Parallel solution methods  
and preconditioners for evolution equations

Owe Axelsson$^{a,b}$, Maya Neytcheva$^b$, and Zhao-Zhang Liang$^{c,b}$

\textit{a}Institute of Geonics, Czech Academy of Sciences, Ostrava, Czech Republic  
\textit{b}Department of Information Technology, Uppsala University, Uppsala, Sweden  
\textit{c}School of Mathematics and Statistics, Lanzhou University, Lanzhou, People's Republic of China  
owe.axelsson@it.uu.se, maya.neytcheva@it.uu.se, liangzz11@lzu.edu.cn

Abstract. The recent development of the high performance computer platforms shows a clear trend towards heterogeneity and hierarchy. In order to utilize the computational power particular attention must be paid to finding new algorithms or adjust existing ones so that they better match the HPC computer architecture.

In this work we consider an alternative to classical time-stepping methods based on use of time-harmonic properties and discuss solution approaches that allow efficient utilization of modern HPC resources.

The method in focus is based on a truncated Fourier expansion of the solution of an evolutionary problem. The analysis is done for linear equations and it is remarked on the possibility to use two- or multilevel methods for nonlinear problems, which can add to an even higher degree of parallelization.

The arising block matrix system to be solved admits a two-by-two block form with square blocks, for which a very efficient preconditioner exists. It leads to tight eigenvalue bounds for the preconditioned matrix and, hence, to a very fast convergence of a preconditioned Krylov subspace or iterative refinement method. The analytical background is shown as well as some illustrating numerical examples.

1 Introduction

Traditional solution methods for time-dependent partial differential equations are based on time-stepping methods. These methods can be explicit or implicit, however they are sequential by nature and, thus, slow. The methods are also often computationally very costly. For reasons of numerical stability explicit methods require often the use of very small time-steps. Even if an implicit method is stable one must still choose somewhat small time-steps to achieve a sufficiently small time-discretization error or use higher order stable methods which adds to the computational complexity.

Each step is costly to execute as it requires the solution of a normally elliptic, though scaled, equation on each time interval. Clearly, the methods are sequential. Therefore, they are costly and time-consuming. However, there exist alternatives to time-stepping methods.

In many practically important problems, such as in optimal control problems, the control function is periodic, for instance an alternating current in electromagnetic
problems. Then the problem becomes time-harmonic and the solution can be approximated with a truncated Fourier series expansion. Due to the orthogonality of the trigonometric functions, for linear problems the computation of the Fourier coefficients separate and one can compute the solution for each period fully in parallel. Hence, the solution process is perfectly parallelizable across the different frequencies. Due to the large size of the discretized systems one must use iterative solution methods, see e.g. [1],[2],[3],[4]. The preconditioner used should then also be well parallelizable.

For each frequency, a system on two-by-two or four-by-four block matrix arises. In a series of publications ([5–10]) it has been demonstrated that for such problems a very efficient preconditioner, the preconditioned square matrix block, (PRESB) method, exists and leads to very tight eigenvalue bounds and hence few iterations. This holds uniformly with respect to mesh size, control cost and other regularization parameters as well as to the problem parameters. The method has been shown to outperform other published methods for the problems considered. We argue further, that the structure of the PRESB method allows for additional levels of parallelization, achieved by applying well-developed and efficient computational toolboxes for the basic operator equations arising in the PRESB method, such as the Algebraic Multigrid (AMG) method, provided via publicly available scientific computing libraries.

For nonlinear problems one can use a two-grid or multi-level method, such as presented in [11–13], i.e. solve the nonlinear equation on a coarse grid, interpolate the solution on the fine grid and compute a corrected solution of the linearized Newton type equation just once, which normally suffices to get a solution with an error of the same order as the fine mesh discretization error. Such approaches can save much computational labor.

The paper presents the idea of using truncated Fourier series expansions in Section 2, an introductory example of an optimal control problem for an evolution equation in Section 3, the description of the solution method and the preconditioner in Section 4, a short presentation of the suggested approach for nonlinear problems in Section 5 and some discussion of the computational and communication costs of the methods are described in Section 6. An illustrating example for the heat equation and, following [8], a more involved example in the form of an eddy current electromagnetic problem with numerical test results are presented in Section 7.

2 Truncated Fourier series expansions for linear evolution equations

Consider an evolution equation

$$\frac{\partial u}{\partial t} + Lu = f, \quad 0 < t < T, \quad x \in \Omega \subset \mathbb{R}^d, \quad d = 1, 2 \text{ or } 3,$$

(1)

where $L$ is a time-independent linear and coercive elliptic type differential operator with boundary and initial conditions

$$u = u(x,t), \quad u(x,t) = 0 \quad \text{on } \partial \Omega, \quad u(x,0) = \hat{u}_0(x).$$
2.1 Truncated Fourier expansion

In order to get a periodic solution we extend the problem (1) symmetrically to the time interval to \((0,2T]\), where

\[
f(x,2T-t) = f(x,t), \quad 0 < t \leq T.
\]

Hence, it holds also that \(u(x,2T-t) = u(x,t), \quad 0 < t \leq T\), that is, the solution on the interval \([T,2T]\) is the mirror of the solution on \([0,T]\), see Figure 2.1 for an illustration.

![A symmetrically extended function](image)

**Fig. 1.** A symmetrically extended function

The differential equation then becomes an evolution equation with periodic conditions,

\[
\begin{align*}
\frac{\partial u}{\partial t} + Lu &= f, \quad 0 < t < 2T, \quad x \in \Omega, \\
u(x,t) &= 0, \quad x \in \partial \Omega, \quad \text{and} \\
u(x,0) &= u(x,2T).
\end{align*}
\]

This enables the use of a truncated Fourier series,

\[
u(x,t) = \sum_{k=0}^{N} u_k(x)\cos(\omega_k t) + iv_k(x)\sin(\omega_k t),
\]

where \(\omega_k = k \frac{\pi}{T}, \quad k = 0,1,\ldots,N\) are the angular velocities and \(u_k(x),v_k(x)\) are the Fourier coefficients at \(x \in \Omega\). Substituting (3) in the differential equation (2), we obtain

\[
\sum_{k=0}^{N} \left( -\omega_k u_k(x)\sin(\omega_k t) + i\omega_k v_k(x)\cos(\omega_k t) + \sum_{k=0}^{N} \nabla u_k(x)\cos(\omega_k t) + i\nabla v_k(x)\sin(\omega_k t) \right) = f.
\]
A multiplication with $\cos(\omega_k t)$ respectively with $\sin(\omega_k t)$, and integration over the time interval $(0, 2T)$, thereby using the orthogonality of the trigonometric functions, results in the equations

$$\begin{cases}
i \omega_k v_k(x) + Lu_k(x) = f^c_k(x) := \frac{1}{T} \int_0^{2T} f(x, s) \cos(\omega_k s) ds, \\
-\omega_k u_k(x) + iLv_k(x) = f^s_k(x) := \frac{1}{T} \int_0^{2T} f(x, s) \sin(\omega_k s) ds = 0.
\end{cases} \quad (4)$$

Here we have used the elementary relations $\frac{1}{T} \int_0^{2T} \cos^2(\omega_k s) ds = \frac{1}{T} \int_0^{2T} \sin^2(\omega_k s) ds = 1$ and the symmetry of $f(x, t) = f(x, 2T - t)$, which gives $f^s_k(x) = 0$.

A proper choice of $N$ can be made by trial and error. In some problems one can instead utilize classical truncation error expressions and estimate the solution error via the truncation error. We refer to [14, 15] for such an estimate showing that the error in the Fourier expansion decreases at least as $O(1/\sqrt{N})$, $N \to \infty$. In practice, the solution is often close to a few term Fourier expansion, so it is not needed to choose a large value of $N$.

