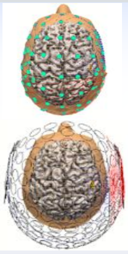
A Full-Multigrid Solver for EEG-based Reconstruction of Dipolar Current Sources in Human Head

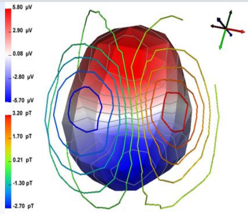
Dominik Bartuschat, Dominik Haspel, Eugenio Carvalheira, Yongqi Sun
BGCE Project 2007
Computer Science 10 LSS, University of Erlangen-Nürnberg
Friedrich-Alexander-Universität Erlangen-Nürnberg

Motivation

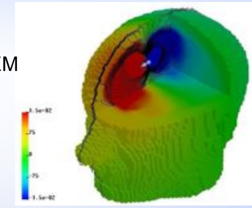
- Localization of dipole in human brain
- Data from EEG/MEG measurement
- Application example:
  Presurgical epilepsy diagnosis
- Reconstruction by NeuroFEM


A dipole (red arrow) in human brain


EEG
MEG


Result of Measurement

NeuroFEM


Reconstructed dipole source

NeuroFEM

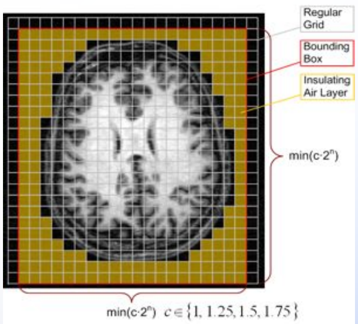
- A software package for dipole localization
- Optimization strategy:
  1) Calculate the electic field of the dipole at an assumed position
  2) Compare with EEG/MEG measurement and correct the position
  3) Go to 1) with corrected position until tolerance reached
- Several solvers integrated: e.g. AMG-preconditioned-CG (AMG-CG)

ParExPDE

- C++ library for Finite Element solvers of PDE
- User-friendly and easy to use interface
- Efficient code by Expression Templates
- Regularly refined hexahedral grids
- Capability for high-performance computing
  e.g. parallelism with MPI

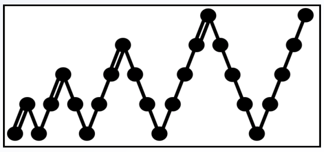
Task

- Integration of a Full-Multigrid solver into NeuroFEM
- Initialization procedure:
  - Build a structured volume grid from the original unstructured grid
  - Construct the local stiffness matrix with insulating air layer
  - Initialize all multi-grid levels
- Solving procedure:
  - Initialize the solution
  - Get the right hand side
  - Solve the equation system
  - Store the result back to NeuroFEM
- Minimizing the bounding box:
  - e. g. for original grid with 94 cells,
    use 96 (=$1.5 \cdot 2^6$) instead of 128 (=2^7) cells



Full-Multigrid Method

- Start on coarsest level
- Use solution on coarser level as initial guess for finer level

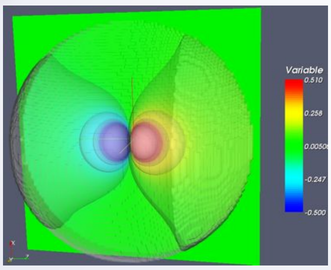


- Restriction using ParExPDE:

```
for (int level = 0; level < number_levels-1; ++level) {
  for (int smooth = 0; smooth < presmooth; ++smooth) {
    u = u + (f - op(u)) / ParExPDE::Diag(op)
      | ParExPDE::interior_points; // Gauß-Seidel method
  }
  r = f - op(u) | ParExPDE::interior_points;
  r = 0.0 | ParExPDE::boundary_points;
  r.doRestrict(); // Restrict to coarser grid
  f.levelDown();
  f = r;
  u.levelDown();
  u = 0.0;
} // Restriction from finest grid to coarsest grid
```

Results & Conclusion

- Test case:
  - Resolution of 2mm
  - Cube discretized into 94 x 94 x 94 cells
  - A ball with a dipole at the center
  - Four layers with different conductivities

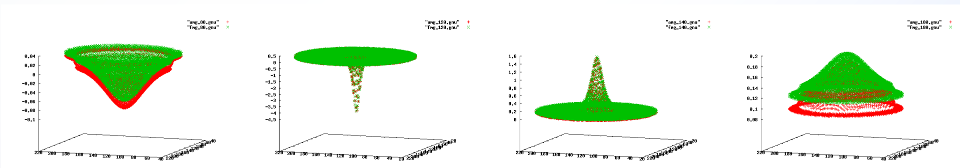


Electric field of the dipole at the center of the ball

- Comparison with AMG-CG
  - Faster setup time
  - Less memory consumption
  - Solving is slower than AMG-CG
  - Slow convergence with conductivity jumps

Performance of FMG (using Gauss-Seidel smoother) and AMG-CG

	Memory[MB]	Setup Time[s]	Runtime[s]
AMG-CG	1147	54.59	14.40
FMG	885	23.27	15.06


(a) z=80 (b) z=120 (c) z=140 (d) z=180

Electric potential (red marks AMG-CG, green marks FMG) in x-y planes at different z-positions.
Differing results for AMG and FMG due to different boundary conditions and insulating layer