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

Peter Arbenz$^1$

$^1$Institute of Computational Science, ETH Zürich,

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. $\mu$FE 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 $\mu$FE 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 ($\mu$FE) analysis.
Cortical vs. trabecular bone
In vivo assessment of bone strength

High-res. pQCT → Create FE

Strength

Strains (%)

pQCT: Peripheral Quantitative Computed Tomography

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

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

with Lamé's constants $\lambda, \mu$, volume forces $f$, boundary tractions $g$, symmetric strain tensor

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

- Domain $\Omega$ is a union of voxels
Discretization using $\mu$FE

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

- System of equation

\[ Kx = b \]

- \( A \) 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
  \[
  K = \sum_{e=1}^{n_{el}} T_e K_e T_e^T, \tag{1}
  \]

  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

1: Define the number of levels, \( L \)
2: \textbf{for} level \( \ell = 0, \ldots, L - 1 \) \textbf{do}
3: \quad \textbf{if} \( \ell < L - 1 \) \textbf{then}
4: \quad \quad Define prolongator \( P_\ell \);
5: \quad \quad Define restriction \( R_\ell = P_\ell^T \);
6: \quad \quad \( K_{\ell+1} = R_\ell K_\ell P_\ell \);
7: \quad \quad Define smoother \( S_\ell \);
8: \quad \textbf{else}
9: \quad \quad Prepare for solving with \( K_\ell \);
10: \quad \textbf{end if}
11: \textbf{end for}
Smoothed aggregation (SA) AMG preconditioner I

1. Build adjacency graph \( G_0 \) of \( K_0 = K \).
   (Take 3 \( \times \) 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 (!!!)
Define a grid transfer operator:

- Low-energy modes, in our case, the rigid body modes (near-kernel) are ‘chopped’ according to aggregation:

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

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

\[ B_\ell = \tilde{P}_\ell B_{\ell+1}, \quad \tilde{P}_\ell^T \tilde{P}_\ell = I_{n_{\ell+1}}, \]

with

\[ \tilde{P}_\ell = \text{diag}(Q_1^{(\ell)}, \ldots, Q_{n_{\ell+1}}^{(\ell)}) \]

and

\[ B_{\ell+1} = \begin{bmatrix} R_1^{(\ell)} \\ \vdots \\ R_{n_{\ell+1}}^{(\ell)} \end{bmatrix}. \]

Columns of \( B_{\ell+1} \) span the near kernel of \( K_{\ell+1} \).

- Notice: matrices \( K_\ell \) are not used in constructing tentative prolongators \( \tilde{P}_\ell \), near kernels \( B_\ell \), and graphs \( G_\ell \).
4 For elliptic problems, it is advisable to perform an additional step, to obtain smoothed aggregation (SA).

\[
P_\ell = (I_\ell - \omega_\ell D_\ell^{-1} K_\ell) \tilde{P}_\ell, \quad \omega_\ell = \frac{4/3}{\lambda_{\text{max}}(D_\ell^{-1} K_\ell)},
\]

smoothed prolongator

In non-smoothed aggregation: \( P_\ell = \tilde{P}_\ell \)

5 Smoother \( S_\ell \): polynomial smoother
  - Choose a Chebyshev polynomial that is small on the upper part of the spectrum of \( K_\ell \) (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

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

- In our implementation: $P_0$ is not smoothed.
- Matrices $K_1, K_2, \ldots$ are formed.
- All graphs, including $G_0$ 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 $G_0$.

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 $\Phi_i$ is the $i$-th block column of $P_0$.

If two $\Phi_j$ and $\Phi_k$ 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 $G_1$ of $K_1$, by working on $G_0$
   - Color $G_1$ using (parallel) distance-2 coloring
   - Apply $K_0$ to all $\Phi_j$ 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 $D_0 = \text{diag}(K_0)$ 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

<table>
<thead>
<tr>
<th>name</th>
<th>elements</th>
<th>nodes</th>
<th>matrix rows</th>
<th>file size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>c01</td>
<td>98'381</td>
<td>60'482</td>
<td>295'143</td>
<td>9</td>
</tr>
<tr>
<td>c02</td>
<td>774'717</td>
<td>483'856</td>
<td>2'324'151</td>
<td>74</td>
</tr>
<tr>
<td>c03</td>
<td>2'609'611</td>
<td>1'633'014</td>
<td>7'828'833</td>
<td>250</td>
</tr>
<tr>
<td>c04</td>
<td>6'164'270</td>
<td>3'870'848</td>
<td>18'492'810</td>
<td>593</td>
</tr>
<tr>
<td>c05</td>
<td>12'038'629</td>
<td>7'560'250</td>
<td>36'115'887</td>
<td>1'157</td>
</tr>
<tr>
<td>c06</td>
<td>20'766'855</td>
<td>13'064'112</td>
<td>62'300'565</td>
<td>1'859</td>
</tr>
<tr>
<td>c07</td>
<td>32'983'631</td>
<td>20'745'326</td>
<td>98'950'893</td>
<td>3'172</td>
</tr>
<tr>
<td>c08</td>
<td>49'180'668</td>
<td>30'966'784</td>
<td>147'542'004</td>
<td>4'732</td>
</tr>
<tr>
<td>c09</td>
<td>70'042'813</td>
<td>44'091'378</td>
<td>210'128'439</td>
<td>6'737</td>
</tr>
<tr>
<td>c10</td>
<td>96'003'905</td>
<td>60'482'000</td>
<td>288'011'715</td>
<td>9'235</td>
</tr>
<tr>
<td>c12</td>
<td>104'512'896</td>
<td>165'834'762</td>
<td>497'504'286</td>
<td>15'953</td>
</tr>
<tr>
<td>c14</td>
<td>165'962'608</td>
<td>263'271'435</td>
<td>789'814'305</td>
<td>25'327</td>
</tr>
<tr>
<td>c15</td>
<td>204'126'750</td>
<td>323'887'399</td>
<td>971'662'197</td>
<td>31'155</td>
</tr>
<tr>
<td>c16</td>
<td>247'734'272</td>
<td>392'912'120</td>
<td>1'178'736'360</td>
<td>37'798</td>
</tr>
</tbody>
</table>
Weak scalability of plain ML preconditioning (Cray XT3)

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

Problem size $n \approx \#\text{ CPUs} \times 295'143$

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)

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

Convergence criterion: $\|b - Ax_k\| \leq 10^{-5}\|b - Ax_0\| = 10^{-5}\|b\|$.

Measurements by Cyril Flaig (Inst. Comput. Science, ETH Zurich)
Matrix-free weak scalability (cont’d)
Weak scalability of matrix-free preconditioning (Blue Gene/L)

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

Convergence criterion: \(\|b - Ax_k\| \leq 10^{-5}\|b - Ax_0\| = 10^{-5}\|b\|\).

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 $n = 14'523'162$.

<table>
<thead>
<tr>
<th>$p$</th>
<th>12</th>
<th>20</th>
<th>40</th>
<th>58</th>
<th>60</th>
<th>80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>†</td>
<td>†</td>
<td>†</td>
<td>110.4</td>
<td>116.2</td>
<td>82.7</td>
<td>70.2</td>
</tr>
<tr>
<td></td>
<td>951.6</td>
<td>699.6</td>
<td>311.3</td>
<td>182.8</td>
<td>185.3</td>
<td>163.1</td>
<td>125.2</td>
</tr>
</tbody>
</table>

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 $\mu$FE 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
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


