Official Report

Summary

Since work really got started (around Sept. 9-10th), I feel like I have made good progress in learning about CFD as a whole through lots of tutorials, and a series of biophysically based small projects that I plan to bring together into a final FEA/CFD model, as well as a more real-time model in Unreal Engine. A majority of the first week was spent generally absorbing as much as I can CFD and related software, linking things back to previous related courses, and hopefully filling in any gaps in knowledge. This consisted of watching tutorials, googling unfamiliar terms and concepts, reading forums, getting software set up (OpenFOAM and eventually Ansys Fluent in the coming week), completing the equivalent of 'hello world' for sims, and generally absorbing as much as I can CFD and related software. I felt a bit like a headless chicken in my researching during this time was confined to 'OpenFOAM', with other key words like 'flesh','heart','blood',etc. and looking for any sorts of biological simulations that I could deconstruct and learn from. I have maintained a list of all the resources/videos/tutorials at the bottom of this section I have come watched and found useful.

During week 2, I became a little more comfortable with OpenFOAM, but eventually came across Ansys Fluent after reading through forums and built equivalent 'hello world' models with much more ease and clarity in learning. The general consensus seems to be that the open source nature of OpenFOAM makes the learning curve much steeper but more powerful in the hands of an expert (due to its modularity and flexibility), and that Fluent is easier to learn and much more user friendly, letting beginners get familiar with things like discretization methods, turbulence models, enabling and disabling models and features etc. by 'ticking boxes'. This also lead me to much better learning sources given this fundamental difference (not that OpenFOAM's was bad, just far more veteran), as more biological topics (see resources) were appearing in my searches. Venturing through this rabbit hole of information was much more fruitful than last weeks however. I also got my Unreal plugin to the point where it renders tetmeshes in a halfedge data structure. The dynamic UV's are not implemented yet, but what I have functionally works for the time being.

For weeks 3 and 4 (which brings me up to Oct 13th), I have been working through a variety small projects of more complex, biologically based workflows and simulations.

This includes static bone model, adaptive deformation of a similar bone model, blood and arteries. Cornell's CFD course ansys.com/cornell, which seems to be biologically based, had the models and meshes that were readily usable from CT data that Ansys converted to usable geometry. The intention is to eventually couple these examples together in a modular fashion, while simultaneously thinking about how I want to put together the final, larger, and more complexed coupled model. This is the majority of what you will see below.

