Robust PRESB Preconditioning of a 3-Dimensional Space-Time Finite Element Method for Parabolic Problems *†

Ladislav Foltyn‡, Dalibor Lukáš§, and Marco Zank¶

March 31, 2023

Abstract

We present a recently developed preconditioning of square block matrices (PRESB) to be used within a parallel method of solution to linear systems of equations arising from tensor-product discretizations of initial boundary-value problems for parabolic-elliptic partial differential equations. We consider weak formulations in Bochner–Sobolev spaces and tensor-product finite element approximations for the heat and eddy current equations. The fast diagonalization method is employed to decouple the arising linear system of equations into auxiliary spatial complex-valued linear systems that are concurrent-in-time. It is proved that the real part of the system matrix is positive definite, which allows us to accelerate the flexible generalized minimal residual method (FGMRES) by the PRESB method. The action of the PRESB method includes two solutions of positive definite systems. The spectrum of the preconditioned system lies between 1/2 and 1. Finally, we combine the PRESB-FGMRES method with multigrid-CG iterations and document numerical efficiency and robustness for spatial discretizations up to 12 millions degrees of freedom.

Keywords: parabolic-elliptic problems, parallel space-time methods, finite element methods, fast diagonalization method, PRESB preconditioning, multigrid

1 Introduction

Nowadays, the computational power of hardware grows by means of massive parallelism rather than increasing the clock rate. This trend is reflected in the development of efficient mathematical methods and software. Physical phenomena governed by partial differential equations such as heat conduction or mechanics can be simulated by a well-established class of parallel methods, the domain decomposition methods [1, 2, 3, 4]. The methods iteratively solve local quasi-static subproblems in parallel and correct the solution by a global coarse problem. The latter gives rise to communication and limits the total system

*The work is dedicated to the memory of Owe Axelsson.

†The work was partially supported by project SGS No. SP2023/67, VŠB-Technical University of Ostrava, Czech Republic. This work was also supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140).

‡VŠB-Technical University of Ostrava, Department of Applied Mathematics, Faculty of Electrical Engineering and Computer Science & IT4Innovations, Czech Republic, e-mail: ladislav.foltyn@vsb.cz

§VŠB-Technical University of Ostrava, Department of Applied Mathematics, Faculty of Electrical Engineering and Computer Science, Czech Republic, e-mail: dalibor.lukas@vsb.cz, Corresponding author.

¶University of Vienna, Faculty of Mathematics, Vienna, Austria, e-mail: marco.zank@univie.ac.at

size such that the just arising exascale supercomputers cannot be fully exploited yet. Therefore, another parallelization in the temporal variable has been a subject of active research, see [5] for an overview.

For a long time, it remained unclear how the sequential matter of time evolution can be simulated in parallel. A breakthrough paper [6] gave birth to the parareal method. It can be viewed as a predictor-corrector iterative method. Concurrent predictions on local time intervals are corrected by a global coarse problem. Although the convergence rate is often super-linear, the actual speedup is inversely proportional to the number of iterations. This is combined with the Schwarz domain decomposition in the method of waveform relaxation [7], where concurrent time-dependent local problems are solved on spatial subdomains throughout the whole time interval. In other words, the method first decomposes the spatial part, and in a second step, the temporal part. The other way around is presented in [8], i.e., first, the parareal method is applied and then a non-overlapping spatial domain decomposition is employed within each time slice. Neither of the approaches can achieve optimal parallel efficiency, i.e., the computational time is not inversely proportional to the number of computing cores.

A more flexible and promising approach is to treat time as another spatial variable. Outstanding results on space-time multigrid methods applied to parabolic problems, including 2d Navier–Stokes equations, were published in [9, 10]. The methods enjoy parallel scalability up to 10^8 space-time degrees of freedom and 10^3 cores. The theory of space-time weak formulations dates back to [11], where the problem is posed in space-time Sobolev spaces. This is nowadays followed by constructions of solvers [12, 13]. In this paper, we follow [14], where the authors discretize a formulation in the fractional-order Sobolev spaces by tensor-product finite elements and employ the fast diagonalization method [15].

The purpose of the present paper is twofold. Firstly, we show that the techniques of [14] also apply to a simpler Bochner–Sobolev space setting [16]. Secondly, since the arising saddle-point problems are related to complex-valued systems, they enjoy a specific block structure, for which a tailored preconditioner of square block matrices (PRESB) was developed [17]. This preconditioner is at least as efficient as a more general Uzawa-type methods [19] used in [12].

The rest of the paper is as follows: In Section 2, the weak formulation and the tensor-product space-time discretization of the initial-boundary value problem for parabolic-elliptic problems, including the 3d heat equation and the 3d eddy current equations are recalled. In Section 3, we describe the solver comprising the fast diagonalization method and PRESB preconditioner. Here we also prove that the real part of the underlying temporal eigenvalue problem is positive. In Section 4, we give numerical experiments, where we employ a geometric multigrid for the solution to the inner-most elliptic systems. We conclude in Section 5.

2 Space-time parabolic problems

Consider the initial-boundary value problem for the parabolic-elliptic partial differential equation

$$\left.\begin{aligned} c\,\frac{\partial u}{\partial t}(x,t) + L_x^*\left[k(x)\,L_x u(x,t)\right] + \varepsilon\,u(x,t) &= f(x,t), && (x,t)\in Q,\\ \gamma u(x,t) &= 0, && (x,t)\in \Gamma\times[0,T],\\ u(x,0) &= u_0(x), && x\in\Omega, \end{aligned}\right\} \tag{1}$$

where $Q := \Omega\times(0,T)$, $T>0$, and $\Omega\subset\mathbb{R}^3$ is a bounded domain with Lipschitz boundary $\Gamma := \partial\Omega$. The symbol $x := (x_1,x_2,x_3)$ denotes the spatial coordinates while t denotes time. The spatial elliptic operator is composed of a first-order operator L_x, the dual operator L_x^*, and the regularization term $\varepsilon\,u$. The operator γ denotes the corresponding trace. We shall consider two particular problems:

(i) For the heat equation, we set $L_x := \nabla_x$, $L_x^* := -\mathrm{div}_x$, $\varepsilon := 0$, and γ is the scalar trace operator defined as $\gamma w := w|_\Gamma$ for a sufficiently smooth scalar function w of x.

(ii) For the eddy current equations, we choose $L_x = L_x^* := \mathrm{curl}_x$, $0 < \varepsilon \ll 1$, and γ is the tangential trace operator, which is given as $\gamma w := w|_\Gamma \times n$ for a sufficiently smooth vector-valued function w of x, where n denotes the outward unit normal to Ω.

