Scalable Algebraic Multigrid for Human Trabecular Bone Structures

Marzio Sala†

ETH Zürich
Institute of Computational Science

† : inf Informatik Computer Science

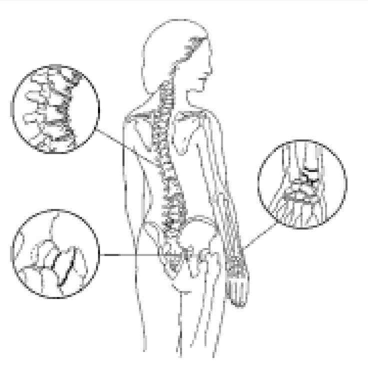
IBM Zürich Research Laboratory, November 8, 2006

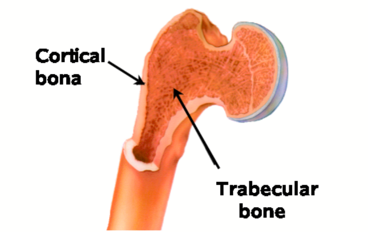
Scalable	§6	$\longrightarrow$	Production code using $\mathcal{O}(1000)$ CPUs
Algebraic	§5	$\longrightarrow$	black-box, no geometric information
Multigrid	§4	$\longrightarrow$	optimal preconditioner
for Human	§2	$\longrightarrow$	applications to real-life problems
Trabecular Bone	§1	$\longrightarrow$	bone type that mostly suffers from osteoporosis
Structures	§3	$\longrightarrow$	data structures arising from CT scans, modeled with $\mu-$Finite Elements (linear elasticity); needs a linear solver

Trabecular Bone

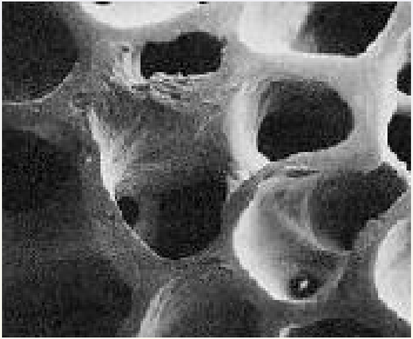
From Wikipedia:

> *Trabecular bone is one of two main types of bone. Trabecular bone is spongy, and makes up the bulk of the interior of most bones, including the vertebrae, while cortical bone is dense and forms the surface of bones.*





Trabecular Bone (cont'd)

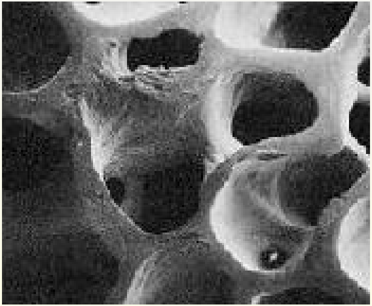


- Bones are living tissues
- structure evolves to adapt to environment
- a disease, *osteoporosis*, can damage the bone

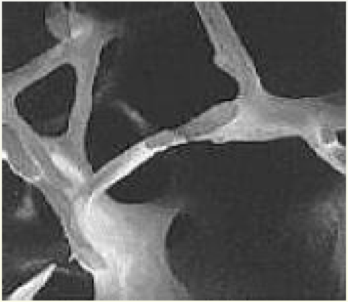
Osteoporosis

- Lifetime risk for osteoporotic fractures in women is estimated close to 40%; in men risk is 13%
- Enormous impact on individual, society, health care systems (as health care problem second only to cardiovascular diseases)

"healthy"



"osteoporotic"



How to estimate that a bone can "carry the load"?

