AN EFFICIENT SOLVER FOR LARGE-SCALE SIMULATIONS OF VOXEL-BASED STRUCTURES USING A NONLINEAR DAMAGE MATERIAL MODEL

Monika Stipsitz[1] and Dieter H. Pahr[1,2]

[1] Institute of Lightweight Design and Structural Biomechanics, TU Wien,
Getreidemarkt 9, 1060 Vienna, Austria. https://www.ilsb.tuwien.ac.at
stipsitz@ilsb.tuwien.ac.at

[2] Division Biomechanics, Karl Landsteiner University,
Dr.-Karl-Dorrek-Straße 30, 3500 Krems an der Donau, Austria. https://www.kl.ac.at
dieter.pahr@kl.ac.at, pahr@ilsb.tuwien.ac.at

Key words: Bone biomechanics, High Performance Computing (HPC), Micro Finite Element, Nonlinear Material Model, Iterative Equation Solver

Abstract. Understanding the failure mechanisms of highly heterogeneous materials requires a detailed knowledge of the micro-structure and the underlying material properties. For bone, this level of information can be obtained from high-resolution computed tomography (CT). In finite element analysis (FEA), the amount of structural information combined with nonlinear material models, calls for highly efficient and parallel software tools. In order to provide such a solver, ParOSol, an existing high-performance linear solver, was extended to incorporate a simple damaged based material model. The FEA solution is found by iteratively applying the linear solver. The new solver was tested on several structures: A simple cuboid for numerical verification, a trabecular bone cube for optimizing the solver parameters and a radius segment (688 million degrees of freedom) for performance evaluation.

1 INTRODUCTION

The global behaviour of heterogeneous materials strongly depends on its micro structure. Therefore, simulations are based on detailed knowledge of the underlying architecture. Especially in case of a nonlinear material behaviour, immense computational power and efficient numerical tools are required.

One field of application for such studies is bone biomechanics. Trabecular bone is a highly hierarchical material. It consists of fine substructers ($\sim 0.1\,\mathrm{mm}$ diameter) which can be resolved by high-resolution µCT imaging devices. Simulations of macro-sized bone segments are necessary to better understand failure mechanisms. This leads to two different approaches for investigating such problems: homogeneous FE models using spatial averaging or µFE models using the full resolution µCT scan. The first approach requires

only low computational resources. However, the material model needs to account for the loss of structural information to provide meaningful results. The second approach requires much less assumptions and simpler material models. However, typical meshes have multiple hundred million to a couple of billion degrees of freedom (dof). Thus, simulations are computationally much more demanding and require efficient parallel solvers.

Many different implementations exist which solve FE problems. However, most of them account for linear material behaviour only or are limited to a couple of million (mio) elements [1, 2, 3]. To our knowledge, three solvers are able to simulate such large models with reasonable computational effort: ParFE [4], FEAP [5] and ParOSol [6, 7]. Originally only for linear problems, ParFE has already been adapted for nonlinear constitutive models by Christen et al. [8]. It scales well with the number of CPUs but shows convergence problems above about 100 mio elements. Adams et al. [9] have extended the serial code FEAP using a parallelisation layer. However, the full version is not publicly available. A similar version is also used by Nawathe et al. [10] for large scale simulations. Later on, ParOSol was developed by Flaig et al. [6]. It is a highly efficient code for linear elastic FE simulations with a low memory footprint. ParOSol is freely available under an open-source licence and can use thousands of CPU cores in parallel. Furthermore, it uses µCT images directly to generate the FE mesh.

To gain a deeper understanding of the structural behaviour of bone with little structural averaging, a highly-parallel and efficient finite element analysis (FEA) software with a nonlinear material model is needed. In this work, ParOSol is extended to the nonlinear range by adding a simple damage based material model. The new implementation, ParOSolNL, will make nonlinear simulations of large digitized structures (e.g. bones) at micro scale feasible, while keeping its numerical efficiency.