Resources

Getting Started with Ansys Fluent - https://www.youtube.com/watch?v=3pgSIOGDOP8

Cardiac Simulations - https://www.youtube.com/watch?v=yaHgLItK8AY

Shorter Cardiac Talk - https://www.youtube.com/watch?v=pqgrOKVYbi4

FSI in Biological Simulations - https://www.youtube.com/watch?v=4_VO_YJcte4

DynamicMesh - https://www.youtube.com/watch?v=72RvIkETWLg

Ansys Fluids Courses - https://courses.ansys.com/index.php/fluids/

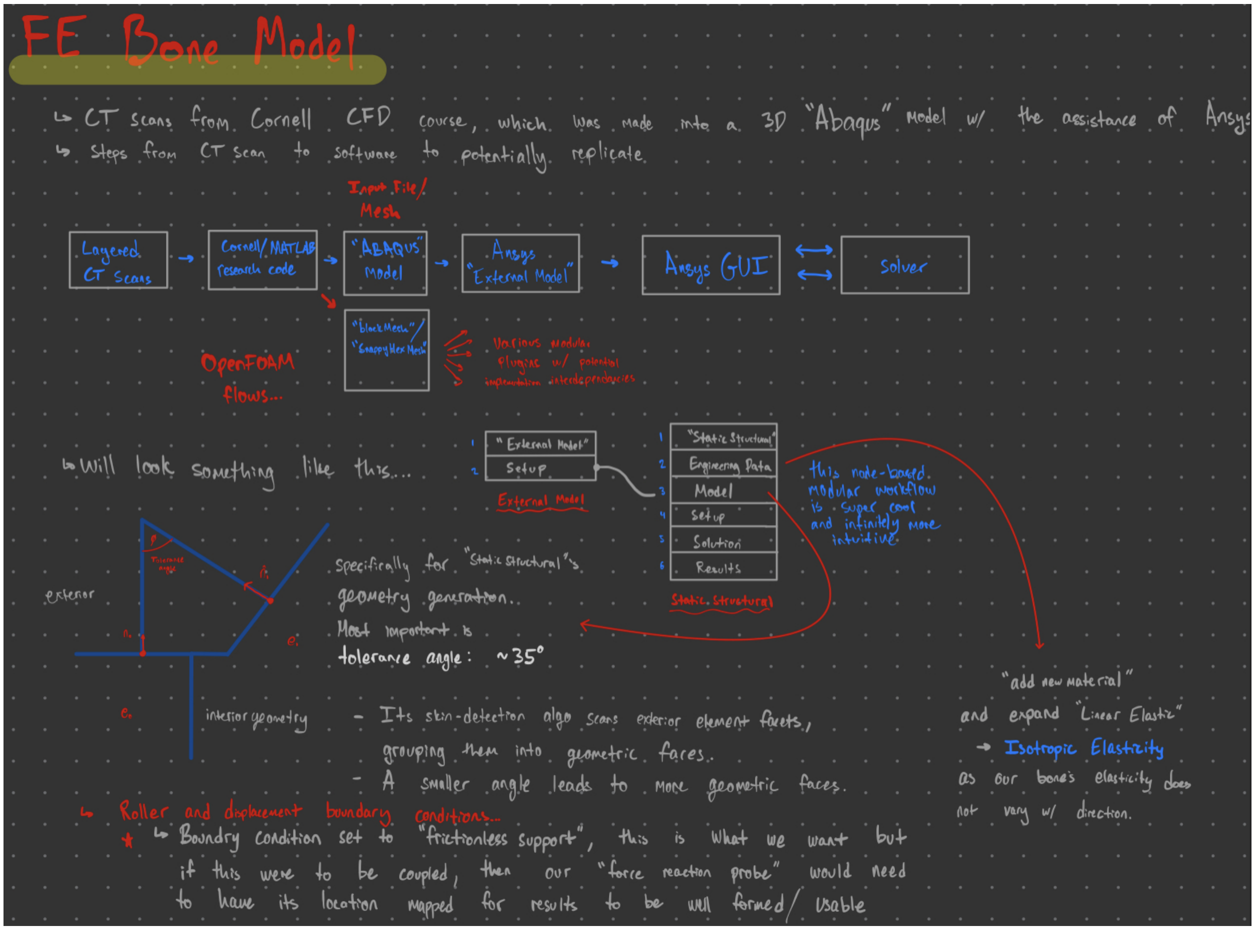
Ansys Structures Courses - https://courses.ansys.com/index.php/structures/

