A PYTHON PACKAGE TO ASSIGN MATERIAL PROPERTIES OF BONE TO FINITE ELEMENT MODELS FROM WITHIN ABAQUS SOFTWARE

EC Pegg, HS Gill

Centre for Orthopaedic Biomechanics, Dept. of Mechanical Engineering, University of Bath, UK.

centre for orthopaedic biomechanics

Introduction

Bonemat software is a great tool which can be used to assign the material properties of bone to a finite element model using data from a CT scan to enable more accurate simulations. Recent improvements in Bonemat have enabled Abaqus input files as well as Ansys files to be processed. However, Bonemat still needs to be run separately from Abaqus and some information contained in the input file (such as element sets or contact definitions) is lost after material assignment.

We introduce a Python package which has been written to interface directly with Abaqus software enabling all pre-processing, material assignment, solving, and post-processing to be fully automated. This improvement in workflow is particularly useful when running multiple models parametrically.

Materials & Methods

The objectives of this study was to:

(a) check equivalence between the results of Bonemat 3.2 (current version) and the py_bonemat_abaqus script

(b) compare processing speeds

Equivalence with Bonemat

The software packages were compared using a CT scan of a half pelvis downloaded from the VAKHUM database, and the associated hexahedral finite element mesh of the left half pelvis. To examine different element types, the hexahedral mesh was converted to linear and quadratic tetrahedral elements by dividing each hexahedron into 5 tetrahedral elements. The equations used to convert the Hounsfield Unit (HU) values to apparent density (papp) and to convert the apparent density to elastic modulus (E) were based on published work [1] and are shown in the following equations.

$$p_{app} = -0.021075 + 0.000786HU$$

$$E = 2.0173\ p_{app}^{2.46}$$

Comparison of processing speed

The time taken to analyse the models by each software was measured using a Windows 7 PC with a 64-bit operating system, 4 CPUS, 8 GB of RAM and an Intel Core i5-3470 processor.

Results

The Python package py_bonemat_abaqus took a similar time to run for all element types; this was between 109 and 126 s. Bonemat 3.2 software was considerably faster, and took between 5 and 20 s (Fig 1).

The mean difference in modulus assignment made by py_bonemat_abaqus and Bonemat 3.2 was -0.05 kPa (range -10.19 to 4.5 kPa, standard deviation 0.62 kPa) (Fig 2).

An error was noticed in the linear tetrahedron jacobian calculation in the old version of Bonemat (3.0) which resulted in up to 2 GPa difference in modulus. We therefore strongly recommend that users update their version of Bonemat to the newest release (www.bonemat.org).

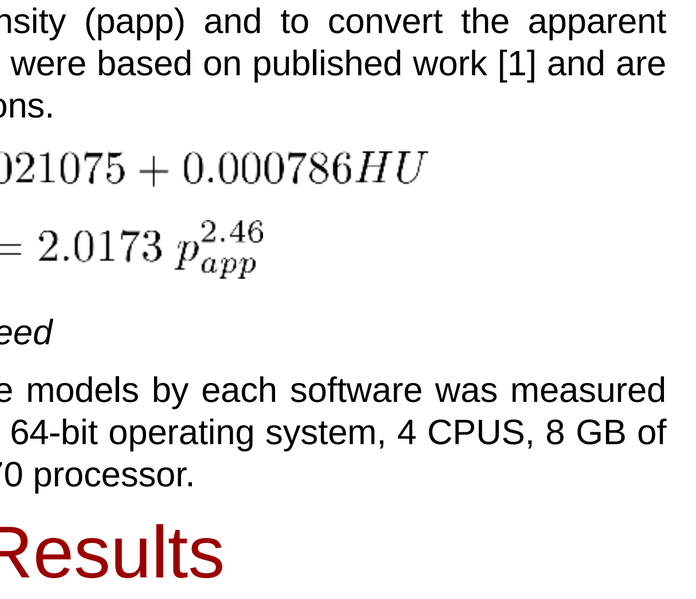

Fig 1: Speed comparison between the software for different element types.

