Barbara Głut [ID]
Maciej Paszyński [ID]

OVERVIEW OF ADAPTIVE AND LOW-RANK APPROXIMATION ALGORITHMS FOR MODELING INFLUENCE OF ELECTROMAGNETIC WAVES GENERATED BY CELLPHONE ANTENNA ON HUMAN HEAD

Abstract

This paper presents an overview of formulations and algorithms that are dedicated to modeling the influence of electromagnetic waves on the human head. We start from h adaptive approximation of a three-dimensional MRI scan of the human head. Next, we solve the time-harmonic Maxwell equations with a 1.8 GHz cellphone antenna. We compute the specific absorption rate used as the heat source for the Pennes bioheat equation modeling the heat generated by EM waves inside the head. We propose an adaptive algorithm mixed with time-stepping iterations where we simultaneously refine the computational mesh, solve the Maxwell and Pennes equations, and iterate the time steps. We employ the sparse Gaussian elimination algorithm with the low-rank compression of the off-diagonal matrix blocks for the factorization of the matrices. We conclude with the statement that 15 minutes of talking with a 1.8 GHz antenna with one watt of power results in increased brain tissue temperatures (up to $38.4°C$).

Citation

Computer Science 22(4) 2021: 433–461

1. Introduction

In this paper, we present an overview of the adaptive finite element method [4, 11, 17] application to the problem of simulations of the influence of an electromagnetic wave antenna on the biological tissues of the human head.

This is a very challenging and important problem. This problem was solved for an artificial "plastic" model of the human head in the Ph.D. thesis of Kyungjoo Kim [14]. This problem was also solved in [19] by using an approximated heat source that represented cellphone radiation based on the results obtained in [14]. The formulation of a cellphone antenna with equivalent currents was described in [8]. This topic is subject to intense research [7, 12, 20], which is focused on different locations of cellphone antennas, its influence on the human heads of adults and children, and the different kinds of cellphone antennas.

In this paper, we employ the following: (1) L2 projection problem for approximating MRI scan of human head; (2) time-harmonic Maxwell equations modeling waves emitted by antenna of cellphone; (3) Pennes bioheat-transfer equations. We combine the following algorithms: (1) h adaptive algorithm employing finite element method with quadratic basis functions to solve L2 projection problem; (2) hp adaptive algorithm employing finite element method with hierarchical basis functions to solve Maxwell problem; (3) time-stepping algorithm using Crank-Nicolson time-integration scheme; (4) hp adaptive algorithm employing finite element method with hierarchical basis functions for solving Pennes bioheat problem; (5) low-rank approximation solver algorithm employing SVD decomposition of off-diagonal matrix blocks.

The time-harmonic Maxwell equation concerns the electromagnetic field propagation on the model of the human head. Its input is the distribution of the materials of the human head with permeability, permittivity, and conductivity. We assume that these electromagnetic properties do not depend on the temperature, only on the type of material. Thus, even if we update the temperature distribution of the human head, the electromagnetic properties of materials do not change; therefore, the input to the time-harmonic Maxwell equations remains the same. The elliptic problem concerns the Pennes bioheat equation. Its input is the material properties, and the right-hand side is computed based on the electromagnetic wave distribution that is estimated by the Maxwell equations. The heat transfer is a non-stationary time-dependent phenomenon, while the electromagnetic wave distribution does not change over time in our model.

The structure of this paper is as follows. We start by introducing three-dimensional finite elements and basis functions, and we illustrate their application on the problem of computing a projection of the human head based on MRI scan data. This first section is an illustrative example of how the adaptive algorithm works by using the two-grid paradigm. The next section focuses on a cellphone antenna model, with a weak formulation of the time-harmonic Maxwell equations. We comment on the values of the parameters for modeling the human head. The next section derives the Crank-Nicolson time-stepping algorithm for the heat-transport equations.

We instantiate the problem for the Pennes bioheat-transfer equations. Finally, we combine these ideas, and we introduce the space-adaptive time-stepping algorithm. Finally, we summarize the paper with the our conclusions, and in the appendix, we describe the sparse Gaussian elimination algorithm with the low-rank compression of off-diagonal blocks that is utilized for the solution of the encountered systems of equations.

2. Adaptive generation of approximation of human head based on MRI scan data

This section aims to derive the adaptive algorithm for the approximation of the MRI scan data of the human head that is presented in Figure 1. The result of the algorithm is to replace the three-dimensional MRI scan of the human head with the continuous approximation of the material data, which is more suitable for the finite element method computations.
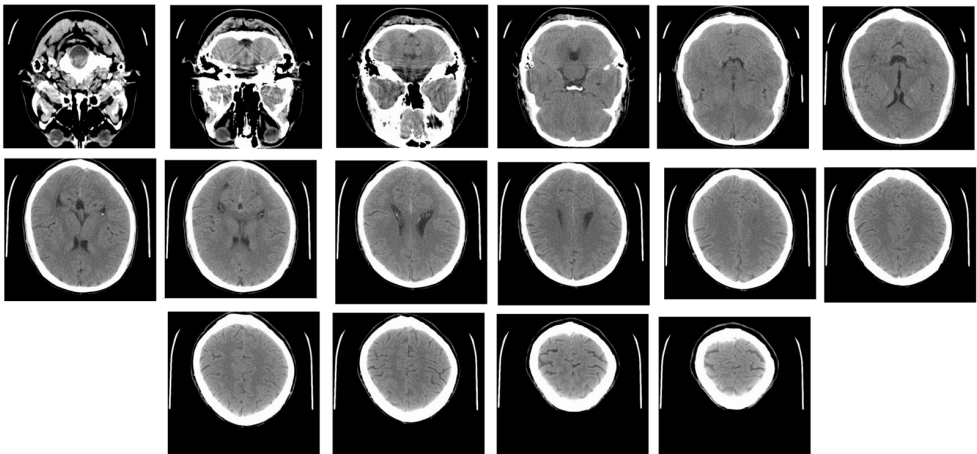

Figure 1. MRI scan of head of Maciej Paszyński

This section introduces the *hp* finite element with hierarchical basis functions (as was proposed by [3, 4]). We start by defining the one-dimensional hierarchical basis functions that will be used later on to define the two- and three-dimensional basis functions by their tensor products:

$$
\begin{aligned}
\hat{\chi}_1(\xi) &= 1 - \xi \\
\hat{\chi}_2(\xi) &= \xi \\
\hat{\chi}_3(\xi) &= (1 - \xi)\xi \\
\hat{\chi}_p(\xi) &= (1 - \xi)\xi(2\xi - 1)^{p-3} \quad p = 4, 5, ..., 9
\end{aligned}
\tag{1}
$$

These 1D basis functions are used to define the 2D basis functions over the two-dimensional finite element that is presented in Figure 2.
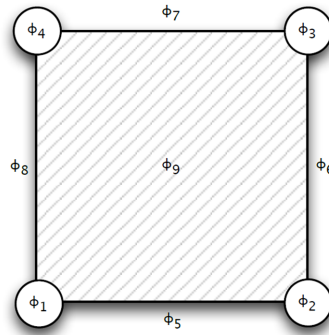
Figure 2. Two-dimensional finite element with four vertex nodes, four edge nodes, and one interior node

For the L2 projection and time-harmonic Maxwell problems, we will employ quadratic basis functions; for the Pennes bioheat equation, we will employ higher-order basis functions as were proposed by the *hp* adaptive algorithm. Equation 2 and Figure 3 define the basis functions over each of the four vertices of the element:

$$
\begin{aligned}
\hat{\phi}_1(\xi_1, \xi_2) &= \hat{\chi}_1(\xi_1)\hat{\chi}_1(\xi_2) \\
\hat{\phi}_2(\xi_1, \xi_2) &= \hat{\chi}_2(\xi_1)\hat{\chi}_1(\xi_2) \\
\hat{\phi}_3(\xi_1, \xi_2) &= \hat{\chi}_2(\xi_1)\hat{\chi}_2(\xi_2) \\
\hat{\phi}_4(\xi_1, \xi_2) &= \hat{\chi}_1(\xi_1)\hat{\chi}_2(\xi_2)
\end{aligned}
\tag{2}
$$

Next, Equation 3 and Figure 4 define the basis functions over each of the four edges of the element:

$$
\begin{aligned}
\hat{\phi}_5(\xi_1, \xi_2) &= \hat{\chi}_3(\xi_1)\hat{\chi}_1(\xi_2) \\
\hat{\phi}_6(\xi_1, \xi_2) &= \hat{\chi}_2(\xi_1)\hat{\chi}_3(\xi_2) \\
\hat{\phi}_7(\xi_1, \xi_2) &= \hat{\chi}_3(\xi_1)\hat{\chi}_2(\xi_2) \\
\hat{\phi}_8(\xi_1, \xi_2) &= \hat{\chi}_1(\xi_1)\hat{\chi}_3(\xi_2)
\end{aligned}
\tag{3}
$$

Equation 4 and Figure 5 define the basis functions over the element's interior:

$$
\hat{\phi}_9(\xi_1, \xi_2) = \hat{\chi}_3(\xi_1)\hat{\chi}_3(\xi_2)
\tag{4}
$$

