



HAL
open science

Physics-driven inverse problems made tractable with cosparse regularization

Srđan Kitić, Laurent Albera, Nancy Bertin, Rémi Gribonval

► **To cite this version:**

Srđan Kitić, Laurent Albera, Nancy Bertin, Rémi Gribonval. Physics-driven inverse problems made tractable with cosparse regularization. *IEEE Transactions on Signal Processing*, 2015, pp.27. hal-01133087v2

HAL Id: hal-01133087

<https://inria.hal.science/hal-01133087v2>

Submitted on 25 Jun 2015 (v2), last revised 21 Aug 2015 (v3)

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.

Physics-driven inverse problems made tractable with cosparse regularization

S. Kitić^{1,3} *Student member, IEEE*, L. Albera^{1,2,3} *Senior member, IEEE*, N. Bertin⁴ *Member, IEEE*,
and R. Gribonval¹, *Fellow, IEEE*

Abstract

Sparse data models are powerful tools for solving ill-posed inverse problems. We present a regularization framework based on the sparse synthesis and sparse analysis models for problems governed by linear partial differential equations. Although nominally equivalent, we show that the two models differ substantially from a computational perspective: unlike the sparse synthesis model, its analysis counterpart has much better scaling capabilities and can indeed be faster when more measurement data is available. Our findings are illustrated on two examples, sound source localization and brain source localization, which also serve as showcases for the regularization framework. To address this type of inverse problems, we develop a specially tailored convex optimization algorithm based on the Alternating Direction Method of Multipliers.

Index Terms

sparsity, cosparsity, source localization, inverse problems, acoustic wave equation, Poisson's equation.

I. INTRODUCTION

Inverse problems are ubiquitous in signal processing: source localization for robotics [1], radar imaging [2], image processing [3], acoustic imaging and tomography for oceanography [4], medical imaging [5] or compressed sensing [6], to cite only a few. In their most general formulation, linear inverse problems can be expressed as the problem of recovering a signal $\mathbf{x} \in \mathbb{R}^n$ from measurements $\mathbf{y} \in \mathbb{R}^m$:

$$\mathbf{y} \approx \mathbf{M}\mathbf{x}, \quad (1)$$

where $\mathbf{M} \in \mathbb{R}^{m \times n}$ is a transfer matrix modeling the signal acquisition process. The problem is well-posed if it has a unique solution, which depends continuously on the data [7], otherwise, the problem is termed *ill-posed*. It is no surprise that ill-posed problems often arise when one tries to infer an information about a physical phenomenon based on some indirect observations. To remedy the ill-posedness of a problem, one needs to embed additional

¹Inria, Centre Inria Rennes - Bretagne Atlantique, France.

²Inserm, U1099, Rennes, France.

³Université de Rennes 1, Rennes, France.

⁴IRISA - CNRS UMR 6074, Rennes, France.

knowledge in the problem formulation, in order to *regularize* it. This is often done by introducing some assumptions on the estimated signal, which is known as imposing a *data model*. In particular, *sparse* data models have emerged as a pervasive and invaluable tool when addressing difficult signal estimation problems.

The seminal paper of Mallat and Zhang [8] marked the beginning of “the sparse revolution” in mathematical signal processing. The idea behind this type of regularization is based on the interpretation of the *Occam’s razor* principle: the underlying data model should not be more complicated than necessary to accurately represent the observations. Essentially, the sparse framework relies on the premise that the signal in question has some property allowing us to design a sparse data model. This premise is most commonly expressed through the *sparse synthesis* model. It states that the signal in question ($\mathbf{x} \in \mathbb{R}^n$) is constructed by a linear combination of a few column vectors (*atoms*) taken from a large matrix called *the dictionary*, denoted here by $\mathbf{D} \in \mathbb{R}^{n \times d}$. More particularly, we have:

$$\mathbf{x} = \mathbf{D}\mathbf{s}, \quad (2)$$

where the coefficient vector $\mathbf{s} \in \mathbb{R}^d$, which contains (relatively to its size) very few non-zero elements, is called a *sparse representation* of the signal \mathbf{x} .

The “twin” of the sparse synthesis model is *the sparse analysis* or *cosparse* data model [9]. While the sparse synthesis data model has proven to be extremely useful in regularizing a wide spectrum of ill-posed inverse problems, the cosparse data model has just recently gained the attention of the scientific community [9]–[11]. For this model, we assume that there exists an analysis operator $\mathbf{A} \in \mathbb{R}^{b \times n}$ ($b \geq n$), such that the following *analysis representation*:

$$\mathbf{z} = \mathbf{A}\mathbf{x} \quad (3)$$

of the signal \mathbf{x} is sparse. A signal \mathbf{x} whose analysis representation $\mathbf{z} = \mathbf{A}\mathbf{x} \in \mathbb{R}^b$ contains ℓ zero elements is said to be ℓ -cosparse, and the index set of the zero entries in \mathbf{z} , corresponding to the rows of \mathbf{A} that are orthogonal to \mathbf{x} , is called its *cosupport*. The analysis and the synthesis sparse models are nominally equivalent in only one special case: when $\mathbf{A} = \mathbf{D}^{-1}$ (analogously: $\mathbf{z} = \mathbf{s}$), *i.e.* the analysis operator and the dictionary are nonsingular matrices [10].

A primary issue when dealing with a particular instance of a linear inverse problem is to choose an appropriate model to regularize it. For the sparse synthesis model, this amounts to choosing the dictionary \mathbf{D} ; for the cosparse analysis model, this consists in choosing the analysis operator \mathbf{A} . A first objective in this paper is to highlight the fact that in many settings one can leverage existing knowledge encoded with known *Partial Differential Equations (PDEs)*—such as those governing underlying physical phenomena— to design these models. For the synthesis model, this is traditionally done by computing Green’s functions and gathering them as the atoms of a dictionary \mathbf{D} [12]–[16]. In parallel, it has been recently observed in an acoustic scenario [17]–[19] that the cosparse model can very naturally incorporate the underlying physics through an analysis operator \mathbf{A} resulting from a direct discretization of the wave equation. We extend this observation to a more general class of PDEs and propose a framework for regularization of physics-driven inverse problems.

The second contribution of this paper is to demonstrate, from a computational perspective, significant differences between the synthesis and analysis approaches. Through two showcase applications we show that:

- the sparse analysis model has scaling abilities orders of magnitude better than the synthesis one;
- regularization by the sparse analysis model can be actually speeded up when more measurement data is available; this phenomenon does not seem to occur with the plain sparse synthesis model.

In order to address these computational issues, we i) provide a detailed description of an optimization method based on the Alternating Direction Method of Multipliers (ADMM) [20] framework and ii) consider two particular linear inverse problems: sound source localization and brain source localization [12], [21], [22]. The two problems impose different conceptual and computational requirements, thus serving as showcases on how to appropriately regularize and efficiently optimize the problem at hand.

This work unifies and substantially extends previous contributions published in conference papers [19], [23], [24]. It includes new insights about problem modeling, algorithmic design, computational aspects, and a wider experimental validation. From the optimization point of view, physics-driven cosparsity can also be seen as a preconditioned sparse synthesis approach, while for some other specific aspects, the studied regularization framework relates with data assimilation.

The paper is structured as follows: Section II introduces the notations to be used throughout the paper. In Section III we define the (general) inverse problems of our interest, and present the regularization framework. Section IV describes the two illustrative source localization applications. The optimization method used to solve regularized problems is presented in Section V together with an analysis of its computational aspects. Numerical experiments are presented in Section VI. The last section gathers final remarks and future directions.

II. NOTATIONS

In the following text, lowercase Greek symbols generally denote scalar constants (*e.g.* γ ; the exception are the symbol “ δ ” which is the standard notation for Dirac’s impulse distribution). Real-valued scalar variables are denoted by lowercase italic symbols (*e.g.* c), while a lowercase regular font (*e.g.* i) is used for integer-valued variables. Vectors are denoted with the lowercase regular bold font (*e.g.* \mathbf{v}), while uppercase regular bold (*e.g.* \mathbf{M}) is used for matrices. By uppercase italic letters we denote functionals (*e.g.* F), and uppercase regular Greek letters (*e.g.* Ω) denote sets. The set of real numbers is \mathbb{R} .

III. PDE-DRIVEN (CO)SPARSE REGULARIZATION

Many signals of interest are physical quantities obeying certain known physical laws. For example, sound propagates according to the acoustic wave equation, Magnetic Resonance Imaging (MRI) is based on Bloch’s equations, Maxwell’s equations are at the foundation of wireless communications etc. It is also well-known that PDEs are ubiquitous in mathematical models of physical laws. In the context of ill-posed inverse problems, the knowledge that a signal satisfies some PDE is a strong prior information that could be useful to perform regularization. More specifically, we are interested in coupling the (co)sparse data model with PDE-governed signal models.

In this study, we are interested in signals modeled by *linear PDEs*, which take the following form:

$$\sum_{|\mathbf{k}| \leq \zeta} A_{\mathbf{k}}(\mathbf{w}) D^{\mathbf{k}} X(\mathbf{w}) = Z(\mathbf{w}), \quad \mathbf{w} \in \Omega \quad (4)$$

where $A_{\mathbf{k}}$, X and Z are functions of the vector variable \mathbf{w} (e.g., space and/or time) in the domain Ω , and \mathbf{k} is the *multiindex* variable with $|\mathbf{k}| = k_1 + \dots + k_n, k_i \in \mathbb{N}_0$.

For a given $\mathbf{k} = (k_1, \dots, k_n)$, $D^{\mathbf{k}}X(\mathbf{w})$ denotes the \mathbf{k}^{th} partial differential of X with respect to \mathbf{w} , defined as:

$$D^{\mathbf{k}}X(\mathbf{w}) = \frac{\partial^{|\mathbf{k}|} X}{\partial w_1^{k_1} \partial w_2^{k_2} \dots \partial w_n^{k_n}}.$$

The boundary conditions can be, e.g., of standard *Dirichlet*, *Neumann*, *Cauchy*, *Robin* or *mixed* type [25].

The right-hand side $Z(\mathbf{w})$ can often be physically interpreted as a “*source*”, a “*sink*”, or some type of singularity, and there is often side information indicating that there are relatively few locations in the domain Ω where $Z(\mathbf{w})$ is nonzero. This crucial fact is the prior information we intend to use for regularizing the ill-posedness of an inverse problem using a form of sparsity.

A. PDE-driven sparse regularization.

Green’s functions can be seen as impulse responses of a linear system defined by a PDE. In other words, they are the solutions $X_{\mathbf{v}}(\mathbf{w})$ of (4) when the right hand side is the Dirac delta $Z(\mathbf{w}) = \delta_{\mathbf{v}}(\mathbf{w})$, for each $\mathbf{v} \in \Omega$. By the superposition principle, knowledge of Green’s functions suffices to solve a PDE for any right hand side Z .

The atoms of the dictionary \mathbf{D} are then the discretized versions of Green’s functions. They are computed either analytically for simple configurations (e.g., for the free field wave equation) or (more often) numerically, depending on the PDE type, the domain Ω , and the initial and boundary conditions.

The goal of sparse regularization is to recover the discretized right hand side \mathbf{s} by solving the optimization problem¹

$$\underset{\mathbf{s}}{\text{minimize}} F_1(\mathbf{s}) + F_2(\mathbf{MDs} - \mathbf{y}), \quad (5)$$

where $F_1(\cdot)$ is a sparsity-inducing penalty, e.g. the ℓ_1 -norm [9], [26] (see Appendix B for more details, especially about structured norms) and $F_2(\cdot)$ is a measure of data-fidelity. One example is the indicator function $F_2(\cdot) = \mathbf{1}_{\ell_2 \leq \varepsilon}(\cdot)$ of the set of vectors whose norm is bounded by ε :

$$\mathbf{1}_{\ell_2 \leq \varepsilon}(\mathbf{v}) := \begin{cases} 0 & \|\mathbf{v}\|_2 \leq \varepsilon, \\ +\infty & \text{otherwise.} \end{cases} \quad (6)$$

B. PDE-driven cosparse regularization.

Cosparse regularization can be inferred directly from (4), without resorting to Green’s functions, by discretizing the PDE itself. In a way, the linear operator \mathbf{A} “encodes” the underlying PDE. Assuming that the discretized vector \mathbf{x} is cosparse, one can envision addressing the inverse problem through the following optimization problem:

$$\underset{\mathbf{x}}{\text{minimize}} F_1(\mathbf{Ax}) + F_2(\mathbf{Mx} - \mathbf{y}). \quad (7)$$

The functional F_2 , as in the sparse synthesis case, can encode different types of data-fidelities.

