



**HAL**  
open science

# Inversion of Eddy-Current Signals Using a Level-Set Method and Block Krylov Solvers

Lorenzo Audibert, Hugo Girardon, Housseem Haddar, Pierre Jolivet

► **To cite this version:**

Lorenzo Audibert, Hugo Girardon, Housseem Haddar, Pierre Jolivet. Inversion of Eddy-Current Signals Using a Level-Set Method and Block Krylov Solvers. *SIAM Journal on Scientific Computing*, 2023, 45 (3), pp.B366-B389. 10.1137/20M1382064 . hal-03043491v2

**HAL Id: hal-03043491**

**<https://hal.science/hal-03043491v2>**

Submitted on 3 Aug 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

1 **INVERSION OF EDDY-CURRENT SIGNALS USING A LEVEL-SET**  
2 **METHOD AND BLOCK KRYLOV SOLVERS\***

3 LORENZO AUDIBERT<sup>†</sup>, HUGO GIRARDON<sup>†‡</sup>, HOUSSEM HADDAR<sup>‡</sup>, AND PIERRE  
4 JOLIVET<sup>§</sup>

5 **Abstract.** The application motivating this work is related to the identification of deposits inside  
6 nuclear power plant steam generators using eddy-current probes. We consider a realistic experimental  
7 process that relies on the scan of a domain by sweeping along a tube axis a probe consisting of coils,  
8 playing the role of the sources/receivers. Solving the inverse shape problem associated with these  
9 measurements using a least squares method requires solutions to the eddy-current and the adjoint  
10 problems for a large number of right-hand sides at each gradient-descent iteration. Additional cost  
11 in the forward solver comes from the use of a potential formulation of the problem that provides  
12 independence from the conductive media topology (that may vary during iterations). We use a  
13 level-set approach to avoid remeshing and handle unknown topologies. The crucial ingredient in  
14 our algorithm is an optimized way of handling high numbers of right-hand sides for iterative solvers  
15 of large-scale problems. We first benchmark various block Krylov methods, block GMRES and  
16 block BGCRODR, to test their effectiveness compared to their standard counterpart, i.e., GMRES  
17 and GCRODR. Then, we propose for BGCRODR a new implementation for recycling information  
18 from previously generated Krylov bases that scales better than traditional approaches. This part  
19 is independent from the practical inverse problem at hand. The efficiency of the overall inversion  
20 procedure is finally demonstrated on realistic synthetic 3D examples.

21 **Key words.** eddy currents, domain decomposition preconditioners, block Krylov methods,  
22 large-scale inverse problems

23 **AMS subject classifications.** 35Q61, 49Q10, 78A46, 65F10

24 **1. Introduction.** Nuclear power plants are thermal power plants using nuclear  
25 fuel to produce electricity: heat given off by the nuclear reaction is transferred by  
26 water to a steam generator (SG) where it is used to vaporize colder water. The  
27 resulting vapor goes through a turbine to generate electricity. The focus here is the  
28 SG where hot water vaporizes cold water: it is composed of U-shaped tubes where  
29 hot water flows. These tubes are plunged inside cold water. Contact with the heated  
30 tube wall vaporizes the cold water: the vapor then streams upwards to the turbines.  
31 Over the course of the operation, the cold water creates metallic deposits on the tube  
32 outer wall [30]. These deposits deteriorate heat transfer on the tube [13], alter the  
33 flow of the water and may create additional mechanical constraints on the device:  
34 detection of such deposits is essential for the nuclear plant operator.

35 As direct inspection is impossible, indirect methods are used. Since the deposit  
36 and the tube inside the SG are conductive, eddy-current testing (ECT) constitutes the  
37 most suitable approach. ECT can be applied to different problematics, for instance,  
38 crack detection inside SG [28, 24] or in a different setting [14], or paired with thermog-  
39 raphy by using Joule effect [43]. The detection process using ECT is the following.  
40 After emptying the device from the water, probes are inserted from one end to the  
41 other end of a tube. By pulling them out at a constant speed, the operator is able  
42 to make measurements at regular positions alongside the tube. The resulting signal  
43 contains information on the medium configuration and, after some post-processing

---

\*Submitted to the editors 22/11/2020.

<sup>†</sup>PRISME, EDF R&D, 6 Quai Watier, 78400 Chatou, France

<sup>‡</sup>INRIA and Ecole Polytechnique, CMAP, Institut Polytechnique de Paris, 91128, Palaiseau, France

<sup>§</sup>IRIT-ENSEEIH, CNRS, 2 rue Camichel, 31000 Toulouse, France

44 steps, provides information on the shape and position of deposits.

45 The probes are composed of a given set of coils: when a coil, called the emitter, is  
 46 subject to a current, it produces an incident electromagnetic field. On the surface of  
 47 conductive materials, eddy currents generate another electromagnetic field, disturbing  
 48 the former. Another coil, called the receiver, then measures the flow of the distorted  
 49 field and compares it to that of the incident field: the difference between the different  
 50 flows is called impedance.

51

52 We propose to analyze the impedance signals using an inverse shape problem  
 53 approach. We formulate the inverse problem using a classical least squares functional  
 54 and solve the resulting optimization problem using a gradient descent with adjoint  
 55 state approach as in [17, 31]. These latter employ a boundary variation method or  
 56 parametric encoding of the geometry to update the shape boundary. In this paper  
 57 however, we choose to implicitly model the shape using a level-set function. The use  
 58 of level-set functions in shape optimization is widespread in recent papers, for instance  
 59 in optimal structure conceptions [42, 1], in electromagnetic inverse scattering [12], in  
 60 optical tomography [35], or in fluid mechanics [25]. It handles more easily topological  
 61 changes in the shape like merging or splitting in two connected components, while  
 62 at the same time reduces computational costs compared to a boundary variation  
 63 approach as the shape update does not require to re-mesh the computational domain  
 64 anymore. We adapt here the level-set approach to the inverse problem at hand,  
 65 introducing a regularization of the gradient descent and decoupling the mesh for the  
 66 level-set function from the mesh for computing the solution of the forward problem.

67 Computation of an impedance signal for a given configuration requires the solu-  
 68 tion of 3D time-harmonic Maxwell equations under the eddy-current approximation.  
 69 Since the level-set approach may lead during iterations to complex topologies of the  
 70 conductive domain, we choose to use an equivalent  $(\mathbf{A}, V)$  potential formulation of  
 71 the problem, see for instance [32] and references therein. This formulation has the  
 72 advantage of depending only on the topology of the whole computational domain.  
 73 In order to avoid adapting the mesh to different probe positions, we reformulate the  
 74 problem in terms of scattered fields and extend the potential formulation to this set-  
 75 ting. The main drawback of  $(\mathbf{A}, V)$  formulation is that it increases the size of the  
 76 discrete system as compared to other classical formulations in terms of electric or  
 77 magnetic fields [32]. In addition, for ECT, depending on the nature of the probe and  
 78 the scan width, the number of problems to solve can be very large (about a thousand)  
 79 at each gradient-descent iteration. Using the formulation in terms of scattered fields,  
 80 we are then faced with a critical issue encountered in large-scale inverse problems:  
 81 how to efficiently solve a large-scale forward problem for a large number of right-hand  
 82 sides?

83 For large-scale simulations, exact  $LU$  factorizations are not tractable using a  
 84 direct solver such as MUMPS [2]. Instead, specialized iterative methods may be  
 85 used. Indeed, they leverage the fact that the available right-hand sides, yielded by  
 86 the different coils and their positions, are available simultaneously. Block Krylov  
 87 methods are part of these specialized iterative methods. They have a higher arithmetic  
 88 intensity than standard Krylov methods, and typically converge in fewer iterates since  
 89 they generate larger Krylov subspaces at each iteration. In practice, these methods  
 90 are already used in geophysics [8] or tomography [41], where there are similar needs  
 91 for efficient solvers capable of dealing with multiple right-hand sides.

92 We here benchmark four different iterative solvers: GMRES [34], GCRODR [26],  
 93 block GMRES [15], and block GCRODR, first implemented in Belos [5] and then

94 independently formalized in [23] and [27]. While GMRES and to a lesser extent  
 95 GCRODR are widely used algorithms to solve linear systems, they turned out to be  
 96 non-effective for our inverse problem as they badly scale with the number of right-  
 97 hand sides. Block iterative solvers allow the user to solve blocks of right-hand sides at  
 98 the same time and are more adapted. However, they are more memory demanding.  
 99 Given the size of the blocks in our problem, handling all right-hand sides (RHS) at  
 100 the same time is not tractable. As such, we split the full block of RHS into smaller  
 101 sub-blocks and try different sub-block size to determine an optimal parameter for  
 102 our application. The difference between block GMRES and block GCRODR lies in  
 103 the recycling option provided by the latter: from one block to another, we are able  
 104 to recycle basis vectors from one sub-block solve to another. Recycling is supposed  
 105 to provide a faster convergence, interested readers are referred to [37] for a survey  
 106 on recycling methods. In our case, while the number of iterates indeed lowers with  
 107 block GCRODR, we observed that the solve time increases. We thus propose a new  
 108 redistribution scheme to increase the performance of block GCRODR.

109 The effectiveness of the whole inversion procedure is tested for realistic experimen-  
 110 tal scenarios and realistic physical parameters provided by our industrial partner. We  
 111 simulate measurements associated with so-called SAX probe (axisymmetric probe)  
 112 and measurements associated with so-called SMX probe (non-axisymmetric). We  
 113 demonstrate in particular the efficiency of our algorithm in handling both scenarios.  
 114 Thanks to block Krylov methods, inversion of data provided by SMX probe for a  
 115 typical experiment is feasible within a reasonable time.

116 The paper is organized as follows. In section 2, the  $(\mathbf{A}, V)$  formulation is presented  
 117 and extended to the scattered field. In section 3, the general context of the appli-  
 118 cation at hand is explained and the inversion algorithm for reconstructing deposits  
 119 is described. Multiple numerical solution strategies are benchmarked in section 4.  
 120 The optimal configuration is then used for complete inverse simulations in section 5.  
 121 Concluding remarks are given in section 6.

## 122 2. The forward problem.

123 **2.1. Problem formulation.** Let  $\Omega \subset \mathbb{R}^3$  be the computational domain of inter-  
 124 est inside the SG that is assumed to be with Lipschitz boundary and later will  
 125 be assumed to be also simply connected with connected boundary and either regular  
 126 or convex polyhedral. The medium physical parameters, namely the electric permit-  
 127 tivity  $\varepsilon(\mathbf{x}) > 0$ , the conductivity  $\sigma(\mathbf{x}) \geq 0$  and the magnetic permeability  $\mu(\mathbf{x}) > 0$   
 128 are assumed to be piecewise constant functions. Let  $\Omega_C$  be the conductive domain,  
 129 i.e., the region where  $\sigma \neq 0$  and  $\Omega_I = \Omega \setminus \overline{\Omega_C}$  be the insulator domain. We denote  
 130  $\Gamma := \partial\Omega_I \cap \partial\Omega_C$  the interface between insulator and conductor domains.

131 Let  $\mathbf{J}$  be the current density and  $(\mathbf{E}, \mathbf{H})$  be the electromagnetic field induced by  
 132 the current. Considering a time-harmonic framework, with  $\omega$  being the pulsation and  
 133 the eddy-current approximation  $\omega\varepsilon \ll \sigma$ , the 3D time-harmonic Maxwell equations  
 134 lead to the following system:

$$135 \quad (2.1) \quad \begin{cases} \operatorname{curl} \mathbf{E} - i\omega\mu\mathbf{H} = \mathbf{0} & \text{in } \Omega, \\ \operatorname{curl} \mathbf{H} - \sigma\mathbf{E} = \mathbf{J} & \text{in } \Omega, \\ \operatorname{div}(\varepsilon\mathbf{E}) = 0 & \text{in } \Omega_I, \end{cases}$$

136 that has to be complemented with some appropriate boundary conditions on  $\partial\Omega$  and  
 137 some compatibility conditions on  $\partial\Omega_I$  for the normal component of  $\mathbf{E}|_{\partial\Omega_I}$ . The latter  
 138 will not be specified since it is not needed in the adopted formulation hereafter. For

139 the boundary conditions on  $\partial\Omega$ , we shall impose  $\mathbf{H} \times \mathbf{n} = \mathbf{0}$  where  $\mathbf{n}$  denotes the  
140 outward normal.