In other words, we have basis functions that are constructed as the tensor products of two linear one-dimensional basis functions at each vertex node. We have basis functions that are constructed as the tensor products of one linear and one quadratic one-dimensional basis function over each element edge. Finally, we have basis functions that are constructed as the tensor products of two quadratic basis functions over each element interior.
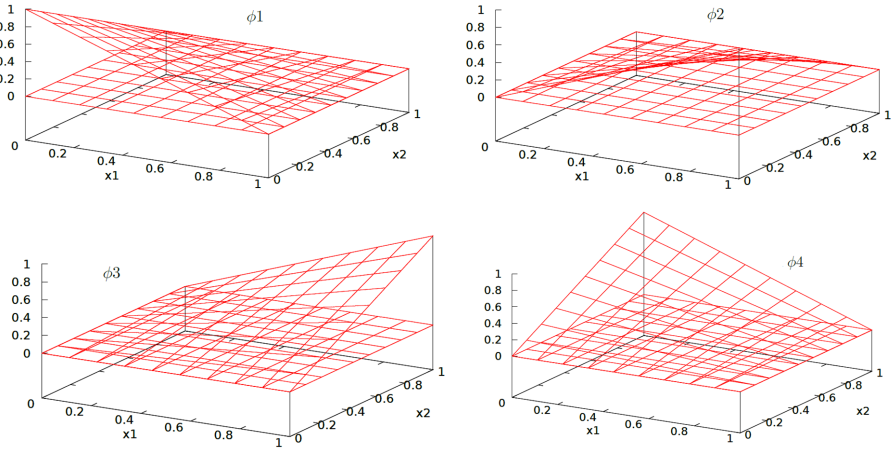
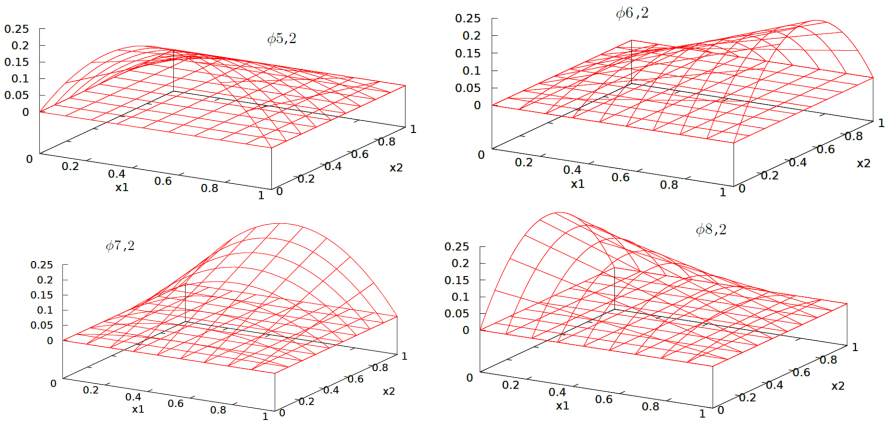**Figure 3.** Basis functions related to finite element vertices

Figure 4. Basis functions related to finite element edge nodes
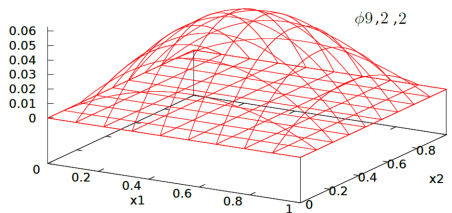
Figure 5. Basis function related to finite element interior node

This construction can be generalized into three-dimensional finite elements and into higher-order basis functions. Now, we define the basis functions over the three--dimensional hexahedral *hp* finite element. Equation 5 defines the basis functions over each of the eight vertices of the element:

$$
\begin{aligned}
\hat{\phi}_1(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_1(\xi_1)\hat{\chi}_1(\xi_2)\hat{\chi}_1(\xi_3) \\
\hat{\phi}_2(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_2(\xi_1)\hat{\chi}_1(\xi_2)\hat{\chi}_1(\xi_3) \\
\hat{\phi}_3(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_2(\xi_1)\hat{\chi}_2(\xi_2)\hat{\chi}_1(\xi_3) \\
\hat{\phi}_4(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_1(\xi_1)\hat{\chi}_2(\xi_2)\hat{\chi}_1(\xi_3) \\
\hat{\phi}_5(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_1(\xi_1)\hat{\chi}_1(\xi_2)\hat{\chi}_2(\xi_3) \\
\hat{\phi}_6(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_2(\xi_1)\hat{\chi}_1(\xi_2)\hat{\chi}_2(\xi_3) \\
\hat{\phi}_7(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_2(\xi_1)\hat{\chi}_2(\xi_2)\hat{\chi}_2(\xi_3) \\
\hat{\phi}_8(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_1(\xi_1)\hat{\chi}_2(\xi_2)\hat{\chi}_2(\xi_3)
\end{aligned}
\tag{5}
$$

Next, Equation 6 defines the $p_i - 1$ shape functions over each of the twelve edges of the element:

$$
\begin{aligned}
\hat{\phi}_{9,j}(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_{2+j}(\xi_1)\hat{\chi}_1(\xi_2)\hat{\chi}_1(\xi_3) & j &= 1, ..., p_1 - 1 \\
\hat{\phi}_{10,j}(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_2(\xi_1)\hat{\chi}_{2+j}(\xi_2)\hat{\chi}_1(\xi_3) & j &= 1, ..., p_2 - 1 \\
\hat{\phi}_{11,j}(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_{2+j}(\xi_1)\hat{\chi}_2(\xi_2)\hat{\chi}_1(\xi_3) & j &= 1, ..., p_3 - 1 \\
\hat{\phi}_{12,j}(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_1(\xi_1)\hat{\chi}_{2+j}(\xi_2)\hat{\chi}_1(\xi_3) & j &= 1, ..., p_4 - 1 \\
\hat{\phi}_{13,j}(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_{2+j}(\xi_1)\hat{\chi}_1(\xi_2)\hat{\chi}_2(\xi_3) & j &= 1, ..., p_5 - 1 \\
\hat{\phi}_{14,j}(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_2(\xi_1)\hat{\chi}_{2+j}(\xi_2)\hat{\chi}_2(\xi_3) & j &= 1, ..., p_6 - 1 \\
\hat{\phi}_{15,j}(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_{2+j}(\xi_1)\hat{\chi}_2(\xi_2)\hat{\chi}_2(\xi_3) & j &= 1, ..., p_7 - 1 \\
\hat{\phi}_{16,j}(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_1(\xi_1)\hat{\chi}_{2+j}(\xi_2)\hat{\chi}_2(\xi_3) & j &= 1, ..., p_8 - 1 \\
\hat{\phi}_{17,j}(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_1(\xi_1)\hat{\chi}_1(\xi_2)\hat{\chi}_{2+j}(\xi_3) & j &= 1, ..., p_9 - 1 \\
\hat{\phi}_{18,j}(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_2(\xi_1)\hat{\chi}_1(\xi_2)\hat{\chi}_{2+j}(\xi_3) & j &= 1, ..., p_{10} - 1 \\
\hat{\phi}_{19,j}(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_2(\xi_1)\hat{\chi}_2(\xi_2)\hat{\chi}_{2+j}(\xi_3) & j &= 1, ..., p_{11} - 1 \\
\hat{\phi}_{20,j}(\xi_1, \xi_2, \xi_3) &= \hat{\chi}_1(\xi_1)\hat{\chi}_2(\xi_2)\hat{\chi}_{2+j}(\xi_3) & j &= 1, ..., p_{12} - 1
\end{aligned}
\tag{6}
$$

Here, p_i stands for the polynomial order of approximation that was utilized over the *i*-th edge. Equation 7 also defines the $(p_{ih}-1) \times (p_{iv}-1)$ shape functions over each of the six faces of the element:

$$\hat{\phi}_{21}(\xi_1, \xi_2, \xi_3) = \hat{\chi}_{2+j}(\xi_1)\hat{\chi}_{2+k}(\xi_2)\hat{\chi}_1(\xi_3) \quad j = 1, ..., p_{13h} - 1, \ k = 1, ..., p_{13v} - 1$$

$$\hat{\phi}_{22}(\xi_1, \xi_2, \xi_3) = \hat{\chi}_{2+j}(\xi_1)\hat{\chi}_{2+k}(\xi_2)\hat{\chi}_2(\xi_3) \quad j = 1, ..., p_{14h} - 1, \ k = 1, ..., p_{14v} - 1$$

$$\hat{\phi}_{23}(\xi_1, \xi_2, \xi_3) = \hat{\chi}_{2+j}(\xi_1)\hat{\chi}_1(\xi_2)\hat{\chi}_{2+k}(\xi_3) \quad j = 1, ..., p_{15h} - 1, \ k = 1, ..., p_{15v} - 1$$

$$\hat{\phi}_{24}(\xi_1, \xi_2, \xi_3) = \hat{\chi}_2(\xi_1)\hat{\chi}_{2+j}(\xi_2)\hat{\chi}_{2+k}(\xi_3) \quad j = 1, ..., p_{16h} - 1, \ k = 1, ..., p_{16v} - 1 \tag{7}$$

$$\hat{\phi}_{25}(\xi_1, \xi_2, \xi_3) = \hat{\chi}_{2+j}(\xi_1)\hat{\chi}_2(\xi_2)\hat{\chi}_{2+k}(\xi_3) \quad j = 1, ..., p_{17h} - 1, \ k = 1, ..., p_{17v} - 1$$

$$\hat{\phi}_{26}(\xi_1, \xi_2, \xi_3) = \hat{\chi}_1(\xi_1)\hat{\chi}_{2+j}(\xi_2)\hat{\chi}_{2+k}(\xi_3) \quad j = 1, ..., p_{18h} - 1, \ k = 1, ..., p_{18v} - 1$$

Here, p_{ih} and p_{iv} are the polynomial orders of approximations in the two directions over the i-th face of the element. Finally, we define the $(p_x-1)\times(p_y-1)\times(p_z-1)$ basis functions over an element interior:

$$\hat{\phi}_{27,ij}(\xi_1, \xi_2) = \hat{\chi}_{2+i}(\xi_1)\hat{\chi}_{2+j}(\xi_2)\hat{\chi}_{2+k}(\xi_3)$$
$$i = 1, ..., p_x - 1, j = 1, ..., p_y - 1, k = 1, ..., p_z - 1 \tag{8}$$

where (p_x, p_y, p_z) are the polynomial orders of approximation along the x, y, and z directions.

In order to obtain a smooth approximation of the material data that is suitable for the finite element method computations, we employ the finite element method implementing the following L2 projections problem.