Further, we are given the material constant $c > 0$ and a positive scalar function k. Finally, the source term f and the initial datum u_0 are given.

2.1 Weak formulation

As it was stated in [16], the variational formulation corresponding to (1) is to find $u \in U$ satisfying $u(x,0) = u_0(x)$ for $x \in \Omega$ and

$$\underbrace{c\left\langle \frac{\partial u}{\partial t}, v\right\rangle_Q + \int_Q \left[k(x)\, L_x u(x,t) \cdot L_x v(x,t) + \varepsilon\, u(x,t)\cdot v(x,t)\right] \mathrm{d}x\,\mathrm{d}t}_{=:a(u,v)} = \langle f, v\rangle_Q \qquad (2)$$

for all $v \in V$. Here, we assume $k \in L^\infty(\Omega)$, $k(x) \geq k_0 > 0$ for almost all $x \in \Omega$, $u_0 \in H(\Omega)$, $f \in V^*$, where

$$H(\Omega) := \left\{ w \in \left[L^2(\Omega)\right]^m \,:\, L_x w \in \left[L^2(\Omega)\right]^3 \text{ and } \gamma w(x) = 0 \text{ for } x \in \Gamma \right\},$$

where $m := 1$ for the scalar case (heat equation) and $m := 3$ for the vectorial one (eddy current equations),

$$U := L^2(0,T;H(\Omega)) \cap H^1(0,T;H(\Omega)^*)\,, \quad V := L^2(0,T;H(\Omega))\,, \quad V^* := L^2(0,T;H(\Omega)^*)\,.$$

Namely, $H(\Omega) := H_0^1(\Omega)$ or $H(\Omega) := H_0(\mathrm{curl};\Omega)$ in case of the heat or eddy current equations and $H(\Omega)^*$ denotes the dual space to $H(\Omega)$. Further, $\langle\cdot,\cdot\rangle_Q$ is the duality pairing on V and V^* as extension of the inner product in $[L^2(Q)]^m$, where $m = 1$ for the heat equation or $m = 3$ for the eddy current equations.

As we treat the initial condition $u_0 \in H(\Omega)$ as a Dirichlet condition, the solution u can be split into a homogeneous and particular part, i.e., $u(x,t) = \widetilde{u}(x,t) + u_0(x)$ for $(x,t) \in Q$. Then, the variational formulation (2) is to find $\widetilde{u} \in \widetilde{U} := L^2(0,T;H(\Omega)) \cap H^1_{0,}(0,T;H(\Omega)^*)$ such that

$$a(\widetilde{u}, v) = \underbrace{\langle f, v\rangle_Q - a(u_0, v)}_{=:b(v)} \qquad (3)$$

for all $v \in V$, where

$$H^1_{0,}(0,T;H(\Omega)^*) := \left\{ v \in H^1(0,T;H(\Omega)^*) :\, v(\cdot,0) = 0 \text{ in } H(\Omega)^* \right\}.$$

In case of the heat equation, i.e., $H(\Omega) = H_0^1(\Omega)$, $L_x = \nabla_x$, $\varepsilon = 0$, there exists a unique solution $\widetilde{u} \in \widetilde{U}$ of the variational formulation (3) satisfying the stability estimate

$$\|\widetilde{u}\|_U \leq 2\sqrt{2}\left[\|f\|_{V^*} + \sqrt{2T\|k\|_{L^\infty(\Omega)}}\|L_x u_0\|_{[L^2(\Omega)]^3}\right], \qquad (4)$$

where V is equipped with the Hilbertian norm

$$\|v\|_V := \left(\int_Q k(x)\, L_x v(x,t)\cdot L_x v(x,t)\,\mathrm{d}x\,\mathrm{d}t\right)^{1/2},$$

its dual space $V^* = L^2(0,T;H(\Omega)^*)$ is equipped with

$$\|f\|_{V^*} := \sup_{0\neq v\in V} \frac{c\langle f, v\rangle_Q}{\|v\|_V},$$

and U is equipped with

$$\|w\|_U := \left(\left\| \frac{\partial w}{\partial t} \right\|_{V^*}^2 + \|w\|_V^2 \right)^{1/2}.$$

The stability estimate (4) follows from [16, Corollary 2.3] when choosing there $\overline{u}_0(x,t) = u_0(x)$ for $(x,t) \in Q$. Note that similar results hold true for the more general case $u_0 \in L^2(\Omega)$, see [13, Theorem 3.2.4].

2.2 Tensor-product FEM discretization

Let Ω be a polyhedral Lipschitz domain. We consider an admissible and shape-regular finite element discretization of Ω into M_x tetrahedral elements ω_i, i.e.,

$$\overline{\Omega} = \bigcup_{i=1}^{M_x} \overline{\omega_i}, \quad \omega_i \cap \omega_j = \emptyset \text{ if } i \neq j,$$

and two connected elements share either a face, an edge, or a vertex. We shall denote the element mesh sizes by $h_{x,i} := \operatorname{diam} \omega_{x,i}$ and the global mesh size by $h_x := \max_i h_{x,i}$. Furthermore, we divide the time interval $(0,T)$ into M_t non-overlapping subintervals, i.e.,

$$0 =: t_0 < t_1 < t_2 < \cdots < t_{M_t} := T$$

with the local time steps denoted by $h_{t,j} := t_j - t_{j-1}$ and the global time step $h_t := \max_j h_{t,j}$. Hence, the tensor-product decomposition of Q into $M := M_x M_t$ elements reads

$$\overline{Q} = \bigcup_{i=1}^{M_x} \bigcup_{j=1}^{M_t} \overline{\omega_i} \times [t_{j-1}, t_j].$$

Next, we introduce the lowest-order conforming finite element subspaces of $H(\Omega)$. These are the Lagrange finite element spaces of continuous element-wise linear functions with nodal degrees of freedom in case of the heat equation and the Nédélec-I [21] spaces of tangential-continuous element-wise linear vectorial functions with one degree of freedom per edge in case of the eddy current equations. We shall denote both spaces by

$$S^1_{h_x}(\Omega) := \operatorname{span} \{\psi_i\}_{i=1}^{N_x} \subset H(\Omega),$$

where N_x is the total number of spatial degrees of freedom, i.e., the number of interior (non-Dirichlet) nodes or the interior edges for the respective cases.

