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

Lorenzo Audibert, Hugo Girardon, Houssem Haddar, Pierre Jolivet

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

LORENZO AUDIBERT†, HUGO GIRARDON†‡, HOUSSEM HADDAR‡, AND PIERRE JOLIVET§

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

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

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

1. Introduction.

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

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

*Submitted to the editors 22/11/2020.

†PRISME, EDF R&D, 6 Quai Watier, 78400 Chatou, France

‡INRIA and Ecole Polytechnique, CMAP, Institut Polytechnique de Paris, 91128, Palaiseau, France

§IRIT–ENSEEIHT, CNRS, 2 rue Camichel, 31000 Toulouse, France

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

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

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

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

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

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

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

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

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

2. The forward problem.

2.1. Problem formulation. Let $\Omega \subset \mathbb{R}^3$ be the computational domain of interest inside the SG that is assumed to be with Lipschitz boundary and later will be assumed to be also simply connected with connected boundary and either regular or convex polyhedral. The medium physical parameters, namely the electric permittivity $\varepsilon(\mathbf{x}) > 0$, the conductivity $\sigma(\mathbf{x}) \geq 0$ and the magnetic permeability $\mu(\mathbf{x}) > 0$ are assumed to be piecewise constant functions. Let $\Omega_{\mathcal{C}}$ be the conductive domain, i.e., the region where $\sigma \neq 0$ and $\Omega_{\mathcal{I}} = \Omega \setminus \overline{\Omega_{\mathcal{C}}}$ be the insulator domain. We denote $\Gamma := \partial\Omega_{\mathcal{I}} \cap \partial\Omega_{\mathcal{C}}$ the interface between insulator and conductor domains.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Let us introduce the function spaces

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

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

and set $\mathbf{X}(\Omega) := \mathbf{H}(\mathrm{curl},\Omega) \cap \mathbf{H}_0(\mathrm{div},\Omega)$. Assume that Ω_C has M connected components Ω_C^i, $i = 1,\dots,M$, then we define

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

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

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

$$\begin{aligned} \text{with:}\quad \mathcal{A}((\mathbf{A},V),(\mathbf{B},Q)) &:= \int_\Omega \left[\mu^{-1}\mathbf{curl}\,\mathbf{A}\cdot\mathbf{curl}\,\overline{\mathbf{B}} + \mu_*^{-1}(\mathrm{div}\,\mathbf{A})(\mathrm{div}\,\overline{\mathbf{B}})\right] \mathrm{d}\mathbf{x} \\ &\quad + \frac{1}{i\omega}\int_{\Omega_{\mathcal{C}}} \sigma(i\omega\mathbf{A} + \boldsymbol{\nabla} V)\cdot\overline{(i\omega\mathbf{B} + \boldsymbol{\nabla} Q)}\,\mathrm{d}\mathbf{x}, \\ \mathcal{L}((\mathbf{B},Q)) &:= \int_\Omega \mathbf{J}\cdot\overline{\mathbf{B}}\,\mathrm{d}\mathbf{x}. \end{aligned}$$

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

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

2.2. Scattered field formulation. As explained later, the goal is to reconstruct a deposit that appears in a reference configuration characterized by the physical parameters σ_0 and μ_0, and the associated conductive part $\Omega_{\mathcal{C}}^0$. Let us denote by $(\mathbf{E}^0,\mathbf{H}^0)$ the electromagnetic field associated with this configuration and a source term $\mathbf{J}$. This field verifies in particular

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

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

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

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

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

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

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

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

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

with $\mathbf{J}_0$ defined by

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

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

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

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

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

where the right-hand side can be expressed as

$$\mathcal{L}^s((\mathbf{B}, Q)) := \int_\Omega \left(1 - \frac{\mu_0}{\mu}\right)\mathbf{H}^0 \cdot \mathbf{curl}\,\overline{\mathbf{B}}\,\mathrm{d}\mathbf{x} - \frac{1}{i\omega}\int_{\Omega_{\mathcal{C}}} (\sigma - \sigma_0)\mathbf{E}^0 \cdot \overline{(i\omega\mathbf{B} + \boldsymbol{\nabla} Q)}\,\mathrm{d}\mathbf{x}.$$

As an immediate corollary of Theorem 2.2, we can state the following theorem on the well-posedness of this problem and the equivalence with the original problem.