Given $\Omega \ni (x, y, z) \rightarrow F(x, y, z) \in R$, find the linear combination $\sum_{i=1,...,N_h} u_h^i e_i(x, y, z)$ such that

$$\|F(x, y, z) - \sum_{i=1,...,N_h} u_h^i e_i(x, y, z)\|_{L2} \rightarrow MIN \tag{9}$$

Here, $\{u_i\}_{i=1}^{N_h}$ are the coefficients, and $\{e_i(x, y, z)\}_{i=1}^{N_h} \in V_h$ are the basis functions obtained by gluing together the shape functions that are spread over a given computational mesh covering Ω. The projection problem can be formulated in a weak form:

$$\int_\Omega u(x, y, z)v(x, y, z)dxdydz = \int_\Omega F(x, y, z)v(x, y, z)dxdydz \ \forall v \tag{10}$$

and discretized into

$$\int_\Omega u_h(x, y, z)v(x, y, z)dxdydz = \int_\Omega F(x, y, z)v(x, y, z)dxdydz \ \forall v \in V_h \tag{11}$$

which results in a system of N_h linear equations to solve:

$$\sum_{i=1,...,N_h} \int_\Omega u_h^i e^i(x, y, z)v_j(x, y, z)dxdydz =$$
$$\int_\Omega F(x, y, z)v_j(x, y, z)dxdydz \quad j = 1, ..., N_h \tag{12}$$

We employ the adaptive mesh refinement procedure following the general *hp* adaptive algorithm described in [3,4]. In this case, we only consider *h* refinement (we only break the selected elements into eight smaller elements), and we use quadratic basis functions. In the coarse-mesh problem, we seek approximation u_h over a given computational mesh. In the fine-mesh problem, we seek approximation $u_{h/2}$ over the computational mesh that was obtained from the coarse mesh by breaking each element into eight new elements. The fine-mesh solution is more accurate, but it is more expensive to compute since we have a larger system of equations to solve in (12). Thus, our algorithm tries to find the minimal computational mesh that provides a solution with a given accuracy. We summarized the pseudo-code of the algorithm below.

```
     function adaptive_L2projection_fem (initial_mesh,desired_err,coef)
1    coarse_mesh = initial_mesh
2    repeat
3      u_h = solve coarse_mesh projection problem
4      fine_mesh = copy coarse_mesh
5      divide each element K of fine mesh into eight new elements
6      u_h/2 = solve fine_mesh projection problem
7      max_err = 0
8      for each element K of fine mesh do
9        K_err = compute relative error on K in L2 norm using u_h and u_h/2
10       if K_err > max_err then
11          max_err = K_err
12       endif
13     enddo
14     adapted_mesh = new empty_mesh
15     for each element K of coarse_mesh do
16       if K_err > coeff * max_err then
17          execute h refinement of element K
18          add K from coarse to adapted_mesh
19       endif
20     enddo
21         coarse_mesh = adapted_mesh
22   until max_err < desired_err
```

The algorithm iteratively breaks the selected finite elements into eight smaller elements to increase the accuracy of the numerical solution. We focus on a digital MRI scan data with 29 two-dimensional slices, each one with 532×565 pixels. Each pixel's intensity is a value within a range of $[0, 255]$; it is proportional to the material's density (including skull, skin, tissue, and air). The selected slices are presented in Figure 1. We construct a 3D bitmap out of these and we run the adaptive L2 projection algorithm. The iterations of the algorithm are illustrated in Figures 6–9.

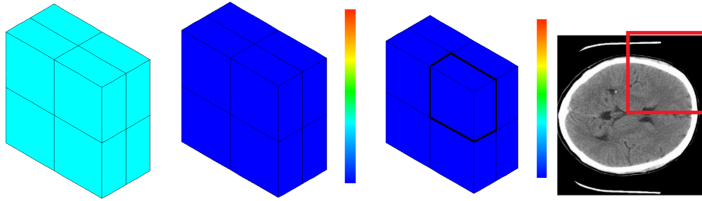**Figure 6.** Coarse mesh, coarse-mesh solution, and relationship between one finite element and MRI scan data: color bar from blue to red represents values from 0 to 255

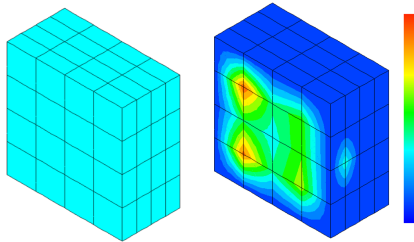**Figure 7.** Coarse- and fine-mesh solutions: color bar from blue to red represents values from 0 to 255
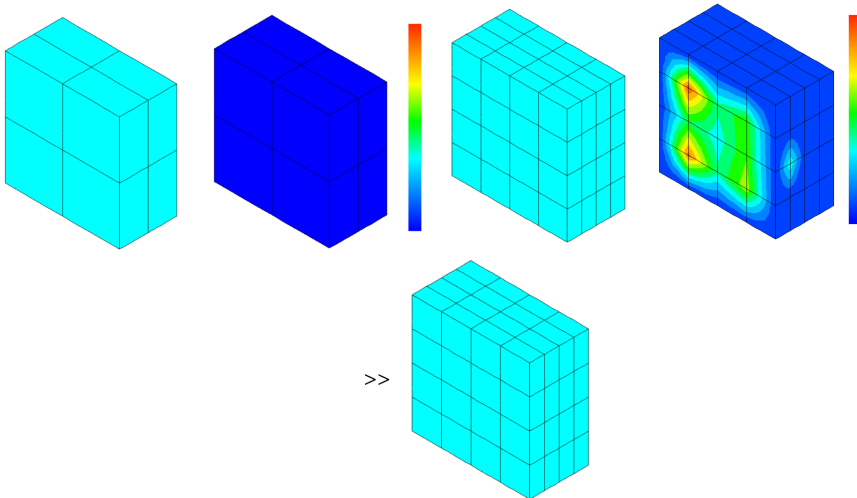

Figure 8. Comparison between coarse and fine meshes and resulting optimal mesh: color bar from blue to red represents values from 0 to 255
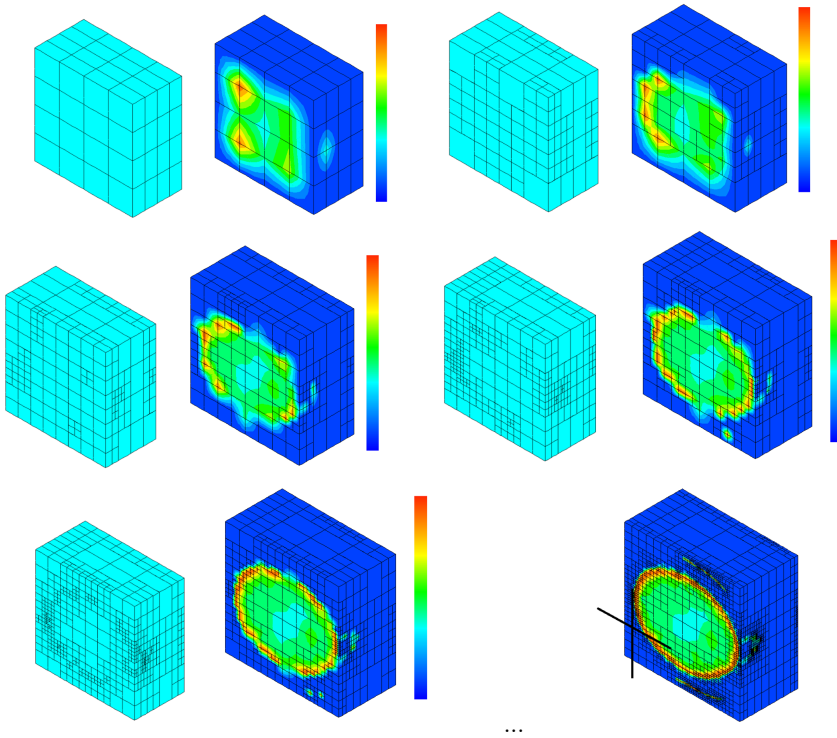
Figure 9. Following iterations of head-approximation algorithm and final approximation of MRI scan of human head, with cellphone antenna location denoted by cross: color bar from blue to red represents values from 0 to 255

We start from the initial computational mesh with eight finite elements (Line 1 of Algorithm 1). We span 27 basis functions at 27 vertices of 8 elements that are constructed as the tensor products of 3 one-dimensional linear basis functions. We also span 66 basis functions at 66 edges of 8 elements that are constructed as the tensor products of 2 one-dimensional linear basis functions and 1 one-dimensional quadratic basis function. Next, we span 36 basis functions over 36 faces of 8 elements that are constructed as the tensor products of 2 one-dimensional quadratic basis functions and 1 one-dimensional linear basis function. Finally, we span eight basis functions on eight interiors of the eight elements that are constructed as the tensor products of three quadratic one-dimensional basis functions. The mesh is illustrated in Figure 6. It covers half of the human head so that, when we solve the L2 projection problem, we can see the projection of the cross-section of the human head on one of the faces of the mesh. We solve the L2 projection problem on this coarse mesh (Line 3 of Algorithm 1). This mesh is too coarse to resolve the MRI scan data. Thus, we break each element into eight, which is illustrated in Figure 7 (Line 5 of Algorithm 1). We have 64 finite elements with basis functions spanning over the element vertices, edges,

faces, and interiors. We solve the projection problem again – this time resolving the MRI scan data on the larger mesh (Line 6 of Algorithm 1). Now, we compare the coarse- and fine-mesh solutions as presented in Figure 8 (Lines 7–12). Namely, we compute $\frac{\|u_h - u_{h/2}\|_{L_2}}{\|u_{h/2}\|_{L_2}} = \frac{\int_K (u_h - u_{h/2})^2 dx}{\int_K (u_{h/2})^2 dx}$ for each element K from the coarse mesh, where u_h is the coarse-mesh solution over element K, and $u_{h/2}$ is the fine-mesh solution projected from the fine mesh over element K. We iterate through the eight coarse-mesh elements and compute the relative error between the coarse- and fine-mesh solutions. We store the maximum element error (Line 11 of Algorithm 1). Since the error is uniformly distributed into eight elements in our example, we decide to break all of the eight elements to produce the next optimal mesh (Lines 14–21). Namely, we iterate through the eight elements of the coarse mesh, and we check if the element local relative error is greater than coeff=33 percent of the maximum error. In this case, this happens for all of the elements; so, we break all elements of the coarse mesh (in Line 17 of Algorithm 1). This mesh becomes the coarse mesh for the next iteration (Line 21 of Algorithm 1). The convergence of the algorithm is stopped when the relative error is less than 1 percent (Line 22 of Algorithm 1). The optimal meshes and the solutions generated in the particular iterations of the algorithm are illustrated in Figure 9. The final mesh is presented in Figure 9. The cross in this figure represents the assumed location of the cellphone antenna (presented in Figure 10). This continuous smooth approximation of the human head will be used for the finite element method computations presented in the following sections.
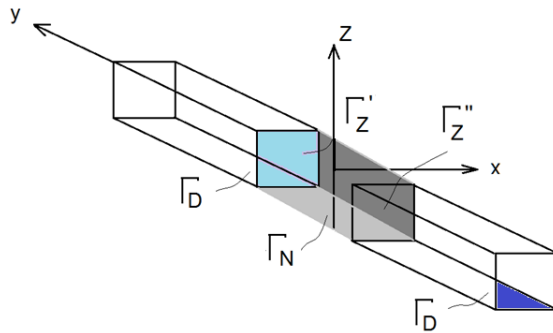
Figure 10. Modeling antenna by boundary conditions and equivalent current

