**Process Meshes:**

* Smooth meshes
* Remesh so that bones have triangle edge lengths ~1mm and cartilage has edge lengths ~.8 mm
* For cartilage:
  + make sure there are no triangles with bad aspect ratios (one side much smaller than the others)
  + remove “inside” surface. I’m not sure if this will be possible to code, but it’s worth a try!
* For femur and tibia bones
  + Check height:width ratio
  + If more than for reference mesh (mean mesh), crop the mesh to have the same ratio
  + If less, add a cylinder on to the end of the bone to make it longer (also unsure if this can be coded)
* Save as new .stl files

**Create anatomical coordinate systems for the femur, tibia, and patella:**

* Existing Matlab code (ACS.zip), partially converted to python

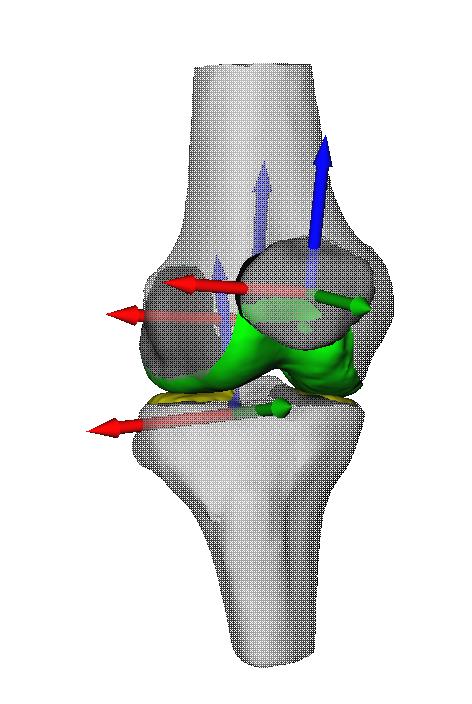
The functions to create the coordinate systems are buildfACS.m, buildtACS.m, and patellaACS.m. Those call a bunch of subfunctions, so I included all of those too. There are some functions in there that you don't need, but I just kept everything to try not to miss anything, but let me know if I did miss a subfunction.

The goal is to read in the stl mesh of the bone and calculate the anatomical coordinate system and write it to a text file. It would also be good to create a plot showing the bone with the coordinate system to verify it's correct.

Some notes:

* the functions were written to work with .iv files for the meshes, but you should switch to .stl for the python version. Also, the triangles/connections in the iv files start from index 0, which is why you might see +1 in some spots for that in the matlab code.
* the femur and tibia matlab functions write out a bunch of cropped mesh files as a part of the process, but there is no need to keep those, and it would actually be better if they weren't saved
* the tibia function requires a point on the front (anywhere on the front of the tibia) of the tibia to be input as well.
* the femur function sometimes doesn't find the long axis correctly if the bone is too short (occurs fairly often with MRIs), so that is a known issue.

I also included an example of some iv meshes that work with the matlab code. And here's an image of the coordinate systems (red=x, green=y, blue=z). I also included an stl version of the same meshes.



*Determine ligament attachments:*

* *Get node correspondence for bone meshes with the reference mesh*
  + *Use* [*https://github.com/ohirose/bcpd*](https://github.com/ohirose/bcpd) *for this*
* *Use the node numbers ligaments are attached to in the reference mesh to update the attachments in the patient-specific model.*

*Determine wrapping surfaces:*

* *Also uses node correspondence*
* *Use specific nodes to fit cylinders/ellipsoids*

*Write new model to xml file*

* *Align bone and cartilage meshes to model*
* *Update links to geometry*
* *Update ligament/muscle attachment points*
* *Update wrapping surface parameters*

*Model is here* [*https://simtk.org/projects/opensim-jam*](https://simtk.org/projects/opensim-jam)