A Scalable Multi-level Preconditioner for Matrix-Free μ-Finite Element Analysis of Human Bone Structures

Peter Arbenz[1]

[1]Institute of Computational Science, ETH Zürich,

Comput. Methods with Applications, Harrachov, Aug 20–24, 2007

Coworkers

- Institute of Computational Science, ETH Zürich
  - Uche Mennel
  - Marzio Sala
  - Cyril Flaig
- Institute for Biomechanics, ETH Zürich
  - Harry van Lenthe
  - Ralph Müller
  - Andreas Wirth
- IBM Research Division, Zürich Research Lab
  - Costas Bekas
  - Alessandro Curioni

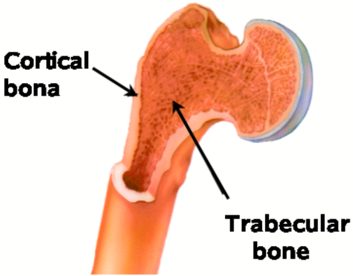
Outline of the talk

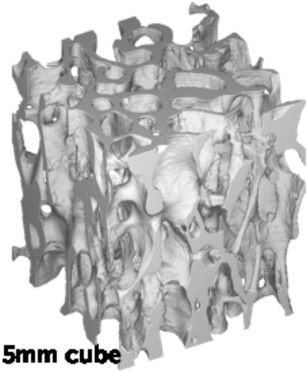
1. μFE Modeling of Trabecular Bone Structures
2. The Mathematical Model
3. Solving the system of equations
4. Algebraic multilevel preconditioning
5. Numerical experiments
6. Conclusions

The need for μFE analysis of bones

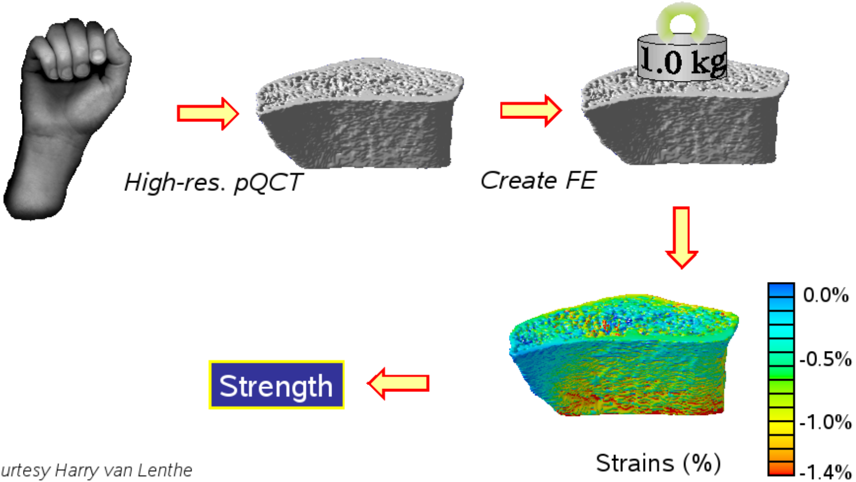
- *Osteoporosis* is disease characterized by low bone mass and deterioration of bone microarchitecture.
- Lifetime risk for osteoporotic fractures in women is estimated close to 40%; in men risk is 13%
- Enormous impact on individual, society and health care systems (as health care problem second only to cardiovascular diseases)
- Since global parameters like bone density do not admit to predict the fracture risk, patients have to be treated in a more individual way.
- Today's approach consists of combining 3D high-resolution CT scans of individual bones with a micro-finite element (μFE) analysis.

Cortical vs. trabecular bone





In vivo assessment of bone strength



Courtesy Harry van Lenthe
University and ETH Zurich

pQCT: Peripheral Quantitative Computed Tomography

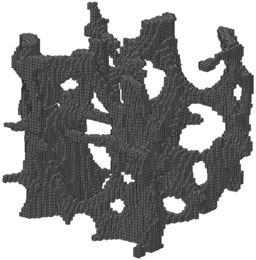
The mathematical model

- Equations of linearized 3D elasticity (pure displacement formulation): Find displacement field **u** that minimizes total potential energy

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

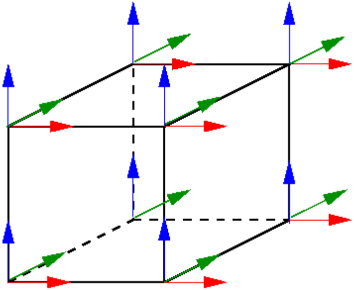
with Lamé's constants λ, μ, volume forces **f**, boundary tractions **g**, symmetric strain tensor