3. Time-harmonic Maxwell equations

This chapter presents an overview of the time-harmonic Maxwell equations in the weak form that were utilized for the solution of the problem of the propagation of the electromagnetic waves as generated by the cellphone antenna. From the Maxwell problem results, we are particularly interested in the heat source q_{SAR} that is generated from the cellphone antenna (as computed from the time-harmonic Maxwell equations).

We now focus on the time-harmonic Maxwell equations:

$$\nabla \times \left(\frac{1}{\mu}\nabla \times \boldsymbol{E}\right) - (\omega^2\varepsilon - \omega\sigma)\boldsymbol{E} = -i\omega\boldsymbol{J}^{imp} \text{ over } \Omega \tag{13}$$

where we seek $\boldsymbol{E}$ – electric field intensity. For the time-harmonic Maxwell equations, we assume that

$$\boldsymbol{E}(\boldsymbol{x},t) = Re(\boldsymbol{E}(x)e^{i\omega t}) \tag{14}$$

(the electromagnetic field oscillates over time; at each point in space, the complex number codes the amplitude and length of the wave). We introduce μ permeability, ε permittivity, σ conductivity, $\boldsymbol{J}^{imp}$ – impressed current of the electric field, ω frequency. For weak formulation

$$\nabla \times \left(\frac{1}{\mu}\nabla \times \boldsymbol{E}\right) - (\omega^2\varepsilon - i\omega\sigma)\boldsymbol{E} = -i\omega\boldsymbol{J}^{imp} \text{ over } \Omega \tag{15}$$

we multiply by the test functions from functional space $\boldsymbol{V} = \boldsymbol{H}(curl,\Omega) = \{\boldsymbol{F} \in \boldsymbol{L}^2 : \nabla \times \boldsymbol{F} \in \boldsymbol{L}^2\}$ to get

$$\int_\Omega \nabla \times \left(\frac{1}{\mu}\nabla \times \boldsymbol{E}\right) \circ \boldsymbol{F}dx - \int_\Omega (\omega^2\varepsilon - i\omega\sigma)\boldsymbol{E} \circ \boldsymbol{F}dx =$$
$$-\int_\Omega i\omega\boldsymbol{J}^{imp} \circ \boldsymbol{F}dx \ \forall \ \boldsymbol{F} \in \boldsymbol{V} \text{ over } \Omega \tag{16}$$

We integrate by parts

$$\int_\Omega (\nabla \times \boldsymbol{E}) \circ \boldsymbol{F}dx = \int_\Omega \boldsymbol{E} \circ (\nabla \times \boldsymbol{F})dx + \int_\Gamma (n \times \boldsymbol{E})\boldsymbol{F}_t ds \ \text{ over } \ \Omega \tag{17}$$

where $\boldsymbol{F}_t = \boldsymbol{F} - (\boldsymbol{F} \circ n)n$ is the tangent component.

$$\int_\Omega \left(\frac{1}{\mu}\nabla \times \boldsymbol{E}\right) \circ (\nabla \times \boldsymbol{F})dx + \int_\Gamma \left(n \times \left(\frac{1}{\mu}\nabla \times \boldsymbol{E}\right)\right)\boldsymbol{F}_t dS$$
$$-\int_\Omega (\omega^2\varepsilon - i\omega\sigma)\boldsymbol{E} \circ \boldsymbol{F}dx = -\int_\Omega i\omega\boldsymbol{J}^{imp} \circ \boldsymbol{F}dx \ \forall \ \boldsymbol{F} \in \boldsymbol{V} \text{ over } \Omega \tag{18}$$

We introduce the Dirichlet boundary conditions on the Γ_D boundary denoted in Figure 10:

$$n \times E = 0 \text{ on } \Gamma_D \tag{19}$$

where n denotes the versor normal to the boundary, and the Neumann boundary conditions on the Γ_N boundary as denoted in Figure 10.

$$n \times \frac{1}{\mu}(\nabla \times E) = \omega J_S^{imp} \text{ on } \Gamma_N \tag{20}$$

J_S^{imp} denotes the impressed surface current. On the boundary of the domain, we introduce the perfectly matching layer (PML) trick, which allows for for closure of the

computational domain. More details in PML can be found in [15]. This represents the layer that absorbs the electromagnetic waves:

$$
\int_\Omega \left(\frac{1}{\mu}\nabla \times \boldsymbol{E}\right)\circ(\nabla \times \boldsymbol{F})dx - \int_\Omega (\omega^2\varepsilon - i\omega\sigma)\boldsymbol{E}\circ \boldsymbol{F}dx =
$$
$$
-\int_\Omega i\omega \boldsymbol{J}^{imp}\circ \boldsymbol{F}dx + \int_{\Gamma_N} i\omega \boldsymbol{J}_S^{imp}dS \ \forall \ \boldsymbol{F}\in \boldsymbol{V} \text{ over } \Omega \tag{21}
$$

We utilize the dimensionless formulation; namely, we scale all the quantities according to the following: ε_0 – electric permeability of air ($8.854 \cdot 10^{-12}$ F/m); μ_0 – magnetic permeability of air ($4.0pi \cdot 10^{-7}$ H/m); $\varepsilon_r = \frac{\varepsilon}{\varepsilon_0}$ relative electric permittivity; $\varepsilon_\delta = \frac{\sigma}{\omega\varepsilon_0}$ relative resistivity; $\mu_r = \frac{\mu}{\mu_0}$ relative magnetic permittivity $k_0 = \omega\sqrt{\varepsilon_0\mu_0}$, $Z_0 = \sqrt{\frac{\mu_0}{\varepsilon_0}}$. We get

$$
\int_\Omega \left(\frac{1}{\mu_r\mu_0}\nabla \times \boldsymbol{E}\right)\circ(\nabla \times \boldsymbol{F})dx - \int_\Omega (\omega^2\varepsilon - i\omega\sigma)\boldsymbol{E}\circ \boldsymbol{F}dx =
$$
$$
-\int_\Omega i\omega \boldsymbol{J}^{imp}\circ \boldsymbol{F}dx + \int_{\Gamma_N} i\omega \boldsymbol{J}_S^{imp}dS \ \forall \ \boldsymbol{F}\in \boldsymbol{V} \text{ over } \Omega \tag{22}
$$

with

$$
\mu_r = \frac{\mu}{\mu_0}, \varepsilon_r = \frac{\varepsilon}{\varepsilon_0}, \varepsilon_\delta = \frac{\sigma}{\omega\varepsilon_0} \tag{23}
$$

We multiply by μ_0

$$
\int_\Omega \left(\frac{1}{\mu_r}\nabla \times \boldsymbol{E}\right)\circ(\nabla \times \boldsymbol{F})dx - \int_\Omega \left(\omega^2\varepsilon\mu_0 - i\omega\sigma\mu_0\right)\boldsymbol{E}\circ \boldsymbol{F}dx =
$$
$$
-\int_\Omega i\omega\mu_0 \boldsymbol{J}^{imp}\circ \boldsymbol{F}dx + \int_{\Gamma_N} i\omega\mu_0 \boldsymbol{J}_S^{imp}dS \ \forall \boldsymbol{F}\in \boldsymbol{V} \text{ over } \Omega \tag{24}
$$

we use $k_0 = \omega\sqrt{\varepsilon_0\mu_0}$, $Z_0 = \sqrt{\frac{\mu_0}{\varepsilon_0}}$ and $k_0 Z_0 = \omega\sqrt{\varepsilon_0\mu_0}\sqrt{\frac{\mu_0}{\varepsilon_0}} = \omega\mu_0$ to get

$$
\int_\Omega \frac{1}{\mu_r}\nabla \times \boldsymbol{E}\circ\nabla \times \boldsymbol{F}dx - \int_\Omega (\omega^2\varepsilon\mu_0 - i\omega\sigma\mu_0)\boldsymbol{E}\circ \boldsymbol{F}dx =
$$
$$
-\int_\Omega ik_0 Z_0 \boldsymbol{J}^{imp}\circ \boldsymbol{F}dx + \int_\Omega ik_0 Z_0 \boldsymbol{J}_S^{imp}\circ \boldsymbol{F}dx \ \forall \boldsymbol{F}\in \boldsymbol{V} \text{ over } \Omega \tag{25}
$$

and $\varepsilon_\delta = \frac{\sigma}{\omega\varepsilon_0}$

$$
\int_\Omega \frac{1}{\mu_r}\nabla \times \boldsymbol{E}\circ\nabla \times \boldsymbol{F}dx + -\int_\Omega k_0^2(\varepsilon_r - i\varepsilon_\sigma)\boldsymbol{E}\circ \boldsymbol{F}dx =
$$
$$
-\int_\Omega ik_0 Z_0 \boldsymbol{J}^{imp}\circ \boldsymbol{F}dx + \int_\Omega ik_0 Z_0 \boldsymbol{J}_S^{imp}\circ \boldsymbol{F}dx \ \forall \boldsymbol{F}\in \boldsymbol{V} \text{ over } \Omega \tag{26}
$$