### 2.2 Galerkin variational FEM

Let $u_k \in U$, $v_k = V$, where $U = V$ is a finite element subspace in $H_0(\Omega)$. Let $M$ be the corresponding Galerkin mass matrix and $L$ the corresponding stiffness matrix, for the inner product in $L_2(\Omega)$. Then after multiplying the equations (4) with the basis functions in $U$ and $V$ and forming the finite element (FEM) variational formulation of (4), the equations take the algebraic form

$$\begin{cases}
L u_k + i\omega_k M v_k = f^c_k \\
-\omega_k M u_k + iLv_k(x) = 0
\end{cases} \quad (5)$$

where $u_k, v_k$ are the discrete FE solution vectors and $f^c_k$ the vector representation of the source function corresponding to $\omega_k$ and the chosen basis functions.

For $k=0$ we get the solutions

$L u_0 = f^c_0, \quad L v_0 = 0, \quad i.e. \; v_0 = 0.$

We multiply the second equation in (5) by the complex unit number $i$ to get the block matrix system,

$$\begin{pmatrix}
L & i\omega_k M \\
-\omega_k M & -L
\end{pmatrix}
\begin{bmatrix}
u_k \\
v_k
\end{bmatrix} =
\begin{bmatrix}
f^c_k \\
0
\end{bmatrix}. \quad (6)$$

The system in (6) can be further reduced to the fourth order problem,

$$(LM^{-1}L + \omega_k^2 M)u_k = LM^{-1}f^c_k. \quad (7)$$

However, to avoid working with 4'th order operators we keep the two-by-two block form in (6). Note that the matrix in (6) is indefinite and Hermitian positive definite. If one prefers to use only real arithmetics, the two-by-two block complex matrix can be reformulated as a four-by-four real-valued block matrix, see e.g. [8], for an example of this.

Optimal control problems with a PDE as state equation has been studied in many publications. To this paper related such publications are for instance [16, 17], [7], and the previously mentioned [5–10].
3 An optimal control problem with an evolution state equation

In general the solution to an evolution equation shows a less smooth behaviour initially before the exponential decay caused by the elliptic nature of the problem has had any significant damping influence. This will occur particularly late in time if the diffusion coefficient in $L$ is relatively small. However, eventually the solution becomes close to the stationary solution, $Lu_\infty = f$, as $t \to \infty$.

Our aim is to find an initial value function which makes the convergence to the stationary solution smooth for all times, $0 < t < T$. In addition, we want to find a source control function such that the solution for $t = T$ is close to some given, target solution $u_d(x)$.

Hence, given a target (desired) solution $u_d(x,t), x \in \Omega, t \in (0,T)$, we want to find a solution $u(x,t)$ of the evolution equation,

$$\frac{\partial u}{\partial t} + Lu = v(x,t), \quad (x,t) \in \Omega \times (0,T],$$

where $v$ is the control function with, for simplicity, homogeneous boundary conditions $u(x,t) = 0, x \in \Omega,$ that is close to $u_d$. Here $\Omega$ is a given bounded Lipschitz domain in $\mathbb{R}^d$, $d=1,2,$ or $3$ and $L$ is an elliptic operator, i.e. satisfies a coercivity condition, $(Lu, \tilde{u}) \geq \alpha (u, \tilde{u}), \alpha > 0$, for all $\tilde{u} \in H^1(\Omega)$. As an example $L$ is a convection-diffusion operator, $Lu = -\nu \Delta u + \mathbf{w} \cdot \nabla u + cu$, where $\nu > 0$ and $c - \frac{1}{2} \nabla \cdot \mathbf{w} \geq 0$. The solution will be controlled by the source function $v \in L^2(\Omega) \times (0,T]$ which, for simplicity we assume is distributed in the whole domain and also satisfies the homogeneous boundary conditions. Such optimal control problems must be regularized to obtain a well-posed problem. For this purpose, we use the simplest Tichonov regularization form. Let $\lambda$ be a Lagrange multiplier to handle the constraint equation $\frac{\partial u}{\partial t} + Lu = v$. The optimal control problem can then be formulated as a saddle type functional,

$$\min_{u,v} \max_{\lambda} J(u,v,\lambda),$$

where the Lagrangian functional equals

$$J(u,v,\lambda) = \frac{1}{2} \int_0^T \int_{\Omega} (u(x,s) - u_d(x,s))^2 dx ds + \int_0^T \int_{\Omega} \lambda \left( \frac{\partial u}{\partial t} + Lu - v \right) dx ds + \frac{1}{2} \beta \int_0^T \int_{\Omega} v^2(x,s) dx ds. \quad (8)$$

Here $\beta > 0$ is the regularization parameter for the cost of the control function $v$. The first order necessary conditions, $\nabla_{u,v,\lambda} J(u,v,\lambda) = 0$, are also sufficient for the existence of a solution. Here $\frac{\partial J}{\partial v} = 0$ leads to the simple relation, $-\lambda + \beta v = 0$, which
enables reducing the arising linear system. Hence, substituting $\lambda = \beta v$ in (8) gives

$$\tilde{J}(u,v) := \mathcal{L}(u,v,\beta v) = \frac{1}{2} \int_0^T \int_{\Omega} (u(x,s) - u_d(x,s))^2 dx ds +$$

$$+ \beta \int_0^T \int_{\Omega} v \frac{\partial u}{\partial s} + Lu \, dx ds - \frac{1}{2} \beta \int_0^T \int_{\Omega} v^2(x,s) ds =$$

$$= \frac{1}{2} \int_0^T \int_{\Omega} (u(x,s) - u_d(x,s))^2 dx ds + \beta \int_0^T \int_{\Omega} \left( - \frac{\partial v}{\partial s} + L^* v \right) ud ds$$

where $J^*$ is the adjoint operator to $J$ and where we have made use of homogeneous boundary conditions. The first order conditions give now

$$\begin{cases} u + \beta (-\frac{\partial u}{\partial s} + L^* v) = u_d \\ \beta (\frac{\partial u}{\partial s} + Lu) - \beta v = 0 \end{cases} \text{ or } \begin{cases} u + \sqrt{\beta} (-\frac{\partial v}{\partial s} + L^* \tilde{v}) = u_d \\ -\sqrt{\beta} (\frac{\partial u}{\partial s} + Lu) + \tilde{v} = 0 \end{cases}$$

(9)

where $\tilde{v} = \sqrt{\beta} v$.

Choosing a proper set of finite element basis functions and multiplying the equations in (9)(right) with these basis functions as test functions, we obtain the FEM formulation, corresponding to (9),

$$\begin{cases} M \hat{u}(t) + \sqrt{\beta} (-\frac{\partial}{\partial s} M \tilde{v}(t) + L^* \tilde{v}(t)) = u_d(t) \\ -\sqrt{\beta} (\frac{\partial}{\partial s} M \hat{u}(t) + Lu(t)) + M \tilde{v}(t) = 0 \end{cases} \text{ .}$$

(10)

Here $M$ is the mass matrix corresponding to the basis functions and the $L_2$-inner product, $L$ is the stiffness matrix corresponding to the operator $\mathcal{L}$, and $u(t)$, $u_d(t)$, $\tilde{v}(t)$ denote the finite element vectors, corresponding to the interior nodepoints in $\Omega_h$, the discrete finite element mesh.

We approximate the time-dependent vectors $u(t)$, $u_d(t)$, $\tilde{v}(t)$ with truncated Fourier series expansions,

$$\sum_{k=1}^{N} \frac{u_k^e + iu_k^s}{v_k^e + iv_k^s} e^{-i\omega_k t} ,$$

(11)

where the frequencies $\omega_k = k\pi/T$, $k = 1,2,\ldots,N$, and where $\begin{bmatrix} u(t) \\ v(t) \end{bmatrix}$ equals its real-valued part, $\sum_{k=1}^{N} \frac{u_k^e}{v_k^e} \cos(\omega_k t) + \frac{u_k^s}{v_k^s} \sin(\omega_k t)$.

