Voxel-based finite element modelling with VOX-FE2

Richard Holbrey (University of Hull)
Neelofer Banglawala (EPCC)
Overview

Part I - Development of the VOX-FE2 plugin
  • Finite element analysis and biomechanics
  • Modelling voxel data
  • VOX-FE2 and ParaView

Part II - The new VOX-FE2 solver and PetSc

Part III - A scheme for remodelling
Part I

Development of the VOX-FE2 plugin for ParaView
In this section

i. Finite element (FE) analysis
   • Application in biomechanical engineering

ii. Voxel data
   • Turning images into FE models

iii. Development of VOX-FE2
   • A brief history
   • Development as a ParaView plugin
Finite Element Analysis (FEA)

- Well-proven and common place technique in engineering
- Becoming increasingly popular for the modelling of biological systems
- Used to calculate the deformation of any object when subjected to loads and to predict:
  - Resultant stress and strain distributions
  - Reaction forces
FE in solid mechanics

FE treats the model as a series of much smaller elements

- For each element, we compute the distortion
  - Stretching (or squashing)
  - Shear
- The body is assumed to be elastic
  - For very small displacements
FEA in biomechanics

Bone is an obvious target for elastic modelling; but some form of simplification is usually required if current commercial software (eg. ANSYS or ABAQUS) is going to be used for analysis.

- During that simplification, fine structures, especially trabeculae, may be lost.
- Geometries are normally ‘homogenized’
  - Elastic moduli are averaged over a small region, say
  - But, what happens near edges?
- Voxel-based modelling offers an alternative, multi-scale approach.
Voxel-based FEA

- Millions of voxels/elements needed to model small sections
How to set the resolution?

- A common metric is Trabecular Thickness (Tb.Th)
  - Largest diameter sphere contained within the model
  - Varies from species to species
- Some typical values for Tb.Th
  - 430µm Elephant
  - 134µm Human adult (F)
  - 20-60 µm Rodent
- For FE: working hypothesis
  - ≥ 5 voxels thick - 10 better!
  - cf Guldberg et al J Bio Eng 1998(120)

Source: BoneJ.org
Example: MicroCT data (rodent)

Besides resolution, for bone studies, we have to consider:

- Nature of the problem
- Noise removal
- Image processing & segmentation

Source: Bouxhein et al. J Bone Min Res 25(7): 1471
Background to VOX-FE

Bone exists in a continual state of change

- Growth peaks in young adult life and after 35 years, begins to decrease
- At 70 years of age, more than 30% of the bone mass is lost
- Bone mass and structural morphology are affected by lack of exercise and disease

In 2003, Sisias established a framework for the investigation of remodelling with respect to:-

- Natural growth
- Metabolic disease (osteoporosis)
- External stimuli
The VOX-FE GUI

In 2009, in conjunction with Hull-York Medical School, a GUI was added

- Developed the script file format, for use with external solver
- Allowed display of boundary conditions, strains etc
- Optimised for voxel data

- Used Borland/Embarcadero C++ Builder (Windows)
  - Powerful RAD tool, but in retrospect, an unfortunate choice
  - No 64-bit version until recently (limiting to ~20M elements)
  - Relied upon graphics component developed by user community
ParaView

- Open source, cross-platform, 32 or 64-bit architecture
- Library of visualization tools, based upon Kitware’s Visualization Toolkit (VTK)
  - A standard display model
  - Multiple views
  - Many filters already available
  - Undo/redo stack
- Well-documented pipeline-based interface
  - Tutorials, videos etc
ParaView/VOX-FE2 (example session)
Vox-FE2 plugin

Currently, the plugin has 6 main functions:-

Pre-processing

• Import/clean image data to create model (uses ITK)
• Group extraction (for ‘grouped’ data eg bone)
• Specify boundary conditions
• Display glyphs to illustrate boundary conditions
• Option to selectively export boundary conditions to a constraints file

Post-processing

• Display of displacements, strains and stress
Glyphs/Level of detail operator
Colour-mapping strains
Clipping
Adding displacement field
Summary

Now have much more stable, capable GUI

- Model exists as a standard VTK object
- ParaView incorporates many VTK filters
- Restricts the use of filters to the correct type for that object
  - Contouring, clipping, gradients, histograms etc
- Multiple views, spreadsheets, 1-3D & animation
- 100M elements of desktop (32 GB RAM)
- VOX-FE2 plugin builds on Archer
- Expect to be able to using RSIP
  - http://www.archer.ac.uk/documentation/white-papers/rsip/ARCHER_wp_RSIP.pdf
Future work

• Features
  • Muscle-wrapping
  • Property definition based on grey scale
  • Examples & tutorials

• Modelling
  • MicroCT
  • Synchrotron data
Part II

The new VOX-FE2 solver
Solver: overview

- What the solver does
- Before VOX-FE2
- PETSc-based redesign of solver
- Performance results
- Summary
Solver: what does it do?