Next, we introduce the dimensionless quantities: $x := \frac{x}{a}$, a – characteristic spatial scale $a = 1\ m$; $\omega := k_0 a$ – frequency, $\boldsymbol{E} := \frac{\boldsymbol{E}}{E_0}$, E_0 – characteristic intensity of vector electric field; $\boldsymbol{J}^{imp} := \frac{aZ_0}{E_0}\boldsymbol{J}$, $\boldsymbol{J}_S^{imp} := \frac{Z_0}{E_0}\boldsymbol{J}_S$. After this scaling, we have $\omega = 3.77$ (which relates to the 1.8 GHz frequency):

$$
\int\limits_\Omega \frac{1}{\mu_r} \nabla \times \boldsymbol{E} \circ \nabla \times \boldsymbol{F} dx + - \int\limits_\Omega \omega^2(\varepsilon_r - i\varepsilon_\sigma)\boldsymbol{E} \circ \boldsymbol{F} dx =
$$
$$
- i\omega \int\limits_\Omega \boldsymbol{J}^{imp} \circ \boldsymbol{F} dx + i\omega \int\limits_\Omega \boldsymbol{J}_S^{imp} \circ \boldsymbol{F} dx \ \forall \boldsymbol{F} \in \boldsymbol{V} \ \text{over} \ \Omega
$$
(27)

In the domain of the human head, we distinguish different materials by looking at the values of the projected functions. In the original bitmap, the pixel intensity varies for the tissue, air, and bones. The monochromatic bitmap varies from 0 to 255 per pixel. We define the air area for pixels where the intensity is less than or equal to 1. We define the bones for pixel intensities that are between 1 to 240. Finally, we define the skull for pixel intensities that are greater than 240. The material data is defined as $\epsilon_r = \epsilon_\sigma = 1$ for the air, $\epsilon_r = 43.54, \epsilon_\sigma = 11.51$ for the tissue, and $\epsilon_r = 15.56; \epsilon_\sigma = 4.31$ for the skull. The cellphone antenna is modeled in the following way. We put on inside the area marked in gray $J_s^{imp} = (0, 1, 0)$, zero Dirichlet b.c. on Γ_D. Having the solution (electromagnetic field E), we can calculate q_{SAR} as average energy per unit volume $\frac{1}{2}\sigma\|E\|^2$, where $\|E\|^2$ denotes the squared L2 norm of the electric field intensity, and σ denotes the conductivity of the tissue. The definition of q_{SAR} follows the one that was presented in the Ph.D. dissertation of Kyungjoo Kim (see [14], p. 121). More details on the time-harmonic Maxwell equations can be found at [1,9,18]. We will solve the time-harmonic Maxwell equations using the finite element method and the complex version of the multi-frontal solver [5,6,10,16], augmented with the low-rank compression of the off-diagonal blocks. Next, we will compute the q_{SAR}, and we plug it in as the heat source for the heat-transfer problem.

4. Heat transfer with Pennes bioheat equations and Crank-Nicolson time-integration scheme

This chapter aims to derive a weak formulation of the Pennes bioheat equations with the Crank-Nicolson time-integration scheme, which is suitable for the adaptive processing with the hp finite element method. We will also define the model parameters based on the continuous approximation of the MRI scan data of the human head. As the heat source, we employ the q_{SAR} that was generated from the cellphone antenna (as computed from the time-harmonic Maxwell equations). The term q_{SAR} is computed based on the distribution of the electromagnetic waves (as computed by the time-harmonic Maxwell equation). This does not change over time, and we treat it as the forcing term.

We employ the Crank-Nicolson time-integration scheme since this is an unconditionally stable scheme and the time-step size there is not restricted by the CFL condition.

We start from the heat-transfer problem:

$$\frac{\partial u}{\partial t} = \nabla \cdot (k\nabla u) + ru + f \tag{28}$$

where $u(x,t)$ denotes the temperature scalar field at point x and time moment t, $k(x,t)$ is the heat conduction (assumed to be isotropic and equal in all directions at a given point x, and constant in time t), and $r(x,t)$ corresponds to the relaxation (cooling) effects due to the presence of the vascular system in the human head. We assume that both parameter fields are constant over time. We employ the Crank–Nicolson time-integration scheme with time step Δt:

$$\frac{u_{t+1} - u_t}{\Delta t} = \nabla \cdot (k\nabla u_*) + ru_* + f_t \text{ where } u_* = \frac{u_{t+1} + u_t}{2} \tag{29}$$

$$\frac{u_{t+1} - u_t}{\Delta t} = \frac{1}{2}[\nabla \cdot (k\nabla u_{t+1}) + \nabla \cdot (k\nabla u_t)] + r\frac{1}{2}(u_t + u_{t+1}) + f_t \tag{30}$$

$$u_{t+1} - \frac{1}{2}\Delta t \left(\nabla \cdot (k\nabla u_{t+1}) + lu_{t+1}\right) = u_t + \frac{1}{2}\Delta t[\nabla \cdot (k\nabla u_t)] + \Delta t[ru_t + f_t] \tag{31}$$

We multiply by test functions v and integrate to obtained the Galerkin method "in space":

$$\int_\Omega \left((1 - r\Delta t)u_{t+1} - \frac{1}{2}\Delta t \left(\nabla \cdot (k\nabla u_{t+1})\right)\right)vdx =$$
$$\int_\Omega \left(u_t + \frac{1}{2}\Delta t[\nabla \cdot (k\nabla u_t)] + \Delta t[ru_t + f_t]\right)vdx_t \ \forall \ v \in V \tag{32}$$

where

$$V = H_0(\Omega) = \left\{f : \Omega \to R : \int_\Omega |f^2| + |\nabla f^2|dx < \infty\right\} \tag{33}$$

We integrate by parts

$$\int_\Omega (1 - r\Delta t)u_{t+1}vdx + \frac{1}{2}\Delta t \int_\Omega (k\nabla u_{t+1}) \cdot \nabla vdx - \frac{1}{2}\Delta t \int_\Gamma (k\nabla u_{t+1}) \cdot ndS =$$
$$\int_\Omega u_tvdx + \frac{1}{2}\Delta t \int_\Omega (k\nabla u_t) \cdot \nabla vdx - \frac{1}{2}\Delta t \int_\Gamma (k\nabla u_t) \cdot ndS + \Delta t \int_\Omega [ru_t + f_t]vdx \tag{34}$$

and we assume that the computational domain is large enough to assume the zero Neumann boundary conditions $(k\nabla u_t) \cdot n = k\frac{\partial u_t}{\partial n} = 0$ to get

$$\int_\Omega (1 - r\Delta t)u_{t+1}vdx + \frac{1}{2}\Delta t \int_\Omega (k\nabla u_{t+1}) \cdot \nabla vdx =$$
$$\int_\Omega u_tvdx + \frac{1}{2}\Delta t \int_\Omega (k\nabla u_t) \cdot \nabla vdx + \Delta t \int_\Omega [ru_t + f_t]vdx \tag{35}$$

Particularly in the first time step,

$$
\int_\Omega (1 - r\Delta t)u_1 v dx + \frac{1}{2}\Delta t \int_\Omega (k\nabla u_1) \cdot \nabla v dx =
$$
$$
= \int_\Omega u_0 v dx + \frac{1}{2}\Delta t \int_\Omega (k\nabla u_0) \cdot \nabla v dx + \Delta t \int_\Omega [ru_0 + f_0]v dx
$$

(36)

where u_0 is a given initial configuration. We introduce bilinear form $b(u_{t+1}, v)$ and the form $l(u_t, v)$ that has a sub-form $l_1(v)$ that is linear with respect to the v argument as well as the sub-form $l_2(v_t, v)$ that is bilinear:

$$
b(u_{t+1}, v) = l(u_t, v) \ \forall \ v \in V
$$
$$
b(u_{t+1}, v) = \int_\Omega u_{t+1} v dx + \frac{1}{2}\Delta t \int_\Omega (k\nabla u_{t+1}) \cdot \nabla v dx - \Delta t \int_\Omega ru_{t+1} dx
$$
$$
l(u_t, v) = l_1(v) + l_2(u_t, v)
$$
$$
l_1(v) = \int_\Omega f_t v dx
$$
$$
l_2(u_t, v) = \int_\Omega u_t v dx + \frac{1}{2}\Delta t \int_\Omega (k\nabla u_t) \cdot \nabla v dx + \Delta t \int_\Omega ru_t v dx
$$

(37)

The above problem is satisfied for all $\forall \ v \in V$, so it is also fulfilled for the particularly selected basis functions. We take the space that was defined by the basis functions that span over the finite elements of adaptive mesh $V \supset V_h = \mathrm{span}\{e_i\}$, where e_i are the global basis functions that were obtained by gluing together the element local basis functions (also called the shape functions) that were defined over the vertices, edges, faces, and interiors of the elements. In this space, we seek the solution to $u^t \approx \sum_i a_i^t e_i^t, u^{t+1} \approx \sum_i a_i^{t+1} e_i^{t+1}$. We also test with basis functions from this space:

$$
b(u_{t+1}, v) = l_1(v) + l_2(u_t, v) \ \forall \ v \in V
$$
$$
b\Big(\sum_i a_i^{t+1} e_i^{t+1}, e_j\Big) = l_1\Big(e_j\Big) + l_2\Big(\sum_i a_i^t e_i^t, e_j\Big) \ \forall \ e_j \in V_h
$$
$$
\sum_i a_i^{t+1} b(e_i^{t+1}, e_j) = l_1(e_j + \sum_i a_i^t l_2(e_i^t, e_j) \ \forall \ e_j \in V_h
$$

(38)

