An exact Schur complement method for time-harmonic optimal control problems*

Owe Axelsson, Dalibor Lukáš, Maya Neytcheva

The Czech Academy of Sciences, Institute of Geonics,
Ostrava, Czech Republic, Department of Information Technology, Uppsala University,
VSB - University of Technology, Ostrava, Czech Republic
Department of Information Technology, Uppsala University, Uppsala, Sweden
owe.axelsson@it.uu.se, dalibor.lukas@vsb.cz, maya.neytcheva@it.uu.se

Abstract. By use of Fourier time series expansions in an angular frequency variable, time-harmonic optimal control problems constrained by a linear differential equation decouples for the different frequencies. Hence, for the analysis of a solution method one can consider the frequency as a parameter. There are three variables to be determined, the state solution, the control variable and the adjoint variable.
The first order optimality conditions lead to a three-by-three block matrix system where the adjoint optimality variable can be eliminated. For the so arising two-by-two block system, in this paper we study a factorization method involving an exact Schur complement method and illustrate the performance of an inexact version of it.

Keywords: PDE-constrained optimal control problems·distributed control ·preconditioning·time-harmonic Maxwell equations

1 Introduction

In an optimal control problem for time-dependent differential equations, see e.g. [1], we seek the solution of the state variable and the control variable that minimizes the functional,

$$\mathcal{J}(u,v)=\frac{1}{2}\int\limits_{\Omega\times[0,T]}\|u-u_d\|^2dxdt+\frac{1}{2}\beta\int\limits_{\Omega\times[0,T]}\|v\|^2dxdt,$$

subject a differential equation $\mathcal{L}u=g$, defined in a bounded domain $\Omega\subset\mathbb{R}^m$, $m=2,3$ and time interval $[0,T]$. The control variable acts as an additional source function to the differential equation and $\beta>0$ is a regularization parameter that determines the control cost. Further u_d is a given target solution. Hence, the corresponding Lagrange optimality functional incorporating the adjoint variable w has the form

$$\mathcal{L}(u,v,w)=\mathcal{J}(u,v)+\int\limits_{\Omega\times[0,T]}(\mathfrak{L}u(\mathbf{x},t)-\mathbf{g}-v)w(\mathbf{x},t)d\mathbf{x}dt. \tag{1}$$

* Supported by EURAD project, European Joint Programme (EJP) Cofund Action No. 847593, VR Grant 2017-03749 *Mathematics and numerics in PDE-constrained optimization problems with state and control constraints*, 2018-2022.

We assume periodic boundary conditions $u(\mathbf{x},0)=u(\mathbf{x},T)$ and consider time-harmonic solutions $u(\mathbf{x},t)=\widetilde{u}(\mathbf{x})e^{i\omega t/T}$ for a given frequency ω, being a multiple of 2π.

Using appropriate finite elements, after space discretization and using the time-harmonic setting, we can formulate the first order necessary optimality conditions. The so arising system is of a saddle point form with a three-by-three block matrix as follows,

$$\begin{bmatrix} A & 0 & B-i\omega A \\ 0 & \beta A & -A \\ B+i\omega A & -A & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} A\mathbf{u}_d \\ \mathbf{0} \\ \mathbf{g} \end{bmatrix}, \tag{2}$$

where A is typically a mass matrix.

When the heat equation is the state constraint, see e.g. [2, 3], then u and v are scalar functions and A and B are matrices of size n, equal to the number of space degrees of freedom. For the case of eddy current curl-curl (see e.g. [4]) equations, u and v are vector variables with two or three components per space discretization point and matrices A and B are block matrices of size $2n$ or $3n$.

We assume that A is symmetric positive definite (spd), B is real and symmetric positive semi-definite (spsd). After elimination of the adjoint variable $\mathbf{w}$ we obtain the system

$$\begin{bmatrix} A & \beta C^* \\ C & -A \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} A\mathbf{u}_d \\ \mathbf{g} \end{bmatrix},$$

with $C=B+i\omega A$, $C^*=B^*-i\omega A$. We scale the system and introduce $\widetilde{\mathbf{v}}=-\sqrt{\beta}\mathbf{v}$ to get

$$\mathcal{A}\begin{bmatrix} \mathbf{u} \\ \widetilde{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} A & -\widetilde{C}^* \\ \widetilde{C} & A \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \widetilde{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} A\mathbf{u}_d \\ \widetilde{\mathbf{g}} \end{bmatrix}, \tag{3}$$

where $\widetilde{C}=\sqrt{\beta}C=\widetilde{B}+i\widetilde{\omega}A$, $\widetilde{B}=\sqrt{\beta}B$, $\widetilde{\omega}=\sqrt{\beta}\omega$ and $\widetilde{\mathbf{g}}=\sqrt{\beta}\mathbf{g}$.