Due to the orthogonality of the trigonometric functions, for linear problems, as we consider here, the computations of the different Fourier coefficients separate, so it suffices to consider the resulting equations for just one index $k$. From (10) follows then,
Here we separate the real and imaginary parts for each equation, to get
\[
\begin{align*}
\begin{bmatrix}
M & 0 \\
0 & M
\end{bmatrix}
\begin{bmatrix}
u_k^c \\
u_k^s
\end{bmatrix}
+ \sqrt{\beta} w_k
\begin{bmatrix}
0 & -M \\
M & 0
\end{bmatrix}
\begin{bmatrix}
v_k^c \\
v_k^s
\end{bmatrix}
+ \sqrt{\beta}
\begin{bmatrix}
L^* & 0 \\
0 & L^*
\end{bmatrix}
\begin{bmatrix}
v_k^c \\
v_k^s
\end{bmatrix}
= 
\begin{bmatrix}
u_{d,k}^c \\
u_{d,k}^s
\end{bmatrix},
\end{align*}
\]
\[
\begin{align*}
\sqrt{\beta} w_k
\begin{bmatrix}
0 & -M \\
M & 0
\end{bmatrix}
\begin{bmatrix}
u_k^c \\
u_k^s
\end{bmatrix}
- \sqrt{\beta}
\begin{bmatrix}
L & 0 \\
0 & L
\end{bmatrix}
\begin{bmatrix}
u_k^c \\
u_k^s
\end{bmatrix}
+ \sqrt{\beta}
\begin{bmatrix}
L^* & 0 \\
0 & L^*
\end{bmatrix}
\begin{bmatrix}
u_k^c \\
u_k^s
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0
\end{bmatrix}.
\end{align*}
\]

or, in block matrix form,
\[
\begin{bmatrix}
M & 0 \\
0 & M
\end{bmatrix}
\begin{bmatrix}
u_k^c \\
u_k^s
\end{bmatrix}
\begin{bmatrix}
\sqrt{\beta} & 0 \\
0 & \sqrt{\beta}
\end{bmatrix}
\begin{bmatrix}
L & 0 \\
0 & L
\end{bmatrix}
\begin{bmatrix}
u_k^c \\
u_k^s
\end{bmatrix}
\begin{bmatrix}
0 \\
0
\end{bmatrix}
= 
\begin{bmatrix}
u_{d,k}^c \\
u_{d,k}^s
\end{bmatrix}.
\]

(12)

Note that the off-diagonal blocks are skew-Hermitian. Let \( \hat{A} = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix}, \hat{B} = -\sqrt{\beta} \begin{bmatrix} L & \omega_k M \\ -\omega_k M & L \end{bmatrix} \). Then the block matrix in (12) can be written,
\[
\begin{bmatrix}
\hat{A} - \hat{B}^* \\
\hat{B} & \hat{A}
\end{bmatrix}.
\]

4 Preconditioning method

Following the approach used in numerous earlier papers, [5–10], as a preconditioner to \( \mathcal{A} \), we choose
\[
\begin{align*}
\mathcal{C} = \begin{bmatrix}
\hat{A} + \hat{B} + \hat{B}^* - \hat{B}^*

\end{bmatrix} = \begin{bmatrix}
\hat{A} + (\hat{L} + \hat{L}^*) - \hat{B}^*

\end{bmatrix},
\end{align*}
\]

(13)

where \( \hat{L} = \sqrt{\beta} \begin{bmatrix} L & 0 \\ 0 & L \end{bmatrix} \). The preconditioned matrix can be written,
\[
\mathcal{C}^{-1} \mathcal{A} = \begin{bmatrix}
I & 0 \\
0 & I
\end{bmatrix} \begin{bmatrix}
\hat{L} + \hat{L}^* & 0 \\
0 & 0
\end{bmatrix}.
\]

Further, as shown below,
\[
\mathcal{C}^{-1} = \begin{bmatrix}
I & 0 \\
0 & I
\end{bmatrix} \begin{bmatrix}
\hat{A} + \hat{L} \\
-\hat{L}
\end{bmatrix} \begin{bmatrix}
(\hat{A} + \hat{L})^{-1} & 0 \\
0 & I
\end{bmatrix},
\]

so
\[
\mathcal{C}^{-1} \begin{bmatrix}
\hat{L} + \hat{L}^* \\
0 & 0
\end{bmatrix} = \begin{bmatrix}
I & 0 \\
0 & I
\end{bmatrix} \begin{bmatrix}
(\hat{A} + \hat{L})^{-1} & 0 \\
0 & I
\end{bmatrix} \begin{bmatrix}
(\hat{A} + \hat{L})^{-1} & 0 \\
0 & I
\end{bmatrix}.
\]
Therefore, besides some matrix vector multiplication, an action of the preconditioned matrix, involves only a solution with the elliptic type matrices \( A + L^* \) and \( A + L \).

In addition, as outlined below and has been shown in previous publications [7–9], the preconditioning leads to tightly clustered eigenvalues and, hence, very fast convergence.

We present now the preconditioner for a matrix in the general form \( A = \begin{bmatrix} A & B_2 \\ B_1 & -A \end{bmatrix} \), where \( A \), of order \( n \times n \), is assumed to be symmetric and positive definite and \( A + B_i \), \( i = 1, 2 \) are nonsingular. Let

\[
C = \begin{bmatrix} A + B_1 + B_2 & B_2 \\ B_1 & -A \end{bmatrix}
\]

be a preconditioner to \( A \), to be used in a Krylov subspace type of iteration method, such as GMRES [18] or MinRes [19]. Given a linear matrix preconditioning equation

\[
C \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix},
\]

by changing the sign of the second equation and adding the first, the system can be written in the equivalent form,

\[
\begin{cases}
(A + B_1 + B_2)x + B_2y = f \\
(A + B_2)x + (A + B_2)y = f - g
\end{cases}
\]

i.e.,

\[
\begin{bmatrix} A + B_1 & B_2 \\ 0 & A + B_2 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} f \\ f - g \end{bmatrix},
\]

where \( z = x + y \). Hence, \( z = (A + B_2)^{-1}(f - g) \) and \( (A + B_1)x = f - B_2z \). Therefore, the algorithm to compute the solution of (15) can be written as

1. Solve \( (A + B_2)z = f - g \)
2. Compute \( f = f - B_2z \)
3. Solve \( (A + B_1)x = \tilde{f} \)
4. Compute \( y = z - x \)

Alternatively, it is readily seen that

\[
C = \begin{bmatrix} A + B_1 + B_2 & B_2 \\ I & I \end{bmatrix} \begin{bmatrix} I & 0 \\ I & -(A + B_2) \end{bmatrix} \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} = \begin{bmatrix} A + B_1 & 0 \\ 0 & I \end{bmatrix}
\]

so

\[
C^{-1} = \begin{bmatrix} (A + B_1)^{-1} & \tilde{0} \\ -(A + B_1)^{-1} \tilde{I} & \tilde{I} \end{bmatrix} \begin{bmatrix} I & 0 \\ -B_2 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ (A + B_2)^{-1} & -(A + B_2)^{-1} \end{bmatrix}.
\]

Hence, besides some vector additions, the algorithm involves a solution of a linear systems with \( A + B_2 \), a matrix vector multiplication with \( B_2 \) and a solution with matrix \( A + B_1 \). In practice, the solution of the two linear systems contribute to the major cost of computing an action of \( C^{-1} \). As we shall see, in our applications they correspond to elliptic operators.

As the following proposition shows, the eigenvalues are tightly clustered near the unit value.

**Proposition 1.** Let \( A = \begin{bmatrix} A & B_2 \\ -B_1 & A \end{bmatrix} \), \( C = A + \begin{bmatrix} B_1 + B_2 & 0 \\ 0 & 0 \end{bmatrix} \), where we assume that \( A \), of order \( n \times n \), is symmetric positive definite and \( B_2 = B_1^* \), \( B_1 = B \), \( B + B^* \)}
is positive semidefinite and $A + B$ is nonsingular. Then the eigenvalues of $C^{-1}A$ are real and satisfy $\frac{1}{2} \leq \frac{1}{1+\alpha} \leq \lambda \leq 1$, where $\alpha = \max \{\text{Re}(\mu)/|\mu|\}$, and $\mu$ is an eigenvalue of $Bz = \mu Az$, $z \neq 0$. The dimension of the eigenvalue $\lambda = 1$ is $n + n_0$, where $n_0$ is the dimension of the nontrivial nullspace of $B_1 + B_2$. It follows that $\lambda = \frac{1 + |\mu|^2}{1 + |\mu|^2 + 2\text{Re}(\mu)} = \frac{1 + |\mu|^2}{1 + |\mu|^2 + 2\alpha}$, where $\alpha$ is the dimension of the nontrivial nullspace of $A$.