2 METHODS

2.1 Theoretical background

The starting point of a CT based FEA simulation is a micro structure which is divided into small elements – the finite elements. Since ParOSol uses direct voxel to element conversion, eight-noded hexahedrons are generated. Assuming small displacements and strains, this implies that the following linear system of equations is solved for the unknown nodal displacements U_j:

$$F_i = K_{ij}U_j. \tag{1}$$

The stiffness matrix K_{ij} is a measure of the structural stiffness and depends on the micro-structure and tissue properties (e.g. Young's modulus). The nodal forces F_i represent the force boundary conditions and are equal to zero for all internal nodes. In this work, geometric nonlinearities are neglected.

The global force is obtained as the sum of all nodal forces on the boundary planes. In a nonlinear damage based material simulation, the load is applied incrementally (i.e. ΔF_i) to obtain the load–displacement curves as well as the ultimate load which characterizes the global strength of the geometry. Additionally, stress, strain and other quantities can be computed for each element.

2.2 Material model

A simple material model is used which allows damage and fracture of elements. It consists of three regions: a linear-elastic, a damaged and a fractured region (Figure 1, left). The material is linear-elastic until the stress exceeds a critical value. This leads to a reduced Young's modulus E by introducing a scalar damage variable D:

$$E = (1 - D)E_0, \tag{2}$$

where E_0 is the initial tissue Young's modulus. If D is higher than the critical damage D_c, the material fails and E is set to zero. No plasticity is involved.

In one dimension, the critical tensile (σ_0^+) and compressive stress (σ_0^-) are scalars. In three dimensions, the critical stress is represented by a damage surface (Figure 1, right). A quadric damage surface as proposed in [11] is used. The shape of the surface is defined by the parameter ζ_0. If a stress state lies outside of the damage surface, D is determined by projecting the stress back onto the damage surface.
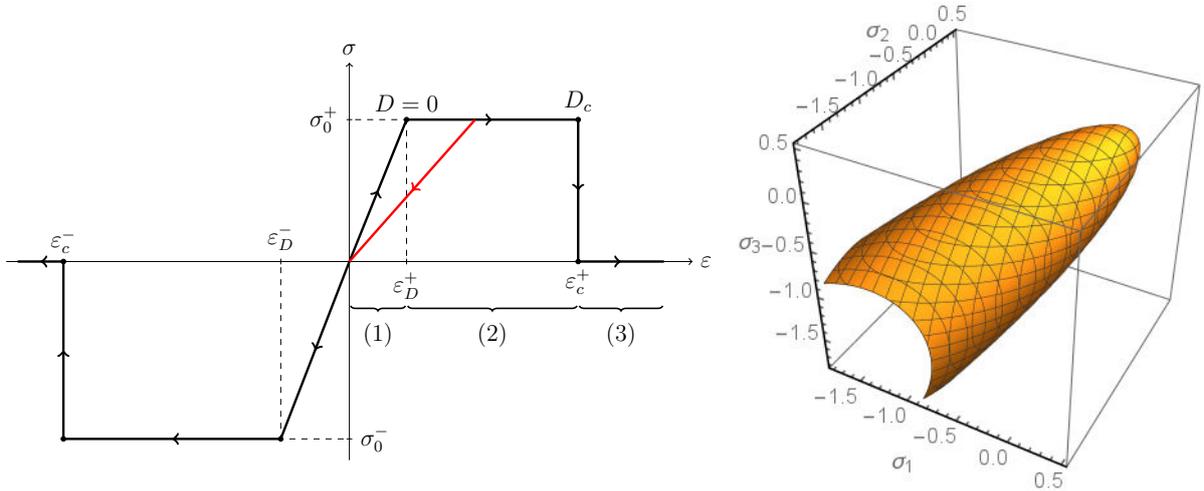
Figure 1: Left: Sketch of the one dimensional material model showing the linear-elastic (1), damaged (2) and fractured (3) region. Right: In three dimensions, the critical stress $\sigma_0^\pm$ is replaced by a damage surface.