We focus on the solution of (3) via its Schur complement of $\mathcal{A}$, that is,

$$S=A+\widetilde{C}A^{-1}\widetilde{C}^*=(1+\widetilde{\omega}^2)A+\widetilde{B}A^{-1}\widetilde{B}$$

is spd, like A, thus $\mathcal{A}$ is regular. We assume that the problem sizes are large, which motivates the use of preconditioned iterative methods of conjugate gradient type.

As we shall see, the method involves solving two complex valued systems. For this a complex-to-real (C-to-R) method, see e.g. [5], can be used, see next section.

There are two other methods with a similar performance. One is based on the PRESB method (see, e.g. [6]) and one is based on an approximate factorization of the Schur complement in real-valued factors, see [4, 7]. The PRESB method involves solving two complex valued systems at each iteration step. These methods are based on preconditioned outer iteration methods of Krylov subspace type which should be flexible, because the inner iterations are not solved exactly, e.g. FGMRES, cf. [8] and GCG, cf. [9].

In practice, the systems in the exact Schur complement method are not solved exactly. Hence, this method needs also an outer iteration method, which however can be a simple defect-correction method.

The method has been dealt with in [10] but with another way of derivation. A numerical comparison of the three methods can be found in [13].

In the next section the method is presented. A method to solve the arising complex valued systems is presented in Section 3. In Section 4 two related methods are shortly presented and Section 5 shows numerical tests.

2 An exact Schur complement method

The system in (2) can be factorized as

$$\mathcal{A}=\begin{bmatrix} A & 0 \\ \widetilde{B}+i\widetilde{\omega}A & H_1 \end{bmatrix}\begin{bmatrix} A^{-1} & 0 \\ 0 & A^{-1} \end{bmatrix}\begin{bmatrix} A & -(\widetilde{B}-i\widetilde{\omega}A) \\ 0 & H_2 \end{bmatrix},$$

where $H_1A^{-1}H_2 = S = (1+\widetilde{\omega}^2)A + \widetilde{B}A^{-1}\widetilde{B}$, i.e., $H_1 = \sqrt{1+\widetilde{\omega}^2}A - i\widetilde{B}$, $H_2 = \sqrt{1+\widetilde{\omega}^2}A+i\widetilde{B}=H_1^*$ are factors in the complex-valued exact factorization of the Schur complement matrix S of $\mathcal{A}$. Then

$$\mathcal{A}^{-1}=\begin{bmatrix} A^{-1} & A^{-1}(\widetilde{B}-i\widetilde{\omega}A)H_2^{-1} \\ 0 & H_2^{-1} \end{bmatrix}\begin{bmatrix} I & 0 \\ -AH_1^{-1}(\widetilde{B}+i\widetilde{\omega}A)A^{-1} & AH_1^{-1} \end{bmatrix}. \tag{4}$$

The computation of a solution of the system $\mathcal{A}\begin{bmatrix}\mathbf{u}\\ \widetilde{\mathbf{v}}\end{bmatrix} = \begin{bmatrix}\mathbf{f}\\ \widetilde{\mathbf{g}}\end{bmatrix}$ can be done by direct use of (4) but, besides solutions with the matrices H_1 and H_2, it would also involve two solutions with A. We show now that the latter solutions with A can be avoided. Consider first the matrix-vector multiplication arising from the second factor in (4):

$$\begin{bmatrix} I & 0 \\ -AH_1^{-1}(\widetilde{B}+i\widetilde{\omega}A)A^{-1} & AH_1^{-1} \end{bmatrix}\begin{bmatrix}\mathbf{f}\\ \widetilde{\mathbf{g}}\end{bmatrix}=\begin{bmatrix} \mathbf{f} \\ AH_1^{-1}(\widetilde{\mathbf{g}}-(\widetilde{B}+i\widetilde{\omega}A)A^{-1}\mathbf{f}) \end{bmatrix}.$$

