Linear computational cost implicit variational splitting solver with non-regular material data for parabolic problems

Paweł Maczuga[1], Maciej Paszyński[1][0000−0001−7766−6052], and Victor Calo[2][0000−0002−1805−4045]

[1] AGH University of Science and Technology, Kraków, Poland
maciej.paszynski@agh.edu.pl, pmaczuga97@gmail.com
[2] Curtin University, Perth, Australia
vmcalo@gmail.com

Abstract. We employ a variational splitting for the Crank-Nicolson method and Pennes bioheat equation modeling the heating of the human head as a result of the cellphone antenna radiation. The solution of the system of equations resulting from the 3D discretization of the implicit time integration scheme with the Crank-Nicolson method has $\mathcal{O}(N^2)$ complexity using direct solver, resulting in the exact solution. Iterative solvers (e.g., multi-grid solvers) deliver $\mathcal{O}(Nk)$ computational cost resulting in an approximate solution. The alternating direction implicit solver delivers $\mathcal{O}(N)$ complexity instead; it provides the exact solution (as the direct solver). Still, it requires a regular tensor product structure of the material data. In this paper, we propose a method for generalizing the linear computational cost alternating direction implicit solver using the Crank-Nicolson scheme into non-regular material data.

Keywords: Pennes problem, Variational splitting, Implicit method, Non-regular material data, Linear computational cost

1 Introduction

Splitting methods modify the original linear systems of equations seeking to reduce computation costs [11]. The methods have been originally proposed for finite differences [8, 9] and later generalized for isogeometric analysis (IGA) [3]. We focus on parabolic equations discretized in space using tensor-product IGA grid. We adopt an implicit time integration scheme based on the Crank-Nicolson method. We employ the Kronecker product decomposition of the matrix $\mathcal{M} = \mathcal{M}_x \otimes \mathcal{M}_y \otimes \mathcal{M}_z$, that allows for linear cost factorization $\mathcal{M}^{-1} = \mathcal{M}_x^{-1} \otimes \mathcal{M}_y^{-1} \otimes \mathcal{M}_z^{-1}$. The results detailed in [1, 5–7] show that the Kronecker product-based solvers result in a linear cost for every time step. They offer an attractive alternative for multi-frontal solver algorithms [2]. Splitting solvers require the regular tensor product structure of the material data. We propose a method for generalizing the solvers into non-regular material data, preserving the linear cost. We present exemplary numerical results of the heating of the

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_18

human head based on a non-regular MRI scan and Pennes equations [10], and the cellphone radiation data available from [4]. Our method can be employed to a broad spectrum of applications of the variational-splitting solvers [1, 5, 6].

2 Varying coefficients in alternating directions solver

Lemma 1. *It is possible to factorize in a linear $\mathcal{O}(N)$ computational cost a system of equations, resulting from the problem $\frac{\partial u(x,y;t)}{\partial t} - \nabla\left(\epsilon(x,y;t)\nabla u(x,y;t)\right) = f(x,y;t)$ discretized in space with B-spline basis functions and in time with Crank-Nicolson implicit scheme, having the non-regular material data $\epsilon(x,y;t)$.*

Proof. We discretize and test with B-splines. We however cut each one into $N_q N_q$ pieces (see Fig.1), so we have a total of $\{\mathcal{I}_k^x(x)\mathcal{I}_l^y\}_{k=1,...,N_q N_x; l=1,...,N_q N_y}$ test functions. In such a way we obtain a weak formulation. Each equation with $B_m^x B_n^y$ test function is replaced by $N_q N_q$ equations with suitable test functions $\{\mathcal{I}_k^x(x)\mathcal{I}_l^y\}_{k=1,...,N_q; l=1,...,N_q}$. In the limit of N_q the integrals equations converge

$$\sum_{k=1,...,N_q; l=1,...,N_q} \int \mathcal{F}(x,y)\mathcal{I}_k^x\mathcal{I}_l^y dxdy \xrightarrow[N_q\to\infty]{} \int \mathcal{F}(x,y)B_m^x B_n^y dxdy \tag{1}$$