The following bone tissue material parameters are used: initial isotropic Young's modulus $E_0 = 10\,\text{GPa}$ [12, 13], Poisson's ratio $\nu = 0.3$. The damage surface is given by the tissue yield stress in tension $\sigma_0^+ = 41\,\text{MPa}$ and compression $\sigma_0^- = 83\,\text{MPa}$ [14] and the shape parameter $\zeta_0 = 0.3$. The critical Damage is $D_c = 0.6$ [15].

2.3 Implementation

Since ParOSol can only solve linear-elastic problems, it is extended to nonlinear material behaviour by applying an incremental procedure (Figure 2). The solution of each load increment is found by repeatedly applying the linear solver of the original ParOSol (line 4). During each iteration, the Young's moduli of the structure are kept constant and the stress state of each element is obtained. Due to the applied local damage approach,

the damage state of an element depends only on the local stress. Thus, for each element, the distance to the damage surface $Y(\sigma_T)$ is calculated (line 6) and D is adapted (lines 8). In case of material failure, E is set to a small fraction of E_0 (line 9). This leads to a better better convergence compared to setting $E = 0$. After E of each element is changed according to its current stress state, the new linear problem is solved to ensure equilibrium. A converged solution for the increment is found when a global and local steady state is reached (line 12).

Data: Model
Result: Ultimate load
1 **while** *maximum load not reached* **do**
2   Apply increment of load;
3   **while** *not converged* **do**
4   Solve linear problem $F = KU$, get σ_{T};
5   **foreach** *element* **do**
6   Evaluate $Y(^{\langle e\rangle}\sigma_{\mathrm{T}})$;
7   **if** $Y(^{\langle e\rangle}\sigma_{\mathrm{T}}) > 0$ **then**
8   Increase Damage (back projection);
9   Reduce Young's modulus;
10   **end**
11   **end**
12   Check convergence;
13   **end**
14 **end**

Figure 2: Nonlinear solving procedure.

The local damage approach helps to minimize the interdependence of the parallel computing processes since the adaptation of E can be done independently. Communication between processes is only needed during the application of the linear solver and to check if a converged state is found.

Two convergence criteria are applied: The local change in damage of each element, $^{\langle e\rangle}R_1$, from iteration $(i-1)$ to iteration (i),

$$^{\langle e\rangle}R_1 = {^{\langle e\rangle}}D_{\mathrm{n}}^{(\mathrm{i})} - {^{\langle e\rangle}}D_{\mathrm{n}}^{(\mathrm{i}-1)} \leq 100\delta \tag{3}$$

and the average change in damage per element, R_2,

$$R_2 = \frac{\sum_{\text{all elements}} {^{\langle e\rangle}}R_1}{N} < \delta. \tag{4}$$

δ is the convergence tolerance. Here, $\sum_{\text{all elements}}$ denotes the sum over all elements and N is the number of elements for which the damage has changed in the current iteration.

4

2.4   FEA model

ParOSolNL is based on unstructured grids, initial tissue material properties and external displacement boundary conditions. Three different model types are investigated: (M1) a homogeneous cuboid for quantitative verification, (M2) a trabecular (virtual) biopsy for parameter optimization and (M3) a section of a human radius for performance measurements. Model 1 is a homogeneous truss with uniform displacement boundary conditions to mimic a one dimensional stress state. A loading–unloading cycle is applied. Model 2 and 3 are created from a μCT scan of a radius with a resolution of $16.4\,\mu\text{m}$. For model 2, a cuboid of 150 voxels side length was cropped from the centre of the radius. The resulting specimen consists only of trabecular bone, has approx. 0.6 mio elements (2.24 mio dof) and a bone volume density of $19\,\%$. For model 3, a slice of 416 voxels height was cropped from the centre of the radius resulting in approx. 213 mio elements (688 mio dof). Compression boundary conditions were applied: The bottom plane was fixed in all directions. On the top plane, displacements in the negative z-direction were applied while x- and y-directions were fixed.