Since $H_1=\sqrt{1+\widetilde{\omega}^2}A-i\widetilde{B}$ it follows that

$$\begin{aligned} -AH_1^{-1}(\widetilde{B}+i\widetilde{\omega}A)A^{-1} &= -iAH_1^{-1}(-i\widetilde{B}+\sqrt{1+\widetilde{\omega}^2}A-(\sqrt{1+\widetilde{\omega}^2}-\widetilde{\omega})A)A^{-1} \\ &= -i(I-b_{\widetilde{\omega}}AH_1^{-1}), \end{aligned}$$

where $b_{\widetilde{\omega}}=\sqrt{1+\widetilde{\omega}^2}-\widetilde{\omega}=\dfrac{1}{\sqrt{1+\widetilde{\omega}^2}+\widetilde{\omega}}\le 1$. Hence,

$$\begin{bmatrix} I & 0 \\ -AH_1^{-1}(\widetilde{B}+i\widetilde{\omega}A)A^{-1} & AH_1^{-1} \end{bmatrix}\begin{bmatrix}\mathbf{f}\\ \widetilde{\mathbf{g}}\end{bmatrix}=\begin{bmatrix} \mathbf{f} \\ -i\mathbf{f}+ib_{\widetilde{\omega}}AH_1^{-1}\mathbf{f}+AH_1^{-1}\widetilde{\mathbf{g}} \end{bmatrix}=\begin{bmatrix} \mathbf{f} \\ -i(\mathbf{f}-A\mathbf{h}), \end{bmatrix} \tag{5}$$

where $\mathbf{h}=H_1^{-1}(b_{\widetilde{\omega}}\mathbf{f}-i\widetilde{\mathbf{g}})$. Further, it follows from (4) that $\widetilde{\mathbf{v}}=-iH_2^{-1}(\mathbf{f}-A\mathbf{h})$. To find the component $\mathbf{u}$, note now that

$$A^{-1}(\widetilde{B}-i\widetilde{\omega}A)H_2^{-1}=iA^{-1}(-i\widetilde{B}-\sqrt{1+\widetilde{\omega}^2}A+b_{\widetilde{\omega}}A)H_2^{-1}=-iA^{-1}+ib_{\widetilde{\omega}}H_2^{-1}.$$

Hence, it follows from (4) and (5) that

$$\begin{aligned} \mathbf{u} &= \begin{bmatrix} A^{-1} & A^{-1}(\widetilde{B}-i\widetilde{\omega}A)H_2^{-1} \end{bmatrix}\begin{bmatrix} A\mathbf{h}+\mathbf{f}-A\mathbf{h} \\ -i(\mathbf{f}-A\mathbf{h}) \end{bmatrix} \\ &= \mathbf{h}+A^{-1}(\mathbf{f}-A\mathbf{h})-A^{-1}(\mathbf{f}-A\mathbf{h})+b_{\widetilde{\omega}}H_2^{-1}(\mathbf{f}-A\mathbf{h}), \end{aligned}$$

that is, $\mathbf{u}=\mathbf{h}+b_{\widetilde{\omega}}H_2^{-1}(\mathbf{f}-A\mathbf{h})=\mathbf{h}+ib_{\widetilde{\omega}}\widetilde{\mathbf{v}}$. The above relations show that, besides a matrix-vector multiplication with A and some complex vector additions, the computation of the vectors $\mathbf{u}$ and $\widetilde{\mathbf{v}}$ requires one solution with H_1 to compute the vector $\mathbf{h}$ and one solution with H_2 to compute $\widetilde{\mathbf{v}}$. Both systems involve complex matrices, which can

be solved by an iterative method using the method in Section 3. In Algorithm 1 we summarize the computations, needed to compute the action of $\mathcal{A}^{-1}$ on a vector $\left[\mathbf{f}^T,\ \widetilde{\mathbf{g}}^T\right]$.