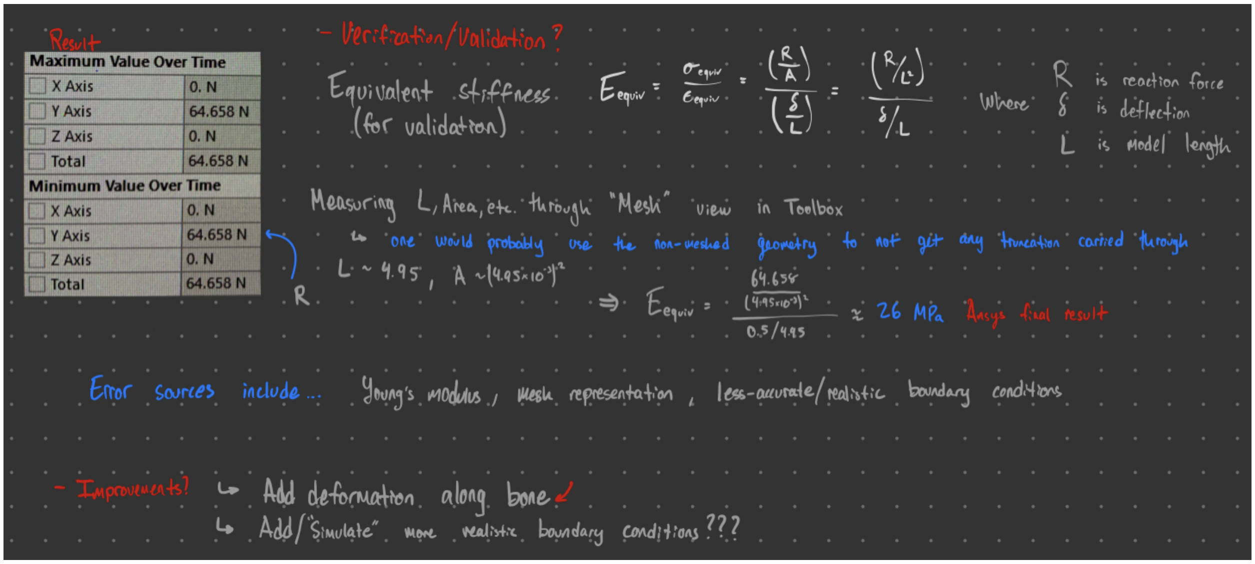
Ansys Materials Courses - https://courses.ansys.com/index.php/materials/

High Resolution FE Model of Bone

→ Geometry file (made by Sean Harvey at Ansys) can be found in the git repository, and was downloaded through Ansys Learning Forum with the associated exercise: https://forum.ansys.com/.

I decided to start structurally rather than with fluids as it seemed more intuitive. It started with an ABAQUS geometry file already sorted for us, leaving work contained within the given specifications for the geometry/problem and contained within Ansys Workbench and Static Structural. The main focus was on the workflow itself, from geometry generation to setting material parameters, and finally setting boundary conditions. After the model was simulated, it checked out to be within 10% of the experimental result, with the main potential error contributions from Young's modulus, 'non-realistic' boundary conditions, and mesh representation. An interesting take away was realizing how much the geometry itself could bias or effect the simulation, which lead into a broader exploration of mesh quality, skewness, etc. The results for this mini-project was not very visual (since displacement was only in the Y-axis), but served as a good start point for adding bone deformation

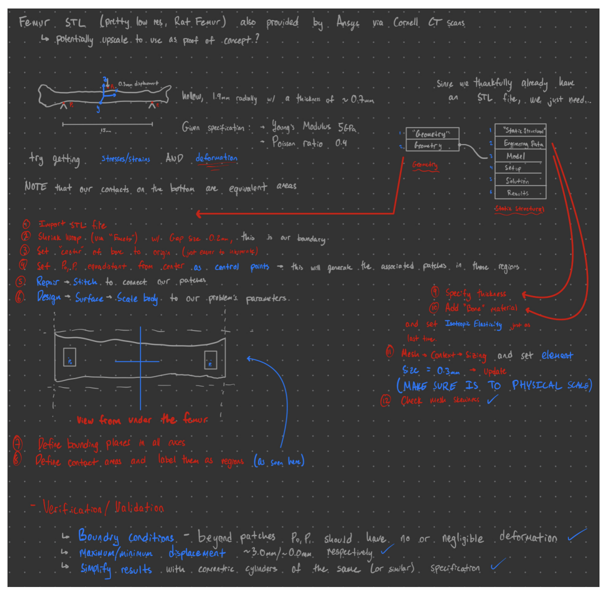




Bone Deformation

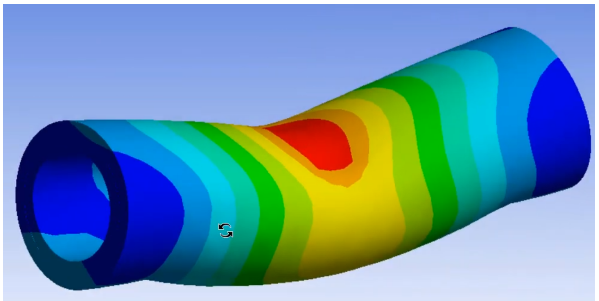
→ Geometry file (again provided by Ansys) can be found in the git repository.