2.5   Performance measurements

The reported results were obtained using two different compute servers: (S1) a small shared memory system for testing and parameter study using models M1 and M2 (2×12 cores Intel Xeon E5-2697 @2.70GHz, 384 GB RAM) and (S2) Vienna Scientific Cluster (VSC) for performance evaluation on model M3 (2×8 cores Intel Xeon E5-2650v2 @2.6GHz, 64 GB RAM per node, Intel QDR-80 dual-link high-speed InfiniBand).

When comparing two simulation results, the relative error δF_{rel} is defined via the maximum force of a reference force–displacement curve:

$$\delta F_{\text{rel}} = \frac{F(U_{\text{ref}}) - F_{\text{max}}}{F_{\text{max}}}, \tag{5}$$

where U_{ref} and F_{max} are the displacement and force at the maximum point of the reference curve.

3   RESULTS

The results for model M1, the quasi-one dimensional structure, show excellent agreement with the analytical solution (Figure 3, left). The stress–strain path correctly follows the loading and unloading cycle from tension over unloading to failure in compression. The material shows a homogeneous stress and damage distribution.
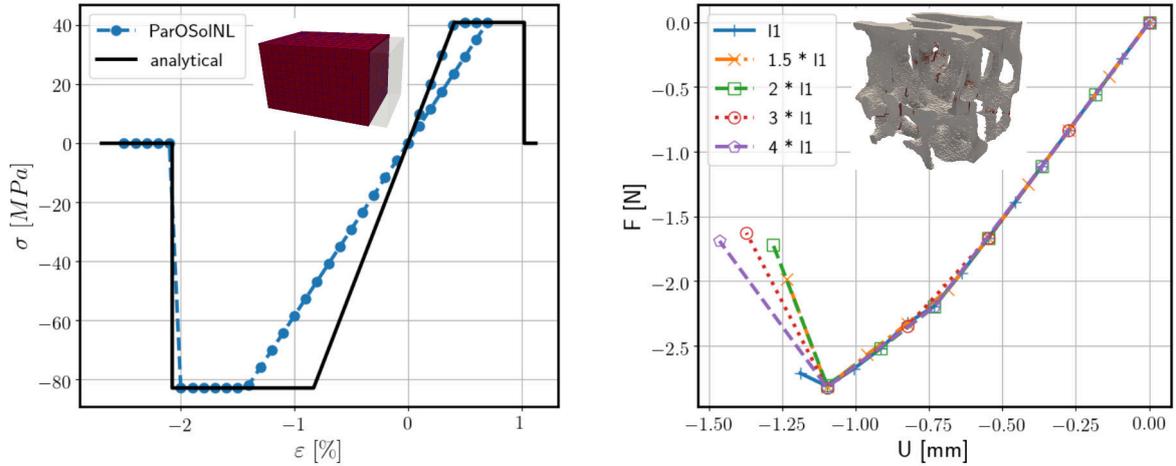
Figure 3: Left: σ–ε for a homogeneous truss (M1) in loading–unloading cycle: Simulation with ParOSolNL (dots), analytic results (solid line). Right: Global force–displacement curve of a trabecular bone sample (M2) for different increment sizes. The colours in the models indicate the damage.

Model M2 was compressed until ultimate load. The resulting damage distribution shows a realistic pattern (Figure (3), right). The performance of the nonlinear solver depends strongly on the solver control parameters. Using model M2, three solver control parameters were identified that have a major impact on the run time of the nonlinear simulation: (1) Lowering the linear convergence tolerance from 10^{-8} to 10^{-5} leads to a more than 4 times shorter run time (Table 1). Until the ultimate stress, the relative error in the global stress is still acceptable using a linear convergence of 10^{-5} ($\leq 0.07\%$). (2) The size of the load increments has no large effect on the ultimate load (Figure 3, right) but on the post-yield behaviour. The run times for different increment sizes differ by less than 10%. (3) The ratio ($E_{\mathrm{fracture}}/E_0$) between failed and initial tissue Young's modulus greatly affects the convergence rate (Table 2). The run times can be decreased by a factor of two by increasing E_{fracture} from 0 to $10^{-3}E_0$. To reduce computational requirements ($E_{\mathrm{fracture}}/E_0$) = 10^{-3} is chosen for performance tests, this comes with an error of up to 7% in the ultimate load.

