μ-Finite Element Analysis of Human Bone Structures

Peter Arbenz

Chair of Computational Science, ETH Zürich
E-mail: arbenz@inf.ethz.ch
http://people.inf.ethz.ch/arbenz/

Collaborators

- Chair of Computational Science, ETH Zurich
  - Uche Mennel
  - Marzio Sala
  - Cyril Flaig
  - Yves Ineichen
  - Sumit Paranjape
- Institute for Biomechanics, ETH Zurich
  - Harry van Lenthe
  - Andreas Wirth
  - David Christen
  - Ralph Müller
- IBM Zurich Research Lab
  - Costas Bekas
  - Alessandro Curioni

Outline of the talk

Introduction / Osteoporosis

- In healthy bone, there is continuous matrix remodeling of bone. Up to 10% of all bone mass may be undergoing remodeling at any point in time.
- Osteoporosis is a disease characterized by an imbalance between bone resorption and bone formation.
- Osteoporosis causes a low bone mass and a deterioration of the bone microarchitecture.

[From http://www.osteoswiss.ch]

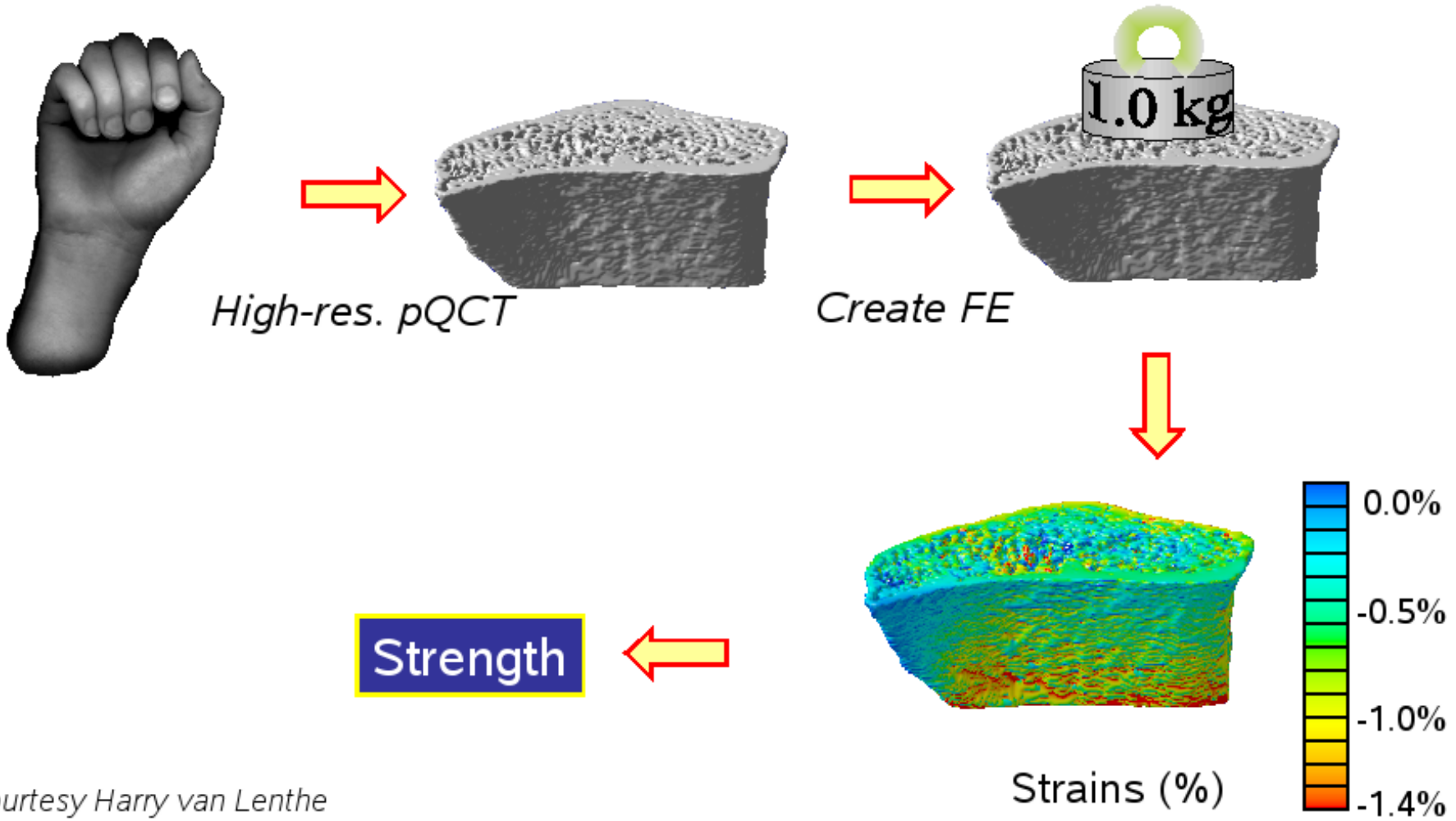
Introduction / Osteoporosis (cont.)

- In Switzerland, the risk for an osteoporotic fracture in women above 50 years is about 50%, for men the risk is about 20%.
- Mostly affected bones are vertebra, femur, and radius.
- Not all fractures are actually noticed, but osteoporosis means a considerable reduction in the quality of life for many affected people.
- Enormous impact on individuals, society and health care systems (as health care problem second only to cardiovascular diseases).
- Osteoporosis is tried to be detected by Densitometry or other imaging technologies such as Quantitative Computer Tomography (QCT). These techniques allow to estimate / measure the bone mineral density.

Introduction / Osteoporosis (cont.)

- Since global parameters like bone density do not admit to predict the fracture risk, patients have to be treated in ways that take into account the structure of individual bones.
- Patient-specific finite element (FE) models of bones based on 3D high-resolution CT images become increasingly attractive for evaluation of stiffness and strength *in vivo*.
- This project is a part of the VPHOP or the Osteoporotic Virtual Physiological Human Project in FP7. One of the goals: predict the absolute risk of fracture in patients with low bone mass

In vitro/*in vivo* assessment of bone strength



Courtesy Harry van Lenthe
University and ETH Zurich

pQCT: peripheral Quantitative Computed Tomography

In vitro/*in vivo* assessment of bone strength (cont'd)

1. μCT / high-resolution pQCT scan
2. CT voxel information is converted directly into 8-noded hexahedral elements $\longrightarrow$ large number of degrees of freedom
3. A mathematical model for trabecular bones: linearized 3D elasticity.
4. Bone tissue is assumed to be isotropic
   Elastic parameters (elasticity modulus $E = 17$ GPa, Poisson ratio $\nu = 0.3$) are taken from measurements.
5. A scalable, robust and reliable parallel solver
6. A scalable high-performance computer

The mathematical model

- Equations of linearized 3D elasticity (pure displacement formulation): Find displacement field $\mathbf{u}$ that minimizes total potential energy

$$\int_\Omega \left[\mu\, \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{u}) + \frac{\lambda}{2} (\operatorname{div} \mathbf{u})^2 - \mathbf{f}^t \mathbf{u} \right] d\Omega - \int_{\Gamma_N} \mathbf{g}_S^t \mathbf{u} d\Gamma,$$