Adding deformation was done with a lower resolution CT scan for a rat's femur, which was again provided by the Cornell course.

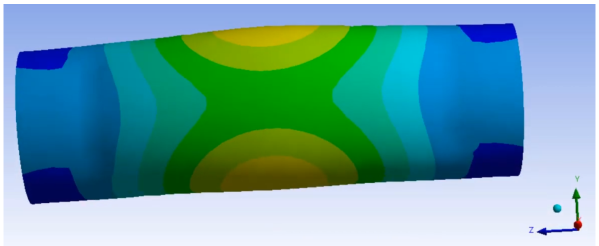


The problem specification consisted of a femur with supports 15 mm apart, with a central displacement of 0.3 mm downwards. The femur has Young's Modulus of 5 GPa, and Poisson of 0.4. The bone is hollow and has a radius of 1.9 mm and 0.7 mm thickness.

This exploration was much more involved and tedious, but made tons of sense once I got the hang of it. I made a note of the general work flow, and the steps I took according to the specification. The supports were 'contact patches' of equal area that were projected onto the bone from a planes below and above it. This also showed the modular nature of Ansys Workbench in being able to connect 'nodes' or 'objects' as inputs or outputs to different sims, allowing for complexity in highly coupled simulations.



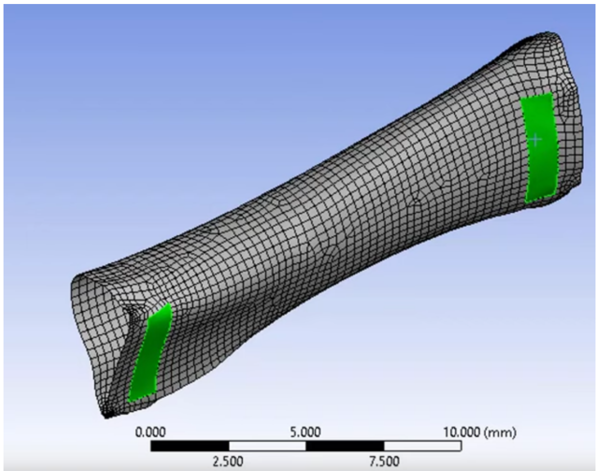
Side view of simplified cylinder model.



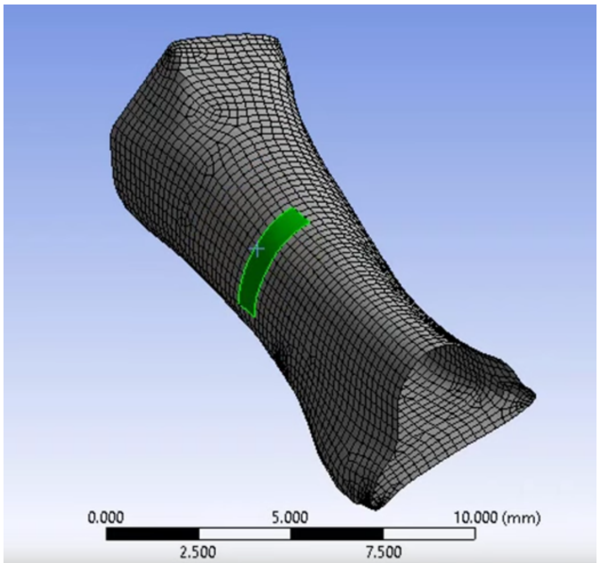
Bottom view of simplified cylinder bone model, confirming that our boundary conditions were respected.