We end up with the following system of linear equations:

$$
\begin{bmatrix} b(e_1^{t+1}, e_1) & \dots & b(e_N^{t+1}, e_1) \\ \dots & & \dots \\ b(e_1^{t+1}, e_N) & \dots & b(e_N^{t+1}, e_N) \end{bmatrix} \begin{Bmatrix} a_1^{t+1} \\ \dots \\ a_N^{t+1} \end{Bmatrix} = \begin{Bmatrix} l_1(e_1) + \sum_i a_i^t l_2(e_i^t, e_1) \\ \dots \\ l_1(e_1) + \sum_i a_i^t l_2(e_i^t, e_N) \end{Bmatrix}
$$

(39)

For the simulations of the heating of the human head, we transform the equations into the Pennes bioheat equation. Assuming thermal conductivity k as well as metabolism q_m, perfusion $W_b c_b$, and the heat source from the cellphone q_{SAR} (obtained by post-processing the Maxwell solution), we solve the Pennes bioheat equations:

$$\rho c \frac{\partial u}{\partial t} = \nabla \cdot (K \nabla u) + W_b c_b (u_{a0} - u) + q_m + q_{SAR} \text{ in } \Omega \qquad (40)$$

This section presents our algorithm for the simultaneous simulations of Maxwell equations and Pennes equations with the Crank-Nicolson scheme, where k is the thermal conductivity (here, it is scaled by ρ density and c specific heat), q_m is the metabolism, W_b is the blood-flow rate, c_b is the blood's specific heat capacity, and u_{a0} is the core body temperature. Namely, we define convection term $k(x,t) = \frac{K(x,t)}{\rho c}$, and we define reaction term $r(x,t) = -\frac{W_b c_b}{\rho c}$ and $f(x,t) = \frac{W_b c_b u_{a0} + q_m + q_{SAR}}{\rho c}$. The values of the parameters are summarized in Table 1. They are selected according to the type of tissue, as suggested by the L2 projection of the MRI scan data of the human head. We also assume that the air is located (where the intensity of the bitmap is ≤ 1), the skin or brain (tissue in general) (where the intensity is within a range of $[1, 240]$), and the skull (where the intensity is ≥ 240). The equations will be solved by using the multi-frontal solver [5, 6, 10, 16] augmented with the low-rank compression of the off-diagonal blocks (see Appendix).

Table 1
Material data used in simulation

Material	Air	Brain	Skull
ρ	1.16	1039	1645
C	1006	3700	1300
K	0.02	0.57	0.4
q_m	0	7100	590
$W_b c_b$	0	40,000	3300
u_{a0}	30	36.6	36.6

5. Space-adaptive time-stepping algorithm for simulation of heating of human head

This section presents the algorithm for simultaneous simulations of the Maxwell and Pennes equations using the Crank-Nicolson scheme. We utilize two versions of the three-dimensional hp-adaptation algorithm [4] with hexahedral finite elements: one implementing the Maxwell equations, and the other implementing the Pennes equations. In every adaptive iteration and every time step, we utilize the multi-frontal direct solver [5,6] implemented in the MUMPS library [2,13] augmented with the low--rank compression of the off-diagonal blocks. We start with the generation of the globally continuous h adaptive approximation of the MRI scan of the human head

using the L2 projection algorithm described in Section 2. We keep the adaptive mesh as the reference for the human head material data.

We start again from an initial mesh. We solve the Maxwell equation with the assumed cellphone source antenna and the material data that is read from the L2 projection of the MRI scan data that is measured at the Gauss integration points. Next, we run one iteration of the hp-adaptivity for the Maxwell equations. The code selects the optimal hp refinements and executes them over the selected elements. At this point, we dump out the computational mesh from the Maxwell code and dump it into the Pennes equation code. We solve the one-time step of the Pennes equations with Crank-Nicolson. We use the hp-refined mesh that was generated by the Maxwell code with the q_{SAR} that was obtained by post-processing the solution of the Maxwell equations and with the material data that was scaled according to the L2 projections of the MRI scan data that was measured at the Gauss integration points. We execute one iteration of the hp-adaptivity. We dump out the mesh from the projection code. Now, we dump the mesh into the Maxwell code and proceed with the second iterations of the hp-adaptivity for the Maxwell code. At the same time, the Pennes equation and Maxwell solutions are read by another instance of the code that is executed in the next process. This starts the next iteration of the Pennes equation in the next time step. The general idea of the algorithm can be summarized in the following pseudo-code:

```
1    generate initial_mesh
2    coarse_mesh = initial_mesh
3    time_step = MYRANK
4    do iteration of self-adaptive hp-FEM for time_step
5       if time_step is not 0 then
6          receive previous_coarse_mesh_solution from processor MYRANK-1
             (including Maxwell and Pennes solutions)
7       else
8          previous_coarse_mesh_solution = initial solution
             (including Maxwell and Pennes solutions)
9       endif
10      project previous_coarse_mesh_solution into coarse_mesh
             (including Maxwell and Pennes solutions)
11      coarse_mesh_solution = solve coarse_mesh problem
             (including Maxwell and Pennes equations)
12      send coarse_mesh_solution into processor MYRANK+1
             (including Maxwell and Pennes solutions)
13      generate fine_mesh
14      if time_step is not 0 then
15         receive previous_fine_mesh_solution from processor MYRANK-1
             (including Maxwell and Pennes solutions)
16      else
```

17   previous_fine_mesh_solution = initial solution
         (including Maxwell and Pennes solutions)
18   **endif**
19   project previous_fine_mesh_solution into fine_mesh
         (including Maxwell and Pennes solutions)
20   fine_mesh_solution = solve fine_mesh problem
         (including Maxwell and Pennes equations)
21   **send** fine_mesh_solution into processors MYRANK+1
         (including Maxwell and Pennes solutions)
22   make decision about optimal refinements
         (H1 norm for Pennes and Hcurl norm for Maxwell)
23   generate optimal_mesh from coarse_mesh
24   coarse_mesh = optimal_mesh
26 **enddo**

We design our algorithm to enable the concurrent execution of several Maxwell- and Pennes-equation solvers in particular time steps. The sends and receives communicate by dumping out and dumping in mesh files with solutions. Our algorithm can execute the coarse-mesh computations for the Maxwell and Pennes equations on Core 1 in Time Step 1. After obtaining the coarse-mesh solution, we can proceed with the fine-mesh solutions on Core 1 in Time Step 2 and the coarse-mesh solutions for both the Maxwell and Pennes equations on Core 2 in Time Step 2. Next, Core 1 can execute the optimization algorithm, selecting the optimal hp refinements of the mesh on Core 1 in Time Step 1, Core 2 can execute the fine-mesh solutions for both problems in Time Step 2, and Core 3 can start with the coarse-mesh solution in Time Step 3.

We present some snapshots from the execution of our algorithm. We employ the low-rank multi-frontal solver that is implemented in the MUMPS library [2,13]. First, we present the execution times for the sequence of the fine-mesh problems that are generated by the hp-adaptive algorithm for the Maxwell equations in Figure 11.

We follow the hp adaptive algorithm described in [4]. The horizontal axis presents the number of degrees of freedom for the fine-mesh problems. The vertical axis presents the execution times for the fine-mesh solvers. We compare the traditional multi-frontal solver and the low-rank approximation solver to deliver the linear computational cost. Second, we present the execution times for the sequence of fine meshes that are generated by the hp-adaptive solver for one-time step iteration of the Pennes equations in Figure 12. The structure of the figure is identical to the one that was generated for the Maxwell equations. Notice that we have a system of three equations for the Maxwell equations and one equation for the one time-step of the Pennes equation; therefore, the Maxwell problems are larger.
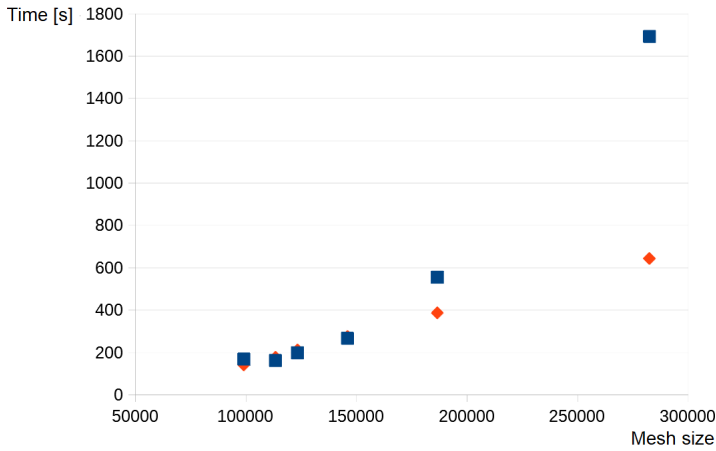
Figure 11. Reduction of computational cost for Maxwell equation; comparison of standard multi-frontal solver and low-rank solver (increasing mesh dimensions correspond to mesh-adaptation procedure)
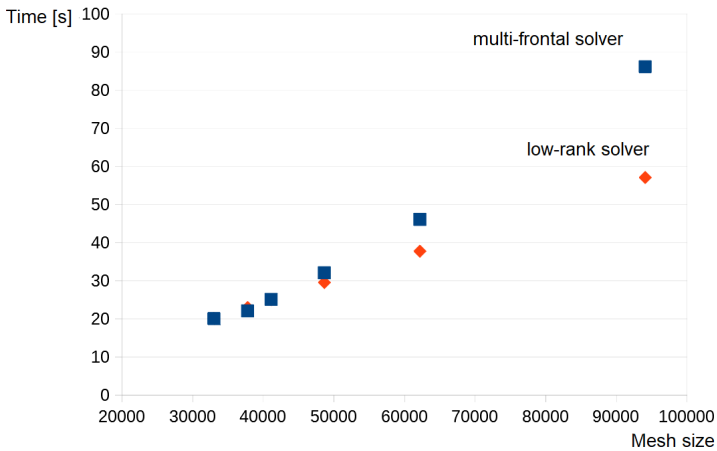

Figure 12. Reduction of computational cost for Pennes bioheat problem; comparison of standard multi-frontal solver and low-rank solver (increasing mesh dimensions correspond to mesh-adaptation procedure)

Third, we present the snapshots from the real part components, the imaginary part components, and the absolute value of the electromagnetic field that is generated by our cellphone antenna model in Figures 13–19. Fourth, we present the snapshots of the hp-adaptive grids where we solve the Pennes equations with the q_{SAR} obtained from the post-processing of the Maxwell equation solution in Figures 20–22.