Table 1: Total run time (using 24 processes on compute server S1) and relative error in the ultimate load, δF_{rel}, for different combinations of linear and nonlinear convergence tolerances. The grey row was chosen for Model M3.

linear tolerance	nonlinear tolerance	run time[s]	$\delta F_{\mathrm{rel}}[\%]$
10^{-8}	10^{-7}	4930	reference
10^{-7}	10^{-6}	2744	-0.02
10^{-6}	10^{-5}	1562	-0.03
10^{-5}	10^{-4}	1043	-0.07
10^{-4}	10^{-3}	891	-0.26

Table 2: Total run time (using 24 processes on compute server S1) and relative error in the ultimate load, δF_{rel}, for different ($E_{\mathrm{fracture}}/E_0$). Due to limited computational resources, the grey row was chosen for model M3.

$E_{\mathrm{fracture}}/E_0$	run time[s]	$\delta F_{\mathrm{rel}}[\%]$
0	5521	reference
10^{-5}	4190	-4.2
10^{-4}	4431	-4.7
10^{-3}	2744	-6.9
10^{-2}	1525	-15.1

The performance of the nonlinear solver was investigated using model M3, a radius section with approx. 213 mio elements (Figure 4). The nonlinear simulation used 5650 single-core hours and approx. 76 GB of total memory on 256 cores. The reported memory is computed as the average resident set size (RSS) per process times the number of cores.
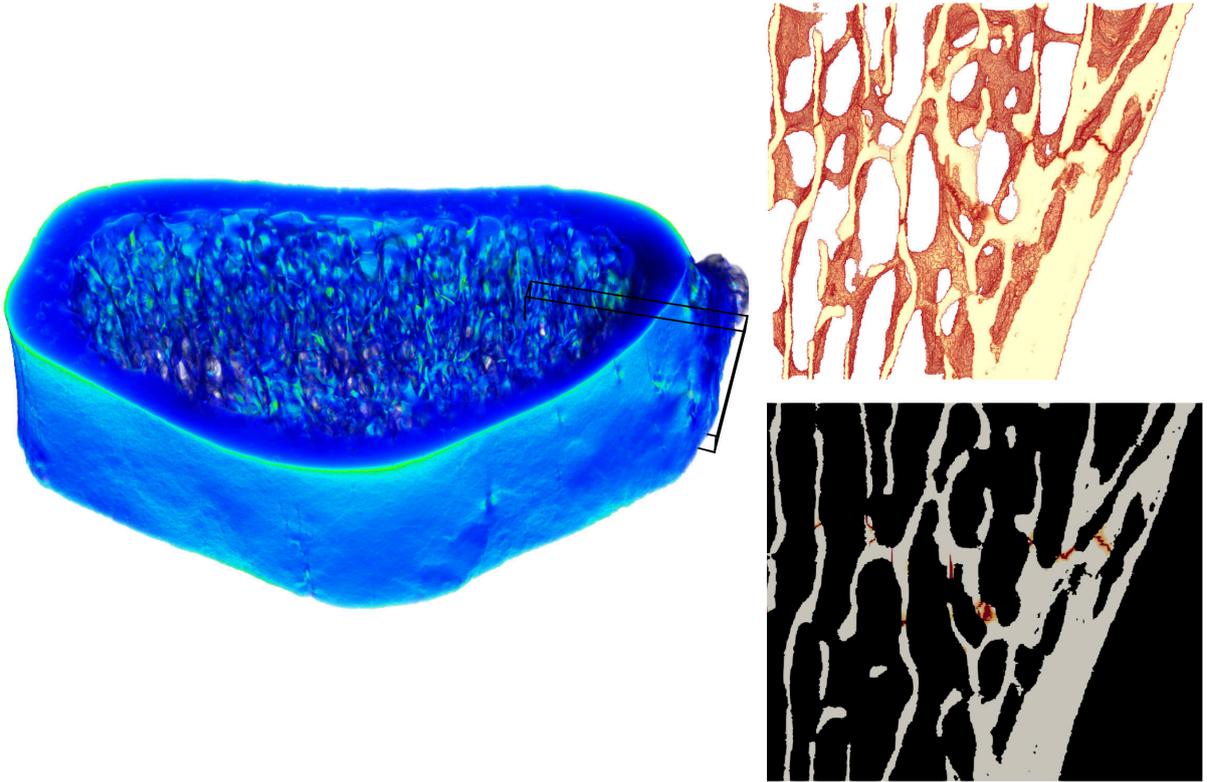
Figure 4: Simulation result for the radius segment (M3) at the ultimate stress point. Left: 3-dimensional image (colours indicate von Mises stress). Right: Damage distribution for small segment (3-dimensional view at top, central layer at bottom).

