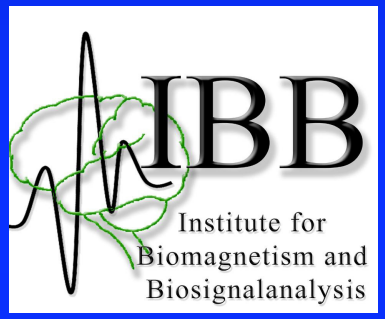
Linear complexity of finite element method based forward modeling in bioelectromagnetism



Carsten Wolters

Institut für Biomagnetismus und Biosignalanalyse, Westf. Wilhelms-Universität Münster, Germany

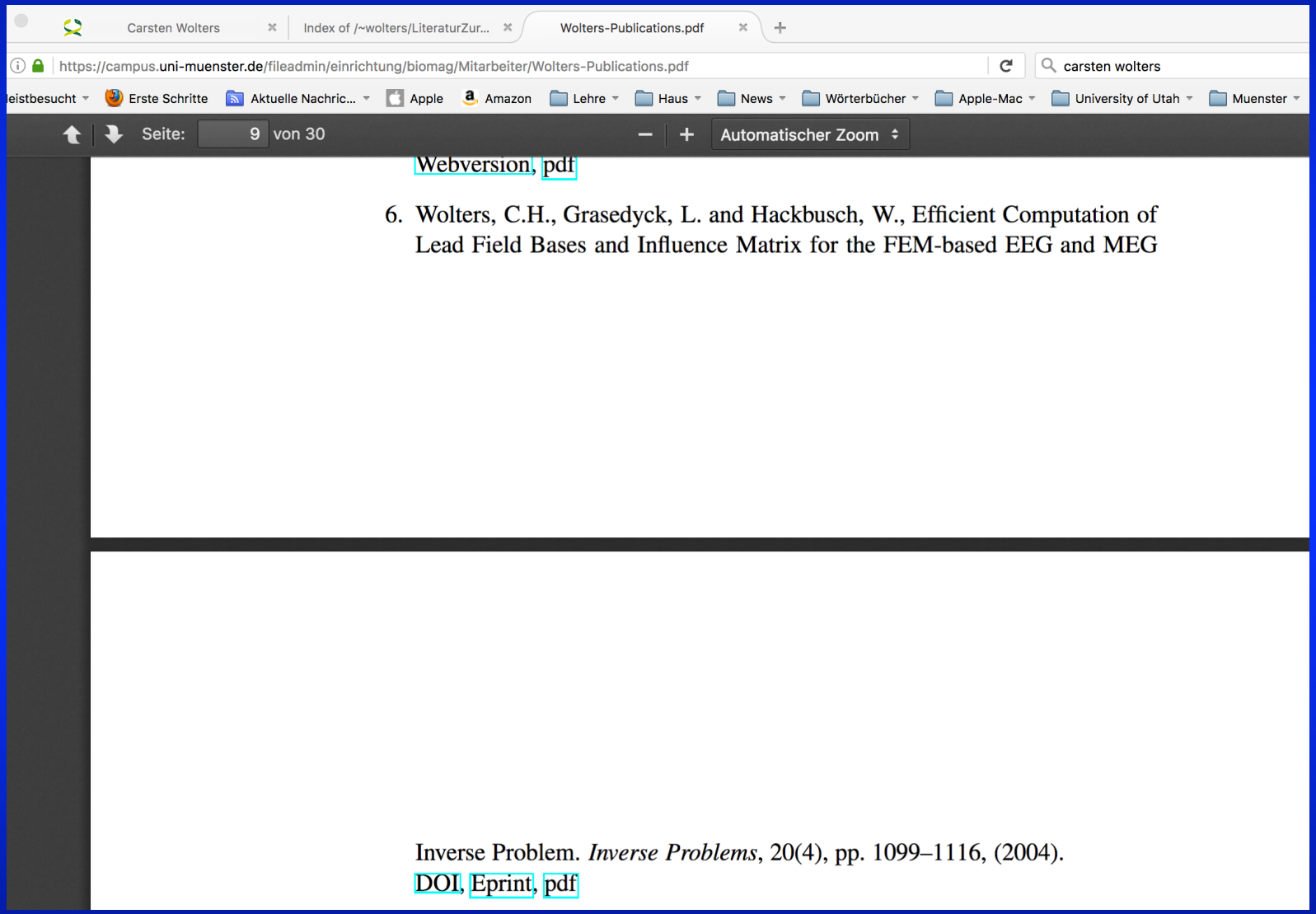
Lecture on Nov.26, 2024

Structure of the lecture

- Linear complexity for FEM forward modeling: Transfer matrices
- Fast FE solver methods

[Wolters, Grasedyck & Hackbusch, *Inverse Problems*, 2004]

Link to own work for transfer matrices:



[Weinstein, Zhukov & Johnson, *Annals Biomed. Eng.*,2000]
[Wolters, Grasedyck & Hackbusch, *Inverse Problems,* 2004]

EEG transfer matrix

$$\begin{aligned} \nabla\cdot(\sigma\nabla\Phi) &= J_p = \nabla\cdot\vec{j}_p \quad in \quad \Omega \\ (\sigma\nabla\Phi)\cdot\vec{n} &= 0 \quad on \quad \Gamma = \partial\Omega \\ \Phi/_{\mathrm{Re}f} &= 0 \end{aligned}$$

$$(*)\mathbf{K}\underline{\Phi} = \underline{J} \qquad \mathbf{K} \in R^{N\times N}$$

$$\underline{\Phi}^{EEG} = \mathbf{R}\underline{\Phi}$$

$$\overset{(*)}{=} \mathbf{R}(\mathbf{K}^{-1}\underline{J})$$

$$= (\mathbf{R}\mathbf{K}^{-1})\underline{J}$$

$$\mathbf{R} = \begin{pmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & \dots & \dots & 0 \\ 1 & 0 & \dots & \dots & \dots & \dots & 0 \end{pmatrix} \in R^{(s^{EEG}-1)\times N}$$

$$T^{EEG} := \mathbf{R}\mathbf{K}^{-1} \in R^{(s^{EEG}-1)\times N}$$

$$T^{EEG} : \underline{J} \rightarrow \underline{\Phi}^{EEG}$$

$s^{EEG} - 1$

N

[Wolters, Grasedyck & Hackbusch, *Inverse Problems,* 2004]

MEG transfer matrix

$$\underline{\Psi}_i = \underline{\Psi}_i^{\mathrm{s}} + \underline{\Psi}_i^{\mathrm{sek}} = \frac{\mu}{4\pi}\int_{\Omega}\left\langle \vec{j}^{\,\mathrm{s}}(\vec{y}), \oint_{\partial F_i}\frac{1}{|\vec{x}-\vec{y}|}\,d\vec{x}\right\rangle d\vec{y} - \frac{\mu}{4\pi}\int_{\Omega}\left\langle \sigma(\vec{y})\nabla\Phi(\vec{y}), \oint_{\partial F_i}\frac{1}{|\vec{x}-\vec{y}|}\,d\vec{x}\right\rangle d\vec{y}$$

[Wolters, Grasedyck & Hackbusch, *Inverse Problems,* 2004]

MEG transfer matrix

$$\underline{\Psi}_i = \underline{\Psi}_i^{\mathrm{s}} + \underline{\Psi}_i^{\mathrm{sek}} = \frac{\mu}{4\pi}\int_{\Omega}\left\langle \vec{j}^{\,\mathrm{s}}(\vec{y}), \oint_{\partial F_i}\frac{1}{|\vec{x}-\vec{y}|}\,d\vec{x}\right\rangle d\vec{y} - \frac{\mu}{4\pi}\int_{\Omega}\left\langle \sigma(\vec{y})\nabla\Phi(\vec{y}), \oint_{\partial F_i}\frac{1}{|\vec{x}-\vec{y}|}\,d\vec{x}\right\rangle d\vec{y}$$

$$\nabla\Phi(\vec{y}) \approx \nabla\left(\sum_{j=1}^{N}\psi_j(\vec{y})\underline{\Phi}_j\right) = \sum_{j=1}^{N}\nabla\psi_j(\vec{y})\underline{\Phi}_j$$

[Wolters, Grasedyck & Hackbusch, *Inverse Problems,* 2004]