Figure 13. Cross-section of mesh in OXY plane; real part of x-component of electromagnetic field generated by cellphone antenna

Figure 14. Cross-section of mesh in OXY plane; real part of y-component of electromagnetic field generated by cellphone antenna

Figure 15. Cross-section of mesh in OXY plane; real part of z-component of electromagnetic field generated by cellphone antenna

Figure 16. Cross-section of mesh in OXY plane; real part of x-component of electromagnetic field generated by cellphone antenna
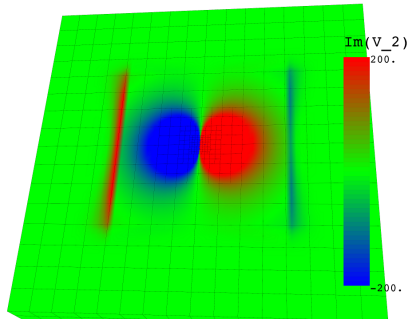
Figure 17. Cross-section of mesh in OXY plane; real part of y-component of electromagnetic field generated by cellphone antenna
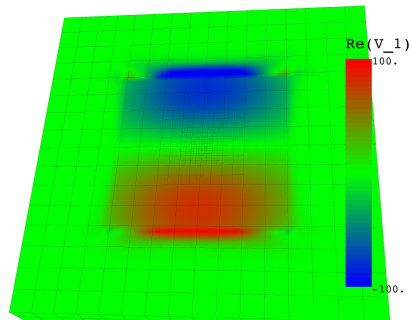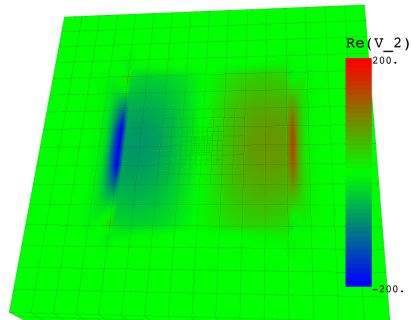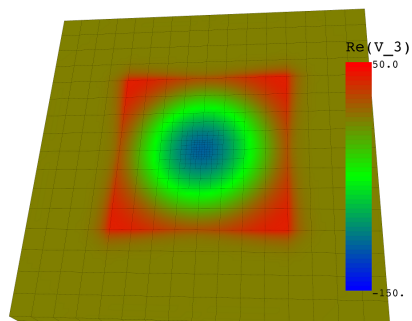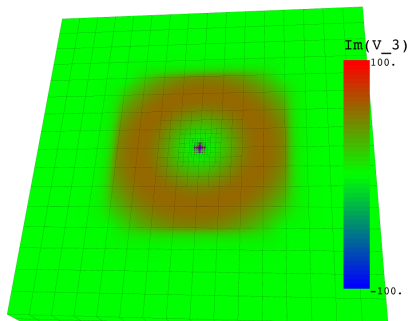

Figure 18. Cross-section of mesh in OXY plane; real part of z-component of electromagnetic field generated by cellphone antenna
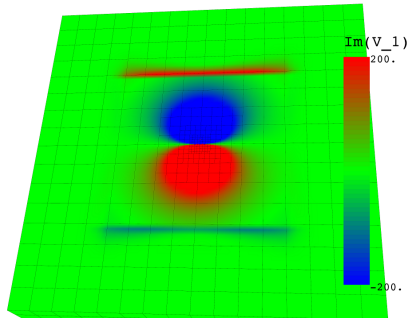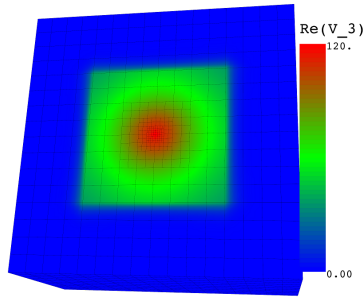
Figure 19. Cross-section of mesh in OXY plane; absolute value of z-component of electromagnetic field generated by cellphone antenna
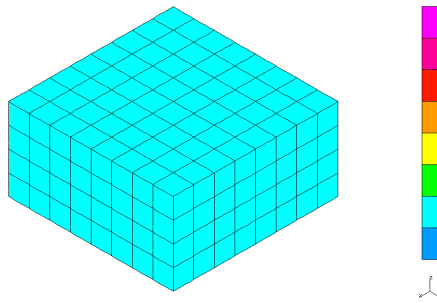
Figure 20. Cross-section of mesh in OYZ plane presenting initial mesh for Pennes equation; light-blue color represents quadratic polynomials on finite element edges and faces
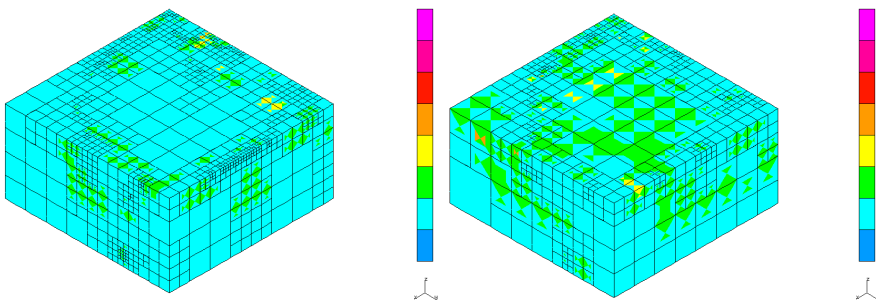
Figure 21. Cross-section in OYZ plane of two exemplary meshes generated by hp adaptive algorithm for solution of Pennes equation: dark-blue represents linear polynomials; light-blue represents quadratic polynomials on finite element edges and faces; green represents cubic polynomials on selected finite element edges and faces; brown represents fourth-order polynomials
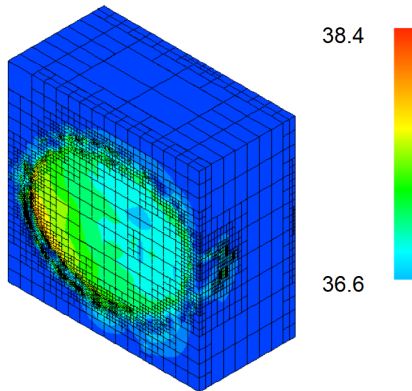
Figure 22. Cross-section of refined mesh in OXY plane and solution to Pennes equation (temperature distribution after 600 seconds of cellphone usage): red corresponds to maximum temperature of 38.4°C; light-blue corresponds to 36.6°C

From our numerical results, it follows that a 10-minute (600-second) exposure to cellphone radiation implies up to a 2°C increase in the temperature of the brain in the range close to the cellphone (compare Figure 22).

6. Conclusions

We presented an overview of the formulations and adaptive algorithms that are dedicated to modeling the electromagnetic waves' influence on the human head. Our simulations used the MRI scan data of the second author. We derived the time--harmonic Maxwell equations and the non-stationary heat-transfer equations (both in their weak forms). For the former, we employed the Crank-Nicolson time-integration scheme. We employed the MUMPS solver for the resulting systems' solution with the low-rank compression of blocks and fast factorizations. We concluded that 15 minutes of exposure to a 1.8 GHz antenna of one watt of power increases the brain tissue temperature to up to 38.4°C. Future work will include replacing the multi-frontal solver by an alternating direction solver, incorporating the direction-splitting procedure for the time-harmonic Maxwell equations and heat-transfer equations.

Acknowledgements

Appendix: Fast solver based on Singular Value Decomposition

Each matrix $A \in \mathcal{R}^{n \times m}$ can be decomposed:

$$A = U\Sigma V^T$$

where $U \in \mathcal{R}^{n \times n}$, $U^T = U^{-1}$, $V \in \mathcal{R}^{m \times m}$, $V^T = V^{-1}$ (orthogonal matrix), where $\Sigma \in \mathcal{R}^{n \times m}$ are the singular values matrices, σ_i are the singular values

$$\Sigma = \begin{bmatrix} \sigma_1 & & & & & & \\ & \sigma_2 & & & & & \\ & & \ddots & & & & \\ & & & \sigma_r & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{bmatrix}$$

$r = rank(A)$ = number of linearly independent rows. Each matrix $A \in \mathcal{R}^{n \times m}$ is orthogonally equivalent to diagonal matrix. For example:

$$A = \left[\left[u_1 \right] \left[u_2 \right] \left[u_3 \right] \left[u_4 \right] \right] \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \\ 0 & 0 & 0 & \sigma_4 \end{bmatrix} \begin{bmatrix} [& v_1 &] \\ [& v_2 &] \\ [& v_3 &] \\ [& v_4 &] \end{bmatrix}$$

where $A \in \mathcal{R}^{4 \times 4}$, $\mathcal{U} \in \mathcal{R}^{4 \times 4}$, $\mathcal{V} \in \mathcal{R}^{4 \times 4}$, $\sigma_i = \sqrt{\lambda_i}$ are root of singular values λ_i of square matrix AA^T, where $u_i \in \mathcal{R}^{4 \times 1}$ are eigenvectors of AA^T (left singular values of A), $v_i \in \mathcal{R}^{4 \times 1}$ are eigenvectors of $A^T A$ (right singular values of A); e.g.,

$$\left[\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \end{bmatrix} \begin{bmatrix} 9 \\ 10 \\ 11 \\ 12 \end{bmatrix} \begin{bmatrix} 13 \\ 14 \\ 15 \\ 16 \end{bmatrix} \right] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.000002 & 0 \\ 0 & 0 & 0 & 0.0000001 \end{bmatrix} \begin{bmatrix} [1 & 2 & 3 & 4] \\ [5 & 6 & 7 & 8] \\ [9 & 10 & 11 & 12] \\ [13 & 14 & 15 & 16] \end{bmatrix}$$