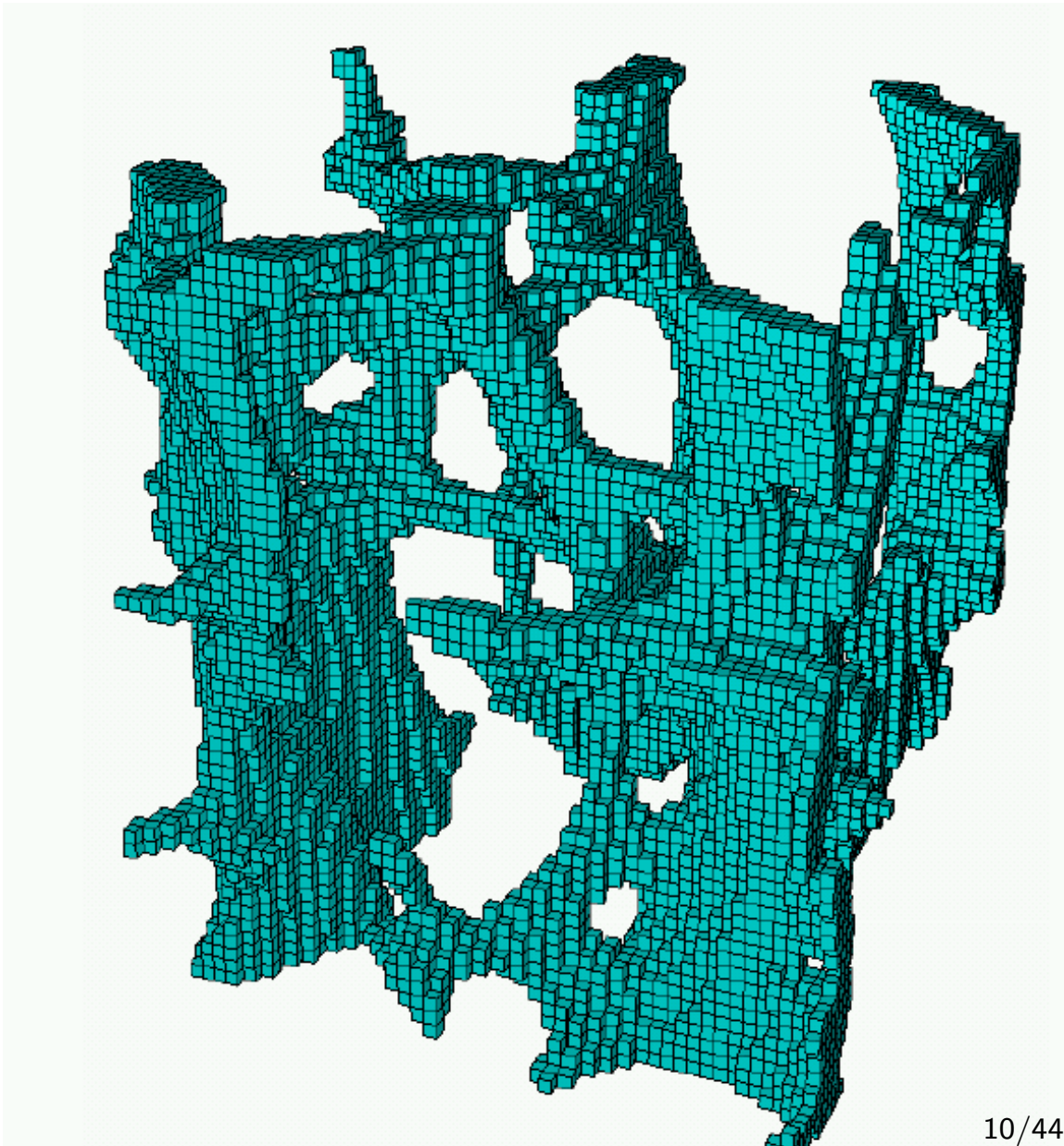
  with Lamé's constants λ, μ, volume forces $\mathbf{f}$, boundary tractions $\mathbf{g}$, symmetric strain tensor

$$\varepsilon(\mathbf{u}) := \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T).$$

- In typical analyses: Vertical displacements at top and bottom boundary planes are (partially) prescribed (Dirichlet boundary conditions).
  Otherwise boundary is free (homogeneous Neumann b.c.)

Continuum finite element (FE) models using μFE

Size of voxels
10-100μm $\longrightarrow$ μFE
analysis.



Solving the system of equations

- System of equation

$$K\mathbf{x} = \mathbf{b}$$

  K is very large, sparse, symmetric positive definite.
- Original (Rietbergen 1996 [3]) approach by people of ETH Biomechanics: preconditioned conjugate gradient (PCG) algorithm
  - element-by-element (EBE) matrix multiplication

$$K = \sum_{e=1}^{n_{\text{el}}} T_e K_e T_e^T, \tag{1}$$

    Note: all element matrices are identical!
  - diagonal (Jacobi) preconditioning
  - very memory economic, slow convergence as problems get big

Solving the system of equations (cont.)

- Our first approach targeted at reducing the iteration count: PCG which smoothed aggregation AMG preconditioning [4] (Similar to Adams *et al.* SC'04 [1])
- Requires assembling $K \longrightarrow$ large memory requirement.
- Needs parallelization for distributed memory machines.
- Employ software: Trilinos (Sandia NL, Albuquerque, NM) In particular we use
  - Distributed (multi)vectors and (sparse) matrices (Epetra).
  - Domain decomposition (load balance) with ParMETIS
  - Iterative solvers and preconditioners (AztecOO)
  - Smoothed aggregation AMG preconditioner (ML)
  - Direct solver on coarsest level (AMESOS)

Solving the system of equations (cont.)

- Our second approach targeted at reducing memory consumption:
  Want to stick with PCG and devise a scalable (multigrid) preconditioner that can be built without assembling K. [2]
  - Typical operator complexity of SA-AMG is 1.4. Hope to reduce the memory requirements by a factor

$$3.5 = \frac{1.4}{0.4}.$$

- Smoother: Chebyshev polynomial smoother (Adams *et al.* 2003)
  - Determine a polynomial that is small on the 'upper part of the spectrum'.
  - Only needs matrix-vector product.
  - Perfect scalability.

Constructing a multigrid preconditioner without system matrix

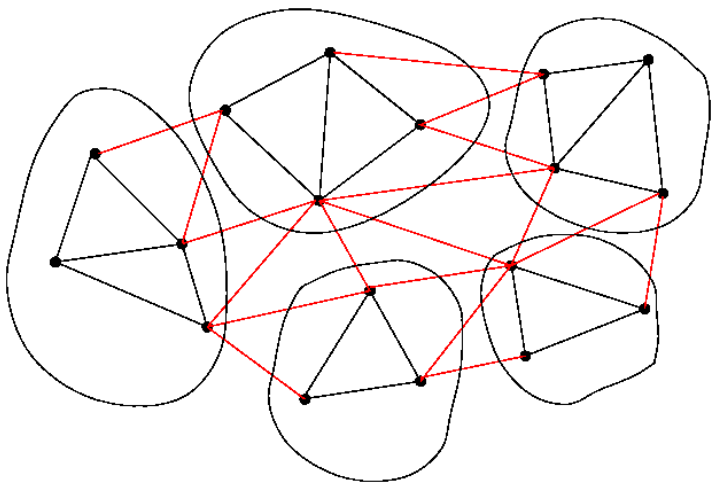
- Smoothed Aggregation based Algebraic MultiGrid (SA-AMG) preconditioning does not need system matrix to build multigrid hierarchy!
- The aggregates are constructed by means of the matrix graph. Each node of the grid gives rise to a 3×3 block in the system matrix. Graph is an order of magnitude smaller.
  We use METIS to construct aggregates. Aggregates consist of ca. 100 nodes.
- Prolongator is based on near-kernel (near-nullspace) which in linear elasticity are translations and rotations that are determined from the node coordinates.
- Our new approach: Generate the second-finest multigrid level matrix-free with unsmoothed aggregates, and the further levels in the standard way.

Setup procedure for an abstract multigrid solver

1: Define the number of levels, L
2: **for** level $\ell = 0, \ldots, L-1$ **do**
3: **if** $\ell < L-1$ **then**
4: Define prolongator P_ℓ;
5: Define restriction $R_\ell = P_\ell^T$;
6: $K_{\ell+1} = R_\ell K_\ell P_\ell$;
7: Define smoother S_ℓ;
8: **else**
9: Prepare for solving with K_ℓ;
10: **end if**
11: **end for**

Smoothed aggregation (SA) AMG preconditioner I

1. Build adjacency graph $\mathcal{G}_0$ of $K_0 = K$.
   (Take 3×3 block structure into account.)
2. Group graph vertices into contiguous subsets, called *aggregates*. Each aggregate represents a coarser grid vertex.
   - Typical aggregates: $3 \times 3 \times 3$ nodes (of the graph) up to $5 \times 5 \times 5$ nodes (if aggressive coarsening is used)
   - ParMETIS
   - Note: The matrices $K_1, K_2, \ldots$ need much less memory space than K_0!



Smoothed aggregation (SA) AMG preconditioner II

3. Define a grid transfer operator:
   - Low-energy modes (near kernel), in our case, the rigid body modes are 'chopped' according to aggregation

$$B_\ell = \begin{bmatrix} B_1^{(\ell)} \\ \vdots \\ B_{n_{\ell+1}}^{(\ell)} \end{bmatrix} \qquad B_j^{(\ell)} = \text{rows of } B_\ell \text{ corresponding to grid points assigned to } j^{th} \text{ aggregate.}$$

   - Let $B_j^{(\ell)} = Q_j^{(\ell)} R_j^{(\ell)}$ be QR factorization of $B_j^{(\ell)}$ then

$$B_\ell = \widetilde{P}_\ell B_{\ell+1}, \qquad \widetilde{P}_\ell^T \widetilde{P}_\ell = I_{n_{\ell+1}},$$

     with

$$\widetilde{P}_\ell = \text{diag}(Q_1^{(\ell)}, \ldots, Q_{n_{\ell+1}}^{(\ell)}) \qquad \text{and} \qquad B_{\ell+1} = \begin{bmatrix} R_1^{(\ell)} \\ \vdots \\ R_{n_{\ell+1}}^{(\ell)} \end{bmatrix}.$$

     Columns of $B_{\ell+1}$ span the near kernel of $K_{\ell+1}$.
   - Notice: matrices K_ℓ are *not* used in constructing tentative prolongators $\widetilde{P}_\ell$, near kernels B_ℓ, and graphs $\mathcal{G}_\ell$.