THEOREM 2.3. *Assume that Ω is simply connected with connected boundary and 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 $(\mathbf{A}^s, V^s) \in \mathbf{X}(\Omega) \times \tilde{H}^1(\Omega_{\mathcal{C}})$. Assume in addition that $(\mathbf{E}^0, \mathbf{H}^0)$ satisfies (2.6) with $\operatorname{div}\mathbf{J} = 0$ in Ω. Then, the solution of (2.10) satisfies (2.9) or equivalently (2.8) and the fields $(\mathbf{E}, \mathbf{H})$ defined by (2.7) verify the first two equations in (2.6).*

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

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

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

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

Since the scalar potential is assumed to a have zero mean value in each connected component of $\Omega_{\mathcal{C}}$, we enforce this condition by adding a penalization of the form $\int_{\Omega_{\mathcal{C}}} \delta_0 \sigma V \overline{Q}\, \mathrm{d}\mathbf{x}$ where $0 < \delta_0 \ll 1$. The value of δ_0 is ~~chosen empirically using calibration on a test case~~ merely used to ensure the zero mean value constraint. It may be chosen arbitrarly low as long as solvers can handle it numerically when solving the corresponding discrete system. To summarize, the discrete system is built by replacing the sesquilinear form $\mathcal{A}$ by

$$\mathcal{A}((\mathbf{A},V),(\mathbf{B},Q)) + \frac{1}{i\omega}\int_{\Omega_{\mathcal{C}}} \delta_0 \sigma V \overline{Q}\, \mathrm{d}\mathbf{x},$$

and replacing the variational space $X(\Omega) \times \tilde{H}^1(\Omega)$ with $\mathbf{X}_h^1(\Omega) \times V_h^1(\Omega_{\mathcal{C}})$. The numerical implementation of the resulting scheme has been done using the finite element library FreeFEM [18].

3. The inverse problem.

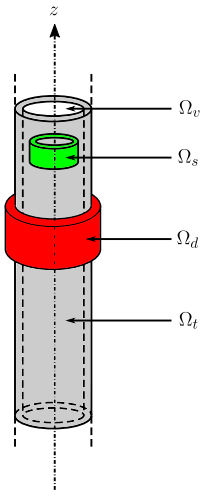


Fig. 1: Domain Ω for the Maxwell equations

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

- Ω_v the vacuum domain inside and outside the tube, with physical parameters $(\sigma_v = 0, \mu_v)$;
- Ω_t the domain occupied by the tube consisting of a conductive material, with physical parameters (σ_t, μ_t);
- Ω_d the deposit domain located in the exterior of the tube, with physical parameters (σ_d, μ_d);
- Ω_s the domain occupied by the probe placed inside the tube.

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

The physical parameters are assumed to be known a priori and the only unknown for the inverse problem is the deposit Ω_d. In this context and referring to the notation of the previous section, the reference media is defined by (with χ_O denoting the characteristic function of a domain O),

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

while

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

(a) SAX probe (b) SMX probe

Fig. 2: Two probes used for ECT

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

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

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

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) and (2.6) where the source term $\mathbf{J}$ is supported by the coil k. Note that from this definition, we have the equality $\Delta Z_{kl} = \Delta Z_{lk}$ for any k and l.