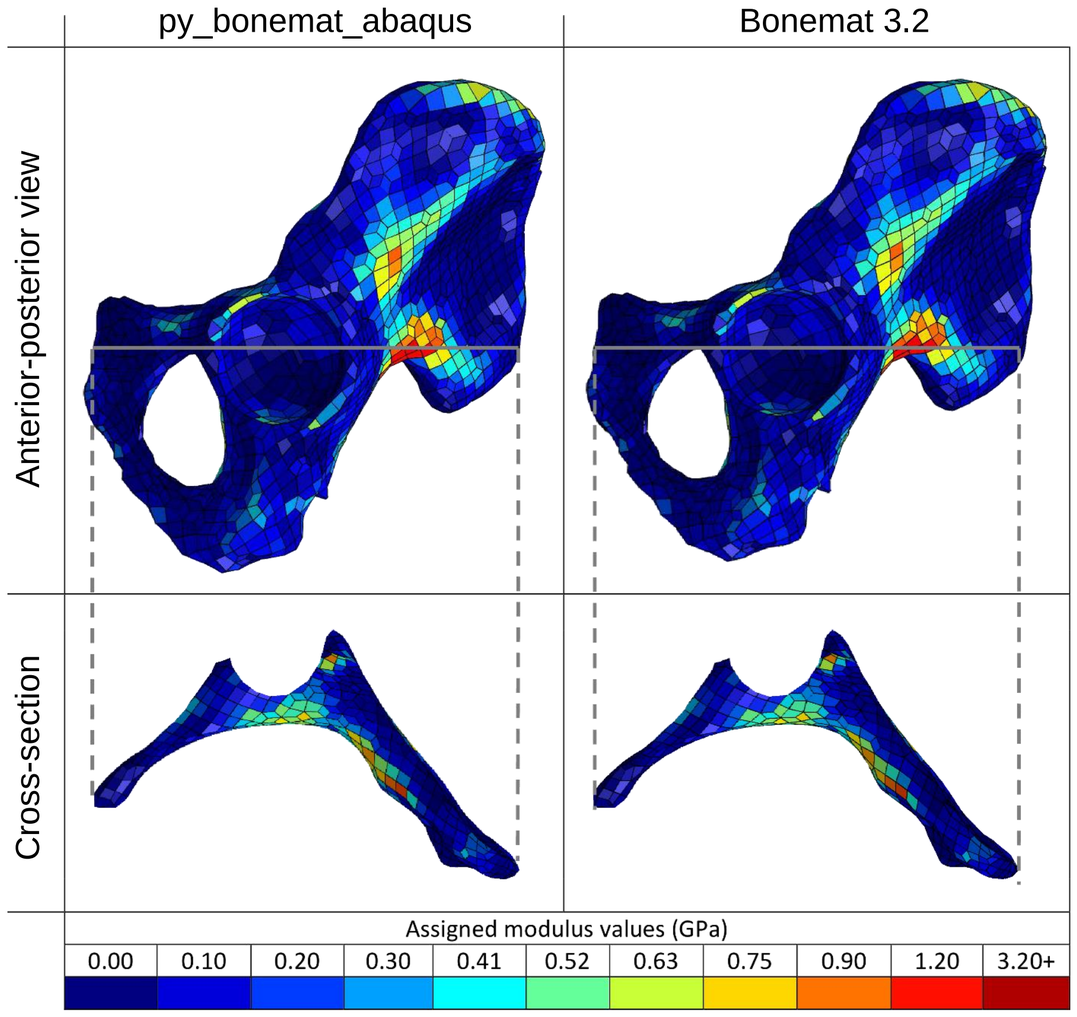

Fig 2: Assigned modulus values from py_bonemat_abaqus and Bonemat 3.2 (V3 algorithm) for the hex mesh of the hemi-pelvis.

```
# import py_bonemat_abaqus module
from py_bonemat_abaqus import run
# create input file of a cube
myModel = mdb.models['Model']
mySquare = myModel.ConstrainedSketch(name='square',
                                     sheetSize=200.0)
mySquare.rectangle(point1=0,0, point2=30,30)
myPart = myModel.Part(dimensionality=THREE_D,
                      name='Cube',
                      type=DEFORMABLE_BODY)
myPart.BaseSolidExtrude(depth=30.0, sketch=mySquare)
myModel.rootAssembly.Instance(dependent=ON,
                              name = 'Cube-1'
                              part = myPart)
myModel.StaticStep(name='Step-1', previous='Initial')
myPart.seedPart(deviationFactor = 0.1,
                minSizeFactor = 0.1, size = 3.0)
myJob = mdb.Job(name='Cube.inp',
                model='Model', description='')
myJob.writeInput()
# use py_bonemat_abaqus to assign materials
run('ParametersFile.txt', 'CubeCT.dcm', 'Cube.inp')
# solve model
mdb.ModelFromInputFile(inputFileName ='Cube_MAT.inp',
                       name = 'Cube_MAT)
myJob = mdb.Job(name = 'Cube_MAT.inp',
                model='Cube_MAT', description = '')
myJob.submit(consistencyChecking=OFF)
# ... go on to post-process results
```

Fig 3: Example usage of py_bonemat_abaqus

Discussion

Material assignments were almost equivalent between the two software packages, with any differences explainable by rounding effects. To put the differences into context, a difference of -0.05 kPa is 0.00000002% of the typical modulus of cortical bone (20.7 Gpa), and 0.00000003% of the modulus of trabecular bone (14.8 GPa) [3]. The Python package was slower to process the models, but was successfully able to assign material properties from within Abaqus software as part of an automated script (Fig 3).

The Python script can be downloaded for free from
`https://pypi.python.org/pypi/py_bonemat_abaqus`

Acknowledgements

We would like to acknowledge the Laboratory of Human Anatomy and Embryology, University of Brussels (ULB) for supplying the CT scan and finite element model from the VAKHUM database. The study was unfunded.

Corresponding Author: e.c.pegg@bath.ac.uk

Department of Mechanical Engineering, University of Bath, Bath, BA2 7AY, UK

UNIVERSITY OF BATH

EORS 2016
ISTITUTO ORTOPEDICO RIZZOLI
14-16 SEPTEMBER 2016