Finally, we introduce the lowest-order conforming finite element subspace of $H^1_{0,}(0,T) := \{v \in H^1(0,T) : \, v(0) = 0\}$, i.e., the Lagrange nodal finite element space for both cases,

$$S^1_{h_t}(0,T) := \operatorname{span} \{\varphi_j\}_{j=1}^{N_t} \subset H^1_{0,}(0,T),$$

where $N_t := M_t$ is the number of non-initial time steps $0 \neq t_j$, $j = 1, \ldots, M_t$. We arrive at the tensor-product finite element space

$$\widetilde{U}_h := S^1_h(Q) := S^1_{h_x}(\Omega) \otimes S^1_{h_t}(0,T) \subset \widetilde{U}$$

of dimension $N := N_t N_x$. To preserve nice properties of a temporal matrix-pencil discussed later in Section 3.1, we shall take this discrete ansatz space as a test space too, i.e., $V_h := S^1_h(Q) \subset V$.

The discrete variational formulation to (3) is to find $\widetilde{u}_h \in \widetilde{U}_h$ such that

$$a(\widetilde{u}_h, v_h) = b(v_h) \tag{5}$$

for all $v_h \in V_h$. Like in [14], the discrete space-time variational formulations (5) lead to the following systems of linear equations

$$(\mathrm{A}_{h_t} \otimes \mathrm{M}_{h_x} + \mathrm{M}_{h_t} \otimes \mathrm{A}_{h_x})\, \widetilde{\mathbf{u}} = \mathbf{b}, \tag{6}$$

where for $k, l = 1, 2, \ldots, N_t$,

$$(\mathrm{A}_{h_t})_{k,l} := \int_0^T \varphi_l'(t)\, \varphi_k(t)\, \mathrm{d}t, \quad (\mathrm{M}_{h_t})_{k,l} := \int_0^T \varphi_l(t)\, \varphi_k(t)\, \mathrm{d}t,$$

for $i, j = 1, 2, \ldots, N_x$,

$$(\mathrm{A}_{h_x})_{i,j} := \int_\Omega \left[k(x)\, L_x\psi_j(x) \cdot L_x\psi_i(x) + \varepsilon\, \psi_j(x) \cdot \psi_i(x)\right]\, \mathrm{d}x, \quad (\mathrm{M}_{h_x})_{i,j} := c \int_\Omega \psi_j(x) \cdot \psi_i(x)\, \mathrm{d}x,$$

and $\mathbf{b} := (\mathbf{b}_1, \ldots, \mathbf{b}_{N_t}) \in \mathbb{R}^N$, where for $i = 1, 2, \ldots, N_x$ and $k = 1, 2, \ldots, N_t$,

$$(\mathbf{b}_k)_i := \langle f, \varphi_k\, \psi_i \rangle_Q - \int_Q \left[k(x)\, L_x u_0(x) \cdot L_x\psi_i(x) + \varepsilon\, u_0(x) \cdot \psi_i(x)\right] \varphi_k(t)\, \mathrm{d}x\, \mathrm{d}t.$$

In the case of the heat equation, by [16, Theorem 3.2], there exists a unique and stable solution $\widetilde{u}_h \in \widetilde{U}_h$ of the variational formulation (5), which converges to the solution of (3).

3 PRESB-based space-time solver

To solve the linear system (6), we exploit the tensor-product structure and employ the fast diagonalization method [15]. The idea is to solve N_t independent auxiliary spatial problems. In fact, it is similar to the discrete Fourier analysis in time, cf. [17], in which case the solution is assumed to be time-periodic. In this paper, we do not have such restrictions. Here, the additional price of the fast diagonalization method is that we shall orthogonalize (by means of eigenvalue decomposition) the temporal basis functions, which is negligible since $N_t \ll N_x$. Luckily, the arising spatial complex-valued linear systems can be efficiently solved by the dedicated preconditioner method PRESB [17].

3.1 Eigenvalues of the temporal matrix pencil

The matrices M_{h_x}, M_{h_t}, and A_{h_x} are Gram matrices. Hence, those are symmetric and positive definite, due to the regularization parameter $\varepsilon > 0$ for the eddy current equations. The matrix A_{h_t} is nonsymmetric and positive-semidefinite, as

$$\mathbf{v}^\top\, \mathrm{A}_{h_t}\, \mathbf{v} = \int_0^T v_{h_t}'(t)\, v_{h_t}(t)\, \mathrm{d}t = -\int_0^T v_{h_t}(t)\, v_{h_t}'(t)\, \mathrm{d}t + [v_{h_t}(T)]^2 - \underbrace{[v_{h_t}(0)]^2}_{=0}$$

and, hence

$$\mathbf{v}^\top\, \mathrm{A}_{h_t}\, \mathbf{v} = \frac{1}{2}[v_{h_t}(T)]^2 \geq 0 \tag{7}$$

hold true, where the vector $\mathbf{v} \in \mathbb{R}^{N_t}$ defines the function $v_{h_t}(t) = \sum_{j=1}^{N_t} v_j \varphi_j(t)$. Obviously, for functions with $v_{h_t}(T) = 0$, the quadratic form vanishes, $\mathbf{v}^\top\, \mathrm{A}_{h_t}\, \mathbf{v} = 0$.

Next, we investigate the temporal matrix pencil, namely, the generalized eigenvalue problem

$$\mathrm{A}_{h_t}\, \mathbf{z} = \xi\, \mathrm{M}_{h_t}\, \mathbf{z}, \quad \mathbf{z} \neq \mathbf{0}, \tag{8}$$

with the generalized eigenvector $\mathbf{z} = (z_1, \ldots, z_{N_t})^\top \in \mathbb{C}^{N_t}$ and the generalized eigenvalue $\xi = \mu + \imath\nu \in \mathbb{C}$ with $\mu, \nu \in \mathbb{R}$.

Lemma 1. *The generalized eigenvalues $\xi = \mu + \imath\nu$ of the generalized eigenvalue problem* (8) *satisfy*

$$\Re\xi = \mu > 0.$$

Proof. Let $\xi = \mu + \imath\nu$, where $\mu, \nu \in \mathbb{R}$, be a generalized eigenvalue of (8), and $\mathbf{z} = \mathbf{x} + \imath\mathbf{y}$ be a related generalized eigenvector, where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{N_t}$.

First, [12, Lemma 3.2] and estimate (7) yield $\Re\xi = \mu \geq 0$.

The proof of $\Re\xi = \mu > 0$ is based on the same idea as [12, Remark 3.5]. Note that $z_{h_t}(T) = 0$ if and only if $z_{N_t} = 0$ with $z_{h_t} = \sum_{k=1}^{N_t} z_k \varphi_k$. With this, we distinguish two cases.