A detailed investigation of the total run time shows that over 94 % of the total time was spend in the linear solver (Figure 5, left). The nonlinear overhead, i.e. the loop over

all elements and the application of the constitutive model, takes less than 5% of the total time. In addition, strong scaling of one increment where no damage is accumulated, shows that the linear solver scales very good (Figure 5, right). Preprocessing and I/O do not scale, but are only done once per simulation or increment, respectively. The completion of a nonlinear analysis takes more than one order of magnitude more time than one linear-elastic analysis. This is due to the high number of iterations needed until a converged state is found.
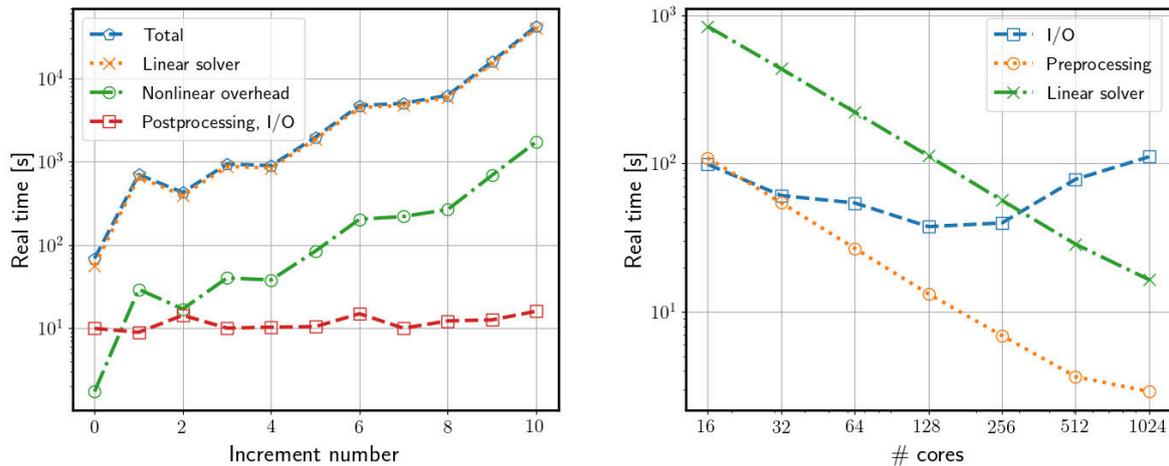
Figure 5: Left: Run time per increment for the radius section (M3, 688 mio dof) using 256 cores. Right: Strong scaling of one linear increment.