141 Solving (2.1) can be proven to be rather difficult, as the eddy-current approxima-  
142 tion introduces a different behavior of the fields in the insulator and in the conductor  
143 domains. For instance, when  $\Omega_{\mathcal{I}}$  and/or its boundary are not simply connected the  
144 computation of  $(\mathbf{E}, \mathbf{H})$  may require the introduction of cuts and associated harmonic  
145 functions. For the application described later, the computational domain  $\Omega$  is simply  
146 connected with connected boundary. Therefore, in order to remove the issue related  
147 to the connectivity of  $\Omega_{\mathcal{I}}$ , we propose here to formulate (2.1) using the potentials  
148  $(\mathbf{A}, V)$  defined by (see for instance [32]):

$$149 \quad (2.2) \quad \mu\mathbf{H} = \mathbf{curl} \mathbf{A} \text{ in } \Omega, \quad \mathbf{E} = i\omega\mathbf{A} + \nabla V \text{ in } \Omega_C,$$

150 together with the Coulomb gauge  $\operatorname{div} \mathbf{A} = 0$  in  $\Omega$  and the additional the boundary  
151 condition  $\mathbf{A} \cdot \mathbf{n} = 0$  on  $\partial\Omega$ . The existence of the decomposition is motivated by the  
152 first equation in (2.1). Inserting these definitions into the second equation of (2.1)  
153 and the boundary conditions on  $\partial\Omega$  yield the following system:

$$154 \quad (2.3) \quad \begin{cases} \mathbf{curl}(\mu^{-1}\mathbf{curl} \mathbf{A}) - \sigma(i\omega\mathbf{A} + \nabla V) = \mathbf{J} & \text{in } \Omega, \\ \operatorname{div} \mathbf{A} = 0 & \text{in } \Omega, \\ \mathbf{A} \cdot \mathbf{n} = 0 \text{ and } (\mu^{-1}\mathbf{curl} \mathbf{A}) \times \mathbf{n} = \mathbf{0} & \text{on } \partial\Omega. \end{cases}$$

155 These equations allow to compute the electromagnetic field inside the conductive  
156 material and would be sufficient to simulate the impedance measurements we are  
157 interested in later.

158 *Remark 2.1.* Note that  $V$  is defined up to an additive constant in each connected  
159 component of  $\Omega_C$ . In fact, since  $\Omega$  is simply connected with connected boundary, there  
160 exists a function  $\tilde{V}$  such that  $\mathbf{E} = i\omega\mathbf{A} + \nabla\tilde{V}$  in  $\Omega$ . The function  $\tilde{V}$  may therefore differ  
161 from the function  $V$  determined by (2.3) by a constant in each connected component  
162 of  $\Omega_C$ .

163 In a finite element framework, when solving numerically (2.3), the gauge condition is  
164 difficult to implement as it requires one to build a discrete function space of divergence-  
165 free functions. To remove the condition from the functional space, we adopt the  
166 procedure introduced in [9] and modify the first equation in (2.3) as

$$167 \quad \mathbf{curl}(\mu^{-1}\mathbf{curl} \mathbf{A}) - \mu_*^{-1}\nabla(\operatorname{div} \mathbf{A}) - \sigma(i\omega\mathbf{A} + \nabla V) = \mathbf{J} \quad \text{in } \Omega,$$

168 where  $\mu_*$  is a positive constant. It can be chosen in practice as an average value of  
169  $\mu$ . By adding the penalization term, we lose the relation that links  $\mathbf{E}$  and  $\mathbf{J}$  inside  $\Omega$ ,  
170 namely  $\operatorname{div}(\sigma\mathbf{E}) = -\operatorname{div} \mathbf{J}$  in  $\Omega$ , that should be added to the system. Assuming that  
171  $\operatorname{div} \mathbf{J} = 0$  in  $\Omega$ , we then obtain the following equivalent system

$$172 \quad (2.4) \quad \begin{cases} \mathbf{curl}(\mu^{-1}\mathbf{curl} \mathbf{A}) - \mu_*^{-1}\nabla(\operatorname{div} \mathbf{A}) - \sigma(i\omega\mathbf{A} + \nabla V) = \mathbf{J} & \text{in } \Omega, \\ \operatorname{div}(\sigma(i\omega\mathbf{A} + \nabla V)) = 0 & \text{in } \Omega_C, \\ \sigma(i\omega\mathbf{A} + \nabla V) \cdot \mathbf{n} = 0 & \text{on } \Gamma, \\ \mathbf{A} \cdot \mathbf{n} = 0 \text{ and } (\mu^{-1}\mathbf{curl} \mathbf{A}) \times \mathbf{n} = \mathbf{0} & \text{on } \partial\Omega. \end{cases}$$

173 Let us introduce the function spaces

$$174 \quad \mathbf{H}(\operatorname{curl}, \Omega) := \{\mathbf{A} \in L^2(\Omega)^3, \mathbf{curl} \mathbf{A} \in L^2(\Omega)^3\}$$

175  
176

$$\mathbf{H}_0(\operatorname{div}, \Omega) := \{\mathbf{A} \in L^2(\Omega)^3, \operatorname{div} \mathbf{A} \in L^2(\Omega), \mathbf{A} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega\}$$

177 and set  $\mathbf{X}(\Omega) := \mathbf{H}(\operatorname{curl}, \Omega) \cap \mathbf{H}_0(\operatorname{div}, \Omega)$ . Assume that  $\Omega_C$  has  $M$  connected compo-  
178 nents  $\Omega_C^i$ ,  $i = 1, \dots, M$ , then we define

$$179 \quad \tilde{H}^1(\Omega_C) := \left\{ V \in H^1(\Omega), \int_{\Omega_C^i} V dx = 0, i = 1, \dots, M \right\}.$$

180 Multiplying the first equation, resp. the second equation, in (2.4) with a test function  
181  $\bar{\mathbf{B}} \in \mathbf{X}(\Omega)$ , resp.  $\bar{Q}/i\omega \in \tilde{H}^1(\Omega_C)$ , integrating by parts over  $\Omega$ , using the boundary  
182 conditions, and summing the resulting equations leads to the following variational  
183 formulation [32, 17]:

$$184 \quad (2.5) \quad \mathcal{A}((\mathbf{A}, V), (\mathbf{B}, Q)) = \mathcal{L}((\mathbf{B}, Q)), \quad \forall (\mathbf{B}, Q) \in \mathbf{X}(\Omega) \times \tilde{H}^1(\Omega_C)$$

185

$$\begin{aligned} \text{with: } \mathcal{A}((\mathbf{A}, V), (\mathbf{B}, Q)) &:= \int_{\Omega} [\mu^{-1} \operatorname{curl} \mathbf{A} \cdot \operatorname{curl} \bar{\mathbf{B}} + \mu_*^{-1} (\operatorname{div} \mathbf{A})(\operatorname{div} \bar{\mathbf{B}})] \, dx \\ &+ \frac{1}{i\omega} \int_{\Omega_C} \sigma (i\omega \mathbf{A} + \nabla V) \cdot \overline{(i\omega \mathbf{B} + \nabla Q)} \, dx, \\ \mathcal{L}((\mathbf{B}, Q)) &:= \int_{\Omega} \mathbf{J} \cdot \bar{\mathbf{B}} \, dx. \end{aligned}$$

186

187 We now can state the following theorem that can be deduced from a more general  
188 result in [32, chapter 6], see also [17].

189 **THEOREM 2.2.** *Assume that  $\Omega$  is simply connected with a connected boundary*  
190 *and that  $\mathbf{J} \in X(\Omega)'$ . Then problem (2.5) admits a unique solution  $(\mathbf{A}, V) \in \mathbf{X}(\Omega) \times$*   
191  *$\tilde{H}^1(\Omega_C)$ . Assume in addition that  $\operatorname{div} \mathbf{J} = 0$  in  $\Omega$ . Then the solution of (2.5) satisfies*  
192 *(2.4) or equivalently (2.3) and the fields  $(\mathbf{E}, \mathbf{H})$  defined by (2.2) verify the first two*  
193 *equations in (2.1).*

194 **2.2. Scattered field formulation.** As explained later, the goal is to reconstruct  
195 a deposit that appears in a reference configuration characterized by the physical pa-  
196 rameters  $\sigma_0$  and  $\mu_0$ , and the associated conductive part  $\Omega_C^0$ . Let us denote by  $(\mathbf{E}^0, \mathbf{H}^0)$   
197 the electromagnetic field associated with this configuration and a source term  $\mathbf{J}$ . This  
198 field verifies in particular

$$199 \quad (2.6) \quad \begin{cases} \operatorname{curl} \mathbf{E}^0 - i\omega \mu_0 \mathbf{H}^0 = \mathbf{0} & \text{in } \Omega, \\ \operatorname{curl} \mathbf{H}^0 - \sigma_0 \mathbf{E}^0 = \mathbf{J} & \text{in } \Omega, \\ \mathbf{H}^0 \times \mathbf{n} = \mathbf{0} & \text{on } \partial\Omega. \end{cases}$$

200 In order to speed up calculations in cases where a large number of different source  
201 terms  $\mathbf{J}$  is used and avoid remeshing (see the discussion at the end of this section), it  
202 is more advantageous to solve for the scattered fields

$$203 \quad (\mathbf{E}^s, \mathbf{H}^s) := (\mathbf{E}, \mathbf{H}) - (\mathbf{E}^0, \mathbf{H}^0)$$

204 assuming that  $(\mathbf{E}^0, \mathbf{H}^0)$  has been computed offline. As above, one can set up equations  
205 for the scattering field in terms of potentials. We hereafter give an outline. Taking  
206 the difference between the two first equations of (2.1) and (2.6) we obtain

$$207 \quad \operatorname{curl} \mathbf{E}^s - i\omega(\mu \mathbf{H} - \mu_0 \mathbf{H}^0) = \mathbf{0} \quad \text{in } \Omega.$$

208 As previously, when  $\Omega$  is simply connected with connected boundary, this equation  
209 implies the existence of potentials  $(\mathbf{A}^s, V^s)$  such that

$$210 \quad (2.7) \quad \mu \mathbf{H} - \mu_0 \mathbf{H}^0 = \mathbf{curl} \mathbf{A}^s \text{ in } \Omega, \quad \mathbf{E}^s = i\omega \mathbf{A}^s + \nabla V^s \text{ in } \Omega_C,$$

211 with the Coulomb gauge  $\operatorname{div} \mathbf{A}^s = 0$  in  $\Omega$  and the additional boundary condition  
212  $\mathbf{A}^s \cdot \mathbf{n} = 0$  on  $\partial\Omega$ . Inserting these definitions into the second equation of (2.1) and  
213 the boundary conditions on  $\partial\Omega$  yield the following system:

$$214 \quad (2.8) \quad \begin{cases} \mathbf{curl}(\mu^{-1} \mathbf{curl} \mathbf{A}^s) - \sigma(i\omega \mathbf{A}^s + \nabla V^s + \mathbf{E}^0) = \mathbf{J}_0 & \text{in } \Omega, \\ \operatorname{div} \mathbf{A}^s = 0 & \text{in } \Omega, \\ \mathbf{A}^s \cdot \mathbf{n} = 0 \text{ and } (\mu^{-1} \mathbf{curl} \mathbf{A}^s) \times \mathbf{n} = \mathbf{0} & \text{on } \partial\Omega, \end{cases}$$

215 with  $\mathbf{J}_0$  defined by

$$216 \quad \mathbf{J}_0 := \mathbf{curl} \left( \left( 1 - \frac{\mu_0}{\mu} \right) \mathbf{H}^0 \right) - \sigma_0 \mathbf{E}^0 \quad \text{in } \Omega.$$

217 Clearly, if  $\operatorname{div} \mathbf{J} = 0$  in  $\Omega$ , then we also have  $\operatorname{div} \mathbf{J}_0 = 0$  in  $\Omega$  and therefore as explained  
218 previously, the system (2.8) can be equivalently written as

$$219 \quad (2.9) \quad \begin{cases} \mathbf{curl}(\mu^{-1} \mathbf{curl} \mathbf{A}^s) - \mu_*^{-1} \nabla(\operatorname{div} \mathbf{A}^s) - \sigma(i\omega \mathbf{A}^s + \nabla V^s + \mathbf{E}^0) = \mathbf{J}_0 & \text{in } \Omega, \\ \operatorname{div}(\sigma(i\omega \mathbf{A}^s + \nabla V^s + \mathbf{E}^0)) = 0 & \text{in } \Omega_C, \\ \sigma(i\omega \mathbf{A}^s + \nabla V^s + \mathbf{E}^0) \cdot \mathbf{n} = 0 & \text{on } \Gamma, \\ \mathbf{A}^s \cdot \mathbf{n} = 0 \text{ and } (\mu^{-1} \mathbf{curl} \mathbf{A}^s) \times \mathbf{n} = \mathbf{0} & \text{on } \partial\Omega. \end{cases}$$

220 Similarly to (2.5), the variational formulation of this problem can be written as

$$221 \quad (2.10) \quad \mathcal{A}((\mathbf{A}^s, V^s), (\mathbf{B}, Q)) = \mathcal{L}^s((\mathbf{B}, Q)), \quad \forall (\mathbf{B}, Q) \in \mathbf{X}(\Omega) \times \tilde{H}^1(\Omega_C)$$

222 where the right-hand side can be expressed as

$$223 \quad \mathcal{L}^s((\mathbf{B}, Q)) := \int_{\Omega} \left( 1 - \frac{\mu_0}{\mu} \right) \mathbf{H}^0 \cdot \mathbf{curl} \bar{\mathbf{B}} \, dx - \frac{1}{i\omega} \int_{\Omega_C} (\sigma - \sigma_0) \mathbf{E}^0 \cdot \overline{(i\omega \mathbf{B} + \nabla Q)} \, dx.$$

224 As an immediate corollary of [Theorem 2.2](#), we can state the following theorem on the  
225 well-posedness of this problem and the equivalence with the original problem.