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



- Domain Ω is a union of voxels

Discretization using μFE



- Voxel has 8 nodes/vertices
- In each node we have 3 degrees of freedom: displacements in (x-, y-, z-direction)
- In total 24 degrees of freedom
- Finite element approximation: displacements $\mathbf{u}$ represented by piecewise trilinear polynomials
- strains / stresses computable by means of nodal displacements

Solving the system of equations I

- System of equation

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

  A is large (actually HUGE) sparse, symmetric positive definite.
- 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 II

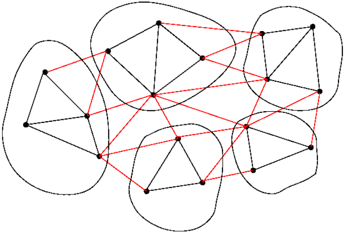
- Our new approach: pcg which smoothed aggregation AMG preconditioning
  (It is known that this works, see Adams et al. [3])
- Requires assembling K
- Parallelization for distributed memory machines
- Employ software: Trilinos (Sandia Nat'l Lab)
  In particular we use
  - Distributed (multi)vectors and (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)

Setup procedure for an abstract multigrid solver

```
Define the number of levels, $L$
for level $\ell = 0, \dots, L-1$ do
    if $\ell < L-1$ then
        Define prolongator $P_\ell$;
        Define restriction $R_\ell = P_\ell^T$;
        $K_{\ell+1} = R_\ell K_\ell P_\ell$;
        Define smoother $S_\ell$;
    else
        Prepare for solving with $K_\ell$;
    end if
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!
     Typical operator complexity for SA: 1.4 (!!!)



Smoothed aggregation (SA) AMG preconditioner II

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

     $$B_\ell = \begin{bmatrix} B_1^{(\ell)} \\ \vdots \\ B_{n_{\ell+1}}^{(\ell)} \end{bmatrix}$$

     $B_j^{(\ell)}$ = rows of B_ℓ corresponding to grid points assigned to j^{th} 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

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

     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: $P_\ell = \widetilde{P}_\ell$

5. Smoother S_ℓ: polynomial smoother
   - Choose a Chebyshev polynomial that is small on the upper part of the spectrum of K_ℓ (Adams, Brezina, Hu, Tuminaro, 2003).