Proof. From $\mathbb{C}^\text{C}[x, y] = A[x, y], \|x\| + \|y\| \neq 0$, follows $(1 - \lambda)\mathbb{C}[x, y] = (B_1 + B_2)x$. Since $B_1 + B_2$ are positive semidefinite, it follows that $\lambda \leq 1$. Further, for $\lambda \neq 1$, it holds $B_1x = Ay$. Hence $(1 - \lambda)(A + B_1 + B_2 + B_2A^{-1}B_1)x = (B_1 + B_2)x$ or since $A$ is spd, $(1 - \lambda)(I + \tilde{B} + \tilde{B}^* + \tilde{B}^*\tilde{B})x = (\tilde{B} + \tilde{B}^*)x$, where $\tilde{B} = A^{-1/2}BA^{-1/2}$. Therefore $(1 - \lambda)(1 + 2\text{Re}(\mu) + |\mu|^2) = 2\text{Re}(\mu) \text{ or } \frac{2\text{Re}(\mu)/|\mu|}{|\mu|^2 + |\mu|^2} \leq \alpha/(1 + \alpha)$. See also [8, 9].

Proposition 1 shows that the relative size, $\text{Re}(\mu)/|\mu|$ of the real part of the spectrum of $\tilde{B} = A^{-1/2}BA^{-1/2}$ determines the lower eigenvalue bound of $C^{-1}A$ and hence the rate of convergence of the preconditioned iterative solution method. For a small such relative part the convergence of the iterative solution method will be exceptionally rapid. Such small parts can occur for harmonic problems with a large value of the frequency.

It is further readily seen that the eigenvalue problem has a full eigenvector solution space, i.e. the preconditioned matrix $C^{-1}A$ is diagonalizable by a similarity transformation formulation, see e.g. [4].

5 A multilevel mesh solution method for solving nonlinear evolution equations

For a nonlinear problem, such as $\frac{\partial u}{\partial t} + \mathcal{L}(u)u = f$ where $\mathcal{L}(u)u = -\nabla \cdot (a(u))\nabla u + w(u) \cdot \nabla u = f$ the arising block matrix system is nonlinear and one must use some form of nonlinear solver.

In a similar way as done for the classical Newton method, to ensure convergence and fast convergence of the nonlinear solver, we need a high quality initial guess see e.g. [11]. For this purpose we can use (nested) discretization meshes, $T_0 \subset T_1 \subset ... \subset T_m$, obtained by regular refinements of some given coarse mesh ($T_0$). A scheme to obtain a good starting vector, used e.g., in [20], is depicted in Algorithm 1.

**Algorithm 1** Constructing an initial guess for the nonlinear solver using the full mesh hierarchy

1. Initialize $a, b$ and the solution at $T_0$
2. For all $i = 0, 1, ..., m$
   1. Solve the nonlinear system on level $i$
   2. If $i < m$, prolong the current solution to level $i + 1$
3. EndFor

Thus, in order to solve the nonlinear system on level $m$ we solve it also on all coarser meshes. This, of course, makes the solution procedure computationally demanding.
In [20, 10] the following idea is shown to be very fruitful: solve (accurately) the nonlinear problem on some coarse mesh, prolong the so-obtained solution to the finest \((T_m)\) and solve the linearized problem only once, gaining under certain conditions the order of the accuracy of the discretization of the finest level.

We note that this idea is not always directly applicable for optimal control problems. For instance, in the framework of the semi-smooth Newton method applied for problems with a box-constrained solution, the stopping criterion is not related to a certain tolerance but to a set of point identifiers where the box-constraint values are taken exactly and that stop changing. In this sense we cannot control the accuracy of the solution on a mesh by a tolerance and continue the nonlinear iterations until convergence is achieved. The two-level framework for the semi-smooth Newton method can then be applied as follows. Instead of using all levels in the mesh hierarchy, we choose one (or more, not necessarily consecutive coarser meshes), solve the nonlinear problem there, prolong to \(T_m\) and perform only one Newton step there, thus, solve the linear problem only once. In [10], the effect of using different numbers of hybrid meshes is nicely illustrated.

6 Computational and communication complexity of the proposed approach

To compute the vectors \(u_k\) and \(v_k\) for a linear problem as in (11) for each frequency \(\omega_k\) can solve \(N\) systems (12) independently. These solutions do not require any communication. Communication occurs only when the individual components per frequency must be summed up.

The system in (12) is assumed to be solved by an iterative solution method that accommodates variable preconditioning, such as FGMRES ([18]). Each iteration requires one matrix-vector multiplication, a few vector updates and scalar products and a solution of a system with the preconditioner \(C\). In turn, the latter requires two solutions with the matrices

\[
F = \begin{bmatrix}
M + \sqrt{\beta}L & -\omega_k \sqrt{\beta}M \\
\omega_k \sqrt{\beta}M & M + \sqrt{\beta}L
\end{bmatrix}
\text{ and } F^* = \begin{bmatrix}
M + \sqrt{\beta}L^* & \omega_k \sqrt{\beta}M \\
-\omega_k \sqrt{\beta}M & M + \sqrt{\beta}L^*
\end{bmatrix}
\]

(16)

plus one multiplication with \(L^* - \omega_k M\).

Using our preconditioner the solution with \(F\) and \(F^*\) boils down to solutions with the matrices

\[
Q = (1 + \omega_k \sqrt{\beta})M + \sqrt{\beta}L\text{ and } Q^* = (1 + \omega_k \sqrt{\beta})M + \sqrt{\beta}L^*,
\]

(17)

and a multiplication with \(\omega_k \sqrt{\beta}M\). For the solution with \(Q\) and \(Q^*\) we can use the Conjugate gradient method, preconditioned by AMG, see e.g. [21, 22] or by some other well-parallelizable preconditioner such as special versions of algebraic multilevel methods [23, 24, 8] or such as the gaining popularity Monte Carlo-based approximate inverse preconditioning methods (cf. e.g. [25]).

Assuming that the discretization mesh partitioning is done using an appropriate software tool, such as ParMetis ([26]), we expect a well-balanced distribution of the computational work, good data locality, no communication bottlenecks and good scalability and efficiency of the proposed solution method.
7 Illustrating test problems

For the purpose of illustrating the approximation of the time-dependent solution and the robustness of the method with respect to mesh and regularization parameters we consider first the heat problem. Then, to illustrate the robustness of the preconditioning method with respect to several parameters, i.e. mesh, various regularization, angular frequency and problem parameter, we consider an eddy-current time-harmonic problem. All numerical experiments are performed in Matlab.

7.1 The heat equation

We consider the heat problem as state equation in the optimal control problem, namely, Problem 1.

\[
\frac{\partial}{\partial t} u(x,t) - \nabla \cdot (\kappa(x) \nabla u(x,t)) = v(x,t) \quad \text{for all } (x,t) \in \Omega \times (0,T],
\]

\[
u(x,t) = 0 \quad \text{for all } (x,t) \in \partial \Omega \times (0,T],
\]

\[
u(x,0) = u_d(x,0) = 0 \quad \text{for all } x \in \Omega,
\]

where \(u_d\) is the desired solution and is taken as in [15],

\[
u_d = \chi_{[\frac{1}{4}, \frac{1}{2}]}(t) \chi_{[\frac{1}{2}, 1]}^2(x).
\]

Above, \(\Omega = [0,1]^2\), \(\kappa(x)\) is the thermal conductivity. The function \(\chi\) is the characteristic function, being equal to one when the argument is in the given interval and zero elsewhere. In alignment with the assumptions, the solution is extended to the interval \([0,2T]\) by use of symmetric mirroring. The space discretization is done using bilinear finite element basis functions on square discretization mesh.