Now, we can assume $\epsilon = 0.00001$, and we can drop the singular values that are less than ϵ together with the corresponding rows and columns.

$$\left[\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \end{bmatrix} \right] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \left[\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} \right]$$

We can remove the singular value matrix and just multiply the rows.

$$
\begin{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} & \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} \\ \begin{bmatrix} 0.5 \cdot 5 & 0.5 \cdot 6 & 0.5 \cdot 7 & 0.5 \cdot 8 \end{bmatrix} \end{bmatrix}
$$

We obtain the compression of matrix A by a sequence of rank-1-updates. The SVD decomposition of the matrix of size N costs $O(N^3)$, so we can apply this algorithm for small blocks of the global matrix.

$$
\begin{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} & \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} \\ \begin{bmatrix} 2.5 & 3 & 3.5 & 4 \end{bmatrix} \end{bmatrix} =
$$

$$
\begin{bmatrix}
1 \cdot 1 + 5 \cdot 2.5 & 1 \cdot 2 + 5 \cdot 3 & 1 \cdot 3 + 5 \cdot 3.5 & 1 \cdot 4 + 5 \cdot 4 \\
2 \cdot 1 + 6 \cdot 2.5 & 2 \cdot 2 + 6 \cdot 3 & 2 \cdot 3 + 6 \cdot 3.5 & 2 \cdot 4 + 6 \cdot 4 \\
3 \cdot 1 + 7 \cdot 2.5 & 3 \cdot 2 + 7 \cdot 3 & 3 \cdot 3 + 7 \cdot 3.5 & 3 \cdot 4 + 7 \cdot 4 \\
4 \cdot 1 + 8 \cdot 2.5 & 4 \cdot 2 + 8 \cdot 3 & 4 \cdot 3 + 8 \cdot 3.5 & 4 \cdot 4 + 8 \cdot 4
\end{bmatrix}
$$

The low-rank r compression of two matrices of dimension $n \times m$ and $m \times l$ reduces the cost of multiplication from $O(nml)$ to $O(srl)$. Here, r and s are the numbers of rows and columns, respectively, after the compression. Namely, r is the number of singular values that are larger than ϵ in the first matrix, and s is the number of singular values that are larger than ϵ in the second matrix. This reduction is presented in Figure 23. It can be generalized to blocked matrices (see Figure 24).

For the large-matrix processing, the off-diagonal blocks are low-rank compressed. Having the matrix $A \in \mathcal{R}^{N \times N}$ decomposed into blocks $A(k,l) \in \mathcal{R}^{n \times n}$ (where the off-diagonal blocks are compressed by the low-rank compression SVD algorithm), we can employ the low-rank multi-frontal solver that is implemented in the MUMPS library [2, 13]. The benefits from the low-rank approximation solver are obvious; it delivers a linear $\mathcal{O}(N)$ computational cost solution in the Maxwell problem as well as in every time step of the Crank-Nicolson time-integration scheme. We illustrate this benefit for both problems in Figures 11–12.
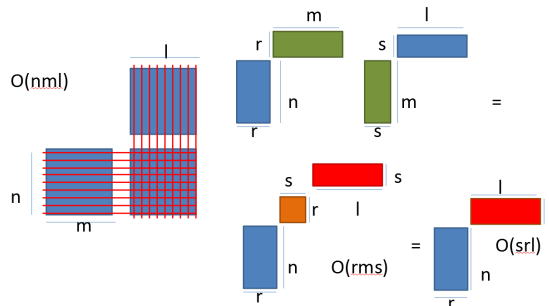

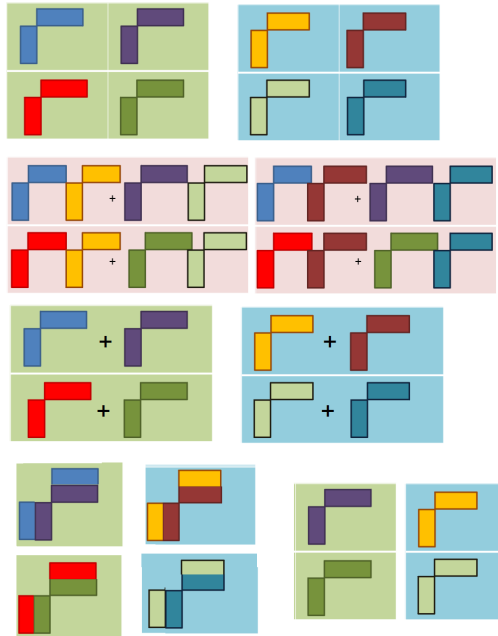**Figure 23.** Benefit from multiplication of low-rank compressed matrices

Figure 24. Recursive generalization of multiplications of low-rank compressed matrices

References

[1] Albers B., Savidis S., Tasan H.E., von Estorff O., Gehlken M.: BEM and FEM results of displacements in a poroelastic column, *International Journal of Applied Mathematics and Computer Science*, vol. 22(4), pp. 883–896, 2012.

[2] Amestoy P.R., Buttari A., L'Excellent J.Y., Mary T.: Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures, *ACM Transactions on Mathematical Software*, vol. 45(1), pp. 1–26, 2019. doi: 10.1145/3242094.

[3] Demkowicz L.: *Computing with hp-ADAPTIVE FINITE ELEMENTS, Vol. I. One and Two Dimensional Elliptic and Maxwell Problems*, Chapman and Hall/CRC Applied Mathematics and Nonlinear Science, 2006.

[4] Demkowicz L., Kurtz J., Pardo D., Paszyński M., Rachowicz W., Zdunek A.: *Computing with hp-ADAPTIVE FINITE ELEMENTS, Vol. II. Frontiers. Three Dimensional Elliptic and Maxwell Problems with Applications*, Chapman and Hall/CRC Applied Mathematics and Nonlinear Science, 2007.

[5] Duff I.S., Reid J.K.: The Multifrontal Solution of Indefinite Sparse Symmetric Linear, *ACM Transactions on Mathematical Software*, vol. 9(3), pp. 302–325, 1983.

[6] Duff I.S., Reid J.K.: The Multifrontal Solution of Unsymmetric Sets of Linear Equations, *Journal on Scientific and Statistical Computing*, vol. 5, pp. 633–641 1984.

[7] Forouharmajd F., Ebrahimi H., Pourabdian S.: Mobile Phone Distance From Head and Temperature Changes of Radio Frequency Waves on Brain Tissue, *International Journal of Preventive Medicine*, vol. 9(1), p. 61, 2018. doi: 10.4103/ijpvm.IJPVM_70_17.

[8] Garcia-Castillo L., Gomez-Revuelto I., Amor-Martin A., Łoś M., Paszyński M.: Algorithm for simultaneous adaptation and time step iterations for the electromagnetic waves propagation and heating of the human head induced by cell phone, *Procedia Computer Science*, vol. 108, pp. 2448–2452, 2017.

[9] Garcia-Dooro D.: *A new software suite for electromagnetics*, Ph.D. thesis, Universidad Carlos III de Madrid, 2014.

[10] Geng P., Oden T.J., van de Geijn R.A.: A parallel multifrontal algorithm and its implementation, *Computer Methods in Applied Mechanics and Engineering*, vol. 149(1–4), pp. 289–301, 1997. doi: 10.1016/S0045-7825(97)00052-2.

[11] Hughues T.J.R.: *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover Civil and Mechanical Engineering, 2000.

[12] Iqbal-Faruque M., Aisyah-Husni N., Ikbal-Hossain M., Tariqul-Islam M., Misran N.: Effects of Mobile Phone Radiation onto Human Head with Variation of Holding Cheek and Tilt Positions, *Journal of Applied Research and Technology*, vol. 12(5), pp. 871–876, 2014. doi: 10.1016/S1665-6423(14)70593-0.

[13] Jeannerod C.P., Mary T., Pernet C., Roche D.S.: Improving the Complexity of Block Low-Rank Factorizations with Fast Matrix Arithmetic, *SIAM Journal on Matrix Analysis and Applications*, vol. 40(4), pp. 1478–1496, 2019. doi: 10.1137/19M1255628.

[14] Kyunogjoo K.: *Finite element modeling of the radiation and induced heat transfer in the human body*, Ph.D. thesis, The University of Texas at Austin, 2013.

[15] Michler C., Demkowicz L., Kurtz J., Pardo D.: Improving the performance of perfectly matched layers by means of hp-adaptivity, *Numerical Methods for Partial Differential Equations*, vol. 23(4), pp. 832–858, 2007. doi: 10.1002/num.20252.

[16] Paszyński M.: *Fast Solvers for Mesh Based Computations*, Taylor & Francis, CRC Press, 2016.

[17] Pilat A.: FEMLab software applied to Active Magnetic Bearing analysis, *International Journal of Applied Mathematics and Computer Science*, vol. 14(4), pp. 497–501, 2004.

[18] Salazar-Palma M., Sarkar T.K., Garcia-Castillo L.E., Roy T., Djordjevic A.R.: *Iterative and Self-Adaptive Finite-Elements in Electromagnetic Modeling*, Artech House Publishers, 1998.

[19] Schaefer R., Łoś M., Sieniek M., Demkowicz L., Paszyński M.: Quasi-linear computational cost adaptive solvers for three dimensional modeling of heating of a human head induced by cell phone, *Journal of Computational Science*, vol. 11, pp. 163–174, 2015.

[20] Stanković V., Jovanović D., Krstić D., Marković V., Cvetković N.: Temperature distribution and Specific Absorption Rate inside a child's head, *International Journal of Heat and Mass Transfer*, vol. 104, pp. 559–565, 2017. doi: 10.1016/j.ijheatmasstransfer.2016.08.094.

Affiliations

Barbara Głut
AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Krakow, Poland, glut@agh.edu.pl, ORCID ID: https://orcid.org/0000-0003-1522-873X

Maciej Paszyński
AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Krakow, Poland, paszynsk@agh.edu.pl, ORCID ID: https://orcid.org/0000-0003-1522-873X