   - Parallelizes perfectly, quality independent of processor number.

'Matrix-free' multigrid

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

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

- In our implementation: P_0 is not smoothed.
- Matrices $K_1, K_2, \ldots$ are formed.
- All graphs, including $\mathcal{G}_0$ are constructed.
- Memory savings (crude approximation): $\frac{1.4}{0.4} = 3.5$
- Clever formation of K_1.

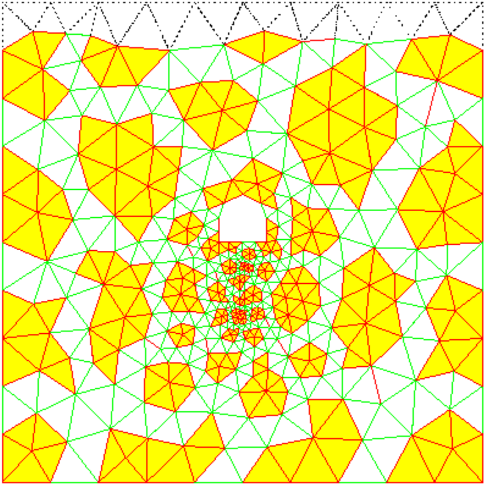
Procedure I

1. Definition of the aggregates on $\mathcal{G}_0$.
2. Definition of the (tentative) prolongator P_0. This requires the aggregates defined in step 1, and the 'near null space'.
3. Computation of the (i, j) block-elements of K_1 for non-smoothed aggregation:

$$K_1(i,j) = \Phi_i^T \, K_0 \, \Phi_j,$$

where Φ_i is the i-th block column of P_0.
If two Φ_j and Φ_k are "far-away", we can group them together in a $\Phi' = \Phi_j + \Phi_k$, then compute $K_0 \Phi'$ with one matvec

Procedure II

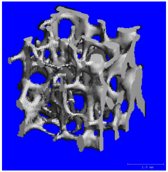


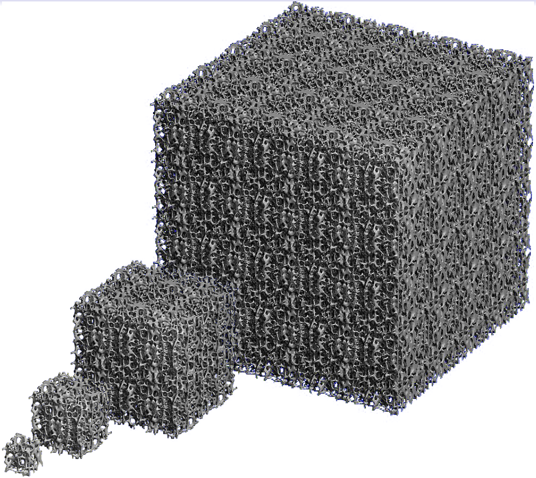
Courtesy Radim Blaheta, U. of Ostrava

Procedure III

4. Building K_1:
   - Construct (in parallel) the graph $\mathcal{G}_1$ of K_1, by working on $\mathcal{G}_0$
   - Color $\mathcal{G}_1$ using (parallel) distance-2 coloring
   - Apply K_0 to all Φ_j belonging to the same color
   - Fewer colors for non-smoothed aggregation (typically from 15 to 25 colors)
5. Smoother for level 0:
   - Chebyshev polynomials
   - need to determine $D_0 = diag(K_0)$ with a distance-1 coloring

Weak scalability test





Problem size scales with the number of processors.
Computations done on Cray XT3 at Swiss National Supercomputer Center (CSCS) and on IBM Blue Gene/L at Zürich Research Lab

Weak scalability test: problem sizes

name	elements	nodes	matrix rows	file size (MB)
c01	98'381	60'482	295'143	9
c02	774'717	483'856	2'324'151	74
c03	2'609'611	1'633'014	7'828'833	250
c04	6'164'270	3'870'848	18'492'810	593
c05	12'038'629	7'560'250	36'115'887	1'157
c06	20'766'855	13'064'112	62'300'565	1'859
c07	32'983'631	20'745'326	98'950'893	3'172
c08	49'180'668	30'966'784	147'542'004	4'732
c09	70'042'813	44'091'378	210'128'439	6'737
c10	96'003'905	60'482'000	288'011'715	9'235