226 **THEOREM 2.3.** *Assume that  $\Omega$  is simply connected with connected boundary and*  
227 *that  $(\mathbf{E}^0, \mathbf{H}^0) \in L^2(\Omega)^3 \times L^2(\Omega)^3$ . Then, problem (2.10) admits a unique solution*  
228  *$(\mathbf{A}^s, V^s) \in \mathbf{X}(\Omega) \times \tilde{H}^1(\Omega_C)$ . Assume in addition that  $(\mathbf{E}^0, \mathbf{H}^0)$  satisfies (2.6) with*  
229  *$\operatorname{div} \mathbf{J} = 0$  in  $\Omega$ . Then, the solution of (2.10) satisfies (2.9) or equivalently (2.8) and*  
230 *the fields  $(\mathbf{E}, \mathbf{H})$  defined by (2.7) verify the first two equations in (2.6).*

231 In the context of deposit detection presented below, the probe takes measurements at  
232 regular positions alongside the tube. A typical scan requires up to 140 probe positions:  
233 for each one, a new computation of the electromagnetic field is required. Considering  
234 the variational formulation (2.5), as the right-hand side depends on the current density  
235  $\mathbf{J}$  located in the probe, computation of the field for each position requires a different  
236 mesh where the probe is at the proper position. Conversely, in (2.10), the right-side  
237 depends on the incident field only on the domain where  $\sigma \neq \sigma_0$  or  $\mu \neq \mu_0$ , which is  
238 independent from the probe position. Consequently, assuming that the incident fields  
239 are computed offline for any probe position, solving (2.10) do not require remeshing  
240 for different probe positions and therefore is more computationally efficient.

241 **2.3. Finite element discretization.** To solve (2.5) or (2.10), we shall use  
 242 finite elements on a tetrahedralization  $\mathcal{T}_h$  of the domain  $\Omega$ , where  $h$  is the mesh  
 243 size. A tetrahedralization of the conductive domain would be a restriction of  $\mathcal{T}_h$   
 244 to  $\Omega_C$ . Let  $K$  indicate an element of the tetrahedralization. We choose piece-  
 245 wise affine Lagrange elements to discretize the scalar potential  $V$ . We denote by  
 246  $V_h^1(\Omega_C) := \{v_h \in \mathcal{C}^0(\Omega_C) / \forall K \subset \mathcal{T}_h, v_h|_K \in \mathbb{P}^1(K)\}$  the discrete space associated  
 247 with  $H^1(\Omega_C)$ .

248 Assuming that  $\Omega$  is a regular domain, the space  $\mathbf{H}(\text{curl}, \Omega) \cap \mathbf{H}_0(\text{div}, \Omega)$  is iso-  
 249 metric to  $H^1(\Omega)^3 \cap \mathbf{H}_0(\text{div}, \Omega)$ . This is also the case when  $\Omega$  is a convex cylinder or  
 250 polyhedron [9], which is the case of our numerical experiments. In this case, piecewise  
 251 affine Lagrange elements can also be used to discretize each component of the vector  
 252 potential. The discrete space associated with  $X(\Omega)$  is then

$$253 \quad \mathbf{X}_h^1(\Omega) = \{\mathbf{w}_h \in (\mathcal{C}^0(\Omega))^3 / \mathbf{w}_h|_K \in (\mathbb{P}^1(K))^3 \forall K \in \mathcal{T}_h, \mathbf{w}_h \cdot \mathbf{n} = 0 \text{ on } \partial\Omega\}.$$

254 Since the scalar potential is assumed to have zero mean value in each connected  
 255 component of  $\Omega_C$ , we enforce this condition by adding a penalization of the form  
 256  $\int_{\Omega_C} \delta_0 \sigma V \bar{Q} \, d\mathbf{x}$  where  $0 < \delta_0 \ll 1$ . The value of  $\delta_0$  is chosen empirically using cali-  
 257 bration on a test case merely used to ensure the zero mean value constraint. It may  
 258 be chosen arbitrarily low as long as solvers can handle it numerically when solving  
 259 the corresponding discrete system. To summarize, the discrete system is built by  
 260 replacing the sesquilinear form  $\mathcal{A}$  by

$$261 \quad \mathcal{A}((\mathbf{A}, V), (\mathbf{B}, Q)) + \frac{1}{i\omega} \int_{\Omega_C} \delta_0 \sigma V \bar{Q} \, d\mathbf{x},$$

262 and replacing the variational space  $X(\Omega) \times \tilde{H}^1(\Omega)$  with  $\mathbf{X}_h^1(\Omega) \times V_h^1(\Omega_C)$ . The numer-  
 263 ical implementation of the resulting scheme has been done using the finite element  
 264 library FreeFEM [18].

### 265 3. The inverse problem.

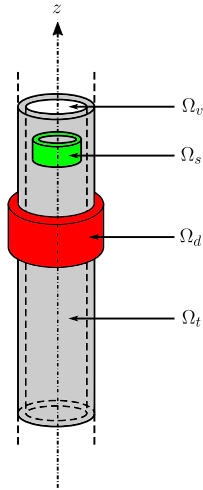


Fig. 1: Domain  $\Omega$  for the Maxwell equations



266 **3.1. Description of the experiment.** Let us specify the composition of the  
 267 computational domain in the context of shape reconstruction inside a SG [28]. Con-  
 268 sider a U-shaped tube, the focus being placed here on the straight part of the tube  
 269 which is assumed to be a cylinder around the  $z$ -axis. We denote by:

- 270 •  $\Omega_v$  the vacuum domain inside and outside the tube, with physical parameters  
 271  $(\sigma_v = 0, \mu_v)$ ;
- 272 •  $\Omega_t$  the domain occupied by the tube consisting of a conductive material, with  
 273 physical parameters  $(\sigma_t, \mu_t)$ ;
- 274 •  $\Omega_d$  the deposit domain located in the exterior of the tube, with physical  
 275 parameters  $(\sigma_d, \mu_d)$ ;
- 276 •  $\Omega_s$  the domain occupied by the probe placed inside the tube.

277 The current density  $\mathbf{J}$  is considered to be compactly supported in  $\Omega_s$  and divergence-  
 278 free. Figure 1 displays the main features of the domain. We assume here that the  
 279 probe conductivity can be neglected compared to the remaining conductive materials.  
 280 The conducting part  $\Omega_c$  consists of the tube and the deposit while the insulating part  
 281 consists of the vacuum and the probe. The computational domain  $\Omega$  is a cylinder of  
 282 height  $H$  and radius  $R$  chosen to be sufficiently large.

283  
 284 The physical parameters are assumed to be known a priori and the only unknown  
 285 for the inverse problem is the deposit  $\Omega_d$ . In this context and referring to the notation  
 286 of the previous section, the reference media is defined by (with  $\chi_O$  denoting the  
 287 characteristic function of a domain  $O$ ),

$$288 \quad \sigma_0 := \sigma_t \chi_{\Omega_t} \quad \mu_0 := \mu_t \chi_{\Omega_t} + \mu_v \chi_{\Omega_v} + \mu_v \chi_{\Omega_s}$$

289 while

$$290 \quad \sigma := \sigma_t \chi_{\Omega_t} + \sigma_d \chi_{\Omega_d} \quad \mu := \mu_t \chi_{\Omega_t} + \mu_v \chi_{\Omega_v} + \mu_v \chi_{\Omega_s} + (\mu_d - \mu_v) \chi_{\Omega_d}.$$

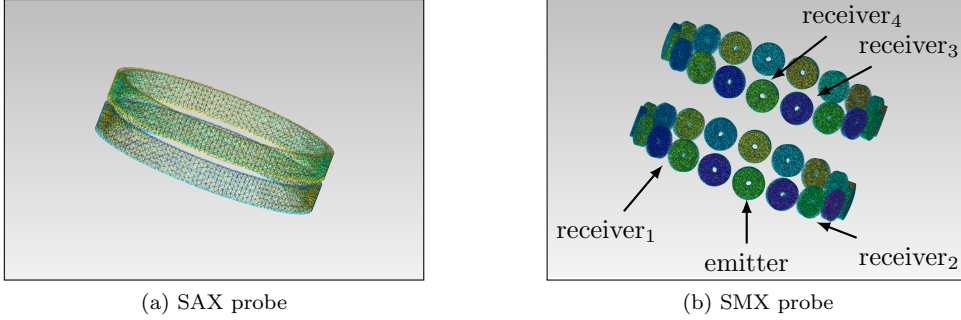


Fig. 2: Two probes used for ECT

291 The data for the deposit identification is collected as follows. A probe composed  
 292 of  $N_c$  coils is inserted inside the tube to one end. The probe is then pulled at constant  
 293 speed to the other end. At regular positions, the coils are subjected to a current  $I$ ,  
 294 inducing an electromagnetic wave. Figure 2 displays two examples of probes used  
 295 in ECT: while the SAX probe is composed of two coaxial coils, the SMX probe  
 296 has two rows of coils placed around the probe axis. Note that due to its structure,

297 the former provides information that is averaged on the azimuthal component. We  
 298 denote by  $N_p$  the number of probe positions. The coil  $l$  used to generate the fields  
 299 is called the emitter, while the coil  $k$  measuring the flow is called the receiver. The  
 300 corresponding measured signal, called impedance signal and denoted  $\Delta Z_{kl}$ , has the  
 301 following expression [24]:

$$302 \quad (3.1) \quad \Delta Z_{kl} = \frac{1}{i\omega I^2} \int_{\Omega_d} \left( \left( \frac{1}{\mu} - \frac{1}{\mu_0} \right) (\mathbf{curl} \mathbf{E}_k) \cdot (\mathbf{curl} \mathbf{E}_l^0) - i\omega(\sigma - \sigma_0) \mathbf{E}_k \cdot \mathbf{E}_l^0 \right) dx,$$

303 where the notation  $(\mathbf{E}_k, \mathbf{H}_k)$  and  $(\mathbf{E}_k^0, \mathbf{H}_k^0)$  respectively refers to the solution of (2.1)  
 304 and (2.6) where the source term  $\mathbf{J}$  is supported by the coil  $k$ . Note that from this  
 305 definition, we have the equality  $\Delta Z_{kl} = \Delta Z_{lk}$  for any  $k$  and  $l$ .

306

307 In practice, the probes cannot measure  $\Delta Z_{kl}$ , but rather linear combinations of  
 308 these quantities called modes. Consider two coils  $k$  and  $l$ , then there are two main  
 309 modes for these coils:

310 **differential mode**  $Z_F = 0.5i(\Delta Z_{ll} - \Delta Z_{kk})$ ;

311 **absolute mode**  $Z_{FA} = 0.5i(\Delta Z_{ll} + \Delta Z_{kk})$ .

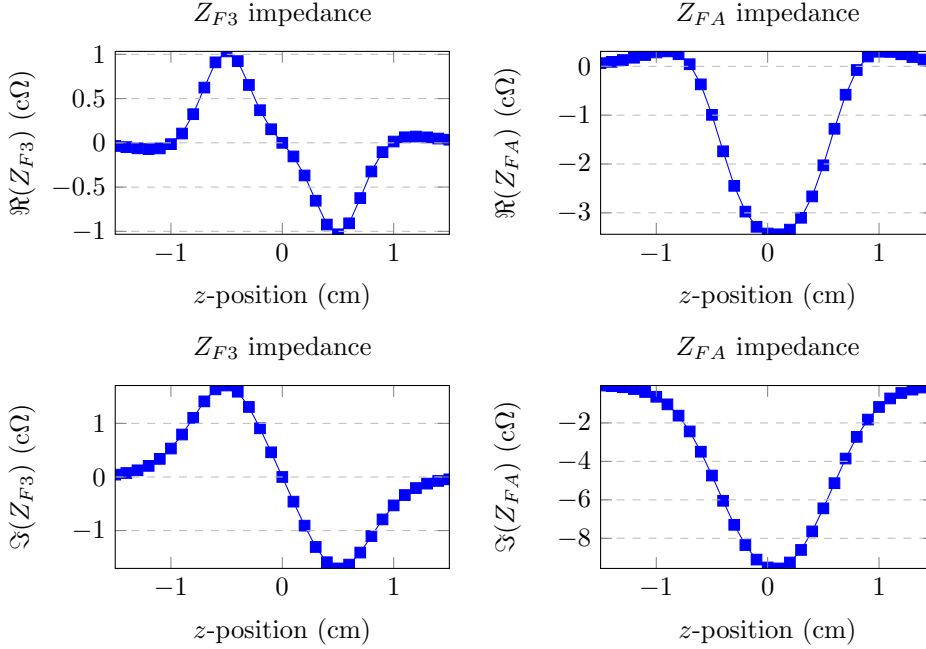


Fig. 3: Example of SAX impedance signals for an annular deposit between  $z_- = -0.5$  cm and  $z_+ = 0.5$  cm, at frequency 100 kHz.

