Solving the EEG forward problem using the subtraction approach and adaptivity with hanging nodes in Dune-FEM

Falk Meyer

fmeye10@uni-muenster.de

18.02.2013

Westfälische Wilhelms-Universität Münster

IBB Institute for Biomagnetism and Biosignalanalysis

1. Subtraction forward problem
2. Adaptive finite element approach
3. Detection and treatment of hanging nodes
4. Dune-FEM, setting and examples
5. To Do

Outline

The continous EEG forward problem in source analysis is to find a solution for the electric potential u such that

$$\begin{aligned} div\sigma(x)\nabla u(x) &= \mathbf{J}^p(x) && \forall x \in \ \Omega \\ <\sigma(x)\nabla u(x), \mathbf{n}(x)> &= 0 && \forall x \in \ \partial\Omega \end{aligned}$$

This partial differential equation discribes the distribution of the electric potential u in the volume condutor Ω caused by means of the primary current $\mathbf{J}^p$ representing brain activity, where $\sigma \in \mathbb{R}$ depicts the conductivity in the certain head compartments in the isotropic case.

To apply standard discretisation techniques for this problem we then solve the subtraction forward problem, that is to find a solution for the *correction potential* $u^{corr,y}$ such that

$$\begin{aligned} div\sigma(x)\nabla(u^{corr,y}(x)+u^{\infty,y}(x)) &= \mathbf{J}^p(x) && \forall x \in \Omega \\ <\sigma(x)\nabla(u^{corr,y}(x)+u^{\infty,y}(x)),\mathbf{n}(x)> &= 0 && \forall x \in \partial\Omega \\ \int_\Omega (u^{corr,y}(x)+u^{\infty,y}(x))\mathrm{d}x &= 0 \end{aligned}$$

With $u^{\infty,y}$ being the singularity potential given analytically and $y \in \Omega$ the source position (the mathematic dipole). A unique solution $u^{corr,y} \in \mathbf{H}^1(\Omega)$ of this analytical forward problem exists according to *Wolters et al.* (2007*a*).

Knowing that the primary current density function is described by

$$\mathbf{J}^p(x) = \mathbf{J}^y(x) = div\mathbf{M}(y)\delta(x-y) \qquad\qquad \mathbf{M}(y) \in \mathbb{R}^3$$

the above system of equations can be written in the following form:

$$\begin{aligned}
div\sigma(x)\nabla u^{corr,y}(x) &= div(\sigma(y)-\sigma(x))\nabla u^{\infty,y} && \forall x \in \Omega\\
<\sigma(x)\nabla u^{corr,y}(x), \mathbf{n}(x)> &= -<\sigma(x)\nabla u^{\infty,y}, \mathbf{n}(x)> && \forall x \in \partial\Omega\\
\int_\Omega u^{corr,y}(x)\mathrm{d}x &= -\int_\Omega u^{\infty,y}(x)\mathrm{d}x
\end{aligned}$$

This problem is now well suitable for a finite element approach, because the singularity of $u^{\infty,y}$ in the source position y has no impact on the right-hand side anymore.

Outline

We now specify the finite element space V_T: Let $T = \{t_1, t_2, ..., t_t\}$ be a decomposition of a polygonal domain $\Omega \subset \mathbb{R}^3$ into cubes, then

$$V_T := \{\varphi \in V | \varphi \,|_{T_i} \, affine, i = 1, ..., t\}$$

denotes the standard Lagrangian finite element space for $V \subset \mathbf{H}^1(\Omega)$.

Using standard conforming linear basis functions φ_i at nodal positions v_i, $i = 1, ..., N$ the finite element approach leads to the following linear system:

$$\mathbf{K}u = \mathbf{b}^y \qquad \mathbf{K} \in \mathbb{R}^{N\times N}, u, b^y \in \mathbb{R}^N$$

with the stiffness matrix

$$\mathbf{K}_{i,j} := \int_\Omega < \sigma(x)\nabla\varphi_j, \nabla\varphi_i > \mathrm{d}x \qquad i,j = 1, ..., N$$

and right-hand side

$$\begin{aligned}\mathbf{b}_i^y := \quad & \int_\Omega < (\sigma(y) - \sigma(x))\nabla u^{\infty,y}(x), \nabla\varphi_i(x) > \mathrm{d}x \\ & - \int_{\partial\Omega} \varphi_i(x) < \mathbf{n}(x), \sigma(y)\nabla u^{\infty,y}(x) > \mathrm{d}x, i = 1, ..., N.\end{aligned}$$

For adaptive refinements we then introduce a residual based error estimator η_T satisfying (*Zhao et al*. 2010)

$$\|u^{corr,y} - u_h^{corr,y}\|_\Omega^2 \leq C \cdot \eta_T^2 = C \sum_{t \in T} \eta_t^2$$

for a positive constant C and $u_h^{corr,y}$ as the solution of the before seen FEM-approach. The local error estimator η_t is defined by

$$\eta_t^2 = h_t^2 \|f - div\sigma \nabla u_h^{corr,y}\|_{L^2(t)}^2 + h_t \sum_{\iota \in I_t} \|r(u_h^{corr,y})\Big|_\iota\|_{L^2(\partial t)}^2$$

with

$$r(u_h^{corr,y})\Big|_\iota = \begin{cases} n_t \cdot [\sigma \nabla u_h^{corr,y}], & \text{if } \iota \subset \Omega \setminus \partial\Omega, \\ < \sigma \nabla u^{corr,y}, \mathbf{n} > -g, & \text{if } \iota \subset \partial\Omega \end{cases}$$

where f and g denote the appropriate right-hand sides in the subtraction forward problem.