Case $z_{N_t} \neq 0$: Then, it follows from [12, equation (3.5)] and estimate (7) that

$$\mu = \frac{a}{b} > 0$$

with $a := \mathbf{x}^\top \mathrm{A}_{h_t} \mathbf{x} + \mathbf{y}^\top \mathrm{A}_{h_t} \mathbf{y} > 0$ and $b := \mathbf{x}^\top \mathrm{M}_{h_t} \mathbf{x} + \mathbf{y}^\top \mathrm{M}_{h_t} \mathbf{y} > 0$.

Case $z_{N_t} = 0$: We shall prove that $\mathbf{z} = \mathbf{0}$ when $\mu = 0$, i.e., $\mathbf{z}$ can not be an eigenvector. Let us assume $\mu = 0$. The matrices take the form:

$$\mathrm{M}_{h_t} = \frac{1}{6} \begin{pmatrix} 2h_{t,1} + 2h_{t,2} & h_{t,2} & & & & \\ h_{t,2} & 2h_{t,2} + 2h_{t,3} & h_{t,3} & & & \\ & \ddots & \ddots & \ddots & & \\ & & h_{t,N_t-1} & 2h_{t,N_t-1} + 2h_{t,N_t} & h_{t,N_t} & \\ & & & h_{t,N_t} & 2h_{t,N_t} \end{pmatrix}$$

and

$$\mathrm{A}_{h_t} = \frac{1}{2} \begin{pmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -1 & 1 \end{pmatrix}.$$

Thus, the last line of (8) reads

$$-z_{N_t-1} + z_{N_t} = \frac{\imath\nu}{3}(h_{t,N_t}\, z_{N_t-1} + 2h_{t,N_t}\, z_{N_t}),$$

which gives $z_{N_t-1} = 0$. Next, the last but one line of (8) reads

$$-z_{N_t-2} + z_{N_t} = \frac{\imath\nu}{3}\left[h_{t,N_t-1}\, z_{N_t-2} + 2\,(h_{t,N_t-1} + h_{t,N_t})\, z_{N_t-1} + h_{t,N_t}\, z_{N_t}\right],$$

which yields $z_{N_t-2} = 0$. By induction we have $z_1 = \cdots = z_{N_t} = 0$, i.e., $\mathbf{z} = \mathbf{0}$, which is in contradiction to (8). Hence, $\mu > 0$. □

The following is a trivial consequence.

Corollary 1. *The generalized eigenvalue problem* (8) *is equivalent to the problem*

$$\mathrm{M}_{h_t}\, \mathbf{z} = \lambda\, \mathrm{A}_{h_t}\, \mathbf{z}, \quad \mathbf{z} \neq \mathbf{0}. \tag{9}$$

Namely, the generalized eigenvalues $\lambda = \alpha + \imath\beta$, $\alpha, \beta \in \mathbb{R}$, of (9) *satisfy $\lambda = 1/\xi$, i.e.,*

$$\alpha = \frac{\mu}{\mu^2 + \nu^2}, \quad \beta = -\frac{\nu}{\mu^2 + \nu^2},$$

where $\xi = \mu + \imath\nu$ and $\mathbf{z}$ are those generalized eigenvalues and eigenvectors of (8)*. Hence,*

$$\Re\lambda = \alpha > 0.$$

3.2 Fast diagonalization method

The fast diagonalization method applied to the tensor-product system (6) was proposed in [15], see also [14] and [12, 18] for investigations in isogeometric analysis. The idea is to construct N_t independent spatial subproblems which can be solved in parallel. We rewrite (9) as follows:

$$\mathrm{A}_{h_t}^{-1}\,\mathrm{M}_{h_t} = \mathrm{Z}_{h_t}\,\Lambda_{h_t}\,\mathrm{Z}_{h_t}^{-1} \tag{10}$$

with $\mathrm{Z}_{h_t} := [\mathbf{z}_1, \ldots, \mathbf{z}_{N_t}]$ and $\Lambda_{h_t} := \mathrm{diag}\,(\lambda_1, \ldots, \lambda_{N_t})$, where λ_k and $\mathbf{z}_k$ are the eigenvalues and eigenvectors of (9). We shall further denote

$$\mathrm{Y}_{h_t} := \mathrm{Z}_{h_t}^{-1}\,\mathrm{A}_{h_t}^{-1}.$$

The solution of (6) admits the formula

$$\widetilde{\mathbf{u}} = (\mathrm{Z}_{h_t} \otimes \mathrm{I}_{N_x})\,(\mathrm{I}_{N_t} \otimes \mathrm{M}_{h_x} + \Lambda_{h_t} \otimes \mathrm{A}_{h_x})^{-1}\,(\mathrm{Y}_{h_t} \otimes \mathrm{I}_{N_x})\,\mathbf{b},$$

where $\mathrm{I}_{N_x} \in \mathbb{R}^{N_x \times N_x}$ and $\mathrm{I}_{N_t} \in \mathbb{R}^{N_t \times N_t}$ denote the identity matrices.

We summarize the fast diagonalization method as a direct solution method to (6) as follows:

1. Solve the eigenvalue decomposition (10).
2. Transform the right-hand side $\mathbf{b} \in \mathbb{R}^{N_x\,N_t}$ to $\mathbf{g} \in \mathbb{C}^{N_x\,N_t}$,
$$\mathbf{g} := (\mathbf{g}_1, \ldots, \mathbf{g}_{N_t})^\top := (\mathrm{Y}_{h_t} \otimes \mathrm{I}_{N_x})\,\mathbf{b}.$$
3. In parallel, for $k = 1, \ldots, N_t$, solve the complex-valued systems
$$(\mathrm{M}_{h_x} + \lambda_k \mathrm{A}_{h_x})\,\mathbf{w}_k = \mathbf{g}_k. \tag{11}$$
4. Assemble the solution
$$\widetilde{\mathbf{u}} := (\mathrm{Z}_{h_t} \otimes \mathrm{I}_{N_x})\,\mathbf{w},$$
where $\mathbf{w} := (\mathbf{w}_1, \ldots, \mathbf{w}_{N_t})^\top$.