Smoothed aggregation (SA) AMG preconditioner III

4. For elliptic problems, it is advisable to perform an additional step, to obtain *smoothed aggregation* (SA).

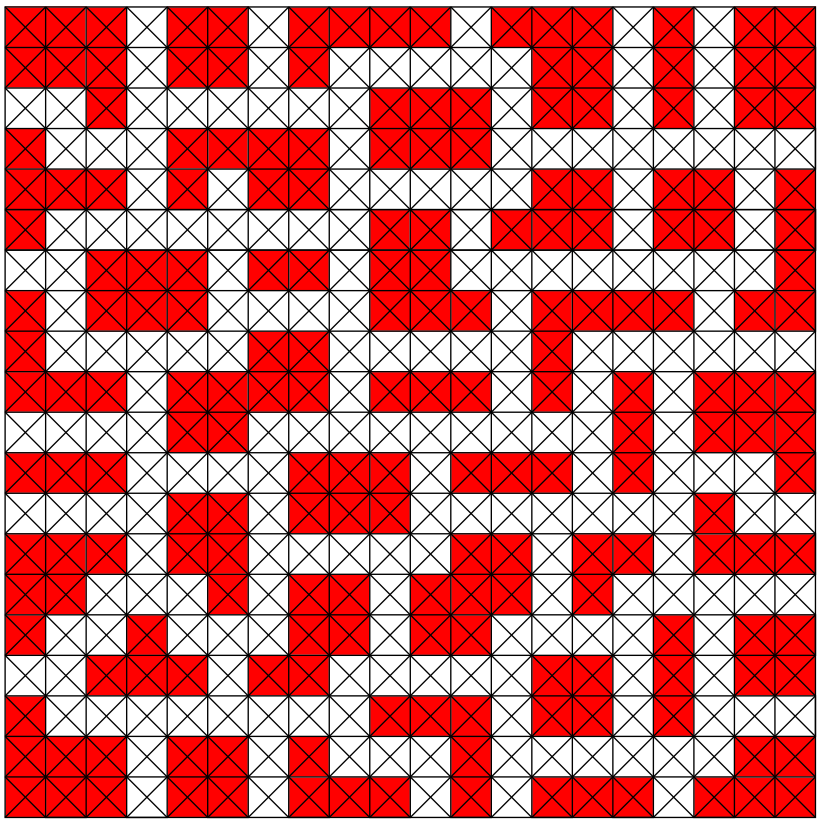
$$P_\ell = (I_\ell - \omega_\ell D_\ell^{-1} K_\ell)\,\widetilde{P}_\ell, \qquad \omega_\ell = \frac{4/3}{\lambda_{\max}(D_\ell^{-1} K_\ell)},$$

smoothed prolongator

In *non-smoothed* aggregation: $P_\ell = \widetilde{P}_\ell$

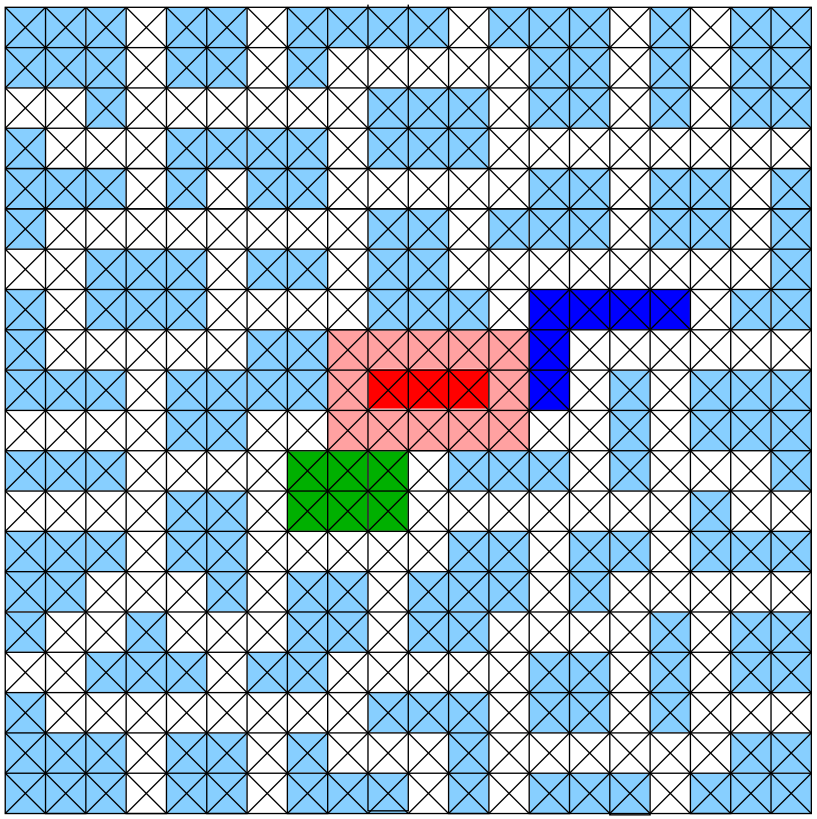
Computing $K_1 = P_0^T K_0 P_0$

- P_0 maps the degrees of freedom of the finest level (nodes) to the degrees of freedom of the second finest level (aggregates).
- In (block) column i those rows are nonzero that correspond to nodes of aggregate i.



Computing $K_1 = P_0^T K_0 P_0$ (cont.)

- Multiplying K_0 with the i-th column of P_0 introduces nonzeros at nodes directly connected to aggregate i.



Computing $K_1 = P_0^T K_0 P_0$ (cont.)

- Want to collapse different columns of P_0 in one column of $\hat{P}_0$ such that nonzeros in one column of $K_0\hat{P}_0$ do not interfere with each other.
  Aggregates have to be sufficiently far apart.
- Since other neighbors of a neighbor can interfere with the original aggregate, neighbors of neighbors cannot be in equal columns.
- Can use graph coloring algorithm of the graph of K_1:
  aggregates that are two edges apart must have different colors (distance-2 coloring).

'Matrix-free' multigrid

- We do NOT form $K = K_0$ but do an element-by-element (EBE) matrix multiplication

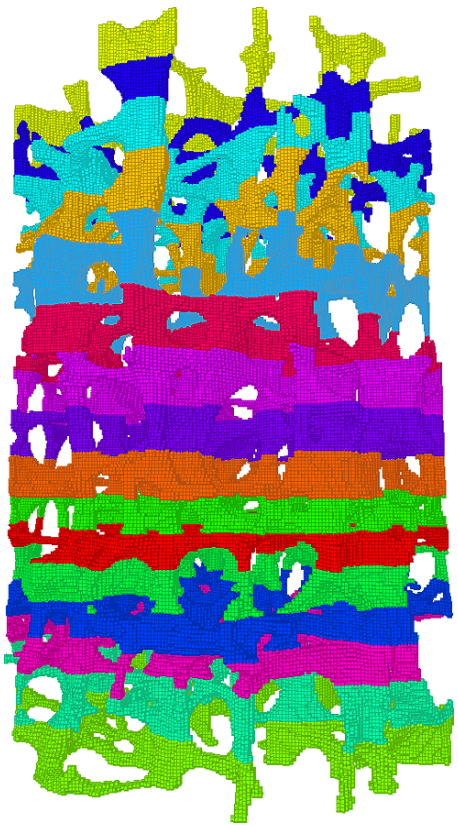
$$K = \sum_{e=1}^{n_{\text{el}}} T_e K_e T_e^T$$

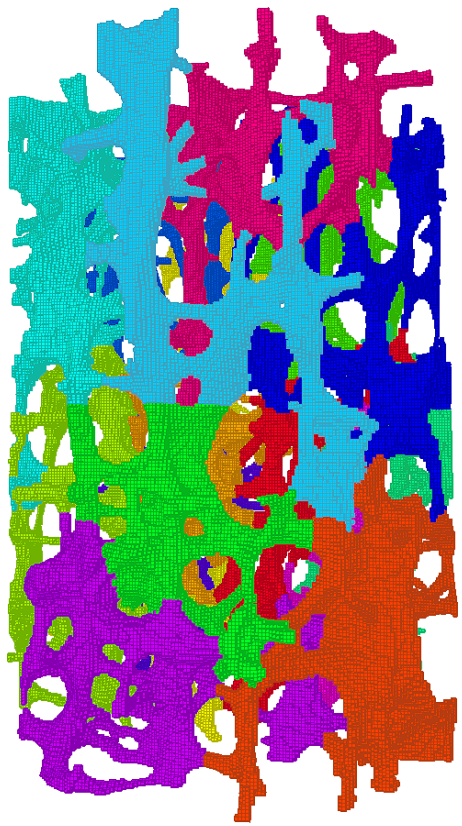
- We do NOT smooth the first prolongator P_0.
- Matrices $K_2, K_3, \ldots$ are formed with smoothed prolongators from $K_1, K_2, \ldots$
- All graphs, including $\mathcal{G}_0$ are constructed.
- Memory savings (crude approximation): ~ 4.
- Needs clever construction of $K_1 = P_0^T K_0 P_0$ since forming $K_0 P_0 = \sum_{e=1}^{n_{\text{el}}} T_e K_e T_e^T P_0$ is infeasible.