¹ \mathbf{Ds} can be seen as the discretized version of a superposition integral.

There are many approaches to discretize the PDE, such as the Finite Difference Method (FDM) or the Finite Element Method (FEM). The choice of the discretization method is determined by the particular problem in hand. For the sake of simplicity, the illustrations in this paper rely on the FDM.

For well-posed PDEs, the analysis operator \mathbf{A} and the dictionary \mathbf{D} are inverse to each other. The discretized operator \mathbf{A} , which encodes the boundary conditions too, is square and invertible. Its inverse $\mathbf{D} = \mathbf{A}^{-1}$ is a dictionary associated to a twin synthesis model. Therefore, the two problems are nominally equivalent ($\mathbf{A}\mathbf{x} = \mathbf{z} = \mathbf{s}$). However, as we will see in Section VI, their computational aspects can be radically different. In a way, the cosparsity formulation can be seen as an efficient pre-conditioning of the sparse one.

Eventually, (7) is reminiscent to a general formulation of *data assimilation* [27]. However, through a sparsity-inducing penalty F_1 , in (7) we encourage cosparsity, whilst traditional data assimilation minimizes some sum-of-squares objective. Moreover, data assimilation usually considers streaming data, whereas we assume that the vector \mathbf{y} is readily available for batch optimization.

C. Encoding additional knowledge through penalties

In many settings, it is known beforehand which region Ω_1 of the domain Ω may contain ‘‘sources’’ and which region Ω_2 is source-free. Consequently, in the synthesis context, regarding the minimization of $F_1(\mathbf{z})$, one can envision a separable problem of the form $F_1(\mathbf{z}_{\Omega_1}) + F_3(\mathbf{z}_{\Omega_2})$, where \mathbf{z}_{Ω_1} and \mathbf{z}_{Ω_2} denote the corresponding subvectors of \mathbf{z} . The $F_3(\cdot)$ penalty can be, for instance, another indicator function $\mathbf{1}_{\ell_2 \leq \sigma}(\cdot)$. Accordingly, the rows of the discretized analysis operator \mathbf{A} can be split into \mathbf{A}_{Ω_1} and \mathbf{A}_{Ω_2} . Taking this into account leads to an equivalent sparse analysis problem of the form $F_1(\mathbf{A}_{\Omega_1}\mathbf{x}) + F_3(\mathbf{A}_{\Omega_2}\mathbf{x})$.

Other variants can be envisioned to encode other types of prior knowledge at different levels of precision. In the same manner, the framework can be extended to account for multiple constraints, by taking F_3 to be the sum of convex functionals $F_3 = \sum_{i=3}^f F_i$ (the same holds for the optimization algorithm described in section V).

D. General optimization framework

Once equipped with penalties that appropriately reflect available prior knowledge, the optimization problems corresponding to sparse and cosparsity regularization read as:

$$\underset{\mathbf{z}}{\text{minimize}} F_1(\mathbf{z}) + F_2(\mathbf{M}\mathbf{D}\mathbf{z} - \mathbf{y}) + F_3(\mathbf{C}\mathbf{D}\mathbf{z} - \mathbf{c}), \quad (8)$$

$$\underset{\mathbf{x}}{\text{minimize}} F_1(\mathbf{A}\mathbf{x}) + F_2(\mathbf{M}\mathbf{x} - \mathbf{y}) + F_3(\mathbf{C}\mathbf{x} - \mathbf{c}). \quad (9)$$

Here F_1 is an objective, while F_2 and F_3 are the (extended-valued²) penalty functionals for the measurements and additional problem constraints, respectively. We will see in Section V how to solve these optimization problems with a variant of the Alternating Direction of Multipliers Method (ADMM).

²Allowed to take $+\infty$ values to encode hard constraints.

IV. TWO WORKED EXAMPLES

To demonstrate the usage of the above PDE-driven (co)sparse regularization framework, we now present two source localization problems arising in different contexts: acoustics, and human electrophysiology. In both cases, the observation process is modeled by $\mathbf{y} = \mathbf{M}\mathbf{x}$ where \mathbf{M} is a simple spatial sub-sampling operator. More particularly, this amounts to assuming that the microphones/electrodes are placed in fixed positions in space (in the discrete setting, this is equivalent to an appropriate row-reduced identity matrix \mathbf{M}).

A. First example: regularization with the wave equation

Acoustic source localization. Acoustic source localization is, in general, the problem of determining the spatial coordinates of sound source(s) based on microphone recordings. It is an inverse problem, since the goal is to infer locations given the measurements of the sound field. This problem arises in different fields, such as speech and sound enhancement [21], [28], speech recognition [29], acoustic tomography [4], robotics [1] and aeroacoustics [30].

To address this problem, a common family of approaches is based on *Time Difference of Arrival (TDOA)*, i.e., the exploitation of the time offset between recordings [31]. However, such approaches usually only provide the direction of arrival of the sound sources. In contrast, the approaches considered here aim at estimating their full spatial position.

Physical modeling. Sound waves propagate according to the *acoustic wave equation* [32]:

$$\Delta P(\mathbf{r}, t) - \frac{1}{C(\mathbf{r}, t)^2} \frac{\partial^2 P(\mathbf{r}, t)}{\partial t^2} = \sum_{j=1}^g Z_j(t) \delta(\mathbf{r} - \mathbf{r}_j) \quad (10)$$

where $P(\mathbf{r}, t)$ denotes the sound pressure at spatial position \mathbf{r} and time t , in some spatial domain Ω and temporal domain $[t_0, t_1]$. The functional $C(\mathbf{r}, t)$ represents the speed of sound and the functional $Z_j(t)\delta(\mathbf{r} - \mathbf{r}_j)$ is the *source signal* at the same position and time. The pair (\mathbf{r}_j, t) denotes the spatio-temporal coordinate of the j^{th} source, out of a total of g sources. The sound speed C is temperature dependent, but it is usually assumed to be around 343ms^{-1} in perfect gases at room temperature. The information acquired by the microphones is limited by the emission and recording durations, an effect which needs to be taken into account.

Boundary conditions and initial conditions. Eq. (10) holds in the interior of domain Ω and must be accompanied with appropriate initial and boundary conditions so that the acoustic wave equation determines a unique sound pressure field. Since it is a second-order PDE, the initial conditions require both the pressure value and its time derivative. Hence, the initial condition typically corresponds to a *Cauchy boundary condition* at $t = 0$. For the spatial boundary conditions, we use the model introduced in [33], based on the impedance boundary conditions:

$$\frac{\partial P(\mathbf{r}, t)}{\partial t} = -C(\mathbf{r}, t)E(\mathbf{r})\nabla P(\mathbf{r}, t) \cdot \mathbf{n} \quad (11)$$

where $\mathbf{r} \in \partial\Omega$ and \mathbf{n} denotes the normal vector. The model assumes so-called *locally reacting surfaces*, and is physically justified if the boundary structure does not allow propagation parallel to the boundary surface [32]. The specific acoustic impedance coefficient $E(\mathbf{r})$ is generally frequency-dependent, but for the purpose of this study we will assume frequency-independent boundaries ($E(\mathbf{r})$ is constant over all frequencies). This formulation is general

enough to approximate some other boundary conditions - *e.g.* if $E(\mathbf{r})$ is very large, the equation models a very rigid boundary (usually interpreted in terms of the Neumann boundary condition).

Discretization. To discretize the wave equation and yield the matrix \mathbf{A} , we use the Finite Difference Time Domain (FDTD) Standard Leap Frog (SLF) method [17], [19], [34], which corresponds to second-order centered finite differences in space and time (briefly discussed in appendix A). After discretization, and including boundary and initial conditions, we obtain a full-rank, square invertible matrix operator $\mathbf{A} \in \mathbb{R}^{st \times st}$, where s and t are the numbers of points used to discretize space and time, respectively. The matrix operator \mathbf{A} is a banded lower triangular matrix. Moreover, the matrix \mathbf{A} is very sparse, as it can have only a very limited number of non-zeros per row (*e.g.* maximum seven in the 2D case).

Analogously, the discretized spatio-temporal pressure field $\mathbf{x} \in \mathbb{R}^{st}$ and the discretized spatio-temporal source component $\mathbf{z} \in \mathbb{R}^{st}$ are built by vectorization and sequential concatenation of t corresponding s -dimensional vector fields.

Incorporating additional knowledge or constraints. Knowledge about the targeted sparse signal can help choose the sparsity measure F_1 . The most common convex relaxation of the non-convex ℓ_0 objective is its substitution by the ℓ_1 -norm, which is known to promote generally sparse solutions. If we assume spatial stationarity of the sources (for sufficiently short acquisition time, the sources are assumed to keep fixed positions in space), this usual norm can be replaced by a so-called *mixed norm*, such as the $\ell_{2,1}$ -norm, in order to favor solutions for which all temporal slices of the sparse estimate have the same support (also known as *jointly sparse* vectors). In the forthcoming experimental section, we use the standard ℓ_1 , joint $\ell_{2,1}$, and a so-called hierarchical $\ell_{2,1}$ norm (more details on these structured norms are given in appendix B).

Analysis versus synthesis. Here, as previously described, the dictionary of Green’s functions is the inverse of the analysis operator $\mathbf{D} = \mathbf{A}^{-1}$. This equivalence is not surprising, since we are dealing with a well-posed PDE. However, given the structure of the matrix \mathbf{A} , the dictionary \mathbf{D} is *not* sparse, even though it is also a lower triangular matrix. This becomes obvious if one rewrites the discretization provided in the Appendix A in the causal (explicit) form. Then, the columns of the dictionary \mathbf{D} are simply the truncated impulse responses of an infinite impulse response filter. It is important to note that one cannot derive an analytical solution for (10) in the general case of arbitrary combination of initial and boundary conditions³. Thus, in many cases solutions are computed numerically, by discretizing the spatial and temporal domains.

An important remark is that the number of nonzero entries $\text{nnz}(\cdot) := \|\cdot\|_0$ in the two matrices \mathbf{A} and \mathbf{D} have radically different dependencies on the spatio-temporal “resolution” $n := st$, since $\text{nnz}(\mathbf{A}) = \mathcal{O}(n)$ while $\text{nnz}(\mathbf{D}) = \mathcal{O}(n^2)$. Note that one usually needs only $\mathbf{MD} \in \mathbb{R}^{m \times n}$, *i.e.* a subsampled dictionary with $\text{nnz}(\mathbf{MD}) = \mathcal{O}(mn) = \mathcal{O}(\tilde{m}st^2)$, where \tilde{m} is the number of microphones. Overall, \mathbf{MD} has $\mathcal{O}(\tilde{m}t)$ times more nonzero entries than \mathbf{A} . We will see that this leads to substantially different computational behaviors of sparse and cosparsity regularization.

³Not to be confused with the well-posedness of the PDE problem itself.

With the analysis operator and the dictionary available in discrete form, we can readily apply sparse and cospase regularizations to recover the pressure and source vectors and therefore solve the sound source localization inverse problem.

B. Second example: regularizing with Poisson's equation

Electrical potentials produced by neuronal activity can be measured at the surface of the head, up to an additive constant, using ElectroEncephaloGraphy (EEG). The sources of this neuronal activity can be approximately modeled as a small number of equivalent current dipoles. Numerous methods were developed to localize equivalent current dipoles from EEG recordings. Among them, beamforming techniques [35], subspace approaches [12] and sparse methods [36] are the most popular.