MEG transfer matrix

$$\underline{\Psi}_i = \underline{\Psi}_i^{\mathrm{p}} + \underline{\Psi}_i^{\mathrm{sek}} = \frac{\mu}{4\pi}\int_{\Omega}\left\langle \vec{j}^{\,\mathrm{s}}(\vec{y}), \oint_{\partial F_i}\frac{1}{|\vec{x}-\vec{y}|}\,d\vec{x}\right\rangle d\vec{y} - \frac{\mu}{4\pi}\int_{\Omega}\left\langle \sigma(\vec{y})\nabla\Phi(\vec{y}), \oint_{\partial F_i}\frac{1}{|\vec{x}-\vec{y}|}\,d\vec{x}\right\rangle d\vec{y}$$

$$\mathbf{S}_{ij} := -\frac{\mu}{4\pi}\int_{\Omega}\left\langle \sigma(\vec{y})\nabla\psi_j(\vec{y}), \oint_{\partial F_i}\frac{1}{|\vec{x}-\vec{y}|}\,d\vec{x}\right\rangle d\vec{y} \in R^{s^{\mathrm{MEG}}\times N}$$

$$\underline{\Psi}^{\mathrm{sek}} = \mathbf{S}\underline{\Phi} \overset{(*)}{=} \mathbf{S}(\mathbf{K}^{-1}\underline{J}) = (\mathbf{S}\mathbf{K}^{-1})\underline{J}$$

$$T^{\mathrm{MEG}} := \mathbf{S}\mathbf{K}^{-1} \in R^{s^{\mathrm{MEG}}\times N}$$

s^{MEG}

N

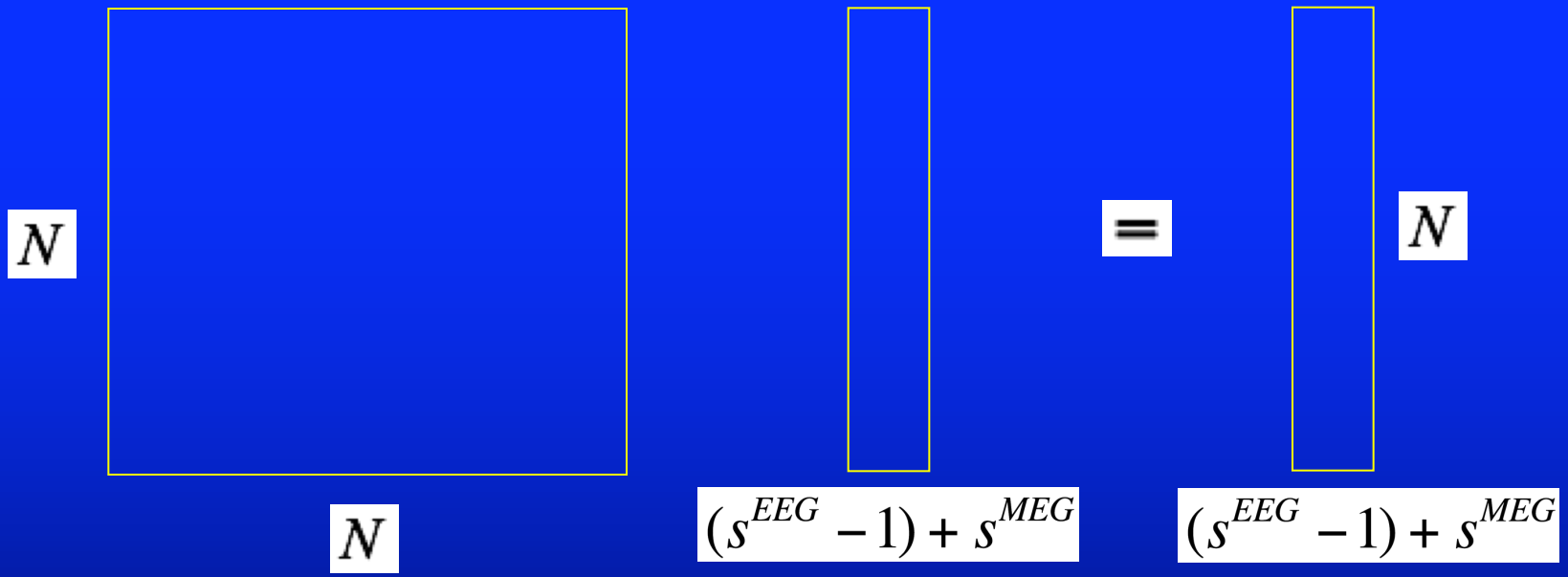
$$T^{\mathrm{MEG}} : \underline{J} \rightarrow \underline{\Psi}^{\mathrm{sek}}$$

[Wolters, Grasedyck & Hackbusch, *Inverse Problems,* 2004]

Computing the transfer matrices

$$\begin{pmatrix} T^{EEG} \\ T^{MEG} \end{pmatrix} := \begin{pmatrix} \mathbf{R} \\ \mathbf{S} \end{pmatrix} \mathbf{K}^{-1} \Leftrightarrow \begin{pmatrix} T^{EEG} \\ T^{MEG} \end{pmatrix} \mathbf{K} = \begin{pmatrix} \mathbf{R} \\ \mathbf{S} \end{pmatrix}$$

$$\Leftrightarrow \mathbf{K} \begin{pmatrix} \left(T^{EEG}\right)^{tr} & \left(T^{MEG}\right)^{tr} \end{pmatrix} = \begin{pmatrix} \mathbf{R}^{tr} & \mathbf{S}^{tr} \end{pmatrix}$$

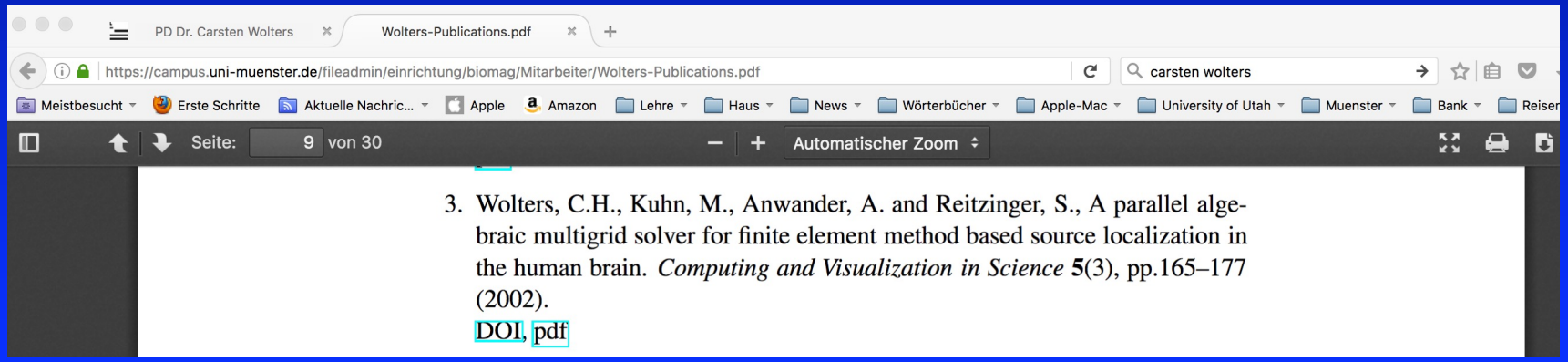


Outline

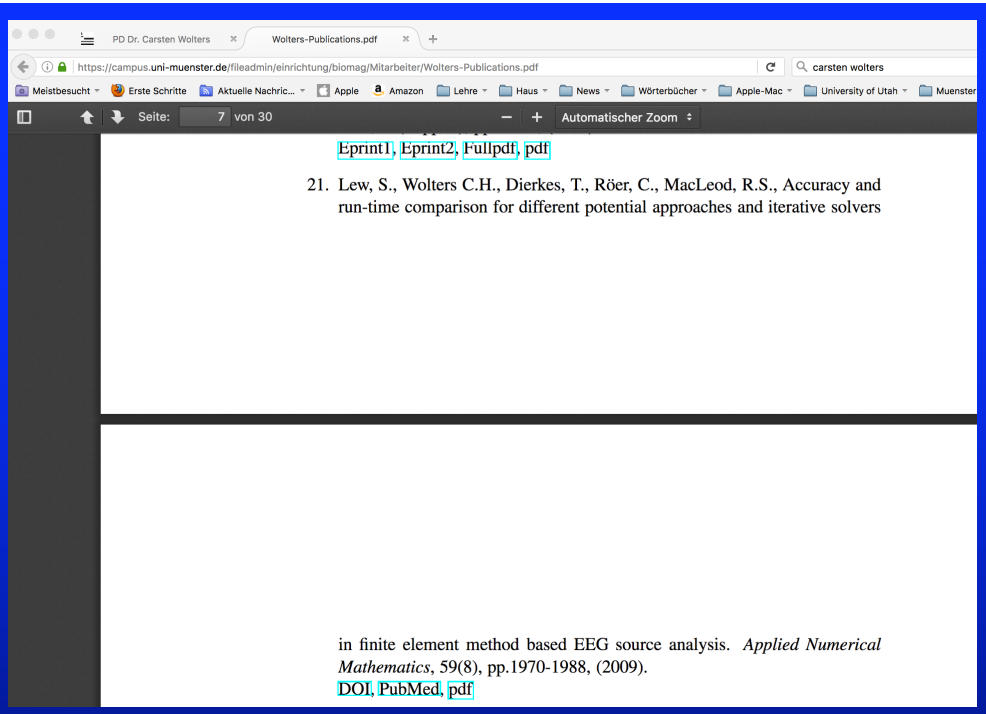
- Linear complexity for FEM forward modeling: Transfer matrices
- Fast FE solver methods

[Wolters, Vorlesungsskriptum, Chapter 7.2]
[Wolters, Kuhn, Anwander & Reitzinger, *Comp. Vis. Sci.*., 2002]
[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

Link to this work:



3. Wolters, C.H., Kuhn, M., Anwander, A. and Reitzinger, S., A parallel algebraic multigrid solver for finite element method based source localization in the human brain. *Computing and Visualization in Science* **5**(3), pp.165–177 (2002).
DOI, pdf