Now we give the adaptive finite element algorithm as follows:

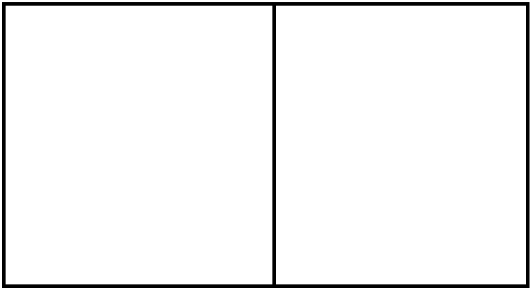
1. Give the initial conforming hexahedral mesh T_0 and parameter θ, set $l = 0$;
2. Solve subtraction forward problem on T_l and obtain solution $u_l^{corr,y}$;
3. Compute the error estimator η_t for each element $t \in T_l$;
4. Mark minimal elements set M_l such that for all $t \in M_l$

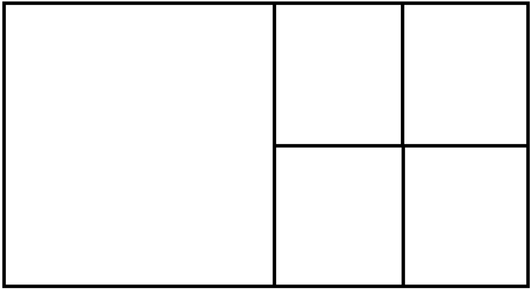
$$\eta_t^2 \geq \theta \cdot \max_{t' \in T_l} \eta_{t'}^2;$$

5. Refine T_l by bisection of all elements in M_l to get T_{l+1};
6. Detect and treat resulting hanging nodes appropriately;
7. Set $l := l + 1$ and go to step 2.

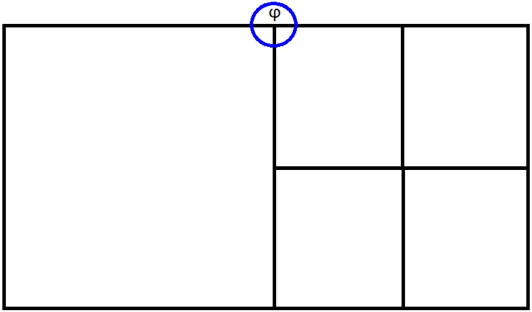
Outline

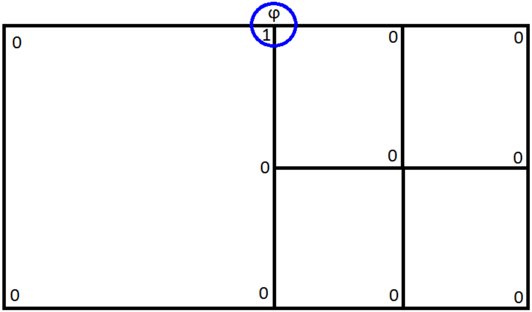
Hanging nodes occur after bisection of a cube in the grid while one of its neighbors is not refined:



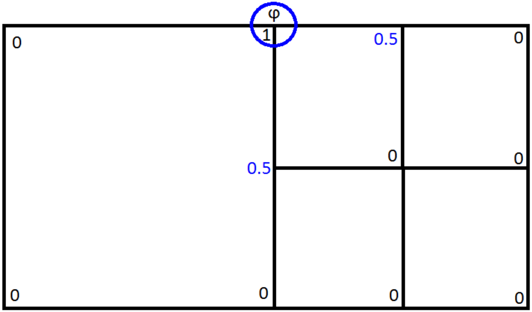


For this type of a grid a standard finite element discretization creates a non conforming basis with discontinous piecewise linear functions φ:




φ
0
1
0
0
0
0
0
0
0
0
0
0

To solve this problem we build a composite basis function φ as follows:



To detect hanging nodes in a grid G we use the following algorithm:

- vector<IndexType> hangingnodeVector;
- **for** leaf entity $i = 0, ..., \#$ {Codim 0 leaf entities of G}
  - **for** intersection $j = 0, ..., \#$ {intersections of entity i}
    - **if** $j.insideEntity.level() > j.outsideEntity.level()$
      Save indexes of all occuring hanging nodes h as corners of j in hangingnodeVector by checking whether a corner of j is a corner in $j.outsideEntity$.