Algorithm 1

1: Let $H_1=\sqrt{1+\widetilde{\omega}^2}A-i\widetilde{B}$, $b_{\widetilde{\omega}}=\frac{1}{\sqrt{1+\widetilde{\omega}^2}+\widetilde{\omega}}$, $H_2=H_1^*=\sqrt{1+\widetilde{\omega}^2}A+i\widetilde{B}$

2: Solve $H_1\mathbf{h}=b_{\widetilde{\omega}}\mathbf{f}-i\widetilde{\mathbf{g}}$

3: Solve $H_2\widetilde{\mathbf{v}}=-i(\mathbf{f}-A\mathbf{h})$

4: Compute $\mathbf{u}=\mathbf{h}+ib_{\widetilde{\omega}}\widetilde{\mathbf{v}}$

Clearly, if we solve exactly with H_1 and H_2, we have a direct solution method for $\mathcal{A}$. In practice they are solved to some limited tolerance. It can then be efficient to imbed the method in a defect-correction framework, applied at least one step.

Namely, after one step of Algorithm 1, where we compute an approximate solution $\begin{bmatrix}\mathbf{u}_0\\ \widetilde{\mathbf{v}}_0\end{bmatrix}$ of $\mathcal{A}\begin{bmatrix}\mathbf{u}\\ \widetilde{\mathbf{v}}\end{bmatrix}=\begin{bmatrix}\mathbf{f}\\ \widetilde{\mathbf{g}}\end{bmatrix}$, we solve

$$\mathcal{A}\left(\begin{bmatrix}\delta\mathbf{u}\\ \delta\widetilde{\mathbf{v}}\end{bmatrix}\right)=\begin{bmatrix}\mathbf{f}\\ \widetilde{\mathbf{g}}\end{bmatrix}-\mathcal{A}\begin{bmatrix}\mathbf{u}_0\\ \widetilde{\mathbf{v}}_0\end{bmatrix},$$

using again Algorithm 1. Here $\delta\mathbf{u}=\mathbf{u}-\mathbf{u}_0$ and $\delta\widetilde{\mathbf{v}}=\widetilde{\mathbf{v}}-\widetilde{\mathbf{v}}_0$. Depending on the size of the residual, it can be repeated once more.

3 A complex-to-real solution nethod

To avoid complex arithmetics in the solution with matrices H_2, $i=1,2$ in Algorithm 1 we can use a complex-to-real method, that is rewrite, say the system with H_2 as,

$$(\sqrt{1+\widetilde{\omega}^2}A+i\widetilde{B})(x+iy)=a+ib$$

in real valued form, where a,b are defined by Step 3 of the Algorithm 1. Hence,

$$\begin{bmatrix}\sqrt{1+\widetilde{\omega}^2}A & -\widetilde{B}\\ \widetilde{B} & \sqrt{1+\widetilde{\omega}^2}A\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}a\\ b\end{bmatrix}.$$

This can be solved efficiently by use the PRESB preconditioning method, see e.g. [5, 12], that is preconditioned by

$$\begin{bmatrix}\sqrt{1+\widetilde{\omega}^2}A & -\widetilde{B}\\ \widetilde{B} & \sqrt{1+\widetilde{\omega}^2}A+2\widetilde{B}\end{bmatrix}=\begin{bmatrix}I & -I\\ 0 & I\end{bmatrix}\begin{bmatrix}\sqrt{1+\widetilde{\omega}^2}A+\widetilde{B} & 0\\ \widetilde{B} & \sqrt{1+\widetilde{\omega}^2}A+\widetilde{B}\end{bmatrix}\begin{bmatrix}I & I\\ 0 & I\end{bmatrix} \tag{6}$$

which can be factorized as shown above. Hence, each iteration step involves two solutions with the real valued matrix,

$$\sqrt{1+\widetilde{\omega}^2}A+\widetilde{B}.$$

The rate of convergence of this method follows from its spectral eigenvalue analysis which we include here for completeness of the paper,

$$\mathcal{A}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}=\lambda\mathcal{P}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}\quad\text{that is,}\quad\begin{bmatrix}\widehat{A} & -\widetilde{B}\\ \widetilde{B} & \widehat{A}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}=\lambda\begin{bmatrix}\widehat{A} & -\widetilde{B}\\ \widetilde{B} & \widehat{A}+2\widetilde{B}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}$$

where $\widehat{A}=\sqrt{1+\widetilde{\omega}^2}A$.

