

## WEEK 3

ARDEN RASMUSSEN

### MESH

I need to implement boundary condition parsing in the PSLG, and when I make the mesh constraint system, I need to be able to specify the boundary edges, So that I can then use the boundary conditions in the actual FEA. I should make a new struct that stores the boundary edge, and the values representing the boundary condition. I don't think that this should be too difficult.

I really need to implement constrained mesh generation. This has proven difficult in the past. I should re-read the Sloan paper and try their method again, although they annoyingly don't give the FORTRAN code :(. Read through it very carefully, and give it a try. I'm pretty sure that I have the unconstrained version working fine. The constraint implementation is also where the boundary conditions will be propagated into the mesh from the PSLG. I have two ideas on how to do this. One way is to store a list of edge references with the associated boundary condition, or change each edge to have no boundary condition, and then change the appropriate edges. The second would be simpler, but the first would be better.

Then I need to implement the triangle virus. This does not seem too bad. Just a variation on depth first searching. I think that I should be able to implement it fairly quickly. Just keep a list of the newly infected triangles, then remove the last infected triangles. Repeating the process should remove all unwanted triangles.

Finally we do the mesh refinement. I think that I will use Chews algorithm. This I don't think will be too hard to get working. The only somewhat challenging part is the edge splitting. If I'm splitting a constrained edge, I need to make sure to propagate the boundary conditions to the two new splits.

---

*Date:* February 13, 2019.

## REFINEMENT

- Which boundary conditions should be implemented, and how?
  - Dirichlet. Just a constant value, or time dependent?
- Thoughts on boundary condition implementation? Should every edge have a boundary condition, and just have a null condition, or should they be stored by reference to an edge. What about in the file? Should the file have a null condition for each edge? This would make it easier to read, but would result in a large file size.