With our partitioned test functions, *we assume* that each equation can be approximated by one quadrature point. We generate a *rectangular* system of equations, with $N \times N_q N_q$ equations and N unknowns, repeated for $N_q N_q$ sets of N equations (see Fig. 1). Later, we will replace N equations from $N N_q N_q$ system of equations by a linear combination of $N_q N_q$ equations to obtain quadratic $N \times N$ system. We will glue the test functions $B_m^x B_n^y$ back together from the pieces $\{\mathcal{I}_k^x\mathcal{I}_l^y\}_{k=1,...,N_q, l=1,...,N_q}$. The left-hand side is of the following form

$$\begin{aligned} LHS = \sum_{i,j} \Bigg(\int B_i^x B_j^y \mathcal{I}_k^x \mathcal{I}_l^y + \eta \int \epsilon(x,y)\partial_x B_i^x B_j^y \partial_x \mathcal{I}_k^x \mathcal{I}_l^y + \\ \eta \int \epsilon(x,y) B_i^x \partial_y B_j^y \mathcal{I}_k^x \partial_y \mathcal{I}_l^y \Bigg) \mathcal{U}_{i,j} \quad \forall k = 1,...,N_x N_q; l = 1,...,N_y N_q \end{aligned} \tag{2}$$

and the right-hand side terms, one for each test function $\mathcal{I}_k^x\mathcal{I}_l^y$, are derived from the Crank-Nicolson scheme. We introduce quadrature points (ξ_k, ξ_l), one for each test function $\mathcal{I}_k^x(x)\mathcal{I}_l^y(y)$, and the Kronecker product approximation of the left-hand side, ignoring terms of higher order with respect to η

$$\begin{aligned} LHS = \sum_{i=1,...,N_x; j=1,...,N_y} w_k w_l \big(B_i^x(\xi_k)\mathcal{I}_k^x(\xi_k) B_j^y(\xi_l)\mathcal{I}_l^y(\xi_l) + \\ \eta\epsilon(\xi_k,\xi_l) B_i^x(\xi_k)\mathcal{I}_k^x(\xi_k)\partial_y B_j^y(\xi_l)\partial_y \mathcal{I}_l^y(\xi_l) + \\ \eta\epsilon(\xi_k,\xi_l)\partial_x B_i^x(\xi_k)\partial_x \mathcal{I}_k^x(\xi_k) B_j^y(\xi_l)\mathcal{I}_l^y(\xi_l)\big)\mathcal{U}_{i,j} \approx \end{aligned} \tag{3}$$

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_18

$$\sum_{i=1,...,N_x;j=1,...,N_y} w_k w_l \left(B_i^x(\xi_k)\mathcal{I}_k^x(\xi_k) + \eta\epsilon_{k,l}\partial_x B_i^x(\xi_k)\partial_x\mathcal{I}_k^x(\xi_k)\right) \left(B_j^y(\xi_l)\mathcal{I}_l^y(\xi_l) + \eta\epsilon_{k,l}\partial_y B_j^y(\xi_l)\partial_y\mathcal{I}_l^y(\xi_l)\right)\mathcal{U}_{i,j} = RHS \quad \forall k = 1,...,N_xN_q; l = 1,...,N_yN_q \tag{4}$$

Notice that $\epsilon_{k,l} = \epsilon(\xi_k, \xi_l)$ is a given constant value of material data at the quadrature point defined for a test function $\mathcal{I}_k^x\mathcal{I}_l^y$.

$$\hat{\mathcal{A}}_k = w_k \times \begin{bmatrix} (B_1^x(\xi_1)\mathcal{I}_1^x(\xi_1) + \eta\epsilon_{1,k}\partial_x B_1^x(\xi_1)\partial_x\mathcal{I}_1^x(\xi_1))\cdots \\ \cdots \left(B_{N_x}^x(\xi_1)\mathcal{I}_1^x(\xi_k) + \eta\epsilon_{1,1}\partial_x B_{N_x}^x(\xi_1)\partial_x\mathcal{I}_1^x(\xi_1)\right) \\ \cdots \\ \left(B_1^x(\xi_{N_xN_q})\mathcal{I}_{N_xN_q}^x(\xi_{N_xN_q}) + \eta\epsilon_{N_xN_q,k}\partial_x B_1^x(\xi_{N_xN_q})\partial_x\mathcal{I}_{N_xN_q}^x(\xi_{N_xN_q})\right)\cdots \\ \cdots \left(B_{N_x}^x(\xi_{N_xN_q})\mathcal{I}_{N_xN_q}^x(\xi_{N_xN_q}) + \eta\epsilon_{N_xN_q,k}\partial_x B_{N_x}^x(\xi_{N_xN_q})\partial_x\mathcal{I}_{N_xN_q}^x(\xi_{N_xN_q})\right) \end{bmatrix} \tag{5}$$