To treat hanging nodes we first define

$$N_h = \{v_0, ..., v_{M-1}, v_M, ..., v_{N-1}\}$$

as the set of all nodes in the grid G where v_i is a hanging node for $i \in \{M, ..., N-1\}$ and a regular node for all other indexes i.
Let ψ_i denote the linear basis function representing a non-hanging node v_i.
To arrive at a conforming basis of the finite element space V_T we combine each ψ_i with appropriately scaled functions ψ_k of neighboring hanging nodes such that

$$\forall 0 \leq i < M : \varphi_i = \psi_i + \sum_{k=M}^{N-1} \alpha_{ik} \psi_k$$

is a conforming basis function φ_i.

Let $t \in T$ a cube with non-hanging node v_i and hanging node v_k, then α_{ik} is the evaluation of the element shape function ϕ_i^f at node v_k on the father element f of e. Now we arrive at a conforming basis

$$B = \{\varphi_i | 0 \leq i < M\}$$

with $< B >= V_T$, because for $u \in V_T$

$$\begin{aligned} u = & \sum_{j=0}^{M-1} x_j \varphi_j = \sum_{j=0}^{M-1} x_j \Big(\psi_j + \sum_{k=M}^{N-1} \alpha_{jk} \psi_k \Big) = \sum_{j=0}^{M-1} x_j \psi_j + \sum_{k=M}^{N-1} \underbrace{\Big(\sum_{j=0}^{M-1} \alpha_{jk} x_j \Big)}_{=: x_k} \psi_k \\ = & \sum_{j=0}^{N-1} x_j \psi_j \end{aligned}$$

holds.

With

$$a(\psi_i,\psi_j) = \int_\Omega < \sigma(x)\nabla\varphi_j, \nabla\varphi_i > \mathrm{d}x$$

we then can modify the linear system $\mathbf{K}u = \mathbf{b}^y$ as follows:

$$\begin{pmatrix} K_{RR} & K_{RH} \\ K_{HR} & I \end{pmatrix} \begin{pmatrix} x_R \\ x_H \end{pmatrix} = \begin{pmatrix} b_R^y \\ 0 \end{pmatrix}$$

where

$$\begin{array}{lcl} 0 \leq i < M & : & (K)_{ij} = a(\psi_j,\psi_i) + \sum_{k=M}^{N-1} \alpha_{ik} a(\psi_j,\psi_k) \\ M \leq i < N, 0 \leq j < M & : & (K)_{ij} = -\alpha_{ji} \\ M \leq i,j < N & : & (K)_{ij} = \delta_{ij} \end{array}$$

Outline

DUNE, the Distributed and Unified Numerics Environment is a modular toolbox for solving partial differential equations (PDEs) with grid-based methods.[...] DUNE is based on the following main principles:
Separation of data structures and algorithms by abstract interfaces, Efficient implementation of these interfaces using generic programming techniques, Reuse of existing finite element packages with a large body of functionality.
(http://www.dune-project.org/dune.html)

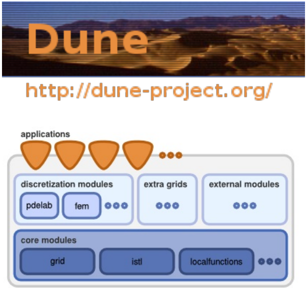


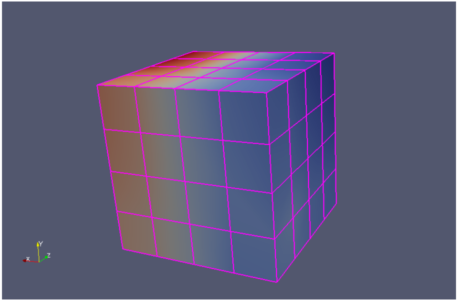
Figure : The design principle of Dune

DUNE-FEM is a DUNE module which defines interfaces for implementing discretization methods like Finite Element Methods (FEM) and Finite Volume Methods (FV) and Discontinuous Galerkin Methods (DG).
DUNE-FEM offers interfaces to solve the Poisson equation on different grids (ALBERTAGRID, ALUGRID etc.) with various solver.
To summarize the advantages of DUNE-FEM:

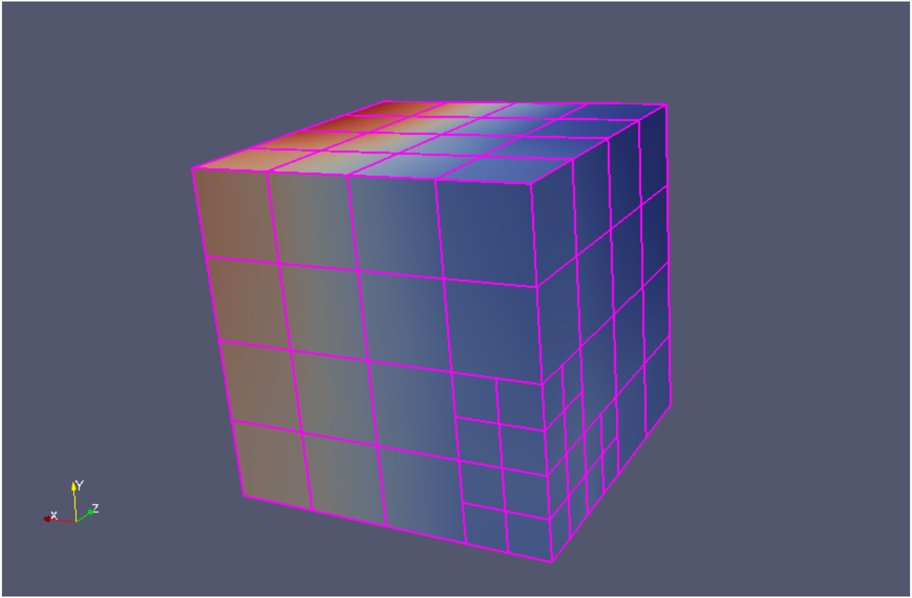
- flexibility through template-based implementations
- different solvers and optimization opportunities already given
- use of different grid managers independent of current implementation