Eprint1, Eprint2, Fullpdf, pdf

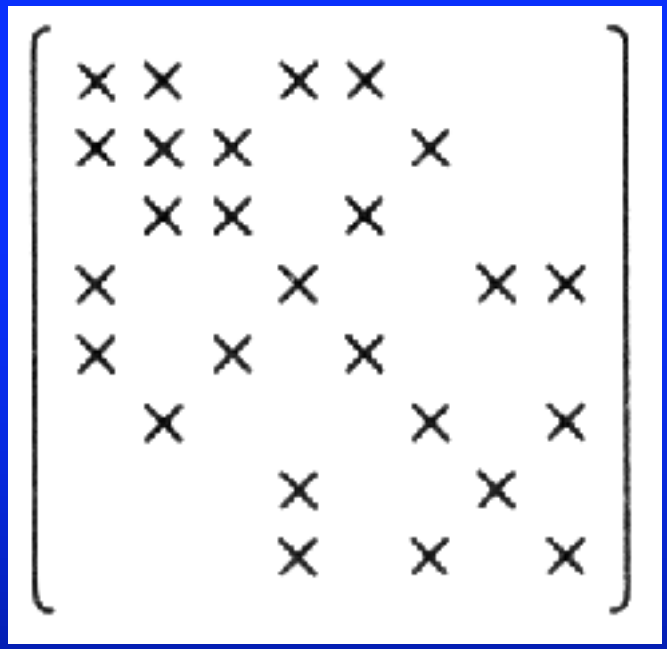
21. Lew, S., Wolters C.H., Dierkes, T., Röer, C., MacLeod, R.S., Accuracy and run-time comparison for different potential approaches and iterative solvers

in finite element method based EEG source analysis. *Applied Numerical Mathematics*, 59(8), pp.1970-1988, (2009).
DOI, PubMed, pdf

[Wolters, Vorlesungsskriptum, Chapter 7.2]
[Wolters, Kuhn, Anwander & Reitzinger, *Comp. Vis. Sci..*, 2002]
[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

FEM solver aspects

$$\mathbf{K}_h \underline{u}_h = \underline{j}_h$$



$$\underline{u}_h = K_h^{-1} \underline{j}_h \; ??$$

[Hackbusch, *Springer*, 1994]

Preconditioned Conjugate Gradient Method

Algorithm 1 PCG : $(K_h, \underline{u}_h, \underline{j}_h, C_h, \text{ACCURACY}) \rightarrow (\underline{u}_h)$

$\underline{r}_h = \underline{r}_h^0 = \underline{j}_h - K_h \underline{u}_h$

SOLVE $C_h \underline{w}_h = \underline{r}_h$

$\underline{s}_h = \underline{w}_h$

$\gamma^0 = \gamma = \gamma_{\text{OLD}} = < \underline{w}_h, \underline{r}_h >$

while $\left(\gamma/\gamma^0 = \left(\frac{||\underline{r}_h||_{C_h^{-1}}}{||\underline{r}_h^0||_{C_h^{-1}}} \right)^2 = \left(\frac{||K_h \underline{e}_h||_{C_h^{-1}}}{||K_h \underline{e}_h^0||_{C_h^{-1}}} \right)^2 = \left(\frac{||\underline{e}_h^i||_{K_h C_h^{-1} K_h}}{||\underline{e}_h^0||_{K_h C_h^{-1} K_h}} \right)^2 > \text{ACCURACY}^2 \right)$ **do**

$\underline{v}_h = K_h \underline{s}_h$

$\alpha = \gamma / < \underline{s}_h, \underline{v}_h >$

$\underline{u}_h = \underline{u}_h + \alpha \underline{s}_h$

$\underline{r}_h = \underline{r}_h - \alpha \underline{v}_h$

SOLVE $C_h \underline{w}_h = \underline{r}_h$

$\gamma = < \underline{w}_h, \underline{r}_h >$

$\beta = \gamma / \gamma_{\text{OLD}}$, $\gamma_{\text{OLD}} = \gamma$

$\underline{s}_h = \underline{w}_h + \beta \underline{s}_h$

end while

PCG accuracy

[Wolters, Vorlesungsskriptum, Chapter 7.2]
[Wolters, Kuhn, Anwander & Reitzinger, *Comp. Vis. Sci..*, 2002]
[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

Iterative solver for FE-equation system

- **FE discretization, meshsize:** h

$$\mathbf{K}_h \underline{u}_h = \underline{j}_h$$

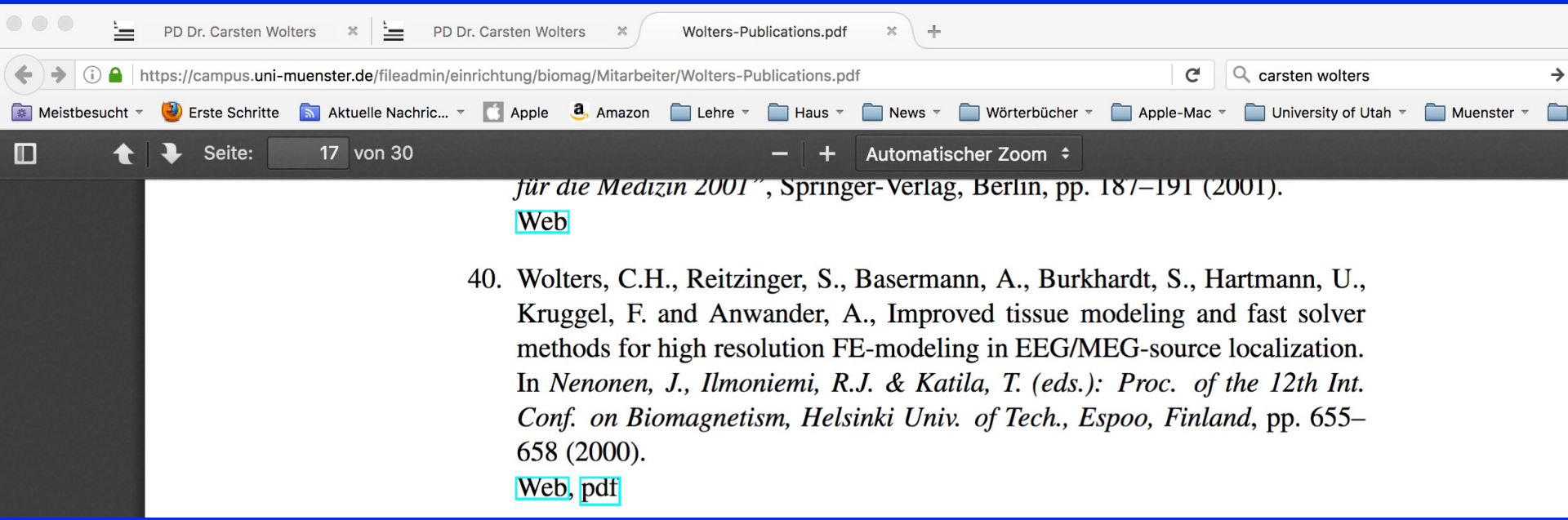
with condition number

$$\kappa(K_h) = \frac{\lambda_h^{\max}}{\lambda_h^{\min}} = \mathrm{O}(h^{-2})$$

- **Example:** $h \approx 2mm$

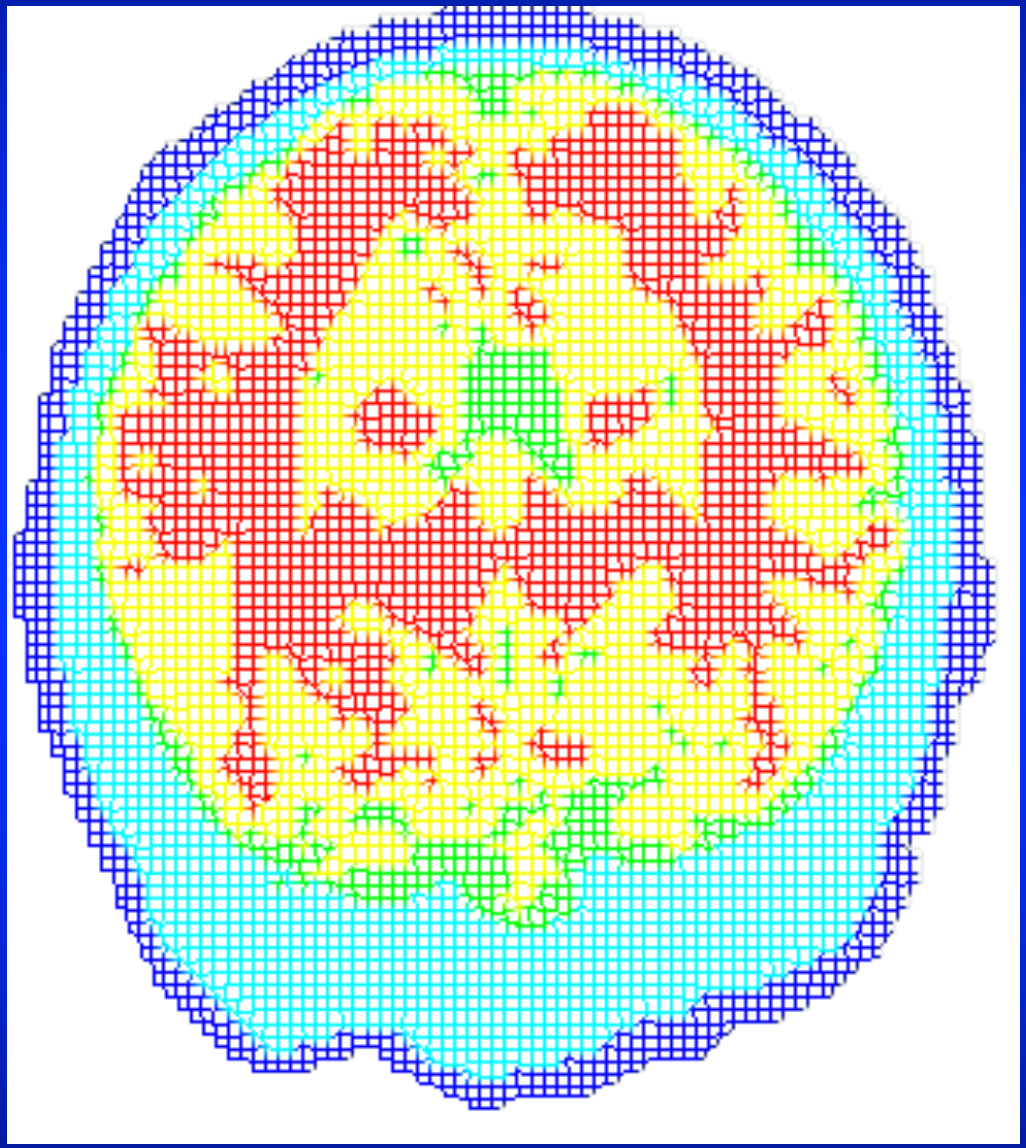
[Wolters, Reitzinger, Basermann, Burkhardt, Hartmann, Kruggel & Anwander, Proceedings of *BIOMAG Helsinki, 2000*]

