

# Homework 1

---

A triangle mesh is a special type of simplicial complex. Edges and faces are simplexes and have orientation. When they together form a simplicial complex, we would like their orientation to be coherent.

Consider the orientation of each half-edge: source vertex to target vertex, and the orientation of each face: the normal direction decided by the linked edge cycle and right hand rule. When the orientation of a mesh is consistent, each non-boundary edge is shared by two half-edges with opposite directions.

Some mesh formats, such as the ".m" file, strictly follows this consistency. Meshes are stored as (1) an array of vertices, and followed by (2) an array of faces, each of which indicates consecutive ids of its three vertices. And these three vertex ids define the cycle and the local normal direction.

Many commonly used formats, such as ".obj" and ".off", also allow the mesh to be stored as an unorganized set of triangles. When the triangles are unordered, geometric algorithms or renderer usually do a repair to make it coherent.

The Homework 1 therefore is:

Write a program to read in a ".obj" format mesh file, check its coherence and perform repair if necessary, and then output it to a ".m" file.

A Trivial Example:

A rectangle [v1,v2,v3,v4] represented by two triangles [v1, v2, v4] and [v2, v3, v4].

If the input .obj file is as follows:

```
v -1 0 0
v 0 -1 0
v 1 0 0
v 0 1 0
f 1 2 4
f 2 4 3
```

Then it is inconsistent.

The correct face order should be:

```
...
f 1 2 4
f 2 3 4
```

Therefore, your program should repair the second face by changing its vertex-id order to 4 2 3 (or 2 3 4, or 3 4 2).

A .obj files example is provided, please write a console program to correct it.

Format: yourprogram.exe [in.obj] [out.m]

Input: in.obj file

Output: the repaired out.m file

You can get the viewer G3dOGL.exe from "<http://www.ece.lsu.edu/xinli/Tools/G3dOGL.exe>"

You should be able to directly see your output .m file.

(1) it should succeed to open your out.m file

(2) the mesh I provided is a closed genus-3 surface with only one connected component

If G3dOGL fails to open it or you see a lot of unconnected pieces, that means your program has bugs. You can use the mesh library to try to read in your .m file and debug.

When you submit your homework (your program and the compiled .exe file), please indicate your computational complexity.