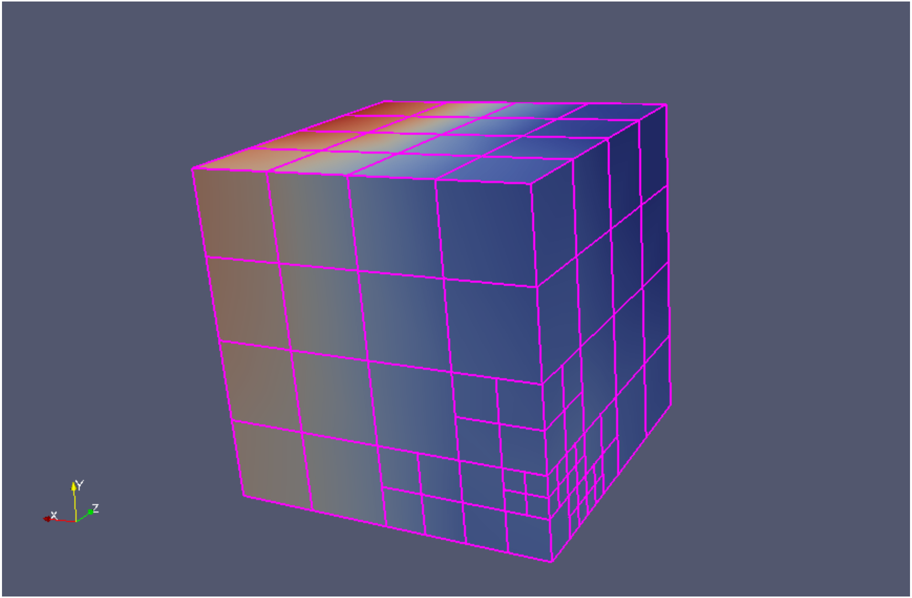
First example

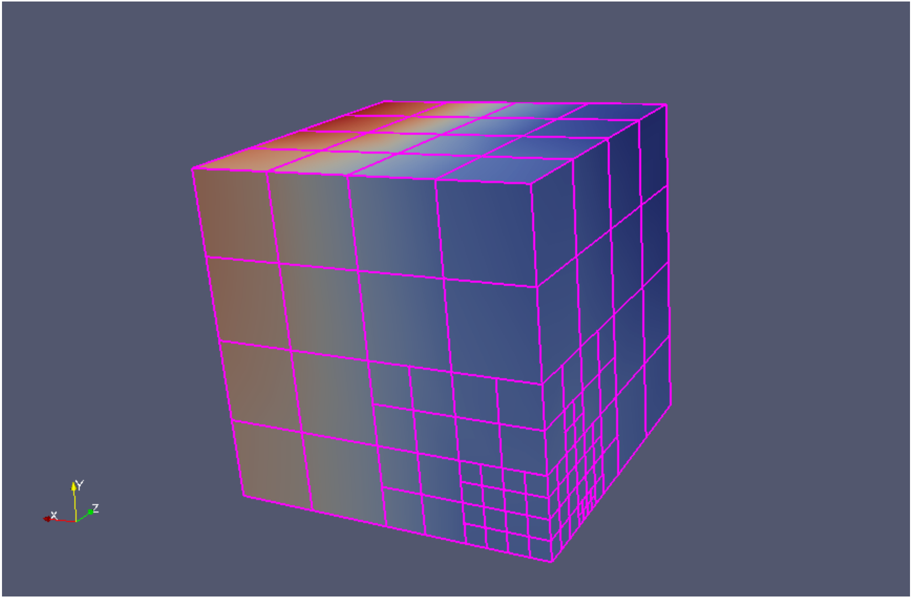
Let $\Omega = [0,1]^3 \subset \mathbb{R}^3$ und T be a decomposition of Ω into 64 cubes of the same edge-length. Then we solve the subtraction approach with $y = (0.6, 0.6, 0.6)$, $M = (0.0001, 0, 0)$ and $\theta = 0.2$ on this area as seen below:

