Developing a preprocessing pipeline to enable contact analysis in a novel finite element analysis solver

Elisha Gretton, Supervised by: Frederik Trommer

INSIGNEO
Institute for *in silico* Medicine

Background

Finite element analysis is a proven tool for the mechanical analysis of bones. Due to the ability to predict stresses and strains within the bone it can help predict the risk of bone fractures. Voxel meshes and corresponding FE solvers, such as ParOSol, are widely used due to the simplicity of obtaining them from computed tomography scans and their computational advantages. However, simulating contact has proved difficult due to the jagged surface of the bone geometry from the discretisation onto a cartesian grid resulting in inaccurate surface stresses. Nevertheless, contact stresses play an important role in the prediction of fractures in joints such as the wrist joint. To remedy this, a new formulation was published by Bhattacharya et al. [1] called the “Simulated smooth surface” formulation (SS-SC). This formulation was recently implemented in the FE solver ParOSol [2]. However, there exists currently no convenient method to automatically create the input files. This project intends to create code which will automatically generate the ParOSol input files from any given voxel mesh.

Methods

Code was developed in Python in order to write the ParOSol input files [3]. The boundary conditions are extracted from the image data and the files are saved in the hdf5 file format. A virtual machine running linux was set up on which ParOSol was built in order to run the created files. The results were then visualized and analysed using the software ParaView. The code was tested on a simple geometry seen in Figure 4. The body was simulated as linear elastic with a Poisson's ratio of 0.33. The boundary conditions were chosen so that the bottom nodes were fixed in all directions, and the nodes with the largest values in the y dimension were displaced in -z direction by 0.2 times the voxel size.

Results

The files generated by the code were run by ParOSol without errors and produced the desired results. Figure 4 plots the von Mises stresses in MPa.

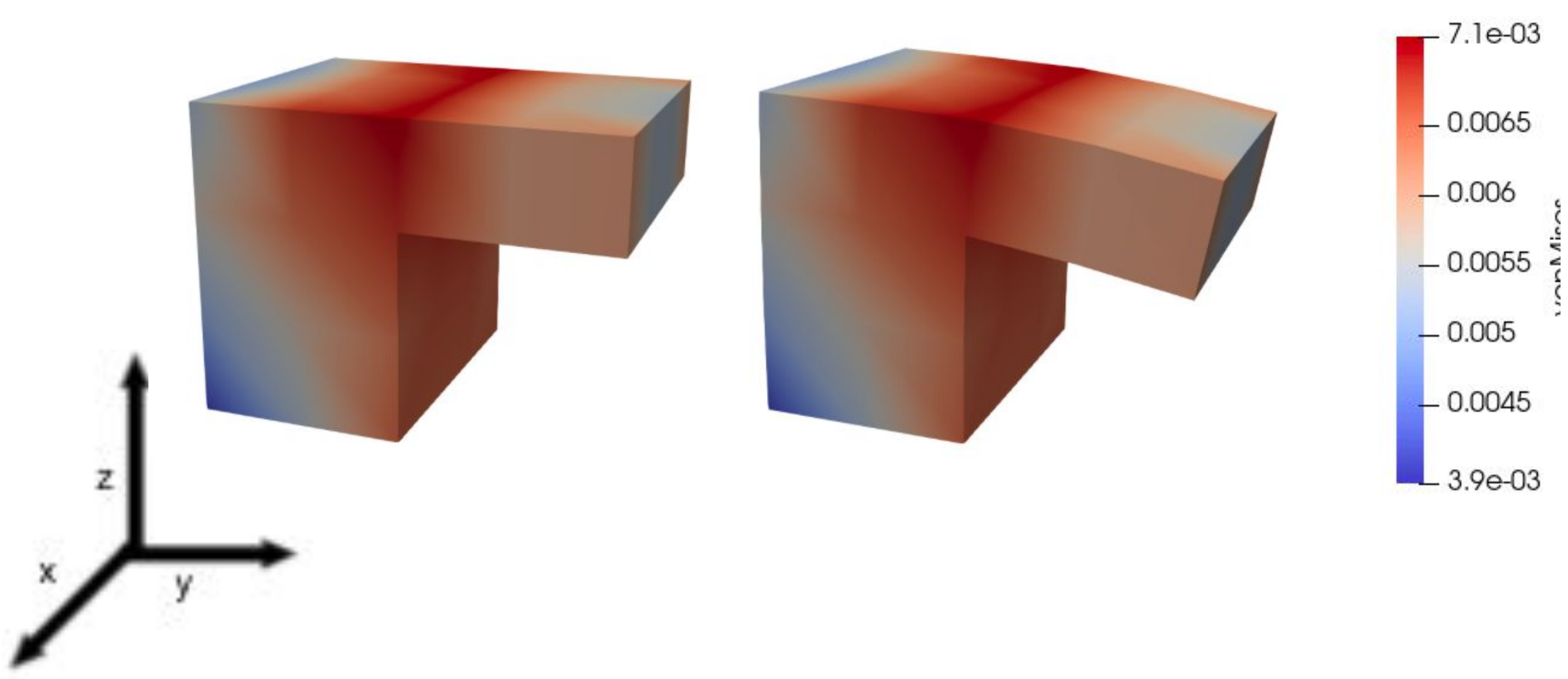

Figure 4: von Mises stress contours on a 3D voxel mesh model in undeformed (left) and deformed (right) configurations, plotted using visualisation software ParaView

References

[1] P. Bhattacharya, D. Betts, and G. H. van Lenthe, “A novel contact interaction formulation for voxelbased micro-finite-element models of bone,” *International journal for numerical methods in engineering*, vol. 115, no. 4, pp. 411–426, 2018.