We assume that \(u_d\) is of the form (3) and use only up to five terms in the extension, i.e., \(N = 0, 1, \ldots, 5\).

The solution procedure is as follows. The outer solution method and the solver for the blocks \(\mathcal{F}\) and \(\mathcal{F}^*\), appearing in the preconditioner \(C ((16))\) is FGMRES. The very inner solver for the blocks \(Q\) and \(Q^*\) (cf. (17)) is the conjugate gradient, preconditioned by AGMG (cf. [21]). The (relative) stopping criteria are \(10^{-6}\), \(10^{-3}\) and \(10^{-3}\), correspondingly. The results for various \(h, N\) and \(\beta\) are presented in Table 1. As in [15], \(\omega\) is fixed as \(2\pi\). The iterations are shown in the form \(IT_{outer}(IT_{inner}^{av}/IT_{agmg}^{av})\), where \(IT_{outer}\) is the number of outer FGMRES iterations, \(IT_{inner}^{av}\) is the average number of inner FGMRES iterations and \(IT_{agmg}^{av}\) is the average number of the AGMG-preconditioned conjugate gradient. Although the implementation is in Matlab, we include as a reference the total execution time (in seconds).

As shown in Table 1, the number of outer iterations varies between 6 and 8 for all chosen mesh sizes \(h\) and regularization parameter \(\beta\). The number of inner iterations varies between 1 and 4 and the inner-most iterations using the AGMG-preconditioned conjugate gradient varies also between 2 and 7. This means that the method shows a very robust behaviour. It is worth noting that even though we use Matlab, the total
Table 1. Problem 1: performance results for FGMRES, preconditioned by $C$ in (13)

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>$h = \frac{1}{64}$</th>
<th>$h = \frac{1}{128}$</th>
<th>$h = \frac{1}{256}$</th>
<th>$h = \frac{1}{512}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta = 10^{-2}$</td>
<td>0 7(1/3) 0.33</td>
<td>1 7(3/7) 2.38</td>
<td>2 7(4/7) 14.01</td>
<td>0 7(1/4) 22.52</td>
</tr>
<tr>
<td></td>
<td>0.33 8(1/3) 0.32</td>
<td>1.74 8(2/6) 1.74</td>
<td>14.01 8(3/6) 12.13</td>
<td>22.52 8(1/4) 18.81</td>
</tr>
<tr>
<td>$\beta = 10^{-4}$</td>
<td>1 7(3/6) 0.58</td>
<td>2 7(4/6) 2.92</td>
<td>3 6(4/6) 2.60</td>
<td>1 7(3/7) 66.40</td>
</tr>
<tr>
<td></td>
<td>0.43 8(2/5) 0.43</td>
<td>2.52 8(3/6) 2.52</td>
<td>2.60 8(3/6) 2.47</td>
<td>45.75 8(2/7) 45.75</td>
</tr>
<tr>
<td>$\beta = 10^{-6}$</td>
<td>2 7(4/6) 0.71</td>
<td>3 6(4/6) 2.60</td>
<td>4 6(4/6) 2.67</td>
<td>2 7(4/7) 71.25</td>
</tr>
<tr>
<td></td>
<td>0.62 8(3/5) 0.62</td>
<td>2.47 8(3/6) 2.47</td>
<td>2.67 8(3/6) 2.44</td>
<td>63.22 8(3/7) 63.22</td>
</tr>
<tr>
<td>$\beta = 10^{-8}$</td>
<td>3 6(4/6) 0.67</td>
<td>4 6(4/6) 2.67</td>
<td>5 6(4/6) 2.62</td>
<td>3 6(4/6) 65.58</td>
</tr>
<tr>
<td></td>
<td>0.32 8(2/4) 0.32</td>
<td>2.44 8(2/5) 2.44</td>
<td>2.62 8(4/6) 3.06</td>
<td>63.87 8(3/7) 63.87</td>
</tr>
</tbody>
</table>

computing time increases only with a factor, approximately equal to 5, when $h^{-1}$ is doubled, which is close to the optimal value 4.

For the sake of comparisons, in Table 2 we include the performance of the block-diagonally preconditioned MINRES, as described in [15]. The stopping criteria are $10^{-6}$ for the outer MINRES and $10^{-3}$ for the blocks, solved by AGMG.

Table 2 confirms the robust behaviour of MINRES, documented in [15]. We see, however, that for small values of $\beta$, which must be used to obtain a solution sufficiently close to the desired solution, our method outperforms MINRES.

To test the accuracy of the solution the resulting approximate solution was compared with the solution of an implicit time-stepping method, the midpoint trapezoidal (or the implicit midpoint) method, briefly recalled for completeness. Consider the ordinary differential equation $x'(t) = f(t, x(t))$ with initial condition $x(0) = x_0$. The
midpoint trapezoidal methods reads as follows:

\[ x_{k+1} = x_k + \delta f \left( \frac{t_k + t_{k+1}}{2}, \frac{x_k + x_{k+1}}{2} \right), k \geq 0. \]

The method is implicit, second order accurate and unconditionally stable for linear problems, as the one in hand. Consider the equation in (1). After space discretization and applying the above time-stepping scheme, the problem becomes as follows:

\[ \text{For } k=0,1,\ldots,m, \]
\[ \text{Solve } (M + \frac{1}{2} \Delta_t L)u_{k+1} = (M - \frac{1}{2} \Delta_t L)u_k + \Delta_t v_{\text{opt}, k+1/2}. \]

Here, \( \Delta_t \) is the timestep, \( m = T/\Delta_t \) and \( v_{\text{opt}} \) is the optimal control, obtained from the optimization problem, evaluated at the required time. For the solution of systems with \( M + \frac{1}{2} \delta L \) we use the AGMG-preconditioned conjugate gradient method with
a relative stopping criterion $10^{-3}$. We compare the solutions at the final time $T$. The results are shown in Table 3.

**Table 3.** Problem 1: Performance of the midpoint trapezoidal method ($N=5$)

<table>
<thead>
<tr>
<th>$\beta$ = $10^{-2}$</th>
<th>$\beta$ = $10^{-4}$</th>
<th>$\beta$ = $10^{-6}$</th>
<th>$\beta$ = $10^{-8}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta t$</td>
<td>IT</td>
<td>CPU</td>
<td>IT</td>
</tr>
<tr>
<td>$h = \frac{1}{64}$</td>
<td>0.1</td>
<td>0.06</td>
<td>6</td>
</tr>
<tr>
<td>$h = \frac{1}{128}$</td>
<td>0.05</td>
<td>0.19</td>
<td>5</td>
</tr>
<tr>
<td>$h = \frac{1}{256}$</td>
<td>0.025</td>
<td>0.35</td>
<td>5</td>
</tr>
<tr>
<td>$h = \frac{1}{512}$</td>
<td>0.1</td>
<td>0.41</td>
<td>6</td>
</tr>
<tr>
<td>$h = \frac{1}{1024}$</td>
<td>0.05</td>
<td>0.68</td>
<td>6</td>
</tr>
<tr>
<td>$h = \frac{1}{2048}$</td>
<td>0.025</td>
<td>1.18</td>
<td>6</td>
</tr>
<tr>
<td>$h = \frac{1}{4096}$</td>
<td>0.1</td>
<td>1.42</td>
<td>7</td>
</tr>
<tr>
<td>$h = \frac{1}{8192}$</td>
<td>0.05</td>
<td>2.74</td>
<td>7</td>
</tr>
<tr>
<td>$h = \frac{1}{16384}$</td>
<td>0.025</td>
<td>4.70</td>
<td>6</td>
</tr>
</tbody>
</table>

The difference in the solution obtained within the optimal control and the solution obtained via the time stepping scheme is illustrated in Table 4. The difference is computed $\frac{\|u_{opt} - u_k\|_2}{\|u_{opt}\|_2}$ with $u_{opt}$ being the state computed by the optimal control method and $u_t$ computed by the time-stepping method.