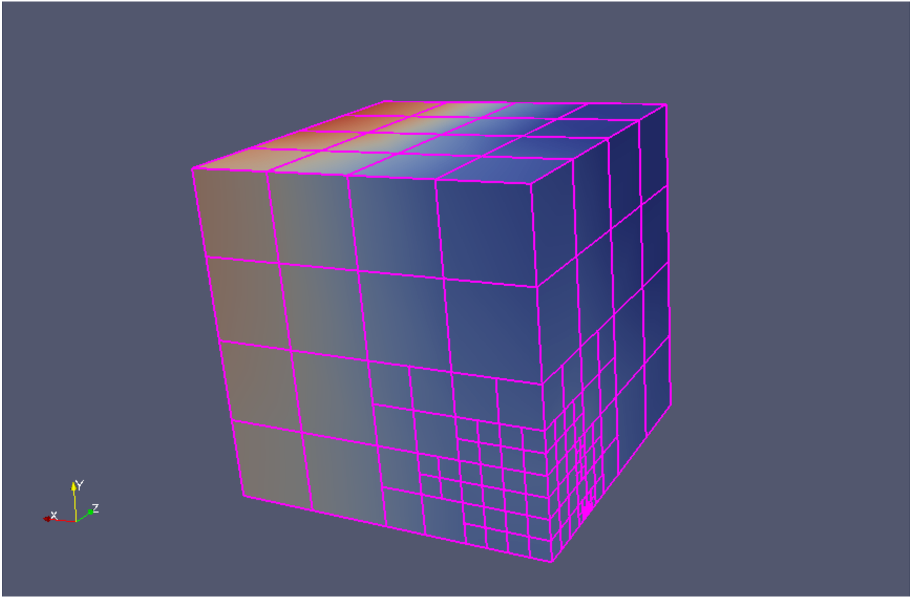


The cube with corner $(0,0,0)$ obained a different conductivity σ than all other cubes. The adaption process with hanging node treatment is shown on the next slices.