312 Experimental observations show that each of these modes has different sensitivi-  
 313 ties with respect to the deposit. Roughly speaking,  $Z_F$  better detects sharp variations  
 314 in the shape geometry while  $Z_{FA}$  is more suited to identify smooth variations. To  
 315 illustrate these observations, Figure 3 displays examples of impedance signals for an  
 316 annular deposit between  $z_- = -0.5$  cm and  $z_+ = 0.5$  cm, at frequency 100 kHz, with  
 317 the SAX probe. We do not elaborate here on the specifics of the acquisitions and  
 318 refer to [28] for more details.

319 In practice, the SAX probe can work with three different pulsations  $\omega_1 > \omega_2 > \omega_3$   
 320 and generates a differential mode for each pulsation. Note that for the lowest, it  
 321 generates an additional absolute mode. The conductivity of the tube absorbs most  
 322 of the energy delivered by the coil, the absorption rate increasing with the frequency.  
 323 This is why we focus here only on  $\omega_3 = 2\pi \cdot 10^5$  rad/sec as it ensures that some of  
 324 the electromagnetic energy reaches the outer part of the tube. As such, a SAX probe  
 325 provides two impedance values for each  $z$ -position in this paper.

326 The SMX probe can also operate at a fourth pulsation  $\omega_4 < \omega_3$ , though we  
 327 consider here only  $\omega_3$ . It generates absolute mode signals according to the following  
 328 acquisition rule: the emitter coils are contained on the lower row. For each emitter,  
 329 there are four associated receiver coils, as displayed in [Figure 2b](#). Since the probe has  
 330 19 coils on each row, the device provides 76 impedance values for each  $z$ -position in  
 331 this paper.

332 **3.2. Description of the inversion algorithm.** We here give a description of  
 333 the adopted inversion strategy to reconstruct the shape and position of deposits from  
 334 measured signals. For a given  $z$ -position of the probe, let us denote by  $\mathbf{Z}_{\text{meas}}^i(z)$   
 335 the measured signal and  $\mathbf{Z}^i(\Omega_d, z)$  the computed signal for a given shape  $\Omega_d$ , where  
 336  $i = 1, \dots, N_s$  refer to the index of the signals ( $N_s = 2$  for SAX and  $N_s = 76$  for SMX).  
 337 The numerical evaluation of the impedance signal is obtained by solving (2.10). We  
 338 now formulate the inverse problem as an optimization problem for a least squares  
 339 misfit cost functional:

$$340 \quad \min_{\Omega_d} \left( \mathcal{J}(\Omega_d) = \frac{1}{N_s} \sum_{i=1}^{N_s} \int_{z_-}^{z_+} |\mathbf{Z}^i(\Omega_d, \zeta) - \mathbf{Z}_{\text{meas}}^i(\zeta)|^2 d\zeta \right).$$

342 This formulation has been solved [21, 17, 31] using a gradient-descent method and a  
 343 boundary variation technique. One drawback of such an approach is that it modifies  
 344 at each iteration the computational mesh by moving the shape boundary accordingly,  
 345 imposing a remeshing step. A possible remedy has been proposed in [31] by the  
 346 use of a fixed Cartesian mesh in the region containing the deposit and consider only  
 347 boundaries that can be exactly represented by the considered mesh. Indeed, such a  
 348 procedure imposes a strong bias on the reconstruction and cannot handle complex  
 349 topologies. We propose here to use a level-set approach [25] where the boundary  $\partial\Omega_d$   
 350 is implicitly encoded by the zero level-set of a function  $\psi$ . Denote by  $D \subset \Omega$  a domain  
 351 containing all admissible shapes  $\Omega_d$ , called Region Of Interest (ROI). The function  $\psi$   
 352 is then defined on  $D$  and verifies

$$353 \quad \psi(\mathbf{x}) \begin{cases} < 0 & \text{if } \mathbf{x} \in \Omega_d, \\ = 0 & \text{if } \mathbf{x} \in \partial\Omega_d, \\ > 0 & \text{if } \mathbf{x} \in D/\overline{\Omega_d}. \end{cases}$$

354 Since it is not directly correlated to the solution of the forward problem, the domain  
 355  $D$  is meshed independently from the mesh used for computing the solutions to (2.10).  
 356 However, the two meshes are kept fixed during iterations. The following scheme then  
 357 requires only interpolation operations from one mesh to another.

358 Following previous works on level-set-based shape optimization [1, 10], the shape  
 359 update at each iteration under this model is equivalent to convecting the level-set  
 360 according to the Hamilton–Jacobi equation:

$$361 \quad (3.2) \quad \frac{\partial\psi}{\partial t} + G|\nabla\psi| = 0 \quad \text{in } D,$$

362 where  $G$  is a deformation direction defined on  $D$  as an extension of the shape gradient  
 363 of the cost functional  $\mathcal{J}$  as explained below. Note that the convection time is a step  
 364 that needs to be adjusted at each iteration in order to ensure a fast convergence. In  
 365 our numerical algorithm, (3.2) is solved using a backward method of characteristics  
 366 as proposed in [6].

367 *Evaluation of the descent direction  $G$ .* The computation of the descent direc-  
 368 tion uses the solution of an additional variational problem for so-called adjoint state.  
 369 Following [21, 31], it is defined as the field  $(\mathbf{P}, W) \in X(D) \times \tilde{H}^1(\Omega_C)$ , solution of:

$$370 \quad \mathcal{A}^*((\mathbf{P}, W), (\mathbf{B}, Q)) = \overline{\mathcal{A}((\mathbf{B}, Q), (\mathbf{P}, W))} = \mathcal{L}^*((\mathbf{B}, Q)), \forall (\mathbf{B}, Q) \in \mathbf{X}(\Omega) \times \tilde{H}^1(\Omega_C),$$

$$371 \quad \text{with } \mathcal{L}^*((\mathbf{B}, Q)) := \int_{\Omega_d} \left( \frac{-1}{i\omega} \left( \frac{1}{\mu} - \frac{1}{\mu_0} \right) \mathbf{curl} \bar{\mathbf{B}} \cdot \mathbf{curl} \bar{\mathbf{E}}^0 \right. \\ \left. + \frac{1}{i\omega} (\sigma - \sigma_0) \overline{(i\omega \mathbf{B} + \nabla Q)} \cdot \bar{\mathbf{E}}^0 \right) dx.$$

372  
 373  
 374 These fields can be seen as the Lagrange multiplier of the optimization problem  
 375 where the constraint is the variational formulation (2.5). Using these fields, we are  
 376 able to write the shape derivative of the cost function as (see [21] for more details on  
 377 the calculations):

$$378 \quad \mathcal{J}'(\Omega_d)(\boldsymbol{\theta}) = -\frac{1}{N_s} \sum_{i=1}^{N_s} \frac{\omega}{I^2} \int_{\partial\Omega_d} (\boldsymbol{\theta} \cdot \mathbf{n}) g^i ds.$$

379 For a signal  $i$ , let us denote by  $k_i$  (resp.  $l_i$ ) the index of the receiver (resp. emitter)  
 380 coil. The vector  $\mathbf{g} = (g^1, \dots, g^{N_s})$  of gradients is then defined by:

$$381 \quad g^i = \begin{cases} g_{l_i l_i} - g_{k_i k_i} & \text{differential mode} \\ g_{l_i l_i} + g_{k_i k_i} & \text{absolute mode,} \end{cases}$$

382 where, for a given emitter coil  $l$  and receiver coil  $k$ ,

$$383 \quad g_{kl} := \int_{z_{\min}}^{z_{\max}} \Re \left( \overline{(Z(\Omega_d, \zeta) - Z_{\text{meas}}(\zeta))} \left\{ \left[ \frac{1}{\mu} \right] (\mathbf{n} \cdot \mathbf{curl} \mathbf{A}_k) (\mathbf{n} \cdot \mathbf{curl} \bar{\mathbf{P}}_1 - \mathbf{n} \cdot \mathbf{curl} \mathbf{A}_l^0) \right. \right. \\ \left. \left. - [\mu] \left( \frac{1}{\mu} (\mathbf{curl} \mathbf{A}_k) \times \mathbf{n} \right) \cdot \left( \frac{1}{\mu_0} (\mathbf{curl} (\bar{\mathbf{P}}_1)_+) \times \mathbf{n} - \frac{1}{\mu_0} (\mathbf{curl} \mathbf{A}_l^0) \times \mathbf{n} \right) \right. \right. \\ \left. \left. + \frac{[\sigma]}{i\omega} (i\omega (\mathbf{A}_k)_\tau + \nabla_\tau V_k) \cdot \overline{(i\omega (\mathbf{P}_l)_\tau + \nabla_\tau W_l + (\mathbf{E}_l^0)_\tau)} \right\} \Big|_\zeta \right) d\zeta.$$

384 The notation  $\mathbf{A}|_\zeta$  refers to the solution of the forward problem with the source term  
 385 generated by the coils at position  $\zeta$ . For a vector field  $\mathbf{a}$ ,  $\mathbf{a}_\tau = \mathbf{a} - (\mathbf{a} \cdot \mathbf{n})\mathbf{n}$  denotes the  
 386 tangential part of  $\mathbf{a}$  on a surface  $\Gamma$  with normal vector  $\mathbf{n}$ .  $\nabla_\tau V$  denotes the tangential  
 387 part of  $\nabla V$ .

388 If one chooses a descent direction  $\boldsymbol{\theta}$  such that,

$$389 \quad \boldsymbol{\theta} = \gamma \frac{1}{N_s} \sum_{i=1}^{N_s} g^i \mathbf{n} \quad \text{on } \partial\Omega_d,$$

390 where  $\gamma$  is a positive constant sufficiently small, then  $\boldsymbol{\theta}$  is a descent direction for  $\mathcal{J}$ .  
 391 Note that under a level-set approach, the role of  $\gamma$  is replaced by the length  $\Delta t$  of the

392 time interval that we use to convect  $\psi$  between two iterations using (3.2), see Figure 4.  
393

394 In order to solve the convection problem (3.2), from an initial state defined by  
395  $\psi_0$  (where  $\psi_0$  is a level-set encoding  $\partial\Omega_d$ ), the deformation direction  $G$  needs to be  
396 specified for any point in  $D$ . We first define a  $H^1(D)$  shape gradient associated with  
397  $\mathcal{J}$  by considering  $\mathbf{G} \in H^1(D)^3$ , solution of,  $\forall \boldsymbol{\theta} \in H^1(D)^3$ ,

$$398 \quad (3.3) \quad \int_D \left( \sum_{i=1}^3 \alpha \nabla \mathbf{G}_i \cdot \nabla \boldsymbol{\theta}_i + \mathbf{G}_i \boldsymbol{\theta}_i \right) dx = -\mathcal{J}'(\Omega_d)(\boldsymbol{\theta}) = \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{\omega}{I^2} \int_{\partial\Omega_d} (\boldsymbol{\theta} \cdot \mathbf{n}) g^i ds$$

399 where  $\alpha > 0$  is chosen empirically and can be seen as a regularization parameter for  
400 the descent direction. Obviously, taking  $\boldsymbol{\theta} = \mathbf{G}$  provides a descent direction for  $\mathcal{J}$ .  
401 We then set  $G = |\mathbf{G}|$  in  $D$ . The right-hand side of (3.3) requires in principle explicit  
402 identification of  $\partial\Omega_d$  which we would like to avoid during iterations. This is done by  
403 observing that

$$404 \quad \int_{\partial\Omega_d} (\boldsymbol{\theta} \cdot \mathbf{n}) g^i ds = \frac{1}{2} \int_D \nabla(\text{sgn}(\psi_0)) \cdot \boldsymbol{\theta} g^i dx \quad \forall \boldsymbol{\theta} \in H^1(D)^3,$$

405 where the right-hand side is to be understood as a duality product and  $\text{sgn}(\psi_0) :=$   
406  $\psi_0/|\psi_0|$ . The resulting complete inversion algorithm is summarized in Figure 4.

```

1: input:  $N_p \times N_c$  impedance measurements on a  $z$  interval
2:   incident fields ( $\mathbf{E}^0$ ), for each coil at each probe position
3: initialize  $\psi = \psi_0$ , the gradient speed  $G$  and choose  $\Delta t$  and  $\alpha$ 
4: while  $\mathcal{J} > \eta$  do
5:    $\mathcal{J}_0 = \mathcal{J}$ 
6:   compute the gradient  $\mathbf{G}$  on  $D$ 
7:   convect level-set  $\psi$  for a time interval  $\Delta t$  and a deformation speed  $G$ 
8:   use  $\psi(\Delta t)$  to encode the functions  $\sigma$  and  $\mu$ 
9:   solve the forward problem for each probe position and coil
10:  compute  $\mathcal{J}$ 
11:  if  $\mathcal{J} < \mathcal{J}_0$  then
12:    solve the adjoint problem for each probe position and coil
13:    compute gradient  $\mathbf{g}$  for each signal
14:     $\psi_0 = \psi(\Delta t)$ 
15:  else
16:    decrease time-step  $\Delta t = \Delta t/2$ 
17:  end if
18: end while

```

Fig. 4: Reconstruction algorithm

407 As indicated in Figure 4, for one loop iteration, the number of finite element  
408 problems to solve is  $p = N_p \times N_c$  (resp.  $N_p \times N_c/2$ ) for the forward (resp. adjoint)  
409 problem. Note that for the adjoint problem, problems are solved only for the emitters.  
410 We denote by  $n$  the number of degrees of freedom of the problem:  $n = n_A + n_V$ , with  
411  $n_A$  (resp.  $n_V$ ) the number of degrees of freedom for  $\mathbf{A}^s$  (resp.  $V^s$ ).