**Table 4.** Problem 1: Difference between the optimal control solution $u_{opt}$ and the solution from the time stepping method $u_k$ for $t_k = T$

<table>
<thead>
<tr>
<th>$\beta$ = $10^{-2}$</th>
<th>$\beta$ = $10^{-4}$</th>
<th>$\beta$ = $10^{-6}$</th>
<th>$\beta$ = $10^{-8}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta t$</td>
<td>$N = 5$</td>
<td>$N = 10$</td>
<td>$N = 5$</td>
</tr>
<tr>
<td>$h = \frac{1}{64}$</td>
<td>0.1</td>
<td>0.45</td>
<td>0.50</td>
</tr>
<tr>
<td>$h = \frac{1}{128}$</td>
<td>0.05</td>
<td>0.12</td>
<td>0.15</td>
</tr>
<tr>
<td>$h = \frac{1}{256}$</td>
<td>0.025</td>
<td>0.03</td>
<td>0.04</td>
</tr>
<tr>
<td>$h = \frac{1}{512}$</td>
<td>0.1</td>
<td>0.45</td>
<td>0.50</td>
</tr>
<tr>
<td>$h = \frac{1}{1024}$</td>
<td>0.05</td>
<td>0.12</td>
<td>0.15</td>
</tr>
<tr>
<td>$h = \frac{1}{2048}$</td>
<td>0.025</td>
<td>0.03</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Figure 2 illustrates the obtained optimal control and state vectors, as well as the form of the desired state, approximated as truncated Fourier expansion with $N = 5$. 


Figure 2. Problem 1: Surface plots of the desired state $u_d$, the computed state $u$ and the control $v$ on a $64 \times 64$ grid and $\beta = 10^{-6}$.

Figure 3 visualizes the solution from the time-stepping procedure for different timesteps.

7.2 Eddy current electromagnetic problem

The second test problem involves several regularization and problem parameters. The aim here is to demonstrate the robustness of our method with respect to all parameters.

We consider then the linear eddy current case of Maxwell’s equations in a bounded domain $\Omega$ with Lipschitz boundary $\Gamma$. The problem can be formulated finding a time-dependent magnetic vector potential $y$ such that
Problem 2.

\[ \begin{align*}
\sigma \frac{\partial y}{\partial t} + \text{curl}(\nu \text{curl} y) &= j & \text{in } \Omega \times (0,T), \\
y \times n &= 0 & \text{on } \partial \Omega \times (0,T), \\
y &= y_0, & \text{on } \partial \Omega \times \{0\},
\end{align*} \]

where \( \sigma, \nu \) and \( j \) denote the electrical conductivity, magnetic reluctivity, and the external current density, respectively, and where \( n \) is the outward normal vector to \( \partial \Omega \). Note that the conductivity can be zero, such as in air.

Due to the discontinuity of \( \sigma \) and to obtain uniqueness in the nonconducting regions, the state equation must be regularized unless the solution is divergence free and a classical inf–sup stability relation holds. We do this here by adding a positive
term $\varepsilon y, \varepsilon > 0$ to the state equation. The regularized problem takes then the form,

$$
\begin{aligned}
\begin{cases}
\sigma \frac{\partial u}{\partial t} + \text{curl}(\nu \text{curl} y) + \varepsilon y = u & \text{in } \Omega \times (0,T) \\
y \times n = 0 & \text{on } \partial \Omega \times (0,T) \\
y = y_0, & \text{in } \Omega \times \{0\}.
\end{cases}
\end{aligned}
$$

In the time-harmonic regime with angular frequency $\omega$, $y(x,t) = \text{Re}\{\hat{y}(x)e^{i\omega t}\}$, $j(x,t) = \text{Re}\{\hat{j}(x)e^{i\omega t}\}$ it leads to finding the complex-valued amplitude $\hat{y}$ satisfying

$$
\begin{aligned}
\begin{cases}
i\omega\sigma\hat{y} + \text{curl}(\nu \text{curl} y) + \varepsilon \hat{y} = \hat{j} & \text{in } \Omega, \\
\hat{y} \times n = 0 & \text{on } \partial \Omega.
\end{cases}
\end{aligned}
$$

The problem is formulated in sense of distributions to find $\hat{y} \in H_0(\text{curl}, \Omega)$:

$$
i\omega \int_\Omega \sigma(x) \hat{y}(x) \cdot v(x) dx + \int_\Omega (\nu(x) \text{curl} \hat{y}(x) \cdot \text{curl} v(x) + \varepsilon \hat{y}(x) \cdot v(x)) dx = \int_\Omega \hat{j}(x) \cdot v(x) dx
$$

for all complex-valued test functions $v \in H_0(\text{curl}, \Omega)$, where $H_0(\text{curl}, \Omega) := \{v \in L^2(\Omega)^3 : \text{curl} v \in L^2(\Omega)^3, v \times n = 0 \text{ on } \Gamma\}$. Assuming $\hat{j} \in L^2(\Omega)^3$ the linear form is bounded. Assuming further $\sigma, \nu \in L^\infty(\Omega), \sigma(x) \geq 0$, and $\nu(x) \geq \nu_0 > 0$ the bilinear form is bounded and elliptic, therefore, the problem is uniquely solvable and the solution depends continuously on the data. Note that the problem can be equivalently viewed as a 2-by-2 real elliptic system solved for the cosine and sine parts, $\hat{y}(x) = y^e(x) + iy^s(x)$.

Conforming finite element approximation of functions $v \in H_0(\text{curl}, \Omega)$ requires continuity of the traces $v \times n$. In [27] and [28] two classes of such elements were proposed. Here we use the lowest-order finite elements of the former class, which is referred to as Nédélec-I elements. On a tetrahedral mesh the finite element functions takes the local form $v(x) = a \times x + b, a, b \in \mathbb{R}^3$. In order to preserve global continuity of the tangential components the degrees of freedom are tangential moments along edges. We arrive at the linear system of equations

$$(i\omega M + K)z = b, \quad (20)$$

where $M_{ij} := \int_\Omega \sigma \varphi_j \cdot \varphi_i, \quad K_{ij} := \int_\Omega \nu \text{curl} \varphi_j \cdot \text{curl} \varphi_i + \varepsilon \varphi_j \cdot \varphi_i, \quad b_i := \int_\Omega \hat{j} \cdot \varphi_i$ with $\varphi_i(x), i,j = 1,...,n$, being the Nédélec-I basis functions. To avoid complex arithmetics we can rewrite (20) in real-valued block matrix form

$$
\begin{bmatrix}
K & -\omega M \\
\omega M & K
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
= 
\begin{bmatrix}
\xi \\
\eta
\end{bmatrix},
$$

where $z = x + iy$ and $b = \xi + i\eta$. Such types of block matrices arise also in some optimal control problems for PDEs, including time-harmonic problems.