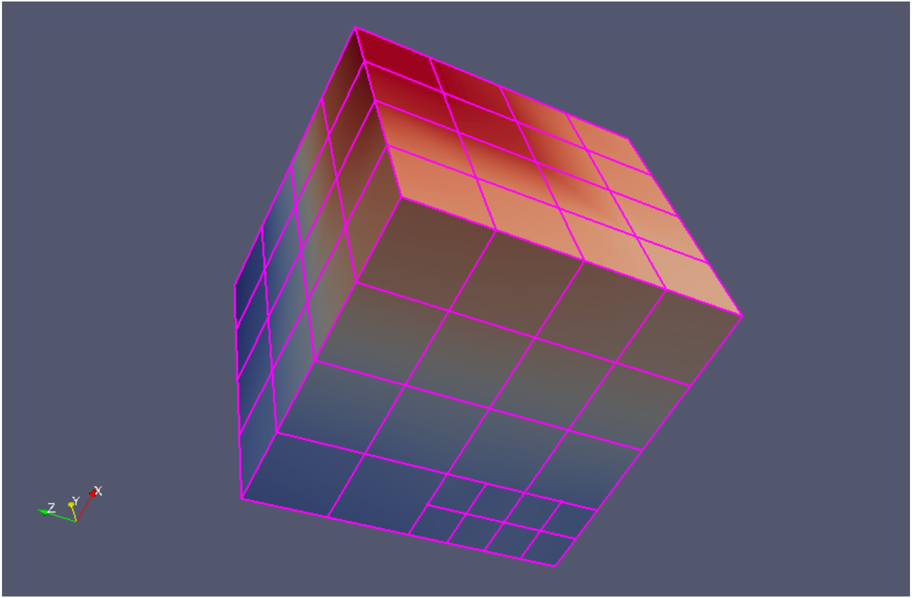

Y
Z
X

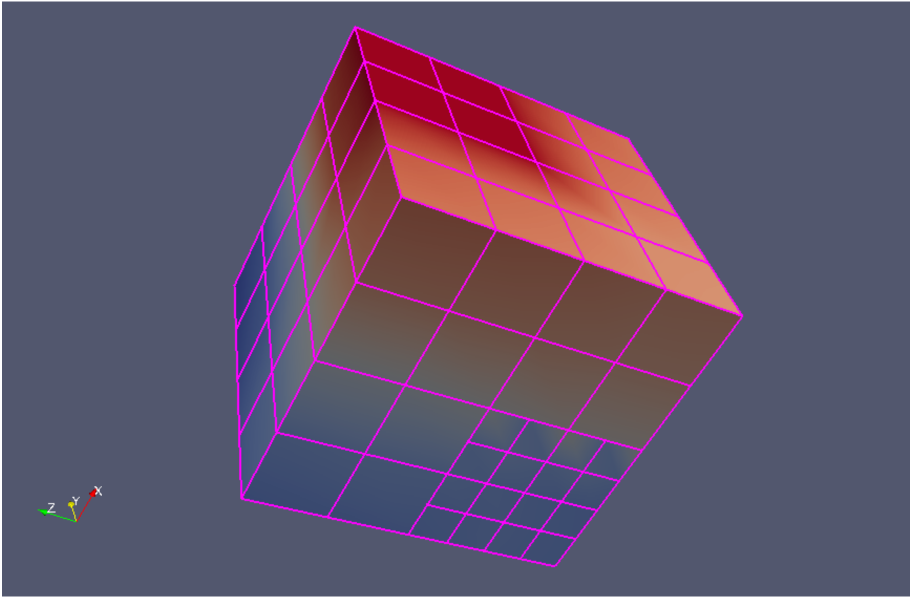

Y
X
Z

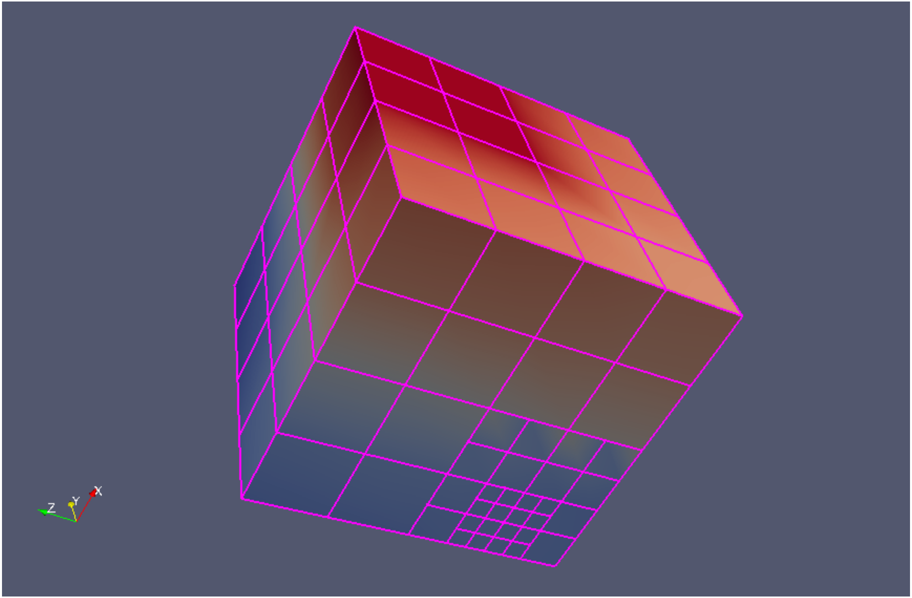

Y
X
Z