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

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

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

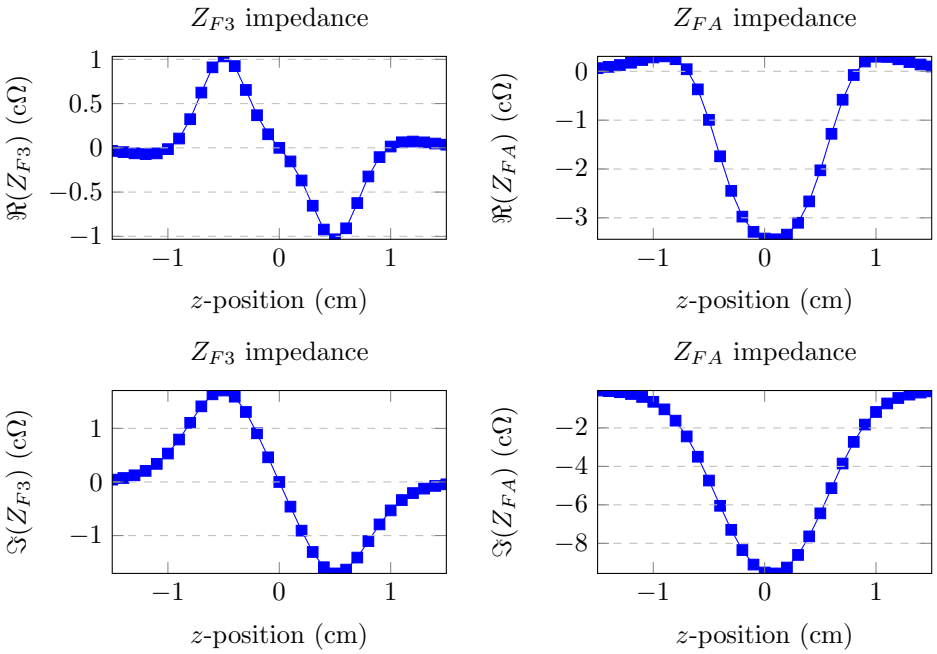


Fig. 3: Example of SAX impedance signals for an annular deposit between $z_- = -0.5\,\mathrm{cm}$ and $z_+ = 0.5\,\mathrm{cm}$, at frequency 100 kHz.

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

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

The SMX probe can also operate at a fourth pulsation $\omega_4 < \omega_3$, though we consider here only ω_3. It generates absolute mode signals according to the following acquisition rule: the emitter coils are contained on the lower row. For each emitter, there are four associated receiver coils, as displayed in Figure 2b. Since the probe has 19 coils on each row, the device provides 76 impedance values for each z-position in this paper.

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

$$\min_{\Omega_d} \left(\mathcal{J}(\Omega_d) = \frac{1}{N_s} \sum_{i=0}^{N_s} \int_{z_-}^{z_+} |\mathbf{Z}^i(\Omega_d, \zeta) - \mathbf{Z}^i_{\text{meas}}(\zeta)|^2 \, \mathrm{d}\zeta \right).$$

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

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

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

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

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

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

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

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

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

$$\left. + \frac{1}{i\omega} (\sigma - \sigma_0) \overline{(i\omega \mathbf{B} + \boldsymbol{\nabla} Q)} \cdot \overline{\mathbf{E}^0} \right) \mathrm{d}\mathbf{x}.$$

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

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

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

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

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

$$g_{kl} := \int_{z_{\min}}^{z_{\max}} \Re \left(\overline{(Z(\Omega_d,\zeta) - Z_{\mathrm{meas}}(\zeta))} \left\{ \left[\frac{1}{\mu} \right] (\mathbf{n} \cdot \mathbf{curl}\, \mathbf{A}_k)(\mathbf{n} \cdot \mathbf{curl}\, \overline{\mathbf{P_l}} - \mathbf{n} \cdot \mathbf{curl}\, \mathbf{A}_l^0) \right. \right.$$

$$- [\mu] \left(\frac{1}{\mu} (\mathbf{curl}\, \mathbf{A}_k) \times \mathbf{n} \right) \cdot \left(\frac{1}{\mu_0} (\mathbf{curl}\, (\overline{\mathbf{P_l}})_+) \times \mathbf{n} - \frac{1}{\mu_0} (\mathbf{curl}\, \mathbf{A}_l^0) \times \mathbf{n} \right)$$

$$\left. \left. + \frac{[\sigma]}{i\omega} (i\omega (\mathbf{A}_k)_\tau + \boldsymbol{\nabla}_\tau V_k) \cdot \overline{(i\omega (\mathbf{P}_l)_\tau + \boldsymbol{\nabla}_\tau W_l + (\mathbf{E}_l^0)_\tau)} \right\} \Big|_\zeta \right) \mathrm{d}\zeta.$$

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

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

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

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

time interval that we use to convect ψ between two iterations using (3.2), see Figure 4.

In order to solve the convection problem (3.2), from an initial state defined by ψ_0 (where ψ_0 is a level-set encoding $\partial\Omega_d$), the deformation direction G needs to be specified for any point in D. We first define a $H^1(D)$ shape gradient associated with $\mathcal{J}$ by considering $\mathbf{G} \in H^1(D)^3$, solution of, $\forall \boldsymbol{\theta} \in H^1(D)^3$,