412 Let us consider the forward problem. Using the scattered field formulation (2.10),  
413 from one problem to another, the sesquilinear form  $\mathcal{A}$  remains the same. Only the

414 right-hand side changes. Solving for all probe positions is equivalent to solving a block  
 415 system of the form:

$$416 \quad (3.4) \quad \begin{pmatrix} \mathbb{M}_{AA} & \mathbb{M}_{AV} \\ \mathbb{M}_{VA} & \mathbb{M}_{VV} \end{pmatrix} \begin{pmatrix} \mathbf{X}_A \\ \mathbf{X}_V \end{pmatrix} = \begin{pmatrix} \mathbf{B}_A \\ \mathbf{B}_V \end{pmatrix},$$

417 where  $\mathbb{M}_{AA} \in \mathcal{M}_{n_A, n_A}(\mathbb{C})$ ,  $\mathbb{M}_{AV} \in \mathcal{M}_{n_A, n_V}(\mathbb{C})$ ,  $\mathbb{M}_{VA} \in \mathcal{M}_{n_V, n_A}(\mathbb{C})$ ,  $\mathbb{M}_{VV} \in$   
 418  $\mathcal{M}_{n_V, n_V}(\mathbb{C})$ ,  $\mathbf{X}_A$  and  $\mathbf{B}_A \in \mathcal{M}_{n_A, p}(\mathbb{C})$ , and  $\mathbf{X}_V$  and  $\mathbf{B}_V \in \mathcal{M}_{n_V, p}(\mathbb{C})$ .

419

420 Depending on the nature of the probe and the length of the tube scanning, the  
 421 number of RHS  $p$  can greatly increase. With the SAX probe containing two coils,  
 422 the number of RHS may remain fairly low. However, using it in ECT may prove  
 423 to be inefficient to reconstruct deposit as it averages the configuration around the  
 424 azimuthal angle. The best strategy would be to use the SMX probe. As a standard  
 425 probe contains 38 coils, the number of RHS increases quite rapidly. Considering that  
 426 a typical scan may span up to 140 positions,  $p$  is expected to exceed 1,000.

427 Note that measures can be taken to reduce the RHS block size for the SMX probe.  
 428 Indeed, as explained before, to generate all signals, one needs to compute the elec-  
 429 tromagnetic potentials for each coil at each probe position, cf. the definition of  $\Delta Z_{kl}$   
 430 in (3.1). However, unlike the SAX probe, the SMX generates only absolute modes.  
 431 Thus, by using the equality  $\Delta Z_{kl} = \Delta Z_{lk}$  for any coil numbers  $k$  and  $l$ , we are able  
 432 bring down the size of the RHS block from  $N_c \times N_p$  to  $N_c/2 \times N_p$  as we only need  
 433 the direct fields of the emitters, that is to say the lower row.

434

435 The choice to implicitly define the deposit shape  $\Omega_d$  with a level-set function  
 436  $\psi$  requires using a fine mesh in the region containing  $\Omega_d$ . Furthermore, due to the  
 437 potential formulation each mesh node inside the conductor region has four unknowns.  
 438 As a result, the total number of degrees of freedom may exceed one million in a  
 439 typical configuration for the forward problem. These constraints combined with the  
 440 high number of RHS motivate the investigation of efficient solution strategies. Being  
 441 able to tackle this task efficiently is critical as it directly impacts the performance of  
 442 the reconstruction algorithm as a whole.

443 **4. Efficient solution strategies.** It was shown in the previous section that  
 444 most of the computational burden of the reconstruction algorithm 4 is the successive  
 445 solutions of linear systems with large number of right-hand sides, see lines 9 and 12.  
 446 Given the size of the discrete problem, exact  $LU$  factorizations are not tractable. We  
 447 therefore use iterative solvers. The main issue is how to efficient handle a large number  
 448 or RHS. It will be shown in particular that block Krylov methods can efficiently  
 449 fix this difficulty. We also propose some adaptations that significantly speedup the  
 450 proposed reconstruction algorithm from section 3. Since this step is the most critical  
 451 in our inversion algorithm and may apply to other large-scale inverse problems, more  
 452 technical details will be provided on some practical implementations.

453 **4.1. Problem formulation.** For simplicity, the notations from (3.4) are cast  
 454 into the following condensed form:

$$455 \quad (4.1) \quad \mathbf{A}\mathbf{X} = \mathbf{B},$$

456 with  $A \in \mathcal{M}_{n, n}(\mathbb{C})$ , and both  $\mathbf{X}$  and  $\mathbf{B} \in \mathcal{M}_{n, p}(\mathbb{C})$ . This is done using PETSc [3,  
 457 4], the linear algebra backend used in our numerical tests, which can convert the

458 coefficient matrix from (3.4) stored using a MatNest, into the more general matrix  
 459 format MatAIJ. As it is common with high-dimensional problems, a first step for  
 460 setting up an efficient solver is the definition of a preconditioner. This preconditioner  
 461 must also be able to deal with multiple right-hand sides efficiently. A commonly  
 462 used method in this context is the restricted additive Schwarz method [7] (RAS) as  
 463 implemented in PETSc. In a distributed-memory parallel context, given a number of  
 464 processes  $N$ , this preconditioner may be written algebraically as:

$$465 \quad (4.2) \quad M^{-1} = \sum_{i=1}^N \tilde{R}_i^T (R_i A R_i^T)^{-1} R_i,$$

466 where  $\{R_i\}_{i=1}^N$  are restriction operators from a global to local vectors on each sub-  
 467 domain (or process), possibly with some overlap.  $\{\tilde{R}_i\}_{i=1}^N$  are similar operators for  
 468 which coefficients on the overlap are set to 0 instead. Readers interested in do-  
 469 main decomposition methods are referred to one of the many available monographs  
 470 on this matter [36, 40, 11]. An appealing feature of domain decomposition meth-  
 471 ods is that the action of local subdomain solvers  $\{(R_i A R_i^T)^{-1}\}_{i=1}^N$  are usually com-  
 472 puted using exact factorizations with libraries such as MUMPS or MKL PARDISO.  
 473 These libraries provide optimized routines for forward eliminations and backward  
 474 substitutions on blocks of multiple vectors. Thus, they are good candidates for our  
 475 solver which has to deal with thousands of right-hand sides. To conclude this sub-  
 476 section, the command line options provided next can be used to setup a PETSc  
 477 preconditioner as defined mathematically above: `-pc.type asm -sub_pc_type lu`  
 478 `-sub_pc_factor_mat_solver_type mkl_pardiso`. The results in this section were  
 479 obtained on Irène, a system composed of 1,656 nodes with two 24-core Intel Xeon  
 480 Platinum 8168 clocked at 2.7 GHz.

481 **4.2. Benchmark of available strategies.** PETSc, through its KSPHPDDM  
 482 [22] interface to HPDDM [23], implements multiple block Krylov methods. In partic-  
 483 ular, the following methods will be considered:

- 484 • standard GMRES [34];
- 485 • standard GCRODR [26];
- 486 • pseudo-block GMRES;
- 487 • pseudo-block GCRODR;
- 488 • block GMRES [15];
- 489 • block GCRODR recalled Figure 7 in order to keep the paper self-contained.

490 Here, standard means that the method is not able to deal with multiple right-hand  
 491 sides available simultaneously. Pseudo-block means that the method is mathemati-  
 492 cally equivalent to the standard one, in the sense that it generates the same Krylov  
 493 subspace, but it fuses similar operations together, e.g., multiple simultaneous sparse  
 494 matrix–vector multiplications become a single sparse matrix–dense matrix multipli-  
 495 cation. Such blocked operations, also used in block Krylov methods, have higher  
 496 arithmetic intensities. In Figure 2 of [23], authors compare the cost of doing multi-  
 497 ple sparse matrix–vector products versus a single sparse matrix–dense matrix product  
 498 and show that it can be up to 400% more efficient with 64 columns. In Figure 6 of [22],  
 499 authors compare the cost of doing multiple forward eliminations and backward substi-  
 500 tutions with a single column vector versus a single forward elimination and backward  
 501 substitution with a block of column vectors and show that it can be up to 250%  
 502 more efficient with 128 columns. In KSPHPDDM,  $QR$  factorizations are computed  
 503 using the CholeskyQR method which was thoroughly studied in [38]. Users are free

504 to manipulate PETSc built-in types or provide their own preconditioned operator,  
 505 but the data structure provided by PETSc and expected by KSPHPDDM assumes  
 506 that blocks of vectors are stored in a standard contiguous column-major dense format  
 507 distributed following a one-dimensional row partitioning among MPI processes. As  
 508 said in the introduction paragraph of this section, block Krylov methods generate  
 509 different subspaces than their standard counterpart. Throughout this section, the re-  
 510 lative convergence tolerance is set to  $10^{-3}$  and the overlapping Schwarz preconditioner  
 511 defined in subsection 4.1 is applied on the right. A restart parameter of size 40 is  
 512 used for standard and pseudo-block methods, and it is set to 30 for block methods,  
 513 which require more memory.

514 **4.2.1. Performance of non-block Krylov methods.** For standard GMRES  
 515 and GCRODR, instead of solving the full system (4.1), we consider only the first  
 516 column of  $\mathbf{B}$  and  $\mathbf{X}$ . Results for the complete block of  $p$  columns may be extrapolated  
 517 by multiplying the timings obtained by  $p$ , since it is expected that the number of  
 518 GMRES and GCRODR iterations will be similar as  $\mathbf{B}$  is traversed column by column.  
 519 These standard solvers could solve the full system, but as highlighted next, they are  
 520 extremely inefficient so it would only be a waste of resources. Indeed, for only the  
 521 first column of  $\mathbf{B}$ , GMRES (resp. GCRODR) converges in 197 (resp. 125) iterations.  
 522 This shows an advantage of such a recycling Krylov method, which also translates  
 523 to runtime: 8.6s against 6.3s. However, these timings are not satisfactory, since by  
 524 extrapolation, it would approximately take 1.8 h (resp. 1.4 h) to solve the full system  
 525 with  $p = 779$  right-hand sides. The command line options provided next can be used  
 526 to setup a PETSc Krylov method as described above: `-ksp_rtol 1e-3 -ksp_pc_side`  
 527 `right -ksp_type hpddm -ksp_gmres_restart 40 -ksp_hpddm_type`  
 528 `gmres`. With GCRODR, five vectors are recycled throughout the restarts. The last  
 529 option has to be switched with `-ksp_hpddm_type gcrodr -ksp_hpddm_recycle 5`.

530 **4.2.2. Performance of (pseudo-)block Krylov methods.** For pseudo-block  
 531 methods, again, it will be shown next that the timings are not satisfactory. Again,  
 532 GCRODR has the edge over GMRES, both in terms of iterates, 130 against 171,  
 533 and in terms of runtime, 20.7 min against 26.3 min. This is a significant improvement  
 534 compared to the standard methods, with approximately a 4x speedup. The previous  
 535 command line options remain unchanged, as HPDDM will by default switch to the  
 536 pseudo-block variants when solving systems with multiple right-hand sides.