Mesh partitioning

- Load balance: Each processor gets the same number of nodes
- Minimize solver communication: Minimize the surface area of the interprocessor interfaces
- Crucial for efficient parallel execution

Initial partition based on node coordinates





ParMETIS/ recursive coordinate bisection (RCB) repartition

The Trilinos software framework

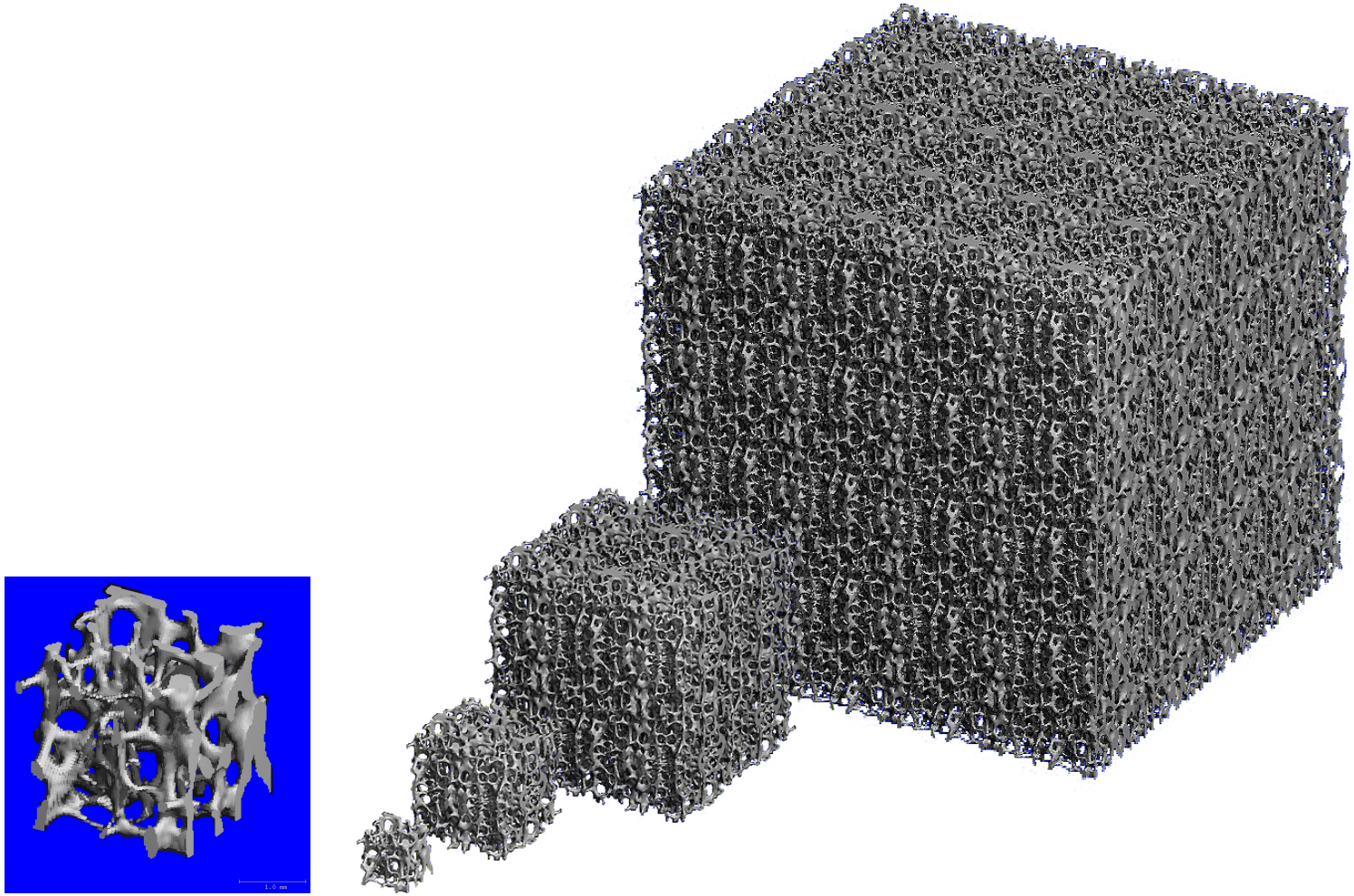
- The Trilinos Project is an effort to develop parallel solver algorithms and libraries within an object-oriented software framework for the solution of large-scale, complex multi-physics engineering and scientific applications.
- See http://trilinos.sandia.gov/
- We used the following packages
  - Epetra (distributed linear algebra objects)
  - AztecOO (PCG)
  - AMESOS (coarsest level solver)
  - IFPACK (Chebyshev)
  - ML (multilevel preconditioner, includes matrix-free version)
  - (Par)METIS (repartitioning, aggregation)
  - Isorropia (RCB repartitioning)

Numerical experiments

- Weak scalability test with an artificial bone.
  Number of degrees of freedom ranges from 300'000 to 2 billion.
- Strong scability test with a large real bone with $1.3 \cdot 10^9$ degrees of freedom.
- Typical production sized problem of $15 \cdot 10^6$ degrees of freedom.
- All numbers have been obtained on the Cray XT5 at the Swiss Supercomputing Center CSCS.

The machine has AMD Istanbul nodes with 6 cores and 8 GB RAM each.

Weak scalability of matrix-free preconditioning



Problem size scales with the number of processors.
Execution times should stay constant.

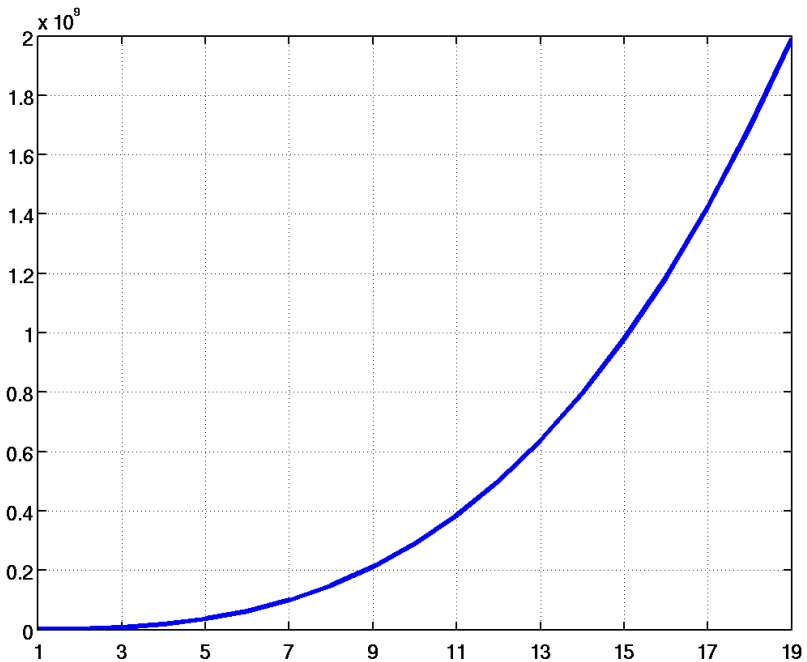
Weak scalability: problem sizes

We identify the artificial bones by C_x where $x = 1, \ldots, 19$.