Spatial physical modeling. It is commonly admitted that the electrical potential $V(\mathbf{r})$ recorded at location \mathbf{r} of the head mostly reflects the activity of pyramidal cells which are located in the gray matter Υ and oriented perpendicularly to the cortical surface. As mentioned above, this activity is generally modeled by current dipoles with orientations perpendicular to the surface of the cortex. Given the geometry and the scalar field $\{W(\mathbf{r})\}$ of electrical conductivities within the head, Poisson's equation [37] establishes a link between the electrical potential $V(\mathbf{r})$ and the electrical activity of the current dipoles:

$$\operatorname{div}_{\mathbf{r}}(W(\mathbf{r}) \nabla_{\mathbf{r}} V(\mathbf{r})) = \sum_{j=1}^g Z_j (\delta(\mathbf{r} - \mathbf{r}_j^-) - \delta(\mathbf{r} - \mathbf{r}_j^+)) / d \quad (12)$$

where \mathbf{r}_j^- and \mathbf{r}_j^+ denote the positions of the two monopoles representing the current sink and source of the j -th dipole of Υ , respectively. Z_j is the amplitude of the j -th dipole of Υ and the operator div denotes divergence. Regarding the distance d between the two monopoles, it is of the order of the millimeter (typically of the order of the grid step). Indeed the human gray matter is about four millimetres thick.

Discretization. To discretize Poisson's equation and derive the operator \mathbf{A} , we use the Finite Difference Method (FDM) of Witwer et al. [38] that enforces Kirchhoff's law at each node of the cubic grid. The natural indeterminacy of the problem, stemming from the measurement of electric potential differences rather than potentials themselves, is removed by fixing the potential at one electrode. After discretization and incorporation of boundary conditions, this leads to a symmetric, positive definite and full-rank matrix $\mathbf{A} \in \mathbb{R}^{s \times s}$, where $(s + 1)$ is the total number of voxels in the head. Since the problem is time-independent, we have $n := s$.

As for the acoustic case, the matrix \mathbf{A} is very sparse, with at most seven non-zero components in each row [37]. The resulting discrete Poisson's equation reads $\mathbf{A}\mathbf{x} = \mathbf{i}$, where \mathbf{x} is the vector of potentials in the volume and \mathbf{i} is the vector of source currents.

Encoding additional knowledge. Additional knowledge about the sources of interest can be encoded in a matrix form, leading to a modified cospase model.

First, knowledge about the orientations of the q dipoles covering the gray matter Υ and about the opposite amplitudes of the corresponding monopoles can be encoded through a sparse matrix \mathbf{B} . Let u^- and u^+ index the elements of \mathbf{i} giving the amplitude of the monopoles located at positions \mathbf{r}^- and \mathbf{r}^+ , respectively. Knowing the geometry of the cortex, it is possible to build a set of q different couples (u_j^-, u_j^+) indexing the q dipoles of Υ .

Indeed, by covering the surface of the gray matter with q monopoles indexed by the integers u_j^- , we can deduce the q corresponding integers u_j^+ such that each dipole is oriented orthogonally to the surface of Υ , as the pyramidal cells it represents. We can thus express the vector \mathbf{i} as $\mathbf{i} = \mathbf{B}\mathbf{z}$, where $\mathbf{B} = (B_{u_1, u_2})$ is an $(s \times s)$ sparse matrix with entries

$$B_{u_1, u_2} = \begin{cases} 1 & \text{if } u_1 = u_2 \\ -1 & \text{if } u_1 = u_j^+ \text{ and } u_2 = u_j^- \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

and \mathbf{z} is an s -dimensional g -sparse vector with $s - q$ known zero elements (the u -th element of \mathbf{z} is zero if $u \neq u_j^-$ for $j \in \{1, \dots, q\}$). Non-zero elements of \mathbf{z} represent the amplitude of monopoles restricted to the cortical surface.

Matrix \mathbf{B} conveys our knowledge about the perpendicular orientations of the dipoles of Υ and the opposite amplitudes of each couple of monopoles. By construction, \mathbf{B} is invertible, and the cost of computing some matrix-vector product $\mathbf{B}^{-1}\mathbf{v}$ is $\mathcal{O}(s)$. To summarize, the model $\mathbf{A}\mathbf{x} = \mathbf{i} = \mathbf{B}\mathbf{z}$ can be rewritten $\tilde{\mathbf{A}}\mathbf{x} = \mathbf{z}$ with $\tilde{\mathbf{A}} := \mathbf{B}^{-1}\mathbf{A}$. By abuse of notation, \mathbf{A} will now denote the modified analysis operator $\tilde{\mathbf{A}}$, which is sparse as a product of two sparse matrices.

As mentioned in section III-C for a large class of PDEs, Poisson's equation (12) can be split between a source-region Ω_1 and a source-free region Ω_2 . As explained above, Ω_1 is the set of possible locations of monopoles at the surface of the gray matter Υ . The rows of the analysis operator \mathbf{A} can then be accordingly split into \mathbf{A}_{Ω_1} and \mathbf{A}_{Ω_2} , where \mathbf{A}_{Ω_1} is the $(q \times g)$ submatrix of \mathbf{A} obtained by extracting the rows of \mathbf{A} corresponding to the source support set Ω_1 , and \mathbf{A}_{Ω_2} corresponds to the rows indexed by the complementary set Ω_2 . Another way of encoding knowledge about the source support set Ω_1 is to factorize \mathbf{z} as $\mathbf{z} = \mathbf{R}\mathbf{s}$, where $\mathbf{R} = (R_{u_1, u_2})$ is an $(s \times q)$ expanding sparse matrix defined by:

$$R_{u_1, u_2} = \begin{cases} 1 & \text{if } u_1 = u_j^- \text{ and } u_2 = j \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

for $j \in \{1, \dots, q\}$.

Analysis versus synthesis. As in the acoustic context, the Green's functions dictionary \mathbf{D} can be computed as the inverse of the analysis operator \mathbf{A} . The brain sources can then be localized by solving either the analysis sparse problem (9) or the synthesis sparse problem (8) using the constraint matrix $\mathbf{C} = \mathbf{A}_{\Omega_2}$ and $\mathbf{c} = \mathbf{0}$. Both problems are equivalent, but, unlike the matrix \mathbf{A} , the dictionary \mathbf{D} is *not* sparse. The physical interpretation is that Poisson's equation models a non-zero steady-state electrical field on the domain given a source distribution. This can be easily verified for the spherical model where the analytical solution of the forward problem is available [12].

Using the expanding matrix \mathbf{R} (14), a second synthesis sparse problem can be solved in order to localize brain sources. It consists in solving (5) using the dictionary $\mathbf{D} = \mathbf{A}^{-1}\mathbf{R}$. In section VI, the corresponding optimization algorithm will be named MCE (Minimum Current Estimates) to refer to a similar approach proposed in [36]. Note that, in this case, an equivalent cospase optimization problem cannot be designed, since the matrix \mathbf{R} is not square invertible.

V. ALTERNATING DIRECTION METHOD OF MULTIPLIERS

Even with convex objective functions and constraints, the optimization problems (8)-(9) are often ill-conditioned and difficult to address, especially at large scale. Discretization of physical phenomena often requires handling a very large number of variables: this is particularly pronounced in models of time-dependent physical processes such as wave propagation.

To perform very large-scale convex optimization we are bound to use a first order method [20], [39]–[41]. After preliminary experiments with several state of the art methods, we retained the Alternating Direction Method of Multipliers (ADMM) [20] framework since it seemed generally faster to converge to higher accuracy. However, we obtained analogous results by using [39], implying that the conclusions hold regardless of the applied optimization algorithm.

A. Algorithm description

In its canonical form, ADMM allows for minimization of a sum of two objectives. Fortunately, there is a straightforward way to extend the canonical ADMM (appendix D) to account for more than two objective terms, known as *Simultaneous Direction Method of Multipliers (SDMM)* [42].

Generic SDMM. Consider the following convex optimization problem:

$$\underset{\mathbf{x}, \mathbf{z}_i}{\text{minimize}} \sum_{i=1}^f F_i(\mathbf{z}_i) \text{ subject to } \mathbf{H}_i \mathbf{x} - \mathbf{h}_i = \mathbf{z}_i.$$

Given an arbitrary choice of multipliers $\rho_i > 0$, the following alternating minimization steps converge (in the sense of the objective function) towards its minimum:

$$\mathbf{z}_i^{(k+1)} = \text{prox}_{\frac{1}{\rho_i} F_i} \left(\mathbf{H}_i \mathbf{x}^{(k)} - \mathbf{h}_i + \mathbf{u}_i^{(k)} \right), \quad (15)$$

$$\mathbf{x}^{(k+1)} = \arg \min_{\mathbf{x}} \sum_{i=1}^f \frac{\rho_i}{2} \|\mathbf{H}_i \mathbf{x} - \mathbf{h}_i + \mathbf{u}_i^{(k)} - \mathbf{z}_i^{(k+1)}\|_2^2, \quad (16)$$

$$\mathbf{u}_i^{(k+1)} = \mathbf{u}_i^{(k)} + \mathbf{H}_i \mathbf{x}^{(k+1)} - \mathbf{h}_i - \mathbf{z}_i^{(k+1)}. \quad (17)$$

The expression

$$\text{prox}_{\frac{1}{\rho_i} F}(\mathbf{z}) = \arg \min_{\mathbf{x}} \frac{1}{\rho_i} F(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2, \quad (18)$$

is the so-called *proximal operator* [42] of the function $\frac{1}{\rho_i} F$.

Synthesis / Analysis SDMM. Using SDMM for both the synthesis and analysis problems is straightforward. For synthesis, the problem is parametrized by $\mathbf{H}_1 = \mathbf{I}$, $\mathbf{H}_2 = \mathbf{M}\mathbf{D}$ and $\mathbf{H}_3 = \mathbf{C}\mathbf{D}$, where \mathbf{I} denotes the identity. For analysis, we set $\mathbf{H}_1 = \mathbf{A}$, $\mathbf{H}_2 = \mathbf{M}$ and $\mathbf{H}_3 = \mathbf{C}$. In both cases we have $\mathbf{h}_1 = \mathbf{0}$, $\mathbf{h}_2 = \mathbf{y}$ and $\mathbf{h}_3 = \mathbf{c}$.

Choice of multipliers. In the considered showcase applications, we have $f = 2$ or $f = 3$, depending whether the penalty $F_3(\cdot)$ is present or not. The functionals F_2 and F_3 are used to constrain the problem, hence they penalize the distance to the measurements and enforce additional constraints. The functional F_1 is a sparsity-promoting term, for which we envision three types of penalties, defined in Appendix B. Their proximal operators are provided in Appendix C.

Note that the multipliers ρ_i can be chosen differently for each \mathbf{z}_i , since this simply corresponds to rescaling the expression $\mathbf{H}_i \mathbf{x} = \mathbf{z}_i$ (described in Appendix D). For any choice of multiplier, the iterates are guaranteed to converge (in the sense of the objective function) towards a minimum of (15); however, the particular choice of ρ_i may influence the convergence speed, since the \mathbf{x} -update step (16) involves a weighted least squares problem.

Empirically, to quickly attain a feasible point and preserve convergence speed, it seems appropriate to adjust the multipliers ρ_1 , ρ_2 and ρ_3 relative to the constraint parameters. Our strategy is to first fix the value $\rho_1 = \rho$, and then set i) $\rho_2 = \max(\rho, \rho/\varepsilon)$ and $\rho_3 = \max(\rho, \rho/\sigma)$ if F_2 and F_3 are the indicator functions bounding a norm of their arguments by ε and σ , respectively; or ii) $\rho_2 = \max(\rho, \rho\sqrt{\varepsilon})$ and $\rho_3 = \max(\rho, \rho\sqrt{\sigma})$ if F_2 and F_3 are norm-squared penalties weighted by ε and σ , respectively. Other types of penalties are allowed, but may require different weighting heuristics.