[2] C. Flaig, "A highly scalable memory efficient multigrid solver for μ-finite element analyses". PhD thesis, ETH Zürich, 2012.

[3] F. Trommer, and P. Bhattacharya. "A penalty contact implementation on a highly parallelisable Cartesian mesh finite element solver", In Congress of the European Society of Biomechanics, Milan, July 2021.

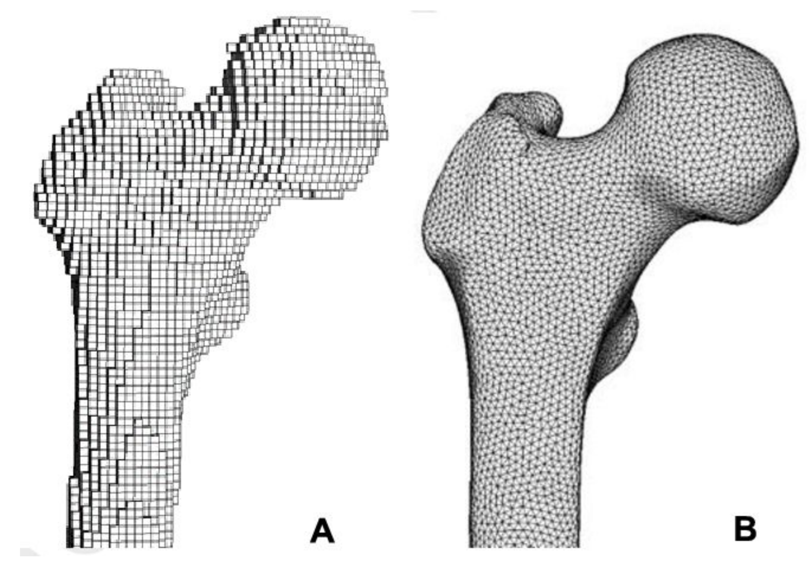

Figure 1: the voxel mesh of a bone at different refinements

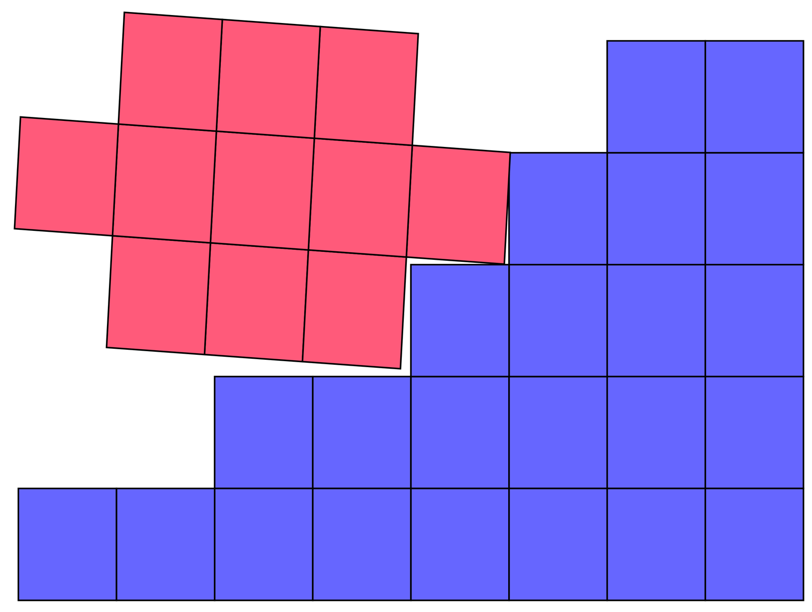

Figure 2: 2D diagram of two disjoint, voxelized bodies sliding over one another

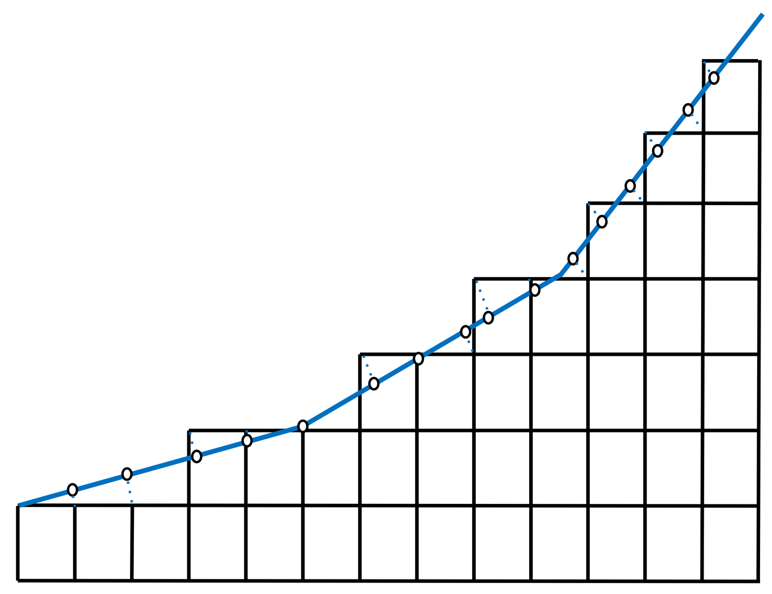

Figure 3: 2D diagram of SS-SC, simulating a smooth surface on one body

Discussion

The von Mises stress contours appear reasonable, thereby providing confidence in the preprocessing pipeline. The next step of the project would be to expand the abilities of the code to implement the data necessary for modelling contact. During the project, I have developed a deeper understanding of the Linux operating system by working on a Linux-based high performance computing cluster and on a Linux virtual machine that I installed. I have also learned how to use GitHub more efficiently and how it can be used for version control. It was interesting to learn about how mathematical techniques can be applied to biomechanical concepts, and being able to implement this using software.