Link to this work:



[Wolters, Reitzinger, Basermann, Burkhardt, Hartmann, Kruggel & Anwander, Proceedings of *BIOMAG Helsinki, 2000*]

Iterative solver for FE-equation system



[Wolters, Reitzinger, Basermann, Burkhardt, Hartmann, Kruggel & Anwander, Proceedings of *BIOMAG Helsinki, 2000*]

Iterative solver for FE-equation system

- **FE discretization, meshsize:** h

$$\mathbf{K}_h \underline{u}_h = \underline{j}_h$$

with condition number

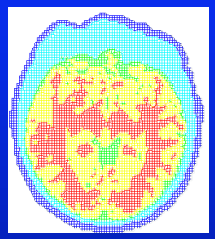
$$\kappa(K_h) = \frac{\lambda_h^{\max}}{\lambda_h^{\min}} = \mathrm{O}(h^{-2})$$

- **Convergence of the CG solver:**

$$\left\| \underline{u}_h^* - \underline{u}_h^{(i)} \right\|_{K_h} \le 2c_h^i \left\| \underline{u}_h^* - \underline{u}_h^{(0)} \right\|_{\mathbf{K}_h}$$

$$c_h = \frac{\sqrt{\kappa(K_h)} - 1}{\sqrt{\kappa(K_h)} + 1}$$

- **Example:** $h \approx 2mm$



$\Rightarrow$ $\kappa(K_h) \approx 10^7$ $\Rightarrow$ $c_h = 0.9993$

Strategy: Preconditioning!

[Wolters, Vorlesungsskriptum, Chapter 7.2]
[Wolters, Kuhn, Anwander & Reitzinger, *Comp. Vis. Sci..*, 2002]
[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

Jacobi preconditioning

Jacobi preconditioning or scaling

The simplest preconditioner is the scaling or Jacobi-preconditioning ([264, pp.265f], [281, pp.257f]), where

$$C_h := D_h^2, \qquad D_h := \mathrm{DIAG}(\sqrt{K_h^{[11]}}, \ldots, \sqrt{K_h^{[N_h N_h]}}).$$

Theorem 6.5.8. *Let K_h be SPD and $C_h := D_h^2$ the Jacobi-preconditioner. Assume that each row of K_h does not contain more than* d *nonzero entries. Then, for all diagonal matrices $\tilde{D}_h^{-1}$, it is*

$$\kappa_2(C_h^{-1} K_h) \leq \mathrm{d}\ \kappa_2(\tilde{D}_h^{-1} K_h),$$

[Wolters, Vorlesungsskriptum, Chapter 7.2]
[Wolters, Kuhn, Anwander & Reitzinger, *Comp. Vis. Sci.*., 2002]
[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

Incomplete Cholesky preconditioning

Incomplete Cholesky preconditioning

The SPD stiffness matrix K_h can be decomposed into a left triangular matrix L_h and its transpose using the Cholesky-decomposition, $K_h = L_h L_h^{tr}$ [281, pp.209f]. Algorithm 19 shows the Cholesky-decomposition.

Algorithm 19 Cholesky decomposition $\mathsf{K}_h = L_h L_h^{tr}$

for $p = 1, \ldots, N_h$ **do**
  $L_h^{[pp]} = \sqrt{\mathsf{K}_h^{[pp]}}$
  for $i = (p+1), \ldots, N_h$ **do**
    $L_h^{[ip]} = \mathsf{K}_h^{[ip]} / L_h^{[pp]}$
    for $k = (p+1), \ldots, i$ **do**
      $\mathsf{K}_h^{[ik]} = \mathsf{K}_h^{[ik]} - L_h^{[ip]} / L_h^{[kp]}$
    end for
  end for
end for

[Wolters, Vorlesungsskriptum, Chapter 7.2]
[Wolters, Kuhn, Anwander & Reitzinger, *Comp. Vis. Sci..*, 2002]
[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

Incomplete Cholesky preconditioning

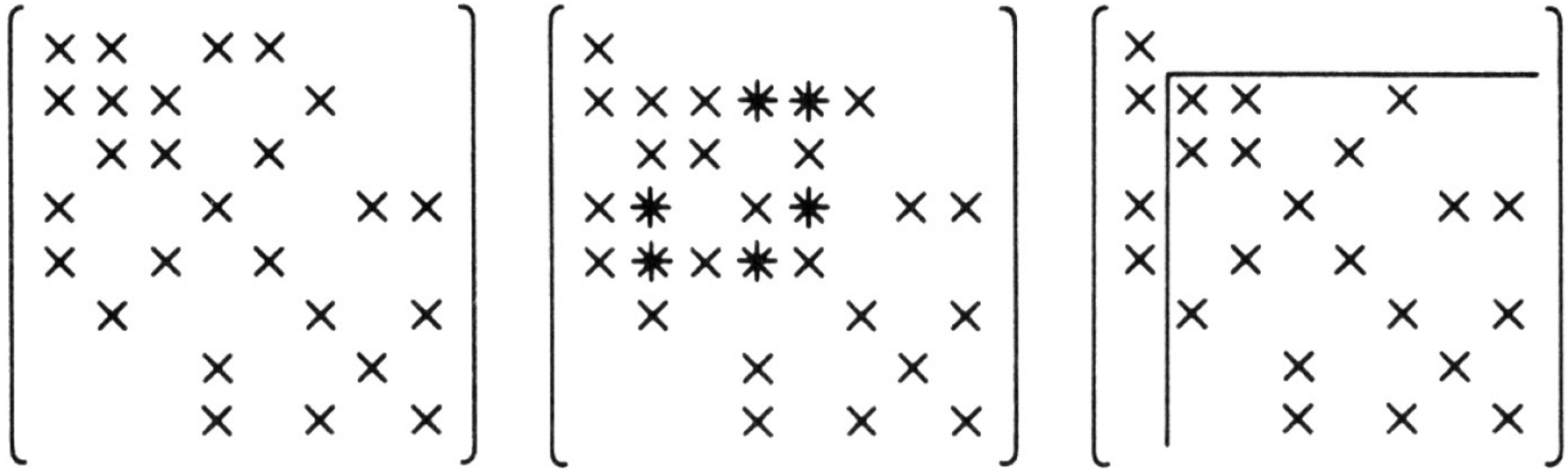


Figure 7.1: *Principle of IC0 (taken from Schwarz [281]): nonzero matrix pattern (left), situation after first step of Algorithm 19 (fill-in is marked with *), situation after the first step of IC0.*

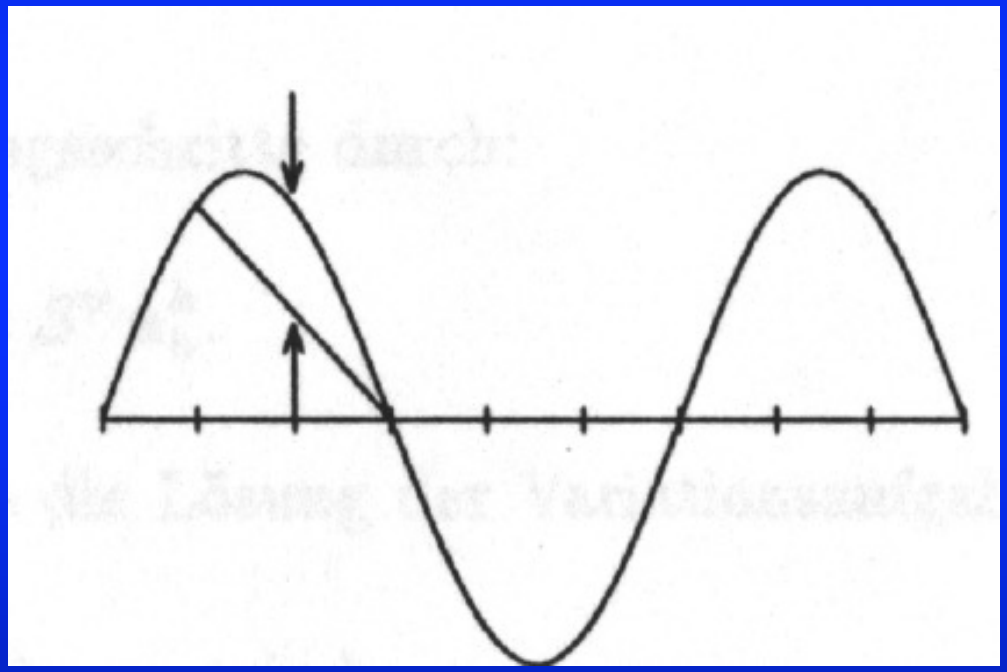
[Hackbusch, *Springer*, 1994]
[Wolters, Vorlesungsskriptum, Chapter 7.2]
[Wolters, Kuhn, Anwander & Reitzinger, *Comp. Vis. Sci.*., 2002]
[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