Direct vs iterative solvers. For problems of modest scale, the least squares minimization step (16) can (and should) be performed exactly, by means of a direct method, *i.e.* matrix inversion. For computational efficiency, this requires relying on matrix factorization such as the Cholesky decomposition. An important observation is that many discretization schemes yield banded matrices (such as in our two examples). Then, the coefficient matrix arising from the normal equations in the analysis least squares step is again banded, which leads to a *banded Cholesky factor* [43, Theorem 1.5.1]. Further, a desirable property is to obtain Cholesky factors essentially as sparse as the factorized matrix. Many efficient algorithms heuristically achieve this goal (such as the sparse Cholesky decomposition [44] used in our computations). They will be referred to as *sparsity-preserving* algorithms. However, the number of non-zero elements of the factorized matrix is a lower bound on the number of non-zero elements of its Cholesky decomposition [45, Theorem 4.2]. Accordingly, due to the density of the dictionary \mathbf{D} , only the sparse analysis regularization will benefit from these sparse factorizations.

For large scale problems one needs to resort to iterative algorithms and approximate the solution of (16). An important advantage of ADMM is that it ensures convergence even with inexact computations of intermediate steps, as long as the accumulated error is finite [46]. Moreover, these algorithms can be usually initialized (*warm-started*) using the estimate from the previous ADMM iteration, which can have a huge influence on the overall speed of convergence. For the weighted SDMM we use, a good choice is the Least Squares Minimum Residual (LSMR) algorithm [47], which is discussed in the experimental section and Appendix D.

B. Computational aspects

It can be shown that the above-described SDMM algorithm yields *numerically identical* solutions for the synthesis and the analysis problems, as long as all evaluations in (16) are exact (this corresponds to the usage of direct methods, described in the previous subsection). However, as detailed below, the overall cost of the analysis minimization is driven by that of the multiplication with \mathbf{A} and its transpose, which is $\mathcal{O}(n)$ thanks to the sparsity of the analysis operator \mathbf{A} . This is in stark contrast with synthesis minimization, whose cost is dominated by much heavier multiplications with the dense matrix \mathbf{MD} and its transpose. As a result, and as will be confirmed experimentally in Section VI, *the analysis minimization is computationally much more efficient.*

Initialization costs. Generating the analysis operator $\mathbf{A} \in \mathbb{R}^{n \times n}$ in matrix form is problem dependent, but usually of moderate cost. The cost of building the transfer matrix \mathbf{M} is negligible, since it often corresponds to simple acquisition models (e.g. the row-reduced identity). To efficiently compute *the reduced dictionary* $\mathbf{G} = \mathbf{M}\mathbf{D} \in \mathbb{R}^{m \times n}$, which satisfies $\mathbf{A}^\top \mathbf{G}^\top = \mathbf{M}^\top$, one needs to solve m linear systems⁴ $\mathbf{A}^\top \mathbf{g}_j = \mathbf{m}_j$ of order n , where \mathbf{g}_j^\top and $\mathbf{m}_j^\top \in \mathbb{R}^n$ are the rows of \mathbf{G} and \mathbf{M} , respectively. Thus, it adds at least⁵ $\mathcal{O}(mn)$ operations on the price of computing \mathbf{A} and \mathbf{M} , unless an analytical expression of Green’s functions is available.

If the direct method is used to solve the linear least squares step, classical algebraic manipulations show that we first need to compute the coefficient matrix $\mathbf{H}_A = \rho_1 \mathbf{A}^\top \mathbf{A} + \rho_2 \mathbf{M}^\top \mathbf{M}$, in the analysis case, or $\mathbf{H}_S = \rho_1 \mathbf{I} + \rho_2 (\mathbf{M}\mathbf{D})^\top \mathbf{M}\mathbf{D}$, in the synthesis case. Due to the sparse structure of \mathbf{A} and \mathbf{M} , the former can be computed in $\mathcal{O}(n)$, while the latter requires $\mathcal{O}(n^2 m^2)$ operations. The Cholesky factorization requires $\mathcal{O}(n^3)$ operations in general, but this is significantly reduced for sparse matrices [45]. However, it is known [49] that the minimum fill-in problem –finding the best sparsity-preserving permutation– is an NP-complete problem, thus all available algorithms are (usually very efficient) heuristics. Therefore, it is very difficult to derive tight bounds on initialization complexity.

Iteration costs. Each iteration of SDMM involves three types of operations.

a) *Component-wise scalar operations:* evaluation of proximal operators and update of scaled Lagrangian multipliers \mathbf{u}_i given $\mathbf{H}_i \mathbf{x}^{(k)}$ and $\mathbf{H}_i \mathbf{x}^{(k+1)}$. These have $\mathcal{O}(n)$ complexity with respect to the problem size n , since they involve only component-wise thresholding and vector norm computations (see appendix C).

b) *Matrix-vector products:* computation of $\mathbf{H}_i \mathbf{x}^{(j)}$. For analysis, the matrices \mathbf{H}_1 and \mathbf{H}_2 are both sparse with $\mathcal{O}(n)$ nonzero entries, hence these matrix-vector products also have an $\mathcal{O}(n)$ complexity. This is also the case for \mathbf{H}_3 in the considered scenarios. In contrast, for synthesis, the matrix $\mathbf{H}_2 = \mathbf{M}\mathbf{D}$ is dense, reflecting the discretized Green’s functions. For example, in the case of the wave equation, given \tilde{m} microphones, a spatial “resolution” of s voxels with $s > \tilde{m}$, and a temporal “resolution” t such that $n = st$ and $m = \tilde{m}t$, the complexity of this matrix-vector product is $\mathcal{O}(\tilde{m}t \times n) = \mathcal{O}(\tilde{m}st^2)$. It dominates the cost of all other matrix-vector products.

c) *Least squares:* the solution of problem (16). When an iterative solver is used to address the least squares step, we assume that a properly preconditioned and warm-started iterative method would terminate to sufficient accuracy in considerably less than n iterations. The overall computational complexity is governed by the cost of matrix-vector products $\mathbf{H}_i \mathbf{v}$ and $(\mathbf{H}_i)^\top \mathbf{w}$ for some intermediate vectors \mathbf{v} , \mathbf{w} , which as just seen have very different complexities in the analysis and synthesis settings.

When a direct method is used to evaluate the linear least squares step the complexity analysis is more delicate. Since the matrix \mathbf{H}_A is banded for both acoustic and EEG problems, we know that regular Cholesky decomposition will indeed produce factors, which are much sparser in the analysis than in the synthesis case. For example, for the wave equation, we have $\mathbf{H}_A = \mathbf{L}_A \mathbf{L}_A^\top$, where $\text{nnz}(\mathbf{L}_A) = \mathcal{O}(s^2 t)$, and $\mathbf{H}_S = \mathbf{L}_S \mathbf{L}_S^\top$, where $\text{nnz}(\mathbf{L}_S) = \mathcal{O}(s^2 t^2)$. However, due to the mentioned NP-hardness of the minimum fill-in problem, it is impossible to exactly evaluate the sparsity of the yielded *sparse* Cholesky factorization $\mathbf{P}^\top \mathbf{H}_A \mathbf{P}$ (\mathbf{P} is a permutation matrix), and estimate the

⁴The reader may notice the similarity with time-reversal [48].

⁵Assuming the favorable scenario where the linear system can be solved in $\mathcal{O}(n)$ operations, such as for the SLF method.

Problem size $s \times t$	$(19 \times 19) \times 61$	$(30 \times 30) \times 97$	$(48 \times 48) \times 155$	$(60 \times 60) \times 194$	$(76 \times 76) \times 246$	$(95 \times 95) \times 307$
Synthesis (GB)	0.1	0.6	4.1	10	26	63
Analysis (GB)	0.001	0.005	0.02	0.04	0.07	0.2

Table I

MEMORY REQUIREMENTS RELATIVE TO THE PROBLEM SIZE, WITH $\tilde{m} = 10$ MICROPHONES.

computational complexity. But, it is reasonable to assume that the permuted coefficient matrix would yield even sparser Cholesky factor than \mathbf{L}_A . Therefore, one may expect the analysis model to be computationally much more efficient. This result was empirically checked through simulations.

Additionally, for the proposed weighting scheme, the matrix \mathbf{H}_A is usually better conditioned than \mathbf{H}_S . The rationale comes from the fact that the applied multipliers $\rho_1 \leq \rho_2$ usually assign a large weight to the diagonal elements of the matrix \mathbf{H}_A . And since $\mathbf{H}_A = \mathbf{A}^\top \mathbf{H}_S \mathbf{A}$, one can see this as *preconditioned* synthesis approach.

C. Memory aspects

Another view on computational scalability is through memory requirements. For the synthesis model, assuming the general case where the analytical expression of the Green functions is not available, the least requirement is storing the $(m \times n)$ matrix \mathbf{MD} in memory (to avoid computational overhead, it is usually necessary to also store $(\mathbf{MD})^\top$). Hence, the minimum storage requirement for the synthesis case is $\mathcal{O}(mn)$. As we will see numerically in the experimental section (see Table I), this cost can quickly become prohibitive (*e.g.* in the acoustic setting where $\mathcal{O}(mn) = \mathcal{O}(\tilde{m}st^2)$), even for modern computers with large amounts of RAM. On the other hand, memory requirement for storing the analysis operator is $\mathcal{O}(n)$.

Evaluating the storage requirements of sparse Cholesky factorization is not viable, due to the NP-hardness of the fill-in problem. It is, however, presumed (and experimentally checked below) that it requires substantially less than $\mathcal{O}(n^2)$ memory units, which is the requirement for the regular (synthesis) Cholesky factor.

D. Structured matrices

In some cases, a special matrix structure can be exploited in order to further reduce storage size and computational effort. For instance, when the wave equation is time-invariant, the use of local discretization schemes leads to an almost-block-Toeplitz⁶ analysis operator, which means that we can store \mathbf{A} with $\mathcal{O}(s)$ cost. Moreover, if an explicit scheme (such as the FDTD-SLF) is used, the analysis operator becomes lower triangular [50]. Consequently, its inverse, the dictionary matrix \mathbf{D} , is also triangular almost-block-Toeplitz, and we can represent \mathbf{MD} with $\mathcal{O}(\tilde{m}st)$ parameters. Unfortunately, this is possible only in special settings where the assumptions are made on the time-invariance of the physical process *and* on the chosen discretization. Indeed, the inverse of an arbitrary (block) Toeplitz matrix is not necessarily (block) Toeplitz [43].

⁶excluding initial conditions.

Our goal in this work is not to exploit such matrix structures, but to emphasize that in the analysis case, one can “naively” manipulate the matrix (as long as the discretization has local support) and still gain in terms of computational and storage resources. This is not necessarily the case with the synthesis approach, and even if possible, requires specialized techniques in order to exploit particular matrix structure. For example, in the case where $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a Toeplitz matrix, this means first embedding it in a circulant form, and then applying Fast Fourier Transform at cost $\mathcal{O}(n \log n)$ [51] (even in this case the analysis approach is cheaper, since it requires $\mathcal{O}(n)$ calculations).

Finally, note that while such specific matrix structures in \mathbf{A} and \mathbf{D} may be useful when iterative algorithms are used to solve the least squares step in SDMM, these become useless for the direct computation of linear systems with matrices \mathbf{H}_A and \mathbf{H}_S (since forming the normal equations usually disturbs the structure). For all these reasons, in the subsequent experiments, we disregard any additional structures of the involved matrices except their sparsity patterns.

VI. SIMULATIONS

Numerical results comprise different scenarios both in the acoustic and biomedical contexts, with the goal to highlight different aspects of the sparse and cosparsity regularizations. Since the two problems are formally equivalent, the expected computational differences will be due to the sparse structure of the analysis operator \mathbf{A} .