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

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

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

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

input: $N_p \times N_c$ impedance measurements on a z interval
    incident fields ($\mathbf{E}^0$), for each coil at each probe position
initialize $\psi = \psi_0$, the gradient speed G and choose Δt and α
while $\mathcal{J} > \eta$ **do**
  $\mathcal{J}_0 = \mathcal{J}$
  compute the gradient $\mathbf{G}$ on D
  convect level-set ψ for a time interval Δt and a deformation speed G
  use $\psi(\Delta t)$ to encode the functions σ and μ
  solve the forward problem **for each probe position and coil**
  compute $\mathcal{J}$
  if $\mathcal{J} < \mathcal{J}_0$ **then**
    solve the adjoint problem **for each probe position and coil**
    compute gradient **g for each signal**
    $\psi_0 = \psi(\Delta t)$
  else
    decrease time-step $\Delta t = \Delta t/2$
  end if
end while

Fig. 4: Reconstruction algorithm

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

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

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

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

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

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

Note that measures can be taken to reduce the RHS block size for the SMX probe. Indeed, as explained before, to generate all signals, one needs to compute the electromagnetic potentials for each coil at each probe position, cf. the definition of ΔZ_{kl} in (3.1). However, unlike the SAX probe, the SMX generates only absolute modes. Thus, by using the equality $\Delta Z_{kl} = \Delta Z_{lk}$ for any coil numbers k and l, we are able 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 the direct fields of the emitters, that is to say the lower row.

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

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

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

$$A\mathbf{X} = \mathbf{B}, \tag{4.1}$$

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, 4], the linear algebra backend used in our numerical tests, which can convert the

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

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

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

4.2. Benchmark of available strategies. PETSc, through its KSPHPDDM [22] interface to HPDDM [23], implements multiple block Krylov methods. In particular, the following methods will be considered:

- standard GMRES [34];
- standard GCRODR [26];
- pseudo-block GMRES;
- pseudo-block GCRODR;
- block GMRES [15];
- block GCRODR recalled Figure 7 in order to keep the paper self-contained.

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

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

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

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

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

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

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

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

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

Krylov method	# of blocks	# of RHS/block	$\sum$(# 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
BGMRES(30)	16	49	900	5.6 min	0.43 s	19.3
	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

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

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

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

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

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

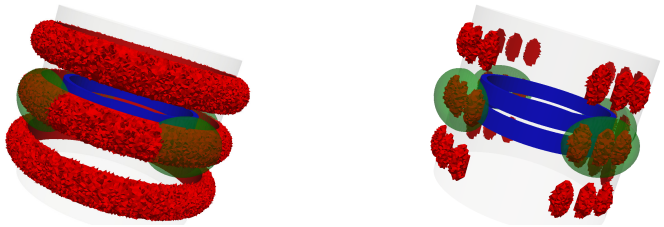
Krylov method	$\sum$(# of iterates)	A	M^{-1}	$\perp$	Recycling	Total
BGMRES(30)	263	58.1 s	104.3 s	60.1 s	—	3.8 min
BGCRODR(30, 1) + $N' = 9$	166	37.8 s	70.4 s	32.2 s	6.5 s	2.6 min

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

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

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

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



(a) Axisymmetric initialization

(b) Non-axisymmetric initialization

(c) Optimal solution for axisymmetric initialization (11 iterations)

(d) Optimal solution for non-axisymmetric initialization (14 iterations)

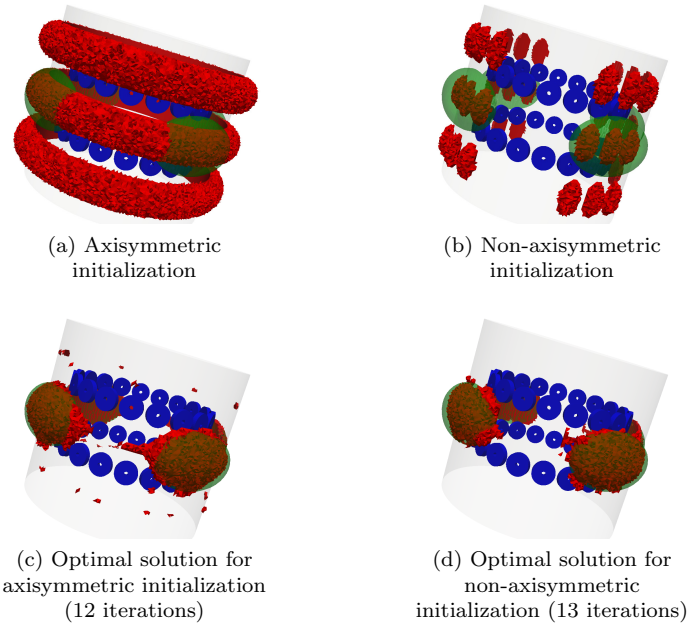
Fig. 5: Convergence results for the SAX probe (in blue) for a target of four ellipsoids (in green) on 960 processes and different shapes of deposits (in red)