MultiGrid (MG) preconditioning

Principle of MG

- **Smoother for high-frequency error components is effective**

$$\underline{d}_h = K_h \underline{u}_h - \underline{j}_h$$



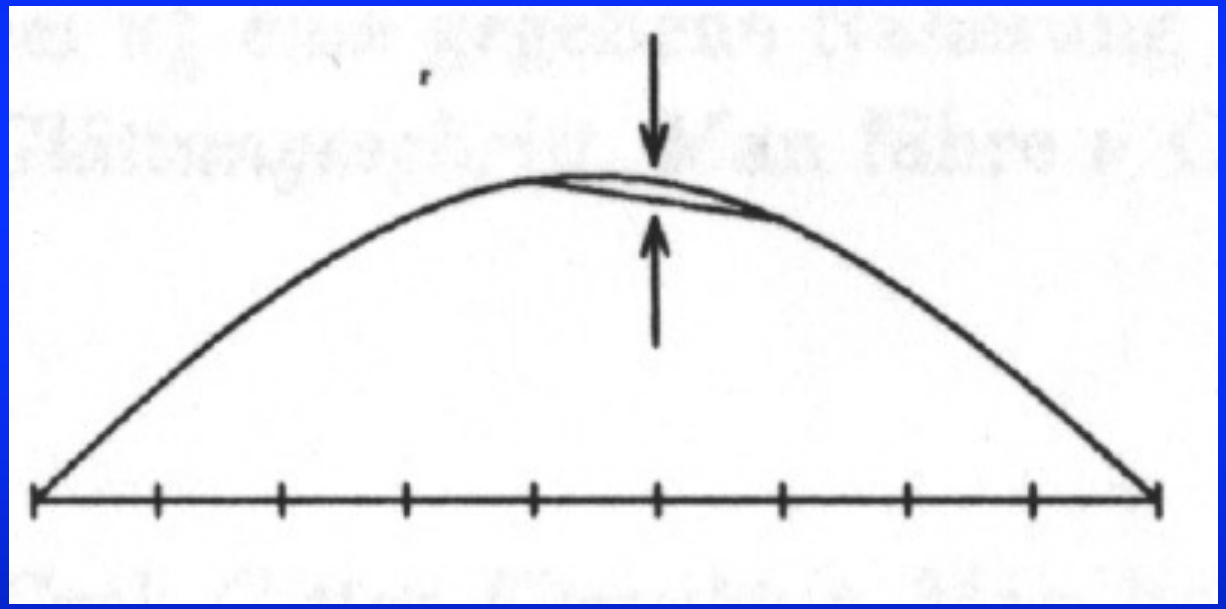
[Hackbusch, *Springer*, 1994]
[Wolters, Vorlesungsskriptum, Chapter 7.2]
[Wolters, Kuhn, Anwander & Reitzinger, *Comp. Vis. Sci.*., 2002]
[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

MultiGrid (MG) preconditioning

Principle of MG

- **Smoother for low-frequency error components not effective**

$$\underline{d}_h = K_h \underline{u}_h - \underline{j}_h$$



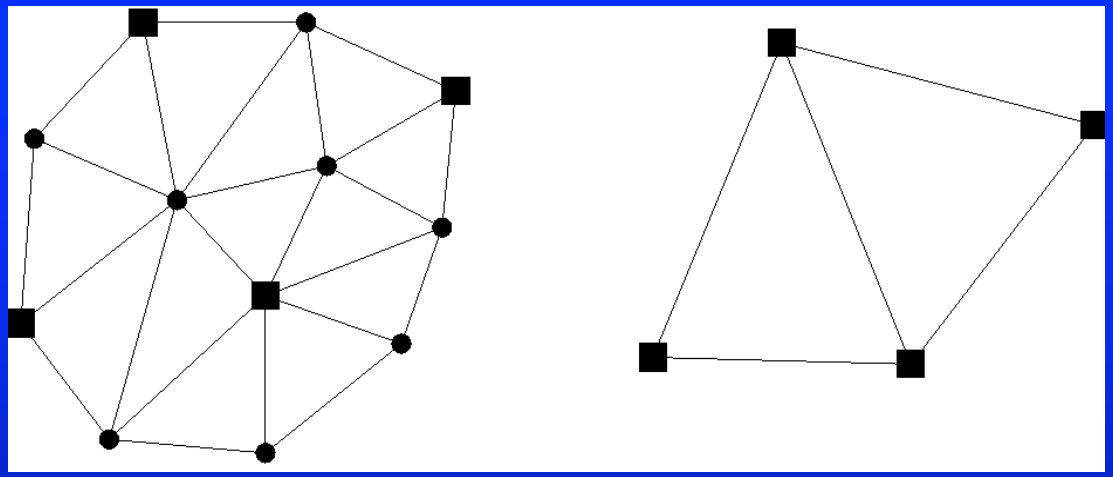
[Hackbusch, *Springer*, 1994]
[Wolters, Vorlesungsskriptum, Chapter 7.2]
[Wolters, Kuhn, Anwander & Reitzinger, *Comp. Vis. Sci.*., 2002]
[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

MultiGrid (MG) preconditioning

Principle of the MG

- Smoother for high-frequency error components
- Coarse grid correction for low-frequency error components

$$\underline{d}_H = P^{tr}_{h,H}\underline{d}_h$$



[Hackbusch, *Springer*, 1994]
[Wolters, Vorlesungsskriptum, Chapter 7.2]
[Wolters, Kuhn, Anwander & Reitzinger, *Comp. Vis. Sci.*., 2002]
[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

MultiGrid (MG) preconditioning

Algorithm 3 V-cycle MG : $(K_h, \underline{u}_h, \underline{j}_h, \nu_F, \nu_B) \rightarrow (\underline{u}_h)$

if CoarseGrid **then**

$\underline{u}_h \leftarrow \text{DirectSolve}(K_h \underline{u}_h = \underline{j}_h)$

else

$\underline{u}_h \leftarrow \nu_F$ TIMES SMOOTH FORWARD$(K_h, \underline{u}_h, \underline{j}_h)$

$\underline{d}_h = K_h \underline{u}_h - \underline{j}_h$

$\underline{d}_H = P^{tr}_{h,H} \underline{d}_h$

$\underline{w}_H = 0$

$\underline{w}_H = \text{MG}(K_H, \underline{w}_H, \underline{d}_H)$

$\underline{w}_h = P_{h,H} \underline{w}_H$

$\underline{u}_h = \underline{u}_h - \underline{w}_h$

$\underline{u}_h \leftarrow \nu_B$ TIMES SMOOTH BACKWARD$(K_h, \underline{u}_h, \underline{j}_h)$

end if

[Hackbusch, *Springer,* 1994]
[Wolters, Vorlesungsskriptum, Chapter 7.2]
[Wolters, Kuhn, Anwander & Reitzinger, *Comp. Vis. Sci..*, 2002]
[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

Geometric Multigrid

- **Geometric MG (GMG)**
  - Problem-specific smoother for high-frequency error components
  - Coarse grid correction for low-frequency error components

  } Complexity:
  • O(N)
  Convergence rate:
  • h-independent
  • high

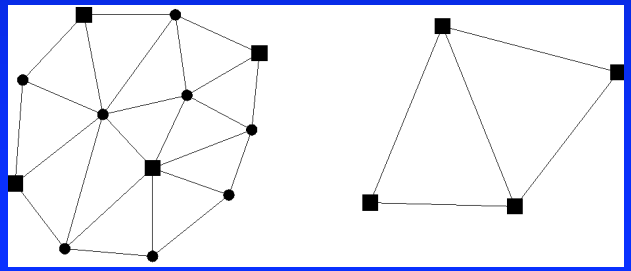
- **Problems in our application**
  - Choice of the smoother is difficult (Inhomogeneities/Anisotropies)
  - Generation of the coarse grids difficult

Strategy: Algebraic MG (AMG)!

Algebraic Multigrid

[Ruge and Stüben, *SIAM,* 1986; Stüben, GMD, Tech.Report, 1999]
[Wolters, Vorlesungsskriptum, Chapter 7.2]
[Wolters, Kuhn, Anwander & Reitzinger, *Comp. Vis. Sci..*, 2002]
[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

- **Principle of the AMG:**
  - Smoother is given (Gauss-Seidel)
  - Coarse grids and interpolation matrices are constructed from the entries of *K*
    - Diagonal entries <-> nodes
    - Nondiagonal entries <-> edges
  - Only one high resolution FE mesh!