Verification and validation was also more rigorous. There is the standard check of making sure that deformations happen only within the area around the displacement (and not beyond the supports), and making sure that the minimum and maximum displacements were close to 0 and 3.0 mm respectively. The most interesting by far was from following a video of a TA from the course constructing a very basic cylinder bone model (see resources), and checking if it behaves similarly under the same conditions.

For a long time I was running into crashing and geometry issues (being too big, too many faces, etc.) and it ended up being due to the physical scale of the model you are importing. It was set to m scale instead of mm, causing all sorts of issues and my answers blowing up, but this ended up being an easy fix and a common beginner mistake.



Bottom view of Femur. Contact patches (green) locations p_0, p_1 (left/right)



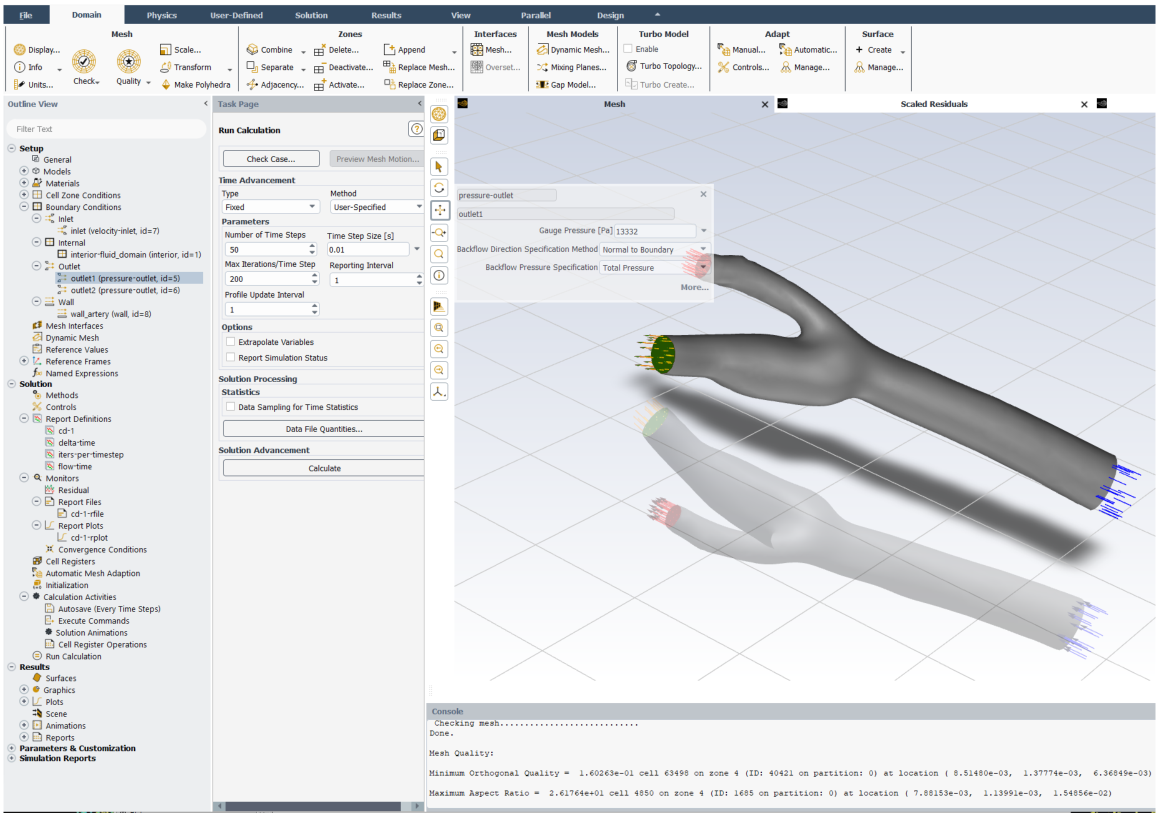
Top view of Femur. Contact patch (green) location p_2