A. General settings

Objective functions. In the following experiments, the sparsity promoting objective F_1 is chosen as follows: standard ℓ_1 , joint $\ell_{2,1}$, and hierarchical $\ell_{2,1}$ norm (see appendix B) in the acoustic case, ℓ_1 in the biomedical case. This optimization is performed under the measurement constraint $F_2(\cdot) = \mathbf{1}_{\ell_2 \leq \varepsilon}(\cdot)$, where $\varepsilon = 10^{-5}$ in all cases but one, where we explicitly specify this parameter. In addition, for all EEG experiments, we use the additional constraint term $F_3(\cdot) = \mathbf{1}_{\ell_2 \leq \sigma}(\cdot)$ with $\sigma = 10^{-5}$.

Stopping criteria. Unless otherwise specified, we use the stopping criteria (38) and (39) with the relative accuracies $\epsilon = 10^{-2}$ (in the acoustic setting) and $\epsilon = 10^{-3}$ (in the EEG setting) or the maximum number of SDMM iterations (5000 in the acoustic case and 10000 in the biomedical context). The SDMM multiplier ρ_1 is set to $\rho_1 = \rho = 10$ and the remaining ones are computed using the heuristics explained in section V. In the acoustic experiments where the LSMR algorithm is used to estimate the solution, we set its stopping criterion to $\|\mathbf{H}^T(\mathbf{H}\mathbf{v} - \mathbf{h})\|_2 / \|\mathbf{h}\|_2 \leq 10^{-4}\epsilon$ (given a least squares problem $\mathbf{v}^* = \arg \min_{\mathbf{v}} \|\mathbf{H}\mathbf{v} - \mathbf{h}\|_2^2$).

Data simulation and processing. As far as the acoustic experiments are concerned, the sampling model simulates *noiseless* recordings taken by fixed, but randomly placed microphones in reverberant environment ($E(\mathbf{r}) = \xi = 100$). The point sources are also randomly distributed in space and modeled as white noise signals.

Regarding the brain source localization experiments, $g = 3$ distant synchronous epileptic dipoles are randomly arranged in the gray matter of the superior cortex. A physiologically-relevant model [52] is used to generate the spike-like interictal epileptic activity. It is noteworthy that this activity is the same for the three epileptic dipoles,

leading to synchronous epileptic sources. On the other hand, the background activity, *i.e.*, the activity of non-epileptic dipoles of the gray matter, is generated as Gaussian and as temporally and spatially white. Its power is controlled by a multiplicative coefficient in order to get different Signal to Noise Ratio (SNR) values. We use three nested concentric spheres for the head model. Since Poisson’s equation models an instantaneous phenomenon, we select the electrode measurements matching the time sample where the highest signal amplitude is observed (this should correspond to the top of the epileptic activity spike).

Assuming that the number of sources g is given, we determine the set of the sound source location estimates from $\mathbf{z} = \mathbf{A}\mathbf{x}$ (the analysis case) or directly from \mathbf{z} (the synthesis case). Namely, we index the elements of \mathbf{z} by (\mathbf{r}, t) (in the acoustic setting) and by \mathbf{r} (in the EEG setting). Then, we simply declare the locations corresponding to the g highest values of $\|\mathbf{z}_{\mathbf{r},\cdot}\|_2$ (in the acoustic case) or $|z_{\mathbf{r}}|$ (in the EEG case) to be the source positions.

Performance measure. The quality of localization is presented as an estimated error per source. The error is computed as the Euclidean distance between pairs $(\hat{\mathbf{r}}_i, \tilde{\mathbf{r}}_i)$, where $\hat{\mathbf{r}}_i$ and $\tilde{\mathbf{r}}_i$ denote the estimated and the true position of the i^{th} source, respectively. The pairs are chosen such that the overall error is minimal, by means of the Hungarian algorithm [53]. The simulation results are averaged over 50 realizations. Note that we resampled the sources and their location between these 50 experiments. Acoustic experiments were run on Intel® Xeon® 2.4GHz cores, equipped with 8GB RAM, in single-core/single-thread mode, while all biomedical experiments were run on Intel® Xeon® 4-Core 2.8GHz, equipped with 32GB RAM.

B. Equivalence of synthesis and analysis results

A first set of source localization experiments provides empirical evidence of the equivalence of the analysis and synthesis models when the analysis operator is square and invertible. In particular, in the acoustic context, we compare their respective localization performance as a function of the number of sources, while using the three chosen convex functionals F_1 (ℓ_1 , $\ell_{2,1}$ and hierarchical $\ell_{2,1}$ norms). These experiments are purely numerical, and we postpone experiments with a more physical interpretation to Section VI-D. We vary the number of sources from $g = 1$ to $g = 15$ and use $\tilde{m} = 10$ microphones for this simulation. To avoid a potential bias in the results if iterative methods are used to solve (16), we restrict the experiments to a small scale problem for which direct methods are still applicable. The domain is an artificial two dimensional spatial grid of size 15×15 , simulated through 50 “temporal slices”. The emission duration of white noise sources is five time samples.

In all these settings, when the same objective F_1 is used and the same number of sources is active, we verify point-to-point that both models provide numerically identical results.

C. Scalability

A second series of source localization experiments compares the scalability potential of the two models for the acoustic wave and Poissons’s equations. Here we are interested in studying the computational cost as a function of both the problem size and the number of measurements.

In the experiments dedicated to the acoustic wave equation, we use the LSMR iterative method to approximately solve the least squares problem (16) without building and storing a fully dense coefficient matrix for the synthesis

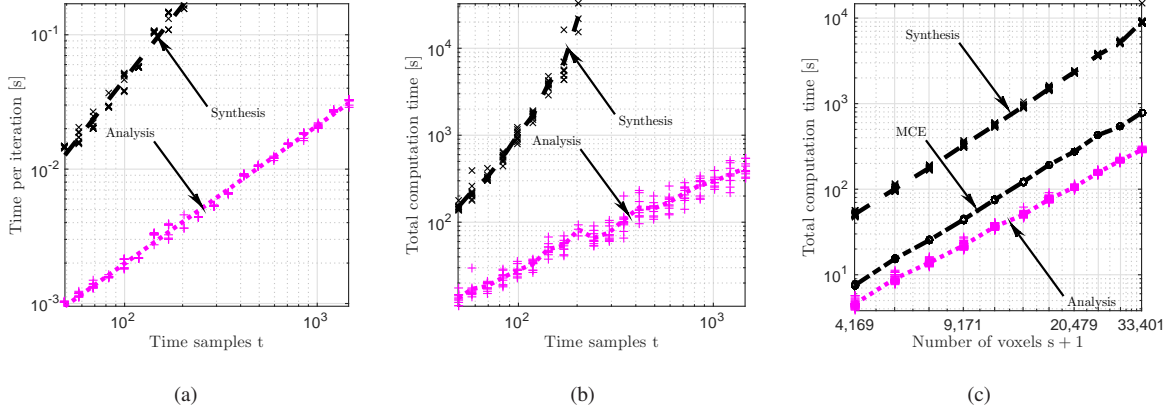


Figure 1. Computation time relative to the problem size. (a) Time per LSMR iteration vs t (wave equation). (b) Total computation time vs t (wave equation). (c) Total computation time vs $s + 1$ (Poisson's equation).

model (its storage cost would be of the order of 10^{11} bytes in double-precision floating point format). In order to ensure there is no bias towards any of the two models (since the primal/dual residuals defined in appendix D may be influenced by norms of the involved matrices), an oracle stopping criterion is used: iterations stop when the objective function $F_1(\mathbf{z}^{(k)})$ falls below $c \cdot F_1(\tilde{\mathbf{z}})$ with $c = 1.1$ and $\tilde{\mathbf{z}}$ the ground truth source signal.

Influence of the problem size. The first experimental scenario concerns sound source localization. We vary the number of time samples of the problem to verify that the two models scale differently with respect to temporal dimension. Using the results obtained by the experiments in the subsection VI-B, we restrict the experimental setup to the regime where perfect localization is possible, and we consider 20 different values of t from 50 to 1455. The results on Fig. 1(a) confirm our predictions: the computational cost per iteration for the cospase optimization problem grows linearly with t , while the cost of its synthesis counterpart is nearly quadratic. Moreover, the difference between the two models becomes striking when the total computation time is considered (Fig. 1(b)), since the synthesis-based problem exhibits cubic growth. Finally, we are unable to scale the synthesis problem above $t = 203$, due to significantly increased memory requirements (Table I) and computation time.

In the second scenario, dedicated to Poisson's equation, we vary the discretization at 11 different scales, yielding uniform grids with a number of nodes ranging between $s + 1 = 4169$ and $s + 1 = 33401$. We recall that the number of optimization variables is equal to s for the synthesis/analysis approaches, and equal to $q \ll s$ for the MCE-like approach. Therefore, the results on Fig. 1(c) are presented with respect to the number $s + 1$ of voxels in the head, which is the same for all methods. As for the acoustic case, the computational cost of the synthesis approach is higher and grows faster than that of its analysis counterpart. Interestingly, the latter and the MCE-like approach show a similar behavior. In fact, the direct computation of the proximal operators and the SDMM involved in the MCE-like approach require a slightly lower cost, but the additional cost due to the initial computation of the dictionary slightly increase the MCE-like technique total cost ; the impact of this additional cost is not negligible, since the dictionary has to be recomputed for each patient in clinical practice.

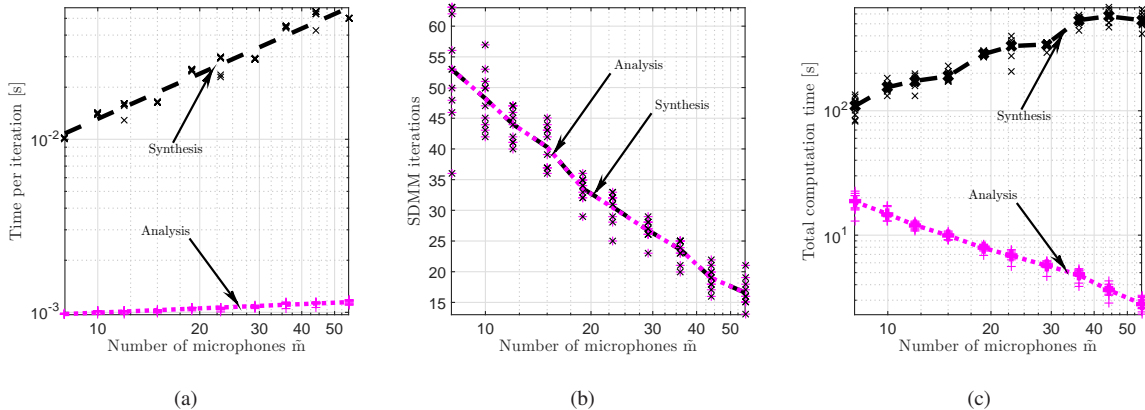


Figure 2. Computational cost vs number of microphones \tilde{m} . (a) Time per LSMR iteration. (b) Number of SDMM iterations. (c) Total computation time.

Influence of the number of measurements. Keeping the number of variables n fixed, we now vary the number of measurements in the acoustic scenario. Given the complexity analysis in Section V-B, we expect the per-iteration complexity of the analysis model to be approximately independent of the number of microphones \tilde{m} , while the cost of the synthesis model should grow linearly with \tilde{m} . The results shown on Fig. 2(c) and Fig. 2(b) confirm this behavior in terms of computational cost per inner iteration: in the synthesis case, it grows at almost linear rate, while being practically independent of \tilde{m} in the analysis case. However, the number of (outer) SDMM iterations decreases with \tilde{m} for both models. Overall, the total computation time increases in the synthesis case, but it *decreases* with the number of microphones in the analysis case. While perhaps a surprise, this is in line with recent theoretical studies [54] suggesting that the availability of more data may enable the acceleration of certain machine learning tasks. Here the acceleration is only revealed when adopting the analysis viewpoint rather than the synthesis one.