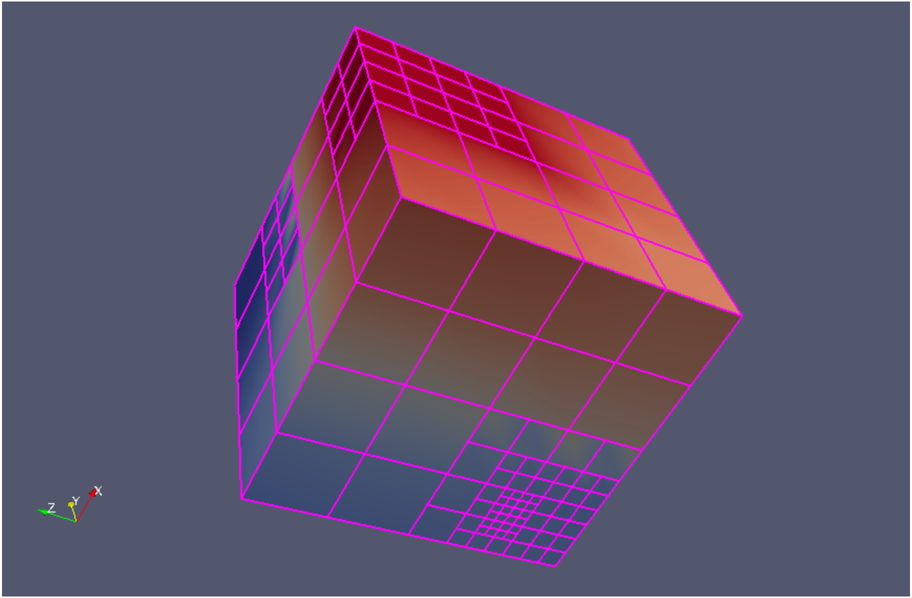

Y
X
Z

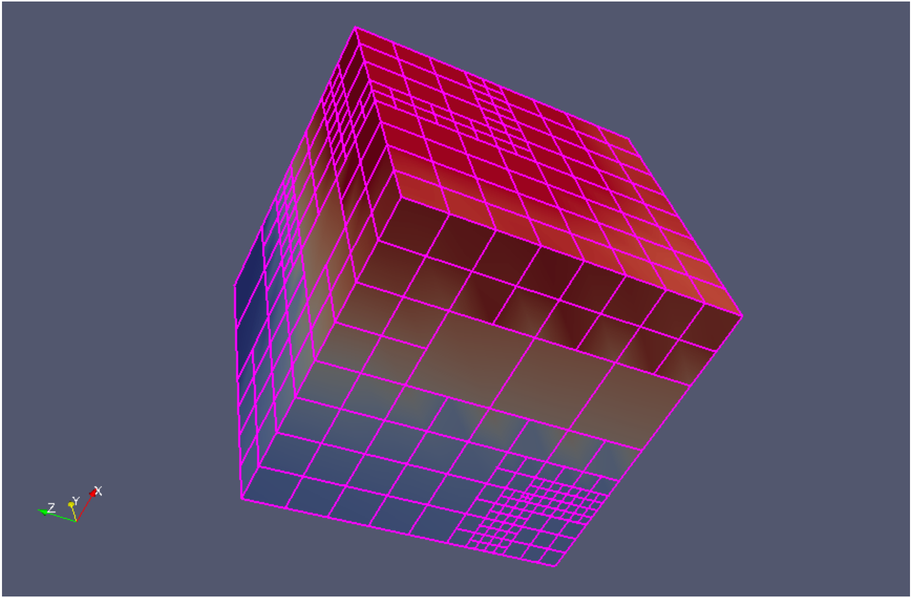
The following slices show the adaption process without hanging node treatment