for each $k = 1,...,N_yN_q$. Here, to save space, we have partitioned the first and the last row of the matrix into two consecutive rows. The size of $\mathcal{A}_k$ is $N_xN_q \times N_x$.

$$\hat{\mathcal{B}}_{l,j} = w_l \begin{bmatrix} \left(B_j^y(\xi_l)\mathcal{I}_l^y(\xi_l) + \eta\hat{\epsilon}_{1,l}\partial_y B_j^y(\xi_l)\partial_y\mathcal{I}_l^y(\xi_l)\right) & \cdots & 0 \\ \cdots & & \\ 0 & \cdots & \left(B_j^y(\xi_l)\mathcal{I}_l^y(\xi_l) + \eta\hat{\epsilon}_{N_x,l}\partial_y B_j^y(\xi_l)\partial_y\mathcal{I}_l^y(\xi_l)\right) \end{bmatrix} \tag{6}$$

for each $l = 1,...,N_yN_q$, $j = 1,...,N_y$. The size of each $\mathcal{B}_{l,j}$ block is $N_x \times N_x$. Using this notation, see Figure 1, we can rewrite our system in the following form $\mathcal{ABU} = \mathcal{F}$

$$\begin{bmatrix} \hat{\mathcal{A}}_1 \cdots & 0 \\ \cdots \cdots & \cdots \\ 0 & \cdots \hat{\mathcal{A}}_{N_yN_q} \end{bmatrix} \begin{bmatrix} \hat{\mathcal{B}}_{1,1} & \cdots & \hat{\mathcal{B}}_{1,N_y} \\ \cdots & \cdots & \cdots \\ \hat{\mathcal{B}}_{N_yN_q,1} & \cdots & \hat{\mathcal{B}}_{N_yN_q,N_y} \end{bmatrix} \begin{bmatrix} \mathcal{U}_{1,1} \\ \cdots \\ \mathcal{U}_{N_x,N_y} \end{bmatrix} = \begin{bmatrix} \hat{\mathcal{F}}_{1,1} \\ \cdots \\ \hat{\mathcal{F}}_{N_xN_q,N_yN_q} \end{bmatrix} \tag{7}$$

where $\hat{\mathcal{F}}_{k,l} = \int \mathcal{RHS}_{k,l}(x,y)\mathcal{I}_k^x\mathcal{I}_j^y$ for $k = 1,...,N_xN_q$, $l = 1,...,N_yN_q$, and $\mathcal{RHS}_{k,l}$ is defined according to the Crank-Nicolson employed time integration scheme. $\mathcal{A}$ matrix has dimension $N_xN_qN_yN_q \times N_xN_yN_q$, $\mathcal{B}$ matrix has dimension $N_xN_qN_y \times N_xN_y$, $\mathcal{U}$ has dimension N_xN_y and $\mathcal{F}$ have dimensions $N_xN_qN_yN_q$.

$$\begin{bmatrix} \hat{\mathcal{G}}_1 \\ \cdots \\ \hat{\mathcal{G}}_{N_x,N_yN_q} \end{bmatrix} = \begin{bmatrix} \hat{\mathcal{B}}_{1,1} & \cdots & \hat{\mathcal{B}}_{1,N_y} \\ \cdots & \cdots & \cdots \\ \hat{\mathcal{B}}_{NyNq,1} & \cdots & \hat{\mathcal{B}}_{N_yN_q,N_y} \end{bmatrix} \begin{bmatrix} \mathcal{U}_{1,1} \\ \cdots \\ \mathcal{U}_{N_x,N_y} \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} \hat{\mathcal{A}}_1 \cdots & 0 \\ \cdots \cdots & \cdots \\ 0 & \cdots \hat{\mathcal{A}}_{N_yN_q} \end{bmatrix} \begin{bmatrix} \hat{\mathcal{G}}_1 \\ \cdots \\ \hat{\mathcal{G}}_{N_x,N_yN_q} \end{bmatrix} = \begin{bmatrix} \hat{\mathcal{F}}_1 \\ \cdots \\ \hat{\mathcal{F}}_{N_xN_q,N_yN_q} \end{bmatrix} \tag{9}$$