D. Robustness and versatility

We conclude with a final series of experiments in more realistic settings to provide empirical evidence of the versatility and robustness of the proposed framework. We no longer evaluate the sparse synthesis model, since it has been shown to have identical numerical performance compared to the analysis model, with much poorer large-scale abilities.

Versatility. The proposed cospase approach can be exploited with any shape of the spatial domain, as long as the information emitted from the sources can reach the microphones. On the other hand, expressing Green’s functions analytically is possible only for a restricted set of geometries, otherwise, one must evaluate them numerically. As mentioned in subsection V-B, this means solving a large number of linear systems with the coefficient matrix given as the analysis operator.

A challenging scenario previously considered in [19] is a “split room”, where microphones and sources are separated by a “wall” obstacle preventing a direct propagation path. Sound propagation is simulated in a virtual 3D space of size $2.5 \times 2.5 \times 2.5\text{m}^3$, with a separating wall of length 1.5m. The recording time is set to $t_1 = 5\text{s}$ and

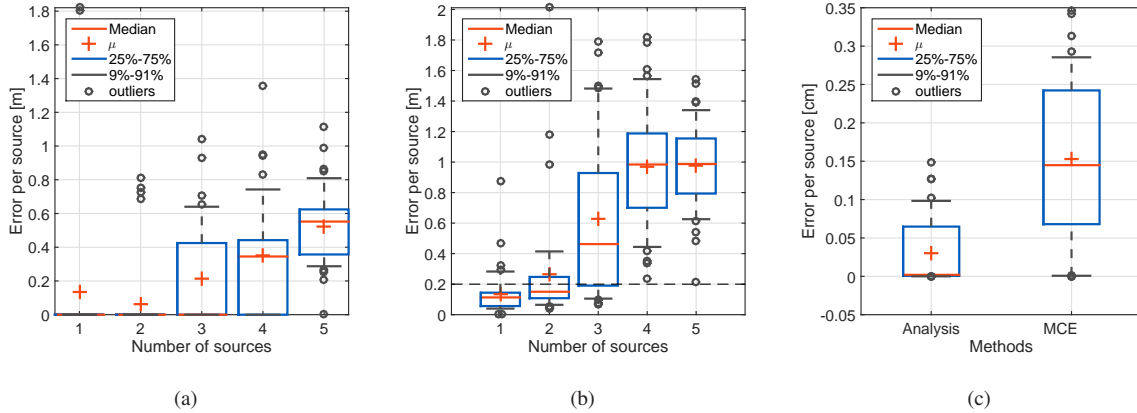


Figure 3. Performance analysis in realistic contexts: (a) The “split room” acoustic experiment. (b) Avoiding the inverse crime (dashed line represents perfect localization tolerance of the given crude grid). (c) Brain source localization error at the output of the analysis and MCE approaches for an SNR of -20 dB.

the white noise sources emit during the entire acquisition period. The spatial domain step sizes are $d_x = d_y = d_z = 0.25\text{m}$ and the sampling frequency is $f_s \approx 2.38\text{kHz}$ to ensure the marginal stability of the SLF scheme. This results in 1.2×10^7 variables for the $\ell_{2,1}$ -norm minimization problem. The computer results presented in the Fig. 3(a), indicate that we are able to perfectly localize up to three sources. Reverberation, which is usually considered as undesirable in audio inverse problems, is thus essentially exploited in this approach. This is reminiscent of time-reversal techniques [48] as well as of recent echo-based techniques for source localization and room-shape estimation [16].

Model errors. So far we have addressed inverse problems where the simulated data perfectly matches the model assumed for regularization. This is known as *the inverse crime* [55]. To avoid it, we now consider an acoustic 2D setting where the simulated data is first generated on a fine grid of size $121 \times 121 \times 6121$, before solving the inverse problem on a coarse grid of size $25 \times 25 \times 1225$ (both grids simulate a virtual 2D space of size $5 \times 5\text{m}^2$ with recording and emission times set to 5s). Microphone positions correspond to the nodes of the crude grid, while white noise sources are arbitrarily distributed at the nodes of the fine grid. Before downsampling, the fine model data is temporally low-pass filtered to reduce the aliasing effects. The product⁷ $\mathbf{A}\tilde{\mathbf{x}}$ is now only *approximately* sparse, and to account for this model error we increased the data fidelity parameter ε to 0.1. The results shown in the Fig. 3(b) imply that we are able to localize up to two sources with an error on the order of the crude grid’s spatial step size.

Noise. Another type of model error encountered in the biomedical context is due to the presence of background activity in non-epileptic regions of the gray matter. Non-epileptic dipoles of the gray matter also have a non-zero amplitude, even if it should be ideally lower than that of epileptic dipoles. Consequently, the n -dimensional vector $\mathbf{A}\mathbf{x}$ is not really g -sparse, but it should have g dominant components. We conduct a second experiment to evaluate the influence of such model errors. Fig. 3(c) presents the simulation results for an SNR value of -20 dB. This

⁷ $\tilde{\mathbf{x}}$ is the crude version of the “fine” data vector \mathbf{x} .

is representative of localization problems with synchronous and focal epileptic sources, which are challenging for existing techniques. The analysis approach exhibits a remarkable robustness with respect to background activity and manages to perfectly localize epileptic sources even for a very low SNR value. This is, however, not the case for the MCE approach, which requires significantly higher SNR to achieve the same performance. Given these results, along with the results from the previous subsection, the analysis approach is a clear winner in terms of both localization performance and computational cost.

VII. CONCLUSION

A unified framework for (co)sparse regularization of inverse problems emerging from discretized linear PDEs has been presented. We also described a simple, but general algorithm based on ADMM which may be used to solve optimization problems generated by this framework.

After demonstrating the nominal equivalence between the sparse synthesis and sparse analysis models driven by the same underlying physics, we highlighted the much more favorable properties of the latter from a computational point of view. We showed indeed that the synthesis model does not scale well with the problem size in terms of computation time and memory. In stark contrast, not only does the analysis model scale up linearly with the problem size, but it is also speeded up when the amount of measurement data increases. We emphasized that the computational differences are intimately due to the difference in inherited sparsity of the analysis operator \mathbf{A} and the synthesis dictionary \mathbf{D} , rather than to a particular choice of algorithmic approach or discretization scheme. In the analysis approach, discretization of the underlying PDE leads to a system of difference equations, while numerical integration underpins the construction of synthesis dictionaries of Green's functions.

Even in specific practical contexts such as brain source localization where additional knowledge about the source supports can reduce the computational cost of the sparse synthesis model, the analysis approach remains competitive thanks to its improved robustness to the presence of background activity, and its ability to work without pre-computing Green's functions.

For the purpose of illustration of these fundamental differences, the simulations were conducted with finite difference discretization schemes. As evoked in the Appendix, this comes with constraints such as the *Courant-Friedrich-Lewy* (CFL) conditions –somewhat limiting the scalability in 3D+t problems– and limited geometric flexibility, for example with respect to orientations of the boundaries. Extensions to other discretization schemes –such as Finite Element Methods (FEM)– are straightforward, provided they still yield a sparse discretized analysis operator. Combining these with multilevel methods such as algebraic multigrid [56], parallelized solvers or smoothing strategies [57], or GPU implementations, should further accelerate the optimization process and allow applications to real-life acoustic and EEG data.

In the considered illustrative scenarios, the physical parameters (head conductivities, sound celerity, boundary conditions) and geometry of the domains were assumed perfectly known. From a practical perspective, it is important to handle inaccurate models. This can be seen from two viewpoints: the robustness to model errors, and the ability to incorporate the estimation of certain unknown physical parameters. The former has to be further investigated and confronted to real data, in order to strengthen preliminary evidence gathered in this paper.

Regarding the latter, several scenarios arise. For the estimation of certain unknown physical parameters, one can envision expressing an extended linear inverse problem by appending new variables to the unknown vector \mathbf{x} , and incorporating prior knowledge on these parameters through appropriate penalties. This is for example the case of initial or boundary conditions in the wave equation. This may however raise new identifiability, conditioning or scalability questions. Other cases, such as unknown celerities or conductivities, are more challenging. Preliminary work [58] suggests the potential of bilinear formalisms and lifting techniques to address them.

APPENDIX A

FDTD DISCRETIZATION OF THE WAVE EQUATION

To discretize the isotropic⁸ wave-equation part of the operator (for $\mathbf{r} \in \Omega \setminus \partial\Omega$ and $t > t_0$) in a 2D setting, we rely on the following expression:

$$\begin{aligned} \frac{\partial^2 p_{i,j}^t}{\partial x^2} + \frac{\partial^2 p_{i,j}^t}{\partial y^2} - \frac{1}{c^2} \frac{\partial^2 p_{i,j}^t}{\partial t^2} &= \frac{p_{i-1,j}^t - 2p_{i,j}^t + p_{i+1,j}^t}{d_x^2} \\ &+ \frac{p_{i,j-1}^t - 2p_{i,j}^t + p_{i,j+1}^t}{d_y^2} - \frac{1}{c^2} \frac{p_{i,j}^{t+1} - 2p_{i,j}^t + p_{i,j}^{t-1}}{d_t^2} \\ &+ \mathcal{O}(\max(d_x, d_y, d_t)^2) \end{aligned} \quad (19)$$

where d_x, d_y and d_t denote the discretized spatial and temporal step sizes, respectively. Neglecting the $\mathcal{O}(\cdot)$ term yields a convenient *explicit* scheme [59] to compute $p_{i,j}^{t+1}$ using pressure values at the previous two discrete time instances ($p_{(\cdot,\cdot)}^t$ and $p_{(\cdot,\cdot)}^{t-1}$). Initial conditions provide $p_{i,j}^t$, for all (i,j) and $t \in \{0,1\}$. Boundary conditions are discretized by substituting a non-existent spatial point in the scheme (19) by the expression obtained from (11). For example, the missing point behind the right “wall” is evaluated as:

$$p_{i+1,j}^t = p_{i-1,j}^t + \frac{d_x}{cd_t \xi_{i,j}} \left(p_{i,j}^{t-1} - p_{i,j}^{t+1} \right) \quad (20)$$

To ensure the stability of the scheme, spatial and temporal step sizes are bound to respect the *Courant-Friedrich-Lewy* (CFL) condition [59]: $cd_t / \min(d_x, d_y) \leq 1/\sqrt{2}$.

Concatenating these difference equations for the entire spatio-temporal dimension $s \times t$ yields a full rank / square-invertible matrix operator $\mathbf{A} \in \mathbb{R}^{st \times st}$.

APPENDIX B

SPARSITY MEASURES

The most common convex relaxation of the non-convex sparsity-promoting ℓ_0 objective is the ℓ_1 -norm,

$$\|\mathbf{v}\|_1 = \sum_i |v_i| \quad (21)$$

which is known to promote sparse solutions.

⁸The speed of sound $C(\mathbf{r}, t) = c = 343\text{m/s}$ is uniform in all directions.

If local spatial stationarity of the sources is assumed (say, for sufficiently short acquisition time), the sources retain fixed positions in space. Then we favor solutions for which all temporal slices of the sparse estimate have the same support (also known as *jointly sparse* vectors). This is promoted the $\ell_{2,1}$ -norm:

$$\|\mathbf{v}\|_{2,1} = \sum_i \sqrt{\sum_j |v_{i,j}|^2}, \quad (22)$$

where $v_{i,j}$ denotes the (i,j) th element obtained by transforming the vector \mathbf{v} into a matrix \mathbf{V} whose columns are jointly sparse subvectors.