Z
Y
X


Z
Y
X


Z
Y
X

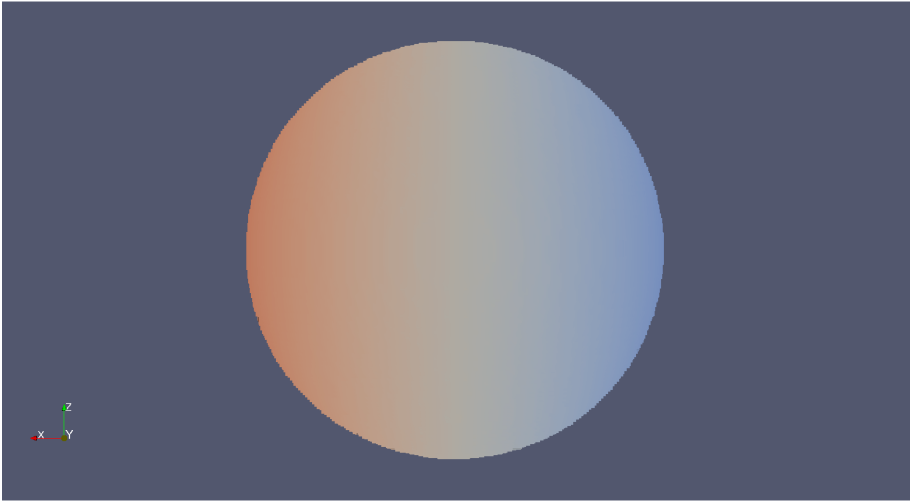

Z
Y
X

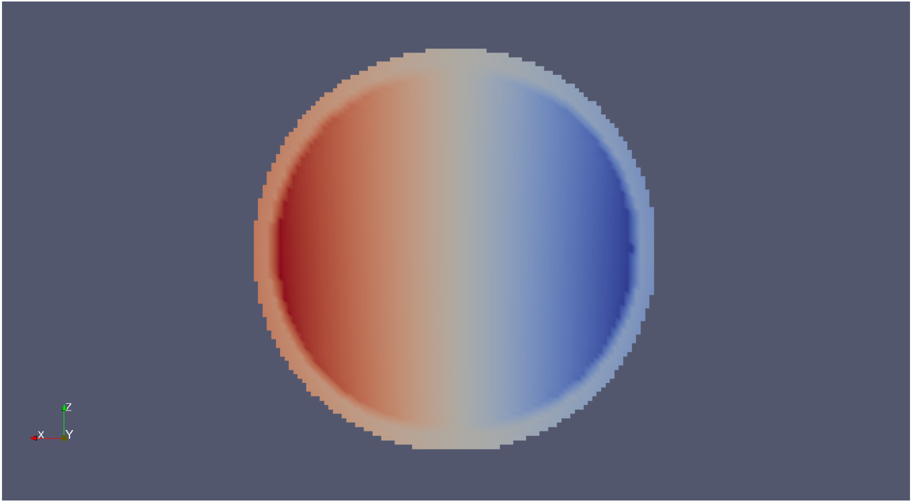

Z
Y
X

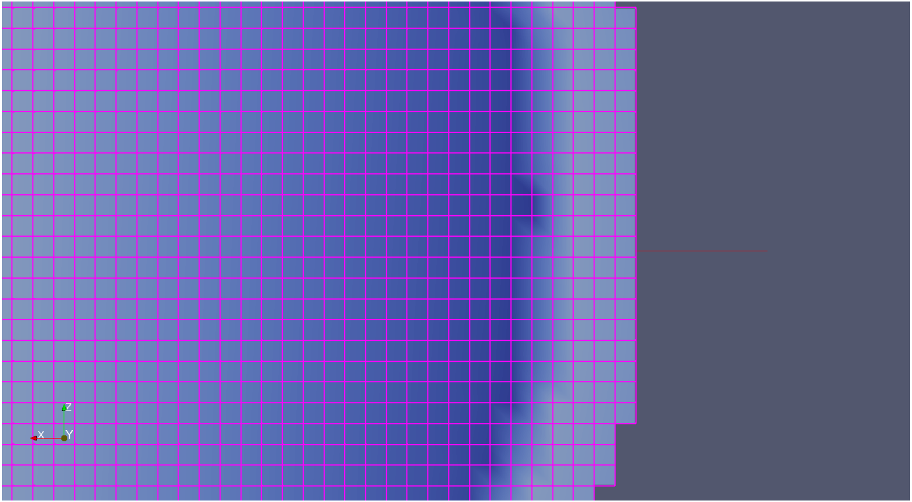
Second example

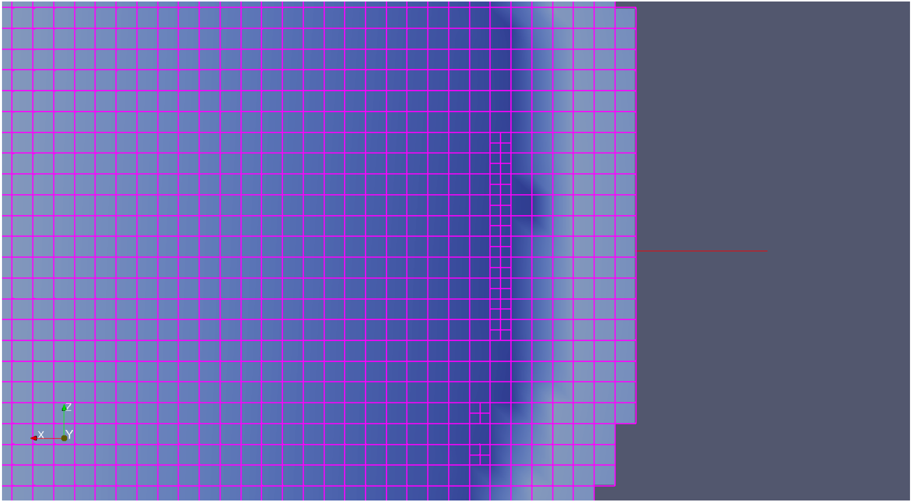
Let $\Omega \subset \mathbb{R}^3$ be a 4layer sphere model with radius 92($\cdot$2mm), centered in (127, 127, 127), and T be a conforming decomposition of Ω into 405545 2mm cubes. Then we solve the subtraction approach with $y = (127.5, 127.3, 126)$, $M = (1, 0, 0)$ and $\theta = 0.9$ on this area.
The 4 layers represent (from inside to outside)

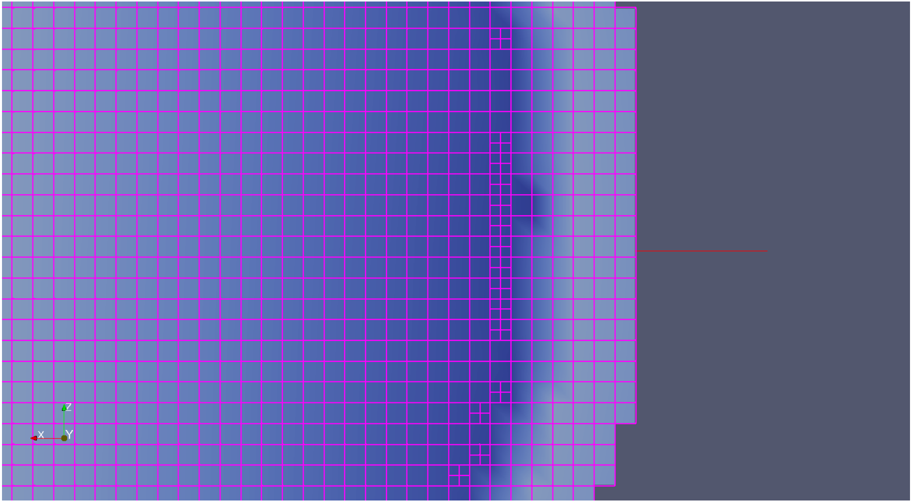
- the brain (inner sphere radius 78) with $\sigma_{104} = 0.00033$ S/mm
- the CSF (inner sphere radius 78 to 80) with $\sigma_{103} = 0.00179$ S/mm
- the Skull (inner sphere radius 80 to 86) with $\sigma_{102} = 0.000042$ S/mm
- the Skin (inner sphere radius 86 to 92) with $\sigma_{101} = 0.00033$ S/mm


Z
X
Y


Z
X
Y


Z
X
Y


Z
X
Y


Z
X
Y

The following observations regarding the RE and MAG error between the (deMunck) exact and numerical solution of the subtraction problem were made:

adaption step	# hanging nodes	RE	MAG
0	0	0.021882	1.01893
1	$>$1000	0.0216304	1.01861
2	$>$2000	0.0214971	1.01845

Outline

- Implement anisotropy
- Detailed case study
- Improve runtime and implementation
- Figure out future possibilities of Dune
- Finish master thesis

Thank you for your Attention!