537 Eventually, the performance of BGMRES and BGCRODR are compared. Block  
 538 Krylov methods have higher arithmetic intensities and require more involved kernels  
 539 such as block orthogonalizations. They are also more memory demanding, since, for  
 540 example, the block Arnoldi process generates block Hessenberg matrices, whose  $QR$   
 541 factorizations are costlier to compute using Householder reflectors than plain Hessen-  
 542 berg matrices factorized with Givens rotations [16]. For that reason, solving the full  
 543 system (4.1) with  $p = 779$  right-hand sides is not tractable. Instead, the complete  
 544 block of right-hand sides is decomposed into contiguous sub-blocks which are then  
 545 solved in sequence. At the beginning of each new cycle, deflation is performed us-  
 546 ing a tolerance of  $10^{-10}$ . That is, a rank-revealing  $QR$  factorization of the block of  
 547 initial residuals is computed, and the Arnoldi process only iterates on blocks of size  
 548  $i \in \llbracket 1; p \rrbracket$  such that  $R_{ii} \leq 10^{-10} R_{11}$ . This is achieved in PETSc using the option  
 549 `-ksp_matsolve_batch_size  $p'$` , which will then successively solve  $\lfloor \frac{p}{p'} \rfloor$  subsystems  
 550 with at most  $p'$  right-hand sides. The complete set of options now reads `-ksp_rtol`  
 551 `1e-3 -ksp_pc_side right -ksp_type hpddm -ksp_gmres_restart 30 -ksp_hpddm`



552 `_deflation_tol 1e-10 -ksp_matsolve_batch_size p' -ksp_hpddm_type bgmres.`  
 553 Four different values of  $p'$  are used: 390, 195, 98, and 49. This corresponds to respec-  
 554 tively 2, 4, 8, and 16 successive subsystem solves. The number of iterations, summed  
 555 over all subsystem solves, is respectively 42, 101, 263, and 900. Looking at these num-  
 556 bers, the configuration  $p' = 390$  is the most efficient numerically, as expected, since  
 557 it is the one that enlarges the generated Krylov subspace the most per block Arnoldi  
 558 iteration. However, the higher rate of convergence does not translate into faster run-  
 559 times. Indeed, the time to solution for the previous four block sizes is respectively  
 560 4.7 min, 3.9 min, 3.8 min, and 5.6 min. This highlights the fact that one has to care-  
 561 fully pick the number of right-hand sides treated simultaneously. On the one hand,  
 562 the higher this number, the faster the convergence. On the other hand, the lower this  
 563 number, the cheaper block Krylov kernels are, e.g., block orthogonalizations.

564 **4.2.3. The special case of BGCRODR.** BGCRODR has the advantage of  
 565 handling both blocking and recycling. This is of great interest here, since multiple  
 566 solves with the same coefficient matrix  $A$  are performed while traversing all sub-  
 567 blocks of  $\mathbf{B}$ . For one of the two near-optimal configurations with BGMRES,  $p' = 98$ ,  
 568 we instead now switch to BGCRODR. A single basis vector is recycled throughout  
 569 successive solves. However, it is important to keep in mind that a basis vector in the  
 570 block Krylov sense is in practice a set of  $p'$  vectors. This is achieved by replacing  
 571 `-ksp_hpddm_type bgmres` by `-ksp_hpddm_type bgcrodr -ksp_hpddm_recycle 1` in  
 572 the previous set of options. As expected, the number of iterations, summed over all  
 573 subsystem solves, is lowered with respect to BGMRES. It becomes 166 instead of  
 574 263. One could then expect faster timings than with BGMRES, but this is in practice  
 575 not the case. The time to solution is indeed 7.0 min, which is a great deterioration  
 576 of the BGMRES timing: almost 3 min slower. Indeed, BGCRODR solves eigenvalue  
 577 problems at the end of each cycle, see lines 14 and 31 of Figure 7. The dimension of  
 578 these problems scales linearly with  $p'$ , the number of right-hand sides per sub-block.  
 579 The runtime with a single OpenMP thread of the `zgeev` routine as implemented in  
 580 Intel oneAPI is 26.9 s and 314.1 s with sub-blocks of dimensions 49 and 98, respectively.  
 581 Clearly, one can see that these timings are not acceptable for large  $p'$  since the cost of  
 582 this LAPACK routine scales superlinearly with the dimension of the block Hessenberg  
 583 matrix generated by the block Arnoldi process. The decrease of total number of  
 584 iterations does not compensate the high cost of the BGCRODR eigenvalue problems.  
 585 This will be further investigated in the next paragraph.

586 All the obtained results are gathered in Table 1. Results that are extrapolated  
 587 are typeset in gray, just to highlight that the figures may slightly vary if complete but  
 588 wasteful runs were performed instead. Clearly, the use of block Krylov methods is  
 589 highly beneficial for solving efficiently (4.1). The most effective methods, BGMRES  
 590 with block size of 98 or 195, exhibit a near 28x speedup with respect to a standard  
 591 GMRES implementation which does not use either blocking or recycling. For the  
 592 sake of thoroughness, we also report the time needed to setup the restricted additive  
 593 Schwarz preconditioner: 1.6 s. Since the coefficient matrix does not change while  
 594 solving (4.1), the preconditioner is only computed once, so this timing, compared to  
 595 the ones from Table 1, is negligible.

596 **4.3. Increasing the efficiency of recycling block Krylov methods.** Though  
 597 recycling block Krylov methods have been used successfully in the past [33], results  
 598 shown in the previous section are not encouraging. There is at least one explana-  
 599 tion for this discrepancy. Previous studies, e.g., [8, 41], deal with rather moderate  
 600 numbers of right-hand sides, in the hundreds. In the present work, there is one order

Krylov method	# of blocks	# of RHS/block	$\sum(\# \text{ of iterates})$	Time	/RHS	Speedup
GMRES(40)	779	1	153,463	1.8 h	8.3 s	—
GCRODR(40, 5)	779	1	97,375	1.4 h	6.4 s	1.3
P-BGMRES	1	779	171	26.3 min	2.0 s	4.1
P-BGCRODR	1	779	130	20.7 min	1.6 s	5.2
	16	49	900	5.6 min	0.43 s	19.3
BGMRES(30)	8	98	263	3.8 min	0.29 s	28.6
	4	195	101	3.9 min	0.30 s	27.6
	2	390	42	4.7 min	0.36 s	23.0
BGCRODR(30, 1)	8	98	166	7.0 min	0.53 s	15.7

Table 1: Comparison of GMRES, GCRODR, their pseudo-block and block variants, for solving (4.1) on 960 processes using a restricted additive Schwarz preconditioner

601 of magnitude more vectors, in the thousands. Thus, all algebraic operations from  
602 BGCRODR that scale superlinearly with the dimension of the Krylov subspace are  
603 difficult to amortize. Indeed, these operations are often done redundantly by each  
604 process. Similar considerations apply to, for example, GMRES, where Hessenberg  
605 matrices generated by the Arnoldi process are stored redundantly by each process,  
606 at least as implemented in PETSc, Trilinos [20] and more specifically its Belos pack-  
607 age [5], and HPDDM.

608 **4.3.1. Redistribution algorithm for the extraction of information.** In  
609 order to alleviate this severe limitation, we propose to redistribute the standard  
610 (resp. generalized) eigenvalue problem from BGCRODR line 14 (resp. 31) of Figure 7  
611 on a small subset of  $N' < N$  processes. This is achieved using the PETSc option  
612 `-ksp_hpddm_recycle_redistribute  $N'$` . Then, the “small” dense distributed opera-  
613 tors are passed to SLEPc [19], which is used to solve the problem instead of using se-  
614 quential LAPACK routines redundantly. The computed eigenvectors are then broad-  
615 cast to the other  $N - N'$  processes. This redistribution scheme has the advantage that  
616 not all  $N$  processes used for solving (4.1) will be involved in the eigensolves. The par-  
617 allel granularity of this workload is way too fine: solving dense eigenproblems with a  
618 few thousand unknowns on thousands of processes is likely to perform very poorly due  
619 to the very high communication-to-computation ratio. There is also no available com-  
620 putational routine in distributed dense linear algebra libraries such as ScaLAPACK  
621 for nonsymmetric eigenproblems. With SLEPc, we instead use the Krylov–Schur  
622 method [39], coupled either by a shift or a shift-and-invert spectral transformation.  
623 In this spectral transformation, an exact distributed  $LU$  factorization is computed by  
624 Elemental [29] with the following option `-ksp_hpddm_recycle_mat_type elemental`.

625 **4.3.2. Performance of the proposed algorithm.** This strategy is investi-  
626 gated next with a small communicator of size  $N' = 9$ . In the previous section, calls  
627 to LAPACK for recycling information took 4.5 min, which explains why the naive  
628 BGCRODR implementation was not competitive against BGMRES. With this new  
629 distributed strategy, again with  $p' = 98$ , only 6.4 s are spent in `EPSSolve`, SLEPc  
630 computational routine for solving eigenproblems. Even if the recycled information is  
631 now computed iteratively, instead of directly with LAPACK, the overall convergence  
632 of BGCRODR is not impacted, and it still takes 166 iterations to solve all sub-blocks.

633 However, the time to solution is now 2.6 min. This is more competitive than the pre-  
 634 vious BGMRES timing of 3.8 min, and it also makes recycling much more affordable  
 635 than in the naive BGCRODR implementation which converged in 7.0 min. The most  
 636 efficient strategy has a 41x speedup with respect to a standard GMRES implementa-  
 637 tion which does not use blocking. With this efficient BGCRODR implementation, the  
 638 case  $p' = 195$  is also investigated. The time spent in `EPSSolve` now becomes 18.0s,  
 639 so the effect of the number of right-hand sides in the sub-block is clearly highlighted.  
 640 The number of iterations (resp. time to solution) is now 70 (resp. 3.1 min), which is  
 641 indeed less than with BGMRES, but still does not beat BGCRODR with  $p' = 98$ .  
 642 These results are gathered in [Table 2](#), where the first, sixth, seventh, and last row  
 643 of [Table 1](#) are recalled first.

Krylov method	# of blocks	# of RHS/block	Time	/RHS	Speedup
GMRES(40)	779	1	1.8 h	8.3 s	—
BGMRES(30)	8	98	3.8 min	0.29 s	28.6
	4	195	3.9 min	0.30 s	27.6
Naive BGCRODR(30, 1)	8	98	7.0 min	0.53 s	15.7
BGCRODR(30, 1) + $N' = 9$	8	98	2.6 min	0.20 s	41.5
	4	195	3.1 min	0.24 s	34.6

Table 2: Improvements of the proposed method over previous results from [Table 1](#)

644 To summarize this section, in [Table 3](#), the computational cost of the best BGM-  
 645 RES and BGCRODR configurations from [Table 2](#), i.e., the second and fifth lines, is  
 646 broken down by major components. These include: operator and preconditioner ap-  
 647 plications (third and fourth columns), orthogonalizations and  $QR$  factorizations (fifth  
 648 column), and for BGCRODR, computation of recycling information using `EPSSolve`  
 649 (sixth column). The total computational time from the previous table, which includes  
 650 all other miscellaneous costs, is repeated in the last and seventh column. Again, one  
 651 may notice that the decrease in total number of iterates for sub-blocks of dimension  
 652  $p' = 98$  with BGCRODR plus the low cost of recycling information thanks to our  
 653 redistribution scheme allows for a significant improvement over BGMRES.

Krylov method	$\Sigma(\# \text{ of iterates})$	$A$	$M^{-1}$	$\perp$	Recycling	Total
BGMRES(30)	263	58.1 s	104.3 s	60.1 s	—	3.8 min
BGCRODR(30, 1) + $N' = 9$	166	37.8 s	70.4 s	32.2 s	6.5 s	2.6 min

Table 3: Most time-consuming operations of the linear solvers for sub-blocks of dimension  $p' = 96$ . Column  $A$ : multiplication by the operator from (4.1);  $M^{-1}$ : application of the preconditioner from (4.2);  $\perp$ : orthogonalizations and  $QR$  factorizations; Recycling: SLEPc eigenvalue solves for BGCRODR

654 **5. Deposit reconstruction.** In the previous section, we defined block iterative  
 655 solvers which efficiently deal with the forward problem (2.5) for a large number of  
 656 source terms. These strategies are investigated now for the full inverse algorithm.  
 657 Indeed, at each iteration, two block systems have to be solved: one for the forward

658 and one for the adjoint state. Given the large number of right-hand sides and degrees  
 659 of freedom, we expect the resolution of the finite element problems to be the limiting  
 660 factor at each reconstruction iteration. Let us illustrate this point with the following  
 661 test case.

662 We consider synthetic input data, generated numerically. To avoid any bias in the  
 663 measurements, the deposit is explicitly defined in the computational mesh to generate  
 664 the input data. The target shape here is composed of four ellipsoids at angles  $0$ ,  $\pi/2$ ,  
 665  $\pi$ , and  $3\pi/2$ , of  $z$ -radius 3.25 mm,  $r$ -radius 2.5 mm, and  $\theta$ -radius 5 mm. Assume here  
 666 that the probe scans 41 positions in the  $z$ -axis. For the resolution of the different  
 667 systems in [Figure 4](#), the optimal parameters found in the previous section are used,  
 668 i.e., BGCRODR with a single recycled multivector and sub-blocks of size at most 98  
 669 (see the before last line from [Table 2](#)).