4 DISCUSSION

A simple nonlinear material model has been implemented into the existing framework ParOSol and successfully tested. The correct implementation of the material model was verified using a quasi single–element test with a homogeneous cuboid under uniform uni-axial compression. Nonlinear analysis of a small trabecular biopsy and a radius section showed qualitatively plausible results. Damage occured primarily at thin parts of the structure and at imperfections, e.g. holes in the cortex.

Using a radius section with 213 mio elements (688 mio dof), it has been proven that large simulations are feasible using the new implementation. The linear solver scales very well up to the tested 1024 CPU cores, comparable to the original ParOSol [6]. Although only little overhead is added in each iteration by the application of the nonlinear material model, a large number of iterations is needed until convergence is reached.

The performance of the new solver strongly depends on the solver control parameters. By carefully choosing the convergence tolerance and E_{fracture}, the run time can be considerably reduced. For E_{fracture} this is due to the preconditioner of the linear solver: ParOSol uses a geometric multigrid algorithm to precondition the system. The larger the modulus differences, the worse is the preconditioned system and thus the solving time increases. Due to very limited computational resources, we chose solver parameters that lead to an

error of approx. 10 %. This dependence needs to be taken into account when comparing to literature.

Only two comparably large simulations are known to the authors: One simulation of femura [10] and one of vertebra samples [16]. The reported run times and memory requirements per sample are listed in Table 3. A direct comparison of the performance is not possible since [10] and [16] use a geometrically and material nonlinear framework, while we incorporated only material nonlinearity. Furthermore, the implemented material models differ and the solver control parameters are not known.

Taking all this into account, the new ParOSolNL has at least a comparable performance. One major advantage is the low memory requirements. ParOSolNL needs approx. 76 GB for the simulation of a model containing 213 mio elements which is more than 3 orders of magnitude less than reported in [10]. This low memory footprint is due to the completely matrix-free implementation. It allows the simulation of models with 55 mio elements on an average shared memory server comparable to S1.

Table 3: Performance of large material (m) and / or geometric (g) nonlinear simulations in literature.

Reference	Model size [mio elements]	CPU hours	# cores	memory	nonlinearity
Nawathe et al.[10]	120 (dof: 400)	44.000	4096	120 TB	m, g
Fields et al.[16]	485	30.000	2000	unknown	m, g
ParOSolNL	213 (dof: 688)	5650	256	76 GB	m [1]

[1] The reported performance strongly depends on the value of the solver control parameters.

This study has a number of limitations: First, the reported run times and memory requirements were measured only once due to limited resources. Although the compute nodes on server S2 were reserved for the jobs, the storage servers are shared resources. Furthermore, the run time depends on the layout of the used compute nodes, which were allocated automatically to the jobs. Second, we use a simple algorithm which iteratively applies the linear solver of ParOSol. A large number of iterations is needed to reach a converged state. A reduction of the number of iterations could possibly be achieved by implementing a Newton–Raphson based algorithm. However, it is likely that the abrupt change of the Young's modulus, when fracture occurs, would lead to convergence problems. Third, no geometric nonlinearity was considered in this study. Large deformations occur mainly when elements fracture and parts of the structure fail. Although we are mainly interested in the results until the ultimate yield point, there were already about 1 % of the elements fractured. Thus, the linear approximation of strains and displacements can lead to a perceptible error. Fourth, we use a simple material model which does not include hardening, plasticity and anisotropy. The material parameters are taken directly from literature for comparable material models without direct validation based on experiments. Thus, the next step will be to test the validity of our approach in direct comparison with experimental data.

The implementation of a simple nonlinear material model in an efficient linear-elastic

FEA framework enables simulations of large digitized structures, e.g. bones. The new implementation shows a good parallel performance, comparable to existing codes, and a lower memory footprint . Detailed insight into the local failure pattern will be helpful for a better understanding of the complex failure behaviour of bone.