For the time-harmonic problem the aim is to compute a periodic steady state solution $(y,v)$ that satisfies (19) but not necessarily the initial condition $y = y_0$. Including instead the periodicity condition, $y(0) = y(2T)$, the state equation takes the form

$$
\begin{aligned}
\begin{cases}
\sigma \frac{\partial y}{\partial t} + \text{curl}(\nu \text{curl} y) + \varepsilon y = v & \text{in } \Omega \times (0,T) \\
y \times n = 0 & \text{on } \partial \Omega \times (0,T) \\
y(0) = y(2T) & \text{in } \Omega.
\end{cases}
\end{aligned}
$$
We denote the adjoint variable with \( w \). Similarly, the condition \( w(2T) = 0 \) is replaced by the periodicity condition, \( w(0) = w(2T) \). We consider then a time-harmonic desired state,

\[
y_d(x,t) = y_d^c(x) \cos(\omega t) + y_d^s(x) \sin(\omega t).
\]

Due to the linearity of the problem, the state \( y \), the Lagrange multiplier, i.e. co-state \( w \) and the control \( u \) are time-harmonic as well with the same frequency \( \omega \),

\[
y(x,t) = y^c(x) \cos(\omega t) + y^s(x) \sin(\omega t)
\]

\[
u(x,t) = u^c(x) \cos(\omega t) + u^s(x) \sin(\omega t)
\]

\[
w(x,t) = w^c(x) \cos(\omega t) + w^s(x) \sin(\omega t).
\]

The Fourier coefficients \( u^c(x) \) and \( u^s(x) \) are then related to the corresponding coefficients \( w^c(x) \), \( w^s(x) \) as before. Using the above time-harmonic representation of the solution and the replacement of the initial and end conditions with the periodicity conditions, as is shown in [29, 8], the optimality system can be written

\[
\begin{align*}
\omega \sigma y^s + \text{curl}(\nu \text{curl} y^c) + \varepsilon y^s - \beta^{-1} w^c &= 0 \quad \text{in } \Omega \\
-\omega \sigma y^c + \text{curl}(\nu \text{curl} y^s) + \varepsilon y^c - \beta^{-1} w^s &= 0 \quad \text{in } \Omega \\
-\omega \sigma w^s + \text{curl}(\nu \text{curl} w^c) + \varepsilon w^s + y^c &= y_d^c \quad \text{in } \Omega \\
\omega \sigma w^c + \text{curl}(\nu \text{curl} w^s) + \varepsilon w^c + y^s &= y_d^s \quad \text{in } \Omega \\
\omega \sigma w^c + \varepsilon y^c + w^c &= 0 \quad \text{on } \partial \Omega \\
y^c \times n = 0 \quad \text{on } \partial \Omega \\
w^s \times n = 0 \quad \text{on } \partial \Omega
\end{align*}
\]

We consider the case where the control \( u \) is prescribed only on a subset, such as an electric coil, i.e. not distributed on the whole domain \( \Omega \). It vanishes then outside this subregion. We assume that the observation region is also restricted to this subdomain, \( \Omega_d \), which is defined by a characteristic function,

\[
\tau(x) = \begin{cases} 
1, & x \in \Omega_d \\
0, & x \in \Omega \setminus \Omega_d
\end{cases}
\]

The corresponding cost functional is then

\[
J(y,u) = \frac{1}{2} \int_{\Omega \times (0,T)} \tau(x)|y - y_d|^2 \, dx \, dt + \frac{1}{2} \beta \int_{\Omega \times (0,T)} \tau(x)|u|^2 \, dx \, dt
\]

and the optimization problem is subject to the state equation,

\[
\begin{cases}
\sigma \frac{\partial y}{\partial t} + \text{curl}(\nu \text{curl } y) = \tau(x)u & \text{in } \Omega \times (0,T) \\
y \times n = 0 & \text{on } \partial \Omega \times (0,T) \\
y(0) = y(T) & \text{in } \Omega
\end{cases}
\]
After a similar transformation with \( \sqrt{\beta} \) as has been done previously, the corresponding finite element matrix takes now the form,

\[
\mathbf{A} = \begin{bmatrix}
M_0 & 0 & K & -M_\omega \\
0 & M_0 & M_\omega & K \\
K & M_\omega & -M_0 & 0 \\
-M_\omega & K & 0 & -M_0
\end{bmatrix}
\]

where \( M_0 \) is the mass matrix corresponding to the observation subdomain \( \Omega_d \) and has zero entries at nodepoints in \( \Omega\setminus\Omega_d \). For ease of presentation from now on we denote \( \tilde{K} \) and \( \tilde{M}_\omega \) by \( K \) and \( M_\omega \), respectively.

Following the previously presented general approach, the first step in constructing a preconditioner is to add

\[
\begin{bmatrix}
K & -M_\omega \\
M_\omega & K
\end{bmatrix} + \begin{bmatrix}
K & M_\omega \\
-M_\omega & K
\end{bmatrix} = 2 \begin{bmatrix}
K & 0 \\
0 & K
\end{bmatrix}
\]

to the primal two-by-two block diagonal matrix, which gives

\[
\mathbf{C}_0 = \begin{bmatrix}
M_0 + 2K & 0 & K & -M_\omega \\
0 & M_0 + 2K & M_\omega & K \\
K & M_\omega & -M_0 & 0 \\
-M_\omega & K & 0 & -M_0
\end{bmatrix}, \tag{21}
\]

In a similar way as has been done for two-by-two block matrices with square blocks in Section 4, we transform \( \mathbf{C}_0 \) to \( \tilde{\mathbf{C}}_0 = \mathbf{L}\mathbf{C}_0\mathbf{L} \) where

\[
\mathbf{L} = \begin{bmatrix}
I & 0 & 0 & 0 \\
0 & I & 0 & 0 \\
-I & 0 & I & 0 \\
0 & -I & 0 & I
\end{bmatrix},
\]

which gives

\[
\tilde{\mathbf{C}}_0 = \mathbf{L} \begin{bmatrix}
M_0 + K & M_\omega & K & -M_\omega \\
-M_\omega & M_0 + K & M_\omega & K \\
K + M_0 & M_\omega & -M_0 & 0 \\
-M_\omega & K + M_0 & 0 & -M_0
\end{bmatrix} \mathbf{L} =
\begin{bmatrix}
M_0 + K & M_\omega & K & -M_\omega \\
-M_\omega & M_0 + K & M_\omega & K \\
0 & 0 & -(M_0 + K) & M_\omega \\
0 & 0 & -M_\omega & -(M_0 + K)
\end{bmatrix}, \tag{22}
\]

i.e. a block triangular form.

Here we can replace the two block diagonal matrices

\[
\begin{bmatrix}
M_0 + K & M_\omega \\
-M_\omega & M_0 + K
\end{bmatrix} \quad \text{with} \quad \begin{bmatrix}
M_0 + K + 2M_\omega & M_\omega \\
-M_\omega & M_0 + K
\end{bmatrix}
\]
\[
\begin{bmatrix}
-(M_0 + K) & M_\omega \\
-M_\omega & -(M_0 + K)
\end{bmatrix} \quad \text{with} \quad \begin{bmatrix}
-(M_0 + K) & M_\omega \\
-M_\omega & -(M_0 + K + 2M_\omega)
\end{bmatrix},
\]
respectively.

We solve the systems with $\tilde{C}_0$ in (22) using a coupled inner-outer iteration method. Then the arising systems with the block diagonal matrix $\begin{bmatrix} M_0 + K & M_\omega \\ -M_\omega & M_0 + K \end{bmatrix}$ are solved with the preconditioner

\[
A_0 := \begin{bmatrix} M_0 + K + 2M_\omega & M_\omega \\ -M_\omega & M_0 + K \end{bmatrix} = \begin{bmatrix} D_0 + M_\omega & M_\omega \\ -M_\omega & M_0 + K \end{bmatrix}.
\]

Following the previously used approach, to solve an equation $A_0 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$ we rewrite the equations as

\[
\begin{bmatrix} D_0 & M_\omega \\ 0 & D_0 \end{bmatrix} \begin{bmatrix} x \\ x+y \end{bmatrix} = \begin{bmatrix} f \\ f+g \end{bmatrix},
\]
which involves two solutions with the elliptic matrix $D_0 = M_0 + K + M_\omega$. The arising systems with the other block diagonal matrix in $\tilde{C}_0$ i.e. with $\begin{bmatrix} -(M_0 + K) & M_\omega \\ -M_\omega & -(M_0 + K) \end{bmatrix}$, are solved in a similar way. Even though one now gets coupled inner-outer iterations, which multiply up, this is a viable approach since, as follows from Proposition 1, the arising condition numbers for both the outer and inner iterations are bounded by a not large number $1 + \alpha$, $0 < \alpha \leq 1$, for different values of $\alpha$. Hence there will be few iterations. Furthermore, as is seen by the numerical tests, in practice it suffices to solve the inner systems to a fairly rough relative accuracy, say $10^{-2}$ to get the smallest or nearly the smallest number of outer iterations, so that there will be very few iterations. This is clearly demonstrated in the next section.