3.3 PRESB method

We rewrite the complex-valued system (11),

$$\Big(\mathrm{M}_{h_x} + \overbrace{(\alpha_k + \imath\beta_k)}^{=\lambda_k}\,\mathrm{A}_{h_x}\Big)\,\overbrace{(\mathbf{u}_k + \imath\mathbf{v}_k)}^{=\mathbf{w}_k} = \overbrace{(\mathbf{b}_k + \imath\mathbf{c}_k)}^{=\mathbf{g}_k},$$

as the two-by-two real system

$$\begin{pmatrix} \mathrm{M}_{h_x} + \alpha_k \mathrm{A}_{h_x} & -\beta_k \mathrm{A}_{h_x} \\ \beta_k \mathrm{A}_{h_x} & \mathrm{M}_{h_x} + \alpha_k \mathrm{A}_{h_x} \end{pmatrix} \begin{pmatrix} \mathbf{u}_k \\ \mathbf{v}_k \end{pmatrix} = \begin{pmatrix} \mathbf{b}_k \\ \mathbf{c}_k \end{pmatrix}, \tag{12}$$

which is solved iteratively by the flexible generalized minimal residual method (FGMRES) [20] accelerated by the PRESB method [17]. The PRESB method is described in the following.

We shall consider a complex-valued system rewritten as the two-by-two real system

$$\underbrace{\begin{pmatrix} \mathcal{A} & \mathcal{B} \\ -\mathcal{B} & \mathcal{A} \end{pmatrix}}_{=:\mathcal{M}} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix}, \tag{13}$$

where $\mathcal{A}$ and $\mathcal{B}$ are square matrices and $\mathcal{A}+\mathcal{B}$ is nonsingular. The PRESB preconditioner reads as follows:

$$\mathcal{C} := \begin{pmatrix} \mathcal{A} + 2\mathcal{B} & \mathcal{B} \\ -\mathcal{B} & \mathcal{A} \end{pmatrix}. \tag{14}$$

As described in [17], the action of the preconditioner

$$\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} := \mathcal{C}^{-1}\begin{pmatrix}\mathbf{p}\\ \mathbf{q}\end{pmatrix}$$

proceeds as follows:

1. Solve $(\mathcal{A}+\mathcal{B})\,\mathbf{r} = \mathbf{p}+\mathbf{q}$.
2. Compute $\widetilde{\mathbf{p}} := \mathbf{p} - \mathcal{B}\,\mathbf{r}$.
3. Solve $(\mathcal{A}+\mathcal{B})\;\mathbf{x} = \widetilde{\mathbf{p}}$.
4. Compute $\mathbf{y} := \mathbf{r} - \mathbf{x}$.

Lemma 2. *Let $\mathcal{A}$ and $\mathcal{B}$ be symmetric and positive semidefinite, $\mathcal{A}+2\mathcal{B}$ be positive definite, and $\mathcal{A}+\mathcal{B}$ be nonsingular. Then the eigenvalues σ of $\mathcal{C}^{-1}\,\mathcal{M}$ are real and satisfy*

$$\frac{1}{2} \le \sigma \le 1.$$

Proof. [17, Proposition 3.1] □

Next, we shall apply the PRESB preconditioner to (12). We shall additionally rely on the positive definiteness of $\mathcal{A}+\mathcal{B}$ in (13). To that aim, we distinguish two cases.

In case $\beta_k \le 0$, the use of PRESB method is straightforward, i.e.,

$$\mathcal{A} := \mathrm{M}_{h_x} + \alpha_k \mathrm{A}_{h_x}, \quad \mathcal{B} := -\beta_k \mathrm{A}_{h_x}, \quad \mathbf{u} := \mathbf{u}_k, \quad \mathbf{v} := \mathbf{v}_k, \quad \mathbf{b} := \mathbf{b}_k, \quad \mathbf{c} := \mathbf{c}_k.$$

In this case, in steps 1 and 3 of PRESB method, we shall solve the symmetric and positive definite systems $\mathcal{A}+\mathcal{B} = \mathrm{M}_{h_x} + (\alpha_k - \beta_k)\mathrm{A}_{h_x}$.

In case $\beta_k > 0$, we can change the sign of the second column and the second row in (12), hence,

$$\mathcal{A} := \mathrm{M}_{h_x} + \alpha_k \mathrm{A}_{h_x}, \quad \mathcal{B} := \beta_k \mathrm{A}_{h_x}, \quad \mathbf{u} := \mathbf{u}_k, \quad \mathbf{v} := -\mathbf{v}_k, \quad \mathbf{b} := \mathbf{b}_k, \quad \mathbf{c} := -\mathbf{c}_k.$$

Then, in steps 1 and 3 of PRESB method, we shall solve again the symmetric and positive definite systems $\mathcal{A}+\mathcal{B} = \mathrm{M}_{h_x} + (\alpha_k + \beta_k)\mathrm{A}_{h_x}$.

3.4 Space-time solver

In this subsection, we summarize the space-time solver of the linear system (6) using the fast diagonalization and the PRESB methods.

1: Compute the eigenvalue decomposition $\mathrm{A}_{h_t}^{-1}\,\mathrm{M}_{h_t} = \mathrm{Z}_{h_t}\,\Lambda_{h_t}\,\mathrm{Z}_{h_t}^{-1}$ in (10) with $\Lambda_{h_t} = \mathrm{diag}\,(\lambda_1,\ldots,\lambda_{N_t})$.

2: Solve $\mathbf{g} := (\mathbf{g}_1,\ldots,\mathbf{g}_{N_t})^\top := \left(\mathrm{Z}_{h_t}^{-1}\,\mathrm{A}_{h_t}^{-1}\otimes \mathrm{I}_{N_x}\right)\mathbf{b}$, where $\mathbf{b}$ is the right side of (6).

3: Compute $\mathbf{w} := (\mathbf{w}_1,\ldots,\mathbf{w}_{N_t})^\top$ independently by

for $k = 1, 2, \ldots, N_t$ **do**

Set $\alpha_k = \Re\lambda_k$ and $\beta_k = \Im\lambda_k$.

