Bespoke bone modelling with VOX-FE

Neelofer Banglawala(1),*

Iain Bethune(1), Richard Holbrey(2), Michael J. Fagan(2)

(1) EPCC, The University of Edinburgh, (2) The University of Hull

Modelling bones: why and how?

Bone is in a continual state of flux through ossification (growth) and resorption (loss). Bone loss begins at age 35, with > 30% bone lost by age 70. Bone has a complex geometry and is made up of different materials e.g. hard outer cortical shell, spongey inner trabeculae and soft marrow.

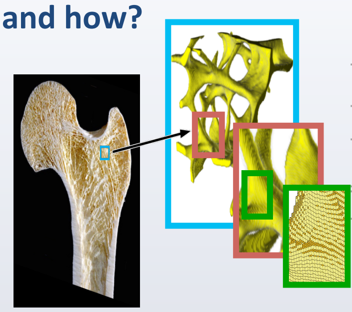


Figure 1. Bone model made up of small 3d finite elements (voxels).

How does bone respond to forces and constraints e.g. an implant? How does bone formation respond to stress and strain? We can explore these questions with *in silico* experiments on bone models. The complex geometry of bone can be modelled as consisting of millions of small cubic elements known as voxels.

VOX-FE : legacy solver

Developed jointly by Hull [1] and EPCC, VOX-FE [2] is a voxel-based finite element software suite (solver & GUI) for the analysis of bone models. The solver outputs the final voxel displacements of a bone model subject to linear forces/ constraints. Written in C++/MPI, earlier versions of VOX-FE could:

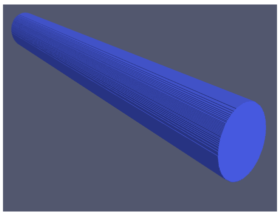


Figure 2. Dense cylinder "bone" model, used for comparing analytical solutions to simulation results.

- Manipulate/solve bone models of ~20x10^6 elements, max 4 material types
- Parallelise models along 1 dimension
- Scale up to 256 cores on ARCHER [3]

But realistic bone modelling requires more complex models with several hundred million voxel elements with an arbitrary number of materials (e.g. surrounding tissue). Exploiting HPC resources becomes crucial.

Solver Development

To exploit large-scale HPC resources such as ARCHER, bone models need to be parallelised along all dimensions, and the solver must scale well to thousands of cores. Through two ARCHER Embedded CSE funded development projects, we replaced the legacy solver with an entirely new design that utilises the powerful functionality of the PETSc library [4]. This has the added benefit of expanding the range of solution algorithms available to the solver. We also improved the load balancing capabilities of VOX-FE by using the fast, parallel graph partitioning library ParMETIS [5]. The solver developments are summarised below:

Stage I – solver redesign using PETSc library

- New solver interface with PETSc functionality
- Redesign representation of bone model to support large, complex models with an arbitrary number of materials
- New and improved interface between solver and GUI

Stage II – load balancing using ParMETIS

- Representing the bone model as a graph of elements, ParMETIS then finds the optimal partition of the bone model across processes

Performance Improvements

The solver development has resulted in the latest release, VOX-FE3, which can manipulate and solve models of hundreds of millions of elements with an arbitrary number of materials. Furthermore, the solver scales well to thousands of cores, as shown in Figure 3.

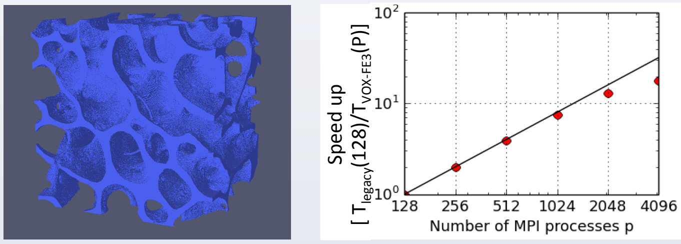
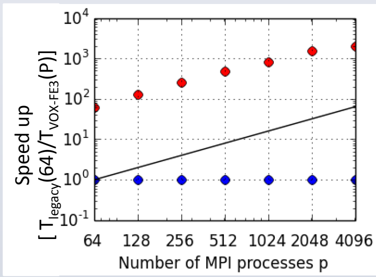


Figure 3. Right: Horse head trabeculae model (200x10^6 elements). Left: Strong scaling speedup of VOX-FE3 solver using horse head model. Solid line is ideal speedup.



Better load balancing has resulted in less memory usage per process, improving the overall performance of the solver compared with the legacy solver

Figure 4. Strong scaling speedup of model loading times: legacy solver (blue), VOX-FE3 solver (red), ideal speedup (black).

GUI Development

A new ParaView [6] based GUI has replaced the legacy GUI, which was written in Borland C++ and was tied to the Windows OS. The new GUI can manipulate larger models and has additional features, such as being able to account for the effect of muscle tissue on bone.

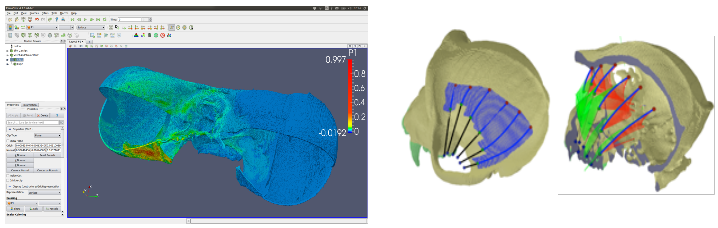


Figure 5. Left: screenshot of new ParaView GUI, visualising a dragonfly head. Right: example muscle wrapping implementation [7].

Further development: bone remodelling

Bone remodelling (growth/loss) can now be investigated by setting a threshold stress above/below which bone is added/lost when subject to differential stress and strain.

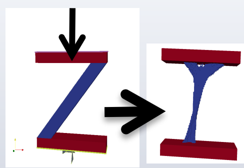


Figure 6. 'Z' bone model deforms into 'T' bone model when compressed from above.

References

[1] Biological & Medical Engineering Group, University of Hull

[2] VOX-FE is open source software: https://sourceforge.net/projects/vox-fe/

[3] ARCHER is the UK's National Supercomputing Service: http://www.archer.ac.uk

[4] PETSc: https://www.mcs.anl.gov/petsc/

[5] ParMETIS, Karypsis lab: http://www.karypsis.ac.uk

[6] ParaView: http://www.paraview.org/

[7] J Liu et al. 2012 Biomechanics and Modeling in Mechanobiology. 11:1 35-47.

* n.banglawala@epcc.ed.ac.uk