Bone C_x has

#elements	$60'482\,x^3$
#nodes	$\sim 100'000\,x^3$
#matrix rows	$\sim 290'000\,x^3$

Number of degrees of freedom



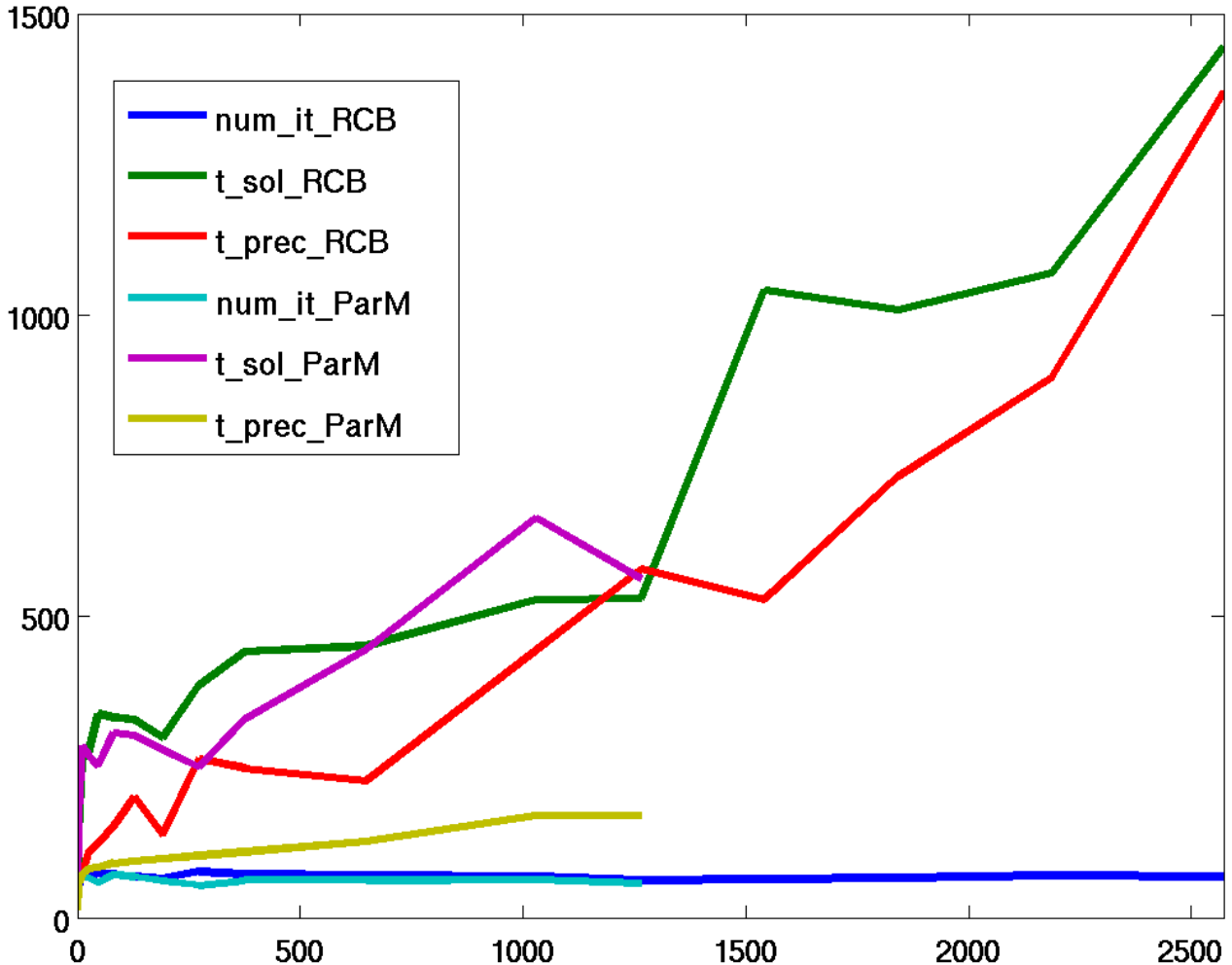
Weak scalability: ParMETIS

name	#cores	#it_steps	solver time	prec setup time
1	1	45	32	14
2	3	67	169	55
3	10	71	285	72
4	24	66	270	81
5	46	58	252	84
6	81	71	306	90
7	128	68	303	94
8	192	61	278	98
9	273	54	250	103
10	375	62	329	109
12	648	62	445	127
14	1029	63	664	169
15	1265	57	562	170
16	1536	—	—	—
17	1842	—	—	—
18	2187	—	—	—
19	2572	—	—	—

Weak scalability: Recursive Coordinate Bisection

name	#cores	#it_steps	solver time	prec setup time
1	1	45	30	14
2	3	53	137	67
3	10	66	264	75
4	24	66	273	108
5	46	74	338	123
6	81	73	332	152
7	128	68	329	200
8	192	65	298	137
9	273	76	386	264
10	375	72	442	248
12	648	70	451	227
14	1029	69	527	443
15	1265	62	530	578
16	1542	64	1042	528
17	1842	66	1009	733
18	2187	70	1070	897
19	2572	68	1444	1370

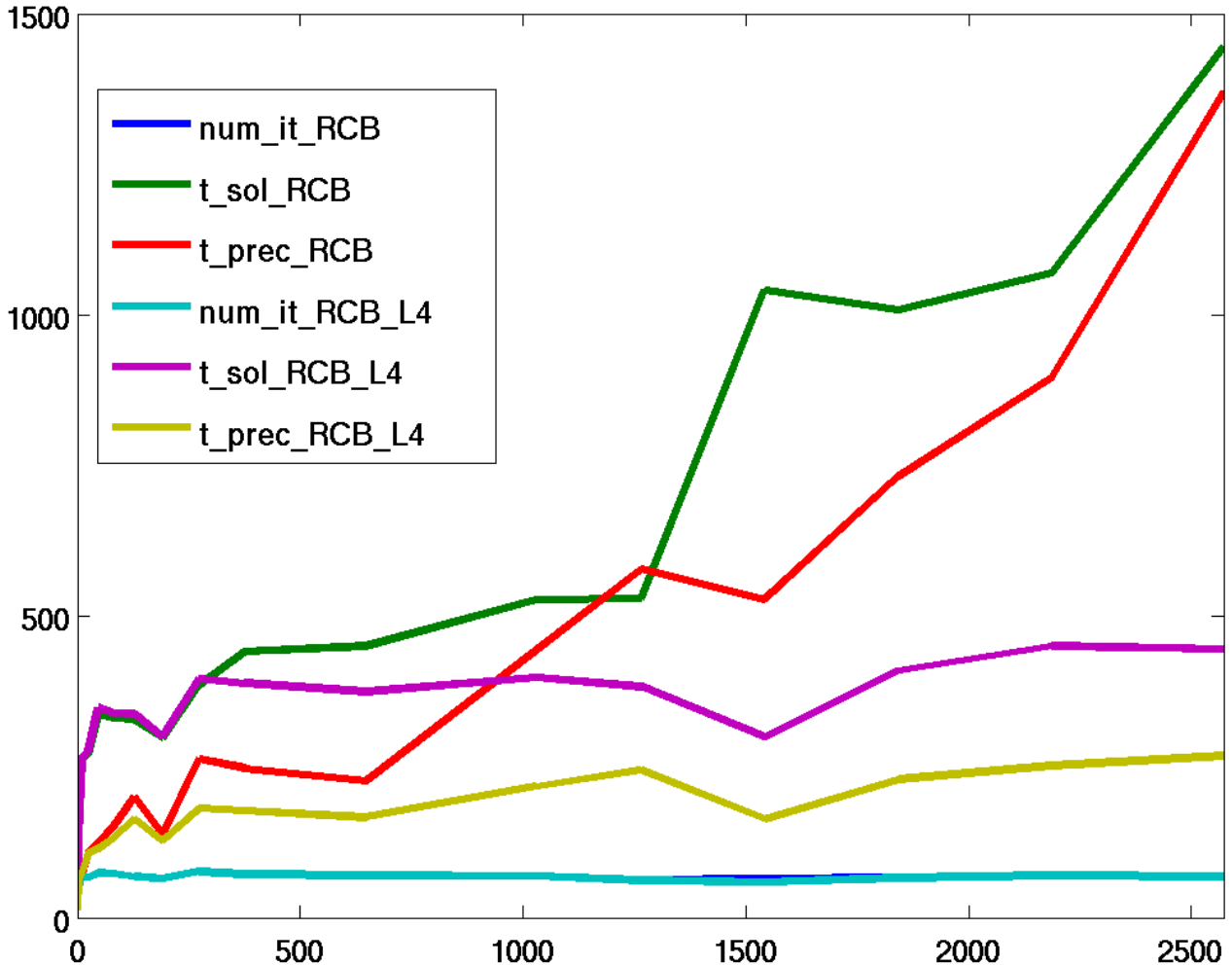
Weak scalability: ParMETIS vs. RCB



Weak scalability: ParMETIS vs. RCB (Discussion)