if $\beta_k \le 0$ **then**

Solve

$$\begin{pmatrix}\mathrm{M}_{h_x}+\alpha_k\mathrm{A}_{h_x} & -\beta_k\mathrm{A}_{h_x}\\ \beta_k\mathrm{A}_{h_x} & \mathrm{M}_{h_x}+\alpha_k\mathrm{A}_{h_x}\end{pmatrix}\begin{pmatrix}\mathbf{u}_k\\ \mathbf{v}_k\end{pmatrix} = \begin{pmatrix}\Re\mathbf{g}_k\\ \Im\mathbf{g}_k\end{pmatrix}$$

using the PRESB method as described in Section 3.3 with the PRESB preconditioner

$$\mathcal{C} := \begin{pmatrix}\mathrm{M}_{h_x}+(\alpha_k-2\beta_k)\mathrm{A}_{h_x} & -\beta_k\mathrm{A}_{h_x}\\ \beta_k\mathrm{A}_{h_x} & \mathrm{M}_{h_x}+\alpha_k\mathrm{A}_{h_x}\end{pmatrix}.$$

else

Solve

$$\begin{pmatrix} \mathrm{M}_{h_x} + \alpha_k \mathrm{A}_{h_x} & \beta_k \mathrm{A}_{h_x} \\ -\beta_k \mathrm{A}_{h_x} & \mathrm{M}_{h_x} + \alpha_k \mathrm{A}_{h_x} \end{pmatrix} \begin{pmatrix} \mathbf{u}_k \\ -\mathbf{v}_k \end{pmatrix} = \begin{pmatrix} \Re \mathbf{g}_k \\ -\Im \mathbf{g}_k \end{pmatrix}$$

using the PRESB method as discribed in Section 3.3 with the PRESB preconditioner

$$\mathcal{C} := \begin{pmatrix} \mathrm{M}_{h_x} + (\alpha_k + 2\beta_k) \mathrm{A}_{h_x} & \beta_k \mathrm{A}_{h_x} \\ -\beta_k \mathrm{A}_{h_x} & \mathrm{M}_{h_x} + \alpha_k \mathrm{A}_{h_x} \end{pmatrix}.$$

end if

Set $\mathbf{w}_k := \mathbf{u}_k + \imath \mathbf{v}_k$.

end for

4: Compute the solution $\widetilde{\mathbf{u}} := (\mathrm{Z}_{h_t} \otimes \mathrm{I}_{N_x})\, \mathbf{w}$.

4 Numerical experiments

First of all, we specify the parameters of the problem (1). The computational domain is the unit cube $\Omega := (0,1)^3$, the simulation time $T := 1$. The temporal coefficient is set to $c := 1$ as well as the source term $f(x,t) := 1$ in case of the heat equation and $f(x,t) := (1,1,1)^\top$ in case of the eddy current equations. We consider the decomposition of Ω into two subdomains $\Omega_1 := (0,1/2) \times (0,1) \times (0,1)$ and $\Omega_2 := (1/2,1) \times (0,1) \times (0,1)$, over which we prescribe jumping coefficients

$$k(x) := \begin{cases} k_1, & x \in \Omega_1, \\ k_2, & x \in \Omega_2, \end{cases}$$

where $k_1 := 1$ and $k_2 > 0$ is a parameter varying from 10^{-6} to 10^6 to document robustness of the solver. The regularization parameter is $\varepsilon := 0$ in case of the heat equation, where no regularization is needed, and $\varepsilon := 10^{-6}$ in case of the eddy current equations to establish the well-posedness. We consider the homogeneous initial conditions, i.e., $u_0(x) := 0$ or $u_0(x) := (0,0,0)^\top$ in case of the heat or eddy current equations, respectively.

As summarized in Section 3.4, we solve N_t independent auxiliary spatial systems (12). Next, we describe the solver setup. It relies on inner-outer iterations. Since the systems can be seen as symmetric indefinite, we could opt for the minimal residual (MinRes) method. However, as we employ an iterative method within the preconditioner, the outer iterations are resolved by FGMRES [20]. To accelerate their convergence, we use the PRESB preconditioner (14). The relative tolerance of this PRESB-FGMRES method is 10^{-8}. Within the action of PRESB, two symmetric positive definite systems are solved. Here, we introduce inner iterations by the conjugate gradient method (CG) preconditioned with a geometric multigrid. We set the relative tolerance for these inner multigrid-CG iterations to 10^{-2}. The geometric multigrid consists of a V-cycle through up to 4 levels of uniform refinements. As a smoother, we employ the additive Jacobi method with a relaxation factor 0.01 in some cases of the heat equation or the multiplicative nodal-patch smoother by Arnold, Falk, and Winther [22] in the other cases of the heat equation as well as in all the cases of the eddy current equations. At the coarsest level, the system is solved directly by the Cholesky decomposition. We rely on an in-house MPI-C++ code with two external libraries: EIGEN [23] used to solve (10) within the fast diagonalization method and the column re-ordering algorithm COLAMD [24] used within our sparse direct solver. The coarsest level was discretized in Netgen [25]. The numerical experiments for up to millions of DOFs were performed on a laptop equipped with the 12-core processor Intel Core i7-8750H, 2.2 GHz and 32 GB memory, while the finest level experiments for cca twelve million of DOFs were performed at Karolina cluster at the IT4Innovations, VSB-Technical University of Ostrava, Czech Republic. A node of Karolina is equipped with two 64-cores AMD 7H12 processors, 2.6 GHz, and 256 GB memory.

4.1 Space-time simulations

We present complete space-time simulations with an embarrassingly parallel implementation of the independent auxiliary spatial systems. Here, we solve the heat equation without jumping coefficients, i.e., $k_2 := 1$. We decompose the time interval $(0, 1)$ into $M_t = N_t \in \{32, 64, 128, 256\}$ temporal elements and employ the method at 3 levels of uniformly refined spatial discretizations. The range of numbers of outer and inner iterations is displayed in Tab. 1, while the lowest numbers were achieved for the highest real parts α_k of the eigenvalues and the highest numbers were observed for the lowest α_k. Nonetheless, the iterations are stable regarding both the temporal and spatial sizes of the space-time discretizations. Here, we employed the multiplicative nodal-patch smoother.

Table 1: Ranges of the numbers of the outer PRESB-FGMRES iterations and, in brackets, the corresponding ranges of the total numbers of the inner multigrid-CG iterations for the solutions of the heat equation discretized into $N_x\, N_t$ space-time DOFs.

multigrid level	N_x	$N_t := 32$	$N_t := 64$	$N_t := 128$	$N_t := 256$
0	2 395	7-12(14-24)	7-12 (14-24)	7-12 (14-24)	7-12(14-24)
1	16 433	8-13(17-27)	8-13(17-27)	8-13(17-27)	8-13(17-27)
2	121 265	8-13(19-27)	8-13(19-27)	8-13(19-27)	8-13(19-27)

4.2 Robustness of the PRESB preconditioner

In this subsection, we present robustness of the preconditioner with respect to the spatial discretization, which was also observed in Section 4.1, further, robustness with respect to varying the contrast of the real parts α and the imaginary parts β of the eigenvalues $\lambda = \alpha + \imath\beta$, and, finally, robustness with respect to varying the coefficient jump k_2. We shall consider both the heat and eddy current equations.