- **Problems:**
  - AMG-interpolation non-optimal:
    - Some few error components are not well reduced, i.e., some few eigenvalues of the AMG iteration matrix are close to 1

Strategy: AMG-CG!

[Hackbusch, *Springer*, 1994]
[Wolters, Vorlesungsskriptum, Chapter 7.2]
[Wolters, Kuhn, Anwander & Reitzinger, *Comp. Vis. Sci.*., 2002]
[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

Preconditioned Conjugate Gradient Method

Algorithm 1 PCG : $(K_h, \underline{u}_h, \underline{j}_h, C_h, \text{ACCURACY}) \rightarrow (\underline{u}_h)$

$\underline{r}_h = \underline{r}_h^0 = \underline{j}_h - K_h \underline{u}_h$

SOLVE $C_h \underline{w}_h = \underline{r}_h$

$\underline{s}_h = \underline{w}_h$

$\gamma^0 = \gamma = \gamma_{\text{OLD}} = < \underline{w}_h, \underline{r}_h >$

while $\left(\gamma/\gamma^0 = \left(\frac{||\underline{r}_h||_{C_h^{-1}}}{||\underline{r}_h^0||_{C_h^{-1}}} \right)^2 = \left(\frac{||K_h \underline{e}_h||_{C_h^{-1}}}{||K_h \underline{e}_h^0||_{C_h^{-1}}} \right)^2 = \left(\frac{||\underline{e}_h^i||_{K_h C_h^{-1} K_h}}{||\underline{e}_h^0||_{K_h C_h^{-1} K_h}} \right)^2 > \text{ACCURACY}^2 \right)$ **do**

$\underline{v}_h = K_h \underline{s}_h$

$\alpha = \gamma / < \underline{s}_h, \underline{v}_h >$

$\underline{u}_h = \underline{u}_h + \alpha \underline{s}_h$

$\underline{r}_h = \underline{r}_h - \alpha \underline{v}_h$

SOLVE $C_h \underline{w}_h = \underline{r}_h$

$\gamma = < \underline{w}_h, \underline{r}_h >$

$\beta = \gamma / \gamma_{\text{OLD}}$, $\gamma_{\text{OLD}} = \gamma$

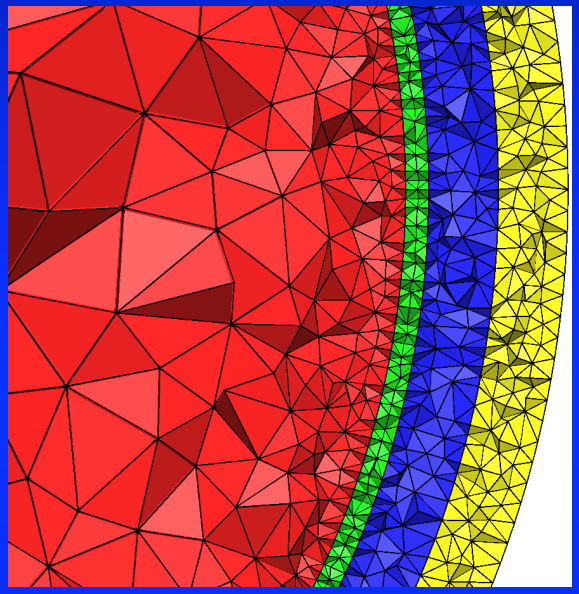
$\underline{s}_h = \underline{w}_h + \beta \underline{s}_h$

end while

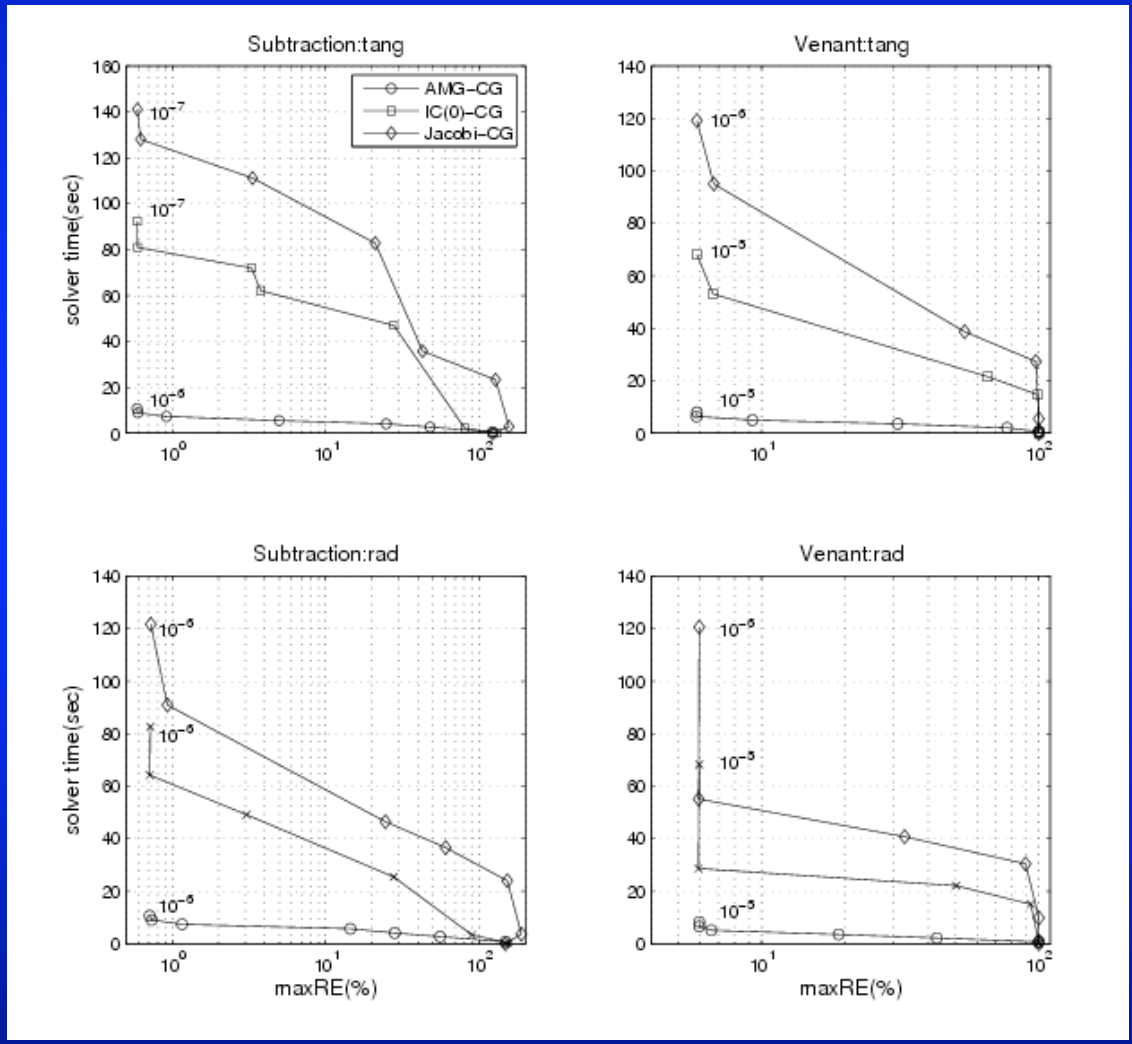
PCG accuracy

[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

Solver: "Maximal relative error over all eccentricities" versus "solution time"