To test the robustness of the method, we report a result form [8], where different values of the reluctivity and two meshsizes are reported, see Table 5. Here, as inner iteration method, a geometric multigrid method is used. It can be seen that there is a very minor variation of outer iterations for all parameter values. Also the number of inner iterations per outer iteration is very stable, bounded by 4 for the rough $(10^{-2})$ relative inner precision. It is somewhat larger for the more strict inner iteration relative precision $(10^{-6})$, but there is a no need to use such a strict precision. There are hardly any variations of iterations for different values of the mesh parameter $h$.

Table 6 illustrates the robustness of the method with respect to variation of the angular frequency $\omega$. It is noticed that there is some increase of the number of inner iterations for values of $\omega = 1$ up to $\omega = 10^4$. In this case also the behaviour of the geometric multigrid is more sensitive to the value of $\omega$. A comparison with AMG has not been done but would be of interest.

In Table 7 we present the results where nonzero conductivity holds only in a subdomain, $\Omega_2 = (0,1)^2 \times (0,1/2)$ of the given domain $(0,1)^3$. Further, the control and observation domain equals $\Omega_d = (1/4, 3/4)^3$.
Table 5. Problem 2: Eddy current optimal control, 4-by-4 system: Robustness of outer and total inner (in brackets) FGMRES iterations with respect to $\beta$, $\nu$, and $h$, while fixing $\omega=\sigma_2=1$ and outer rel. prec. $10^{-8}$

<table>
<thead>
<tr>
<th>$h$</th>
<th>$\beta$</th>
<th>inner rel. prec. $10^{-2}$</th>
<th>inner rel. prec. $10^{-6}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\nu$ $10^8$</td>
<td>$\nu$ $10^4$</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>2(4)</td>
<td>3(6) 11(22) 4(8) 3(6)</td>
<td>2(6) 3(6) 11(33) 4(12) 3(9)</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>2(4)</td>
<td>4(8) 11(22) 3(6) 3(6)</td>
<td>2(8) 4(16) 11(36) 3(11) 3(11)</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>3(6)</td>
<td>6(12) 9(18) 4(8) 4(8)</td>
<td>2(8) 6(30) 9(45) 2(8) 3(12)</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>2(8)</td>
<td>10(40) 6(23) 4(13) 5(17)</td>
<td>2(8) 10(87) 5(30) 2(8) 3(18)</td>
</tr>
<tr>
<td>$10^0$</td>
<td>2(8)</td>
<td>10(57) 5(19) 3(12) 5(20)</td>
<td>2(8) 9(125) 4(24) 3(12) 5(40)</td>
</tr>
</tbody>
</table>

Table 6. Problem 2: Eddy current optimal control on a subset, 4-by-4 system: Robustness of outer FGMRES iterations, total inner (first number in brackets) FGMRES iterations, and total inner-most (second number in brackets) multigrid-preconditioned PCG iterations with respect to $\beta$, $\omega$, and $h$, while fixing $\nu=\sigma_2=1$ and outer rel. prec. $10^{-8}$

<table>
<thead>
<tr>
<th>$h$</th>
<th>$\beta$</th>
<th>$\omega$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td>$1/32$</td>
<td>10^{-8}</td>
<td>8(18,44) 8(20,55) 10(29,109) 10(64,128) 3(8,16)</td>
</tr>
<tr>
<td></td>
<td>10^{-6}</td>
<td>10(21,74) 10(21,76) 11(37,141) 7(50,100) 3(8,16)</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>8(16,64) 8(16,64) 11(41,163) 6(42,84) 3(8,16)</td>
</tr>
<tr>
<td></td>
<td>10^{-2}</td>
<td>6(13,52) 6(13,52) 9(49,196) 7(47,94) 3(7,14)</td>
</tr>
<tr>
<td></td>
<td>$10^0$</td>
<td>6(15,60) 6(16,64) 7(42,168) 7(47,94) 3(7,14)</td>
</tr>
<tr>
<td>$1/64$</td>
<td>10^{-8}</td>
<td>9(18,64) 9(18,68) 10(27,108) 11(67,134) 4(12,24)</td>
</tr>
<tr>
<td></td>
<td>10^{-6}</td>
<td>8(17,68) 8(17,68) 11(35,140) 7(51,102) 4(12,24)</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>9(19,94) 9(19,94) 11(42,172) 6(44,88) 4(12,24)</td>
</tr>
<tr>
<td></td>
<td>10^{-2}</td>
<td>6(12,68) 6(12,68) 9(49,197) 6(47,94) 4(11,22)</td>
</tr>
<tr>
<td></td>
<td>$10^0$</td>
<td>6(12,68) 6(12,68) 8(52,230) 6(47,94) 4(11,22)</td>
</tr>
</tbody>
</table>

We notice also here the strong robustness of the method, now with respect to different frequencies, $\omega$. In this case, the behaviour of the geometric multigrid method was somewhat more sensitive, in particular to frequencies around unity. A comparison with AMG has not been done. As a general conclusion, the method shows a fully reliable behaviour with respect to all parameters and needs very few outer iterations.
Table 7. Problem 2: Eddy current optimal control, 4-by-4 system: Robustness of outer and total inner (in brackets) FGMRES iterations with respect to $\beta$, $\sigma_2$, and $h$, while fixing $\omega = \nu = 1$ and outer rel. prec. $10^{-8}$

<table>
<thead>
<tr>
<th>$h$</th>
<th>$\beta$</th>
<th>$\sigma_2$ inner rel. prec. $10^{-2}$</th>
<th>$\sigma_2$ inner rel. prec. $10^{-6}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/16</td>
<td>$10^{-8}$</td>
<td>11(22) 11(22) 11(22) 11(65) 6(23) 11(33) 11(33) 11(33) 11(175) 6(42) 11(43) 11(43) 11(36) 10(161) 5(34)</td>
<td>9(45) 9(45) 9(45) 9(160) 4(28) 9(54) 9(54) 9(54) 9(160) 4(28)</td>
</tr>
<tr>
<td></td>
<td>$10^{-6}$</td>
<td>11(22) 11(22) 11(22) 12(75) 6(23) 11(43) 11(43) 11(36) 10(161) 5(34)</td>
<td>6(54) 6(54) 5(30) 6(112) 3(21) 6(54) 6(54) 5(30) 6(112) 3(21)</td>
</tr>
<tr>
<td></td>
<td>$10^{-4}$</td>
<td>10(21) 10(21) 9(18) 10(80) 5(19) 9(45) 9(45) 9(45) 9(160) 4(28)</td>
<td>4(51) 4(51) 4(24) 4(69) 2(13) 4(51) 4(51) 4(24) 4(69) 2(13)</td>
</tr>
<tr>
<td></td>
<td>$10^{-2}$</td>
<td>7(24) 7(24) 6(23) 7(56) 3(12) 6(54) 6(54) 5(30) 6(112) 3(21)</td>
<td>4(51) 4(51) 4(24) 4(69) 2(13) 4(51) 4(51) 4(24) 4(69) 2(13)</td>
</tr>
<tr>
<td></td>
<td>$10^0$</td>
<td>7(32) 7(32) 5(19) 6(46) 2(8) 4(51) 4(51) 4(24) 4(69) 2(13)</td>
<td>4(51) 4(51) 4(24) 4(69) 2(13) 4(51) 4(51) 4(24) 4(69) 2(13)</td>
</tr>
</tbody>
</table>

Acknowledgements

The work of the first author is supported by the projects LD15105 Ultrascale computing in geosciences and LQ1602 IT4Innovations excellence in science supported by the Ministry of Education, Youth and Sports of the Czech Republic. The work of the third author is funded by the China Scholarship Council (CSC) and his work is performed during his visit at Uppsala University, Sweden.

References