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

The convergence figures demonstrate the limits of the SAX probe. As the two coils of the probe have the same revolution axis as the tube, the information computed is averaged on the azimuthal component. As a consequence, when initializing the algorithm with an axisymmetric deposit like in Figure 5a, the optimal shape remains axisymmetric, see Figure 5c, and does not match the target shape. However, by choosing a more accurate initialization, e.g., Figure 5b, where we limit the initial-

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

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



(a) Axisymmetric initialization

(b) Non-axisymmetric initialization

(c) Optimal solution for axisymmetric initialization (12 iterations)

(d) Optimal solution for non-axisymmetric initialization (13 iterations)

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)

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

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

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

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

Type of probe	SAX	SMX
Level-set convection	**214 s**	2.9 min
Direct solve	14 s	2.6 min
Adjoint solve	14 s	2.6 min
Gradient computation	151 s	**7.5 min**
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

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

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

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

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

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

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

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

REFERENCES

[1] G. Allaire, F. De Gournay, F. Jouve, and A.-M. Toader, *Structural optimization using topological and shape sensitivity via a level-set method*, Control and Cybernetics, 34 (2005), p. 59.

[2] P. Amestoy, I. Duff, J.-Y. L'Excellent, and J. Koster, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM Journal on Matrix Analysis and Applications, 23 (2001), pp. 15–41, http://mumps.enseeiht.fr.

[3] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, *PETSc users manual*, Tech. Report ANL-95/11 - Revision 3.13, Argonne National Laboratory, 2020.

[4] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, *PETSc web page*. http://www.mcs.anl.gov/petsc, 2020.

[5] E. Bavier, M. Hoemmen, S. Rajamanickam, and H. Thornquist, *Amesos2 and Belos: direct and iterative solvers for large sparse linear systems*, Scientific Programming, 20 (2012), pp. 241–255.

[6] C. T. T. Bui, C. Dapogny, and P. Frey, *An accurate anisotropic adaptation method for solving the level-set advection equation*, International Journal for Numerical Methods in Fluids, 70 (2012), pp. 899–922.

[7] X.-C. Cai and M. Sarkis, *A restricted additive Schwarz preconditioner for general sparse linear systems*, SIAM Journal on Scientific Computing, 21 (1999), pp. 792–797.

[8] H. Calandra, S. Gratton, J. Langou, X. Pinel, and X. Vasseur, *Flexible variants of block restarted GMRES methods with application to geophysics*, SIAM Journal on Scientific Computing, 34 (2012), pp. A714–A736.

[9] M. Costabel and M. Dauge, *Singularities of electromagnetic fields in polyhedral domains*, Archive for Rational Mechanics and Analysis, 151 (2000), pp. 221–276.

[10] C. Dapogny, P. Frey, F. Omnès, and Y. Privat, *Geometrical shape optimization in fluid mechanics using FreeFem++*, Structural and Multidisciplinary Optimization, 58 (2018), pp. 2761–2788.

[11] V. Dolean, P. Jolivet, and F. Nataf, *An Introduction to Domain Decomposition Methods: Algorithms, Theory and Parallel Implementation*, SIAM, 2015.

[12] O. Dorn and D. Lesselier, *Level-set methods for inverse scattering*, Inverse Problems, 22 (2006), p. R67.

[13] T. Dupuy, *Modélisation des transferts thermiques dans les dépôts d'encrassement des générateurs de vapeur*, PhD thesis, Ecole Centrale de Marseille, 2019.

[14] J. García-Martín, J. Gómez-Gil, and E. Vázquez-Sánchez, *Non-destructive techniques based on eddy current testing*, Sensors, 11 (2011), pp. 2525–2565.

[15] M. H. Gutknecht, *Block Krylov space methods for linear systems with multiple right-hand sides: an introduction*, in Modern Mathematical Models, Methods and Algorithms for Real World Systems, A. Siddiqui, I. Duff, and O. Christensen, eds., 2006, pp. 420–447.

[16] M. H. Gutknecht and T. Schmelzer, *Updating the QR decomposition of block tridiagonal and block Hessenberg matrices*, Applied Numerical Mathematics, 58 (2008), pp. 871–883.