We proceed with the simulations on 4 levels of uniformly refined discretizations of Ω. The numbers of interior nodes, which are the degrees of freedom (DOFs) in case of the heat equation, and the numbers of interior edges, which are the DOFs in case of the eddy current equations, are listed in Tab. 2. At the coarsest level, we want to have cca 10^4 DOFs. Thus, we start at the second discretization level in case of the heat equation (nodal DOFs) and at the first discretization level in case of the eddy current equations (edge DOFs).

Table 2: Non-Dirichlet degrees of freedom (DOFs) at respective discretization levels.

heat eq.	eddy current eq.	N_x	
multigrid levels	multigrid levels	nodal DOFs	edge DOFs
	0	3 569	22 576
0	1	26 145	173 856
1	2	200 001	1 364 288
2		1 564 289	10 808 960
3		12 373 249	86 052 096

First of all, we consider the heat equation and employ the nodal-patch smoother within the multigrid method. In Tab. 3, we present numbers of outer PRESB-FGMRES iterations and corresponding total numbers of inner multigrid-CG iterations when varying from 10^{-6} to 10^6 the real part α of the eigenvalues $\lambda = \alpha + \imath\beta$, while the imaginary part $\beta := 1$ is fixed, and when varying coefficient jump k_2 in the same range while the other value $k_1 := 1$ is fixed. We observe that the numbers of iterations are more less aligned with Lemma 2, i.e., one or two iterations per order of the relative precision 10^{-8} in the

worst cases of small α. The convergence becomes super-optimal for higher α. We do not see much influence of jumping coefficients, which is due to the robust smoother within the multigrid. The numbers of iterations also seem to be bounded independently of the problem size, which is another nice property of the multigrid.

Table 3: Numbers of the outer PRESB-FGMRES iterations and, in brackets, the corresponding total numbers of the inner multigrid-CG iterations for the solutions of (12) for the heat equation and the nodal-patch smoother at multigrid levels 0, 1, 2, and 3 (the respective entries are separated by commas). At level 0, the assembling time ranged between 257 and 300 seconds, the solution time ranged between 1 to 10 seconds and the memory consumption between 343 and 355 MB. At level 1, the assembling time ranged between 293 and 422 seconds, the solution time ranged between 118 to 268 seconds, and the memory consumption was always 1.097 GB. At level 2, the assembling time ranged between 460 and 673 seconds, the solution time ranged between 772 to 3434 seconds, and the memory consumption was always 7.165 GB. At level 3, the assembling time ranged between 4086 and 4151 seconds, the solution time ranged between 14002 to 43819 seconds, and the memory consumption was always 55.633 GB.

k_2	$\alpha := 10^{-6}$	$\alpha := 10^{-3}$	$\alpha := 1$	$\alpha := 10^{3}$	$\alpha := 10^{6}$
10^{-6}	9(18),12(35), 14(38),14(41)	9(18),12(35), 14(38),14(41)	7(14),9(19), 11(26),13(32)	3(6),5(10), 7(14),8(16)	2(4),4(14), 4(14),4(20)
10^{-3}	11(22),14(31), 14(41),14(43)	11(22),14(31), 14(41),14(43)	10(20),11(22), 11(23),12(35)	3(6),4(14), 4(14),4(20)	2(4),4(14), 4(16),4(20)
1	10(20),11(35), 12(42),13(48)	10(20),11(35), 12(42),13(48)	5(10),8(27), 9(32),9(34)	2(4),4(14), 4(14),5(23)	2(4),4(14), 4(14),5(22)
10^{3}	8(16),11(35), 13(43),16(53)	8(16),11(35), 13(43),16(53)	4(8),9(24), 10(29),13(37)	2(4),4(14), 5(18),5(18)	2(4),4(14), 5(18),5(18)
10^{6}	8(16),11(35), 13(43),16(53)	8(16),11(35), 13(43),16(53)	4(8),9(24), 10(29),13(37)	2(4),4(14), 5(18),5(18)	2(4),4(14), 5(18),5(18)

Secondly, we do similar simulations for the eddy current equations and the nodal-patch smoother within the multigrid method. In Tab. 4, we present numbers of outer PRESB-FGMRES iterations and corresponding total numbers of inner multigrid-CG iterations for varying α and k_2. They are again very stable and aligned with the theoretical estimate of the condition number of the PRESB method. The entry, $k_2 = \alpha = 10^6$, produced a numerically singular matrix, for which we could not proceed with the method at levels 1 and 2.

Finally, we present additional results for the heat equation with the Jacobi additive smoother. It is less stable in terms of numbers of iterations, but on the other hand, less memory and time-consuming than the nodal-patch smoother presented in Tab. 3. The results for levels 1 and 2 are presented in Tab. 5. The coarsest level results remain those from Tab. 3.

5 Conclusion

We showed that the preconditioning method PRESB that has been recently developed [17] for a class of complex-valued systems works efficiently for linear systems arising in tensor-product finite element discretizations of parabolic-elliptic problems when solved by the fast diagonalization method. We proved that the generalized eigenvalue problem that underlines the fast diagonalization method produces eigenvalues with positive real parts, which then leads to symmetric positive definite systems to be solved at the inner level. In this paper, we combined the methodology with a geometric multigrid for these inner

Table 4: Numbers of the outer PRESB-FGMRES iterations and, in brackets, the corresponding total numbers of the inner multigrid-CG iterations for the solutions of (12) for the eddy current equations and the nodal-patch smoother at multigrid levels 0, 1, and 2 (the respective entries are separated by commas). At level 0, the assembling time ranged between 167 and 183 seconds, the solution time ranged between 2 to 28 seconds and the memory consumption between 204 and 214 MB. At level 1, the assembling time ranged between 203 and 254 seconds, the solution time ranged between 95 to 250 seconds, and the memory consumption was always 838 MB. At level 2, the assembling time ranged between 550 and 829 seconds, the solution time ranged between 814 to 2429 seconds, and the memory consumption was always 5.797 GB.

k_2	$\alpha := 10^{-6}$	$\alpha := 10^{-3}$	$\alpha := 1$	$\alpha := 10^{3}$	$\alpha := 10^{6}$
10^{-6}	8(16),10(34), 10(36)	8(16),10(34), 10(36)	6(12),7(26), 9(38)	3(6),4(16), 4(22)	7(14),7(32), 8(44)
10^{-3}	11(22),13(29), 13(40)	11(22),13(29), 13(40)	10(20),10(36), 10(51)	3(6),4(16), 4(22)	4(8),6(28), 7(38)
1	8(16),9(34), 10(43)	8(16),9(34), 10(43)	5(10),6(27), 6(34)	3(6),4(16), 4(22)	4(8),6(28), 7(38)
10^{3}	8(16),9(34), 10(44)	8(16),9(34), 10(44)	4(8),6(28), 6(34)	3(6),4(16), 4(22)	5(10),6(26), 7(38)
10^{6}	8(16),9(33), 10(44)	8(16),9(34), 10(44)	4(8),6(28), 6(34)	3(6),4(16), 4(22)	14(136),—, —