Computation shows that

$$(1-\lambda)\begin{bmatrix}\widehat{A} & -\widetilde{B}\\ \widetilde{B} & \widehat{A}+2\widetilde{B}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}=\begin{bmatrix}0 & 0\\ 0 & 2\widetilde{B}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}.$$

Clearly, for $\mathbf{y}=\mathcal{N}(\widetilde{B})$, $\mathcal{N}(\widetilde{B})$ being the null space of $\widetilde{B}$, and any x we see that $\lambda=1$. For $\lambda\neq 1$ it holds that

$$(1-\lambda)\mathbf{y}^T(\widetilde{B}\widehat{A}^{-1}\widetilde{B}+\widehat{A}+2\widetilde{B})\mathbf{y}=2\mathbf{y}^T\widetilde{B}\mathbf{y}.$$

Transforming the latter equality by multiplying by $\widehat{A}^{-1/2}$ from left and right and denoting $\widehat{B}=\widehat{A}^{-1/2}\widetilde{B}\widehat{A}^{-1/2}$ we obtain

$$(1-\lambda)\widehat{\mathbf{y}}^T(\widehat{B}^2+I+2\widehat{B})\widehat{\mathbf{y}}=2\widehat{\mathbf{y}}^T\widehat{B}\widehat{\mathbf{y}},\quad \text{i.e.,}\quad 1-\lambda=\frac{2\widehat{\mathbf{y}}^T\widehat{B}\widehat{\mathbf{y}}}{\widehat{\mathbf{y}}^T(\widehat{B}+I)^2\widehat{\mathbf{y}}}\leq\frac{1}{2},$$

that is, $\frac{1}{2}\leq\lambda\leq 1$, where $\widehat{\mathbf{y}}=\widehat{A}^{-1/2}\mathbf{y}$. Hence, Krylov subspace methods, applied to $\mathcal{A}$, preconditioned by $\mathcal{P}$, converge rapidly and with a convergence speed that holds uniformly with respect to all parameters, including the discretization parameter. For small values of $\widetilde{\omega}$, when $\|\widehat{B}\|$ can become small, the eigenvalues cluster at unity. Note that $\widetilde{\omega}=\sqrt{\beta}\omega$, where β is normally small.

4 Two related methods

There are two related methods. One is based directly on the use of the PRESB method for equation (3), that in the preconditioner equals

$$\mathcal{P}=\begin{bmatrix}A & -\widetilde{C}^*\\ \widetilde{C} & A+\widetilde{C}+\widetilde{C}^*\end{bmatrix}=\begin{bmatrix}A & -\widetilde{C}^*\\ \widetilde{C} & A+\widetilde{B}+\widetilde{B}^T\end{bmatrix},$$

which can be factorized as in (6), so it involves solving inner systems with $A+\widetilde{C}$ and $A+\widetilde{C}^*$. Here

$$A+\widetilde{C}=A+\widetilde{B}+i\widetilde{\omega}A,\quad A+\widetilde{C}^*=A+\widetilde{B}-i\widetilde{\omega}A.$$

Thus, the systems are similar to the ones for H_1 and H_2 in Section 2. They are solved in a similar way as in Section 3. Hence, this method is based on coupled inner-outer iterations, but where the outer iterations should best also involve a flexible conjugate gradient acceleration method. To sum up, this method involves two uses of such acceleration methods, while for the exact Schur complement method one can use a simple defect-correction method for the outer iterations.

In the other related method an approximation of S in real-valued factors, $G_1A^{-1}G_2=(\sqrt{1+\widetilde{\omega}^2}A+\widetilde{B})A^{-1}(\sqrt{1+\widetilde{\omega}^2}A+\widetilde{B})=(1+\widetilde{\omega}^2)A+\widetilde{B}A^{-1}\widetilde{B}^T+\sqrt{1+\widetilde{\omega}^2}(\widetilde{B}+\widetilde{B}^T)$ is used. Here we assume that B is not neccessarily symmetric. The corresponding approximate factorization of $\mathcal{A}$ becomes

$$\mathcal{A}\approx\mathcal{B}=\begin{bmatrix}A & 0\\ C & G_1\end{bmatrix}\begin{bmatrix}A^{-1} & 0\\ 0 & A^{-1}\end{bmatrix}\begin{bmatrix}A & -C^*\\ 0 & G_2\end{bmatrix}.$$