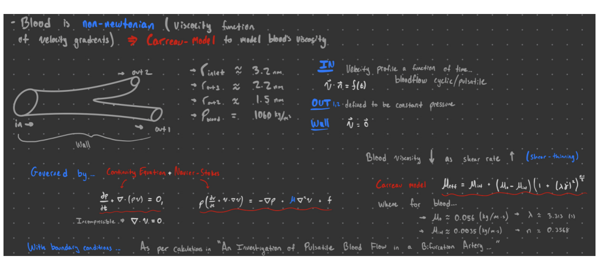
When constructing the simpler cylindrical model, an important rule of thumb that was emphasized by the TA was that our thickness of our geometry should be at the minimum two elements thick, so that there is at least some behavior to be captured. This heat map model is something that I may find useful when trying to physically inform real-time simulations towards the end of the semester. I am wondering if there is some easy way to get that sort of data in a format where I can map colored regions to weights on an arbitrary mesh given that the shape is the same. Not too sure how that may help specifically yet, but it is something that I am thinking about.

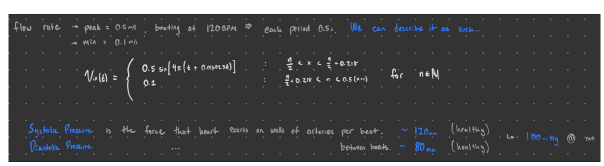
Blood and Arteries

→ Geometry file (again provided by Ansys and Cornell) can be found in the git repository.

→ There are two output video animations that can be found in 'videos' in the git repository.







This last mini-project ended up being the most rigorous and time consuming (both computationally and effort wise), and was my first dip into proper CFD and that specific workflow. This simulates the flow of blood through a static/non-adaptive, non-breaking artery (although adaptive, breaking artery is a goal to aim for now). We can visualize the shear on the walls of the arteries as well as velocities, allowing us to trace paths through the artery as visualization of flow. This was done by setting up a function for velocity at the inlet, simulating a heart beat with the piece-wise function found on the right above. The pressures at the outlets were kept at a normal, constant value for the human body. The dynamics of blood was something that I did not expect to be this deep. Blood flow is laminar for the most part, but under conditions of high flow, laminar flow can be disrupted and become turbulent, as we will see in one of the animations in the repository. This non-Newtonian behavior is captured through the Carreau-model, but I am currently in the middle of trying to get the Casson Fluid model implemented, but this requires playing around with user-defined functions, which I still need to look into a bit deeper.

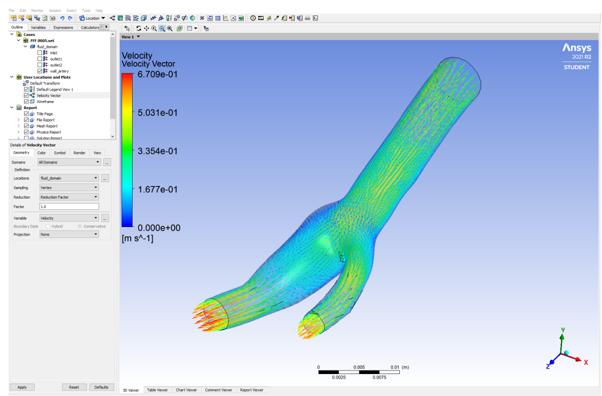
Our results consisted of velocity visualization, visualization of shear on walls (although the wall shear animation was not coherent as I did not have enough timesteps when I captured it), and path tracing of two separate particles.

- Verification / Validation ?
  - Mass conservation (net mass flow rate close to 0)
  - inlet / outlet boundary conditions (inlet matches $V_{in}(t)$ in sim, outlet matches constants)
  - meta-parameters to sim / Mesh-refinement related
  - compare to experimental data (not sure how one would do this here)
- Improvement / Iteration ?
  - Elastic walls (couple FEA and CFD) → next step?
  - Casson Fluid Model for non-newtonian (user-defined function)
  - Breakages ? What would that take?
    - Would that require remeshing + mesh swapping on command ?