Table 5: Numbers of the outer PRESB-FGMRES iterations and, in brackets, the corresponding total numbers of the inner multigrid-CG iterations for the solutions of (12) for the heat equation and the Jacobi smoother at multigrid level 1 and 2 (the respective entries are separated by columns). At level 1, the assembling time ranged between 300 and 409 seconds, the solution time ranged between 56 to 188 seconds, and the memory consumption was always 657 MB. At level 2, the assembling time ranged between 386 and 521 seconds, the solution time ranged between 336 to 1840 seconds, and the memory consumption was always 3.068 GB.

k_2	$\alpha := 10^{-6}$	$\alpha := 10^{-3}$	$\alpha := 1$	$\alpha := 10^{3}$	$\alpha := 10^{6}$
10^{-6}	20(247),24(389)	20(247),23(388)	14(179),17(285)	8(114),7(166)	6(112),5(146)
10^{-3}	20(249),22(417)	20(249),22(417)	14(155),15(254)	6(112),5(146)	6(112),5(146)
1	20(238),23(413)	20(238),23(413)	16(190),18(293)	6(108),6(165)	6(108),6(166)
10^{3}	22(250),28(440)	22(253),28(445)	15(183),15(310)	6(96),6(154)	6(96),6(154)
10^{6}	25(276),39(621)	26(283),39(587)	16(211),30(559)	8(115),14(251)	8(118),13(283)

auxiliary spatial elliptic problems. The overall method is robust with respect to discretization levels, coefficient jumps, and complex eigenvalues related to orthogonalized temporal basis functions. This was numerically confirmed for the 3d heat equation and the 3d eddy current equations. In our next research step, we shall replace multigrid with a domain decomposition method.

References

[1] B. Smith, P. Bjørstad and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 2004.

[2] A. Toselli and O. Widlund, *Domain Decomposition Methods — Algorithms and Theory*, Springer, 2005.

[3] C. Pechstein, *Finite and Boundary Element Tearing and Interconnecting Solvers for Multiscale Problems*, Springer, 2013.

[4] Z. Dostál, T. Kozubek, M. Sadowská and V. Vondrák, *Scalable Algorithms for Contact Problems*, Springer, 2016.

[5] M. Gander, 50 years of time parallel time integration, in: *Multiple Shooting and Time Domain Decomposition Methods*, Springer (2015), 69-113.

[6] J.-L. Lions, Y. Maday and G. Turinici, Résolution d'EDP par un schéma en temps 'pararéel', *C. R. Acad. Sci. Paris, Série I*, **332** (2001), 661-668.

[7] M. Gander, L. Halpern and F. Nataf, Optimal Schwarz waveform relaxation for the one dimensional wave equation, *SIAM J. Numer. Anal.*, **41** (2003), 1643-1681.

[8] L. Foltyn, D. Lukáš and I. Peterek, Domain decomposition methods coupled with parareal for the transient heat equation in 1 and 2 spatial dimensions, *Appl. Math.*, **65** (2020), 173-190.

[9] M. Neumüller, Space-Time Methods: Fast Solvers and Applications, *Monographic Series TU Graz*, **20** (2013).

[10] M. Gander and M. Neumüller, Analysis of a new space-time parallel multigrid algorithm for parabolic problems, *SIAM J. Sci. Comput.*, **38** (2016), A2173–A2208.

[11] O.A. Ladyshenskaja, V.A. Solonnikov and N. N. Uralceva, *Linear and Quasi-Linear Equations of Parabolic Type*, American Mathematical Society, 1968.

[12] C. Hofer, U. Langer, M. Neumüller and R. Schneckenleitner, Parallel and robust preconditioning for space-time isogeometric analysis of parabolic evolution problems, *SIAM J. Sci. Comput.*, **41** (2019), A1793-A1821.

[13] M. Zank, Inf-Sup Stable Space-Time Methods for Time-Dependent Partial Differential Equations, *Monographic Series TU Graz*, **36** (2020).

[14] U. Langer and M. Zank, Efficient direct space-time finite element solvers for parabolic initial-boundary value problems in anisotropic Sobolev spaces, *SIAM J. Sci. Comput.*, **43** (2021), A2714-A2736.

[15] R. Lynch, J. R. Rice and D. H. Thomas, Direct solution of partial difference equations by tensor product methods, *Numer. Math.* **6** (1964), 185-199.

[16] O. Steinbach, Space-time finite element methods for parabolic problems, *Comput. Methods Appl. Math.*, **15** (2015), 551-566.

[17] O. Axelsson and D. Lukáš, Preconditioning methods for eddy-current optimally controlled time-harmonic electromagnetic problems, *J. Numer. Math.*, **27** (2019), 1-21.

[18] G. Loli, M. Montardini, G. Sangalli, and M. Tani, An efficient solver for space-time isogeometric Galerkin methods for parabolic problems, *Comput. Math. Appl.*, **80** (2020), 2586–2603.

[19] W. Zulehner, Nonstandard norms and robust estimates for saddle point problems, *SIAM J. Matrix Anal. Appl.* **32** (2011), 536-560.

[20] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, *SIAM J. Sci. Comput.*, **14** (1993), 461-469.

[21] J. C. Nédélec, Mixed finite elements in $\mathbb{R}^3$, *Numer. Math.* **35** (1980), 315-341.

[22] D. N. Arnold, R. S. Falk and R. Winther, Multigrid in H(div) and H(curl), *Numer. Math.*, **85** (2000), 197-217.

[23] G. Guennebaud, B. Jacob et al., Eigen v3, *http://eigen.tuxfamily.org*, 2010.

[24] T. A. Davis, J. R. Gilbert, S. I. Larimore and E. G. Ng, Algorithm 836: COLAMD, a column approximate minimum degree ordering algorithm, *AMS Trans. Math. Software* **30** (2004), 377-380.

[25] J. Schöberl, NETGEN - an advancing front 2d/3d-mesh generator based on abstract rules, *Comp. Vis. Sci.* **1** (1997), 41-52.