Hence,

$$\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}=\mathcal{B}^{-1}\begin{bmatrix}\mathbf{f}\\ \widetilde{\mathbf{g}}\end{bmatrix}=\begin{bmatrix}A^{-1} & A^{-1}C^*G_2^{-1}\\ 0 & G_2^{-1}\end{bmatrix}\begin{bmatrix}I & 0\\ -AG_1^{-1}CA^{-1} & AG_1^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{f}\\ \widetilde{\mathbf{g}}\end{bmatrix}.$$

A computation shows that the corresponding Algorithm takes the following form.

Algorithm 2

1: Let $G_1=\sqrt{1+\widetilde{\omega}^2}A+\widetilde{B}$, $G_2=G_1^T=\sqrt{1+\widetilde{\omega}^2}A+\widetilde{B}^T$,$c_{\widetilde{\omega}}^*=\sqrt{1+\widetilde{\omega}^2}+i\widetilde{\omega}$

2: Solve $G_1\mathbf{h}=c_{\widetilde{\omega}}\mathbf{f}+\mathbf{g}$

3: Solve $G_2\widetilde{\mathbf{v}}=A\mathbf{h}-\mathbf{f}$

4: Compute $\mathbf{u}=\mathbf{h}-c_{\widetilde{\omega}}^*\widetilde{\mathbf{v}}$

Thus, in this method we solve two inner systems with real matrices, but the outer iteration may need more iterations. A spectral analysis shows that, denoting $\widehat{B}=A^{-1/2}\widetilde{B}A^{-1/2}$

$$\widehat{\alpha}=\frac{\mathbf{y}^T\sqrt{1+\widetilde{\omega}^2}(\widehat{B}+\widehat{B}^T)\mathbf{y}}{\mathbf{y}^T((1+\widetilde{\omega}^2)I+\widehat{B}\widehat{B}^T)\mathbf{y}},\quad \text{i.e.,}\quad 0\le\widehat{\alpha}\le 1$$

it follows that

$$\frac{1}{\lambda}-1\le\widehat{\alpha},\quad \text{that is,}\quad \frac{1}{2}\le\frac{1}{1+\widehat{\alpha}}\le\lambda\le 1.$$

The rate of convergence is similar to that of PRESB but here we solve two real valued inner systems instead of two complex valued, which corresponds to about half the cost.

5 Numerical illustrations

The performance of the preconditioning method from Algorithm 1 in its inexact version is illustrated numerically on the discrete optimal control problem from (1) with a constraint, given by the time-harmonic eddy current problem in $\Omega=[0,1]^3$

$$\mathfrak{L}u\equiv\sigma\frac{\partial u}{\partial t}+\nabla\times(\frac{1}{\mu}\nabla\times u)=\mathfrak{g}\quad \text{in}\,\Omega\times[0,T] \tag{7}$$

with proper boundary and initial conditions. Here u is the magnetic potential, the problem parameter μ is the permeability of the media and σ is the conductivity. In the numerical experiments it is assumed that $\sigma=1$.

To numerically solve (1) we follow the 'discretize-then-optimze' framework. The discretization is done using Nédélec finite elements (cf. e.g. [11]) on a regular tetrahedral mesh. We test Algorithm 1 with inexact solution of the systems H_1 and H_2 and one defect-correction step. The systems with matrices H_1 and H_2 are solved via their equivalent twice larger real formulations using PRESB-preconditioned Generalized Conjugate Gradient (GCR) method ([8]) and a (relative) stopping criterion 10^{-8}. The inner systems in PRESB are solved again via GCR, preconditioned by a V-cycle algebraic multigrid AGMG, see [14]. The (relative) stopping criterion for the inner AGMG solvers is chosen as 10^{-6} for the initial step and 10^{-3} for the defect-correction step.

All experiments are performed in Matlab R2019a on Lenovo ThinkPad T450s with 12 GB of memory.