c12	104'512'896	165'834'762	497'504'286	15'953
c14	165'962'608	263'271'435	789'814'305	25'327
c15	204'126'750	323'887'399	971'662'197	31'155
c16	247'734'272	392'912'120	1'178'736'360	37'798

Weak scalability of plain ML preconditioning (Cray XT3)

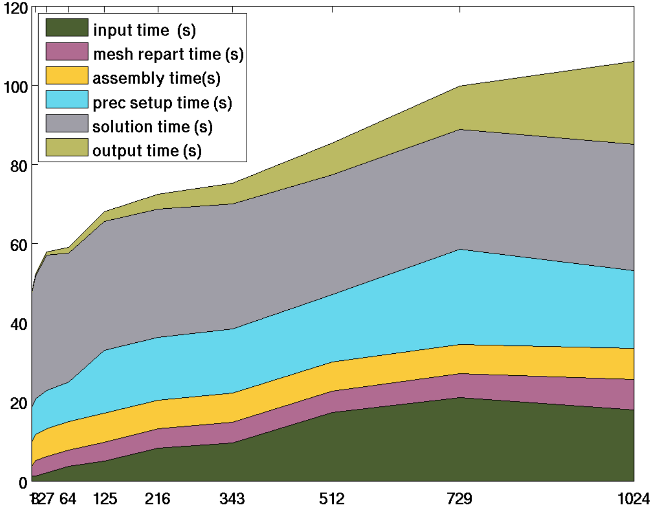
CPUs	input	repart.	assembly	precond.	solution	output	total	iters
1	1.25	2.28	6.25	8.58	28.9	0.10	47.3	51
8	1.27	3.84	6.64	9.03	31.0	0.52	52.3	53
27	2.00	4.18	7.03	9.67	34.2	0.78	57.9	56
64	3.65	4.20	7.12	10.1	32.6	1.33	58.9	53
125	5.03	4.78	7.26	15.9	32.7	2.33	68.0	52
216	8.23	4.92	7.26	15.9	32.3	3.81	72.5	51
343	9.58	5.27	7.38	16.1	31.6	5.25	75.2	49
512	17.3	5.39	7.29	17.0	30.2	8.03	85.3	47
729	21.0	6.18	7.36	24.0	30.2	11.0	99.8	45
1000	17.9	7.68	7.76	19.8	31.8	21.0	106.0	45

Problem size $n \approx \#$ CPUs $\times$ 295’143

Convergence criterion: $\|\mathbf{b} - A\mathbf{x}_k\| \leq 10^{-5}\|\mathbf{b} - A\mathbf{x}_0\| = 10^{-5}\|\mathbf{b}\|$.

Measurements by Uche Mennel (Inst. Comput. Science, ETH Zurich)

Weak scalability of plain ML preconditioning (cont'd)



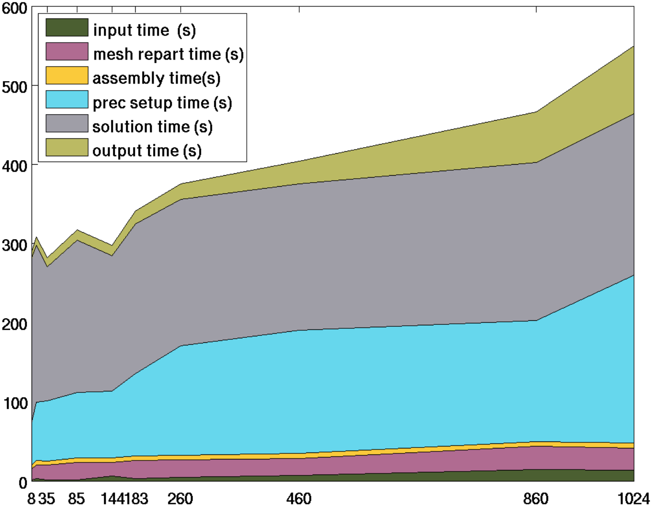
Weak scalability of matrix-free preconditioning (Cray XT3)

name	CPUs	t_{prec}	t_{solve}	t_{total}	n_{it}	χ	m_{prec}
c02	8	52.7	207.9	306.1	66	15	459
c04	16	73.5	198.4	415.6	58	16	437
c05	35	76.0	170.0	356.8	53	16	474
c07	85	82.1	192.4	436.9	53	17	505
c08	144	84.9	170.7	404.7	53	18	480
c09	183	104.0	188.9	476.5	52	16	517
c10	260	137.9	185.5	466.3	53	17	487
c12	460	155.6	185.6	479.9	53	18	507
c15	860	152.6	199.8	608.0	53	17	516
c16	1024	212.2	203.9	725.0	53	17	444

Convergence criterion: $\|\mathbf{b} - A\mathbf{x}_k\| \leq 10^{-5}\|\mathbf{b} - A\mathbf{x}_0\| = 10^{-5}\|\mathbf{b}\|$.

Measurements by Cyril Flaig (Inst. Comput. Science, ETH Zurich)

Matrix-free weak scalability (cont'd)



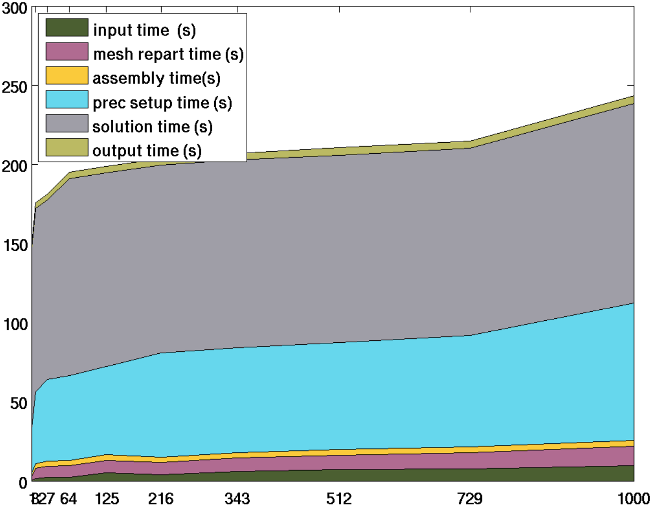
Weak scalability of matrix-free preconditioning (Blue Gene/L)

CPUs	input	repart.	assembly	precond.	solution	output	total	iters
1	0.33	2.50	1.60	27.5	113	1.80	149	94
8	1.40	6.60	3.00	45.2	116	3.50	179	86
27	2.30	7.10	3.20	51.5	113	3.80	185	80
64	2.40	7.10	3.30	53.6	124	4.00	199	86
125	5.20	7.60	3.70	55.7	122	4.00	202	81
216	3.72	8.00	3.42	65.6	119	4.10	207	79
343	5.81	8.60	3.50	66.0	119	4.20	211	77