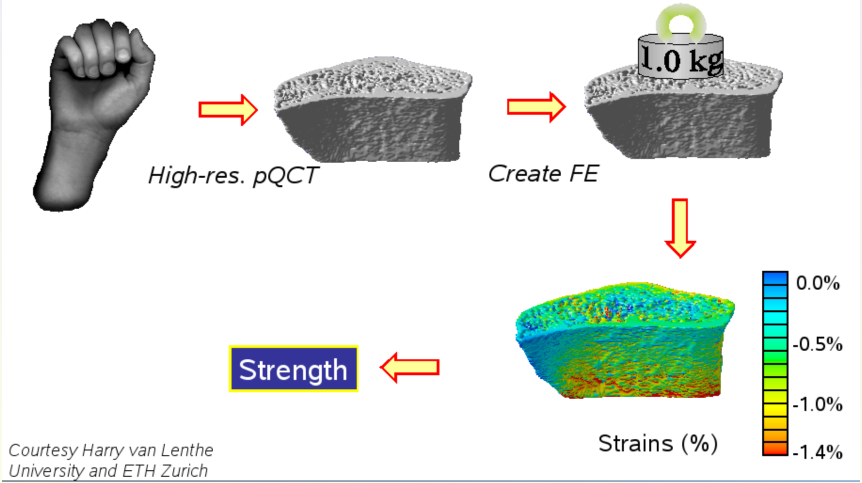
In-vivo Assessment of Bone Strength

Our aim: develop techniques to predict the bone strength, with a strong focus on in-vivo assessment

Current procedure:

- For the clinician, the prediction of bone strength for individual patients is, so far, more or less restricted to the quantitative analysis of bone density alone, although there is convincing evidence that bone microarchitecture plays a significant role as well.
- Since global parameters like bone density do not admit to predict the fracture risk, the microarchitecture has to be taken into account: use a finite element solver

In-vivo Assessment of Bone Strength (cont'd)



pQCT: Peripheral Quantitative Computed Tomography

In-vivo Assessment of Bone Strength (cont'd)

(Some of) required tools:

1. pQCT / high-resolution CT scan
2. filtering software to generate FE meshes
3. a mathematical model for trabecular bones
4. E, ν parameters
5. a scalable, robust and reliable parallel solver
6. data mining techniques
7. last but not least, a high-performance scalable computer or a computational grid

The Mathematical Formulation

- Equations of linear elasticity (weak formulation):
  Find $\mathbf{u} \in [H^1_E(\Omega)]^3 = \{v \in [H^1(\Omega)]^3 : \mathbf{v}_{|\Gamma_D} = \mathbf{u}_S\}$ s.t.

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

  for all $\mathbf{v} \in [H^1_0(\Omega)]^3 = \{v \in [H^1(\Omega)]^3 : \mathbf{v}_{|\Gamma_D} = 0\}$.
  with Lamé's constants λ, μ, volume forces $\mathbf{f}$, boundary tractions $\mathbf{g}$, symmetric strains

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

- Domain Ω is a union of voxels

Discretization using μ-FE

- Finite element approximation: displacements $\mathbf{u}$ represented by piecewise trilinear polynomials, integration using quadrature formula
- Boundary conditions on displacements (Dirichlet)
- After discretization, we have to solve the system of linear equations

$$K\mathbf{x} = \mathbf{b} \qquad \text{or} \qquad (KM)M^{-1}\mathbf{x} = \mathbf{b}$$

  K is large sparse, symmetric positive definite and M a preconditioner.
- kernel also for time-dependent or nonlinear problems
- We use preconditioned conjugate gradient method
- Current solver: use Jacobi or Element-by-Element preconditioner
- need a scalable preconditioner: **multigrid**

A Simple Multigrid V-cycle

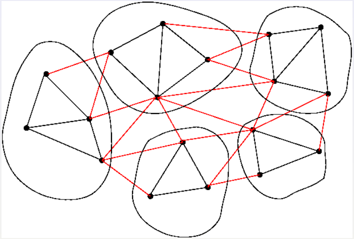
procedure multilevel($K_\ell, \mathbf{b}_\ell, \mathbf{u}_\ell, \ell$)
if $\ell < L$ **then**
  $\mathbf{u}_\ell = S_\ell(K_\ell, \mathbf{b}_\ell, \mathbf{u}_\ell)$; {Presmoothing}
  $\mathbf{r}_{\ell+1} = R_\ell(\mathbf{b}_\ell - K_\ell \mathbf{u}_\ell)$; {Coarse grid correction}
  $\mathbf{v}_{\ell+1} = \mathbf{0}$;
  multilevel($K_{\ell+1}, \mathbf{r}_{\ell+1}, \mathbf{v}_{\ell+1}, \ell + 1$);
  $\mathbf{u}_\ell = \mathbf{u}_\ell + P_\ell \mathbf{v}_{\ell+1}$;
  $\mathbf{u}_\ell = S_\ell(K_\ell, \mathbf{b}_\ell, \mathbf{u}_\ell)$; {Postsmoothing}
else
  Solve $K_L \mathbf{u}_L = \mathbf{b}_L$;
end if