The iteration counts of the test runs are shown in Table 1. Each column contains two groups of numbers of the form $X(Y)$, separated by /. Here X denotes the average

ω	β				
	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}
$nDOFs = 293$					
10^{-8}	5(9) / 3(12)	5(9) / 3(12)	5(9) / 4(12)	3(3) / 2(4)	2(3) / 1(4)
10^{-4}	5(4) / 3(5)	5(4) / 3(5)	5(4) / 4(5)	3(3) / 2(4)	2(3) / 1(4)
1	5(3) / 3(3)	5(3) / 3(3)	5(3) / 3(3)	3(3) / 2(4)	2(3) / 1(4)
10^4	3(3) / 1(4)	3(3) / 1(4)	3(3) / 2(4)	3(3) / 2(4)	2(3) / 1(4)
10^8	2(3) / 1(4)	2(3) / 1(4)	2(3) / 1(4)	2(3) / 1(4)	2(3) / 1(4)
$nDOFs = 2903$					
10^{-8}	4(19) / 4(19)	4(19) / 4(19)	4(19) / 4(20)	4(3) / 3(3)	1(4) / 1(4)
10^{-4}	4(7) / 4(8)	4(7) / 4(8)	5(7) / 4(8)	4(3) / 3(3)	1(4) / 1(4)
1	4(4) / 3(4)	4(4) / 3(4)	4(4) / 4(4)	4(3) / 3(3)	1(4) / 1(4)
10^4	4(4) / 3(4)	4(4) / 3(4)	4(4) / 3(3)	4(3) / 3(3)	1(4) / 1(4)
10^8	2(4) / 1(4)	2(4) / 1(4)	2(4) / 1(4)	2(4) / 2(4)	1(4) / 1(4)
$nDOFs = 25602$					
10^{-8}	4(44) / 4(43)	4(44) / 4(43)	4(44) / 4(44)	4(4) / 4(4)	1(5) / 1(4)
10^{-4}	5(16) / 4(16)	5(16) / 4(16)	5(16) / 4(16)	4(4) / 4(4)	1(5) / 1(4)
1	4(7) / 4(7)	4(7) / 4(7)	4(7) / 4(7)	4(4) / 4(4)	1(5) / 1(4)
10^4	4(4) / 3(4)	4(4) / 3(4)	4(4) / 3(4)	4(4) / 4(4)	1(5) / 1(4)
10^8	3(4) / 2(4)	3(4) / 2(4)	3(4) / 2(4)	3(4) / 2(4)	1(5) / 1(4)

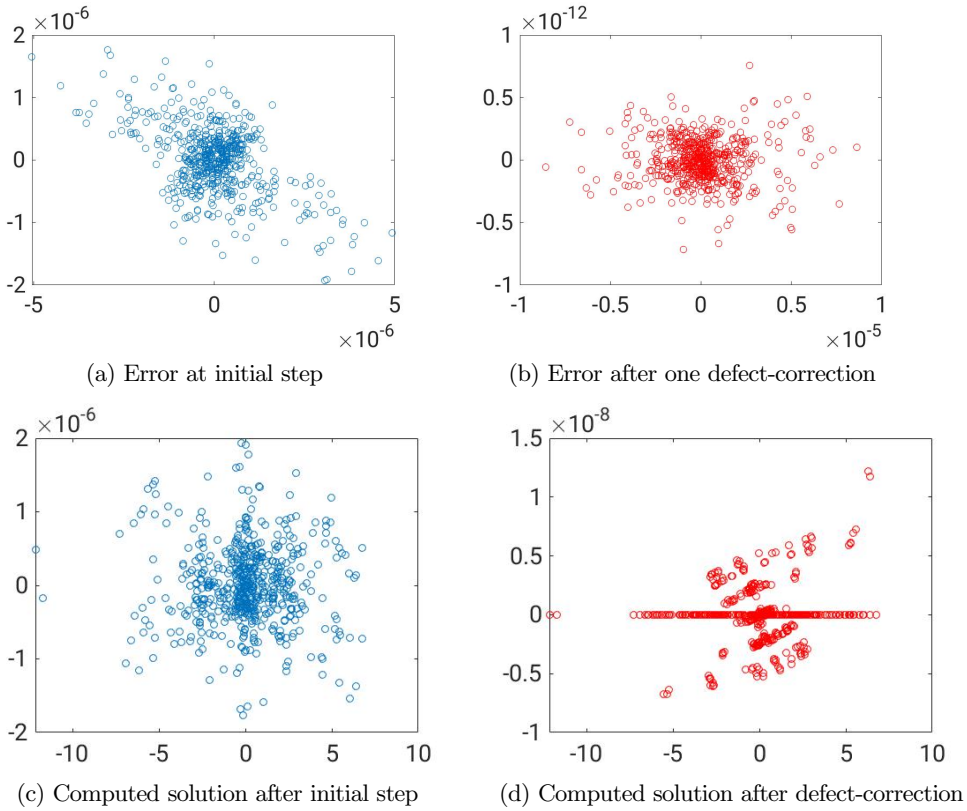
Table 1: Algorithm 1 with inexactly solved H_1 and H_2 for various values of β and ω