- When it works, ParMETIS performs better than RCB.
- Repartitioning by ParMETIS breaks down above $\sim$1250 cores.
- Reason: above this limit the program crashes with *unexpected message buffer overflows*. (Messages have arrived but corresponding receives have not been issued (yet).)
- This problem has been resolved by C. Bekas for the IBM Blue Gene by introducing collective communication.
- RCB run times increase above $\sim$1250 as well.
- Reason: On coarse grids matrices get dense. This entails
  - sparse matrix techniques for almost dense matrices
  - all-to-all communication in the aggregation phase
- Remedy: Limit levels and iterate on the (then) coarsest level (Tuminaro & Tong, SC'00)

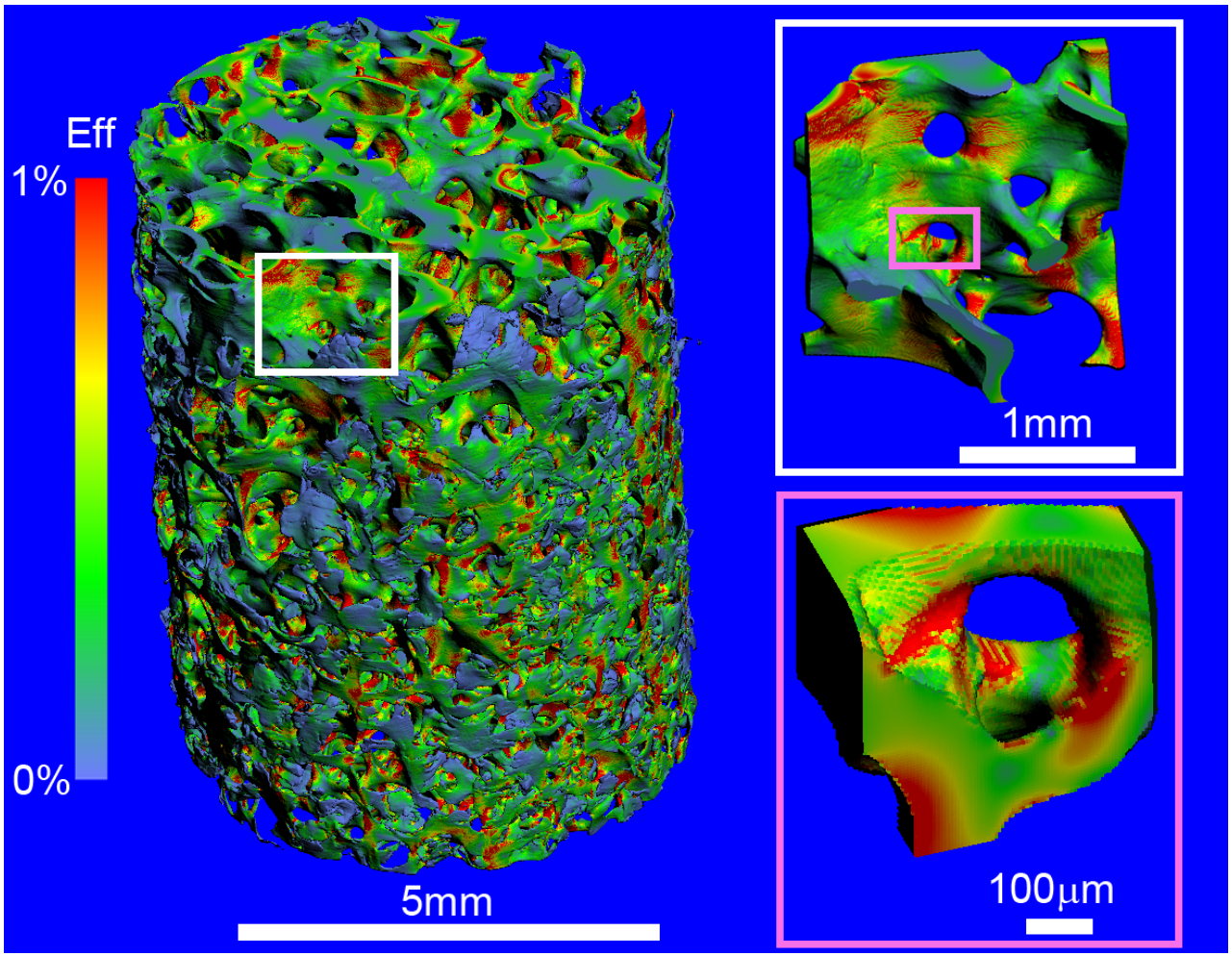
Weak scalability: RCB full vs. RCB reduced levels



Weak scalability: RCB full / reduced levels (Discussion)

- RCB repartitioning combined with a limitation of AMG levels (here 4) provides a scalable solver.
- Iterative solver on the coarsest level: PCG with Chebyshev polynomial preconditioner (here of degree 15).
- Problem sizes are limited by order $2^{31} \approx 2.15 \cdot 10^9$ as Trilinos (Epetra) does not use 64-bit integers.

Strong scalability: very large bone



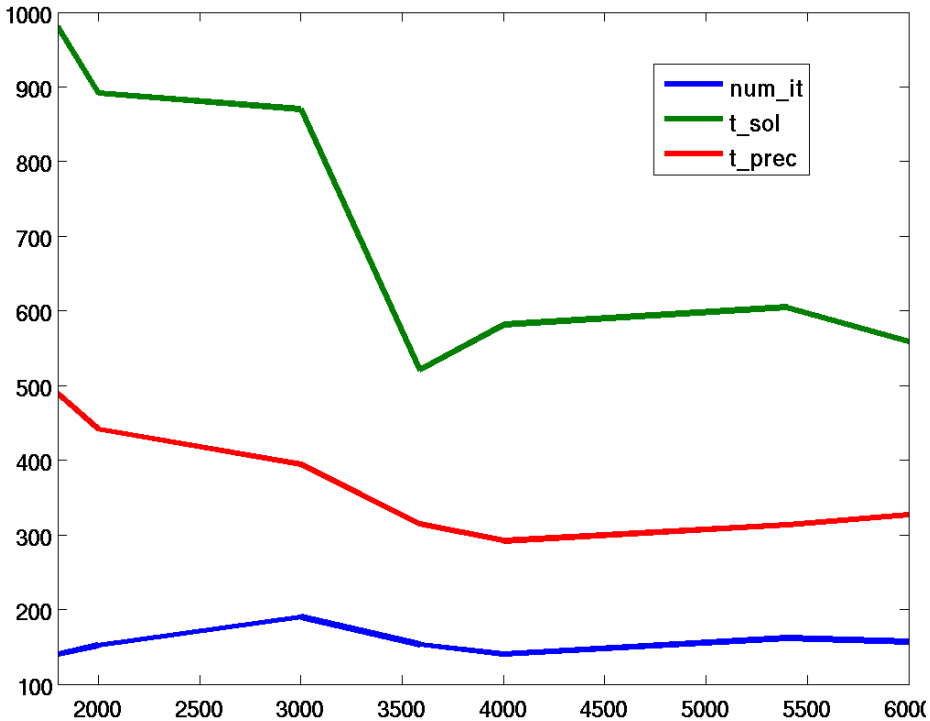
Effective strains with zooms.
(Image by Jean Favre, CSCS)

Strong scalability: very large bone (cont.)

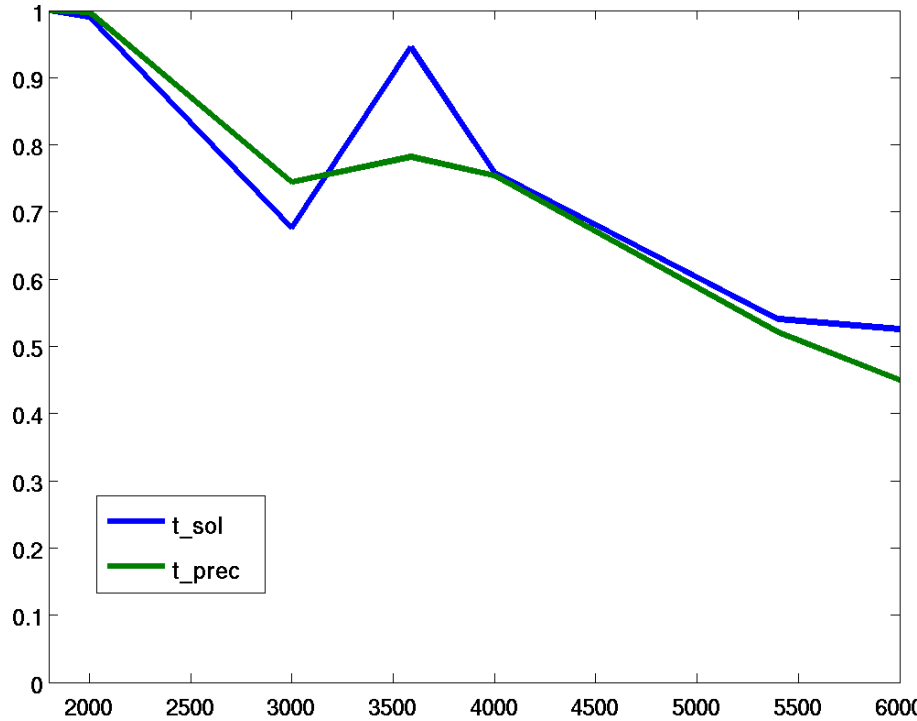
Fixed problem size: $n = 1.342 \cdot 10^9$ dof's

# cores	# iterations	solution time	precond. constr. time
1800	140	981	489
2000	152	892	441
3000	190	870	394
3588	153	521	314
4000	140	581	292
5400	162	605	313
6000	157	560	327

Strong scalability: very large bone (cont.)