We sum up blocks of N_q rows to recover B-spline test functions along x

$$\begin{bmatrix} \mathcal{A}_1 \cdots & 0 \\ \cdots \cdots & \cdots \\ 0 & \cdots \mathcal{A}_{N_y} \end{bmatrix} \begin{bmatrix} \hat{\mathcal{G}}_1 \\ \cdots \\ \hat{\mathcal{G}}_{N_x,N_yN_q} \end{bmatrix} = \begin{bmatrix} \mathcal{F}_1 \\ \cdots \\ \mathcal{F}_{N_x,N_yN_q} \end{bmatrix} \tag{10}$$

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_18

$$\mathcal{A}_k = w_k \begin{bmatrix} (B_1^x(\xi_k)B_1^x(\xi_k) + \eta\hat{\epsilon}_{1,k}\partial_x B_1^x(\xi_k)\partial_x B_1^x(\xi_k)) & \cdots \\ \cdots & \left(B_{N_x}^x(\xi_k)B_1^x(\xi_k) + \eta\hat{\epsilon}_{1,k}\partial_x B_{N_x}^x(\xi_k)\partial_x B_1^x(\xi_k)\right) \\ & \cdots \\ \left(B_1^x(\xi_k)B_{N_x}^x(\xi_k) + \eta\hat{\epsilon}_{N_x,k}\partial_x B_1^x(\xi_k)\partial_x B_{N_x}^x(\xi_k)\right) & \cdots \\ \cdots & \left(B_{N_x}^x(\xi_k)B_{N_x}^x(\xi_k) + \eta\hat{\epsilon}_{N_x,k}\partial_x B_{N_x}^x(\xi_k)\partial_x B_{N_x}^x(\xi_k)\right) \end{bmatrix} \tag{11}$$

for each $k = 1, ..., N_x$. Here, again, to save space, we have partitioned the first and the last row of the matrix into two consecutive rows. Here

$$\left(B_i^x(\xi_k)B_j^x(\xi_k) + \eta\hat{\epsilon}_{i,k}\partial_x B_i^x(\xi_k)\partial_x B_j^x(\xi_k)\right) \approx \sum_{r=1,...,N_q} (B_i^x(\xi_k)\mathcal{I}_r^x(\xi_k) + \eta\epsilon_{r,k}\partial_x B_i^x(\xi_k)\partial_x \mathcal{I}_r^x(\xi_k)) \tag{12}$$

for $i = 1, ..., N_x$ and in particular $\hat{\epsilon}_{i,k} = \frac{\sum_{r=1,...,N_q} \epsilon_{r,k}}{N_q}$ is selected in such a way that these sums are as close as possible. Moreover $\mathcal{F}_{k,l} = \int \mathcal{RHS}_{k,l}(x,y)B_k^x B_l^y$ for $k = 1, ..., N_x$, $l = 1, ..., N_y N_q$. Notice, that this summation of rows does not change the number of unknowns, which is related to the number of columns in matrix $\mathcal{A}$. We solve this system for $\left[\hat{\mathcal{G}}_1 \cdots \hat{\mathcal{G}}_{N_x,N_y N_q}\right]^T$. Next, we take

$$\begin{bmatrix} \hat{\mathcal{B}}_{1,1} & \cdots & \hat{\mathcal{B}}_{1,N_y} \\ \cdots & \cdots & \cdots \\ \hat{\mathcal{B}}_{N_y N_q,1} & \cdots & \hat{\mathcal{B}}_{N_y N_q,N_y} \end{bmatrix} \begin{bmatrix} \mathcal{U}_{1,1} \\ \cdots \\ \mathcal{U}_{N_x,N_y} \end{bmatrix} = \begin{bmatrix} \hat{\mathcal{G}}_1 \\ \cdots \\ \hat{\mathcal{G}}_{N_x,N_y N_q} \end{bmatrix} \tag{13}$$