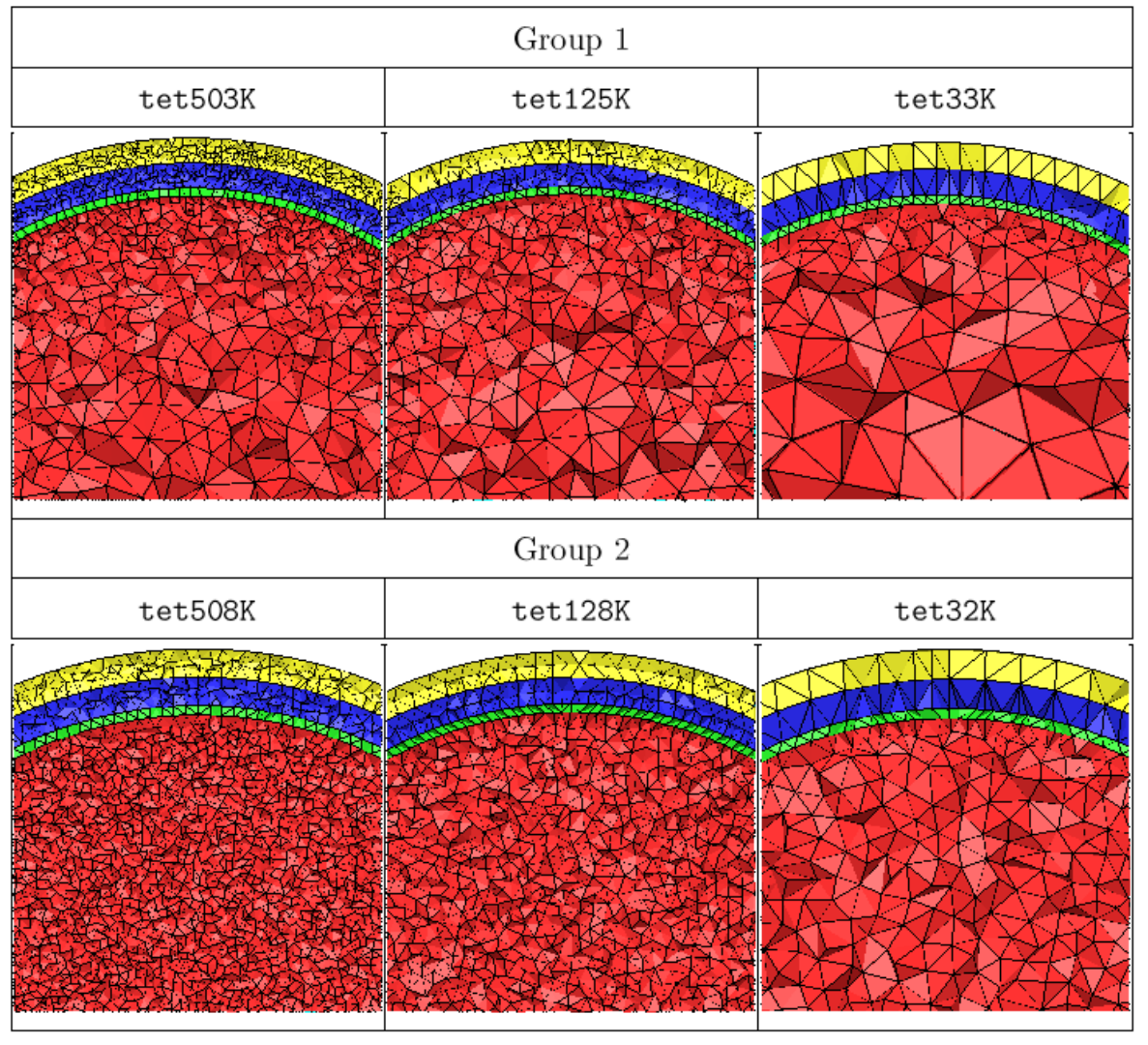


Nodes: 360,056
Elements: 2,165,281



[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

FE models tuned for subtraction (group 1) and for the direct potential methods (group 2)



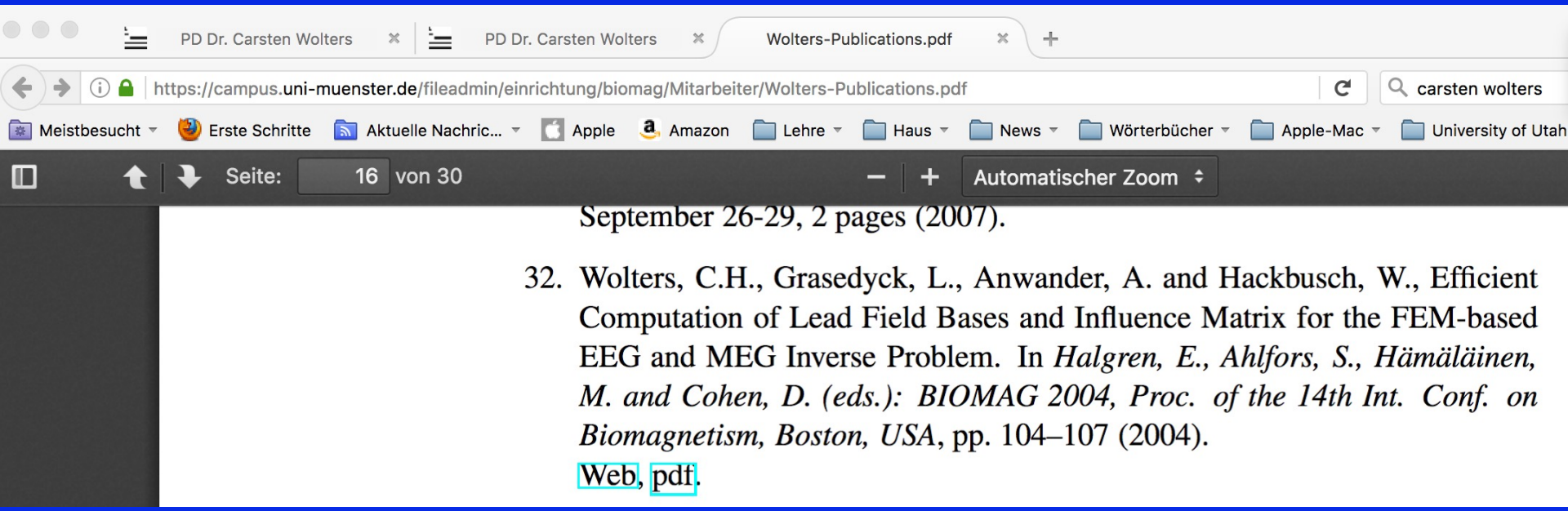
[Lew, Wolters, Dierkes, Roer & MacLeod, *Appl.Num.Math.*, 2009]

Comparison of different preconditioners

	group 1						group 2					
	tet503K		tet125K		tet33K		tet508K		tet128K		tet32K	
solver	time	iter	time	iter	time	iter	time	iter	time	iter	time	iter
AMG-CG	12.25	11.20	1.87	9.04	0.18	5.89	9.18	10.40	1.36	7.27	0.15	5.81
IC(0)-CG	112.03	233.43	8.40	128.39	0.45	72.63	72.41	215.05	5.20	98.96	0.31	52.84
Jacobi-CG	167.82	679.43	16.98	414.00	0.76	229.52	99.60	578.04	9.62	331.15	0.47	161.68
gain factor	13.70	60.66	9.08	45.8	4.22	38.97	10.85	55.58	7.07	45.55	3.13	27.83

[Wolters, Grasedyck, Anwander & Hackbusch, *Biomag*, 2004]

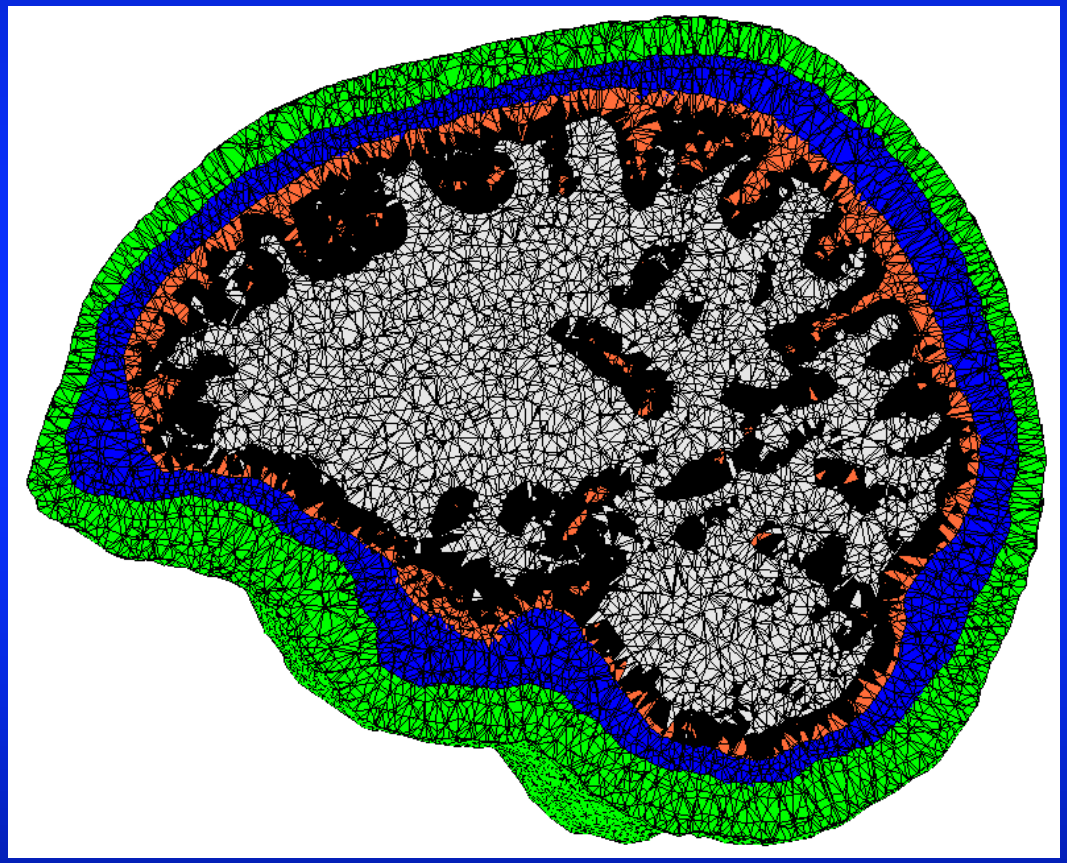
Link to this work:



[Wolters, Grasedyck, Anwander & Hackbusch, *Biomag*, 2004]