5   ACKNOWLEDGEMENTS

REFERENCES

[1] van Rietbergen, B., Weinans, H., Huiskes, R., and Odgaard, A. A new method to determine trabecular bone elastic properties and loading using micromechanical finite-element models. *Journal of Biomechanics* (1995) **28**(1):69–81.

[2] Harrison, N. M., McDonnell, P. F., O'Mahoney, D. C., Kennedy, O. D., O'Brien, F. J., and McHugh, P. E. Heterogeneous linear elastic trabecular bone modelling using micro-CT attenuation data and experimentally measured heterogeneous tissue properties. *Journal of Biomechanics* (2008) **41**:2589–2596.

[3] Verhulp, E., van Rietbergen, B., Müller, R., and Huiskes, R. Indirect determination of trabecular bone effective tissue failure properties using micro-finite element simulations. *Journal of Biomechanics* (2008) **41**(7):1479–1485.

[4] Arbenz, P., van Lenthe, G. H., Mennel, U., Müller, R., and Sala, M. A scalable multilevel preconditioner for matrix-free µ-finite element analysis of human bone structures. *International Journal for Numerical Methods in Engineering* (2008) **73**(7):927–947.

[5] Taylor, R. Feap – finite element analysis program (2014). URL `http://www.ce.berkeley/feap`.

[6] Flaig, C., and Arbenz, P. A scalable memory efficient multigrid solver for micro-finite element analyses based on ct images. *Parallel Computing* (2011) **37**(12):846–854.

[7] Flaig, C. *A highly scalable memory efficient multigrid solver for µ-finite element analyses*. PhD thesis, Eidgenössische Technische Hochschule ETH Zürich, 2012.

[8] Christen, D. B. *Nonlinear failure prediction in human bone – a clinical approach based on high resolution imaging*. PhD thesis, Eidgenössische Technische Hochschule ETH Zürich, 2012.

[9] Adams, M. F., Bayraktar, H. H., Keaveny, T. M., and Papadopoulos, P. Ultrascalable implicit finite element analyses in solid mechanics with over a half a billion degrees of freedom. *In Proceedings of the 2004 ACM/IEEE conference on Supercomputing* (2004), IEEE Computer Society, p. 34.

[10] Nawathe, S., Akhlaghpour, H., Bouxsein, M. L., and Keaveny, T. M. Microstructural failure mechanisms in the human proximal femur for sideways fall loading. *Journal of bone and mineral research* (2014) **29**:507–515.

[11] Schwiedrzik, J. J., Wolfram, U., and Zysset, P. K. A generalized anisotropic quadric yield criterion and its application to bone tissue at multiple length scales. *Biomech Model Mechanobiol* (2013) **12**(6)1155–1168.

[12] Ashman, R. B., and Rho, J. Y. Elastic modulus of trabecular bone material. *Journal of Biomechanics* (1988) **21**(3):177–181.

[13] Homminga, J., McCreadie, B., Ciarelli, T., Weinans, H., Goldstein, S., and Huiskes, R. Cancellous bone mechanical properties from normals and patients with hip fractures differ on the structure level, not on the bone hard tissue level. *Bone* (2002) **30**(5):759–764.

[14] Bayraktar, H. H., Morgan, E. F., Niebur, G. L., Morris, G. E., Wong, E. K., and Keaveny, T. M. Comparison of the elastic and yield properties of human femoral trabecular and cortical bone tissue. *Journal of Biomechanics* (2004) **37**(1):27–35.

[15] Schwiedrzik, J. J., and Zysset, P. K. The influence of yield surface shape and damage in the depth-dependent response of bone tissue to nanoindentation using spherical and Berkovich indenters. *Computer Methods in Biomechanics and Biomedical Engineering* (2015) **18**(5):492–505.

[16] Fields, A. J., Nawathe, S., Eswaran, S. K., Jekir, M. G., Adams, M. F., Papadopoulos, P., and Keaveny, T. M. Vertebral fragility and structural redundancy. *Journal of Bone and Mineral Research* (2012) **27**(10):2152–2158.