and we sum-up N_q rows to recover the full B-splines along y

$$\begin{bmatrix} \mathcal{B}_{1,1} & \cdots & \mathcal{B}_{1,N_y} & \\ \cdots & \cdots & \cdots & \cdots \\ \mathcal{B}_{N_y,1} & \cdots & \mathcal{B}_{N_y,N_y} & \end{bmatrix} \begin{bmatrix} \mathcal{U}_{1,1} \\ \cdots \\ \mathcal{U}_{N_x,N_y} \end{bmatrix} = \begin{bmatrix} \mathcal{G}_1 \\ \cdots \\ \mathcal{G}_{N_x,N_y} \end{bmatrix} \tag{14}$$

$$\mathcal{B}_{l,j} = \begin{bmatrix} \left(\int_y B_1^y(\xi_l)B_l^y(\xi_l) + \int_y \eta\tilde{\epsilon}_{1,l}\partial_y B_j^y(\xi_l)\partial_y B_l^y(\xi_l)\right) & \cdots & 0 \\ & \cdots & \\ 0 & \cdots & \left(\int_y B_j^y(\xi_l)B_l^y(\xi_l) + \int_y \eta\tilde{\epsilon}_{N_x,l}\partial_y B_j^y(\xi_l)\partial_y B_l^y(\xi_l)\right) \end{bmatrix} \tag{15}$$

for each $l = 1, ..., N_y$, $j = 1, ..., N_y$. Here

$$\left(B_j^y(\xi_l)B_j^y(\xi_l) + \eta\tilde{\epsilon}_{j,l}\partial_y B_j^x(\xi_l)\partial_y B_j^y(\xi_l)\right) \approx \sum_{r=1,...,N_q} \left(B_j^y(\xi_l)\mathcal{I}_r^x(\xi_l) + \eta\hat{\epsilon}_{r,l}\partial_y B_j^y(\xi_l)\partial_y \mathcal{I}_r^x(\xi_l)\right) \tag{16}$$

for $j = 1, ..., N_y$ and in particular $\tilde{\epsilon}_{j,l} = \frac{\sum_{r=1,...,N_q} \hat{\epsilon}_{r,l}}{N_q}$ is selected in such a way that these sums are as close as possible. Finally, we solve for $\mathcal{U}_1, ..., \mathcal{U}_{N_x,N_y}$

$$\begin{bmatrix} \mathcal{B}_1 & \cdots & \mathcal{B}_{N_y} \\ \cdots & \cdots & \cdots & \cdots \\ \mathcal{B}_1 & \cdots & \mathcal{B}_{N_y} \end{bmatrix} \begin{bmatrix} \mathcal{U}_1 \\ \cdots \\ \mathcal{U}_{N_x N_y} \end{bmatrix} = \begin{bmatrix} \mathcal{G}_1 \\ \cdots \\ \mathcal{G}_{N_x N_y} \end{bmatrix} \tag{17}$$

Both systems (10) and (17) can be solved in a linear computational cost due to the banded structures of matrices build with one-dimensional B-splines.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_18

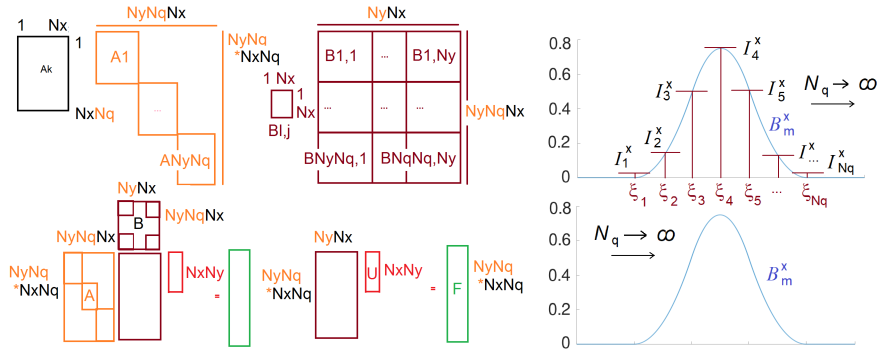

Fig. 1: Partitioning of a matrices. Partition of a test function B_m^x.