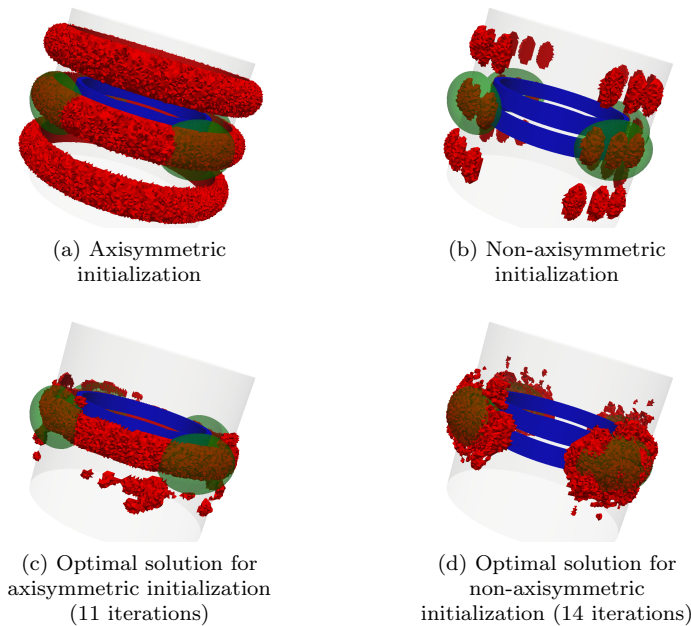


Fig. 5: Convergence results for the SAX probe (in blue) for a target of four ellipsoids (in green) on 960 processes and different shapes of deposits (in red)

670 Let us first recall the algorithm using a SAX probe. It has only two probes and  
 671 generates two signals: one differential and one absolute mode. Hence, there are 82  
 672 source terms for the forward problem for one inversion iteration, and as many terms for  
 673 the adjoint problem. As such, we expect the resolution of the finite element problems  
 674 to be faster than for the SMX probe. We consider two initializations, displayed in  
 675 [Figure 5](#).

676 The convergence figures demonstrate the limits of the SAX probe. As the two  
 677 coils of the probe have the same revolution axis as the tube, the information com-  
 678 puted is averaged on the azimuthal component. As a consequence, when initializing  
 679 the algorithm with an axisymmetric deposit like in [Figure 5a](#), the optimal shape re-  
 680 mains axisymmetric, see [Figure 5c](#), and does not match the target shape. However,  
 681 by choosing a more accurate initialization, e.g., [Figure 5b](#), where we limit the initial-

682 ization on a domain around the target shape, the algorithm reconstructs more valid  
 683 deposits in [Figure 5d](#) i.e. the optimal solution is now non-axisymmetric. Note that  
 684 the optimal solution contains small artefacts due to the initialization that does not  
 685 completely vanish through the convergence as they barely influence the impedance  
 686 signal. They could be removed by adding constraints to the optimization problem,  
 687 for instance surface penalization.

688 In an industrial framework, where little information on the deposit shape can  
 689 be provided a priori, non-axisymmetric initializations shall not be considered as they  
 690 might disregard some solutions. Axisymmetric initializations shall be preferred as  
 691 they do not introduce bias in the target shape.

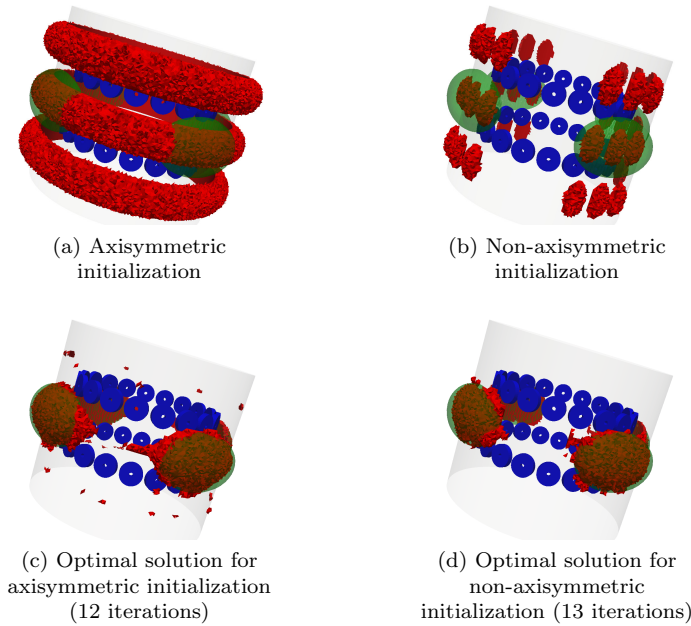


Fig. 6: Convergence results for the SMX probe (in blue) for a target of four ellipsoids (in green) on 960 processes and different shapes of deposits (in red)

692 Let us now compare these SAX reconstruction results with those of the SMX  
 693 probe. The probe is made of 38 coils and generates 76 signals: the emitters are taken  
 694 on the lower row, each emitter being associated with four receivers on both rows. As  
 695 such, the forward problem has 779 source terms, for a problem with approximately  
 696 two million finite element unknowns. These numbers are the same as in [section 4](#).  
 697 The same initializations displayed in [Figure 6](#) are tried with the SMX probe.

698 In terms of performance, the method with SMX (resp. SAX) converged in about  
 699 5 hours (resp. 1 hour and 27 minutes), in 13 iterations, at a rate of about 27 (resp. 8)  
 700 minutes per iteration when the descent direction is accepted, and 7 (resp. 4) minutes  
 701 per iteration when the descent direction is rejected. The rejection of a descent direc-  
 702 tion is faster as there is no computation of the adjoint state or the gradient. [Table 4](#)  
 703 summarizes the computational time of the most time-consuming operations for one  
 704 iteration of the inversion algorithm for the two types of probe.

705 Thanks to the use of block Krylov methods, the time spent in both solves is

706 roughly similar to the time spent in other operations. Compared to the computational  
 707 time of about one hour with standard methods like GMRES or GCRODR, see [Table 1](#),  
 708 this is a substantial improvement.

Type of probe	SAX	SMX
Level-set convection	<b>214 s</b>	2.9 min
Direct solve	14 s	2.6 min
Adjoint solve	14 s	2.6 min
Gradient computation	151 s	<b>7.5 min</b>
Total	436 s	17 min

Table 4: Different operations for one iteration in the inversion algorithm on 960 processes. The most time-consuming operation is typeset in bold

709 When comparing the results with the SAX and SMX probes, it appears quite  
 710 easily that the computations with the former are faster since the number of source  
 711 terms does not exceed a hundred, compared to the 779 RHS of the latter. However,  
 712 when comparing the reconstruction results, it appears that the SMX probe provides  
 713 more interesting information on the target deposit.

714 Note that due to the fast resolution of the finite element problems, the remaining  
 715 limiting operations are the level-set convection and the gradient computation. The  
 716 first operation is currently done sequentially, but can be easily parallelized. For the  
 717 second operation, the gradient is computed directly inside FreeFEM domain-specific  
 718 language. It could be made more efficient by offloading this operation to a specific  
 719 kernel written in a lower-level language, e.g., C++, but this goes beyond the scope of  
 720 this paper.

721 **6. Conclusion.** We proposed a complete efficient strategy to solve realistic ECT  
 722 for the reconstruction of deposits inside SG. Using a classical least squares formulation  
 723 of the inverse problem, the main challenges are:

- 724 • use a formulation of the eddy-current problem that does not require remeshing  
 725 and is independent from the topology of the conductor;
- 726 • use an adapted topological shape optimization method;
- 727 • design an efficient solution strategy that allows a reasonable inversion time.

728 We proposed for the first one the use of a potential formulation combined with rewrit-  
 729 ing the problem in terms of scattered field. For the second point, a level-set method  
 730 is used combined with appropriate regularization of the descent direction. The bot-  
 731 tleneck of the inversion algorithm is the third point where the issue was to efficiently  
 732 handle large-scale problems with a large number of RHS. Domain decomposition-  
 733 preconditioned Krylov methods proved to be a tool of choice in this case. We com-  
 734 pared two different block Krylov algorithms: BGMRES and BGCRODR. We proposed  
 735 for the latter a new redistribution scheme to increase its performance. This part is  
 736 quite general and may be applied to other large-scale inverse problems. In terms  
 737 of deposit reconstructions, though the SAX probe offers less costly computations, it  
 738 may fail to reconstruct properly the deposit. In contrast, the SMX probe contains  
 739 more information and leads to satisfying results. Thanks to block Krylov methods,  
 740 we are able to converge in less than 5 h for a typical industrial problem. Additional  
 741 accelerations of the inversion scheme can be obtained by further optimizing the shape  
 742 convection step or the computation of the gradient with FreeFEM. These issues will

743 be explored in a future work where we would like to apply the inversion scheme on  
 744 experimental data. Reconstructing at the same time the deposit shape, the material  
 745 properties and other possible defects (manufacturing defects, cracks, etc.) is also a  
 746 future perspective of this work where the use of a larger set of data may be needed.

747 **Acknowledgments.** This work was granted access to the GENCI-sponsored  
 748 HPC resources of TGCC@CEA under allocations A0070607519 and A0090607519.

