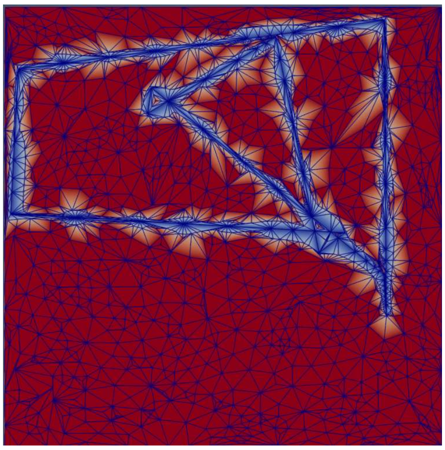
Remeshing Summary

this block reported some problem I found during remeshing part's code work. you can find this mesh in the "output" file.

~~Split Process On The Interior Area~~

this part records some specific situation on the split operations. on the pictures, I drew a personal solution for these cases. the delete operation below relates to the inflation edges' vector, so actually we won't delete the edge in the topology.

Special Cases	
BLUES is the input, RED is the exterior area, WHITE is the offset	
corner split case	
	
	we need do addtional two split operation for the two enges on the faces but not belongs to the input.
interior split case	
	first, we do a split, then deleted the offset anotation for the orange edge, then add new green edge to into the offset vector

Special Cases	
	same as the situation above, but in this case, there are three neighbour vertices
	actually, I have implemented this part, I don't know if this is more better or not since it I thought the performance is not good as I assumed after applying such a modification, maybe this is because that I didn't implemented corner case? have no idea now. you can check these two .hdf files, their names are split_interior_berore and split_interior_after
	before
	after

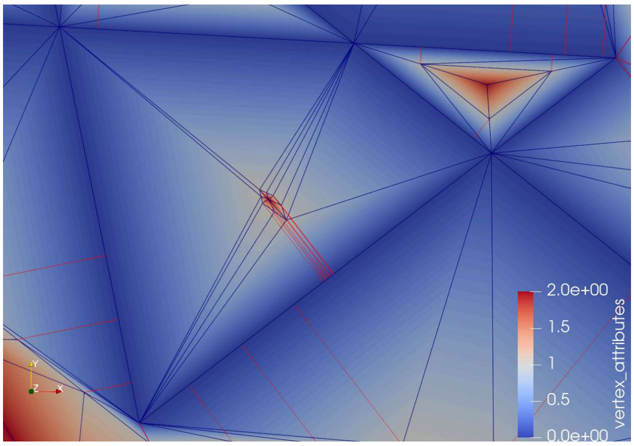
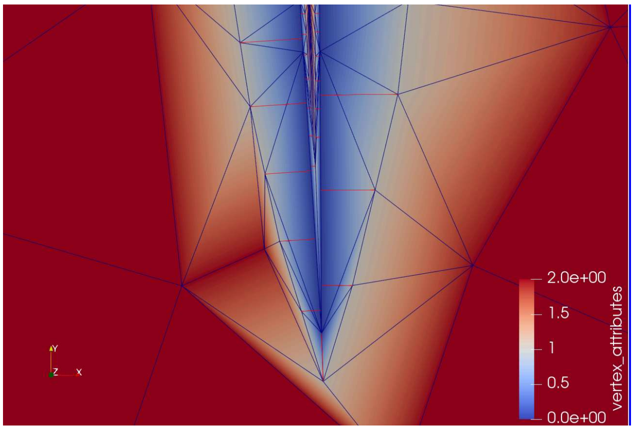
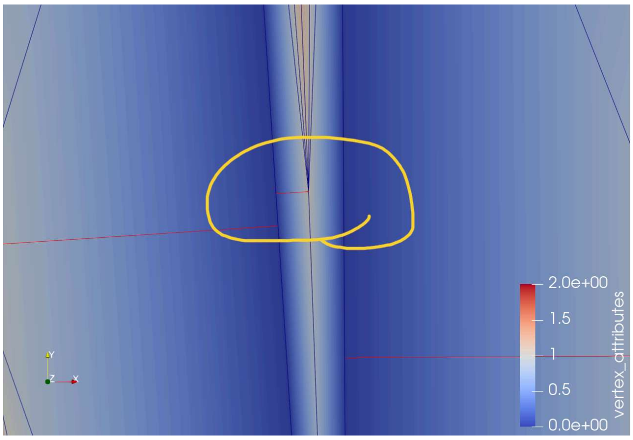
~~Swap/Flip Tradegies~~

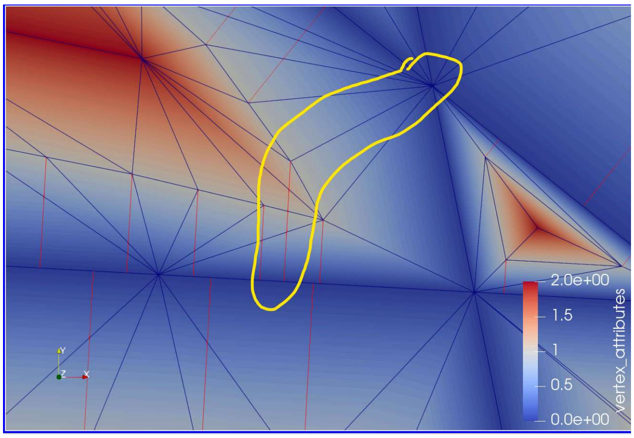
I used two swap/flip methods to do remeshing. equalized valence and the mean ratio value.

I found that I made a mistake while I implenmenting the mean-ratio method, for now, the mean-ratio method has a better performance.

Projection Process

All these problems actually more likely to happen in 3D surfaces' boundary or feature-line area.

Problems	
BLUES is the input, RED is the exterior area, WHITE is the offset	
case 1	
	projection points are not reasonable in some specific cases. this problem could be solved by using incident vertices information. but it still have a potential problem likes below.
~~case 2~~	
	the corner situation display an another bad projection. again, this could be solved by using incident vertices information. maybe we could do a resample technique on the corner, something like doing a split.
	
~~case 3~~	

Problems	
	this is quite similar to the case 1.

~~**Solution Discussion**~~	
idea 1	we can avoid it by not finding global closest simplex, just finding it in vertices' incident one-ring area
idea 2	maybe set a threshold to detect the sharp corners, then directy connect the offset vertices to the input vertices
idea 3	I thought this idea is exactly the idea mentioned in the notes, but it seems not to use the "incident" vertices directly. I remember that, in the last discussion, we said there is an advantage of our project is that we can simply use the incident vertices instead of finding them by using BVH or other acceleration structure. Therefore, I thought maybe this part is still waited to be discussed. In my opinion, I had a very simple solution. I am not sure if this would work, but may be we can connect the offset points with all of its incident input's vertices, then do something like phong-normal. and these normals will be optimized in the next stage of our pipeline