[17] H. Haddar, Z. Jiang, and M. K. Riahi, *A robust inversion method for quantitative 3D shape reconstruction from coaxial eddy current measurements*, Journal of Scientific Computing, 70 (2017), pp. 29–59.

[18] F. Hecht, *New development in FreeFem++*, Journal of Numerical Mathematics, 20 (2012), pp. 251–266.

[19] V. Hernandez, J. E. Roman, and V. Vidal, *SLEPc: a scalable and flexible toolkit for the*

$R_0 = B_i - AX_0$
if U_k is defined (from solving a previous sub-block) **then**
  $[Q, R] =$ distributed qr(AU_k)
  $C_k = Q$
  $U_k = U_k R^{-1}$
  $X_1 = X_0 + U_k C_k^H R_0$
  $R_1 = R_0 - C_k C_k^H R_0$
else
  $[V_1, S_1] =$ distributed qr(R_0)
  perform m steps of BGMRES, thus generating V_{m+1} and $[Q, R] = \text{qr}(\overline{H}_m)$ (Arnoldi basis and Hessenberg matrix)
  find Y_m such that $RY_m = Q^{-1} \begin{bmatrix} S_1 \\ 0_{p\cdot(m-1)\times p} \end{bmatrix}$
  $X_1 = X_0 + V_m Y_m$
  $R_1 = B_i - AX_1$
  solve $\left(H_m + QR^{-H} \begin{bmatrix} 0_{p\cdot(m-1)\times p\cdot(m-1)} & 0_{p\cdot(m-1)\times p} \\ 0_{p\times p\cdot(m-1)} & h_{m+1,m}^H h_{m+1,m} \end{bmatrix} \right) z_\lambda = \theta_\lambda z_\lambda$
  store the k eigenvectors z_λ associated to the smallest eigenvalues in magnitude in P_k
  $[Q, R] = \text{qr}(\overline{H}_m P_k)$
  $C_k = V_{m+1} Q$
  $U_k = V_m P_k R^{-1}$
end if
$j = 1$
while convergence not reached **do**
  $[V_k, S_k] =$ distributed qr(R_j)
  j += 1
  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 $E_k = C_k A V_{m-k}$
  find Y_{m-k} such that $RY_{m-k} = Q^{-1} \begin{bmatrix} S_k \\ 0_{p\cdot(m-k-1)\times p} \end{bmatrix}$
  $Y_k = C_k^H R_{j-1} - E_k Y_{m-k}$
  $X_j = X_{j-1} + U_k Y_k + V_{m-k} Y_{m-k}$
  $R_j = B_i - AX_j$
  scale the columns of U_k so that they are of unit norm and store the scaling coefficients in D_k
  define $G_m = \begin{bmatrix} D_k & E_k \\ 0_{p\cdot(m-k+1)\times p\cdot k} & \overline{H}_{m-k} \end{bmatrix}$
  solve $G_m^H G_m z_\lambda = \theta_\lambda G_m^H \left(\begin{bmatrix} C_k^H U_k & 0_{p\cdot k\times p\cdot(m-k)} \\ V_{m-k+1}^H U_k & I_{p\cdot(m-k+1)\times p\cdot(m-k)} \end{bmatrix} \right) z_\lambda$
  store the k eigenvectors z_λ associated to the smallest eigenvalues in magnitude in P_k
  $[Q, R] = \text{qr}(\overline{H}_m P_k)$
  $C_k = \begin{bmatrix} C_k & V_{m-k+1} \end{bmatrix} Q$
  $U_k = \begin{bmatrix} U_k P_k & V_{m-k} P_k \end{bmatrix} R^{-1}$
end while

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

solution of eigenvalue problems, ACM Transactions on Mathematical Software, 31 (2005), pp. 351–362, https://slepc.upv.es.

[20] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, et al., *An overview of the Trilinos project*, ACM Transactions on Mathematical Software (TOMS), 31 (2005), pp. 397–423, https://trilinos.github.io.

[21] Z. Jiang, H. Haddar, A. Lechleiter, and M. El-Guedri, *Identification of magnetic deposits in 2D axisymmetric eddy current models via shape optimization*, Inverse Problems in Science and Engineering, 24 (2016), pp. 1385–1410.