Preconditioner: Call procedure multilevel($K_0 = K, \mathbf{b}, \mathbf{u} = \mathbf{0}, 0$)

Multigrid Techniques

- multigrid methods require the definition of the auxiliary operators P_ℓ, R_ℓ, and K_ℓ
- this can be done with *geometric* or *algebraic* procedures
- we adopt *algebraic multigrid*; two variants:
    - Algebraically coarsen on each level by identifying a set of coarser-level nodes (C-nodes) and finer-level nodes (F-nodes) (Ruge, Stüben, 1987).
    - Algebraically coarsen on each level by grouping the nodes into contiguous subsets, called aggregates, as done in *smoothed aggregation* (SA) (Vanek, Brezina, Mandel, 2001).

Smoothed Aggregation Setup



Define the maximum number of levels, L
for each level ℓ **do**
    if on coarsest level **then**
        Define the coarse solver; Return
    endif
    Define P_ℓ
    $R_\ell = P_\ell^T$
    $K_{\ell+1} = R_\ell K_\ell P_\ell$
    Define the smoother S_ℓ
endfor

To Store or not To Store K*?*

Two approaches:

1. *matrix-ready methods*: K is assembled and fully stored in memory, requiring memory space for about `nnz` floating point numbers. (Multigrid solvers already available.)
2. *matrix-free methods*: K is never assembled, instead

$$K = \sum_e T_e K_e T_e^T$$

   can apply K to a given vector. This is called EBE: element-by-element. (Which preconditioner?)

Matrix-Free Multigrid

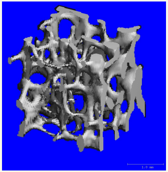
- Matrix-free methods are particularly favorable for problems arising from voxel conversions: all elements generated in the voxel conversion have exactly the same shape, size and orientation.
- All K_e related to a given material are *identical*: required storage is $24 \times 24 \times$ `sizeof(double)` bytes.

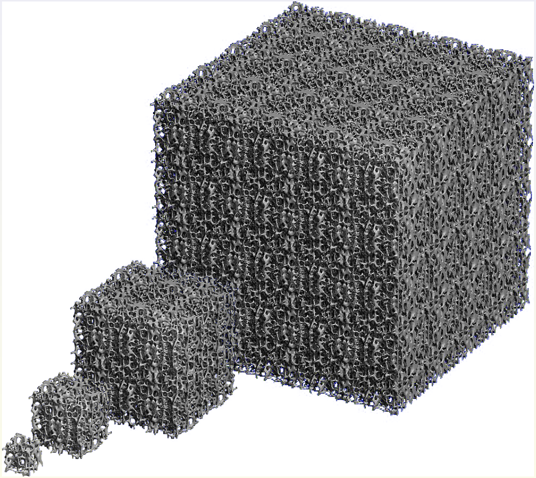
However:

- Few preconditioners are available: Jacobi, EBE, polynomial, ...

We want to define an algebraic multigrid preconditioner that operates in matrix-free environments

Weak scalability test





Problem size scales with the number of processors

Weak scalability test (cont'd)

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 for Matrix-Free

matrix rows	CPUs	prec setup	sol	iters
7'828'833	2	56.33	85.40	21
18'492'810	16	79.37	88.29	20
36'115'887	32	88.66	87.83	20
98'950'893	86	87.44	94.95	21
147'542'004	144	88.64	86.22	21
210'128'439	183	96.72	96.44	21
288'011'715	260	86.711	98.41	22
497'504'286	460	103.25	97.82	22
971'662'197	860	156.49	105.29	22
1'178'736'360	1024	225.71	106.34	22

Algebraic Multigrid for μ-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 (`http://software.sandia.gov/trilinos/packages/ml`); see

ML 5.0 Smoothed Aggregation User's Guide,
M. Gee, C. Siefert, J. Hu, R. Tuminaro, M. Sala