512	7.12	9.10	3.60	67.5	118	4.75	214	75
729	7.50	10.40	3.60	70.5	118	4.64	216	74
1000	9.78	12.03	3.67	87.0	126	4.70	248	77

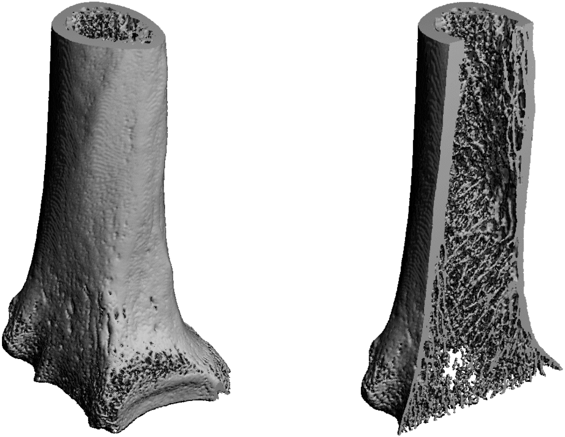
Convergence criterion: $\|\mathbf{b} - A\mathbf{x}_k\| \leq 10^{-5}\|\mathbf{b} - A\mathbf{x}_0\| = 10^{-5}\|\mathbf{b}\|$.

Measurements by Costas Bekas (IBM Research Zurich)

Matrix-free weak scalability on BG/L (cont'd)



Human bone problems



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

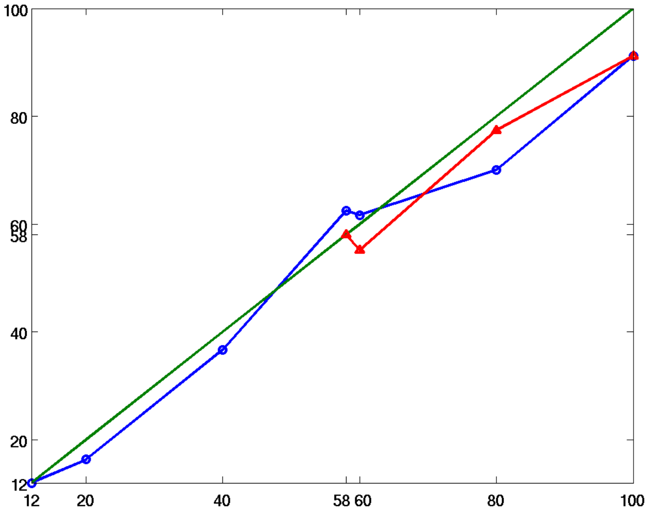
Human bone problems (cont'd)

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'd)



Upshot on algebraic multigrid for μFE problems

1. If enough memory: assemble K and use "standard" smoothed aggregation with Chebyshev or symmetric Gauss-Seidel smoothers, diameter-3 aggregates
2. If not enough memory: prepare K to be applied with EBE approaches, use matrix-free multigrid with Chebyshev smoother for level 0, use aggressive coarsening (50 to 200 nodes per aggregate on level 0)

Both approaches available through ML; see

M. Gee, C. Siefert, J. Hu, R. Tuminaro, and M. Sala:
ML 5.0 Smoothed Aggregation User's Guide.
Sandia National Laboratories Report SAND2006-2649.
(http://software.sandia.gov/trilinos/packages/ml)

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/
- On the CRAY XT3, all phases but the I/O scale very well
- For $\gg$ 1000 processors, ParMETIS computes imbalanced partitions that can cause memory problems (as tested on 4K cpus on BG/L)
- Smoothed aggregation preconditioner not too sensitive to jumps in coefficients. (Results from problem sets not shown)
- The 200M degrees of freedom test is solved in less than 100 seconds on the Cray XT3
- The 1 billion degrees of freedom test is solved in about 12 minutes using pcg with matrix-free AMG preconditioning.

References I

[1] P. Arbenz, U. Mennel, H. van Lenthe, 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. (2007), doi:10.1002/nme.2101.

[2] Scalable Parallel Algebraic Multigrid Preconditioners: http://software.sandia.gov/trilinos/packages/ml

[3] 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. See http://www.sc-conference.org/sc2004/schedule/pdfs/pap111.pdf.

[4] 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