[22] P. Jolivet, J. E. Roman, and S. Zampini, *KSPHPDDM and PCHPDDM: extending PETSc with robust overlapping Schwarz preconditioners and advanced Krylov methods*, Computers & Mathematics with Applications, 84 (2021), pp. 277–295, https://github.com/prj-/jolivet2020petsc.

[23] P. Jolivet and P.-H. Tournier, *Block iterative methods and recycling for improved scalability of linear solvers*, in Proceedings of the 2016 International Conference for High Performance Computing, Networking, Storage and Analysis, SC16, IEEE, 2016.

[24] L. Maurice, V. Costan, E. Guillot, and P. Thomas, *Eddy current NDE performance demonstrations using simulation tools*, in AIP Conference Proceedings, vol. 1511, American Institute of Physics, 2013, pp. 464–471.

[25] S. Osher and R. Fedkiw, *Level-set methods and dynamic implicit surfaces*, Applied Mechanics Reviews, 57 (2004), pp. B15–B15.

[26] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM Journal on Scientific Computing, 28 (2006), pp. 1651–1674.

[27] M. L. Parks, K. M. Soodhalter, and D. B. Szyld, *A block recycled GMRES method with investigations into aspects of solver performance*, arXiv:1604.01713, (2016).

[28] T. Pierre and G. Benjamin, *La simulation des CND-CF complexes à la portée des ingénieurs*, in Proceedings of the 2017 Journées COFREND, Modélisation courants de Foucault, COFREND, 2017.

[29] J. Poulson, B. Marker, R. A. Van de Geijn, J. R. Hammond, and N. A. Romero, *Elemental: a new framework for distributed memory dense matrix computations*, ACM Transactions on Mathematical Software, 39 (2013).

[30] T. Prusek, *Modélisation et simulation numérique du colmatage à l'échelle du sous-canal dans les générateurs de vapeur*, PhD thesis, Aix-Marseille, 2012.

[31] M. K. Riahi, *A fast eddy-current non destructive testing finite element solver in steam generator*, Journal of Coupled Systems and Multiscale Dynamics, 4 (2016), pp. 60–68.

[32] A. A. Rodríguez and A. Valli, *Eddy current approximation of Maxwell equations: theory, algorithms and applications*, vol. 4, Springer Science & Business Media, 2010.

[33] F.-X. Roux and A. Barka, *Block Krylov recycling algorithms for FETI-2LM applied to 3D electromagnetic wave scattering and radiation*, IEEE Transactions on Antennas and Propagation, 65 (2017), pp. 1886–1895.

[34] Y. Saad and M. H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856–869.

[35] M. Schweiger, S. R. Arridge, O. Dorn, A. Zacharopoulos, and V. Kolehmainen, *Reconstructing absorption and diffusion shape profiles in optical tomography by a level-set technique*, Optics Letters, 31 (2006), pp. 471–473.

[36] B. F. Smith, P. Bjørstad, and W. D. Gropp, *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, 2004.

[37] K. M. Soodhalter, E. de Sturler, and M. E. Kilmer, *A survey of subspace recycling iterative methods*, GAMM-Mitteilungen, 43 (2020), p. e202000016.

[38] A. Stathopoulos and K. Wu, *A block orthogonalization procedure with constant synchronization requirements*, SIAM Journal on Scientific Computing, 23 (2002), pp. 2165–2182.

[39] G. W. Stewart, *A Krylov–Schur Algorithm for Large Eigenproblems*, SIAM Journal on Matrix Analysis and Applications, 23 (2002), pp. 601–614.

[40] A. Toselli and O. B. Widlund, *Domain decomposition methods: algorithms and theory*, vol. 34 of Series in Computational Mathematics, Springer, 2005.

[41] P.-H. Tournier, I. Aliferis, M. Bonazzoli, M. de Buhan, M. Darbas, V. Dolean, F. Hecht, P. Jolivet, I. El Kanfoud, C. Migliaccio, F. Nataf, C. Pichot, and S. Semenov, *Microwave tomographic imaging of cerebrovascular accidents by using high-performance computing*, Parallel Computing, 85 (2019), pp. 88–97.

[42] M. Y. Wang, X. Wang, and D. Guo, *A level-set method for structural topology optimization*,

Computer methods in applied mechanics and engineering, 192 (2003), pp. 227–246.
[43] G. Zenzinger, J. Bamberg, W. Satzger, and V. Carl, *Thermographic crack detection by eddy current excitation*, Nondestructive Testing and Evaluation, 22 (2007), pp. 101–111.