749

## REFERENCES

- 750 [1] G. ALLAIRE, F. DE GOURNAY, F. JOUVE, AND A.-M. TOADER, *Structural optimization using*  
 751 *topological and shape sensitivity via a level-set method*, Control and Cybernetics, 34 (2005),  
 752 p. 59.
- 753 [2] P. AMESTOY, I. DUFF, J.-Y. L'EXCELLENT, AND J. KOSTER, *A fully asynchronous multifrontal*  
 754 *solver using distributed dynamic scheduling*, SIAM Journal on Matrix Analysis and Appli-  
 755 cations, 23 (2001), pp. 15–41, <http://mumps.enseiht.fr>.
- 756 [3] S. BALAY, S. ABHYANKAR, M. F. ADAMS, J. BROWN, P. BRUNE, K. BUSCHELMAN, L. DALCIN,  
 757 A. DENER, V. ELJKHOUT, W. D. GROPP, D. KARPEYEV, D. KAUSHIK, M. G. KNEPLEY,  
 758 D. A. MAY, L. C. MCINNES, R. T. MILLS, T. MUNSON, K. RUPP, P. SANAN, B. F. SMITH,  
 759 S. ZAMPINI, H. ZHANG, AND H. ZHANG, *PETSc users manual*, Tech. Report ANL-95/11 -  
 760 Revision 3.13, Argonne National Laboratory, 2020.
- 761 [4] S. BALAY, S. ABHYANKAR, M. F. ADAMS, J. BROWN, P. BRUNE, K. BUSCHELMAN, L. DALCIN,  
 762 A. DENER, V. ELJKHOUT, W. D. GROPP, D. KARPEYEV, D. KAUSHIK, M. G. KNEPLEY,  
 763 D. A. MAY, L. C. MCINNES, R. T. MILLS, T. MUNSON, K. RUPP, P. SANAN, B. F. SMITH,  
 764 S. ZAMPINI, H. ZHANG, AND H. ZHANG, *PETSc web page*. <http://www.mcs.anl.gov/petsc>,  
 765 2020.
- 766 [5] E. BAVIER, M. HOEMMEN, S. RAJAMANICKAM, AND H. THORNQUIST, *Amesos2 and Belos: direct*  
 767 *and iterative solvers for large sparse linear systems*, Scientific Programming, 20 (2012),  
 768 pp. 241–255.
- 769 [6] C. T. T. BUI, C. DAPOGNY, AND P. FREY, *An accurate anisotropic adaptation method for*  
 770 *solving the level-set advection equation*, International Journal for Numerical Methods in  
 771 Fluids, 70 (2012), pp. 899–922.
- 772 [7] X.-C. CAI AND M. SARKIS, *A restricted additive Schwarz preconditioner for general sparse*  
 773 *linear systems*, SIAM Journal on Scientific Computing, 21 (1999), pp. 792–797.
- 774 [8] H. CALANDRA, S. GRATTON, J. LANGOU, X. PINEL, AND X. VASSEUR, *Flexible variants of*  
 775 *block restarted GMRES methods with application to geophysics*, SIAM Journal on Scientific  
 776 Computing, 34 (2012), pp. A714–A736.
- 777 [9] M. COSTABEL AND M. DAUGE, *Singularities of electromagnetic fields in polyhedral domains*,  
 778 Archive for Rational Mechanics and Analysis, 151 (2000), pp. 221–276.
- 779 [10] C. DAPOGNY, P. FREY, F. OMNÈS, AND Y. PRIVAT, *Geometrical shape optimization in fluid*  
 780 *mechanics using FreeFem++*, Structural and Multidisciplinary Optimization, 58 (2018),  
 781 pp. 2761–2788.
- 782 [11] V. DOLEAN, P. JOLIVET, AND F. NATAF, *An Introduction to Domain Decomposition Methods:*  
 783 *Algorithms, Theory and Parallel Implementation*, SIAM, 2015.
- 784 [12] O. DORN AND D. LESSELIER, *Level-set methods for inverse scattering*, Inverse Problems, 22  
 785 (2006), p. R67.
- 786 [13] T. DUPUY, *Modélisation des transferts thermiques dans les dépôts d'encrassement des*  
 787 *générateurs de vapeur*, PhD thesis, Ecole Centrale de Marseille, 2019.
- 788 [14] J. GARCÍA-MARTÍN, J. GÓMEZ-GIL, AND E. VÁZQUEZ-SÁNCHEZ, *Non-destructive techniques*  
 789 *based on eddy current testing*, Sensors, 11 (2011), pp. 2525–2565.
- 790 [15] M. H. GUTKNECHT, *Block Krylov space methods for linear systems with multiple right-hand*  
 791 *sides: an introduction*, in Modern Mathematical Models, Methods and Algorithms for Real  
 792 World Systems, A. Siddiqui, I. Duff, and O. Christensen, eds., 2006, pp. 420–447.
- 793 [16] M. H. GUTKNECHT AND T. SCHMELZER, *Updating the QR decomposition of block tridiagonal*  
 794 *and block Hessenberg matrices*, Applied Numerical Mathematics, 58 (2008), pp. 871–883.
- 795 [17] H. HADDAR, Z. JIANG, AND M. K. RIAHI, *A robust inversion method for quantitative 3D shape*  
 796 *reconstruction from coaxial eddy current measurements*, Journal of Scientific Computing,  
 797 70 (2017), pp. 29–59.
- 798 [18] F. HECHT, *New development in FreeFem++*, Journal of Numerical Mathematics, 20 (2012),  
 799 pp. 251–266.
- 800 [19] V. HERNANDEZ, J. E. ROMAN, AND V. VIDAL, *SLEPc: a scalable and flexible toolkit for the*

```

1:  $R_0 = B_i - AX_0$ 
2: if  $U_k$  is defined (from solving a previous sub-block) then
3:    $[Q, R] = \text{distributed qr}(AU_k)$ 
4:    $C_k = Q$ 
5:    $U_k = U_k R^{-1}$ 
6:    $X_1 = X_0 + U_k C_k^H R_0$ 
7:    $R_1 = R_0 - C_k C_k^H R_0$ 
8: else
9:    $[V_1, S_1] = \text{distributed qr}(R_0)$ 
10:  perform  $m$  steps of BGMRES, thus generating  $V_{m+1}$  and  $[Q, R] = \text{qr}(\overline{H}_m)$ 
    (Arnoldi basis and Hessenberg matrix)
11:  find  $Y_m$  such that  $RY_m = Q^{-1} \begin{bmatrix} S_1 \\ 0_{p \cdot (m-1) \times p} \end{bmatrix}$ 
12:   $X_1 = X_0 + V_m Y_m$ 
13:   $R_1 = B_i - AX_1$ 
14:  solve  $\begin{pmatrix} H_m + QR^{-H} \begin{bmatrix} 0_{p \cdot (m-1) \times p \cdot (m-1)} & 0_{p \cdot (m-1) \times p} \\ 0_{p \times p \cdot (m-1)} & h_{m+1, m}^H h_{m+1, m} \end{bmatrix} \end{pmatrix} z_\lambda = \theta_\lambda z_\lambda$ 
15:  store the  $k$  eigenvectors  $z_\lambda$  associated to the smallest eigenvalues in magnitude
    in  $P_k$ 
16:   $[Q, R] = \text{qr}(\overline{H}_m P_k)$ 
17:   $C_k = V_{m+1} Q$ 
18:   $U_k = V_m P_k R^{-1}$ 
19: end if
20:  $j = 1$ 
21: while convergence not reached do
22:    $[V_k, S_k] = \text{distributed qr}(R_j)$ 
23:    $j += 1$ 
24:   perform  $m - k$  steps of BGMRES with the linear operator  $(I - C_k C_k^H)A$ , thus
    generating  $V_{m+1-k}$ ,  $[Q, R] = \text{qr}(\overline{H}_{m-k})$ , and  $E_k = C_k A V_{m-k}$ 
25:   find  $Y_{m-k}$  such that  $RY_{m-k} = Q^{-1} \begin{bmatrix} S_k \\ 0_{p \cdot (m-k-1) \times p} \end{bmatrix}$ 
26:    $Y_k = C_k^H R_{j-1} - E_k Y_{m-k}$ 
27:    $X_j = X_{j-1} + U_k Y_k + V_{m-k} Y_{m-k}$ 
28:    $R_j = B_i - AX_j$ 
29:   scale the columns of  $U_k$  so that they are of unit norm and store the scaling
    coefficients in  $D_k$ 
30:   define  $G_m = \begin{bmatrix} D_k & E_k \\ 0_{p \cdot (m-k+1) \times p \cdot k} & \overline{H}_{m-k} \end{bmatrix}$ 
31:   solve  $G_m^H G_m z_\lambda = \theta_\lambda G_m^H \begin{pmatrix} C_k^H U_k & 0_{p \cdot k \times p \cdot (m-k)} \\ V_{m-k+1}^H U_k & I_{p \cdot (m-k+1) \times p \cdot (m-k)} \end{pmatrix} z_\lambda$ 
32:   store the  $k$  eigenvectors  $z_\lambda$  associated to the smallest eigenvalues in magnitude
    in  $P_k$ 
33:    $[Q, R] = \text{qr}(\overline{H}_m P_k)$ 
34:    $C_k = \begin{bmatrix} C_k & V_{m-k+1} \end{bmatrix} Q$ 
35:    $U_k = \begin{bmatrix} U_k P_k & V_{m-k} P_k \end{bmatrix} R^{-1}$ 
36: end while

```

Fig. 7: BGCRODR as written by Jolivet and Tournier [23]



- 801 *solution of eigenvalue problems*, ACM Transactions on Mathematical Software, 31 (2005),  
 802 pp. 351–362, <https://slepc.upv.es>.
- 803 [20] M. A. HEROUX, R. A. BARTLETT, V. E. HOWLE, R. J. HOEKSTRA, J. J. HU, T. G. KOLDA,  
 804 R. B. LEHOUCQ, K. R. LONG, R. P. PAWLOWSKI, E. T. PHIPPS, ET AL., *An overview of*  
 805 *the Trilinos project*, ACM Transactions on Mathematical Software (TOMS), 31 (2005),  
 806 pp. 397–423, <https://trilinos.github.io>.
- 807 [21] Z. JIANG, H. HADDAR, A. LECHLEITER, AND M. EL-GUEDRI, *Identification of magnetic de-*  
 808 *posits in 2D axisymmetric eddy current models via shape optimization*, Inverse Problems  
 809 in Science and Engineering, 24 (2016), pp. 1385–1410.
- 810 [22] P. JOLIVET, J. E. ROMAN, AND S. ZAMPINI, *KSPHPDDM and PCHPDDM: extending PETSc*  
 811 *with robust overlapping Schwarz preconditioners and advanced Krylov methods*, Comput-  
 812 *ers & Mathematics with Applications*, 84 (2021), pp. 277–295, [https://github.com/prj-/](https://github.com/prj-/jolivet2020petsc)  
 813 [jolivet2020petsc](https://github.com/prj-/jolivet2020petsc).
- 814 [23] P. JOLIVET AND P.-H. TOURNIER, *Block iterative methods and recycling for improved scalability*  
 815 *of linear solvers*, in Proceedings of the 2016 International Conference for High Performance  
 816 Computing, Networking, Storage and Analysis, SC16, IEEE, 2016.
- 817 [24] L. MAURICE, V. COSTAN, E. GUILLOT, AND P. THOMAS, *Eddy current NDE performance*  
 818 *demonstrations using simulation tools*, in AIP Conference Proceedings, vol. 1511, American  
 819 Institute of Physics, 2013, pp. 464–471.
- 820 [25] S. OSHER AND R. FEDKIW, *Level-set methods and dynamic implicit surfaces*, Applied Mechanics  
 821 Reviews, 57 (2004), pp. B15–B15.
- 822 [26] M. L. PARKS, E. DE STURLER, G. MACKEY, D. D. JOHNSON, AND S. MAITI, *Recycling Krylov*  
 823 *subspaces for sequences of linear systems*, SIAM Journal on Scientific Computing, 28  
 824 (2006), pp. 1651–1674.
- 825 [27] M. L. PARKS, K. M. SOODHALTER, AND D. B. SZYLD, *A block recycled GMRES method with*  
 826 *investigations into aspects of solver performance*, arXiv:1604.01713, (2016).
- 827 [28] T. PIERRE AND G. BENJAMIN, *La simulation des CND-CF complexes à la portée des ingénieurs*,  
 828 in Proceedings of the 2017 Journées COFREND, Modélisation courants de Foucault,  
 829 COFREND, 2017.
- 830 [29] J. POULSON, B. MARKER, R. A. VAN DE GELIJN, J. R. HAMMOND, AND N. A. ROMERO, *Element-*  
 831 *tal: a new framework for distributed memory dense matrix computations*, ACM Transac-  
 832 *tions on Mathematical Software*, 39 (2013).
- 833 [30] T. PRUSEK, *Modélisation et simulation numérique du colmatage à l'échelle du sous-canal dans*  
 834 *les générateurs de vapeur*, PhD thesis, Aix-Marseille, 2012.
- 835 [31] M. K. RIAHI, *A fast eddy-current non destructive testing finite element solver in steam gener-*  
 836 *ator*, Journal of Coupled Systems and Multiscale Dynamics, 4 (2016), pp. 60–68.
- 837 [32] A. A. RODRÍGUEZ AND A. VALLI, *Eddy current approximation of Maxwell equations: theory,*  
 838 *algorithms and applications*, vol. 4, Springer Science & Business Media, 2010.
- 839 [33] F.-X. ROUX AND A. BARKA, *Block Krylov recycling algorithms for FETI-2LM applied to 3D*  
 840 *electromagnetic wave scattering and radiation*, IEEE Transactions on Antennas and Prop-  
 841 *agation*, 65 (2017), pp. 1886–1895.
- 842 [34] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving*  
 843 *nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7  
 844 (1986), pp. 856–869.
- 845 [35] M. SCHWEIGER, S. R. ARRIDGE, O. DORN, A. ZACHAROPOULOS, AND V. KOLEHMAINEN, *Re-*  
 846 *constructing absorption and diffusion shape profiles in optical tomography by a level-set*  
 847 *technique*, Optics Letters, 31 (2006), pp. 471–473.
- 848 [36] B. F. SMITH, P. BJØRSTAD, AND W. D. GROPP, *Domain decomposition: parallel multilevel*  
 849 *methods for elliptic partial differential equations*, Cambridge University Press, 2004.
- 850 [37] K. M. SOODHALTER, E. DE STURLER, AND M. E. KILMER, *A survey of subspace recycling*  
 851 *iterative methods*, GAMM-Mitteilungen, 43 (2020), p. e202000016.
- 852 [38] A. STATHOPOULOS AND K. WU, *A block orthogonalization procedure with constant synchroniza-*  
 853 *tion requirements*, SIAM Journal on Scientific Computing, 23 (2002), pp. 2165–2182.
- 854 [39] G. W. STEWART, *A Krylov–Schur Algorithm for Large Eigenproblems*, SIAM Journal on Matrix  
 855 *Analysis and Applications*, 23 (2002), pp. 601–614.
- 856 [40] A. TOSELLI AND O. B. WIDLUND, *Domain decomposition methods: algorithms and theory*,  
 857 vol. 34 of Series in Computational Mathematics, Springer, 2005.
- 858 [41] P.-H. TOURNIER, I. ALIFERIS, M. BONAZZOLI, M. DE BUHAN, M. DARBAS, V. DOLEAN,  
 859 F. HECHT, P. JOLIVET, I. EL KANFOUD, C. MIGLIACCIO, F. NATAF, C. PICHOT, AND  
 860 S. SEMENOV, *Microwave tomographic imaging of cerebrovascular accidents by using high-*  
 861 *performance computing*, Parallel Computing, 85 (2019), pp. 88–97.
- 862 [42] M. Y. WANG, X. WANG, AND D. GUO, *A level-set method for structural topology optimization*,

- 863 Computer methods in applied mechanics and engineering, 192 (2003), pp. 227–246.  
864 [43] G. ZENZINGER, J. BAMBERG, W. SATZGER, AND V. CARL, *Thermographic crack detection by*  
865 *eddy current excitation*, *Nondestructive Testing and Evaluation*, 22 (2007), pp. 101–111.