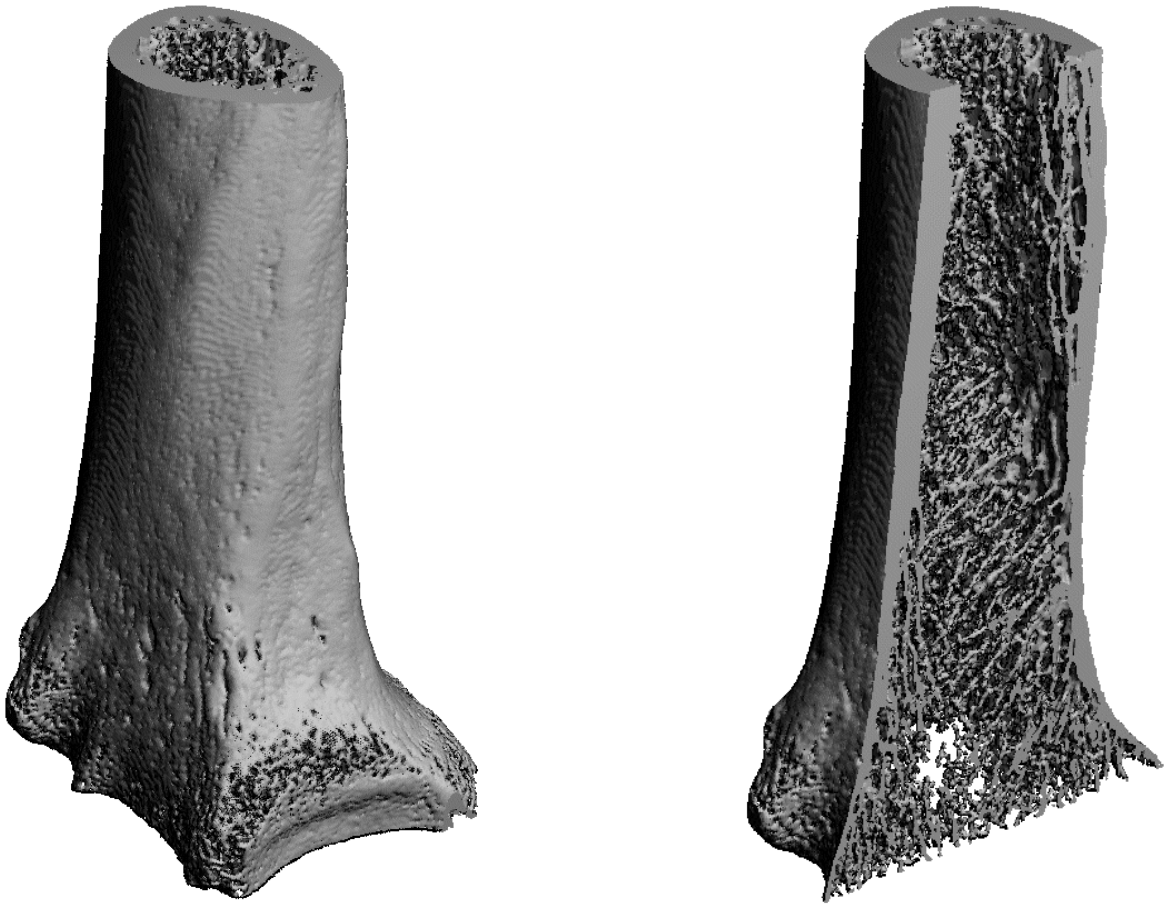


Execution times for solution and construction of preconditioner



Relative efficiencies for solution and construction of preconditioner

Human bone problems



Distal part (20% of the length) of the radius in a human forearm.

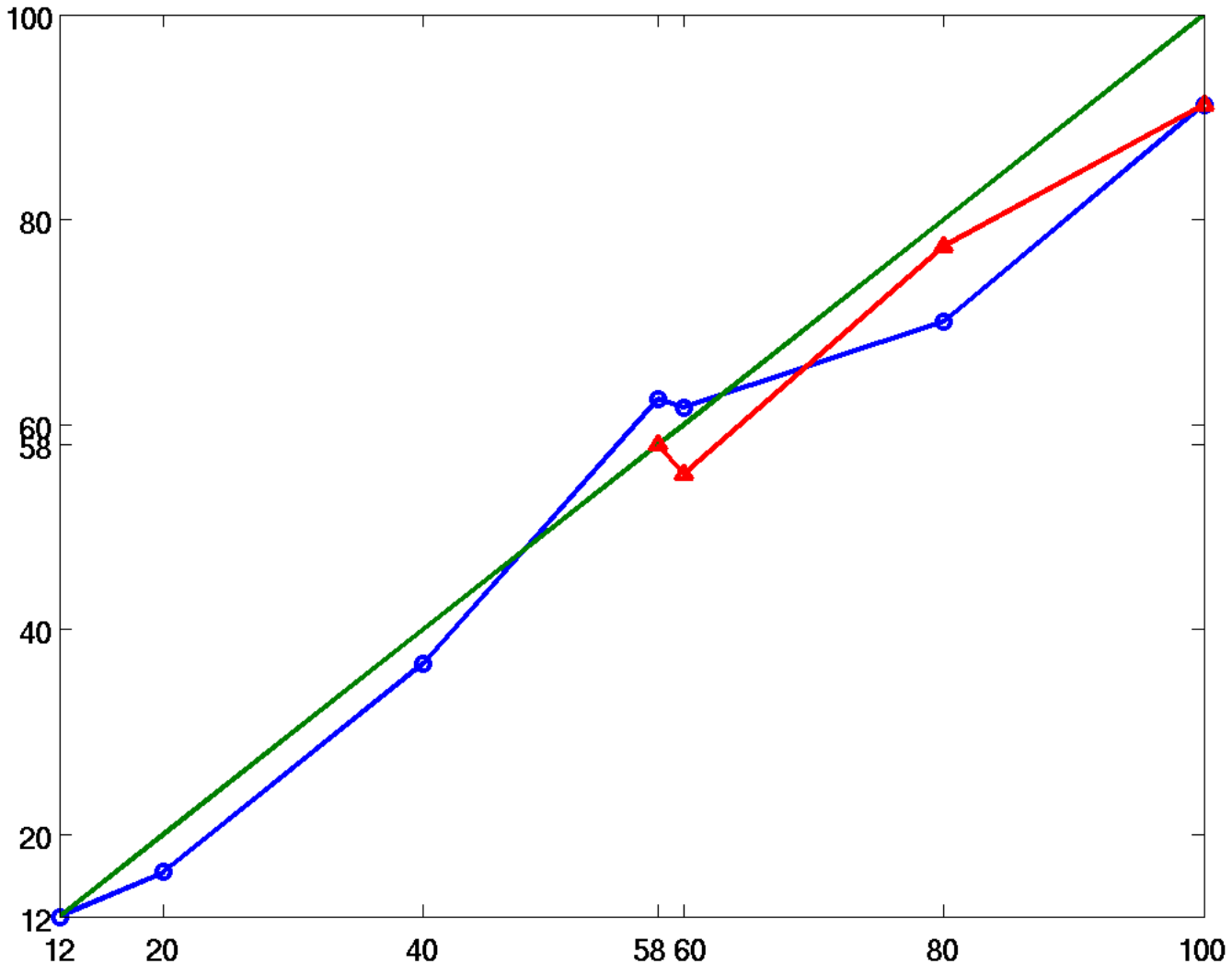
Human bone problems (cont.)

Fixed problem size $n = 14'523'162$.