number of PRESB-preconditioned GCR iterations when solving H_1 and H_2 and Y denotes the average number of AGMG-preconditioned GCR to solve the blocks, arising in the PRESB preconditioning. The left $X(Y)$ group shows the iteration counts for the initial application of Algorithm 1 and on the right of / are the corresponding results of the defect correction step. We see that the convergence of AGMG somewhat deteriorates for the larger values of the regularization parameter β and for small ω.

In Figure 1 we show the effect of the solution scheme on the difference $\mathbf{z}_{ex} - \mathbf{z}_{it}$, where $\mathbf{z}_{ex}$ is obtained by a direct solution of the system in (3). Just one defect-correction step substantially improves the quality of the numerically computed solution, to be explained by less influence of arising rounding errors.

References

1. F. Tröltzsch, Optimal control of partial differential equations: theory, methods, and applications. *Grad. Stud. Math.* Vol. 112. American Math. Society, Providence, RI, 2010.
2. O. Axelsson, M. Neytcheva, A. Ström, An efficient preconditioning method for the state box-constrained optimal control problem. *J. Num. Math.* 26 (2018), 185-207.
3. Z.-Z. Liang, O. Axelsson, M. Neytcheva, A robust structured preconditioner for time-harmonic parabolic optimal control problems. *Numer. Algor.* 79 (2018), 575-596.
4. O. Axelsson, D. Lukáš, Preconditioning methods for eddy-current optimally controlled time-harmonic electromagnetic problems. *J. Numer. Math.* 27 (2019), 1-21.
5. O. Axelsson, M. Neytcheva, B. Ahmad, A comparison of iterative methods to solve complex valued linear algebraic systems. *Numer. Alg.*, 66(2014), 811-841.



(a) Error at initial step

(b) Error after one defect-correction

(c) Computed solution after initial step

(d) Computed solution after defect-correction

Fig. 1: Difference between exact and computed solution

6. O. Axelsson, M. Neytcheva, Preconditioners for two-by-two block matrices with square blocks, TR 2018-010 May 2018, Dept of Inf. Technology, Uppsala University
7. O. Axelsson, M. Béreš, R. Blaheta, Computational methods for boundary optimal control and identification problems, *Mathematics and Computers in Simulation*, 2021, in press.
8. Y. Saad, *Iterative methods for sparse linear systems*, SIAM, 2003, 2nd edition.
9. O. Axelsson, *Iterative solution methods*. Cambridge University Press, 1994.
10. Z.-Z. Liang, O. Axelsson, Exact inverse solution techniques for a class of complex-valued block two-by-two linear systems. Submitted.
11. H. Ammari, A. Buffa and J.-C. Nédélec, A Justification of Eddy Currents Model for the Maxwell Equations, *SIAM Journal on Applied Mathematics*, 60 (2000), 1805-1823. 2018, 424-431
12. O. Axelsson, J. Karatsson, Superlinear convergence using block preconditioners for the real system formulation of complex Helmholtz equations, *J. Computational and Applied Mathematics*, 340-431.
13. O. Axelsson, D. Lukáš, M. Neytcheva, An exact and approximate Schur complement method for time-harmonic optimal control problems. In preparetion.
14. Y. Notay, An aggregation-based algebraic multigrid method. *Electron. Trans. Numer. Anal.*, 37 (2010), 123–146.