The joint $\ell_{2,1}$ norm is a special case of group $\ell_{2,1}$ norms. Another interesting case is a special type of hierarchical $\ell_{2,1}$ -norms where groups are either singletons or disjoint subsets of elements [60]. This objective function should encourage solutions with a small number of active *groups* (such as the temporal groups for jointly sparse vectors) and which are overall sparse. It is evaluated as a sum $\|\mathbf{v}\|_{2,1} + \|\mathbf{v}\|_1$. This objective could be useful in the acoustic setting also: recall that the acoustic wave equation models a time-dynamic system in which the information propagates with finite speed C . As a consequence, one cannot hope to recover a source whose information cannot reach the microphones in a given acquisition time (it can be interpreted as a *sound event horizon*). Thus, regardless of its duration, source emission appears “trimmed”. Another possibility is a source emitting speech signal, which usually contains silent intervals. This knowledge can be casted into the objective function: we still seek the solutions which are spatially sparse, but additionally we prefer if the solutions have a small number of zeros per spatial group.

APPENDIX C PROXIMAL OPERATORS

Proximal operator for the objective F_1 depends on the choice of the objective from the pool of cost functions mentioned in Appendix B. Conveniently, all of them have explicit solutions (these are well-known results by now, available in the references):

$$\begin{aligned} (\text{prox}_{\frac{1}{\rho}\ell_1}(\mathbf{v}))_i &= v_i \left(1 - \frac{1}{\rho|v_i|}\right)_+, \\ (\text{prox}_{\frac{1}{\rho}\ell_{2,1}}(\mathbf{v}))_i &= v_i \left(1 - \frac{1}{\rho\|\mathbf{v}_{\Upsilon}\|_2}\right)_+, \\ \text{prox}_{\frac{1}{\rho}(\ell_{2,1}+\ell_1)}(\mathbf{v}) &= \text{prox}_{\frac{1}{\rho}\ell_{2,1}}\left(\text{prox}_{\frac{1}{\rho}\ell_1}(\mathbf{v})\right). \end{aligned}$$

The operator $(\cdot)_+$ denotes component-wise positive thresholding: $(\mathbf{v})_+ := \{\forall i \mid \max(v_i, 0)\}$, and \mathbf{v}_{Υ} denotes a vector composed by the elements of a vector \mathbf{v} indexed by the indice-set Υ .

For the constraint functionals F_2 and F_3 , one may use various penalty functions. In this paper, we use the indicator function $\mathbf{1}_{\ell_2 \leq \mu}$, *i.e.* bounding the ℓ_2 norm of \mathbf{v} by some noise level: $\|\mathbf{v}\|_2 \leq \mu$. The proximal operator is the projection of the vector to the ℓ_2 -ball of radius μ :

$$\text{prox}_{\ell_2 \leq \mu}(\mathbf{v}) = \begin{cases} \mathbf{v}, & \text{if } \|\mathbf{v}\|_2 \leq \mu \\ \frac{\mu\mathbf{v}}{\|\mathbf{v}\|_2}, & \text{otherwise.} \end{cases} \quad (23)$$

There are many other useful proximal operators which may be efficiently computed, even if they do not admit a closed-form solution [42].

APPENDIX D
WEIGHTED SDMM

ADMM belongs to the class of proximal algorithms [42] and given that all objectives and constraints are convex, ADMM is proven to converge under rather weak conditions [46]. In its canonical form (adopted from [20]), ADMM can be used to solve problems in the following form:

$$\underset{\mathbf{x}, \mathbf{z}}{\text{minimize}} F_1(\mathbf{x}) + F_2(\mathbf{z}) \text{ subject to } \mathbf{Ax} - \mathbf{Bz} = \mathbf{h}. \quad (24)$$

where $\mathbf{A} \in \mathbb{R}^{p \times n}$, $\mathbf{B} \in \mathbb{R}^{p \times m}$ and $\mathbf{h} \in \mathbb{R}^p$. To solve this problem, algorithm alternates the minimization with respect to each primal variable (\mathbf{x}, \mathbf{z}) and the dual ascent step:

$$\mathbf{x}^{(k+1)} = \arg \min_{\mathbf{x}} F_1(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{Ax} - \mathbf{Bz}^{(k)} - \mathbf{h} + \mathbf{u}^{(k)}\|_2^2 \quad (25)$$

$$\mathbf{z}^{(k+1)} = \arg \min_{\mathbf{z}} F_2(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{Bz} - \mathbf{Ax}^{(k+1)} + \mathbf{h} - \mathbf{u}^{(k)}\|_2^2 \quad (26)$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{Ax}^{(k+1)} - \mathbf{Bz}^{(k+1)} - \mathbf{h} \quad (27)$$

Here $\mathbf{x}^{(k)}$, $\mathbf{z}^{(k)}$ and $\mathbf{u}^{(k)}$ represent the estimate, auxiliary and scaled dual variable at the k^{th} iteration, respectively.

When the optimization problem is the sum of more than two functionals, it can be solved by standard (non-weighted) SDMM:

$$\underset{\mathbf{x}, \mathbf{z}_i}{\text{minimize}} \sum_{i=1}^f F_i(\mathbf{z}_i) \text{ subject to } \mathbf{H}_i \mathbf{x} - \mathbf{h}_i = \mathbf{z}_i. \quad (28)$$

It is easy to reformulate it as an ADMM problem (hence, its convergence is guaranteed through ADMM framework).

Define:

$$J_1(\mathbf{z}) = \sum_{i=1}^f F_i(\mathbf{z}_i), \text{ where } \mathbf{z} = \begin{bmatrix} \mathbf{z}_1^T & \mathbf{z}_2^T & \dots & \mathbf{z}_f^T \end{bmatrix}^T. \quad (29)$$

By choosing $J_2(\mathbf{x}) = 0$, we can express the canonical ADMM problem as follows:

$$\underset{\mathbf{x}, \mathbf{z}}{\text{minimize}} J_1(\mathbf{z}) + J_2(\mathbf{x}) \text{ subject to } \mathbf{Hx} - \mathbf{h} = \mathbf{z}, \quad (30)$$

where $\mathbf{H} = \begin{bmatrix} \mathbf{H}_1^T & \mathbf{H}_2^T & \dots & \mathbf{H}_f^T \end{bmatrix}^T$ and $\mathbf{h} = \begin{bmatrix} \mathbf{h}_1^T & \mathbf{h}_2^T & \dots & \mathbf{h}_f^T \end{bmatrix}^T$.

The iterates, given in (25), are now expressed as follows:

$$\begin{aligned} \mathbf{z}^{(k+1)} &= \text{prox}_{\frac{1}{\rho} J_1} \left(\mathbf{Hx}^{(k)} - \mathbf{h} + \mathbf{u}^{(k)} \right), \\ \mathbf{x}^{(k+1)} &= \arg \min_{\mathbf{x}} \frac{\rho}{2} \|\mathbf{Hx} - \mathbf{h} + \mathbf{u}^{(k)} - \mathbf{z}^{(k+1)}\|_2^2, \\ \mathbf{u}^{(k+1)} &= \mathbf{u}^{(k)} + \mathbf{Hx}^{(k+1)} - \mathbf{h} - \mathbf{z}^{(k+1)}. \end{aligned} \quad (31)$$

Since $J_1(\mathbf{z})$ (29) is block-separable, so is the proximal operator $\text{prox}_{\frac{1}{\rho}J_1}(\cdot)$ [42] (the least squares step and the \mathbf{u} -updates are trivially separable, too). Finally, we can recover previously defined SDMM iterates (16):

$$\begin{aligned} \mathbf{z}_i^{(k+1)} &= \text{prox}_{\frac{1}{\rho}F_i} \left(\mathbf{H}_i \mathbf{x}^{(k)} - \mathbf{h}_i + \mathbf{u}_i^{(k)} \right), \\ \mathbf{x}^{(k+1)} &= \arg \min_{\mathbf{x}} \sum_{i=1}^N \frac{\rho}{2} \|\mathbf{H}_i \mathbf{x} - \mathbf{h}_i + \mathbf{u}_i^{(k)} - \mathbf{z}_i^{(k+1)}\|_2^2, \\ \mathbf{u}_i^{(k+1)} &= \mathbf{u}_i^{(k)} + \mathbf{H}_i \mathbf{x}^{(k+1)} - \mathbf{h}_i - \mathbf{z}_i^{(k+1)}. \end{aligned} \quad (32)$$

Given this formulation, one may notice that the functionals F_i encode an objective and constraints. However, the least squares step treats all \mathbf{z}_i equally, meaning that \mathbf{x} is not guaranteed to satisfy the constraints. Moreover, in practice, \mathbf{x} is often far from being feasible.

To alleviate this problem, the natural solution is to set different weights to different blocks $[\mathbf{H}_i \mathbf{x} + \mathbf{u}_i^{(k)} - \mathbf{z}_i^{(k+1)}]$ of the sum of squares in (32). This weighting can be seen as choosing different SDMM multipliers ρ_i for different functionals $F_i(\cdot)$:

$$\begin{aligned} \mathbf{z}_i^{(k+1)} &= \text{prox}_{\frac{1}{\rho_i}F_i} \left(\mathbf{H}_i \mathbf{x}^{(k)} - \mathbf{h}_i + \mathbf{u}_i^{(k)} \right), \\ \mathbf{x}^{(k+1)} &= \arg \min_{\mathbf{x}} \sum_{i=1}^f \frac{\rho_i}{2} \|\mathbf{H}_i \mathbf{x} - \mathbf{h}_i + \mathbf{u}_i^{(k)} - \mathbf{z}_i^{(k+1)}\|_2^2, \\ \mathbf{u}_i^{(k+1)} &= \mathbf{u}_i^{(k)} + \mathbf{H}_i \mathbf{x}^{(k+1)} - \mathbf{h}_i - \mathbf{z}_i^{(k+1)}. \end{aligned} \quad (33)$$

To derive this, consider the following splitting (with $\rho = 1$):

$$\begin{aligned} &\underset{\mathbf{x}, \mathbf{z}_i}{\text{minimize}} \sum_{i=1}^f F_i \left(\frac{1}{\sqrt{\rho_i}} \mathbf{z}_i \right), \\ &\text{subject to } \sqrt{\rho_i} (\mathbf{H}_i \mathbf{x} - \mathbf{h}_i) = \mathbf{z}_i. \end{aligned} \quad (34)$$

Then, the SDMM iterates are defined as follows:

$$\begin{aligned} \mathbf{z}_i^{(k+1)} &= \arg \min_{\mathbf{z}_i} \left(F_i \left(\frac{1}{\sqrt{\rho_i}} \mathbf{z}_i \right) \right. \\ &\quad \left. + \frac{1}{2} \|\sqrt{\rho_i} (\mathbf{H}_i \mathbf{x}^{(k)} - \mathbf{h}_i) - \mathbf{z}_i + \mathbf{u}_i^{(k)}\|_2^2 \right), \\ \mathbf{x}^{(k+1)} &= \arg \min_{\mathbf{x}} \sum_{i=1}^f \frac{1}{2} \|\sqrt{\rho_i} (\mathbf{H}_i \mathbf{x} - \mathbf{h}_i) + \mathbf{u}_i^{(k)} - \mathbf{z}_i^{(k+1)}\|_2^2, \\ \mathbf{u}_i^{(k+1)} &= \mathbf{u}_i^{(k)} + \sqrt{\rho_i} (\mathbf{H}_i \mathbf{x}^{(k+1)} - \mathbf{h}_i) - \mathbf{z}_i^{(k+1)}. \end{aligned} \quad (35)$$

By abuse of notation $\mathbf{z}_i := \frac{1}{\sqrt{\rho_i}} \mathbf{z}_i$, $\mathbf{z}_i^{(k)} := \frac{1}{\sqrt{\rho_i}} \mathbf{z}_i^{(k)}$ and $\mathbf{u}_i^{(k)} := \frac{1}{\sqrt{\rho_i}} \mathbf{u}_i^{(k)}$, we arrive at the expression (33).