- Iteratively solves linear elastic problem

\[ A \mathbf{x} = \mathbf{f} \]

\( A \) Coefficient matrix
\( \mathbf{x} \) Displacement vector
\( \mathbf{f} \) Force vector

Solve using standard algorithms e.g. Conjugate Gradient Method

- **Challenge**: how to store and manipulate large matrices?

- For 10M element model, \( A \) has \( \sim O(10^{13}) \) entries of which \( \sim 0.0003\% \) are non-zero... \( A \) is **sparse**

- Find smart ways to store, manipulate non-zero values only
Solver: before VOX-FE2

Optimised, parallel, linear solver written in C++, MPI

- Models up to 3 materials
- MPI partitioning along z-axis
- Solves up to 20M elements
- Scales well up to 256 cores

- Developer’s $10^6$ question: extend code or start from scratch?
- Start from scratch because...
  - Difficult to implement all the above in old solver
  - There exist libraries that specialise in solving linear systems

MORE MATERIALS
PARALLELISE ALL DIMENSIONS
LARGER MODELS
MORE CORES, BETTER SCALING
Solver : PETSc-based redesign

• PETSc* is a library of data structures and routines for the scalable (parallel) solution of scientific applications
  *Argonne National Laboratory

• A PETSc-based solver has the potential to offer
  • Capability of running large models (~100M+ elements) efficiently with good parallel scaling along all dimensions
  • Greater choice of routines for solving the system, using different combinations of routines to get the best results
  • Fine tune PETSc to improve speed and scalability
  • Interface with ParMETIS for matrix/graph partitioning (remodelling)
  • Flexibility to run on heterogeneous architectures e.g. with GPUs
Solver: performance results

Strong scaling of VOX-FE solvers (≈10M element model)

Time to solve (≈10M element model)

- VOX-FE2 solver
- Old solver
Solver: summary

In replacing old solver with an entirely new PETSc-based design, VOX-FE2 now has improved

- Scalability, speed, flexibility, extensibility, multi-material capability...

Future work

- Improve parallel decomposition using ParMETIS (currently solver splits matrix evenly between processors)

VOX-FE2 will be available on ARCHER and on Sourceforge ([http://sourceforge.net/projects/vox-fe/](http://sourceforge.net/projects/vox-fe/)) in the next few days!
Part III

VOX-FE2:
A scheme for remodelling
Some concepts

- As observed earlier, bone:-
  - Exists in a state of flux resorption/growth
  - Age, disease and stress all have influence

- At this stage
  - Proving the concept
  - Naïve, can our models adapt to stress?
  - How can HPC help best?
Components needed

- Solver
  - Reads VOX-FE model data
  - Standalone (computes displacements)
- Organizing utility
  - Maintains a 'graph' of element connectivity
  - Computes strain data to add/delete elements
- Some form of 'glue' code
Input(1): VOX-FE script / model / materials / constraints

Input(2): Upper/Lower threshold

Convert to Metis (and, optionally, vtk) format

Make element graph

Remodeller

Find surface voxels

Compute SED

Solv...
Remodel/graph component

Reads in connectivity data from Metis

- Installed on Archer
  - Creates a very compact, but fixed graph
- VOX-FE Graph
  - Any voxel can only have 26 neighbours
  - Create bits for each neighbour (32-bit integer)
  - Easy search for surface voxels
  - 2-pass algorithm to remove disconnected elements (‘islands’)

archer
University of Hull
epcc
Glue code

Working in batch mode:-

- Reuse GUI reader as offline utility,
  - convertVoxFEScript
  - m2gmetis
- Remodelling/graph component
  - Reads metis data and displacements
  - Reuse GUI strain component
Scripts

Run remodelling from 2 scripts

• Job submission launches ‘remodel’ script
  • Waits for solve to complete
• Once solver finished, waits for remodeling
• Currently, a new script/model file is created
• Process repeats until time or steps done
Choice of thresholds

- Thresholds entered manually into job submission scripts
- Can be aided by first solve & histogram plot in ParaView
Example 1: ‘Z’ model
‘Z’ model → ‘I’
Example 2: Biomedtownd ‘cube’

- Loaded evenly across top
- Based fixed
- Top/base cannot remodel
Remodelling in action (0)
Remodelling in action (1)
Remodelling in action (2)
Remodelling in action (3)
Remodelling in action (4)
Remodelling in action (5)
Remodelling in action (6)
Remodelling in action (.)
HPC

- 2M element trabecular model
  - solve 280 – 370s (16 cores, wallclock)
  - remodel < 10s
- Choice of thresholds problem
- Would prefer to solve in each dimension
- Larger models
Future work

- Adaptive remodelling
  - Statistical selected thresholds
- Graph partitioning scheme
  - (Par)Metis
Youtube video links:

Remodelling:
  VOXFE:  http://youtu.be/fjtjrM1Z1JQ
  Biomedtown:  http://youtu.be/WVp1u1jID3g
  ‘O’ model:  http://youtu.be/nT4xW1cm7aQ

Tutorial:  http://youtu.be/DDeAxaZnE8U