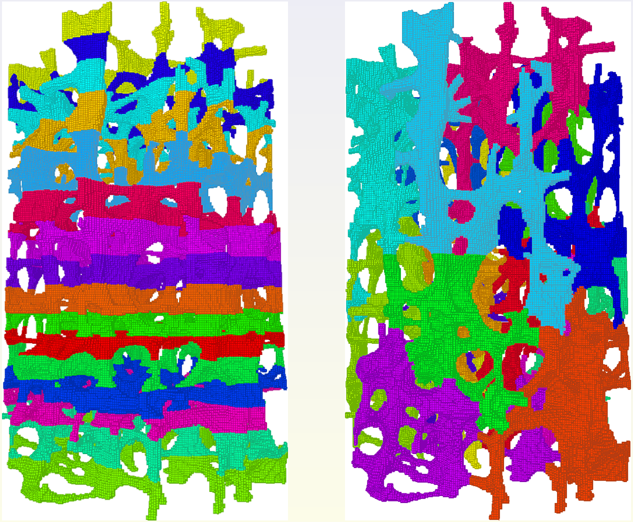
Sandia National Laboratories Report SAND2006-2649.

Our Code: ParFE

`sourceforge.net` project based on

- parallel I/O using HDF5:
  - binary, portable
  - mesh reading scales (reasonably) well with number of processors
- load balance using ParMETIS
- several Trilinos packages:
  - domain decomposition techniques for matrix assembly and linear system solves
  - algebraic multigrid preconditioners

The ParFE *Project (cont'd)*



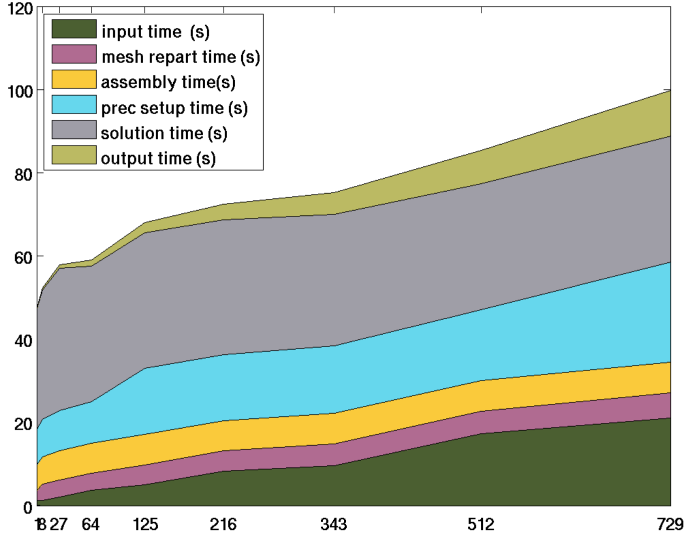
Initial partition (left) based on node coordinates
ParMETIS repartition (right)

Weak Scalability

CPUs	input	repart.	assembly	precond.	solution	output	total	iters
1	1.25	2.28	6.25	8.58	28.86	0.10	**47.32**	51
8	1.27	3.84	6.64	9.03	30.98	0.52	**52.28**	53
27	2.00	4.18	7.03	9.67	34.23	0.78	57.88	56
64	3.65	4.20	7.12	10.05	32.60	1.33	58.94	53
125	5.03	4.78	7.26	15.86	32.71	2.33	67.97	52
216	8.23	4.92	7.26	15.91	32.34	3.81	72.47	51
343	9.58	5.27	7.38	16.09	31.64	5.25	75.21	49
512	17.34	5.39	7.29	17.04	30.24	8.03	85.33	47
729	20.98	6.18	7.36	23.98	30.24	11.05	**99.78**	45

Problem size $n = \#$ CPUs $\times$ 295'143

Weak Scalability (cont'd)



Conclusions

- Our code C++, PARFE, is a parallel highly scalable FE solver for bone structure analysis based on PCG with aggregation multilevel preconditioners
- On the CRAY XT3, all phases but the I/O scale very well
  - Poor scaling of I/O is mostly due to the limited number of available I/O nodes; however
  - time-dependent or non-linear problems less dependent on I/O
- Smoothed aggregation preconditioner not too sensitive to jumps in coefficients
- The 200M degrees of freedom test is solved in less than 100 seconds on the Cray XT3
- the one billion degrees of freedom test is solved in less than 10 minutes using a matrix-free algebraic multigrid

References

My web site: `http://marzio.sala.googlepages.com`

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. Article in preparation.

PARA'06 proceeding:
`http://people.inf.ethz.ch/arbenz/para06.pdf`

Other publications on ParFE:
`http://parfe.sourceforge.net/publications.php`

Scalable Parallel Algebraic Multigrid Preconditioners:
`http://software.sandia.gov/trilinos/packages/ml`