The downside of this approach is that conditioning of the weighted matrix \mathbf{H} is usually worse, which is why applying the standard conjugate gradient method to the normal equations $\mathbf{H}^\top \mathbf{H} \mathbf{x} = \mathbf{H}^\top (\mathbf{z}^{(k+1)} + \mathbf{h} - \mathbf{u}^{(k)})$ has to be avoided. Instead, we use the Least Squares Minimal Residual (LSMR) method [47], which is less sensitive to matrix conditioning. Assuming no a priori knowledge on the structure of \mathbf{H} , we use recommended diagonal (right) preconditioner, whose elements are reciprocal to the ℓ_2 -norms of the columns of \mathbf{H} . Even though there exist

more efficient preconditioners (such as incomplete Cholesky / LU factorizations), there are two advantages that this diagonal preconditioner provides: i), there are no issues with stability, as with the incomplete preconditioners, and ii), it can be efficiently computed in the function handle implementation of the synthesis problem (for which only \mathbf{MD} exists in the matrix form).

Following [20], the SDMM stopping criterion is based on primal and dual residuals $\mathbf{r}_{\text{prim}}^{(k)}$ and $\mathbf{r}_{\text{dual}}^{(k)}$, which are defined as:

$$\mathbf{r}_{\text{prim}}^{(k)} = \mathbf{H}\mathbf{x}^{(k)} - \mathbf{h} - \mathbf{z}^{(k)} \quad (36)$$

$$\mathbf{r}_{\text{dual}}^{(k)} = \mathbf{H} \left(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \right). \quad (37)$$

We stop iterating once their norms fall below the thresholds:

$$q_{\text{prim}}^{(k)} = \epsilon \min\{\|\mathbf{H}\mathbf{x}^{(k)} - \mathbf{h}\|_2, \|\mathbf{z}^{(k)}\|_2\} \quad (38)$$

$$q_{\text{dual}}^{(k)} = \epsilon \left\| \begin{bmatrix} \sqrt{\rho_1} \mathbf{u}_1^{(k)} \\ \vdots \\ \sqrt{\rho_f} \mathbf{u}_f^{(k)} \end{bmatrix} \right\|_2, \quad (39)$$

where ϵ denotes the relative accuracy.

ACKNOWLEDGMENT

This work was supported in part by the European Research Council, PLEASE project (ERC-StG-2011-277906).

The authors thank anonymous reviewers for their insightful remarks helping us to improve the quality of the paper.

REFERENCES

- [1] J. Huang, T. Supaongprapa, I. Terakura, F. Wang, N. Ohnishi, and N. Sugie, "A model-based sound localization system and its application to robot navigation," *Robotics and Autonomous Systems*, vol. 27, no. 4, pp. 199–209, 1999.
- [2] M. Cheney and B. Borden, "Problems in synthetic-aperture radar imaging," *Inverse Problems*, vol. 25, 2009.
- [3] A. Ribes and F. Schmitt, "Linear inverse problems in imaging," *IEEE Signal Processing Magazine*, vol. 25, pp. 84–99, 2008.
- [4] W. Munk, P. Worcester, and C. Wunsch, *Ocean acoustic tomography*, Cambridge University Press, 2009.
- [5] T. Szabo, *Diagnostic ultrasound imaging: inside out*, Academic Press, 2004.
- [6] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*, Springer, 2013.
- [7] A. Kirsch, *An introduction to the mathematical theory of inverse problems*, vol. 120, Springer, 2011.
- [8] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [9] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, "The cospase analysis model and algorithms," *Applied and Computational Harmonic Analysis*, vol. 34, no. 1, pp. 30–56, 2013.
- [10] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse problems*, vol. 23, no. 3, pp. 947, 2007.
- [11] I. Selesnick and M. Figueiredo, "Signal Restoration with Overcomplete Wavelet Transforms: Comparison of Analysis and Synthesis Priors," *In Proceedings of SPIE*, vol. 7446 (Wavelets XIII), 2009.
- [12] L. Albera, A. Ferréol, D. Cosandier-Rimélé, I. Merlet, and F. Wendling, "Brain source localization using a fourth-order deflation scheme," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 2, pp. 490–501, 2008.
- [13] A. Fannjiang, T. Strohmer, and P. Yan, "Compressed remote sensing of sparse objects," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 595–618, 2010.

- [14] A. Asaci, M. Golbabaee, H. Boulard, and V. Cevher, “Structured Sparsity Models for Multiparty Speech Recovery from Reverberant Recordings,” *arXiv*, 2012.
- [15] G. Chardon and L. Daudet, “Source localization in reverberant rooms using sparse modeling and narrowband measurements,” *arXiv preprint arXiv:1307.4894*, 2013.
- [16] I. Dokmanic and M. Vetterli, “Room helps: Acoustic localization with finite elements,” in *37th IEEE ICASSP*, Kyoto, Japan, 2012, pp. 2617–2620.
- [17] S. Nam and R. Gribonval, “Physics-driven structured cosparsity modeling for source localization,” in *37th IEEE ICASSP*, Kyoto, Japan, 2012, pp. 5397–5400.
- [18] N. Antonello, T. van Waterschoot, M. Moonen, and P. A. Naylor, “Source localization and signal reconstruction in a reverberant field using the fddd method,” in *22th EURASIP EUSIPCO*, Lisbon, Portugal, 2014.
- [19] S. Kitić, N. Bertin, and R. Gribonval, “Hearing behind walls: localizing sources in the room next door with cosparsity,” in *38th IEEE ICASSP*, Vancouver, Canada, 2013.
- [20] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [21] J. Chen, K. Yao, and R. E. Hudson, “Source localization and beamforming,” *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 30–39, 2002.
- [22] X. Sheng and Y. Hu, “Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. 53, no. 1, pp. 44–53, 2005.
- [23] S. Kitić, N. Bertin, and R. Gribonval, “A review of cosparsity signal recovery methods applied to sound source localization,” in *24th GRETSI*, Brest, France, 2013.
- [24] L. Albera, S. Kitić, N. Bertin, G. Puy, and R. Gribonval, “Brain source localization using a physics-driven structured cosparsity representation of eeg signals,” in *24th IEEE MLSP*, Reims, France, 2014, pp. 1–6.
- [25] L. Evans, “Partial differential equations (graduate studies in mathematics, vol. 19),” *Instructor*, p. 67, 2009.
- [26] S. Chen, D. Donoho, and M. Saunders, “Atomic decomposition by basis pursuit,” *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.
- [27] G. Evensen, *Data assimilation: the ensemble Kalman filter*, Springer Science & Business Media, 2009.
- [28] E. Hänsler and G. Schmidt, *Speech and audio processing in adverse environments*, Springer, 2008.
- [29] F. Asano, M. Goto, K. Itou, and H. Asoh, “Real-time sound source localization and separation system and its application to automatic speech recognition,” in *INTERSPEECH*, 2001, pp. 1013–1016.
- [30] B. Sklanka, J. Tuss, R. Buehrle, J. Klos, E. Williams, and N. Valdivia, “Acoustic source localization in aircraft interiors using microphone array technologies,” in *paper No. AIAA-2006-2714, 12th AIAA/CEAS Aeroacoustics Conference, Cambridge MA*, 2006.
- [31] M. Pollefeys and D. Nister, “Direct computation of sound and microphone locations from time-difference-of-arrival data,” in *33th IEEE ICASSP*, Las Vegas, USA, 2008, pp. 2445–2448.
- [32] H. Kuttruff, *Room acoustics*, CRC Press, 2009.
- [33] K. Kowalczyk and M. van Walstijn, “Formulation of locally reacting surfaces in FDTD/K-DWM modelling of acoustic spaces,” *Acta Acustica united with Acustica*, vol. 94, no. 6, pp. 891–906, 2008.
- [34] R. LeVeque, *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*, vol. 98, SIAM, 2007.
- [35] T. Piotrowski, D. Gutierrez, I. Yamada, and J. Zygierevicz, “Reduced-rank neural activity index for eeg/meg multi-source localization,” in *39th IEEE ICASSP*, Florence, Italy, 2014.
- [36] K. Uutela, M. Hämäläinen, and E. Somersalo, “Visualization of magnetoencephalographic data using minimum current estimates,” *NeuroImage*, vol. 10, no. 2, pp. 173–180, 1999.
- [37] M. Mohr and B. Vanrumste, “Comparing iterative solvers for linear systems associated with the finite difference discretisation of the forward problem in electro-encephalographic source analysis,” *Medical and Biological Engineering and Computing*, vol. 41, no. 1, pp. 75–84, 2003.
- [38] J. G. Witwer, G. J. Trezek, and D. L. Jewett, “The effect of media inhomogeneities upon intracranial electrical fields,” *IEEE Transactions on Biomedical Engineering*, vol. 5, pp. 352–362, 1972.
- [39] A. Chambolle and T. Pock, “A first-order primal-dual algorithm for convex problems with applications to imaging,” *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

- [40] P. L. Combettes and J.-C. Pesquet, “Primal-dual splitting algorithm for solving inclusions with mixtures of composite, lipschitzian, and parallel-sum type monotone operators,” *Set-Valued and variational analysis*, vol. 20, no. 2, pp. 307–330, 2012.
- [41] L. Condat, “A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms,” *Journal of Optimization Theory and Applications*, vol. 158, no. 2, pp. 460–479, 2013.
- [42] P. L. Combettes and J.-C. Pesquet, “Proximal splitting methods in signal processing,” in *Fixed-point algorithms for inverse problems in science and engineering*, pp. 185–212. Springer, 2011.
- [43] Å. Björck, *Numerical Methods in Matrix Computations*, vol. 59, Springer, 2014.
- [44] Y. Chen, T. A. Davis, W. Hager, and S. Rajamanickam, “Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 35, no. 3, pp. 22, 2008.
- [45] T. A. Davis, *Direct methods for sparse linear systems*, vol. 2, SIAM, 2006.
- [46] J. Eckstein and D. Bertsekas, “On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators,” *Mathematical Programming*, vol. 55, no. 1-3, pp. 293–318, 1992.
- [47] D. C. Fong and M. Saunders, “LSMR: An iterative algorithm for sparse least-squares problems,” *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2950–2971, 2011.
- [48] M. Fink, “Time-reversal acoustics in complex environments,” *Geophysics*, vol. 71, no. 4, pp. SI151–SI164, 2006.
- [49] M. Yannakakis, “Computing the minimum fill-in is np-complete,” *SIAM Journal on Algebraic Discrete Methods*, vol. 2, no. 1, pp. 77–79, 1981.
- [50] J. O. Smith, *Introduction to Digital Filters with Audio Applications*, W3K Publishing, 2007.
- [51] G. Golub and C. Van Loan, *Matrix computations*, vol. 3, JHU Press, 2012.
- [52] B. H. Jansen and V. G. Rit, “Electroencephalogram and visual evoked potential generation in a mathematical model of coupled cortical columns?,” *Biological Cybernetics*, vol. 73, no. 4, pp. 357–366, 1995.
- [53] H. Kuhn, “The Hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [54] S. Shalev-Shwartz and N. Srebro, “SVM optimization: inverse dependence on training set size,” in *ICML*. ACM, 2008, pp. 928–935.
- [55] J. Kaipio and E. Somersalo, *Statistical and computational inverse problems*, vol. 160, Springer, 2005.
- [56] Y. Saad, *Iterative methods for sparse linear systems*, SIAM, 2003.
- [57] J. Bruer, J. Tropp, V. Cevher, and S. Becker, “Time–data tradeoffs by aggressive smoothing,” in *NIPS Proceedings*, 2014, pp. 1664–1672.
- [58] C. Bilen, S. Kitić, N. Bertin, and R. Gribonval, “Sparse acoustic source localization with blind calibration for unknown medium characteristics,” in *ITWIST*, 2014.
- [59] J. Strikwerda, *Finite difference schemes and partial differential equations*, SIAM, 2004.
- [60] R. Jenatton, J.-Y. Audibert, and F. Bach, “Structured variable selection with sparsity-inducing norms,” *The Journal of Machine Learning Research*, vol. 12, pp. 2777–2824, 2011.