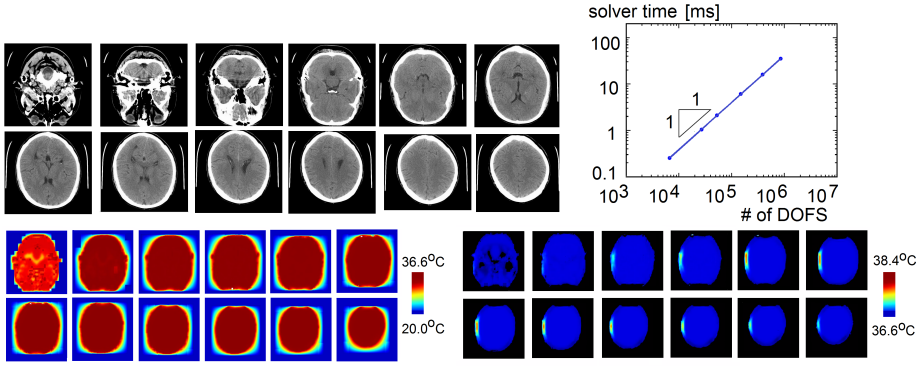

Fig. 2: (First row) Slices of the MRI scan of the human head. Linear cost of the solver. (Second row) The temperature of the head without the cell-phone antenna. The temperature of the head with the cell-phone antenna radiation.

Material	Air	Brain	Skull	Material	Air	Brain	Skull
$\rho\ [kg/m^3]$	1.16	1039	1645	$q_m\ [W/m^3]$	0	7100	590
$c\ [J/kg^oC]$	1006	3700	1300	$W_b c_b\ [W/m^{3o}C]$	0	40000	3300
$K\ [W/m^oC]$	0.02	0.57	0.4	$u_{a0}\ [^oC]$	20	36.6	36.6

Table 1: Material data used in simulation.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_18

3 Variational splitting for Pennes bioheat equation

In this example, we start from the MRI scan of the head of Maciej Paszyński, transformed into a 3D bitmap. We formulate the Pennes bio-heat equation

$$\rho c \frac{\partial u}{\partial t} - \nabla \cdot (K \nabla u) = W_b c_b (u_{a0} - u) + q_m + q_{SAR} \text{ in } \Omega \tag{18}$$

where K represents the thermal conductivity, q_m is the metabolism, $W_b c_b$ stands for the perfusion, and q_{SAR} is the heat source from the cellphone. We employ the parameters from Table 1, following [4]. They are selected according to the MRI scan data. We employ the alternating-direction solver with varying material data. We derive the weak form with the Crank-Nicolson scheme; we discretize with B-splines, and we approximate the left-hand side with Kronecker product form ignoring the terms which are of the higher-order with respect to τ

$$\sum_{i,j,k} \left(\int_{\Omega_x} B_i^x B_m^x - \frac{1}{\rho c} \tau \frac{K}{2} \partial_x B_i^x \partial_x B_m^x \right)$$
$$\left(\int_{\Omega_y} B_j^y B_n^y - \frac{1}{\rho c} \tau \frac{K}{2} \partial_y B_j^y \partial_y B_n^y \right) \left(\int_{\Omega_z} B_k^z B_l^z - \frac{1}{\rho c} \tau \frac{K}{2} \partial_z B_k^z \partial_z B_l^z \right) u_{ijk}^{t+1} =$$
$$\sum_{i,j,k} \int_\Omega B_i^x B_j^y B_k^z B_m^x B_n^y B_l^z u_{ijk}^t + \frac{\tau}{\rho c} \int_\Omega \frac{K}{2} \sum_{i,j,k} \nabla B_i^x B_j^y B_k^z u_{ijk}^t \cdot \nabla (B_m^x B_n^y B_l^z)$$
$$+ \frac{\tau}{\rho c} \int_\Omega (W_b [c_b u_{a0} - c_b \sum_{i,j,k} B_i^x B_j^y B_k^z u_{ijk}^t] + u_t + q_m + q_{SAR}) B_m^x B_n^y B_l^z \tag{19}$$