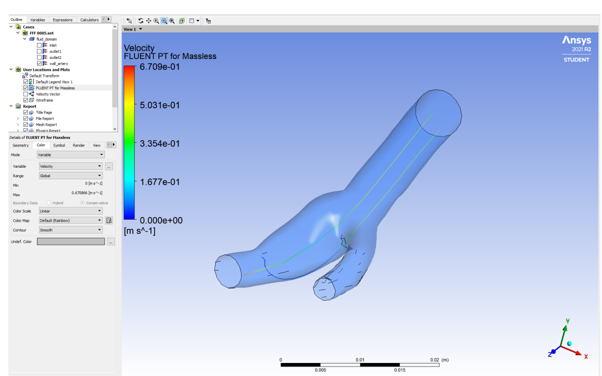
w/ $\mu_\infty = \sqrt{0.625\,Hct}$, $N_\infty = \sqrt{\mu_p (1-Hct)^{-0.25}}$ for blood $Hct = 0.4$, $\mu_p = 0.00145\,[Pa \cdot s]$

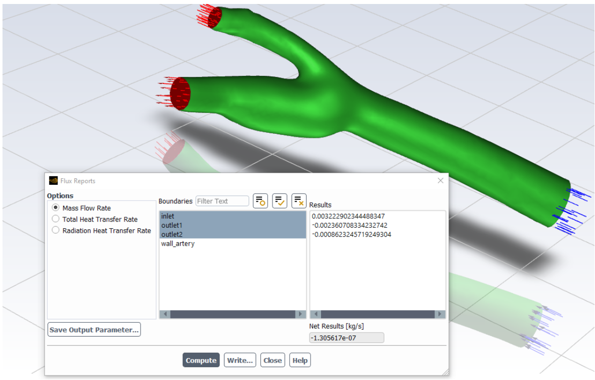
$$\mu = \frac{\mu_\infty^2}{\dot{\gamma}} + \frac{2\mu_\infty N_\infty}{\sqrt{\dot{\gamma}}} + N_\infty^2$$

Casson Model



Verification/validation was also the most rigorous so far, perhaps just due to this being a more complicated fluid simulation, rather than a structural one. Firstly, it consisted of checking our inlet/outlet boundary conditions, which only required setting up monitors to check velocities at those locations during the course of the simulation. Second had to do with mass conservation, checking that the net flux is zero as it should be.





Lastly, we need to do a check relating to mesh-refinement and meta-parameters to the simulation. One can do this by performing mesh refinement and use a smaller time-step to check whether the results are consistent with the original calculation. By using a finer mesh and a smaller time-step, we can also investigate the effects of truncation error caused by spatial discretization and temporal discretization. Surprisingly enough, results were too similar to tell apart, which is good news.

The next step for this is implementing elastic walls, and getting a dynamic heat map animation of that.

Other topics and thoughts

Just as I have done for bones and blood above, I plan complete further mini-projects relating Volumetric and Deviatoric Behavior (for plasticity of skin and flesh), Hyperelasticity (muscle), and Structural Instability (for its broad applicability). This is extremely important to take the deepest dive possible, as I want to get into causing 'breakages' or 'bursting' in simulations that are visually convincing. Materials might also be worth taking a look at if I see that I have enough time to do so, and if it is interesting enough to contribute to the mega-model.

Pertaining to physically informing a real-time simulation, the heatmaps of displacement (or other related measurements) may be a good way to 'bake' physical information. I'm imagining some system where you can paint geometry with colors corresponding to heat maps to give tetrahedral nodes more or less structural strength, allowing them to deform more. However, before I start exploring this, I feel like I still need to couple blood with artery wall displacement (to get some visual jiggling effect from the heart beat rhythm), as well as get some ground of tangibly constructing. My best idea for this so far is a brain-skull-skin coupled simulation, that involves as much coupling and detail that I can handle once I have placidity and hyperelasticity under my belt.