Efficiency of the fast FE transfer matrix approaches

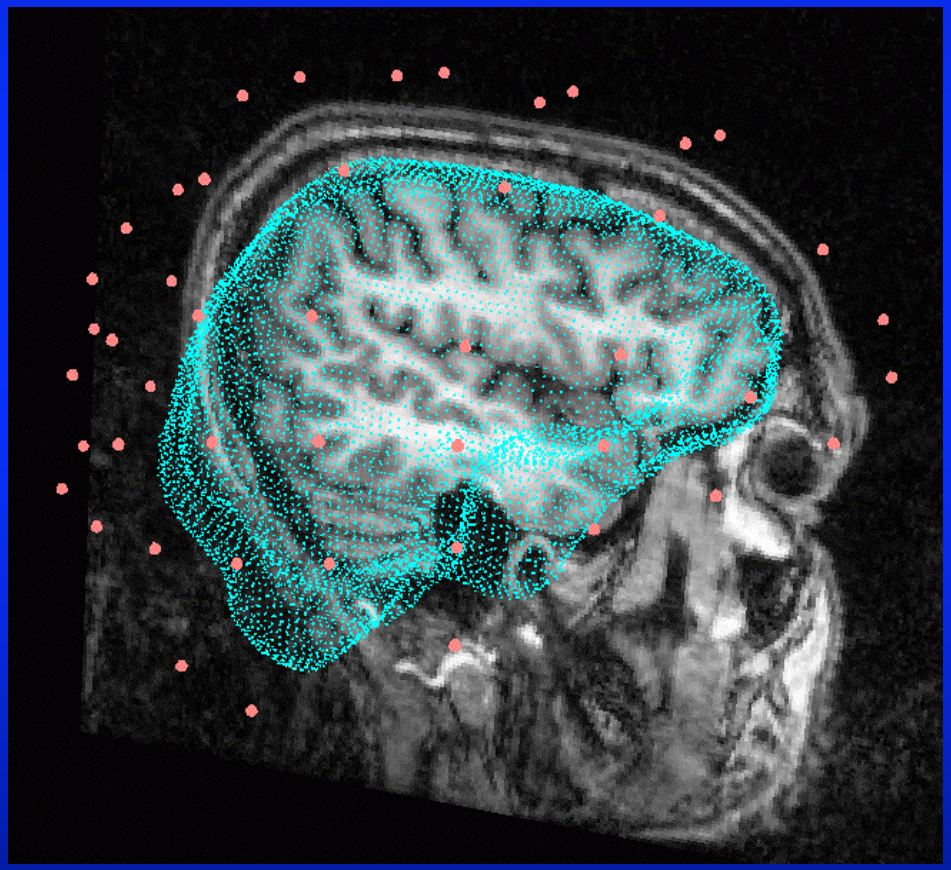
- **Head model: Tetrahedral FE, 147,287 nodes, 892,119 elements**



[Wolters, Grasedyck, Anwander & Hackbusch, *Biomag*, 2004]

Efficiency of the fast FE transfer matrix approaches

- Head model: Tetrahedral FE, 147,287 nodes, 892,119 elements
- Influence space: brain surface mesh, 2mm resolution, 9555 nodes



[Wolters, Grasedyck, Anwander & Hackbusch, *Biomag*, 2004]

Efficiency of the fast FE transfer matrix approaches

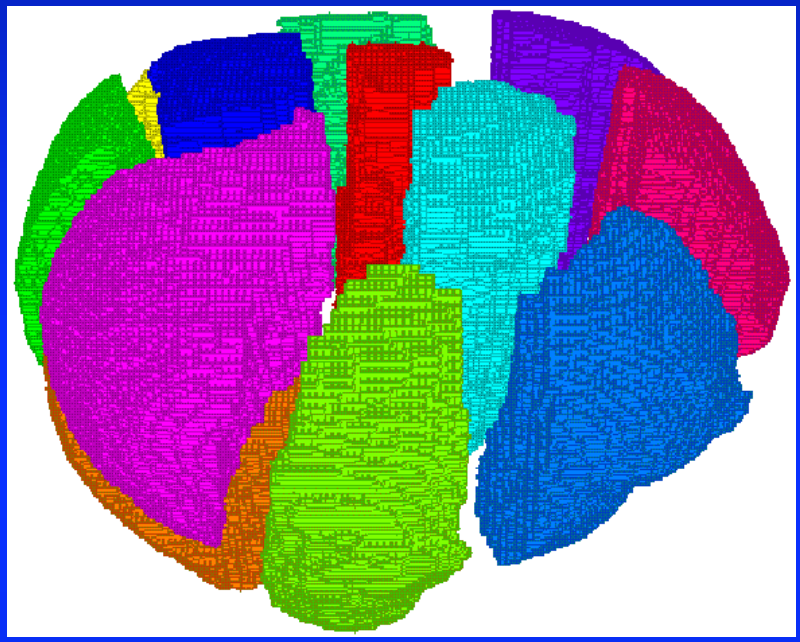
- Head model: Tetrahedral FE, 147,287 nodes, 892,119 elements
- Influence space: brain surface mesh, 2mm resolution, 9555 nodes
- Number of FE forward solutions:
  EEG, 71 electrodes: 9555 * 3 = 28665
  MEG, 147 channels: 9555 * 2 = 19110 (tangential constraint)
- FE approach and dipole model: Venant

Platform	Method	Solver method	Setup time		Influence matrix		Max.memory (in MB)	
			MEG	EEG	MEG	EEG	MEG	EEG
Mac G4	New Lead field bases approach	3RHS-AMG-CG	14min	6.5min	63 sec	56 sec	654	405
	Standard	symIC(0)-CG	0.5sec	0.5sec	230 h	302 h	432	219

[Wolters, Kuhn, Anwander & Reitzinger, *Comp.Vis.Sci,* 2002]

Parallel AMG-CG on distributed memory computers

``Element-wise'' partitioning into subdomains

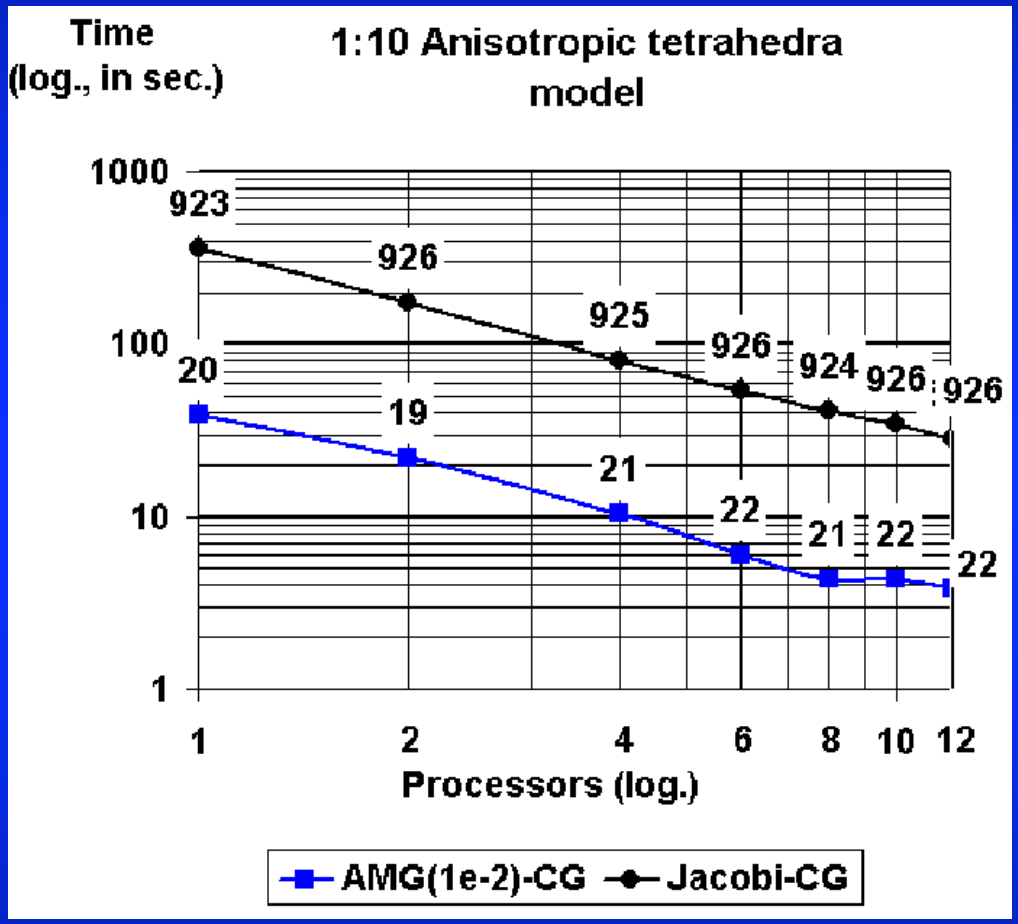


- **Parallel MultiRHS-AMG-CG: Communication is only necessary for:**
  - **Smoother (ω-Jacobi within interface nodes, Gauss-Seidel between blocks and for inner nodes)**
  - **distribution of coarse grid solution**
  - **inner products within PCG method**

[Wolters, Kuhn, Anwander & Reitzinger, *Comp.Vis.Sci,* 2002]

Results for parallel solver methods

SGI Origin2000, each processor 195MHz, MIPS 10000



Solver time comparison: Unknowns:147.287

- Distribution of memory
- About a linear speedup for moderate processor number
- Jacobi-CG on 1 proc <-> AMG-CG on 8 procs: Speedup factor of about 80 (10 MG, 8 parallel.)
- Anisotropy and inhomogeneity does not change performance results
- Stable preconditioning for moderate processor numbers

Thank you for your attention!

SIM-NEURO work-group at IBB