This allows applying Lemma 1 generalized for 3D. We 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 in the range of $(1, 240)$ and the skull, where the intensity is ≥ 240. As the initial condition, we select the temperature of 36.6 Celsius of the human head and 20.0 Celsjus of the air. We assume first no additional heat source and no presence of the cell phone antenna radiation, $q_{SAR} = 0$. The cross-sections of the 3D mesh after 10 minutes of the simulation are presented in Figure 2. As denoted by the red color, the maximum temperature is 36.6 Celsius. The blue color outside the head represents the air with a temperature of 20.0 Celsius. Next, we assume the q_{SAR} as estimated in Figure 6.13 [4]. The resulting heating of the human head after 10 minutes of the radiation is illustrated in Figure 2. As denoted by the red color, the maximum temperature is 38.4 Celsius degrees. The blue color of the human head represents the temperature of 36.6 Celsius. We illustrate the linear computational cost of the solver in Figure 2. We use computational grids of size $16 \times 16 \times 16$, which using cubic B-splines results in $N = (16+3)^3 = 6,859$, then $24 \times 24 \times 24$, which results in $N = (24+3)^3 = 19,683$, then $(32+3)^3 = 42,875$, then $(48+3)^3 = 132,651$, then $(64+3)^3 = 300,763$ and finally $(96+3)^3 = 970,299$. The computations are performed on the Linux workstation with a 2.4GHz processor with 64 GB of RAM. The computational burden related to distinguishing different material

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_18

data in comparison to homogeneous material is negligibly small. Our solver is a direct solver, and it provides the exact solution.

Conclusion. We can vary material data in implicit variational splitting solvers, preserving the linear cost. We can vary the material data $\epsilon_{k,l}$ at quadrature points. In the solver, we average them along lines parallel to the axis of the coordinate system for each support of the test functions. We test the method on the Pennes bioheat equation, and we verify the linear cost of the solver.

Acknowledgement. National Science Centre, Poland grant no. 2017/26/M/ST1/00281. Research project partly supported by program "Excellence initiative – research university" for the University of Science and Technology. The research presented in this paper was partially supported by the funds of Polish Ministry of Education and Science assigned to AGH University of Science and Technology. The European Union's Horizon 2020 Research and Innovation Program of the Marie Skłodowska-Curie grant agreement No. 777778 provided additional support.

References

1. Behnoudfar, P., Calo, V.M., Deng, Q., Minev, P.D.: A variationally separable splitting for the generalized-α method for parabolic equations. International Journal for Numerical Methods in Engineering **121**(5), 828–841 (2020)
2. Calo, V., Collier, N., Pardo, D., Paszyński, M.: Computational complexity and memory usage for multi-frontal direct solvers used in p finite element analysis. Procedia Computer Science **4**, 1854–1861 (2011)
3. Gao, L., Calo, V.M.: Fast isogeometric solvers for explicit dynamics. Computer Methods in Applied Mechanics and Engineering **274**, 19–41 (2014)
4. Kyunogjoo, K.: Finite element modeling of the radiation and induced heat transfer in the human body. Ph.D. dissertation, The University of Texas at Austin. (2013)
5. Łoś, M., Munoz-Matute, J., Muga, I., Paszyński, M.: Isogeometric residual minimization method (igrm) with direction splitting for non-stationary advection–diffusion problems. Computers & Mathematics with Applications **79**(2), 213–229 (2020)
6. Łoś, M., Paszyński, M., Kłusek, A., Dzwinel, W.: Application of fast isogeometric l2 projection solver for tumor growth simulations. Computer Methods in Applied Mechanics and Engineering **316**, 1257–1269 (2017)
7. Łoś, M., Woźniak, M., Paszyński, M., Dalcin, L., Calo, V.M.: Dynamics with matrices possessing Kronecker product structure. Procedia Computer Science **51**, 286–295 (2015)
8. Peaceman, D.W., Rachford, Jr, H.H.: The numerical solution of parabolic and elliptic differential equations. Journal of the Society for industrial and Applied Mathematics **3**(1), 28–41 (1955)
9. Samarskii, A.A.: The theory of difference schemes, vol. 240. CRC Press (2001)
10. Schaefer, R., Los, M., Sieniek, M., Demkowicz, L.F., 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 **11**, 163–174 (2015)
11. Sportisse, B.: An analysis of operator splitting techniques in the stiff case. Journal of Computational Physics **161**(1), 140 – 168 (2000)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_18