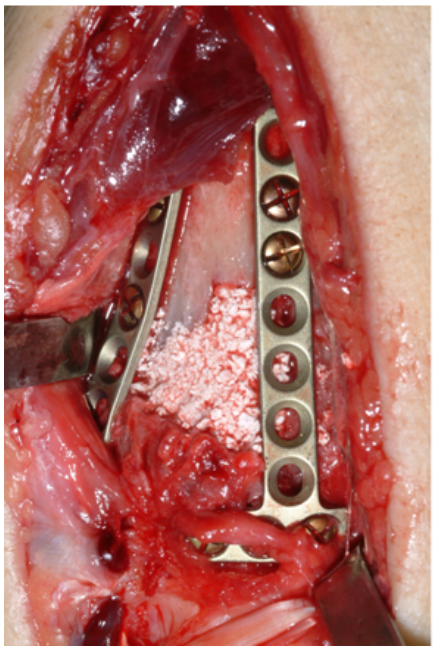
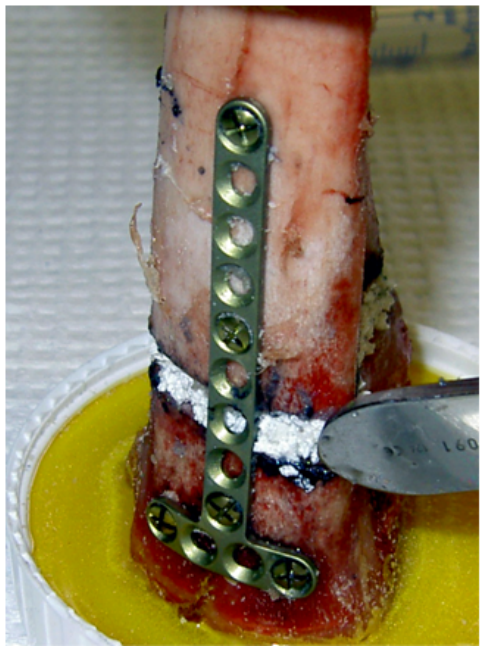
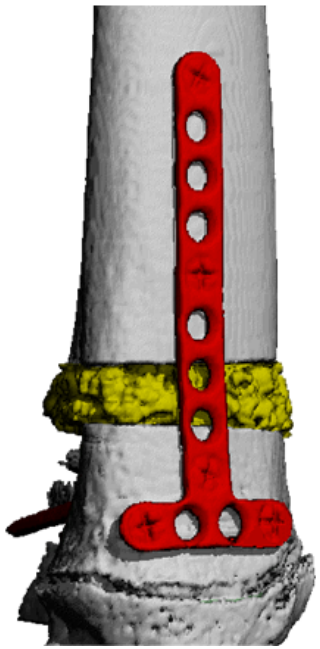
$p = 12$	$p = 20$	$p = 40$	$p = 58$	$p = 60$	$p = 80$	$p = 100$
†	†	†	110.4	116.2	82.7	70.2
951.6	699.6	311.3	182.8	185.3	163.1	125.2

Total CPU time in seconds required to solve the problem using matrix-ready (top) and matrix-free preconditioners (bottom) on p processors. The symbol † indicates failure to run because of lack of memory.

Human bone problems (cont.)

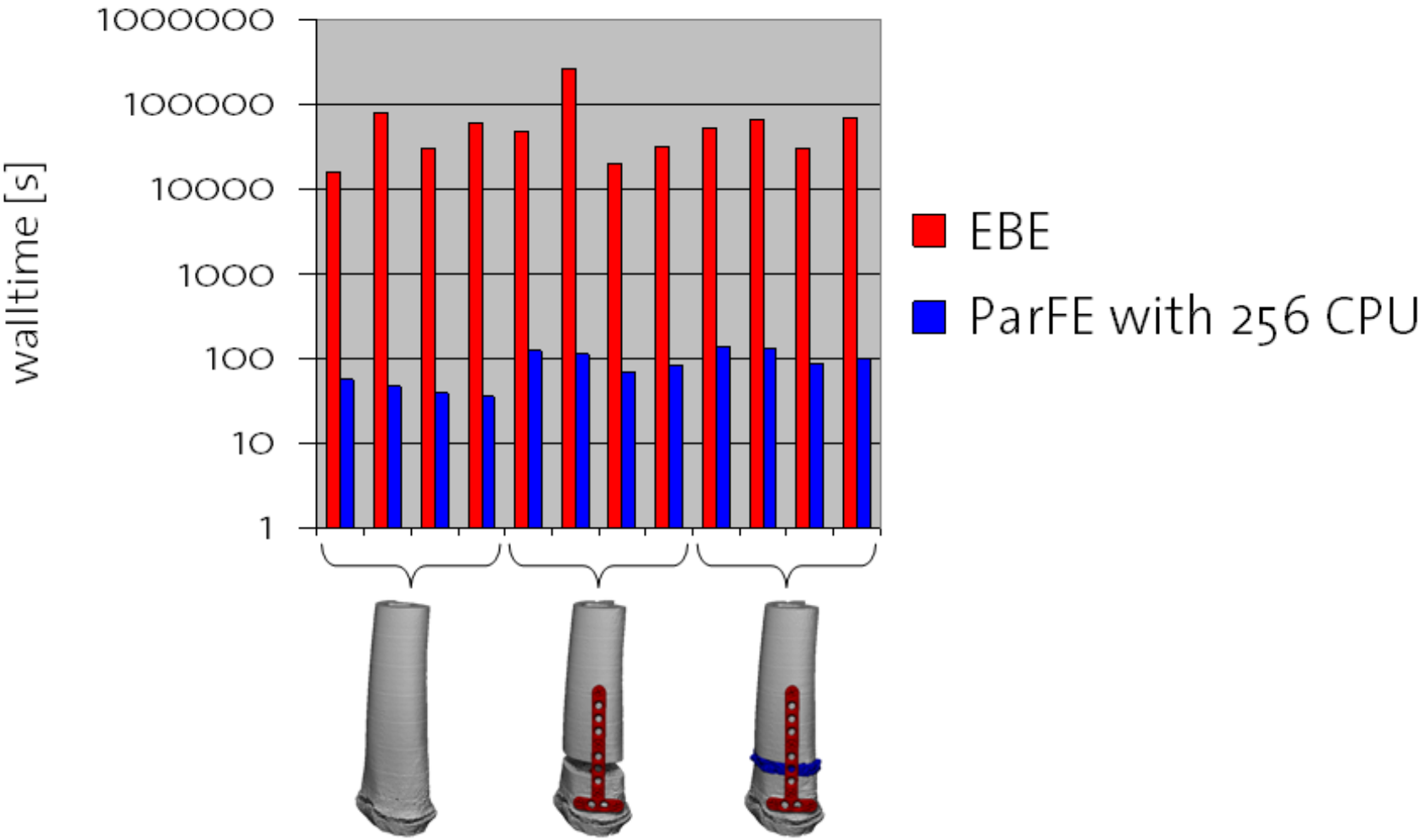


Biomechanics problem: Stability of bone-implant constructs

Bone grafts and biomaterials are often used to aid the repair of complicated fractures.
Simplified *ex vivo* model: ovine radii, *post mortem* implanted T-plates, cortical screws and artificial bone graft.

Biomechanics problem: Stability of bone-implant constructs (cont.)



Conclusions

- Our C++ code, ParFE, is a parallel highly scalable FE solver for bone structure analysis based on PCG with aggregation multilevel preconditioners, see http://parfe.sourceforge.net/

- The algorithm scales quite well on the Cray XT5.
- For processor numbers > 1000 partitioning becomes a problem with ParMETIS.
- RCB comes to our rescue.
- Dense coarse problems must be avoided. Limitation of the number of levels helps. The coarsest problem must be solved iteratively.
- With sufficient hardware, problems with hundreds of millions of degrees of freedom can be solved in a few ($\sim$10) minutes.

References


M. F. Adams, H. H. Bayraktar, T. M. Keaveny, and P. Papadopoulos: *Ultrascalable implicit finite element analyses in solid mechanics with over a half a billion degrees of freedom*. ACM/IEEE Proceedings of SC2004: High Performance Networking and Computing, 2004.
12


P. Arbenz, H. van Lenthe, U. Mennel, R. Müller, and M. Sala. A Scalable Multi-level Preconditioner for Matrix-Free μ-Finite Element Analysis of Human Bone Structures. Internat. J. Numer. Methods in Engrg., 73(7):927–947 (2007).
13

References (cont.)


B. van Rietbergen, H. Weinans, R. Huiskes, and B. J. W. Polman: *Computational strategies for iterative solutions of large FEM applications employing voxel data*. Internat. J. Numer. Methods Engrg. 39 (16): 2743-2767 (1996).
11


P. Vaněk, M. Brezina, and J. Mandel. Algebraic multigrid based on smoothed aggregation for second and fourth order problems. *Computing*, 56(3):179–196, 1996. doi:10.1007/BF02238511
12